



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Ανίχνευση αντικειμένων και ημιαυτόματος χαρακτηρισμός εικόνων

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήστος Ι. Βαρυτιμίδης

Επιβλέπων : Στέφανος Κόλλιας  
Καθηγητής

Αθήνα, Ιούνιος 2008





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Ανίχνευση αντικειμένων και ημιαυτόματος χαρακτηρισμός εικόνων

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήστος Ι. Βαρυτιμίδης

Επιβλέπων : Στέφανος Κόλλιας  
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10<sup>η</sup> Ιουνίου 2008.

.....  
Στέφανος Κόλλιας  
Καθηγητής

.....  
Ανδρέας Σταφυλοπάτης  
Καθηγητής

.....  
Γεώργιος Στάμου  
Λέκτορας

Αθήνα, Ιούνιος 2008

.....

Χρήστος Ι. Βαρυτιμίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Χρήστος Ι. Βαρυτιμίδης, 2008.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Η ανίχνευση αντικειμένων σε εικόνες αποτελεί πεδίο της ανάλυσης εικόνων που ερευνάται έντονα τα τελευταία χρόνια. Σε αυτή τη διπλωματική παρουσιάζεται μία ολοκληρωμένη μέθοδος ανίχνευσης αντικειμένων σε εικόνες, η οποία δημιουργήθηκε από τους Viola και Jones το 2001. Για την περιγραφή των εικόνων χρησιμοποιούνται τα χαρακτηριστικά τύπου Haar, ενώ η ταξινόμηση των υποψήφιων περιοχών της εικόνας γίνεται με τον αλγόριθμο AdaBoost χρησιμοποιώντας διαδοχικά συνδεδεμένους ταξινομητές (ΔΣΤ) για αύξηση της ταχύτητας ανίχνευσης. Χρησιμοποιώντας αυτή τη μέθοδο, εκπαιδεύτηκαν διάφοροι ανιχνευτές για το εσωτερικό και το εξωτερικό μέρος αυτοκινήτων, με εικόνες από το σύνολο LabelMe κ.α. Παρουσιάζουμε και εξηγούμε τις επιλογές που έγιναν για την εκπαίδευση κάθε ανιχνευτή. Τα αποτελέσματα της αξιολόγησης κάθε ανιχνευτή παρουσιάζονται με διαγράμματα ακρίβειας-επανάκλησης και χαρακτηριστικής λειτουργίας δέκτη (ROC). Παρουσιάζονται επίσης συμπεράσματα που βοηθούν στην επίτευξη βέλτιστων αποτελεσμάτων με χρήση της συγκεκριμένης μεθόδου. Στα πλαίσια αυτής της διπλωματικής κατασκευάστηκε ένα πρόγραμμα ημιαυτόματου χαρακτηρισμού εικόνων, το οποίο ανιχνεύει αντικείμενα στην υπό χαρακτηρισμό εικόνα χρησιμοποιώντας τη μέθοδο που παρουσιάζουμε.

## Λέξεις κλειδιά

Ανίχνευση αντικειμένων, χαρακτηριστικά τύπου Haar, AdaBoost, διαδοχικά συνδεδεμένοι ταξινομητές, ταξινόμηση, LabelMe, ημιαυτόματος χαρακτηρισμός εικόνων.

## **Abstract**

Object detection in images is a field of image analysis that is searched intensively during the past few years. In this diploma thesis we present a complete object detection method which was created by Viola and Jones in 2001. Haar-like features are used to describe images, while the classification of the candidate regions of an image is performed by a cascade of classifiers created by the AdaBoost algorithm in order to increase detection speed. By using this method, we trained several detectors for interior parts of a car, as well as its exterior, with sample images from the LabelMe dataset. We show and explain the choices that were made in every detector training. The results of the evaluation of every detector are presented in precision-recall and receiver operator characteristic (ROC) diagrams. We also present some conclusions in order to achieve the best results from this method. In this diploma thesis we created a program for semi-automatic annotation of images, which detects objects in images using the presented method.

## **Key words**

Object detection, Haar-like features, AdaBoost, cascade of classifiers, classification, LabelMe, semi-automatic image annotation.

## Πρόλογος

Αυτή η διπλωματική εκπονήθηκε κατά το ακαδημαϊκό έτος 2007-2008 στο Εργαστήριο Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων (IVML) του Εθνικού Μετσόβιου Πολυτεχνείου. Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Στέφανο Κόλλια για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την εργασία αυτή. Ακόμα, θα ήθελα να ευχαριστήσω ιδιαίτερα τους Δρ. Ιωάννη Αβριθή και Δρ. Κωνσταντίνο Ραπαντζίκο για την ουσιαστική βοήθεια και καθοδήγηση που μου παρείχαν καθ' όλη τη διάρκεια της εκπόνησης αυτής της εργασίας.





# Πίνακας Περιεχομένων

<b>Κεφάλαιο 1. Εισαγωγή.....</b>	<b>15</b>
1.1 Ανίχνευση αντικειμένων .....	15
1.2 Συνεισφορά της διπλωματικής .....	18
1.3 Οργάνωση κειμένου .....	18
<b>Κεφάλαιο 2. Ανίχνευση αντικειμένων με χαρακτηριστικά τύπου Haar.....</b>	<b>21</b>
2.1 Χαρακτηριστικά τύπου Haar .....	21
2.2 Πίνακας Προστιθέμενου Εμβადού.....	24
2.3 Αλγόριθμος εκπαίδευσης AdaBoost.....	27
2.4 Διαδοχικά Συνδεδεμένος Ταξινομητής (ΔΣΤ) .....	29
<b>Κεφάλαιο 3. Πειραματική μεθοδολογία .....</b>	<b>33</b>
3.1 Προετοιμασία .....	33
3.2 Επιλογές εκπαίδευσης.....	36
3.3 Επιλογές εκτέλεσης .....	38
3.4 Κατάταξη αποτελεσμάτων .....	39
3.5 Αξιολόγηση ανιχνευτή.....	41
<b>Κεφάλαιο 4. Πειραματικά αποτελέσματα .....</b>	<b>45</b>
4.1 Εσωτερικό αυτοκινήτου.....	45
4.1.1 Τιμόνια αυτοκινήτων .....	45
4.1.2 Αεραγωγοί .....	48
4.1.3 Μοχλός ταχυτήτων.....	51
4.1.4 Πεντάλ .....	54
4.2 Σύνολο εικόνων LabelMe.....	57
4.2.1 Εκπαίδευση ανιχνευτών για αυτοκίνητα .....	58
4.2.2 Αξιολόγηση επιδόσεων ταξινομητή για το εμπρός και πίσω μέρος αυτοκινήτων .....	59
4.2.3 Αξιολόγηση επιδόσεων ταξινομητή για το πλαϊνό μέρος αυτοκινήτων.....	62
4.2.4 Αξιολόγηση επιδόσεων όλων των ταξινομητών για τα μέρη αυτοκινήτων .....	65
<b>Κεφάλαιο 5. Ημιαυτόματος χαρακτηρισμός εικόνων.....</b>	<b>67</b>
5.1 Εισαγωγή.....	67
5.2 Περιγραφή προγράμματος.....	69
5.2.1 Παράμετροι εισόδου.....	71
5.2.2 Λειτουργίες πληκτρολογίου.....	71
5.2.3 Λειτουργίες ποντικιού .....	73

5.2.4 Αρχεία χαρακτηρισμού εικόνων .....	74
<b>Κεφάλαιο 6. Συμπεράσματα και μελλοντικές επεκτάσεις .....</b>	<b>75</b>
6.1 Ανίχνευση αντικειμένων με χαρακτηριστικά τύπου Haar .....	75
6.2 Ημιαυτόματος χαρακτηρισμός εικόνων .....	77
<b>Παράρτημα Α. Ανίχνευση αντικειμένων με χρήση της OpenCV .....</b>	<b>79</b>
A.1 Πρόγραμμα προετοιμασίας θετικών παραδειγμάτων .....	79
A.2 Πρόγραμμα εκπαίδευσης ταξινομητή .....	80
A.3 Πρόγραμμα αξιολόγησης της απόδοσης του ανιχνευτή .....	84
A.4 Αλλαγές στο πρόγραμμα αξιολόγησης της απόδοσης του ανιχνευτή .....	86
<b>Βιβλιογραφία .....</b>	<b>89</b>



Σχήμα 4.14 - Καμπύλες ακρίβειας–επανάκλησης και ROC του ανιχνευτή εμπρός και πίσω μέρους αυτοκινήτων.....	60
Σχήμα 4.15 - Αποτελέσματα της ανίχνευσης για το εμπρός και πίσω μέρος αυτοκινήτων στο πρώτο σύνολο εικόνων δοκιμής. ....	61
Σχήμα 4.16 - Αποτελέσματα της ανίχνευσης για το εμπρός και πίσω μέρος αυτοκινήτων στο δεύτερο σύνολο εικόνων δοκιμής.....	62
Σχήμα 4.17 - Καμπύλες ακρίβειας–επανάκλησης και ROC του ανιχνευτή του πλαϊνού μέρους αυτοκινήτων. ....	63
Σχήμα 4.18 - Αποτελέσματα της ανίχνευσης του πλαϊνού μέρους αυτοκινήτων	64
Σχήμα 4.19 - Καμπύλες ακρίβειας–επανάκλησης και ROC του συνδυασμού των ανιχνευτών για αυτοκίνητα .....	65
Σχήμα 4.20 - Αποτελέσματα της ανίχνευσης αυτοκινήτων σε τυχαίες γωνίες. ..	66
Σχήμα 5.1 - Διεπαφή του προγράμματος χαρακτηρισμού εικόνων Annotator .....	70
Σχήμα 5.2 - Λειτουργίες πληκτρολογίου.....	73
Σχήμα 5.3 - Παράδειγμα αρχείου εξόδου προγράμματος χαρακτηρισμού εικόνων.....	74
Σχήμα 6.1 - Ομοιογένεια θετικών παραδειγμάτων. ....	76

## Κατάλογος Πινάκων

Πίνακας 2.1 - Πλήθος χαρακτηριστικών τύπου Haar σε παράθυρο 20x20 .....	23
Πίνακας 3.1 - Πίνακας Σύγχυσης (Confusion Matrix).....	41
Πίνακας 4.1 - Σύγκριση διάφορων ανιχνευτών για το πλαϊνό μέρος αυτοκινήτων. .....	64



# Κεφάλαιο 1.

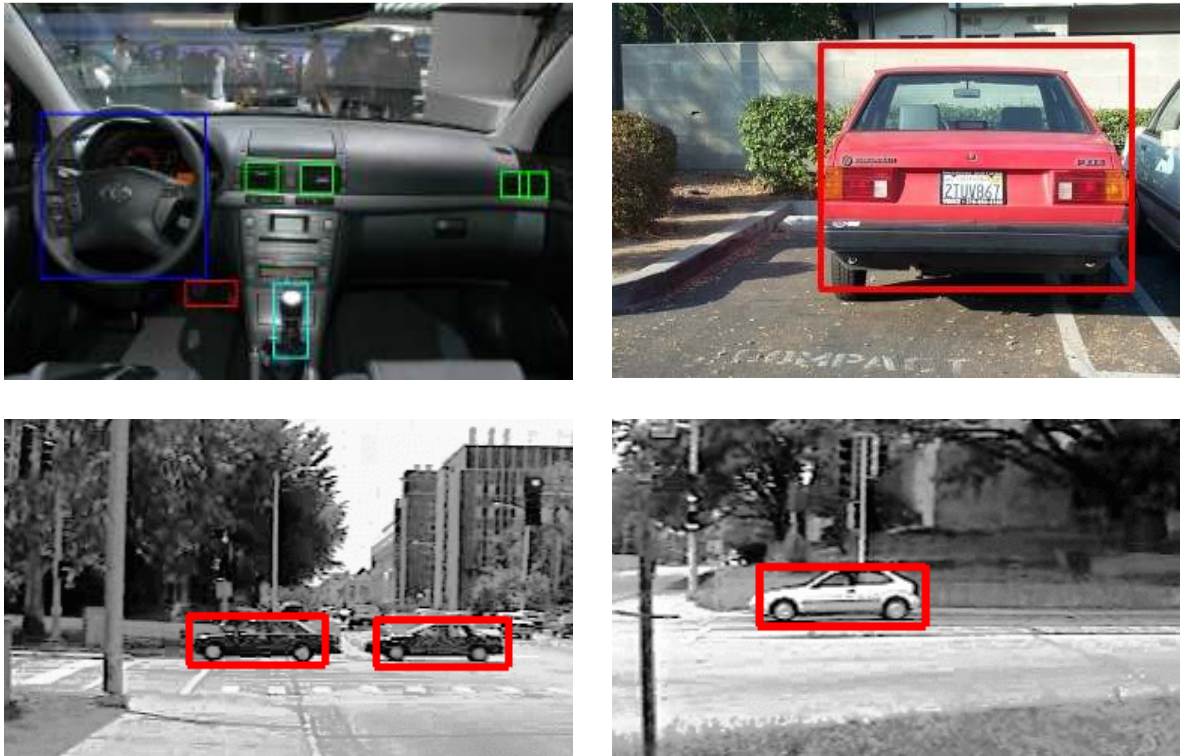
## Εισαγωγή

Η ανίχνευση αντικειμένων σε εικόνες είναι μία διαδικασία η οποία πραγματοποιείται από τον άνθρωπο πολλές φορές κάθε δευτερόλεπτο και για πάρα πολλά αντικείμενα. Τις περισσότερες φορές ο άνθρωπος δε χρειάζεται να προσπαθήσει για να αναγνωρίσει ένα αντικείμενο, ακόμα και αν δεν έχει δει ποτέ στο παρελθόν το συγκεκριμένο, αλλά κάποιο άλλο της ίδιας κατηγορίας. Έτσι, για παράδειγμα, καταλαβαίνουμε ότι τα "αντικείμενα" που βλέπουμε σε έναν δρόμο είναι άνθρωποι, παρό το γεγονός ότι μπορεί να μην έχουμε δει τους ίδιους ανθρώπους προηγουμένως ή το συγκεκριμένο συνδυασμό ρούχων που φοράνε. Επίσης, αναγνωρίζουμε ότι ένα αντικείμενο που βλέπουμε ανήκει στην κατηγορία αυτοκίνητο, ακόμα και αν είναι πολύ διαφορετικό από κάθε άλλο αυτοκίνητο που έχουμε δει μέχρι εκείνη τη στιγμή, δεν φαίνεται ολόκληρο, ή οι συνθήκες φωτισμού είναι κακές. Αντιλαμβανόμαστε, λοιπόν, ότι το ανθρώπινο σύστημα ανάλυσης εικόνων και αναγνώρισης αντικειμένων είναι πολύ ανεπτυγμένο.

Στη γενικότερη προσπάθεια αυτοματοποίησης κάποιων διαδικασιών, εντάσσεται και η ανίχνευση αντικειμένων σε εικόνες (κινούμενες ή στατικές) με τη χρήση ηλεκτρονικών υπολογιστών. Η διαδικασία αυτή μπορεί να φανεί χρήσιμη σε πολλούς τομείς, όπως τη ρομποτική, την πληροφορική κ.α. Για παράδειγμα, ένα σύστημα όρασης σε αυτοκίνητα που ανιχνεύει διάφορα αντικείμενα, όπως το οδόστρωμα, άλλα αυτοκίνητα, δέντρα και βράχους, μπορεί να βοηθήσει σημαντικά στη βελτίωση της ασφάλειας στην οδήγηση. Ακόμα, χρησιμοποιώντας τη διαδικασία ανίχνευσης, μπορούμε να αναζητήσουμε σε μία βάση εικόνων τις εμφανίσεις ενός συγκεκριμένου αντικειμένου, χωρίς να είναι απαραίτητη η προηγούμενη επεξεργασία τους από τον άνθρωπο. Τα τελευταία χρόνια έχει ενταθεί η έρευνα στον τομέα αυτό, που όμως εξακολουθεί να αποτελεί ένα ανοιχτό πρόβλημα της επιστημονικής περιοχής της ανάλυσης εικόνων και αναγνώρισης προτύπων.

### 1.1 Ανίχνευση αντικειμένων

Το πρόβλημα της ανίχνευσης αντικειμένων ορίζεται ως η διαδικασία κατά την οποία ένα υπολογιστικό σύστημα παίρνει ως είσοδο μια εικόνα και βγάζει ως έξοδο τις περιοχές της εικόνας στις οποίες εμφανίζεται κάποιο συγκεκριμένο αντικείμενο. Το αποτέλεσμα, δηλαδή, της διαδικασίας αποτελείται από την κατηγορία κάθε αντικειμένου που ανιχνεύθηκε καθώς και τη θέση και διάστασή του. Η ανίχνευση αντικειμένων σε εικόνες διαφοροποιείται από την ταξινόμηση εικόνων. Στην ταξινόμηση εικόνων, η έξοδος του υπολογιστικού συστήματος περιλαμβάνει μόνο την πληροφορία αν η εικόνα απεικονίζει ή όχι ένα συγκεκριμένο αντικείμενο και όχι τη θέση και τις διαστάσεις του αντικειμένου. Αντίθετα, στην ανίχνευση αντικειμένων, χρειάζεται να ελεγχθεί αν εμφανίζεται κάποιο αντικείμενο σε οποιαδήποτε θέση και με οποιοδήποτε μέγεθος μέσα σε μια εικόνα. Βλέπουμε, λοιπόν, ότι η ανίχνευση αντικειμένων είναι μια διαδικασία αρκετά σύνθετη και πιο πολύπλοκη από τη διαδικασία της ταξινόμησης εικόνων. Στο Σχήμα 1.1 βλέπουμε τα αποτελέσματα της διαδικασίας ανίχνευσης για διάφορες κατηγορίες αντικειμένων.



Σχήμα 1.1 - Παραδείγματα αποτελεσμάτων της διαδικασίας ανίχνευσης για διάφορες κατηγορίες αντικειμένων.

Μια εικόνα αποτελείται από ένα σύνολο εικονοστοιχείων, που το καθένα παίρνει μια τιμή που αντιστοιχεί στη φωτεινότητα ή στην τιμή του χρώματος στη συγκεκριμένη θέση της εικόνας. Αναπαριστούμε, δηλαδή, μια εικόνα με μια δισδιάστατη δομή από στοιχεία που παίρνουν διακριτές τιμές, άρα μπορούμε να τη θεωρήσουμε ως ένα ψηφιακό σήμα δύο διαστάσεων. Έτσι, μπορούμε να χρησιμοποιήσουμε διάφορα εργαλεία της θεωρίας σημάτων, ώστε να αντλήσουμε πληροφορίες που περιέχονται σε μια εικόνα. Μεταξύ άλλων, έχουν χρησιμοποιηθεί οι παρακάτω τεχνικές: α) ο μετασχηματισμός Hough [DuHa72], β) ο μετασχηματισμός SIFT (Scale-Invariant Feature Transform – Μετασχηματισμός Χαρακτηριστικών Ανεξάρτητων Κλίμακας) [Lowe04], γ) ο μετασχηματισμός Gabor, δ) τα ιστογράμματα HOG (Histogrammes of Oriented Gradients – Ιστογράμματα Κατευθύνσεων Κλίσης) [DaTr05], ε) ιστογράμματα προσανατολισμού ακμών (Edge Orientation Histograms – EOH) [LeWe04], στ) ιστογράμματα έντασης (Histogram of Intensities), ζ) η ανίχνευση περιγράμματος [ShBC05], η) ο μετασχηματισμός κυματιδίων με συναρτήσεις βάσης τύπου Haar (χαρακτηριστικά τύπου Haar) [OPS+97] και θ) η ανάλυση ανεξάρτητων συνιστωσών (Independent Component Analysis – ICA).

Η διαδικασία της ανίχνευσης ενός αντικειμένου σε μια εικόνα είναι εφαρμογή της θεωρίας αναγνώρισης προτύπων. Μετά την περιγραφή του προτύπου (εικόνας ή τμήματος εικόνας) με κάποια από τις μεθόδους που περιγράψαμε, θα πρέπει να ελεγχθεί αν το πρότυπο ανήκει στην κατηγορία που μας ενδιαφέρει. Για τον έλεγχο αυτό, που ονομάζεται ταξινόμηση, μπορούν να χρησιμοποιηθούν διάφοροι αλγόριθμοι, όπως: α) τα νευρωνικά δίκτυα, β) τα SVM (Support Vector Machine – Μηχανή Διανυσμάτων Στήριξης) [Burg98], γ) ο αλγόριθμος Winnow και δ) αλγόριθμοι ενδυνάμωσης (boosting), όπως ο αλγόριθμος AdaBoost [FrSc95].



Στη βιβλιογραφία βρίσκουμε διάφορες μεθόδους για ανίχνευση αντικειμένων σε εικόνες. Οι Agarwal και Roth το 2002 παρουσίασαν έναν ανιχνευτή στηριζόμενο σε τοπικά χαρακτηριστικά της εικόνας που εντοπίζονται με χρήση ενός ανιχνευτή περιοχών ενδιαφέροντος και περιγράφονται με ένα λεξιλόγιο τμημάτων (vocabulary of parts) της εικόνας [AgRo02]. Χρησιμοποιώντας τον αλγόριθμο Winnow για την ταξινόμηση, πέτυχαν καλά αποτελέσματα σε εικόνες με αυτοκίνητα από την πλάγια όψη τους. Το 2004 οι ίδιοι συγγραφείς (Agarwal et al.) παρουσίασαν μια σειρά πειραμάτων και άλλαξαν τον τη μέθοδο αξιολόγησης που χρησιμοποιούσαν, παίρνοντας ελαφρώς χειρότερα αποτελέσματα [AgAR04]. Οι Garg et al. το 2002 χρησιμοποίησαν ένα συνδυασμό τοπικών και γενικών πληροφοριών από εικόνες [GaAH02]. Δημιούργησαν ουσιαστικά δύο ανεξάρτητους ανιχνευτές, έναν βασιζόμενο σε αυτόν του [AgRo02] και έναν στηριζόμενο σε γενικότερα χαρακτηριστικά της εικόνας χρησιμοποιώντας ανάλυση ανεξάρτητων συνιστωσών. Τα αποτελέσματα των δύο ανιχνευτών συνδυάστηκαν, δίνοντας καλύτερα αποτελέσματα από το [GaAH02]. Οι Fergus et al. το 2003 χρησιμοποίησαν έναν ανιχνευτή περιοχών ενδιαφέροντος σε συνδυασμό με ιστογράμματα έντασης, επιλέγοντας έτσι τμήματα της εικόνας τα οποία αναπαριστούνται με τη θέση και την κλίμακά τους [FePZ03]. Έπειτα υπολογίζεται η πιθανότητα εμφάνισης κάθε τμήματος της εικόνας σε μια κατηγορία αντικειμένων, δεδομένης της θέσης και της κλίμακας του. Κατά την ανίχνευση εξετάζεται αν εμφανίζονται τα συστατικά μέρη του αντικείμενου. Το 2004 οι Leibe et al. συνδύασαν την ανίχνευση αντικειμένων με την τμηματοποίηση [LeLS04]. Μαθαίνοντας ποιες δομές μπορεί να εμφανιστούν σε ένα συγκεκριμένο αντικείμενο και πού, δημιουργούν ένα αφηρημένο μοντέλο σχήματος (Implicit Shape Model – ISM) και οδηγούνται στην ανίχνευση του αντικείμενου. Με τον τρόπο αυτό μπορούν να ανιχνευθούν και αντικείμενα με επικαλύψεις, καθώς το μοντέλο δεν θεωρεί ότι πρέπει να εμφανίζονται όλα τα μέρη ενός αντικείμενου για τον εντοπισμό του. Από το αρχικό αποτέλεσμα της ανίχνευσης δημιουργείται μια τμηματοποίηση της εικόνας, η οποία έπειτα βελτιστοποιεί τα αποτελέσματα της ανίχνευσης. Το 2008 οι ίδιοι συγγραφείς παρουσίασαν εκτεταμένα πειράματα πάνω σε αυτή τη μέθοδο με πολύ καλά αποτελέσματα σε διάφορα σύνολα εικόνων [LeLS08]. Το 2005 οι Fritz et al. συνδύασαν το αφηρημένο μοντέλο σχήματος από το [LeLS04] με έναν ταξινομητή SVM για να μειώσουν τις λανθασμένες θετικές ταξινομήσεις του πρώτου. Οι Mutch και Lowe το 2006 χρησιμοποίησαν στη μέθοδό τους τον μετασχηματισμό Gabor σε κάθε δυνατή θέση και κλίμακα [MuLo06]. Έπειτα, επιλέγεται μια αραιή αναπαράσταση από τις συναρτήσεις βάσης και χρησιμοποιώντας έναν ταξινομητή SVM οδηγούνται στην ανίχνευση αντικειμένων με πολύ καλά αποτελέσματα.

Σε αυτή τη διπλωματική ασχοληθήκαμε με την ανίχνευση αντικειμένων με χρήση των χαρακτηριστικών τύπου Haar. Τα χαρακτηριστικά τύπου Haar χρησιμοποιήθηκαν πρώτη φορά για την ανίχνευση ανθρώπων στο [OPS+97]. Έπειτα παρουσιάστηκε ένα γενικότερο πλαίσιο για ανίχνευση αντικειμένων από τους Papageorgiou et al., με το οποίο ανιχνεύθηκαν άνθρωποι και πρόσωπα [PaOP98]. Το 2001 οι Viola και Jones επέκτειναν τη μέθοδο, προσθέτοντάς της διάφορα χαρακτηριστικά [ViJo01]. Η συνεισφορά τους είναι πολύ μεγάλη, καθώς πέρα από τα επιπλέον χαρακτηριστικά τύπου Haar που πρόσθεσαν, εισήγαγαν τη χρήση πινάκων για το γρήγορο υπολογισμό των χαρακτηριστικών (όπως θα δούμε στην ενότητα 2.2) και τη χρήση του αλγόριθμου AdaBoost για την ταξινόμηση. Δημιούργησαν έτσι, ένα ολοκληρωμένο σύστημα ανίχνευσης το οποίο είχε μεγάλη απόδοση και λειτουργούσε σε πραγματικό χρόνο. Το 2002 οι Lienhart και Maydt επέκτειναν το σύστημα ανίχνευσης των Viola και Jones προσθέτοντας ένα σύνολο περιστραμμένων χαρακτηριστικών τύπου Haar, όπως θα δούμε στην ενότητα 2.1 [LiMa02].

Η υλοποίηση του συστήματος αυτού ενσωματώθηκε στη βιβλιοθήκη λογισμικού OpenCV και είναι διαθέσιμη ελεύθερα ως λογισμικό ανοιχτού κώδικα [OpenCV].

Η μέθοδος των χαρακτηριστικών τύπου Haar επιλέχθηκε ανάμεσα στις άλλες μεθόδους που παρουσιάστηκαν, καθώς παρουσιάζει πολύ καλά αποτελέσματα ανίχνευσης απαιτώντας πολύ λίγο χρόνο για την ολοκλήρωση της διαδικασίας. Για τις περισσότερες διαθέσιμες μεθόδους ανίχνευσης ξέρουμε τις επιδόσεις ως προς τα αποτελέσματα ανίχνευσης, αλλά δεν αναφέρεται ο χρόνος που απαιτεί η διαδικασία, καθώς συνήθως αυτός είναι πολύ αυξημένος. Η μέθοδος των Viola και Jones απαιτεί πολύ λίγο χρόνο για την ανίχνευση (όπως θα επαληθεύσουμε στο Κεφάλαιο 4 από τα πειράματά μας) και μπορεί να χρησιμοποιηθεί και σε συστήματα ανίχνευσης αντικειμένων πραγματικού χρόνου.

## 1.2 Συνεισφορά της διπλωματικής

Σε αυτή τη διπλωματική εργασία ασχοληθήκαμε με δύο αντικείμενα\* την ανίχνευση αντικειμένων και τον ημιαυτόματο χαρακτηρισμό εικόνων. Για την ανίχνευση αντικειμένων χρησιμοποιήσαμε τη μέθοδο των χαρακτηριστικών τύπου Haar. Εκπαιδεύσαμε ανιχνευτές για διάφορα αντικείμενα, όπως θα δούμε στο Κεφάλαιο 4, σύμφωνα με τη μεθοδολογία που περιγράφουμε στο Κεφάλαιο 3. Επιλέξαμε κάθε παράμετρο της εκπαίδευσης των ανιχνευτών ανάλογα με τις απαιτήσεις κάθε περίπτωσης, αλλάζοντας την αρχιτεκτονική και τα συστατικά μέρη των ανιχνευτών.

Η απόδοση κάθε ανιχνευτή που εκπαιδεύσαμε ελέγχθηκε με πειράματα που παρουσιάζονται αναλυτικά στο Κεφάλαιο 4. Για τα πειράματα αυτά χρησιμοποιήσαμε διάφορα σύνολα εικόνων που χρησιμοποιούνται ευρέως στη βιβλιογραφία. Πειραματιστήκαμε με τον τρόπο κατάταξης των αποτελεσμάτων της ανίχνευσης για την αξιολόγηση των ανιχνευτών, όπως θα δούμε στις ενότητες 3.5 και Α.4. Επίσης, συγκρίναμε την απόδοση των ανιχνευτών μας με τα αποτελέσματα των πιο γνωστών μεθόδων που έχουν εφαρμοστεί στα ίδια σύνολα εικόνων.

Για τις ανάγκες της διαδικασίας εκπαίδευσης των ανιχνευτών κατασκευάσαμε ένα πρόγραμμα χαρακτηρισμού εικόνων. Το πρόγραμμα κατασκευάστηκε με γνώμονα την ευχρηστία και την ταχύτητα στη διεκπεραίωση της διαδικασίας χαρακτηρισμού. Τα πολύ καλά αποτελέσματα των ανιχνευτών μας οδήγησαν στην ενσωμάτωση της διαδικασίας ανίχνευσης στο πρόγραμμα χαρακτηρισμού εικόνων. Έτσι, δημιουργήσαμε ένα πρόγραμμα ημιαυτόματου χαρακτηρισμού εικόνων με πολύ καλή απόδοση και μεγάλη ευχρηστία.

## 1.3 Οργάνωση κειμένου

Η διπλωματική αυτή χωρίζεται σε πέντε κεφάλαια. Το πρώτο κεφάλαιο αποτελείται από την παρούσα εισαγωγή. Στο δεύτερο κεφάλαιο αναλύεται η μέθοδος ανίχνευσης αντικειμένων των Viola και Jones που χρησιμοποιήσαμε και περιγράφεται κάθε συστατικό μέρος της μεθόδου. Στο τρίτο κεφάλαιο παρουσιάζεται η πειραματική μεθοδολογία που ακολουθήσαμε για να

εκπαιδεύσουμε συστήματα ανίχνευσης διάφορων αντικειμένων. Στο τέταρτο κεφάλαιο περιγράφουμε όλες τις επιλογές που έγιναν για την εκπαίδευση κάθε συστήματος ανίχνευσης, καθώς και τα αποτελέσματα της αξιολόγησης καθενός από αυτά. Στο πέμπτο κεφάλαιο περιγράφουμε το πρόγραμμα που δημιουργήσαμε για τον ημιαυτόματο χαρακτηρισμό εικόνων. Τέλος, στο παράρτημα περιγράφεται η διαδικασία εκπαίδευσης ενός ταξινομητή με τα προγράμματα της βιβλιοθήκης OpenCV.



## Κεφάλαιο 2.

### Ανίχνευση αντικειμένων με χαρακτηριστικά τύπου Haar

Η διαδικασία της ανίχνευσης αντικειμένων με χαρακτηριστικά τύπου Haar βασίζεται σε τέσσερα στοιχεία. Το πρώτο και βασικότερο είναι η χρήση των χαρακτηριστικών Haar για την περιγραφή των αντικειμένων. Πέρα από τις αρχικές συναρτήσεις Haar, χρησιμοποιούνται και άλλες του ίδιου τύπου όπως θα δούμε στην ενότητα 2.1. Δεύτερο βασικό στοιχείο της διαδικασίας είναι η χρήση δύο πινάκων για τον γρήγορο υπολογισμό των χαρακτηριστικών, που παρουσιάζεται στην ενότητα 2.2. Τρίτο στοιχείο αποτελεί ο αλγόριθμος εκπαίδευσης AdaBoost που χρησιμοποιείται για την εκπαίδευση του ανιχνευτή και περιγράφεται αναλυτικά στην ενότητα 2.3. Τελευταίο στοιχείο της διαδικασίας είναι η χρήση πολλών μικρών και γρήγορων διαδοχικά συνδεδεμένων ταξινομητών, αντί ενός μεγάλου και αργού ταξινομητή. Με την τεχνική αυτή, που παρουσιάζεται στην ενότητα 2.4, αυξάνεται πολύ η ταχύτητα της διαδικασίας ανίχνευσης.

#### 2.1 Χαρακτηριστικά τύπου Haar

Για να ανιχνεύσουμε αντικείμενα σε εικόνες απαιτείται κατάλληλη επεξεργασία και αναπαράσταση του περιεχομένου τους. Για την αναπαράσταση του περιεχομένου της εικόνας στη μέθοδο που εξετάζουμε, χρησιμοποιούμε τα χαρακτηριστικά τύπου Haar, τα οποία προκύπτουν από την εφαρμογή του μετασχηματισμού κυματιδίων σε μια εικόνα με χρήση των συναρτήσεων τύπου Haar [ViJo01].

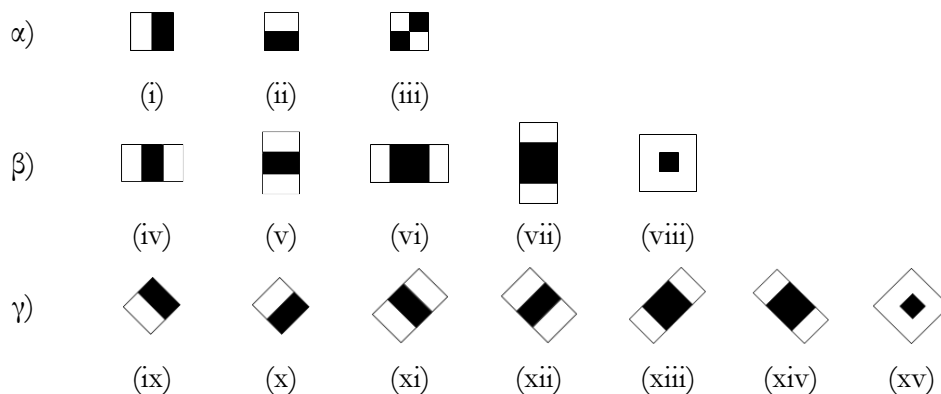
Η χρησιμοποίηση των συναρτήσεων Haar στο μετασχηματισμό κυματιδίων ξεκινά από την παρατήρηση ότι η τιμή της φωτεινότητας κάθε εικονοστοιχείου επηρεάζεται έντονα από τις αλλαγές στο φωτισμό της σκηνής [OPS+97]. Αυτή η αλλαγή όμως, επηρεάζει αρκετά ομοιόμορφα όλα τα εικονοστοιχεία της εικόνας. Έτσι, η τιμή μιας συνάρτησης που εξετάζει τη μέση διαφορά ανάμεσα σε δύο ή τρεις περιοχές της ίδιας εικόνας, θα παραμένει σε μεγάλο βαθμό ανεπηρέαστη. Χρησιμοποιώντας, λοιπόν, τις συναρτήσεις Haar, η διαδικασία της ανίχνευσης αντικειμένων δε θα επηρεάζεται από τις διαφορές στη φωτεινότητα από εικόνα σε εικόνα.

Οι συναρτήσεις Haar υπολογίζουν τη διαφορά ανάμεσα στους μέσους όρους των τιμών των εικονοστοιχείων δύο (ή τριών) περιοχών. Ας θεωρήσουμε τη συνάρτηση Haar που παριστάνεται με το ορθογώνιο  $i$  από το Σχήμα 2.1. Υπολογίζεται ο μέσος όρος των εικονοστοιχείων που βρίσκονται μέσα στο άσπρο ορθογώνιο, καθώς και αυτών που βρίσκονται μέσα στο μαύρο ορθογώνιο. Έπειτα, ο μέσος όρος του μαύρου ορθογωνίου αφαιρείται από τον μέσο όρο του άσπρου. Η τιμή που προκύπτει αποτελεί την τιμή του Haar χαρακτηριστικού.

Εφαρμόζοντας τον μετασχηματισμό κυματιδίων με τη συναρτησιακή βάση Haar, προκύπτει ένας περιορισμένος αριθμός χαρακτηριστικών [OPS+97]. Στο μονοδιάστατο μετασχηματισμό, η απόσταση ανάμεσα σε δύο γειτονικά κυματίδια, σε επίπεδο  $n$ , θα είναι  $2^n$ . Η απόσταση αυτή είναι πολύ μεγάλη, κι έτσι δεν λαμβάνουμε όσες πληροφορίες θέλουμε από μια εικόνα ώστε να την

περιγράψουμε λεπτομερώς. Για να έχουμε, λοιπόν, μια πιο λεπτομερή, χωρικά, αναπαράσταση του περιεχομένου της εικόνας χρειαζόμαστε ένα σύνολο από πλεονάζουσες συναρτήσεις βάσης. Για να το πετύχουμε αυτό, εφαρμόζουμε τις συναρτήσεις Haar με μεταξύ τους απόσταση ένα εικονοστοιχείο κάθε φορά. Έτσι, θα έχουμε μια πολύ πιο πυκνή αναπαράσταση. Επίσης, στο μετασχηματισμό κυματιδίων, το μέγεθος των συναρτήσεων Haar, κανονικά διπλασιάζεται σε κάθε επανάληψη. Για να αυξήσουμε ακόμα περισσότερο την λαμβανόμενη πληροφορία από την εικόνα, ορίζουμε ότι το μέγεθος των συναρτήσεων Haar θα αυξάνει κάθε φορά κατά ένα μόνο εικονοστοιχείο. Έτσι, το σύνολο των χαρακτηριστικών Haar σε μία εικόνα γίνεται υπερπολλαπλάσιο του αρχικού. Αυξάνουμε, δηλαδή, την ποσότητα της πληροφορίας που αντλούμε από μια εικόνα, αυξάνοντας τα χαρακτηριστικά τύπου Haar που θα υπολογιστούν σε αυτήν.

Τα κλασσικά Haar χαρακτηριστικά φαίνονται στο Σχήμα 2.1α [OPS+97]. Είναι σχετικά απλά και μπορούν να εντοπίσουν ακμές οριζόντια και κατακόρυφα καθώς και διαγώνιες γραμμές. Για να μπορέσουμε να αναπαραστήσουμε γραμμές, ράβδους και τετράγωνα καλύτερα, προσθέτουμε τα χαρακτηριστικά που φαίνονται στο Σχήμα 2.1β (τα χαρακτηριστικά  $iv$  και  $v$  εμφανίζονται στο [Vi]o01], ενώ τα υπόλοιπα στο [LiMa02]), τα οποία υπολογίζονται χωρίς να αυξάνεται ιδιαίτερα η πολυπλοκότητα, όπως θα δούμε στην ενότητα 2.2. Μια μεγάλη προσθήκη είναι τα χαρακτηριστικά που είναι περιστραμμένα κατά  $45^\circ$  και φαίνονται στο Σχήμα 2.1γ [LiMa02]. Με τη χρήση αυτών βελτιώνεται σημαντικά η αναπαράσταση των διαγώνιων σχημάτων. Με την προσθήκη όλων αυτών των χαρακτηριστικών, το σύνολο γίνεται υπερπλήρες και αναπαριστά πολύ καλύτερα την πληροφορία που περιέχεται σε μία εικόνα.



Σχήμα 2.1 - Χαρακτηριστικά τύπου Haar. α) Κλασσικά Haar χαρακτηριστικά, β) Χαρακτηριστικά για γραμμές και ράβδους, γ) Χαρακτηριστικά περιστραμμένα κατά  $45^\circ$
















Για τον υπολογισμό του πλήθους των Haar χαρακτηριστικών σε κάποιο παράθυρο εικόνας πλάτους  $W$  και ύψους  $H$ , ακολουθούμε την παρακάτω διαδικασία [LiMa02]. Έστω ότι  $w$  και  $h$  είναι το πλάτος και ύψος του ορθογωνίου της συνάρτησης Haar που εξετάζουμε. Το μέγεθος του ορθογωνίου θα αυξάνεται κατά ένα σε κάθε βήμα. Άρα, οι μέγιστοι συντελεστές μεγέθυνσης των ορθογωνίων σε πλάτος και ύψος θα είναι  $X = \left\lfloor \frac{W}{w} \right\rfloor$  και  $Y = \left\lfloor \frac{H}{h} \right\rfloor$ , αντίστοιχα. Το πλήθος των χαρακτηριστικών που προκύπτουν από την εφαρμογή ενός κατακόρυφου Haar χαρακτηριστικού στο παράθυρο εικόνας, είναι:

$$XY \cdot \left( W + 1 - w \frac{X+1}{2} \right) \cdot \left( H + 1 - h \frac{Y+1}{2} \right),$$

ενώ από την εφαρμογή ενός περιστραμμένου κατά  $45^0$  Haar χαρακτηριστικού:

$$XY \cdot \left( W + 1 - z \frac{X+1}{2} \right) \cdot \left( H + 1 - z \frac{Y+1}{2} \right), \text{ όπου } z = w + h.$$

Εφαρμόζοντας τους παραπάνω τύπους σε ένα παράθυρο εικόνας διαστάσεων  $W = 20$  και  $H = 20$ , βρίσκουμε ότι το πλήθος των χαρακτηριστικών είναι αυτό που φαίνεται στον Πίνακα 2.1.

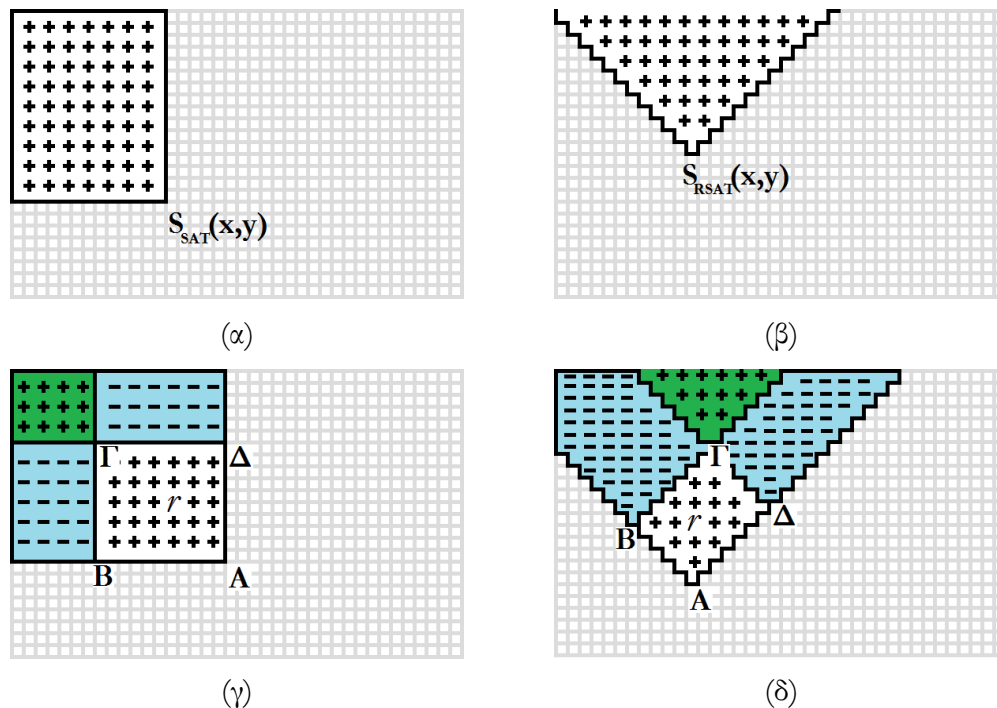
τύπος χαρακτηριστικού	w	h	X	Y	χαρακτηριστικά
	2	1	10	20	21.000
	1	2	20	10	21.000
	2	2	10	10	10.000
	3	1	6	20	13.230
	1	3	20	6	13.230
	4	1	5	20	9.450
	1	4	20	5	9.450
	3	3	6	6	3.969
	1	2	6	6	3.969
	2	1	6	6	3.969
	3	1	5	5	2.025
	1	3	5	5	2.025
	4	1	4	4	1.156
	1	4	4	4	1.156
	3	3	3	3	729
Σύνολο					116.358

Πίνακας 2.1 - Πλήθος χαρακτηριστικών τύπου Haar σε παράθυρο 20x20

Βλέπουμε, λοιπόν, ότι ενώ ένα παράθυρο εικόνας με διαστάσεις 20x20 έχει 400 εικονοστοιχεία, το πλήθος των χαρακτηριστικών που προκύπτουν είναι υπερ-πολλαπλάσιο αυτού. Αυτό οφείλεται στο γεγονός ότι το νέο σύνολο Haar χαρακτηριστικών είναι υπερπλήρες. Παρά τον πολύ μεγάλο αριθμό τους, τα Haar χαρακτηριστικά μπορούν να υπολογιστούν πολύ γρήγορα με τη βοήθεια του Πίνακα Προστιθέμενου Εμβαδού.

## 2.2 Πίνακας Προστιθέμενου Εμβαδού

Όπως είδαμε στην ενότητα 2.1, και φαίνεται από τον Πίνακα 2.1, ο αριθμός των Haar χαρακτηριστικών είναι πολύ μεγάλος (περίπου 120.000 για μία εικόνα διαστάσεων 20x20). Για να υπολογιστεί κάθε χαρακτηριστικό, ο προφανής τρόπος είναι να προσθέσουμε τις τιμές όλων των εικονοστοιχείων που περιλαμβάνει κάθε ορθογώνιο. Το σύνολο αυτών των πράξεων είναι πολύ χρονοβόρο, και θα καθιστούσε τη χρήση των χαρακτηριστικών τύπου Haar, δεδομένου του μεγάλου πλήθους τους, απαγορευτική. Μπορούμε όμως, να εκμεταλλευτούμε τον Πίνακα Προστιθέμενου Εμβαδού (Summed Area Table –  $S_{SAT}$ ) [ViJo01], καθώς και τον Περιστραμμένο Πίνακα Προστιθέμενου Εμβαδού (Rotated Summed Area Table –  $S_{RSAT}$ ) [LiMa02], [LiKP03], οι οποίοι επιτρέπουν τον υπολογισμό των χαρακτηριστικών τύπου Haar σε σταθερό χρόνο.



Σχήμα 2.2 - Οι πίνακες που βοηθούν στο γρήγορο υπολογισμό των χαρακτηριστικών.

α) Πίνακας Προστιθέμενου Εμβαδού  $S_{SAT}$ , β) Περιστραμμένος Πίνακας Προστιθέμενου Εμβαδού  $S_{RSAT}$ , γ) Υπολογισμός κατακόρυφου ορθογωνίου, δ) Υπολογισμός περιστραμμένου ορθογωνίου



Ο Πίνακας Προστιθέμενου Εμβαδού ( $S_{SAT}$ ) είναι ένας πίνακας ιδίων διαστάσεων με την αρχική εικόνα, που χρησιμεύει στο γρήγορο υπολογισμό όλων των κατακόρυφων χαρακτηριστικών [Vi01]. Ο  $S_{SAT}$  στη θέση  $(x, y)$  περιλαμβάνει το άθροισμα των τιμών όλων των εικονοστοιχείων που βρίσκονται επάνω και αριστερά του  $(x, y)$ , συμπεριλαμβανομένου και αυτού (βλ. Σχήμα 2.2α). Ισχύει δηλαδή:

$$S_{SAT}(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

όπου  $i(x', y')$  είναι η τιμή του εικονοστοιχείου στη θέση  $(x', y')$ . Για να κατασκευάσουμε όλο τον  $S_{SAT}$  αρκεί ένα μόνο πέρασμα από όλα τα στοιχεία της εικόνας, από τα αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Ο υπολογισμός γίνεται με τη χρήση της σχέσης:

$$S_{SAT}(x, y) = S_{SAT}(x, y-1) + S_{SAT}(x-1, y) - S_{SAT}(x-1, y-1) + i(x, y),$$

με χρήση των αρχικών τιμών  $S_{SAT}(-1, y) = S_{SAT}(x, -1) = 0$ .

Με τη βοήθεια του  $S_{SAT}$  μπορούμε να υπολογίσουμε το άθροισμα των τιμών των εικονοστοιχείων μέσα σε οποιοδήποτε ορθογώνιο με μόλις τέσσερις αναφορές στον  $S_{SAT}$ . Για να υπολογίσουμε το άθροισμα των τιμών των εικονοστοιχείων που βρίσκονται μέσα στο ορθογώνιο  $r$  στο Σχήμα 2.2γ (που ορίζεται από τα σημεία  $A(x_A, y_A)$ ,  $B(x_B, y_B)$ ,  $\Gamma(x_\Gamma, y_\Gamma)$  και  $\Delta(x_\Delta, y_\Delta)$ ), αρκεί να εκτελέσουμε τις παρακάτω πράξεις:

$$RecSum(r) = S_{SAT}(x_A, y_A) - S_{SAT}(x_B, y_B) - S_{SAT}(x_\Delta, y_\Delta) + S_{SAT}(x_\Gamma, y_\Gamma),$$

όπου  $RecSum(r)$  είναι το άθροισμα των τιμών των εικονοστοιχείων εντός του ορθογώνιου  $r$ .

Για την κατασκευή του  $S_{SAT}$  χρειάζονται ακριβώς  $n$  βήματα, όπου  $n$  το πλήθος των εικονοστοιχείων της εικόνας. Σε κάθε βήμα εκτελούνται τέσσερις αναφορές σε πίνακα και τρεις απλές αριθμητικές πράξεις. Άρα συνολικά, για την κατασκευή του  $S_{SAT}$  χρειαζόμαστε  $4n$  αναφορές σε πίνακα και  $3n$  πράξεις. Έπειτα, μπορούμε να υπολογίσουμε οποιοδήποτε κατακόρυφο ορθογώνιο σε σταθερό χρόνο, με τέσσερις αναφορές στον πίνακα και τρεις αριθμητικές πράξεις. Παρακάτω θα δούμε το συνολικό όφελος από τη χρήση του  $S_{SAT}$  και του  $S_{RSAT}$ .

Για τον υπολογισμό των τιμών των περιστραμμένων κατά  $45^\circ$  χαρακτηριστικών, χρησιμοποιούμε έναν άλλο πίνακα, τον Περιστραμμένο Πίνακα Προστιθέμενου Εμβαδού ( $S_{RSAT}$ ) [LiMa02], [LiKP03]. Ο  $S_{RSAT}$  έχει κι αυτός τις ίδιες διαστάσεις με αυτές της αρχικής εικόνας. Στη θέση  $(x, y)$  περιλαμβάνει το άθροισμα των τιμών όλων των εικονοστοιχείων που βρίσκονται στο στραμμένο κατά  $45^\circ$  ορθογώνιο, που έχει το κατώτερο άκρο του στο σημείο  $(x, y)$  (βλ. Σχήμα 2.2β). Ισχύει δηλαδή:

$$S_{RSAT}(x, y) = \sum_{y' \leq y, y' \leq y - |x - x'|} i(x', y').$$

Για να κατασκευάσουμε τον  $S_{RSAT}$  χρειάζεται ένα πέρασμα από τα στοιχεία της αρχικής εικόνας από τα αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Ο υπολογισμός γίνεται με τη χρήση της σχέσης:

$S_{RSAT}(x, y) = S_{RSAT}(x-1, y-1) + S_{RSAT}(x+1, y-1) - S_{RSAT}(x, y-2) + i(x, y) + i(x, y-1)$   
και των αρχικών τιμών  $S_{RSAT}(-1, y) = S_{RSAT}(x, -1) = S_{RSAT}(x, -2) = 0$  και  $i(x, -1) = 0$ .




Με τη βοήθεια του  $S_{RSAT}$  μπορούμε να υπολογίσουμε το άθροισμα των τιμών των εικονοστοιχείων μέσα σε οποιοδήποτε περιστραμμένο ορθογώνιο με μόλις τέσσερις αναφορές σε αυτόν. Έτσι, όπως και για τα μη περιστραμμένα χαρακτηριστικά, για να υπολογίσουμε το άθροισμα των τιμών των εικονοστοιχείων που βρίσκονται μέσα στο ορθογώνιο  $r$  στο Σχήμα 2.2δ (που ορίζεται από τα σημεία  $A(x_A, y_A)$ ,  $B(x_B, y_B)$ ,  $\Gamma(x_\Gamma, y_\Gamma)$  και  $\Delta(x_\Delta, y_\Delta)$ ), αρκεί να εκτελέσουμε τις παρακάτω πράξεις:

$$RecSum(r) = S_{RSAT}(x_A, y_A) - S_{RSAT}(x_B, y_B) - S_{RSAT}(x_\Delta, y_\Delta) + S_{RSAT}(x_\Gamma, y_\Gamma)$$

Για τον υπολογισμό του  $S_{RSAT}$ , λοιπόν, θα χρειαστεί να εκτελεστούν  $n$  βήματα, στα οποία θα γίνουν πέντε αναφορές σε πίνακα και τέσσερις αριθμητικές πράξεις. Άρα συνολικά, για τον υπολογισμό του  $S_{RSAT}$  χρειαζόμαστε  $5n$  αναφορές σε πίνακα και  $4n$  πράξεις. Έπειτα, όπως και με τον  $S_{SAT}$ , μπορούμε να υπολογίσουμε οποιοδήποτε περιστραμμένο ορθογώνιο, με μόλις τέσσερις αναφορές σε πίνακα και τρεις αριθμητικές πράξεις.

Συνολικά, για να κατασκευάσουμε τον  $S_{SAT}$  και τον  $S_{RSAT}$  μιας εικόνας που αποτελείται από  $n$  εικονοστοιχεία, θα χρειαστούμε  $9n$  αναφορές σε πίνακα και  $7n$  πράξεις. Έπειτα, όλα τα χαρακτηριστικά θα υπολογίζονται σε σταθερό χρόνο, όπως θα δούμε αναλυτικά παρακάτω. Το γεγονός αυτό, κάνει το κόστος κατασκευής των δύο πινάκων να φαίνεται αρκετά μικρό, συγκριτικά με το χρόνο που κερδίζουμε από τη χρήση τους.

Ας δούμε τώρα, το κόστος υπολογισμού κάθε χαρακτηριστικού που χρησιμοποιούμε, με χρήση των παραπάνω πινάκων. Για τον υπολογισμό των χαρακτηριστικών που αποτελούνται από δύο γειτονικά ορθογώνια (από το Σχήμα 2.1 τα i, ii, ix, x) θα χρειαστούμε 6 αναφορές σε πίνακα και 6 αριθμητικές πράξεις για τον υπολογισμό των δύο ορθογωνίων και 1 αριθμητική πράξη για τη μεταξύ τους αφαίρεση. Άρα, συνολικά 6 αναφορές σε πίνακα και 7 αριθμητικές πράξεις. Για τον υπολογισμό των χαρακτηριστικών που αποτελούνται από τρία γειτονικά ορθογώνια (από το Σχήμα 2.1 τα iv-vii, xi-xiv) θα χρειαστούμε 8 αναφορές σε πίνακα και 11 αριθμητικές πράξεις (9 για τον υπολογισμό των ορθογωνίων και 2 για τις μεταξύ τους πράξεις). Οι πράξεις τελικά μειώνονται σε 8 ακολουθώντας το σκεπτικό που φαίνεται στο Σχήμα 2.3 [LiMa02]. Τα χαρακτηριστικά viii και xv που φαίνονται στο Σχήμα 2.1, παρότι αποτελούνται από δύο ορθογώνια, αυτά δεν είναι γειτονικά μεταξύ τους. Έτσι, για τον υπολογισμό τους, σύμφωνα με το Σχήμα 2.3, χρειάζονται 8 αναφορές σε πίνακα και 8 αριθμητικές πράξεις. Για το χαρακτηριστικό iii που αποτελείται από τέσσερα ορθογώνια, απαιτούνται 9 αναφορές σε πίνακα και 12 αριθμητικές πράξεις. Βλέπουμε, λοιπόν, ότι όλα τα χαρακτηριστικά που περιγράφηκαν στην ενότητα 2.1 μπορούν να υπολογιστούν σε σταθερό χρόνο, ανεξάρτητα από το μέγεθος του χαρακτηριστικού, με τη χρήση των δύο αυτών πινάκων. Το γεγονός αυτό, επιταχύνει δραστικά το σύστημα ανίχνευσης.

	3 ορθογώνια $\Rightarrow$ 8 αναφορές & 11 πράξεις
	2 ορθογώνια $\Rightarrow$ 8 αναφορές & 6 πράξεις
	8 αναφορές & 8 πράξεις

Σχήμα 2.3 - Μείωση του κόστους υπολογισμού χαρακτηριστικών με 3 ορθογώνια

Η ανίχνευση αντικειμένων, θα πρέπει να γίνει σε κάθε δυνατή θέση της εικόνας, καθώς επίσης και σε κάθε δυνατή κλίμακα. Για τον έλεγχο σε κάθε θέση, ο ανιχνευτής κινείται μέσα στην εικόνα, διατρέχοντάς την ολόκληρη, και εφαρμόζοντας τη μέθοδο ανίχνευσης σε κάθε υποπαράθυρο. Για τον έλεγχο σε κάθε κλίμακα, άλλες μέθοδοι δημιουργούν μια πυραμίδα από σμικρύνσεις της εικόνας, και εφαρμόζουν σε κάθε κλίμακα της πυραμίδας τον ανιχνευτή, διατηρώντας σταθερό το μέγεθός του. Με αυτή τη μέθοδο, πέρα από το κόστος της ίδιας της ανίχνευσης, προστίθεται και το χρονικό κόστος της κατασκευής της πυραμίδας εικόνων, το οποίο είναι αρκετά σημαντικό. Στη μέθοδο που εξετάζουμε, αντί να αλλάζουμε το μέγεθος της εικόνας όπου γίνεται η ανίχνευση, αλλάζουμε το μέγεθος του ίδιου του ανιχνευτή. Αυτό είναι εφικτό, καθώς τα χαρακτηριστικά τύπου Haar μπορούν να μεταβληθούν σε μέγεθος. Επίσης, με τη χρήση των δύο πινάκων που είδαμε προηγουμένως, ο υπολογισμός ενός χαρακτηριστικού δεν επηρεάζεται χρονικά από το μέγεθός του. Έτσι, η εφαρμογή της μεθόδου μπορεί να γίνει στον ίδιο ακριβώς χρόνο για οποιαδήποτε κλίμακα του παραθύρου ανίχνευσης. Από μετρήσεις που έχουν γίνει, έχει βρεθεί ότι ο χρόνος που χρειάζεται για την κατασκευή της πυραμίδας εικόνων που χρησιμοποιούν άλλες μέθοδοι, είναι παραπλήσιος με το χρόνο που χρειάζεται η μέθοδος που εξετάζουμε, για όλη τη διαδικασία ανίχνευσης [Vi02]. Έτσι, βλέπουμε ότι οποιαδήποτε μέθοδος χρησιμοποιεί πυραμίδες εικόνων για την ανίχνευση αντικειμένων, θα είναι αναγκαστικά πιο χρονοβόρα από την εξεταζόμενη μέθοδο.

## 2.3 Αλγόριθμος εκπαίδευσης AdaBoost

Υπάρχουν πολλές μέθοδοι για την υλοποίηση ενός ταξινομητή, δεδομένου ενός συνόλου χαρακτηριστικών και ενός συνόλου εκμάθησης θετικών και αρνητικών εικόνων. Έχουν χρησιμοποιηθεί νευρωνικά δίκτυα, μηχανές διανυσμάτων υποστήριξης κ.α. Όπως είδαμε προηγουμένως, τα χαρακτηριστικά που χρησιμοποιούμε μπορούν να υπολογιστούν πάρα πολύ γρήγορα. Το σύνολο όμως των χαρακτηριστικών είναι πολύ μεγάλο (περίπου 120.000 για μια εικόνα 20x20). Έτσι, ο υπολογισμός του πλήρους συνόλου των χαρακτηριστικών για την ανίχνευση σε κάθε υποπαράθυρο της εικόνας, θα ήταν και πάλι πολύ χρονοβόρος [Vi01]. Θα πρέπει, λοιπόν, να επιλέξουμε ένα μικρό αριθμό χαρακτηριστικών από το διαθέσιμο σύνολο, και να

κατασκευάσουμε από αυτά τον ταξινομητή μας. Η επιλογή αυτών των χαρακτηριστικών είναι αρκετά δύσκολη.

Στην μέθοδο ανίχνευσης που εξετάζουμε, χρησιμοποιήθηκε ο αλγόριθμος AdaBoost τόσο για την επιλογή των χαρακτηριστικών που θα χρησιμοποιηθούν, όσο και για την εκπαίδευση του ταξινομητή [FrSc95]. Ο αλγόριθμος εκμάθησης AdaBoost ανήκει στην κατηγορία των αλγορίθμων ενδυνάμωσης (boosting) και χρησιμοποιείται για να αυξήσει την απόδοση ενός οποιουδήποτε απλού αλγορίθμου ταξινόμησης. Ο απλός αλγόριθμος ταξινόμησης λέγεται και ασθενής αλγόριθμος ταξινόμησης, καθώς ακόμα και η καλύτερη συνάρτηση ταξινόμησης που μπορεί να προκύψει από αυτόν, δεν αναμένεται να ταξινομεί καλά τα δεδομένα. Συγκεκριμένα, αρκεί η συνάρτηση ταξινόμησης να έχει απόδοση ελαφρώς καλύτερη από την τυχαία ταξινόμηση (50%). Για να αυξήσει, λοιπόν, την απόδοση ενός ασθενούς αλγορίθμου ταξινόμησης, ο AdaBoost συνδυάζει μια συλλογή ασθενών συναρτήσεων ταξινόμησης χρησιμοποιώντας άπληστο αλγόριθμο, ώστε να σχηματίσει από αυτούς έναν ισχυρότερο ταξινομητή.

Η βελτίωση του ασθενούς αλγορίθμου ταξινόμησης πραγματοποιείται, καλώντας τον αλγόριθμο να επιλύσει μια αλληλουχία προβλημάτων ταξινόμησης. Αρχικά, όλα τα παραδείγματα (θετικά και αρνητικά) παίρνουν μια τιμή βάρους, η οποία είναι ίδια για όλα. Δίνονται στον αλγόριθμο τα παραδείγματα και πραγματοποιείται ο πρώτος κύκλος εκμάθησης, όπου ο αλγόριθμος ταξινομεί όλα τα παραδείγματα με κάθε διαθέσιμη συνάρτηση ταξινόμησης. Έπειτα, οι συναρτήσεις ταξινόμησης διατάσσονται σύμφωνα με τα αποτελέσματά τους, λαμβάνοντας υπόψη το βάρος κάθε παραδείγματος. Επιλέγεται ένας μικρός αριθμός συναρτήσεων ταξινόμησης, από αυτές με τα καλύτερα αποτελέσματα, που αποτελούν τον πρώτο ασθενή ταξινομητή. Ο πρώτος κύκλος εκμάθησης ολοκληρώνεται και τα βάρη των παραδειγμάτων ισοσταθμίζονται, δίνοντας μεγαλύτερο βάρος στα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον πρώτο ασθενή ταξινομητή. Έτσι, στον δεύτερο κύκλο εκμάθησης ο αλγόριθμος ταξινόμησης θα θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον προηγούμενο ταξινομητή. Τα βήματα επαναλαμβάνονται διαδοχικά, μέχρι να φτάσουμε στο επίπεδο του συνολικού λόγου λανθασμένης ταξινόμησης που επιθυμούμε. Τελικά, ο ισχυρός ταξινομητής προκύπτει από τον συνδυασμό των ασθενών ταξινομητών που επιλέχθηκαν και ένα κατώφλι. Κατά την διαδικασία της ταξινόμησης ενός υποπαράθυρου εικόνας από τον ισχυρό ταξινομητή, εφαρμόζονται στο υποπαράθυρο όλοι οι ασθενείς ταξινομητές. Τα αποτελέσματα των ασθενών ταξινομητών αθροίζονται, και αν το άθροισμα ξεπερνά το κατώφλι του ταξινομητή, το υπό εξέταση αντικείμενο ταξινομείται ως θετικό, αλλιώς ως αρνητικό.

Υπάρχουν τέσσερις εκδοχές του αλγορίθμου AdaBoost [FrHT00]. Η αρχική εκδοχή ονομάζεται Διακριτός AdaBoost (Discrete AdaBoost – DAB) καθώς η συνάρτηση ταξινόμησης κάθε ασθενή ταξινομητή παίρνει μόνο δύο διακριτές τιμές, τις  $\{-1,1\}$  ανάλογα με το αν ένα δείγμα ταξινομείται ως θετικό ή αρνητικό. Η δεύτερη εκδοχή ονομάζεται Πραγματικός AdaBoost (Real AdaBoost – RAB), καθώς η συνάρτηση ταξινόμησης κάθε ασθενή ταξινομητή παίρνει όλες τις πραγματικές τιμές στο διάστημα  $[0,1]$ . Με τη χρήση του RAB, μπορούμε να έχουμε μια ένδειξη εμπιστοσύνης για τα αποτελέσματα της ταξινόμησης, χρησιμοποιώντας τις τιμές που επιστρέφονται από τον αλγόριθμο και όχι μόνο το αποτέλεσμα της θετικής ή αρνητικής ταξινόμησης. Άλλη εκδοχή είναι ο Ήπιος AdaBoost (Gentle AdaBoost – GAB), ο οποίος βασίζεται στον Πραγματικό AdaBoost αλλά χρησιμοποιεί βήματα της μεθόδου Newton αντί για ακριβή υπολογισμό. Τέλος, υπάρχει και ο LogitBoost, ο οποίος έχει δύο παραλλαγές, αυτή που χρησιμοποιεί δύο κλάσεις και αυτή που χρησιμοποιεί J κλάσεις. Ο αριθμός των κλάσεων επηρεάζει

την τιμή της εκτίμησης πιθανότητας κάθε δείγματος  $x_i$ , η οποία ισούται με  $p(x_i) = \frac{1}{2}$  στη μία περίπτωση και  $p(x_i) = \frac{1}{J}$  στην άλλη.

Στη μέθοδο ανίχνευσης αντικειμένων που χρησιμοποιούμε, κάθε ασθενής αλγόριθμος εκμάθησης περιορίζεται στο σύνολο των συναρτήσεων ταξινόμησης που αποτελούνται από ένα μόνο χαρακτηριστικό τύπου Haar. Προφανώς, από ένα μόνο χαρακτηριστικό δε μπορούμε να περιμένουμε ιδιαίτερα χαμηλό λόγο σφάλματος. Σε κάθε στάδιο του αλγορίθμου AdaBoost επιλέγεται το χαρακτηριστικό που διαχωρίζει καλύτερα τα θετικά από τα αρνητικά δείγματα. Για κάθε χαρακτηριστικό, ο ασθενής αλγόριθμος εκμάθησης προσδιορίζει ένα κατώφλι της τιμής του χαρακτηριστικού, που ελέγχοντάς το περιορίζονται οι λανθασμένες ταξινομήσεις από το συγκεκριμένο χαρακτηριστικό στις ελάχιστες δυνατές. Έπειτα, επιλέγεται ως ασθενής ταξινομητής το χαρακτηριστικό τύπου Haar, που, για το δεδομένο κατώφλι του, κάνει τη συνολικά καλύτερη ταξινόμηση. Ο AdaBoost συνεχίζει εκπαιδύοντας όλους τους ασθενείς ταξινομητές, μέχρι το σημείο που ο ισχυρός συνολικός ταξινομητής επιτυγχάνει το επίπεδο ταξινόμησης που ζητάμε.

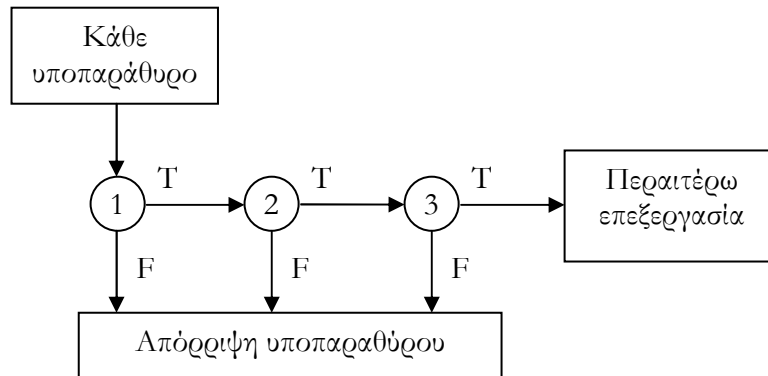
Ο αλγόριθμος AdaBoost παρέχει αρκετά ισχυρές εγγυήσεις για την ορθότητά του. Έχει αποδειχθεί, ότι το σφάλμα ταξινόμησης του ισχυρού ταξινομητή που προκύπτει από την εφαρμογή του αλγορίθμου, τείνει προς το μηδέν εκθετικά ως προς τον αριθμό των κύκλων εκπαίδευσης [SFBL98]. Επίσης, η όλη διαδικασία της εκμάθησης πραγματοποιείται με μεγάλη ταχύτητα. Ας θεωρήσουμε ότι έχουμε στη διάθεσή μας  $K$  χαρακτηριστικά τύπου Haar και  $N$  εικόνες-παραδείγματα. Για να κατασκευαστεί ένας ισχυρός ταξινομητής από τον αλγόριθμο AdaBoost, που αποτελείται από  $M$  ασθενείς ταξινομητές, χρειάζονται  $O(MNK)$  βήματα, σε αντίθεση με άλλους αλγορίθμους που χρειάζονται  $O(MNKN)$  βήματα.

## 2.4 Διαδοχικά Συνδεδεμένος Ταξινομητής (ΔΣΤ)

Σε αυτή την ενότητα παρουσιάζεται η μέθοδος ταξινόμησης με χρήση ενός Διαδοχικά Συνδεδεμένου Ταξινομητή (ΔΣΤ) [Vi01]. Η μέθοδος αυτή βοηθά στη επίτευξη υψηλού λόγου ανίχνευσης, μειώνοντας σημαντικά τον απαιτούμενο χρόνο. Η ιδέα στηρίζεται στο γεγονός ότι μπορούμε να κατασκευάσουμε πολλούς μικρούς σε μέγεθος ταξινομητές, τους οποίους θα συνδέσουμε διαδοχικά. Μπορούμε, λοιπόν, να χρησιμοποιήσουμε απλούστερους και πολύ γρήγορους ταξινομητές αρχικά, οι οποίοι θα απορρίπτουν γρήγορα την πλειονότητα των αρνητικών υποπαράθυρων, και πιο σύνθετους και χρονοβόρους αργότερα, ώστε να μειώσουμε το λόγο λανθασμένων ανιχνεύσεων.

Η διάταξη του ΔΣΤ θυμίζει τη μορφή ενός εκφυλισμένου δέντρου απόφασης. Κάθε υποπαράθυρο της εικόνας εισέρχεται στον πρώτο ταξινομητή. Αν ο ταξινομητής το κατατάξει ως θετικό, αυτό περνά ως είσοδος στον δεύτερο ταξινομητή. Αν και αυτός το κατατάξει ως θετικό, τότε περνά στον τρίτο κ.ο.κ. Αν σε αυτή τη διαδοχή κάποιος ταξινομητής κατατάξει το υποπαράθυρο ως αρνητικό, τότε αυτό απορρίπτεται και δεν εξετάζεται από κανένα άλλο ταξινομητή (βλέπε Σχήμα 2.4). Θα μπορούσαμε να παρομοιάσουμε τον ΔΣΤ με έναν μεγάλο ταξινομητή που αποτελείται από το σύνολο των χαρακτηριστικών του ΔΣΤ, όπου όμως δεν περιμένουμε να υπολογιστεί όλο το πλήθος των χαρακτηριστικών. Αντίθετα, ελέγχουμε ανά μερικά

χαρακτηριστικά το άθροισμα των τιμών των χαρακτηριστικών για να αποφασίσουμε αν το υπό εξέταση υποπαράθυρο απορρίπτεται ή όχι.



Σχήμα 2.4 - Διαδοχικά Συνδεδεμένος Ταξινομητής

Κάθε ταξινομητής του ΔΣΤ εκπαιδεύεται χρησιμοποιώντας ένα σύνολο θετικών και ένα σύνολο αρνητικών παραδειγμάτων, όπως είδαμε στην ενότητα 2.3. Το σύνολο των θετικών παραδειγμάτων είναι το ίδιο κατά την εκπαίδευση κάθε ταξινομητή. Το σύνολο των αρνητικών παραδειγμάτων όμως, μεταβάλλεται. Συγκεκριμένα, κάθε ταξινομητής εκπαιδεύεται χρησιμοποιώντας ως αρνητικά παραδείγματα, τα παραδείγματα που ταξινομούνται από τους προηγούμενους ταξινομητές του ΔΣΤ ως θετικά [ViJo01]. Αυτό αυξάνει σε πολύ μεγάλο βαθμό τα αρνητικά παραδείγματα τα οποία θα εξεταστούν συνολικά. Για να φτάσει ένας συγκεκριμένος αριθμός αρνητικών παραδειγμάτων στον τρέχοντα ταξινομητή, θα πρέπει τα παραδείγματα αυτά να ταξινομηθούν από όλους τους προηγούμενους ταξινομητές του ΔΣΤ (λανθασμένα) ως θετικά. Ας θεωρήσουμε ότι σε κάθε στάδιο ενός ΔΣΤ θέλουμε να εξετάζονται 1.000 αρνητικά παραδείγματα και κάθε στάδιο έχει λόγο λανθασμένης θετικής ανίχνευσης 0,5. Τότε, για να περάσουν στο δέκατο στάδιο 1.000 αρνητικά παραδείγματα, αυτά θα χρειαστεί να έχουν ταξινομηθεί από τα προηγούμενα εννιά στάδια ως θετικά. Έτσι, συνολικά θα πρέπει να εξεταστούν περίπου 512.000 αρνητικά παραδείγματα.

Η εξέταση πολύ μεγαλύτερου αριθμού αρνητικών παραδειγμάτων αυξάνει την τελική απόδοση του ΔΣΤ. Αντίθετα, κάθε ταξινομητής καλείται να πραγματοποιήσει μια πιο δύσκολη ταξινόμηση από αυτές των προηγούμενων ταξινομητών. Τα αρνητικά παραδείγματα που θα έχει στη διάθεσή του θα είναι πιο δύσκολα στην ταξινόμηση από τα παραδείγματα που είχαν τα προηγούμενα από αυτό στάδια. Έχοντας, λοιπόν, πιο δύσκολο σύνολο εκπαίδευσης ένας ταξινομητής που βρίσκεται σε προχωρημένο στάδιο, θα παρουσιάσει αυξημένες λανθασμένες ταξινομήσεις, θετικές και αρνητικές.

Οι απλοί ταξινομητές ενός ΔΣΤ, θα πρέπει να έχουν πολύ χαμηλό λόγο λανθασμένων αρνητικών ταξινομήσεων, ώστε να μην χάνονται τα πραγματικά αντικείμενα στη συνολική ταξινόμηση. Για να διασφαλίσουμε τη σωστή λειτουργία του ΔΣΤ, θα πρέπει να αυξήσουμε περαιτέρω τις θετικές ταξινομήσεις (είτε αφορούν πραγματικά αντικείμενα είτε όχι) [ViJo01]. Μια τεχνική για να πετύχουμε αυτό το αποτέλεσμα είναι να επέμβουμε στις τιμές των κατωφλίων των ταξινομητών που προσδιόρισε ο αλγόριθμος AdaBoost κατά την εκπαίδευση. Το κατώφλι ενός ταξινομητή ορίζει την ελάχιστη τιμή του σταθμισμένου με βάρη άθροισματος των τιμών των χαρακτηριστικών που θα πρέπει να έχει ένα υποπαράθυρο για να ταξινομηθεί ως θετικό. Ένα υποπαράθυρο ταξινομείται, δηλαδή, ως θετικό, όταν το (σταθμισμένο) άθροισμα των τιμών των

χαρακτηριστικών που υπολογίστηκε για αυτό ξεπερνά το κατώφλι του ταξινομητή. Έτσι, μειώνοντας τις τιμές των κατωφλίων θα αυξηθεί ο αριθμός των παραθύρων που ταξινομούνται ως θετικά, άρα και ο λόγος θετικών ταξινομήσεων.

Η απόδοση ενός ΔΣΤ εξαρτάται από την απόδοση κάθε σταδίου του, καθώς και από το πλήθος των σταδίων που χρησιμοποιούνται. Έστω ότι έχουμε ένα ΔΣΤ που αποτελείται από  $K$  στάδια, που το στάδιο  $i$  έχει λόγο λανθασμένης ανίχνευσης  $f_i$ . Τότε, ο συνολικός λόγος λανθασμένης ανίχνευσης  $F$  του ΔΣΤ θα είναι:

$$F = \prod_{i=1}^K f_i$$

Αν το στάδιο  $i$  του ΔΣΤ έχει λόγο σωστής ανίχνευσης  $d_i$ , τότε ο συνολικός λόγος ανίχνευσης  $D$  του ταξινομητή θα είναι:

$$D = \prod_{i=1}^K d_i$$

Αν το στάδιο  $i$  αποτελείται από  $n_i$  χαρακτηριστικά, τότε ο μέσος αριθμός χαρακτηριστικών  $N$  που εκτιμάται ότι θα χρειαστεί να υπολογιστούν, είναι:

$$N = n_0 + \sum_{i=1}^K \left( n_i \prod_{j < i} f_j \right)$$

Ας πάρουμε ως παράδειγμα μια εικόνα διαστάσεων 600x400 εικονοστοιχείων. Αυτή η εικόνα περιέχει περίπου 350.000 υποπαράθυρα τα οποία θα πρέπει να εξεταστούν από τον ταξινομητή μας. Έστω ότι έχουμε έναν ταξινομητή των 200 χαρακτηριστικών. Για κάθε εξεταζόμενο υποπαράθυρο της εικόνας, θα πρέπει να υπολογίσουμε και τα 200 χαρακτηριστικά για να αποφασίσουμε αν το υποπαράθυρο θα απορριφθεί ή όχι. Για το σύνολο της εικόνας, λοιπόν, θα πρέπει να υπολογιστούν:  $350.000 \cdot 200 = 70.000.000$  χαρακτηριστικά, μέχρι να ολοκληρωθεί η ανίχνευση. Έστω τώρα, ότι χρησιμοποιούμε ένα ΔΣΤ με 10 στάδια, καθένα από τα οποία έχει 20 χαρακτηριστικά και λόγο λανθασμένης ανίχνευσης 0,5. Τότε ο μέσος όρος των χαρακτηριστικών που θα χρειαστεί να υπολογιστεί για κάθε υποπαράθυρο θα είναι περίπου 33. Άρα, για να ταξινομηθούν όλα τα υποπαράθυρα της εικόνας, θα χρειαστεί να υπολογίσουμε περίπου:  $350.000 \cdot 33 = 11.550.000$  χαρακτηριστικά. Βλέπουμε, λοιπόν, ότι χρησιμοποιώντας αυτή την τεχνική, αυξάνουμε την ταχύτητα της ταξινόμησης.

Είδαμε, λοιπόν, ότι ένας ΔΣΤ προσδιορίζεται από τον αριθμό των σταδίων του  $K$ , το λόγο λανθασμένης ανίχνευσης  $f_i$  και το λόγο ανίχνευσης  $d_i$  του κάθε σταδίου, καθώς και τον αριθμό των Haar χαρακτηριστικών  $n_i$  του κάθε σταδίου. Θα λέγαμε ότι ο ιδανικός ΔΣΤ έχει λόγο ανίχνευσης  $d_i$  σε κάθε στάδιο που προσεγγίζει τη μονάδα, λόγο λανθασμένης ανίχνευσης  $f_i$  που τείνει στο μηδέν και αριθμό χαρακτηριστικών ανά στάδιο  $n_i$  πολύ χαμηλό, ακόμα και ίσο με ένα. Οι απαιτήσεις αυτές όμως είναι ασυμβίβαστες μεταξύ τους. Ο λόγος λανθασμένης ανίχνευσης ενός σταδίου εξαρτάται από τον αριθμό των χαρακτηριστικών του ταξινομητή του σταδίου αυτού. Δε μπορούμε να περιμένουμε, λοιπόν, πολύ χαμηλούς λόγους λανθασμένης ανίχνευσης με χρήση λίγων χαρακτηριστικών. Αυτό μπορεί να συμβεί μόνο στα πρώτα στάδια ενός ΔΣΤ. Σε αυτά είναι δυνατή η εύρεση κάποιων χαρακτηριστικών τύπου Haar, τα οποία αντιστοιχούν σε έντονα γεωμετρικά μοτίβα του υπό ανίχνευση αντικειμένου και χαρακτηρίζουν τα θετικά παραδείγματα. Έτσι, για να μειώσουμε το λόγο λανθασμένης ανίχνευσης σε ένα στάδιο, θα πρέπει να αυξήσουμε

τον αριθμό των Haar χαρακτηριστικών στο στάδιο αυτό. Αυξάνοντας τα χαρακτηριστικά ενός σταδίου, αμέσως αυξάνεται και ο μέσος αριθμός χαρακτηριστικών  $N$  που εκτιμάται ότι θα χρειαστεί να υπολογιστούν για μία ανίχνευση, άρα και ο συνολικός χρόνος της ανίχνευσης. Επίσης, με την αύξηση των χαρακτηριστικών, αυξάνεται η δυσκολία που αντιμετωπίζει ένα θετικό παράδειγμα για να ταξινομηθεί ως θετικό, καθώς θα πρέπει να ελεγχθεί περισσότερες φορές μέχρι να αποφασιστεί αν είναι θετικό ή όχι. Έτσι, μειώνεται ο αριθμός των θετικών υποπαράθρων που ταξινομούνται ως θετικά, άρα μειώνεται και ο λόγος σωστής ανίχνευσης. Τελικά, βλέπουμε ότι για κάθε ταξινομητή που εκπαιδεύουμε θα πρέπει να κάνουμε κάποιες παραχωρήσεις και επιλογές που θα επηρεάσουν την απόδοση του ανιχνευτή μας. Όλες οι επιλογές που χρειάζεται να κάνουμε εξηγούνται στο Κεφάλαιο 3.



## Κεφάλαιο 3.

### Πειραματική μεθοδολογία

Είδαμε στο Κεφάλαιο 2 ότι για την εκπαίδευση ενός ανιχνευτή που χρησιμοποιεί τη μέθοδο των Viola και Jones, χρειάζεται να γίνουν κάποιες επιλογές βασικών παραμέτρων. Από τις παραμέτρους αυτές επηρεάζεται η δομή, η απόδοση και η ταχύτητα του ανιχνευτή που θα προκύψει. Σε αυτό το κεφάλαιο περιγράφονται με αναλυτικό τρόπο όλες αυτές οι επιλογές που πρέπει να γίνουν, από την προετοιμασία των παραδειγμάτων που θα χρησιμοποιηθούν μέχρι την τελική δομή του ανιχνευτή.

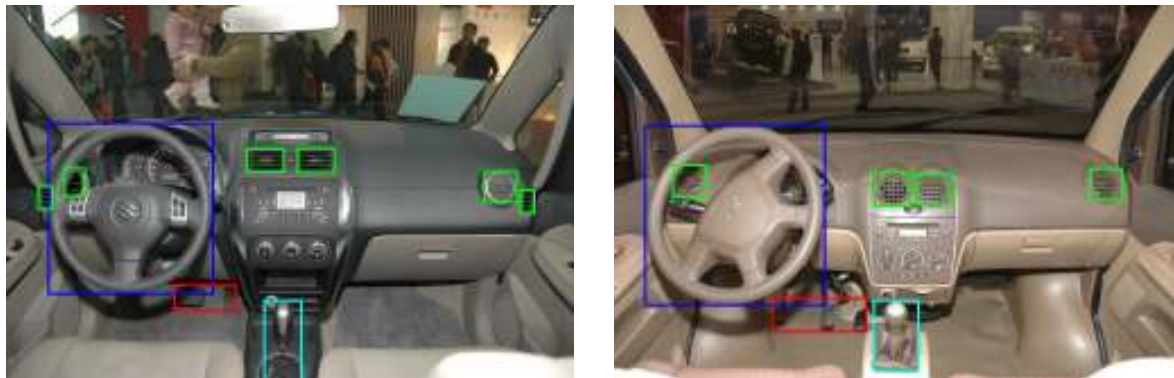
#### 3.1 Προετοιμασία

Είδαμε σε προηγούμενες ενότητες, ότι για την εκπαίδευση του ταξινομητή χρειαζόμαστε ένα σύνολο θετικών παραδειγμάτων, καθώς και ένα σύνολο αρνητικών. Ως θετικά παραδείγματα ορίζονται οι εικόνες που περιέχουν μόνο το αντικείμενο που θέλουμε να ανιχνεύσουμε. Τα αρνητικά παραδείγματα περιέχουν οτιδήποτε άλλο, εκτός από το αντικείμενο που θέλουμε να ανιχνεύσουμε. Εκτός από τα δύο αυτά σύνολα παραδειγμάτων, θα χρειαστούμε και ένα τρίτο σύνολο εικόνων δοκιμής, το οποίο θα χρησιμοποιήσουμε για να αξιολογήσουμε τον ανιχνευτή που θα κατασκευάσουμε.

Τα θετικά παραδείγματα προκύπτουν από εικόνες οι οποίες να περιέχουν το αντικείμενο για το οποίο θέλουμε να εκπαιδεύσουμε τον ανιχνευτή μας. Σε αυτές τις εικόνες το αντικείμενο θα πρέπει να φαίνεται ολόκληρο, χωρίς επικαλύψεις από άλλα αντικείμενα. Η μέθοδος που χρησιμοποιούμε ψάχνει για γεωμετρικά σχήματα που χαρακτηρίζουν ένα αντικείμενο σε όλα τα θετικά παραδείγματα, σε συγκεκριμένες θέσεις. Αν για παράδειγμα κατασκευάζουμε έναν ανιχνευτή για εντοπισμό ολόκληρου του σώματος ενός όρθιου ανθρώπου, δίνοντας ως θετικό παράδειγμα τη φωτογραφία ενός ανθρώπου από τη μέση και επάνω δεν προσφέρει σωστές πληροφορίες για την εκμάθηση του ανιχνευτή. Ακόμα, δίνοντας τη φωτογραφία ενός ανθρώπου που το μισό του σώμα κρύβεται από ένα αντικείμενο, ή από έναν άλλο άνθρωπο, και πάλι δε βοηθά στην κατασκευή του ανιχνευτή, καθώς δε θα μπορούν να εντοπιστούν τα χαρακτηριστικά που περιγράφουν το αντικείμενο που μας ενδιαφέρει. Επίσης, το αντικείμενο θα πρέπει να βρίσκεται στην εικόνα υπό την ίδια, περίπου, γωνία, όπως θα διαπιστώσουμε από τα πειράματα στο Κεφάλαιο 4. Ένας άνθρωπος σε όρθια στάση (κατακόρυφη εικόνα) και ένας άνθρωπος ξαπλωμένος (οριζόντια εικόνα), έχουν τα ίδια, περίπου, χαρακτηριστικά, αλλά δεν τα έχουν στην ίδια θέση. Άρα, τα δύο αυτά παραδείγματα δε θα βοηθήσουν στην κατασκευή ενός ενιαίου ανιχνευτή.

Τα θετικά παραδείγματα, θα πρέπει να είναι εικόνες που περιέχουν μόνο το αντικείμενο που μας ενδιαφέρει. Για τον ορισμό του αντικειμένου μέσα σε μια εικόνα, αρκεί να προσδιορίσουμε το ορθογώνιο παραλληλόγραμμο το οποίο περιέχει όλο το αντικείμενο, χωρίς περιττά κενά. Θα πρέπει λοιπόν, αφού συγκεντρώσουμε το σύνολο των εικόνων, να τις χαρακτηρίσουμε, αναφέροντας

τις περιοχές που περιέχουν το αντικείμενο. Στα πλαίσια αυτής της διπλωματικής, κατασκευάσαμε ένα πρόγραμμα χαρακτηρισμού εικόνων με πολλές δυνατότητες, το οποίο θα παρουσιαστεί αναλυτικά στο Κεφάλαιο 5. Στο Σχήμα 3.1 φαίνονται δύο παραδείγματα εικόνων που έχουν χαρακτηριστεί για διάφορα αντικείμενα.



Σχήμα 3.1 - Παραδείγματα χαρακτηρισμένων εικόνων.

Το πλήθος των θετικών παραδειγμάτων θα πρέπει να είναι αρκετά μεγάλο (3.000 έως 7.000). Η εύρεση τόσων θετικών παραδειγμάτων, ειδικά όταν θέλουμε να εκπαιδεύσουμε έναν ταξινομητή για αντικείμενα όχι τόσο κοινά όσο οι άνθρωποι και τα πρόσωπα, δεν είναι πάντα εφικτή. Το χωρικό μέγεθος των θετικών παραδειγμάτων θα πρέπει να είναι αρκετά μικρό ώστε να μπορούν να γίνουν σε λογικό χρόνο οι υπολογισμοί που χρειάζονται. Αντίθετα, όσο μεγαλύτερο είναι το μέγεθός τους, τόσο καλύτερη ανάλυση του αντικειμένου θα μπορέσουμε να κάνουμε. Στη βιβλιογραφία αναφέρονται μεγέθη από 17x17 εικονοστοιχεία, έως 24x24. Προφανώς ο λόγος των διαστάσεων των παραδειγμάτων εξαρτάται από το είδος του αντικειμένου. Για παράδειγμα, άλλο λόγο διαστάσεων θα έχουν τα παραδείγματα για την κατηγορία "όρθιοι άνθρωποι" και άλλο για την κατηγορία "αυτοκίνητα σε πλάγια όψη". Το χαρακτηριστικό μέγεθος που πρέπει να έχουμε υπ' όψιν είναι το εμβαδό των παραδειγμάτων, το οποίο θα πρέπει να είναι από 300 έως 600 εικονοστοιχεία.

Για τα αρνητικά παραδείγματα δεν χρειάζεται τόση επεξεργασία. Αρκεί να βρούμε εικόνες που να μην περιέχουν καμία εμφάνιση του αντικειμένου που μας ενδιαφέρει. Μια τεχνική που μπορεί να δώσει γρήγορα, αρκετά αρνητικά παραδείγματα, είναι να καλύψουμε τα θετικά παραδείγματα από τις εικόνες που χρησιμοποιήσαμε προηγουμένως. Εδώ θα πρέπει να προσέξουμε, ότι μπορεί στις εικόνες αυτές να υπάρχουν κάποια αντικείμενα τα οποία κρίναμε πως δεν έπρεπε να χαρακτηριστούν ως θετικά (λόγω επικαλύψεων, γωνιών κλήσης κτλ). Αυτά θα πρέπει επίσης να καλυφθούν, ώστε να μην υπάρχει κανένα απολύτως αντικείμενο της κατηγορίας που μας ενδιαφέρει, στα αρνητικά παραδείγματα.

Κατά την εκπαίδευση του ταξινομητή, θέτουμε ως παράμετρο το πλήθος των αρνητικών παραδειγμάτων τα οποία θέλουμε να ελέγχονται σε κάθε στάδιο του Διαδοχικά Συνδεδεμένου Ταξινομητή μας. Όπως είδαμε στην ενότητα 2.4, για να έχουμε π.χ. 1.000 αρνητικά παραδείγματα σε κάθε στάδιο, με κάθε στάδιο να έχει λόγο λανθασμένης ανίχνευσης 0,5, στο δέκατο στάδιο θα χρειαστεί να εξεταστούν περίπου 512.000 αρνητικά παραθύρα. Βλέπουμε λοιπόν, ότι το πλήθος των αρνητικών παραδειγμάτων θα πρέπει να είναι πάρα πολύ μεγάλο.

Κάθε αρνητικό παράδειγμα έχει διαστάσεις ίδιες με αυτές των θετικών παραδειγμάτων, ή πολλαπλάσια αυτών. Μπορούμε να ορίσουμε ως αρνητικά παράθυρα εικόνες οι οποίες έχουν ακριβώς το ίδιο μέγεθος με τα θετικά παραδείγματα και προφανώς δεν περιέχουν εμφανίσεις του υπό εξέταση αντικειμένου. Η επιλογή αυτή παρότι φαίνεται καλή, οδηγεί στην ανάγκη για κατασκευή εκατομμυρίων αρχείων εικόνων, γεγονός το οποίο δεν είναι δυνατό. Αντίθετα, μπορούμε να επιλέξουμε για τα αρνητικά παραδείγματα εικόνες μεγάλης ανάλυσης οι οποίες δεν περιέχουν καμία εμφάνιση του υπό εξέταση αντικειμένου. Τότε, ο αλγόριθμος εκπαίδευσης θα διασχίζει αυτές τις εικόνες ψάχνοντας για υποπαράθυρα τα οποία θα αποτελέσουν τα αρνητικά παραδείγματα. Έτσι, με λίγες χιλιάδες εικόνες υψηλής ανάλυσης (περίπου τόσες εικόνες όσες και ο αριθμός των θετικών παραδειγμάτων) μπορούμε να ικανοποιήσουμε τις ανάγκες του αλγορίθμου εκπαίδευσης.



(α)



(β)



(γ)

Σχήμα 3.2 - Παραδείγματα εικόνων για την εκπαίδευση και τη δοκιμή ενός ανιχνευτή. α) Θετικά παραδείγματα εικόνων, β) αρνητικά παραδείγματα εικόνων, γ) εικόνες δοκιμής.

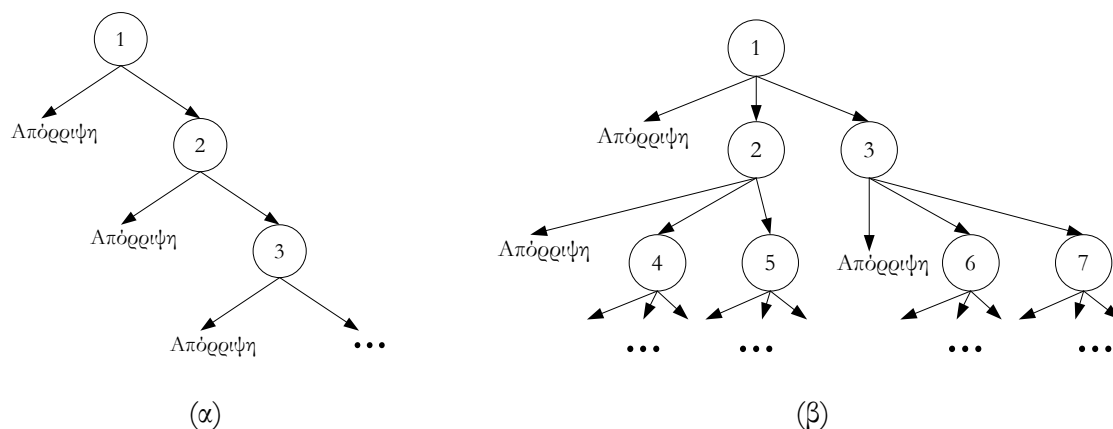
Το σύνολο των εικόνων δοκιμής είναι ένα σύνολο εικόνων που περιέχουν αντικείμενα του τύπου που θέλουμε να εντοπίσουμε. Αυτές τις εικόνες θα τις χρειαστούμε μετά την κατασκευή του ανιχνευτή, για να μετρήσουμε την απόδοσή του. Οι εικόνες αυτές δε θα πρέπει να ανήκουν στο σύνολο των αρνητικών ή θετικών παραδειγμάτων που δημιουργήσαμε προηγουμένως. Οι εικόνες δοκιμής θα πρέπει να χαρακτηριστούν ώστε να μπορέσουμε με αυτόματο τρόπο να μετρήσουμε την απόδοση του ανιχνευτή, όπως θα δούμε στην ενότητα 3.5.

Αφού ολοκληρωθούν τα παραπάνω στάδια της συγκέντρωσης των θετικών και αρνητικών παραδειγμάτων, καθώς και των εικόνων δοκιμής, τελειώνει το στάδιο της προετοιμασίας και μπορούμε να ξεκινήσουμε την εκπαίδευση ενός ανιχνευτή. Παραδείγματα των τριών κατηγοριών εικόνων φαίνονται στο Σχήμα 3.2.

## 3.2 Επιλογές εκπαίδευσης

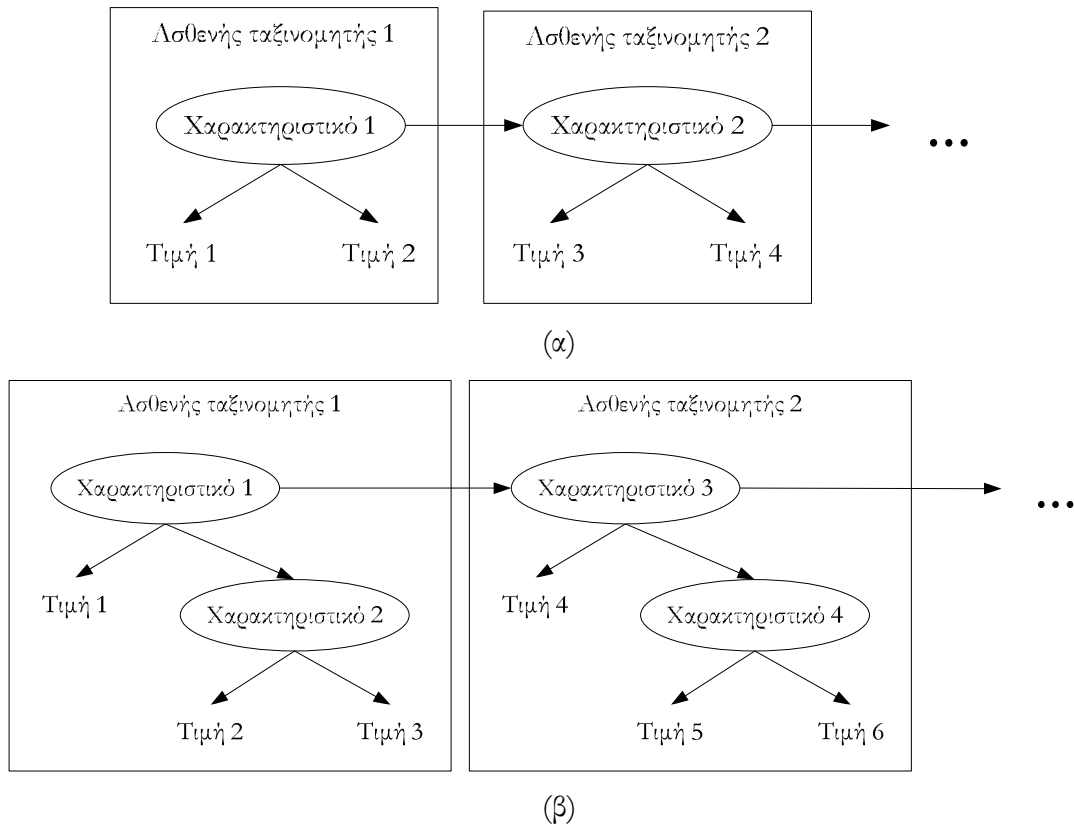
Ο ταξινομητής που περιγράψαμε αποτελείται από ένα πλήθος διαδοχικά συνδεδεμένων ταξινομητών, όπως είδαμε στην ενότητα 2.4. Ο αριθμός των ταξινομητών που θα συνδυαστούν λέγεται και αριθμός σταδίων του ταξινομητή και ορίζεται από την παράμετρο  $K$ . Ο αριθμός των σταδίων μπορεί να πάρει τιμές σε ένα μεγάλο εύρος και σε συνδυασμό με άλλες παραμέτρους επηρεάζει την απόδοση του ταξινομητή. Αριθμός σταδίων ανάμεσα σε 15 και 25 είναι αρκετά συνηθισμένος.

Η δομή του συνολικού ταξινομητή ορίζεται με την παράμετρο  $N_{tree-splits}$  και μπορεί να είναι είτε ενός εκφυλισμένου δέντρου (όπως περιγράψαμε στην ενότητα 2.4), είτε αυτή ενός Δέντρου Ταξινόμησης και Παρεμβολής (ΔΤΠ) (CART – Classification and Regression Tree). Αν επιλέξουμε τη δομή του εκφυλισμένου δέντρου, τότε σε κάθε κόμβο θα έχουμε δύο επιλογές, είτε να απορριφθεί ένα υποπαράθυρο, είτε να εξεταστεί από τον επόμενο στη σειρά ταξινομητή. Αν επιλέξουμε τη δομή ενός πλήρους ΔΤΠ, τότε σε κάθε κόμβο του δέντρου θα έχουμε περισσότερες επιλογές, αφού ένα υποπαράθυρο, ανάλογα με το αποτέλεσμα της ταξινόμησης που πραγματοποιήθηκε, μπορεί είτε να απορριφθεί, είτε να εξεταστεί από διάφορους ταξινομητές. Ένα παράδειγμα των δομών αυτών φαίνεται στο Σχήμα 3.3.



Σχήμα 3.3 - Δομή διαδοχικά συνδεδεμένου ταξινομητή. α) Εκφυλισμένο δέντρο, β) πλήρες δέντρο ταξινόμησης και παρεμβολής

Η δομή κάθε ασθενούς ταξινομητή επηρεάζεται από την παράμετρο  $N_{features}$ , η οποία ορίζει τον αριθμό των χαρακτηριστικών τύπου Haar από τα οποία θα χρειαστεί να υπολογιστούν. Μπορούμε να δώσουμε στην παράμετρο  $N_{features}$  την τιμή 1, ώστε να έχουμε ένα χαρακτηριστικό ανά ασθενή ταξινομητή, όπως περιγράψαμε στην ενότητα 2.3 και φαίνεται στο Σχήμα 3.4α. Αλλιώς μπορούμε να δώσουμε μεγαλύτερη τιμή στην παράμετρο  $N_{features}$  ώστε να υπολογίζουμε περισσότερα χαρακτηριστικά σε κάθε ασθενή ταξινομητή, όπως φαίνεται στο Σχήμα 3.4β.



Σχήμα 3.4 - Δομή ασθενούς ταξινομητή. α) Δομή για  $N_{features} = 1$ , β) δομή για  $N_{features} = 2$

Για τους ισχυρούς ταξινομητές πρέπει να ορίσουμε τον ελάχιστο λόγο σωστής ανίχνευσης  $d$  και το μέγιστο λόγο λανθασμένης ανίχνευσης  $f$ . Όπως είδαμε στην ενότητα 2.4 ο συνδυασμός καθενός από τα δύο αυτά μεγέθη με τον αριθμό των σταδίων  $K$ , προσδιορίζει την αναμενόμενη απόδοση του ανιχνευτή. Ακόμα, θα πρέπει να ορίσουμε τον αριθμό των θετικών και των αρνητικών παραδειγμάτων που θα χρησιμοποιηθούν στην εκπαίδευση κάθε σταδίου,  $N_{pos}$  και  $N_{neg}$  αντίστοιχα. Επίσης, μπορούμε να επιλέξουμε ποια χαρακτηριστικά τύπου Haar από αυτά που παρουσιάστηκαν στην ενότητα 2.1 θα χρησιμοποιηθούν. Μπορούμε δηλαδή, να μην χρησιμοποιήσουμε όλα τα χαρακτηριστικά που είδαμε, αλλά ένα υποσύνολό τους, ώστε να επιταχύνουμε τη διαδικασία της εκπαίδευσης.

Μια ακόμα παράμετρος που επηρεάζει την εκπαίδευση του ανιχνευτή είναι η συμμετρία του αντικειμένου. Μπορούμε, δηλαδή, να ορίσουμε αν το αντικείμενο το οποίο θέλουμε να ανιχνεύεται είναι συμμετρικό ως προς τον κατακόρυφο άξονα, ή όχι. Ορίζοντας ότι το αντικείμενο είναι

συμμετρικό, ο αλγόριθμος εκπαίδευσης θα ελέγξει μόνο το αριστερό μισό των θετικών παραδειγμάτων για κάθε χαρακτηριστικό τύπου Haar και το συμμετρικό του. Έτσι ο χρόνος εκπαίδευσης μειώνεται, αλλά αν τα αντικείμενα δεν είναι εντελώς συμμετρικά, οι πληροφορίες που βρίσκονται στο δεξιό μέρος των θετικών παραδειγμάτων δεν λαμβάνονται υπόψη, με αποτέλεσμα η περιγραφή τους να μην είναι σωστή και η απόδοση της διαδικασίας να είναι χαμηλή.

Τέλος, μπορούμε να επιλέξουμε τον αλγόριθμο που θα χρησιμοποιηθεί για την εκπαίδευση, καθώς και τα χαρακτηριστικά του. Μπορούμε να ορίσουμε ποια παραλλαγή του αλγορίθμου AdaBoost θα χρησιμοποιηθεί, από τις Discrete, Real, Gentle AdaBoost και LogitBoost που είδαμε στην ενότητα 2.3. Για κάθε παραλλαγή του AdaBoost, μπορούμε να επιλέξουμε το κριτήριο σύμφωνα με το οποίο υπολογίζεται το σφάλμα ταξινόμησης. Οι επιλογές είναι οι ελάχιστες λανθασμένες ταξινομήσεις, το κριτήριο Gini και η ελάχιστη εντροπία [FrHT00]. Ακόμα, μπορούμε να καθορίσουμε αν τα αρνητικά και τα θετικά παραδείγματα θα αρχικοποιούνται με το ίδιο βάρος ή με ξεχωριστές τιμές βάρους, όπως είδαμε στην ενότητα 2.3. Τέλος, μπορούμε να ορίσουμε και την τιμή του συντελεστή ρύθμισης των βαρών των παραδειγμάτων σε κάθε κύκλο εκπαίδευσης.

### 3.3 Επιλογές εκτέλεσης

Κατά την εκτέλεση της ανίχνευσης σε μία εικόνα, υπάρχουν διάφοροι παράμετροι που επηρεάζουν είτε την ίδια τη διαδικασία, είτε μόνο τα αποτελέσματά της. Μπορούμε να επιλέξουμε τον παράγοντα κλίμακας  $sf$  με τον οποίο θα μεταβάλλεται το μέγεθος του ανιχνευτή. Δίνοντάς του τιμή κοντά στη μονάδα, θα εκτελεστεί η ανίχνευση για περισσότερα μεγέθη του ανιχνευτή. Θα εξεταστούν, δηλαδή, περισσότερα υποπαράθυρα κάθε εικόνας, με αποτέλεσμα να έχουμε περισσότερες θετικές ανιχνεύσεις (σωστές και λανθασμένες) και ο χρόνος που απαιτείται για τη διαδικασία να είναι μεγαλύτερος. Αντίθετα, δίνοντας στην παράμετρο  $sf$  μεγαλύτερη τιμή, εξετάζονται λιγότερα υποπαράθυρα, άρα μειώνονται οι θετικές ανιχνεύσεις και ο απαιτούμενος χρόνος για την ολοκλήρωση της διαδικασίας. Μπορούμε επίσης, να επιλέξουμε τον αριθμό των σταδίων του ΔΣΤ που θα χρησιμοποιηθούν. Έτσι, μπορούμε να χρησιμοποιήσουμε λιγότερα στάδια από όσα εκπαιδεύσαμε, αυξάνοντας έτσι τις θετικές ταξινομήσεις (σωστές και λανθασμένες).

Αφού ολοκληρωθεί η διαδικασία της ανίχνευσης για μια εικόνα, πραγματοποιείται ένα φιλτράρισμα των αποτελεσμάτων. Για κάθε υποπαράθυρο που ανιχνεύεται ως θετικό, μετριέται ο αριθμός των υποπαράθυρων που είναι γειτονικά με αυτό και ανιχνεύθηκαν ως θετικά. Γειτονικά θεωρούνται δύο υποπαράθυρα, όταν υπάρχει επικάλυψη των περιοχών που ορίζουν. Μπορούμε να ορίσουμε τον ελάχιστο αριθμό γειτονικών υποπαράθυρων που θα πρέπει να έχει ένα αποτέλεσμα για να επιστραφεί από τον ανιχνευτή. Έτσι, με τον αριθμό των ελάχιστων γειτόνων κάθε υποπαράθυρου μπορούμε να μεταβάλουμε τα αποτελέσματα της ανίχνευσης. Αυξάνοντας τον ελάχιστο αριθμό γειτονικών υποπαράθυρων, μειώνονται τα υποπαράθυρα που επιστρέφει το σύστημα ανίχνευσης. Το μέγεθος αυτό μπορεί να χρησιμοποιηθεί ως μέτρο εμπιστοσύνης για τα αποτελέσματα της διαδικασίας ανίχνευσης.

### 3.4 Κατάταξη αποτελεσμάτων

Τα αποτελέσματα της διαδικασίας ανίχνευσης πρέπει να μπορούμε να τα κατατάξουμε σε σωστά και λανθασμένα με αυτόματο τρόπο. Περιγράφουμε δύο μεθόδους που χρησιμοποιούνται ευρέως στη βιβλιογραφία.

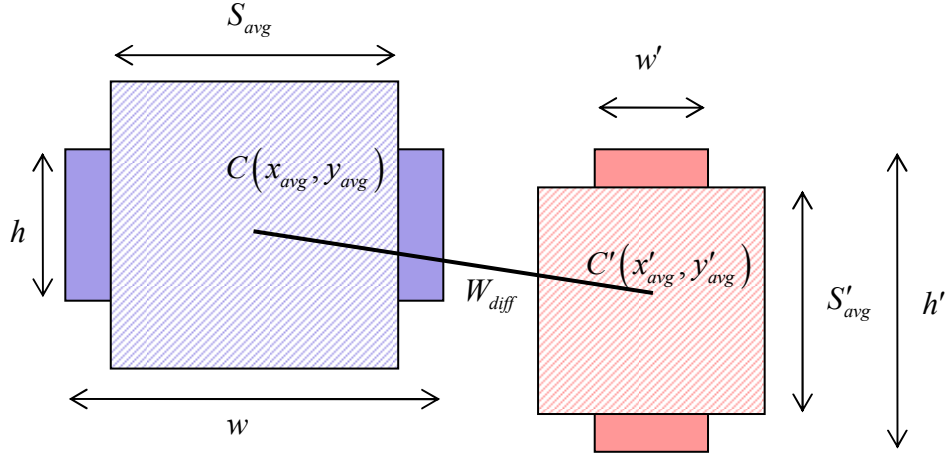
Στην πρώτη μέθοδο κατασκευάζεται ένα τετράγωνο για κάθε παραλληλόγραμμο που περιλαμβάνει ένα πραγματικό αντικείμενο της εικόνας [LiMa02]. Το τετράγωνο θα έχει πλάτος  $S_{avg}$  και κέντρο το  $C(x_{avg}, y_{avg})$  που δίνονται από τους παρακάτω τύπους:

$$\begin{aligned}x_{avg} &= x_0 + \frac{w}{2} \\y_{avg} &= y_0 + \frac{h}{2} \\S_{avg} &= \sqrt{\frac{w^2 + h^2}{2}}\end{aligned}$$

όπου  $(x_0, y_0)$  είναι το πάνω αριστερό άκρο του παραλληλογράμμου,  $w$  το πλάτος του και  $h$  το ύψος του. Για κάθε παραλληλόγραμμο που ταξινομείται ως θετικό, υπολογίζονται τα ίδια στοιχεία  $C'(x'_{avg}, y'_{avg})$  και  $S'_{avg}$ . Για να θεωρηθεί ότι το παραλληλόγραμμο που επέστρεψε ο ανιχνευτής ταυτίζεται με το πραγματικό αντικείμενο (ώστε να θεωρηθεί σωστή η ανίχνευση) θα πρέπει να ικανοποιούνται οι δύο παρακάτω σχέσεις:

$$\begin{aligned}W_{diff} &= |(x_{avg}, y_{avg}) - (x'_{avg}, y'_{avg})| < S_{avg} \cdot D_{pos} \\S'_{avg} &\in \left( \frac{S_{avg}}{D_{size}}, S_{avg} \cdot D_{size} \right)\end{aligned}$$

όπου  $D_{pos}$  και  $D_{size}$  είναι παράμετροι που ορίζουν τη μέγιστη διαφορά θέσης και μεγέθους, αντίστοιχα. Ορίζεται, δηλαδή, ότι η απόσταση  $W_{diff}$  ανάμεσα στο κέντρο του μέσου τετραγώνου του πραγματικού αντικειμένου και αυτού που ανιχνεύτηκε, θα πρέπει να είναι μικρότερη από  $D_{pos}$  φορές το μέγεθος του μέσου τετραγώνου του πραγματικού αντικειμένου. Επίσης, το μέσο μέγεθος του αντικειμένου που ανιχνεύτηκε θα πρέπει να είναι από  $\frac{1}{D_{size}}$  έως  $D_{size}$  φορές το μέσο μέγεθος του πραγματικού αντικειμένου. Μειώνοντας τις τιμές των παραμέτρων αυτών, το κριτήριο για τον χαρακτηρισμό ενός ανιχνευμένου παραλληλογράμμου ως σωστό ή όχι, γίνεται πιο αυστηρό. Αντίθετα, αυξάνοντας τις τιμές αυτές, το κριτήριο γίνεται πιο χαλαρό.

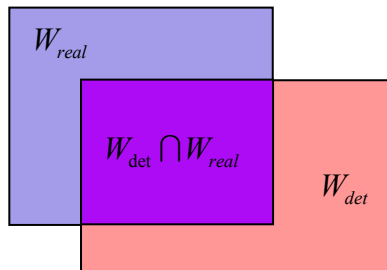


Σχήμα 3.5 - Παράδειγμα της πρώτης μεθόδου κατάταξης των αποτελεσμάτων της διαδικασίας ανίχνευσης.

Στη δεύτερη μέθοδο, μετριέται η επικάλυψη ανάμεσα στο υποπαράθυρο που επιστρέφει ο ανιχνευτής ( $W_{det}$ ) και το πραγματικό αντικείμενο ( $W_{real}$ ) [EZW+06]. Αν η επικάλυψη ξεπερνά ένα όριο που θέτουμε, τότε η ανίχνευση θεωρείται σωστή, αλλιώς θεωρείται λανθασμένη. Συγκεκριμένα, υπολογίζεται ο λόγος της περιοχής επικάλυψης προς την περιοχή που ορίζουν τα δύο υποπαράθυρα μαζί, και ελέγχεται αν ξεπερνά ή όχι το κατώφλι  $t$  που ορίζουμε. Για μια σωστή ταξινόμηση, λοιπόν, θα πρέπει:

$$\frac{|W_{det} \cap W_{real}|}{|W_{det} \cup W_{real}|} > t$$

Η τιμή που θα θέσουμε στο κατώφλι ανίχνευσης επηρεάζει σημαντικά τα αποτελέσματα που θα πάρουμε. Όσο μικρότερη είναι η τιμή, τόσο καλύτερα είναι τα αποτελέσματα για τον ανιχνευτή μας. Βέβαια, αν θέσουμε το κατώφλι πολύ χαμηλά, τα αποτελέσματα θα είναι καλά αριθμητικά, αλλά θα απέχουν από την πραγματικότητα, καθώς θα θεωρούνται σωστές, ανιχνεύσεις που έχουν πολύ μικρή επικάλυψη με το πραγματικό αντικείμενο. Έτσι, θα πρέπει να βρεθεί μια τιμή για το κατώφλι που να ανταποκρίνεται στην πραγματικότητα. Στο φαίνεται ένα υποπαράθυρο με το πραγματικό αντικείμενο ( $W_{real}$ ) και ένα υποπαράθυρο που επέστρεψε ο ανιχνευτής ( $W_{det}$ ), καθώς και η περιοχή στην οποία επικαλύπτονται.



Σχήμα 3.6 - Παράδειγμα της δεύτερης μεθόδου κατάταξης των αποτελεσμάτων της διαδικασίας ανίχνευσης.



### 3.5 Αξιολόγηση ανιχνευτή

Για την αξιολόγηση ενός ανιχνευτή μπορούν να χρησιμοποιηθούν διάφορες μετρικές σχέσεις. Σε αυτή τη διπλωματική χρησιμοποιούμε τα διαγράμματα ακρίβειας-επανάκλησης (precision-recall) και χαρακτηριστικής λειτουργίας δέκτη (ROC). Παρακάτω εξηγούμε τα μεγέθη που σχετίζονται με τα διαγράμματα αυτά.

Αρχικά, θα πρέπει να ορίσουμε τα σύνολα των πραγματικά θετικών (true positive –  $TP$ ), λανθασμένα θετικών (false positive –  $FP$ ), πραγματικά αρνητικών (true negative –  $TN$ ) και λανθασμένα αρνητικών (false negative –  $FN$ ) ταξινομήσεων. Επίσης, θα πρέπει να ορίσουμε τα σύνολα των θετικών (positive  $P$ ) και αρνητικών (negative  $N$ ) δειγμάτων, καθώς και τα σύνολα των θετικών (positive  $P'$ ) και αρνητικών (negative  $N'$ ) ταξινομήσεων. Έστω ότι δίνουμε μια εικόνα στον ανιχνευτή για την εύρεση αντικειμένων. Ο ανιχνευτής μας θα ελέγξει όλη την εικόνα, ελέγχοντας ξεχωριστά πολλά υποπαράθυρα αυτής. Κάθε υποπαράθυρο, μπορεί να περιέχει ή όχι το αντικείμενο που ψάχνουμε. Επίσης, κάθε υποπαράθυρο μπορεί να ταξινομηθεί ως υποπαράθυρο που το περιέχει το αντικείμενο, ή όχι. Αν ένα υποπαράθυρο περιέχει το αντικείμενο, και ο ανιχνευτής το εντοπίσει, τότε λέμε ότι έχουμε μια πραγματικά θετική ταξινόμηση ( $TP$ ). Αν ο ανιχνευτής δεν το εντοπίσει, τότε έχουμε μια λανθασμένα αρνητική ταξινόμηση ( $FN$ ). Το σύνολο των υποπαράθυρων που περιέχουν το αντικείμενο που ψάχνουμε είναι το σύνολο των θετικών δειγμάτων ( $P$ ), ενώ το σύνολο των υποπαράθυρων που ο ανιχνευτής θεώρησε ότι περιέχουν το αντικείμενο, είναι το σύνολο των θετικών ταξινομήσεων ( $P'$ ). Αν τώρα ο ανιχνευτής ελέγχει ένα υποπαράθυρο που δεν περιέχει το αντικείμενο που ψάχνουμε, αλλά το εντοπίσει, τότε έχουμε μια λανθασμένα θετική ταξινόμηση ( $FP$ ). Αν, σωστά, δεν το εντοπίσει, τότε έχουμε μια πραγματικά αρνητική ταξινόμηση ( $TN$ ). Το σύνολο όλων των υποπαράθυρων που δεν περιέχουν το αντικείμενο ονομάζεται σύνολο αρνητικών δειγμάτων ( $N$ ), ενώ το σύνολο των υποπαράθυρων που ο ανιχνευτής απέρριψε, θεωρώντας ότι δεν περιέχουν το αντικείμενο, ονομάζεται σύνολο αρνητικών ταξινομήσεων ( $N'$ ). Τα σύνολα αυτά, συνδέονται μεταξύ τους. Γραφικά μπορούμε να τα αναπαραστήσουμε με τον πίνακα σύγχυσης (Confusion Matrix), που φαίνεται στον Πίνακας 3.1.

Πλήθος υποπαράθυρων ( $T$ )	$P$	$N$
$P'$	True Positive ( $TP$ )	False Positive ( $FP$ )
$N'$	False Negative ( $FN$ )	True Negative ( $TN$ )

Πίνακας 3.1 - Πίνακας Σύγχυσης (Confusion Matrix)

Τα πλήθη των συνόλων που περιγράψαμε προηγουμένως συνδέονται με τις παρακάτω σχέσεις:

$$|TP| + |FP| = |P'|$$

$$|FN| + |TN| = |N'|$$

$$|TP| + |FN| = |P|$$

$$|FP| + |TN| = |N|$$

$$|P| + |N| = T$$

$$|P'| + |N'| = T$$

Αν ορίζαμε αυστηρά τα υποπαράθυρα που περιέχουν τα αντικείμενα που ψάχνουμε, θα έπρεπε κάθε αντικείμενο να αντιστοιχεί σε ένα μόνο υποπαράθυρο. Έτσι, το πλήθος του συνόλου P θα ήταν ίσο με τον αριθμό των αντικειμένων στην εικόνα και το πλήθος του συνόλου N θα ήταν ίσο με το πλήθος όλων των υποπαράθυρων της εικόνας μείον το πλήθος των αντικειμένων. Στην ανίχνευση αντικειμένων, όμως, δε χρησιμοποιούμε τόσο αυστηρούς ορισμούς, οπότε θεωρούμε μια ανίχνευση σωστή, ακόμα και αν το υποπαράθυρο που επιστρέφεται απέχει λίγο από το πραγματικό αντικείμενο ή το μέγεθός του είναι λίγο διαφορετικό. Έτσι, για κάθε αντικείμενο που ανήκει στην εικόνα, δημιουργείται ένα σύνολο υποπαράθυρων στην περιοχή του αντικειμένου, που αν ταξινομηθούν ως θετικά από τον ταξινομητή θα θεωρήσουμε ότι έγινε σωστή ανίχνευση.

Οι μετρικές σχέσεις που χρησιμοποιούμε για να αξιολογήσουμε την απόδοση ενός ταξινομητή είναι η ακρίβεια (precision), η επανάκληση ή λόγος πραγματικών θετικών ταξινομήσεων (recall, true positive rate – *TPR*) και ο λόγος λανθασμένων θετικών ταξινομήσεων (false positive rate – *FPR*). Η ακρίβεια, ισούται με το λόγο του πλήθους των πραγματικά θετικών ταξινομήσεων προς το πλήθος όλων των θετικών ταξινομήσεων. Εκφράζει, δηλαδή, το ποσοστό των θετικών ανιχνεύσεων που είναι σωστές, και υπολογίζεται με τους παρακάτω τύπους:

$$precision = \frac{|TP|}{|TP| + |FP|}, \text{ ή}$$

$$precision = \frac{|TP|}{|P'|}$$

Η επανάκληση ισούται με τον λόγο του πλήθους των πραγματικά θετικών ταξινομήσεων προς το πλήθος όλων των θετικών δειγμάτων και εκφράζει το ποσοστό των θετικών δειγμάτων που ανιχνεύθηκαν. Για τον υπολογισμό της επανάκλησης, χρησιμοποιούμε τους παρακάτω τύπους:

$$recall = \frac{|TP|}{|TP| + |FN|}, \text{ ή}$$

$$recall = \frac{|TP|}{|P|}$$

Ο λόγος λανθασμένων θετικών ταξινομήσεων (*FPR*), ισούται με το λόγο των λανθασμένα θετικών ταξινομήσεων προς το πλήθος όλων των αρνητικών δειγμάτων. Με τη μετρική αυτή σχέση, εκφράζουμε το ποσοστό των αρνητικών υποπαράθυρων που ταξινομήθηκαν ως θετικά. Για τον υπολογισμό του *FPR*, χρησιμοποιούμε τις παρακάτω σχέσεις:

$$FPR = \frac{|FP|}{|FP| + |TN|}, \text{ ή}$$

$$FPR = \frac{|FP|}{|N|}$$

Είδαμε, ότι το πλήθος των θετικών υποπαράθυρων είναι μεγαλύτερο από τον αριθμό των εμφανίσεων ενός αντικειμένου σε μια εικόνα. Το πλήθος των αρνητικών υποπαράθυρων, θα ισούται με το πλήθος όλων των υποπαράθυρων που εξετάζονται, μείον τα υποπαράθυρα που θεωρήσαμε ως θετικά. Για να προσδιορίσουμε τον αριθμό των θετικών και αρνητικών υποπαράθυρων, θα πρέπει να βρούμε όλα τα υποπαράθυρα που θα εξεταστούν από τον ταξινομητή μας. Ο αριθμός αυτός εξαρτάται από το μέγεθος της εικόνας, το μέγεθος του αρχικού παραθύρου ανίχνευσης και τον παράγοντα κλίμακας που χρησιμοποιούμε κατά την ανίχνευση. Έπειτα, θα πρέπει να υπολογίσουμε τον αριθμό των θετικών υποπαράθυρων. Ο αριθμός αυτός εξαρτάται από τον αριθμό των αντικειμένων στην εικόνα, το μέγεθός τους και τις τιμές των παραμέτρων που χρησιμοποιήσαμε για την κατάταξη των αποτελεσμάτων του ανιχνευτή.

Εφαρμόζοντας τον ανιχνευτή σε ένα σύνολο εικόνων δοκιμής, θα έχουμε έναν αριθμό θετικών ταξινομήσεων και έναν αριθμό αρνητικών (κάποιες από τις οποίες θα είναι σωστές και οι υπόλοιπες λανθασμένες). Έτσι μπορούμε να υπολογίσουμε μια τιμή ακρίβειας, μια τιμή επανάκλησης και μια τιμή για το *FPR*. Από μία τιμή όμως δε μπορούμε να λάβουμε ασφαλή συμπεράσματα για την απόδοση του ανιχνευτή μας. Θα χρειαστούμε, λοιπόν, ένα σύνολο τιμών για κάθε μετρική σχέση, ώστε να κατασκευάσουμε διαγράμματα που θα παρουσιάζουν καλύτερα την απόδοση του ανιχνευτή. Για να λάβουμε αυτό το σύνολο τιμών θα πρέπει να μεταβάλλουμε κάποια παράμετρο της διαδικασίας ανίχνευσης.

Μια επιλογή είναι να μεταβάλλουμε τον αριθμό των σταδίων του ταξινομητή που χρησιμοποιούνται για την ανίχνευση. Έτσι, θα πρέπει να εκτελέσουμε πολλές φορές όλη τη διαδικασία της ανίχνευσης. Αρχικά χρησιμοποιούμε όλα τα στάδια του ταξινομητή, και σε κάθε επόμενη εκτέλεση χρησιμοποιούμε ένα στάδιο λιγότερο. Σε κάθε εφαρμογή του ανιχνευτή, μειώνοντας τον αριθμό των σταδίων του ταξινομητή, θα αυξάνεται ο αριθμός των θετικά ταξινομημένων υποπαράθυρων. Έτσι, θα προκύψει ένα σύνολο τιμών για κάθε μετρική σχέση, οπότε θα είμαστε σε θέση να κατασκευάσουμε διαγράμματα για να αξιολογήσουμε την απόδοση του ανιχνευτή. Η διαδικασία αυτή, όμως, είναι αρκετά χρονοβόρα, καθώς θα πρέπει να εκτελεστεί όλη η διαδικασία της ανίχνευσης πολλές φορές.

Μια άλλη επιλογή, είναι να μεταβάλλουμε την τιμή των ελάχιστων γειτόνων. Ξεκινάμε εκτελώντας την ανίχνευση με ελάχιστο αριθμό γειτόνων ίσο με ένα, και σε κάθε βήμα αυξάνουμε την τιμή κατά ένα. Αυξάνοντας τον αριθμό των ελάχιστων γειτόνων που θα πρέπει να έχει ένα υποπαράθυρο που περνά από όλα τα στάδια του ταξινομητή, ώστε να ταξινομηθεί ως θετικό, ο τελικός αριθμός των θετικά ταξινομημένων υποπαράθυρων θα μειώνεται. Το πλεονέκτημα αυτής της μεθόδου, είναι ότι δε χρειάζεται να εκτελέσουμε όλη τη διαδικασία ανίχνευσης πολλές φορές. Αρκεί από το σύνολο των θετικών ανιχνεύσεων που προέκυψαν για την αρχική τιμή των ελάχιστων γειτόνων, να αφαιρούμε τα υποπαράθυρα που δεν έχουν τον απαιτούμενο αριθμό γειτόνων κάθε φορά. Έτσι, αρκεί να εκτελέσουμε μία φορά τη διαδικασία της ανίχνευσης για όλες τις εικόνες και να υπολογίσουμε όλες τις μετρικές σχέσεις για κάθε τιμή των ελάχιστων γειτόνων.

Έχοντας ένα σύνολο τιμών κάθε μετρικής σχέσης, μπορούμε να κατασκευάσουμε τις καμπύλες ακρίβειας–επανάκλησης (precision–recall) και ROC. Η καμπύλη ακρίβειας–επανάκλησης μας δείχνει την ακρίβεια της διαδικασίας ανίχνευσης για κάθε τιμή επανάκλησης που μπορούμε να πετύχουμε. Γενικά, καθώς αυξάνεται η επανάκληση, μειώνεται η ακρίβεια του ανιχνευτή, και αντίστροφα. Έτσι, θα πρέπει παρατηρώντας την καμπύλη αυτή, να επιλέξουμε ένα συνδυασμό ακρίβειας και επανάκλησης που μας ικανοποιεί.

Η καμπύλη χαρακτηριστικής λειτουργίας δέκτη (ROC) μας δείχνει την τιμή επανάκλησης που μπορούμε να πετύχουμε για κάθε τιμή του λόγου λανθασμένων θετικών ταξινομήσεων. Εκφράζει, δηλαδή, τη μεταβολή των πραγματικών θετικών ταξινομήσεων, όταν μεταβάλλεται ο αριθμός των λανθασμένων θετικών ταξινομήσεων. Γενικά, καθώς αυξάνεται ο αριθμός των λανθασμένων θετικών ταξινομήσεων, αυξάνεται και ο αριθμός των πραγματικών θετικών ταξινομήσεων. Η καμπύλη ROC ενός ταξινομητή που χρησιμοποιεί την τύχη για την ταξινόμηση των δειγμάτων, θα είναι μια ευθεία γραμμή από το σημείο  $(0,0)$ , στο σημείο  $(1,1)$ . Ένας ταξινομητής A είναι καλύτερος από έναν ταξινομητή B, αν η καμπύλη του A βρίσκεται επάνω και αριστερά της καμπύλης του B. Έτσι, η καμπύλη ROC κάθε ταξινομητή, που δεν απαντά τυχαία, θα πρέπει να βρίσκεται στο επάνω αριστερό τμήμα του χώρου  $[0,0] \times [1,1]$ .

## Κεφάλαιο 4.

### Πειραματικά αποτελέσματα

Σε αυτή τη διπλωματική, εφαρμόσαμε τη μέθοδο ανίχνευσης που παρουσιάσαμε, σε διαφορετικά σύνολα εικόνων. Αρχικά, ασχοληθήκαμε με τον εντοπισμό αντικειμένων από το εσωτερικό ενός αυτοκινήτου. Επιλέξαμε τέσσερα βασικά αντικείμενα του εσωτερικού ενός αυτοκινήτου και εκπαιδεύσαμε ανιχνευτές για κάθε κατηγορία. Έπειτα ασχοληθήκαμε με το εξωτερικό μέρος του αυτοκινήτου. Εκπαιδεύσαμε ανιχνευτές για τέσσερις διαφορετικές οπτικές γωνίες του αμαξώματος ενός αυτοκινήτου. Παρακάτω περιγράφουμε κάθε κατηγορία αντικειμένων με την οποία ασχοληθήκαμε, δίνουμε παραδείγματα αυτών, περιγράφουμε τις επιλογές που κάναμε για την εκπαίδευση του αντίστοιχου ανιχνευτή και τέλος παρουσιάζουμε και σχολιάζουμε την απόδοση κάθε ανιχνευτή.

#### 4.1 Εσωτερικό αυτοκινήτου

Το πρώτο σύνολο εικόνων που χρησιμοποιήσαμε περιέχει εικόνες από το εσωτερικό μέρος αυτοκινήτων. Είχαμε στη διάθεσή μας περίπου 750 εικόνες. Εξετάζοντας τη συλλογή, επιλέξαμε να εκπαιδεύσουμε ανιχνευτές για το τιμόνι, τους αεραγωγούς, τον μοχλό ταχυτήτων και τα πεντάλ. Για την εκπαίδευση χρησιμοποιήσαμε τις 690 από αυτές, ενώ οι υπόλοιπες 60 χρησιμοποιήθηκαν για τη μέτρηση της απόδοσης των ανιχνευτών. Ο αριθμός των εικόνων φαίνεται σχετικά μικρός, συγκρινόμενος με τις χιλιάδες εικόνων που χρησιμοποιήθηκαν στη βιβλιογραφία. Θα δούμε, όμως, ότι δεδομένης της δυσκολίας του προβλήματος, μπορέσαμε να κατασκευάσουμε ικανοποιητικής απόδοσης ανιχνευτές.

Αρχικά, χρειάστηκε να χαρακτηρίσουμε όλες τις εικόνες που είχαμε στη διάθεσή μας. Χρησιμοποιήσαμε το πρόγραμμα χαρακτηρισμού εικόνων που κατασκευάσαμε, το οποίο θα παρουσιαστεί αναλυτικά στο Κεφάλαιο 5. Η διαδικασία του χαρακτηρισμού είναι αριετάρικη, καθώς θα πρέπει να χαρακτηριστούν 750 εικόνες με αριετάρικα αντικείμενα η κάθε μία.

##### 4.1.1 Τιμόνια αυτοκινήτων

Το πρώτο αντικείμενο με το οποίο ασχοληθήκαμε ήταν το τιμόνι αυτοκινήτου. Σε κάθε εικόνα υπήρχε ένα τιμόνι, άρα είχαμε στη διάθεσή μας 690 εικόνες με τιμόνια για την εκπαίδευση. Αφαιρώντας κάποιες εικόνες, όπου τα τιμόνια κρύβονταν από άλλα αντικείμενα, καταλήξαμε σε 686 θετικά παραδείγματα. Τα παραδείγματα αυτά, αν και φαίνονται αριετάρικα, δεν ήταν πολύ ομοιογενή. Υπήρχαν ανομοιότητες στη μορφή των τιμονιών, καθώς υπήρχαν τιμόνια με δύο, τρεις και τέσσερις ακτίνες. Επίσης, υπήρχαν σημαντικές διαφορές στη θέση στην οποία εμφανιζόταν κάθε χαρακτηριστικό τμήμα του τιμονιού, καθώς υπήρχαν τιμόνια έντονα περιστραμμένα. Έτσι,

είναι δύσκολο να επιλεγούν τα χαρακτηριστικά τύπου Haar που θα χρησιμοποιηθούν για την ταξινόμηση των τιμονιών, καθώς τα δείγματα δεν έχουν τα ίδια γεωμετρικά χαρακτηριστικά στις ίδιες θέσεις. Μερικά αντιπροσωπευτικά παραδείγματα τιμονιών φαίνονται στο Σχήμα 4.1.

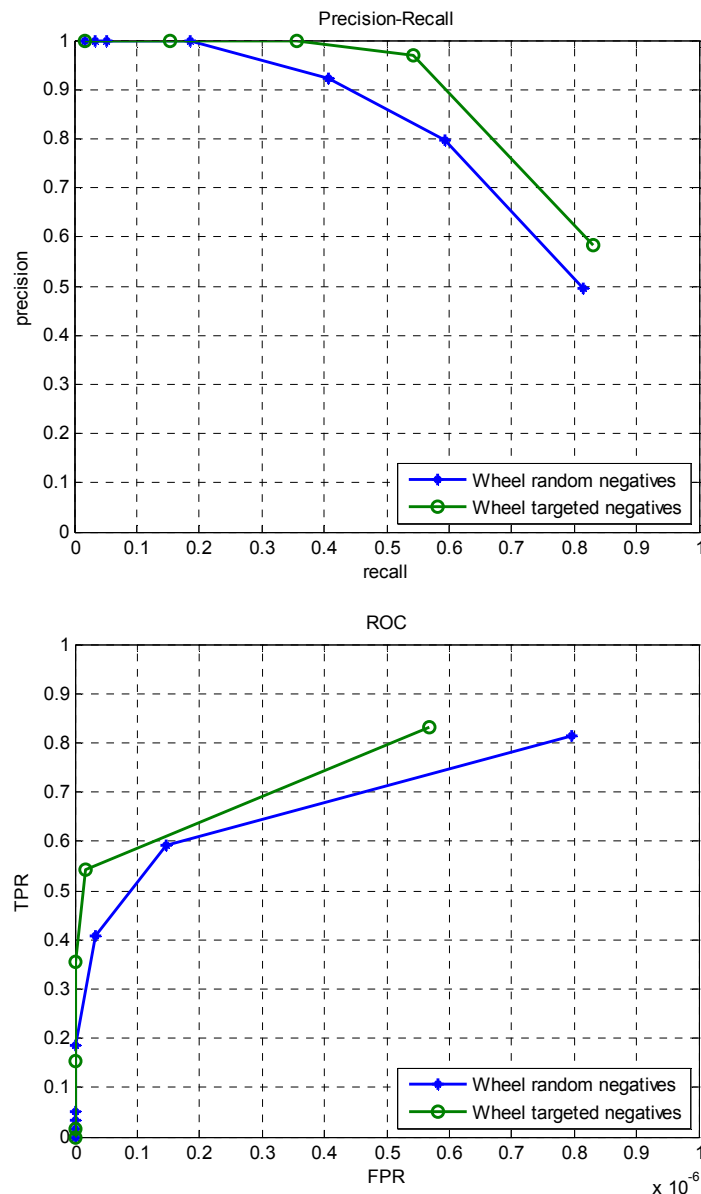


Σχήμα 4.1 - Παραδείγματα της κατηγορίας τιμόνια. Φαίνεται η ποικιλία των μορφών και των γωνιών περιστροφής.

Εκπαιδεύσαμε δύο ανιχνευτές για τα τιμόνια. Επιλέξαμε να έχουν και οι δύο ανιχνευτές από 17 στάδια. Σε κάθε στάδιο θέσαμε τον ελάχιστο λόγο εντοπισμού στο 0,999. Η τιμή αυτή είναι αρκετά υψηλή, ώστε να πετύχουμε υψηλό λόγο ανίχνευσης στο τέλος. Θέσαμε το μέγιστο λόγο λανθασμένων ανιχνεύσεων στο 0,5. Η τιμή αυτή είναι αρκετά υψηλή, αλλά χρησιμοποιώντας αρκετά στάδια (πάνω από 10) ο συνολικός λόγος λανθασμένων ανιχνεύσεων θα είναι αρκετά χαμηλός. Ακόμα, επειδή η τιμή είναι υψηλή, ο ανιχνευτής θα έχει λίγους ασθενείς ταξινομητές ανά στάδιο, που σημαίνει ότι θα έχει αυξημένη ταχύτητα. Επιλέξαμε να χρησιμοποιήσουμε όλο το σύνολο των χαρακτηριστικών τύπου Haar που παρουσιάστηκε στην ενότητα 2.1, ώστε η περιγραφή των γεωμετρικών χαρακτηριστικών των αντικειμένων να είναι πιο ακριβής. Δε χρησιμοποιήσαμε την επιλογή της συμμετρίας, ώστε η εκπαίδευση να γίνει σε όλο το πλάτος των αντικειμένων. Τέλος, επιλέξαμε κάθε ασθενής ταξινομητής να αποτελεί ένα δέντρο με δύο χαρακτηριστικά.

Η μόνη διαφορά ανάμεσα στους δύο ανιχνευτές, βρίσκεται στο σύνολο των αρνητικών παραδειγμάτων. Ο πρώτος ανιχνευτής είχε ως αρνητικά παραδείγματα διάφορες τυχαίες εικόνες από το διαδίκτυο που δεν περιείχαν τιμόνια, αλλά ανθρώπους, δέντρα, κτίρια κ.α. Ένας ανιχνευτής για τιμόνια αυτοκινήτων, θα χρησιμοποιείται σε φωτογραφίες που απεικονίζουν το εσωτερικό ενός αυτοκινήτου. Έτσι, επιλέξαμε να χρησιμοποιήσουμε για την εκπαίδευση του δεύτερου ταξινομητή, αρνητικά παραδείγματα από το εσωτερικό των αυτοκινήτων από τα οποία "κρύψαμε" τα τιμόνια. Ο πρώτος ανιχνευτής, λοιπόν, έχει εκπαιδευτεί με τυχαία αρνητικά παραδείγματα, ενώ ο δεύτερος με στοχευμένα στο εσωτερικό των αυτοκινήτων.

Η απόδοση των δύο ανιχνευτών, μετρήθηκε σε ένα σύνολο 59 εικόνων με εσωτερικά αυτοκινήτων ανάλυσης 750x500 εικονοστοιχείων. Στο Σχήμα 4.2 βλέπουμε τις καμπύλες ακρίβειας-επανάκλησης, καθώς και τις καμπύλες ROC των δύο ανιχνευτών.



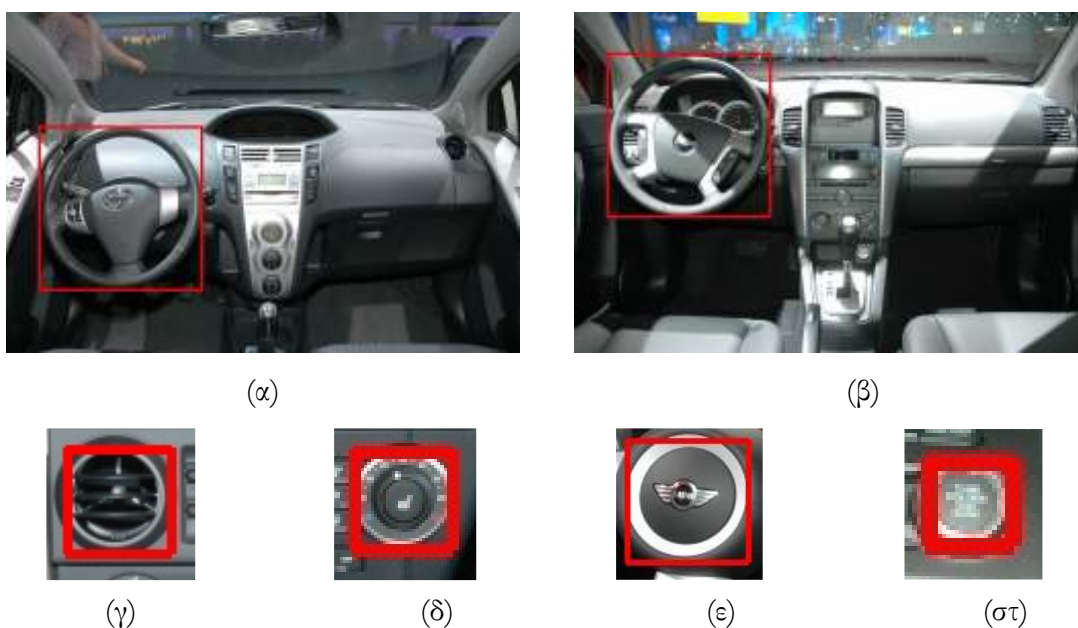
Σχήμα 4.2 - Καμπύλες ακρίβειας-επανάκλησης και ROC των ανιχνευτών για τα τιμόνια αυτοκινητών

Βλέπουμε ότι ο ανιχνευτής που εκπαιδεύσαμε με αρνητικά παραδείγματα από το εσωτερικό τμήμα αυτοκινήτων αποδίδει καλύτερα από ό,τι ο ανιχνευτής που εκπαιδεύσαμε με τυχαία αρνητικά παραδείγματα. Η ακρίβεια που πετυχαίνουμε για κάθε τιμή επανάκλησης είναι αρκετά υψηλή, όπως φαίνεται στη γραφική παράσταση. Από τις καμπύλες ROC βλέπουμε ότι μπορούμε να πετύχουμε υψηλή επανάκληση (της τάξης του 0,8) για πολύ χαμηλούς λόγους λανθασμένων θετικών ταξινομήσεων. Ο λόγος λανθασμένων θετικών ταξινομήσεων της τάξης του  $10^{-6}$ , σημαίνει ότι παρουσιάζεται μια λανθασμένη θετική ταξινόμηση ανά περίπου 1.000.000 υποπαράθυρα που ελέγχονται. Ο λόγος αυτός είναι πολύ χαμηλός, και δείχνει την καλή ποιότητα του ταξινομητή μας.

Ο χρόνος ανίχνευσης ήταν παραπλήσιος για τους δύο ανιχνευτές. Ο πρώτος ανιχνευτής (εκπαιδευμένος με τυχαία αρνητικά παραδείγματα) χρειάστηκε 18,6 δευτερόλεπτα για να ανιχνεύσει τα τιμόνια σε 59 εικόνες ανάλυσης 750x500 εικονοστοιχείων, χρειάστηκε δηλαδή,

περίπου 0,31 δευτερόλεπτα για κάθε εικόνα. Ο δεύτερος ανιχνευτής (εμπαιδευμένος με αρνητικά παραδείγματα από το εσωτερικό μέρος αυτοκινήτων) χρειάστηκε 16,7 δευτερόλεπτα για να ανιχνεύσει τα τιμόνια στις ίδιες εικόνες. Χρειάστηκε δηλαδή, περίπου 0,28 δευτερόλεπτα για κάθε εικόνα. Δεδομένης της μεγάλης ανάλυσης των εικόνων, ο χρόνος των δύο ανιχνευτών θεωρείται αρκετά χαμηλός. Συγκεκριμένα, καθένas από τους δύο ανιχνευτές χρειάστηκε να ελέγξει 61.588.209 υποπαράθυρα. Έτσι, προκύπτει ότι ο πρώτος ανιχνευτής χρειάζεται κατά μέσο όρο  $3,0 \cdot 10^{-7}$  δευτερόλεπτα για να ελέγξει κάθε υποπαράθυρο, ενώ ο δεύτερος  $2,7 \cdot 10^{-7}$  δευτερόλεπτα.

Στο Σχήμα 4.3 βλέπουμε κάποια αποτελέσματα της ανίχνευσης στο σύνολο εικόνων δοκιμής. Τα περισσότερα τιμόνια εντοπίστηκαν σωστά όπως φαίνεται στις εικόνες (α) και (β) στο Σχήμα 4.3. Κατά την ανίχνευση προέκυψαν και κάποια λανθασμένα αποτελέσματα, όπως αυτά που φαίνονται στις εικόνες (γ)-(στ) στο Σχήμα 4.3. Στην εικόνα (γ) φαίνεται ένας αεραγωγός, στην (ε) ένα τμήμα τιμονιού και στις (δ) και (στ) κάποια κουμπιά της κεντρικής κονσόλας. Βλέπουμε ότι τα γεωμετρικά χαρακτηριστικά των περιοχών που λανθασμένα θεωρήθηκαν ότι εμφανίζουν ένα τιμόνι, ταιριάζουν σε μεγάλο βαθμό με αυτά των τιμονιών (το σχήμα είναι κυκλικό, στο κέντρο υπάρχει αλλαγή στη φωτεινότητα, κ.α.).



Σχήμα 4.3 - Αποτελέσματα της ανίχνευσης τιμονιών. Στις εικόνες (α), (β) βλέπουμε σωστές ανιχνεύσεις χωρίς λάθη, ενώ στις (γ)-(στ) λανθασμένες ανιχνεύσεις.

#### 4.1.2 Αεραγωγοί

Το δεύτερο αντικείμενο με το οποίο ασχοληθήκαμε ήταν οι αεραγωγοί αυτοκινήτων. Σε κάθε μία από τις 690 εικόνες που είχαμε στη διάθεσή μας, υπήρχαν από ένας έως και οχτώ αεραγωγοί. Έτσι, το συνολικό πλήθος των θετικών παραδειγμάτων για τους αεραγωγούς ήταν 2909. Το πλήθος αυτό είναι αρκετά μεγάλο για να κατασκευαστεί ένας καλός ταξινομητής. Τα παραδείγματα, όμως,



παρουσίαζαν πολύ έντονες διαφορές μεταξύ τους ως προς το σχήμα, τη μορφή και το λόγο διαστάσεων. Υπήρχαν αεραγωγοί παραλληλόγραμμοι και κυκλικοί, κατακόρυφοι και οριζόντιοι, ανοιχτοί και κλειστοί. Παραδείγματα τυπικών δειγμάτων φαίνονται στο Σχήμα 4.4. Με τόσο έντονη ανομοιομορφία στα δείγματα, περιμένουμε ότι η διαδικασία της επιλογής των χαρακτηριστικών τύπου Haar θα είναι δύσκολη.



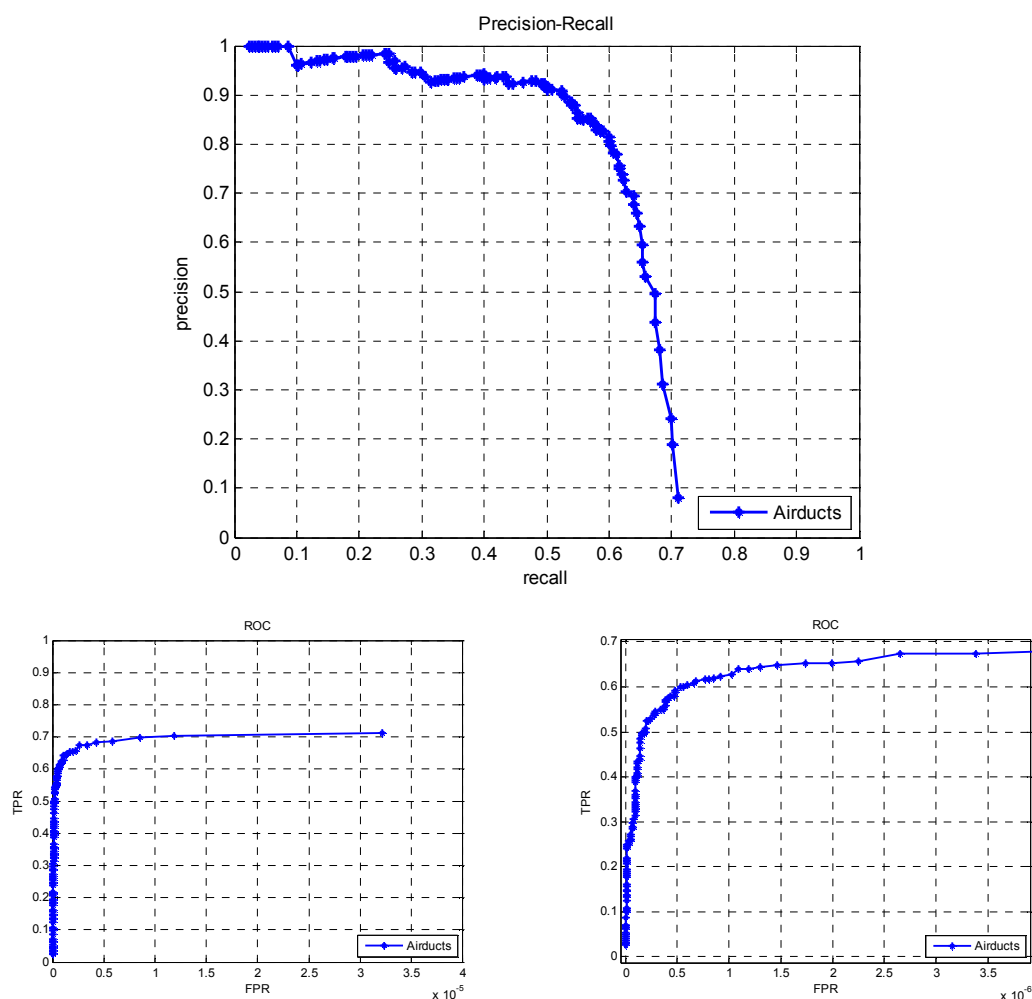
Σχήμα 4.4 - Παραδείγματα εικόνων αεραγωγών

Επιλέξαμε να εκπαιδεύσουμε έναν ταξινομητή αποτελούμενο από 15 στάδια. Δεδομένης της πολύ έντονης ανομοιογένειας των δειγμάτων, επιλέξαμε να χρησιμοποιήσουμε πολύ χαμηλό μέγιστο λόγο λανθασμένων ανιχνεύσεων. Έτσι, θέσαμε την τιμή του στο 0,025. Σε συνδυασμό με τα 15 στάδια του ταξινομητή, αναμένεται να ξεπεραστεί η έντονη ανομοιογένεια των δειγμάτων, και να είναι περιορισμένος ο αριθμός των λανθασμένων θετικών ανιχνεύσεων. Αντίθετα, χρησιμοποιώντας τόσο χαμηλό λόγο λανθασμένων ανιχνεύσεων σε 15 στάδια, ο ταξινομητής αναμένεται να είναι πιο αργός συγκριτικά με τους ανιχνευτές για τα τιμόνια. Ο ελάχιστος λόγος εντοπισμού επιλέχθηκε και πάλι πολύ υψηλός, ίσως με 0,999, ώστε να επιτύχουμε και υψηλό συνολικό λόγο εντοπισμού. Χρησιμοποιήσαμε το πλήρες σύνολο χαρακτηριστικών τύπου Haar και επιλέξαμε ασυμμετρία στην εκπαίδευση, ώστε να πετύχουμε το καλύτερο δυνατό αποτέλεσμα. Τα αρνητικά παραδείγματα αντλήθηκαν από εικόνες του εσωτερικού μέρους αυτοκινήτων, που δεν περιείχαν κανέναν αεραγωγό.

Η απόδοση του ανιχνευτή μετρήθηκε στο ίδιο σύνολο εικόνων που χρησιμοποιήθηκε και για τα τιμόνια. Στο Σχήμα 4.5 βλέπουμε τις καμπύλες ακρίβειας–επανάκλησης και ROC του ανιχνευτή. Βλέπουμε ότι η επανάκληση του ανιχνευτή αεραγωγών είναι χαμηλότερη από ό,τι του ανιχνευτή τιμονιών, γεγονός που οφείλεται στη πολύ μεγάλη ανομοιογένεια των εικόνων αεραγωγών. Αντίθετα, για τιμές χαμηλότερης επανάκλησης (περίπου 0,5), βλέπουμε ότι μπορούμε να επιτύχουμε πολύ υψηλή ακρίβεια στις ανιχνεύσεις(περίπου 0,9), γεγονός που οφείλεται στο μεγάλο αριθμό θετικών παραδειγμάτων που είχαμε στη διάθεσή μας. Παρατηρώντας την καμπύλη ROC, βλέπουμε ότι ο λόγος λανθασμένων θετικών ταξινομήσεων παραμένει πολύ χαμηλός (κάτω από  $10^{-5}$ ), και αυξάνεται μόνο όταν ξεπεράσουμε την τιμή 0,7 της επανάκλησης.

Ο χρόνος ανίχνευσης ήταν, όπως αναμενόταν, μεγαλύτερος από το χρόνο που χρειάστηκε για να ανιχνευτούν τα τιμόνια. Συγκεκριμένα, για να ανιχνευτούν οι αεραγωγοί στις 59 εικόνες ανάλυσης 750x500 εικονοστοιχείων που χρησιμοποιήσαμε για τα τιμόνια, χρειάστηκαν 114,5 δευτερόλεπτα, χρόνος περίπου εφτά φορές περισσότερος. Για την ανίχνευση των αεραγωγών σε κάθε εικόνα, δηλαδή, ο ανιχνευτής χρειάζεται περίπου 2 δευτερόλεπτα. Ο χρόνος αυτός είναι αρκετά υψηλός, αλλά δικαιολογείται από την μεγάλη ανομοιογένεια των θετικών παραδειγμάτων εκπαίδευσης, που είχε ως αποτέλεσμα την κατασκευή ενός πολύπλοκου ταξινομητή. Συγκεκριμένα, ο ανιχνευτής χρειάστηκε να ελέγξει 62.199.303 υποπαράθυρα για να ολοκληρώσει τη διαδικασία

της ανίχνευσης. Έτσι, προκύπτει ότι για κάθε υποπαράθυρο απαιτούσε κατά μέσο όρο  $18,4 \times 10^{-7}$  δευτερόλεπτα. Ο χρόνος αυτός είναι περίπου 6 φορές περισσότερος από το χρόνο που απαιτούσε ο ανιχνευτής για τα τιμόνια.



Σχήμα 4.5 - Καμπύλη ακρίβειας-επανάληψης και ROC του ανιχνευτή για τους αεραγωγούς αυτοκινητών

Στο Σχήμα 4.6 βλέπουμε κάποια αποτελέσματα από τη διαδικασία ανίχνευσης αεραγωγών στις εικόνες του συνόλου δοκιμής. Οι περισσότεροι αεραγωγοί εντοπίστηκαν, χωρίς να έχουμε πολλές λανθασμένους ανιχνεύσεις, όπως φαίνεται στις εικόνες (α) και (β) στο Σχήμα 4.6. Στις εικόνες (γ)-(ζ) στο Σχήμα 4.6 φαίνονται κάποιες από τις λανθασμένες ανιχνεύσεις. Συγκεκριμένα, στην εικόνα (α) φαίνεται ένα πεντάλ, στη (β) ένα κομμάτι από πατάκι, στην (ε) ένα τμήμα τοίχου, στην (στ) ένα τμήμα ενός μοχλού ταχυτήτων και στην (ζ) ένα κουμπί της κεντρικής κονσόλας. Παρότι οι περιοχές που ανιχνεύθηκαν δεν περιέχουν κανέναν αεραγωγό, βλέπουμε ότι τα γεωμετρικά χαρακτηριστικά που χαρακτηρίζουν τις περιοχές αυτές μοιάζουν πάρα πολύ με αυτά των αεραγωγών. Σε κάποιες περιπτώσεις ακόμα και το ανθρώπινο μάτι θα μπορούσε να κάνει λάθος, θεωρώντας ότι εξετάζει μόνο το συγκεκριμένο υποπαράθυρο της εικόνας, και όχι την εικόνα στο σύνολό της. Βλέπουμε, δηλαδή, ότι οι εσφαλμένες ανιχνεύσεις αεραγωγών είναι, σε ένα βαθμό, λογικές και αναμενόμενες.



(α)



(β)



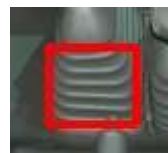
(γ)



(δ)



(ε)



(στ)



(ζ)

Σχήμα 4.6 - Αποτελέσματα της ανίχνευσης αεραγωγών. Στις εικόνες (α), (β) βλέπουμε σωστές ανιχνεύσεις χωρίς λάθη, ενώ στις (γ)-(ζ) λανθασμένες ανιχνεύσεις.

### 4.1.3 Μοχλός ταχυτήτων

Ένα άλλο αντικείμενο από το εσωτερικό ενός αυτοκινήτου που μας απασχόλησε, ήταν οι μοχλοί ταχυτήτων. Από τις εικόνες που είχαμε στη διάθεσή μας για την εκπαίδευση του ταξινομητή, προέκυψαν 600 θετικά παραδείγματα. Τα παραδείγματα δεν έχουν όλα το ίδιο σχήμα, αλλά είναι γενικά αρκετά ομοιογενή. Στο Σχήμα 4.7 βλέπουμε κάποια παραδείγματα που χρησιμοποιήθηκαν στην εκπαίδευση του ταξινομητή. Παρατηρώντας τα παραδείγματα αυτά, βλέπουμε ότι ο μοχλός ταχυτήτων δεν αποτελείται από πολλά γεωμετρικά χαρακτηριστικά.



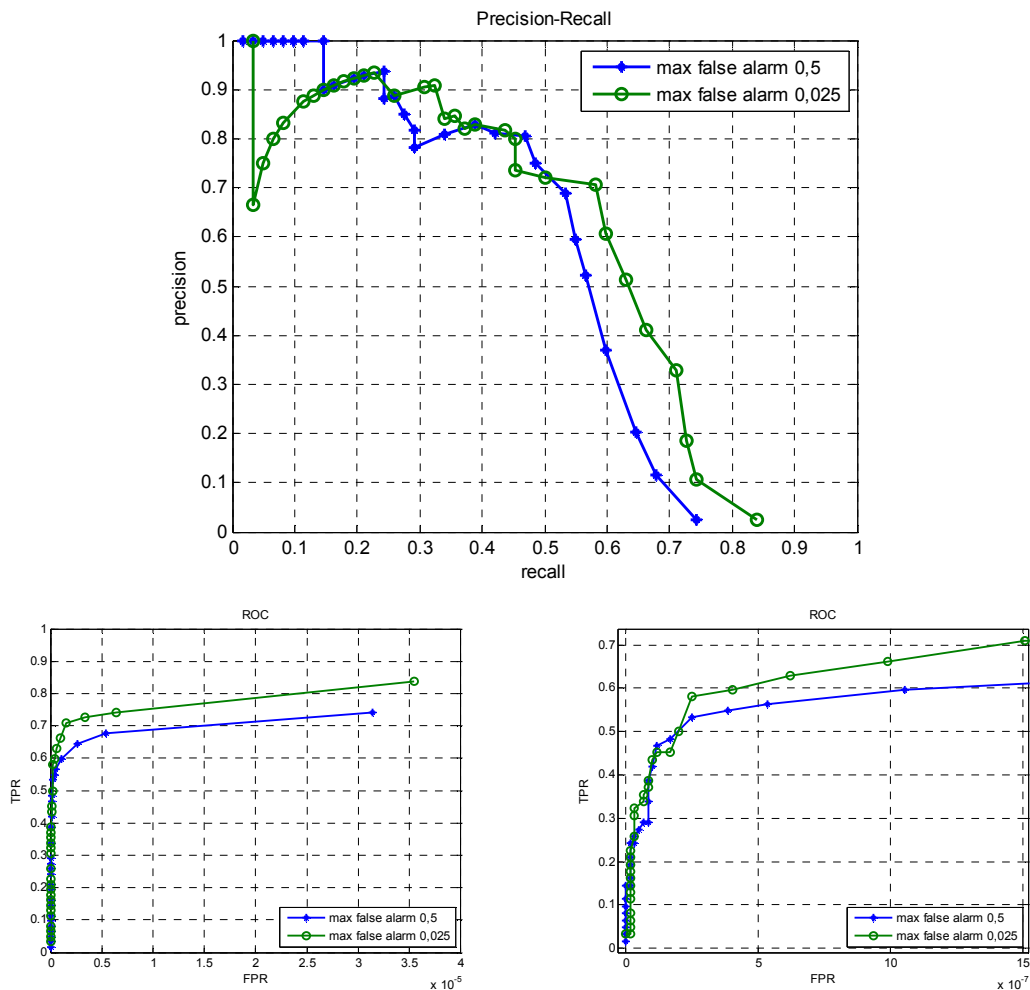
Σχήμα 4.7 - Θετικά παραδείγματα εκπαίδευσης του αντικειμένου μοχλός ταχυτήτων

Συνολικά, εκπαιδεύσαμε δύο ταξινομητές για τους μοχλούς ταχυτήτων. Και οι δύο ταξινομητές είχαν ελάχιστο λόγο εντοπισμού 0,999, τιμή αρκετά υψηλή, ώστε να μας δώσει υψηλό λόγο ανίχνευσης στο τέλος. Χρησιμοποιήθηκε το πλήρες σύνολο των χαρακτηριστικών τύπου Haar και δε χρησιμοποιήθηκε η ιδιότητα της συμμετρίας των θετικών παραδειγμάτων. Τα αρνητικά παραδείγματα αντλήθηκαν και πάλι από το σύνολο εικόνων με το εσωτερικό μέρος αυτοκινήτων, όπου οι μοχλοί ταχυτήτων είχαν καλυφθεί. Η διαφορά των δύο ταξινομητών που εκπαιδεύσαμε, βρίσκεται στο συνδυασμό αριθμού σταδίων και μέγιστου λόγου λανθασμένων ανιχνεύσεων. Στον πρώτο ταξινομητή, επιλέξαμε τον συνδυασμό μέγιστου λόγου λανθασμένων ανιχνεύσεων 0,5 και αριθμού σταδίων 20. Με το συνδυασμό αυτό, αναμένουμε συνολικό λόγο λανθασμένων ανιχνεύσεων της τάξης του  $10^{-6}$ , και έναν αρκετά γρήγορο ταξινομητή. Λόγω της έλλειψης πολλών γεωμετρικών χαρακτηριστικών στο σύνολο των αντικειμένων, θεωρήσαμε ότι έπρεπε να κατασκευάσουμε και έναν δεύτερο ταξινομητή με πολύ πιο χαμηλό αναμενόμενο συνολικό λόγο λανθασμένων ανιχνεύσεων. Έτσι, επιλέξαμε ο δεύτερος ταξινομητής να έχει μόλις 7 στάδια, αλλά λόγο λανθασμένων ανιχνεύσεων σε κάθε στάδιο 0,025. Από το συνδυασμό αυτό, αναμένεται ότι ο συνολικός λόγος λανθασμένων ανιχνεύσεων θα είναι της τάξης του  $10^{-11}$ . Η τιμή αυτή είναι εξαιρετικά χαμηλή. Κάθε στάδιο του δεύτερου ταξινομητή αναμένεται να έχει αυξημένο αριθμού χαρακτηριστικών με συνέπεια ο ταξινομητής να είναι πιο αργός.

Μετρήσαμε την απόδοση κάθε ανιχνευτή στο ίδιο σύνολο εικόνων που χρησιμοποιήσαμε για τη μέτρηση της απόδοσης των προηγούμενων ανιχνευτών. Τα αποτελέσματα φαίνονται στο Σχήμα 4.8.

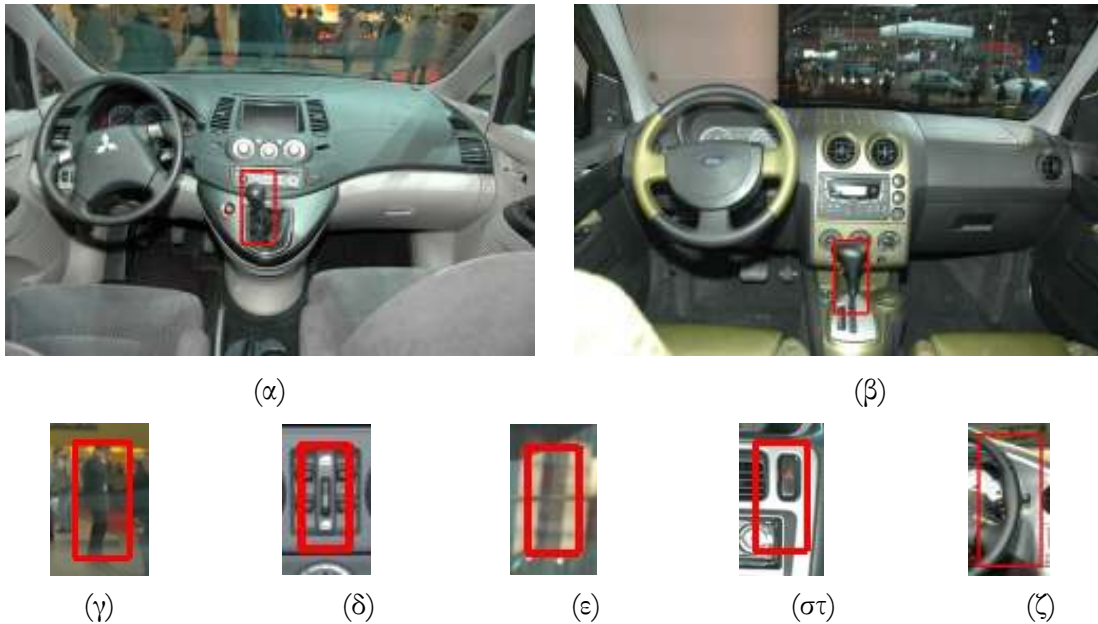
Στην καμπύλη ακρίβειας–επανάκλησης, βλέπουμε ότι έχουμε υψηλή ακρίβεια μόνο για πολύ χαμηλές τιμές επανάκλησης. Η επανάκληση μπορεί να πάρει υψηλές τιμές, αλλά η ακρίβεια θα μειωθεί πολύ. Παρατηρούμε ότι ειδικά σε υψηλές τιμές επανάκλησης, ο ανιχνευτής με τον πιο χαμηλό λόγο λανθασμένης ανίχνευσης έχει μεγαλύτερη ακρίβεια, συγκριτικά με τον πρώτο ανιχνευτή. Το γεγονός αυτό ήταν αναμενόμενο, αφού η πρόβλεψη για το συνολικό λόγο λανθασμένης ανίχνευσης ήταν πολύ χαμηλότερη για τον δεύτερο ταξινομητή. Στην καμπύλη ROC επιβεβαιώνεται το συμπέρασμά μας ότι ο δεύτερος ανιχνευτής δίνει καλύτερα αποτελέσματα, αφού για ίδιο λόγο λανθασμένων θετικών ταξινομήσεων, έχει μεγαλύτερο λόγο πραγματικών θετικών ταξινομήσεων. Βλέπουμε ακόμα, ότι ο λόγος λανθασμένων ανιχνεύσεων είναι αρκετά χαμηλός, αφού ξεπερνά το  $10^{-6}$  μόνο όταν η επανάκληση γίνει 0,7.

Ο πρώτος ανιχνευτής χρειάστηκε 43,5 δευτερόλεπτα για να διατρέξει και τις 59 εικόνες του συνόλου δοκιμής, δηλαδή περίπου 0,7 δευτερόλεπτα για κάθε εικόνα. Ο δεύτερος ανιχνευτής χρειάστηκε 53 δευτερόλεπτα, δηλαδή περίπου 0,9 δευτερόλεπτα για κάθε εικόνα. Κάθε ανιχνευτής έλεγξε συνολικά 59.687.527 υποπαράθυρα, άρα ο πρώτος ανιχνευτής χρειάστηκε  $7,3 \times 10^{-7}$  δευτερόλεπτα για κάθε υποπαράθυρο, ενώ ο δεύτερος  $8,9 \times 10^{-7}$ . Οι χρόνοι και των δύο ανιχνευτών είναι αρκετά καλοί, αν και υψηλότεροι από το χρόνο που απαιτούσε ο ανιχνευτής για τα τιμόνια. Το γεγονός αυτό οφείλεται στην πολυπλοκότητά τους, λόγω της έλλειψης πολλών γεωμετρικών χαρακτηριστικών στα αντικείμενα. Η εκτίμησή μας, ότι ο δεύτερος ανιχνευτής θα είναι πιο αργός, λόγω της συγκριτικά αυξημένης πολυπλοκότητάς του, επιβεβαιώθηκε.



Σχήμα 4.8 - Καμπύλες ακρίβειας–επανάκλησης και ROC των ανιχνευτών για τους μοχλούς ταχυτήτων

Στο Σχήμα 4.9 βλέπουμε κάποια αποτελέσματα της διαδικασίας ανίχνευσης. Στις εικόνες (α) και (β) στο Σχήμα 4.9 βλέπουμε αποτελέσματα της ανίχνευσης που ήταν απόλυτα επιτυχημένα, καθώς εντόπισαν το μοχλό ταχυτήτων σωστά, χωρίς να επιστρέψουν καμία λανθασμένη ανίχνευση. Λόγω του περιορισμένου αριθμού γεωμετρικών χαρακτηριστικών σε ένα μοχλό ταχυτήτων, παρουσιάζονταν και λανθασμένες ανιχνεύσεις. Στις εικόνες (γ)-(ζ) στο Σχήμα 4.9 βλέπουμε τα αποτελέσματα κάποιων λανθασμένων ανιχνεύσεων. Στην εικόνα (γ) φαίνεται ένας άνθρωπος, στη (δ) και (στ) μέρη της κεντρικής κονσόλας, στη (ε) τμήμα του φρόντου και στην (ζ) τμήμα του τιμονιού. Σε μερικά από αυτά (όπως τα (γ) και (δ)) μπορούμε να εντοπίσουμε εύκολα τα γεωμετρικά χαρακτηριστικά που ταιριάζουν με αυτά ενός μοχλού ταχυτήτων, ενώ σε άλλα ο εντοπισμός αυτός δεν είναι το ίδιο εύκολος.



Σχήμα 4.9 - Αποτελέσματα της ανίχνευσης μοχλών ταχυτήτων. Στις εικόνες (α), (β) βλέπουμε σωστές ανιχνεύσεις χωρίς λάθη, ενώ στις (γ)-(ζ) λανθασμένες ανιχνεύσεις.

#### 4.1.4 Πεντάλ

Το τελευταίο αντικείμενο από το εσωτερικό μέρος των αυτοκινήτων με το οποίο ασχοληθήκαμε, ήταν τα πεντάλ. Συγκεκριμένα επιλέξαμε να ασχοληθούμε με το σύνολο των πεντάλ σε ένα αυτοκίνητο, και όχι με κάθε πεντάλ ξεχωριστά. Από τις 690 εικόνες μπορέσαμε να πάρουμε 674 θετικά παραδείγματα. Λόγω της θέσης των πεντάλ (κάτω από το τιμόνι και την κεντρική κονσόλα του αυτοκινήτου), η επιλογή καλών θετικών παραδειγμάτων ήταν δύσκολη. Έτσι, χρησιμοποιήθηκαν ακόμα και εικόνες με μερικές επικαλύψεις. Το μεγαλύτερο πρόβλημα στα θετικά παραδείγματα, είναι ότι ο αριθμός των πεντάλ, το σχήμα τους και η μεταξύ τους απόσταση δεν είναι ίδιες κάθε φορά. Υπάρχουν αυτοκίνητα που έχουν δύο πεντάλ και άλλα που έχουν τρία. Υπάρχουν πεντάλ τετράγωνα, οριζόντια παραλληλόγραμμα και κατακόρυφα. Μερικά θετικά παραδείγματα εκπαίδευσης που χρησιμοποιήσαμε φαίνονται στο Σχήμα 4.10. Παρατηρώντας τις εικόνες στο Σχήμα 4.10, βλέπουμε ότι ο εντοπισμός κοινών γεωμετρικών χαρακτηριστικών θα είναι δύσκολος.

Για τα πεντάλ εκπαιδεύσαμε δύο ανιχνευτές. Λόγω των πολύ έντονων διαφορών που παρουσίαζαν τα θετικά παραδείγματα, έπρεπε να γίνουν τέτοιες επιλογές, ώστε ο συνολικός λόγος λανθασμένων ανιχνεύσεων, που είναι συνδυασμός του αριθμού των σταδίων και του λόγου λανθασμένων ανιχνεύσεων σε κάθε στάδιο, να είναι πολύ χαμηλός. Επιλέξαμε στον ένα ανιχνευτή να χρησιμοποιήσουμε πολύ χαμηλό λόγο λανθασμένων ανιχνεύσεων σε κάθε στάδιο, ίσο με 0,025. Ο ανιχνευτής αυτός δεν χρειάζεται να αποτελείται από πολλά στάδια. Έτσι, επιλέξαμε να περιλαμβάνει 10 στάδια. Στον δεύτερο ανιχνευτή, επιλέξαμε να χρησιμοποιήσουμε περισσότερα στάδια και μεγαλύτερο λόγο λανθασμένων ανιχνεύσεων σε κάθε στάδιο. Έτσι, ο δεύτερος ταξινομητής κατασκευάστηκε ώστε να αποτελείται από 25 στάδια που το καθένα να έχει λόγο



λανθασμένων ανιχνεύσεων ίσο με 0,5. Και στους δύο ανιχνευτές επιλέξαμε να χρησιμοποιήσουμε το πλήρες σύνολο των διαθέσιμων χαρακτηριστικών τύπου Haar, και να μην κάνουμε χρήση της ιδιότητας της συμμετρίας στα θετικά παραδείγματα. Τα αρνητικά παραδείγματα αντλήθηκαν και για τους δύο ανιχνευτές από εικόνες εσωτερικών αυτοκινητών, στις οποίες είχαν καλυφθεί τα πεντάλ. Ο πρώτος ανιχνευτής αναμένεται να είναι πιο αργός, καθώς για να επιτευχθεί τόσο χαμηλός λόγος λανθασμένης ανίχνευσης σε κάθε στάδιο, αυτά θα πρέπει να είναι πολύπλοκα και να αποτελούνται από πολλά χαρακτηριστικά τύπου Haar.

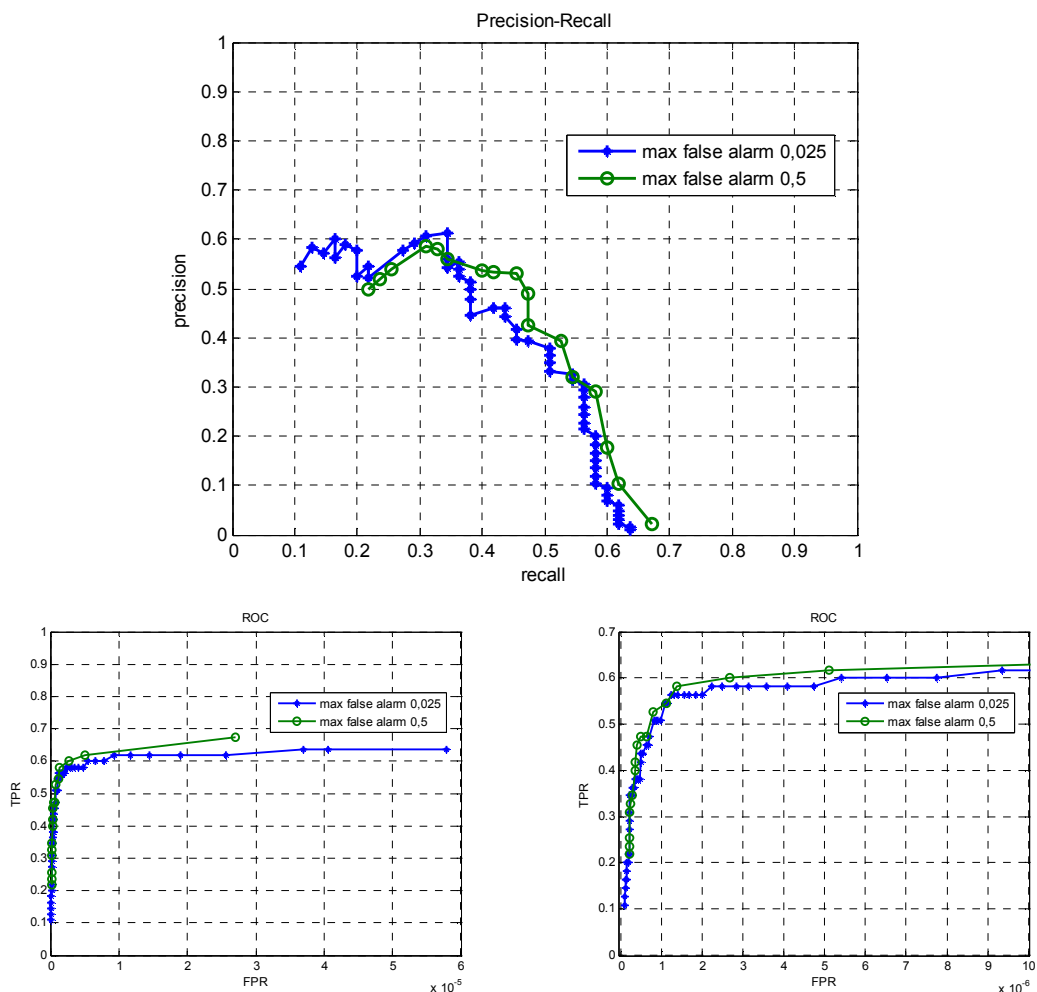


Σχήμα 4.10 - Θετικά παραδείγματα εκπαίδευσης των πεντάλ

Η εκπαίδευση κανενός από τους δύο ανιχνευτές δε μπόρεσε να ολοκληρωθεί. Και στις δύο εκπαιδεύσεις, όταν ο λόγος λανθασμένης ταξινόμησης έφτασε το επίπεδο του  $10^{-4}$ , ο αλγόριθμος εκπαίδευσης AdaBoost δε μπορούσε να επιλέξει χαρακτηριστικά τύπου Haar τα οποία θα ταξινομούσαν καλύτερα τα παραδείγματα. Η εκπαίδευση του πρώτου ανιχνευτή σταμάτησε στο 5° στάδιο, ενώ του δεύτερου στο 18°. Το γεγονός αυτό οφείλεται στη μεγάλη διαφορετικότητα που παρουσίαζαν τα θετικά παραδείγματα.

Στο Σχήμα 4.11 φαίνονται οι καμπύλες ακρίβειας–επανάκλησης και ROC των δύο ανιχνευτών που εκπαιδεύσαμε. Στην καμπύλη ακρίβειας–επανάκλησης βλέπουμε ότι ανεξάρτητα της επανάκλησης των ανιχνευτών, η ακρίβειά τους είναι χαμηλή, γεγονός που οφείλεται στον υψηλό συνολικό λόγο λανθασμένης ανίχνευσης του ταξινομητή μας. Επίσης, η επανάκληση δεν είναι αρκετά υψηλή, γεγονός που οφείλεται στο μεγάλο βαθμό ανομοιογένειας των θετικών παραδειγμάτων. Ανάμεσα στους δύο ανιχνευτές, ο δεύτερος παρουσιάζει καλύτερες επιδόσεις, αφού για την ίδια τιμή επανάκλησης παρουσιάζει μεγαλύτερη ακρίβεια, και μπορεί να επιτύχει συνολικά μεγαλύτερη επανάκληση. Στην καμπύλη ROC επιβεβαιώνεται ότι ο δεύτερος ανιχνευτής είναι καλύτερος, αφού η καμπύλη του βρίσκεται σχεδόν συνέχεια πιο πάνω από την καμπύλη του πρώτου ανιχνευτή.

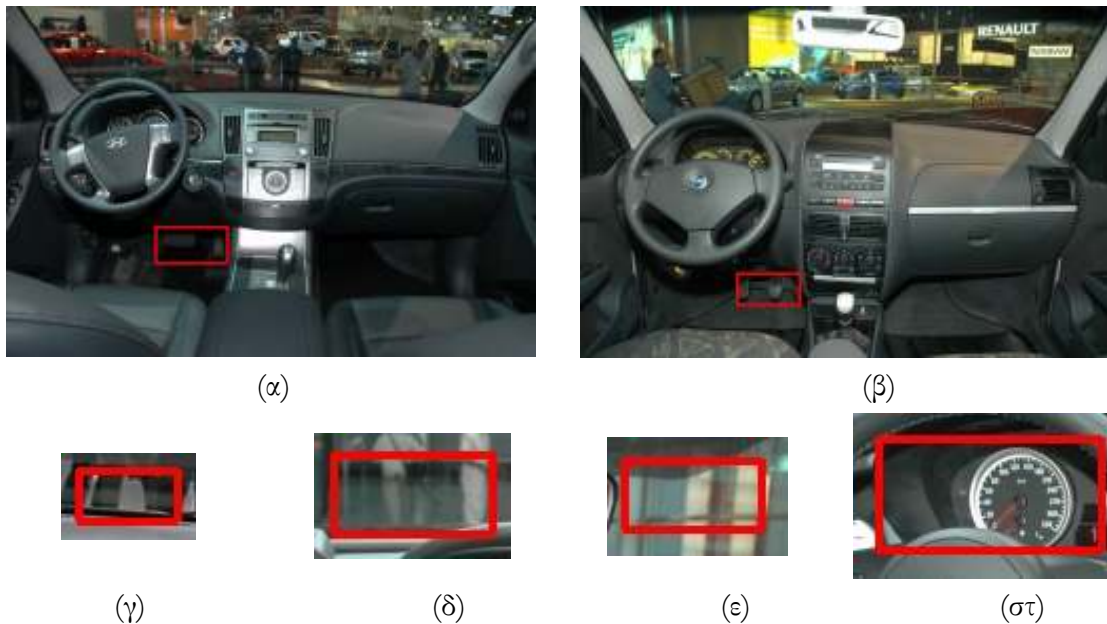
Το σύνολο των εικόνων δοκιμής περιείχε 55 εικόνες. Ο πρώτος ανιχνευτής χρειάστηκε 85,5 δευτερόλεπτα για να τις διατρέξει όλες. Χρειάστηκε, δηλαδή, περίπου 1,5 δευτερόλεπτα για κάθε εικόνα. Ο δεύτερος ανιχνευτής χρειάστηκε 60 δευτερόλεπτα για να ολοκληρώσει τη διαδικασία, δηλαδή περίπου 1,1 δευτερόλεπτα για κάθε εικόνα. Κάθε ανιχνευτής χρειάστηκε να ταξινομήσει 57.617.498 υποπαράθυρα, άρα, ο πρώτος ανιχνευτής χρειάστηκε κατά μέσο όρο  $14,8 \times 10^{-7}$  δευτερόλεπτα, ενώ ο δεύτερος  $10,4 \times 10^{-7}$  ανά υποπαράθυρο. Οι χρόνοι των δύο ανιχνευτών είναι αρκετά υψηλοί. Ο χρόνος του πρώτου ανιχνευτή είναι μεγαλύτερος από αυτόν του δεύτερου, όπως αναμενόταν, καθώς ο πρώτος χρησιμοποιεί λιγότερους και πιο πολύπλοκους ταξινομητές.



Σχήμα 4.11 - Καμπύλες ακρίβειας–επανάκλησης και ROC των ανιχνευτών για τα πεντάλ

Στο Σχήμα 4.12 βλέπουμε κάποια αποτελέσματα της διαδικασίας ανίχνευσης. Στις εικόνες (α) και (β) στο Σχήμα 4.12 βλέπουμε δύο παραδείγματα σωστών ανιχνεύσεων, όπου έχουν ανιχνευθεί σωστά τα πεντάλ, και δεν έχει γίνει καμία λανθασμένη ανίχνευση. Λόγω της μεγάλης ανομοιογένειας των θετικών δειγμάτων, ο ανιχνευτής κάνει και λανθασμένες ταξινομήσεις. Οι εικόνες (γ)-(στ) στο Σχήμα 4.12 περιέχουν κάποιες λανθασμένες ταξινομήσεις που έκανε ο ανιχνευτής μας. Στις εικόνες (γ)-(ε) φαίνονται τμήματα του φόντου, ενώ στην εικόνα (στ) τμήμα των οργάνων ελέγχου. Σε μερικές από αυτές (όπως οι (γ), (δ) και (ε)), μπορούμε να διακρίνουμε εύκολα τα γεωμετρικά χαρακτηριστικά που είναι κοινά με τα θετικά παραδείγματα των πεντάλ.





Σχήμα 4.12 - Αποτελέσματα της ανίχνευσης πεντάλ. Στις εικόνες (α), (β) βλέπουμε σωστές ανιχνεύσεις χωρίς λάθη, ενώ στις (γ)-(ζ) λανθασμένες ανιχνεύσεις.

## 4.2 Σύνολο εικόνων LabelMe

Το LabelMe είναι ένα εργαλείο χαρακτηρισμού εικόνων που λειτουργεί μέσω του διαδικτύου [RTMF05]. Χρησιμοποιεί μια βάση εικόνων και μια βάση αρχείων χαρακτηρισμού των εικόνων αυτών. Στην ιστοσελίδα του LabelMe, μπορεί κανείς να χαρακτηρίσει τα αντικείμενα που βρίσκονται στις εικόνες που του δίνονται. Με τη συνεισφορά κάθε χρήστη, δημιουργήθηκε στην ιστοσελίδα του LabelMe μια πολύ μεγάλη συλλογή από χαρακτηρισμένες εικόνες, που διατίθεται ελεύθερα. Έτσι, αποκτά κανείς εύκολα πρόσβαση σε μια τεράστια συλλογή χαρακτηρισμένων εικόνων κάθε είδους. Στην ιστοσελίδα διατίθεται, επίσης, και μια συλλογή εργαλείων για τη διαχείριση των εικόνων και των αρχείων χαρακτηρισμού τους. Τα εργαλεία αυτά είναι γραμμένα σε κώδικα για το πρόγραμμα Matlab. Περιλαμβάνονται εργαλεία για την εύρεση αντικειμένων στη βάση σύμφωνα με το όνομα του αντικειμένου, εμφάνιση των εικόνων ή των αντικειμένων ξεχωριστά, διάβασμα μιας εικόνας από τη βάση κ.α. Επίσης, περιλαμβάνεται ένα χρήσιμο εργαλείο για τη μετατροπή των αρχείων χαρακτηρισμού των εικόνων από τη μορφή που υποστηρίζει το LabelMe, σε μορφή κατάλληλη για την εκπαίδευση με τη βιβλιοθήκη OpenCV που θα δούμε στο παράρτημα Α.

Η βάση εικόνων του LabelMe περιλαμβάνει πάρα πολλές εικόνες, εκ των οποίων πολλές είναι ήδη χαρακτηρισμένες. Έτσι, έχουμε μια έτοιμη πηγή εικόνων για να εκπαιδεύσουμε ανιχνευτές για πολλές κατηγορίες αντικειμένων. Δυστυχώς όμως, για την εκπαίδευση ενός ανιχνευτή με τη μέθοδο που εξετάζουμε, χρειάζονται πάρα πολλές εικόνες. Στη βάση του LabelMe βρήκαμε μερικές κατηγορίες αντικειμένων που είχαν τον απαραίτητο αριθμό εμφανίσεων σε εικόνες.

Υπάρχουν όμως και μειονεκτήματα στη χρήση της συλλογής του LabelMe. Πρώτον, ο χαρακτηρισμός των εικόνων γίνεται από τυχαίους χρήστες του διαδικτύου. Έτσι, δεν εξασφαλίζεται

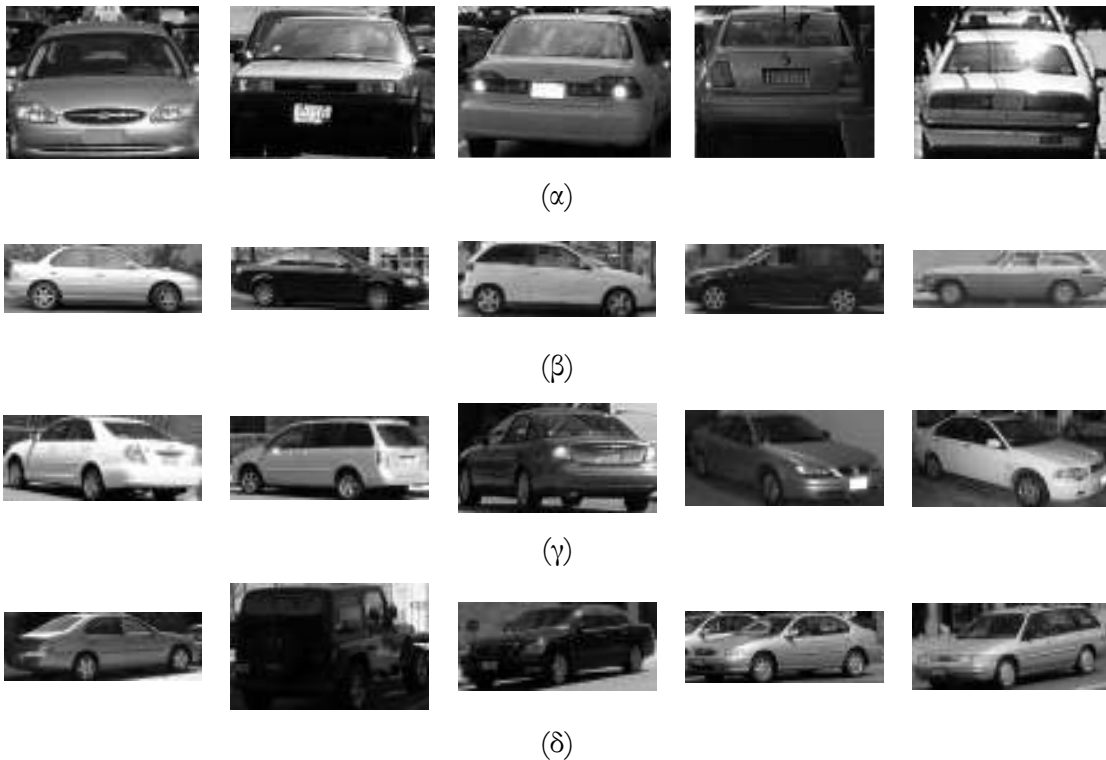
ότι θα είναι ακριβής. Ακόμα, δεν εξασφαλίζεται ούτε ότι ο χρήστης δε θα χαρακτηρίσει τις εικόνες με εντελώς λανθασμένο τρόπο, ακούσια, ή και εκούσια. Επίσης, στη βάση εικόνων του LabelMe περιλαμβάνονται χαρακτηρισμοί για πολλά αντικείμενα με έντονες επικαλύψεις από άλλα αντικείμενα, γεγονός το οποίο δυσχεραίνει την εκπαίδευση ενός ανιχνευτή.

#### 4.2.1 Εκπαίδευση ανιχνευτών για αυτοκίνητα

Μια κατηγορία αντικειμένων που είχε πολλές εμφανίσεις στις εικόνες του LabelMe, ήταν τα αυτοκίνητα. Υπήρχαν φωτογραφίες που απεικόνιζαν το εξωτερικό μέρος αυτοκινήτων από διάφορες γωνίες. Επιλέξαμε λοιπόν, να ασχοληθούμε με την ανίχνευση αυτοκινήτων. Ψάχνοντας για τη λέξη "αυτοκίνητο" στη βάση εικόνων, βρήκαμε 15.652 αντικείμενα. Από αυτά, ένας μεγάλος αριθμός ήταν αντικείμενα που περιλάμβαναν τη λέξη αυτοκίνητο, όπως αυτοκινητόδρομος, ρόδα αυτοκινήτου, παράθυρο αυτοκινήτου κ.α. Μετά από διάφορες επιλογές, καταλήξαμε να έχουμε στη διάθεσή μας 2.821 εικόνες αυτοκινήτων, οι οποίες απεικόνιζαν ολόκληρα αυτοκίνητα, χωρίς έντονες επικαλύψεις από άλλα αντικείμενα, υπό διάφορες γωνίες. Όπως περιγράφεται και στη βιβλιογραφία, επιλέξαμε να χωρίσουμε τις εικόνες σε υποκατηγορίες, ανάλογα με την γωνία με τη οποία φαίνεται το αυτοκίνητο [ScKa00]. Επιλέξαμε να δημιουργήσουμε τέσσερις ομάδες. Η επιλογή αυτή έγινε, ώστε να υπάρχει ομοιομορφία στις εικόνες που ανήκουν σε κάθε ομάδα, αλλά και να υπάρχουν αρκετά αντικείμενα σε κάθε ομάδα ώστε να μπορέσει να εκπαιδευτεί ένας ανιχνευτής υψηλών επιδόσεων. Η πρώτη ομάδα αποτελείται από αυτοκίνητα τα οποία έχουν φωτογραφηθεί από το εμπρός ή το πίσω μέρος τους, με την ελάχιστη δυνατή γωνία λήψης. Η δεύτερη ομάδα περιλαμβάνει αυτοκίνητα που φωτογραφήθηκαν από την πλαϊνή τους μεριά, κοιτώντας είτε προς τα δεξιά, είτε προς τα αριστερά, και πάλι με την ελάχιστη δυνατή γωνία. Η τρίτη και η τέταρτη ομάδα αποτελούνται από αυτοκίνητα που φωτογραφήθηκαν υπό μεγάλη γωνία. Στην τρίτη ομάδα η διεύθυνση των αυτοκινήτων είναι αυτή στην οποία το πλαϊνό τμήμα του αυτοκινήτου βρίσκεται στο αριστερό μέρος της εικόνας. Στην τέταρτη ομάδα, αντίθετα, το πλαϊνό τμήμα του αυτοκινήτου βρίσκεται στο δεξιό μέρος της εικόνας. Στο Σχήμα 4.13 φαίνονται μερικά παραδείγματα από κάθε ομάδα.

Σε κάθε ομάδα υπάρχουν αρκετές εικόνες, ώστε να εκπαιδευτεί κάθε ταξινομητής. Από τη δεύτερη και την τρίτη ομάδα, αφαιρέθηκαν τέσσερις εικόνες καθώς το μέγεθός τους ήταν πολύ μικρό. Έτσι, καταλήγουμε η πρώτη ομάδα να αποτελείται από 920 εικόνες, η δεύτερη από 666, η τρίτη από 612 και η τέταρτη από 619 εικόνες. Το σύνολο των εικόνων σε κάθε ομάδα είναι αρκετά ομοιογενές, λόγω του διαχωρισμού που κάναμε. Η ομοιογένεια των δύο πρώτων ομάδων είναι ακόμα μεγαλύτερη, καθώς οι γωνίες εμφάνισης των αντικειμένων είναι πιο συγκεκριμένες. Έτσι, για την εκπαίδευση κάθε ανιχνευτή, χρησιμοποιήσαμε όλες τις εικόνες της αντίστοιχης ομάδας ως θετικά παραδείγματα.

Για αρνητικά παραδείγματα σε κάθε ανιχνευτή, χρησιμοποιήσαμε τις εικόνες στις οποίες ανήκαν τα θετικά παραδείγματα. Σε κάθε μία από αυτές, καλύψαμε κάθε αντικείμενο που ταίριαζε στην περιγραφή "αυτοκίνητο" της βάσης του LabelMe. Έτσι, εξασφαλίσουμε ότι δεν θα υπήρχε κανένα αυτοκίνητο στα αρνητικά παραδείγματα, ούτε από αυτά που επιλέχθηκαν ως θετικά παραδείγματα, αλλά ούτε και από αυτά που απορρίφθηκαν ως ακατάλληλα για θετικά παραδείγματα, π.χ. λόγω επικαλύψεων.



Σχήμα 4.13 - Υποομάδες της κατηγορίας αντικειμένων "αυτοκίνητα". α) Εμπρός και πίσω μέρος αυτοκινήτων, β) πλαϊνό μέρος αυτοκινήτων, γ) αυτοκίνητα υπό έντονη γωνία, με το πλαϊνό τους τμήμα αριστερά, δ) αυτοκίνητα υπό έντονη γωνία, με το πλαϊνό τους τμήμα δεξιά

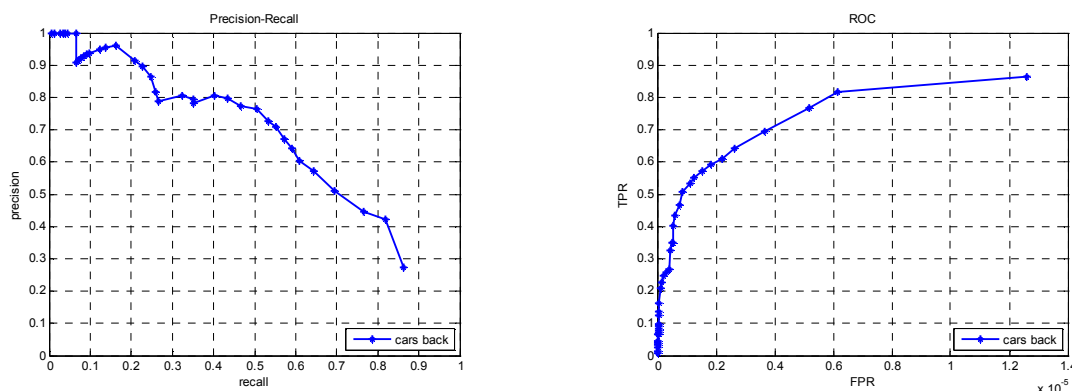
Οι υπόλοιπες παράμετροι εκπαίδευσης επιλέχθηκε να είναι ίδιες για όλους τους ανιχνευτές. Καθώς οι εικόνες είχαν αρκετά μεγάλη ομοιογένεια μεταξύ τους (σε κάθε ομάδα), επιλέχθηκε η χρησιμοποίηση υψηλού λόγου λανθασμένης ανίχνευσης σε κάθε στάδιο, ίσου με 0,5. Για να μειωθεί ο συνολικός λόγος λανθασμένων ανιχνεύσεων, επιλέξαμε να χρησιμοποιήσουμε 20 στάδια σε κάθε ανιχνευτή. Επιλέξαμε τη χρησιμοποίηση όλων των διαθέσιμων χαρακτηριστικών τύπου Haar για να πετύχουμε καλύτερη περιγραφή των γεωμετρικών χαρακτηριστικών των αντικειμένων. Τέλος, δε χρησιμοποιήθηκε συμμετρική εκπαίδευση, καθώς τα αντικείμενα των δύο τελευταίων ομάδων δεν είναι συμμετρικά. Επίσης, με τη χρήση συμμετρίας η αναζήτηση για χαρακτηριστικά γίνεται λιγότερο εξαντλητική, γεγονός το οποίο οδηγεί σε χαμηλότερες επιδόσεις.

#### 4.2.2 Αξιολόγηση επιδόσεων ταξινομητή για το εμπρός και πίσω μέρος αυτοκινήτων

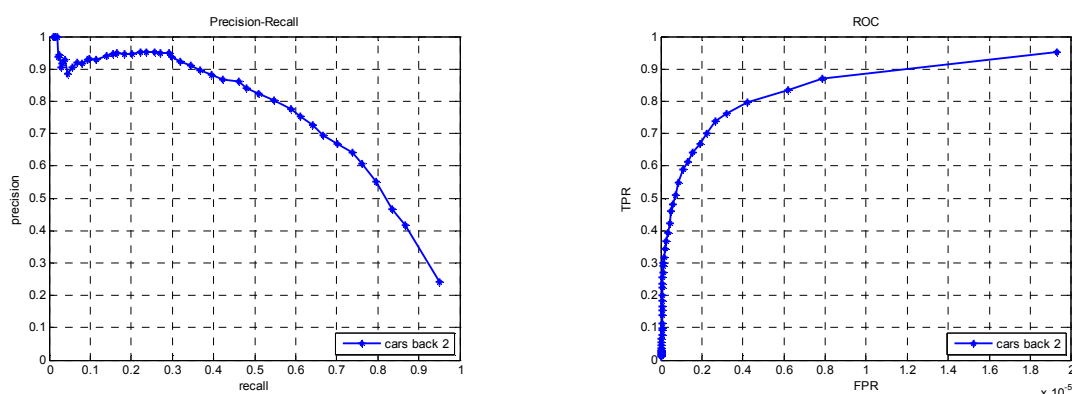
Για να μετρήσουμε την απόδοση του κάθε ανιχνευτή που εκπαιδεύσαμε, χρειαζόμασταν σύνολα εικόνων που να περιείχαν εμφανίσεις αυτοκινήτων υπό συγκεκριμένες γωνίες. Τις εικόνες από τη βάση του LabelMe, τις χρησιμοποιήσαμε όλες για την εκπαίδευση. Έτσι, έπρεπε να βρούμε άλλα σύνολα εικόνων που να περιέχουν αυτοκίνητα υπό συγκεκριμένες γωνίες, για να μετρήσουμε την απόδοση κάθε ανιχνευτή. Δυστυχώς δεν υπάρχουν πολλές συλλογές εικόνων αυτοκινήτων

διαθέσιμες στο διαδίκτυο, και ιδιαίτερα υπό συγκεκριμένη γωνία εμφάνισης. Βρήκαμε τρεις συλλογές, τις οποίες χρησιμοποιήσαμε όπως περιγράφεται παρακάτω.

Ο πρώτος ανιχνευτής που εκπαιδεύσαμε εντοπίζει το εμπρός και πίσω μέρος αυτοκινήτων. Για να μετρήσουμε την απόδοσή του χρησιμοποιήσαμε δύο σύνολα εικόνων του Πανεπιστημίου Caltech (California Institute of Technology) που περιέχουν φωτογραφίες αυτοκινήτων τραβηγμένες σε ευθεία με το πίσω μέρος τους. Το πρώτο σύνολο [WeWP00] περιέχει 126 εικόνες με αυτοκίνητα, ενώ το δεύτερο [FePZ03] περιέχει 526 εικόνες. Εφαρμόσαμε τον ανιχνευτή μας σε αυτά τα δύο σύνολα εικόνων, και τα αποτελέσματα φαίνονται στο Σχήμα 4.14.



(α)



(β)

Σχήμα 4.14 - Καμπύλες ακρίβειας–επανάκλησης και ROC του ανιχνευτή εμπρός και πίσω μέρους αυτοκινήτων. α) Αποτελέσματα από την πρώτη ομάδα εικόνων, β) αποτελέσματα από τη δεύτερη ομάδα εικόνων

Τα αποτελέσματα του πρώτου συνόλου εικόνων φαίνονται στο Σχήμα 4.14α. Στην καμπύλη ακρίβειας–επανάκλησης, βλέπουμε ότι μπορούμε να πετύχουμε αρκετά υψηλή επανάκληση, της τάξης του 0,85. Η ακρίβεια παραμένει υψηλά για χαμηλή επανάκληση, αλλά όσο αυξάνεται η επανάκληση, μειώνεται η ακρίβεια. Ακόμα και όταν έχουμε υψηλή επανάκληση, της τάξης του 0,7, η ακρίβεια παραμένει ικανοποιητική, με τιμές περίπου στο 0,5. Στην καμπύλη ROC επιβεβαιώνεται η υψηλή απόδοση του ανιχνευτή μας. Ακόμα και σε πολύ υψηλή επανάκληση, ο λόγος λανθασμένης θετικής ανίχνευσης παραμένει αρκετά χαμηλός, μικρότερος από  $10^{-5}$ .

Για την ανίχνευση των 126 εικόνων ανάλυσης 378x250 εικονοστοιχείων χρειάστηκαν 20,9 δευτερόλεπτα. Άρα, ο ανιχνευτής χρειαζόταν 0,17 δευτερόλεπτα για να ολοκληρώσει την ανίχνευση σε κάθε εικόνα. Για το σύνολο της διαδικασίας, ο ανιχνευτής χρειάστηκε να ελέγξει 28.213.164 υποπαράθυρα της εικόνας. Άρα, χρειάστηκε κατά μέσο όρο  $7,4 \times 10^{-7}$  δευτερόλεπτα για να ελέγξει κάθε υποπαράθυρο. Ο χρόνος αυτός είναι αρκετά καλός, αν και περίπου διπλάσιος από τον καλύτερο που έχουμε μετρήσει, που αφορά την ανίχνευση τιμονιών στο εσωτερικό αυτοκινήτου. Μερικά αποτελέσματα της ανίχνευσης φαίνονται στο Σχήμα 4.15.

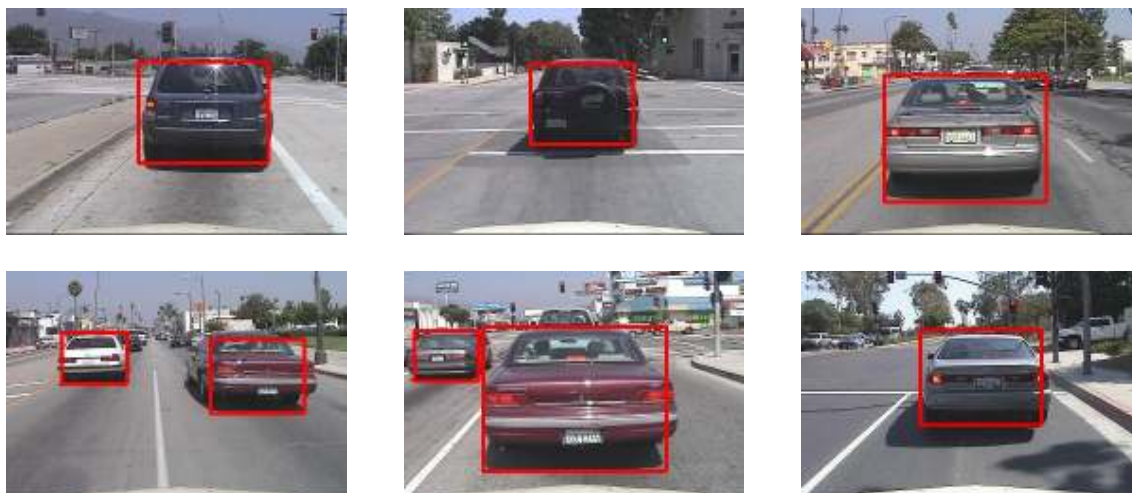


Σχήμα 4.15 - Αποτελέσματα της ανίχνευσης για το εμπρός και πίσω μέρος αυτοκινήτων στο πρώτο σύνολο εικόνων δοκιμής.

Στο Σχήμα 4.14β βλέπουμε τα αποτελέσματα του δεύτερου συνόλου εικόνων. Η καμπύλη ακρίβειας-επανάκλησης δείχνει καλύτερα αποτελέσματα από αυτήν του πρώτου συνόλου εικόνων. Συγκεκριμένα, για κάθε τιμή ακρίβειας έχουμε επανάκληση μεγαλύτερη κατά περίπου 0,1. Η καμπύλη ROC επιβεβαιώνει ότι ο ανιχνευτής μας είχε καλύτερα αποτελέσματα στο δεύτερο σύνολο εικόνων. Ακόμα, η μέγιστη τιμή επανάκλησης που μπορούμε να πετύχουμε φτάνει το 0,95, ή 95%, τιμή αρκετά υψηλή. Οι Fergus et al. χρησιμοποιώντας το ίδιο σύνολο εικόνων δοκιμής, πέτυχαν μέγιστη επανάκληση ίση με 90,3% [FePZ03]. Βλέπουμε λοιπόν, ότι ο ανιχνευτής μας δίνει πολύ καλά αποτελέσματα. Αποτελέσματα της διαδικασίας ανίχνευσης φαίνονται στο Σχήμα 4.16.

Για την ανίχνευση στις 526 εικόνες ανάλυσης 360x240 εικονοστοιχείων, χρειάστηκαν 71,8 δευτερόλεπτα, άρα περίπου 0,14 δευτερόλεπτα για κάθε εικόνα. Συνολικά ο ανιχνευτής χρειάστηκε να ελέγξει 106.286.716 υποπαράθυρα, άρα χρειάστηκε περίπου  $6,8 \times 10^{-7}$  δευτερόλεπτα για κάθε υποπαράθυρο. Η διαφορά που παρατηρείται στο χρόνο που απαιτείται για την εξέταση ενός υποπαράθυρου στο πρώτο και το δεύτερο σύνολο εικόνων είναι μικρή και μπορεί να οφείλεται ακόμα και σε τυχαίες παραμέτρους, όπως τα προγράμματα που τρέχουν παράλληλα στον υπολογιστή.



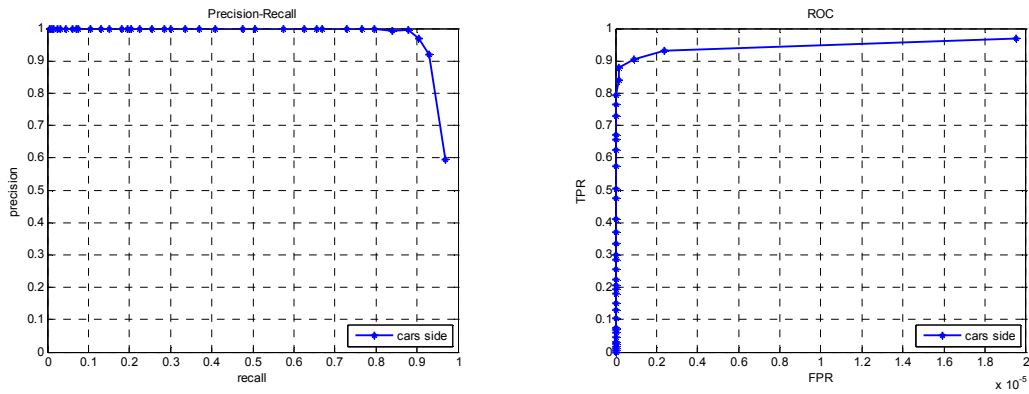


Σχήμα 4.16 - Αποτελέσματα της ανίχνευσης για το εμπρός και πίσω μέρος αυτοκινήτων στο δεύτερο σύνολο εικόνων δοκιμής.

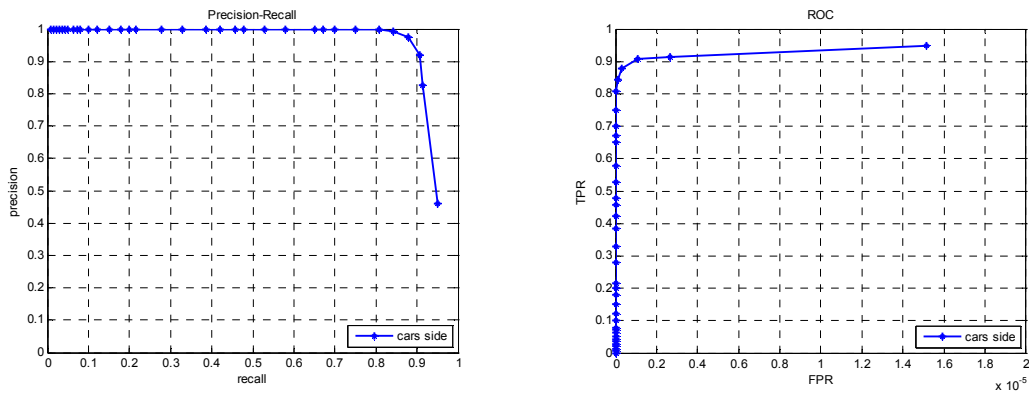
#### 4.2.3 Αξιολόγηση επιδόσεων ταξινομητή για το πλαϊνό μέρος αυτοκινήτων

Ο δεύτερος ανιχνευτής που εκπαιδεύσαμε, ανιχνεύει το πλαϊνό μέρος αυτοκινήτων. Για να μετρήσουμε την απόδοσή του, χρησιμοποιήσαμε ένα σύνολο εικόνων στο οποίο τα αυτοκίνητα φαίνονταν από το πλάι, με φορά είτε προς τα δεξιά, είτε προς τα αριστερά [AgAR04]. Το σύνολο αυτό χωρίζεται σε δύο ομάδες. Η πρώτη αποτελείται από 170 εικόνες με 200 εμφανίσεις αυτοκινήτων σε σταθερή κλίμακα. Η δεύτερη αποτελείται από 108 εικόνες με 139 αυτοκίνητα σε διάφορες κλίμακες. Εφαρμόσαμε τον ανιχνευτή και στις δύο ομάδες εικόνων. Τα αποτελέσματα των δύο ομάδων φαίνονται στο Σχήμα 4.17.

Στο Σχήμα 4.17α βλέπουμε τις χαρακτηριστικές καμπύλες για την πρώτη ομάδα εικόνων, όπου το μέγεθος των αυτοκινήτων είναι σταθερό. Η επανάκληση μπορεί να πάρει πολύ υψηλές τιμές, φτάνοντας ακόμα και το 0,95. Η ακρίβεια για τιμές επανάκλησης μέχρι 0,8 (που είναι αρκετά μεγάλη τιμή), είναι ίση με τη μονάδα, δηλαδή δεν έχουμε καμία λανθασμένη θετική ανίχνευση. Η τιμή αυτή για την ακρίβεια είναι η ιδανική. Ακόμα και για επανάκληση 0,9, η ακρίβεια παραμένει σε πάρα πολύ υψηλή τιμή, ίση με 0,95. Μόνο όταν η επανάκληση φτάσει στο 0,95 μειώνεται η τιμή της ακρίβειας στο 0,6. Στην καμπύλη ROC βλέπουμε ότι για τιμές επανάκλησης μέχρι 0,8 ο λόγος λανθασμένων θετικών ανιχνεύσεων είναι ίσος με μηδέν, αφού δεν εμφανίζεται καμία λανθασμένη θετική ανίχνευση. Ακόμα και για πολύ υψηλή επανάκληση (0,95), ο λόγος λανθασμένων θετικών ανιχνεύσεων παίρνει τιμές της τάξης του  $10^{-5}$ . Τα πολύ καλά αυτά αποτελέσματα επαναλαμβάνονται και στη δεύτερη ομάδα εικόνων (Σχήμα 4.17β), όπου τα αυτοκίνητα δεν εμφανίζονται υπό συγκεκριμένη κλίμακα. Οι καμπύλες ακρίβειας–επανάκλησης και ROC φανερώσουν την ίδια καλή απόδοση και σε αυτή την ομάδα εικόνων. Άρα, οι τόσο καλές επιδόσεις του ανιχνευτή μας δεν περιορίζονται σε αντικείμενα με συγκεκριμένο μέγεθος. Οι υψηλές τιμές επανάκλησης αποδίδονται στη μεγάλη ομοιογένεια των θετικών παραδειγμάτων εκπαίδευσης.



(α)



(β)

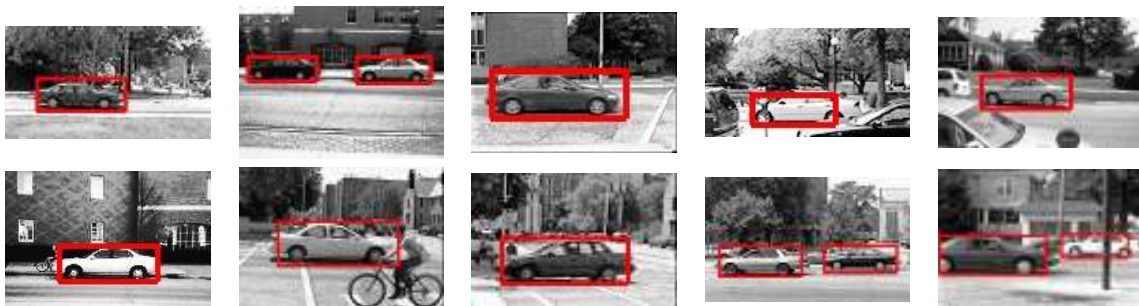
Σχήμα 4.17 - Καμπύλες ακρίβειας-επανάκλησης και ROC του ανιχνευτή του πλαϊνού μέρους αυτοκινήτων. α) Αποτελέσματα από την πρώτη ομάδα εικόνων (αντικείμενα σταθερής κλίμακας), β) αποτελέσματα από τη δεύτερη ομάδα εικόνων (αντικείμενα μεταβλητής κλίμακας)

Στον Πίνακα 4.1 παραθέτουμε τα αποτελέσματα ανίχνευσης στα δύο αυτά σύνολα εικόνων που βρήκαμε στη βιβλιογραφία. Οι τιμές δίνουν την επανάκληση για ίσους λόγους σφαλμάτων (Equal Error Rate – ERR), δηλαδή για την τιμή επανάκλησης που ισούται με την ακρίβεια των αποτελεσμάτων. Βλέπουμε ότι τα αποτελέσματα που επιτύχαμε με τη μέθοδο αυτή είναι κοντά στα καλύτερα της βιβλιογραφίας. Συγκεκριμένα, καλύτερα αποτελέσματα προκύπτουν μόνο στις εργασίες [LeLS08] και [MuLo06] για το πρώτο σύνολο εικόνων και [LeLS08] για το δεύτερο σύνολο εικόνων. Τα αποτελέσματα των Leibe et al. στο [LeLS08] είναι καλύτερα, καθώς το αφηρημένο μοντέλο σχήματος που χρησιμοποιούν υποστηρίζει και σημαντικές επικαλύψεις του αντικειμένου. Μερικά αποτελέσματα του ανιχνευτή μας στο σύνολο εικόνων, παρουσιάζονται στο Σχήμα 4.18.

Μέθοδος ανίχνευσης	Σύνολο εικόνων 1 (σταθερή κλίμακα)	Σύνολο εικόνων 2 (μεταβλητή κλίμακα)
AgAR04	~77%	~40%
AgRo02	~79%	-
FePZ03	88,5%	-
FLCS05	88,6%	87,8%
GaAH02	~88%	-
LeLS08	97,5%	95%
MuLo06	99,9%	90,6%
Προτεινόμενη	92,7%	90,7%

Πίνακας 4.1 - Σύγκριση διάφορων ανιχνευτών για το πλαϊνό μέρος αυτοκινήτων. Για κάθε μέθοδο βλέπουμε την επανάκληση για ίσους λόγους σφαλμάτων.

Ο χρόνος ανίχνευσης για τις δύο ομάδες εικόνων ήταν 4,7 και 6,8 δευτερόλεπτα για την πρώτη και τη δεύτερη ομάδα, αντίστοιχα. Η πρώτη ομάδα αποτελούνταν από 170 εικόνες ανάλυσης από 110x75 έως 360x176 εικονοστοιχεία. Η δεύτερη ομάδα αποτελούνταν από 107 εικόνες ανάλυσης από 151x101 έως 434x205 εικονοστοιχεία. Λόγω των μεγάλων διακυμάνσεων στις διαστάσεις των εικόνων δε μπορούμε να υπολογίσουμε τη μέση ταχύτητα του ανιχνευτή ανά εικόνα. Μπορούμε όμως να υπολογίσουμε το μέσο χρόνο επεξεργασίας κάθε υποπαραθύρου της εικόνας. Συνολικά, ελέγχθηκαν 6.866.176 υποπαραθύρα στην πρώτη ομάδα και 10.311.190 στη δεύτερη, δηλαδή, 17.177.366 υποπαραθύρα για όλες τις εικόνες δοκιμής. Άρα, ο μέσος χρόνος που απαιτήθηκε για τον έλεγχο κάθε υποπαραθύρου ήταν περίπου  $6,7 \times 10^{-7}$  δευτερόλεπτα, χρόνος αρκετά μικρός. Ο χρόνος που χρειάστηκε η όλη διαδικασία ήταν ιδιαίτερα μικρός, κυρίως λόγω της χαμηλής ανάλυσης των εικόνων.

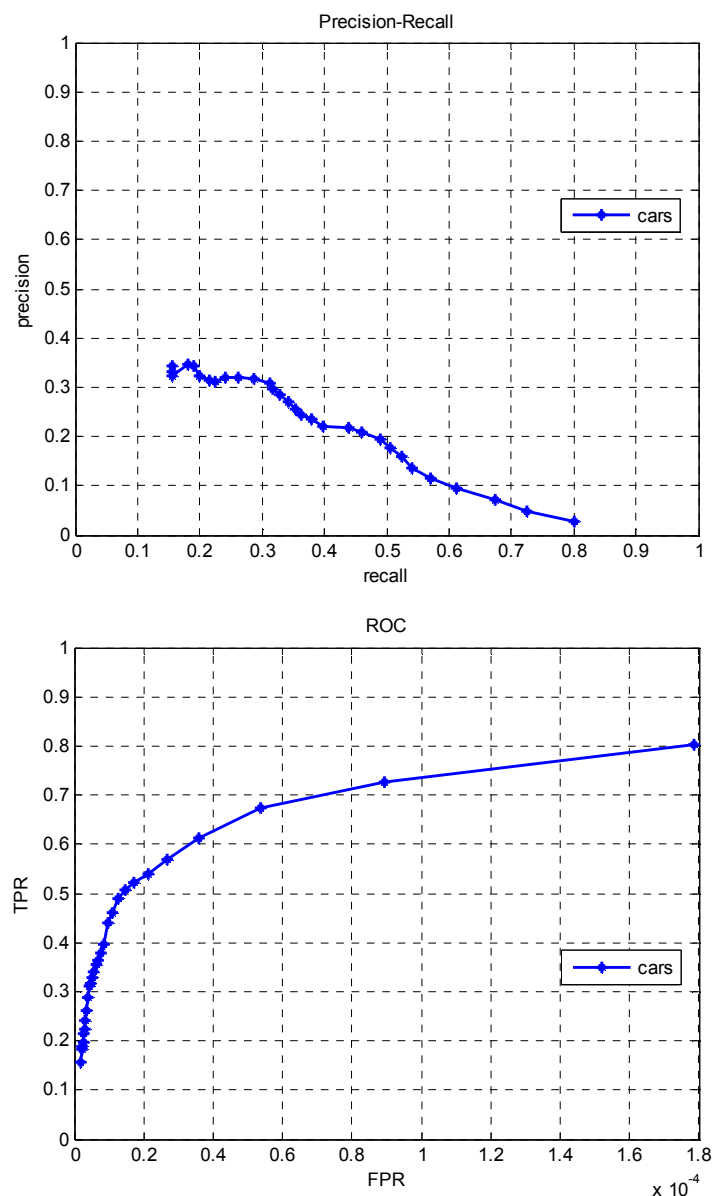


Σχήμα 4.18 - Αποτελέσματα της ανίχνευσης του πλαϊνού μέρους αυτοκινήτων



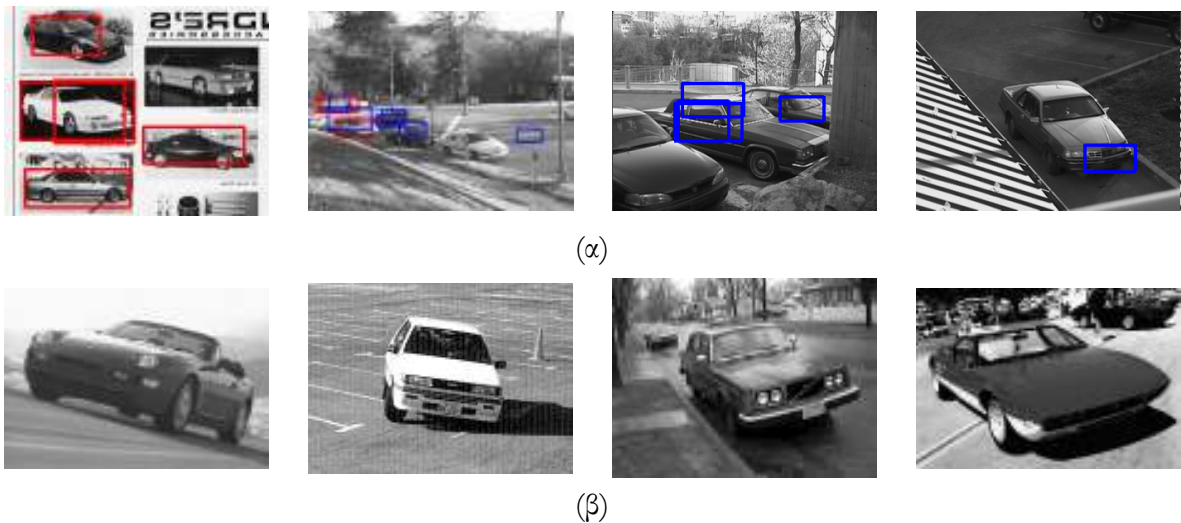
#### 4.2.4 Αξιολόγηση επιδόσεων όλων των ταξινομητών για τα μέρη αυτοκινήτων

Το τρίτο σύνολο εικόνων που είχαμε στη διάθεσή μας για τη μέτρηση επιδόσεων των ανιχνευτών, περιελάμβανε εικόνες αυτοκινήτων υπό διάφορες γωνίες [ScKa00]. Η συλλογή αυτή αποτελείται από 104 εικόνες διαφόρων διαστάσεων και συνολικά 213 εμφανίσεις αυτοκινήτων. Οι γωνίες εμφάνισης των αυτοκινήτων στις εικόνες αυτές είναι πολλές και τυχαίες, κι έτσι επιλέξαμε να εφαρμόσουμε και τους τέσσερις ανιχνευτές σε αυτές τις εικόνες. Σε κάθε εικόνα, εφαρμόσαμε τους τέσσερις ανιχνευτές διαδοχικά, δημιουργήσαμε ένα ενιαίο σύνολο με τα αποτελέσματα των ανιχνεύσεων και στο τέλος ελέγξαμε ποια αποτελέσματα αντιστοιχούν σε αυτοκίνητα και ποια όχι. Οι χαρακτηριστικές καμπύλες φαίνονται στο Σχήμα 4.19.



Σχήμα 4.19 - Καμπύλες ακρίβειας-επανάκλησης και ROC του συνδυασμού των ανιχνευτών για αυτοκίνητα

Παρατηρώντας την καμπύλη ακρίβειας–επανάκλησης στο Σχήμα 4.19, βλέπουμε ότι συνδυάζοντας τους τέσσερις ανιχνευτές μας μπορούμε να εντοπίσουμε μεγάλο ποσοστό αυτοκινητών. Η επανάκληση μπορεί να φτάσει ακόμα και το 0,8, η ακρίβεια, όμως, στην ανίχνευση είναι χαμηλή. Αυτό οφείλεται κυρίως στο γεγονός ότι χρησιμοποιήσαμε τέσσερις ανιχνευτές αντί για έναν. Κάθε ανιχνευτής επιστρέφει έναν αριθμό λανθασμένων θετικών ανιχνεύσεων. Χρησιμοποιώντας κάθε φορά τέσσερις ανιχνευτές, ο συνολικός αριθμός λανθασμένων θετικών ανιχνεύσεων προκύπτει από το άθροισμα των επιμέρους. Έτσι, είναι λογικό η ακρίβεια να είναι μειωμένη. Τα αποτελέσματα αυτά επιβεβαιώνονται και στην καμπύλη ROC, όπου ο λόγος λανθασμένων θετικών ανιχνεύσεων είναι της τάξης του  $10^{-4}$ , για υψηλή επανάκληση. Μερικά αποτελέσματα της ανίχνευσης φαίνονται στο Σχήμα 4.20.



Σχήμα 4.20 - Αποτελέσματα της ανίχνευσης αυτοκινητών σε τυχαίες γωνίες. α) Εικόνες με σωστές (κόκκινο χρώμα) και λανθασμένες (μπλε χρώμα) ανιχνεύσεις, β) εικόνες στις οποίες δεν ανιχνεύθηκε κανένα αντικείμενο.

Παρατηρώντας τις εικόνες στο Σχήμα 4.20 βλέπουμε ότι οι γωνίες εμφάνισης των αυτοκινητών είναι πολύ διαφορετικές από αυτές που παρουσιάστηκαν στο Σχήμα 4.13. Στις εικόνες αυτές έχουμε περιστροφή του αντικειμένου γύρω από όλους τους άξονες, ενώ η εκπαίδευση έγινε με εικόνες που είχαν περιστροφή ως προς τον ένα μόνο άξονα (τον κατακόρυφο). Από την τόσο μεγάλη διαφοροποίηση των εικόνων του συνόλου δοκιμής από τις εικόνες που χρησιμοποιήθηκαν κατά την εκπαίδευση των ανιχνευτών, δικαιολογείται η χαμηλή απόδοση του ανιχνευτή, τόσο ως προς την ακρίβεια, όσο και ως προς την επανάκληση.

Για την εκτέλεση της διαδικασίας της ανίχνευσης στις 104 εικόνες χρειάστηκαν 115 δευτερόλεπτα. Οι εικόνες είχαν πολύ διαφορετικές αναλύσεις, κι έτσι δε μπορούμε να υπολογίσουμε το μέσο χρόνο που χρειάστηκε για κάθε εικόνα. Συνολικά χρειάστηκε να ελεγχθούν 38.431.039 από τέσσερις ανιχνευτές. Έτσι, μπορούμε να πούμε ότι το σύνολο των τεσσάρων ανιχνευτών χρειάστηκε  $3 \times 10^{-6}$  δευτερόλεπτα για κάθε υποπαράθυρο, ή ότι κάθε ξεχωριστός ανιχνευτής χρειάστηκε κατά μέσο όρο  $7,5 \times 10^{-7}$  δευτερόλεπτα. Ο χρόνος αυτός είναι παραπλήσιος με το χρόνο που υπολογίσαμε στα προηγούμενα σύνολα εικόνων.

## Κεφάλαιο 5.

### Ημιαυτόματος χαρακτηρισμός εικόνων

Στο κεφάλαιο αυτό παρουσιάζουμε το πρόγραμμα Annotator που δημιουργήσαμε για τον χαρακτηρισμό εικόνων με ημιαυτόματο τρόπο. Χαρακτηρισμός μιας εικόνας είναι η διαδικασία κατά την οποία προσδιορίζουμε τη θέση και το είδος των αντικειμένων που υπάρχουν σε μια εικόνα. Μπορούμε να χαρακτηρίσουμε όλα τα αντικείμενα σε μια εικόνα, ή μόνο αυτά που μας ενδιαφέρουν. Για να περιγράψουμε το είδος του αντικειμένου που βρίσκεται σε κάθε θέση, συνήθως αρκεί να το ονομάσουμε. Αν ο χαρακτηρισμός γίνεται εξολοκλήρου από το χρήστη, λέμε ότι γίνεται χειροκίνητα, ενώ αν γίνεται αποκλειστικά από ένα είδος μηχανής, τότε λέμε ότι γίνεται αυτόματα. Τέλος, αν γίνεται από κάποια μηχανή με την επίβλεψη και την παρέμβαση του χρήστη, τότε λέμε ότι γίνεται με ημιαυτόματο τρόπο.

#### 5.1 Εισαγωγή

Στη μέθοδο ανίχνευσης που εξετάσαμε σε αυτή τη διπλωματική, για να εκπαιδύσουμε έναν ανιχνευτή έπρεπε να δώσουμε έναν μεγάλο αριθμό θετικών παραδειγμάτων του αντικειμένου που θέλουμε να ανιχνεύσουμε. Τα παραδείγματα αυτά είναι εικόνες που περιλαμβάνουν μόνο το αντικείμενο που μας ενδιαφέρει. Η εύρεση μεγάλου αριθμού τέτοιων εικόνων, που απεικονίζουν μόνο ένα αντικείμενο, είναι αδύνατη. Για να μπορέσουμε, λοιπόν, να κατασκευάσουμε τα θετικά παραδείγματα εκπαίδευσης, θα πρέπει να αποκόψουμε το αντικείμενο που μας ενδιαφέρει από εικόνες που το περιέχουν (μαζί με πολλά άλλα αντικείμενα). Για να γίνει αυτό, θα πρέπει να ξέρουμε που βρίσκεται το αντικείμενο στις εικόνες που έχουμε στη διάθεσή μας. Θα πρέπει δηλαδή, οι εικόνες μας να έχουν χαρακτηριστεί, τουλάχιστον για το αντικείμενο που θέλουμε να ανιχνεύσουμε.

Έπειτα, είδαμε ότι για κάθε ανιχνευτή είναι απαραίτητη η μέτρηση της επίδοσής του. Αρχικά, θα πρέπει να εφαρμοστεί ο ανιχνευτής σε ένα σύνολο εικόνων δοκιμής. Έπειτα, θα πρέπει να ελέγξουμε τα αποτελέσματα της ανίχνευσης, για να τα κατατάξουμε ως σωστά ή λανθασμένα. Αυτή η διαδικασία ελέγχου είναι δύσκολο να γίνει χειροκίνητα. Θα πρέπει να γίνει, λοιπόν, με τη χρήση κάποιου προγράμματος υπολογιστή. Για να μπορέσει όμως ο υπολογιστής να κατατάξει μια ανίχνευση ως σωστή ή λανθασμένη, θα πρέπει να ξέρει που βρίσκεται πραγματικά το αντικείμενο στις εικόνες της δοκιμής. Θα πρέπει να έχει δηλαδή, στη διάθεσή του, το χαρακτηρισμό των εικόνων δοκιμής για το αντικείμενο που μας ενδιαφέρει.

Βλέπουμε λοιπόν, ότι για την εκπαίδευση και την αξιολόγηση ενός ανιχνευτή, είναι απαραίτητος ο χαρακτηρισμός μεγάλου πλήθους εικόνων. Δυστυχώς η διαδικασία του χαρακτηρισμού είναι αρκετά κουραστική. Η ευχρηστία του προγράμματος χαρακτηρισμού παίζει καθοριστικό ρόλο σε αυτό τον τομέα. Το πρόγραμμα χαρακτηρισμού θα πρέπει να κάνει τις λιγότερες δυνατές ερωτήσεις στο χρήστη, δίνοντάς του όμως μεγάλη ευελιξία. Θα πρέπει να

δίνεται στο χρήστη η δυνατότητα να διορθώσει τα λάθη που έχει κάνει. Επίσης, λόγω του συχνά μεγάλου πλήθους εικόνων που πρέπει να χαρακτηριστούν, το πρόγραμμα θα πρέπει να απαιτεί τις λιγότερες δυνατές κινήσεις από το χρήστη (είτε στο πληκτρολόγιο, είτε με το ποντίκι). Πιστεύουμε ότι το πρόγραμμα που κατασκευάσαμε, παρέχει πολλές ευκολίες στο χρήστη, ώστε η διαδικασία του χαρακτηρισμού να γίνεται όσο το δυνατόν ευκολότερα.

Στις περισσότερες περιπτώσεις χαρακτηρισμού εικόνων, χρειάζεται να χαρακτηρίσουμε συγκεκριμένα αντικείμενα σε κάθε εικόνα. Είτε θέλουμε να εκπαιδεύσουμε έναν ανιχνευτή, είτε πολλούς, ξέρουμε εκ των προτέρων ποια αντικείμενα μας ενδιαφέρουν και θέλουμε να χαρακτηρίσουμε στις εικόνες που έχουμε στη διάθεσή μας. Η παρατήρηση αυτή μπορεί να βελτιώσει τρομερά την ευχρηστία του προγράμματος χαρακτηρισμού. Δίνοντας ο χρήστης εκ των προτέρων τις ονομασίες των αντικειμένων που θέλει να ανιχνεύσει, θα έχει τη δυνατότητα αντί να πληκτρολογεί κάθε φορά το όνομα του αντικειμένου που χαρακτήρισε, να διαλέγει ποιο αντικείμενο είναι αυτό από μια σειρά διαθέσιμων επιλογών. Με τον τρόπο αυτό μειώνονται σημαντικά οι κινήσεις που θα πρέπει να κάνει ο χρήστης πάνω στο πληκτρολόγιο, κι έτσι απλοποιείται πολύ η διαδικασία.

Ένα άλλο βήμα που μπορεί να απλοποιήσει τη διαδικασία χαρακτηρισμού εικόνων, είναι αυτό του ημιαυτόματου χαρακτηρισμού. Το πρόγραμμα χαρακτηρισμού θα πρέπει να εντοπίζει αυτόματα τα αντικείμενα που μας ενδιαφέρουν. Αυτά θα προτείνονται στο χρήστη, και ο χρήστης θα πρέπει να μπορεί να τα τροποποιήσει, να τα σβήσει, ή και να προσθέσει άλλα, τα οποία δεν ανιχνεύθηκαν αυτόματα. Αν ο αυτόματος εντοπισμός είναι αποτελεσματικός, η διαδικασία επιταχύνεται σε μεγάλο βαθμό, καθώς ο χρήστης περισσότερο ελέγχει, παρά πραγματοποιεί τη διαδικασία. Έτσι, μειώνεται και η καταπόνηση του χρήστη, καθώς δε χρειάζεται να ελέγχει εξονυχιστικά τις εικόνες, αφού τα περισσότερα αντικείμενα εμφανίζονται αυτόματα. Το πρόγραμμα χαρακτηρισμού που κατασκευάσαμε λειτουργεί και με ημιαυτόματο τρόπο, χρησιμοποιώντας ανιχνευτές βασισμένους στα χαρακτηριστικά τύπου Haar που εξετάσαμε σε αυτή τη διπλωματική.

Το πρόγραμμα χαρακτηρισμού εικόνων που κατασκευάσαμε υποστηρίζει χειροκίνητο και ημιαυτόματο τρόπο λειτουργίας. Για να χρησιμοποιηθεί ο ημιαυτόματος τρόπος λειτουργίας, θα πρέπει να έχουμε στη διάθεσή μας έναν ανιχνευτή όπως αυτοί που παρουσιάστηκαν στο Κεφάλαιο 4. Κάποιοι τέτοιοι ανιχνευτές υπάρχουν διαθέσιμοι ελεύθερα. Αν όμως θέλουμε να χαρακτηρίσουμε ένα αντικείμενο για το οποίο δε βρίσκουμε τον κατάλληλο ανιχνευτή, θα πρέπει να τον εκπαιδεύσουμε μόνοι μας. Μπορούμε να χαρακτηρίσουμε χειροκίνητα όλες τις εικόνες που θα χρειαστούμε για την εκπαίδευση και μετά να χρησιμοποιήσουμε ημιαυτόματο χαρακτηρισμό για τις εικόνες που θα χρειαστούμε για την αξιολόγηση του ανιχνευτή. Εναλλακτικά, μπορούμε να ανιχνεύσουμε χειροκίνητα ένα μέρος μόνο του συνόλου εκπαίδευσης και να εκπαιδεύσουμε έναν προσωρινό ανιχνευτή τον οποίο θα χρησιμοποιήσουμε για να χαρακτηρίσουμε τις υπόλοιπες εικόνες ημιαυτόματα. Έπειτα, στο χαρακτηρισμό των εικόνων δοκιμής, μπορούμε να χρησιμοποιήσουμε τον τελικό ανιχνευτή που εκπαιδεύσαμε.

## 5.2 Περιγραφή προγράμματος

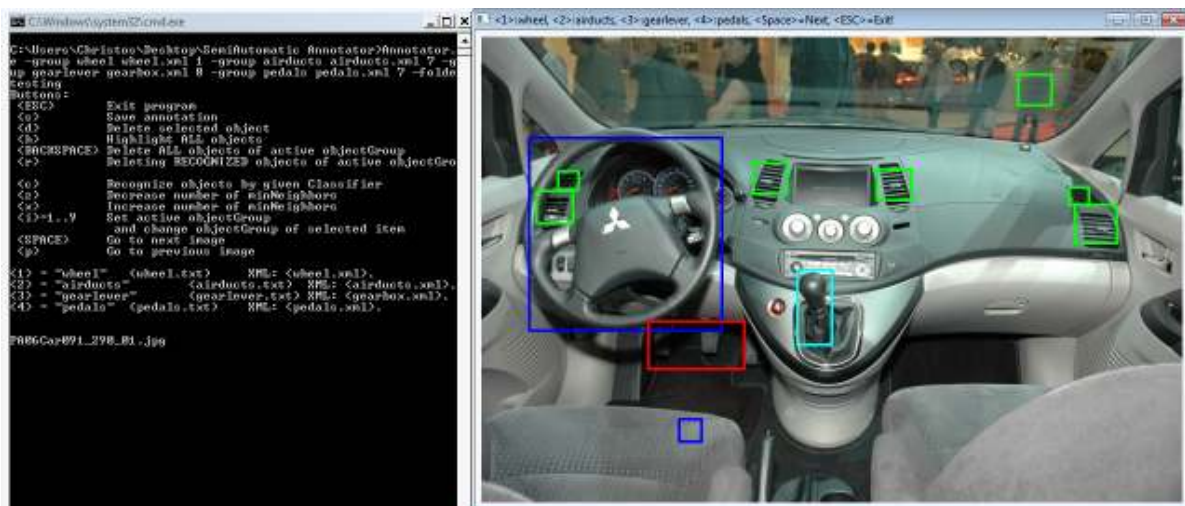
Το πρόγραμμα χαρακτηρισμού εικόνων Annotator που παρουσιάζουμε, δημιουργήθηκε σε γλώσσα προγραμματισμού C++. Επιλέξαμε αυτή τη γλώσσα προγραμματισμού καθώς παρέχει πάρα πολλές δυνατότητες στον προγραμματιστή, όπως δείκτες και αντικειμενοστραφή προγραμματισμό. Για την υλοποίηση της διεπαφής με το χρήστη, τη διαχείριση των εικόνων και του ποντικιού, χρησιμοποιήσαμε τη βιβλιοθήκη OpenCV [OpenCV]. Την OpenCV χρησιμοποιήσαμε και για την εκπαίδευση των ανιχνευτών που χρησιμοποιήσαμε σε αυτή τη διπλωματική, όπως είδαμε σε προηγούμενο κεφάλαιο (Κεφάλαιο 4). Η βιβλιοθήκη αυτή παρέχει συναρτήσεις για διαχείριση και επεξεργασία εικόνων. Επίσης υποστηρίζει την διαδικασία ανίχνευσης αντικειμένων που χρειαζόμαστε για τον ημιαυτόματο χαρακτηρισμό. Τέλος, δίνει και κάποιες βασικές δυνατότητες χρήσης του ποντικιού ως συσκευή εισόδου.

Το πρόγραμμα Annotator εκτελείται από τη γραμμή εντολών, όπου θα πρέπει να του δοθούν οι παράμετροι λειτουργίας ως παράμετροι εισόδου. Αυτές περιγράφονται αναλυτικά στην επόμενη ενότητα (5.2.1). Με την έναρξη του προγράμματος εμφανίζεται μια περιληπτική περιγραφή της λειτουργίας κάθε πλήκτρου του πληκτρολογίου. Έπειτα, εμφανίζονται τα ονόματα των ομάδων αντικειμένων που έχουν δηλωθεί, καθώς και το πλήκτρο στο οποίο αντιστοιχεί κάθε ομάδα και ο ανιχνευτής της ομάδας, αν έχει οριστεί.

Το πρόγραμμα Annotator υποστηρίζει τη διακοπή της διαδικασίας και συνέχισή της. Έτσι, πριν φορτώσει την πρώτη εικόνα που βρίσκεται στο φάκελο που ορίστηκε από το χρήστη, ελέγχει αν υπάρχει προηγούμενος χαρακτηρισμός για κάθε ομάδα αντικειμένων. Αν βρεθεί προηγούμενος χαρακτηρισμός, αυτός φορτώνεται στη μνήμη και γίνεται διαθέσιμος για επεξεργασία.

Αν έχουν οριστεί ανιχνευτές και κάποια αντικείμενα, αυτοί εφαρμόζονται στην τρέχουσα εικόνα πριν αυτή εμφανιστεί στο χρήστη. Ακόμα και αν έχουν οριστεί ανιχνευτές, όταν βρεθεί προηγούμενος χαρακτηρισμός για σε εικόνα (είτε από προηγούμενη εκτέλεση του προγράμματος, είτε από την ίδια εκτέλεση) για έστω και μία ομάδα αντικειμένων, δεν εφαρμόζεται κανένας ανιχνευτής. Αυτό συμβαίνει, καθώς θεωρούμε ότι ο χρήστης έχει ήδη χαρακτηρίσει την εικόνα, και προσθέτοντας τα αποτελέσματα της ανίχνευσης θα χειροτερέψουμε τον υπάρχοντα χαρακτηρισμό. Ο χρήστης, βέβαια, μπορεί να επιλέξει να εφαρμοστεί η ανίχνευση οποιασδήποτε κατηγορίας αντικειμένων σε μια εικόνα ανά πάσα στιγμή.

Κάθε εικόνα που επεξεργάζεται εμφανίζεται σε ένα παράθυρο, όπου γίνεται και ο χαρακτηρισμός της με το ποντίκι. Στο αρχικό παράθυρο της γραμμής εντολών φαίνεται το όνομα του αρχείου εικόνας. Όταν αλλάζουμε εικόνα (πηγαίνοντας είτε στην επόμενη, είτε στην προηγούμενη σε σειρά), προβάλλεται το σύνολο του χαρακτηρισμού που αποθηκεύεται στη μνήμη για την εικόνα που αφήσαμε. Η διαδικασία αυτή επαναλαμβάνεται μέχρι ο χρήστης να επιλέξει την έξοδο του από το πρόγραμμα. Σημειώνεται, ότι κατά τη διάρκεια της διαδικασίας, όλα τα δεδομένα αποθηκεύονται στην κύρια μνήμη του υπολογιστή και όχι σε κάποιο αρχείο στο δίσκο. Έτσι, ο χρήστης θα πρέπει να επιλέγει την αποθήκευση του χαρακτηρισμού, όπως θα δούμε στην επόμενη ενότητα (5.2.2). Κατά την έξοδο από το πρόγραμμα ο χρήστης ρωτάται αν επιθυμεί την αποθήκευση του χαρακτηρισμού. Στο Σχήμα 5.1 φαίνονται τα παράθυρα του προγράμματος και οι πληροφορίες στο παράθυρο της γραμμής εντολών.



(α)



(β)

Σχήμα 5.1 - Διεπαφή του προγράμματος χαρακτηρισμού εικόνων Annotator. α) Έναρξη του προγράμματος. Τα αντικείμενα που φαίνονται είναι οι προτάσεις του προγράμματος ημιαυτόματου χαρακτηρισμού. β) Βλέπουμε την περιγραφή των αντικειμένων της πρώτης εικόνας στο παράθυρο εντολών και τις προτάσεις του προγράμματος για την δεύτερη εικόνα.

### 5.2.1 Παράμετροι εισόδου

Το πρόγραμμα χαρακτηρισμού που κατασκευάσαμε λειτουργεί με χειροκίνητο, αλλά και με ημιαυτόματο τρόπο. Η επιλογή της λειτουργίας γίνεται κατά την έναρξη, από τις παραμέτρους που δίνονται. Παρακάτω περιγράφουμε τις παραμέτρους λειτουργίας του προγράμματος κατά την κλήση του.

```
Annotator.exe -group όνομα_ομάδας [ανιχνευτής [ελάχιστοι_γείτονες]]  
-folder όνομα_φακέλου
```

Με την παράμετρο `group` ορίζουμε κάθε κατηγορία αντικειμένων που θέλουμε να χαρακτηρίσουμε. Το `όνομα_ομάδας` είναι το όνομα που περιγράφει τα αντικείμενα. Κάθε ομάδα αντικειμένων αντιστοιχίζεται σε ένα από τα πλήκτρα 1,2,...,9 ανάλογα με τη σειρά που δηλώνονται. Έτσι, η πρώτη ομάδα που θα δηλωθεί στις παραμέτρους θα αντιστοιχεί στο πλήκτρο 1, η δεύτερη στο πλήκτρο 2 κ.ο.κ. Τα αντικείμενα που ανήκουν σε κάθε ομάδα, σημειώνονται με διαφορετικό χρώμα, για να ξεχωρίζουμε τις ομάδες μεταξύ τους. Ο μέγιστος αριθμός ομάδων αντικειμένων είναι 9, αριθμός αρκετός για τον τρόπο χαρακτηρισμού που περιγράψαμε στην εισαγωγή. Από το όνομα της ομάδας προκύπτει και το όνομα του αρχείου εξόδου. Αυτό θα έχει για όνομα το `όνομα_ομάδας`, και για κατάληξη την τυπική κατάληξη των αρχείων κειμένου ".txt".

Για κάθε ομάδα αντικειμένων που θα χρησιμοποιήσουμε, μπορούμε να επιλέξουμε ξεχωριστά, αν ο χαρακτηρισμός θα γίνει χειροκίνητα ή με ημιαυτόματο τρόπο. Αν θέλουμε για μια συγκεκριμένη ομάδα να γίνει χειροκίνητα ο χαρακτηρισμός, δε δίνουμε καμία άλλη παράμετρο για την ομάδα αυτή. Αν θέλουμε ο χαρακτηρισμός να γίνει ημιαυτόματα, θα πρέπει να δηλώσουμε το όνομα του ανιχνευτή που θα χρησιμοποιηθεί στη θέση `ανιχνευτής`. Είδαμε ότι μια σημαντική παράμετρος κατά την ανίχνευση είναι ο αριθμός των ελάχιστων γειτόνων. Μπορούμε να τον δηλώσουμε με την παράμετρο `ελάχιστοι_γείτονες`. Αν δεν δηλωθεί χρησιμοποιείται από προεπιλογή ο αριθμός 3. Με την παράμετρο `folder` δηλώνεται ο φάκελος που περιέχει τις εικόνες που θέλουμε να χαρακτηρίσουμε, στη θέση `όνομα_φακέλου`.

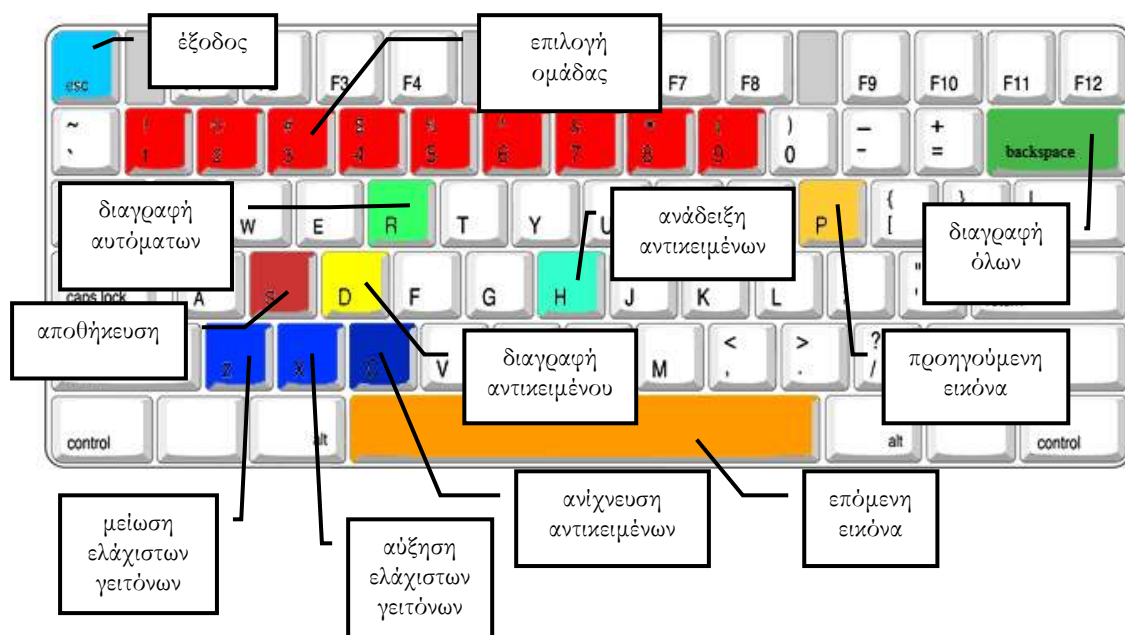
### 5.2.2 Λειτουργίες πληκτρολογίου

Κατά την εκτέλεση του προγράμματος χρησιμοποιούμε κάποια πλήκτρα του πληκτρολογίου για διάφορες λειτουργίες. Η αντιστοίχιση κάθε πλήκτρου με τη λειτουργία του περιγράφεται παρακάτω και φαίνεται στο Σχήμα 5.2. Πατώντας οποιοδήποτε πλήκτρο εκτός από αυτά που περιγράφονται παρακάτω, απλά ανανεώνεται η οθόνη.

<ESC> Έξοδος από το πρόγραμμα. Αμέσως πριν την έξοδο, το πρόγραμμα μας ρωτά αν θέλουμε να αποθηκεύσουμε τον χαρακτηρισμό που έχουμε κάνει μέχρι τη στιγμή της εξόδου.

- <s> Αποθήκευση χαρακτηρισμού στα αρχεία εξόδου.
- <d> Διαγραφή επιλεγμένου αντικειμένου.
- <h> Ανάδειξη όλων των αντικειμένων. Με τη λειτουργία αυτή επιβλέπουμε εύκολα τα αντικείμενα που χαρακτηρίσαμε στην εικόνα.
- <Backspace> Διαγραφή ΟΛΩΝ των αντικειμένων της επιλεγμένης ομάδας στην τρέχουσα εικόνα. Το πρόγραμμα ζητά επιβεβαίωση για τη λειτουργία αυτή.
- <r> Διαγραφή των αντικειμένων της επιλεγμένης ομάδας που προέκυψαν με αυτόματο τρόπο στην τρέχουσα εικόνα. Δεν διαγράφονται όσα αντικείμενα προστέθηκαν από το χρήστη. Επίσης, δε διαγράφονται ούτε τα αντικείμενα που ενώ προέκυψαν με αυτόματο τρόπο, έχουν μετακινηθεί ή αλλάξει μέγεθος από το χρήστη. Το πρόγραμμα δεν ζητά επιβεβαίωση για τη λειτουργία αυτή.
- <c> Εκτέλεση ανίχνευσης για την επιλεγμένη ομάδα.
- <z> Μείωση και αύξηση κατά 1 του αριθμού ελάχιστων γειτόνων που πρέπει να έχει  
<x> ένα αποτέλεσμα της ανίχνευσης για την επιλεγμένη ομάδα για να εμφανιστεί. Κατά την αλλαγή αυτή, τα αποτελέσματα που ανήκουν στην επιλεγμένη ομάδα και προέκυψαν αυτόματα διαγράφονται, και εμφανίζονται τα αποτελέσματα που αντιστοιχούν στην νέα τιμή της παραμέτρου. Αν κάποιο αντικείμενο έχει διαγραφεί, μετακινηθεί ή αλλάξει το μέγεθός του, δεν επιστρέφεται ξανά από τον ανιχνευτή.
- <i>=1..9 Επιλογή της ενεργής ομάδας αντικειμένων και αλλαγή της ομάδας του επιλεγμένου αντικειμένου. Κάθε νέο αντικείμενο που προστίθεται θα ανήκει στην ομάδα αυτή. Αν υπάρχει ήδη κάποιο αντικείμενο επιλεγμένο, τότε αλλάζει η ενεργή ομάδα και το επιλεγμένο αντικείμενο ανήκει πλέον στην ομάδα που επιλέχθηκε.
- <space> Μετάβαση στην επόμενη και προηγούμενη εικόνα. Οι εικόνες διατάσσονται σε  
<p> κύκλο, οπότε η εικόνα μετά την τελευταία είναι η πρώτη, και η εικόνα πριν την πρώτη είναι η τελευταία.





Σχήμα 5.2 - Λειτουργίες πληκτρολογίου

### 5.2.3 Λειτουργίες ποντικιού

Το ποντίκι εκτελεί τις πιο σημαντικές λειτουργίες του προγράμματος χαρακτηρισμού. Με το ποντίκι προσθέτουμε ένα νέο αντικείμενο, αλλάζουμε τις διαστάσεις ή μετακινούμε ένα υπάρχον αντικείμενο και επιλέγουμε αντικείμενα. Η λογική των χειρισμών είναι ίδια με αυτή των περισσότερων σχεδιαστικών προγραμμάτων.

Για να δημιουργήσουμε ένα αντικείμενο, αρκεί να πατήσουμε με το αριστερό κουμπί του ποντικιού (αριστερό κλικ) σε ένα σημείο που θέλουμε να αποτελεί ένα άκρο του αντικειμένου, και κρατώντας το κουμπί πατημένο μετακινούμαστε μέχρι το αντιδιαμετρικό άκρο του αντικειμένου.

Κάνοντας ένα απλό αριστερό κλικ πάνω σε ένα αντικείμενο, το επιλέγουμε. Το επιλεγμένο αντικείμενο ξεχωρίζει από τα υπόλοιπα, καθώς το περίγραμμά του είναι πιο έντονο. Έπειτα, μπορούμε είτε να το μετακινήσουμε, είτε να αλλάξουμε τις διαστάσεις του. Για να το μετακινήσουμε κάνουμε αριστερό κλικ στο εσωτερικό του αντικειμένου, και κρατώντας το κουμπί πατημένο, μετακινούμε το αντικείμενο. Για να αλλάξουμε τις διαστάσεις του, κάνουμε αριστερό κλικ στο άκρο ή την πλευρά του αντικειμένου που θέλουμε να διορθώσουμε, και κρατώντας το κουμπί πατημένο, μετακινούμε το συγκεκριμένο άκρο ή πλευρά στη θέση που θέλουμε. Ο δείκτης του ποντικιού μεταβάλλεται ανάλογα και στις δύο περιπτώσεις, ώστε να αντιλαμβάνεται ο χρήστης την ενέργεια που πρόκειται να συμβεί. Οι δείκτες που χρησιμοποιήθηκαν είναι οι προεπιλογές του λειτουργικού συστήματος για τις ίδιες εργασίες σε παράθυρα προγραμμάτων.

### 5.2.4 Αρχεία χαρακτηρισμού εικόνων

Ο τρόπος αποθήκευση του χαρακτηρισμού των εικόνων είναι αυτός που χρησιμοποιείται από τη βιβλιοθήκη OpenCV [OpenCV]. Έτσι, με το τέλος της διαδικασίας χαρακτηρισμού μπορούμε να χρησιμοποιήσουμε τα αρχεία εξόδου κατευθείαν για την εκπαίδευση ενός ανιχνευτή.

Για κάθε ομάδα αντικειμένων χρησιμοποιούμε ένα αρχείο κειμένου. Σε αυτό το αρχείο περιλαμβάνεται ο χαρακτηρισμός όλων των εικόνων για τη συγκεκριμένη ομάδα αντικειμένων. Για κάθε εικόνα που έχει χαρακτηριστεί υπάρχει μια γραμμή στο αρχείο. Το πρώτο στοιχείο κάθε γραμμής είναι το όνομα της εικόνας, συμπεριλαμβανομένης της διαδρομής φακέλων από το αρχείο με το χαρακτηρισμό, ως την εικόνα. Το δεύτερο στοιχείο είναι ο αριθμός των αντικειμένων της ομάδας που εμφανίζονται στη συγκεκριμένη εικόνα. Έπειτα ακολουθεί μία τετράδα αριθμών για κάθε αντικείμενο, που αποτελείται από: α) την οριζόντια συντεταγμένη του επάνω αριστερού άκρου του παραλληλόγραμμου που περικλείει το αντικείμενο, β) την κατακόρυφη συντεταγμένη του ίδιου σημείου, γ) το πλάτος του παραλληλόγραμμου και δ) το ύψος του. Έτσι, αν μια εικόνα περιλαμβάνει  $n$  αντικείμενα μιας ομάδας, τότε στο αρχείο με το χαρακτηρισμό για τη συγκεκριμένη ομάδα θα υπάρχει μια γραμμή για αυτή την εικόνα με  $4n+2$  στοιχεία. Το πρώτο θα είναι το όνομα της εικόνας, το δεύτερο ο αριθμός  $n$  και θα ακολουθούν  $n$  τετράδες με τις συντεταγμένες και τις διαστάσεις κάθε αντικειμένου. Στο Σχήμα 5.3 φαίνεται ένα παράδειγμα αρχείου εξόδου του προγράμματος χαρακτηρισμού.

```
TestImages/test-0.pgm 1 28 54 93 29
TestImages/test-1.pgm 2 30 69 83 29 138 69 101 31
TestImages/test-10.pgm 2 11 52 72 26 113 51 75 26
TestImages/test-100.pgm 1 12 38 83 26
TestImages/test-101.pgm 1 19 26 84 24
TestImages/test-102.pgm 1 8 35 84 28
TestImages/test-103.pgm 1 99 80 80 25
TestImages/test-104.pgm 2 134 60 75 25 17 53 85 31
TestImages/test-105.pgm 2 145 72 81 28 45 70 82 25
TestImages/test-106.pgm 1 78 59 83 26
TestImages/test-107.pgm 1 28 50 86 24
...
```

Σχήμα 5.3 - Παράδειγμα αρχείου εξόδου προγράμματος χαρακτηρισμού εικόνων

## Κεφάλαιο 6.

### Συμπεράσματα και μελλοντικές επεκτάσεις

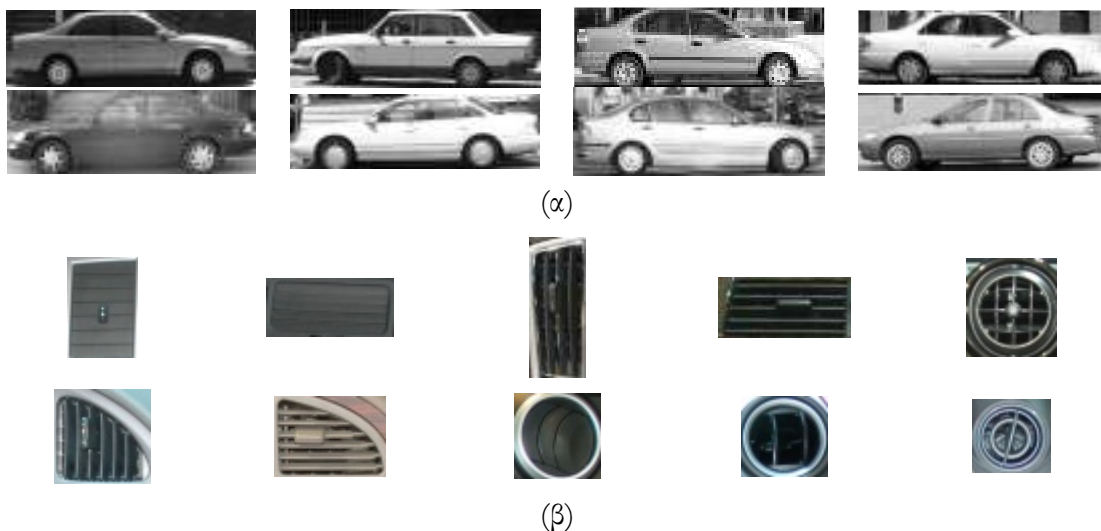
Σε αυτή τη διπλωματική ασχοληθήκαμε με την ανίχνευση αντικειμένων και τον ημιαυτόματο χαρακτηρισμό εικόνων. Έπειτα από πειράματα στους δύο αυτούς τομείς, προκύπτουν κάποια συμπεράσματα για τις μεθόδους που χρησιμοποιήσαμε και υλοποιήσαμε, καθώς και κάποιες επεκτάσεις που θα μπορούσαν να υλοποιηθούν μελλοντικά

#### 6.1 Ανίχνευση αντικειμένων με χαρακτηριστικά τύπου Haar

Στο Κεφάλαιο 4 παρουσιάστηκαν 11 ανιχνευτές που βασίζονται στα χαρακτηριστικά τύπου Haar. Είδαμε για τον κάθε ένα τις επιλογές που έγιναν για την εκπαίδευσή του, τα παραδείγματα εικόνων που χρησιμοποιήθηκαν και τέλος την απόδοσή του σε ένα σύνολο εικόνων δοκιμής. Παρουσιάζοντας τα αποτελέσματα κάθε δοκιμής, τα σχολιάσαμε και προσπαθήσαμε να τα εξηγήσουμε. Από το σχολιασμό όλων αυτών, βγαίνουν κάποια συμπεράσματα για τις επιλογές που πρέπει να γίνουν κατά την εκπαίδευση ενός ανιχνευτή τέτοιου τύπου.

Είδαμε τον καθοριστικό ρόλο που παίζει η ομοιογένεια των παραδειγμάτων εκπαίδευσης, ως προς τα γεωμετρικά χαρακτηριστικά τους και τη θέση των γεωμετρικών αυτών χαρακτηριστικών στο αντικείμενο. Ο ανιχνευτής του πλαϊνού μέρους των αυτοκινήτων εκπαιδεύτηκε με παραδείγματα τα οποία είχαν πολύ μεγάλη ομοιογένεια. Η γωνία υπό την οποία εμφανίζονταν τα αντικείμενα ήταν πολύ περιορισμένη και όλα τα αντικείμενα είχαν περίπου τα ίδια γεωμετρικά χαρακτηριστικά στις ίδιες θέσεις. Έτσι, προέκυψε ένας ανιχνευτής με εξαιρετικές επιδόσεις, έχοντας επανάκληση μέχρι και 0,95 και πολύ υψηλή ακρίβεια. Αντίθετα, στην περίπτωση των αεραγωγών, υπήρχαν πολύ έντονες διαφορές ως προς το σχήμα, τη γωνία εμφάνισης των αντικειμένων, τα γεωμετρικά χαρακτηριστικά τους και τη θέση αυτών των χαρακτηριστικών. Έτσι, προέκυψε ένας ανιχνευτής με χαμηλότερες επιδόσεις, του οποίου η επανάκληση έφτανε μέχρι το 0,7 και η ακρίβεια έπεφτε δραματικά καθώς η επανάκληση ξεπερνούσε το 0,6. Οι διαφορές στα παραδείγματα φαίνονται στις εικόνες στο Σχήμα 6.1.

Είδαμε ακόμα, τον ρόλο που παίζει το πλήθος των γεωμετρικών χαρακτηριστικών των παραδειγμάτων εκπαίδευσης, ανεξάρτητα από την ομοιογένειά τους. Τα παραδείγματα εκπαίδευσης του ανιχνευτή για τους μοχλούς ταχυτήτων ήταν αρκετά ομοιογενή, κι έτσι τα γεωμετρικά χαρακτηριστικά τους μπορούσαν εύκολα να καθοριστούν. Το αντικείμενο "μοχλός ταχυτήτων", όμως, είναι αρκετά απλό στην περιγραφή και δεν αποτελείται από πολλά και ιδιαίτερα γεωμετρικά χαρακτηριστικά. Έτσι, αν και η επανάκληση του ανιχνευτή ήταν σχετικά υψηλή, παρουσίαζε χαμηλή ακρίβεια, με αποτέλεσμα η απόδοσή του να μην θεωρείται ιδιαίτερα υψηλή.



Σχήμα 6.1 - Ομοιογένεια θετικών παραδειγμάτων. α) Θετικά παραδείγματα με μεγάλη ομοιογένεια, β) θετικά παραδείγματα με ελάχιστη ομοιογένεια

Παρατηρήσαμε την επίδραση της παραμέτρου του μέγιστου λόγου λανθασμένων θετικών ανιχνεύσεων ανά στάδιο. Όταν ορίζαμε την τιμή αυτή αρκετά χαμηλά, τα στάδια του ανιχνευτή αποτελούνταν από περισσότερα χαρακτηριστικά τύπου Haar, ήταν δηλαδή πιο πολύπλοκα. Αυτό είχε ως συνέπεια να χρειάζεται περισσότερος χρόνος για να ολοκληρωθεί η ανίχνευση.

Εφαρμόσαμε τους ανιχνευτές που εκπαιδεύσαμε σε σύνολα εικόνων που έχουν χρησιμοποιηθεί ευρέως στη βιβλιογραφία για πειράματα. Συγκρίνοντας τα αποτελέσματα των ανιχνευτών μας με άλλες εργασίες, βλέπουμε ότι τα αποτελέσματα που επιτύχαμε είναι ανάμεσα στα καλύτερα που συναντούμε στη βιβλιογραφία. Επίσης, βλέπουμε ότι η μέθοδος δίνει πολύ καλά αποτελέσματα σε ένα μεγάλο φάσμα αντικειμένων στα οποία τη δοκιμάσαμε. Τέλος, δεδομένης της ταχύτητας της διαδικασίας, μπορούμε να πούμε ότι οι ανιχνευτές με χαρακτηριστικά τύπου Haar μπορούν να χρησιμοποιηθούν σε πολλές πρακτικές εφαρμογές ανίχνευσης αντικειμένων με επιτυχία.

Μια επέκταση μου θα μπορούσε να γίνει σε αυτή τη μέθοδο ανίχνευσης, είναι η εξαγωγή κάποιου είδους εμπιστοσύνης για τα αποτελέσματα. Στην υπάρχουσα μορφή της μεθόδου, τα αποτελέσματα της ανίχνευσης είναι τα υποπαράθυρα της εικόνας τα οποία ταξινομήθηκαν ως θετικά, χωρίς καμία επιπλέον πληροφορία. Το μόνο μέτρο που μπορεί να χρησιμοποιηθεί και να δώσει κάποιο μέτρο εμπιστοσύνης, είναι ο αριθμός των γειτονικών υποπαράθυρων που ταξινομούνται ως θετικά. Το σκεπτικό αυτό βασίζεται στην παρατήρηση ότι στην περιοχή ενός πραγματικού αντικειμένου είναι πολύ πιθανό να γίνουν πολλαπλές θετικές ταξινομήσεις. Αντίθετα, σε μια περιοχή που δεν υπάρχει το αντικείμενο το οποίο αναζητάμε, είναι λιγότερο πιθανό να γίνουν πολλαπλές θετικές ταξινομήσεις. Έτσι, μπορούμε να θεωρήσουμε τον αριθμό των γειτόνων ενός αποτελέσματος της διαδικασίας, ως μέτρο εμπιστοσύνης για την ορθότητα της ανίχνευσης.

Όπως αναφέραμε, για κάθε υποπαράθυρο της εικόνας που εξετάζεται, το αποτέλεσμα της ανίχνευσης είναι αν το συγκεκριμένο υποπαράθυρο περιλαμβάνει ή όχι το αντικείμενο που αναζητάμε. Για να βγει το αποτέλεσμα αυτό, είδαμε ότι οι τιμές των χαρακτηριστικών τύπου Haar που προκύπτουν συγκρίνονται με διάφορα κατώφλια κατά τον έλεγχο με τους διαδοχικά συνδεδεμένους ταξινομητές. Ένα άλλο μέτρο εμπιστοσύνης, λοιπόν, θα μπορούσε να αποτελεί η τιμή που συγκρίνεται με το κατώφλι του κάθε ταξινομητή. Έτσι, όσο μεγαλύτερη είναι αυτή η τιμή

από το κατώφλι, τόσο πιο πιθανό είναι η ταξινόμηση να είναι σωστή. Αντίθετα, όσο πιο οριακά ξεπερνιούνται τα κατώφλια αυτά, η πιθανότητα να είναι σωστή η ταξινόμηση μειώνεται.

Άλλη επέκταση της δουλειάς που κάναμε σε αυτή τη διπλωματική, θα μπορούσε να αποτελεί ο διαχωρισμός των θετικών παραδειγμάτων εκπαίδευσης σε υποκατηγορίες, όπως εφαρμόστηκε στην περίπτωση της ανίχνευσης του εξωτερικού μέρους αυτοκινήτων. Θα μπορούσαμε, λοιπόν, να χωρίσουμε κάποιες κατηγορίες αντικειμένων σε υποκατηγορίες, ώστε να αυξηθεί η ομοιογένεια ανάμεσα στα παραδείγματα κάθε υποκατηγορίας, και να εκπαιδεύσουμε έναν ανιχνευτή για κάθε υποκατηγορία. Ακόμα και στην περίπτωση κατηγοριών αντικειμένων που χωρίστηκαν σε υποκατηγορίες, θα μπορούσε να συνεχιστεί η δουλειά που κάναμε, βρίσκοντας περισσότερα παραδείγματα και χωρίζοντάς τα σε ακόμα περισσότερες υποκατηγορίες. Έτσι, κάθε ανιχνευτής θα έχει εκπαιδευτεί με παραδείγματα με μεγαλύτερη ομοιογένεια μεταξύ τους, και συνολικά η διαδικασία της ανίχνευσης θα είναι πιο ακριβής. Βέβαια, θα πρέπει να έχουμε πάντα υπόψη, ότι οι περισσότεροι ανιχνευτές αυξάνουν το χρόνο ανίχνευσης και μειώνουν τη γενικότητα της διαδικασίας, οπότε δε θα πρέπει να γίνεται υπερβολική κατάτμηση των κατηγοριών αντικειμένων.

## 6.2 Ημιαυτόματος χαρακτηρισμός εικόνων

Ο χαρακτηρισμός μιας μεγάλης συλλογής εικόνων είναι μια κουραστική και χρονοβόρα διαδικασία. Γι' αυτό, το πιο σημαντικό κριτήριο για την επιλογή ενός εργαλείου χαρακτηρισμού είναι η ευχρηστία του και οι ευκολίες που παρέχει ώστε να μειωθεί όσο το δυνατόν περισσότερο ο χρόνος που απαιτείται για την ολοκλήρωση της διαδικασίας. Στην προσπάθεια αυτή, κατασκευάσαμε ένα πρόγραμμα χαρακτηρισμού εικόνων που πιστεύουμε ότι έχει μεγάλο βαθμό ευχρηστίας. Ο χαρακτηρισμός κάθε αντικειμένου γίνεται με το ποντίκι, και ο χρήστης μπορεί να κάνει κάθε διόρθωση σε αυτά, όπως να τα μετακινήσει, να αλλάξει το μέγεθός τους και να σβήσει όποιο θέλει. Οι λειτουργίες που γίνονται από το πληκτρολόγιο και χρησιμοποιούνται πιο συχνά, έγινε προσπάθεια να συγκεντρωθούν σε ένα τμήμα του πληκτρολογίου. Έτσι, ελαχιστοποιούνται οι απαιτούμενες κινήσεις από το χρήστη, με αποτέλεσμα να είναι πιο ξεκούραστη η διαδικασία.

Επίσης, το πρόγραμμα αυτό παρέχει δυνατότητες ημιαυτόματου χαρακτηρισμού εικόνων. Για την ημιαυτόματη λειτουργία, απαιτείται η χρήση κάποιας μεθόδου εντοπισμού αντικειμένων. Επιλέξαμε τη χρησιμοποίηση της μεθόδου ανίχνευσης αντικειμένων με χρήση χαρακτηριστικών τύπου Haar, με την οποία ασχοληθήκαμε σε αυτή τη διπλωματική. Η επιλογή αυτή έγινε, καθώς είδαμε ότι η μέθοδος αυτή δίνει καλά αποτελέσματα απαιτώντας πολύ λίγο χρόνο για την ανίχνευση. Συνολικά, είδαμε ότι η ημιαυτόματη λειτουργία του προγράμματος συνεισφέρει πολύ στη μείωση του χρόνου που απαιτείται για τη διαδικασία.

Τέλος, η έξοδος του προγράμματος είναι συγκεντρωμένη σε ένα αρχείο κειμένου για κάθε κατηγορία αντικειμένων, ανεξαρτήτως του αριθμού των εικόνων. Έτσι, η συνολική έξοδος του προγράμματος αποτελείται από πολύ λίγα αρχεία, γεγονός που βοηθά στην εύκολη μεταφορά, διαχείριση και αποθήκευσή τους.

Το πρόγραμμα χαρακτηρισμού που κατασκευάσαμε τροποποιήθηκε πολλές φορές μέχρι να φτάσει στη μορφή που παρουσιάσαμε. Λειτουργίες προστέθηκαν και τροποποιήθηκαν πολλές φορές, ώστε να επιτευχθεί η μέγιστη δυνατή λειτουργικότητα. Η βιβλιοθήκη OpenCV που χρησιμοποιήσαμε δεν παρείχε δυνατότητες για κατασκευή μιας πιο προσιτής διεπαφής με το χρήστη. Έτσι χρησιμοποιήθηκε η γραμμή εντολών για επικοινωνία με το χρήστη. Μελλοντικά θα

μπορούσαμε να βελτιώσουμε τη διεπαφή με το χρήστη, με τη χρήση γραφικών (κουμπιών, μενού κτλ).

Μια άλλη επέκταση που θα μπορούσε να πραγματοποιηθεί θα ήταν η υποστήριξη εικόνων με ανάλυση μεγαλύτερη από την ανάλυση της οθόνης του υπολογιστή στον οποίο τρέχει το πρόγραμμα. Αυτή τη στιγμή, το πρόγραμμα απεικονίζει τις εικόνες που πρόκειται να χαρακτηριστούν στις φυσικές τους διαστάσεις. Έτσι, αν μια εικόνα έχει ανάλυση που ξεπερνά αυτή της οθόνης του χρήστη, δε θα είναι ορατή όλη η εικόνα.

Μια μεγάλη επέκταση θα μπορούσε να είναι η τροποποίηση του τρόπου με τον οποίο υλοποιείται η ημιαυτόματη λειτουργία του προγράμματος. Είδαμε, ότι για να έχουμε ημιαυτόματο χαρακτηρισμό θα πρέπει να έχουμε έναν έτοιμο ανιχνευτή. Θα μπορούσαμε να χρησιμοποιήσουμε μια μέθοδο εκπαίδευσης για ανιχνευτές, η οποία να λειτουργεί παράλληλα με τη διαδικασία χαρακτηρισμού και να υποστηρίζει προσθετική εκπαίδευση. Δηλαδή, να εκπαιδεύει έναν ανιχνευτή, και κάθε φορά που δίνονται νέα παραδείγματα ο ανιχνευτής να βελτιώνεται. Έτσι, θα μπορούσαμε να εκπαιδεύουμε τον ανιχνευτή κατά τη διάρκεια της διαδικασίας χαρακτηρισμού. Έπειτα από το χαρακτηρισμό ενός αριθμού εικόνων, όταν ο ανιχνευτής δε θα παρουσιάζει υψηλό σφάλμα, θα μπορούσε αυτόματα να αρχίσει να προτείνει κάποια υποπαράθυρα στα οποία εντοπίζει τα αντικείμενα τα οποία μας ενδιαφέρουν. Έτσι, η διαδικασία θα απλοποιούνταν αρκετά. Δυστυχώς η μέθοδος που χρησιμοποιήσαμε για την εκπαίδευση των ανιχνευτών μας δεν υποστηρίζει προσθετική λειτουργία και είναι πολύ χρονοβόρα για να εκτελείται παράλληλα με τη διαδικασία χαρακτηρισμού.

# Παράρτημα Α

## Ανίχνευση αντικειμένων με χρήση της OpenCV

Για την εκπαίδευση των ανιχνευτών μας επιλέξαμε τη χρήση της βιβλιοθήκης OpenCV (Open Source Computer Vision Library – Βιβλιοθήκη Όρασης Υπολογιστών Ανοιχτού Κώδικα). Η OpenCV αναπτύχθηκε από την εταιρία Intel και διατίθεται ελεύθερα για χρήση. Είναι μια βιβλιοθήκη συναρτήσεων που έχει ως πεδίο εφαρμογής της την Όραση Υπολογιστών. Περιέχει μεγάλο πλήθος συναρτήσεων για την επεξεργασία και την παρουσίαση εικόνων και βίντεο. Μαζί με τη βιβλιοθήκη συναρτήσεων, παρέχονται και κάποια έτοιμα προγράμματα που κατασκευάζουν έναν ταξινομητή σύμφωνα με την τεχνική που εξετάζουμε. Παρακάτω εξηγούμε τον τρόπο κατασκευής ενός τέτοιου ταξινομητή.

### A.1 Πρόγραμμα προετοιμασίας θετικών παραδειγμάτων

Το πρόγραμμα της OpenCV, `createsamples`, μπορεί να αποκόψει τις περιοχές των εικόνων που έχουν χαρακτηριστεί για ένα είδος αντικειμένων και να τις αποθηκεύσει σαν ξεχωριστές εικόνες. Όλες οι εικόνες που παράγονται αποθηκεύονται σε ένα αρχείο τύπου `vec`. Αφού, λοιπόν, χαρακτηρίσουμε τις εικόνες για το συγκεκριμένο αντικείμενο που μας ενδιαφέρει, μπορούμε να δώσουμε το αρχείο που περιλαμβάνει το χαρακτηρισμό των εικόνων ως είσοδο στο πρόγραμμα `createsamples`. Θα πρέπει ο χαρακτηρισμός όλων των εικόνων να περιλαμβάνεται σε ένα αρχείο κειμένου. Κάθε γραμμή του αρχείου θα περιλαμβάνει το χαρακτηρισμό μιας συγκεκριμένης εικόνας. Το πρώτο στοιχείο κάθε γραμμής θα πρέπει να είναι το όνομα της εικόνας που έχει χαρακτηριστεί. Έπειτα, ακολουθεί ο αριθμός των αντικειμένων του συγκεκριμένου τύπου που έχουν χαρακτηριστεί στην εικόνα. Τέλος, για κάθε αντικείμενο, ακολουθεί μια τετράδα αριθμών που δηλώνουν την οριζόντια και κατακόρυφη συντεταγμένη του πάνω αριστερού άκρου του ορθογωνίου και το πλάτος και ύψος του ορθογωνίου.

Η έξοδος του προγράμματος `createsamples` είναι μια συλλογή όλων των εικόνων σε ένα αρχείο, και όχι πολλά αρχεία εικόνων ξεχωριστά. Για να κατασκευάσουμε τον ταξινομητή, τα θετικά παραδείγματα θα πρέπει να έχουν όλα το ίδιο ακριβώς μέγεθος. Για το λόγο αυτό, δίνουμε ως παράμετρο στο πρόγραμμα `createsamples` το μέγεθος (πλάτος και ύψος) που θέλουμε να έχουν τα θετικά δείγματα. Το πρόγραμμα μεταβάλλει το μέγεθος των θετικών παραδειγμάτων ώστε να ταιριάζει ακριβώς με το επιθυμητό, ακόμα κι αν χρειαστεί να αλλάξει τον λόγο των διαστάσεων τους.

## A.2 Πρόγραμμα εκπαίδευσης ταξινομητή

Η OpenCV παρέχει το πρόγραμμα `haartraining`, το οποίο εκτελεί όλη την εκπαίδευση ενός ταξινομητή, σύμφωνα με τη θεωρία που αναλύσαμε προηγουμένως. Το πρόγραμμα αυτό παίρνει ως παραμέτρους εισόδου όλες τις επιλογές που μπορούμε να κάνουμε για την εκπαίδευση του ταξινομητή μας. Παρακάτω περιγράφονται οι παράμετροι αυτές και εξηγείται η χρήση τους. Οι παράμετροι που βρίσκονται μέσα σε αγκύλες είναι προαιρετικές. Σε περίπτωση που δε δοθεί μια προαιρετική παράμετρος, λαμβάνει μια εξορισμού τιμή, η οποία φαίνεται πριν την αγκύλη κλεισίματος.

`-data <όνομα_καταλόγου>`

Εδώ δίνουμε το όνομα του καταλόγου στον οποίο θα αποθηκευθεί ο ταξινομητής. Για κάθε στάδιο του ΔΣΤ δημιουργείται ένας υποφάκελος με όνομα τον αριθμό του σταδίου. Μέσα σε αυτόν δημιουργείται ένα αρχείο κειμένου, που περιλαμβάνει τις πληροφορίες του ταξινομητή που αποτελεί το στάδιο. Έτσι, για κάθε Haar χαρακτηριστικό του ταξινομητή αναγράφεται ο τύπος του, η θέση και το μέγεθός του καθώς και το κατώφλι του. Στο τέλος αναφέρεται και το συνολικό κατώφλι του ισχυρού ταξινομητή. Στο τέλος της εκπαίδευσης θα δημιουργηθεί και ένα αρχείο τύπου `xml` με το ίδιο όνομα, που θα έχει την ίδια περίπου δομή. Η εκπαίδευση ενός ΔΣΤ μπορεί να διακοπεί και να συνεχιστεί σε οποιοδήποτε σημείο. Όταν επιλέξουμε να συνεχίσουμε την εκπαίδευση ενός υπάρχοντος ΔΣΤ, ξεκινάμε από το τελευταίο υπάρχον στάδιο το οποίο εντοπίζεται στο φάκελο.

`-vec <όνομα_αρχείου_vec>`

Με αυτή την παράμετρο δηλώνουμε το αρχείο τύπου `vec` που δημιουργήσαμε με το πρόγραμμα `createsamples` κατά την προετοιμασία, το οποίο περιλαμβάνει τη συλλογή με τα θετικά παραδείγματα.

`-bg <όνομα_αρχείου_αρνητικών_παραδειγμάτων>`

Με την παράμετρο αυτή δηλώνουμε το όνομα του αρχείου κειμένου που περιλαμβάνει τα αρνητικά παραδείγματα. Κάθε γραμμή του θα πρέπει να έχει ένα όνομα αρχείου εικόνας που θα αποτελέσει πηγή αρνητικών παραδειγμάτων.

`[-npos <αριθμός_θετικών_παραδειγμάτων = 2000>]`

Εδώ δηλώνουμε τον αριθμό των θετικών παραδειγμάτων που θέλουμε να χρησιμοποιηθούν για την εκπαίδευση του ταξινομητή. Η τιμή αυτή πρέπει να είναι μικρότερη ή ίση με το σύνολο των θετικών παραδειγμάτων που περιέχονται στο αρχείο τύπου `vec` που δηλώσαμε. Τα θετικά παραδείγματα επιλέγονται σύμφωνα με τη σειρά με την οποία έχουν αποθηκευθεί στη συλλογή. Αν δοθεί τιμή μεγαλύτερη από το πλήθος των διαθέσιμων θετικών δειγμάτων, τότε η τιμή των δειγμάτων που θα χρησιμοποιηθούν γίνεται ίση με το πλήθος τους. Αν δεν χρησιμοποιήσουμε αυτή την παράμετρο, δηλώνεται αυτόματα η τιμή 2.000 και ισχύουν τα προηγούμενα.



`[-nneg <αριθμός_αρνητικών_παραδειγμάτων = 2000>]`

Η παράμετρος αυτή δηλώνει τον αριθμό των αρνητικών παραδειγμάτων που θέλουμε να χρησιμοποιηθούν σε κάθε στάδιο. Δηλώνουμε δηλαδή, τον αριθμό των αρνητικών παραδειγμάτων που θα πρέπει να ταξινομηθούν ως θετικά από όλα τα προηγούμενα στάδια, ώστε να χρησιμοποιηθούν στο τρέχον στάδιο ως αρνητικά. Τα αρνητικά παραδείγματα επιλέγονται τυχαία από όλα τα διαθέσιμα υποπαράθυρα των εικόνων που δηλώθηκαν με την παράμετρο `bg`. Σε περίπτωση που το πλήθος των θετικών παραδειγμάτων που ορίστηκε προηγουμένως τροποποιηθεί από το πρόγραμμα (αν είναι μεγαλύτερο από τα διαθέσιμα θετικά παραδείγματα), τότε και το πλήθος των χρησιμοποιούμενων αρνητικών παραδειγμάτων θα αλλάξει ανάλογα. Αν δηλαδή, το πλήθος θετικών παραδειγμάτων ορίστηκε ως  $N_{pos,1}$  αλλάζει σε  $N_{pos,2}$ , τότε το πλήθος των αρνητικών παραδειγμάτων θα αλλάξει από  $N_{neg,1}$  που δηλώθηκε, σε  $N_{neg,2} = \frac{N_{neg,1}}{N_{pos,1}} N_{pos,2}$ . Η εξ' ορισμού τιμή είναι 2.000.

`[-nstages <αριθμός_σταδίων = 14>]`

Με αυτή την παράμετρο δηλώνουμε τον αριθμό των σταδίων που θέλουμε να έχει ο ΔΣΤ. Αν ο φάκελος που θα αποθηκευθεί ο ταξινομητής δεν είναι κενός και περιέχει έναν ΔΣΤ με στάδια λιγότερα από αυτά που δηλώνουμε τώρα, συνεχίζεται η εκπαίδευση του υπάρχοντος ΔΣΤ μέχρι να αποκτήσει ο ΔΣΤ τόσα στάδια όσα δηλώνουμε. Αν δεν δηλώσουμε αριθμό σταδίων, κατασκευάζονται 14 στάδια.

`[-nsplits <αριθμός_διαίρεσεων_ = 1>]`

Με αυτή την παράμετρο δηλώνουμε τον αριθμό των χαρακτηριστικών που θα χρησιμοποιηθούν σε κάθε ασθενή ταξινομητή. Για να κατασκευάσουμε τον ταξινομητή όπως περιγράφηκε στη θεωρία, θα πρέπει να δώσουμε την τιμή 1, που είναι και η προεπιλεγμένη τιμή. Δίνοντας μεγαλύτερη τιμή, κάθε ασθενής ταξινομητής θα αποτελείται από περισσότερα από ένα χαρακτηριστικά τύπου Haar.

`[-mem <διαθέσιμη_μνήμη_σε_MB = 200>]`

Δηλώνουμε την ποσότητα μνήμης σε MB που θέλουμε να χρησιμοποιήσει το πρόγραμμα για την εκπαίδευση. Ο υπολογισμός όλων των χαρακτηριστικών σε όλα τα θετικά παραδείγματα αποθηκεύεται στη μνήμη του υπολογιστή, και γενικά απαιτείται πολύς χώρος. Ό,τι δε χωράει στην κύρια μνήμη, αποθηκεύεται στη δευτερεύουσα (σκληρός δίσκος), με αποτέλεσμα να μειώνεται η ταχύτητα της διαδικασίας. Έτσι, όση περισσότερη κύρια μνήμη διαθέσουμε για την εκτέλεση του προγράμματος, τόσο πιο γρήγορα θα γίνει η εκπαίδευση του ταξινομητή. Δε θα πρέπει όμως να δηλώσουμε περισσότερη μνήμη από όση είναι πραγματικά διαθέσιμη. Σε περίπτωση που δε δηλώσουμε την ποσότητα μνήμης που θέλουμε να χρησιμοποιήσει το πρόγραμμα, αυτό καταλαμβάνει 200MB (τιμή χαμηλή για μια συνήθη εκπαίδευση).

`[-sym (προεπιλογή)] [-nonsym]`

Εδώ ορίζουμε αν θέλουμε να γίνει η εκπαίδευση θεωρώντας ότι τα παραδείγματα είναι συμμετρικά ως προς τον κατακόρυφο άξονα, ή όχι. Αν θεωρήσουμε ότι τα παραδείγματα είναι συμμετρικά, τότε η εκπαίδευση γίνεται στο πρώτο μισό των παραδειγμάτων, υπολογίζοντας για κάθε χαρακτηριστικό και το συμμετρικό του. Με την επιλογή αυτή επιταχύνεται η διαδικασία της εκπαίδευσης, αλλά το αντικείμενο που θέλουμε να ανιχνεύσουμε θα πρέπει να είναι απόλυτα συμμετρικό για να πετύχουμε καλά αποτελέσματα.

`[-minhitrate <ελάχιστος_λόγος_εντοπισμού = 0.99500>]`

Με την παράμετρο αυτή δηλώνουμε τον ελάχιστο λόγο εντοπισμού που θέλουμε να έχει κάθε στάδιο του ΔΣΤ. Για να έχουμε καλή ταξινόμηση, η τιμή αυτή θα πρέπει να είναι πολύ κοντά στη μονάδα. Η προεπιλεγμένη τιμή είναι 0,995.

`[-maxfalsealarm <μέγιστος_λόγος_λανθασμένων_ανιχνεύσεων = 0.500000>]`

Με αυτή την παράμετρο δηλώνουμε το μέγιστο λόγο λανθασμένων θετικών ανιχνεύσεων που θέλουμε να έχει κάθε στάδιο του ΔΣΤ. Με τιμή κοντά στο 0,5 (αλλά πάντα κάτω από αυτό), κατασκευάζεται ένας γρήγορος ταξινομητής που σε κάθε στάδιο απορρίπτει περίπου τα μισά αρνητικά παράθυρα. Αν επιλέξουμε τιμή πολύ χαμηλότερη (στην περιοχή του 0,05), ο ταξινομητής θα απορρίπτει πολύ περισσότερα αρνητικά παράθυρα σε κάθε στάδιο. Έτσι, για τον ίδιο αριθμό σταδίων με το προηγούμενο παράδειγμα, θα έχει πολύ χαμηλότερο συνολικό λόγο λανθασμένων ανιχνεύσεων. Ωστόσο, ο ταξινομητής θα είναι πιο αργός και η εκπαίδευσή του πιο χρονοβόρα. Η προεπιλεγμένη τιμή είναι 0,5.

`[-weighttrimming <μικρορύθμιση_βαρών = 0.950000>]`

Η παράμετρος αυτή δηλώνει το συντελεστή ρύθμισης βαρών  $wf$ . Κατά την εκπαίδευση του ταξινομητή με τη χρήση του αλγορίθμου AdaBoost, τα παραδείγματα που χρησιμοποιούνται έχουν μια τιμή βάρους η οποία μεταβάλλεται σε κάθε κύκλο εκπαίδευσης. Αν ένα παράδειγμα ταξινομήθηκε σωστά σε έναν κύκλο εκπαίδευσης, τότε το βάρος του μειώνεται, ώστε ο AdaBoost να θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα. Το βάρος των παραδειγμάτων αυτών μειώνεται πολλαπλασιάζοντάς το με τον αριθμό  $1 - wf$ , άρα, όσο μεγαλύτερη είναι η τιμή της παραμέτρου `weighttrimming`, τόσο πιο πολύ μειώνεται η τιμή του βάρους των παραδειγμάτων που ταξινομήθηκαν σωστά.

`[-eqw]`

Με την παράμετρο αυτή δηλώνουμε ότι θέλουμε τα βάρη για τα αρνητικά και τα θετικά παραδείγματα να αρχικοποιούνται όλα με την ίδια τιμή. Αν δεν επιλέξουμε την παράμετρο αυτή, τα αρχικά βάρη των θετικών παραδειγμάτων είναι διαφορετικά από τα βάρη των αρνητικών.

```
[-mode <BASIC (default) | CORE | ALL>]
```

Με την παράμετρο αυτή επιλέγουμε ποια χαρακτηριστικά θέλουμε να χρησιμοποιηθούν για την εκπαίδευση του ταξινομητή. Αν δώσουμε την τιμή BASIC, που είναι και η προεπιλεγμένη, χρησιμοποιούνται μόνο τα χαρακτηριστικά (i)-(v) που φαίνονται στο Σχήμα 2.1. Αν δώσουμε την τιμή CORE, χρησιμοποιούνται τα προηγούμενα και επιπλέον τα (vi)-(viii). Τέλος αν δώσουμε την τιμή ALL, χρησιμοποιούνται όλα τα διαθέσιμα χαρακτηριστικά τύπου Haar, δηλαδή όλα τα προηγούμενα προσθέτοντας και όλα τα περιστραμμένα κατά  $45^0$  χαρακτηριστικά.

```
[-w <πλάτος_παραδείγματος = 24>]  
[-h <ύψος_παραδείγματος = 24>]
```

Δηλώνουμε το πλάτος και το ύψος των θετικών και αρνητικών παραδειγμάτων που θα χρησιμοποιηθούν, αντίστοιχα. Η προεπιλεγμένη τιμή είναι 24 εικονοστοιχεία για το ύψος και για το πλάτος.

```
[-bt <DAB | RAB | LB | GAB (προεπιλογή)>]
```

Εδώ επιλέγουμε ποια παραλλαγή του αλγορίθμου AdaBoost θέλουμε να εφαρμοστεί [FrHT00]. Οι επιλογές είναι: α) DAB = Discrete AdaBoost (Διακριτός AdaBoost), β) RAB = Real AdaBoost (Πραγματικός AdaBoost), γ) LB = LogitBoost, δ) GAB = Gentle AdaBoost. Η προεπιλογή (Gentle AdaBoost) χρησιμοποιείται συνήθως, καθώς δίνει καλύτερα αποτελέσματα [LiKP03].

```
[-err <misclass (default) | gini | entropy>]
```

Αν έχει επιλεγεί ως αλγόριθμος κατασκευής του ταξινομητή ο Διακριτός AdaBoost (DAB), τότε με αυτή την παράμετρο επιλέγουμε την συνάρτηση που θα χρησιμοποιηθεί για τον υπολογισμό των σφαλμάτων κατά την κατασκευή του δέντρου του κάθε ταξινομητή [FrHT00]. Αν δώσουμε την τιμή misclass, επιλέγεται το κριτήριο των ελάχιστων λανθασμένων ταξινομήσεων, που είναι και η προεπιλεγμένη τιμή. Αν δώσουμε την τιμή gini, επιλέγεται το κριτήριο Gini, ενώ αν δώσουμε την τιμή entropy, το κριτήριο της ελάχιστης εντροπίας.

```
[-maxtreesplits <μέγιστος_αριθμός_διαχωρισμών_στο_δέντρο_του_ΔΣΤ = 0>]
```

Με την παράμετρο αυτή καθορίζουμε τη δομή του δέντρου του ΔΣΤ. Αν δώσουμε την τιμή 0, που είναι και η προεπιλεγμένη, κατασκευάζεται ένα δέντρο εκφυλισμένης μορφής, όπως παρουσιάστηκε και στη θεωρία (βλέπε Σχήμα 3.3α). Αν δώσουμε τιμή μεγαλύτερη του 0, τότε ο ΔΣΤ θα είναι ένα Δέντρο Ταξινόμησης και Παρεμβολής (ΔΤΠ) (CART – Classification and Regression Tree) (βλέπε Σχήμα 3.3β). Θα έχει, δηλαδή, τη μορφή ενός κανονικού δέντρου απόφασης, όπου κάθε κόμβος θα διαχωρίζεται τόσες φορές, όσες δηλώνουμε με την τιμή αυτή.

```
[-minpos <ελάχιστος_αριθμός_θετικών_παραδειγμάτων_ανά_συστάδα = 500>]
```

Αν έχουμε επιλέξει ο ΔΣΤ να έχει τη μορφή δέντρου (ΔΤΠ), τότε με την παράμετρο αυτή ορίζουμε τον ελάχιστο αριθμό θετικών παραδειγμάτων που θα πρέπει να περνάνε από κάθε κόμβο. Αν αυτό το κριτήριο δεν ικανοποιείται για κάποιον κόμβο, τότε αυτός δεν κατασκευάζεται.

### A.3 Πρόγραμμα αξιολόγησης της απόδοσης του ανιχνευτή

Το πρόγραμμα `performance`, που περιλαμβάνεται στη βιβλιοθήκη OpenCV, εφαρμόζει έναν ανιχνευτή σε ένα σύνολο εικόνων δοκιμής και κατατάσσει τα αποτελέσματα ως σωστά ή λανθασμένα. Για την κατάταξη αυτή, χρησιμοποιεί τη μέθοδο που προτείνεται στο [LiMa02] και περιγράφηκε στην ενότητα 3.4. Το πρόγραμμα δέχεται διάφορες παραμέτρους εισόδου, οι οποίες δίνουν πληροφορίες στο πρόγραμμα για τη σωστή εκτέλεσή του και καθορίζουν κάποιες παραμέτρους ανίχνευσης. Οι παράμετροι που εμφανίζονται μέσα σε αγκύλες είναι προαιρετικές. Σε περίπτωση που δε δοθεί μια προαιρετική παράμετρος, αυτή λαμβάνει μια προεπιλεγμένη τιμή.

```
-data <όνομα_φακέλου_ταξινομητή>
```

Με αυτή την παράμετρο δηλώνουμε τον φάκελο στον οποίο βρίσκεται ο ταξινομητής που δημιουργήσαμε κατά την εκπαίδευση.

```
-info <όνομα_αρχείου_χαρακτηρισμένων_εικόνων_δοκιμής>
```

Εδώ δηλώνουμε το όνομα του αρχείου κειμένου που περιλαμβάνει το χαρακτηρισμό των εικόνων δοκιμής. Για τον έλεγχο θα χρησιμοποιηθούν όλες οι εικόνες που βρίσκονται σε αυτό το αρχείο. Αν μια εικόνα δοκιμής δεν περιλαμβάνει κανένα αντικείμενο του τύπου που θέλουμε να ανιχνεύσουμε και το όνομά της δε βρίσκεται στο αρχείο με το χαρακτηρισμό των εικόνων, τότε η εικόνα αυτή δε θα ελεγχθεί.

```
[-maxSizeDiff <μέγιστη_διαφορά_μεγέθους = 1.500000>]
```

```
[-maxPosDiff < μέγιστη_διαφορά_θέσης = 0.300000>]
```

Οι τιμές των δύο αυτών παραμέτρων καθορίζουν ποιες ανιχνεύσεις θεωρούνται σωστές και ποιες όχι. Μειώνοντας τις τιμές των παραμέτρων αυτών, το κριτήριο για τον προσδιορισμό ενός ανιχνευμένου παραλληλογράμμου ως σωστό ή όχι, γίνεται πιο αυστηρό. Αντίθετα, αυξάνοντας τις τιμές αυτές, το κριτήριο γίνεται πιο χαλαρό. Η προεπιλεγμένη τιμή για τη μέγιστη διαφορά μεγέθους είναι 1,5 ενώ για τη μέγιστη διαφορά θέσης είναι 0,3.

```
[-sf <παράγοντας_κλίμακας = 1.200000>]
```

Η παράμετρος αυτή ορίζει τον συντελεστή με τον οποίο θα πολλαπλασιάζονται οι διαστάσεις του παραθύρου ανίχνευσης. Έτσι, το παράθυρο ανίχνευσης ξεκινά από το πραγματικό μέγεθος του ανιχνευτή και αφού ελέγξει όλη την εικόνα, το μέγεθός του πολλαπλασιάζεται με τον παράγοντα *sf*. Αφού ελέγξει ξανά όλη την εικόνα, το μέγεθος του παραθύρου ανίχνευσης, αυξάνεται και πάλι

κατά  $sf$  κ.ο.κ., μέχρι το παράθυρο ανίχνευσης να μην χωρά στην εικόνα. Αυξάνοντας τον παράγοντα κλίμακας  $sf$ , εξετάζουμε λιγότερα υποπαράθυρα της εικόνας. Αυτό έχει ως αποτέλεσμα να μειώνεται η ακρίβεια του ανιχνευτή μας. Ωστόσο, η ταχύτητα της διαδικασίας ανίχνευσης αυξάνεται σημαντικά, ακόμα και με μικρή αύξηση του παράγοντα κλίμακας. Η προεπιλεγμένη τιμή είναι 1,2.

[`-ni`]

Με την παράμετρο αυτή ορίζουμε ότι δε θέλουμε να αποθηκευθούν τα αποτελέσματα της ανίχνευσης. Αν δεν δώσουμε την παράμετρο αυτή, για κάθε εικόνα που εξετάζεται, θα αποθηκεύεται ένα αντίγραφο της εικόνας, στο οποίο όμως θα απεικονίζονται και τα παραλληλόγραμμα που επέστρεψε ο ανιχνευτής. Έτσι, μπορούμε να ελέγξουμε και οπτικά τα αποτελέσματα της ανίχνευσης.

[`-nos <αριθμός_σταδίων = -1>`]

Η παράμετρος αυτή ορίζει τον αριθμό των σταδίων του ταξινομητή που θέλουμε να χρησιμοποιηθούν κατά την ανίχνευση. Αν δώσουμε την τιμή -1, που είναι και η προεπιλογή, χρησιμοποιούνται όλα τα στάδια του ταξινομητή.

[`-rs <μέγεθος_καμπύλης_ROC = 40>`]

Μεταβάλλοντας τον αριθμό των ελάχιστων γειτόνων των υποπαράθυρων που θα επιστρέφονται από τον ανιχνευτή, μπορούμε να μεταβάλλουμε τα αποτελέσματα της ανίχνευσης. Αυξάνοντάς τον, μειώνονται τα υποπαράθυρα που επιστρέφει ο ανιχνευτής. Έτσι, μπορούμε να κατασκευάσουμε μια καμπύλη χαρακτηριστικής λειτουργίας δέκτη (Receiver Operation Characteristic – ROC) μεταβάλλοντας την τιμή των ελάχιστων γειτόνων από μια χαμηλή τιμή (συνήθως 1) έως μια αρκετά μεγαλύτερη.

Με αυτή την παράμετρο καθορίζουμε τη μέγιστη τιμή που θα πάρει η μεταβλητή του συστήματος για τους ελάχιστους γείτονες κάθε υποπαράθυρου. Η ελάχιστη τιμή είναι το 1, χωρίς να μπορούμε να την αλλάξουμε. Έτσι, μας επιστρέφονται τα αποτελέσματα των ανιχνεύσεων για όλες τις τιμές ελάχιστων γειτόνων από 1 έως `μέγεθος_καμπύλης_ROC`, δηλαδή `μέγεθος_καμπύλης_ROC-1` σημεία για την καμπύλη ROC. Ουσιαστικά, δηλαδή, με αυτή την παράμετρο ορίζουμε το πλήθος των δεδομένων που θέλουμε να μας επιστρέψει το πρόγραμμα για την κατασκευή της καμπύλης ROC. Η προεπιλεγμένη τιμή είναι 40, που είναι αρκετά ικανοποιητική.

[`-w <πλάτος_παραθύρου_ανίχνευσης = 24>`]

[`-h <ύψος_παραθύρου_ανίχνευσης = 24>`]

Με τις παραμέτρους αυτές δηλώνουμε το μέγεθος του αρχικού παραθύρου ανίχνευσης, το οποίο θα πρέπει να ταυτίζεται με το μέγεθος του ταξινομητή. Το μέγεθος του παραθύρου ανίχνευσης θα πολλαπλασιάζεται με το συντελεστή κλίμακας που δηλώσαμε σε προηγούμενη παράμετρο, ώστε να ανιχνευθούν όλα τα πιθανά μεγέθη αντικειμένων.

## A.4 Αλλαγές στο πρόγραμμα αξιολόγησης της απόδοσης του ανιχνευτή

Είδαμε παραπάνω το πρόγραμμα της OpenCV για τον έλεγχο της επίδοσης του ανιχνευτή. Επιλέξαμε να αλλάξουμε τον τρόπο που προσδιορίζεται αν μια ανίχνευση είναι πραγματική ή λανθασμένη. Έτσι, αντί της μεθόδου που χρησιμοποιείται από το πρόγραμμα της OpenCV, επιλέξαμε να χρησιμοποιήσουμε τη μέθοδο που προτείνεται στο [EZW+06] και περιγράφηκε στην ενότητα 3.4. Επίσης, πραγματοποιήσαμε κάποιες αλλαγές στις λειτουργίες του, ώστε να βελτιωθεί η χρηστικότητά του.

```
-data <αρχείο_ταξινομητή_μορφής_XML>
```

Δηλώνουμε, όπως και πριν, το όνομα και τη θέση που βρίσκεται ο ταξινομητής. Εδώ όμως, ο ταξινομητής θα πρέπει να βρίσκεται σε μορφή αρχείου XML. Το αρχείο XML του ταξινομητή δημιουργείται στο τέλος της διαδικασίας εκπαίδευσης από το πρόγραμμα haartraining της OpenCV. Επιλέξαμε τη χρησιμοποίηση του αρχείου XML αντί της αρχικής μορφής φακέλου, καθώς το αρχείο XML μπορεί να μεταφερθεί και να αποθηκευθεί πολύ πιο εύκολα. Έτσι, η μέτρηση της επίδοσης ενός ανιχνευτή μπορεί να γίνει ανά πάσα στιγμή, αποθηκεύοντας τον ταξινομητή στην πολύ πιο ευέλικτη μορφή του αρχείου XML.

```
[-minOverlap <κατώφλι_ανίχνευσης = 0.500000>]
```

Με αυτή την παράμετρο, δηλώνουμε το κατώφλι που θα χρησιμοποιηθεί για την κατάταξη ενός αποτελέσματος του ταξινομητή, ως σωστό ή λανθασμένο, όπως περιγράφηκε στην ενότητα 3.4. Η προεπιλεγμένη τιμή είναι το 0,5, που πειραματικά βρέθηκε ότι δίνει σωστά αποτελέσματα.

```
[-csv <αρχείο_δεδομένων_εξόδου>]
```

Μια ακόμα λειτουργία που προστέθηκε, είναι αυτή της αυτόματης αποθήκευσης των αποτελεσμάτων σε αρχείο κειμένου. Με την παράμετρο αυτή δηλώνουμε το όνομα του αρχείου κειμένου στο οποίο θέλουμε να αποθηκευθούν τα αποτελέσματα της διαδικασίας μέτρησης της επίδοσης του ανιχνευτή. Είδαμε προηγουμένως, ότι για να κατασκευάσουμε μια καμπύλη ROC, αρκεί να μεταβάλλουμε την παράμετρο των ελάχιστων γειτόνων. Έτσι, στο αρχείο εξόδου θα αποθηκεύουμε μια γραμμή με δεδομένα για κάθε τιμή ελάχιστων γειτόνων. Κάθε γραμμή του αρχείου εξόδου θα περιλαμβάνει τα εξής δεδομένα: α) τον παράγοντα κλιμάκωσης του παραθύρου ανίχνευσης, β) το κατώφλι ανίχνευσης, γ) τον αριθμό των σταδίων του ταξινομητή που χρησιμοποιείται, δ) τον αριθμό των ελάχιστων γειτόνων, ε) τον αριθμό των ανιχνευμένων υποπαραθύρων που βρέθηκαν σε όλες τις εικόνες και κατατάχθηκαν ως σωστά, στ) τον αριθμό όλων των εμφανίσεων του αντικειμένου στις εικόνες, και ζ) τον αριθμό των ανιχνευμένων υποπαραθύρων που κατατάχθηκαν ως λανθασμένα. Οι τιμές διαχωρίζονται μεταξύ τους με τον χαρακτήρα κόμμα ",". Οι πρώτες τρεις τιμές κάθε γραμμής δηλώνουν τις παραμέτρους που χρησιμοποιήθηκαν κατά τις ανιχνεύσεις, και είναι σταθερές για κάθε εκτέλεση του προγράμματος. Τοποθετούνται όμως, γιατί μπορούμε να τρέχουμε το πρόγραμμα με διαφορετικές παραμέτρους κάθε φορά, και να αποθηκεύουμε τα αποτελέσματα στο ίδιο αρχείο εξόδου. Το αρχείο εξόδου δε σβήνεται κάθε φορά που τρέχουμε το πρόγραμμα, αλλά συνεχίζεται το γράψιμο σε αυτό από το

τέλος του. Έτσι, μπορούμε να έχουμε όλα τα αποτελέσματα για έναν ταξινομητή σε ένα αρχείο εξόδου, και να κατασκευάσουμε εύκολα τις καμπύλες ROC που επιθυμούμε.

```
[ -neighbors <ελάχιστος_αριθμός_γειτόνων = 1>]
```

Η παράμετρος αυτή δηλώνει τον ελάχιστο αριθμό γειτόνων που θα πρέπει να έχουν τα υποπαράθυρα που θα επιστραφούν από τον ανιχνευτή. Τα δεδομένα που θα πάρουμε στο τέλος για την κατασκευή της καμπύλης ROC, θα είναι για αριθμό ελάχιστων γειτόνων από την παράμετρο που δώσαμε, ως την τιμή της παραμέτρου μέγεθος\_καμπύλης\_ROC. Η προεπιλεγμένη τιμή είναι 1. Αν δώσουμε την τιμή 0, θα μας επιστραφούν από τον ανιχνευτή όλα τα υποπαράθυρα που πέρασαν από τον ΔΣΤ, χωρίς να γίνει καμία επεξεργασία σε αυτά. Ο αριθμός των υποπαράθυρων αυτών θα είναι πολύ μεγαλύτερος από ό,τι με την προεπιλεγμένη τιμή, και θα περιέχει και αρκετές λανθασμένες ανιχνεύσεις. Όσο μεγαλύτερη τιμή δώσουμε σε αυτή την παράμετρο, τόσο λιγότερα θα είναι τα υποπαράθυρα που θα μας επιστρέψει ο ανιχνευτής, και η καμπύλη ROC θα είναι περιορισμένη.

Στο τροποποιημένο πρόγραμμα δεν χρησιμοποιούνται οι παράμετροι maxSizeDiff και maxPosDiff, καθώς έχουν αντικατασταθεί από την παράμετρο minOverlap. Επίσης οι παράμετροι w και h δεν χρειάζονται, καθώς οι διαστάσεις του ταξινομητή είναι αποθηκευμένες στο αρχείο XML. Οι παράμετροι info, sf, ni, nos και rs χρησιμοποιούνται ακριβώς όπως περιγράφηκε προηγουμένως.





## Βιβλιογραφία

- [AgAR04] Shivani Agarwal, Aatif Awan, and Dan Roth, Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11):1475-1490, 2004.  
<http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/>
- [AgRo02] S Agarwal, D Roth, Learning a sparse representation for object detection, Proc. ECCV, 2002
- [Burg98] CJC Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998
- [DaTr05] N Dalal, B Triggs, Histograms of oriented gradients for human detection, Proc. CVPR, 2005
- [DuHa72] RO Duda, PE Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM, 1972
- [DuHS00] RO Duda, PE Hart, DG Stork, Pattern Classification, Wiley-Interscience, 2nd edition, 2000
- [EZW+06] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang, The 2005 PASCAL Visual Object Classes Challenge, In Selected Proceedings of the First PASCAL Challenges Workshop, LNAI, Springer-Verlag, 2006
- [FePZ03] R Fergus, P Perona, A Zisserman, Object class recognition by unsupervised scale-invariant learning, Computer Vision and Pattern Recognition, 2003
- [FLCS05] Mario Fritz, Bastian Leibe, Barbara Caputo, Bernt Schiele, Integrating Representative and Discriminative Models for Object Category Detection, iccv, pp. 1363-1370, Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 2, 2005
- [FrHT00] J Friedman, T Hastie, R Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Statist, 2000
- [FrSc95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, In Computational Learning Theory: Eurocolt '95, pages 23–37. Springer-Verlag, 1995
- [GaAH02] A Garg, S Agarwal, T Huang, Fusion of global and local information for object detection, ICPR'02, 2002

- [LeLS04] B Leibe, A Leonardis, B Schiele, Combined object categorization and segmentation with an implicit shape model, Workshop on Statistical Learning in Computer Vision, ECCV, 2004
- [LeLS08] B Leibe, A Leonardis, B Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision, 2008
- [LeWe04] K Levi, Y Weiss, Learning object detection from a small number of examples: The importance of good features, Proc. CVPR, 2004
- [LiKP03] R Lienhart, A Kuranov, V Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, DAGM 25th Pattern Recognition Symposium, 2003
- [LiMa02] R Lienhart, J Maydt, An extended set of Haar-like features for rapid object detection, Image Processing, 2002
- [Lowe04] DG Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 2004
- [MuLo06] J Mutch, DG Lowe, Multiclass object recognition with sparse localized features, IEEE CVPR, 2006
- [OpenCV] Open Source Computer Vision Library  
<http://opencvlibrary.sourceforge.net/>
- [OPS+97] M Oren, C Papageorgiou, P Sinha, E Osuna, T Poggio, Pedestrian detection using wavelet templates, Proc. Computer Vision and Pattern Recognition, 1997
- [PaOP98] CP Papageorgiou, M Oren, T Poggio, A general framework for object detection, Computer Vision, 1998
- [RTMF05] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation. MIT AI Lab Memo AIM-2005-025, September, 2005.  
<http://labelme.csail.mit.edu/>
- [ScKa00] H. Schneiderman and T. Kanade, A Statistical Model for 3D Object Detection Applied to Faces and Cars. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, June 2000.  
<http://vasc.ri.cmu.edu/idb/html/car/index.html>
- [SFBL98] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Stat., 26(5):1651–1686, 1998
- [ShBC05] J Shotton, A Blake, R Cipolla, Contour-based learning for object detection, Proc. ICCV, 2005

- [ViJo01] P Viola, M Jones, Rapid object detection using a boosted cascade of simple features, Proc. CVPR, 2001
- [ViJo02] P Viola, M Jones, Robust real-time object detection, International Journal of Computer Vision, 2002
- [ViJo04] P Viola, M Jones, Robust real-time face detection, International Journal of Computer Vision, 2004
- [WeWP00] M. Weber, M. Welling and P. Perona, 'Towards Automatic Discovery of Object Categories. Proc. IEEE Comp. Soc. Conf. Comp. Vision and Pattern Recog. (CVPR), 2000 June.  
<http://www.vision.caltech.edu/archive.html>