Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Μετάδοση γνώσης αλγορίθμων εντοπισμού αντικειμένων σε ενσωματωμένα συστήματα πραγματικού χρόνου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ Σ. ΑΘΑΝΑΣΙΑΔΗ

Επιβλέπων: Λουκάς Πέτρου
Καθηγητής Α.Π.Θ.

Εργαστήριο Ευφυών Συστημάτων & Τεχνολογίας λογισμικού
Θεσσαλονίκη, ???? 2018

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων & Τεχνολογίας λογισμικού

Μετάδοση γνώσης αλγορίθμων εντοπισμού αντικειμένων σε ενσωματωμένα συστήματα πραγματικού χρόνου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ Σ. ΑΘΑΝΑΣΙΑΔΗ

Επιβλέπων: Λουκάς Πέτρου
Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ??η ?????? 2018.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Λουκάς Πέτρου	Κάποιος	Κάποιος άλλος
Καθηγητής Α.Π.Θ.	Καθηγητής Α.Π.Θ.	Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, ?????? 2018

(Υπογραφή)

.....
Ιωάννης Αθανασιάδης
Υποψήφιος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Α.Π.Θ.

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ήλεκτρονικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων & Τεχνολογίας λογισμικού

Ευχαριστίες

Περίληψη

Λέξεις Κλειδιά

Abstract

Keywords

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

Η Τεχνητή Νοημοσύνη είναι ο κλάδος/τομέας της επιστήμης της πληροφορικής, που ασχολείται με την σχεδίαση και κατασκευή ευφυών συστημάτων, δηλαδή συστημάτων που διαθέτουν χαρακτηριστικά που σχετίζονται με την ανθρώπινη νοημοσύνη και συμπεριφορά. Ο πιο ισχυρός υπολογιστής που γνωρίζει ο άνθρωπος μέχρι στιγμής είναι ο ίδιο εγκεφαλός του(όπως βέβαια και οι εγκεφάλοι άλλων ζώων). Αυτός για να φτάσει στο σημείο να χαίρει της τωρινής υπολογιστικής του ικανότητας χρειάστηκε εκατομύρια χρόνια βιολογικής εξέλιξης. Ακόμα και με αυτή την υπολογιστική ικανότητα που διαθέτει συνεχίζει να χρησιμοποιεί τη τεχνική μετάδοσης γνώσης, παρά τα χρόνια εξελιξής του. Αυτό σύμφωνα με τη δαρβινική θεωρία[?] δείχνει πως ήταν μία βελτιστοποίηση στην εκπαιδευσή του (δεν ισχυρίζεται η τελειότητα αυτού) η οποία δοκιμάστηκε και επέζησε μέχρι στιγμής τη φυσική επιλογή.

Ως εκ τούτου, ο σχεδιασμός αλγορίθμων και μοντέλων που εκπαιδεύονται μέσω της μετάδοσης γνώσης είναι μία αρκετά δοκιμασμένη επιλογή και μία αρκετά δικαιολογημένη απόφαση. Ωστόσο, πάλι παραδείγματι τον εγκέφαλο τα νευρωνικά δίκτυα εντοπισμού αντικειμένων θα έπρεπε να απαιτούν λιγότερες παραμέτρους και να επεξεργάζονται την εισοδό τους σε πραγματικό χρόνο (τουλ. 24 καρέ /δευτερόλεπτο). Η τελευταία αυτή απαίτηση σε συνδιασμό με την προηγούμενη οδηγούν στη διαμόρφωση του προβλήματος που περιγράφεται παρακάτω.

Επιπλέον, η μετάδοση γνώσης είναι ένα από τα εργαλεία που είναι στενά συνδεδεμένο με την έννοια της Γενικής Τεχνητής νοημοσύνης(General Artificial Intelligence). Αυτό συμβαίνει γιατί μοντελοποιείται ένα πολύ σημαντικό κομμάτι της διαδικασίας του εγκεφάλου: η ίδια η μάθηση.

Ένας σημαντικός τομέας που ωθεί τα σύγχρονα επιτεύγματα στη τεχνητή νοημοσύνη είναι η εμφάνιση και εξέλιξη του κλάδου της Βαθιάς Μηχανικής Μάθησης (Deep learning - DL).Η χρήση τεχνικών βαθιάς μάθησης στην επίλυση προβλημάτων Μηχανικής Όρασης, έχει κατορθώσει να αντιμετωπίσει περίπλοκα προβλήματα τα οποία μέχρι και πριν από λίγα χρόνια θεωρείτο ακατόρθωτο να λυθούν. Ακόμα και η χρήση της DL σε ενσωματωμένα συστήματα με μικρή μνήμη και απαιτήσεις ταχύτατης επεξεργασίας δίνει καλύτερα αποτελέσματα από ότι άλλοι αλγόριθμοι.

Σήμερα, το γενικότερο πρόβλημα της ταυτόχρονης αναγνώρισης και εντοπισμού αντικειμένων σε εικόνες χρησιμοποιεί εκτεταμένα Νευρωνικά Δίκτυα Συνέλιξης (Convolutional Neural Networks - CNNs). Η εκπαίδευση και η επανεκπαίδευση αυτών έχει επισημανθεί ότι οφελείται αρκετά από τη χρήση τεχνικών μετάδοση γνώσης [?]. Σε αυτό παίζουν ρόλο και άλλοι οικονομοτεχνικοί λόγοι όπως το μέγεθος των διαθέσιμων δεδομένων. Σε συνδιασμό όλων των παραπάνω η εργασία καλείται να δώσει κάποιες παρατηρήσεις προς την επίλυση του παρακάτω προβλήματος.

1.1 Περιγραφή του Προβλήματος

Παρόλο που τα νευρωνικά δίκτυα εντοπισμού αντικειμένων από εικόνες έχουν φτάσει στο σημείο να έχουν ικανοποιητική ακρίβεια και να εκτελούνται σε ενσωματωμένα συστήματα, δεν έχουν μελετηθεί όλες οι διαφορές τους και οι σχέσεις τους με αυτά που έχουν μεγάλο αριθμό παραμέτρων και εκτελούνται με μεγαλύτερη απαίτηση ισχύος. Η κύρια μελέτη αυτών των συστημάτων έγκειται στον χρόνο εκτέλεσης, την ακρίβεια και πρόσφατα τη μνήμη[?] και την απαίτηση σε ισχύ. Ένα μεγάλο κεφάλαιο στην κατεύθυνση της έρευνας για τα νευρωνικά δίκτυα που μπορούν να εκτελούνται σε ενσωματωμένα συστήματα είναι η μετάδοση γνώσης.

Είναι σημαντικό για παράδειγμα ένα ρομπότ να μπορεί να χρησιμοποιήσει την προηγούμενα επεξεργασμένη (και μη) πληροφορία που διαθέτει για την ανάκτηση και την επεξεργασία καινούριας. Κατά αυτό τον τρόπο μειώνεται ο χρόνος εκπαιδευσής του σε μία καινούρια εργασία. Ωστόσο, θέλουμε οι ενσωματωμένες συσκευές να είναι όσο πιο "ελκυστικές" γίνεται και συνήθως αρκετά μικρότερες σε μέγεθος από μία συστοιχία υπολογιστών [?, ?], ανάλογα με την εργασία που επιθυμούμε να εκτελέσουν. Αυτό, έχει ως αποτέλεσμα να μην μπορούμε να τοποθετήσουμε ογκώδη, άρα με μεγάλη επεξεργαστική ισχύ, υπολογιστικά συστήματα στα ενσωματομένα συστήματα.

Ήδη η μετάδοση γνώσης χρησιμοποιήται συνεχώς σε προβλήματα ενισχυόμενης μάθησης(RL)[?, ?], ώστε να μπορεί το μοντέλο ενός πράκτορα να μεταφερθεί από την προσομοίωση στην πραγματικότητα. Το ίδιο απαιτείται και στην περίπτωση της τεχνητής όρασης με σκοπό την εξοικονόμηση δεδομένων, την ταχύτερη και την αποτελεσματικότερη εκπαίδευση. Το πρόβλημα είναι πως δεν έχουν γίνει πειραματισμοί και μελέτες για την καταγραφή της συμπεριφοράς των νευρωνικών δικτύων εντοπισμού αντικειμένων ενσωματωμένων συστημάτων ως προς τη μετάδοση γνώσης.

1.2 Σκοπός-Συνεισφορά της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία μελετά την επαναχρησιμοποίηση νευρωνικών δικτύων συνέ- λιξης (CNN) σε εφαρμογές ταυτόχρονης αναγνώρισης και εντοπισμού αντικειμένων (object recognition and localization - object detection) σε εικόνες.

Βασικός σκοπός είναι η μεταφορά πληροφορίας μοντέλων CNN από προηγούμενες εφαρμογές, διατηρώντας κατά το δυνατόν σταθερή την αρχιτεκτονική ώστε να επιτρέπεται συνεχώς η εφαρμογή τους σε προβλήματα πραγματικού χρόνου. Μία ακόμα απαίτηση είναι η ελάχιστη μνήμη του αλγορίθμου, προκειμένου να είναι δυνατή η εκτελεσή του από όσο περισσότερες ενσωματωμένες συσκευές. Επίσης, ζητήται η κατά το δυνατόν υψηλότερη ακρίβεια του μοντέλου μετά από την εκπαίδευση του και μεταφορά γνώσης. Τέλος, απαιτείται η κατά το δυνατόν γρηγορότερη επανεκπαιδευσή του. Όλες αυτές οι απαιτήσεις εξετάζονται κατά το πόσο είναι εφικτές και τι περιορισμούς προϋποθέτουν. Η εξέταση αυτή γίνεται πρώτη φορά μέχρις στιγμής καθότι η μετάδοση γνώσης είναι αρκετά καινούρια τεχνική στο τομέα της τεχνητής νοϋμοσύνης όπως και τα νευρωνικά δίκτυα εντοπισμού πραγματικού χρόνου.

Επίσης, παρουσιάζεται ένας πιο γρήγορος αλγόριθμος για την εκπαίδευση του νευρωνικού δικτύου εντοπισμού *SqueezeDet*, όπως και η εφαρμογή ενός νέου τρόπου βελτιστοποίησης υπερπαραμέτρων των νευρωνικών δικτύων. Τέλος, γίνεται και μία πειραματική αναφορά στην πλαστικότητα των νευρώνων των δικτύων εντοπισμού αντικειμένων για ενσωματωμένα συστήματα.

1.3 Διάρθρωση της αναφοράς

Η διάρθρωση της παρούσας διπλωματικής εργασίας είναι η εξής:

- **Κεφάλαιο ??** Περιγράφονται τα σημαντικότερα νευρωνικά δίκτυα εντοπισμού αντικειμένων σε εικόνα. Γίνεται αναφορά στη δυνατότητα εκτελεσής τους από ενσωματωμένες συσκευές και προς το σκοπό αυτό γίνονται και μετρήσεις.

- **Κεφάλαιο ??** Δίνονται οι ορισμοί που αφορούν τη μετάδοση γνώσης. Αναφέρονται παραδείγματα και τεχνικές μετάδοσης γνώσης, όπως επίσης και το θεωρητικό υπόβαθρο για το μετέπειτα πειραματισμό.

- **Κεφάλαιο ??** Παρουσιάζονται τα πειράματα από την περιγραφή τους, τα αποτελεσματά τους και τέλος τα συμπεράσματα που προέρχονται από αυτά.

- **Κεφάλαιο ??** Επιγραμματική περιγραφή της υλοποίησης των πειραμάτων όσον αφορά το υλικό και το λογισμικό που χρησιμοποιήθηκε.

- **Κεφάλαιο ??** Αναφέρονται τα προβλήματα που προέκυψαν και προτείνονται θέματα για μελλοντική μελέτη, αλλαγές και επεκτάσεις.

Κεφάλαιο 2

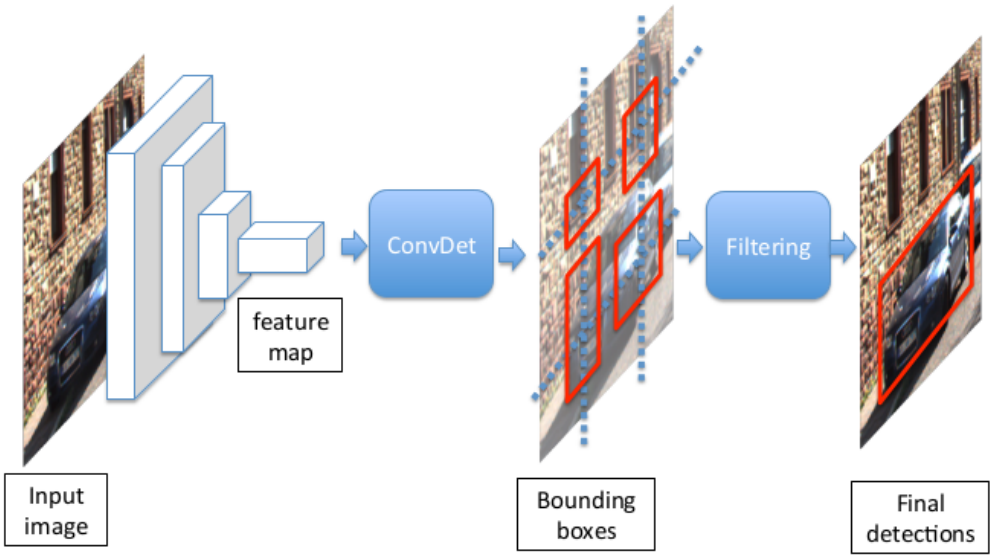
Νευρωνικά Δίκτυα για Εντοπισμό Αντικειμένων

2.1 Εισαγωγή

Τα τελευταία χρόνια έχει προταθεί ένας μεγάλος αριθμός νευρωνικών δικτύων για τον εντοπισμό αντικειμένων από εικόνες. Η εισαγωγή της έννοιας του CNN, όπου πρόκειται για νευρωνικό δίκτυο το οποίο έχει επίπεδα που εκτελούν συνελίξεις για την αναγνώριση χειρόγραφων αριθμών είχε ήδη προταθεί το 1998 [?] και το 2006 [?]. Ωστόσο, λόγω υψηλής απαίτησης σε υπολογιστική ισχύ δεν προτεινόταν η χρήση τους στην επίλυση προβλημάτων αναγνώρισης αντικειμένων ή εντοπισμό. Η αλλαγή της πορείας έγινε το 2012, όταν προτάθηκε το AlexNet [?] το οποίο είναι ένα νευρωνικό δίκτυο πολλών επιπέδων (deep) το οποίο έχει επίπεδα συνελίσουν την είσοδό τους με τα βάρη τους (convolutional). Από τότε και έπειτα άρχισε η χρήση και η δημιουργία καινούριων CNN για προβλήματα που εμπεριέχουν την αναγνώριση και τον εντοπισμό αντικειμένων. Τα δίκτυα που προτάθηκαν το δεύτερο μισό του 2017 και μόνο για την αναγνώριση/εντοπισμό αντικειμένων ξεπερνούν τα δέκα. Επιπλέον η απαίτηση χρήση τους σε πραγματικού χρόνου εφαρμογές και σε ενσωματωμένα συστήματα έδωσε μια άλλη οπτική ανάπτυξής τους. Σε αυτό το κεφάλαιο αναλύονται τα σημαντικότερα δίκτυα που αποτελούν ορόσημα για την χρήση των νευρωνικών σε ενσωματωμένες συσκευές.

2.2 SqueezeDet-Net [?] [?]

Τα δίκτυα αυτά προτάθηκαν από ερευνητές του πανεπιστημίου του Berkeley και της εταιρίας Deepscale ως μια λύση για το πρόβλημα του μεγάλου αριθμού παραμέτρων ενός νευρωνικού δικτύου για εντοπισμό αντικειμένων. Το SqueezeNet ουσιαστικά εμπεριέχεται στο SqueezeDet ως ένα από τα επίπεδά του . Και τα δύο προτάθηκαν για επίλυση προβλημάτων που αφορούν την αυτοκινητοβιομηχανία. Η μείωση του αριθμού των παραμέτρων κάνει εφικτή την αναγνώριση αντικειμένων σε πραγματικό χρόνο και επιτυγχάνει μεγαλύτερη ενεργειακή αποδοτικότητα. Σε σύγκριση με το AlexNet[?], επιτυγχάνει την ίδια ακρίβεια με 50x μικρότερο μέγεθος παραμέτρων. Η είσοδος που δέχεται είναι το ήδη επεξεργασμένο αποτέλεσμα του SqueezeNet.



Σχήμα 2.1: Η σειρά επεξεργασίας του SqueezeDet για εντοπισμό αντικειμένων. Ένα CNN π.χ. SqueezeNet εξάγει χαρακτηριστικά από την εικόνα εισόδου και τα δίνει ως είσοδο στο επίπεδο ConvDet. Με τη σειρά του, το επίπεδο ConvDet υπολογίζει τα ορθογώνια περιβλήματα γύρω από τα ομοιόμορφα κατανεμημένα $W \times H$ κέντρα των πιθανών αντικειμένων. Κάθε ορθογώνιο περίβλημα σχετίζεται με 1 σκορ εμπιστοσύνης και C υπό συνθήκη πιθανότητες. Κατά το επίπεδο *Filtering* κρατούμε τα N περιβλήματα με τα κυρίαρχα σκορ εμπιστοσύνης και χρησιμοποιούνται αλγόριθμοι *NMS* για τη λήψη των τελικά εντοπισμένων αντικειμένων.

Η αρχιτεκτονική του SqueezeDet, του επιτρέπει να προτείνει την ορθογώνια περιοχή μέσα στην οποία εντοπίζει ένα αντικείμενο σε μία εικόνα αλλά και να το κατηγοριοποιεί ταυτόχρονα. Ουσιαστικά αποτελείται από ένα επίπεδο συνελίξεων το οποίο προτείνει τις περιοχές ύπαρξης αντικειμένων και ένα NMS (Non-Maximum Suppression) φίλτρο για υπολογισμό της πιθανότητας ύπαρξης κάποιου αντικειμένου στις προτεινόμενες περιοχές. Το φίλτρο εφαρμόζεται σε όλη την έξοδο του ConvNet και η πιθανότητα υπολογίζεται από τον τύπο:

$$max\{Pr\left(class_C|Object\right)\} * Pr(Object) * IOU_{truth}^{pred}$$

Το SqueezeNet με τη σειρά αποτελεί ιδανική υλοποίηση του επιπέδου συνελίξεων του SqueezeDet γιατί στοχεύει στον μικρό αριθμό παραμέτρων. Επίσης κατά τη σχεδιασή του δόθηκε περισσότερη έμφαση στον διανυσματικό χώρο των βαρών με δεδομένη ακρίβεια και όχι το ανάποδο. Αυτό ήταν που οδήγησε στις παρακάτω τρεις στρατηγικές για τη μείωση του αριθμού των παραμέτρων:

1. Αντικατάσταση των 3×3 φίλτρων με 1×1.
2. Μείωση του αριθμού καναλιών στα 3×3 φίλτρα, με χρήση επιπλέον επιπέδων (*squeeze layers*).
3. Η υποδειγματοληψία γίνεται στα τελευταία layers του δικτύου.

Με εφαρμογή αυτών των τριών στρατηγικών και με υλοποίηση της λογικής "Network in Network" των ResNet [?] και GoogleNet [?] το νευρωνικό αποτελείται από μικρότερες

οντότητες, οι οποίες ονομάζονται *fire modules*. Κάθε τέτοια οντότητα όπως φαίνεται στο Σχήμα 1.2 ορίζεται από τις παραμέτρους:

- $s_{1\times1}$ ο αριθμός φίλτρων στο *squeeze layer*.
- $e_{1\times1}$ ο αριθμός 1×1 φίλτρων στο *expand layer*.
- $e_{3\times3}$ ο αριθμός 3×3 φίλτρων στο *expand layer*.

Προκειμένου να επιτευχθεί η στρατηγική 2 απαιτείται $s_{1\times1} < (e_{1\times1} + e_{3\times3})$. Έπειτα ολόκληρη η αρχιτεκτονική αποτυπώνεται καλύτερα στο Σχήμα 1.2. Επιπλέον στην είσοδο των 3×3 φίλτρων γίνεται *zero-padding* κατά 1 εικονοστοιχείων στο σύνορο.

Η εκπαίδευση του δικτύου γίνεται με *Stochastic Gradient Descent* χρησιμοποιώντας την ίδια συνάρτηση κόστους που χρησιμοποιεί το δίκτυο YOLO[?]. Κατά την εκκίνηση το learning rate είναι 0.04, το οποίο μειώνεται κατά την πάροδο των εποχών. Στο τέλος της εκπαίδευσης μπορεί να εφαρμοστεί και η τεχνική Deep Compression στο SqueezeNet με την οποία χρησιμοποιώντας μόνο 0.66 MB επιτυγχάνεται ακρίβεια 80.3% στα 6 bit στο σύνολο δεδομένων του ImageNet. Η συνάρτηση κόστους εκπαιδεύει και τα 2 επίπεδα της αρχιτεκτονικής:

$$\begin{aligned}
&\frac{\lambda_{bbox}}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} I_{ijk}[(\delta x_{ijk} - \delta x_{ijk}^{G})^2 + (\delta y_{ijk} - \delta y_{ijk}^{G})^2 \\
&\qquad + (\delta w_{ijk} - \delta w_{ijk}^{G})^2 + (\delta h_{ijk} - \delta h_{ijk}^{G})^2] \\
&+ \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \frac{\lambda_{conf}^{+}}{N_{obj}} I_{ijk}(\gamma_{ijk} - \gamma_{ijk}^{G})^2 + \frac{\lambda_{conf}^{-}}{WHK - N_{obj}} \bar{I}_{ijk} \gamma_{ijk}^{2} \\
&\qquad + \frac{1}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \sum_{c=1}^{C} I_{ijk} l_c^{G} \log(p_c).
\end{aligned}$$

Το πρώτο κομμάτι είναι για την εύρεση του περιβλήματος των αντικειμένων. Το σημείο $(\delta x_{ijk}, \delta y_{ijk}, \delta w_{ijk}, \delta h_{ijk})$ αφορά τις σχετικές συντεταγμένες του πιθανού περιβλήματος ως προς το σημείο (i, j). Το δεύτερο κομμάτι (η γραμμή με το δεύτερο τριπλό άθροισμα) αφορά την παλινδρόμηση για το σκορ εμπιστοσύνης. Έπειτα, το τρίτο κομμάτι αφορά την cross-entropy για κατηγοριοποίηση των αντικειμένων σε κλάσεις. Η ακρίβεια του αλγορίθμου υπολογίστηκε πάνω στο σύνολο δεδομένων του KITTI και δίνεται με το κριτήριο (mAP) για τους πεζούς, τα αυτοκίνητα και τους ποδηλάτες.

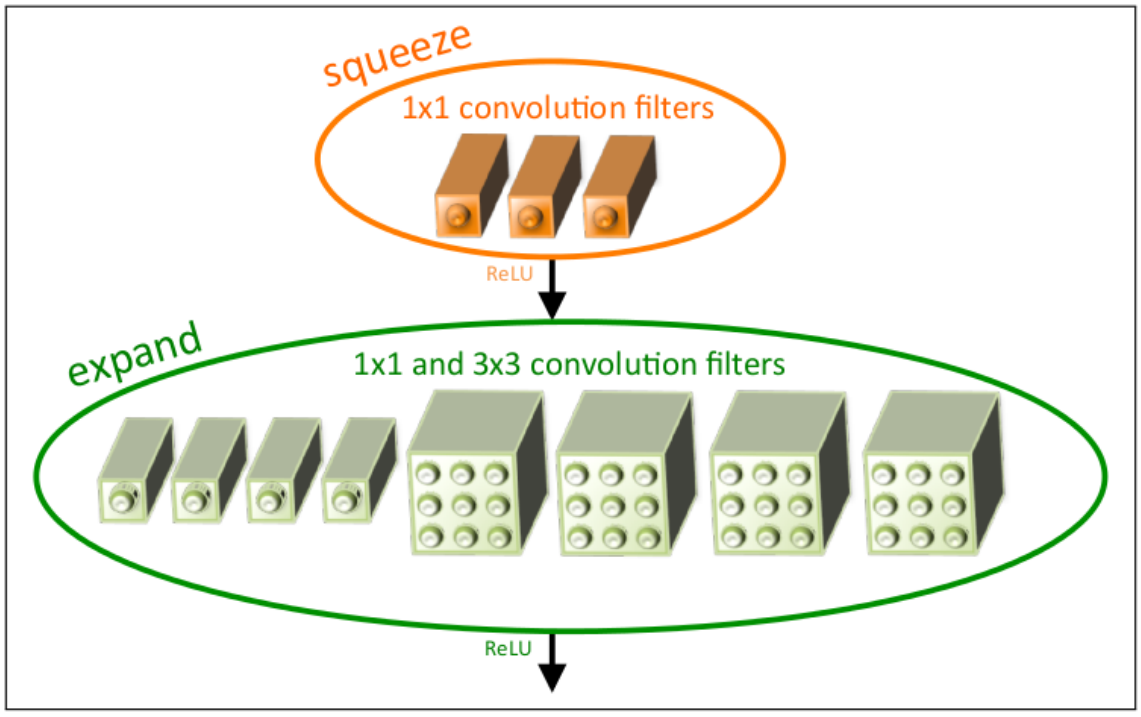
Μέθοδος	Car			Cyclist			Pedestrian		
SqueezeDet	90.2	84.7	73.9	82.9	75.4	72.1	77.1	68.3	65.8
SqueezeDet+	90.4	87.1	78.9	87.6	80.3	78.1	81.4	71.3	68.5

Μέθοδος	Model size (MB)	mAP
SqueezeDet	7.9	76.7
SqueezeDet+	26.8	80.4

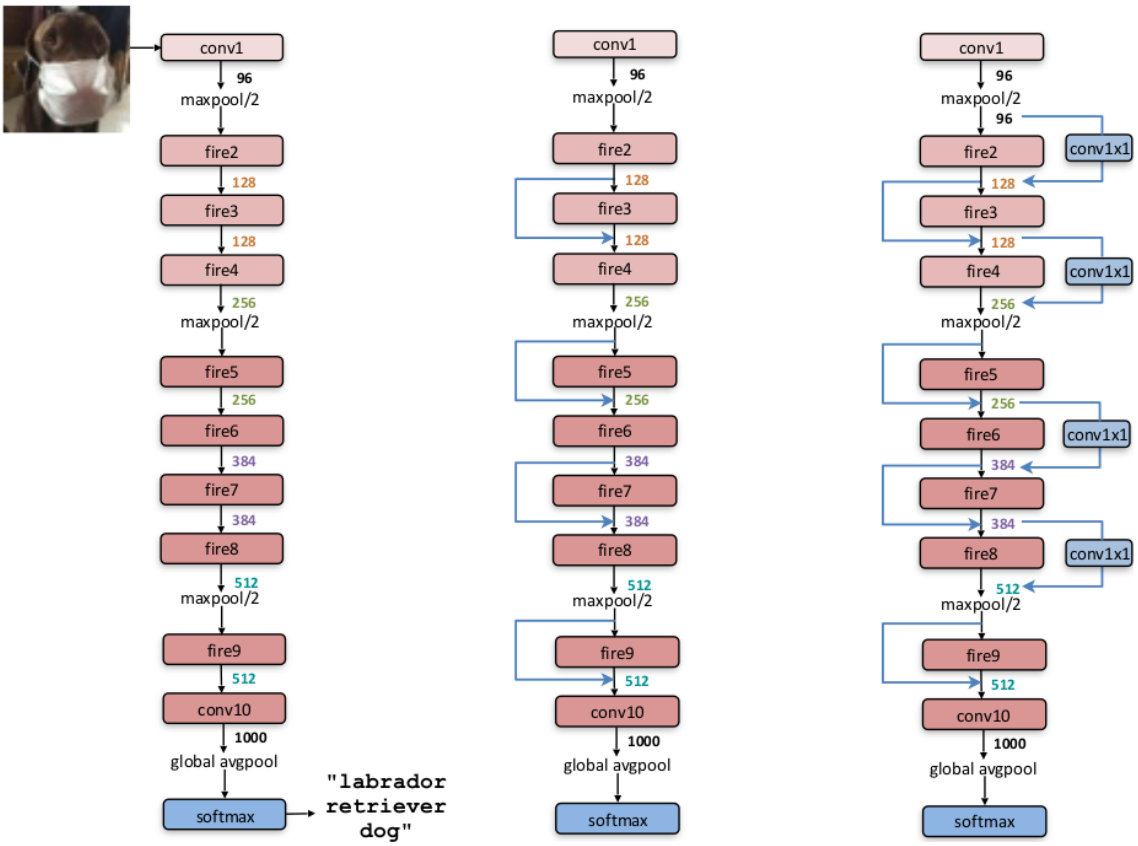
Η απαίτηση μνήμης παρά τον παραπάνω πίνακα μπορεί να μειωθεί περαιτέρω. Με τη χρήση τεχνικών συμπίεσης και επηρεάζοντας τις υπερπαραμέτρους επιτυγχάνεται παραπάνω από 95×

συμπίεση κρατώντας την ίδια ευαισθησία. Ταυτόχρονα κερδίζει σε ταχύτητα και κατανάλωση ενέργειας. Μπορεί να φτάσει έως και τα 57.2 FPS στο KITTI, ενώ η ενισχυμένη εκδοσή του (SqueezeDet+) τα 32.1 FPS. Ο μειωμένος αριθμός παραμέτρων οδηγεί σε λιγότερη προσπέλαση μνήμης και οπότε λιγότερη χρήση της DRAM. Αυτή με τη σειρά της οδηγεί σε χαμηλότερη κατανάλωση ενέργειας είναι από 1.4 έως 4.0J/frame στο KITTI ανάλογα την έκδοση του SqueezeDet.

Ο χρόνος εκτέλεσης μειώνεται και αυτός με τη μείωση του αριθμού των παραμέτρων του δικτύου. Αν και στο ίδιο το έγγραφο που πρωτοπαρουσιάζεται το δίκτυο δε γίνεται λόγος για αυτό, η ενέργεια που καταγράφεται σε ενσωματωμένες συσκευές στο [?] είναι 26.37 J/frame στην πλατφόρμα του Nexus 5 για την εκτέλεση του SqueezeNet μόνο. Μάλιστα η παράλληλη υλοποίησή του οδηγεί σε ακόμα μικρότερη κατανάλωση ενέργειας. Για την ίδια πλατφόρμα έχει 249.47X λιγότερες ενεργειακές απαιτήσεις. Τέλος, η απαίτηση μνήμης, η κατανάλωση ενέργειας και η χρονική απόκρισή του νευρωνικού, το χρήζουν κατάλληλο για ενσωματωμένα συστήματα.



Σχήμα 2.2: Μικροσκελής όψη της αρχιτεκτονικής SqueezeNet. Στην εικόνα φαίνεται το *Fire module*, το οποίο είναι η βασική οντότητα όλου του SqueezeNet. Σε αυτό το παράδειγμα, $s_{1x1} = 3, e_{1x1} = 4, e_{3x3} = 4$.



Σχήμα 2.3: Μακροσκελής όψη της αρχιτεκτονικής SqueezeNet. Τα δίκτυα που παρουσιάζονται είναι τα: απλό SqueezeNet (αριστερά), SqueezeNet με απλό bypass (μέση), SqueezeNet με σύνθετο bypass (δεξιά). Η σύνδεση προηγουμένων επιπέδων με το επόμενο και όχι μόνο του αμέσως προηγούμενου ωφελεί την ακρίβεια του δικτύου.

2.3 YOLO[?]

Το δίκτυο αυτό προτάθηκε ως μια λύση για το πρόβλημα της πραγματικού χρόνου αναγνώρισης αντικειμένων διατηρώντας όσο το δυνατόν μεγαλύτερη μέση ακρίβεια. Χαρακτηρίζεται από το όνομά του *You Only Look Once* που υποδηλώνει πως τόσο κατά τον εντοπισμό/αναγνώρισή όσο και κατά την εκπαίδευση, το δίκτυο λαμβάνει την εικόνα εισόδου μία φορά και δεν την επεξεργάζεται ξανά μετά την είσοδο κατά την εκτέλεση του. Οπότε, λαμβάνεται υπόψιν και η κατά το δυνατόν γρηγορότερη εκπαίδευση του. Οι ικανότητες του πέρα από την ταχύτητα και την ακρίβειά του είναι και ο ταυτόχρονος εντοπισμός πολλών διαφορετικών αντικειμένων σε μία εικόνα και η χωροθέτηση τους.

Αρχικά για την επιλογή της αρχιτεκτονικής οι συγγραφείς θεώρησαν πως η αναγνώριση αντικειμένων μπορεί να θεωρηθεί ως πρόβλημα απλής παλινδρόμησης. Επειδή η επεξεργασία εισόδου γίνεται μία φορά, η αρχιτεκτονική του νευρωνικού αποτελείται από επίπεδα συνέλιξης, υποδειγματοληψίας και από δύο ολικά συνδεδεμένα επίπεδα στο τέλος. Τα επίπεδα αυτά είναι συνδεδεμένα εν σειρά. Η αναγνώριση της κλάσης του αντικειμένου αλλά και του περιβλήματος του γίνεται την ίδια χρονική στιγμή. Για αυτό το αποτέλεσμα του νευρωνικού είναι ένα πλέγμα $S \times S$ κελιών που κάθε κελί χωρίζει ισόποσα την εικόνα. Επίσης προβλέπει περιβλήματα αντικειμένων και για το κάθε ένα από αυτά ένα σκορ εμπιστοσύνης. Το σκορ εμπιστοσύνης δείχνει την πιθανότητα το περίβλημα περιέχει ένα αντικείμενο. Όλα τα περι-

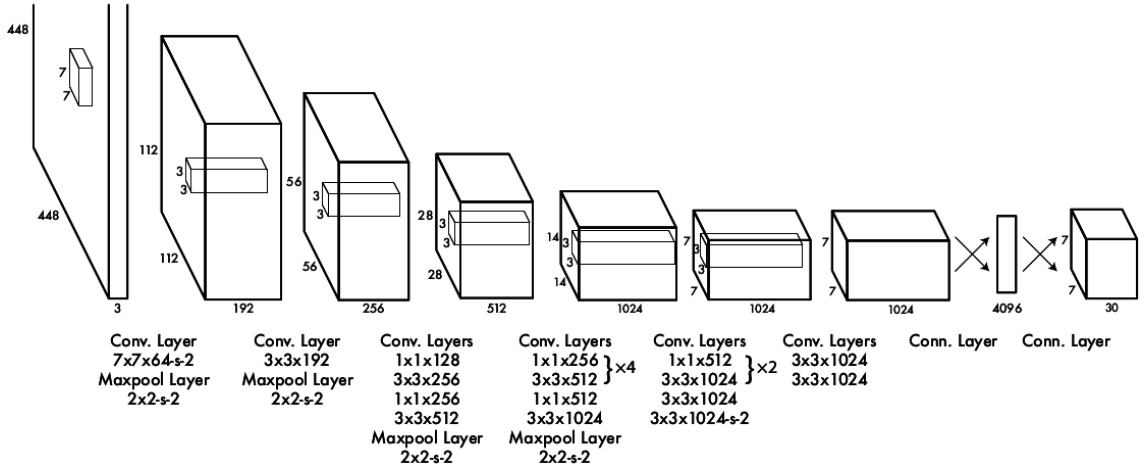
βλήματα του δικτύου YOLO είναι ορθογώνια. Η πληροφορία που περιγράφει ένα περίβλημα είναι $(x, y, w, h, \text{σκορ εμπιστοσύνης})$

(x, y): το κέντρο του περιβλήματος

$w =$ πλάτος ορθογωνίου / πλάτος εικόνας

$h =$ ύψος ορθογωνίου / ύψος εικόνας

Επιπλέον κάθε κελί έχει και C πιθανότητες $PrClass_i|Object,\ i = 1, .., C$. Οπότε συνολικά η έξοδος του νευρωνικού είναι ένας τανυστής $S \times S \times (B \cdot 5 + C)$. Η αρχιτεκτονική του φαίνεται και αναλυτικά στο Σχήμα 1.4.



Σχήμα 2.4: Παράδειγμα της αρχιτεκτονικής YOLO για εικόνα εισόδου 224×224 όπου $S = 7, B = 2, C = 20$. Το YOLO έχει 24 συνελεκτικά επίπεδα ακολουθούμενα από 2 ολικά συνδεδεμένα επίπεδα. Η χρήση 1×1 συνελεκτικών επιπέδων μπορεί να μειώσει τον χώρο των χαρακτηριστικών από τα προηγούμενα επίπεδα. Επίσης τα συνελεκτικά επίπεδα αρχικά εκπαιδεύονται στο ImageNet χρησιμοποιώντας τη μισή ανάλυση εικόνας (224×224 εικόνα εισόδου) και μετά χρησιμοποιώντας τη διπλάσια για τον εντοπισμό αντικειμένων.

Ως συναρτήσεις ενεργοποίησης, αντί για τις πλέον διαδεδομένες $ReLU$ χρησιμοποιείται μια παραπλήσια μορφή:

$$\phi(x) = \begin{cases} x, & x > 0 \\ 0.1x, & \text{αλλιώς} \end{cases}$$

Η εκπαίδευση γίνεται με κύριο σκοπό την βελτιστοποίηση του τετραγωνικού σφάλματος της εξόδου. Ως συνάρτηση απωλειών χρησιμοποιείται η συνάρτηση:

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

Η εκπαίδευση γίνεται αρχικά με *learning rate* 10^{-2} έπειτα 10^{-3} και τελικά 10^{-4}. Επιπλέον χρησιμοποιείται η τεχνική *dropout* [?] με ποσοστό 50%. Όλα αυτά οδηγούν σε ακρίβεια 63.4 mAP στο σύνολο δεδομένων PASCAL VOC 2007. Η ταχύτητα αναγνώρισης αντικειμένων είναι 150 fps στη κάρτα γραφικών TITAN X της nvidia. Αντίστοιχα ο χρόνος που απαιτείται για την εκπαίδευση του νευρωνικού είναι μία εβδομάδα με το ίδιο hardware.

Το YOLO στην πρώτη έκδοσή του πέτυχε παραπάνω από δύο φορές τη μέση ακρίβεια των συστημάτων αναγνώρισης αντικειμένων με καθυστέρηση $25ms$. Αυτό του επιτρέπει να εισαχθεί και στο τέλος του δικτύου *Fast R-CNN* για μια διορθωμένη αναγνώριση αντικειμένων στο φόντο της εικόνας. Στην παρουσίαση του δικτύου δεν γίνεται λόγος για απαίτηση μνήμης του συστήματος ωστόσο εκτελώντας τον αλγόριθμο από το [?], φαίνεται στο [?] ότι όσο λιγότερη μνήμη υπάρχει διαθέσιμη τόσο πιο αργά εκτελείται ο αλγόριθμος. Από το Σχήμα 1.4 υπολογίζεται πως για τον τανυστή εξόδου $7 \times 7 \times 30$ απαιτείται μνήμη ίση με 808 MB για βάρη των 32-bit. Αυτός μας δείχνει πως το YOLO είναι ένα βήμα προς την εισαγωγή των νευρωνικών στα ενσωματωμένα, ωστόσο πάλι απαιτεί αρκετή μνήμη και έχει περιορισμούς οι οποίοι δεν το χρήζουν κατάλληλο για κρίσιμες εφαρμογές. Παραδείγματα αδυναμιών του δικτύου είναι ότι τα περιβλήματα δεν υπολογίζονται πολλές φορές σωστά. Μια άλλη περίπτωση είναι ότι ενώ προσπαθεί να γενικευθεί για μικρά και μεγάλα αντικείμενα, όταν αυτά βρίσκονται σε εικόνες με διαφορετικές αναλογίες διαστάσεων ή επαναλαμβάνονται σε ένα γκρουπ με μικρές διαστάσεις αποτυγχάνει τον εντοπισμό τους.

2.4 YOLO9000 [?]

Η βελτίωση αυτή του YOLO μπορεί να εντοπίσει αντικείμενα σε μια εικόνα από έως και 9000 διαφορετικές κατηγορίες. Μάλιστα, οι διαδικασίες μάθησης και εντοπισμού γίνονται πιο γρήγορα προσφέροντας και μεγαλύτερη ακρίβεια. Χαρακτηριστικό είναι ο λόγος 76.8 mAP στα 67 fps. Ωστόσο αντί να επεκτείνουν τις διαστάσεις του δικτύου, οι συγγραφείς προτίμησαν να το απλοποιήσουν δίνοντας τη δυνατότητα με διάφορες τεχνικές να διευκολύνουν την εκμάθησή του. Τα χαρακτηριστικά του δικτύου φαίνονται στον Πίνακα 1.1 και επεξηγούνται παρακάτω.

batch norm: Ο μέσος όρος αφαιρείται και διαιρείται με την τυπική απόκλιση όχι μόνο στην αρχή του δικτύου, αλλά και μέσα στο δίκτυο ανά κάποιες εισόδους.

hi-res classifier: Κατά την εκπαίδευση για κάποιες εποχές χρησιμοποιείται μεγαλύτερη ανάλυση της εικόνας 448×448 αντί 224×224.

convolutional (with anchor boxes): Χρησιμοποιούνται διαφορετικά περιβλήματα από ότι στην πρώτη έκδοση του νευρωνικού. Τα περιβλήματα αυτά είναι τα ίδια που περιγράφονται στο Faster R-CNN. Η διαφορά είναι ότι πλέον το δίκτυο εντοπίζει πάνω από

	YOLO								YOLOv2
batch norm		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier			✓	✓	✓	✓	✓	✓	✓
convolutional				✓	✓	✓	✓	✓	✓
anchor boxes				✓	✓				
new network					✓	✓	✓	✓	✓
dimension priors						✓	✓	✓	✓
location prediction						✓	✓	✓	✓
passthrough							✓	✓	✓
multi-scale								✓	✓
hi-res detector									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

Πίνακας 2.1: Χαρακτηριστικά του δικτύου YOLO9000.

1000 κουτιά ανά εικόνα και αποκτά βελτιωμένο recall. Αυτή η δυνατότητα αντικαθιστά τα ολικά συνδεδεμένα επίπεδα στο τέλος του δικτύου.

dimension priors: Τα *anchor boxes* έχουν το πρόβλημα πως οι αρχικές συνθήκες των βαρών εντοπισμού δημιουργούν μεγάλο πρόβλημα για την εκπαίδευση του συστήματος. Οπότε χρησιμοποιείται ο k-means για να τις αποφασίσουμε με k = 5 και μετρική:

$$d(box, centroid) = 1 - IOU(box, centroid)$$

location prediction: Η εύρεση της θέσης του περιβλήματος παρουσιάζει αστάθεια. Για την αντιμετώπιση αυτού χρησιμοποιείται η μέθοδος άμεσου εντοπισμού θέσης με τη χρήση ενός ασαφούς ελεγκτή.

passthrough: Απλά ένα επίπεδο για την σύνδεση των δύο τελευταίων διότι κατά την εκπαίδευση αντικαθίσταται το τελευταίο επίπεδο συνέλιξης με 3χ3 συνελεκτικά επίπεδα με 1024 φίλτρα το καθένα ακολουθούμενο από ένα τελευταίο 1χ1 συνελεκτικό επίπεδο.

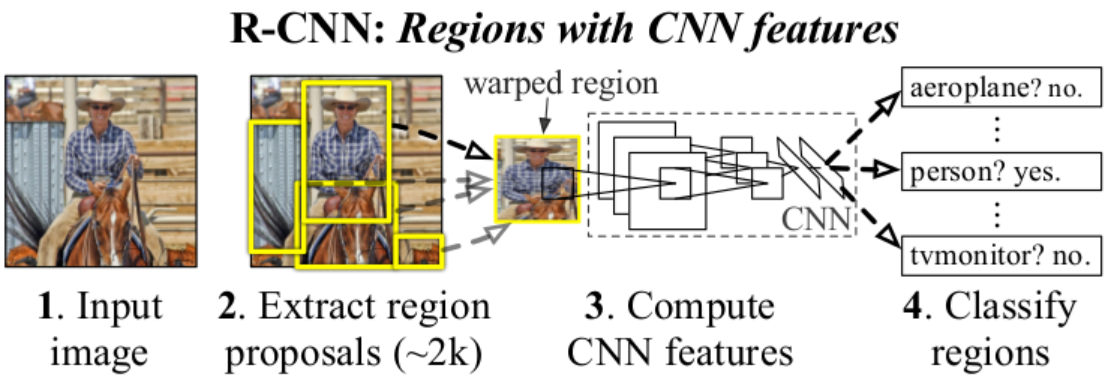
multi-scale: Το μέγεθος της εικόνας αυξάνεται ή ελαττώνεται κατά την εκπαίδευση.

hi-res detector: Ο εντοπισμός αντικειμένων γίνεται χρησιμοποιώντας καλύτερη ανάλυση της εικόνας από ότι το υπόλοιπο δίκτυο.

Η ταχύτητα του δικτύου οφείλεται στη χρήση τεχνικών του GoogleNet[**?**]:

- Network In Network
- Global pooling
- 3×3 και 1×1 συνελεκτικά φίλτρα.

Η δυνατότητα εντοπισμού πολλών κλάσεων οφείλεται στη χρήση ενός δέντρου που εμπεριέχει τις κλάσεις αυτές. Επομένως κάθε αντικείμενο έχει πολλές ετικέτες. Έτσι η τελική κλάση θα είναι κάποιο από τα φύλλα του δέντρου. π.χ. αν μια εικόνα δείχνει ένα "golden retreiver" τότε σίγουρα είναι αντικείμενο τύπου ζώου και τύπου σκύλου. Με αυτό τον τρόπο δεν χρειάζεται να διαχωριστούν όλες οι κλάσεις μεταξύ όλων, αλλά αρκεί να εντοπιστεί ότι το αντικείμενο είναι τύπου 1 ή 2 και μετά να αναζητηθεί η σχέση μεταξύ των υπο-τύπων του έως ότου ο αλγόριθμος καταλήξει σε κάποιο φύλλο του δέντρου. Μάλιστα κατά την εκπαίδευσή του στο ImageNet για 1000 κατηγορίες αντικειμένων προστίθεται επιπλέον θόρυβος (διαφόρων ειδών) στις εικόνες ώστε να ενισχυθεί η ικανότητα εντοπισμού. Τέλος, ενώ η



Σχήμα 2.5: Πανόραμα του συστήματος εντοπισμού αντικειμένων του R-CNN. Το σύστημα λαμβάνει μια εικόνα (1), εξάγει 2000 προτεινόμενα περιβλήματα (2), υπολογίζει τα χαρακτηριστικά για κάθε ένα από τα περιβλήματα χρησιμοποιώντας ένα CNN (3), κατηγοριοποιεί κάθε περιοχή εντός κάποιου από τα προτεινόμενα περιβλήματα χρησιμοποιώντας linear SVM (4). Το R-CNN πετυχαίνει mAP ίσο με 53.7% στο PASCAL VOC 2010 και 31.4 στο *ILSVRC2013*.

πρώτη έκδοση του YOLO ήταν ένα πρώτο βήμα για την χρήση νευρωνικών σε ενσωματωμένα συστήματα, η δεύτερη αποτελεί ένα ικανοποιητικό και αρκετά αξιόπιστο σύστημα για την εφαρμογές όπως η αυτόνομη οδήγηση.

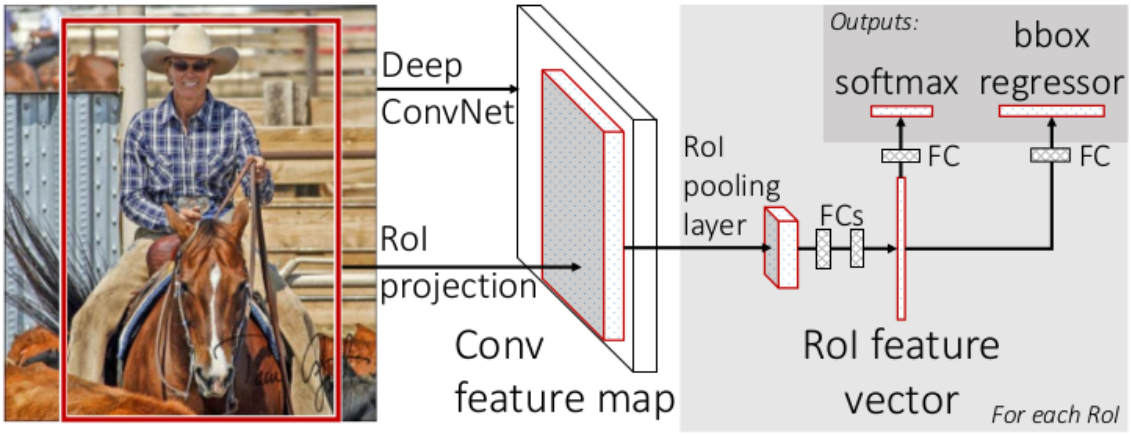
2.5 R-CNN [?]

Το R-CNN ήταν από τα πρώτα δίκτυα που εμπνεύστηκαν από το AlexNet το έτος 2013. Οπότε και χρησιμοποιεί κομμάτια αυτού του δικτύου με μικρές αλλαγές. Ωστόσο προστίθεται παραπάνω λειτουργικότητα για την αναγνώριση πολλών αντικειμένων στην εικόνα και όχι πλέον μόνο για την κατηγοριοποίηση όλης της εικόνας ως ένα αντικείμενο. Για να το πετύχει αυτό το R-CNN χρησιμοποιεί τη τεχνική *Selective Search* [**?**]. Με αυτή μπορεί και εντοπίζει κάθε αντικείμενο στην εικόνα και το περιβάλει με ένα ορθογώνιο. Η εναλλακτική στρατηγική θα ήταν να χρησιμοποιείται ένα κυλιόμενο παράθυρο στην εικόνα για τον εντοπισμό των αντικειμένων. Η τελευταία αυτή τεχνική θα είχε μεγάλο υπολογιστικό κόστος έναντι της *Selective Search*.

Μετά τον εντοπισμό των πιθανών περιοχών ύπαρξης αντικειμένων, οι περιοχές αυτές διαστασιοποιούνται σε τετράγωνες και χρησιμοποιείται μια επεξεργασμένη έκδοση του AlexNet για την κατηγοριοποίηση του αντικειμένου.

Στο τελευταίο επίπεδο του νευρωνικού υπάρχει ένα SVM το οποίο τελικά κατηγοριοποιεί το αντικείμενο σε κάποια κλάση (βήμα 4 στο Σχήμα 1.5). Ωστόσο, όπως έχει αναφερθεί το R-CNN, αφού εκτελέσει το νευρωνικό, υλοποιεί και ένα ακόμη βήμα για βελτιστοποίηση των ορθογώνιων περιβλημάτων. Ουσιαστικά θεωρεί το πρόβλημα εύρεσης περιβλήματος ως πρόβλημα απλής παλινδρόμησης για να δώσει καλύτερα-στενότερα όρια στο περίβλημα. Έτσι στο τελευταίο βήμα λαμβάνει ως εισόδους τις περιοχές (τα ορθογώνια περιβλήματα) που υπάρχουν αντικείμενα και τις επιστρέφει βελτιωμένες.

Προφανώς το R-CNN δεν μπορεί να εκτελεστεί από κάποια ενσωματωμένη συσκευή, διότι χρησιμοποιεί το AlexNet. Ωστόσο, οι διάδοχοί του έχουν βελτιωθεί ώστε να είναι εφικτή η εκτέλεσή τους σε κάποια ενσωματωμένη συσκευή τόσο από άποψη μνήμης, όσο και από χρόνο εκτέλεσης.



Σχήμα 2.6: Η αρχιτεκτονική Fast-RCNN. Η εικόνα εισόδου και οι πολλαπλές περιοχές ενδιαφέροντος (RoI) οι οποίες είναι είσοδοι σε ένα ενοποιημένο CNN. Κάθε RoI συλλέγεται σε ένα χάρτη χαρακτηριστικών δεδομένου μεγέθους και αντιστοιχίζεται σε ένα διάνυσμα χαρακτηριστικών χρησιμοποιώντας ολικά συνδεδεμένα επίπεδα (FCs στο Σχήμα 1.6). Το δίκτυο έχει δύο διανύσματα χαρακτηριστικών σε κάθε RoI. Ένα που προέρχεται από την έξοδο του Softmax και ένα από την γραμμική παλινδρόμηση. Τέλος το κομμάτι του ενοποιημένου δικτύου χρησιμοποιεί κοινή συνάρτηση απωλειών και όλα τα κομμάτια του εκπαιδεύονται μαζί.

2.6 Fast R-CNN [?]

Οι λόγοι για τους οποίους είναι αργό το R-CNN είναι δύο.

1. Η χρήση του εμπρόσθιου περάσματος του AlexNet για κάθε πιθανού ορθογώνιου περιβλήματος από την Selective search το οποίο απαιτεί να εκτελεστεί το εμπρόσθιο πέρασμα του AlexNet περίπου 2000 φορές.

2. Εκπαιδεύει τρία μοντέλα ξεχωριστά: το CNN για την εύρεση των χαρακτηριστικών της εικόνας, το SVM και το μοντέλο της απλής παλινδρόμισης για την βελτίωση των ορθογωνίων περιβλημάτων στο τέλος.

Το Fast R-CNN υλοποιήθηκε με σκοπό να λύσει αυτά τα 2 προβλήματα και τελικά να επιταχύνει το R-CNN. Το πρώτο λύθηκε εκτελώντας τα βήματα σε διαφορετική σειρά. Πρώτα εκτελείται το εμπρόσθιο πέρασμα του CNN σε όλη την εικόνα και μετά γίνεται pooling σε κάθε πιθανό περίβλημα αντικειμένου (αυτό καλείται Region of Interest Pooling -RoIPool). Οπότε από περίπου 2000 εκτελέσεις του εμπρόσθιου περάσματος, πλέον απαιτείται μόνο μία. Το δεύτερο λύθηκε βάζοντας και τα τρία μοντέλα να εκτελούνται μαζί σε ένα κοινό δίκτυο. Η παλινδρόμηση γίνεται παράλληλα με την κατηγοριοποίηση η οποία πλέον δε γίνεται με SVM αλλά με *Softmax classifier* λαμβάνοντας ως είσοδο το αποτέλεσμα του επίπεδου RoIPool.

Η εκπαίδευση του δικτύου γίνεται 2.7 φορές πιο γρήγορα και οι χρόνοι αναγνώρισης αντικειμένων σε εικόνα ξεκινούν από 0.10 sec, ανάλογα με το μέγεθος της εικόνας. Η ακρίβεια (κριτήριο mAP) ελαφρώς αυξάνεται παρά τις αλλαγές στο δίκτυο. Παρόλα αυτά οι χρόνοι εκτέλεσης συνεχίζουν να είναι απαγορευτικοί για ενσωματωμένα συστήματα. Επίσης δε γίνεται λόγος για κατανάλωση ενέργειας, διότι εκτελείται σε GPU (*Nvidia K40 GPU overclocked to 875 MHz*). Επίσης αποφεύγεται η χρήση μεγάλων εικόνων, ώστε να μπορεί το δίκτυο να εμπεριέχεται ολόκληρο σε μια GPU και να μην απαιτείται πλέον χρήση του σκληρού δίσκου για caching.

2.7 Faster R-CNN [?]

Η επόμενη -χρονικά- πρόταση ήταν το Faster R-CNN το οποίο προτάθηκε από το ερευνητικό τμήμα της *Microsoft*. Το καινούριο αυτό δίκτυο βασίστηκε στο γεγονός ότι ο κύριος φόρτος εργασίας του Fast R-CNN γινόταν πλέον στον αλγόριθμο *Selective Search*. Ταυτόχρονα, το βασικό δίκτυο επανα-υπολόγιζε χαρακτηριστικά (features) στις εικόνες στο πρώτο layer, τα οποία υπολόγιζε και η *Selective Search* για να βρει τα περιβλήματα των αντικειμένων.

Επομένως η πρόταση για το Faster R-CNN ήταν η χρήση ενός νευρωνικού δικτύου RPN (*Region Proposal Network*) το οποίο θα κάνει τη δουλειά της *Selective Search*. Αυτό αποτελείται από δύο κομμάτια: ένα ολικά συνδεδεμένο επίπεδο που προτείνει περιοχές και τον ανιχνευτή του Fast R-CNN που χρησιμοποιεί τις προτεινόμενες περιοχές του πρώτου. Με αυτό τον τρόπο όλο το αρχικό R-CNN είναι πλέον ένα ενοποιημένο δίκτυο. Επιπλέον κατά τον εντοπισμό περιοχών αντικειμένων το RPN χρησιμοποιεί τη τεχνική της 'προσοχής'[?] η οποία χρησιμοποιείται πλέον σε αρκετά νευρωνικά δίκτυα πέραν των συνελεκτικών για αύξηση της ακρίβειας.

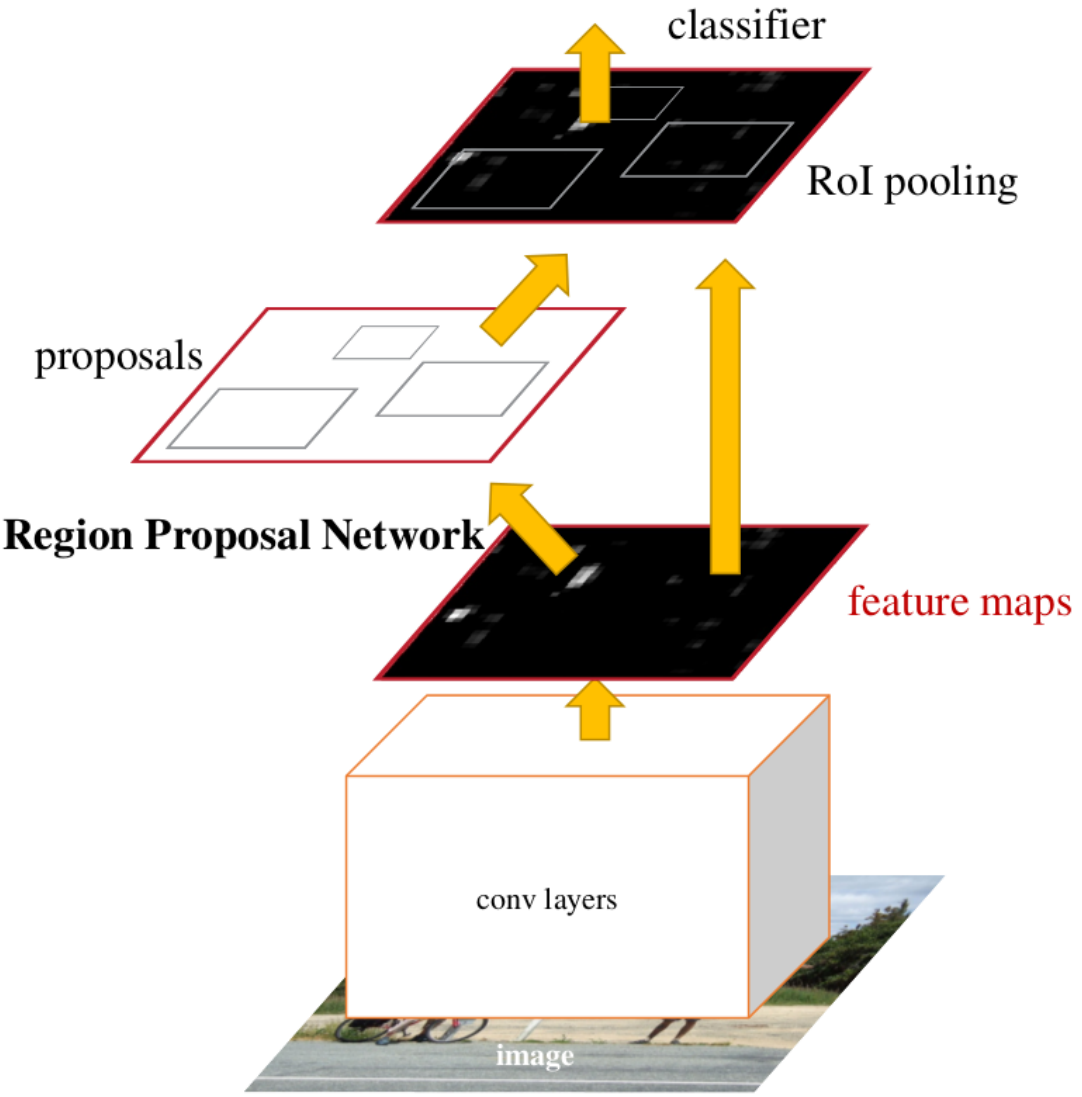
Αξιοσημείωτο είναι ότι το RPN χρησιμοποιείται από το SqueezeDet, αλλά αντί ολικά συνδεδεμένου δικτύου στο πρώτο χρησιμοποιεί συνελεκτικό επίπεδο. Και στις δύο περιπτώσεις υπάρχει ανεξαρτησία στη μετακίνηση των αντικειμένων.

Ουσιαστικά αυτό που συμβαίνει είναι ότι πάνω από τα αποτελέσματα του πρώτου επιπέδου τύπου CNN περνάει ένα παράθυρο το οποίο διαλέγει για κάθε εικονοστοιχείο της εικόνας k πιθανά παράθυρα διαφορετικού μήκους και πλάτους. Οι χρόνοι εκτέλεσης είναι από 5 fps εώς 17 fps ανάλογα με τον τρόπο υλοποίησης του RPN σε GPU NVIDIA Tesla K40. Επίσης στο σύνολο δεδομένων *PASCAL VOC 2012* πετυχαίνει ακρίβεια με κριτήριο mAP 59.9%. Η μνήμη ωστόσο συνεχίζει να αποτελεί πρόβλημα. Για αυτό δεν κρίνεται κατάλληλο για ενσωματωμένα συστήματα.

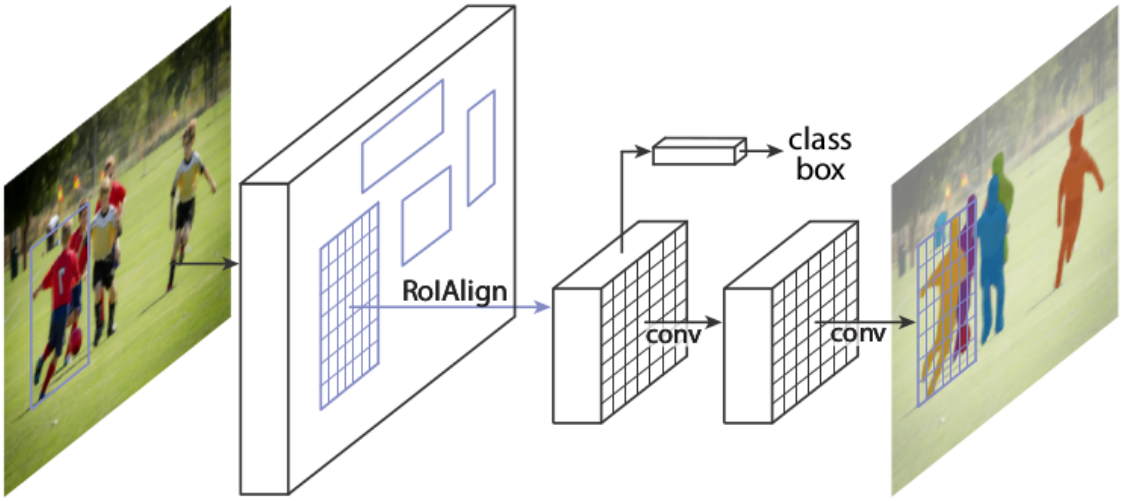
2.8 Mask R-CNN [?]

Η τελευταία πρόταση βασισμένη στο R-CNN είναι το Mask R-CNN το οποίο προτάθηκε από την ερευνητική ομάδα της *Facebook*. Το νευρωνικό δίκτυο αυτό είναι το Faster R-CNN με τη διαφορά πως αντί για ορθογώνια περιβλήματα των αντικειμένων πλέον εντοπίζει τα πραγματικά περιβλήματα στην εικόνα. Δηλαδή το σύνορο ενός αντικειμένου με την υπόλοιπη εικόνα δεν είναι πλέον ένα ορθογώνιο, αλλά ένα αφηρημένο σχήματα που το περιβάλλει με ακρίβεια εικονοστοιχείων. Για να επιτευχθεί αυτό χρησιμοποιείται ένας παράλληλος κλάδος του δικτύου, ο οποίος είναι υπεύθυνος για την κατηγοριοποίηση των εικονοστοιχείων σε αντικείμενα. Στο τέλος αυτού του κλάδου επιστρέφονται πίνακες οι οποίοι έχουν 1 στα εικονοστοιχεία στα οποία εντοπίζεται αντικείμενο και 0 σε αυτά που δεν εντοπίζεται. Κάθε πίνακας συνδέεται με ένα αντικείμενο σε ένα RoI. Αυτοί οι πίνακες είναι και οι μάσκες από όπου πήρε και το δίκτυο το ονομά του. Τέλος πρέπει να τονιστεί πως κάθε μάσκα βρίσκεται με παλινδρόμηση δύο κλάσεων αντικειμένων (δυαδική παλινδρόμηση) και όχι χρησιμοποιώντας *Softmax*.

Προκειμένου να κατηγοριοποιηθεί ένα εικονοστοιχείο σε κάποιο αντικείμενο χρειάζεται να χρησιμοποιηθούν τα χαρακτηριστικά της εικόνας τα οποία υπολογίζονται χρησιμοποιώντας τα πρώτα συνελεκτικά επίπεδα του Faster R-CNN. Το πρόβλημα είναι πως οι διαστάσεις του τανυστή της εικόνας δεν είναι ίδιες με της διαστάσεις του τανυστή των χαρακτηριστικών. Έτσι αν μια περιοχή σημείων ήταν πάνω αριστερά και είχε διαστάσεις 15×15 πλέον θα έχει διαστάσεις 2.93×2.93 για χάρτη χαρακτηριστικών και εικόνα όπως παρουσιάζεται στο Σχήμα 1.9. Σε αυτό το σημείο προτιμήθηκε να χρησιμοποιηθεί μια διγραμμική παρεμβολή, ώστε να υπολογίζονται τα χαρακτηριστικά των διαστάσεων με δεκαδικά. Αυτή η τεχνική



Σχήμα 2.7: Το δίκτυο Faster-RCNN είναι πλέον το ολοκληρωτικά ενοποιημένο δίκτυο για εντοπισμό αντικειμένων που εξελίχθηκε από το R-CNN. Η οντότητα RPN (*Region Proposal Network*) βοηθά ως μέθοδος-τεχνική προσοχής αυτού του ενοποιημένου δικτύου, αντικαθιστώντας τη *Selective Search*.



Σχήμα 2.8: Αντί του RoIPool, η εικόνα περνάει από το καινούριο επίπεδο RoIAlign ώστε οι περιοχές του χάρτη χαρακτηριστικών που διαλέγονται από το RoIPool να αντιστοιχούν με μεγαλύτερη ακρίβεια σε περιοχές της εικόνας εισόδου. Αυτό απαιτείται γιατί η κατηγοριοποίηση ανά εικονοστοιχείο απαιτεί πιο λεπτομερή ευθυγράμμιση μεταξύ περιοχών της αρχικής εικόνας και του χάρτη χαρακτηριστικών

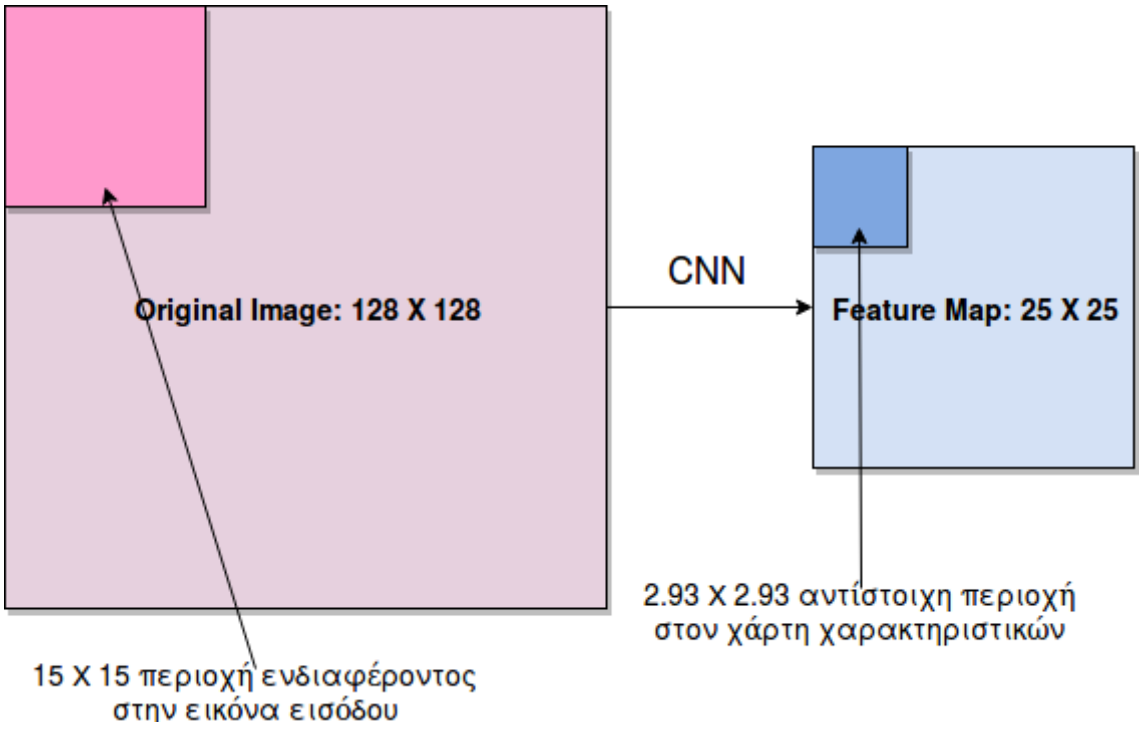
λέγεται RoIAlign. Πρότερα, το εικονοστοιχείο που αντιστοιχούσε στη θέση $x = 2.93$ των χαρακτηριστικών απλά θα αντιστοιχιζόταν στη θέση $x = 2$.

Αυτή η κατηγοριοποίηση δίνει τη δυνατότητα για μεγαλύτερη ακρίβεια στον εντοπισμό αντικειμένων και φέρνει την αναγνώριση αντικειμένων πιο κοντά στα ανθρώπινα αποτελέσματα. Επιπλέον ο αλγόριθμος μπορεί να διαγωνιστεί και σε διαφορετικά περιβάλλοντα όπως το COCO όπου απαιτείται η κατηγοριοποίηση των εικονοστοιχείων από εικόνες σε αντικείμενα. Σε αυτό το σύνολο δεδομένων (COCO 2015, COCO 2016), η μέση ακρίβεια που επιτυγχάνει ο αλγόριθμος είναι $mAP = AP = 37.1$, $AP_{50} = 60.0$, $AP_{75} = 39.4$, $AP_S = 16.9$, $AP_M = 39.9$, $AP_L = 53.5$ (μετρικές COCO[?]). Η ταχύτητα εντοπισμού ανά εικόνα μένει ίδια στα 5 fps για το ίδιο hardware, διότι η καινούρια προσθήκη είναι ένα μικρό κομμάτι νευρωνικού δικτύου. Ταυτόχρονα, η μνήμη που χρειάζεται είναι περίπου η ίδια διότι το δίκτυο για τη μάσκα απαιτεί 355 kB το πολύ για Faster R-CNN τύπου FPN (Σχήμα 10)

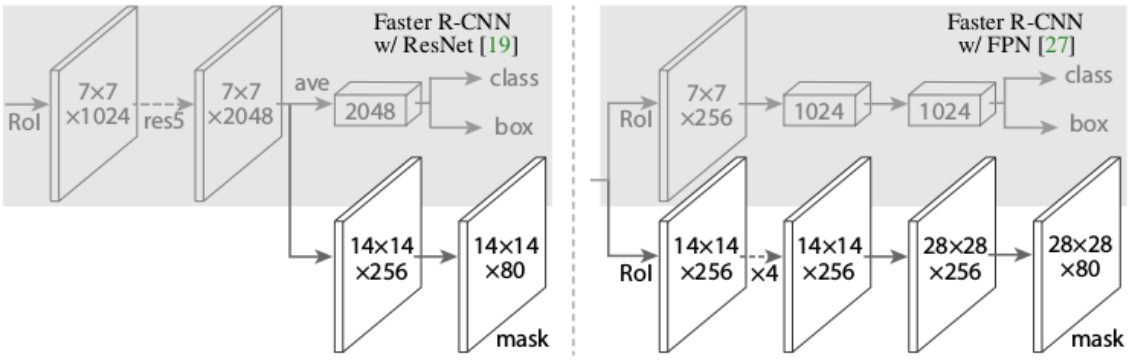
Η εκπαίδευση γίνεται με τον ίδιο τρόπο όπως στο Fast R-CNN, χρησιμοποιώντας gradient descent με ορμή. Ωστόσο, τα RPN και το δίκτυο της μάσκας εκπαιδεύονται ξεχωριστά, θεωρώντας το RoI θετικό όταν η μετρική IoU είναι πάνω από 0.5. Επιπλέον τελικά χαρακτηριστικά που διερευνήθηκαν για το δίκτυο αυτό είναι:

- Πολυκατηγορικές μάσκες ενάντια σε ανεξάρτητες μάσκες. Το αποτέλεσμα είναι ότι προτιμώνται οι ανεξάρτητες μάσκες οι οποίες μπορούν να υπολογιστούν μετά το RPN.
- Μάσκες ανά κατηγορία ή ανεξάρτητες. Το αποτέλεσμα είναι ότι οι ανεξάρτητες μάσκες έχουν πολύ μικρή διαφορά σε κριτήριο mAP από ότι αυτές που βρίσκονται ανά κατηγορία. Αυτό σημαίνει πως η πρόβλεψη μια μάσκας $m \times m$ είναι πιο συμφέρουσα από ότι να έχω μία μάσκα για κάθε πιθανή κλάση του εντοπισμένου αντικειμένου.

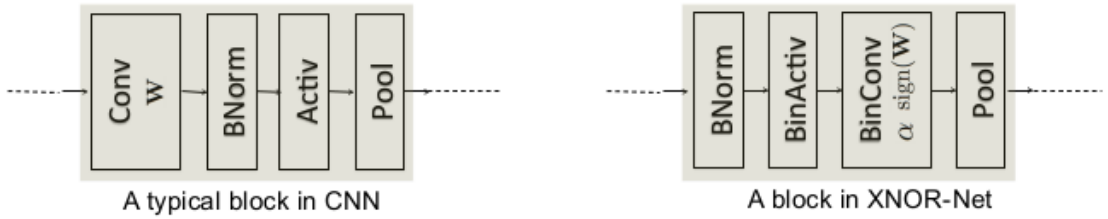
Βέβαια λόγω των υπολογιστικών απαιτήσεων το δίκτυο αυτό δε θεωρείται κατάλληλο για ενσωματωμένα συστήματα. Παρ' όλα αυτά η δυνατότητα εντοπισμού αντικειμένων ανά εικονοστοιχείο το χρήζει πιο εύχρηστο για άλλους αλγορίθμους που χρησιμοποιούνται σε αυτά, όπως το SLAM με vision, η αυτόνομη οδήγηση κ.α.



Σχήμα 2.9: Παράδειγμα αντιστοίχησης περιοχής από τον χάρτη χαρακτηριστικών στην αρχική εικόνα εισόδου. Για τον υπολογισμό των χαρακτηριστικών στο σημείο $(2.93, 2.93)$ χρησιμοποιείται διγραμμική παρεμβολή.



Σχήμα 2.10: Η κεφαλή του δικτύου: Για την κεφαλή του δικτύου χρησιμοποιούνται κομμάτια από άλλα δίκτυα όπως το ResNet[?] και το FPN[?]. Τα βελάκια αντιστοιχούν σε συνέλιξη, αποσυνέλιξη ή πέρασμα από ένα ολικά συνδεδεμένο επίπεδο. Όλες οι συνελίξεις γίνονται από 3×3 φίλτρα, εκτός από τη συνέλιξη εξόδου, η οποία είναι 1×1. Οι αποσυνελίξεις γίνονται από 2×2 φίλτρα με βήμα 2. Επίσης η ενεργοποίηση χρησιμοποιεί ReLU στα κρυφά επίπεδα. Αριστερά φαίνεται η μάσκα του Mask-RCNN χρησιμοποιώντας το πέμπτο στάδιο του ResNet (res5) αλλαγμένο για περιοχή 7×7 με βήμα 1, αντί για 14×14 με βήμα 2. Δεξιά φαίνεται μια υλοποίηση της μάσκας του Mask-RCNN με τέσσερις διαδοχικές συνελίξεις πάνω στο Faster R-CNN με την κεφαλή του FPN.



Σχήμα 2.11: Σύγκριση μεταξύ των μπλοκ του XNOR-Network (δεξιά) και ενός τυπικού CNN (αριστερά).

2.9 XNOR-Net [?]

Το δίκτυο XNOR προσεγγίζει διαφορετικά το πρόβλημα των CNN από ότι όλα τα παραπάνω. Η διαφορά έγκειται στη χρήση δυαδικών βαρών και αναπαράσταση της εισόδου ως δυαδικής. Η προσέγγιση αυτή καθιστά δυνατή την εκτέλεση του δικτύου όχι μόνο σε GPU ή εξειδικευμένο hardware όπως *FPGA* ή *ASIC*, αλλά και σε CPU. Περαιτέρω στην πρόταση του νευρωνικού εξετάζονται δύο εκδοχές

1. Μόνο δυαδικά βάρη που δέχονται τιμές στο σύνολο $\{-1, +1\}$.

2. Ταυτόχρονα δυαδικά βάρη και δυαδική είσοδος που δέχονται τιμές στο σύνολο $\{-1, +1\}$.

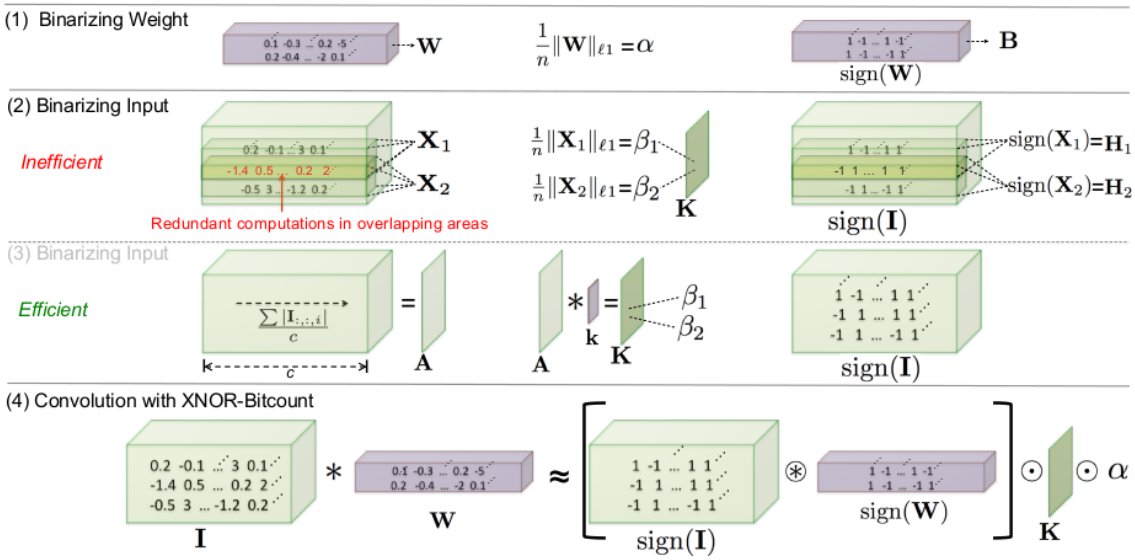
Η δεύτερη από τις δύο εκδοχές είναι και η πιο αποτελεσματική, τόσο από άποψη χρόνου εκτέλεσης, όσο και από άποψη ακρίβειας. Και στις δύο περιπτώσεις η αρχιτεκτονική του δικτύου είναι ίδια με κάποιο άλλο π.χ. AlexNet[?], GoogleNet[?], ResNet[?]. Αυτό που αλλάζει είναι τα βάρη και η σειρά με την οποία γίνονται οι πράξεις. Δηλαδή σε ένα κοινό δίκτυο έχουμε πρώτα τη συνέλιξη(C), μετά την κανονικοποίηση(B), αργότερα την ενεργοποίηση(A) και τέλος τη συγκέντρωση (P). Στο δίκτυο XNOR η σειρά είναι διαφορετική: $B \rightarrow A \rightarrow C \rightarrow P$.

Για να είναι επιτυχής η προσαρμογή του δικτύου σε διάφορες αρχιτεκτονικές οι συγγραφείς επινόησαν έναν αλγόριθμο για το μετασχηματισμό της εισόδου και των βαρών στο σύνολο των δυαδικών τανυστών. Σε αυτό το σύνολο τανυστών τα στοιχεία κάθε τανυστή λαμβάνουν τιμές στο σύνολο $\{-1, +1\}$. Πιο αναλυτικά: Έστω η κανονική είσοδος ενός επιπέδου I και έστω W το σύνολο των βαρών αυτού του επιπέδου. Τότε σύμφωνα με τον προτεινόμενο αλγόριθμο:

$$\mathbf{I} * \mathbf{W} \approx (sign(\mathbf{I}) \circledast sign(\mathbf{W})) \odot \mathbf{K}\alpha$$

Ο τελεστής $\circledast$ αντιπροσωπεύει συνέλιξη με χρήση XNOR αντί για πολλαπλασιασμού. Οπότε ο μετασχηματισμένος πίνακας εισόδου είναι ο $sign(\mathbf{I})$ και ο μετασχηματισμένος πίνακας βαρών είναι ο $sign(\mathbf{W})$. Ο τελευταίος όρος είναι ένας πίνακας που πολλαπλασιάζεται στοιχείο ανά στοιχείο για να κλιμακωθεί σωστά το αποτέλεσμα της συνέλιξης $(\circledast)$ με XNOR.

Όσον αφορά την εκπαίδευση το δίκτυο χρησιμοποιεί τη τεχνική ADAM [?] και πετυχαίνει καλύτερη ακρίβεια από ότι αν χρησιμοποιούσε στοχαστική *gradient descent* (SGD). Επίσης είναι το πρώτο δυαδικό δίκτυο που έχει ελεγχθεί στο διαγωνισμό του *ImageNet*. Στην ακρίβεια μεταξύ των κορυφαίων 5 κλάσεων χρησιμοποιώντας την αρχιτεκτονική του *AlexNet* πετυχαίνει σκορ 69.2 και της κορυφαίας μίας 44.2. Αντίστοιχα χρησιμοποιώντας την αρχιτεκτονική του *ResNet* πετυχαίνει 73.2 και 51.2. Αν χρησιμοποιούσε κανείς το *ResNet* από μόνο του θα πετύχαινε ακρίβεια 89.2 για τις κορυφαίες 5 και 69.3 για τη κορυφαία μία. Βέβαια αυτή η θυσία της ακρίβειας γίνεται προς αύξηση της ταχύτητας, η οποία είναι 32χ φορές πιο αυξημένη. Μεγαλύτερη επιτάχυνση δικτύων επιτυγχάνεται όσο τα φίλτρα στα επίπεδα του νευρωνικού είναι μεγαλύτερα. Αυτό βέβαια βρίσκεται σε αντίθεση με τη πιο μοντέρνα



Σχήμα 2.12: Επεξήγηση της διαδικασίας μετασχηματισμού της συνέλιξης σε συνέλιξη που κάνει χρήση της πράξης XNOR και δυαδικούς τανυστές.

τακτική να ελαττώνονται οι διαστάσεις των δικτύων όπως στο [?]. Από πλευράς ενεργειακής αποδοτικότητας δε γίνεται λόγος στη σχετική έρευνα. Παρόλα αυτά, από τη στιγμή που το δίκτυο απαιτεί λιγότερες πράξεις από πολλαπλασιασμούς και είναι μικρότερο σε μέγεθος εξοικονομείται τόσο ενέργεια από τον επεξεργαστή όσο και από τη χρήση της DRAM.

2.10 SSD [?]

Το δίκτυο αυτό παρουσιάζει κοινή αρχιτεκτονική με το Faster-RCNN και ουσιαστικά αποτελεί μια βελτίωση του (Deep Multibox [?]). Η διαφορά του από άλλα δίκτυα είναι ότι διακριτοποιεί τις προβλέψεις για τα ορθογώνια περιβλήματα αντικειμένων σε ένα σύνολο από προκαθορισμένα aspect ratios και μεγέθη ανά περιοχή του (feature map). Κατά τον χρόνο εκτέλεσης το νευρωνικό εξάγει σκορ για την παρουσία κάθε κλάσης αντικειμένων σε κάθε ένα από τα προκαθορισμένα περιβλήματα και παράγει διορθώσεις ώστε κάθε περίβλημα να ταιριάξει στο μέγεθος του αντικειμένου. Επιπλέον το δίκτυο συνδυάζει προβλέψεις από πολλαπλά (feature maps) με διαφορετικές αναλύσεις για τη φυσική αντιμετώπιση αντικειμένων διαφόρων μεγεθών.

Το SSD είναι πιο απλό σε σχέση με τις μεθόδους που απαιτούν την ύπαρξη δικτύου για να προτείνει αντικείμενα, γιατί καταργεί εντελώς τα ξεχωριστά επίπεδα που παράγουν προτάσεις αντικειμένων και αυτά που τα ακολουθούν για αναδειγματοληψία ανά εικονοστοιχείο (ή ανά feature) και τοποθετεί όλους τους υπολογισμούς σε ένα δίκτυο. Αυτό καθιστά το δίκτυο εύκολο στην εκπαίδευση και στην ενσωμάτωση του σε άλλα συστήματα.

Οι πειραματισμοί στα σύνολα δεδομένων PASCAL VOC, COCO και ILSVRC επιβεβαιώνουν ότι το SSD έχει ανταγωνιστική ακρίβεια σε σχέση με άλλες μεθόδους που χρησιμοποιούν επιπλέον βήματα για την πρόταση του αντικειμένου. Ταυτόχρονα, είναι πιο γρήγορο και προσφέρει μια ενοποιημένη δομή τόσο για την εκπαίδευση όσο και για την απλή εκτέλεση. Για εικόνες εισόδου 300×300 (SSD300) το δίκτυο επιτυγχάνει mAP 74.3% στο σύνολο δεδομένων VOC2007 στα 59 FPS χρησιμοποιώντας την GPU Nvidia Titan X. Ενώ για εικόνες εισόδου 512×512 επιτυγχάνει mAP 76.9% υπερβαίνοντας την απόδοση του Faster-RCNN.

Συγκρινόμενο με άλλες μεθόδους με ένα στάδιο, το SSD έχει κατά πολύ καλύτερη ακρίβεια

ακόμα και με μικρότερη εικόνα εισόδου. Αυτό το χρήζει ικανό για να χρησιμοποιηθεί σε ενσωματωμένα συστήματα, αφού οι απαιτήσεις μνήμης του είναι μικρότερες, λόγω του μονού σταδίου και είναι αρκετά γρήγορο για την εκτέλεσή του.

2.11 Σύνοψη

Στα προηγούμενα μέρη του κεφαλαίου αναλύσαμε κάθε δίκτυο ξεχωριστά παρουσιάζοντας την αρχιτεκτονική, την ακρίβειά και την ταχύτητά του. Επίσης, συμπεράναμε κατά πόσο το κάθε ένα μπορεί να χρησιμοποιηθεί σε ενσωματωμένη συσκευή. Από εκεί προέκυψε πως τα δίκτυα τα οποία μπορούν να χρησιμοποιηθούν για εντοπισμό αντικειμένων είναι τα:

- SqueezeDet
- YOLO
- YOLO9000
- SSD

Από άποψη χρόνου εκτέλεσης

Στη βιβλιογραφία διακρίνονται δύο είδη δικτύων: για εντοπισμό αντικειμένου (object localization) και για αναγνώριση αντικειμένου (object recognition).. Τα πρώτα μελετούνται ως προς το χρόνο εκτέλεσης και την ακρίβεια (mAP). Τα δεύτερα ως προς τον αριθμό παραμέτρων, τον χρόνο εκτέλεσης, την ακρίβειά τους και το πόσο καλά μπορούν να εκπαιδευτούν. Επίσης, τα δίκτυα εντοπισμού εμπεριέχουν μέσα δίκτυα για εξαγωγή χαρακτηριστικών(feature extraction). Οπότε παρατίθενται οι δύο πίνακες για δίκτυα που κάνουν feature extraction (Πίνακας 1.2) και για δίκτυα που κάνουν εντοπισμό αντικειμένων (Πίνακας 1.3).

Τα δίκτυα YOLO και SSD αν και προτείνονται για ενσωματωμένα, στα πειράματα που έχουν γίνει απαιτούν αρκετούς πόρους ακόμα και για ενσωματωμένες συσκευές, διότι εκτελούνται μόνο σε GPU. Επειδή γενικότερα στην βιβλιογραφία επικρατεί μία σύγχυση για τους χρόνους εκτέλεσης των νευρωνικών δικτύων στο παρόν κείμενο χρησιμοποιήθηκε η μετρική frame/ms/Watt , όπου μετριέται ο χρόνος εκτέλεσης ως προς την κατανάλωση ισχύος στο 'forward pass' του νευρωνικού ανά frame. Κατά αυτό τον τρόπο υπάρχει ένα μέτρο το οποίο μπορεί να συγκρίνει τους χρόνους εκτέλεσης ανεξαρτήτως της αρχιτεκτονικής. Ωστόσο, υπάρχει το μειονέκτημα ότι διαφορετικές αρχιτεκτονικές χρησιμοποιούν διαφορετικά ποσά ενέργειας για να πραγματοποιήσουν τις ίδιες πράξεις.

Από άποψη ακρίβειας το επικρατέστερο δίκτυο (από αυτά που αναλύονται) είναι το Faster-RCNN. Παρόλα αυτά, τα δίκτυα δεν μπορούν να συγκριθούν με το Mask-RCNN γιατί αυτό δεν βρίσκει ορθογώνια περιβλήματα μόνο αλλά κάνει και εντοπισμό των αντικειμένων της εικόνας ανά εικονοστοιχείο. Για τα υπόλοιπα η σειρά ακρίβειας είναι η παρακάτω.

1. Faster-RCNN
2. SqueezeDet
3. YOLO9000
4. SSD

Αξιοσημείωτα είναι ότι το SSD πετυχαίνει καλύτερη ακρίβεια, όση και το Faster-RCNN με χρόνους εκπαίδευσης διπλάσιους του YOLO. Επίσης ότι το SqueezeDet μπορεί και ξεπερνάει την ακρίβεια του Faster-RCNN σε κάποια σύνολα δεδομένων. Το τελευταίο στη λίστα

Δίκτυο feature extraction	Top-1 ακρίβεια	Αριθμός Παραμέτρων	Μέσος Χρόνος Εκτέλεσης/W
SqueezeNet	57.5	421,098 (6 bit)	$6.53\,ms$
XNOR-Net	44.2	61M (61MB)	$2.275ms\,/2W$ (ATOM Z530 CPU)
VGG-16	70.5	14,714,688	$41.23ms$
MobileNet-224	83.3	3,191,072	$19ms\,/80W$(Xeon E3-1231 v3 CPU)
ResNet-101	76.4	42,605,504	$2.48ms$
Inception V3	78.0	21,802,784	$2.017ms$
Inception ResNet V2	80.4	54,336,736	$4ms$

Πίνακας 2.2: Σύγκριση δικτύων για feature extraction στο εμπρόσθιο πέρασμα για ένα frame. Όλα τα δίκτυα έχουν βάρη τύπου float32 εκτός από τα SqueezeNet και XNOR-Net και μόνο αυτά τα 2 εκτελούνται σε CPU. Η ακρίβεια των δικτύων μετριέται στο σύνολο δεδομένων του ImageNet. Τα χαρακτηριστικά των δικτύων που δεν αναλύονται στο κεφάλαιο είναι από [?], ενώ άλλα προέρχονται από [?], [?],[?],[?]. Η εικόνες για την μέτρηση είναι στο μέγεθος των εικόνων του ImageNet. Αν η πλατφόρμα εκτέλεσης του δικτύου είναι διαφορετική, τότε αυτή αναγράφεται μέσα σε παρενθέσεις.

Δίκτυο Εντοπισμού	mAP	Μέσος Χρόνος Εκτέλεσης / Watt / frame
SqueezeDet + SqueezeNet	80.4 (KITTI)	$31.2\,ms/128.3W$ (NVIDIA Titan X)
YOLO9000 480×480 + Darknet	77.8 (PASCAL VOC 7+12)	$17\,ms/250W$ (NVIDIA Titan X)
SSD300 + VGG16	74.3 (PASCAL VOC 7+12)	$52\,ms/250W$ (NVIDIA Titan X)
Mask R-CNN + ResNeXt-101	37.1 (MS COCO 2015)	$240\,ms/143.1W$ (NVIDIA Titan X)
Faster R-CNN + ResNet	76.4 (PASCAL VOC 7+12)	$200\,ms/143.1W$ (NVIDIA Titan X)

Πίνακας 2.3: Σύγκριση δικτύων για εντοπισμό αντικειμένων. Η ακρίβεια των δικτύων μετριέται σε διάφορα σύνολα δεδομένων. Η παράθεση αυτή γίνεται, διότι δεν υπάρχει κάποιο κοινό σύνολο δεδομένων στο οποίο να έχουν εκτελεσθεί όλα τα δίκτυα εντοπισμού, σε αντίθεση με τα απλά CNN. Βέβαια αυτό γίνεται για λόγους που αφορούν την ακρίβεια και θα εξετασθούν αναλυτικότερα στο επόμενο κεφάλαιο.

είναι το δίκτυο XNOR-Net το οποίο έχει μεγάλη διαφορά από τα άλλα δίκτυα σε ακρίβεια. Όπως αναφέρεται και στο ίδιο προσπαθεί να λύσει το πρόβλημα εκτέλεσης νευρωνικού σε CPU χρησιμοποιώντας bits για τα βάρη και όχι Bytes, προκειμένου να το χωρέσει και μια συμβατική ενσωματωμένη συσκευή. Η χρησιμότητα του όμως αίρεται στη γενική περίπτωση, με τη λογική ότι το SqueezeDet μπορεί να πετύχει αρκετά μεγαλύτερη ακρίβεια έχοντας μικρότερο μέγεθος. Χρήσεις του XNOR-Net αφορούν περισσότερο σχεδιαστικά προγράμματα π.χ. για αναγνώριση αντικειμένων από σκίτσα [?].

Από άποψη μνήμης

Τα δίκτυα για feature extraction μελετώνται πλέον με τον αριθμό των παραμέτρων. Αυτό γίνεται διότι η αναπαράστασή τους σε 8, 16 ή 32 bit καθορίζεται από τους επιπλέον αλγόριθμους συμπίεσης (π.χ. Ristretto [?]). Ο Πίνακας 1.2 δείχνει τα χαρακτηριστικά κάθε νευρωνικού τέτοιου τύπου. Οι χρόνοι αφορούν την εκτέλεση των δικτύων αυτών με τη χρήση του tensorflow [?] στη GPU GTX 1080 Ti της NVIDIA, και στο σύνολο δεδομένων του Imagenet ILSVRC2012. Οι υπολογισμοί τους έγιναν στη συστοιχία του εργαστηρίου Αρχιτεκτονικής και συστημάτων που αναλύεται σε παρακάτω κεφάλαιο. Παρά ταύτα, παρατίθενται και οι χρόνοι για δίκτυα που δεν ενδείκνυται η εκτέλεσή τους σε GPU, σε άλλη πλατφόρμα.

Το RCNN και οι απόγονοί του τοποθετήθηκαν κυρίως για να φανεί η σειρά στην οποία όλοι βασίστηκαν κατά την ανάπτυξη των νευρωνικών δικτύων εντοπισμού αντικειμένων. Όλες

οι παραπάνω προτάσεις συγκρίνονται με τα Fast-RCNN και Faster-RCNN και όπως φαίνεται τα επόμενα θα συγκριθούν με το Mask-RCNN. Μάλιστα όπως το SqueezeDet εμπνεύστηκε από το Faster R-CNN έτσι και ένα επόμενο δίκτυο ενσωματωμένων μπορεί να εμπνευστεί από το Mask R-CNN. Ήδη παρόμοια λειτουργία σε ενσωματωμένα συστήματα πραγματοποιεί το ENet [**?**].