ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ ΒΙΝΤΕΟ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

Ταχεία ανίχνευση αντικειμένων σε εικόνες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γιώργου Π. Μήτση

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2015.

Εθνικο Μετσοβιο Πολυτεχνειο
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ ΒΙΝΤΕΟ
ΚΑΙ ΠΟΛΥΜΕΣΩΝ

Ταχεία ανίχνευση αντικειμένων σε εικόνες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γιώργου Π. Μήτση

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14/7/2015.

.....	.....	.....
Στέφανος Κόλλιας Καθηγητής Ε.Μ.Π.	Σταφυλοπάτης Ανδρέας-Γεώργιος Καθηγητής Ε.Μ.Π.	Στάμου Γεώργιος Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2015.

.....
Γιώργος Π. Μήτσης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Στέφανο Κόλλια για την δυνατότητα που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα, καθώς και τον ερευνητή κ. Ιωάννη Αβρίθη για την πολύτιμη καθοδήγηση που μου παρείχε κατά την εκπόνηση της διπλωματικής μου εργασίας και για την συνεργασία μας. Ευχαριστώ επίσης τα παιδιά του εργαστηρίου Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων (IVA) που μου παραχώρησαν τον υπολογιστή που χρησιμοποίησα κατά την διάρκεια της διπλωματικής και ήταν πρόθυμοι να βοηθήσουν όποτε χρειάστηκε.

Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου που με στήριξαν με κάθε τρόπο καθ'όλη την διάρκεια των σπουδών μου, καθώς και τις αδελφές μου οι οποίες αν και εξ' αποστάσεως μου πρόσφεραν την απαραίτητη ψυχολογική υποστήριξη.

Τέλος, ένα μεγάλο ευχαριστώ στους φίλους μου, οι οποίοι όλα αυτά τα χρόνια προσπάθησαν με κάθε δυνατό τρόπο να με αποσπάσουν από το διάβασμα και τις υποχρεώσεις, και χάρη στους οποίους τα φοιτητικά μου χρόνια θα μου μείνουν αξέχαστα.

Περίληψη

Η ανίχνευση υποψήφιων θέσεων αντικειμένων είναι ένα σχετικά πρόσφατο πρόβλημα που προέκυψε λόγω της πολυπλοκότητας των αλγορίθμων ανίχνευσης αντικειμένων και του μεγάλου χρόνου εκτέλεσής τους. Σκοπός είναι με ταχείς υπολογισμούς να ανιχνεύονται όλα τα αντικείμενα στην εικόνα ανεξάρτητα από την κλάση στην οποία ανήκουν. Οι ανιχνεύσεις αυτές τροφοδοτούνται στους ανιχνευτές αντικειμένων έτσι ώστε οι τελευταίοι να αποφύγουν την εξαντλητική αναζήτηση με την μέθοδο κινούμενου παραθύρου. Με αυτόν τον τρόπο μειώνεται ο χρόνος που χρειάζονται για να ταυτοποιήσουν μια εικόνα ενώ ταυτόχρονα μπορούν να χρησιμοποιήσουν πιο πολύπλοκους και αποτελεσματικούς αλγορίθμους. Όλοι οι σύγχρονοι ανιχνευτές αντικειμένων χρησιμοποιούν τις υποψήφιες θέσεις αντικειμένων..

Στην διπλωματική μας παρουσιάζουμε όλες τις σύγχρονες μεθόδους για την παραγωγή των υποψήφιων θέσεων αντικειμένων και προτείνουμε μια νέα μέθοδο, την Segment Boxes. Στην μέθοδο αυτή χρησιμοποιούμε κατάτμηση της εικόνας και με βάση τα τμήματα που προκύπτουν βαθμολογούμε παράθυρα μέσα στην εικόνα ανάλογα με την πιθανότητα να υπάρχουν σε αυτά αντικείμενα. Προσπαθούμε να ενσωματώσουμε καλές ιδέες άλλων μεθόδων καθώς και δικές μας για την επίτευξη βέλτιστου αποτελέσματος, κάτι που έχει σαν αποτέλεσμα να καταλήξουμε σε διάφορες προσεγγίσεις της μεθόδου μας.

Συγκρίνουμε τις διάφορες προσεγγίσεις μας και τις καλύτερες τις συγκρίνουμε με τις σύγχρονες μεθόδους με την χρήση κατάλληλων μετρικών πάνω σε εικόνες από τις βάσεις εικόνων PASCAL VOC07 και ImageNet2013. Στην συνέχεια ενσωματώνουμε την μέθοδό μας σε έναν σύγχρονο ανιχνευτή αντικειμένων που χρησιμοποιεί βαθιά μάθηση (deep learning) και συνελικτικά νευρωνικά δίκτυα, τον Fast R-CNN, και συγκρίνουμε και πάλι τα αποτελέσματά μας με αυτά των άλλων μεθόδων, στο πραγματικό πλέον πρόβλημα της ανίχνευσης αντικειμένων.

Στόχος μας ήταν να εξετάσουμε τις δυνατότητες της κατάτμησης για το πρόβλημα της ανίχνευσης υποψήφιων θέσεων αντικειμένων. Τα αποτελέσματα της μεθόδου μας είναι ανταγωνίσιμα και σε μερικές περιπτώσεις ξεπερνούν τα αποτελέσματα των σύγχρονων μεθόδων, επιτυγχάνοντας μικρό χρόνο εκτέλεσης (μέχρι και 0.3 δευτερόλεπτα ανά εικόνα).

Keywords: όραση υπολογιστών, ανίχνευση αντικειμένων, ανίχνευση υποψήφιων θέσεων αντικειμένων, κατάτμηση, Segment Boxes

Abstract

Object proposals is a relatively new problem which appeared due to the complexity of modern object detectors and their high execution time. The purpose of object proposal algorithms is the high speed class-agnostic detection of all objects in the image. The proposals are then passed to the object detectors so that they avoid the exhaustive search of the image using the sliding window approach. This way, the time needed to detect objects is drastically reduced which enables them to use more complex and effective algorithms. Modern object detectors use object proposals.

In our thesis we present most modern methods for the extraction of object proposals and we propose a new method, Segment Boxes. This method uses segmentation of the image and by using the resulting segments we score windows inside the image based on the possibility that they contain objects. We try to encapsulate good ideas of other methods as well as some of our own to achieve best results, so we end up with several approaches of our method.

We compare those different approaches and the best ones are compared with the state-of-the-art methods, using the appropriate metrics, on images from datasets PASCAL VOC07 and ImageNet2013. We then use our proposals with a modern object detector which uses deep learning and convolutional neural networks, Fast R-CNN, and we compare again our results with those of other methods, this time on the problem of object detection.

Our goal was to examine the potential of segmentation on the problem of object proposals. The results of our method are competitive and in some cases exceed those of the state-of-the-art methods, while achieving low execution time (one of our approaches runs on 0.3 seconds per image).

Keywords: computer vision, detection proposals, object detection, object proposals, segmentation, Segment Boxes

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

Η όραση υπολογιστών είναι ένα πεδίο που περιλαμβάνει μεθόδους λήψης, κατεργασίας, ανάλυσης και κατανόησης εικόνων και γενικά δεδομένων πολλών διαστάσεων από τον πραγματικό κόσμο για να παράγει συμβολικές ή αριθμητικές πληροφορίες, πχ με την μορφή απόφασης.

Σαν επιστημονικό πεδίο, η όραση υπολογιστών ασχολείται με την θεωρία πίσω από τα τεχνητά συστήματα που εξάγουν πληροφορίες για εικόνες. Τα δεδομένα της εικόνας μπορεί να έχουν διάφορες μορφές όπως φωτογραφίες, βίντεο, πολυδιάστατες εικόνες από ιατρικό σαρωτή κλπ. Σαν τεχνικός κλάδος, η όραση υπολογιστών εφαρμόζει τις θεωρίες της και τα μοντέλα για την κατασκευή συστημάτων όρασης.

Στις μέρες μας η όραση υπολογιστών βρίσκει τεράστια εφαρμογή σε διάφορους τομείς. Ενδεικτικά μπορούμε να αναφέρουμε τους παρακάτω:

- Έλεγχος διαδικασιών (ένα βιομηχανικό ρομπότ ή ένα αυτόνομο όχημα)
- Ανίχνευση συμβάντων (οπτική επιτήρηση)
- Ανακατασκευή εικόνας (σε περίπτωση απώλειας πληροφορίας πχ λόγω θορύβου)
- Οργάνωση πληροφοριών (ευρετηριοποίηση βάσεων δεδομένων και ακολουθιών εικόνων)
- Εξομοίωση αντικειμένων και περιβαλλόντων (βιομηχανική επιθεώρηση, ιατρική ανάλυση εικόνας ή τοπογραφική εξομοίωση)
- Αλληλεπίδραση χρηστών με υπολογιστικά συστήματα (ως είσοδος σε μια συσκευή επικοινωνίας ανθρώπου/μηχανής, ενισχυμένη πραγματικότητα)
- Ανίχνευση αντικειμένων σε εικόνα (ανίχνευση προσώπου, ανίχνευση πεζών, ταυτοποίηση αντικειμένων)

Στην παρούσα διπλωματική θα ασχοληθούμε με το πρόβλημα της ανίχνευσης αντικειμένων.

1.1 Ανίχνευση αντικειμένων

Η ανίχνευση αντικειμένων είναι μια διαδικασία την οποία ο άνθρωπος καλείται να εκτελεί καθημερινά στην ζωή του. Για τον ανθρώπινο εγκέφαλο αυτή η εργασία δεν απαιτεί ιδιαίτερη προσπάθεια καθώς είναι εκπαιδευμένος να αναγνωρίζει αντικείμενα ακόμα και αν δεν έχει ξαναδεί ποτέ στο παρελθόν τα ίδια, αρκεί να έχει δει κάποιο άλλο της ίδιας κατηγορίας. Είναι πολύ εύκολο να αναγνωρίσει ένα αντικείμενο σαν αυτοκίνητο στον δρόμο ακόμα και αν είναι η πρώτη φορά που συναντάει το συγκεκριμένο μοντέλο. Η εργασία όμως αυτή δεν είναι το ίδιο εύκολη ακόμα και για έναν εξελιγμένο ηλεκτρονικό υπολογιστή.

Στην γενικότερη προσπάθεια αυτοματοποίησης κάποιων διαδικασιών, στις μέρες μας προσπαθούμε να επιτύχουμε τέτοιου είδους ανιχνεύσεις σε εικόνες, κινούμενες ή στατικές, με την χρήση ηλεκτρονικού υπολογιστή. Η διαδικασία αυτή έχει αρκετές εφαρμογές σε πραγματικά προβλήματα όπως το πρόβλημα της οδήγησης αυτοκινήτων χωρίς οδηγό, τα οποία θα πρέπει να έχουν επίγνωση του χώρου γύρω τους και να αναγνωρίζουν τυχόν εμπόδια, ανθρώπους, άλλα αυτοκίνητα ή προβλήματα στο οδόστρωμα, ή στην ανάλυση ιατρικής εικόνας, όπου ένα μηχάνημα θα μπορεί να κάνει διάγνωση χωρίς να υπάρχει ανάγκη επίβλεψης της διαδικασίας από τον άνθρωπο. Με αυτόν τον τρόπο και αφού ο υπολογιστής έχει πρόσβαση σε μεγαλύτερο όγκο πληροφοριών, θα μειωθούν τα λάθη και ο χρόνος που χρειάζεται για την λήψη αποφάσεων.

Στην επιστήμη των υπολογιστών μπορούμε να ορίσουμε την ανίχνευση αντικειμένων ως το πρόβλημα κατά το οποίο με είσοδο μια εικόνα και δεδομένης της γνώσης για τον τρόπο αναπαράστασης των αντικειμένων παίρνουμε σαν έξοδο το *ποιά* αντικείμενα εμφανίζονται στην εικόνα και *πού* (σχήμα 1.2). Το πρόβλημα αυτό μπορούμε να το χωρίσουμε σε δύο ομάδες, την ανίχνευση στιγμιότυπου αντικειμένου και την ανίχνευση κλάσης.

Η ανίχνευση στιγμιότυπου αντικειμένου είναι η διαδικασία κατά την οποία ελέγχεται αν ένα αντικείμενο είναι το ίδιο με ένα άλλο αντικείμενο που είδαμε προηγουμένως. Αυτού του είδους η ανίχνευση, εξαρτάται κυρίως από το χρώμα και από πληροφορίες αποκλειστικές για το αντικείμενο που εξετάζουμε.

Η ανίχνευση κλάσης, είναι η διαδικασία κατά την οποία αντικείμενα που δεν έχουμε ξαναδεί στο παρελθόν, τοποθετούνται στην ίδια ομάδα με άλλα αντικείμενα που έχουμε δει. Η ανίχνευση κλάσης βασίζεται στην γενίκευση ιδιοτήτων ή λειτουργιών των αντικειμένων που γνωρίζουμε, τα οποία συγκρίνονται με τις αντίστοιχες ιδιότητες του καινούργιου, προς αναγνώριση αντικειμένου. Επειδή το εκάστοτε αντικείμενο μπορεί να υποστεί αλλαγές όσον αφορά το μέγεθος, την οπτική γωνία, το αν εμφανίζεται ολόκληρο ή τμήμα του στην εικόνα, καθώς και άλλες παραμορφώσεις, η ανίχνευση κλάσης είναι αρκετά δύσκολο πρόβλημα. Στην εργασία μας, όταν μιλάμε για ανίχνευση αντικειμένου, θα αναφερόμαστε μόνο στην ανίχνευση κλάσης του.

Ένας επιπλέον διαχωρισμός που μπορεί να γίνει, σχετίζεται με τον τρόπο που γίνεται η ανίχνευση στην εικόνα (σχήμα 1.1). Έτσι μπορεί να έχουμε ανίχνευση σε επίπεδο εικόνας (σχήμα 1.1α), σε επίπεδο περιγράμματος (σχήμα 1.1β) και σε επίπεδο παραθύρου (σχήμα 1.1γ). Καθ'όλη τη διάρκεια της διπλωματικής μας, θα

(α) εικόνα (β) περίγραμμα (γ) παράθυρο

Σχήμα 1.1: Τρόποι ανίχνευσης αντικειμένων

αναφερόμαστε στην ανίχνευση σε επίπεδο παραθύρου.

Για να πραγματοποιήσουμε την ανίχνευση, αναπαριστούμε την εικόνα σαν ένα ψηφιακό σήμα δύο διαστάσεων όπου κάθε στοιχείο παίρνει σαν τιμή το χρώμα στην συγκεκριμένη θέση της εικόνας. Με αυτόν τον τρόπο έχουμε την δυνατότητα να χρησιμοποιήσουμε εργαλεία της θεωρίας σημάτων για να επεξεργαστούμε την εικόνα και να αναπαραστήσουμε τις πληροφορίες της σε χρήσιμες μορφές. Παραδείγματα από τις συνηθέστερες τεχνικές που χρησιμοποιούνται σε αλγορίθμους ανίχνευσης αντικειμένων είναι α) τα ιστογράμματα HOG, β) ο μετασχηματισμός Hough, γ) ο μετασχηματισμός SIFT, δ) ιστογράμματα ακμών και ε) ιστογράμματα έντασης.

Στην συνέχεια, παίρνοντας τα χαρακτηριστικά της εικόνας με κάποια από τις παραπάνω μεθόδους, ο αλγόριθμος καλείται να λάβει απόφαση για το αν βρίσκεται ή όχι το αντικείμενο στην εικόνα. Αυτός ο έλεγχος ονομάζεται ταξινόμηση. Για αυτό το κομμάτι της ανίχνευσης αντικειμένων χρησιμοποιούνται άλλες τεχνικές όπως τα νευρωνικά δίκτυα, τα δάση απόφασης και τα SVM (Support Vector Machine - Μηχανές Διανυσματικής Υποστήριξης).

Η σύγχρονη τάση στο συγκεκριμένο πρόβλημα έχει οδηγήσει στην χρήση πιο πολύπλοκων, πολυστρωματικών νευρωνικών δικτύων, την λεγόμενη βαθιά μάθηση (deep learning). Στόχος της συγκεκριμένης μεθόδου είναι να αντικατασταθούν τα χαρακτηριστικά που αναφέραμε με άλλα τα οποία έχουν προέλθει από μια μη επιβλεπόμενη ή ημιεπιβλεπόμενη διαδικασία μάθησης. Πολλές αρχιτεκτονικές της μεθόδου έχουν ήδη δώσει βέλτιστα αποτελέσματα (πχ Scalable object detection using deep convolutional Networks [12] , R-CNN [17], Fast R-CNN [16])

1.2 Υποψήφιες θέσεις αντικειμένων

Μέχρι πριν λίγα χρόνια, οι πιο επιτυχημένες μέθοδοι για ανίχνευση αντικειμένων χρησιμοποιούσαν την τεχνική κυλιόμενου παραθύρου [27, 36, 14], με την οποία αποδοτικοί ταξινομητές ελέγχουν για παρουσία αντικειμένου σε κάθε παράθυρο της εικόνας. Με αυτόν τον τρόπο ελέγχονται 10^4 με 10^5 παράθυρα. Με την αύξηση των εικονοστοιχείων της εικόνας έχουμε τάξεις μεγέθους μεγαλύτερο αριθμό παραθύρων, αφού τα παράθυρα επιλέγονται σε διάφορα μεγέθη, ενώ οι σύγχρονες βάσεις δεδομένων απαιτούν και ανίχνευση της διάστασης του αντικειμένου, κάτι που αυξάνει ακόμα

Σχήμα 1.2: Παραδείγματα ανίχνευσης αντικειμένων

περισσότερο τον χώρο αναζήτησης σε 10^6 με 10^7 παράθυρα.

Η αύξηση της πολυπλοκότητας των αλγορίθμων για την ανίχνευση αντικειμένων οδήγησε σε αύξηση της ποιότητας ανίχνευσης αλλά ταυτόχρονα αυξήθηκε σημαντικά και ο χρόνος που απαιτείται σε κάθε υποψήφιο παράθυρο [37, 17, 33, 18, 8]. Ένας τρόπος αντιμετώπισης του προβλήματος έτσι ώστε να διατηρηθεί ο χρόνος σε λογικά επίπεδα και να παραμείνει υψηλή η ποιότητα της ανίχνευσης είναι η χρήση των υποψήφιων θέσεων αντικειμένων [1, 6, 11, 35]. Θεωρώντας ότι όλα τα αντικείμενα μοιράζονται παρόμοια χαρακτηριστικά για να ξεχωρίζουν από το περιβάλλον τους, σχεδιάστηκαν αλγόριθμοι οι οποίοι παίρνοντας μια εικόνα σαν είσοδο, δίνουν σαν έξοδο ένα σύνολο από θέσεις της εικόνας στις οποίες υπάρχει αυξημένη πιθανότητα να υπάρχει αντικείμενο (σχήμα 1.3). Στόχος των μεθόδων αυτών είναι να επιστρέψουν όσο το δυνατόν περισσότερα αντικείμενα που περιέχει η εικόνα, σε όσο το δυνατών μικρότερο αριθμό παραθύρων. Έτσι λοιπόν, πιο εξειδικευμένοι και πολύπλοκοι αλγόριθμοι τρέχουν σε πολύ μικρότερο χρόνο αφού τρέχουν σε πολύ μικρότερο αριθμό παραθύρων.

Η συγκεκριμένη μεθοδολογία προτάθηκε αρκετά πρόσφατα, μόλις το 2011, από τους B. Alexe, T. Deselaers και V. Ferrari [2]. Στο άρθρο τους ορίζουν τα αντικείμενα σαν πράγματα της εικόνας που ξεχωρίζουν με ένα καλά ορισμένο όριο και κέντρο, όπως αγελάδες, αυτοκίνητα και τηλέφωνα, σε αντίθεση με το άμορφο φόντο όπως ο ουρανός, το γρασίδι και ο δρόμος. Στη συνέχεια, προτείνουν ένα μέτρο για το κατα πόσο ένα σημείο της εικόνας είναι αντικείμενο ή όχι, το οποίο ονομάζουν αντικειμενικότητα (*objectness*). Ένα αντικείμενο πρέπει να έχει τουλάχιστον ένα από τα παρακάτω τρία χαρακτηριστικά:

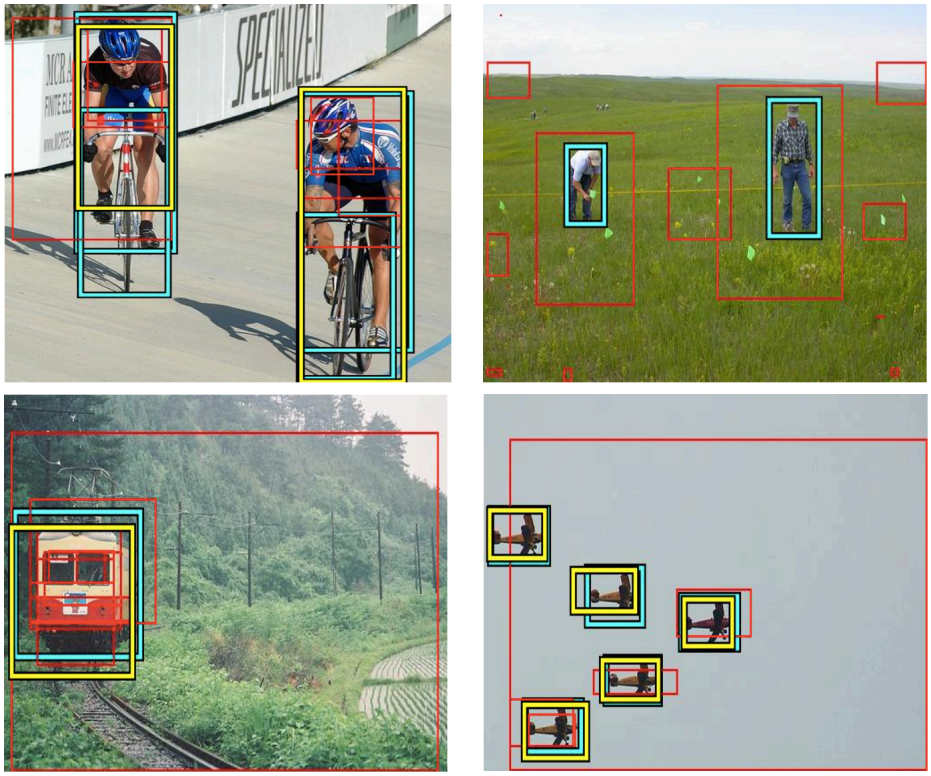
- καλά ορισμένο κλειστό όριο
- διαφορετική εμφάνιση από το περιβάλλον του
- να είναι μοναδικό ή να ξεχωρίζει μέσα στην εικόνα

Πολλά αντικείμενα έχουν περισσότερα από ένα από τα παραπάνω χαρακτηριστικά.

Αξίζει να σημειωθεί ότι οι ανιχνευτές αντικειμένων εξειδικεύονται σε μια κλάση αντικειμένων όπως αυτοκίνητα ή άνθρωποι, ενώ οι υποψήφιες θέσεις αντικειμένων έχουν γενικότερη δράση και εντοπίζουν όλα τα είδη αντικειμένων. Αυτό έχει σαν αποτέλεσμα την εύκολη γενίκευση της εφαρμογής της μεθόδου ακόμα και σε αντικείμενα που δεν έχουν ανιχνευθεί ξανά στο παρελθόν.

Στην λογική τους, οι υποψήφιες θέσεις αντικειμένων θυμίζουν τους ανιχνευτές σημείων ενδιαφέροντος. Όπως αυτοί εφιστούν την προσοχή των αντίστοιχων αλγορίθμων στα πιο σημαντικά και ξεχωριστά σημεία της εικόνας, έτσι και με τις υποψήφιες θέσεις, οι ανιχνευτές αντικειμένων τρέχουν μόνο σε σημεία που υπάρχει πιθανότητα να περικλείονται αντικείμενα.

Οι σύγχρονοι βέλτιστοι ανιχνευτές αντικειμένων για τις βάσεις δεδομένων PASCAL [13] και ImageNet [32], χρησιμοποιούν όλοι υποψήφιες θέσεις αντικειμένων. Εκτός του ότι επιτρέπουν την χρήση πιο προηγμένων ταξινομητών λόγω βελτίωσης του χρόνου εκτέλεσης, η χρήση των υποψήφιων θέσεων αντικειμένων αλλάζει και



Σχήμα 1.3: Παραδείγματα υποψήφιων θέσεων αντικειμένων

τα δεδομένα τα οποία επεξεργάζεται ο ταξινομητής. Αυτό μπορεί να βελτιώσει και την ποιότητα των αποτελεσμάτων μειώνοντας τις εσφαλμένες ανιχνεύσεις (false positives).
Από την στιγμή που το πρότειναν οι Alexe, Deselaers και Ferrari μέχρι σήμερα, έχουν προταθεί πολλές μέθοδοι για τον υπολογισμό των υποψήφιων θέσεων αντικειμένων. Η βασική μετρική για την ποιότητα των αλγορίθμων αυτών είναι η ανάκληση (recall). Ως ανάκληση ορίζουμε το ποσοστό των πραγματικών θέσεων αντικειμένων της εικόνας που επιστρέφονται ως προτεινόμενα από την εκάστοτε μέθοδο.

Στις μέρες μας, έχουν αναπτυχθεί πολλοί καλοί αλγόριθμοι, ο καθένας με επιτυχία σε διαφορετικούς τομείς. Υπάρχουν αλγόριθμοι που έχουν επιτύχει πολύ καλή ανάκληση (98% Selective Search [34]) αλλά επιστρέφουν μεγάλο αριθμό υποψήφιων θέσεων, άλλοι πολύ γρήγοροι (0.2/ Bing [7]) αλλά με χαμηλή ανάκληση και άλλοι με πολύ καλά αποτελέσματα, λίγες υποψήφιες θέσεις αλλά πολύ αργοί (CPMC [6]).

Ωστόσο, το γεγονός ότι δεν υπάρχει ακόμα ένας αλγόριθμος που να τα κάνει όλα σημαίνει ότι υπάρχει αρκετό πρόσφορο έδαφος στο συγκεκριμένο χώρο και ακριβώς εκεί πάει να πατήσει η συγκεκριμένη διπλωματική.

1.3 Συνεισφορά της διπλωματικής

Σε αυτή τη διπλωματική μελετήθηκαν διάφορες μέθοδοι αυτού του καινούργιου σχετικά τομέα και συγκρίνοντάς τες προσπαθήσαμε να δούμε που βασίζεται η επιτυχία της κάθε μεθόδου. Οι περισσότερες μέθοδοι κάνουν χρήση της κατάτμησης της εικόνας για την παραγωγή των αποτελεσμάτων τους. Σαν στόχο της διπλωματικής μας είχαμε να ελέγξουμε κατά πόσο η κατάτμηση έχει εξαντλήσει τις δυνατότητές της σαν χαρακτηριστικό γνώρισμα των αντικειμένων.

Στην προσπάθειά μας αυτή, αναπτύξαμε μια νέα μέθοδο, την Segment Boxes, η οποία χρησιμοποιεί αποκλειστικά και μόνο κατάτμηση της εικόνας για την βαθμολόγηση των υποψήφιων θέσεων. Συνδυάσαμε κάποιες ιδέες από άλλες μεθόδους, τόσο για να επιτύχουμε καλύτερη ανάκληση, όσο και για να έχουμε καλύτερη ταχύτητα, καθώς με έξυπνες δομές δεδομένων πετυχαίνουμε μικρό χρόνο εκτέλεσης, πράγμα πολύ σημαντικό για την ανίχνευση υποψήφιων θέσεων. Παράλληλα, μελετήσαμε και αρκετές ακόμα ιδέες πέρα από την βασική, με σκοπό την βελτίωση της απόδοσης, πάντα όμως μόνο με την χρήση τμημάτων (κεφάλαιο 3.2).

Στην συνέχεια συγκρίναμε την μέθοδό μας με τις σύγχρονες μεθόδους για να έχουμε πλήρη εικόνα της απόδοσής της πάνω στις πιο διαδεδομένες βάσεις δεδομένων για το αντικείμενο καθώς όλες οι καινούργιες εργασίες καλούνται να συγκριθούν σε αυτές. Όπως θα δούμε και από τα αποτελέσματα η μέθοδος που δημιουργήσαμε είναι απλή και γρήγορη, ενώ πετυχαίνει αρκετά καλή ποιότητα ανιχνεύσεων που σε ορισμένες περιπτώσεις ξεπερνά τις ήδη υπάρχουσες μεθόδους.

Τέλος δοκιμάσαμε τις υποψήφιες θέσεις της μεθόδου μας στον ανιχνευτή αντικειμένων Fast R-CNN έτσι ώστε να δούμε την απόδοσή της στο πραγματικό πρόβλημα με το οποίο ασχολούμαστε που είναι η ανίχνευση αντικειμένων σε εικόνα.

1.4 Οργάνωση κειμένου

Η διπλωματική χωρίζεται σε έξι κεφάλαια. Το πρώτο κεφάλαιο είναι η παρούσα εισαγωγή. Στο δεύτερο κεφάλαιο παρουσιάζουμε τις διάφορες μεθόδους που έχουν υλοποιηθεί πάνω στο πρόβλημα ανίχνευσης υποψήφιων θέσεων αντικειμένων, μερικές από τις οποίες παρείχαν και ιδέες για την υλοποίηση της δικής μας μεθόδου. Παρουσιάζουμε επίσης την μέθοδο κατάτμησης του Felzenswalb [15] η οποία μας παρέχει τα χαρακτηριστικά που χρησιμοποιούμε καθώς και της ανίχνευσης ακμών [10] η οποία χρησιμοποιήθηκε σε μια από τις εναλλακτικές υλοποιήσεις. Στο τρίτο κεφάλαιο γίνεται παρουσίαση της δικής μας μεθόδου καθώς και οι παραλλαγές που κάναμε στην προσπάθειά μας να επιτύχουμε το βέλτιστο δυνατό αποτέλεσμα. Στο τέταρτο κεφάλαιο έχουμε την παρουσίαση των αποτελεσμάτων της μεθόδου μας συγκρίνοντας τις διάφορες παραλλαγές που υλοποιήσαμε και την σύγκρισή της βέλτιστης με τις υπόλοιπες σύγχρονες μεθόδους. Ακολουθεί στο πέμπτο κεφάλαιο η παρουσίαση του ανιχνευτή αντικειμένων Fast R-CNN και η σύγκριση με βάση τα αποτελέσματά του για κάθε μέθοδο. Τέλος κλείνουμε στο έκτο κεφάλαιο με τα συμπεράσματα και την μελλοντική έρευνα πάνω στον τομέα.

Κεφάλαιο 2

Σχετικές εργασίες

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις μεθόδους που έχουν προταθεί για την ανίχνευση υποψήφιων θέσεων αντικειμένων σε εικόνα.

Υπάρχουν δύο βασικές προσεγγίσεις για τον υπολογισμό των υποψήφιων θέσεων: *μέθοδοι ομαδοποίησης* και *μέθοδοι βαθμολόγησης παραθύρου*. Αντιπροσωπευτικές μέθοδοι για τις δύο αυτές ομάδες είναι η Selective Search [34] και Objectness [2] αντίστοιχα.

2.1 Μέθοδοι ομαδοποίησης

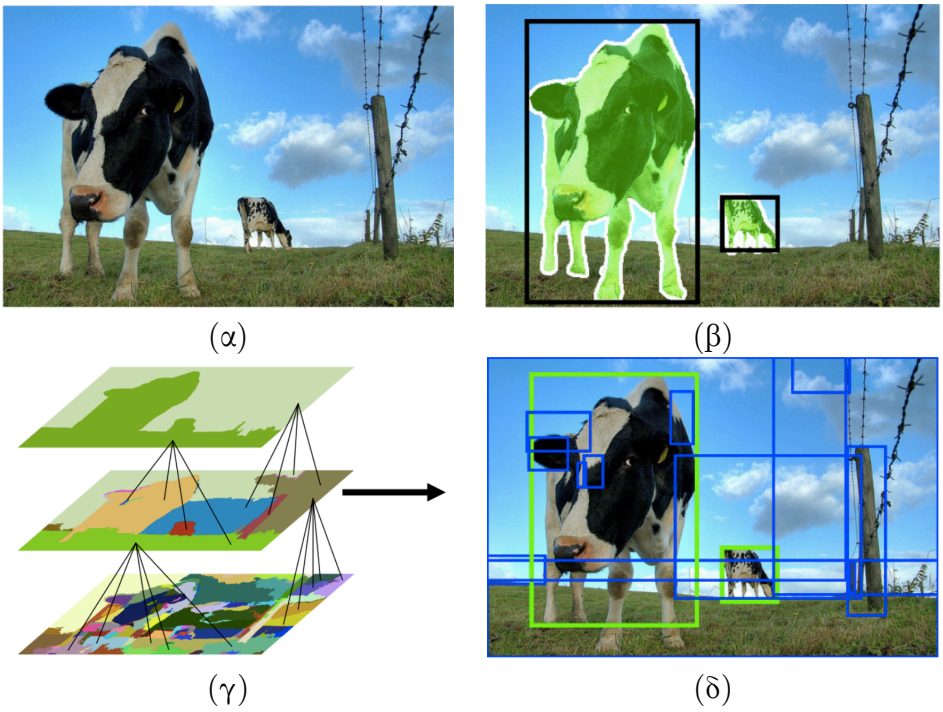
Οι μέθοδοι ομαδοποίησης δημιουργούν πολλαπλά (πιθανώς και επικαλυπτόμενα) τμήματα της εικόνας που πιθανώς να αντιστοιχούν σε αντικείμενα. Στη συνέχεια καλούνται να λάβουν απόφαση για το αν θα συγχωνεύσουν τα τμήματα μεταξύ τους με βάση ένα μεγάλο σύνολο υποδείξεων όπως το σχήμα των τμημάτων, στοιχεία εμφάνισης (χρώμα, υφή κλπ.) και ακμές (συνήθως παίρνονται από [3] και [10]).

Για να αυξηθεί ο αριθμός των τμημάτων συχνά πραγματοποιείται υπερκατάτμηση της εικόνας συνήθως με τον αλγόριθμο του Felzenszwalb [15].

2.1.1 Selective Search

Στην μέθοδο *Selective Search* [34] δημιουργούνται τμήματα της εικόνας που ενώνονται ιεραρχικά σε μεγαλύτερα τμήματα. Ξεκινάει με μια υπερκατάτμηση [15] που παράγει τα αρχικά τμήματα. Στην συνέχεια χρησιμοποιείται ένας άπληστος αλγόριθμος που ενώνει περιοχές μεταξύ τους: Πρώτα υπολογίζεται η ομοιότητα των περιοχών μεταξύ τους. Οι δύο πιο παρόμοιες περιοχές ενώνονται και υπολογίζεται η ομοιότητα της νέα περιοχής με τις γειτονικές της. Η διαδικασία επαναλαμβάνεται μέχρι ολόκληρη η εικόνα να γίνει μία περιοχή.

Για την ομοιότητα $s(r_i, r_j)$ μεταξύ των περιοχών r_i και r_j χρησιμοποιούνται πολλά συμπληρωματικά χαρακτηριστικά με την προϋπόθεση ότι υπολογίζονται γρήγορα.



Σχήμα 2.1: Δεδομένης μιας εικόνας (α) σκοπός είναι να βρεθούν τα αντικείμενα (β). Για να επιτευχθεί αυτό χρησιμοποιείται κατάτμηση: δημιουργούνται περιοχές για όλα τα μεγέθη της εικόνας και λαμβάνονται όσο γίνεται περισσότερες διαφορετικές μορφές της εικόνας με χρήση διαφορετικών χρωματικών χώρων (γ). Παράδειγμα υποψηφίων παραθύρων έχουμε στο (δ)

Αυτό σημαίνει ότι όταν ενώνονται δύο περιοχές r_i και r_j, η τιμές των χαρακτηριστικών της περιοχής που προκύπτει θα πρέπει να υπολογίζεται από τις τιμές των r_i και r_j χωρίς να χρειάζεται να προσπελαύνονται ξανά τα εικονοστοιχεία.

Τα χαρακτηριστικά που χρησιμοποιούνται είναι το s_{colour} που μετράει την ομοιότητα χρώματος, το $s_{texture}$ που μετράει την ομοιότητα υφής, το s_{size} το οποίο βοηθάει να ενώνονται πρώτα μικρότερα τμήματα και το s_{fill} για να ενώνονται πρώτα τμήματα που περιέχονται σε άλλα μεγαλύτερα.

Παράλληλα, για την μεγαλύτερη ποικιλομορφία στα τμήματα που παράγονται και στον τρόπο που ενώνονται, κάτι που είναι και το βασικό στοιχείο της μεθόδου, χρησιμοποιούνται πολλοί διαφορετικοί χρωματικοί χώροι και διαφορετικοί αλγόριθμοι για την αρχική κατάτμηση.

Τέλος οι υποψήφιες θέσεις από κάθε στρατηγική ομαδοποιούνται για να δώσουν την τελική πλέον έξοδο (σχήμα 2.1).

Η μέθοδος δεν έχει παραμέτρους που να απαιτούν εκπαίδευση, ανταυτού, τα χαρακτηριστικά και οι συναρτήσεις ομοιότητας είναι σχεδιασμένες με το χέρι. Η μέθοδος αυτή είναι η επιλογή πολλών από τους πιο σύγχρονους ανιχνευτές αντικειμένων,

(α) Κατάτμηση (β) Αρχικοποίηση

(γ) k-οστή επανάληψη (δ) Υποψήφιο παράθυρο

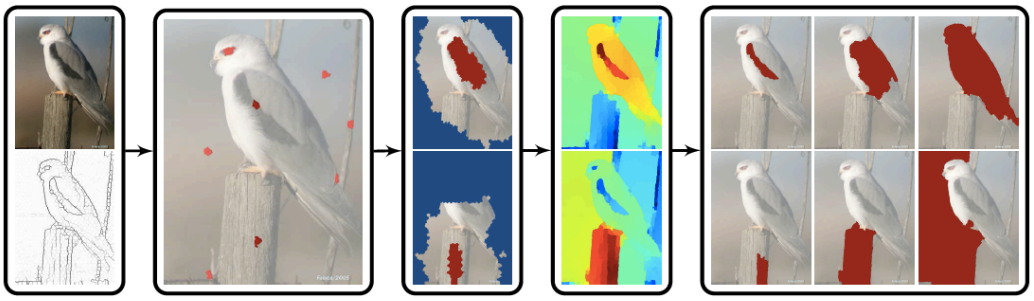
Σχήμα 2.2: Εφαρμόζεται ο αλγόριθμος Randomized Prim's πάνω στον γράφο που δημιουργείται από την κατάτμηση (α'). (β') Ξενικάει με ένα τμήμα (πράσινο). Σε κάθε επανάληψη (γ'), παίρνει ένα γειτονικό τμήμα (κόκκινο) και αποφασίζει αν θα το προσθέσει ή αν θα επιτρέψει το παράθυρο σαν υποψήφια θέση (δ'). Η φωτεινότητα του κόκκινου δείχνει την σχετική πιθανότητα να επιλεγεί το συγκεκριμένο γειτονικό τμήμα (όσο πιο ανοιχτό, τόσο πιο πιθανό). Τα μπλε τμήματα δεν συνδέονται στο τρέχον δέντρο, άρα δεν μπορούν να επιλεγούν.

συμπεριλαμβανομένου και του R-CNN ανιχνευτή [17].

2.1.2 Randomized Prim's

Η *Randomized Prim's* [26] χρησιμοποιεί παρόμοια χαρακτηριστικά με την *Selective Search* (ομοιότητα χρώματος, αναλογία κοινού συνόρου και μέγεθος) αλλά εισάγει τυχαιότητα στον τρόπο που ενώνει τα τμήματα. Δημιουργεί έναν γράφο $G = (V, E, \varrho)$ όπου V τα τμήματα της εικόνας από υπερκατάτμηση [15], E οι ακμές μεταξύ τους και ϱ το βάρος των ακμών που ορίζει την πιθανότητα δύο τμήματα να ανήκουν στο ίδιο αντικείμενο. Ο αλγόριθμος παράγει ανεξάρτητα τυχαία δέντρα επικάλυψης με βάση το βάρος τους τα οποία ωστόσο δεν είναι τα ελάχιστα καθώς εισάγεται τυχαιότητα στον τρόπο που ενώνονται οι κόμβοι. Η επέκταση του κάθε δέντρου επικάλυψης τελειώνει με ένα κριτήριο τερματισμού στο οποίο εισάγεται επίσης τυχαιότητα. Ο αλγόριθμος επαναλαμβάνεται για όσα δέντρα θέλει ο χρήστης.

Επιπλέον, σε αντίθεση με την *Selective Search*, η μέθοδος *Randomized Prim's*



Σχήμα 2.3: Οι σπόροι τοποθετούνται με βάση μια διαδικασία μάθησης, δημιουργούνται οι μάσκες αντικειμένων και φόντου με βάση τους σπόρους, υπολογίζεται ο γεωδαιτικός μετασχηματισμός απόστασης για τις διάφορες μάσκες και παράγονται οι υποψήφιες θέσεις με βάση τις κρίσιμες τιμές για κάθε GDT.

εφαρμόζει μια διαδικασία μάθησης για να υπολογιστούν τα βάρη των ακμών και η αναλογία με την οποία συμμετέχει το κάθε χαρακτηριστικό στην παραγωγή της υποψήφιας θέσης.

Η συγκεκριμένη μέθοδος αυξάνει την ποικιλομορφία των αποτελεσμάτων και βελτιώνει την ταχύτητα μειώνοντας ελαφρώς την ποιότητα.

Στο σχήμα 2.2 μπορούμε να δούμε καλύτερα την ροή εργασίας του αλγορίθμου.

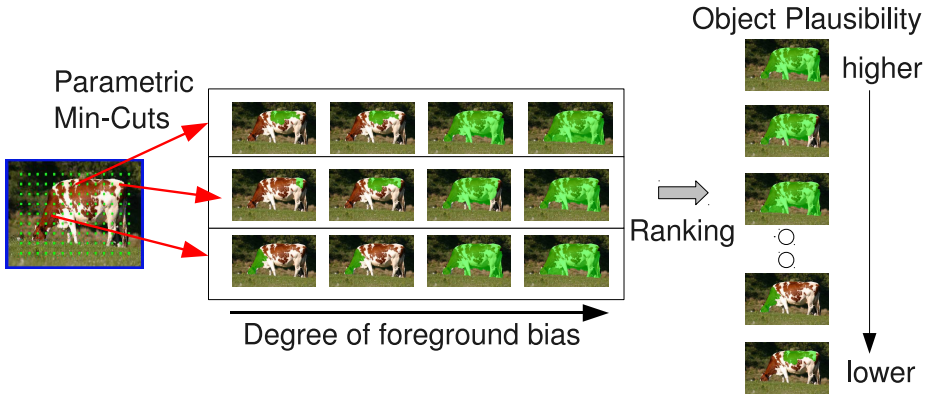
2.1.3 Geodesic Object Proposals

Η μέθοδος αυτή (*Geodesic* [23]) χρησιμοποιεί γεωδαιτικό μετασχηματισμό απόστασης (Geodesic Distance Transform - GDT) που υπολογίζεται από προσεκτικά τοποθετημένους σπόρους (σχήμα 2.3).

Η γεωδαιτική απόσταση $d_{x,y}$ μεταξύ δύο κόμβων $x, y \in V_i$ ορίζεται ως το μήκος του συντομότερου μονοπατιού ανάμεσα στου κόμβους. Ο GDT μετράει την γεωδαιτική απόσταση από ένα σύνολο κόμβων $Y \subset V$ σε κάθε κόμβο $x \in V$,

$$D(x; Y) = \min_{y \in Y} d_{x,y}. \tag{2.1}$$

Πιο συγκεκριμένα, παράγεται μια υπερκατάτμηση, αυτή τη φορά χρησιμοποιώντας τον αλγόριθμο της εργασίας του Dollár [10]. Η πιθανότητα ύπαρξης ακμών για κάθε εικονοστοιχείο που προκύπτει επίσης από τον ίδιο αλγόριθμο χρησιμοποιείται για την υπολογισμό της ομοιότητας μεταξύ των τμημάτων της εικόνας. Στη συνέχεια ορίζεται ένα σύνολο από σπόρους που τοποθετούνται στην εικόνα μετά από μια διαδικασία μάθησης και για κάθε έναν δημιουργείται μια μάσκα που ξεχωρίζει το προσκήνιο από το φόντο που θα χρησιμοποιηθεί για τον υπολογισμό του GDT. Για κάθε μια μάσκα υπολογίζεται ο GDT σε όλη την εικόνα και κάθε επίπεδο του GDT αποτελεί μια περιοχή. Ωστόσο δεν είναι όλες οι περιοχές καλές οπότε εξάγονται μερικές ποιοτικές υποψήφιες θέσεις με βάση κάποιες τιμές κατωφλίου. Τέλος ταξινομούνται όλα τα αποτελέσματα που προτείνονται από κάθε σπόρο και μάσκα για να απορριφθούν παρόμοιες ανιχνεύσεις.



Σχήμα 2.4: Τμήματα εξάγονται γύρω από τους σπόρους του προσκηνίου για διάφορα κατώφλια, με αποτέλεσμα την δημιουργία τμημάτων σε διαφορετικά μεγέθη. Τα τμήματα που προκύπτουν φιλτράρονται και βαθμολογούνται με βάση κάποια χαρακτηριστικά για να προκύψουν οι υποψήφιες θέσεις.

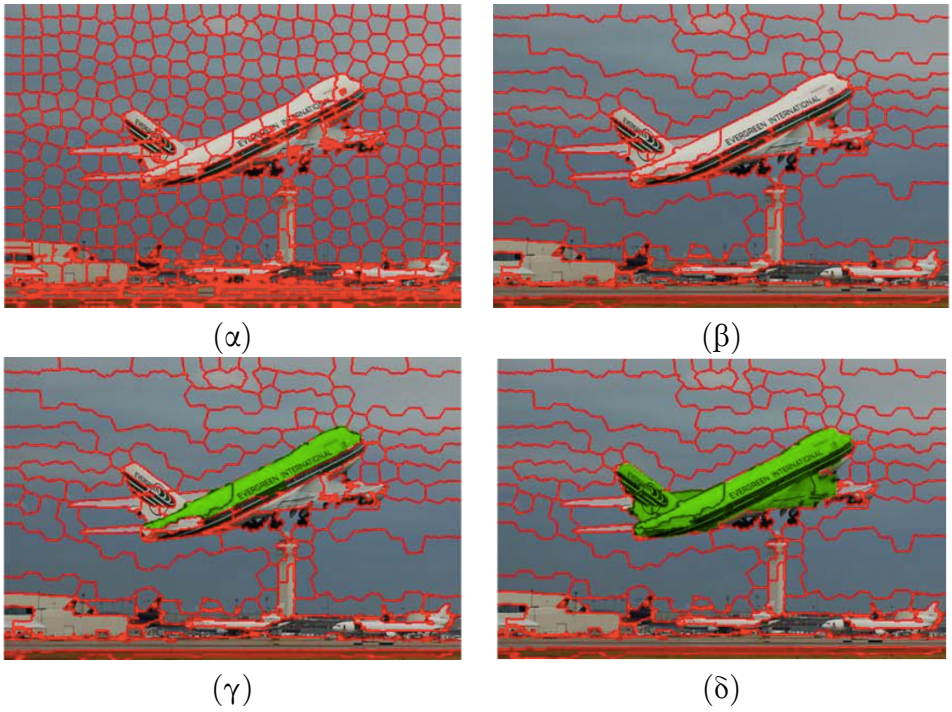
2.1.4 Constrained Parametric Min-Cuts

Η μέθοδος *Constrained Parametric Min-Cuts (CPMC)* [6] αποφεύγει την αρχική κατάτμηση της εικόνας και κάνει κατάτμηση γράφου με την χρήση πολλών σπόρων κατευθείαν πάνω στα εικονοστοιχεία. Τα τμήματα που προκύπτουν βαθμολογούνται με βάση ένα μεγάλο αριθμό χαρακτηριστικών. Η ροή εργασίας φαίνεται στο σχήμα 2.4.

Για κάθε εικόνα, διαφορετικά σύνολα από αρχικά εικονοστοιχεία θεωρούνται ότι ανήκουν στο προσκήνιο και ορίζονται σαν σπόροι προσκηνίου. Στην συνέχεια, σε κάθε εικονοστοιχείο της εικόνας τίθεται μια τιμή ανάλογα με την πιθανότητα να ανήκει και αυτό στο προσκήνιο. Αυτό γίνεται για όλα τα εικονοστοιχεία εκτός από αυτά στα όρια της εικόνας, που ορίζονται σαν σπόροι φόντου. Οι τελευταίοι χρησιμοποιούνται για να μην υπάρχει περίπτωση το σύνολο των εικονοστοιχείων φόντου να είναι κενό. Υπολογίζεται έτσι μια κατάτμηση μεταξύ προσκηνίου και φόντου με την χρήση διάφορων χαρακτηριστικών και κατωφλιών για να προκύψει μεγάλη ποικιλία μεγεθών τμημάτων.

Πιο συγκεκριμένα, για την επιλογή των σπόρων προσκηνίου χρησιμοποιείται ένα πλέγμα 5x5 ενώ για την επιλογή των σπόρων φόντου χρησιμοποιούνται 4 διαφορετικές προσεγγίσεις: μια που περιλαμβάνει όλο το όριο της εικόνας, μια μόνο τις κάθετες ακμές, μια μόνο τις οριζόντιες και μία όλες εκτός από το κάτω όριο. Αυτό γίνεται για την ανίχνευση αντικειμένων που βγαίνουν εκτός του ορίου της εικόνας.

Ακολουθεί φιλτράρισμα και βαθμολογία των τμημάτων που προκύπτουν με την χρήση 34 διαφορετικών χαρακτηριστικών σχετικά με γράφους, περιοχές και ιδιότητες Gestalt.



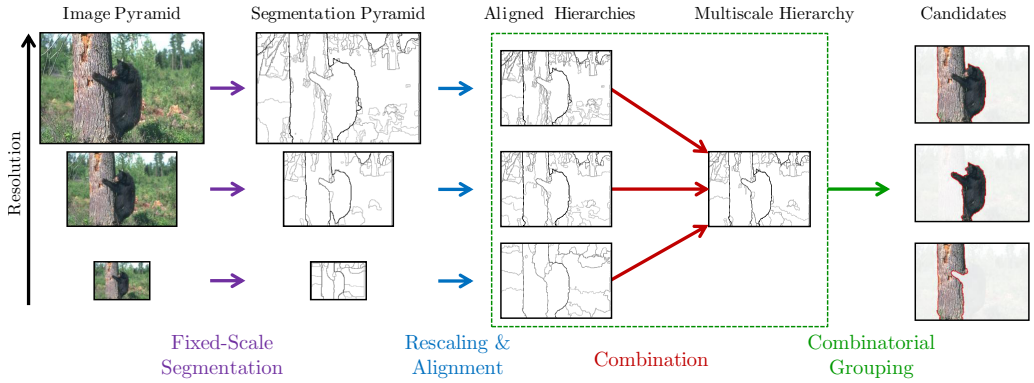
Σχήμα 2.5: Οι διάφορες φάσεις της μεθόδου Rantalankila. Από την αρχική υπερκατάτμηση της εικόνας (α), δημιουργείται μια προτεινόμενη κατάτμηση σε μεγαλύτερα τμήματα (β) τα οποία συγχωνεύονται με την τοπική προσέγγιση (γ). Στο (δ) παρουσιάζεται μια υποψήφια θέση που προκύπτει από την ολική προσέγγιση.

2.1.5 Μέθοδος του Rantalankila

Στην μέθοδο *Rantalankila* [29] η εικόνα υφίσταται υπερκατάτμηση σε μικρά τμήματα και η υποψήφιες θέσεις προκύπτουν με δύο διαφορετικές προσεγγίσεις, μία τοπική και μία ολική της εικόνας στο ύφος της $CPMC$ μεθόδου (σχήμα 2.5).

Η τοπική προσέγγιση είναι παρόμοια με την $SelectiveSearch$ αλλά χρησιμοποιεί διαφορετικά χαρακτηριστικά. Σε αυτή την προσέγγιση, σε κάθε γειτονικό ζευγάρι τμημάτων δίνεται μια τιμή που αντιπροσωπεύει την οπτική ομοιότητα των τμημάτων. Τα ζευγάρια που μοιάζουν περισσότερο συγχωνεύονται σε ένα τμήμα και τα εικονοστοιχεία τους αποθηκεύονται σαν υποψήφια αντικείμενα. Οι τιμές των ομοιοτήτων των γειτόνων ανανεώνονται για το καινούργιο τμήμα που δημιουργήθηκε. Η συγχώνευση συνεχίζεται μέχρι να μείνει ένα μόνο τμήμα που περιλαμβάνει όλη την εικόνα και τότε συλλέγονται όλα τα υποψήφια αντικείμενα που παρήχθησαν κατά την διαδικασία. Αυτή η τοπική προσέγγιση παράγει προτάσεις σε όλα τα μεγέθη αλλά δεν είναι κατάλληλη για την ανίχνευση αντικειμένων που περιλαμβάνουν κομμάτια που διαφέρουν οπτικά.

Στην ολική προσέγγιση, το πρόβλημα ορίζεται ως διαχωρισμός του αντικειμένου από το φόντο, ελαχιστοποιώντας μια συνάρτηση ενέργειας πάνω σε έναν γράφο με



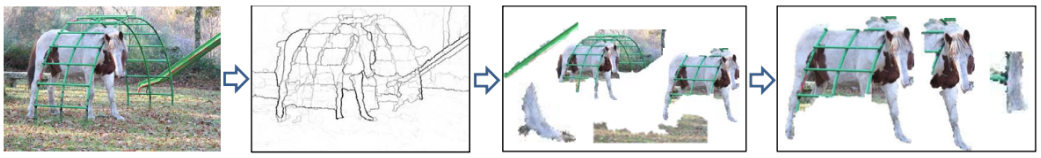
Σχήμα 2.6: Ροή εργασίας για την μέθοδο MCG.

τμήματα. Ο γράφος αυτός έχει για κόμβους τα διάφορα τμήματα και για ακμές τις σχέσεις μεταξύ των τμημάτων. Η συνάρτηση ενέργειας περιλαμβάνει δύο όρους. Ο ένας αντιπροσωπεύει την ομοιότητα μεταξύ των γειτονικών τμημάτων και ο άλλος την υπόθεση αντικειμένου-φόντου. Αλλάζοντας την υπόθεση αντικειμένου-φόντου καθώς και τις παραμέτρους της συνάρτησης ενέργειας, η μέθοδος καταφέρνει να επιστρέψει έναν μεγάλο αριθμό από υποψήφιες θέσεις. Η επιλογή των παραμέτρων γίνεται με εκμάθηση έτσι ώστε να υπάρχει ποικιλία στις υποψήφιες θέσεις που προκύπτουν.

2.1.6 Multiscale Combinatorial Grouping

Στην *Multiscale Combinatorial Grouping (MCG)* [4] προτείνεται ένας σχετικά γρήγορος αλγόριθμος για τον υπολογισμό ιεραρχικής κατάτμησης σε πολλαπλές κλίμακες που βασίζεται στα [3],[10] και [39] από τις οποίες παίρνει και τα βασικά χαρακτηριστικά που χρησιμοποιεί.

Η μέθοδος ξεκινάει από μια πυραμίδα πολλών αναλύσεων της εικόνας και πραγματοποιεί ιεραρχική κατάτμηση σε κάθε ανάλυση ξεχωριστά. Στην συνέχεια, οι πολλές αυτές ιεραρχίες ευθυγραμμίζονται και ενώνονται σε μια πολυεπίπεδη ιεραρχία κατάτμησης. Τα τμήματα της τελευταίας συγχωνεύονται με βάση την ένταση των ακμών μεταξύ τους σε μεγαλύτερα τμήματα και αυτά με την σειρά τους βαθμολογούνται με βάση διάφορα χαρακτηριστικά για να προκύψουν οι τελικές υποψήφιες θέσεις. Τα χαρακτηριστικά που χρησιμοποιούνται για την βαθμολόγηση είναι το μέγεθος και η θέση του υποψήφιου παραθύρου καθώς τα περισσότερα αντικείμενα έχουν συγκεκριμένη θέση και αναλογία στις εικόνες, το σχήμα του αντικειμένου που προτείνεται και η ένταση των ακμών γύρω του. Το σχήμα 2.6 παρουσιάζει ακριβώς αυτή την διαδικασία.



Σχήμα 2.7: Η μέθοδος Endres, υπολογίζει μια ιεραρχική κατάτμηση, παράγει υποψήφιες θέσεις και τις βαθμολογεί.

2.1.7 Μέθοδος του Endres

Οι βασικές συνεισφορές της εργασίας των *Endres και Hoiem* [11] είναι η χρήση χαρακτηριστικών για το όριο και το σχήμα του αντικειμένου σε συνδυασμό με χαμηλού επιπέδου χαρακτηριστικά, για να παράγει μεγάλη ποικιλία από υποψήφια παράθυρα, και η χρήση μιας διαδικασίας βαθμολόγησης με εκμάθηση, που στοχεύει στην ανάκληση όλων των αντικειμένων στην εικόνα.

Αρχικά παράγει μια υπερκατάτμηση της εικόνας με την χρήση του αλγορίθμου του Hoiem [19]. Στη συνέχεια χρησιμοποιεί τα τμήματα σαν σπόρους για να χωρίσει την εικόνα σε περιοχές και παράγει τις υποψήφιες θέσεις. Τα χαρακτηριστικά που χρησιμοποιεί είναι το ιστόγραμμα χρώματος και το ιστόγραμμα υφής, το άθροισμα και τη μέγιστη ένταση του ορίου ανάμεσα το κέντρο μάζας των τμημάτων, και τρία χαρακτηριστικά σχετικά με την διάταξη των τμημάτων. Αυτά ουσιαστικά δείχνουν κατά πόσο υπάρχει συνοχή ανάμεσα στα τμήματα έτσι ώστε να ανήκουν στο ίδιο αντικείμενο. Για παράδειγμα, αν μια περιοχή προβλέπεται ότι είναι στην αριστερή πλευρά ενός αντικειμένου και η περιοχή δεξιά της προβλέπεται ότι είναι στην δεξιά πλευρά του, τότε οι δύο περιοχές έχουν συνοχή. Υπάρχει λοιπόν βαθμολογία για την οριζόντια, την κάθετη και την ολική συνοχή.

Τέλος χρησιμοποιείται ένας αλγόριθμος μάθησης για την βαθμολόγηση των περιοχών και την επιστροφή των περισσότερο πιθανών θέσεων ύπαρξης αντικειμένου στην εικόνα.

2.2 Μέθοδοι βαθμολόγησης παραθύρου

Η εναλλακτική προσέγγιση για την παραγωγή των υποψήφιων θέσεων είναι η βαθμολόγηση των υποψήφιων παραθύρων ανάλογα με την πιθανότητα που έχουν να περιέχουν ένα αντικείμενο. Αντίθετα με τις μεθόδους ομαδοποίησης που μπορούν να επιστρέφουν τα ίδια τα αντικείμενα, εδώ οι αλγόριθμοι επιστρέφουν μόνο το παράθυρο που περιβάλλει το αντικείμενο και γενικά είναι πιο γρήγοροι. Εκτός και αν τα υποψήφια παράθυρα που επιλέγονται είναι πολύ πυκνά, αλγόριθμοι επιστρέφουν υποψήφιες θέσεις με μικρή τοπική ακρίβεια. Για την αντιμετώπιση του φαινομένου, πολλές τεχνικές χρησιμοποιούν μια επιπλέον διαδικασία για την βελτίωση της ακρίβειας.

2.2.1 Objectness

Η *Objectness* [2] είναι μια από τις πρώτες και πιο γνωστές εργασίες για ανίχνευση υποψήφιων θέσεων αντικειμένων. Ουσιαστικά είναι αυτή που πρότεινε την χρήση τους στην ανίχνευση αντικειμένων και εισήγαγε την έννοια του objectness, το οποίο αναφέρεται στην πιθανότητα ύπαρξης αντικειμένου σε μία θέση.

Στην εργασία αυτή, χρησιμοποιούνται πέντε διαφορετικά χαρακτηριστικά για την βαθμολόγηση της ποιότητας της ανίχνευσης. Αρχικά έχουμε την Σημαντικότητα (multiscale saliency - MS) η οποία χρησιμοποιεί τον γρήγορο μετασχηματισμό Fourier (Fast Fourier Transform - FFT) και δίνει μεγαλύτερη βαρύτητα σε περιοχές που έχουν μοναδική εμφάνιση στην εικόνα. Έτσι περιοδικές επαναλήψεις ενός αντικειμένου (πχ γρασίδι ή κήπος από λουλούδια) έχει μικρότερη πιθανότητα να δώσει καλό αποτέλεσμα (σχήμα 2.8α,β) . Στην συνέχεια έχουμε την Αντίθεση Χρώματος (Color Contrast - CC σχήμα) η οποία μας δίνει την χρωματική διαφορά του παραθύρου που επεξεργαζόμαστε με τις άμεσα γειτονικές του περιοχές και χρησιμοποιεί ιστογράμματα LAB για τον υπολογισμό της (2.8ε). Το τρίτο χαρακτηριστικό είναι η πυκνότητα ακμών (Edge Density - ED) που μετράει κατα πόσο υπάρχει ακμή στα όρια του παραθύρου και υπολογίζεται με την βοήθεια του ανιχνευτή Canny (σχήμα 2.8γ,δ) ενώ σαν τέταρτο χρησιμοποιείται η θέση και το μέγεθος του παραθύρου ανεξάρτητα από τα εικονοστοιχεία που περιέχει.

Το πέμπτο, και σύμφωνα με της μετρήσεις τους, μακράν πιο σημαντικό χαρακτηριστικό, είναι η επικάλυψη των τμημάτων (Superpixels Straddling - SS σχήμα). Ουσιαστικά δείχνει τον βαθμό με τον οποίο κάθε τμήμα βγαίνει εκτός του ορίου του παραθύρου που εξετάζουμε (2.8στ):

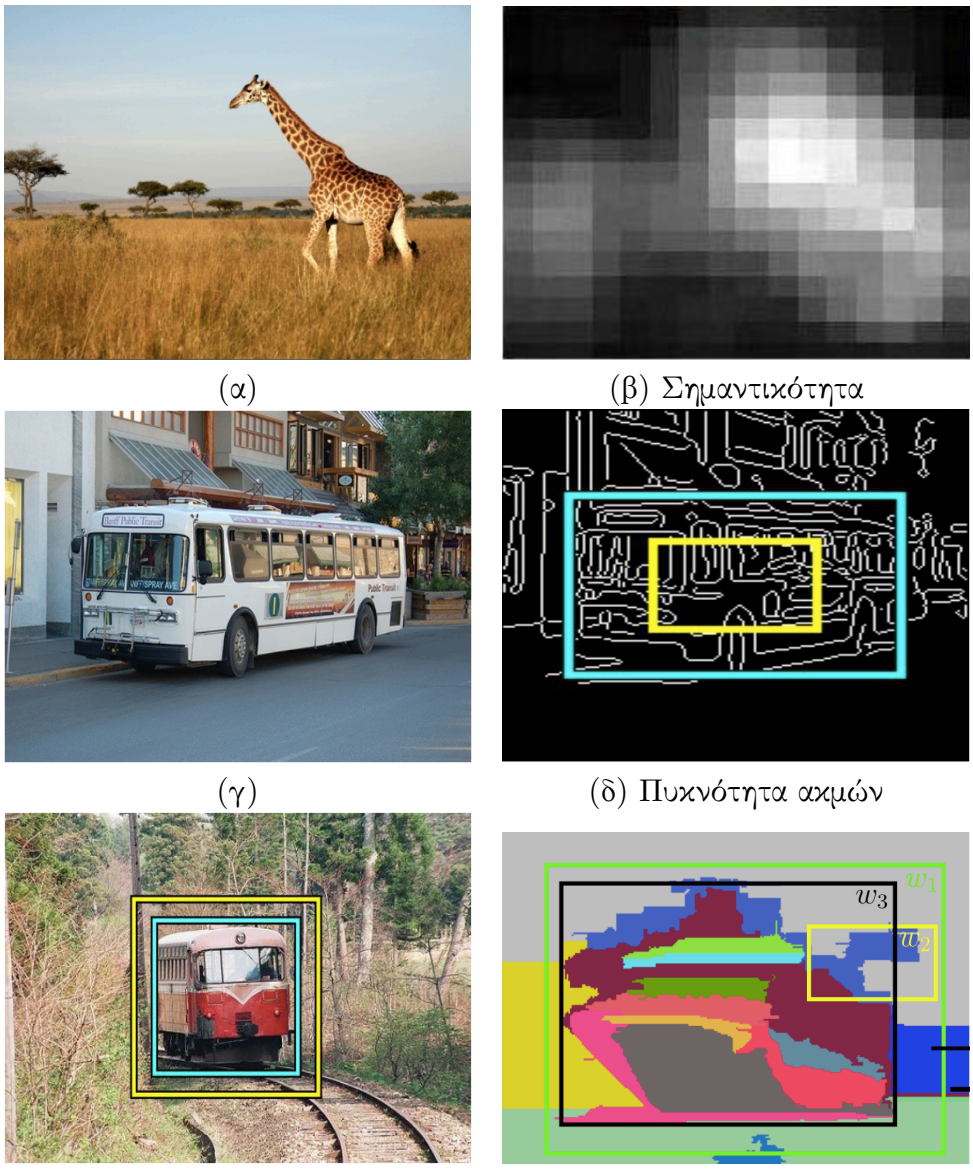
$$SS(w, \theta_{ss}) = 1 - \sum_{s \in S(\theta_{ss})} \frac{min(|s \backslash w|, |s \cap w|)}{|w|} \tag{2.2}$$

όπου:
- $S(\theta_{ss})$ είναι το σύνολο των τμημάτων που πήραμε από τον αλγόριθμο του Felzenszwalb [15],
- $s \cap w$ η περιοχή του τμήματος μέσα στο παράθυρο,
- $s \backslash w$ η περιοχή του τμήματος έξω από το παράθυρο,
- θ_{ss} η παράμετρος η οποία χρησιμοποιείται για την κατάτμηση.

Η παραπάνω σχέση μας λέει ότι τα τμήματα εντελώς μέσα ή έξω από το παράθυρο συμβάλουν μηδενικά στο άθροισμα. Για ένα τμήμα s, η συμβολή είναι μικρότερη αν περιέχεται σχεδόν όλο μέσα στο w ή είναι σχεδόν έξω σαν μέρος του φόντου. Συνεπώς το $SS(w, \theta_{ss})$ είναι μεγαλύτερο για παράθυρα w τα οποία είναι σφιχτά γύρω από αντικείμενα.

Όπως το θ_{ss}, έτσι και τα υπόλοιπα χαρακτηριστικά έχουν παραμέτρους οι οποίες επιλέγονται με βάση μια διαδικασία μάθησης. Όλα τα παραπάνω χαρακτηριστικά είναι συμπληρωματικά οπότε ο συνδυασμός τους αποφέρει καλύτερα αποτελέσματα. Ο συνδυασμός αυτός έγινε με το Αφελές Μπεζυανό μοντέλο (Naives Bayes model).

Τέλος, πραγματοποιείται μια διαδικασία πού ονομάζεται *non-maximum suppression* (nms) με την οποία δειγματοληπτούνται παράθυρα ανάλογα με τη βαθμολογία



(α) (β) Σημαντικότητα

(γ) (δ) Πυκνότητα ακμών

(ε) Αντίθεση χρώματος (στ) Επικάλυψη τμημάτων

Σχήμα 2.8: Χαρακτηριστικά της objectness. Στην (β) βλέπουμε ότι η καμηλοπάρδαλη ξεχωρίζει από το φόντο οπότε με χρήση FFT είναι εύκολο να ανιχνευθεί, στην (δ) το λεωφορίο έχει την μεγαλύτερη πυκνότητα ακμών οπότε και ανιχνεύεται από αυτό το χαρακτηριστικό, στην (ε) το τρένο διαφέρει χρωματικά από το περιβάλλον του και στην (στ) το πλοίο ανιχνεύεται στο σημείο που δεν υπάρχουν πολλά τμήματα που βρίσκονται πάνω στο σύνορο του υποψήφιου παραθύρου.

τους και την θέση τους σε σχέση με άλλα παράθυρα. Ο στόχος είναι διπλός: να πάρουμε τα πιο ποιοτικά παράθυρα και να έχουμε όσο το δυνατόν μεγαλύτερα ποικιλία θέσεων. Αυτό βοηθάει στην ανίχνευση περισσοτέρων αντικειμένων. Ξεκινώντας, επιλέγεται το παράθυρο με την μεγαλύτερη βαθμολογία. Στην συνέχεια επιλέγεται το επόμενο παράθυρο στην σειρά με την μεγαλύτερη βαθμολογία και κρατιέται μόνο αν δεν έχει μεγάλη επικάλυψη με κάποιο από τα πιο υψηλόβαθμα παράθυρα. Η διαδικασία επαναλαμβάνεται για όλα τα παράθυρα και ο τελικός αριθμός των παραθύρων εξαρτάται από την επιθυμητή τιμή της επικάλυψης με την οποία γίνεται η σύγκριση των παραθύρων.

Τα παράθυρα που χρησιμοποιούνται αρχικά και τα οποία είναι αυτά που βαθμολογούνται είναι 100000 και είναι ομοιόμορφα κατανεμημένα στην εικόνα.

Όπως μπορούμε να διαπιστώσουμε, τα διάφορα χαρακτηριστικά ανιχνεύουν και διαφορετικά αντικείμενα στις εικόνες και γι'αυτό χρησιμοποιούνται όλα μαζί, ώστε εκεί που αποτυγχάνει το ένα, να πετυχαίνει το άλλο. Καθώς το σημαντικότερο σύμφωνα με τις μετρήσεις τους είναι η επικάλυψη των τμημάτων, η μέθοδός μας όπως θα δούμε στην συνέχεια εμπνέεται από αυτό και προσπαθεί να εξαντλήσει τις δυνατότητές του σαν κριτήριο για την βαθμολόγηση των παραθύρων.

2.2.2 Μέθοδος του Rahtu

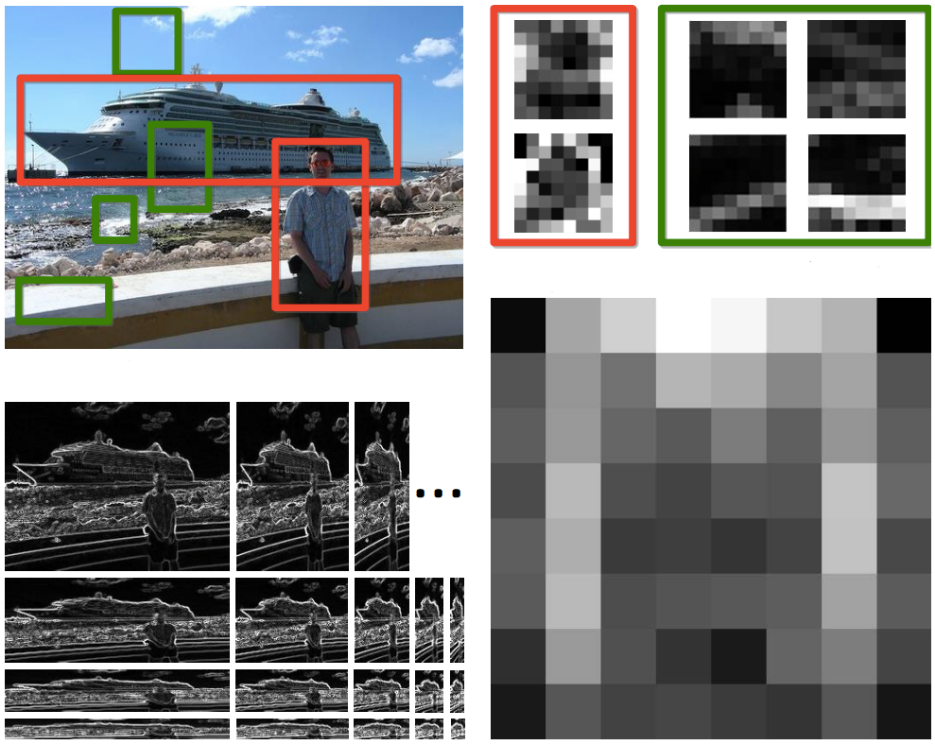
Η μέθοδος του *Rahtu* [28] βασίζεται στην *Objectness* και προσπαθεί να πετύχει καλύτερη βαθμολόγηση χρησιμοποιώντας ελαφρώς διαφορετικά χαρακτηριστικά. Επίσης διαφέρει ο τρόπος που επιλέγονται τα αρχικά παράθυρα και υπογραμμίζει την σημασία μιας καλής non-maximum suppression διαδικασίας.

Τα αρχικά παράθυρα που επιλέγονται προκύπτουν από παράθυρα γύρω από μονάδες, ζευγάρια ή τριάδες από τμήματα σε συνδυασμό με 100000 παράθυρα δειγματοληπτημένα από προηγούμενη διαδικασία μάθησης.

Τα χαρακτηριστικά που επιλέγονται είναι το όριο των τμημάτων, η κατανομή των ακμών και η συμμετρία του παραθύρου. Τα τμήματα προκύπτουν ξανά από τον αλγόριθμο του Felzenszwalb [15] ενώ τα άλλα δύο από τις ακμές και τη κλίση της εικόνας.

Το πρώτο χαρακτηριστικό που χρησιμοποιείται είναι βασισμένο στην επιτυχία της επικάλυψης των τμημάτων της μεθόδου *Objectness* και χρησιμοποιεί την αρχική υπερκατάτμηση σε τμήματα. Ωστόσο, αντί για τα ίδια τα τμήματα, εδώ ελέγχουν κατά πόσο ένα παράθυρο γύρω από το κάθε τμήμα, βγαίνει εκτός του ορίου του παραθύρου. Αυτό το χαρακτηριστικό (Superpixel Boundary Integral - BI) βρέθηκε ότι αποδίδει το ίδιο με το SS αλλά μπορεί να υπολογιστεί ταχύτερα. Το επόμενο χαρακτηριστικό χρησιμοποιεί ακμές και συγκεκριμένα την κατανομή τους μέσα στο υποψήφιο παράθυρο (Boundary Edge Distribution - BE). Όσο περισσότερες οι ακμές κοντά στα όρια του παραθύρου, τόσο μεγαλύτερη βαθμολογία έχει το παράθυρο αφού ένα αντικείμενο περικλείεται από ακμές. Τέλος χρησιμοποιείται η συμμετρία του παραθύρου (Window Symmetry - WS) πάνω στον οριζόντιο και στον κατακόρυφο άξονα. Το WS βασίζεται στις ίδιες κατευθυνόμενες ακμές που βασίζεται και το BE.

2.2.3 Bing



Σχήμα 2.9: Παρόλο που τα αντικείμενα (κόκκινα) και τα μη-αντικείμενα (πράσινα) έχουν μεγάλη ποικιλία στον χώρο της εικόνας (πάνω αριστερά), σε ένα μικρό μέγεθος (κάτω αριστερά), τα αντίστοιχα διανύσματα κλίσης όπως το χαρακτηριστικό NG (πάνω δεξιά), παρουσιάζουν εντυπωσιακή ομοιότητα. Στην μέθοδο γίνεται εκμάθηση ενός 64D γραμμικού μοντέλου (κάτω δεξιά) για την επιλογή των υποψήφιων θέσεων με βάση το χαρακτηριστικό NG.

Πρόκειται για μια αρκετά ενδιαφέρουσα μέθοδο κυρίως λόγω του πολύ μικρού χρόνου εκτέλεσής της. Σαν χαρακτηριστικό για την ανίχνευση των αντικειμένων, η *Bing* [7] χρησιμοποιεί την ένταση των διανυσμάτων κλίσης της εικόνας όταν αυτή έχει υποστεί σμίκρυνση σε μια 8×8 εικόνα, τα οποία τοποθετούνται σε ένα SVM για εκμάθηση. Οι συγγραφείς αναφέρουν ότι με την σμίκρυνση σε τόσο μικρό μέγεθος, η κλίση των διανυσμάτων ανάμεσα στα αντικείμενα δεν διαφέρει, λόγω της μικρής παραλλαγής που μπορεί να έχουν τα κλειστά σύνορα σε μια τόσο αφηριμένη προβολή. Έτσι αντικείμενα με διαφορετικό χρώμα, υφή, φωτεινότητα κλπ, έχουν παρόμοια εμφάνιση στην εικόνα διανυσμάτων κλίσης όπως φαίνεται και στο σχήμα 2.9. Τη χαρακτηριστική αυτή 8×8 αναπαράσταση την ονομάζουν 64D νόρμα κλίσης (Normed Gradient - NG).

Στην εργασία τους, αναφέρεται ανάκληση της τάξης του 95% για 1000 υποψήφια παράθυρα. Ωστόσο, όπως αποδείχθηκε αργότερα με την εργασία *CrackingBing*

[40] η επιτυχία της μεθόδου δεν οφείλεται στον τρόπο αναπαράστασης της εικόνας ούτε στην εκμάθηση του χαρακτηριστικού αυτού, και παρόμοια αποτελέσματα θα μπορούσαν να πάρουν αν ο αλγόριθμός τους δεν κοίταζε καθόλου το περιεχόμενο των εικονοστοιχείων. Αυτό στο οποίο οφείλεται η επιτυχία του αλγορίθμου είναι ο τρόπος που επιλέγονται τα αρχικά υποψήφια παράθυρα, ο οποίος εκμεταλλεύεται την μετρική IoU (για την οποία θα μιλήσουμε στο κεφάλαιο 4.3) με βάση την οποία υπολογίζεται η ανάκληση.

2.2.4 Edge Boxes

Η μέθοδος των Dollár και Zitnick *Edge Boxes* [41] είναι αυτή που μας απασχόλησε περισσότερο και η οποία μας έδωσε τον αρχικό κώδικα στον οποίο δουλέψαμε. Εδώ, οι υποψήφιες θέσεις βαθμολογούνται ανάλογα με τον αριθμό των περιγραμμάτων που περικλείονται εξολοκλήρου μέσα στο παράθυρο που δουλεύουμε κάθε φορά (σχήμα 2.10). Για τα αρχικά παράθυρα χρησιμοποιείται η τεχνική του κυλιόμενου παραθύρου τα οποία τοποθετούνται ανάλογα με τη θέση, το μέγεθος και την αναλογία των πλευρών τους.

Αναλυτικότερα, παίρνοντας μια εικόνα, αρχικά υπολογίζεται το κατά πόσο ένα εικονοστοιχείο αποτελεί ακμή, χρησιμοποιώντας τον [10] ο οποίος χρησιμοποιεί δάση απόφασης για τον γρήγορο υπολογισμό των ακμών. Σκοπός είναι να ανιχνευθούν όλα τα περιγράμματα που εξέχουν από το παράθυρο και τα οποία λοιπόν είναι απίθανο να ανήκουν σε αντικείμενο που περιέχεται σε αυτό. Για υπολογιστική αποδοτικότητα, οι ακμές που έχουν μεγάλη συγγένεια μεταξύ τους, ομαδοποιούνται.

Δεδομένου των ομάδων ακμών $s_i \in S$, υπολογίζεται η συγγένεια μεταξύ των γειτονικών ομάδων. Για ένα ζευγάρι ομάδων s_i και s_j, η συγγένεια υπολογίζεται με βάση την μέση τιμή θέσης τους x_i και x_j και τη μέση τιμή προσανατολισμού τους θ_i και θ_j. Διαισθητικά, οι ομάδες ακμών έχουν μεγάλη συγγένεια αν η γωνία μεταξύ της μέσης τιμής της θέσης τους είναι παρόμοια με τον προσανατολισμό των ομάδων. Πιο συγκεκριμένα, η συγγένεια $\alpha(s_i, s_j)$ υπολογίζεται ως εξής:

$$\alpha(s_i, s_j) = |\cos(\theta_i - \theta_{ij})\cos(\theta_j - \theta_{ij})|^\gamma, \tag{2.3}$$

όπου θ_{ij} είναι η γωνία μεταξύ των x_i και x_j. Η τιμή του γ ορίζει την ευαισθησία στην αλλαγή της κατεύθυνσης και στην μέθοδό τους οι Dollár και Zitnick χρησιμοποιούν την τιμή $\gamma = 2$.

Έχοντας τις ομάδες S και τις συγγένειές τους, μπορεί πλέον να υπολογιστεί η βαθμολογία του υποψήφιου παραθύρου b. Αρχικά υπολογίζεται το άθροισμα m_i της έντασης των ακμών που βρίσκονται σε κάθε ομάδα s_i. Επιλέγεται επίσης ένα εικονοστοιχείο-αντιπρόσωπος $\bar{x}_i$ για κάθε ομάδα. Ποιο εικονοστοιχείο θα επιλεχθεί δεν έχει σημασία.

Για κάθε ομάδα s_i υπολογίζεται μια συνεχής τιμή $w_b(s_i) \in [0, 1]$ η οποία δείχνει αν η s_i περιέχεται ολόκληρη μέσα στο b οπότε $w_b(s_i) = 1$ ή όχι οπότε $w_b(s_i) = 0$. Έστω S_b το σύνολο των ομάδων ακμών που προεξέχουν από τα όρια του παραθύρου b. Για όλα τα $s_i \in S_b$, το $w_b(s_i)$ τίθεται ίσο με 0. Όμοια, θέτουμε $w_b(s_i) = 0$ για

όλα τα s_i για τα οποία $\bar{x}_i \notin b$ αφού όλα τα εικονοστοιχεία πρέπει να είναι είτε έξω από το b είτε $s_i \in S_b$. Για τις υπόλοιπες ομάδες για τις οποίες $\bar{x}_i \in b$ και $s_i \notin S_b$ υπολογίζεται το $w_b(s_i)$ ως εξής:

$$w_b(s_i) = 1 - \max_T \prod_j^{|T|-1} \alpha(t_j, t_{j+1}), \tag{2.4}$$

όπου T είναι ένα μονοπάτι από ομάδες ακμών μήκους $|T|$ που ξεκινάει από κάποιο $t_1 \in S_b$ και τελειώνει σε ένα $t_{|T|} = s_i$. Αν δεν υπάρχει κανένα τέτοιο μονοπάτι, ορίζουμε $w_b(s_i) = 1$. Συνεπώς η παραπάνω εξίσωση βρίσκει το μονοπάτι με τη μεγαλύτερη συγγένεια μεταξύ των ομάδων s_i και μια ομάδα που προεξέχει από το όριο του παραθύρου. Αφού οι περισσότερες συγγένειες είναι μηδενικές, αυτός ο υπολογισμός μπορεί να γίνει γρήγορα.

Τελικά, για την βαθμολόγηση του κάθε παραθύρου, χρησιμοποιούνται οι τιμές του w_b που υπολογίσαμε και έχουμε:

$$h_b = \frac{\sum_i w_b(s_i) m_i}{2(b_w + b_h)^\kappa}, \tag{2.5}$$

όπου b_w και b_h είναι το πλάτος και το ύψος του παραθύρου. Η διαίρεση γίνεται με την περίμετρο του παραθύρου και όχι με το εμβαδόν γιατί το πλάτος των ακμών είναι ένα εικονοστοιχείο ανεξάρτητα από το μέγεθος. Χρησιμοποιείται $\kappa = 1.5$ επειδή μεγαλύτερα παράθυρα έχουν περισσότερες ακμές κατά μέσο όρο.

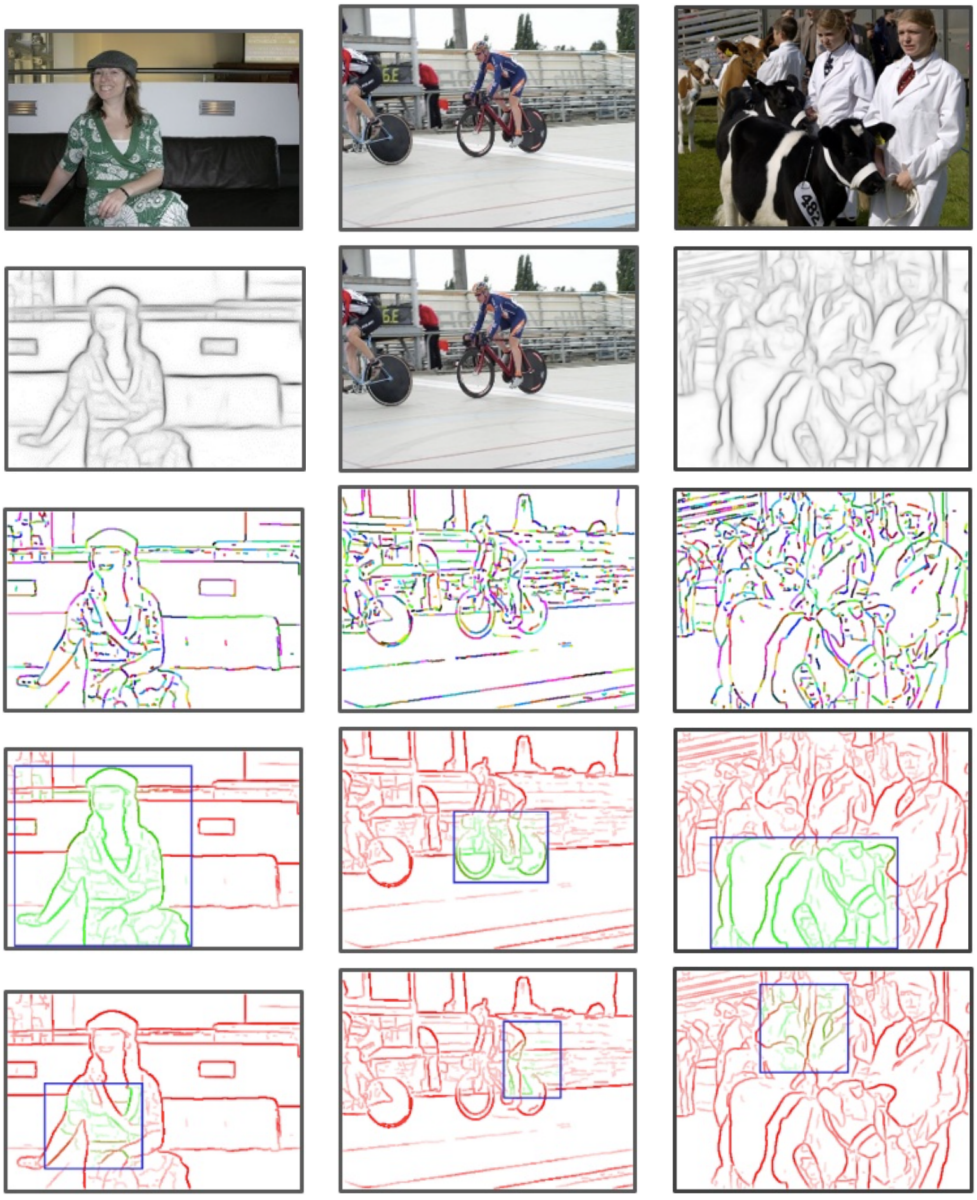
Τέλος παρατηρήθηκε ότι οι ακμές στο κέντρο του παραθύρου είναι μικρότερης σημασία από αυτές κοντά στα άκρα του, αφού υπάρχει μικρότερη πιθανότητα να ανήκουν στην περίμετρο του αντικειμένου [2]. Γι'αυτό το λόγο, για την τελική βαθμολογία αφαιρείται η ένταση των ακμών από ενα μικρότερο παράθυρο στο κέντρο του υποψήφιου παραθύρου:

$$h_b^{in} = h_b - \frac{\sum_{p \in b^{in}} m_p}{2(b_w + b_h)^\kappa}, \tag{2.6}$$

όπου το πλάτος και το μήκος του b^{in} είναι $b_w/2$ και $b_h/2$ αντίστοιχα.

Τα παράθυρα υφίστανται μια διαδικασία τελειοποίησης για την μεγιστοποίηση του h_b^{in} όσον αφορά την θέση, το μέγεθος και την αναλογία του παραθύρου. Τέλος ταξινομούνται και με μια διαδικασία NMS ο αλγόριθμος δίνει τις υποψήφιες θέσεις των αντικειμένων.

Ο αλγόρθμος αυτός δίνει πολύ καλά αποτελέσματα όσον αφορά την ανάκληση και με πολύ καλή ταχύτητα. Η μέθοδος αυτή είναι που έδωσε το έναυσμα στην δική μας εργασία και χάρη στις έξυπνες δομές δεδομένων που χρησιμοποιεί καταφέραμε να πετύχουμε και τα δικά μας αποτελέσματα.



Σχήμα 2.10: Από πάνω προς τα κάτω έχουμε (πρώτη γραμμή) την αρχική εικόνα, (δεύτερη γραμμή) τις ακμές από τον αλγόριθμο [10], (τρίτη γραμμή) τις ομάδες ακμών, (τέταρτη γραμμή) ένα παράδειγμα καλού υποψήφιου παραθύρου, και (πέμπτη γραμμή) ένα παράδειγμα κακού υποψήφιου παραθύρου. Οι πράσινες ακμές προβλέπεται να είναι μέρος του αντικειμένου ($w_b(s_i) = 1$), ενώ οι κόκκινες ακμές όχι ($w_b(s_i) = 0$). Η βαθμολόγηση του υποψήφιου παραθύρου βασίζεται μόνο στον αριθμό των περιγραμμάτων που περικλείονται *εξολοκλήρου* μέσα στο παράθυρο και αποτελούν μια πολύ καλή ένδειξη για το αν είναι αντικείμενο ή όχι.

2.3 Βασικές (baseline) μέθοδοι

Αυτές οι μέθοδοι θα χρησιμοποιηθούν για σύγκριση σαν κατώτερο όριο της ποιότητας των υπόλοιπων αλγορίθμων όπως θα δούμε στο κεφάλαιο 5.

Uniform. Τα παράθυρα είναι ομοιόμορφα κατανεμημένα μέσα στην εικόνα ως προς την θέση του κέντρου τους, το μέγεθός τους και την αναλογία των πλευρών τους.

Gaussian. Ανάλογα με την Uniform, τα παράθυρα τοποθετούνται με βάση ένα Γκαουσιανό μοντέλο ανάλογα με την θέση του κέντρου τους, το μέγεθός τος και την αναλογία των πλευρών τους.

Sliding Window. Τα παράθυρα τοποθετούνται μέσα σε ένα ορθογώνιο πλέγμα όπως γίνεται συνήθως στους ανιχνευτές αντικειμένων που χρησιμοποιούν την μέθοδο του κυλιόμενου παραθύρου. Ο αριθμός των παραθύρων που θέλουμε διανέμεται ανάλογα με το μέγεθός του (ύψος και πλάτος) και για κάθε μέγεθος, τα παράθυρα τοποθετούνται ομοιόμορφα. Η διαδικασία αυτή είναι εμπνευσμένη από την υλοποίηση της *Bing* [7].

Superpixels. Κάθε τμήμα της κατάτμησης θεωρείται υποψήφια θέση. Αυτή η μέθοδος χρησιμοποιείται σαν baseline λόγω της χρήσης της κατάτμησης σε πολλές μεθόδους όπως είδαμε.

2.4 Άλλες εργασίες

Εδώ θα θέλαμε να παρουσιάσουμε κάποιες άλλες εργασίες, οι οποίες δεν είναι πάνω στην ανίχνευση υποψήφιων θέσεων αντικειμένων αλλά χρησιμοποιούνται από αυτές για την εξαγωγή των προτάσεών τους. Θα περιγράψουμε δύο από τις πιό ευρέως χρησιμοποιούμενες εργασίες και τις οποίες χρησιμοποιήσαμε και εμείς για την δημιουργία της δικής μας μεθόδου.

2.4.1 Κατάτμηση εικόνας με γράφους

Στην όραση υπολογιστών, η κατάτμηση εικόνας είναι η διαδικασία του χωρισμού της ψηφιακής εικόνας σε πολλά τμήματα, ομάδες εικονοστοιχείων, γνωστά και ως superpixel. Στόχος της κατάτμησης είναι να απλοποιήσει και/ή να αλλάξει την αναπαράσταση της εικόνας σε μια πιο χρήσιμη και εύκολη μορφή για ανάλυση. Η κατάτμηση χρησιμοποιείται συχνά για να εντοπίζονται αντικείμενα και όρια. Πιο συγκεκριμένα, κατά την διαδικασία αυτή κάθε εικονοστοιχείο αποκτά μια τιμή έτσι ώστε εικονοστοιχεία με την ίδια τιμή να μοιράζονται κοινά χαρακτηριστικά.

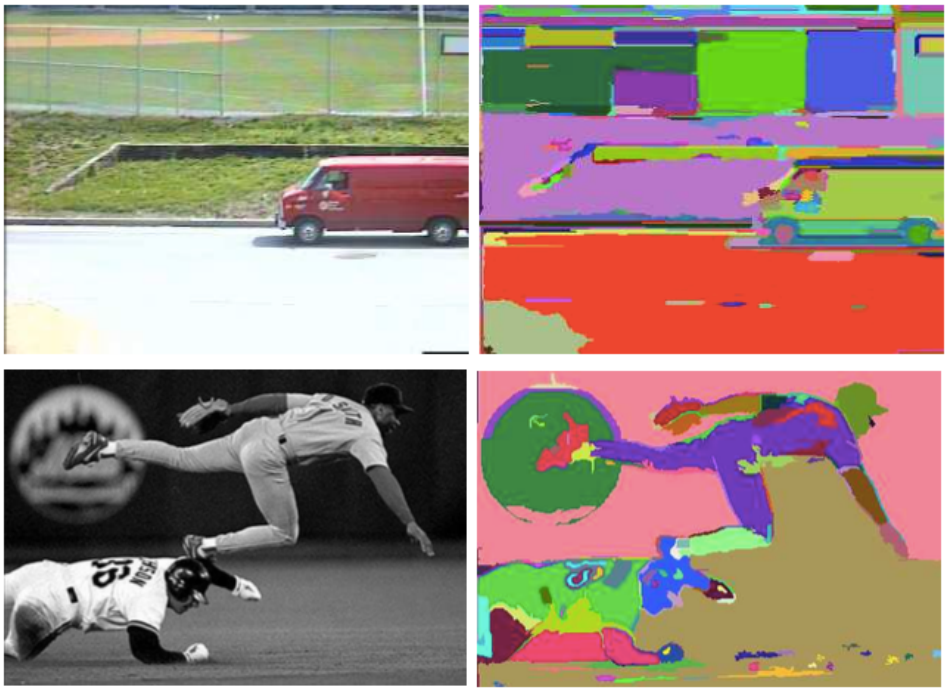
Το αποτέλεσμα της κατάτμησης είναι ένα σύνολο από τμήματα που συνολικά καλύπτουν όλη την εικόνα, ή ένα σύνολο από περιγράμματα τμημάτων (χρήσιμο για ανίχνευση ακμών). Τα εικονοστοιχεία κάθε περιοχής είναι παρόμοια όσον αφορά κάποια χαρακτηριστικά όπως χρώμα, ένταση και υφή. Γειτονικές περιοχές διαφέρουν σημαντικά ως προς αυτά τα χαρακτηριστικά.

Όπως είδαμε και παραπάνω, πολλές από τις τεχνικές ανίχνευσης υποψήφιων θέσεων αντικειμένων χρησιμοποιούν κατάτμηση για την παραγωγή των αποτελεσμάτων τους. Ο πιο διαδεδομένος αλγόριθμος για αυτή τη δουλειά είναι ο αλγόριθμος των Felzenszwalb και Huttenlocher [15]. Παραδείγματα κατάτμησης με τον συγκεκριμένο αλγόριθμο μπορούμε να δούμε στο σχήμα 2.11.

Σε αυτόν, ακολουθείται μια προσέγγιση γράφου. Έστω $G = (V, E)$ ένας μη κατευθυνόμενος γράφος με κόμβους $u_i \in V$, τα στοιχεία που θα κατατμηθούν, στην περίπτωσή μας τα εικονοστοιχεία, και ακμές $(u_i, u_j) \in E$ ανάμεσα στους γειτονικούς κόμβους. Κάθε ακμή έχει το αντίστοιχο βάρος $w(u_i, u_j)$ το οποίο δείχνει την ανομοιομορφία μεταξύ των δύο εικονοστοιχίων που συνδέει και συγκεκριμένα την διαφορά χρώματος. Μία κατάτμηση S είναι ένα διαχωρισμός του V σε κομμάτια τέτοια ώστε κάθε κομμάτι $C \in S$ να αντιστοιχεί σε ένα συνεκτικό κομμάτι σε ένα γράφο $G' = (V, E')$, όπου $E' \subseteq E$. Γενικά θέλουμε ακμές στο ίδιο κομμάτι να έχουν σχετικά μικρό βάρος, και ακμές μεταξύ των διαφορετικών κομματιών να έχουν μεγαλύτερο βάρος.

Για να αποφασίσει ο αλγόριθμος αν θα ενώσει δύο κομμάτια σε ένα τμήμα, χρησιμοποιείται μια συνάρτηση D. Για να οριστεί αυτή, πρέπει πρώτα να ορίσουμε την *εσωτερική διαφορά* ενός κομματιού $C \subseteq V$ ως το μέγιστο βάρος του ελάχιστου συνδετικού δέντρου του κομματιού, $\mathrm{MST}(C, E)$.

$$\mathrm{Int}(C) = \max_{e \in \mathrm{MST}(C,E)} w(e), \tag{2.7}$$



Σχήμα 2.11: Παραδείγματα κατάτμησης με $k = 300$

Στη συνέχεια ορίζεται η διαφορά μεταξύ δύο κομματιών $C_1, C_2 \subseteq V$ ως η μικρότερη τιμή ακμής που συνδέει τα δύο κομμάτια.

$$\text{Diff}(C_1, C_2) = \min_{u_i \in C_1, u_j \in C_2, (u_i, u_j) \in E} w(u_i, u_j), \tag{2.8}$$

Αν δεν υπάρχει ακμή που συνδέει τα C_1 και C_2 τότε θεωρούμε $\text{Diff}(C_1, C_2) = \infty$.

Η συνάρτηση D δείχνει ότι δύο κομμάτια πρέπει να είναι σε διαφορετικά τμήματα, αν υπάρχει ένδειξη για ύπαρξη συνόρου ανάμεσα στο κομμάτια, ελέγχοντας αν η διαφορά $\text{Diff}(C_1, C_2)$ ανάμεσα στα τμήματα είναι μεγάλη σε σχέση με την εσωτερική διαφορά τουλάχιστον ενός από τα δύο κομμάτια, $\text{Int}(C_1)$ και $\text{Int}(C_2)$. Μια συνάρτηση κατωφλίωσης χρησιμοποιείται για να ελέγχει τον βαθμό στον οποίο η διαφορά αυτή πρέπει να είναι μεγαλύτερη από την εσωτερική διαφορά. Έτσι έχουμε:

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if Diff}(C_1, C_2) < \text{MInt}(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases} \tag{2.9}$$

όπου η ελάχιστη εσωτερική διαφορά MInt ορίζεται ως

$$\text{MInt}(C_1, C_2) = \min\left(\text{Int}(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)\right), \tag{2.10}$$

Η συνάρτηση κατωφλίωσης τ ελέγχει τον βαθμό κατά τον οποίο η διαφορά μεταξύ των δύο τμημάτων πρέπει να είναι μεγαλύτερη από την εσωτερική διαφορά έτσι ώστε

να υπάρχει ένδειξη για σύνορο μεταξύ τους (η συνάρτηση D να επιστρέφει true). Για μικρά κομμάτια, η εσωτερική διαφορά $\mathrm{Int}(C)$ δεν είναι καλός εκτιμητής των τοπικών χαρακτηριστικών των δεδομένων. Στην ακραία περίπτωση που $|C| = 1$ τότε το $\mathrm{Int}(C) = 0$. Συνεπώς πρέπει να χρησιμοποιηθεί μια συνάρτηση κατωφλίωσης που βασίζεται στο μέγεθος του κομματιού,

$$\tau(C) = k/|C|, \tag{2.11}$$

όπου $|C|$ δείχνει το μέγεθος του C, και k είναι μια σταθερή παράμετρος. Με αυτόν τον τρόπο, μικρά κομμάτια χρειάζονται μεγαλύτερη ένδειξη για ύπαρξη συνόρου για να μην ενωθούν με γειτονικά τους. Όσο μεγαλύτερο το k, τόσο μεγαλύτερα και τα τμήματα που δίνει σαν έξοδο ο αλγόριθμος.

Algorithm 1: Segmentation algorithm

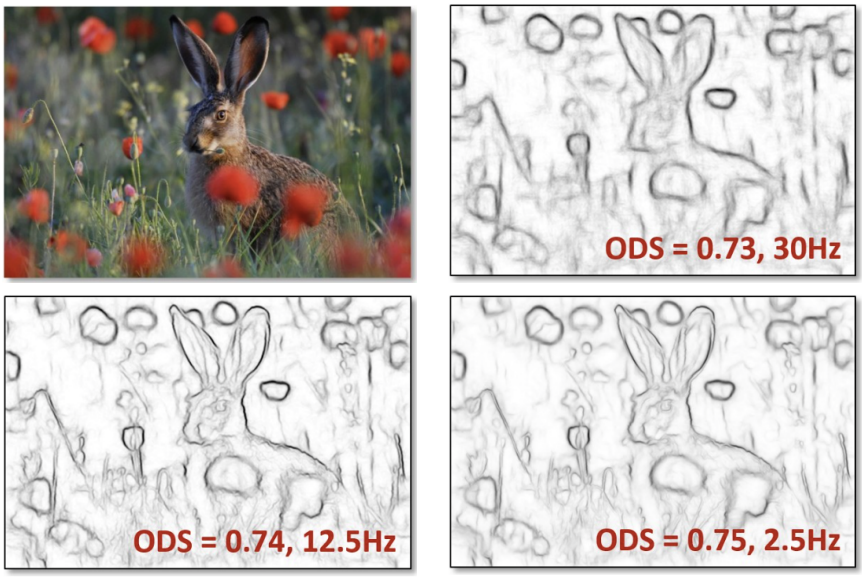
Data: ένας γράφος $G(V, E)$ με n κόμβους και m ακμές
Result: μια κατάτμηση του V σε τμήματα $S = (C_1, ..., C_r)$

1. Ταξινόμησε τις ακμές E στο $\pi = (o_1, ..., o_m)$ σε αύξουσα σειρά
2. Ξεκίνα με μια κατάτμηση S^0, όπου κάθε κόμβος u_i αποτελεί ξεχωριστό κομμάτι
3. Επανέλαβε το βήμα 4 για $q = 1, ..., m$
4. Έστω C_i^{q-1} το κομμάτι του S^{q-1} που περιέχει το u_i και C_j^{q-1} το κομμάτι που περιέχει το u_j. Αν $C_i^{q-1} \neq C_j^{q-1}$ και $w(o_q) \leq \mathrm{MInt}(C_i^{q-1}, C_j^{q-1})$ τότε το S^q παίρνεται από το S^{q-1} συγχωνεύοντας τα C_i^{q-1} και C_j^{q-1}. Αλλιώς $S^q = S^{q-1}$.
5. *return*$(S = S^m)$

Ο αλγόριθμος των Felzenszwalb και Huttenlocher καταφέρνει να κατασκευάσει μια κατάτμηση που δεν είναι ούτε τραχιά ούτε λεπτή (περισσότερα γι'αυτό στην δημοσίευση [15]). Επίσης χρειάζεται πολύ μικρό χρόνο εκτέλεσης καθώς ακολουθεί το μοντέλο του Kruskal για τον ομώνυμο αλγόριθμο κατασκευής ελάχιστου συνδετικού δέντρου με την χρήση ασύνδετου συνόλου (disjoint set), κάτι που οδηγεί σε πολυπλοκότητα $O(m \log m)$.

2.4.2 Ανίχνευση ακμών

Η ανίχνευση ακμών είναι μια πολύ βασική διαδικασία στην όραση υπολογιστών καθώς έχει πολλές εφαρμογές τόσο στην ανίχνευση αντικειμένων, όσο και στην κατάτμηση και στην εξαγωγή συνόρων. Είναι φυσικό λοιπόν να χρησιμοποιηθεί και



Σχήμα 2.12: Ποιότητα των ακμών που ανιχνεύονται για τις διάφορες προσεγγίσεις της μεθόδου. Παρατηρούμε ότι τα αποτελέσματα είναι πολύ καλά ακόμα και για την προσέγγιση πραγματικού χρόνου (30Hz)

για την ανίχνευση πιθανών θέσεων αντικειμένων και όπως είδαμε, πολλές μέθοδοι χρησιμοποιούν τις ακμές σαν πληροφορία για την εξαγωγή των αποτελεσμάτων τους.

Πολλές μέθοδοι έχουν προταθεί για την συγκεκριμένη εργασία, όπως ο γνωστός ανιχνευτής Canny [5]. Πολλές μετέπειτα τεχνικές άρχισαν να εξετάζουν την χρήση εκμάθησης για την ανίχνευση ακμών και πρόσφατα προτάθηκε από τους Dollár και Zitnick μια νέα μέθοδος [10] που χρησιμοποιεί τυχαία δάση απόφασης και δομημένη μάθηση (structured learning) και η οποία παράγει πολύ καλής ποιότητας ακμές σε πραγματικό χρόνο (30 εικόνες το δευτερόλεπτο). Οι μέθοδοι *Edge Boxes*, *MCG* και *Geodesic* χρησιμοποιούν την [10]. Στο σχήμα 2.12 μπορούμε να δούμε κάποια αποτελέσματα αυτής της μεθόδου.

Όπως είπαμε, η μέθοδος αυτή ορίζει το πρόβλημα της ανίχνευσης ακμών σε ένα πλαίσιο δομημένης μάθησης που χρησιμοποιεί τυχαία δάση απόφασης. Ένα δέντρο απόφασης $f_t(x)$ ταξινομεί την είσοδο $x \in X$ χωρίζοντας τα δεδομένα ανάμεσα στο αριστερό και το δεξί υποδέντρο, σύμφωνα με μία δυαδική συνάρτηση διαχωρισμού

$$h(x, \theta_j) \in \{0, 1\} \tag{2.12}$$

με παράμετρο θ_j σε κάθε κόμβο j.

Δεδομένου ενός κόμβου j και ενός συνόλου εκπαίδευσης $S \subset X \times Y$, ο στόχος της εκπαίδευσης του δέντρου απόφασης είναι να βρεθούν οι παράμετροι θ_j που μεγιστοποιούν το κριτήριο κέρδους πληροφορίας I_j. Αυτό ορίζεται ως

$$I_j = I(S_j, S_j^L, S_j^R), \tag{2.13}$$

όπου $S_j^L = (x, y) \in S_j | h(x, \theta_j) = 0$, $S_j^R = S_j \setminus S_j^R$. Όταν η δυαδική συνάρτηση $h(x, \theta_j) = 0$, η είσοδος x ταξινομείται στο αριστερό υποδέντρο, αλλιώς στο δεξί. Η διαδικασία τερματίζει όταν φτάσει σε έναν κόμβο φύλλο (τερματικό). Η έξοδος $y \in Y$ είναι η πρόβλεψη για την είσοδος x και αποθηκεύεται στον κόμβο φύλλο. Το δάσος απόφασης είναι ένα σύνολο από ανεξάρτητα δέντρα απόφασης f_t. Τα αποτελέσματα των προβλέψεων κάθε δέντρου $f_t(x)$ για κάθε είσοδο x συνδυάζονται σε μια τελική απόφαση. Η επιλογή του τρόπου συνδυασμού εξαρτάται από τον τύπο του χώρου Y.

Το βασικό πρόβλημα της ταξινόμησης με δέντρα απόφασης είναι το φαινόμενο της υπερπροσαρμογής. Τα δάση απόφασης λύνουν αυτό το πρόβλημα με την χρήση πολλών, μη συσχετισμένων δέντρων. Σημαντικό λοιπόν στην εκπαίδευση των δέντρων είναι η δημιουργία μεγάλης ποικιλίας διαφοροποιημένων δέντρων. Αυτό γίνεται είτε με την τυχαία υποδειγματοληψία των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του κάθε δέντρου, είτε με την τυχαία υποδειγματοληψία των χαρακτηριστικών και της συνάρτησης διαχωρισμού που χρησιμοποιείται σε κάθε κόμβο του δέντρου. Η δεύτερη επιλογή παράγει μοντέλα καλύτερης ακρίβειας και χρησιμοποιείται περισσότερο στην πράξη. Τελικά, η απώλεια ακρίβειας ενός δέντρου συμβάλει στην μεγαλύτερη ποικιλία του συνόλου.

Στην εργασία τους, οι Dollár και Zitnick επέκτειναν την ιδέα των τυχαίων δασών απόφασης για να παράγουν δομημένη έξοδο Y (structured output). Δεδομένου ενός υποπαραθύρου της εικόνας $x \in X$, η έξοδος $y \in Y$ αποθηκεύει την αντίστοιχη μάσκα κατάτμησης ή την αντίστοιχη δυαδική απεικόνιση ακμών. Η μάσκα κατάτμησης συμβολίζεται ως $y \in Y = \mathbb{Z}^{d \times d}$ και η δυαδική απεικόνιση ακμών ως $y' \in Y' = \{0, 1\}^{d \times d}$, όπου d το πλάτος του υποπαραθύρου. Η εκπαίδευση του ταξινομητή τυχαίου δάσους με δομημένη έξοδο είναι δύσκολή λόγω της μεγάλης διάστασης του χώρου εξόδου και της συνάρτησης που χρησιμοποιείται για να οριστεί το κριτήριο κέρδους πληροφορίας.

Ο κύριος στόχος του δομημένου δάσους απόφασης είναι να απεικονίσει όλες τις δομημένες τιμές σε ένα διακριτό σύνολο $c \in C$. Οι συγγραφείς λύνουν αυτό το πρόβλημα κάνοντας πρώτα μια απεικόνιση του δομημένου χώρου εξόδου Y σε έναν ενδιάμεσο χώρο Z. Το πρόβλημα της μεγάλης διάστασης του δομημένου χώρου εξόδου Y αμβλύνεται δειγματοληπτώντας m διαστάσεις του Z που ακολουθούν την Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA). Όσον αφορά το κριτήριο κέρδους πληροφορίας, η ομοιότητα πάνω στον Y λαμβάνεται υπολογίζοντας την απόσταση στον Z. Τελικά, ο ενδιάμεσος αυτός χώρος Z, απεικονίζεται πάνω σε διακριτές τιμές του χώρου C.

Για βελτίωση της ποιότητας ανίχνευσης, εφαρμόζεται ο αλγόριθμος σε διάφορες αναλύσεις της εικόνας με την συνολική έξοδο να είναι το μέσο όρο των επιμέρους αποτελεσμάτων καθώς και μια τεχνική όξυνσης ακμών.

Η εξαιρετική αποδοτικότητα του αλγορίθμου οφείλεται κυρίως στην χρήση δομημένων τιμών που προβλέπουν πληροφορία για ένα υποπαράθυρο της εικόνας. Με αυτόν τον τρόπο μειώνεται δραστικά ο αριθμός T των δέντρων που πρέπει να αξιολογηθούν.

Κεφάλαιο 3

Μέθοδος Segment Boxes

Μελετώντας τις παραπάνω μεθόδους, διαπιστώσαμε ότι δεν υπάρχει μια βέλτιστη τεχνική για την παραγωγή υποψήφιων θέσεων αντικειμένων σε εικόνα. Στόχος στη διπλωματική μας λοιπόν ήταν αν πάρουμε τις ιδέες που φαίνεται ότι είναι καλές και να δοκιμάσουμε να κατασκευάσουμε μια δική μας μέθοδο. Έτσι μπορούμε να διαπιστώσουμε τι είναι καλό και τι όχι για το συγκεκριμένο έργο και να προσπαθήσουμε να εξαντλήσουμε τις δυνατότητες της κατάτμησης για αυτόν τον σκοπό.

Η μέθοδος Segment Boxes που προτείνουμε χρησιμοποιεί την αρχική επιλογή παραθύρων της μεθόδου Edge Boxes έτσι ώστε με λίγα σχετικά παράθυρα να υπάρχει μεγάλη κάλυψη της εικόνας, αλλά η βαθμολόγησή τους δεν γίνεται πλέον με βάση τις ακμές αλλά τα τμήματα της εικόνας που προκύπτουν από κατάτμηση με τον αλγόριθμο των Felzenszwalb και Huttenlocher. Κάθε παράθυρο βαθμολογείται με βάση το εμβαδόν των τμημάτων που βρίσκονται αποκλειστικά μέσα στο παράθυρο κανονικοποιημένο ως προς το εμβαδόν του παραθύρου, έτσι ώστε να μην δίνεται βαρύτητα στα μεγάλα παράθυρα.

Ο λόγος που επιλέξαμε η μέθοδός μας να είναι μέθοδος βαθμολόγησης παραθύρου και όχι ομαδοποίησης είναι εμφανής στο σχήμα 3.1. Σε μια τέτοια περίπτωση, οι αλγόριθμοι ομαδοποίησης θα έδιναν σαν πιθανές θέσεις τα παράθυρα (β'),(γ') και (δ') αλλά το παράθυρο (ε') δεν θα προέκυπτε ποτέ καθώς για να ενωθούν τα δύο δεξιά τμήματα θα πρέπει πρώτα να ενωθούν τα δύο αριστερά ως πιο κοντινά. Ωστόσο σε μερικές περιπτώσεις αντικειμένων, το παράθυρο (ε') είναι χρήσιμο και δεν θέλουμε να το χάσουμε. Ακολουθώντας την βαθμολόγηση παραθύρου, το παράθυρο (ε') θα προκύψει κατά την εκτέλεση του αλγορίθμου και θα βαθμολογηθεί αν και πιθανώς με λιγότερη βαθμολογία από τα υπόλοιπα.

Στη συνέχεια θα περιγράψουμε τα στοιχεία υλοποίησης της μεθόδου καθώς και διάφορες άλλες τροποποιήσεις που δοκιμάσαμε.

(α) (β) (γ) (δ) (ε)

Σχήμα 3.1: (α) Εικόνα εισόδου. (β)-(δ) Υποψήφια παράθυρα με όλες τις μεθόδους. (ε) Παράθυρο που δεν μπορεί να προκύψει από μεθόδους ομαδοποίησης λόγω της σειράς με την οποία γίνεται η ένωση.

3.1 Περιγραφή μεθόδου

Αρχικά, τρέχουμε τον αλγόριθμο κατάτμησης [15] που περιγράψαμε στο κεφάλαιο 2.4.1 με παράμετρο $k = 150$ έτσι ώστε να πάρουμε μια υπερκατάτμηση, η οποία όμως διατηρεί την μορφή των αντικειμένων της εικόνας. Για τις υπόλοιπες παραμέτρους κρατάμε τις προτεινόμενες τιμές. Σε κάθε τμήμα που προκύπτει δίνουμε και μία ξεχωριστή τιμή τμήματος.

Με μικρή επέμβαση στον παραπάνω αλγόριθμο, εκτός από τα ίδια τα τμήματα μπορούμε να βρούμε και τους γείτονές τους με χρήση των ακμών του γράφου της εικόνας. Αφού πραγματοποιηθεί η κατάτμηση, εξετάζουμε όλες τις ακμές μεταξύ των εικονοστοιχείων. Αν τα εικονοστοιχεία στα δύο άκρα των ακμών αντιστοιχίζονται σε ίδιο τμήμα, άρα και τα εικονοστοιχεία έχουν ενωθεί κατά την εκτέλεση του αλγορίθμου, δεν μας ενδιαφέρουν. Αυτά που βρίσκονται σε διαφορετικά τμήματα μας δείχνουν γειτνίαση.

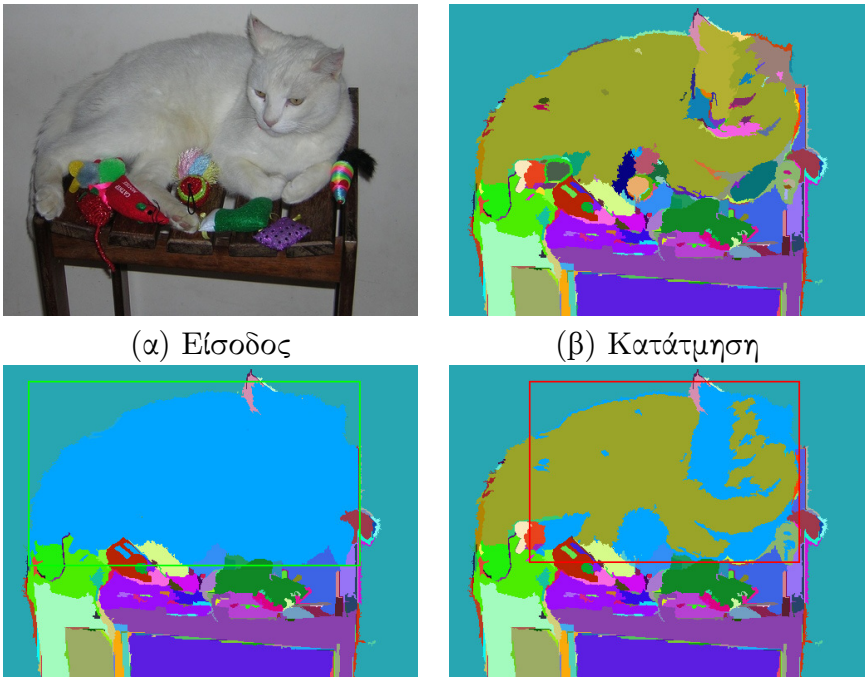
Στη συνέχεια επιλέγουμε τα αρχικά παράθυρα που θα βαθμολογήσουμε, ακολουθώντας την μέθοδο Edge Boxes. Σκοπός μας είναι με όσο δυνατόν λιγότερα παράθυρα να καλύπτονται όλα τα πιθανά αντικείμενα, έτσι ώστε να έχουμε μεγάλη ταχύτητα εκτέλεσης. Συγκεκριμένα διαλέγουμε παράθυρα μεγέθους από 1000 εικονοστοιχεία μέχρι ολόκληρη την εικόνα και με αναλογία πλάτους-μήκους που κυμαίνεται από $1/\tau$ ως τ, όπου $\tau = 3$ στην εργασία μας. Τα παράθυρα τοποθετούνται έτσι ώστε να υπάρχει μεταξύ τους επικάλυψη ίση με α όπου εμείς χρησιμοποιούμε $\alpha = 0.65$. Η παράμετρος αυτή ρυθμίζεται για να πετύχουμε καλύτερα αποτελέσματα σε διαφορετικό IoU (θα συζητήσουμε γι'αυτό στο κεφάλαιο 5).

Για να συμπεριλάβουμε και τα αντικείμενα που βρίσκονται στα όρια της εικόνας, επεκτείνουμε την εικόνα γύρω γύρω κατά ένα εικονοστοιχείο. Στα εικονοστοιχεία αυτά δεν αποδίδουμε τιμή τμήματος καθώς δεν πρόκειται για πραγματικό τμήμα επομένως δεν πρέπει να συμμετέχει στην παραγωγή των υποψήφιων θέσεων. Πρακτικά τα προσθέτουμε για να μπορούμε να πάρουμε παράθυρα στο όριο της εικόνας και να μην βγαίνουμε εκτός του ορίου της όταν κοιτάμε εκατέρωθεν του παραθύρου.

Στη συνέχεια με δεδομένη τη κατάτμηση, βρίσκουμε το εμβαδόν area του κάθε τμήματος u_i δηλαδή το πλήθος των εικονοστοιχείων p_i από τα οποία αποτελείται:

$$\text{area}(u_i) = |p_i|, \tag{3.1}$$

Στην συνέχεια ορίζουμε ένα από αυτά αντιπρόσωπο για κάθε τμήμα ο οποίος



(α) Είσοδος (β) Κατάτμηση

(γ) Καλό υποψήφιο παράθυρο (δ) Κακό υποψήφιο παράθυρο

Σχήμα 3.2: (α) Εικόνα εισόδου. (β) Κατάτμηση που προκύπτει από τον αλγόριθμο των Felzenszwalb και Huttenlocher με $k = 150$. (γ) Υποψήφιο παράθυρο με μεγάλη βαθμολογία από τον αλγόριθμό μας. Βλέπουμε ότι το μεγαλύτερο μέρος του παραθύρου αποτελείται από ολόκληρα τμήματα (μπλέ). (δ) Υποψήφιο παράθυρο με μικρή βαθμολογία. Μεγάλο ποσοστό του περιέχει τμήματα που βγαίνουν εκτός του ορίου του παραθύρου.

χρησιμοποιείται αργότερα για να ελέγξουμε αν το τμήμα βρίσκεται εντός ή εκτός του παραθύρου μας. Η επιλογή του αντιπροσώπου δεν έχει σημασία αλλά στην πράξη χρησιμοποιούμε το κάτω δεξιά εικονοστοιχείο του τμήματος λόγω της φοράς με την οποία κοιτάμε τα εικονοστοιχεία

$$\bar{x}_i = \text{random}\,(p_i). \tag{3.2}$$

Για την βαθμολόγηση των υποψήφιων παραθύρων από τον αλγόριθμό μας εργαζόμαστε ως εξής. Αρχικά βρίσκουμε τα τμήματα τα οποία βγαίνουν εκτός του παραθύρου που μας ενδιαφέρει. Έστω το σύνολο όλων αυτών των τμημάτων S_b. Επειδή αυτή η διαδικασία γίνεται για κάθε παράθυρο και άρα θα εκτελεστεί πολλές φορές κατά την διάρκεια εκτέλεσης του αλγορίθμου μας, είναι απαραίτητη μια έξυπνη διαδικασία για τον υπολογισμό του S_b. Γι'αυτό το λόγο, όπως και στην [41] δημιουργούμε δύο ακόμα δομές. Παρακάτω θα περιγράψουμε τον τρόπο που βρίσκουμε ποιά τμήματα τέμνουν ένα οριζόντιο σύνορο από το εικονοστοιχείο (α, r) ως το (β, r) ενώ για τα τμήματα που τέμνουν κάθετα σύνορα εργαζόμαστε με εντελώς ανάλογο τρόπο. Για τα οριζόν-

τια σύνορα, δημιουργούμε δύο δομές για κάθε γραμμή της εικόνας. Η πρώτη δομή L_r περιέχει τιμές τμημάτων. Η τιμή του κάθε τμήματος αποθηκεύεται με την σειρά που εμφανίζεται πάνω στην γραμμή από αριστερά προς τα δεξιά στην γραμμή r. Μια νέα τιμή προστίθεται μόνο αν αλλάζει η τιμή τμήματος πάνω στην γραμμή. Έτσι, καθώς είναι πολύ πιθανό εικονοστοιχεία πάνω στην γραμμή να ανήκουν σε ίδιο τμήμα, το μήκος της L_r είναι πολύ μικρότερο από το πλάτος της εικόνας και άρα η αναζήτηση σε αυτήν είναι πολύ ταχύτερη. Ωστόσο η δεύτερη δομή είναι αυτή που μας επιτρέπει να προσπελάσουμε την πρώτη για να μπορούμε να χρησιμοποιήσουμε τα δεδομένα που είναι αποθηκευμένα σε αυτήν. Έτσι κατασκευάζουμε μια δομή K_r με μέγεθος ίσο με το πλάτος της εικόνας και η οποία αποθηκεύει τις τιμές της θέσης στην L_r για κάθε εικονοστοιχείο της στήλης c στην γραμμή r. Συνεπώς, αν το εικονοστοιχείο p στην θέση (c, r) είναι μέλος του τμήματος s_i, τότε $L_r(K_r(c)) = i$. Αφού πολλά εικονοστοιχεία ανήκουν σε λίγα τμήματα, οι δύο αυτές δομές μπορούν να χρησιμοποιηθούν αποδοτικά για να βρούμε τα τμήματα που περνάνε το σύνορο που δημιουργείται από τα (α, r) και (β, r) ψάχνοντας τα δεδομένα στην L_r από τη θέση $K_r(\alpha)$ ως την θέση $K_r(\beta)$.

Έπειτα, αρχίζοντας από κάθε τέτοιο τμήμα, κάνουμε μια αναζήτηση για να βρούμε ποια άλλα τμήματα βρίσκονται εντός του παραθύρου. Αυτό γίνεται ελέγχοντας αν το εικονοστοιχείο αντιπρόσωπος του κάθε τμήματος που εντοπίζουμε κατά την αναζήτηση των γειτονικών τμημάτων βρίσκεται εντός του παραθύρου. Αν είναι μέσα, το προσθέτουμε στο σύνολο με τα τμήματα του S_b, έτσι ώστε να κοιτάξουμε στην συνέχεια τους δικούς του γείτονες. Αν δεν είναι μέσα σημαίνει ότι είτε όλο το τμήμα είναι εκτός, είτε ότι το τμήμα τέμνει το σύνορο. Και στις δύο περιπτώσεις το τμήμα δεν μας ενδιαφέρει και αγνοείται. Προφανώς αν το έχουμε ξαναεπισκεφτεί, το προσπερνάμε. Τελικά έχουμε στο S_b όλα τα τμήματα που βρίσκονται μέσα στο παράθυρο καθώς και αυτά που το τέμνουν.

Παίρνοντας ένα ένα τα εσωτερικά τμήματα, αρχίζουμε να τα ενώνουμε με τα γειτονικά τους προσθέτοντας τα εμβαδά τους. Αυτό γίνεται με την χρήση μιας δομής δεδομένων που ονομάζεται disjoint set η οποία λειτουργεί ως εξής: Αρχικά όλα τα τμήματα είναι κόμβοι που έχουν σαν πατέρα τον εαυτό τους, ύψος 1 και τιμή το εμβαδόν του τμήματος. Όταν δύο τμήματα πρόκειται να ενωθούν, ο ένας από τους δύο κόμβους αποκτά τιμή το άθροισμα των εμβαδών και ο άλλος αποκτά πατέρα τον πρώτο. Αν έχουν ίδια τιμή ύψους, τότε το ύψος του πρώτου αυξάνεται κατά 1. Με αυτόν τον τρόπο, αν κάποιο άλλο τμήμα πρόκειται να ενωθεί με τον δεύτερο κόμβο και συνεπώς με τον συνδυασμό των κόμβων, αρκεί να το συνδέσουμε με τον πατέρα του. Η δομή αυτή προσφέρει ευχρηστία και ταχύτητα στην εκτέλεση του αλγορίθμου μας.

Τελικά επιστρέφουμε σαν βαθμολογία του παραθύρου το εμβαδόν του πατέρα του τμήματος που προκύπτει από την ένωση όλων των εσωτερικών τμημάτων, άρα το συνολικό εμβαδόν των τμημάτων, κανονικοποιημένο ως προς τις διαστάσεις του παραθύρου, έτσι ώστε να μην δίνεται έμφαση στα μεγάλα τμήματα

$$\text{score}(b) = \frac{\sum_{i \in R} \text{area}(u_i)}{\text{area}(b)}, \tag{3.3}$$

όπου b το υποψήφιο παράθυρο που εξετάζουμε, R το σύνολο όλων των τμημάτων εξολοκλήρου εντός του υποψήφιου παραθύρου, area(u_i) το εμβαδόν του τμήματος u_i και area(b) το εμβαδόν του υποψήφιου παραθύρου.

Αφού γίνει η βαθμολόγηση του παραθύρου, το περνάμε από μια διαδικασία τελειοποίησης (refinement) έτσι ώστε να μεγιστοποιηθεί η βαθμολογία του πάνω στην θέση, το μέγεθος και την αναλογία του παραθύρου. Πραγματοποιούμε μια άπληστη αναζήτηση της καλύτερης βαθμολογίας μικραίνοντας κάθε φορά το παράθυρο σε μία από τις τέσσερις πλευρές του κατά κάποιο βήμα *step*. Αφού βρούμε το βέλτιστο τώρα παράθυρο, μειώνουμε το *step* στο μισό. Η διαδικασία επαναλαμβάνεται μέχρι το βήμα να είναι μικρότερο από 2 εικονοστοιχεία.

Αφού γίνει αυτό για όλα τα υποψήφια παράθυρα, ταξινομούνται ανάλογα με τη βαθμολογία τους και ακολουθεί ένα non-maximum suppression (nms) για να απομακρυνθούν παράθυρα που συμπίπτουν σε μεγάλο βαθμό. Στην εργασία μας το nms γίνεται με βάση την επικάλυψη των παραθύρων και η τιμή του τίθεται ίση με 0,75 IoU (Intersection over Union).

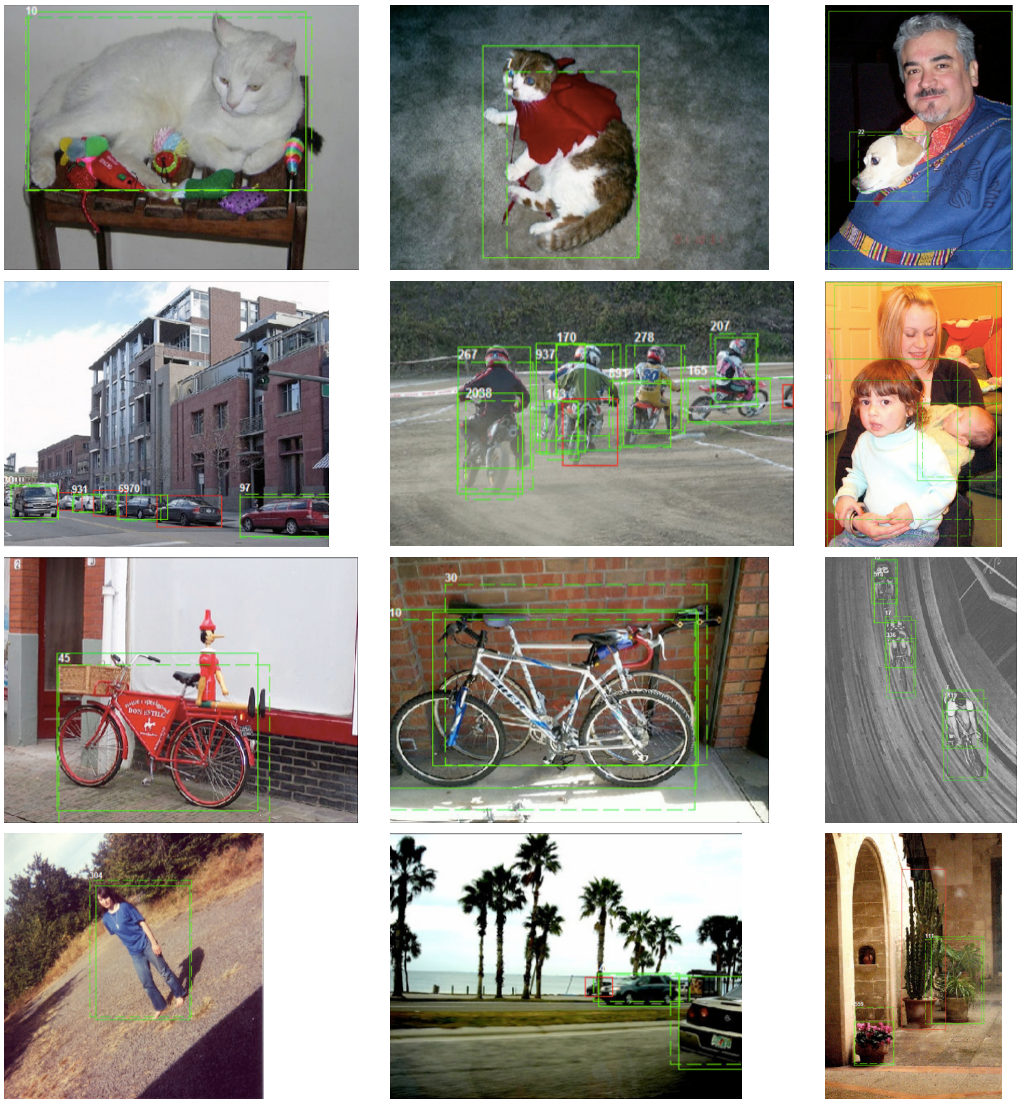
Αξίζει να σημειωθεί ότι ο αλγόριθμός μας (αλγόριθμος 2) δεν χρειάζεται σε κανένα σημείο διαδικασία εκμάθησης που σημαίνει ότι είναι τελείως ανεξάρτητος από υπάρχοντα δεδομένα. Η ιδέα του αλγορίθμου είναι πολύ απλή και χάρη σε έξυπνες δομές δεδομένων, η βαθμολόγηση μπορεί να γίνει πολύ γρήγορα, μόλις σε 2 δευτερόλεπτα ανά εικόνα. Στην συνέχεια θα παρουσιάσουμε και διάφορες άλλες προσεγγίσεις που κάναμε στην προσπάθειά μας να βελτιώσουμε περαιτέρω την μέθοδό μας.

Ακολουθούν παραδείγματα εκτέλεσης του αλγορίθμου μας σε πραγματικές εικόνες της βάσης εικόνων PASCAL VOC07 για υποψήφια παράθυρα με επικάλυψη ίση με 0.7 με το αντίστοιχο παράθυρο αναφοράς (ground truth) (σχήμα 3.3).

Algorithm 2: Segment Boxes

Data: image
Result: object proposal boxes

boxes $\leftarrow \varnothing$;
windows $\leftarrow$ all possible windows to score;
get segments and neighbours from Felzenszwalb;
for $b \in$ windows **do**
    $S_b \leftarrow \varnothing, R_b \leftarrow \varnothing$;
    $S_b \leftarrow$ borderSegments(b);
    visited $\leftarrow S_b$;
    for $s \in S_b$ **do**
        $S_b \leftarrow S_b \setminus s$;
        if ($\bar{x}_s$ in b) $\wedge$ ($s \notin$ borderSegments(b)) **then**
            $R_b \leftarrow R_b \cup s$;
        $A_b \leftarrow$ not visited neighbours[s];
        visited $\leftarrow$ visited $\cup A_b$;
        $S_b \leftarrow S_b \cup A_b$;
    score[b] $\leftarrow \sum_{i \in R_b}$ area(u_i)/ area(b);
    boxes $\leftarrow$ boxes $\cup$ b;
refine(boxes);
sort(boxes);
nms(boxes);
return boxes;



Σχήμα 3.3: Παραδείγματα ανίχνευσης αντικειμένων με την μέθοδο Segment Boxes. Το πράσινο συνεχές δείχνει τα παράθυρα αναφοράς που ανιχνεύθηκαν επιτυχώς, το πράσινο διακεκομμένο τα παράθυρα του αλγορίθμου μας που αντιστοιχούν σε σωστή ανίχνευση και το κόκκινο τα παράθυρα που απέτυχε να ανιχνεύσει. Ο αριθμός πάνω αριστερά δείχνει την θέση που κατέχει το παράθυρο μετά την βαθμολόγηση από τον αλγόριθμό μας.

3.2 Εναλλακτικές προσεγγίσεις

Στην προσπάθειά μας για επίτευξη μιας μεθόδου που να εμφανίζει πλεονεκτήματα ως προς τις ήδη υπάρχουσες μεθόδους, πραγματοποιήσαμε διάφορες τροποποιήσεις στον αλγόριθμός μας, άλλες με καλά και άλλες με λιγότερο καλά αποτελέσματα. Παρακάτω θα περιγράψουμε αυτές μας τις προσπάθειες.

3.2.1 Βέλτιστο εσωτερικό παράθυρο

Τρέχοντας τον αλγόριθμό μας, παρατηρήσαμε ότι το μεγαλύτερο μέρος του χρόνου που χρειάζεται για την εκτέλεσή του, το καταναλώνει στις πολλαπλές βαθμολογήσεις που κάνει κατά τη διάρκεια της τελειοποίησης. Ωστόσο αυτή η διαδικασία είναι απαραίτητη, καθώς τα αρχικά παράθυρα που επιλέγονται είναι αρκετά αποστασιοποιημένα μεταξύ τους, με αποτέλεσμα να μην πετυχαίνουν ακριβώς τα αντικείμενα που βρίσκονται κοντά σε αυτά. Με την διαδικασία τελειοποίησης, μετακινώντας λίγο το παράθυρο και επαναβαθμολογώντας το, βρίσκουμε την βέλτιστή του θέση ώστε να μεγιστοποιήσουμε την βαθμολογία. Έτσι, το κυρίως μέρος του αλγορίθμου μας επαναλαμβάνεται αρκετές φορές.

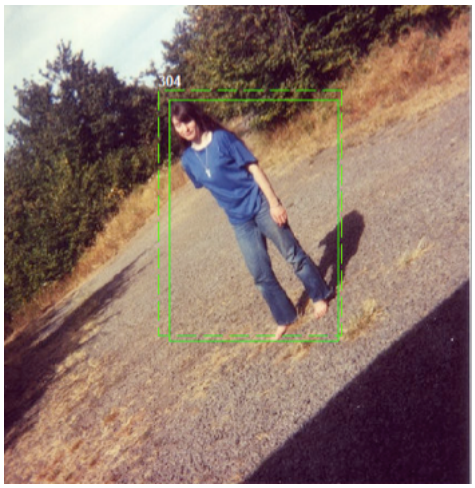
Για να αντιμετωπίσουμε αυτό το πρόβλημα παρατηρήσαμε ότι με τις δομές που έχουμε φτιάξει, και πιο συγκεκριμένα με την δομή disjoint set, μπορούμε να πάρουμε κατευθείαν το βέλτιστο εσωτερικό παράθυρο του παραθύρου που εξετάζουμε, χωρίς μεγάλο επιπρόσθετο χρόνο εκτέλεσης, αφαιρώντας τελείως την διαδικασία της τελειοποίησης.

Αρχικά κατά τον υπολογισμό του εμβαδού του κάθε τμήματος, δημιουργούμε ένα σφιχτό παράθυρο γύρω από το κάθε τμήμα. Για να το κάνουμε αυτό, απλά αποθηκεύουμε την θέση του εικονοστοιχείου που βρίσκεται πιο πάνω, πιο κάτω, δεξιότερα και αριστερότερα στο τμήμα που εξετάζουμε.

Στη συνέχεια, κατά την κατασκευή του disjoint set, όταν ενώνουμε δύο κομμάτια, δημιουργούμε και ένα παράθυρο γύρω από το τμήμα που προκύπτει από την ένωση. Αυτό γίνεται κρατώντας πάλι το δεξιότερο, το αριστερότερο, το πιο πάνω και το πιο κάτω εικονοστοιχείο. Ο υπολογισμός αυτός γίνεται ταχύτατα με τέσσερις απλές συγκρίσεις με βάση τα αρχικά σφιχτά παράθυρα που υπολογίσαμε. Αν συνεχίσουμε να ενώνουμε τμήματα, χρησιμοποιούμε τα νέα παράθυρα που δημιουργήθηκαν κατά την ένωση οπότε η διαδικασία δεν χρειάζεται κάποιον υπολογισμό από την αρχή κάτι που μεταφράζεται επίσης σε υψηλή ταχύτητα εκτέλεσης.

Αφού τελειώσουμε την κατασκευή του disjoint set, μαζί με την βαθμολογία του παραθύρου, έχουμε και το πιο σφιχτό παράθυρο που περικλείει τα τμήματα του παραθύρου μας. Έτσι δεν χρειάζεται περαιτέρω επεξεργασία.

Με αυτή την διαδικασία, έχουμε σημαντική αύξηση της ταχύτητας εκτέλεσης καθώς από 2 δευτερόλεπτα ανά εικόνα, πέφτουμε στα 0.3 δευτερόλεπτα. Επιπλέον έχουμε λιγότερα παράθυρα καθώς είναι πιθανό μερικά να δίνουν καλή βαθμολογία διαφέροντας περισσότερο από 0.75% επικάλυψη, η οποία όμως να οφείλεται στο ίδιο εσωτερικό παράθυρο. Δυστυχώς όμως υπάρχει και ένα ποσοστό στο οποίο δεν συμ-



(α) αρχική υλοποίηση

(β) εμβαδό βέλτιστου εσωτερικού παραθύρου

Σχήμα 3.4: Παράδειγμα προσέγγισης με χρήση εμβαδού βέλτιστου εσωτερικού παραθύρου. Παρατηρούμε ότι η γυναίκα στην αρχική υλοποίηση (α) ανιχνεύεται στο 304*ο* παράθυρο ενώ με το εμβαδόν βέλτιστου εσωτερικού παραθύρου (β) μόλις στο 53*ο*.

βαίνει αυτό με αποτέλεσμα να έχουμε περισσότερες χαμένες ανιχνεύσεις.

3.2.2 Εμβαδόν βέλτιστου εσωτερικού παραθύρου

Μια άλλη τροποποίηση που επιχειρήσαμε να κάνουμε είναι η χρήση του εμβαδού του ίδιου του βέλτιστου εσωτερικού παραθύρου αντί του εμβαδού των τμημάτων που περιέχει. Η ιδέα αυτή βασίζεται στο γεγονός ότι υπάρχουν εικόνες όπως η εικόνα 3.4 στις οποίες το αντικείμενο που μας ενδιαφέρει έχει μια κλίση σε σχέση με τις διαστάσεις της εικόνας. Αυτό έχει σαν συνέπεια αυτά τα αντικείμενα να έχουν μικρότερη πιθανότητα να ανιχνευθούν, αφού λόγω του ότι τα παράθυρα δεν είναι σφιχτά γύρω από αυτά, μετά την κανονικοποίηση με το εμβαδόν του παραθύρου έχουν μικρότερη βαθμολογία απ' ότι αν ήταν παράλληλα με τις διαστάσεις της εικόνας.

Για να υλοποιήσουμε αυτή την ιδέα, χρησιμοποιήσαμε την ιδέα του βέλτιστου εσωτερικού παραθύρου που περιγράψαμε προηγουμένως. Αφού έχουμε τα τέσσερα σημεία του παραθύρου, είναι εύκολο βρούμε το ύψος b_h^{in} και το πλάτος του b_w^{in} και κατά συνέπεια το εμβαδόν του

$$\text{area}(b^{\text{in}}) = b_h^{\text{in}} b_w^{\text{in}} \tag{3.4}$$

Έτσι κατά την βαθμολογία, αντί να κρατάμε το εμβαδόν των τμημάτων που προκύπτουν κάθε φορά, κρατάμε το εμβαδόν του παραθύρου που τα περικλείει

$$\text{score}(b) = \frac{\text{area}(b_{in})}{\text{area}(b)}. \tag{3.5}$$

Αυτό έχει σαν συνέπεια την βελτίωση της ανίχνευσης αντικειμένων όπως η εικόνα 3.4 που είπαμε, αλλά γενικότερα εμφανίζονται και πολλά παράθυρα που δεν περιέχουν αντικείμενα κάτι που συνολικά μειώνει την απόδοση του αλγορίθμου μας.

3.2.3 Τμήματα με βάρη

Καθώς όλα τα τμήματα δεν έχουν την ίδια πιθανότητα να ανήκουν στο ίδιο αντικείμενο, το να παίρνουμε συνδυασμούς τμημάτων και να τους βαθμολογούμε με τον ίδιο τρόπο φαίνεται κάπως αφελές. Κατά την διαδικασία της κατάτμησης, σε ένα τμήμα ήδη βρίσκονται εικονοστοιχεία που έχουν μεγάλη πιθανότητα να ανήκουν στο ίδιο αντικείμενο. Η ιδέα μας εδώ ήταν να συνεχίσουμε κατά κάποιο τρόπο την ένωση που κάνει ο Felzenszwalb [15] στην διαδικασία βαθμολόγησης των παραθύρων μας.

Για αυτό το λόγο, αποφασίσαμε να ορίσουμε ένα βάρος στο κάθε τμήμα, ανάλογα με το οποίο θα συμμετέχει στην διαμόρφωση της βαθμολογίας του παραθύρου. Το βάρος αυτό προκύπτει από την σχέση των εσωτερικών τμημάτων με τα τμήματα που βρίσκονται στο σύνορο του παραθύρου.

Αναλυτικότερα, σαν βάρη χρησιμοποιήσαμε το χαρακτηριστικό του χρώματος, όπως ακριβώς γίνεται και κατά την κατάτμηση από το [15]. Για να το κάνουμε αυτό, αφού παραχθεί η κατάτμηση της εικόνας, στο σημείο που βρίσκουμε τις ακμές ανάμεσα στα τμήματα, κρατάμε για κάθε ακμή και το βάρος που δίνει ο αλγόριθμος [15].

Υπάρχουν πολλές ακμές που ενώνουν δύο τμήματα, αφού ενώνονται παραπάνω από ένα εικονοστοιχεία που βρίσκονται στα δύο τμήματα, οπότε κρατάμε σαν βάρος την μικρότερη ακμή, καθώς αυτή είναι ουσιαστικά αυτή που θα καθόριζε αν θα ενώνονταν τα δύο τμήματα αν συνεχίζαμε τον αλγόριθμο του Felzenszwalb

$$\mathrm{dist}(\alpha, \beta) = \min_{\substack{u_i \in \alpha, u_j \in \beta \\ (u_i, u_j) \in E}} \mathrm{edge}(u_i, u_j), \tag{3.6}$$

όπου α και β τμήματα, dist η απόσταση μεταξύ τμημάτων, u_i και u_j εικονοστοιχεία, E το σύνολο των ακμών και edge το βάρος της ακμής ανάμεσα στο εικονοστοιχεία.

Στη συνέχεια κανονικοποιούμε τις αποστάσεις στην μονάδα, ώστε ο αριθμός που προκύπτει ουσιαστικά να είναι ο βαθμός κατά τον οποίο θα συμμετέχει το κάθε τμήμα στην βαθμολόγηση.

Αφού βρούμε τα τμήματα που τέμνουν το σύνορο του παραθύρου που εξετάζουμε, θέτουμε σαν βάρος $w_{\mathrm{border}} = 0$. Στόχος μας είναι να υπολογίσουμε το βάρος των εσωτερικών τμημάτων w_{inside}.

Όπως και στην αρχική μας προσέγγιση, για να βρούμε τα εσωτερικά τμήματα του παραθύρου, ξεκινώντας από τα τμήματα στο σύνορο κάνουμε αναζήτηση των γειτόνων προς το εσωτερικό του παραθύρου. Ωστόσο, σε αυτή την υλοποίηση, καθώς προχωράμε προς τα γειτονικά, ταυτόχρονα ανανεώνουμε και την τιμή βάρους του κάθε τμήματος θέτοντάς το ίσο με τη απόσταση borderDist από το τμήμα του συνόρου από το οποίο ξεκινήσαμε.

Αν u_i, u_j γειτονικά τότε:

$$\mathrm{borderDist}_j(u_i) = \mathrm{dist}(u_i, u_j), \tag{3.7}$$

αλλιώς:

$$\text{borderDist}_j(u_i) = \max(\text{borderDist}_j(u_k), \text{dist}(u_k, u_i)), \tag{3.8}$$

με j το τμήμα του συνόρου απ' όπου ξεκινάμε, u_i το τμήμα στο οποίο θέλουμε να αποδώσουμε τιμή βάρους και u_k ο προηγούμενος κόμβος από αυτόν που μας ενδιαφέρει πάνω στο μονοπάτι που εξετάζουμε.

Κρατάμε το μέγιστο γιατί μπορεί δύο εσωτερικά τμήματα να διαφέρουν λίγο μεταξύ τους, αλλά εμάς μας ενδιαφέρει πόσο διαφέρουν από το τμήμα συνόρου. Για να γίνει περισσότερο αντιληπτό αυτό, αρκεί να φανταστούμε τρία τμήματα που συνδέονται στην σειρά, εκ των οποίων το ένα είναι πάνω στο όριο του παραθύρου. Αυτό το τμήμα έχει λοιπόν βάρος ίσο με 0. Υπάρχουν δύο περιπτώσεις. Αν $\text{borderDist}_j(u_k) < \text{dist}(u_k, u_i)$, τότε το τμήμα u_i διατηρεί για βάρος τους την $\text{dist}(u_k, u_i)$, αφού το πρώτο ενώνεται κατευθείαν με το συνοριακό τμήμα με μικρό βάρος, ενώ το δεύτερο ενώνεται πρώτα με το πρώτο (με το οποίο έχει μεγάλη διαφορά) και μετά με το συνοριακό. Συνολικά το u_i λοιπόν απέχει πολύ από το σύνορο. Στην άλλη περίπτωση όπου $\text{borderDist}_j(u_k) > \text{dist}(u_k, u_i)$ το u_i αφού διαφέρει λίγο από το u_k το οποίο έχει μεγάλη διαφορά με το συνοριακό, παίρνει την τιμή του $\text{borderDist}_j(u_k)$ καθώς διέρχεται από το u_k για να φτάσει στο σύνορο. Αυτό μπορούμε να το γενικεύσουμε και σε περισσότερα τμήματα.

Αφού όλα τα τμήματα συνδέονται με κάποιο τρόπο μεταξύ τους, προκύπτουν διαφορετικές τιμές βάρους για κάθε τμήμα από κάθε συνοριακό τμήμα. Τελικά θέλουμε να κρατήσουμε την ελάχιστη απόσταση από το σύνορο γενικότερα οπότε:

$$w(u_i) = \min_{j \in S_b} \text{borderDist}_j(u_i), \tag{3.9}$$

όπου $w(u_i)$ το τελικό βάρος του τμήματος και S_b το σύνολο των συνοριακών τμημάτων.

Έτσι σε κάθε παράθυρο βρίσκουμε ένα βάρος για το κάθε τμήμα.

Στη συνέχεια ακολουθούμε δύο διαφορετικές προσεγγίσεις για τον υπολογισμό της βαθμολογίας:

- Στην πρώτη προσέγγισή μας, παίρνουμε ένα ένα τα τμήματα που βρίσκονται εντός του παραθύρου και τα προσθέτουμε στην βαθμολογία του ανάλογα με το βάρος τους. Έτσι κάθε τμήμα συμμετέχει σε διαφορετικό βαθμό στην βαθμολόγηση.

$$\text{score}(b) = \sum_{i \in R} w(u_i)\, \text{area}(u_i), \tag{3.10}$$

- Στην δεύτερη προσέγγιση, ταξινομούμε τα τμήματα εντός του παραθύρου σε φθίνουσα σειρά, έτσι ώστε στην πρώτη θέση να βρίσκεται αυτό με την μεγαλύτερη απόσταση από το σύνορο. Στη συνέχεια παίρνουμε το επόμενο και

αν είναι γειτονικά τα ενώνουμε, αλλιώς το κρατάμε και συνεχίζουμε με το επόμενο. Ενώνουμε λοιπόν μόνο τμήματα που είναι γειτονικά με τμήματα που έχουμε ήδη επισκεφτεί προηγουμένως. Η διαδικασία αυτή συνεχίζεται για όλα τα τμήματα. Όταν ενώνουμε δύο τμήματα, πολλαπλασιάζουμε το εμβαδόν της ένωσής τους με το μικρότερο βάρος των επιμέρους τμημάτων καθώς θέλουμε να έχουμε κάθε φορά την απόσταση του συνολικού τμήματος από το σύνορο

$$\text{score}(b_{\text{in}}) = \min_{i \in R_{\text{in}}} w(u_i) \frac{\sum_{i \in R\text{in}} \text{area}(u_i)}{\text{area}(b_{\text{in}})} \tag{3.11}$$

Κάθε φορά που ενώνουμε ελέγχουμε την τιμή του παραθύρου που προκύπτει και κρατάμε άπληστα το εσωτερικό παράθυρο με την μεγαλύτερη βαθμολογία.

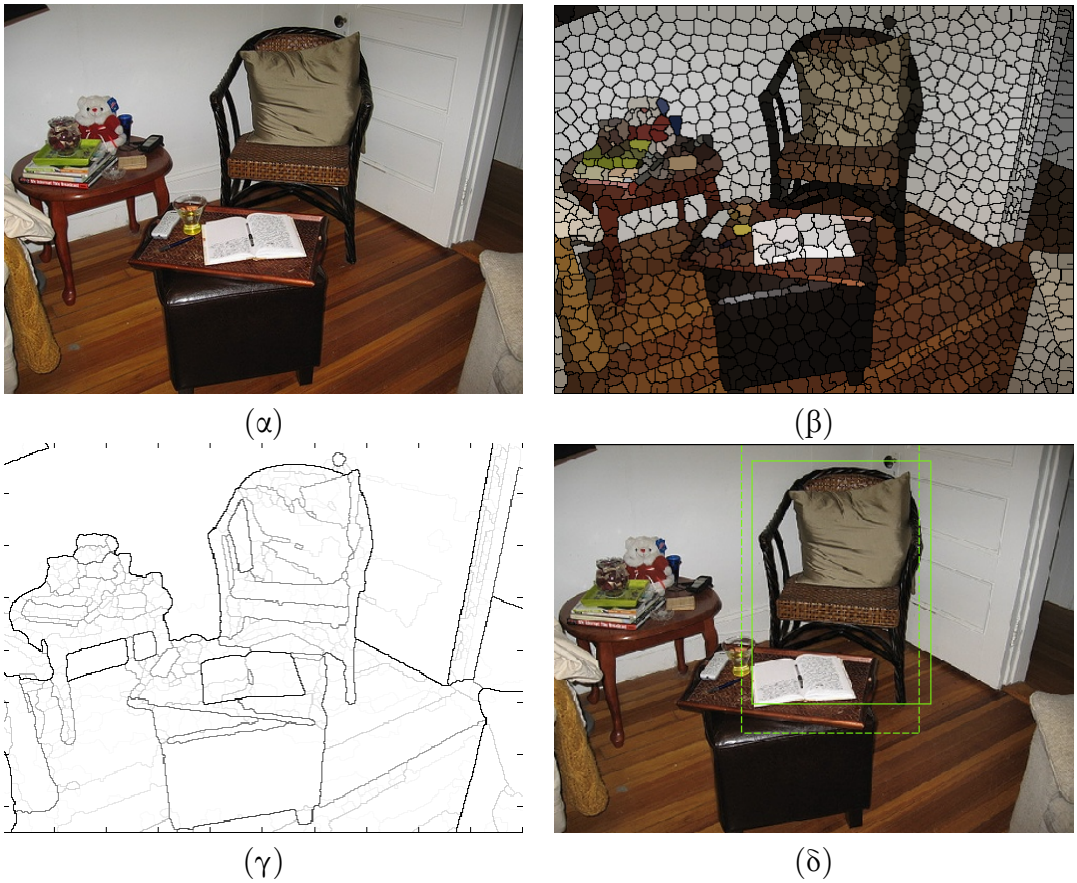
Παρά την φαινομενικά καλή ιδέα, τα αποτελέσματα της εκτέλεσης αυτής της μεθόδου δεν ήταν ιδιαίτερα καλά. Αυτό οφείλεται κυρίως στο γεγονός ότι μικρά μεμονωμένα τμήματα έχουν αυξημένη πιθανότητα να έχουν καλύτερη βαθμολογία αφού όσο ενώνονται τα τμήματα, τόσο μικρότερη η τιμή του βάρους που θα έχει το συνολικό τμήμα. Έτσι, αφού όλα τα τμήματα υφίστανται την ίδια κανονικοποίηση με το εμβαδόν του παραθύρου, δίνεται έμφαση στα μικρά τμήματα που δημιουργούνται κατά την υπερκατάτμηση. Επιπλέον, σημαντικό ρόλο παίζει και το γεγονός ότι η διαφορά χρώματος δεν είναι από μόνη της ιδιαίτερα ενδεικτική για την ύπαρξη διαφορετικού αντικειμένου. Γι'αυτό το λόγο άλλωστε πολλές από τις μεθόδους που εξετάσαμε χρησιμοποιούν μια ποικιλία από χαρακτηριστικά πέρα του χρώματος, όπως υφή, σχήμα και μέγεθος.

3.2.4 Κατάτμηση με ανίχνευση ακμών

Αφού η κατάτμηση είναι το βασικό συστατικό της μεθόδου μας, δεν θα μπορούσαμε παρά να δοκιμάσουμε και μια άλλη κατάτμηση εκτός από αυτή του αλγορίθμου το Felzenszwalb [15]. Για αυτό το λόγο, και επηρεασμένοι από την [23] δοκιμάσαμε να χρησιμοποιήσουμε κατάτμηση από τις ακμές που παράγει ο αλγόριθμος των Dollár και Zitnick [10].

Η συγκεκριμένη κατάτμηση προκύπτει με την χρήση κώδικα που δίνεται από τους συγγραφείς, παρόλο που δεν αναφέρεται και δεν χρησιμοποιείται στην δική τους εργασία. Αυτό που κάνουμε λοιπόν είναι να τεμαχίσουμε την εικόνα σε πολλά περίπου ισομεγέθη τμήματα που ωστόσο διατηρούν τις ακμές των αντικειμένων 3.5(β). Στη συνέχεια ακολουθούμε την ίδια προσέγγιση που ακολουθήσαμε και με τα τμήματα του Felzenszwalb στην βασική μας υλοποίηση, αλλά δοκιμάσαμε και την προσέγγιση με το βέλτιστο εσωτερικό παράθυρο.

Ωστόσο από την δομή των τμημάτων, διαπιστώσαμε ότι από μόνα τους δεν έχουν ιδιαίτερη αξία και γι'αυτό σε δεύτερη φάση τα χρησιμοποιήσαμε με βάρος που δίνεται από τις ακμές που παράγει ο [10]. Με αυτόν τον τρόπο τα τμήματα συμβάλουν με διαφορετικό τρόπο ανάλογα με το αν υπάρχει σύνορο μεταξύ τους. Η τιμή του βάρους ανάμεσα στα τμήματα μπορεί να αναπαρασταθεί σχηματικά στο σχήμα 3.5(γ). Και



Σχήμα 3.5: Κατάτμηση με χρήση ανίχνευσης ακμών. Η αρχική εικόνα (α) υφίσταται κατάτμηση με βάση την ανίχνευση ακμών της [10] και αποδίδονται στα τμήματα βάρη που αντιστοιχούν στην ένταση των ακμών του ίδιου αλγορίθμου (γ). Στη (δ) έχουμε το αποτέλεσμα της εκτέλεσης της μεθόδου μας.

πάλι δοκιμάσαμε και τις δύο προσεγγίσεις με ακμές, με απλή πρόσθεση των τμημάτων ανάλογα με το βάρος που τους αντιστοιχεί και με την χρήση ενιαίου βάρους για όλο το τμήμα που προκύπτει κρατώντας άπληστα το βέλτιστο. Τα αποτελέσματα και πάλι δεν ήταν ιδιαίτερα ελπιδοφόρα όπως θα δούμε και στο κεφάλαιο 4.

Κεφάλαιο 4

Πειράματα

4.1 Υλοποίηση

Η υλοποίηση της μεθόδου μας καθώς και τα πειράματα που εκτελέσαμε έγιναν σε περιβάλλον Microsoft Windows 7 64bit, με την χρήση MATLAB 2010b και ο κώδικάς μας γράφτηκε σε Matlab και C++. Ο μεταγλωττιστής που χρησιμοποιήθηκε είναι ο Microsoft Visual C++ 2010 Express. Όλα τα πειράματα εκτελέστηκαν σε υπολογιστή με Intel(R) Core(TM) i5 CPU K 655 @ 3.20GHz και χωρίς χρήση GPU. Για την αξιολόγηση χρησιμοποιήθηκε το Piotr's Matlab toolbox, η δημοσίευση [20] και ο κώδικας VOCdevkit της βάσης PASCAL. Ο κώδικας της μεθόδου μας είναι διαθέσιμος στο github[1].

4.2 Βάσεις δεδομένων

Οι βάσεις δεδομένων που χρησιμοποιούμε στην εργασία μας και πάνω σε αυτές γίνονται οι μετρήσεις είναι η PASCAL VOC 2007 [13] και ImageNet 2013 [32] καθώς αποτελούν τις δύο από τις πιο ευρέως χρησιμοποιούμενες βάσεις εικόνων του χώρου.

PASCAL Χρησιμοποιούμε όλο το σύνολο των εικόνων της βάσης, η οποία περιλαμβάνει 20 κατηγορίες αντικειμένων σε περίπου 5000 εικόνες. Για την αξιολόγηση, συμπεριλαμβάνουμε όλες τις κατηγορίες και όλα τα παράθυρα αναφοράς, συμπεριλαμβανομένου και των "δύσκολων" που πολλές φορές δεν λαμβάνονται υπόψιν ακριβώς λόγω της μικρής πιθανότητας σωστής ανίχνευσης. Αυτό γίνεται γιατί θέλουμε να μετρήσουμε την μέγιστη δυνατή ανάκληση σε όλα τα αντικείμενα. Αξίζει να σημειωθεί ότι η βάση αυτή περιέχει μόνο συγκεκριμένα αντικείμενα και όχι όλα τα αντικείμενα που μπορεί να περιέχονται στις εικόνες, κάτι που καθιστά αμφισβητήσιμο το αν η ανάκληση των αντικειμένων στην βάση αυτή είναι αρκετά καλή μετρική για τον προσδιορισμό της ποιότητας των αλγορίθμων.

[1] https://github.com/vinPopulaire/ObjectProposals.git

ImageNet Καθώς όπως είπαμε η προηγούμενη βάση δεδομένων περιέχει μόνο 20 κατηγορίες, ενώ η μέθοδός μας όπως και οι υπόλοιπες σύγχρονες μέθοδοι θεωρητικά ανιχνεύουν αντικείμενα ανεξαρτήτως του είδους τους, χρησιμοποιούμε και την βάση ImageNet 2013. Αυτή η βάση περιέχει αντικείμενα από 200 κατηγορίες σε πάνω από 20000 εικόνες. Σημειώνουμε ότι οι κατηγορίες αυτές δεν είναι ειδικές περιπτώσεις των κατηγοριών της PASCAL αλλά καινούργιες όπως διαφορετικά είδη ζώων, τροφές (πχ hot-dogs), αντικείμενα σπιτιού και άλλα. Ωστόσο, παρά την μεγαλύτερη ποικιλία τους, ούτε αυτή είναι πλήρης καθώς και σε αυτήν υπάρχουν αντικείμενα μέσα στις εικόνες που δεν έχουν σημειωθεί.

4.3 Πρωτόκολλο αξιολόγησης

Όπως είπαμε και προηγουμένως, αυτό που είναι σημαντικό στους αλγορίθμους που εξετάζουμε είναι να υπάρχει καλή κάλυψη των αντικειμένων ενδιαφέροντος στην εικόνα που μας ενδιαφέρει, καθώς χαμένα αντικείμενα από αυτούς δεν μπορούν να βρεθούν στην συνέχεια από τους αλγορίθμους ανίχνευσης. Συνεπώς η πιο κοινή πρακτική είναι η αξιολόγηση της ποιότητά τους ανάλογα με την ανάκληση των δεδομένων αναφοράς.

Το πρωτόκολλο παρουσιάστηκε στην εργασία [1] χρησιμοποιώντας τη βάση δεδομένων PASCAL VOC 2007 [13] και χρησιμοποιείται σαν κατευθυντήρια γραμμή για την αξιολόγηση της ποιότητας των περισσοτέρων σύγχρονων αλγορίθμων ανίχνευσης πιθανών θέσεων αντικειμένων.

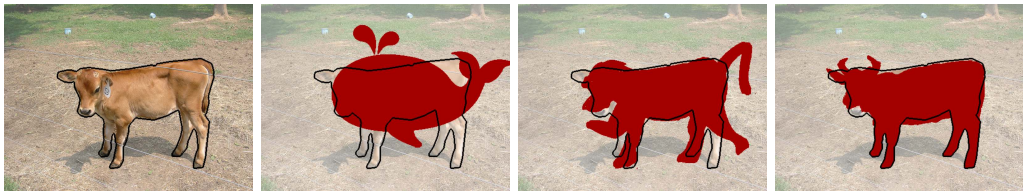
Η αξιολόγηση των υποψήφιων θέσεων είναι αρκετά διαφορετική από την αξιολόγηση της παραδοσιακής ανίχνευσης αντικειμένων καθώς οι περισσότερες μετρικές των τελευταίων δεν έχουν νόημα, όπως η μέση ακρίβεια ανίχνευσης (average precision - AP), η σύγχυση κατηγορίας (class confusion), η σύγχυση φόντου (background confusion) κλπ.

Οι μετρικές που χρησιμοποιούνται εδώ είναι συνήθως συναρτήσεις του IoU (Intersection over Union) ή αλλιώς δείκτης Jaccard μεταξύ των υποψηφίων θέσεων που παράγει ο αλγόριθμος και των πραγματικών θέσεων της βάσης εικόνων. Για δύο τμήματα b_i και b_j μιας εικόνας, το IoU ορίζεται ως:

$$\text{IoU}(b_i, b_j) = \frac{\text{area}(b_i \cap b_j)}{\text{area}(b_i \cup b_j)} \tag{4.1}$$

Για να καταλάβουμε τι αντιπροσωπεύει ποιοτικά κάθε τιμή του IoU μπορούμε να δούμε την εικόνα 4.1.

Με βάση το παραπάνω, οι μετρικές που μας ενδιαφέρουν έχουν να κάνουν με την *ανάκληση* (recall) για κάποιο κατώφλι t του IoU. Με αυτό εννοούμε ότι για κάθε πραγματική θέση της βάσης δεδομένου, κοιτάμε αν το βέλτιστο παράθυρο από άποψη επικάλυψης από αυτά που έχει ανιχνεύσει ο αλγόριθμος έχει IoU μεγαλύτερο από το κατώφλι t. Αν έχει, τότε το παράθυρο αυτό θεωρείται εντοπισμένο.



(a) Ground truth (b) $\mathcal{J} = 0.554$ (c) $\mathcal{J} = 0.703$ (d) $\mathcal{J} = 0.910$

Σχήμα 4.1: Ποιοτική έκφραση των τιμών IoU.

Πιο τυπικά, ορίζουμε την ανάκληση ως

$$\text{Recall} = \frac{\#\text{εντοπισμένα αντικείμενα}}{\#\text{όλα τα αντικείμενα της βάσης}} \tag{4.2}$$

Η υποψήφιες θέσεις λοιπόν αξιολογούνται με βάση δύο γραφικές παραστάσεις:

- την ανάκληση ως προς το IoU, με σταθερό αριθμό υποψήφιων θέσεων
- την ανάκληση ως προς τον αριθμό υποψήφιων θέσεων με μεταβλητό IoU.

Μια άλλη μετρική που προτάθηκε πρόσφατα στην εργασία [20] είναι η *μέση ανάκληση* (Average Recall - AR) κατά την οποία δημιουργείται μια γραφική παράσταση της μέσης ανάκλησης (για IoU από 0.5 μέχρι 1) ως προς τον αριθμό των υποψηφίων θέσεων. Όπως και η AP για την ανίχνευση αντικειμένων, έτσι και η AR συνοψίζει την απόδοση του αλγορίθμου ανίχνευσης υποψήφιων θέσεων μεταξύ κατωφλιών IoU για δεδομένο αριθμό παραθύρων. Η μετρική αυτή φαίνεται να σχετίζεται πιο στενά με την τελική απόδοση ανίχνευσης.

Τέλος, ευρέως χρησιμοποιούμενη είναι και η *μέση καλύτερη επικάλυψη* (Average Best Overlap - ABO), η οποία εξαλείφει την ανάγκη για ύπαρξη κατωφλίου. Αρχικά υπολογίζεται η επικάλυψη μεταξύ των πραγματικών αντικειμένων των παραθύρων αναφοράς $g_i \in G$ και της βέλτιστης υπόθεσης του αλγορίθμου που χρησιμοποιούμε για αυτό το αντικείμενο $l_j \in L$. Η ABO υπολογίζεται ως ο μέσος όρος

$$\text{ABO} = \frac{1}{|G|} \sum_{g_i \in G} \max_{l_j \in L} \text{IoU}(g_i, l_j) \tag{4.3}$$

Η ABO συνήθως υπολογίζεται ανά κατηγορία αντικειμένων. Για τον προσδιορισμό της ABO πάνω σε όλες τις κατηγορίες, χρησιμοποιείται η μετρική MABO (Mean Average Best Overlap).

Στην εργασία μας επιχειρούμε να συγκρίνουμε την ποιότητα της μεθόδου μας σε όλες τις παραπάνω μετρικές με τις υπόλοιπες μεθόδους, για να έχουμε πιο ολοκληρωμένη εικόνα της ανιχνευτικής της ικανότητας.

4.4 Αποτελέσματα

4.4.1 Σύγκριση παραλλαγών της Segment Boxes

Στο παρόν κεφάλαιο θα κάνουμε μια σύγκριση μεταξύ των εναλλακτικών προσεγγίσεων της μεθόδου μας που περιγράψαμε στο κεφάλαιο 3.2 πάνω σε εικόνες της βάσης PASCAL. Αυτό γίνεται με σκοπό να βρούμε ποια από αυτές τις προσεγγίσεις είναι η βέλτιστη ώστε να την συγκρίνουμε με τις σύγχρονες state-of-the-art μεθόδους.

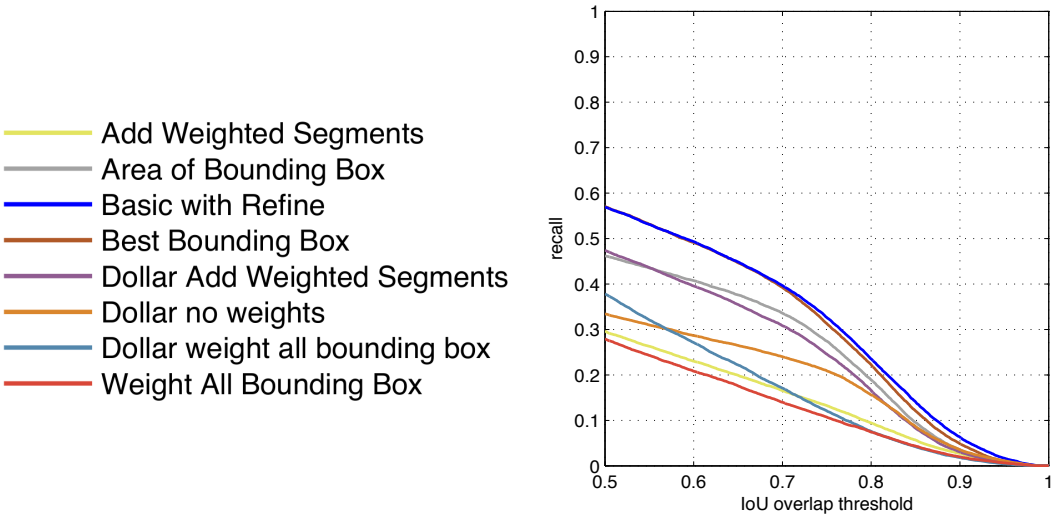
Αρχικά συγκρίνουμε τις παραλλαγές μας όσον αφορά την ανάκληση ως προς την τιμή του IoU για διάφορες τιμές του αριθμού των παραθύρων που επιστρέφει ο αλγόριθμος σαν υποψήφιες (σχήμα 4.2). Σκοπός των μεθόδων ανίχνευσης υποψήφιων θέσεων αντικειμένων είναι να επιτύχουν την μέγιστη δυνατή ανάκληση σε όσο το δυνατόν μικρότερο αριθμό παραθύρων. Συνήθως οι αλγόριθμοι ανίχνευσης αντικειμένων χρησιμοποιούν 1000 υποψήφιες θέσεις καθώς ο χρόνος εκτέλεσής τους καθιστά απαγορευτική την χρήση περισσοτέρων. Παρ'ολα αυτά, σύγχρονοι αλγόριθμοι όπως ο Fast R-CNN [16], καταφέρνει με πολύ μικρό χρόνο εκτέλεσης να έχει πολύ καλή ποιότητα ανίχνευσης. Συνεπώς πλέον και τα 10000 παράθυρα είναι ικανοποιητικός αριθμός. Για τον αλγόριθμο Fast R-CNN θα μιλήσουμε στο κεφάλαιο 5.2.

Στην συνέχεια κάνουμε σύγκριση όσον αφορά την ανάκληση ως προς τον αριθμό των υποψήφιων παραθύρων για κάποια τιμή του IoU (σχήμα 4.3). Γενικά, όσο πιο μεγάλη τιμή έχει το IoU, τόσο πιο δύσκολη είναι η ανίχνευση αλλά πιο ποιοτικά τα αποτελέσματά της (σχήμα 4.1). Σκοπός λοιπόν είναι να έχουμε όσο το δυνατόν καλύτερα αποτελέσματα για μεγαλύτερα IoU. Παρόλα αυτά, οι σύγχρονοι ανιχνευτές αντικειμένων λειτουργούν με IoU $= 0.5$ καθώς για αυτήν την τιμή έχουμε το μεγαλύτερο ποσοστό ανάκλησης. Συνεπώς καλές τιμές ανάκλησης και σε χαμηλότερα IoU είναι επιθυμητές.

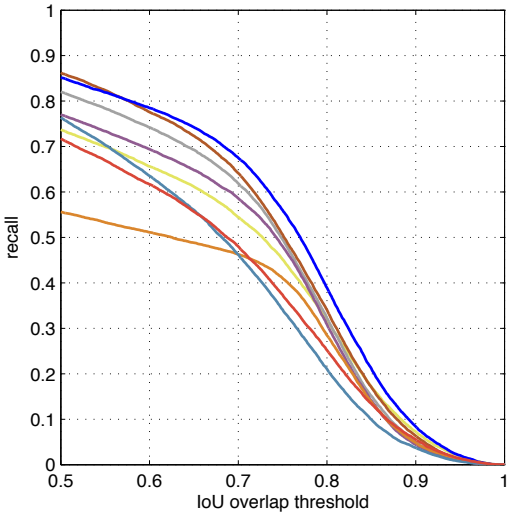
Από τα σχήματα μπορούμε να δούμε εύκολα ότι παρόλο που οι ιδέες μας θεωρητικά φαίνεται να έχουν αξία, πρακτικά οδηγούν σε χειρότερα αποτελέσματα. Κάθε καμπύλη βρίσκεται κάτω από την καμπύλη της βασικής μας υλοποίησης σχεδόν σε όλες τις γραφικές, ειδικά για μεγάλες τιμές του IoU όπως φαίνεται και στα σχήματα 4.3(β,γ,δ). Μόνο για IoU $= 0.5$ βλέπουμε ότι η προσέγγιση με το βέλτιστο εσωτερικό παράθυρο δίνει λίγο καλύτερα αποτελέσματα ενώ για 0.6 ακολουθούν την ίδια πορεία. Αυτό είναι καλό καθώς ο χρόνος εκτέλεσης είναι αρκετά διαφορετικός (2 δευτερόλεπτα η βασική προσέγγιση, 0.3 δευτερόλεπτα η προσέγγιση με βέλτιστο εσωτερικό παράθυρο).

Επιπλέον στο σχήμα 4.2(γ) βλέπουμε ότι για μεγάλο αριθμό υποψήφιων παραθύρων η μέθοδος με τα βάρη παρουσιάζει καλύτερη ανάκληση από τις υπόλοιπες προσεγγίσεις. Ωστόσο, η μέθοδος αυτή, εκτός από αργή, έχει πολύ μικρή αποτελεσματικότητα σε λιγότερο αριθμό παραθύρων με αποτέλεσμα να μην την θεωρούμε συγκρίσιμη σε ποιότητα με την βασική.

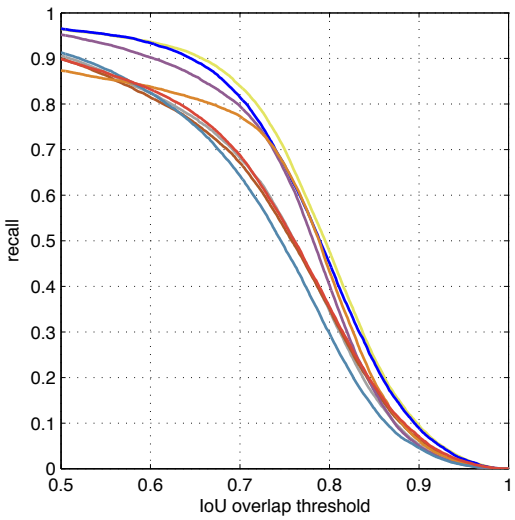
Στην συνέχεια, για την σύγκριση με τις άλλες μεθόδους θα χρησιμοποιήσουμε κυρίως την βασική υλοποίηση, ενώ θα παρουσιάσουμε και μερικές συγκρίσεις με την προσέγγιση βέλτιστου εσωτερικού παραθύρου.



(α) 100 υποψήφια παράθυρα

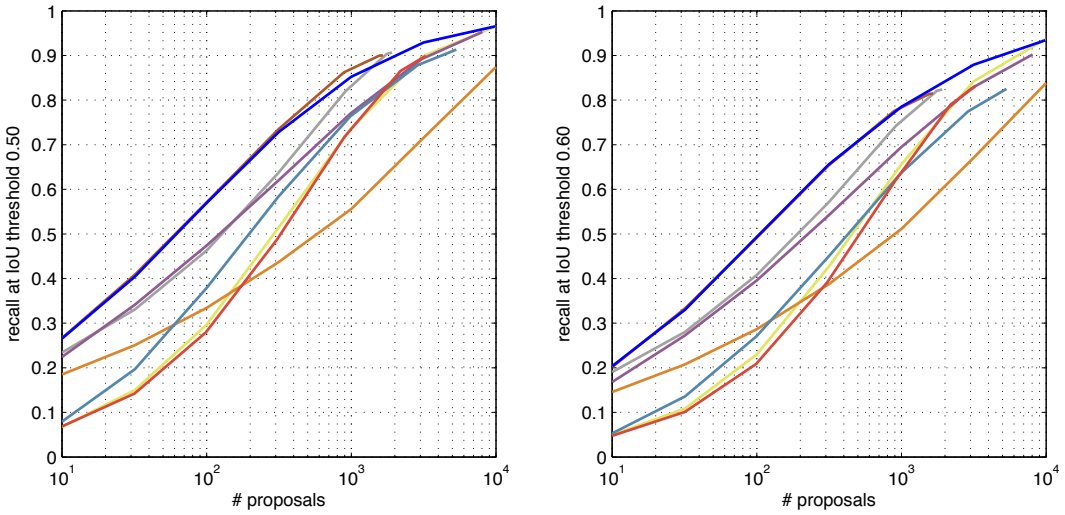


(β) 1000 υποψήφια παράθυρα



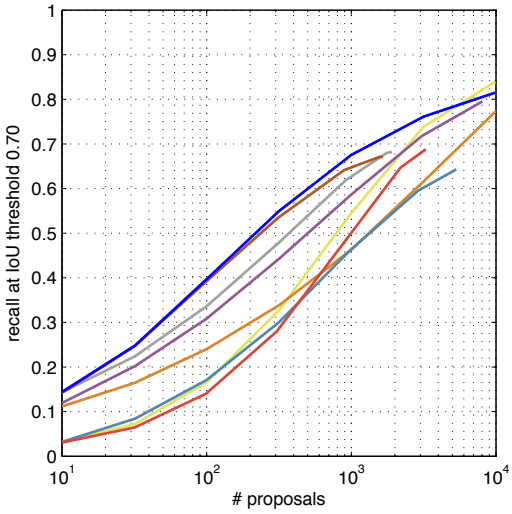
(γ) 10000 υποψήφια παράθυρα

Σχήμα 4.2: Ανάκληση των παραλλαγών της Segment Boxes ως προς το IoU στην βάση PASCAL VOC 2007.

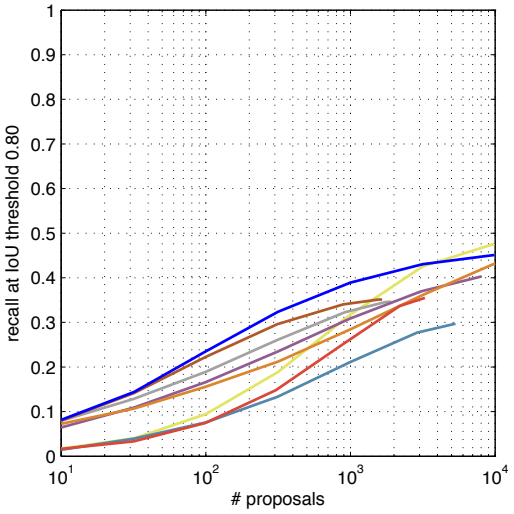


(α) 0.5 IoU

(β) 0.6 IoU



(γ) 0.7 IoU



(δ) 0.8 IoU

Σχήμα 4.3: Ανάκληση των παραλλαγών της Segment Boxes ως προς τον αριθμό των υποψήφιων παραθύρων στην βάση PASCAL VOC 2007.

4.4.2 Σύγκριση με τις άλλες μεθόδους

Τα αποτελέσματα των πειραμάτων όσον αφορά την ανάκληση μπορούμε να τα δούμε στα σχήματα 4.4 και 4.5 όπου βλέπουμε την ανάκληση σαν συνάρτηση του αριθμού των υποψήφιων θέσεων και του IoU αντίστοιχα. Να σημειώσουμε ότι στα πειράματα αυτά χρησιμοποιήσαμε την αρχική υλοποίηση η οποία δίνει και τα συνολικά καλύτερα αποτελέσματα.

Από τα σχήματα αυτά μπορούμε να επιβεβαιώσουμε ότι σε διαφορετικές τιμές των μετρικών, ξεχωρίζουν διαφορετικές μέθοδοι. Για παράδειγμα, για λίγο παράθυρα βλέπουμε ότι η *MCG* είναι η αποτελεσματικότερη ενώ για πολλά παράθυρα, στις υψηλές τιμές IoU, η *Selective Search* ξεχωρίζει με διαφορά. Η δική μας μέθοδος έχει την καλύτερη ανάκληση από όλες τις state-of-the-art μεθόδους για IoU < 0.65 για 10000 υποψήφια παράθυρα, ενώ και για λιγότερα παραμένει ανταγωνίσιμη.

Εξίσου καλά αποτελέσματα έχουμε στην την βάση ImageNet 2013 όπως φαίνεται στο σχήμα 4.6. Παρατηρούμε ότι τα αποτελέσματα και στις δύο βάσεις είναι παρόμοια παρόλο που η ImageNet περιέχει 200 κατηγορίες αντικειμένων, πολλές από τις οποίες δεν έχουν σχέση με τις 20 της PASCAL. Αυτό σημαίνει ότι οι μέθοδοι δεν έχουν υποστεί υπερεκμάθηση πάνω στα δεδομένα της PASCAL και άρα μπορεί να θεωρηθεί ότι όντως μετράνε το κατά πόσο σε μια θέση βρίσκεται ένα αντικείμενο ανεξάρτητα από το είδος του.

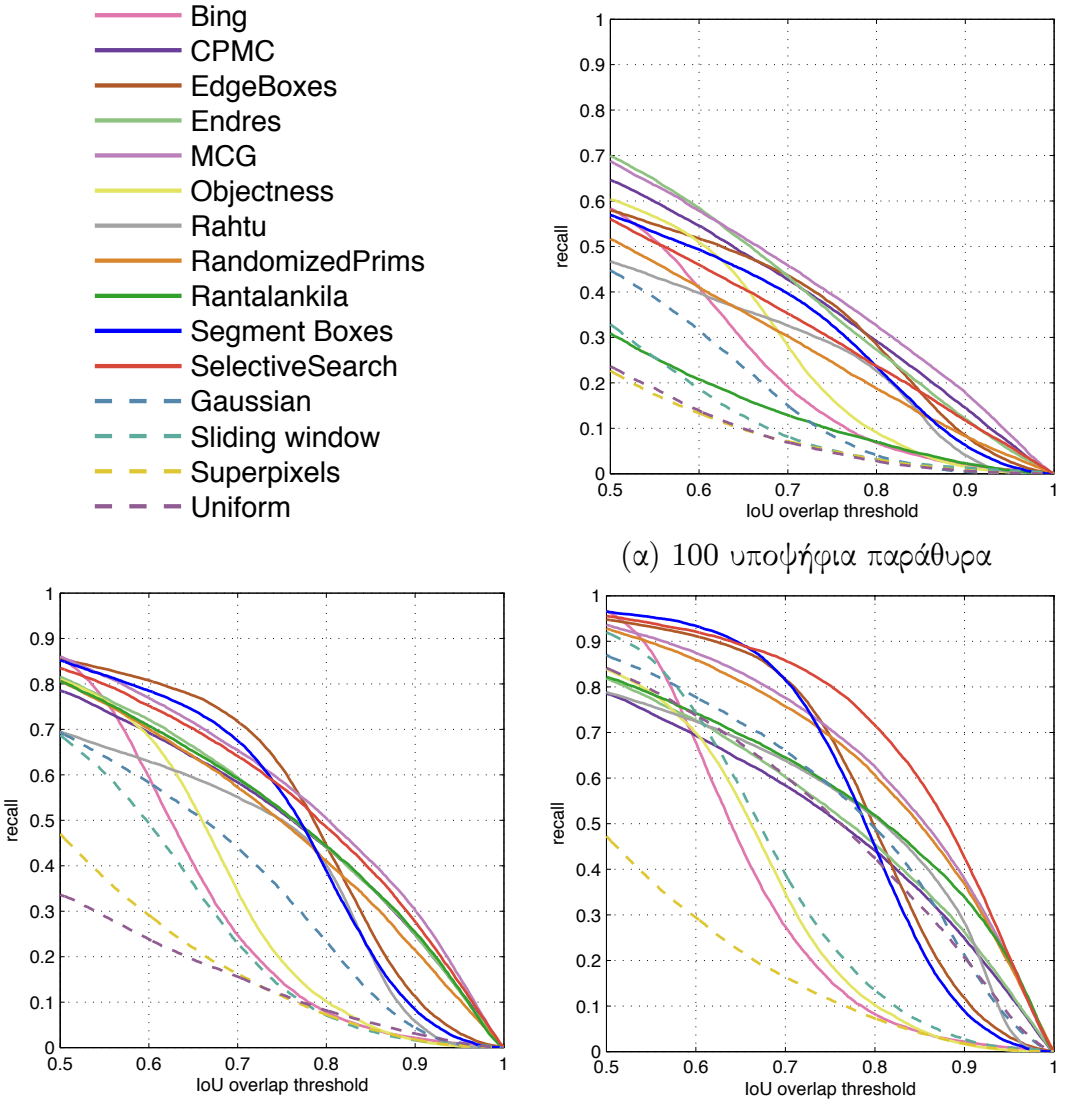
Παρατηρούμε επίσης ότι η καμπύλη μας ακολουθεί σε σχήμα την καμπύλη της *Edge Boxes*. Αυτό οφείλεται στην παρόμοια επιλογή αρχικών παραθύρων για βαθμολόγηση. Παρόλα αυτά, η δική μας μέθοδος επιτυγχάνει μεγαλύτερη ανάκληση σε μερικές περιπτώσεις καθώς η υπερκατάτμηση που κάνουμε κρατάει περισσότερα αντικείμενα που χάνει η προσέγγιση με τις ακμές. Ωστόσο σε λιγότερα παράθυρα η EdgeBoxes έχει καλύτερα αποτελέσματα, πιθανώς γιατί η ακμές περιέχουν περισσότερη και πιο συμπιεσμένη πληροφορία από τα τμήματα.

Στο σχήμα 4.7 συγκρίνουμε την μέθοδό μας με τις υπόλοιπες ως προς την τιμή της μέσης ανάκλησης (AR). Σε αυτή τη μετρική, η μέθοδός μας δεν έχει ιδιαίτερα καλά αποτελέσματα κυρίως λόγω της χαμηλής ανάκλησης σε μεγάλες τιμές του IoU όπως φαίνεται στο σχήμα 4.5(δ).

Στη συνέχεια συγκρίνουμε την μέθοδό μας με με κάποιες από τις μεθόδους που προαναφέραμε ως προς την μετρική MABO (σχήμα 4.8). Για τον υπολογισμό της παίρνουμε τις τιμές της ABO για όλες τις κατηγορίες και βρίσκουμε το μέσο όρο τους ως προς τον αριθμό των κατηγοριών.

Από το σχήμα 4.8 βλέπουμε ότι η μέθοδός μας έχει αρκετά καλή συμπεριφορά και σε αυτή τη μετρική, κυρίως για μικρά IoU όπου είναι εμφανώς καλύτερη από την state-of-the-art *Selective Search*. Για μεγαλύτερα IoU δεν εμφανίζει σημαντική πτώση στην απόδοση και παραμένει καλύτερη από την *Rantalankila* και την *Objectness*.

Όσον αφορά την ταχύτητα, οι διάφορες μέθοδοι έχουν αρκετά διαφορετικό χρόνο εκτέλεσης (πίνακας 4.1) και δεν αρκεί να δούμε μόνο την ανάκληση σαν μέτρο ποιότητας του αλγορίθμου. Να τονίσουμε ότι η ανίχνευση υποψήφιων θέσεων αντικειμένων είναι προπαρασκευαστικό στάδιο για την χρήση πιο ακριβών και εξεζητημένων αλγορίθμων για την ανίχνευση αντικειμένων. Εξ' ορισμού λοιπόν πρέπει να είναι γρήγοροι

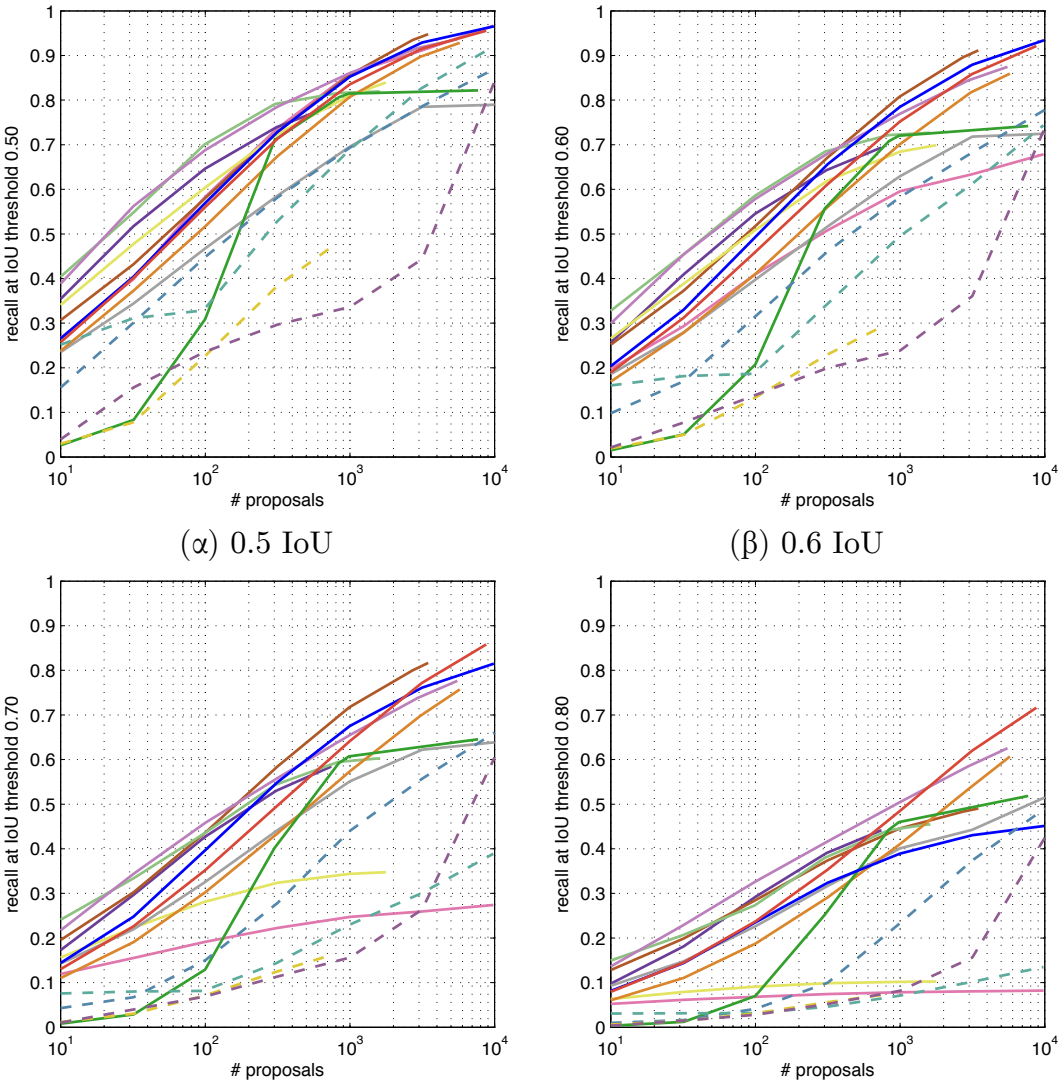


(α) 100 υποψήφια παράθυρα

(β) 1000 υποψήφια παράθυρα

(γ) 10000 υποψήφια παράθυρα

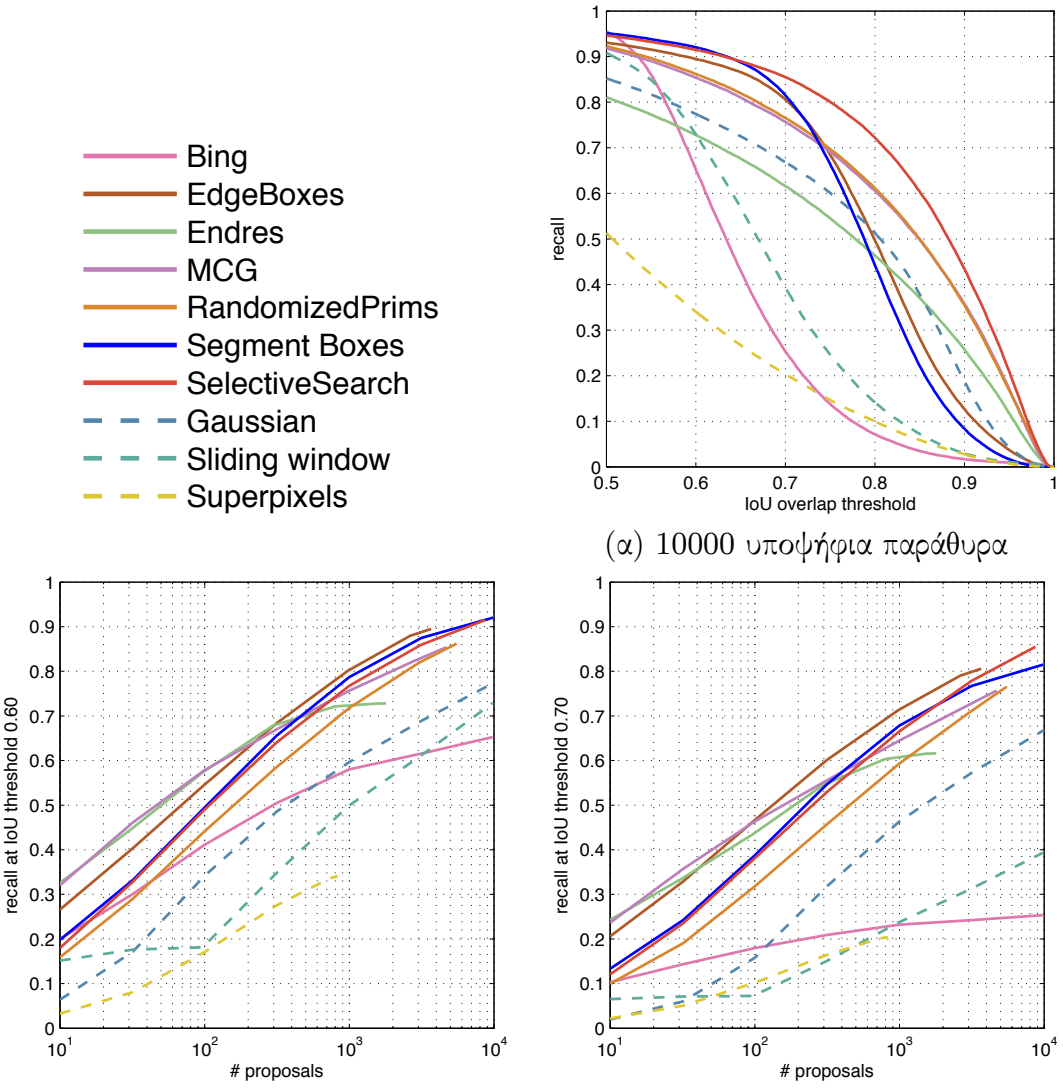
Σχήμα 4.4: Ανάκληση ως προς το IoU στην βάση PASCAL VOC 2007.



(α) 0.5 IoU (β) 0.6 IoU

(γ) 0.7 IoU (δ) 0.8 IoU

Σχήμα 4.5: Ανάκληση ως προς τον αριθμό των υποψήφιων παραθύρων στην βάση PASCAL VOC 2007.

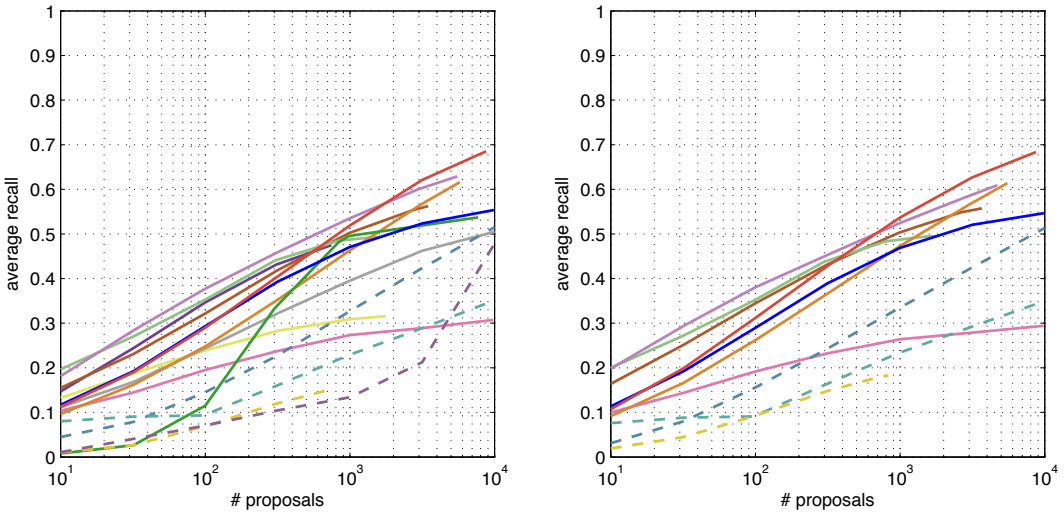


(α) 10000 υποψήφια παράθυρα

(β) 0.6 IoU

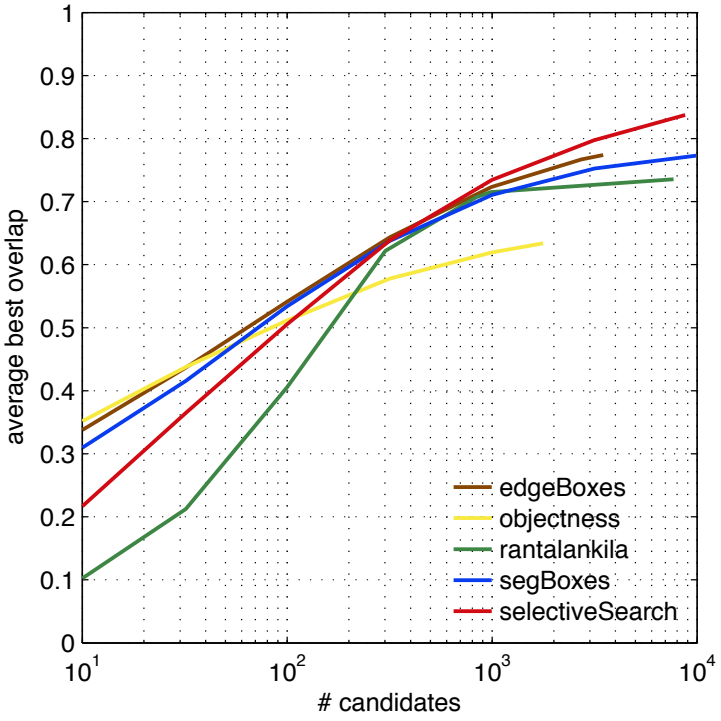
(γ) 0.7 IoU

Σχήμα 4.6: Αποτελέσματα ανάκλησης στην βάση ImageNet 2013.



(α) PASCAL (β) ImageNet

Σχήμα 4.7: Αποτελέσματα μέσης ανάκλησης στις δύο βάσεις εικόνων



Σχήμα 4.8: Αποτελέσματα μέσης βέλτιστης επικάλυψης (ABO) για την βάση PASCAL

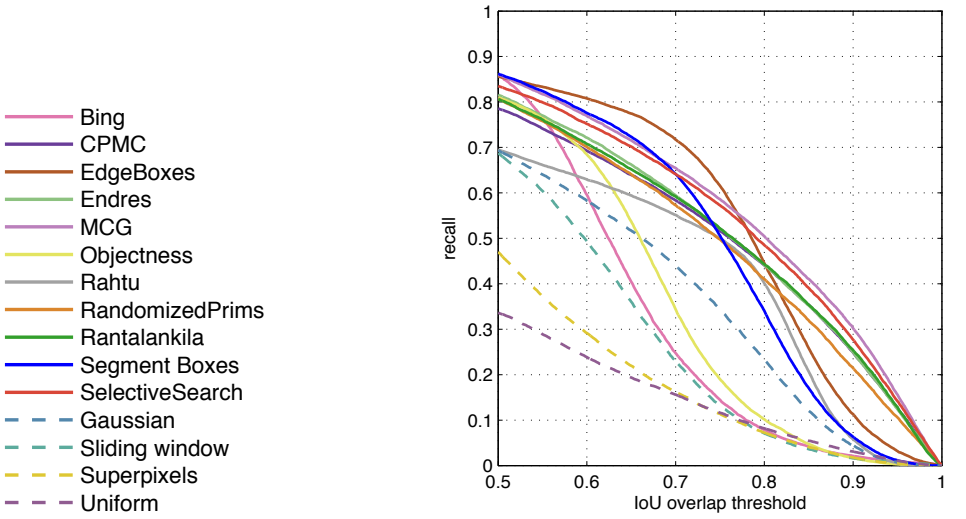
Μέθοδος	Χρήση	Προσέγγιση	Χρόνος (sec)
Bing [7]		ΒΠ	0.2
CPMC [6]	ΚΓ	Ο	250
Edge Boxes [41]	Α	ΒΠ	0.3
Endres [11]	ΚΓ	Ο	100
MCG [4]	Α	Ο	30
Objectness [2]		ΒΠ	3
Rahtu [28]		ΒΠ	3
Randomized Prim's [26]	SP	Ο	1
Rantalankila [29]	SP	Ο	10
Selective Search [34]	SP	Ο	10
Segment Boxes (με refine)	SP	ΒΠ	2
Segment Boxes (με best bounding box)	SP	ΒΠ	0.3
Gaussian			0
SlidingWindow			0
Superpixels			1
Uniform			0

Πίνακας 4.1: Χρόνος εκτέλεσης αλγορίθμων υπολογισμού υποψήφιων θέσεων αντικειμένων.
SP: Superpixels , Α: Ακμές , ΚΓ: Κατάτμηση Γράφων , ΒΠ: Βαθμολόγηση Παραθύρου , Ο: Ομαδοποίηση.

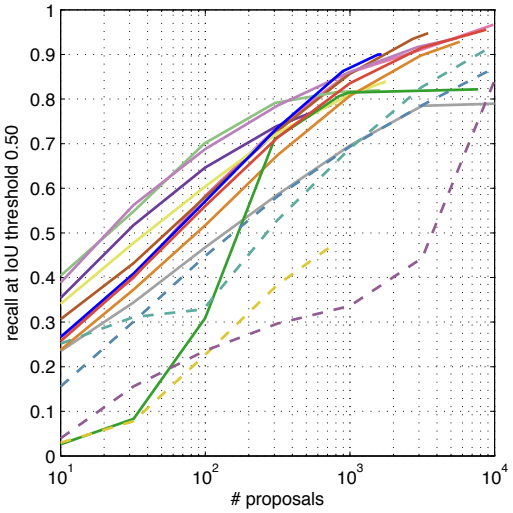
για έχουν νόημα ύπαρξης.

Όπως βλέπουμε στον πίνακα 4.1 οι μέθοδοι βαθμολόγησης παραθύρου τείνουν να είναι πιο γρήγοροι. Η δική μας μέθοδος είναι από τις πιο γρήγορες με χρόνο εκτέλεσης μόλις 2 δευτερόλεπτα ανά εικόνα, ενώ όπως είπαμε καταφέραμε να ρίξουμε τον χρόνο ακόμα περισσότερο με την προσέγγιση εσωτερικού παραθύρου (κεφάλαιο 3.2.1). Ακολουθώντας αυτή την προσέγγιση, ο χρόνος εκτέλεσης φτάνει στα 0.3 δευτερόλεπτα ανά εικόνα, καθιστώντας τον αρκετά γρήγορο και για εφαρμογές πραγματικού χρόνου όπως βίντεο.

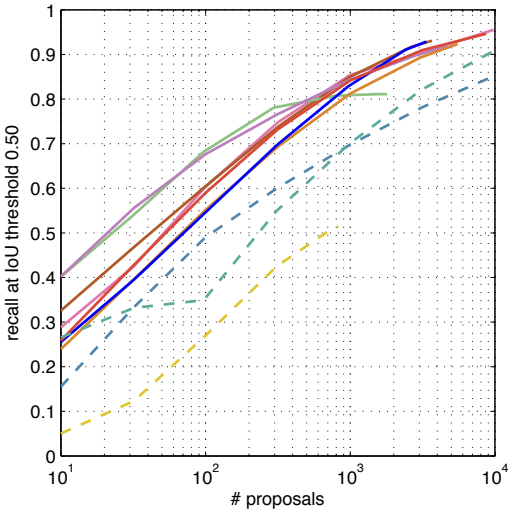
Η προσέγγιση αυτή παρουσιάζει ελαφρώς μειωμένη ποιότητα αποτελεσμάτων αλλά επιτυγχάνει βέλτιστη ανάκληση για IoU $= 0.5$ με 1000 υποψήφια παράθυρα. Στις γραφικές του σχήματος 4.9 μπορούμε να δούμε αναλυτικότερα τα αποτελέσματα.



(α) 1000 παράθυρα στην βάση PASCAL



(β) 0.5 IoU στην βάση PASCAL



(β) 0.5 IoU στην βάση ImageNet

Σχήμα 4.9: Ανάκληση για την προσέγγιση με βέλτιστο εσωτερικό παράθυρο. Σε αυτήν δεν χρησιμοποιείται διαδικασία τελειοποίησης και ο χρόνος εκτέλεσης πέφτει στα 0.3 δευτερόλεπτα ανά εικόνα.

Κεφάλαιο 5

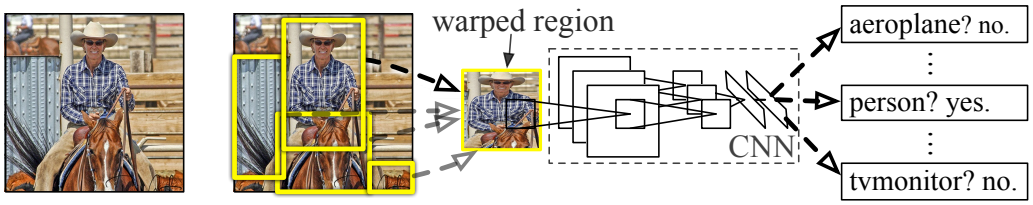
Ανίχνευση Αντικειμένων

Όπως είπαμε και προηγουμένως, οι σύγχρονοι ανιχνευτές αντικειμένων χρησιμοποιούν τις υποψήφιες θέσεις αντικειμένων σαν προπαρασκευαστικό στάδιο στην ανίχνευση, έτσι ώστε να μπορούν να τρέξουν σε εύλογο χρονικό διάστημα. Είναι λοιπόν σημαντικό να δούμε την απόδοση ενός τέτοιου ανιχνευτή χρησιμοποιώντας τις υποψήφιες θέσεις που προκύπτουν από την δική μας μέθοδο, έτσι ώστε να έχουμε καλύτερη εικόνα της ποιότητας των αποτελεσμάτων μας.

Οι παλιότεροι ανιχνευτές αντικειμένων βασίζονται σε χαρακτηριστικά από τον SIFT μετασχηματισμό ή από τα ιστογράμματα προσανατολισμένης κλίσης (Histogram of oriented gradient - HOG) όπως οι [34] και [9]. Ωστόσο, η απόδοση των εν λόγω αλγορίθμων τα τελευταία χρόνια έχει μείνει στάσιμη οπότε και προτάθηκαν νέες εναλλακτικές. Σήμερα, η προσέγγιση που κερδίζει έδαφος ολοένα και περισσότερο είναι η χρήση χαρακτηριστικών που προκύπτουν μετά από εκπαίδευση συνελικτικών νευρωνικών δικτύων (Convolutional Neural Network - CNN).

Δουλειές που βασίζονται στα CNN υπήρχαν ήδη από το 1990 όπως η [25], αλλά η χρήση τους περιορίστηκε λόγω της ανόδου των SVM. Το 2012, επανήλθε το ενδιαφέρον για το CNN χάρη στους Krizhevsky et al [24] οι οποίοι πέτυχαν σημαντικά μεγαλύτερη ποιότητα ανίχνευσης από τους υπόλοιπους στον διαγωνισμό ImageNet (ILSVRC). Μετά από αυτήν την επιτυχία έχουν προταθεί ακόμα καλύτεροι αλγόριθμοι που χρησιμοποιούν CNN όπως ο R-CNN [17], ο SPPnet [18] και ο Fast R-CNN [16]. Και οι τρεις αυτοί αλγόριθμοι χρησιμοποιούν την Caffe [21] υλοποίηση του CNN που περιγράφηκε στην [24].

Στην εργασία μας θα χρησιμοποιήσουμε τον ανιχνευτή Fast R-CNN [16] του Ross Girshick ο οποίος επιτυγχάνει πολύ καλή επίδοση στις βάσεις που εξετάζουμε και θα συγκρίνουμε τα αποτελέσματά της εκτέλεσής του με τη χρήση της μεθόδου μας και των άλλων state-of-the-art μεθόδων.



Σχήμα 5.1: R-CNN. Από αριστερά προς τα δεξιά, το σύστημα παίρνει μια εικόνα εισόδου, εξάγει τις υποψήφιες θέσεις αντικειμένων, υπολογίζει τα χαρακτηριστικά για κάθε υποψήφια θέση χρησιμοποιώντας ένα μεγάλο CNN και τέλος ταξινομεί κάθε υποψήφια θέση σε κλάσεις χρησιμοποιώντας εξειδικευμένα σε κάθε κλάση γραμμικά SVM.

5.1 Μέθοδος R-CNN

Για να καταλάβουμε πώς δουλεύει ο Fast R-CNN, πρέπει πρώτα να δούμε τον προκάτοχό του, R-CNN [17]. Ο συγκεκριμένος ανιχνευτής προτάθηκε το 2014 από τους R. Girshick et al και βασίζεται χαρακτηριστικά που προκύπτουν μετά από εκπαίδευση ενός CNN. Από αυτό προκύπτει άλλωστε και το όνομά του R-CNN: Regions with CNN features (σχήμα 5.1).

Η μέθοδος που ακολουθείται μπορεί να χωριστεί σε 3 κομμάτια. Αρχικά παράγονται οι υποψήφιες θέσεις αντικειμένων που ορίζουν τις θέσεις στις οποίες θα τρέξει το κυρίως μέρος του αλγορίθμου του ανιχνευτή. Το δεύτερο κομμάτι αποτελείται από ένα μεγάλο συνελικτικό νευρωνικό δίκτυο που εξάγει ένα σταθερού μεγέθους διάνυσμα χαρακτηριστικών για κάθε περιοχή. Το τρίτο κομμάτι είναι ένα σύνολο από γραμμικές μηχανές υποστήριξης διανυσμάτων (Support Vector Machine - SVM), εξειδικευμένες σε κάθε κλάση.

Οι υποψήφιες θέσεις εξάγονται με κάποια από τις μεθόδους που αναφέραμε πριν, συμπεριλαμβανομένου της δικής μας. Στο συγκεκριμένο σημείο της εκτέλεσης του αλγορίθμου θα κληθούμε να ενσωματώσουμε την μέθοδό μας για να ελέγξουμε την αποτελεσματικότητά της.

Όσον αφορά τα χαρακτηριστικών, εξάγεται ένα διάνυσμα χαρακτηριστικών 4096 διαστάσεων για κάθε υποψήφια περιοχή. Τα χαρακτηριστικά υπολογίζονται με μια προς τα εμπρός διάδοση στο νευρωνικό μιας 227×227 RGB εικόνας μέσα από πέντε συνελικτικά επίπεδα και δύο πλήρως συνδεδεμένα επίπεδα.

Για να υπολογιστούν τα χαρακτηριστικά για μια υποψήφια περιοχή, πρέπει πρώτα να μετατραπούν τα δεδομένα της περιοχής σε μορφή που είναι συμβατή με το νευρωνικό (η αρχιτεκτονική απαιτεί εισόδους σταθερού 227×227 μεγέθους). Για να το πετύχουν αυτό, οι συγγραφείς επέλεξαν να προσαρμόσουν τα εικονοστοιχεία μέσα στο υποψήφιο παράθυρο στο συγκεκριμένο μέγεθος. Πριν την προσαρμογή, επεκτείνουν το παράθυρο κατά 16 εικονοστοιχεία από όλες τις πλευρές ώστε να συμπεριλάβουν και λίγα "συμφραζόμενα" για το κομμάτι της εικόνας καθώς όπως έδειξαν, με αυτόν τον τρόπο επιτυγχάνεται καλύτερης ποιότητας ανίχνευση.

Εκπαίδευση

Αρχικά γίνεται μια προ-εκπαίδευση του CNN σε μια μεγάλη βοηθητική βάση (ILSVRC2012) χρησιμοποιώντας κατηγοριοποίηση σε επίπεδο εικόνας καθώς δεν υπάρχουν δεδομένα σε επίπεδο αντικειμένων για την συγκεκριμένη βάση.

Για να γίνει η προσαρμογή του CNN στην ανίχνευση αντικειμένων στο νέο είδος εικόνων (προσαρμοσμένα υποψήφια παράθυρα), συνεχίζεται η εκπαίδευση των παραμέτρων του CNN με στοχαστική μέθοδο της πιο απότομης κλίσης (Stochastic Gradient Descent - SGD) χρησιμοποιώντας μόνο τα προσαρμοσμένα υποψήφια παράθυρα. Η αρχιτεκτονική του CNN παραμένει ίδια σε όλα τα σημεία της εκτός από την αντικατάσταση του 1000-way επιπέδου ταξινόμησης για την ImageNet βάση, με ένα τυχαία αρχικοποιημένο 21-way επίπεδο ταξινόμησης της PASCAL (20 για τις κλάσεις + 1 για το φόντο).

Τέλος, αφού τα χαρακτηριστικά έχουν εξαχθεί, βελτιστοποιείται ένα γραμμικό SVM για κάθε κλάση και εκπαιδεύονται οι bounding-box regressor για την καλύτερη εύρεση της θέσης των αντικειμένων στην εικόνα.

Εκτέλεση ανίχνευσης

Κατά την διάρκεια της εκτέλεσης της ανίχνευσης, εξάγονται περίπου 2000 υποψήφιες θέσεις, μετατρέπονται στο μέγεθος που χρειάζεται και πραγματοποιείται η προς τα εμπρός διάδοση στο CNN για τον υπολογισμό των χαρακτηριστικών. Μετά, για κάθε κλάση αντικειμένων, βαθμολογείται κάθε διάνυσμα χαρακτηριστικών με ένα SVM εκπαιδευμένο για κάθε κλάση. Δεδομένων όλων των βαθμολογιών των υποψήφιων θέσεων, εφαρμόζεται ένα άπληστο nms ανεξάρτητα για κάθε κλάση, που απορρίπτει περιοχές που έχουν IoU επικάλυψη μεγαλύτερη κατά ένα κατώφλι με κάποια περιοχή με μεγαλύτερη βαθμολογία. Το κατώφλι αυτό υπολογίζεται με εκπαίδευση.

Δυό είναι οι λόγοι που καθιστούν την ανίχνευση αποδοτική. Πρώτον, όλοι οι παράμετροι του CNN μοιράζονται μεταξύ των κλάσεων και δεύτερον, τα διανύσματα χαρακτηριστικών που υπολογίζονται από το CNN είναι λίγων διαστάσεων αν συγκριθούν με άλλες συχνά χρησιμοποιούμενες προσεγγίσεις. Αυτό έχει σαν αποτέλεσμα ο χρόνος που χρειάζεται να υπολογιστούν οι υποψήφιες θέσεις και τα χαρακτηριστικά (13 δευτερόλεπτα ανά εικόνα σε GPU) να διαμοιράζεται σε όλες τις κλάσεις. Η μόνη εργασία που απευθύνεται σε συγκεκριμένη κλάση είναι ο πολλαπλασιασμός του διανύσματος χαρακτηριστικών με τα βάρη του SVM και το nms. Έτσι ο αλγόριθμος μπορεί να εκτελεστεί χωρίς μεγάλη απώλεια χρόνου και για παραπάνω από τις 20 κλάσεις της βάσης PASCAL.

Με την χρήση της Selective Search για τις υποψήφιες θέσεις, οι συγγραφείς αναφέρουν αποτελέσματα της τάξης του 53.7 mAP στην βάση VOC2010.

Μειονεκτήματα

Ο R-CNN παρόλο που έχει αρκετά καλή ποιότητα ανίχνευσης, εμφανίζει τρία σημαντικά προβλήματα.

Αρχικά, η εκπαίδευση γίνεται σε πολλά στάδια. Στο πρώτο στάδιο προσαρμόζεται το νευρωνικό για ανίχνευση στα δεδομένα του PASCAL. Στην συνέχεια, τα γραμμικά SVM προσαρμόζονται σε χαρακτηριστικά του νευρωνικού που υπολογίζονται από

τις μετασχηματισμένες στο κατάλληλο μέγεθος υποψήφιες θέσεις. Στο τρίτο στάδιο, εκπαιδεύονται οι bounding-box regressors.

Επιπλέον, η ανίχνευση είναι ακριβή από άποψη χώρου και χρόνου. Για την εκπαίδευση των SVM και των regressors, τα χαρακτηριστικά εξάγονται από το κάθε μετασχηματισμένο παράθυρο υποψήφιας θέσης σε κάθε εικόνα και εγγράφονται στον δίσκο. Με την χρήση μεγάλων βαθιών δικτύων (deep networks), όπως του VGG16, η διαδικασία παίρνει 2.5 μέρες GPU για τις 5 χιλιάδες εικόνες του VOC07 και απαιτούνται εκατοντάδες gigabytes αποθηκευτικού χώρου.

Ωστόσο και ο χρόνος ανίχνευσης είναι μεγάλος. Κατά την εκτέλεση της ανίχνευσης, τα χαρακτηριστικά εξάγονται από κάθε μετασχηματισμένη υποψήφια θέση της εικόνας με αποτέλεσμα η ανίχνευση με VGG16 να παίρνει 47 δευτερόλεπτα ανά εικόνα.

Ο R-CNN είναι αργός γιατί μετασχηματίζει τις υποψήφιες θέσεις και μετά τρέχει κάθε υποψήφια θέση ξεχωριστά. Για την επιτάχυνση του R-CNN προτάθηκε ο SPP-net [18]. Ωστόσο και αυτός απαιτεί εκπαίδευση πολλών σταδίων αντίστοιχα με τον προηγούμενο, τα δεδομένα γράφονται και πάλι στον δίσκο, και επιπλέον αντίθετα με τον R-CNN, ο αλγόριθμος για την προσαρμογή του δικτύου μπορεί να ενημερώσει μόνο τα πλήρως συνδεδεμένα επίπεδα του νευρωνικού.

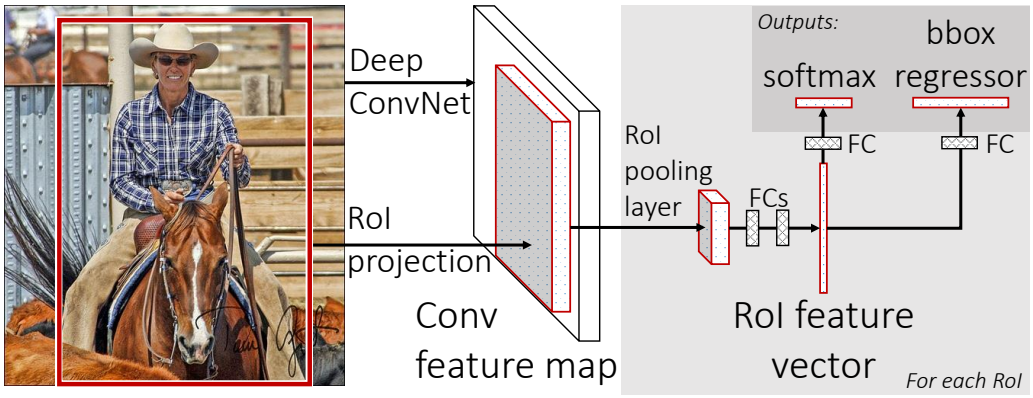
5.2 Μέθοδος Fast R-CNN

Το 2015 προτάθηκε από τον Ross Girshick ένας βελτιωμένος R-CNN ανιχνευτής, ο Fast R-CNN [16] ο οποίος προσπαθεί να βελτιώσει τα σημεία στα οποία υστερεί ο προκάτοχός του.

Ο Fast R-CNN καταφέρνει να πετύχει καλύτερη ποιότητα ανίχνευσης (mAP) από τον R-CNN, καθώς και να βελτιώσει σημαντικά την ταχύτητα και την χρήση χώρου, μετατρέποντας την διαδικασία εκπαίδευσης από τριών σταδίων σε ενός σταδίου, επιτρέποντας όλα τα επίπεδα να ενημερωθούν κατά την εκπαίδευση και αφαιρώντας την ανάγκη για ενδιάμεση αποθήκευση των χαρακτηριστικών στον δίσκο.

Ακολουθώντας την ίδια διαδικασία με τον R-CNN, ξεκινάει από ένα CNN αρχικοποιημένο με προ-εκπαίδευση με δεδομένα της βάση ImageNet. Η αρχιτεκτονική του νευρωνικού περιέχει διάφορα συνελικτικά και max pooling επίπεδα, ακολουθούμενα από ένα pooling επίπεδο περιοχών ενδιαφέροντος (region of interest - RoI) και τελικά πολλά πλήρως συνδεδεμένα επίπεδα. Το προ-εκπαιδευμένο αυτό νευρωνικό τροποποιείται έτσι ώστε να τερματίζει σε δύο επίπεδα, ένα με την softmax πιθανότητα εκτίμησης για τις 20 κλάσεις αντικειμένων + 1 ειδική κλάση φόντου, και το άλλο με 4 πραγματικούς αριθμούς για κάθε μία από τις 20 κλάσεις. Αυτοί οι 80 αριθμοί έχουν την πληροφορία για τους bounding-box regressors. Η μέθοδος Fast R-CNN φαίνεται στο σχήμα 5.2.

Σε αντίθεση με τον R-CNN που ακολουθεί μια διαδικασία εξαγωγής χαρακτηριστικών για κάθε RoI ξεχωριστά, ο Fast R-CNN ακολουθεί μια εικονοκεντρική προσέγγιση όπου υπολογίζονται τα χαρακτηριστικά πάνω σε όλη την εικόνα και τα RoI της ίδιας εικόνας μοιράζονται υπολογισμούς και μνήμη, κάνοντας την εκπαίδευση



Σχήμα 5.2: Fast R-CNN. Το σύστημα παίρνει μια εικόνα και διάφορες περιοχές ενδιαφέροντος (RoI) - υποψήφιες θέσεις αντικειμένων σαν είσοδο του πλήρως συνελικτικού νευρωνικού δικτύου. Κάθε RoI αντιστοιχίζεται με ένα διάνυσμα χαρακτηριστικών μέσω πλήρως συνδεδεμένων επιπέδων. Το νευρωνικό έχει σαν έξοδο δύο διανύσματα ανά RoI: την softmax πιθανότητα και την ανά κλάση απόκλιση του παραθύρου που περικλείει το αντικείμενο. Η αρχιτεκτονική εκπαιδεύεται από άκρη σε άκρη με την χρήση μια ενιαίας συνάρτησης απώλειας.

πιο αποδοτική. Επιπλέον από αυτή την εικονοκεντρική προσέγγιση, ο Fast R-CNN εκπαιδεύει το νευρωνικό με μια ενιαία διαδικασία που βελτιστοποιεί ταυτόχρονα και τον softmax ταξινομητή, και τους bounding box regressors. Για τον σκοπό αυτό χρησιμοποιείται μια ενιαία συνάρτηση απώλειας.

Αφού τελειώσει η εκπαίδευση, η ανίχνευση κοστίζει λίγο παραπάνω από ένα προς τα εμπρός τρέξιμο στο νευρωνικό (θεωρώντας ότι οι υποψήφιες θέσεις είναι ήδη υπολογισμένες). Το νευρωνικό παίρνει σαν είσοδο μια εικόνα και R υποψήφιες θέσεις για βαθμολόγηση. Για κάθε υποψήφια θέση r, το προς τα εμπρός τρέξιμο του νευρωνικού δίνει σαν έξοδο μια κατανομή πιθανότητας κλάσης p και ένα σύνολο από προβλεπόμενα offsets του παραθύρου σχετικά με το r (κάθε κλάση έχει δικό της προβλέπτη offset). Σε κάθε r λοιπόν, για κάθε κλάση k δίνεται μια τιμή εμπιστοσύνης ανίχνευσης που προκύπτει από την εκτιμώμενη πιθανότητα $Pr(class = k|r) \triangleq p_k$. Τέλος εφαρμόζεται nms ανεξάρτητα για κάθε κλάση.

Ο Fast R-CNN δίνει state-of-the-art ποιότητα ανίχνευσης με 66.9% mAP στην βάση PASCAL που εξετάζουμε με χρόνο εκτέλεσης μόλις 0.1 δευτερόλεπτα ανά εικόνα με χρήση GPU, χωρίς να υπολογίσουμε τον χρόνο παραγωγής των υποψήφιων θέσεων. Ο μικρός αυτός χρόνος εκτέλεσης μας επιτρέπει την χρήση περισσότερων υποψήφιων παραθύρων απ' ότι ήταν δυνατόν μέχρι τώρα.

5.3 Πειράματα

Λόγω της εξαιρετικής ποιότητας ανίχνευσης του Fast R-CNN αλλά και του ταχύτατου χρόνου εκτέλεσής του, θεωρήσαμε σκόπιμο να χρησιμοποιήσουμε αυτόν τον ανιχνευτή για να ελέγξουμε την ποιότητα των αποτελεσμάτων της μεθόδου μας.

Η μετρική που χρησιμοποιείται για να μετρήσουμε την απόδοση του ανιχνευτή είναι η μέση ακρίβεια (average precison - AP). Για κάθε μέθοδο που θέλουμε να εξετάσουμε, υπολογίζεται η καμπύλη της *ακρίβειας* ως προς την *ανάκληση*. Η ανάκληση ορίζεται ως το ποσοστό των παραθύρων αναφοράς που επιστρέφει η μέθοδος προς το σύνολο των παραθύρων αναφοράς. Η ακρίβεια ορίζεται ως το ποσοστό των σωστών αποτελεσμάτων που επιστρέφει η μέθοδος προς το σύνολο των αποτελεσμάτων. Η AP συνοψίζει το σχήμα της καμπύλης ακρίβειας/ανάκλησης, και ορίζεται ως η μέση ακρίβεια σε ένα σύνολο από έντεκα ισαπέχοντα επίπεδα ανάκλησης

$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{\mathrm{interp}}(r) \tag{5.1}$$

όπου $p_{\mathrm{interp}}(r)$ είναι η μέγιστη τιμή της ακρίβειας που αντιστοιχεί σε ανάκληση μεγαλύτερη ή ίση του r

$$p_{\mathrm{interp}}(r) = \max_{\bar{r}:\bar{r} \geq r} p(\bar{r}) \tag{5.2}$$

όπου $p(\bar{r})$ είναι η μετρούμενη ακρίβεια σε ανάκληση $\bar{r}$.

Καθώς η AP υπολογίζεται ξεχωριστά για κάθε κλάση, χρησιμοποιούμε και την μέση τιμή όλων των κλάσεων mAP (mean Average Precision) σαν μέτρο απόδοσης.

Ο αλγόριθμος παίρνει σαν είσοδο τις υποψήφιες θέσεις ωστόσο, λόγω του μεγάλου χρόνου εκπαίδευσης καθίσταται αρκετά δύσκολη η εκπαίδευση για κάθε μέθοδο υποψήφιων θέσεων αντικειμένων έτσι ώστε να γίνει η σύγκριση. Γι'αυτό το λόγο χρησιμοποιήσαμε το ήδη εκπαιδευμένο με την μέθοδο Selective Search νευρωνικό, και εισάγαμε τις υποψήφιες θέσεις που προκύπτουν από τις διάφορες μεθόδους μόνο κατά την διαδικασία της εκτέλεσης της ανίχνευσης. Στην πράξη χρησιμοποιήσαμε 10000 υποψήφιες θέσεις. Στους πίνακες 5.2 και 5.3 έχουμε τα αποτελέσματα της ανίχνευσης για κάθε κλάση αντικειμένων ξεχωριστά, ενώ στον πίνακα 5.1 έχουμε το mAP πάνω σε όλες τις κλάσεις.

Συγκρίνοντας τα αποτελέσματα των πινάκων μπορούμε να δούμε ότι η μέθοδος Selective Search εμφανίζει την καλύτερη απόδοση συνολικά με mAP = 66.8. Αυτό είναι λογικό αφού το νευρωνικό έχει εκπαιδευτεί και άρα έχει προσαρμοστεί σε αυτήν. Ωστόσο βλέπουμε ότι η απόκλιση των άλλων μεθόδων δεν είναι πολύ μεγάλη με την edgeBoxes να φτάνει ανίχνευση mAP = 66.1 ενώ ακολουθεί η δική μας με mAP = 65.6, μόλις 1.2 μονάδες κάτω από την Selective Search. Αυτό δείχνει ότι τα αποτελέσματα της μεθόδου μας είναι αρκετά καλής ποιότητας. Επίσης από τους πίνακες 5.2 και 5.3 βλέπουμε ότι διαφορετικές μέθοδοι αποδίδουν καλύτερα σε διαφορετικά είδη αντικειμένων, κάτι που ενισχύει την πεποίθησή μας ότι δεν υπάρχει ακόμα μια μέθοδος που να ξεχωρίζει σε όλους του τομείς.

Πίνακας 5.1: Μέση τιμή της μέσης ακρίβειας για όλες τις κλάσεις

method	mAP
bing	51.4
objectness	54.6
cpmc	64.5
mcg	64.1
rantalankila	62.3
random prim's	64.7
rahtu	59.5
edgeBoxes	66.1
selectiveSearch	**66.8**
segmentBoxes	65.6
segmentBoxesBB	65.6

Όσον αφορά τις δικές μας μεθόδους, χρησιμοποιούμε την κλασική μέθοδο με διαδικασία τελειοποίησης (segmentBoxes) και την μέθοδο με το βέλτιστο εσωτερικό παράθυρο (segmentBoxesBB). Παρατηρούμε ότι παρόλο που στα προηγούμενα πειράματά μας, οι δύο αυτές προσεγγίσεις διαφέρουν αρκετά στην μετρική της ανάκλησης, με την μέθοδο με βέλτιστο εσωτερικό παράθυρο να υστερεί αρκετά, όσον αφορά την μετρική mAP έχουν την ίδια ποιότητα αποτελεσμάτων. Αυτό είναι μια ισχυρή ένδειξη ότι η μετρική της ανάκλησης δεν είναι αρκετή για να αποδώσει την ποιότητα των αλγορίθμων ανίχνευσης υποψήφιων θέσεων στο πραγματικό πρόβλημα της ανίχνευσης αντικειμένων.

Για να εξηγήσουμε αυτό το αποτέλεσμα αρκεί να αναλύσουμε την διαδικασία των δύο μεθόδων. Η μέθοδος με την διαδικασία τελειοποίησης επιστρέφει 10000 παράθυρα που χρησιμοποιούνται όλα στον ανιχνευτή ενώ η μέθοδος με εσωτερικό παράθυρο επιστρέφει περίπου 1000 με 2000. Αυτό έχει σαν αποτέλεσμα η δεύτερη να χάνει κάποια αντικείμενα αλλά η πρώτη να επιστρέφει και πολλές θέσεις στις οποίες δεν υπάρχει αντικείμενο. Έτσι ο αλγόριθμος ανίχνευσης εξετάζει και αυτές τις θέσεις και τους αποδίδει κάποια πιθανότητα ανίχνευσης, σε μερικές περιπτώσεις αρκετά υψηλή, με αποτέλεσμα να μειώνεται η ακρίβειά του (false positives).

Χαρακτηριστικά μπορούμε να δούμε την κλάση των αεροπλάνων που όπως φαίνεται και στον πίνακα 5.2 οι δύο προσεγγίσεις μας διαφέρουν αρκετά. Όπως μπορούμε να δούμε και στο σχήμα 5.3, η πρώτη προσέγγιση επιστρέφει παράθυρα που ταυτοποιούνται εσφαλμένα σαν "αεροπλάνο" όπως τα διάφορα τμήματα του αεροπλάνου του 5.3(α) και το βάζο του 5.3(ζ), κάτι που δεν συμβαίνει στην δεύτερη. Ωστόσο, όπως είπαμε, καθώς η μέθοδος με το εσωτερικό παράθυρο επιστρέφει λιγότερα παράθυρα, χάνει μερικά αντικείμενα όπως τα υπόλοιπα αεροπλάνα του σχήματος 5.3(στ).

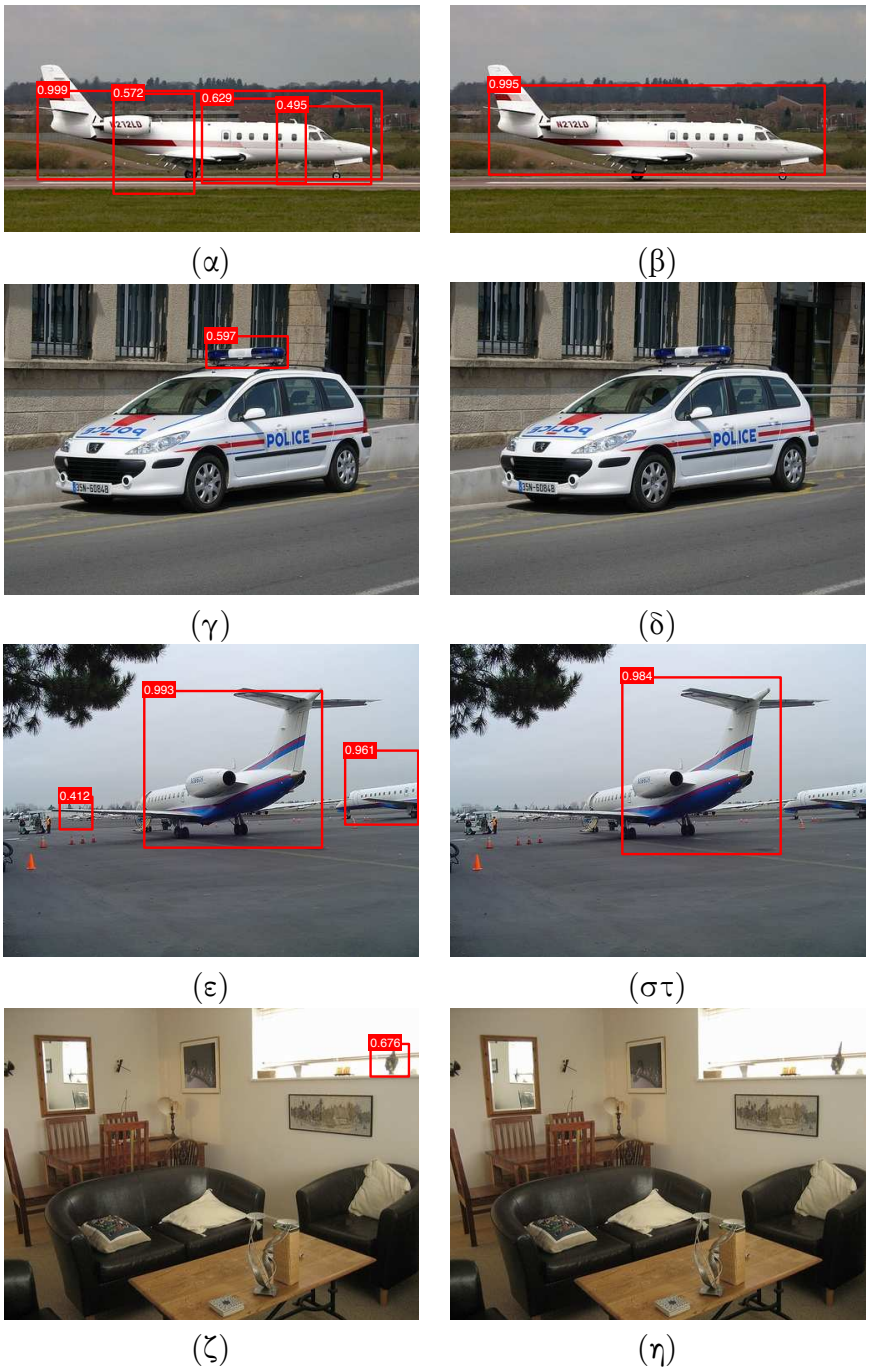
Με βάση το παραπάνω μπορούμε να πούμε ότι ο αριθμός των παραθύρων είναι εξίσου σημαντικό στοιχείο ποιότητας των αλγορίθμων ανίχνευσης υποψήφιων θέσεων με την ανάκληση και δεν πρέπει να υποτιμάται η σημαντικότητά του.

Πίνακας 5.2: Μέση ακρίβεια για κάθε κλάση με τον Fast R-CNN

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
bing	56.7	57.9	49.6	37.2	27.6	64.9	66.3	67.5	25.2	55.7
objectness	58.4	64.4	55.1	40.4	26.1	69.2	65.9	72.2	31.1	61.7
cpmc	69.9	71.5	64.2	50.4	31.7	77.7	70.7	80.7	36.1	69.8
mcg	68.6	72.9	63.8	53.7	39.2	77.6	70.1	81.3	39.7	67.0
rantalankila	66.8	68.6	61.3	48.2	35.5	76.7	68.4	79.7	39.7	67.2
random prim's	70.8	**77.5**	64.8	54.0	41.7	78.4	70.5	78.8	**42.8**	67.1
rahtu	67.0	66.3	59.7	50.5	30.2	76.7	61.8	78.7	32.7	61.2
edgeBoxes	62.7	76.4	**66.4**	**54.7**	**42.7**	**80.6**	77.5	81.1	39.9	72.5
selectiveSearch	**76.0**	76.8	65.3	54.6	38.0	76.5	**78.2**	**81.6**	40.1	**74.1**
segmentBoxes	63.2	76.3	64.7	53.6	42.6	78.8	77.1	**81.6**	40.1	71.3
segmentBoxesBB	72.5	76.8	65.9	54.5	37.1	79.9	77.5	80.3	37.0	72.8

Πίνακας 5.3: Μέση ακρίβεια για κάθε κλάση με τον Fast R-CNN - συνέχεια

method	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
bing	46.8	63.7	68.0	59.4	51.2	21.5	45.2	47.4	61.5	53.9
objectness	50.8	66.7	72.5	62.0	49.7	21.4	49.0	53.2	66.1	55.6
cpmc	67.0	**80.1**	80.8	70.7	63.8	34.5	64.4	65.9	**75.1**	64.5
mcg	**68.9**	76.9	75.6	68.8	60.8	33.8	59.6	64.5	70.6	**68.3**
rantalankila	65.9	76.4	75.9	66.7	59.2	28.5	58.4	65.9	69.4	66.6
random prim's	68.7	76.7	77.9	70.0	60.8	33.0	58.4	66.6	70.3	66.0
rahtu	65.0	70.4	73.6	61.9	59.0	28.8	49.8	62.1	70.3	64.0
edgeBoxes	64.7	78.9	79.8	**76.2**	**67.6**	**36.1**	**65.6**	62.3	68.6	67.2
selectiveSearch	66.5	78.9	**81.8**	74.5	66.2	32.9	**65.6**	**67.7**	73.4	66.8
segmentBoxes	64.0	79.9	79.8	72.5	67.0	35.4	65.2	62.8	71.1	64.1
segmentBoxesBB	61.9	76.8	80.3	76.0	66.1	35.7	63.8	65.0	69.8	62.1



Σχήμα 5.3: Αριστερή στήλη: ανίχνευση αεροπλάνων με την χρήση Segment Boxes με διαδικασία τελειοποίησης. Δεξιά στήλη: ανίχνευση αεροπλάνων με την χρήση Segment Boxes με βέλτιστο εσωτερικό παράθυρο.

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Στόχος μας όπως είπαμε ήταν να δούμε κατά πόσο η κατάτμηση μπορεί να μας δώσει καλές λύσεις στο πρόβλημα που εξετάζουμε. Η ιδέα μας είναι αρκετά απλή, και δεν χρησιμοποιεί πολλά χαρακτηριστικά της εικόνας όπως η objectness και η Selective Search. Παρόλα αυτά, με βάση τις συγκρίσεις που κάναμε, πετύχαμε συγκρίσιμη και σε ορισμένες περιπτώσεις καλύτερη ποιότητα αποτελεσμάτων από πολλές, πιο σύνθετες μεθόδους. Συγκεκριμένα όπως είδαμε πετυχαίνουμε βέλτιστη ανάκληση για

- IoU 0.5 για 1000 προτεινόμενα παράθυρα στην υλοποίηση Βέλτιστο Εσωτερικό Παράθυρο,
- IoU < 0.6 για 10000 προτεινόμενα παράθυρα στην απλή υλοποίηση με διαδικασία τελειοποίησης.

Είναι επίσης αξιοσημείωτο ότι ενώ η μέθοδός μας αποδίδει σχετικά μέτρια ως προς την μετρική της ανάκλησης, στο πραγματικό πρόβλημα που είναι η ανίχνευση αντικειμένων έχουμε ιδιαίτερα καλά αποτελέσματα. Η μέση ακρίβεια της ανίχνευσης του Fast R-CNN με την χρήση της μεθόδου μας είναι μόλις 1,2% mAP κάτω από την Selective Search και με ταχύτητα 0.3 δευτερόλεπτα ανά εικόνα. Αυτό σημαίνει ότι η μετρική της ανάκλησης δεν είναι από μόνη της καλή ένδειξη της ποιότητας των ανιχνεύσεων υποψήφιων θέσεων αντικειμένων και ίσως θα έπρεπε να βρεθεί κάποια καλύτερη μετρική για αυτόν τον σκοπό.

Παράλληλα αναδείξαμε και την σημασία του αριθμού των παραθύρων, αποδεικνύοντας ότι υψηλότερη ανάκληση δεν συνεπάγεται απαραίτητα καλύτερη ποιότητα αποτελεσμάτων κάτι που κατ' επέκταση σημαίνει ότι η χρήση ανίχνευσης υποψήφιων αντικειμένων, προσφέρει σε σχέση με την μέθοδο κυλιόμενου παραθύρου, εκτός από ταχύτητα, και βελτίωση της ποιότητας.

Ένα επίσης από τα χαρακτηριστικά της μεθόδου μας είναι ότι δεν απαιτείται εκμάθηση σε κανένα σημείο της, οπότε και δεν έχει προσαρμοστεί να δουλεύει πάνω σε συγκεκριμένο σύνολο εικόνων. Με απλή χρήση του κώδικά μας, μπορούμε να πάρουμε

τα υποψήφια παράθυρα για οποιαδήποτε εικόνα μας ενδιαφέρει χωρίς την ανάγκη για εκπαίδευση ή προεκπαιδευμένα δεδομένα.

Επιπλέον πρόκειται για ένα γρήγορο αλγόριθμο, αρκετά πιο γρήγορο από τις περισσότερες μεθόδους, με 2 δευτερόλεπτα ανά εικόνα στην βασική υλοποίηση και 0.3 δευτερόλεπτα στην υλοποίηση με το εσωτερικό παράθυρο. Όπως είπαμε η ταχύτητα είναι πολύ σημαντικό χαρακτηριστικό αυτού του είδους μεθόδων καθώς πρόκειται για προπαρασκευαστικό στάδιο την πραγματικής ανίχνευσης αντικειμένων.

Παρόλα αυτά η μέθοδός μας δεν κατάφερε να καλύψει το κενό που αναφέραμε αρχικά για την έλλειψη μιας βέλτιστης μεθόδου ανίχνευσης υποψήφιων θέσεων αντικειμένων.

Ένα από τα βασικά προβλήματα της μεθόδου μας βρίσκεται στην ίδια την κατάτμηση. Η αρχική κατάτμηση της εικόνας παίζει ιδιαίτερα σημαντικό ρόλο και αν ο αλγόριθμος δεν είναι καλός, το αντικείμενο θα χαθεί και δεν μπορεί να εντοπιστεί από τον δικό μας. Ωστόσο δεν υπάρχει αλγόριθμος που να παράγει τέλειας ποιότητας τμήματα, οπότε είναι φυσικό να εισάγονται σφάλματα με αυτή την διαδικασία. Συγκεκριμένα η κατάτμηση δεν διατηρεί εύκολα τα όρια μικρών αντικειμένων, ενώ εμφανίζει και προβλήματα στην ακρίβεια των ορίων των αντικειμένων, και άρα στην ακρίβεια των δικών μας παραθύρων.

Σημαντικό ρόλο παίζουν επίσης και τα αρχικά υποψήφια παράθυρα που επιλέγονται. Αν είναι πολύ αραιά θα έχουμε πολλά χαμένα αντικείμενα ενώ αν είναι πολύ πυκνά θα έχουμε πολλά όχι και τόσο καλά αποτελέσματα ανάμεσα στα καλά καθώς και αυξημένο χρόνο εκτέλεσης.

Ένα άλλο σημείο στο οποίο φαίνεται η μέθοδός μας να έχει προβλήματα σε σχέση με τις σύγχρονες μεθόδους είναι η μικρή ανάκληση για μεγάλα IoU. Αυτό δεν αποτελεί ιδιαίτερο πρόβλημα για την ποιότητα της ανίχνευσης, όπως είδαμε και με την εκτέλεση του Fast R-CNN, καθώς οι περισσότεροι ανιχνευτές χρησιμοποιούν υποψήφιες θέσεις με IoU $= 0.5$ αλλά μπορεί να υπάρξει αν εμφανιστεί ένα αλγόριθμος ανίχνευσης που απαιτεί μεγαλύτερα IoU.

Τέλος, όπως και όλοι οι αλγόριθμοι που βασίζονται σε κατάτμηση, έτσι και ο δικός μας, είναι επιρρεπής σε μικρές αλλαγές της εικόνας, κάτι που έχει σαν αποτέλεσμα μικρή δυνατότητα γενίκευσης σε άλλα δεδομένα [20].

6.2 Μελλοντική έρευνα

Η ανίχνευση αντικειμένων καθώς και η ανίχνευση πιθανών θέσεων, είναι προβλήματα που απασχολούν ιδιαίτερα την επιστημονική κοινότητα τα τελευταία χρόνια, με μεθόδους και τεχνικές να εμφανίζονται με εξαιρετικά μεγάλο ρυθμό. Ήδη από την στιγμή που αρχίσαμε την εκπόνηση της εργασίας μέχρι σήμερα έχουν γίνει εξαιρετικά βήματα προς την επίτευξη καλύτερων και πιο γρήγορων ανιχνεύσεων.

Όσον αφορά την προσέγγιση με την χρήση κατάτμησης, ο καλύτερος τρόπος για την βελτίωση της ποιότητας των αποτελεσμάτων θα ήταν η ανάπτυξη ενός καλύτερου αλγορίθμου κατάτμησης. Ωστόσο υπάρχει περιθώριο βελτίωσης και στο κομμάτι της χρήσης των τμημάτων.

Στην προσέγγιση με την χρήση βαρών δοκιμάσαμε μόνο το χρώμα σαν χαρακτηριστικό των τμημάτων. Πιθανώς σε συνδυασμό με κάποιο άλλο στοιχείο του τμήματος, όπως η υφή και το σχήμα, να απέδιδε καλύτερα κατά την ένωση τους. Είναι επίσης απαραίτητη μια βελτιωμένη διαδικασία κανονικοποίησης έτσι ώστε να μην δίνεται έμφαση στα μικρά τμήματα. Επιπλέον, η μέθοδός μας χρησιμοποιεί παράθυρα παράλληλα στις πλευρές της εικόνας όμως υπάρχουν περιπτώσεις που τα αντικείμενα είναι σε κλίση. Αυτό δεν είναι πολύ συχνό στις βάσεις που χρησιμοποιήσαμε αλλά στην πραγματικότητα είναι αρκετά πιθανό να συναντήσουμε τέτοια εικόνα. Προσπαθήσαμε να επιλύσουμε το πρόβλημα με την προσέγγιση εμβαδού εσωτερικού παραθύρου ανεπιτυχώς. Πιθανώς κάποια ιδέα σε αυτό το κομμάτι να κατάφερνε να εντοπίσει με μεγαλύτερη επιτυχία αυτά τα αντικείμενα. Χρήσιμη επίσης θα ήταν η χρήση συμφραζομένων, δηλαδή πληροφορίας από τα γύρω τμήματα για να αποφασιστεί αν ανήκουν ή όχι στο ίδιο αντικείμενο.

Τέλος μια απλή ιδέα που θα μπορούσε να ενσωματωθεί στην μέθοδό μας προτάθηκε πρόσφατα ([38]) και έχει να κάνει με την διαδικασία της τελειοποίησης. Στην μέθοδό μας προσεγγίζουμε το βέλτιστο παράθυρο εσωτερικά από το παράθυρο που εξετάζουμε. Στην εργασία [38] προτείνεται μια μέθοδο με επέκταση παραθύρου προς τα έξω που φαίνεται να έχει θετικά αποτελέσματα.

Παρά το γεγονός ότι ακόμα οι περισσότεροι αλγόριθμοι περιέχουν επιλεγμένα με το χέρι χαρακτηριστικά της εικόνας για την εξαγωγή των υποψήφιων θέσεων, πλέον, όπως και με τους αλγόριθμους ανίχνευσης αντικειμένων έτσι και εδώ, υπάρχει τάση για την χρήση μάθησης σε βάθος. Ήδη έχουν δημοσιευτεί άρθρα που χρησιμοποιούν βαθιά συνελικτικά νευρωνικά δίκτυα (Deep Convolutional Neural Networks - dCNN) [22, 33] για να επιστρέψουν υποψήφιες θέσεις μέσα από μια διαδικασία εκμάθησης, τα οποία πετυχαίνουν εξίσου καλά αποτελέσματα με τις υπόλοιπες μεθόδους.

Ωστόσο, η εργασία που ξεχωρίζει είναι η Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [31] που προτάθηκε πρόσφατα από τους S Ren, K He, R Girshick και J Sun. Το γεγονός ότι με τον Fast R-CNN ο χρόνος εκτέλεσης της ανίχνευσης (0.1 δευτερόλεπτα) είναι μικρότερος από τον χρόνο παραγωγής των υποψήφιων θέσεων αντικειμένων (βλέπε πίνακα 4.1), οδήγησε τους συγγραφείς να προσθέσουν δύο επιπλέον επίπεδα στο νευρωνικό του Fast R-CNN για να εξάγουν χαρακτηριστικά που οδηγούν στην ανίχνευση υποψήφιων θέσεων. Έτσι με σχεδόν μηδενικό επιπλέον χρονικό κόστος, η μέθοδός τους παράγει τόσο υποψήφιες θέσεις, όσο και ανίχνευση αντικειμένων σε μόλις 0.1 δευτερόλεπτα σε GPU με καλύτερη ποιότητα απ' ότι με την χρήση της Selective Search (73.2 mAP) στην βάση εικόνων PASCAL VOC07.

Επίσης εντυπωσιακή είναι και η εργασία [30] η οποία έχει μεν μικρότερη ποιότητα ανίχνευσης αλλή καταφέρνει να τρέξει σε πραγματικό χρόνο με 45 εικόνες το δευτερόλεπτο σε GPU.

Με βάση το παραπάνω και την εμπειρία μας, μπορούμε να πούμε ότι ο τομέας της ανίχνευσης υποψήφιων θέσεων πιθανότατα θα ξεφύγει από τα επιλεγμένα με το χέρι χαρακτηριστικά και στο μέλλον θα εμφανιστούν πολλοί καλύτεροι αλγόριθμοι που χρησιμοποιούν μάθηση σε βάθος.

Βιβλιογραφία

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73--80. IEEE, 2010.

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2189--2202, 2012.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898--916, 2011.

[4] Pablo Arbelaez, Jordi Pont-Tuset, Jon Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 328--335. IEEE, 2014.

[5] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679--698, 1986.

[6] Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241--3248. IEEE, 2010.

[7] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3286--3293. IEEE, 2014.

[8] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Segmentation driven object detection with fisher vectors. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2968--2975. IEEE, 2013.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886--893. IEEE, 2005.

[10] Piotr Dollár and C Zitnick. Fast edge detection using structured forests. 2014.

[11] Ian Endres and Derek Hoiem. Category independent object proposals. In *Computer Vision--ECCV 2010*, pages 575--588. Springer, 2010.

[12] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2155--2162. IEEE, 2014.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303--338, 2010.

[14] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627--1645, 2010.

[15] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167--181, 2004.

[16] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580--587. IEEE, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv:1406.4729*, 2014.

[19] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3):328--346, 2011.

[20] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *arXiv preprint arXiv:1502.05082*, 2015.

[21] Yangqing Jia. Caffe: An open source convolutional architecture for fast feature embedding, 2013.

[22] Nikolaos Karianakis, Thomas J Fuchs, and Stefano Soatto. Boosting convolutional features for robust object proposals. *arXiv preprint arXiv:1503.06350*, 2015.

[23] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Computer Vision--ECCV 2014*, pages 725--739. Springer, 2014.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097--1105, 2012.

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324, 1998.

[26] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2536--2543. IEEE, 2013.

[27] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15--33, 2000.

[28] Esa Rahtu, Juho Kannala, and Matthew Blaschko. Learning a category independent object detection cascade. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1052--1059. IEEE, 2011.

[29] Pekka Rantalankila, Juho Kannala, and Esa Rahtu. Generating object segmentation proposals using global and local search. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2417--2424. IEEE, 2014.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[33] Christian Szegedy, Scott Reed, Dumitru Erhan, and Dragomir Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[34] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154--171, 2013.

[35] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879--1886. IEEE, 2011.

[36] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137--154, 2004.

[37] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 17--24. IEEE, 2013.

[38] Xiaozhi Chen Huimin Ma Xiang Wang and Zhichen Zhao. Improving object proposals with multi-thresholding straddling expansion. 2015.

[39] Ren Xiaofeng and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *Advances in neural information processing systems*, pages 584--592, 2012.

[40] Qiyang Zhao, Zhibin Liu, and Baolin Yin. Cracking bing and beyond.

[41] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision--ECCV 2014*, pages 391--405. Springer, 2014.