



RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων
Δεδομένων Παρατήρησης Γης

Ανίχνευση αντικειμένων (Object Detection)

Αθηνά Ψάλτα
Βασίλειος Τσιρώνης
Κωνσταντίνος Καράντζαλος

Εαρινό εξάμηνο 2022

Περιεχόμενα

1. Αναγνώριση Αντικειμένων
 - a. Ιστορική αναδρομή
 - b. Τεχνικές cascade
2. Αρχιτεκτονικές ΤΝΔ για ανίχνευση αντικειμένων
 - a. Ταξινόμηση αλγορίθμων
 - b. Faster RCNN
 - c. Single-shot detector
 - d. YoLO
 - e. RetinaNet
 - f. DETR

Ανίχνευση Αντικειμένων

Χαρακτηριστικά και ταξινομητές

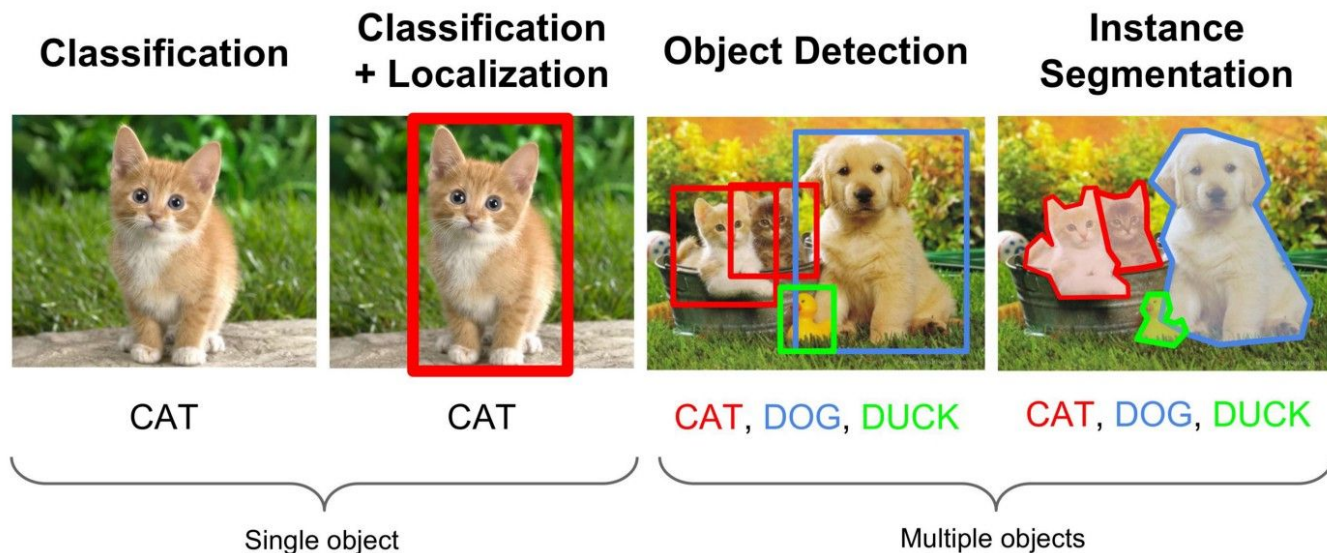
Ανίχνευση Αντικειμένων

Η **ανίχνευση αντικειμένων** (object detection) αφορά τον εντοπισμό μεμονωμένων αντικειμένων συγκεκριμένων (προκαθορισμένων) κατηγοριών σε εικόνες

- Πρόβλημα **ταξινόμησης** (Τι αντικείμενο;)
- Πρόβλημα **παλινδρόμησης** (Που στην εικόνα;)
 - “Τοπικοποίηση” / Localization

Το O.D. αποτελεί βασική συνιστώσα για πληθώρα εφαρμογών της Όρασης Υπολογιστών όπως :

1. Instance / Panoptic segmentation
2. Παρακολούθηση (ενός/πολλαπλών) αντικειμένων
3. Υποτιτλισμός εικόνων

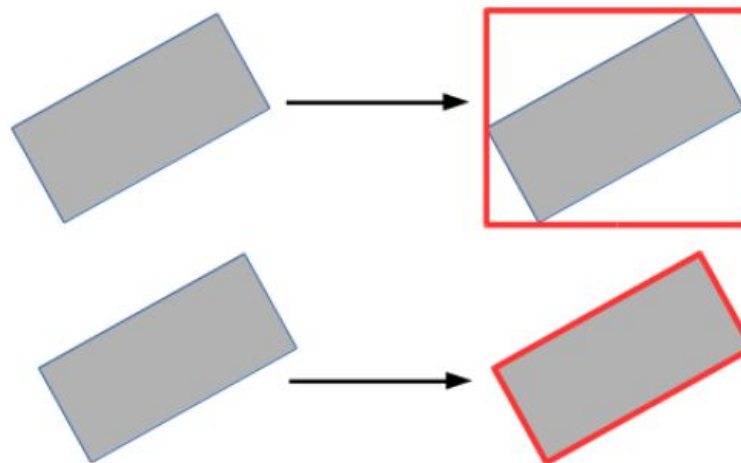


Ανίχνευση Αντικειμένων

Η έννοια του **bounding box** (bb) αποτελεί τη βάση για τον αναλυτικό ορισμό του O.D. Ένα bb ουσιαστικά αποτελεί το ορθογώνιο παραλληλόγραμμο με το ελάχιστο εμβαδόν που περιγράφει την απεικόνιση του εκάστοτε αντικειμένου σε μία εικόνα.

Παραμετροποιήσεις:

1. $(x_{min}, y_{min}, x_{max}, y_{max})$
2. $(x_{min}, y_{min}, width, height)$
3. $(x_{centre}, y_{centre}, (semi-)width, (semi-)height)$



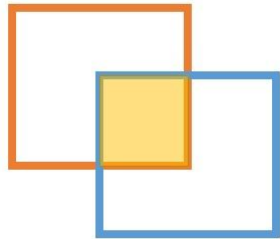
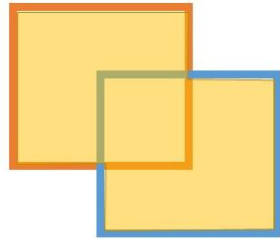
Συνήθως στο O.D. αναφερόμαστε σε bb με **πλευρές παράλληλες στους άξονες (xy)**

- Σπανιότερα, στην υποπερίπτωση του **rotated object detection** χρησιμοποιούνται “ελεύθερα” (ως προς την περιστροφή) bb
 - 1/2/3 Παραμετροποιήσεις + $(angle[-\pi/2, \pi/2])$
 - $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4) \rightarrow$ Υπερ-παραμετροποίηση, ανάγκη δεσμεύσεων

Ανίχνευση Αντικειμένων

Η μετρική **Intersection-over-Union (IoU)** χρησιμοποιείται ευρέως σε προβλήματα O.D.

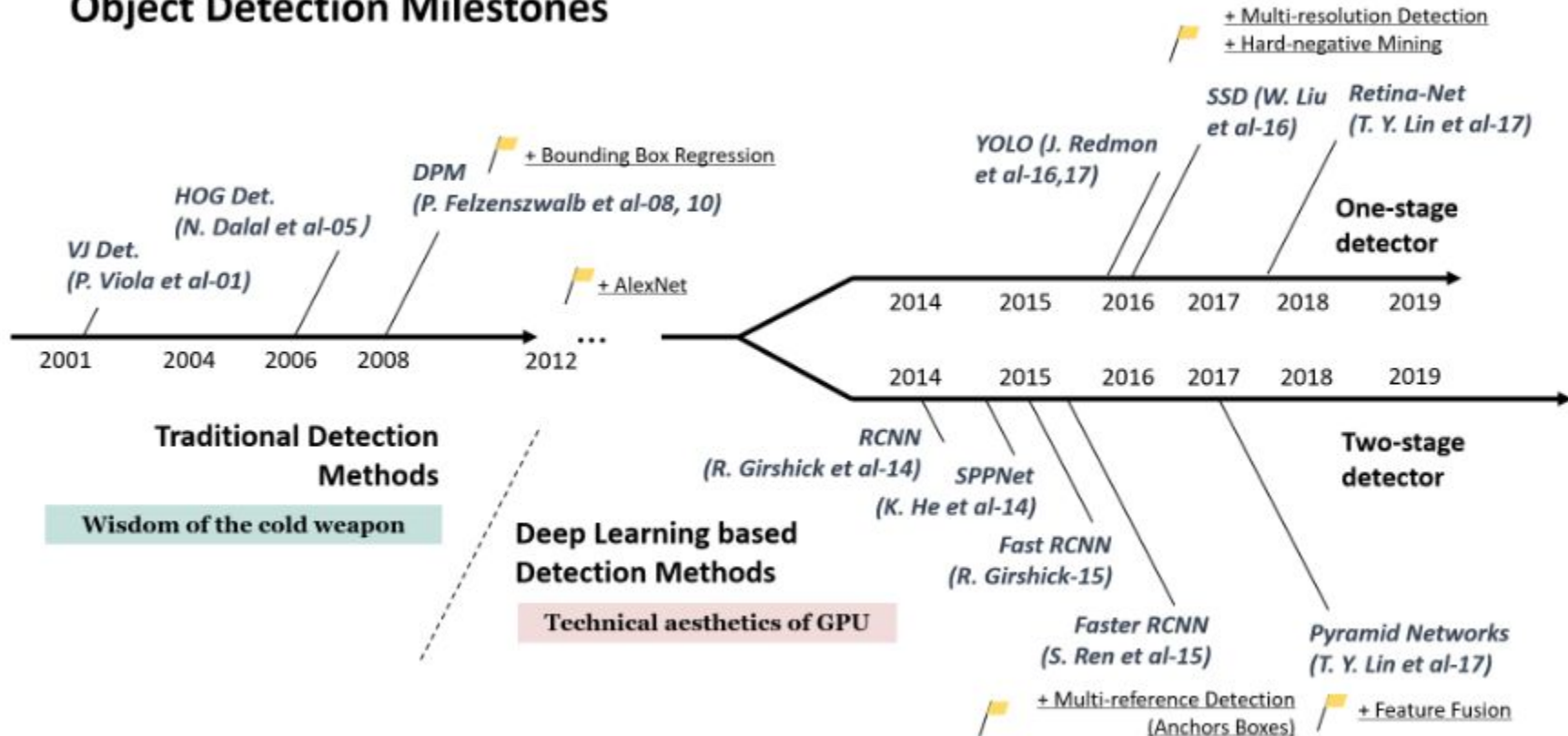
Η κύρια χρήση της αφορά την αντιστοίχιση των προβλέψεων (υπό την μορφή ενός συνόλου από bbs) ενός αλγορίθμου ανίχνευσης με τα δεδομένα αληθείας (επίσης υπό την ίδια μορφή)


$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


— Prediction
— Ground-truth

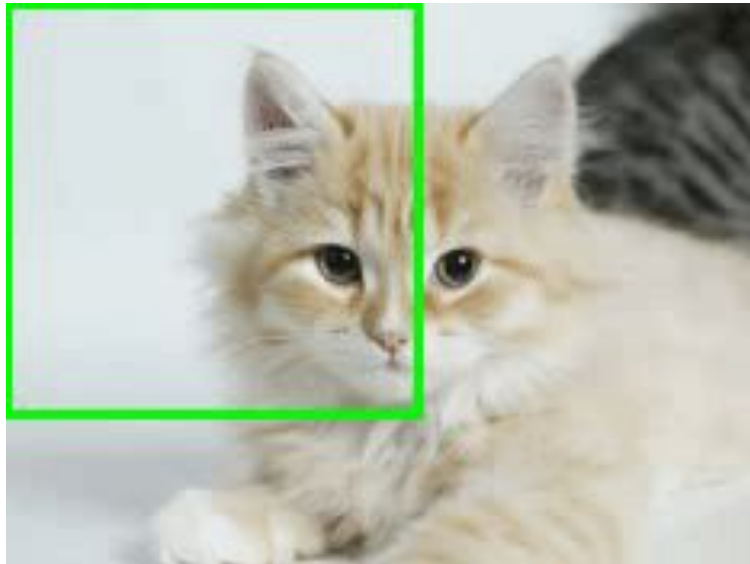
Ανίχνευση Αντικειμένων

Object Detection Milestones



Ανίχνευση Αντικειμένων

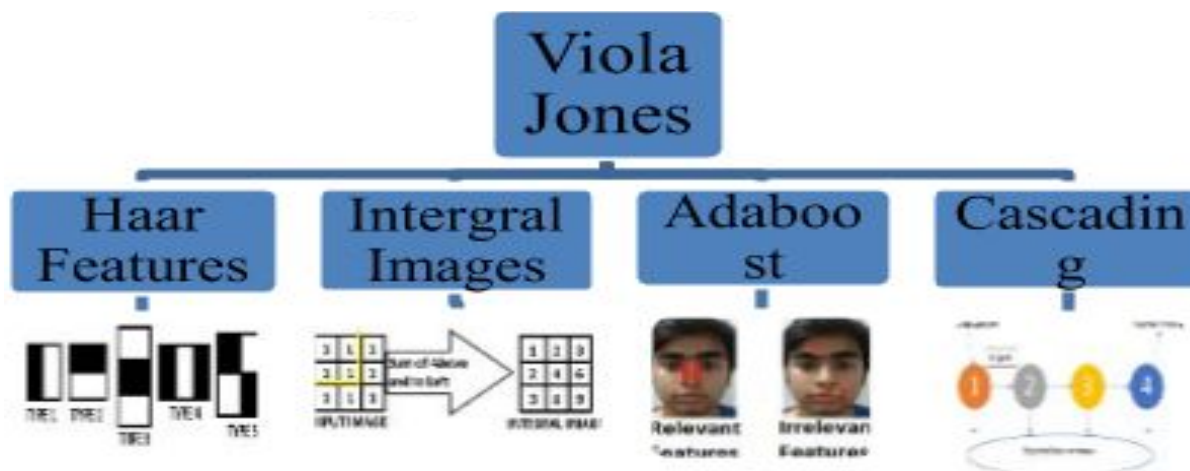
Η απλούστερη προσέγγιση στο πρόβλημα του O.D. αποτελεί η τεχνική των **κυλιόμενων παραθύρων**. Κατά την τεχνική αυτή ένα “παράθυρο” κυλίνεται επί της εικόνας όπου σε κάθε πιθανή θέση του ένας αλγόριθμος ταξινόμησης **ταξινομεί** το παράθυρο σε **κάποια από τις πιθανές κατηγορίες** αντικειμένων ή στη γενική κατηγορία του υποβάθρου (**background**).



Ανίχνευση Αντικειμένων - Viola-Jones

Από τις πρώτες προσπάθειες ανάπτυξης ενός αλγορίθμου ανίχνευσης αποτελεί η δουλειά των Viola & Jones (2001) οι οποίοι βελτίωσαν τον κλασικό ανιχνευτή κυλιόμενων παραθύρων εφαρμόζοντας τις ακόλουθες βελτιώσεις:

1. **Ολοκληρωτική εικόνα (integral image):** Εφάπαξ υπολογισμός σε επίπεδο πλήρους εικόνας μεγεθών που απλοποιούν την εξαγωγή χαρακτηριστικών τυπου “Haar” σε κάθε παράθυρο
2. **Επιλογή χαρακτηριστικών (Feature Selection):** Χρήση ενός Adaboost ταξινομητή για την επιλογή ενός υποσυνόλου χαρακτηριστικών από μία τεράστια “δεξαμενή / pool” διαθέσιμων χαρακτηριστικών ικανά να υποστηρίξουν το πρόβλημα της ανίχνευσης
3. **Πολλαπλή ανίχνευση (detection cascades):** Η ανίχνευση / ταξινόμηση σε κάθε παράθυρο πραγματοποιείται σε μία σειρά διαδοχικών ταξινομήσεων αυξανόμενου υπολογιστικού κόστους. Τα πρώτα βήματα πρέπει να μεγιστοποιούν το Recall της “+” κατηγορίας (foreground)

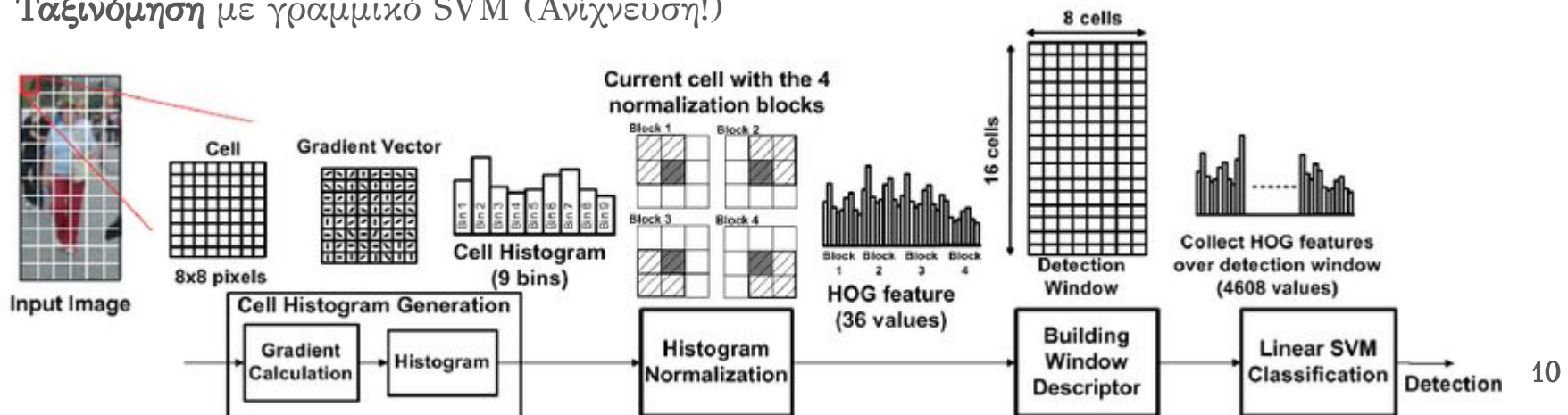
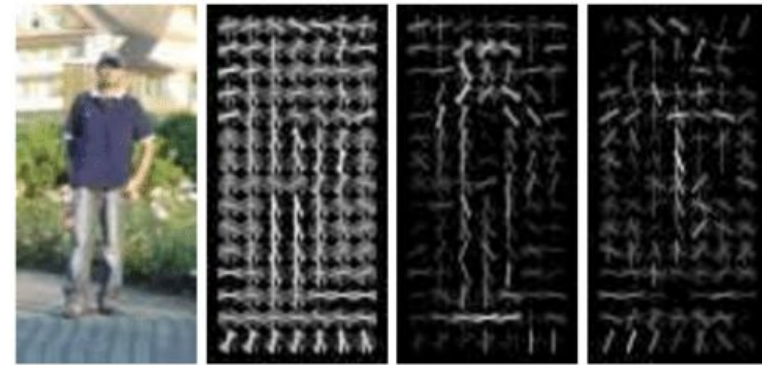
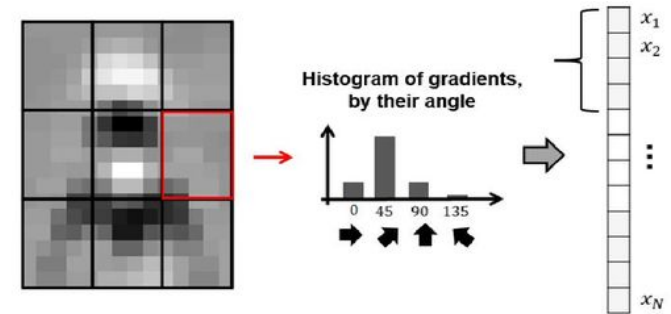


Ανίχνευση Αντικειμένων - HOG detector

Τα **Histograms of Oriented Gradients (HOG)** αποτελούν ένα από τους πλέον δημοφιλείς περιγραφείς χαρακτηριστικών στην Όραση Υπολογιστών.

Ο αλγόριθμος ανίχνευσης (κυλιόμενων παραθύρων) HOG (**HOG-detector**) βασίζεται στον HOG-descriptor για την ανίχνευση σε κάθε “παράθυρο” ανίχνευσης ως εξής:

1. **HOG χαρακτηριστικά** για κάθε 8x8 κελί του παραθύρου (χωρίς επικαλύψεις)
2. **Κανονικοποίηση ιστογράμματος** για κάθε 2x2 μπλοκ κελιών (κυλιόμενα με stride 1 παράθυρο)
3. Δημιουργία διανύσματος περιγραφής εικόνας
4. Ταξινόμηση με γραμμικό SVM (Ανίχνευση!)



Ανίχνευση Αντικειμένων - MS-HOG detector

Ο HOG-detector μπορεί να εφαρμοστεί και σε “πυραμίδες” εικόνες ως “Ανιχνευτής HOG πολλαπλών κλιμάκων” (multi-scale HOG Detector)

Detection Phase

The HOG Detector

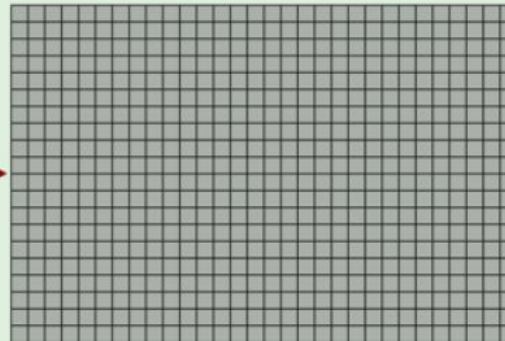
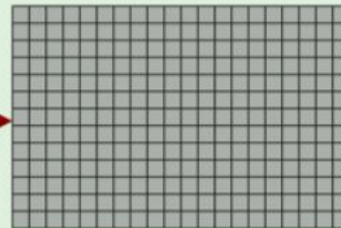
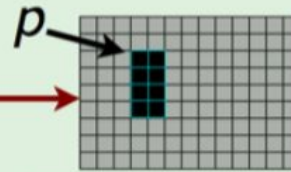


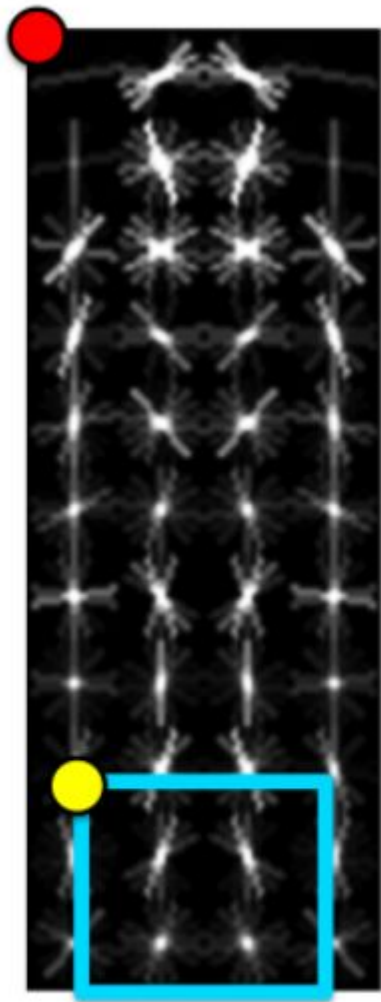
Image pyramid

HOG feature pyramid

$$\text{score}(l, p) = \mathbf{w} \cdot \phi(l, p)$$



Ανίχνευση Αντικειμένων - DPM



Ο αλγόριθμος ανίχνευσης **Deformable Part Model (DPM)** αποτελεί ίσως τον πλέον διαδεδομένο αλγόριθμο ανίχνευσης που δεν αξιοποιεί τεχνικές βαθιάς μάθησης. Φιλοσοφικά, βασίζεται στο “διαίρει και βασίλευε” όπου το εκάστοτε αντικείμενο αποσυντίθεται σε ένα σύνολο διαφορετικών μερών. Αναγνωρίζοντας τα επιμέρους “μέρη” και τη σχετική τους διάταξη μπορεί να εξαχθεί η πρόβλεψη ανίχνευσης συνολικά ενός αντικειμένου.

Ένας DPM ανιχνευτής αποτελείται τυπικά από:

- ένα “**root**” φίλτρο
- ένα πλήθος “**part**” φίλτρων

part location: $\mathbf{v}_1 = (v_{1,x}, v_{1,y})$
and size: 6×6 (in HOG cells)

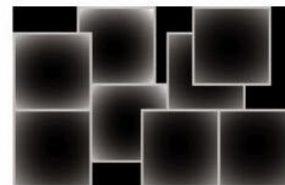
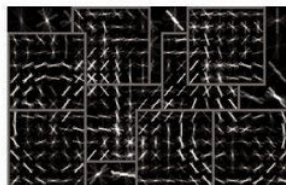
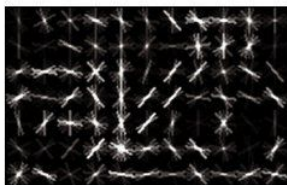
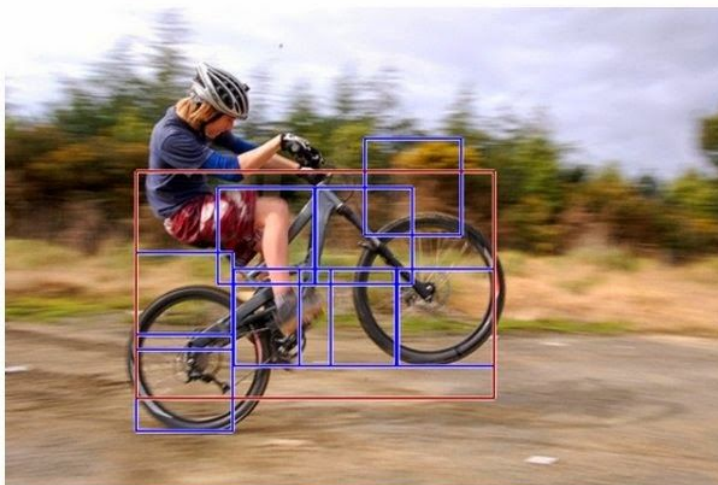
root part (or root filter)

Ανίχνευση Αντικειμένων - DPM

Τα **part** φίλτρα εκπαιδεύονται μέσω ενός αλγορίθμου ασθενούς επίβλεψης (weak-supervision)

Επιπλέον τεχνικές όπως:

- **hard-negative mining**: Εύρεση “αρνητικών” δειγμάτων (background) που δυσκολεύουν τον ταξινομητή ώστε να ενισχυθεί η επίδοσή του
- **bounding box regression**: Πρόβλεψη διορθώσεων (μικρο-μετάθεση, μικρο-κλίμακα) για τα bbs των αντικειμένων ώστε να προσαρμόζονται καλύτερα στο αντικείμενο

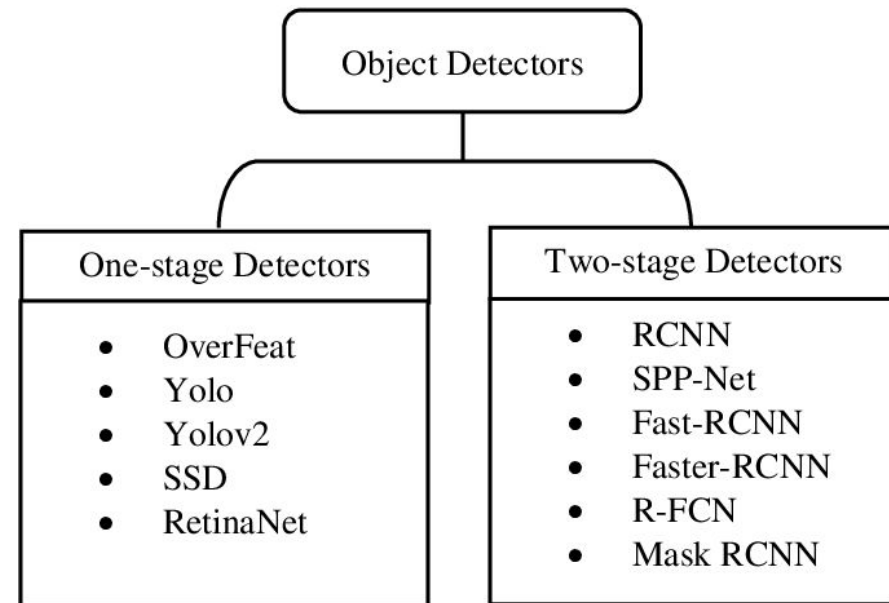


Αρχιτεκτονικές ΤΝΔ για ανίχνευση αντικειμένων

Οι αλγόριθμοι ανίχνευσης βαθιάς μάθησης κατηγοριοποιούνται ως εξής:

1. Αλγόριθμοι 2-σταδίων (2-stage):

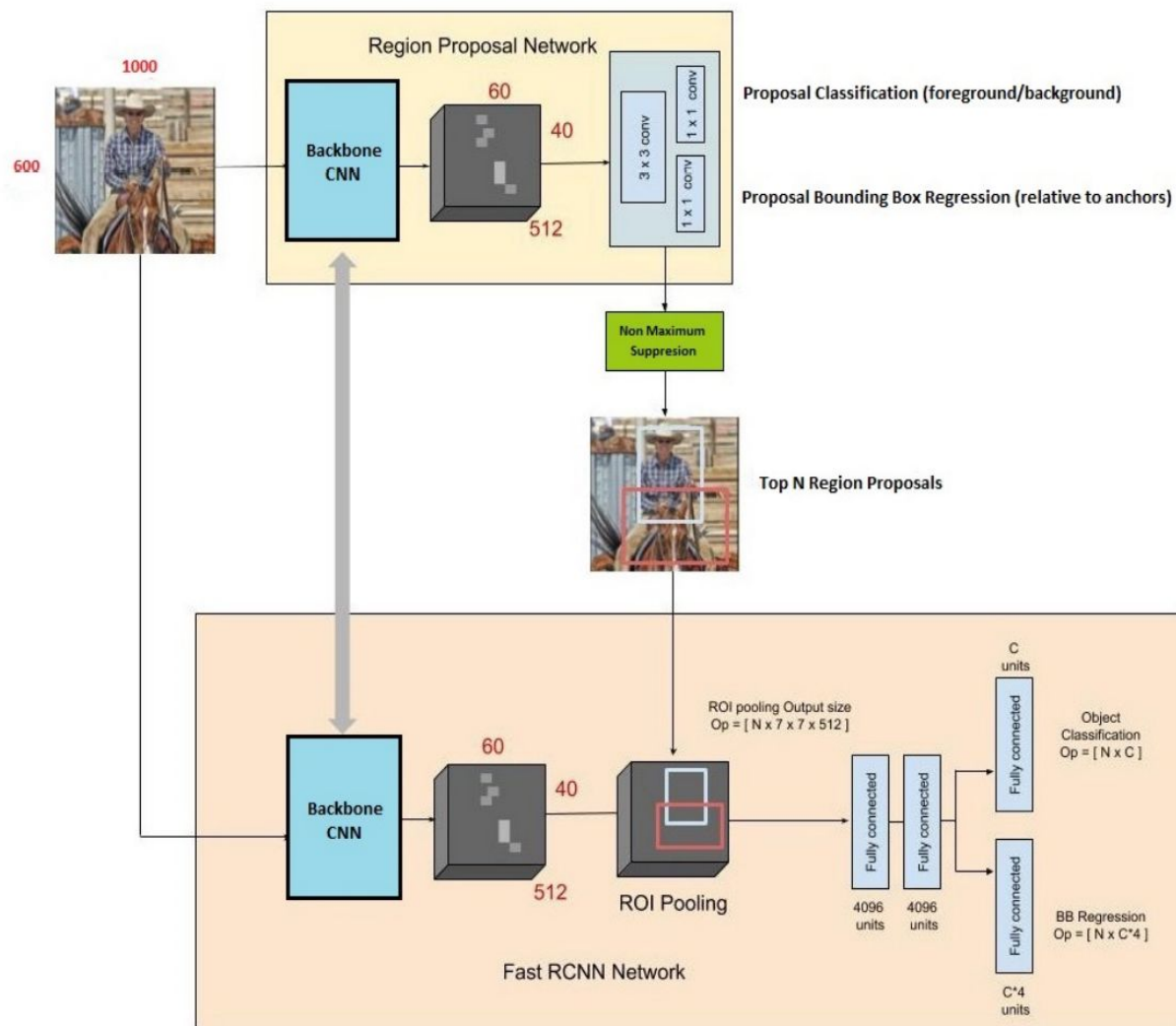
- Υπολογισμός περιοχών ενδιαφέροντος (regions of interest - ROIs)
- **Objectness:** Μέτρο της ύπαρξης αντικειμένου (ανεξαρτήτως κατηγορίας) σε μία περιοχή της εικόνας
- **Object Proposals:** Περιοχές της εικόνας που πιθανώς περιέχουν κάποιο αντικείμενο
- Ανίχνευση ανά περιοχή ενδιαφέροντος
- **Ταξινόμηση** σε κάποια κατηγορία ή απόρριψη (κατηγορία υποβάθρου)
- **Παλινδρόμηση** για τοπικοποίηση (localization)



2. Αλγόριθμοι ενός σταδίου (1-stage):

- **Απευθείας ανίχνευση αντικειμένων** σε ένα πέρασμα από ένα συνεχόμενο δίκτυο
- Συνήθως Πλήρως Συνελικτικές αρχιτεκτονικές

Ο αλγόριθμος Faster R-CNN



Faster R-CNN

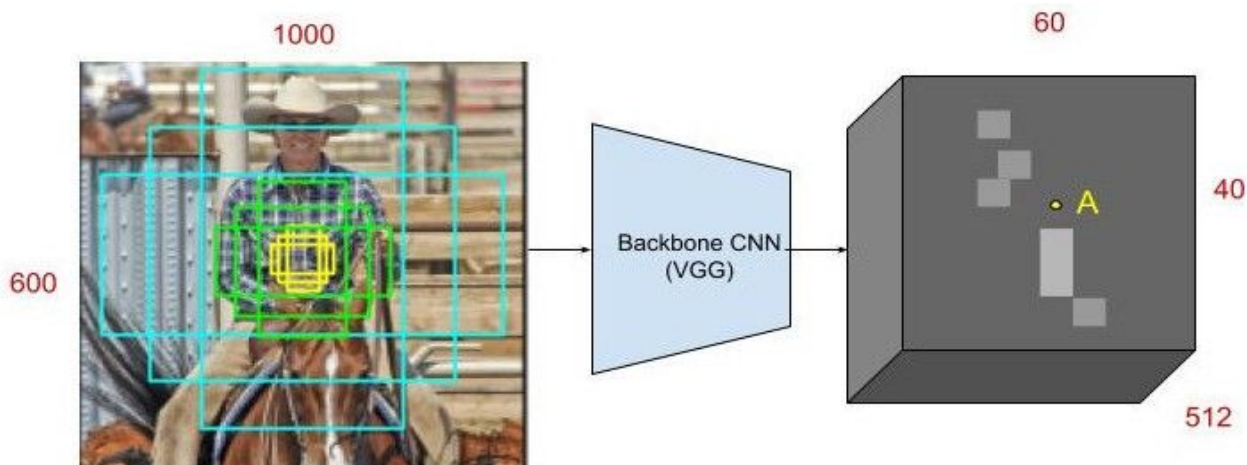
- Two-stage detector που προκύπτει από νέα έκδοση του αλγορίθμου Fast R-CNN
 - Το 1ο stage υποδεικνύει τη δημιουργία των region proposals, δηλαδή την ύπαρξη ή μη ενός αντικειμένου
 - Το 2ο stage αφορά την επεξεργασία αυτών των proposals και την ταξινόμησή τους στις κατάλληλες κατηγορίες

Βασικές συνεισφορές του αλγορίθμου

1. **Region Proposal Network (RPN)** → δίκτυο που ασχολείται αποκλειστικά με τα Object Proposals
 - Ο αρχικός αλγόριθμος Fast R-CNN χρησιμοποιεί έναν αλγόριθμο Selective Search που όμως είναι μια ιδιαίτερα χρονοβόρα διαδικασία
2. Εισαγωγή της έννοιας των **Anchors** αντί χρήσης πολλαπλών φίλτρων με διαφορετικά μεγέθη ή μιας “πυραμίδας” εικόνων, δηλαδή πολλών εκδοχών της ίδιας εικόνας σε διαφορετικές κλίμακες
3. Οι πράξεις συνελίξεων είναι **κοινές** μεταξύ του RPN και του detector Fast R-CNN με αποτέλεσμα να μειώνεται το υπολογιστικό κόστος

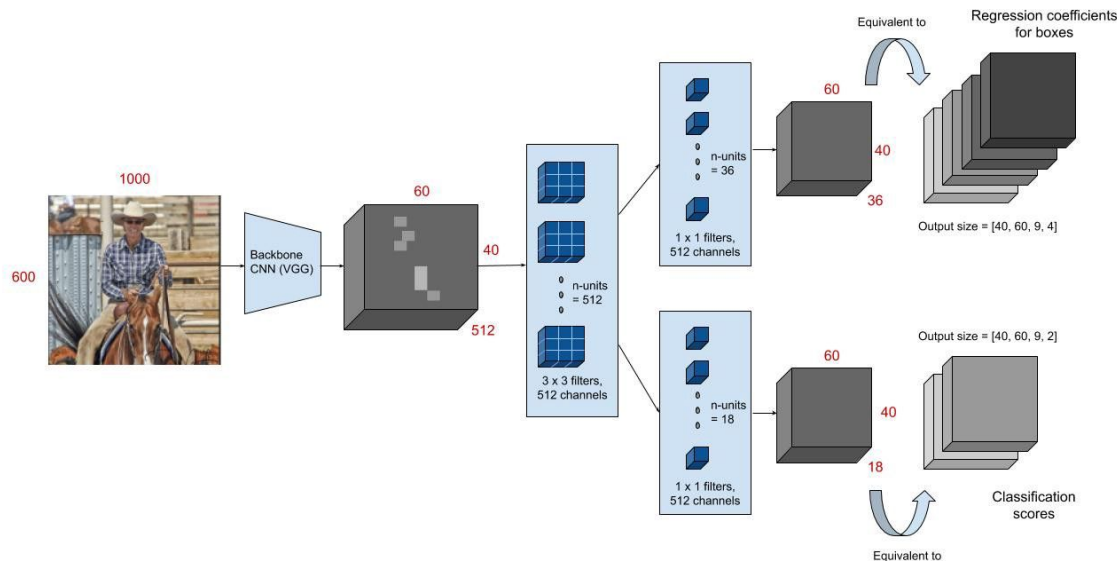
Region Proposal Network (RPN)

- Η εικόνα εισόδου τροφοδοτείται σε ένα backbone CNN δικτυο μετά από ένα κατάλληλο resize
 - Τα χαρακτηριστικά (feature map) που προκύπτουν συνήθως είναι αρκετά μικρότερα (χωρικά) από την αρχική εικόνα ανάλογα με την αρχιτεκτονική του backbone δικτύου
- Τοποθέτηση **Anchors** στην αρχική εικόνα : υποδεικνύουν πιθανά αντικείμενα σε διάφορα μεγέθη και λόγους διαστάσεων (aspect ratio)
 - Κάθε σημείο του feature map, μέσω των anchors αντιστοιχίζεται στην αρχική εικόνα σε ένα συγκεκριμένο BB
 - Στο σχήμα βλέπουμε 9 πιθανά anchors με 3 διαφορετικά μεγέθη και λόγους διαστάσεων στην αρχική εικόνα για το σημείο A του feature map



Region Proposal Network (RPN)

- Το δίκτυο πρέπει να ελέγξει αν τα anchors όντως περιέχουν αντικείμενα στην αρχική εικόνα όσο “κινούμαστε” σε κάθε σημείο του feature map
 - Αντιστοίχιση μέσω της IoU των anchors με τα δεδομένα αληθείας
- Περαιτέρω βελτίωση των συντεταγμένων των anchors ώστε να προκύπτουν BBs ως **Object Proposals** ή αλλιώς **Regions of Interest**
- Αρχικά στο RPN εφαρμόζεται μια συνελικτική στρώση 3x3 στον feature map
- Το RPN δίκτυο χωρίζεται σε 2 μέρη
 - **Ταξινόμηση** του αντικειμένου με 1x1 συνελικτική στρώση → πιθανότητα αν υπάρχει το αντικείμενο μέσα και στα 9 anchors
 - **BB regression** με 1x1 συνελικτική στρώση → 4 συντεταγμένες για κάθε μία από τις 9 anchors



Region Proposal Network (RPN)

- “**Θετικό**” δείγμα anchor : υψηλότερο IoU με κάποιο ground truth BB ή IoU μεγαλύτερο από κάποιο κατώφλι με κάποιο ground truth BB
- “**Αρνητικό**” δείγμα anchor : IoU μικρότερο από κάποιο άλλο κατώφλι
 - Γίνονται θετικά δείγματα για την background κατηγορία!
- Multi-task συνάρτηση κόστους

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

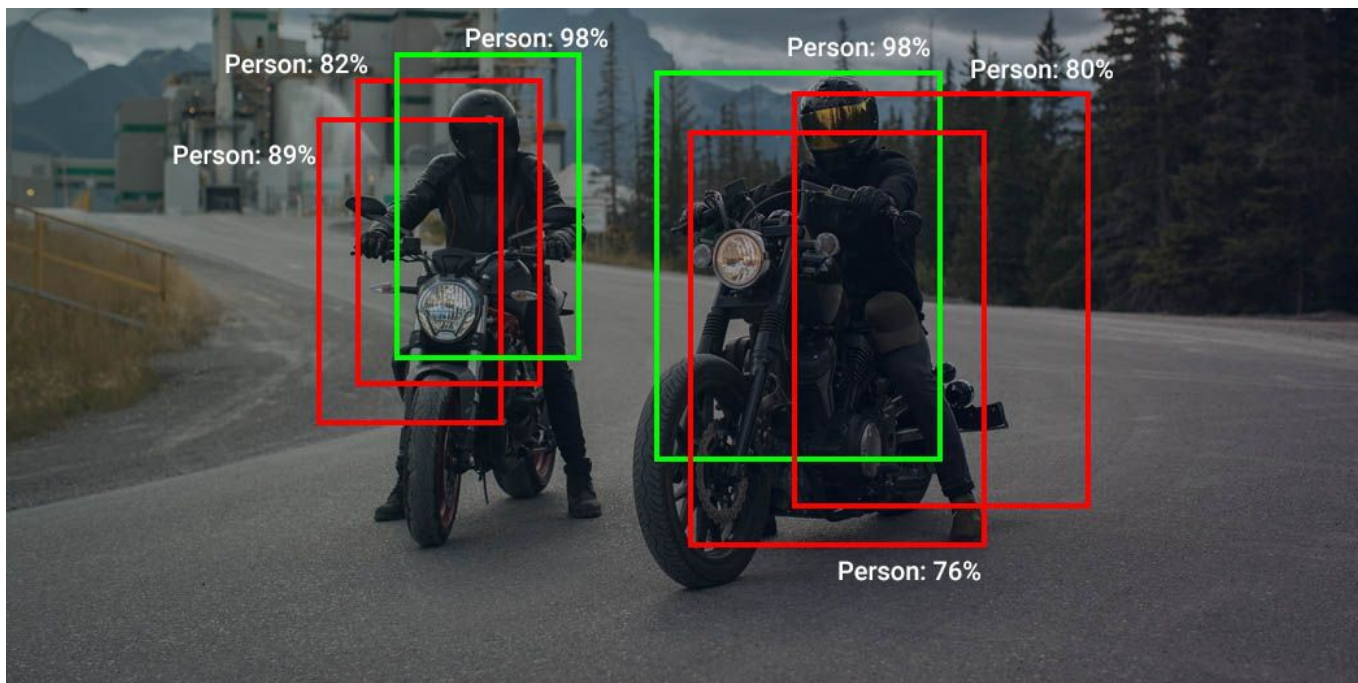
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \quad t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a)$$

$$Objectness_{score}(IoU) = \begin{cases} \text{Positive} \rightarrow \text{IoU} > 0.7 \\ \text{Positive} \rightarrow 0.5 < \text{IoU} \leq 0.7 \\ \text{Negative} \rightarrow \text{IoU} < 0.3 \\ \text{Not Negative/Positive} \rightarrow 0.3 \leq \text{IoU} \leq 0.5 \end{cases}$$

Non-Maximum Suppression (NMS)

Το RPN παράγει εν γένει “**πυκνά**” proposals.

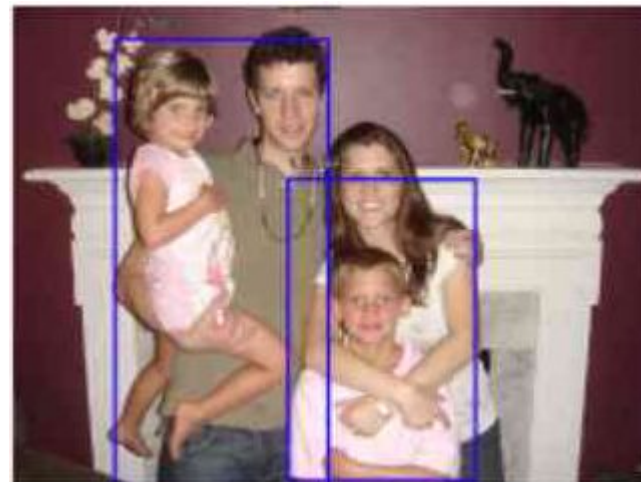
- Αυτό σημαίνει ότι προκύπτουν πολλαπλά proposals τα οποία επικαλύπτονται
 - Διαφέρουν όμως κατά την πιθανότητα πρόβλεψης (confidence)
- Με τη διαδικασία του **NMS** επιλέγονται τελικά τα **proposals** τα οποία έχουν την υψηλότερη πιθανότητα πρόβλεψης συγκριτικά με όσα επικαλύπτονται ($\text{IoU} > 0$)
 - Υπάρχει η δυνατότητα για κατωφλίωση της IoU σε υψηλότερη τιμή (π.χ. $\text{IoU} > 0.3$)



Non-Maximum Suppression (NMS)

Το NMS δεν αποτελεί όμως πανάκεια!

- Στην περίπτωση “**πυκνών**” ή (μερικώς) επικαλυπτόμενων αντικειμένων το NMS είναι προβληματικό
 - Εύρεση κατάλληλης τιμής κατωφλίσωσης της IoU
 - Πιο σύνθετη προσέγγιση:
 - soft-NMS
 - NMS networks
 - Confluence
 - etc...



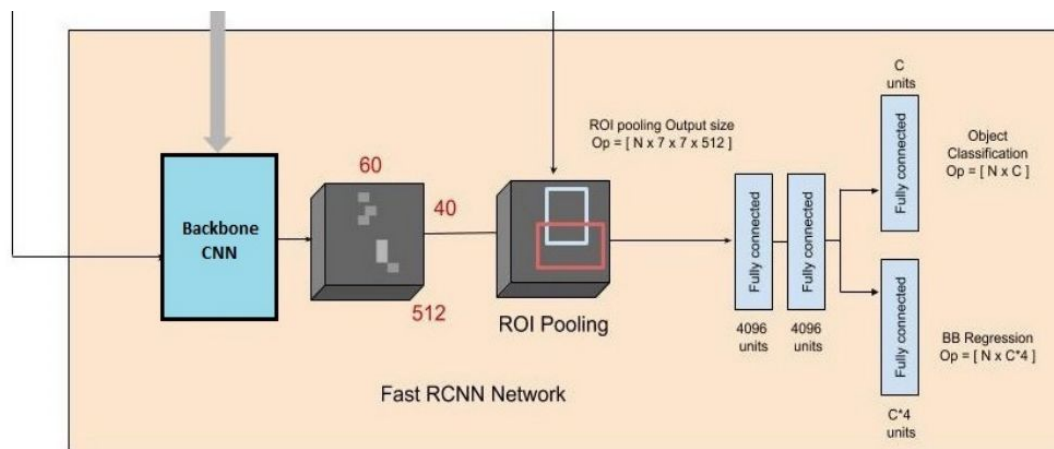
Fast R-CNN

Ο Fast R-CNN detector αλγόριθμος αποτελείται από :

- Ένα backbone CNN δίκτυο,
- Μια **ROI pooling** στρώση
 - Απομονώνει την περιοχή του feature map (από το backbone) που αντιστοιχεί σε ένα Object Proposal
 - Χωρίζει αυτή τη περιοχή σε έναν σταθερό αριθμό (grid) υποπαραθύρων (bins)
 - Εφαρμόζει max pooling σε αυτά τα υποπαραθύρα ώστε να προκύψει έξοδος σταθερού μεγέθους
- 2 πλήρως συνδεδεμένες στρώσεις που τελικά χωρίζονται σε 2 κλάδους (branches) για την **ταξινόμηση** του αντικειμένου και το **BB regression**

input

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

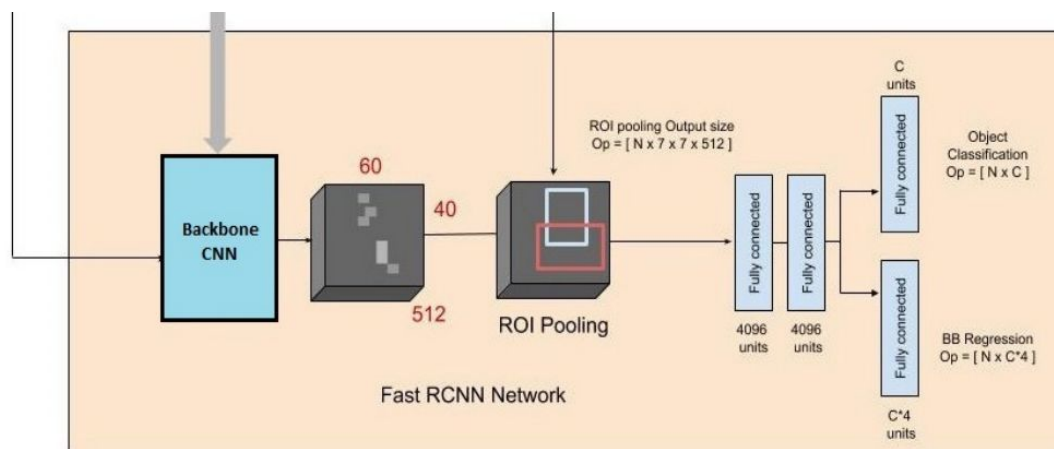


Fast R-CNN

Τα classification και regression μέρη εδώ είναι διαφορετικά από αυτά του RPN!

Ταξινόμηση αντικειμένου : C (πλήθος κατηγοριών) στοιχεία για κάθε μία από τις κατηγορίες του regression μέρους συμπεριλαμβανομένης μιας κατηγορίας για το background

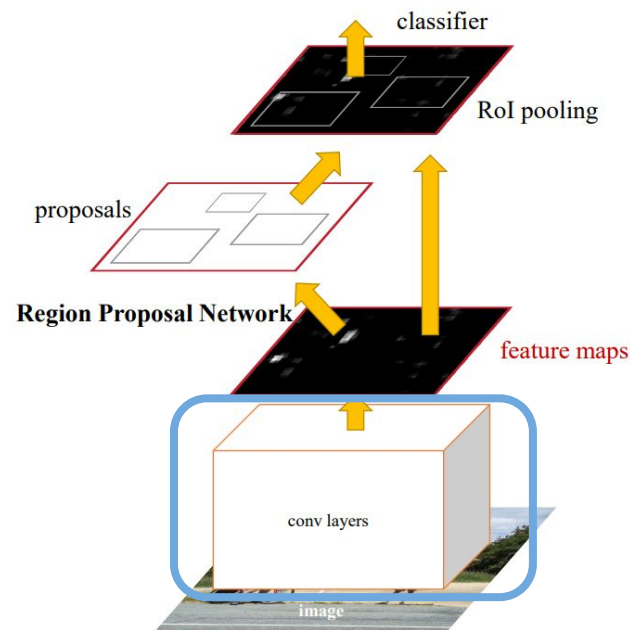
BB regression : χρησιμοποιείται για να βελτιώσει τις θέσεις των προβλεπόμενων BB των αντικειμένων αλλά, σε αντίθεση με το RPN, ο regressor είναι size agnostic και αναφέρεται συγκεκριμένα σε κάθε κατηγορία → κάθε κατηγορία έχει δικό της regressor με 4 παραμέτρους (διορθώσεις ανά συντεταγμένη) → $C \times 4$ στοιχεία



Faster R-CNN

Εκπαίδευση σε 4 βήματα!

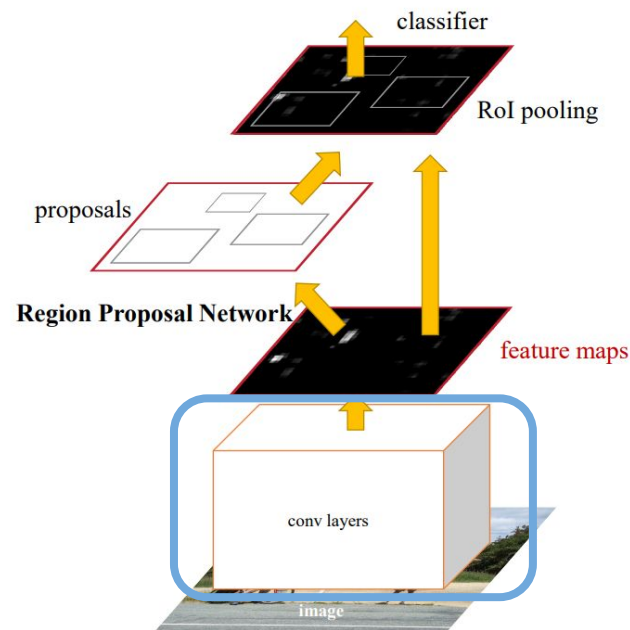
1. Το RPN δίκτυο εκπαιδεύεται **ανεξάρτητα** στο πρόβλημα του Region Proposal. Χρησιμοποιείται ένα **προεκπαιδευμένο στο ImageNet δίκτυο** και μετά γίνεται fine-tuning
2. Ο Fast R-CNN detector εκπαιδεύεται **ανεξάρτητα** στο πρόβλημα του Detection. Χρησιμοποιείται ένα **προεκπαιδευμένο στο ImageNet δίκτυο** και μετά γίνεται fine-tuning
 - Σταθερά βάρη του RPN
 - Τα proposals του RPN αξιοποιούνται για να εκπαιδευτεί ο Faster R-CNN



Faster R-CNN

Εκπαίδευση σε 4 βήματα!

3. Το RPN δίκτυο *αρχικοποιείται τώρα με τα βάρη από τον Faster R-CNN* και εκπαιδεύεται ώστε να γίνει finetune στο πρόβλημα του Region Proposal
 - Τα βάρη στις κοινές στρώσεις του RPN και του Fast R-CNN παραμένουν σταθερά
 - Ουσιαστικά εκπαιδεύονται μόνο τα βάρη που είναι **αποκλειστικά του RPN**
4. Εκπαιδεύεται ο detector Fast R-CNN ξανά χρησιμοποιώντας πλέον το *“τελικό” εκπαιδευμένο δίκτυο RPN*
 - Εκπαιδεύονται οι στρώσεις που βρίσκονται **αποκλειστικά στο detection κομμάτι του Faster R-CNN**



Single Shot Detectors

Τα δίκτυα τύπου **Single Shot Detector (SSD)** αποτελούν ίσως το χαρακτηριστικότερο παράδειγμα οικογένειας αλγορίθμων ανίχνευσης ενός σταδίου.

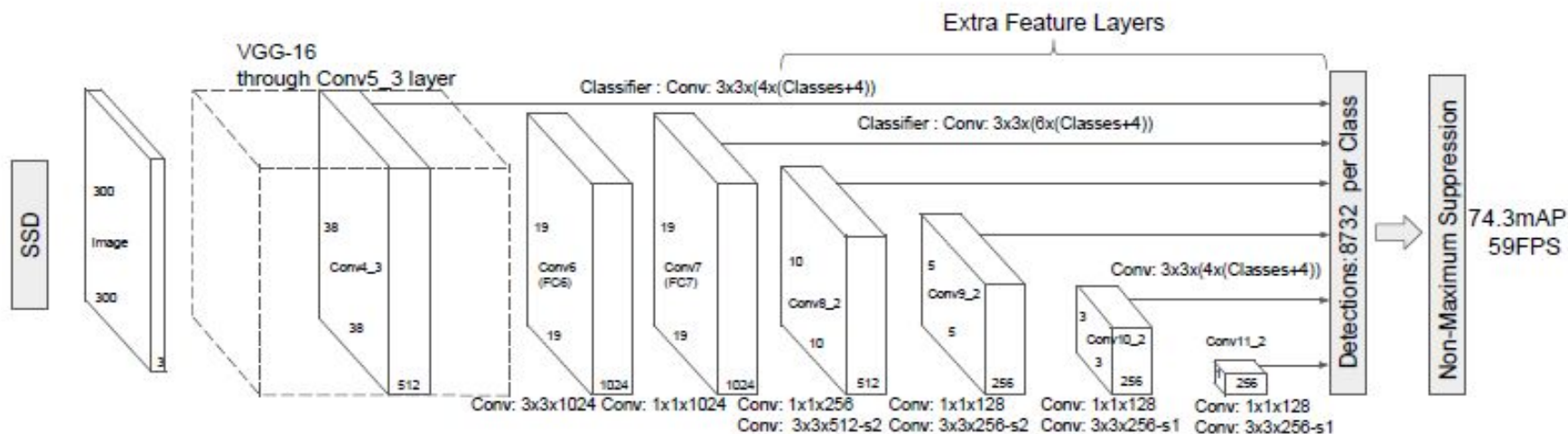
Πρακτικά, ένας SSD αποτελείται από το **encoder τμήμα** ενός τυπικού CNN. Τις περισσότερες φορές το αρχικό μέρος του encoder προέρχεται από κάποιο προεκπαιδευμένο δίκτυο (π. χ. vgg)

- Σε κατάλληλα επιλεγμένα feature maps του encoder τμήματος προσαρτώνται τα λεγόμενα **detection heads**, τα οποία πρόκειται για συνελικτικές στρώσεις 3×3
 - Σε κάθε detection head αντιστοιχούν συγκεκριμένα “**prior boxes**” (~ anchors)
 - Σε κάθε detection head εκτελούνται $num_{priors} * (num_{classes} + 4)$ φίλτρα
 - Τα $num_{priors} * num_{classes}$ φίλτρα αφορούν το τμήμα της ταξινόμησης
 - Τα $num_{priors} * 4$ φίλτρα αφορούν το τμήμα της παλινδρόμησης (localization)



Single Shot Detectors

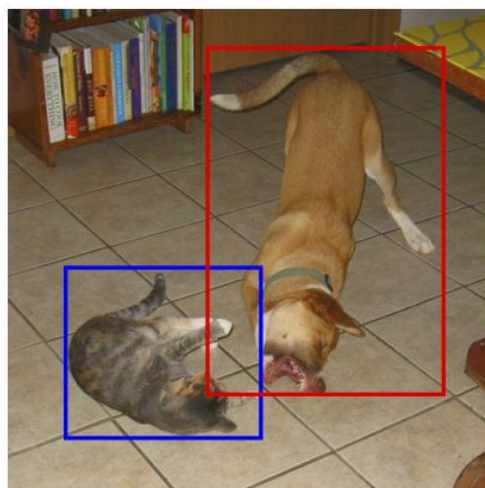
- Πολλές στρώσεις σε μια τυπική αρχιτεκτονική CNN → μειώνεται το μέγεθος του feature map
- Οι βαθιές στρώσεις καλύπτουν πιο μεγάλα receptive fields και αντιπροσωπεύουν πιο abstract αναπαραστάσεις, ενώ οι πιο ρηχές (shallow) στρώσεις καλύπτουν μικρότερα receptive fields
 - **Receptive fields** : ορίζεται ως η περιοχή του χώρου εισόδου που επηρεάζει ένα συγκεκριμένο χαρακτηριστικό του δικτύου (support)
- Βάσει αυτής της λογικής μπορούμε να αξιοποιήσουμε τις πιο “ρηχές” στρώσεις για να προβλέψουμε μικρά αντικείμενα και τις πιο βαθιές στρώσεις του δικτύου για να προβλέψουμε μεγάλα αντικείμενα
 - Μικρά αντικείμενα της εικόνας δεν χρειάζονται μεγάλα receptive fields



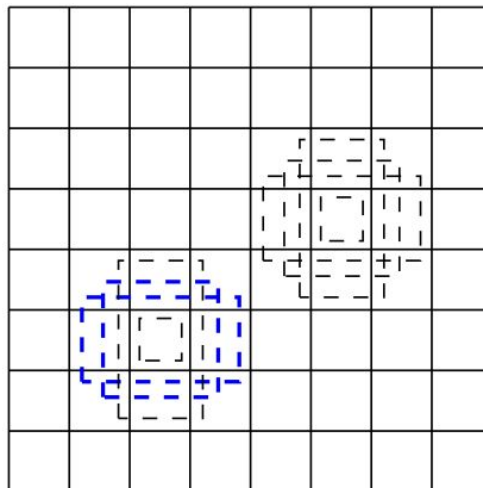
Single Shot Detectors

Τα δίκτυα τύπου **Single Shot Detector (SSD)** αποτελούν ίσως το χαρακτηριστικότερο παράδειγμα οικογένειας αλγορίθμων ανίχνευσης ενός σταδίου.

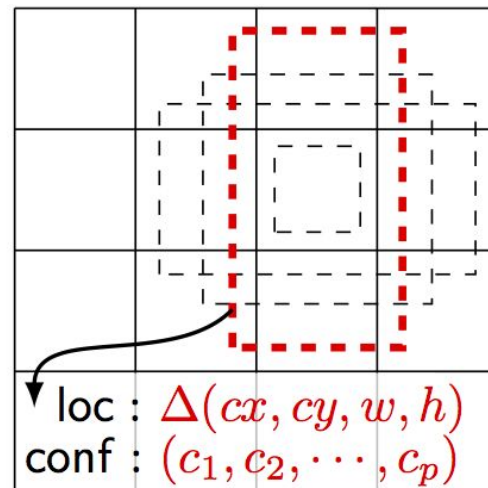
- Τα SSD δίκτυα αντιστοιχίζουν την κλίμακα ανίχνευσης σε ένα feature map
 - Όλα τα priors που συσχετίζονται με ένα feature map διαφέρουν μόνο κατά aspect ratio ή μικρο-διαφορές στο μέγεθος
- Σε κάθε “εμπρός-πέρασμα” του SSD προκύπτουν προβλέψεις για κάθε δυνατή θέση / κλίμακα / aspect ratio!
 - Όσα αντικείμενα ταξινομούνται στην κατηγορία υποβάθρου απορρίπτονται
 - Επειδή κάθε αντικείμενο επικαλύπτεται από πολλαπλά priors, επιλέγονται οι τελικές ανιχνεύσεις έπειτα από τη διαδικασία του **non-maximum suppression**



(a) Image with GT boxes



(b) 8×8 feature map

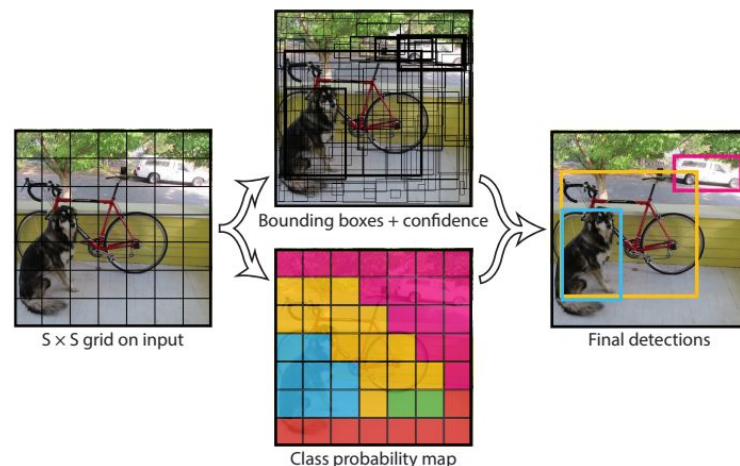


loc : $\Delta(cx, cy, w, h)$
conf : (c_1, c_2, \dots, c_p)

(c) 4×4 feature map

You Only Look Once (YOLO)

- Αλγόριθμος ανίχνευσης **ενός σταδίου**
- Η εικόνα χωρίζεται σε N κελιά (grids) όπου κάθε ένα από αυτά είναι υπεύθυνο για το detection και το localization του αντικειμένου που εμπεριέχει
- Αντίστοιχα αυτά τα κελιά κάνουν προβλέψεις για τις συντεταγμένες του bb σε σχέση με τις δικές τους (bb regression) μαζί με την κατηγορία του αντικειμένου και την πιθανότητα να βρίσκεται σε εκείνα τα κελιά (classification)
- Προκύπτουν αρκετές “διπλότυπες” προβλέψεις διότι πολλά κελιά προβλέπουν το ίδιο αντικείμενο με διαφορετικές προβλέψεις για τη θέση του BB
- Χρήση Non-Maximum Suppression (NMS) για να προκύψουν τα τελικά bbs



You Only Look Once (YOLO)

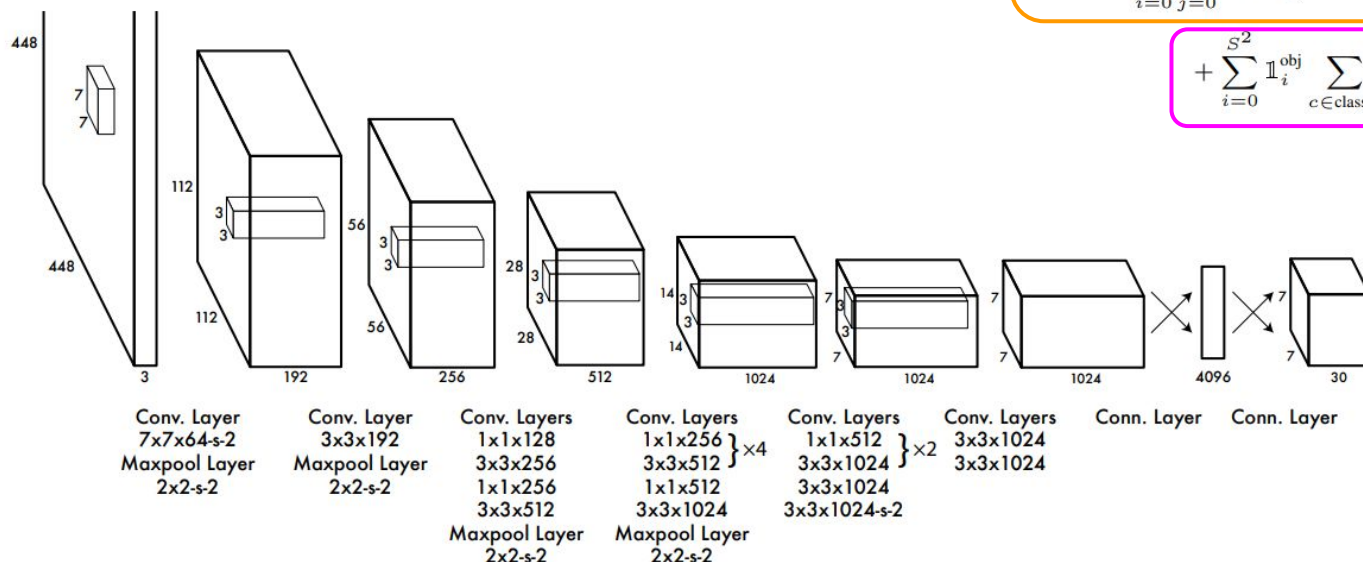
- 24 συνελικτικές στρώσεις & 2 πλήρως συνδεδεμένες στρώσεις
- Αρχικοποίηση των βαρών με προεκπαίδευση στο ImageNet
- Σύνθετη συνάρτηση κόστους
 - 2 όροι **bbox regression**
 - 2 όροι **Cell classification**
 - 1 όρος **κοινός classification + localization**
- Για κάθε κελί, υπολογίζονται **5 παράμετροι** ανά bb που εξετάζεται ($x, y, w, h, \text{confidence} \sim \text{IoU}_{\text{pred/gt}}$) + πιθανότητες ανά κατηγορία (C_i) $\rightarrow S \times S \times (5 \times B + C)$

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{i=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{i=0}^B \mathbb{I}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$



You Only Look Once (YOLO)

Πλεονεκτήματα

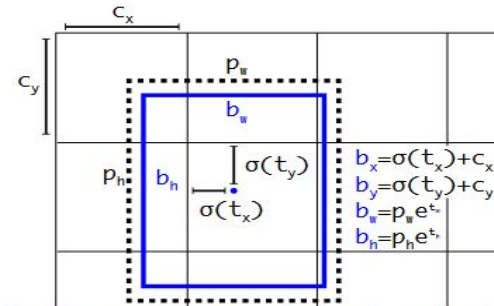
- Ταχύτητα (45 FPS)
- Το δίκτυο αντιλαμβάνεται μια γενικευμένη αναπαράσταση του αντικειμένου
 - Καλά αποτελέσματα προβλέψεων σε εικόνες από σέτ δεδομένων με πίνακες (π.χ. Picasso Dataset) μετά από εκπαίδευση σε πραγματικές εικόνες (διαφορετικά domains)



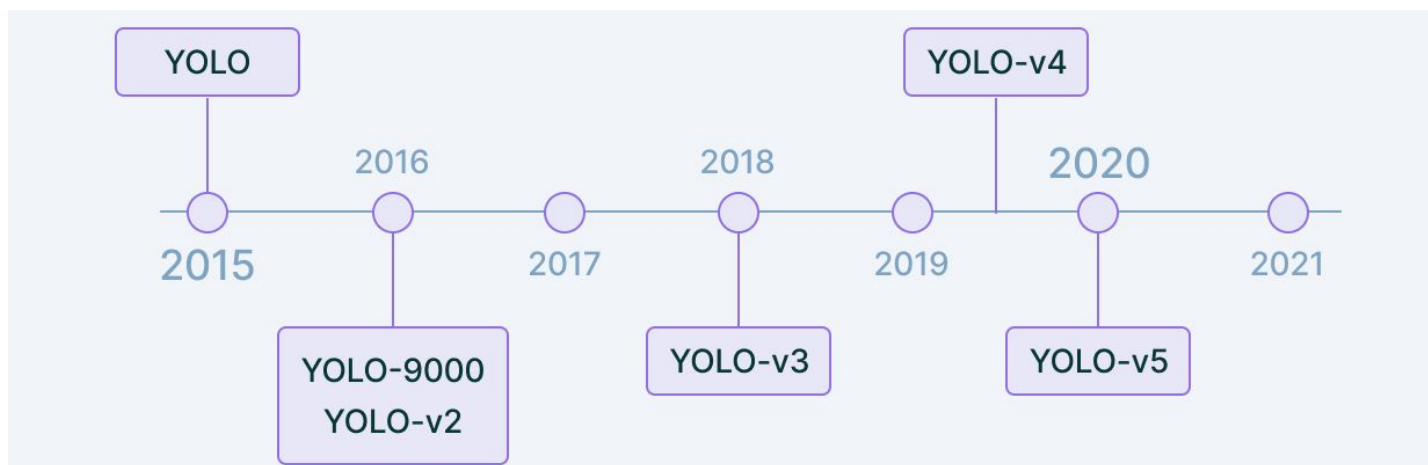
Μειονεκτήματα

- Περιορισμός στα πόσα κοντινά αντικείμενα μπορεί γίνει πρόβλεψη
 - Κάθε κελί προβλέπει μόνο 2 bb (στην αρχική δημοσίευση) και μπορεί να έχει μόνο μια κατηγορία
- Το μοντέλο δυσκολεύεται να προβλέψει αντικείμενα με διαφορετικούς λόγους μεγέθους σε σχέση με τους αντίστοιχους που εκπαιδεύτηκε
- Η συνάρτηση κόστους αντιμετωπίζει τα λάθη στα μικρά BB το ίδιο με τα μεγάλα BB
 - Μικρό λάθος σε μικρό BB μεγαλύτερη επίδραση στην IoU σε σχέση με μικρό λάθος σε ένα μεγάλο BB
 - Μεγαλύτερη πηγή λαθών είναι τα λάθη στο localization των αντικειμένων

YOLO : Διαφορετικές εκδοχές

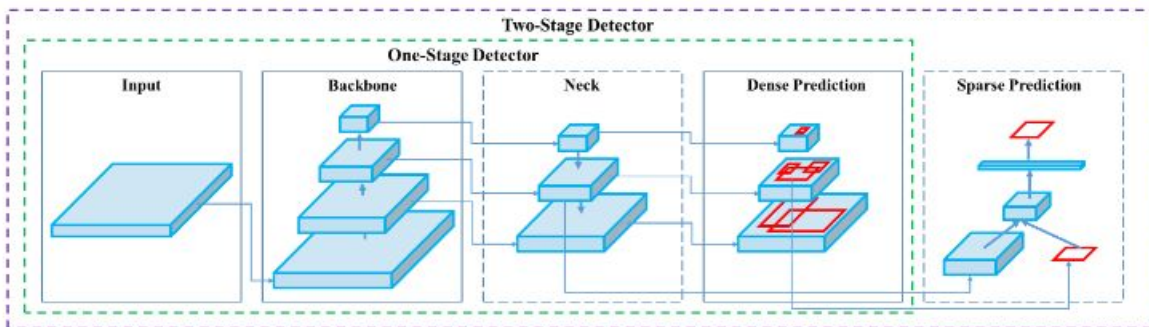


- **YOLO v2** : αντιμετώπιση του προβλήματος της αναγνώρισης μικρών αντικειμένων που βρίσκονται κοντά
 - Το κάθε κελί πλέον μπορεί να προβλέπει έως και 5 BB & εισαγωγή batch normalization και Anchors
- **YOLO 9000** : παρόμοια αρχιτεκτονική με της v2, αλλά σχεδιάστηκε έτσι ώστε να μπορούν να ανιχνευθούν αντικείμενα περισσότερων κατηγοριών από όσες το COCO dataset έχει
 - Χρήση κατηγοριών ταυτόχρονα από το ImageNet και το COCO dataset μέσω ενοποίησης της διαδικασίας ταξινόμησης και ανίχνευσης
 - Χρήση ενός αλγορίθμου ιεραρχικής ταξινόμησης, αφού κάποιες κατηγορίες είναι υποσύνολα σημασιολογικά κάποιων άλλων
 - Χειρότερη ακρίβεια από τη v2, όμως ικανότητα πρόβλεψης σε περισσότερες κατηγορίες

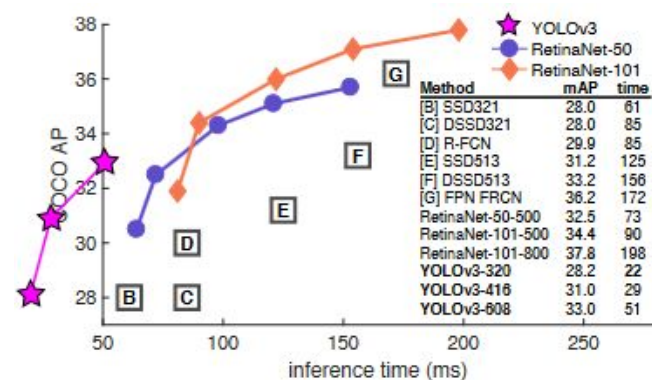


YOLO : Διαφορετικές εκδοχές

- **YOLO v3** : χρήση skip connections και χρήση ενός πιο σύνθετου backbone δικτύου (DarkNet-53)
 - Πρόβλεψη σε 3 διαφορετικές κλίμακες
- **YOLO v4**: Προσθήκη Weighted Residual Connections, Cross Mini Batch Normalization, Self Adversarial Training, Mish Activation, Cross Stage Partial Skip Connections κ.ά.
- **YOLO v4 Tiny** : Ακόμη πιο γρήγορη εκδοχή του αλγορίθμου & σε περισσότερα FPS
- **YOLO v5**: Open source project που εμπεριέχει εκπαιδευμένες εκδοχές του YOLO προεκπαιδευμένο στο COCO dataset



YoLO v4 architecture

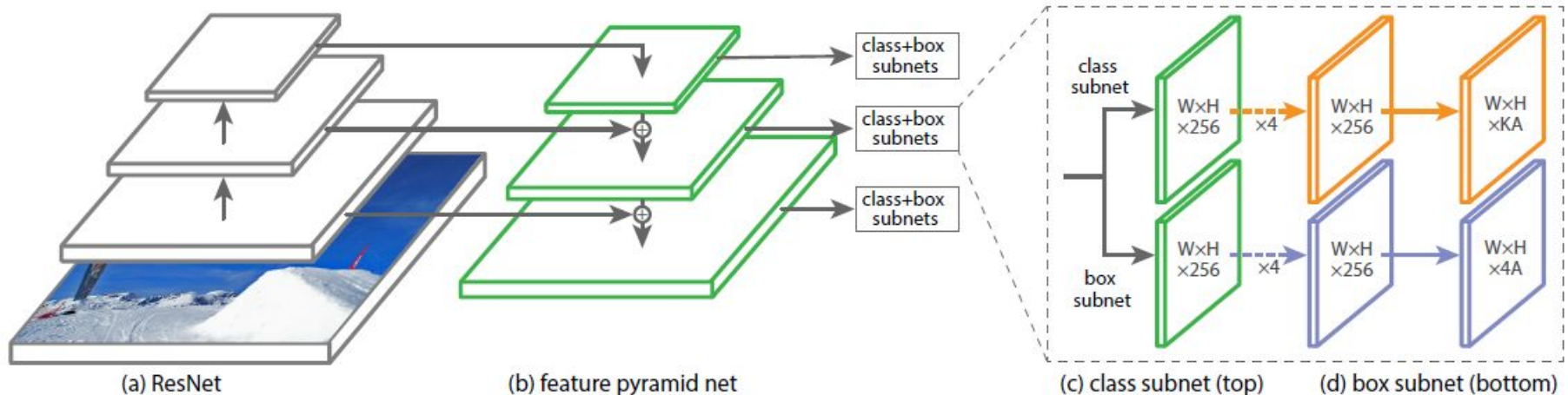


RetinaNet

Ο αλγόριθμος ανίχνευσης **RetinaNet** ανήκει στην οικογένεια ανιχνευτών ενός σταδίου. Αρχικά σχεδιάστηκε για να βελτιώσει την ακρίβεια αυτής της κατηγορίας αλγορίθμων προσεγγίζοντας ακρίβειες εντοπισμού εφάμιλλες με ανιχνευτές 2 σταδίων όπως ο Faster-RCNN

Ως αρχιτεκτονική αποτελείται από 3 διακριτά μέρη:

1. Το προεκπαιδευμένο **backbone** encoder CNN
 - Ελεύθερη επιλογή, καθώς δεν συνδέεται κάποιο detection head σε feature map του backbone
2. Το Feature Pyramid Network (**FPN**)
 - Χρησιμοποιείται για να δημιουργήσει χαρακτηριστικά σε πολλαπλές κλίμακες ώστε να προσαρτηθούν τα detection heads
3. Τα συνελικτικά **detection heads**
 - Όμοια με αυτά του SSD, απλά μεσολαβούν 4 συνελικτικές στρώσεις 256 φίλτρων η κάθε μία



RetinaNet

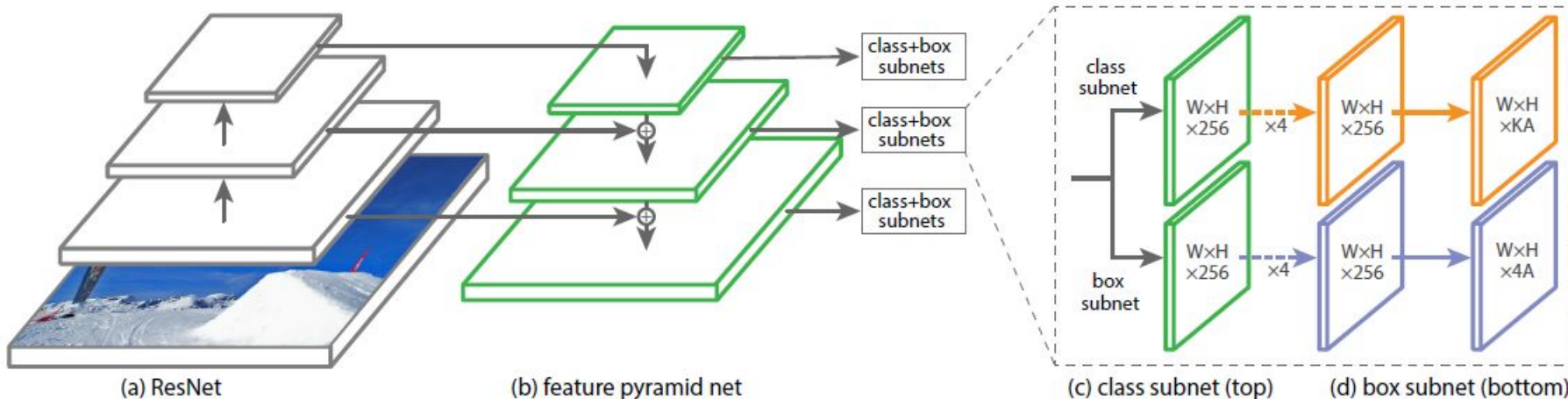
Πέραν της αρχιτεκτονικής του όμως ο RetinaNet ξεχωρίζει και για την ιδιαίτερη συνάρτηση κόστους “**focal loss**”

- Η “**focal loss**” αποτελεί μία **παραλλαγή της cross entropy** κατά την οποία δίνεται μεγαλύτερο βάρος στα “δύσκολα” αρνητικά δείγματα τα οποία ταξινομούνται λάθος κατά την εκπαίδευση
 - Αποτελεί μία λύση στο πρόβλημα ανισοκατανομής των κατηγοριών foreground-background κατά την εκπαίδευση ανιχνευτών ενός σταδίου

$$\text{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise.} \end{cases}$$

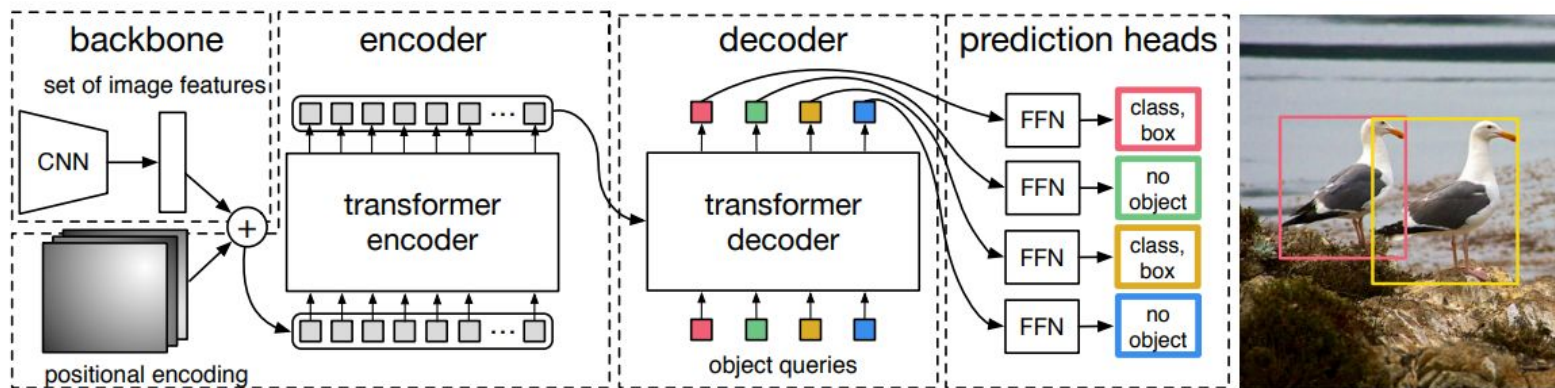
$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t).$$

- Η focal loss μπορεί να αξιοποιηθεί από οποιαδήποτε αρχιτεκτονική ενός σταδίου!
 - Το RPN μόνο του αποτελεί επίσης έναν (δυαδικό) ανιχνευτή ενός σταδίου...



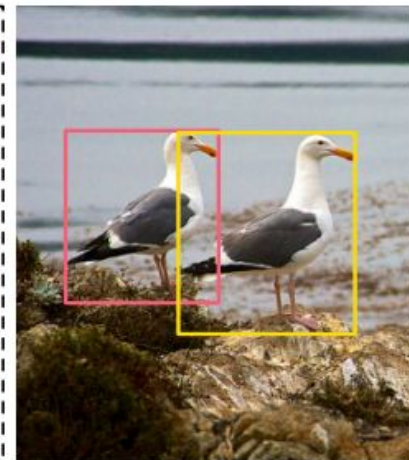
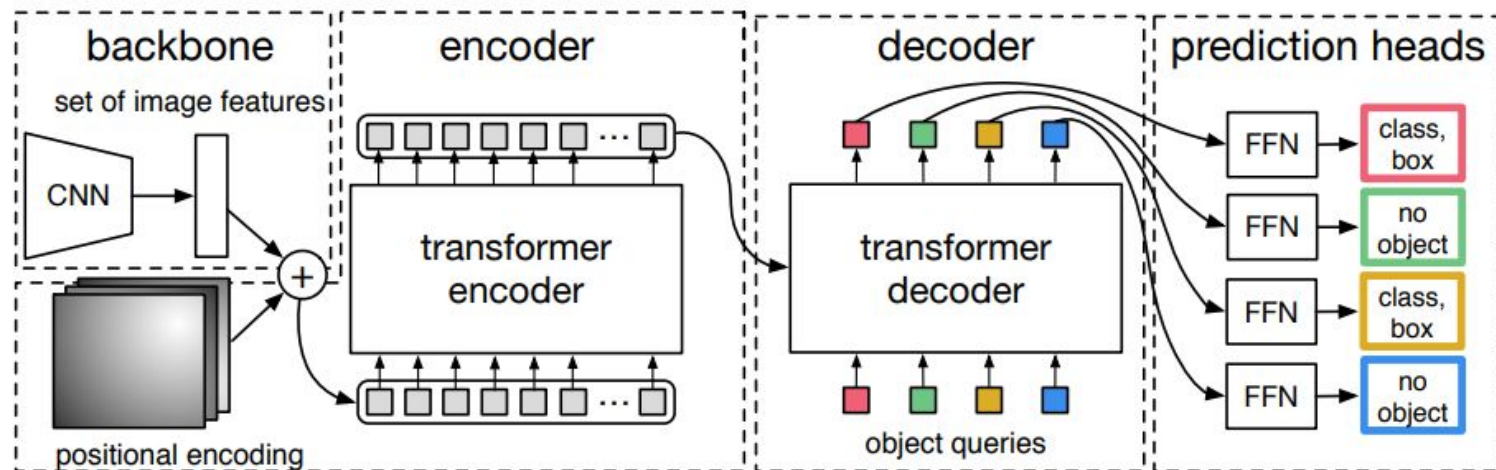
Ο αλγόριθμος DETR

- “*End-to-end Object Detection with Transformers*” της Meta AI το 2020
- Απλή end-to-end ενιαία αρχιτεκτονική που μπορεί να “γενικεύσει” καλά με περιορισμένο αριθμό παραμέτρων
- Το πρόβλημα της ανίχνευσης αντικειμένων μοντελοποιείται ως ένα **image-to-set** πρόβλημα
 - Δεν απαιτούνται αλγόριθμοι που διαθέτουν μια έννοια εμπειρικής / είναι χειροποίητοι όπως π.χ. non-maximum suppression
 - Δεν χρησιμοποιούνται anchors που απλώς μοντελοποιούν τη (prior) γνώση μας για το πρόβλημα (task)
- Βασικές καινοτομίες είναι η χρήση μιας Transformer Encoder-Decoder αρχιτεκτονικής και μιας set-based καθολικής (global) συνάρτησης κόστους που εξαναγκάζει τις προβλέψεις να είναι μοναδικές μέσω bipartite matching
- Δεδομένου ενός σταθερού μικρού συνόλου queries που αφορούν τα αντικείμενα της εικόνας, ο DETR συνυπολογίζει τις σχέσεις μεταξύ των αντικειμένων και το context όλης της εικόνας ώστε να προκύψουν απευθείας οι τελικές προβλέψεις ταυτόχρονα



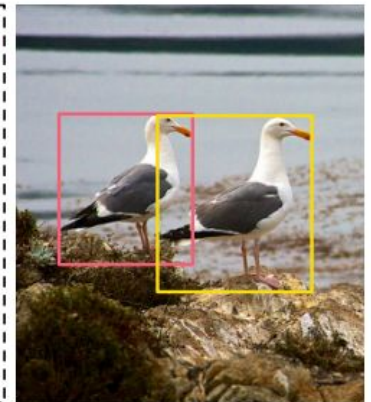
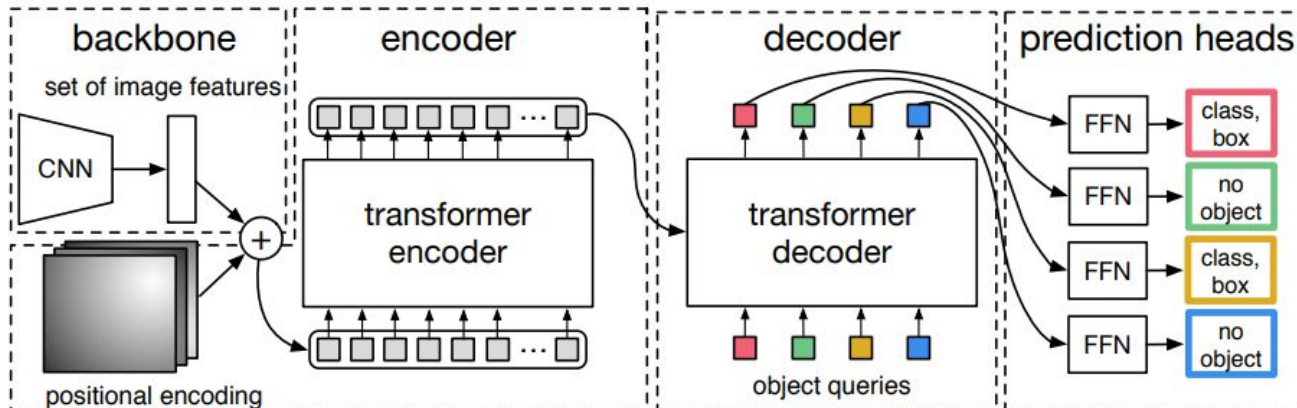
Ο αλγόριθμος DETR

- CNN backbone αρχιτεκτονική ως feature extractor $f \in \mathbb{R}^{C \times H \times W}$, $C=2048$, $H=\frac{H_0}{32}$, $W=\frac{W_0}{32}$
- 1x1 συνέλιξη ώστε να προκύψει μικρότερη διάσταση d των καναλιών $z_0 \in \mathbb{R}^{d \times H \times W}$
- Flattening στη χωρική διάσταση (d×HW)
- 2D χωρικό positional encoding
 - Προστίθεται πριν από κάθε πράξη attention
- Κλασικό encoder τμήμα της Transformer αρχιτεκτονικής



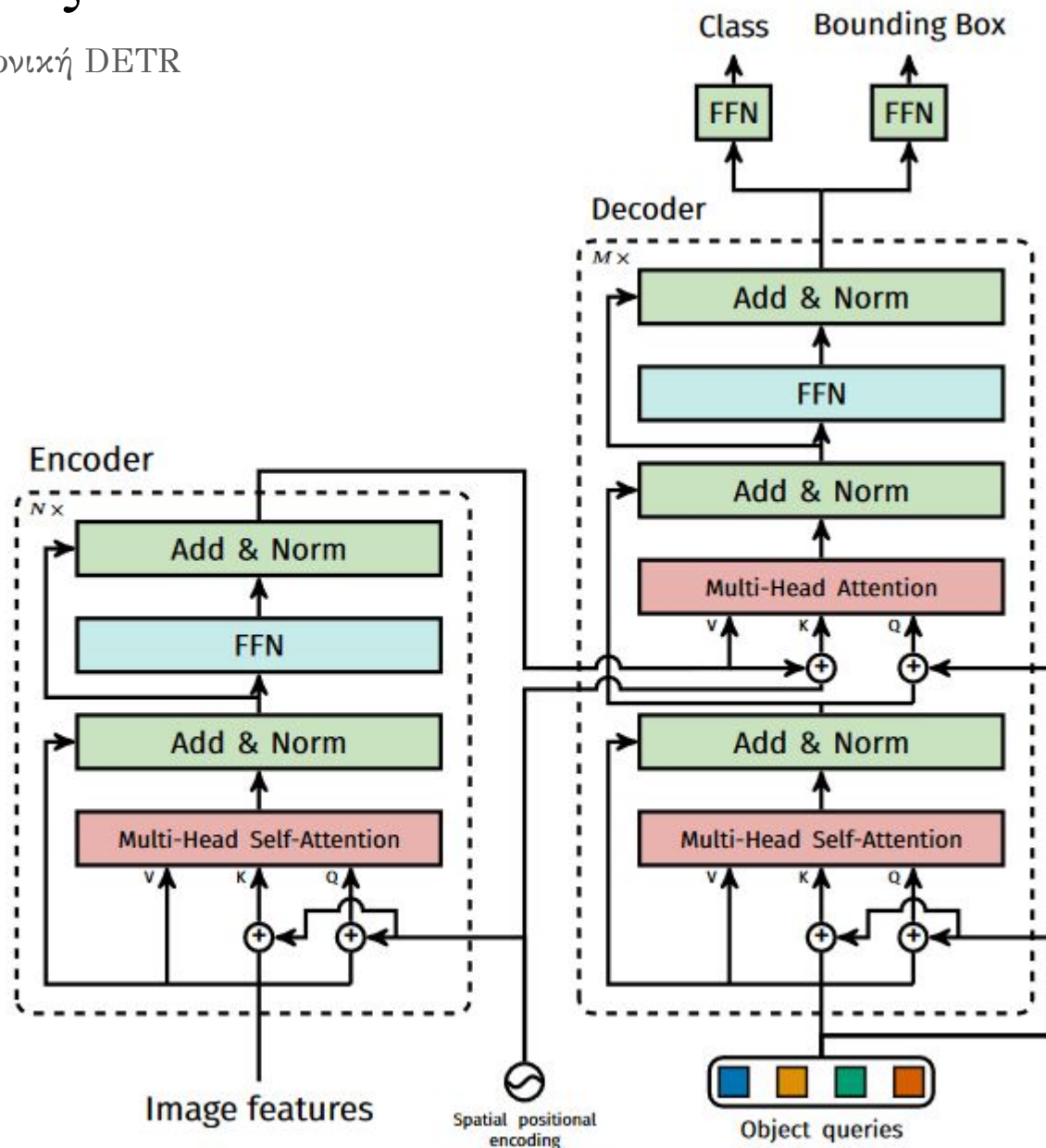
Ο αλγόριθμος DETR

- Κλασικό decoder τμήμα της Transformer αρχιτεκτονικής
 - Μετασχηματισμός N embeddings μεγέθους d μέσω multi-head self & cross attention μηχανισμών
 - Πρόβλεψη σε N αντικείμενα παράλληλα σε κάθε decoder στρώση, αντί του κλασικού Transformer όπου γίνεται πρόβλεψη σε ένα στοιχείο της ακολουθίας εξόδου τη φορά
- Αρχικά εισάγονται “μηδενικά” queries στον decoder!
 - Σε αυτά τα queries προστίθενται **learnable positional encodings** που ονομάζονται **object queries**
 - Τα **object queries** προστίθενται στην είσοδο κάθε στρώσης attention
 - Στην 1η στρώση του decoder ο 1ος μηχανισμός self-attention μπορεί να παραλειφθεί!
 - Τα object queries δρουν εν είδη “anchors” για τον DETR
- Ο decoder αναλαμβάνει να μετασχηματίσει αυτά τα queries σε ένα σύνολο από embeddings στην έξοδο
- Τελικά από αυτά τα τελικά embeddings προκύπτουν οι συντεταγμένες των BB και οι κατηγορίες των αντικειμένων χρησιμοποιώντας ένα feed-forward δίκτυο (N τελικές προβλέψεις)
 - Προστίθεται μία κατηγορία για το “background”



Ο αλγόριθμος DETR

- Τελική αρχιτεκτονική DETR



Ο αλγόριθμος DETR

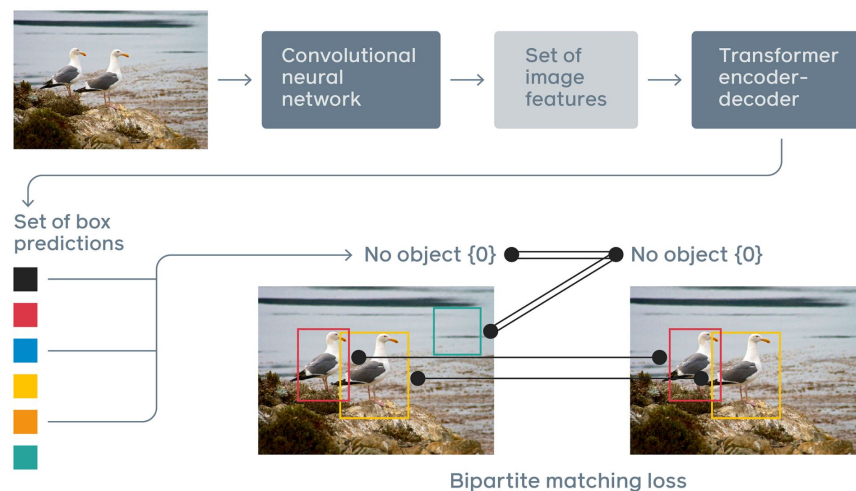
- Bipartite matching problem \rightarrow πρόβλημα αντιστοίχισης
- Βέλτιστο bipartite matching μεταξύ των προβλεπόμενων και των ground truth αντικειμένων
 - Τυπικά επιλύονται από αλγορίθμους σαν τον Hungarian algorithm

$$\hat{\sigma} = \arg \min_{\sigma \in \mathfrak{S}_N} \sum_i^N \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}),$$

- Βελτιστοποίηση classification και bbox regression τμήματος της Bipartite-loss function

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^N \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \right]$$

$\lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} \|b_i - \hat{b}_{\sigma(i)}\|_1$ where $\lambda_{\text{iou}}, \lambda_{\text{L1}} \in \mathbb{R}$ are hyperparameters.



Μετρικές αξιολόγησης

- Ο διαγωνισμός Pascal VOC2007 εισήγαγε τις εξειδικευμένες μετρικές αξιολόγησης της οικογένειας μετρικών “Average Precision” (AP) για προβλήματα ανίχνευσης.
- Η AP υπολογίζεται ως το μέσο precision της ανίχνευσης για διάφορες τιμές του recall
- Η AP μετατρέπεται στην meanAP για προβλήματα ανίχνευσης αντικειμένων σε πολλαπλές κατηγορίες ως ο μ.ο. των επιμέρους AP ανά κατηγορία
- Για να καταστεί δυνατή η μέτρηση **Precision** και **Recall** οι ανιχνεύσεις αντιστοιχίζονται βάσει μέγιστου IoU με τα δεδομένα αληθείας. Ορίζεται επίσης ένα **κατώφλι T** κάτω από το οποίο η προαναφερθείσα αντιστοίχιση απορρίπτεται.
- Η mAP συνήθως αναφέρεται σε κατώφλι ανάθεσης $T=0.5$ και συναντάται και ως mAP_{50}

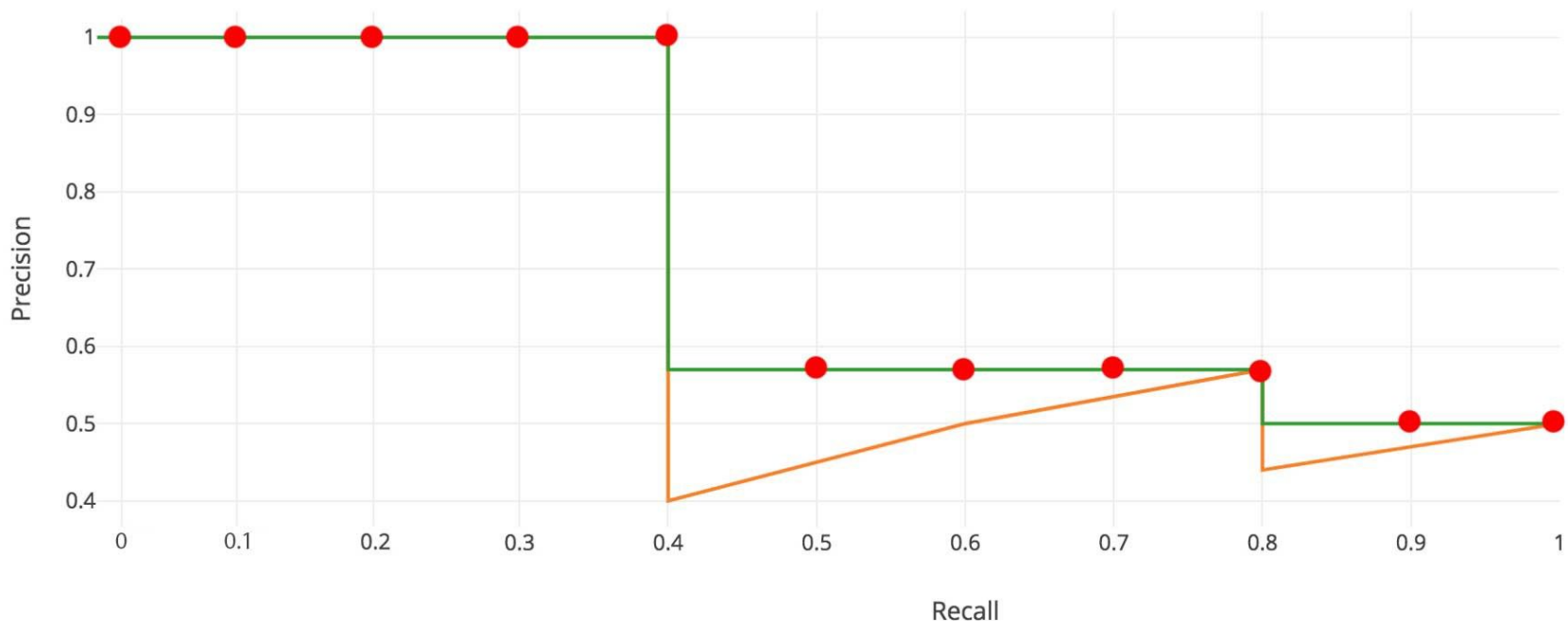
Το 2014 το σετ δεδομένων MS-COCO εισήγαγε την μετρική $mAP_{0.5:0.95}$ η οποία έχει επικρατήσει ως η σημαντικότερη μετρική αξιολόγησης και προκύπτει ως ο μ.ο. των mAP για πολλαπλές τιμές κατωφλίων από $T=0.5$ (αδρή τοπικοποίηση) έως $T=0.95$ (άριστη τοπικοποίηση)

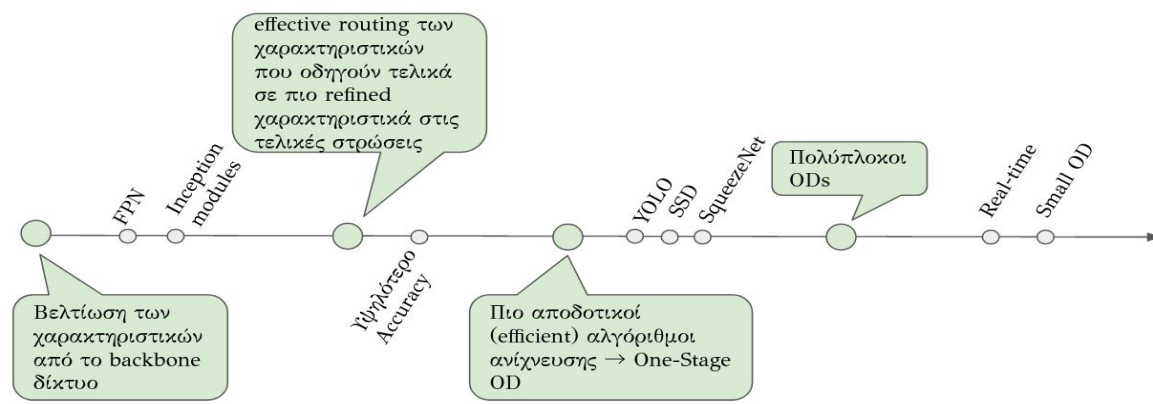
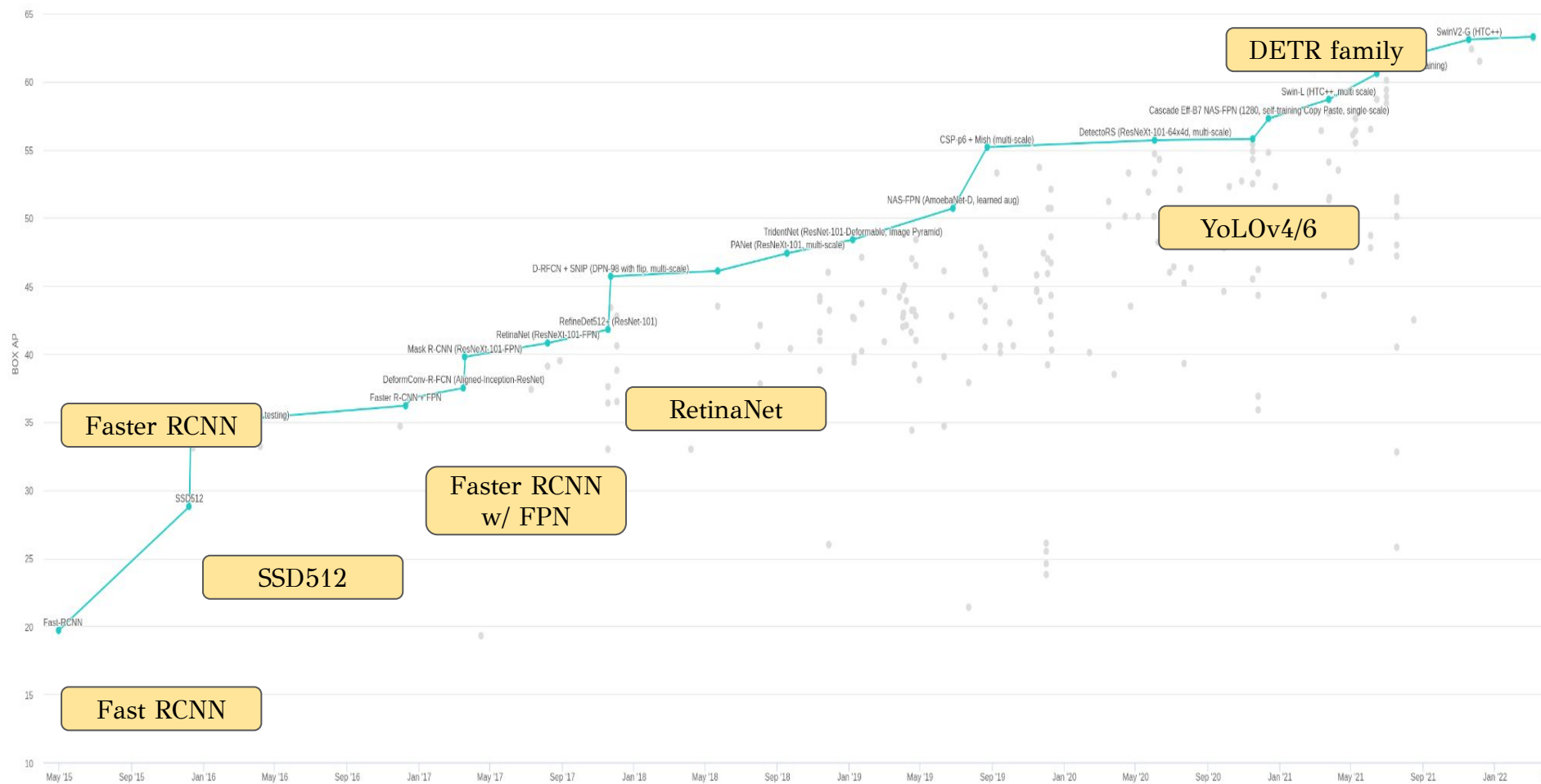
Μετρικές αξιολόγησης AP & mAP

➤ Average Precision (AP) :

- Υπολογίζεται το διάγραμμα Precision/Recall για κάθε confidence threshold ανά κατηγορία
- Μετασχηματισμός του διαγράμματος ώστε η καμπύλη να είναι φθίνουσα, παίρνοντας σε κάθε θέση τη μέγιστη από τις επόμενες τιμές
- Το AP προκύπτει ως ο μέσος όρος Precision τιμών από 11 ίσα χωρισμένων δειγμάτων Recall τιμών ανά κατηγορία.

➤ Mean Average Precision (mAP) : Ο μέσος όρος των ανά κατηγορία AP τιμών.





Σετ δεδομένων και frameworks

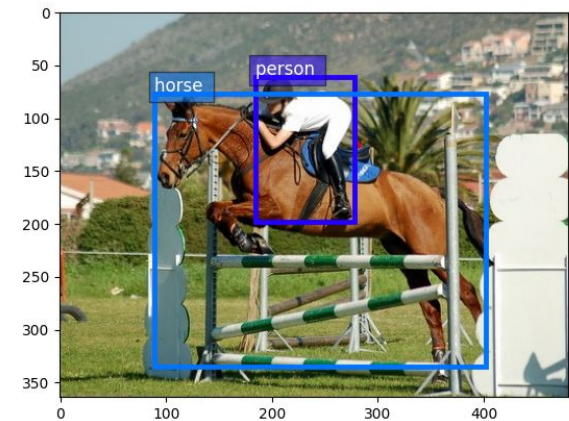
Κλασικά σετ δεδομένων:

- Pascal VOC2007/2012
- MS-COCO
- +++



Γεωχωρικά σετ δεδομένων:

- Kitti
- DOTA_{v2}
- Airborne Maritime Dataset
- +++



Frameworks:

- Tensorflow Object Detection API (tensorflow)
- Detectron2 (pytorch)
- MMDetection (pytorch)
- +++



Object Detection
API





RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων Δεδομένων Παρατήρησης Γης



GitHub

<https://github.com/rslab-ntua>