



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Ήλεκτρονικής και Υπολογιστών

Μετάδοση γνώσης αλγορίθμων εντοπισμού αντικειμένων
σε συστήματα πραγματικού χρόνου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ Σ. ΑΘΑΝΑΣΙΑΔΗ

Επιβλέπων: Λουκάς Πέτρου
Καθηγητής Α.Π.Θ.

Εργαστήριο Ευφών Συστημάτων & Τεχνολογίας Χορηγικού
Θεσσαλονίκη, 2018



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

Μετάδοση γνώσης αλγορίθμων εντοπισμού αντικειμένων σε 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
Καθηγητής Α.Π.Θ.

.....
Κάποιος
Καθηγητής Α.Π.Θ.

.....
Κάποιος άλλος
Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, ????? 2018

(Υπογραφή)

.....

Ιωάννης Αθανασιάδης

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

© 2018 -- All rights reserved



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών
Εργαστήριο Ευφών Συστημάτων & Τεχνολογίας **Χογισμικού**

Copyright ©--All rights reserved Ιωάννης Αθανασιάδης, **·**
Με επιφύλαξη παντός δικαιώματος.

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

Ευχαριστίες

Περίληψη

Λέξεις Κλειδιά

Abstract

Keywords

Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	8
Κατάλογος σχημάτων	9
Κατάλογος πινάκων	11
1 Νευρωνικά Δίκτυα για Εντοπισμό Αντικειμένων	15
1.1 Εισαγωγή	15
1.2 SqueezeDet-Net [61] [29]	15
1.3 YOLO[48]	19
1.4 YOLO9000 [49]	21
1.5 R-CNN [21]	23
1.6 Fast R-CNN [20]	24
1.7 Faster R-CNN [50]	25
1.8 Mask R-CNN [25]	25
1.9 XNOR-Net [46]	29
1.10 SSD [38]	30
1.11 Σύνοψη	31
2 Μετάδοση Γνώσης	35
2.1 Εισαγωγή	35
2.2 Πρόβλημα	35
2.3 Μία ιδέα για τη μετάδοση γνώσης [53]	36
2.4 Ορισμός της Μετάδοσης Γνώσης [42, 23]	36
2.5 Σχέση μετάδοσης γνώσης και μάθησης πολλαπλών έργων	38
2.6 Διάφορα σενάρια μετάδοσης γνώσης	38
2.6.1 Επαγωγική μετάδοση γνώσης	38
2.6.2 Μεταγωγική μετάδοση γνώσης	41
2.6.3 Άνευ επίβλεψης μετάδοση γνώσης	45
2.7 Μεταφερσιμότητα παραμέτρων νευρωνικών δικτύων επεξεργασίας εικόνας [62]	45
2.8 Αρνητική μετάδοση Γνώσης	48
2.9 Καταστροφική λήθη [34]	49
2.10 Προσβασιμότητα	51

3	Πειραματισμοί και αποτελέσματα	53
3.1	Το πρόβλημα και η προτεινόμενη λύση	53
3.2	Το πείραμα	53
3.3	Αυτόματη επιλογή υπερπαραμέτρων	54
3.4	Επιτάχυνση του δικτύου SqueezeDet	55
3.5	Σύνοψη & Συμπεράσματα	55
4	Υλικό & λογισμικό	57
4.1	Περιγραφή του υλικού	57
4.2	Βασικά στοιχεία λογισμικού υλοποίησης	57
4.3	Περιγραφή της υλοποίησης	57
4.3.1	supervisor	58
4.3.2	hypervisor	60
4.3.3	Περιγραφή πακέτων λογισμικού τρίτων	61
5	Μελλοντικές επεκτάσεις	63
	Βιβλιογραφία	64
	Βιβλιογραφία	65

Κατάλογος σχημάτων

1.1	Αρχιτεκτονική SqueezeDet	16
1.2	Αρχιτεκτονική Fire module	18
1.3	Αρχιτεκτονική SqueezeNet	19
1.4	Αρχιτεκτονική YOLO	20
1.5	Αρχιτεκτονική RCNN	23
1.6	Αρχιτεκτονική Fast-RCNN	24
1.7	Η οντότητα RPN	26
1.8	RoIAlign	27
1.9	Παράδειγμα RoIAlign	28
1.10	Αρχιτεκτονική Mask R-CNN	28
1.11	XNOR block vs typical CNN block	29
1.12	Αποτύπωση της συνέλιξης στο XNOR Net	30
2.1	Η διαδικασία μετάδοσης γνώσης	38
2.2	Αρχιτεκτονική νευρωνικής μάθησης δομικής αντιστοιχίας	43
2.3	Οι κατανομές των παραμέτρων του πρώτου και του τελευταίου επιπέδου	47
2.4	Τα αποτελέσματα της μετάδοσης γνώσης από το πείραμα μεταφερσιμότητας παραμέτρων	48
2.5	Γράφος μεταφερσιμότητας	49
2.6	Αλγόριθμος EWC	51
4.1	Η λειτουργία της υλοποίησης	58

Κατάλογος πινάκων

1.1	Χαρακτηριστικά του δικτύου YOLO9000	22
1.2	Σύγκριση δικτύων για feature extraction	32
1.3	Σύγκριση δικτύων για εντοπισμό αντικειμένων	32
3.1	My caption	55

Πίνακας Ακρωνυμίων

Παρακάτω βρίσκεται η λίστα εννοιών που συναντιούνται συχνά στο κείμενο.

CNN Convolutional Neural Network, συνελεκτικό Νευρωνικό Δίκτυο

CPU Central Processing Unit

Frame Καρέ

GPU Graphics Processing Unit

layer Επίπεδο του νευρωνικού δικτύου

mAP medium Average Precision. Μεσαία μέση ακρίβεια

MTL Multi-Task Learning. Μάθηση πολλών έργων ταυτοχρόνως

NMS Non Maximum Suppresion. Αλγόριθμος ο οποίος αν δύο περιβλήματα τέμνονται, τότε κρατάει μόνο αυτό με το μεγαλύτερο σκορ

ReLU Rectified Linear Unit. Πρόκειται για την μη γραμμική συνάρτηση ενεργοποίησης $f(x) = \max(0, x)$



RoI Region of Interest

TL Transfer Learning. Μετάδοση γνώσης


Κεφάλαιο 1

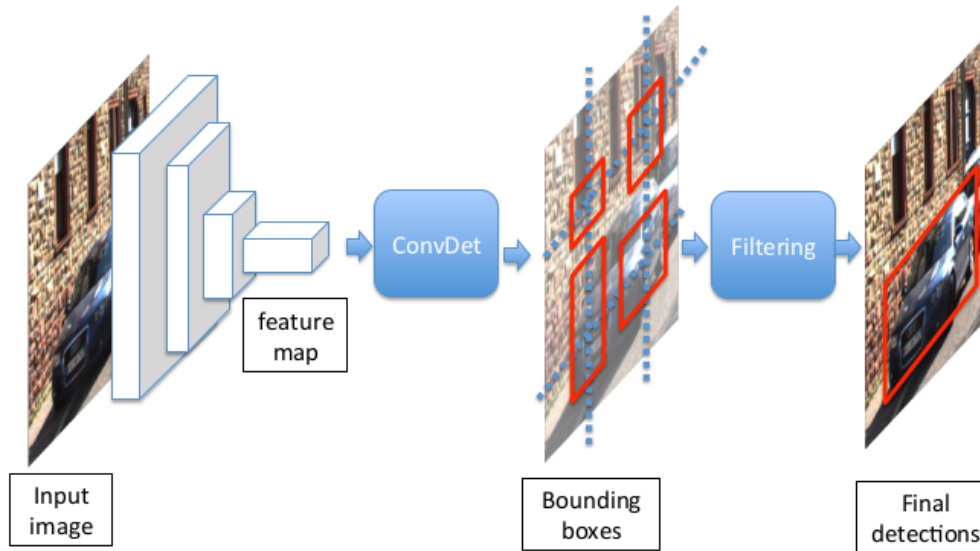
Νευρωνικά Δίκτυα για Εντοπισμό Αντικειμένων

1.1 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 συνεπίξεις για την αναγνώριση χειρόγραφων αριθμών είχε ήδη προταθεί το 1998 [64] και το 2006 [36]. 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 [35] το οποίο είναι ένα νευρωνικό δίκτυο πολλών επιπέδων (deep) το οποίο έχει επίπεδα συνεπίξουν την είσοδό τους με τα βάρη τους (convolutional) τό 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 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.

1.2 SqueezeDet-Net [61] [29]

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[35], επιτυγχάνει την ίδια ακρίβεια με 50x μικρότερο μέγεθος παραμέτρων. Η είσοδος που δέχεται είναι το ήδη επεξεργασμένο αποτέλεσμα του SqueezeNet.



Σχήμα 1.1: 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 κατανομημένα $W \times H$ 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:

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

Το SqueezeNet ~~με τη σειρά~~ αποτελεί ιδανική υλοποίηση του επιπέδου συνελίξεων του SqueezeDet γιατί στοχεύει στον μικρό αριθμό παραμέτρων. Επίσης κατά τη **σχεδίασή** 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 [26] και GoogleNet [56] το νευρωνικό αποτελείται από μικρότερες

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

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



Προκειμένου να επιτευχθεί η στρατηγική 2 απαιτείται $s_{1 \times 1} < (e_{1 \times 1} + e_{3 \times 3})$. 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[48]. 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:

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

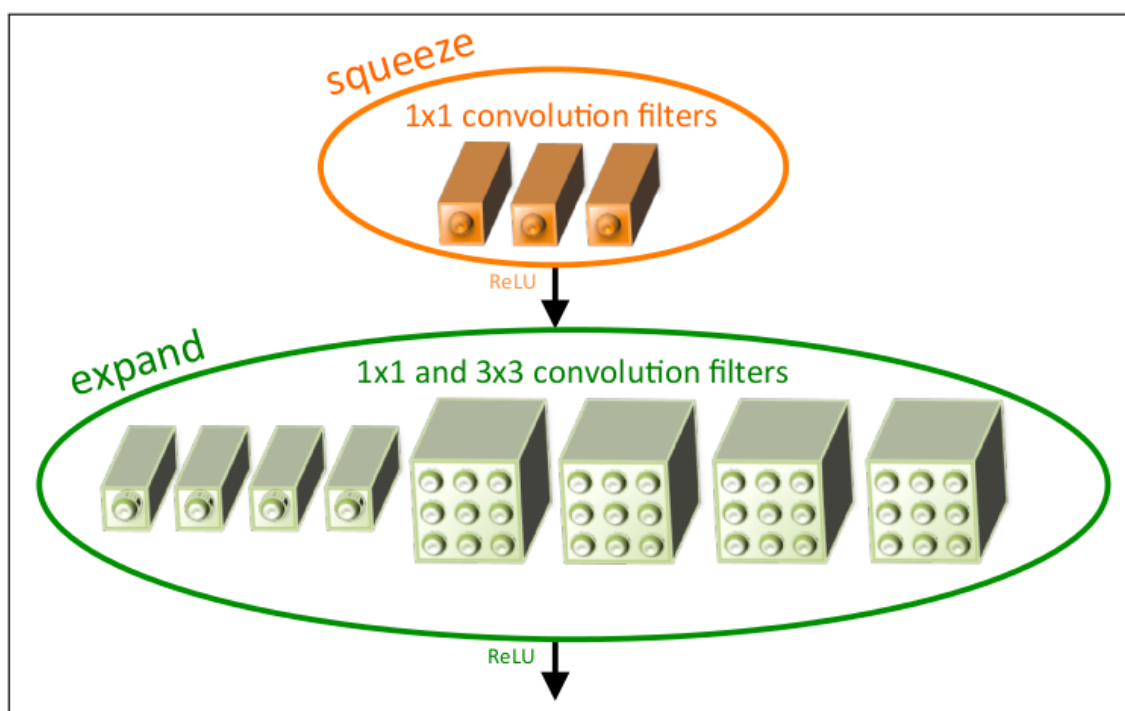
Το πρώτο κομμάτι είναι για την εύρεση του περιβλήματος των αντικειμένων. Το σημείο $(\delta x_{ijk}, \delta y_{ijk}, \delta w_{ijk}, \delta h_{ijk})$ αφορά τις σχετικές συντεταγμένες του πιθανού περιβλήματος ως προς το σημείο (i, j) . Το δεύτερο κομμάτι (η γραμμή με το δεύτερο τριπλό άθροισμα) αφορά 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						

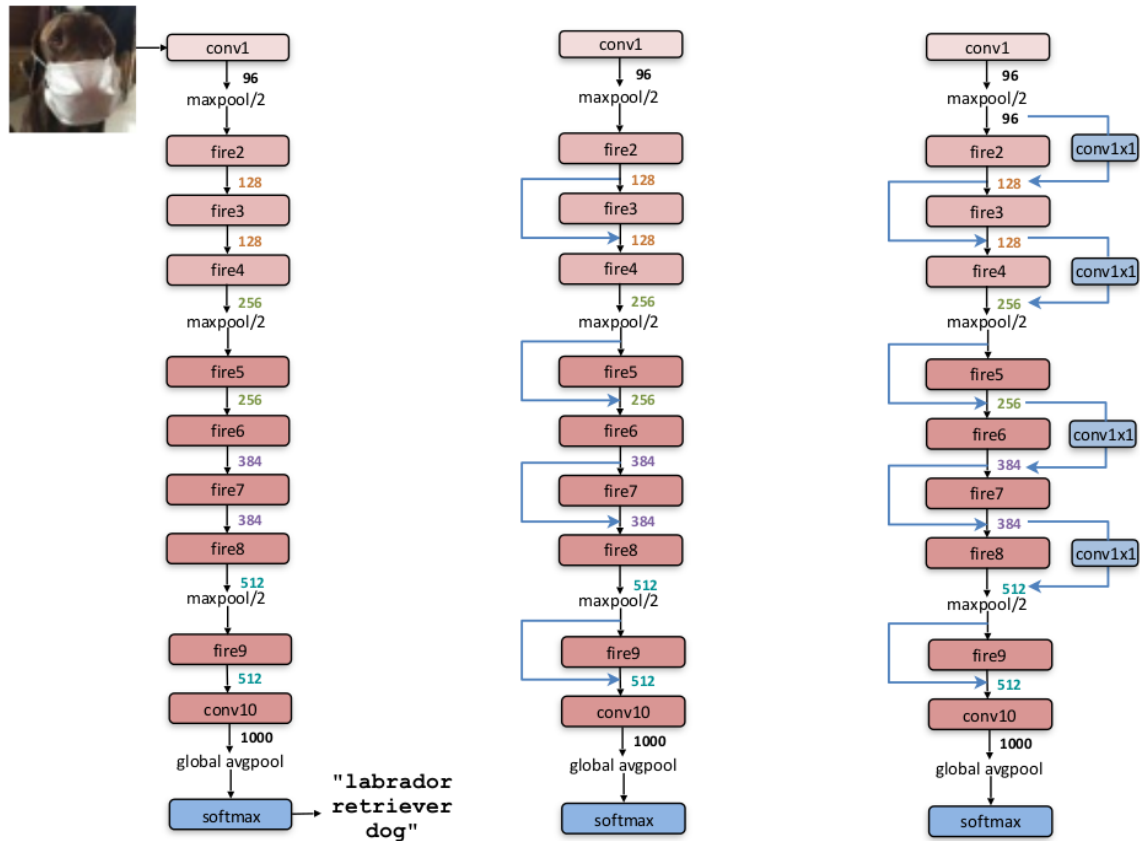
Η απαίτηση μνήμης παρά τον παραπάνω πίνακα μπορεί να μειωθεί περαιτέρω. Με τη χρήση τεχνικών συμπίεσης και επηρεάζοντας τις υπερπαραμέτρους επιτυγχάνεται παραπάνω από 95x

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 έκδοσή του (SqueezeDet+) τα 32.1 FPS. Ο μειωμένος αριθμός παραμέτρων οδηγεί σε λιγότερη προσέλαση 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 [41] 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 1.2: 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 1.3: 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 προηγούμενων επιπέδων με το επόμενο και όχι μόνο του αμέσως προηγούμενου ωφελεί την ακρίβεια του δικτύου.

1.3 YOLO[48]

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/αναγνώριση 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-

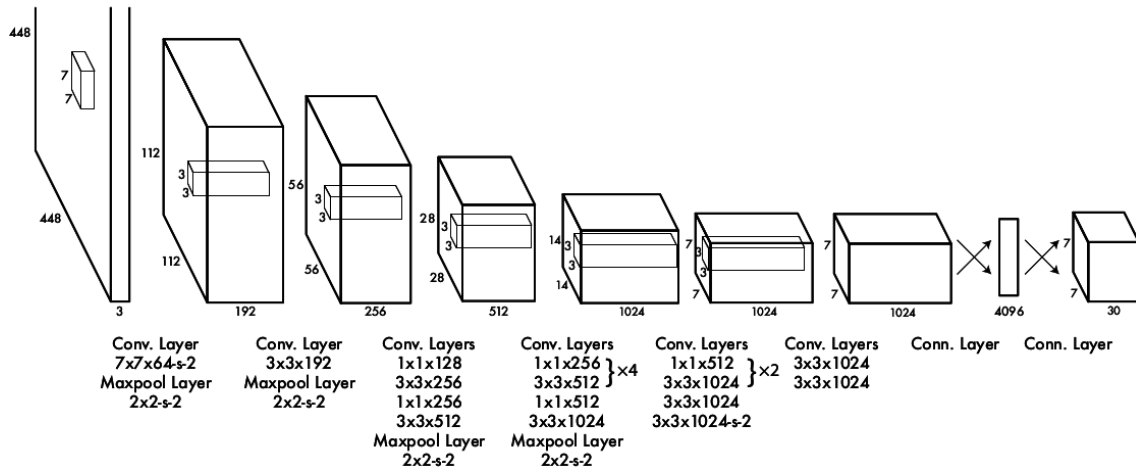
βλήματα του δικτύου YOLO είναι ορθογώνια. Η πληροφορία που περιγράφει ένα περίβλημα είναι $(x, y, w, h, \text{σκορ εμπιστοσύνης})$

(x, y) : το κέντρο του περιβλήματος

w = πλάτος ορθογωνίου / πλάτος εικόνας

h = ύψος ορθογωνίου / ύψος εικόνας

Επιπλέον κάθε κελί έχει και C πιθανότητες $PrClass_i|Object$, $i = 1, \dots, C$. 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 1.4: 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:

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

Η εκπαίδευση γίνεται αρχικά με *learning rate* 10^{-2} έπειτα 10^{-3} και τελικά 10^{-4} . Επιπλέον χρησιμοποιείται η τεχνική *dropout* [54] 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 [47], φαίνεται στο [1] ότι όσο λιγότερη μνήμη υπάρχει διαθέσιμη τόσο πιο αργά εκτελείται ο αλγόριθμος. Από το Σχήμα 1.4 υπολογίζεται πως για τον ταχυστή 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.

1.4 YOLO9000 [49]

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 1.1: 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 3×3 συνελεκτικά επίπεδα με 1024 φίλτρα το καθένα ακολουθούμενο από ένα τελευταίο 1×1 συνελεκτικό επίπεδο.

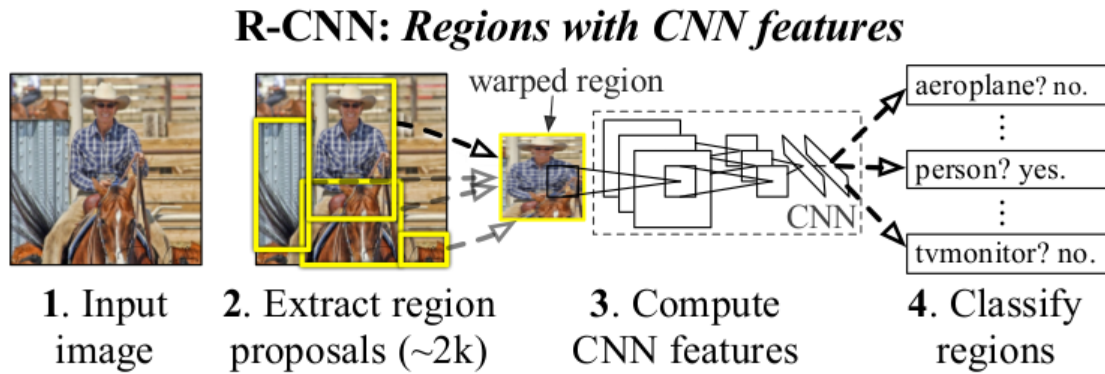
multi-scale: Το μέγεθος της εικόνας αυξάνεται ή ελαττώνεται κατά την εκπαίδευση.

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[55]:

- 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 retriever" 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 1.5: 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.

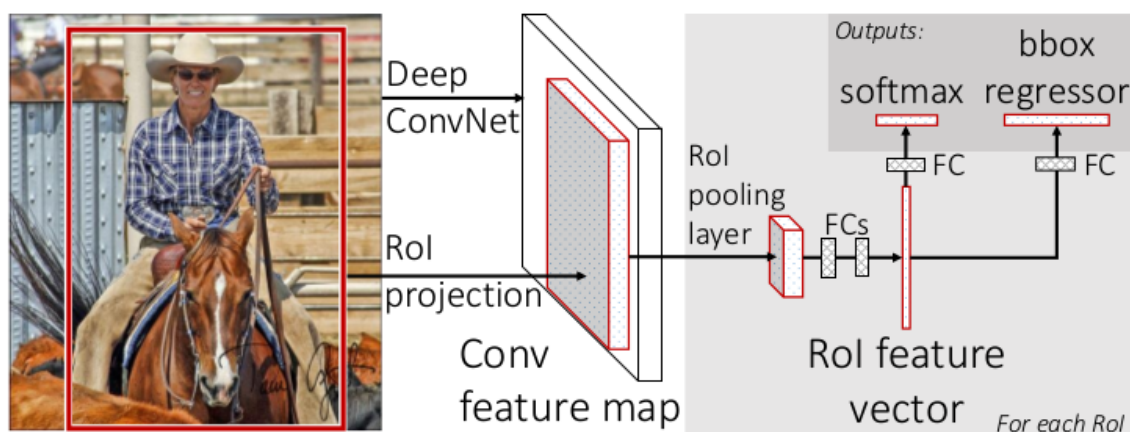
1.5 R-CNN [21]

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* [57]. Με αυτή μπορεί και εντοπίζει κάθε αντικείμενο στην εικόνα και το περιβάλλει 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 τετράγωνα 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 1.6: 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.

1.6 Fast R-CNN [20]

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 παλινδρόμησης 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.

1.7 Faster R-CNN [50]

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'[8] η οποία χρησιμοποιείται πλέον σε αρκετά νευρωνικά δίκτυα πέραν των συνελεκτικών για αύξηση της ακρίβειας.

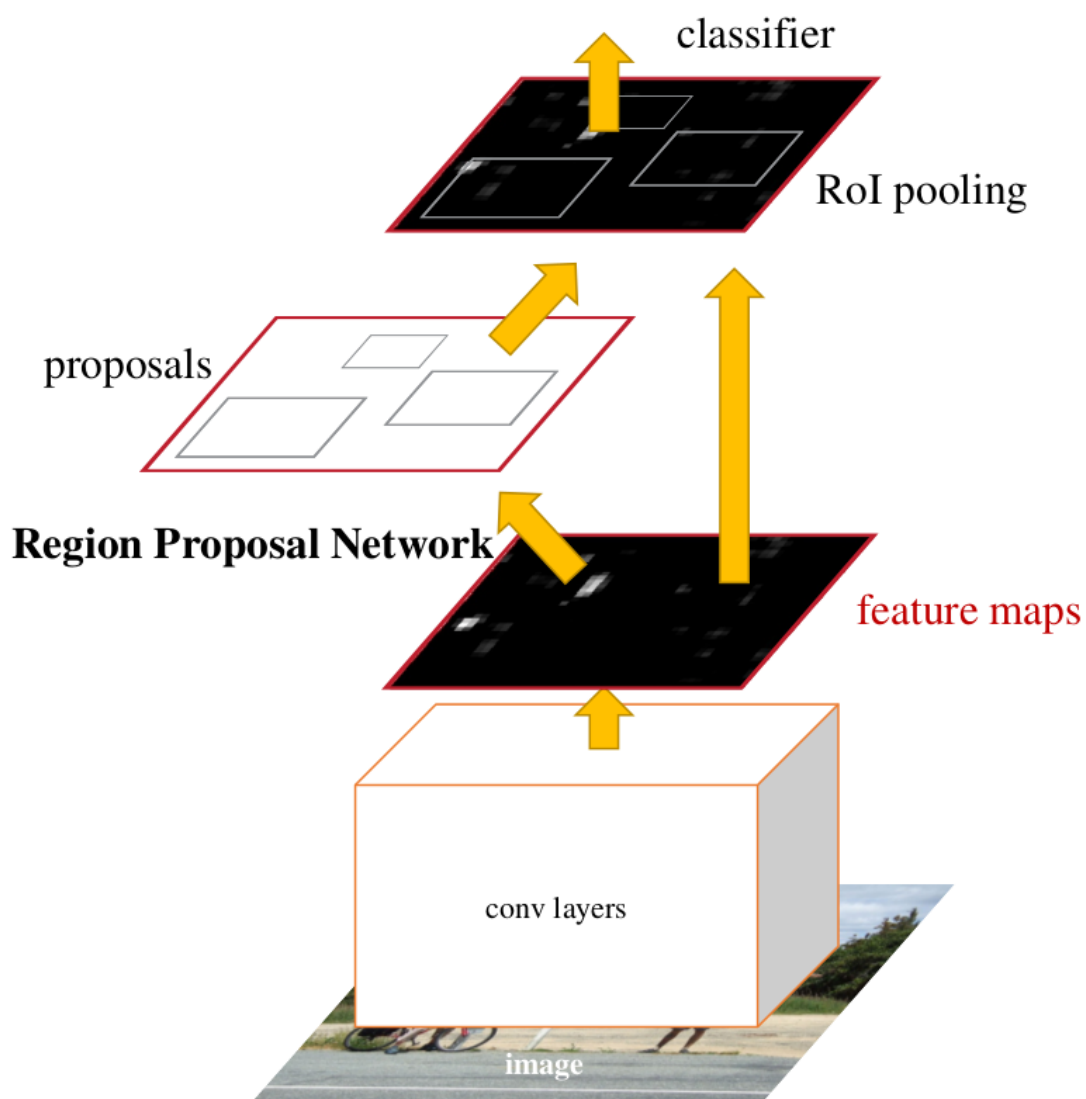
Αξιοσημείωτο είναι ότι το RPN χρησιμοποιείται από το SqueezeDet, αλλά αντί ολικά συνδεδεμένου δικτύου στο πρώτο χρησιμοποιεί συνελεκτικό επίπεδο. Και στις δύο περιπτώσεις υπάρχει ανεξαρτησία στη μετακίνηση των αντικειμένων.

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 έως 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.

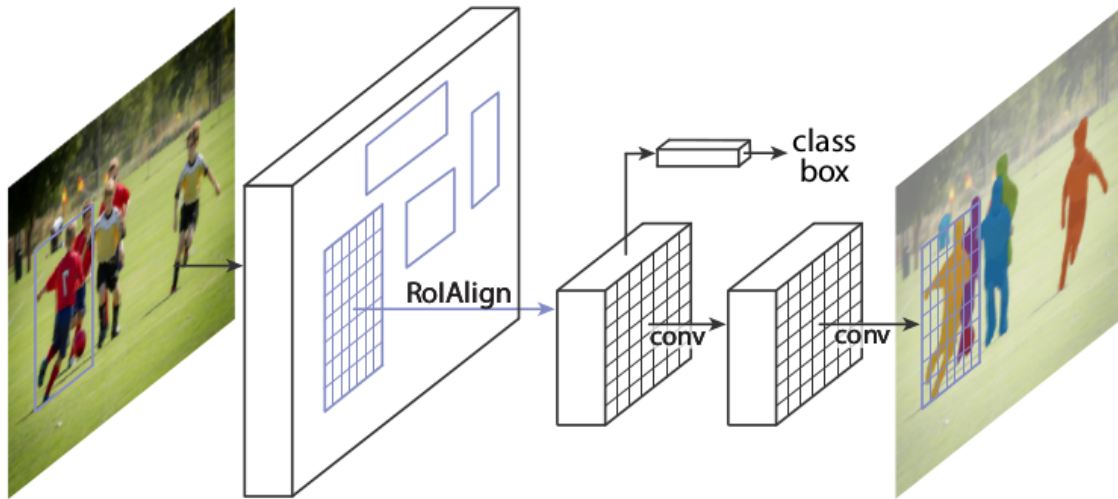
1.8 Mask R-CNN [25]

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 1.7: 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 1.8: 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.

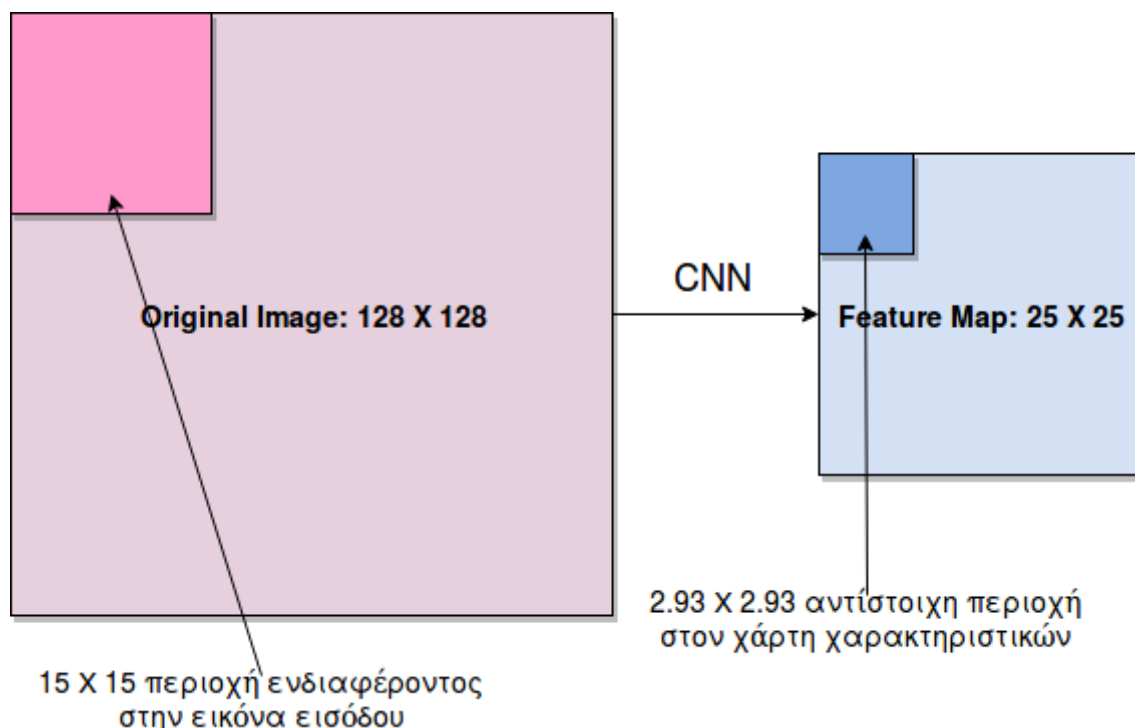
λέγεται RoIAlign. Πρότερα, το εικονοστοιχείο που αντιστοιχούσε στη θέση $x = 2.93$ των χαρακτηριστικών απλά θα αντιστοιχιζόταν στη θέση $x = 2$.

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[9]). 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).

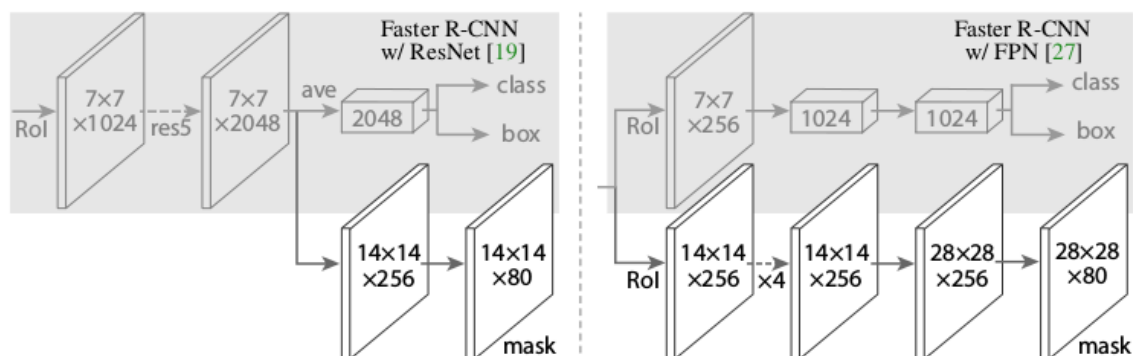
Η εκπαίδευση γίνεται με τον ίδιο τρόπο όπως στο Fast R-CNN, χρησιμοποιώντας gradient descent με ορμή. Ωστόσο, τα RPN και το δίκτυο της μάσκας εκπαιδεύονται ξεχωριστά, θεωρώντας το RoI θετικό όταν η μετρική IoU είναι πάνω από 0.5. Επιπλέον τελικά χαρακτηριστικά που διερευνήθηκαν για το δίκτυο αυτό είναι:

- 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.

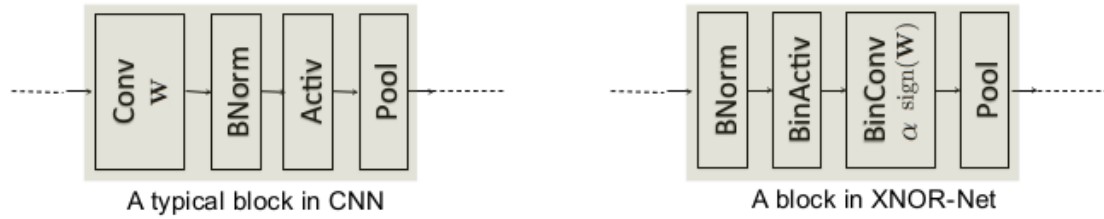
Βέβαια λόγω των υπολογιστικών απαιτήσεων το δίκτυο αυτό δε θεωρείται κατάλληλο για ενσωματωμένα συστήματα. Παρ' όλα αυτά η δυνατότητα εντοπισμού αντικειμένων ανά εικονοστοιχείο το χρήζει πιο εύχρηστο για άλλους αλγορίθμους που χρησιμοποιούνται σε αυτά, όπως το SLAM με vision, η αυτόνομη οδήγηση κ.α.



Σχήμα 1.9: Παράδειγμα αντιστοίχισης περιοχής από τον χάρτη χαρακτηριστικών στην αρχική εικόνα εισόδου. Για τον υπολογισμό των χαρακτηριστικών στο σημείο (2.93, 2.93) χρησιμοποιείται διγραμμική παρεμβολή.



Σχήμα 1.10: Η κεφαλή του δικτύου: Για την κεφαλή του δικτύου χρησιμοποιούνται κομμάτια από άλλα δίκτυα όπως το ResNet[26] και το FPN[37]. Τα βελόκια αντιστοιχούν σε συνέλιξη, αποσυνέλιξη ή πέρασμα από ένα ολικά συνδεδεμένο επίπεδο. Όλες οι συνέλιξεις 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 συνέλιξεις πάνω στο Faster R-CNN με την κεφαλή του FPN.



Σχήμα 1.11: Σύγκριση μεταξύ των μπλοκ του XNOR-Network (δεξιά) και ενός τυπικού CNN (αριστερά).

1.9 XNOR-Net [46]



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 $\{-1, +1\}$.

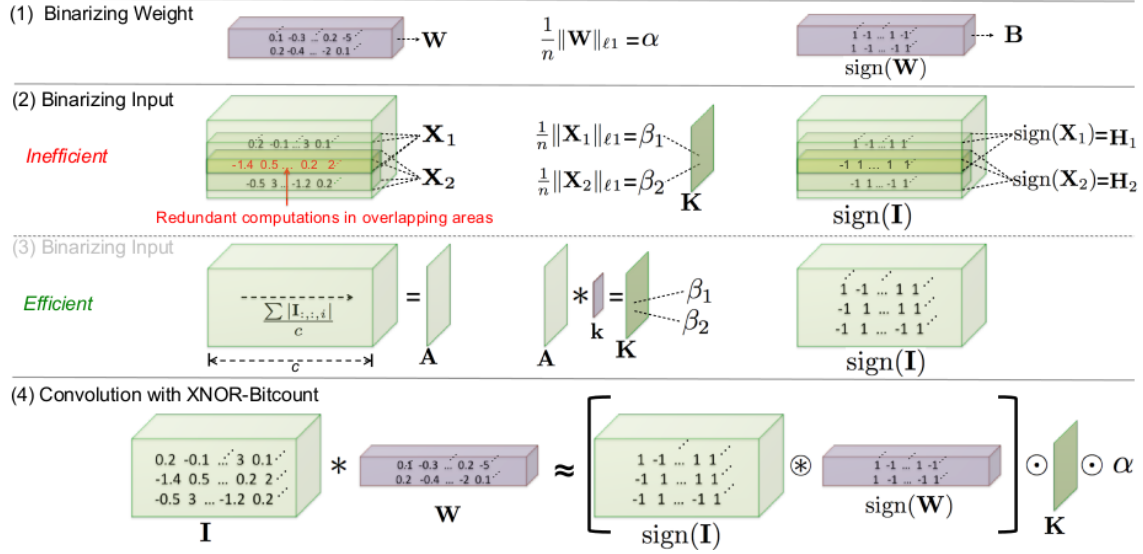
Η δεύτερη από τις δύο εκδοχές είναι και η πιο αποτελεσματική, τόσο από άποψη χρόνου εκτέλεσης, όσο και από άποψη ακρίβειας. Και στις δύο περιπτώσεις η αρχιτεκτονική του δικτύου είναι ίδια με κάποιο άλλο π.χ. AlexNet[35], GoogleNet[55], ResNet[26]. Αυτό που αλλάζει είναι τα βάρη και η σειρά με την οποία 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:

$$I * W \approx (\text{sign}(I) \otimes \text{sign}(W)) \odot K\alpha$$

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

Όσον αφορά την εκπαίδευση το δίκτυο χρησιμοποιεί τη τεχνική ADAM [33] 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 32x φορές πιο αυξημένη. Μεγαλύτερη επιτάχυνση δικτύων επιτυγχάνεται όσο τα φίλτρα στα επίπεδα



Σχήμα 1.12: Επεξήγηση της διαδικασίας μετασχηματισμού της συνέλιξης σε συνέλιξη που κάνει χρήση της πράξης XNOR και δυαδικούς τανυστές.

του νευρωνικού είναι μεγαλύτερα. Αυτό βέβαια βρίσκεται σε αντίθεση με τη πιο μοντέρνα τακτική να ελαττώνονται οι διαστάσεις των δικτύων όπως στο [30]. 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.

1.10 SSD [38]

Το δίκτυο αυτό παρουσιάζει κοινή αρχιτεκτονική με το Faster-RCNN και ουσιαστικά αποτελεί μια βελτίωση του (Deep Multibox [18]). 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 ταιριάζει 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.

1.11 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 μετρίεται 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

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 1.2: 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 μετρείται στο σύνολο δεδομένων του ImageNet. Τα χαρακτηριστικά των δικτύων που δεν αναλύονται στο κεφάλαιο είναι από [28], ενώ άλλα προέρχονται από [11], [32],[27],[3]. 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 1.3: Σύγκριση δικτύων για εντοπισμό αντικειμένων. Η ακρίβεια των δικτύων μετρείται 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.

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 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 [63].

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 [28]). Ο Πίνακας 1.2 δείχνει τα χαρακτηριστικά κάθε νευρωνικού τέτοιου τύπου. Οι χρόνοι αφορούν την εκτέλεση των δικτύων αυτών με τη χρήση του tensorflow [2] στη GPU GTX 1080 Ti της NVIDIA, και στο σύνολο δεδομένων του Imagenet ILSVRC2012. 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 [43].

Κεφάλαιο 2

Μετάδοση Γνώσης

2.1 Εισαγωγή

Ο όρος της μετάδοσης γνώσης (transfer learning) χρησιμοποιείται σε όλο το φάσμα της τεχνητής νοημοσύνης ως η μεταφορά παραμέτρων ενός ήδη εκπαιδευμένου συστήματος σε ένα άλλο. Στο παρόν κεφάλαιο θα αναλυθεί η μετάδοση γνώσης σε αλγόριθμους εντοπισμού αντικειμένων, βέβαια για την καλύτερη ανάλυση αρχικά θα πρέπει να δοθούν κάποιοι ορισμοί και ερμηνευτεί με ακρίβεια ο όρος. Έπειτα, θα αναλυθούν οι τρόποι μετάδοσης γνώσης. Στο τέλος παρουσιάζεται το πείραμα αυτής της εργασίας για τα δίκτυα εντοπισμού αντικειμένων.

2.2 Πρόβλημα

Το πρόβλημα που αντιμετωπίζεται στα νευρωνικά δίκτυα έγκειται στο θεώρημα *no free lunch* [60]. Η θεωρία της μηχανικής μάθησης υποστηρίζει ότι οι αλγόριθμοι μηχανικής μάθησης μπορούν να γενικευτούν σε ένα πεπερασμένο σύνολο από παραδείγματα. Ωστόσο, αυτό δείχνει να αντιτίθεται σε κάποιες βασικές αρχές της λογικής. Η επαγωγή γενικών κανόνων από κάποιο πεπερασμένο σύνολο παραδειγμάτων δεν αποτελεί λογικό εγχείρημα. Για παράδειγμα, προκειμένου να δημιουργηθεί ένας κανόνας ο οποίος να περιγράφει κάθε μέλος ενός συνόλου, θα πρέπει εκ των προτέρων να υπάρχει πληροφορία για κάθε μέλος αυτού του συνόλου.

Εν μέρει η μηχανική μάθηση προσπερνά αυτό το πρόβλημα προσφέροντας μόνο πιθανοτικούς κανόνες, αντί για ακριβείς κανόνες που αποτελούν μέλη της μαθηματικής λογικής. Η μηχανική μάθηση υπόσχεται να βρει κανόνες οι οποίοι είναι πιθανοτικά έγκυροι για τα περισσότερα μέλη κάποιου συνόλου.

Αυστυχώς, ακόμη και αυτό δεν μπορεί να επιλύσει το όλο πρόβλημα. Σύμφωνα με το θεώρημα *no free lunch*, ο μέσος όρος σφάλματος σε όλες τις πιθανές κατανομές, κάθε αλγόριθμο κατηγοριοποίησης είναι ο ίδιος με την κατηγοριοποίηση σημείων τα οποία δεν έχουν παρατηρηθεί εκ των προτέρων.

Αν \mathbb{K} είναι το σύνολο όλων των πιθανών κατανομών και $\kappa_i, \kappa_j \in \mathbb{K}$ δύο κατανομές στις οποίες έχουν εκπαιδευτεί οι αλγόριθμοι a_i, a_j αντίστοιχα. Τότε για συνάρτηση σφάλματος $e(a, \kappa)$ αλγόριθμου a σε κάποια κατανομή $\kappa \in \mathbb{K}$. Έχουμε

$$E_{\kappa \in \mathbb{K} - \kappa_i} [e(a_i, \kappa)] = E_{\kappa \in \mathbb{K} - \kappa_j} [e(a_j, \kappa)], \forall i, j$$

Δηλαδή, κανένας αλγόριθμος δεν είναι καλύτερος στην επίλυση καινούριων προβλημάτων από ότι ένας άλλος. Ευτυχώς, αυτό συμβαίνει μόνο όταν παίρνουμε υπόψη όλες τις πιθανές κατανομές. Στον πραγματικό κόσμο μπορούν να γίνουν κάποιες υποθέσεις για τους τύπους

κατανομών που αντιμετωπίζονται και μπορούμε να πούμε ότι αυτές ανήκουν σε συγκεκριμένες πολλαπλότητες. Παρόλα αυτά, απαιτείται το δίκτυο εντοπισμού να είναι πραγματικού χρόνου και περιορισμένης μνήμης. Έτσι, δεν μπορεί να συμπεριλάβει χαρακτηριστικά για να αφομοιώσει όλη την εμπειρία που απαιτείται για να εντοπίζει εκ των προτέρων κάθε αντικείμενο.

2.3 Μία ιδέα για τη μετάδοση γνώσης [53]

Η μετάδοση γνώσης είναι μία ανθρώπινη διαδικασία η οποία έχει μελετηθεί πολύ στην φιλοσοφία, στην εκπαίδευση και στην εκπαιδευτική ψυχολογία [44]. Στην εκπαίδευση, η μετάδοση γνώσης αναφέρεται ως: "Προηγούμενες μαθημένες γνώσεις ή ικανότητες οι οποίες επηρεάζουν τον τρόπο με τον οποίο μία καινούρια γνώση ή ικανότητα μαθαίνεται και χειρίζεται. Η μετάδοση θεωρείται ως θετική αν η απόκτηση και η επίδοση στην χρήση της καινούριας γνώσης διευκολύνθηκε και ως αρνητική αν εμποδίστηκε." (μετάφραση από: **Leberman et al. 2006, McKeough et al. 1995, Cormier & Hagman 1987**)

Όταν η τεχνική μετάδοσης γνώσης εφαρμόζεται στη μηχανική μάθηση έχει παρόμοιο σκοπό. Το κίνητρο της μηχανικής μάθησης είναι να βελτιώσει τη μάθηση σε ένα συγκεκριμένο πεδίο αποκτώντας πληροφορία από ένα παρόμοιο, αλλά διαφορετικό πεδίο. Οι παραδοσιακές τεχνικές μηχανικής μάθησης όπως λέχθηκε λειτουργούν υπό υποθέσεις για τις κατανομές τους. Για αυτό και παλαιότερα προτεινόταν η μάθηση να αρχίζει *tabula rasa*, χωρίς να ξέρει τίποτε πιο πριν ο αλγόριθμος και όλοι οι παράμετροι των δικτύων ή των αλγορίθμων να αρχίζουν από κάποια τυχαία κατανομή όπως η γκαουσιανή. Επιπλέον, υπάρχει η ανάγκη από την πλειοψηφία των τεχνικών μηχανικής μάθησης τα δεδομένα στα σύνολα της εκπαίδευσης και του ελέγχου να είναι από τον ίδιο χώρο χαρακτηριστικών (feature space). Η τεχνική μετάδοσης γνώσης δείχνει πως αυτό δεν είναι σε κάθε περίπτωση απαραίτητο για τα νευρωνικά δίκτυα, γιατί δίνεται η δυνατότητα να συνδυαστεί προηγούμενη γνώση από μία προηγούμενη εκπαίδευση σε μία καινούρια αλλά σχετική με την προηγούμενη.

Χρησιμοποιώντας αυτή την παρατήρηση μπορεί να παρακάμψει κανείς το *no free lunch theorem*, διότι δεν χρειάζεται ανά πάσα στιγμή να είναι δυνατός ο εντοπισμός όλων των αντικειμένων. Ο αλγόριθμος μπορεί να ξέρει κάθε στιγμή να εντοπίζει ένα μικρό σύνολο αντικειμένων και απλά κάθε φορά να μαθαίνει τον εντοπισμό ενός καινούριου συνόλου.

2.4 Ορισμός της Μετάδοσης Γνώσης [42, 23]

Πριν μεταβούμε στον μαθηματικό ορισμό της τεχνικής μετάδοσης γνώσης θα πρέπει να ορίσουμε κάποια άλλα μεγέθη πρώτα. Οι ορισμοί αυτοί μπορούν να χρησιμοποιηθούν γενικότερα στο αντικείμενο της μηχανικής μάθησης, ωστόσο συνίσταται η χρήση τους μόνο στο αντικείμενο της μετάδοσης γνώσης.

■ Το πεδίο ορισμού (*Domain*) ορίζεται ως μία δυάδα $D = (\mathcal{X}, P)$. Το πρώτο μέλος της δυάδας \mathcal{X} είναι ο χώρος (πολλαπλότητα) στον οποίο ανήκουν τα δείγματα-παρατηρήσεις x κάποιου προβλήματος (π.χ. ο χώρος των **διοδιάστατων** **εικόνων** για εικόνες μεγέθους 1024×768). Το δεύτερο μέλος της δυάδας είναι η αθροιστική συνάρτηση κατανομής $P(X)$ όπου $X = x_1, \dots, x_n \in \mathcal{X}$.

Η εκπαίδευση, η οποία ορίζεται και παρακάτω, γίνεται πάντα σε ένα πεπερασμένο κομμάτι του πεδίου ορισμού, διότι εκτελείται από κάποιον αλγόριθμο και διαρκεί πεπερασμένο χρόνο.

■ Δοθέντος ενός πεδίου ορισμού, το υποψήφιο έργο (task) αποτελείται από την δυάδα $(\mathcal{Y}, f(\cdot))$. Το πρώτο μέλος της δυάδας είναι ο χώρος ετικετών \mathcal{Y} (π.χ. τα ονόματα των κλάσεων των αντικειμένων σε μία εικόνα και ορθογώνια πάνω στην εικόνα όπου αυτά παρατηρούνται). Το

δεύτερο είναι μία συνάρτηση ετικετοποίησης $f : \mathcal{X} \rightarrow Y$. Το έργο συμβολίζεται με το γράμμα T .

Η συνάρτηση ετικετοποίησης "προβλέπει" πάντα ορθά την ετικέτα δοθείσας μίας εισόδου. Από πιθανοτική σκοπιά η $f(x)$ μπορεί να γραφεί ως $P(y|x)$.

■ Το μοντέλο M ορίζεται ως μία συνάρτηση $M(x; W)$, $M : \mathcal{X} \times \mathbb{W} \rightarrow Y$. Το x είναι η είσοδος του μοντέλου, ενώ το w οι παράμετροί του. Το μοντέλο είναι δυνατόν μέσα από την εκπαίδευση (ή ρύθμιση των παραμέτρων του) να προσεγγίσει μία συνάρτηση f στο κοινό τους σύνολο ορισμού.

Η λέξη αλγόριθμος με τη λέξη μοντέλο στο παρακάτω κείμενο έχουν την ίδια σημασία, εκτός αν αναφέρεται διαφορετικά.

■ Το πεδίο ορισμού δεδομένων ή γνωστό ως σύνολο δεδομένων ορίζεται ως $D_d = \{(x_1, y_1), \dots, (x_n, y_n)\}$ όπου $y_i = f(x_i) \in \mathcal{Y}$ και $x_i \in \mathcal{X}$.

Το αρχικό πεδίο ορισμού συμβολίζεται ως D_S και αφορά το πεδίο ορισμού πάνω στο οποίο ήδη έχει εκπαιδευτεί το μοντέλο. Το τελικό πεδίο ορισμού (στόχου) συμβολίζεται ως D_t και αφορά το πεδίο ορισμού στο οποίο πρόκειται να εκπαιδευτεί το μοντέλο. Αντίστοιχα ορίζονται το αρχικό και το τελικό πεδίο ορισμού δεδομένων.

■ Ως πρόβλημα σε ένα πεδίο ορισμού D με ένα έργο (T) ορίζεται η προσέγγιση της συνάρτησης f του T στο D .

Το σύνολο όλων των υποψήφιων έργων θα συμβολίζεται στη συνέχεια ως \mathbb{T} .

■ Το μέτρο επίδοσης $\text{Prf}(T, \hat{f}, \mathcal{X}_{\text{test}})$, όπου $\mathcal{X}_{\text{test}} \subseteq \mathcal{X}$ αναφέρεται στο πόσο πετυχημένα η \hat{f} μπορεί να προσεγγίσει τη συνάρτηση ετικετοποίησης f του έργου T στο σύνολο $\mathcal{X}_{\text{test}}$. Όσο πιο υψηλό, τόσο καλύτερη θεωρείται η προσέγγιση.

Ένα τέτοιο μέτρο θα μπορούσε να είναι η ακρίβεια στο σύνολο ελέγχου του τελικού πεδίου ορισμού.

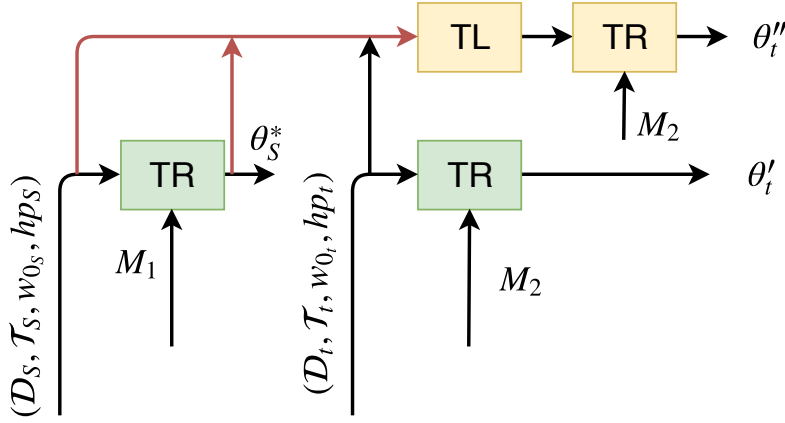
■ Η εκπαίδευση ορίζεται ως η διαδικασία $\text{Tr}(D, T, M, \text{Prf}; \vec{h}p, w_0)$ η οποία έχει ως είσοδο ένα πεδίο ορισμού D , ένα υποψήφιο έργο T και ένα μοντέλο M του τύπου της συνάρτησης f του με αρχική τιμή παραμέτρων την w_0 , ως έξοδο τις βελτιστοποιημένες παραμέτρους w_{opt} του μοντέλου M έτσι ώστε $\text{Prf}(T, M(\mathcal{X}_{\text{test}}, w_{\text{opt}})) > \lambda$ για επιθυμητά μεγάλο $\lambda \geq 0$ και $\mathcal{X}_{\text{test}} \subseteq \mathcal{X}$. Το διάνυσμα $\vec{h}p$ είναι το οι υπερπαραμέτροι που χρησιμοποιήθηκαν για το στιγμιότυπο της εκπαίδευσης.

Οι υπερπαραμέτροι θα μπορούσαν να μην είναι όλες τοποθετημένες σε ένα διάνυσμα, αλλά αυτό δεν είναι διαφορετικό από ότι αν τις περιγράφαμε υπό μία άλλη μαθηματική δομή. Το λ το οποίο αναφέρεται παραπάνω μπορεί να είναι μέλος των υπερπαραμέτρων.

■ Δοθέντος ενός αρχικού πεδίου ορισμού D_S με ένα αρχικό έργο T_S και ενός τελικού πεδίου ορισμού D_t με ένα τελικό έργο T_t η μετάδοση γνώσης ορίζεται ως η διαδικασία TL η οποία έχει ως στόχο τη βελτίωση της εκπαίδευσης στο τελικό πεδίο ορισμού και έργο, δηλαδή έχει ως στόχο τη βελτίωση της προσέγγισης της f_t στο D_t χρησιμοποιώντας πληροφορίες από τα D_S, S , όπου εν γένει $D_S \neq D_t, T_S \neq T_t$.

Στον παραπάνω ορισμό η συνθήκη $D_S \neq D_t$, σημαίνει ότι είτε $\mathcal{X}_S \neq \mathcal{X}_t$ ή ότι $P_S(X) \neq P_t(X)$. Όπως και η συνθήκη $T_S \neq T_t$ σημαίνει ότι είτε $\mathcal{Y}_S \neq \mathcal{Y}_t$, είτε ότι $P(Y_S|X_S) \neq P(Y_t|X_t)$. Αν και οι δύο συνθήκες δεν ισχύουν, τότε δεν υπάρχει μετάδοση γνώσης. Επιπλέον όταν ο χώρος των χαρακτηριστικών στο αρχικό πεδίο ορισμού και στο τελικό σχετίζονται με κάποιο τρόπο, χαρακτηρίζονται ως σχετικά.

Στην παρούσα εργασία $M_s = M_t = M$, δηλαδή χρησιμοποιείται το ίδιο μοντέλο στην αρχική και στην τελική εκπαίδευση, απλά οι παράμετροί τους δεν είναι ίδιες.



Σχήμα 2.1: Η διαδικασία μετάδοσης γνώσης (TL) σε σχέση με την απλή εκπαίδευση (TR) και η σύγκριση της πορείας πληροφορίας. Η μεταβλητή θ_s^* είναι οι παράμετροι που επιστρέφει η εκπαίδευση στο αρχικό πεδίο ορισμού. Η μεταβλητή θ_t' είναι οι παράμετροι που επιστρέφει η εκπαίδευση στο τελικό πεδίο ορισμού χωρίς μετάδοση γνώσης. Η μεταβλητή θ_t'' είναι οι παράμετροι που επιστρέφει η εκπαίδευση στο τελικό πεδίο ορισμού με χρήση μετάδοσης γνώσης.

2.5 Σχέση μετάδοσης γνώσης και μάθησης πολλαπλών έργων

Η έννοια της μετάδοσης γνώσης είναι πολύ κοντινή με τη μάθηση πολλαπλών έργων (MTL) ταυτοχρόνως. Η MTL χρησιμοποιεί παραλληλισμό των έργων στο ίδιο πεδίο ορισμού. Με αυτό τον τρόπο μοιράζονται χαρακτηριστικά που εξάγονται στο ένα έργο και απαιτούνται στα υπόλοιπα. Στο τέλος αυτής της εκπαίδευσης μπορεί το μοντέλο να έχει καλύτερη επίδοση ακόμα και μόνο σε ένα έργο από ότι αν γινόταν εκπαίδευση μόνο σε αυτό ξεχωριστά. Οι λόγοι για τους οποίους συμβαίνει αυτό δε θα αναλυθούν, ωστόσο για παραπάνω διερεύνηση ο αναγνώστης παραπέμπεται στο [52].

Η διαφορά με τη μετάδοση γνώσης είναι ότι η μετάδοση γνώσης λύνει το πρόβλημα σειριακά, μαθαίνοντας πρώτα τα χαρακτηριστικά τα οποία επιλύουν το πρώτο έργο και μετά μεταβαίνει στο επόμενο. Αν όλα τα έργα αφορούν το ίδιο πεδίο ορισμού, τότε η MTL είναι η προτεινόμενη λύση. Ωστόσο, δεν μπορεί να εφαρμοστεί όταν τα έργα αφορούν και διαφορετικά πεδία ορισμού. Αντίθετα, η μετάδοση γνώσης μπορεί να χρησιμοποιήσει χαρακτηριστικά από ένα έργο σε ένα πεδίο ορισμού και έργο D_s, T_s αντίστοιχα, προκειμένου να έχει καλύτερη επίδοση σε ένα πεδίο ορισμού και έργο D_t, T_t .

2.6 Διάφορα σενάρια μετάδοσης γνώσης

Παρακάτω δίνεται μία περίληψη των διαφόρων σεναρίων μετάδοσης γνώσης όπως αναφέρονται στα [53] [42], αλλά με πιο σύγχρονα παραδείγματα.

2.6.1 Επαγωγική μετάδοση γνώσης

Δοθέντος ενός αρχικού πεδίου ορισμού D_s με ένα αρχικό έργο T_s και ένα τελικό πεδίο ορισμού D_t με ένα τελικό έργο T_t , η επαγωγική μετάδοση γνώσης αποσκοπεί στη βελτίωση της μάθησης της τελικής συνάρτησης ετικετοποίησης f_t στο τελικό πεδίο ορισμού D_t χρησιμοποιώντας γνώση από τα D_s, T_s , όπου $T_s \neq T_t$.

Στον παραπάνω ορισμό τα D_S και D_t μπορούν να είναι σχετικά. Βασιζόμενοι στο παραπάνω ορισμό, απαιτούνται μόνο λίγα ετικετοποιημένα δεδομένα για την πρόβλεψη της τελικής συνάρτησης ετικετοποίησης. Γενικά, πρόκειται για πληροφοριοδοτημένη άνευ επίβλεψης μετάδοση γνώσης, όπου εκεί βρίσκεται αρκετό ερευνητικό και πρακτικό ενδιαφέρον. Ωστόσο, υπάρχει και η περίπτωση να είναι Μη-Πληροφοριοδοτημένη άνευ επίβλεψης μετάδοση γνώσης.

Μεταφορά γνώσης από στιγμιότυπα

Η μεταφορά στιγμιότυπων για την επαγωγική μετάδοση γνώσης είναι αρκετά διαισθητική. Αν και η πλειονότητα των δεδομένων του αρχικού πεδίου ορισμού δεν μπορεί να χρησιμοποιηθεί απευθείας, υπάρχουν μέσα σε αυτά μέλη τους τα οποία μπορούν να χρησιμοποιηθούν μαζί με μερικά ετικετοποιημένα δεδομένα του τελικού πεδίου ορισμού.

Για παράδειγμα έχει προταθεί ο αλγόριθμος TrAdaBoost, ο οποίος είναι μία επέκταση του αλγορίθμου AdaBoost, προκειμένου να λύσει τα προβλήματα επαγωγικής μεταφοράς γνώσης. Ο TrAdaBoost υποθέτει ότι το αρχικό και το τελικό πεδίο ορισμού χρησιμοποιούν τις ίδιες ετικέτες και χαρακτηριστικά, αλλά οι κατανομές των δεδομένων στα δύο πεδία ορισμού είναι διαφορετικές. Επιπλέον, ο TrAdaBoost υποθέτει ότι λόγω της διαφοράς των κατανομών, μερικά από τα δεδομένα του αρχικού πεδίου ορισμού μπορεί να βοηθήσουν στην εκπαίδευση στο τελικό πεδίο ορισμού, αλλά μπορεί και να δυσχεραίνουν την εκπαίδευση.

Ο αλγόριθμος προσπαθεί να ζυγιάσει τα δεδομένα προκειμένου να μειώσει τα δεδομένα με αρνητική επίδραση στην εκπαίδευση και να ενισχύσει τα δεδομένα με θετική επίδραση στην εκπαίδευση στο τελικό πεδίο ορισμού. Σε κάθε επανάληψη του, ο TrAdaBoost εκπαιδεύει το βασικό ταξινομητή (συνάρτηση πρόβλεψης - μοντέλο) σε αυτό το ζυγιασμένο σύνολο δεδομένων. Ωστόσο, το σφάλμα υπολογίζεται μόνο στα δεδομένα του τελικού πεδίου ορισμού. Περαιτέρω, ο TrAdaBoost ακολουθεί την ίδια στρατηγική με τον AdaBoost για την ανανέωση των λανθασμένα ταξινομημένων δεδομένων στο τελικό πεδίο ορισμού. Ωστόσο, χρησιμοποιείται διαφορετική στρατηγική για την ανανέωση των λανθασμένα ταξινομημένων δεδομένων στο αρχικό πεδίο ορισμού από το αρχικό μοντέλο. Στο [12] δίνεται μία θεωρητική ανάλυση του αλγορίθμου TrAdaBoost.

Μετάδοση γνώσης της αναπαράστασης χαρακτηριστικών

Η προσέγγιση με τη μετάδοση της αναπαράστασης χαρακτηριστικών στο πρόβλημα της επαγωγικής μετάδοσης γνώσης σκοπεύει στην εύρεση αναπαραστάσεων χαρακτηριστικών για τη μείωση της απόκλισης των πεδίων ορισμού και του σφάλματος ταξινόμησης ή παλινδρόμησης του μοντέλου. Οι στρατηγικές για την εύρεση τέτοιων αναπαραστάσεων χαρακτηριστικών διαφέρουν μεταξύ τους ανάλογα με το αρχικό πεδίο ορισμού. Αν πολλά ετικετοποιημένα δεδομένα είναι διαθέσιμα στο πεδίο ορισμού, μπορούν να χρησιμοποιηθούν τεχνικές επιβλεπόμενης μάθησης για την εύρεση των χαρακτηριστικών. Αυτό είναι όμοιο με τη συνήθη μάθηση πολλαπλών έργων. Αν δεν υπάρχουν ετικετοποιημένα δεδομένα στο αρχικό πεδίο ορισμού, τότε θα πρέπει να χρησιμοποιηθούν μέθοδοι μη επιβλεπόμενης μάθησης, ώστε να κατασκευασθούν τα χαρακτηριστικά που χρειάζονται για αυτή την προσέγγιση.

Παράδειγμα 1ο

Επιβλεπόμενη κατασκευή χαρακτηριστικών:

Όπως επισημάνθηκε παραπάνω οι μέθοδοι επιβλεπόμενης κατασκευής χαρακτηριστικών για την επαγωγική μετάδοση γνώσης είναι όμοιες με αυτές της μάθησης πολλαπλών έργων. Η βασική ιδέα είναι η εκμάθηση μίας χαμηλής διάστασης αναπαράστασης η οποία είναι κοινή μεταξύ σχετικών έργων. Επιπλέον, η καινούρια αυτή αναπαράσταση μπορεί να μειώσει το σφάλμα ταξινόμησης ή παλινδρόμησης του μοντέλου σε κάθε ένα από τα υποψήφια έργα.

Αυτή η προσέγγιση μπορεί να μελετηθεί και ως ένα πρόβλημα βελτιστοποίησης. Σε αυτό τα κοινά χαρακτηριστικά μπορούν να γίνουν γνωστά επιλύοντας το παρακάτω πρόβλημα:

$$\operatorname{argmin} \sum_{t \in \{T, S\}} \sum_{i=1}^{n_t} L(y_{ti}, \langle a_t, U^T x_{ti} \rangle) + \gamma \|A\|_{2,1}^2$$

$$U \in \mathbb{O}^d$$

Στην παραπάνω εξίσωση τα S , T δηλώνουν το αρχικό και το τελικό έργο στα αντίστοιχα πεδία ορισμού. Επίσης ο $A = [a_s, a_t] \in \mathbb{R}^{d \times 2}$ είναι ένας πίνακας των παραμέτρων. Ο U είναι ένας ορθογώνιος πίνακας $d \times d$ που απεικονίζει τα αρχικά δεδομένα υψηλής διάστασης σε αναπαραστάσεις χαμηλής διάστασης. Η νόρμα (r, p) του A ορίζεται ως $\|A\|_{r,p} = \left(\sum_{i=1}^d \|a^i\|_p^r \right)^{\frac{1}{r}}$. Το πρόβλημα αυτό υπολογίζει τις αναπαραστάσεις $U^T X_T$, $U^S X_S$ και τις παραμέτρους, A του μοντέλου ταυτοχρόνως. Το πρόβλημα βελτιστοποίησης μπορεί να μετατραπεί και σε ένα κυρτό πρόβλημα [7]. Η λύση αυτή αναλύεται περισσότερο στο [5], ωστόσο όπως είναι εύληπτο από τον αναγνώστη θα μπορούσε να χρησιμοποιηθεί και ένα νευρωνικό δίκτυο (όχι απαραίτητα βαθύ) για την επίλυση του προβλήματος.

Παράδειγμα 2ο

Άνευ επίβλεψης κατασκευή χαρακτηριστικών:

Ένα πολύ καλό παράδειγμα αυτής της προσέγγιση είναι ο νευρωνικός στατιστικολόγος όπως αναλύεται στο [17]. Σύμφωνα με αυτούς για να μεταφέρει μία μηχανή χαρακτηριστικά από το ένα σύνολο δεδομένων στο άλλο, θα πρέπει να καταλαβαίνει τις ομοιότητες. Για να το πράξει αυτό θα πρέπει να περιγράφει όλο το σύνολο δεδομένων και όχι να περιγράφει τα δεδομένα του απλά ως ανεπεξέργαστα σημεία. Προς αυτό το σκοπό χρησιμοποίησαν και επέκτειναν έναν μεταβλητό αυτόματο κωδικοποιητή [15] για να υπολογίσει στατιστικά ενός συνόλου δεδομένων άνευ επίβλεψης. Οι στατιστικές αυτές μπορούν να χρησιμοποιηθούν για την ομαδοποίηση συνόλων δεδομένων και την ταξινόμηση άγνωστων μέχρι τώρα συνόλων δεδομένων. Ως εκ τούτου και το όνομα του αλγορίθμου, αφού μπορεί να εξάγει στατιστικές των συνόλων δεδομένων άνευ επίβλεψης χρησιμοποιώντας ένα νευρωνικό δίκτυο.

Μία άλλη γνωστή μέθοδος που αναλύει αυτή την περίπτωση είναι η χρήση μάθησης πολλαπλοτήτων. Σε αυτή την εργασία [59] προτείνεται μία προσέγγιση προκρουστικής ανάλυσης για την διάταξη των πολλαπλοτήτων, η οποία μπορεί να χρησιμοποιηθεί στην μετάδοση γνώσης μεταξύ προβλημάτων.

Μετάδοση γνώσης των Παραμέτρων

Οι περισσότερες μέθοδοι μεταφοράς παραμέτρων για την επαγωγική μετάδοση γνώσης υποθέτουν ότι δύο ξεχωριστά ή ίδια μοντέλα για σχετικά έργα θα πρέπει να μοιραστούν κάποιες παραμέτρους ή κατανομές υπερπαραμέτρων. Τα μοντέλα που βρίσκονται υπό αυτή τη μέθοδο συνήθως αφορούν πολλαπλά έργα. Αυτό βέβαια δεν προβάλλει κάποιο πρόβλημα, απλά από το να παραλληλίσουμε τα δύο πεδία ορισμού, τοποθετώντας ένα μοντέλο, λαμβάνουμε το μοντέλο εκπαιδευμένο στο πρώτο πρόβλημα και χρησιμοποιούμε τα δεδομένα εστιάζοντας στην καλύτερη επίδοση του μοντέλου στο δεύτερο (τελικό) πρόβλημα.

Σε αυτή την κατηγορία υπόκειται και το δίκτυο SqueezeDet. Αρχικά οι ερευνητές που το δημιούργησαν στο πρώτο κομμάτι του που είναι το SqueezeNet έλαβαν όλες τις παραμέτρους από προηγούμενη εκπαίδευση του SqueezeNet στο ImageNet. Αυτή είναι περίπου και η μέθοδος που ακολουθείται ~~παρακάτω~~ στην εργασία. Ωστόσο, αυτή έχει και ερωτήματα όπως: ~~πως~~ να αποφασίσει κανείς πόσες παραμέτρους να λάβει από μία προηγούμενη εκπαίδευση, ποιες

και πόσες παραμέτρους χρειάζεται να επαναεκπαιδεύσει, μήπως υπάρχει κάποια σχέση πως να αλλάξει τις υπερπαραμέτρους με βάση την απόφαση λήψης των παραμέτρων; Αυτά και άλλα απαντώνται παρακάτω.

2.6.2 Μεταγωγική μετάδοση γνώσης

Δοθέντος ενός αρχικού πεδίου ορισμού D_s με ένα αρχικό έργο T_s και ένα τελικό πεδίο ορισμού D_t με ένα τελικό έργο T_t , η μεταγωγική μετάδοση γνώσης αποσκοπεί στη βελτίωση της μάθησης της τελικής συνάρτησης ετικετοποίησης f_t στο τελικό πεδίο ορισμού D_t χρησιμοποιώντας γνώση από τα D_s, T_s , όπου $D_s \neq D_t$ και $T_s = T_t$. Επιπλέον, θα πρέπει να υφίστανται κάποια μη ετικετοποιημένα δεδομένα στο τελικό πεδίο ορισμού στη διάρκεια της εκπαίδευσης.

Η έννοια της μεταγωγικής μετάδοσης γνώσης δε θα πρέπει να συγχέεται με την έννοια της μεταγωγικής μάθησης όπου όλα τα δεδομένα ελέγχου πρέπει να ~~γνωρίζονται~~ κατά την εκπαίδευση, η οποία δεν επιτρέπει το μοντέλο να επαναχρησιμοποιηθεί για μελλοντικά δεδομένα. Για να εξάγει αποτελέσματα από τα καινούρια δεδομένα θα πρέπει να τα κατηγοριοποιήσει στις ίδιες κατηγορίες με όλα τα υπάρχοντα. Παρομοίως όμως με τη μεταγωγική μάθηση, στη μεταγωγική μετάδοση γνώσης υποτίθεται ότι υπάρχουν μη ετικετοποιημένα δεδομένα στο τελικό πεδίο ορισμού.

Στον παραπάνω ορισμό της μεταγωγικής μετάδοσης γνώσης το αρχικό και το τελικό πεδίο ορισμού είναι τα ίδια, το οποίο συνεπάγεται ότι μπορεί η συνάρτηση πρόβλεψης του αρχικού προβλήματος να προσαρμοστεί για το δεύτερο και να χρησιμοποιηθεί για τα μη ετικετοποιημένα δεδομένα στο τελικό πεδίο ορισμού. Η προσαρμογή αυτή υπόκειται σε δύο περιπτώσεις: 1) Οι χώροι των των χαρακτηριστικών των δύο πεδίων ορισμού είναι διαφορετικοί $\mathcal{X}_s \neq \mathcal{X}_t$ και 2) οι χώροι των χαρακτηριστικών των πεδίων ορισμού είναι ίδιοι, αλλά οι πιθανοτικές κατανομές είναι διαφορετικές: $P(X_s) \neq P(X_t)$. Αυτό είναι σχεδόν ίδιο με τις απαιτήσεις της προσαρμογής του πεδίου ορισμού και του προβλήματος της πόλωσης των δειγμάτων λόγω δειγματοληψίας.

Μεταφορά γνώσης από στιγμιότυπα

Οι προσεγγίσεις αυτού του τύπου βασίζονται κυρίως στη σημασία της δειγματοληψίας. Προκειμένου να δειχθεί η σημασία της επιλογής μεθόδου δειγματοληψίας, παρουσιάζεται το μεγαλύτερο πρόβλημα της μεταφοράς γνώσης από στιγμιότυπα που είναι η ελαχιστοποίηση του εμπειρικού ρίσκου. Γενικότερα, θα ήταν επιθυμητό οι βελτιστοποιημένες παραμέτροι θ^* του μοντέλου να είναι βελτιστοποιημένες ως προς το αναμενόμενο ρίσκο:

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta} E_{(x,y) \in P} [l(x, y, \theta)]$$

όπου $l(x, y, \theta)$ είναι η συνάρτηση σφάλματος η οποία εξαρτάται από την παράμετρο θ . Ωστόσο είναι δύσκολο να εκτιμηθεί η κατανομή P . Για αυτό προτιμάται η ελαχιστοποίηση του εμπειρικού ρίσκου:

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n [l(x_i, y_i, \theta)]$$

όπου n είναι το μέγεθος των δεδομένων προς εκπαίδευση.

Στη μεταγωγική μετάδοση γνώσης, σκοπός είναι η εύρεση ενός βέλτιστου μοντέλου για το τελικό πεδίο ορισμού δεδομένων ελαχιστοποιώντας το αναμενόμενο ρίσκο:

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta} \sum_{(x,y) \in D_t} P(D_t) l(x, y, \theta)$$

Αν δεν υπάρχουν ετικετοποιημένα δεδομένα στο τελικό πεδίο ορισμού δεδομένων, τότε θα πρέπει το μοντέλο να εξαχθεί από τα δεδομένα του αρχικού πεδίου ορισμού. Αν $P(D_{d_S}) = P(D_{d_t})$, τότε αρκεί να βελτιστοποιήσουμε τις παραμέτρους του μοντέλου όπως παρακάτω:

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta} \sum_{(x,y) \in D_{d_S}} P(D_{d_S}) l(x, y, \theta).$$

Διαφορετικά αν $P(D_{d_S}) \neq P(D_{d_t})$, θα πρέπει να αλλάξουμε το παραπάνω πρόβλημα βελτιστοποίησης ώστε το μοντέλο που θα εξαχθεί από την εκπαίδευση να έχει δυνατότητα μεγάλης γενίκευσης στο τελικό πεδίο ορισμού. Η αλλαγή είναι η εξής:

$$\begin{aligned} \theta^* &= \operatorname{argmin}_{\theta \in \Theta} \sum_{(x,y) \in D_{d_S}} \frac{P(D_{d_t})}{P(D_{d_S})} P(D_{d_S}) l(x, y, \theta) \\ &\approx \operatorname{argmin}_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{P_t(x_{t_i}, y_{t_i})}{P(x_{S_i}, y_{S_i})} l(x_{S_i}, y_{S_i}, \theta). \end{aligned}$$

Σε αυτή την αλλαγή τοποθετούμε διαφορετικές τιμές ποινής στα στιγμιότυπα (x_{S_i}, y_{t_i}) λαπλασιάζοντας τα με $\frac{P_t(x_{t_i}, y_{t_i})}{P(x_{S_i}, y_{S_i})}$. Κατά αυτό τον τρόπο εξάγουμε ένα ακριβές μοντέλο για το τελικό πρόβλημα. Επιπρόσθετα, αφού τα έργα είναι κοινά, ισχύει $P(Y_t|X_t) = P(Y_S|X_S)$. Επομένως, ο λόγος των κατανομών μπορεί να προσεγγισθεί πιο εύκολα:

$$\frac{P(x_{t_i}, y_{t_i})}{P(x_{S_i}, y_{S_i})} = \frac{P(x_{t_i})}{P(x_{S_i})}.$$

Αν μπορέσουμε να υπολογίσουμε τα $\frac{P_t(x_{t_i})}{P(x_{S_i})}$ για κάθε στιγμιότυπο, μπορούμε να επιλύσουμε όλα τα προβλήματα που υπόκεινται στην κατηγορία της μεταγωγικής μετάδοσης γνώσης. Στη βιβλιογραφία υπάρχουν διάφοροι τρόποι για τον υπολογισμό του παραπάνω λόγου, ωστόσο ξεφεύγουν από το αντικείμενο αυτής της διπλωματικής.

Ένα άμεσα σχετιζόμενο παράδειγμα είναι η εργασία της *brighter.ai* [58] για τον εντοπισμό αντικειμένων σε εικόνες υπό διαφορετικές συνθήκες. Δηλαδή στα περισσότερα σύνολα δεδομένων για αλγορίθμους εντοπισμού, όπως και στις περισσότερες εκπαιδεύσεις, δε λαμβάνονται υπόψη περιπτώσεις που οι καιρικές συνθήκες ή η ώρα της ημέρας δυσχεραίνουν την όραση. Έτσι οι περισσότεροι αλγόριθμοι δεν μπορούν να χρησιμοποιηθούν σε αυτές τις περιπτώσεις.

Για την αντιμετώπιση αυτού του προβλήματος οι ερευνητές της *brighter.ai* χρησιμοποίησαν 2 νευρωνικά δίκτυα τύπου **GAN** συνδεδεμένα σειριακά, ώστε να απεικονίσουν τα δεδομένα της μίας συνθήκης στην άλλη κρατώντας τη μορφή των αντικειμένων ώστε να μπορούν να εντοπιστούν. Αν η κατάσταση στην οποία ο αλγόριθμος είναι θεμιτό να λειτουργήσει είναι η νύχτα, τότε: Το πρώτο δίκτυο GAN μετατρέπει την εικόνα από νύχτα σε μέρα και το δεύτερο GAN από μέρα σε νύχτα. Οι ερευνητές τονίζουν τη σημασία του δεύτερου, ώστε να μπορεί να υπάρχει μία ισομορφική απεικόνιση. Επίσης, ως μετρική απόστασης της κατασκευασμένης εικόνας και της πραγματικής από τα δίκτυα GAN χρησιμοποιούν την ευκλείδεια απόσταση μεταξύ των χαρακτηριστικών που εξάγονται κάποιο νευρωνικό δίκτυο κατηγοριοποίησης (πιθανόν εκπαιδευμένο στο αρχικό πεδίο ορισμού) στο αρχικό πεδίο ορισμού και στο τελικό πεδίο ορισμού. Ως σφάλμα προσέγγισης χρησιμοποιούν την απόσταση των εικόνων σε κάθε GAN και ένα άλλο σχεδόν αμετάβλητο χαρακτηριστικό, ανάλογα με τις συνθήκες των εικόνων. Το ίδιο δίκτυο με το οποίο εξήχθησαν τα χαρακτηριστικά, μετά μπορεί να χρησιμοποιηθεί για τον εντοπισμό αντικειμένων.

Μετάδοση γνώσης της αναπαράστασης χαρακτηριστικών

Οι περισσότερες μελέτες μεταγωγικής μετάδοσης γνώσης με τη μεταφορά χαρακτηριστικών, αφορούν αλγόριθμους μηχανικής μάθησης άνευ επίβλεψης. Για την καλύτερη κατανόηση αυτής της μεθόδου παρουσιάζονται δύο αλγόριθμοι.

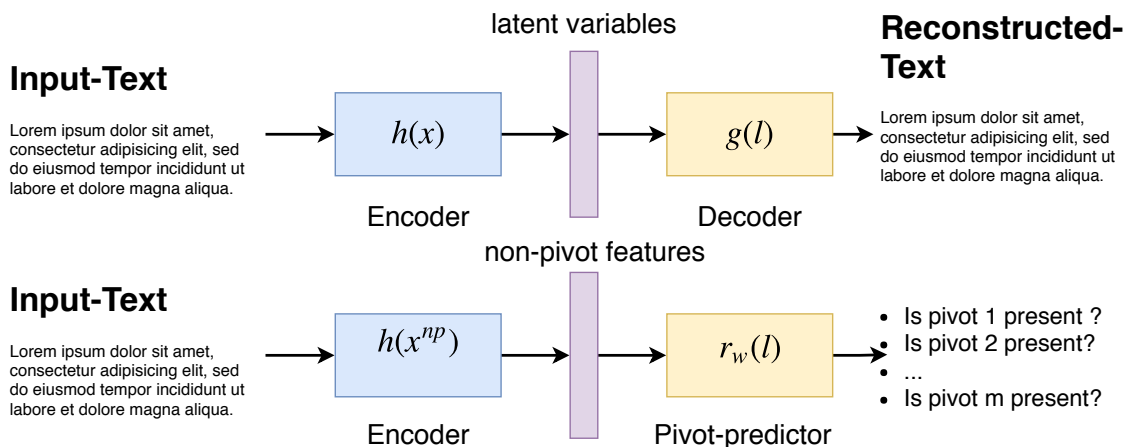
1ος Αλγόριθμος [65]

Νευρωνική μάθηση δομικής αντιστοιχίας:

Σε αυτή την εργασία οι συγγραφείς υιοθέτησαν τη μετάδοση γνώσης με δομική αντιστοιχία [13], αλλά προτίμησαν την αναζήτηση αυτής να την πραγματοποιήσουν με ένα πιο σύγχρονο τρόπο: τους νευρωνικούς αυτόματους κωδικοποιητές όπως εκθέτουν και οι ίδιοι. Το πρόβλημα βέβαια που επιλύουν αφορά την επεξεργασία της φυσικής γλώσσας (NLP). Αυτό βέβαια, δεν εμποδίζει την περίληψη αυτής της εργασίας στην παρούσα, αφού η αντιστοιχία μοντέλων που επιλύουν προβλήματα αυτού του τομέα με αυτά της τεχνητής όρασης είναι αρκετά συχνή.

Αρχικά η προσέγγιση αυτή ξεχωρίζει τα χαρακτηριστικά σε αξονικά χαρακτηριστικά (ή χαρακτηριστικά άξονα) και στα υπόλοιπα. Ο διαχωρισμός αυτός γίνεται εκ των προτέρων, δηλαδή πριν την εφαρμογή του αλγορίθμου και αφορά το αρχικό πεδίο ορισμού. Τα αξονικά χαρακτηριστικά δίνονται ως υπέρ-παράμετροι στον αλγόριθμο και έπειτα αυτός αποφασίζει κατά πόσο εφικτή είναι η περιγραφή των υπολοίπων χαρακτηριστικών από αυτά τα αξονικά χαρακτηριστικά.

Η αρχιτεκτονική της λύσης αναλύεται καλύτερα μέσω του Σχήματος [2.1]. διακρίνονται οι διαφορές μεταξύ ενός απλού αποκωδικοποιητή και αυτής της μεθόδου. Όπως παρατηρείται και στην εργασία της νευρωνικής μάθησης δομικής αντιστοιχίας είναι η πρώτη φορά που χρησιμοποιείται η αρχιτεκτονική του αυτόματου αποκωδικοποιητή για αυτό το σκοπό και για αυτό δεν χρησιμοποιήθηκε και κάποιος βαθύς κωδικοποιητής ή αποκωδικοποιητής.



Σχήμα 2.2: Στο πάνω μέρος της εικόνας βρίσκεται ένας απλός αυτόματος κωδικοποιητής. Στο κάτω μέρος της εικόνας βρίσκεται ο αυτόματος κωδικοποιητής αλλαγμένος στο κομμάτι του αποκωδικοποιητή όπως προβλέπει ο αλγόριθμος. Στο τέλος δε γίνεται ανακατασκευή της εισόδου, αλλά δίνεται η απάντηση αν κάποιο από τα αξονικά χαρακτηριστικά υπάρχει στην έξοδο του κωδικοποιητή. Το και το αποτέλεσμα της συνάρτησης $h(x^{np})$ είναι δυαδικά διανύσματα. Η βελτιστοποίηση γίνεται χρησιμοποιώντας τη συνάρτηση απόκλισης *Kullback-Leibler*.

2ος αλγόριθμος

Transfer Component Analysis (TCA): Ο αλγόριθμος αυτός ουσιαστικά εξάγει χαρακτηριστικά από τα δύο πεδία ορισμού, θεωρώντας ότι όλα τα κοινά χαρακτηριστικά ανήκουν σε

έναν αναπαραγόμενο χώρο πυρήνων Hilbert (RKHS). Τα δύο δεδομένα που λαμβάνει υπόψη του ο αλγόριθμος είναι ότι : $P(X_S) \neq P(X_t)$ και $P(Y_S|X_S) = P(Y_t|X_t)$. Έπειτα θεωρώντας ότι υπάρχει μία λανθάνουσα απεικόνιση $\phi : (X) \rightarrow \mathcal{H}$ για τα X_S, X_t η οποία είναι ισομορφισμός και ουσιαστικά δημιουργεί τα χαρακτηριστικά των δύο πεδίων ορισμού, ψάχνει αυτήν τη συνάρτηση ϕ η οποία ελαχιστοποιεί τη μετρική μέγιστης μέσης ασυμφωνίας (MMD):

$$Dist(X_S', X_t') = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_{S_i}) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(x_{t_i}) \right\|_{\mathcal{H}}$$

$$\phi^* = \operatorname{argmin}_{\phi} Dist(X_S', X_t')$$

Σε αυτό το σημείο κάνει τη θεώρηση κλειδί αυτής της εργασίας, ότι $P(Y_S|\phi(X_S)) = P(Y_t|\phi(X_t))$. Με αυτό και χρησιμοποιώντας το “κόλπο πυρήνα” $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ οδηγείται στον υπολογισμό της μετρικής μέσω του πίνακα K :

$$Dist(X_S', X_t') = \operatorname{tr}(KL), K = \begin{bmatrix} K_{S,S} & K_{S,t} \\ K_{t,S} & K_{t,t} \end{bmatrix}, L_{ij} = \frac{1}{n_i n_j}, n_i = \begin{cases} n_1 & \text{if } x_i \in X_S \\ n_2 & \text{if } x_i \in X_t \end{cases}$$

Βέβαια για τον υπολογισμό του K απαιτείται η λύση μέσω αλγορίθμων SD και η μετέπειτα επεξεργασία της λύσης μέσω PCA η μείωση της διάστασης των χαρακτηριστικών. Αυτό οδηγεί σε μεγάλη πολυπλοκότητα και απώλεια πληροφορίας. Οπότε προτείνεται ο ορισμός ενός δευτέρου πίνακα μέσω της εξίσωσης $K = (K K^{-1/2}) (K^{-1/2} K)$:

$$\tilde{K} = (K K^{-1/2}) \tilde{W} (\tilde{W}^T K^{-1/2} K) = K W W^T K, W = K^{-1/2} \tilde{W}$$

οδηγώντας στη μετρική:

$$Dist(X_S', X_T') = \operatorname{tr}((K W W^T K) L) = \operatorname{tr}(W^T K L K W)$$

Η οποία επιλύεται καλύτερα μέσω της αναζήτησης:

$$W^* = \operatorname{argmin}_W [\operatorname{tr}(W^T W) + \mu \operatorname{tr}(W^T K L K W)], W^T K L K W = I$$

$$H = I_{n_1+n_2} - \frac{1}{n_1 + n_2} \mathbf{1}_{n_1+n_2, n_1+n_2}$$

με μ ορισμένη ως παράμετρος συμβιβασμού. Το σύμβολο $\mathbf{1}_{n,m}$ είναι ένας πίνακας $n \times m$ με όλα τα στοιχεία του ίσα με 1. Η λύση αυτή δίνει πολυπλοκότητα $O(m(n_1 + n_2)^2)$.

Μετάδοση γνώσης των Παραμέτρων

Προφανώς στην μεταγωγική μετάδοση γνώσης δεν μπορεί να υπάρξει μετάδοση της γνώσης των παραμέτρων, διότι τα πεδία ορισμού είναι διαφορετικά. Για καλύτερη κατανόηση θα δώσουμε ένα παράδειγμα: Αν ένα μοντέλο είναι εκπαιδευμένο σε δεδομένα εικόνων σε πολικές συντεταγμένες, τότε δεν μπορούν να χρησιμοποιηθούν οι παράμετροι του για την επίλυση του ίδιου προβλήματος σε καρτεσιανές, απλά μεταφέροντας κάποιες παραμέτρους. Ούτε βέβαια, μπορεί να βοηθηθεί γιατί τα μετέπειτα επίπεδα του μοντέλου με πιο υψηλής διάστασης χαρακτηριστικά βασίζονται στα προγενέστερα επίπεδα με χαμηλότερης διάστασης χαρακτηριστικά τα οποία είναι βασισμένα σε πολικές συντεταγμένες.

2.6.3 Άνευ επίβλεψης μετάδοση γνώσης

Δοθέντος ενός αρχικού πεδίου ορισμού D_s με ένα αρχικό έργο T_s και ένα τελικό πεδίο ορισμού D_t με ένα τελικό έργο T_t , η άνευ επίβλεψης μετάδοση γνώσης αποσκοπεί στη βελτίωση της μάθησης της τελικής συνάρτησης ομαδοποίησης ή περιγραφής των δεδομένων f_t στο τελικό πεδίο ορισμού D_t χρησιμοποιώντας γνώση από τα D_s, T_s , όπου $T_s \neq T_t$ και τα Y_s, Y_t είναι μη παρατηρήσιμα.

Παρομοίως με τους ορισμούς που έχουν δοθεί στην απλή εκπαίδευση, η άνευ επίβλεψης μετάδοση γνώσης αφορά αρχικό και τελικό έργο χωρίς ορισμένες τις ετικέτες, δηλαδή χωρίς αυστηρά ορισμένη (ή είναι άγνωστη) συνάρτηση ετικετοποίησης. Ωστόσο υπάρχουν και άλλοι ορισμοί οι οποίοι πλαισιώνουν καλύτερα αυτό τον ορισμό υποκατηγοριοποιώντας τον σε συγκεκριμένα σενάρια. Σύμφωνα με τον Cook [10] υπάρχουν τέσσερις κατηγορίες

- IS: Πληροφοριοδοτημένη υπό επίβλεψη μετάδοση γνώσης, όπου η συνάρτηση ετικετοποίησης είναι γνωστή τόσο στο αρχικό έργο, όσο και στο τελικό.
- IU: Πληροφοριοδοτημένη άνευ επίβλεψη μετάδοση γνώσης, όπου η συνάρτηση ετικετοποίησης είναι γνωστή μόνο στο αρχικό έργο.
- US: Μη-πληροφοριοδοτημένη υπό επίβλεψη μετάδοση γνώσης, όπου η συνάρτηση ετικετοποίησης είναι γνωστή μόνο στο τελικό έργο.
- UU: Μη-Πληροφοριοδοτημένη άνευ επίβλεψη μετάδοση γνώσης, όπου η συνάρτηση ετικετοποίησης είναι άγνωστη τόσο στο αρχικό έργο, όσο και στο τελικό.

Διάφορες μέθοδοι έχουν αναπτυχθεί για την επίτευξη αποτελεσμάτων σε αυτές τις τέσσερις κατηγορίες. Η άνευ επίβλεψης μετάδοση γνώσης ονομάζεται και μάθηση μηδενικής λήψης (zero-shot learning), οι εργασίες που αφορούν τη μάθηση μηδενικής λήψης προσπαθούν να απεικονίσουν το αρχικό πεδίο ορισμού στο τελικό και χρησιμοποιώντας αυτή την απεικόνιση να εκπαιδεύσουν ευκολότερα το μοντέλο στο τελικό πεδίο ορισμού. Για παράδειγμα αυτό θα μπορούσε να γίνει ακόμα και για ένα SVM [19]. Παρόλα αυτά, σε όλες τις προσεγγίσεις συνεχίζουν να ισχύουν οι περιορισμοί των μεθόδων μάθησης άνευ επίβλεψης.

Αυτή η περίπτωση κυρίως περιλαμβάνει τη μεταφορά ομαδοποίησης από το αρχικό πεδίο ορισμού στο τελικό, ή την κατασκευή της τελικής ομαδοποίησης μέσω της τελικής. Η περίπτωση αναφέρεται απλά για πληρότητα. Για ένα παράδειγμα αυτής της μεθόδου ο αναγνώστης παραπέμπεται στον αλγόριθμο STC(αυτοδίδακτη ομαδοποίηση) [13].

2.7 Μεταφερσιμότητα παραμέτρων νευρωνικών δικτύων επεξεργασίας εικόνας[62]

Όπως έχει παρουσιαστεί στην εργασία των ερευνητών του KTH [14], τα χαρακτηριστικά των εικόνων είναι αρκετά κοινά, επομένως μπορούν οι ίδιοι οι παράμετροι να μεταφερθούν από τον έναν αλγόριθμο στον άλλο με αρκετά καλό αποτέλεσμα και υποδιαιρώντας αρκετά τον χρόνο εκπαίδευσης. Επίσης, οι ίδιοι ερευνητές έδειξαν πως απλά προσθέτοντας ένα *svm* στην κορυφή του δικτύου κρατώντας όλο το υπόλοιπο ίδιο και ανεκπαίδευτο στο τελικό πεδίο ορισμού, ήταν εφικτό τα ξεπεράσουν σχεδόν όλα τα αποτελέσματα των αλγορίθμων εκείνης της χρονικής περιόδου.

Όποτε γεννήθηκε ένα καινούριο ερώτημα στους ερευνητές, κατά πόσο αυτή η τεχνική είναι εφικτή, δηλαδή κατά πόσο είναι μεταφέρισιμοι οι παράμετροι των νευρωνικών δικτύων που επεξεργάζονται εικόνες[62]. Όπως αναφέρουν ότι έχουν παρατηρήσει: τα πρώτα επίπεδα κάθε νευρωνικού δικτύου είναι ίδια, αφού πρόκειται για φίλτρα *Gabor*, οπότε μάλιστα είναι

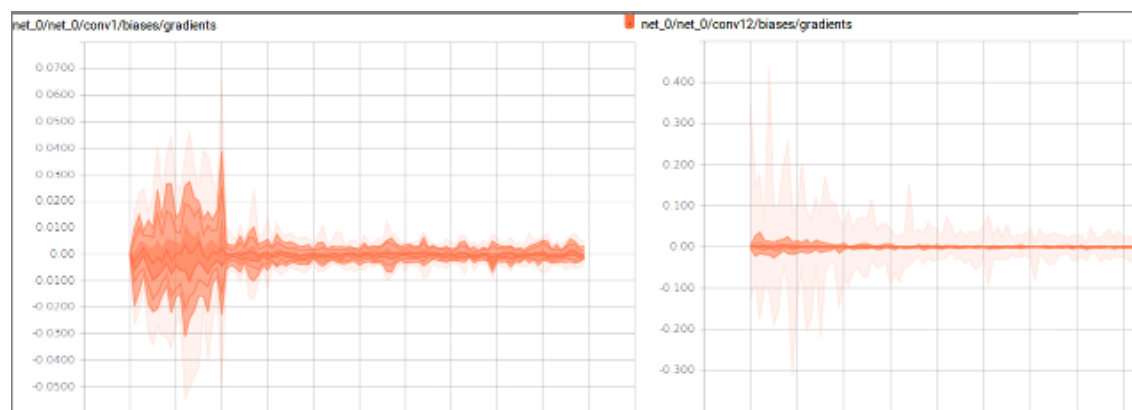
συμβατό με την περίπτωση μας διότι στις μετεκπαιδεύσεις του *SqueezeDet* σε καινούρια σύνολα δεδομένων, τα αρχικά επίπεδα είχαν παραγώγους σφάλματος όπως φαίνεται στο Σχήμα [2.3].

Προκειμένου να απαντήσουν οι ερευνητές στο ερώτημα, πραγματοποίησαν το εξής πείραμα: Λαμβάνοντας ως δίκτυο το AlexNet, πρώτα χώρισαν το ImageNet, με βάση τις κλάσεις του, σε δύο ξεχωριστά σύνολα δεδομένων (πεδία ορισμού) A, B και σε αυτά εκπαιδύσαν ξεχωριστά το δίκτυο, με αποτέλεσμα το δίκτυα Anet, Bnet αντίστοιχα. Ο χωρισμός έγινε με τέσσερις διαφορετικούς τυχαίους τρόπους. Έτσι δοκίμασαν την εκπαίδευση του AlexNet στο σύνολο δεδομένων B με τέσσερις διαφορετικούς τρόπους:

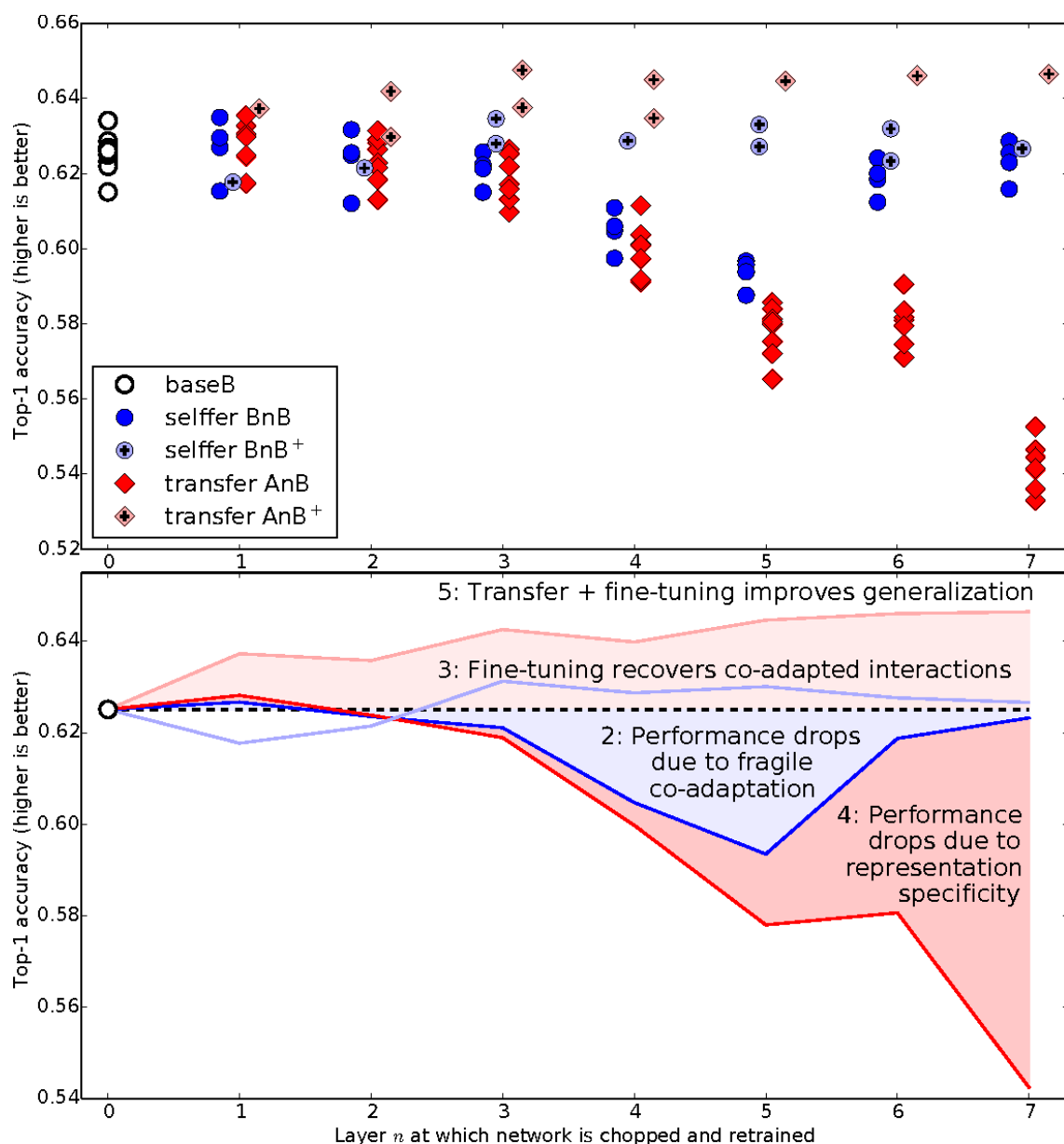
1. Στα n πρώτα επίπεδα μεταφέρθηκαν παράμετροι από το Anet και αυτά έμειναν μη εκπαιδευσιμα, ενώ τα υπόλοιπα εκπαιδευσιμα.
2. Στα n πρώτα επίπεδα μεταφέρθηκαν παράμετροι από το Anet και όλα τα επίπεδα ήταν εκπαιδευσιμα. Αυτό καλείται στην εργασία αυτή τελειοποίηση (*fine-tuning*) των μεταφερόμενων παραμέτρων.
3. Στα n πρώτα επίπεδα μεταφέρθηκαν παράμετροι από το Bnet και αυτά έμειναν μη εκπαιδευσιμα, ενώ τα υπόλοιπα εκπαιδευσιμα.
4. Στα n πρώτα επίπεδα μεταφέρθηκαν παράμετροι από το Bnet και όλα τα επίπεδα ήταν εκπαιδευσιμα. Αυτό καλείται στην εργασία αυτή ως τελειοποίηση των μεταφερόμενων παραμέτρων.

Οι δύο τελευταίες περιπτώσεις, έγιναν για να δείξουν ότι τα επίπεδα εξαρτώνται μεταξύ τους και κατά την εκπαίδευση έχουν εύθραυστη συν-προσαρμογή στα χαρακτηριστικά. Αυτό σημαίνει ότι για να γίνει σωστά η μετάδοση γνώσης δεν μπορούμε να διαλέξουμε τον παραπάνω αριθμό n χωρίς να ελέγξουμε τους αριθμούς $n - 1$, $n + 1$. Αυτό εξηγεί και την πτώση της έντονα κόκκινης γραμμής στο Σχήμα [2.4]. Τα αποτελέσματα που εξήγαγαν φαίνονται στο Σχήμα [2.4]. Αυτά φαίνονται να είναι θετικά για την περίπτωση της μεταφοράς με τελειοποίηση των παραμέτρων βελτιώνοντας την γενικότητα των χαρακτηριστικών. Επίσης, φαίνεται ότι τα τελευταία επίπεδα έχουν πιο ειδικά χαρακτηριστικά, ενώ τα πρώτα γενικότερα με αποτέλεσμα όταν το n πλησιάζει μεγάλες τιμές η ακρίβεια του δικτύου στο σύνολο B να γίνεται αρκετά μικρότερη.

Σε επόμενο πείραμα οι ερευνητές προχώρησαν και χώρισαν το ImageNet [14] πάλι σε δύο ξεχωριστά σύνολα A, B όπως πριν, αλλά με τη διαφορά ότι οι κλάσεις θα ήταν αρκετά διαφορετικές μεταξύ τους και όχι τυχαίες (για A είχαν το σύνολο των αντικειμένων κατασκευασμένων από τον άνθρωπο και για B είχαν το σύνολο πραγμάτων που βρίσκονται στη φύση). Και από αυτό το πείραμα κατέληξαν ότι τα χαρακτηριστικά συνεχίζουν να είναι γενικά και ότι είναι προτιμότερο να χρησιμοποιήσει κανείς αυτά τουλάχιστον στα πρώτα επίπεδα από ότι να κάνει εξολοκλήρου την εκπαίδευση στο σύνολο B.



Σχήμα 2.3: Η κατανομή της παραγώγου σφάλματος των παραμέτρων πόλωσης του πρώτου επιπέδου φαίνεται να έχει χαμηλότερα μέγιστα από ότι αυτές του τελευταίου επιπέδου κατά τάξη μεγέθους, δείχνοντας ότι το πρώτο επίπεδο του *SqueezeNet/SqueezeDet* είναι σχεδόν το ίδιο για τη μετεκπαίδευση στο τελικό πεδίο ορισμού PASCAL VOC2012[22] και στο αρχικό πεδίο ορισμού ImageNet.



Σχήμα 2.4: Πάνω: Κάθε σημείωση στο σχήμα δείχνει τη μέση ακρίβεια στο σύνολο ελέγχου για το εκπαιδευμένο δίκτυο. Οι άσπροι κύκλοι στο $n = 0$ δείχνουν την ακρίβεια του Bnet. Υπάρχουν οχτώ σημεία κάθε σημείωσης για κάθε n , διότι ελέγχθηκαν 4 διαφορετικοί τυχαίοι διαχωρισμοί του ImageNet. Κάθε κλειστός μπλε κύκλος παρουσιάζει την περίπτωση 3, ενώ κάθε μπλε κύκλος με σταυρό παρουσιάζει το 4. Κάθε κόκκινος ρόμβος παρουσιάζει την περίπτωση 1, ενώ κάθε κόκκινος ρόμβος με σταυρό παρουσιάζει την περίπτωση 2. Κάτω: Οι γραμμές που συνδέουν τα μέσα του πάνω διαγράμματος για κάθε μία από τις 4 περιπτώσεις. Οι αριθμοί και τα κείμενα στην εικόνα [επισημαίνουν](#) το λόγο αυτής της συμπεριφοράς.

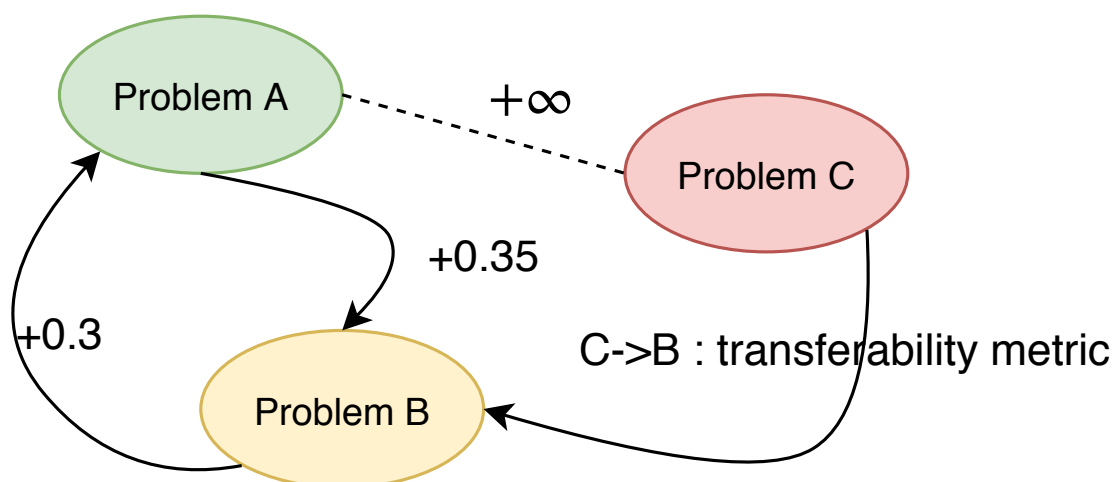
2.8 Αρνητική μετάδοση Γνώσης

Ήδη από το πείραμα της προηγούμενης παραγράφου είναι κατανοητό ότι υπάρχουν όρια στη μεταφερσιμότητα των παραμέτρων. Όσο πιο διαφορετικά είναι το αρχικό με το τελικό πεδίο ορισμού, τόσο πιο δύσκολο είναι για το νευρωνικό να προσαρμοστεί. Μάλιστα, το πόσο

μπορεί να προσαρμοστεί ένας αλγόριθμος έχει και θεωρητικό όριο στην περίπτωση που είναι μορφής Bayes με βάση την πολυπλοκότητα Kolmogorov [39].

Αυτά όλα, είναι αρκετά για να φανταστούμε την περίπτωση της αρνητικής μετάδοσης γνώσης, η οποία συμβαίνει “όταν η μεταφορά γνώσης από το αρχικό στο τελικό πρόβλημα προκαλεί μείωση της επίδοσης του αλγορίθμου” [45]. Αυτό μπορεί να συμβαίνει για τρεις λόγους:

- **Λάθος πληροφορία:** Αν ο αλγόριθμος μετάδοσης γνώσης αφαιρεί πληροφορία από το πρώτο πρόβλημα, την οποία θεωρεί ως επιβλαβή για την τελική εκπαίδευση, τότε η εκπαίδευση δεν είναι καλύτερη από ~~ότι~~ αν ο αλγόριθμος εκπαιδεύεται μόνο στο τελικό πρόβλημα.
- **Επιλογή αρχικού έργου:** Αν υπάρχει παραπάνω από μία επιλογή για το αρχικό σύνολο, ο αλγόριθμος μετάδοσης γνώσης υπάρχει περίπτωση να μην επιλέξει το βέλτιστο ή να διαλέξει να μην επιλέξει κανένα. Για την καλύτερη εικόνα επιλογής αρχικού προβλήματος, ο Eaton [16] πρότεινε τη δημιουργία ενός γράφου όπου η απόσταση κάθε κόμβου από τους υπόλοιπους είναι ίση με μία μετρική μεταφερσιμότητας, όπως φαίνεται στο Σχήμα [2.5]
- **Ανομοιότητα στο αρχικό και το τελικό έργο:** Αν αυτά τα δύο έργα είναι πολύ διαφορετικά, τότε υπάρχει μεγαλύτερο ρίσκο στην μεταγωγική μετάδοση γνώσης να προκληθεί μεγαλύτερο σφάλμα λόγω χρήσης της γνώσης του αρχικού προβλήματος. Αυτό συμβαίνει γιατί όσο πιο όμοια είναι δύο έργα, τόσο πιο κοινά είναι τα χαμηλών διαστάσεων χαρακτηριστικά τους [6].



Σχήμα 2.5: Κατευθυνόμενος γράφος όπου κάθε κόμβος είναι ένα πρόβλημα με το δικό του έργο και το δικό του πεδίο ορισμού και κάθε ακμή έχει βάρος τη μετρική μεταφερσιμότητας από το αρχικό πρόβλημα προς το τελικό. Αυτός ο γράφος δεν παρουσιάζεται με αυτό το σχήμα στην εργασία [16], ωστόσο αυτό εννοείται. Επίσης επιβάλλεται να είναι κατευθυνόμενος λόγω της καταστροφικής λήθης.

2.9 Καταστροφική λήθη [34]

Τα νευρωνικά δίκτυα είναι κυρίως εμπνευσμένα από τη βιολογία του εγκεφάλου των θηλαστικών, ο οποίος εμπεριέχει νευρώνες οι οποίοι μαθαίνουν και εκπαιδεύονται σε καινούρια

προβλήματα καθ' όλη τη διάρκεια της ζωής του ατόμου. Αυτό σημαίνει ότι αν μαθαίνουμε να επιλύουμε ένα πρόβλημα δεν αποκλείει να μάθουμε την επίλυση ενός άλλου τουλάχιστον σε σύντομο χρονικό διάστημα. Δηλαδή οι νευρώνες έχουν ελαστικότητα ως προς τη μάθηση.

Η παρατήρηση που έκαναν οι ερευνητές στο [34] είναι ότι αυτό δε συμβαίνει μέχρι τώρα στις εκπαιδεύσεις των νευρωνικών δικτύων. Αυτό που συμβαίνει στην εφαρμογή της μετάδοσης γνώσης ή και στην απλή εκπαίδευση είναι ότι η βελτίωση της επίδοσης γίνεται μόνο για το αρχικό ή το τελικό πρόβλημα και αν έχουμε μία σειρά προβλημάτων π.χ. στο Σχήμα [2.5] η διαδρομή $C \rightarrow B \rightarrow A$ δεν είναι εφικτή, διότι η μετάδοση γνώσης από το C στο B και η εκπαίδευση στο πρόβλημα B δεν επιτρέπουν το δίκτυο να ξανά-προσαρμοστεί στο A. Ωστόσο αν έχουμε δύο τέλειες λύσεις για τα προβλήματα A, B, τότε αυτές είναι κοντά όπως φαίνεται στο Σχήμα [2.6], αλλά ο χώρος στον οποίο η μία λύση πρέπει να ανήκει για να είναι προσβάσιμη από την άλλη διαμέσου του μοντέλου είναι σε μία περιοχή γύρω από τη τέλεια λύση και όχι πάνω σε αυτή.

Οπότε, είναι θεμιτό να υπολογιστεί αυτή η περιοχή. Ο υπολογισμός αυτής εξαρτάται από τα δεδομένα από ένα υποσύνολο \mathcal{D} του πεδίου ορισμού. Αν δηλαδή θ είναι οι παράμετροι του μοντέλου, τότε η πιθανότητα αυτή να είναι η τέλεια λύση για το πεδίο ορισμού \mathcal{D} είναι με βάση τον κανόνα Bayes:

$$\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}|\theta) + \log p(\theta) - \log p(\mathcal{D})$$

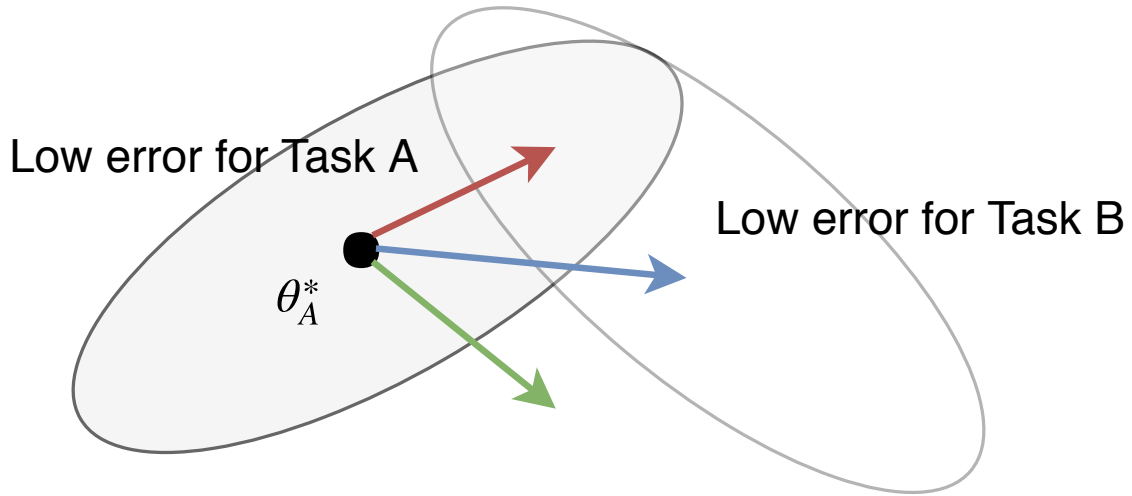
Οπότε αν συμβαίνει μεταφορά δεδομένων από το πρόβλημα A στο B, η πιθανότητα η θ να είναι η τέλεια λύση για κάποιο υποσύνολο δεδομένων \mathcal{D} :

$$\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A) - \log p(\mathcal{D}_B)$$

Η πραγματική πιθανότητα $\log p(\theta|\mathcal{D}_A)$ είναι μη παρατηρήσιμη, οπότε αυτή προσεγγίζεται με μία κατανομή Gauss με μέση τιμή τις βέλτιστες παραμέτρους για το πρόβλημα A: θ_A^* και με τη διαγώνιο του πίνακα ακρίβειας ίση με τη διαγώνιο του πίνακα πληροφορίας Fisher. Το αποτέλεσμα είναι ότι για να κάνουμε τους νευρώνες πιο ελαστικούς κατά την εκπαίδευση στο πρόβλημα B χρησιμοποιούμε τη μετρική:

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_{A,i}^*)^2$$

όπου $\mathcal{L}_B(\theta)$ είναι το σφάλμα κατά την εκπαίδευση στο πρόβλημα B, λαμβάνοντας μόνο υπόψη το πεδίο ορισμού και το έργο του B.



Σχήμα 2.6: Ο αλγόριθμος EWC όπως προτείνεται στο [34]. Προβάλλονται οι περιοχές στις οποίες η λύση είναι ικανοποιητική για τα δύο προβλήματα A,B. Επίσης, δείχνεται ποια η διαφορά μεταξύ των τριών σφαλμάτων: Με κόκκινο η προτεινόμενη μετρική οδηγεί σε μία περιοχή των παραμέτρων η οποία έχει κοινά ικανοποιητικό σφάλμα. Με μπλε η μετρική που δεν έχει βάρος ως προς το μέτρο των παραμέτρων εξέρχεται από την περιοχή της κοινά ικανοποιητικής λύσης. Με πράσινο αν η μετρική σφάλματος λαμβάνει υπόψη το μέτρο των παραμέτρων χρησιμοποιώντας την ευκλείδεια απόσταση αυτών, μπορεί να οδηγηθεί εντελώς εκτός των δύο περιοχών.

2.10 Προσβασιμότητα



Με βάση τη θεωρία αυτομάτου ελέγχου, το παραπάνω πρόβλημα μπορεί να αντιστοιχηθεί σε πρόβλημα προσβασιμότητας. Οι ορισμοί από τη θεωρία προσβασιμότητας είναι οι εξής:

■ Δοθέντος ενός συστήματος $S(x, u, t)$ και μίας αρχικής κατάστασης x_0 , η κατάσταση x_f θεωρείται προσβάσιμη από την x_0 με είσοδο $u(t)$ αν το σύστημα εντός πεπερασμένου χρόνου t_f μπορεί να μεταβεί από την κατάσταση x_0 στην κατάσταση x_f .

Επίσης υπάρχει και ο χρήσιμος ορισμός του προσβάσιμου συνόλου:

■ Δοθέντος ενός συστήματος $S(x, u, t)$ και μίας αρχικής κατάστασης x_0 , όλες οι καταστάσεις x_j οι οποίες είναι προσβάσιμες δημιουργούν το προσβάσιμο σύνολο $R^S(x_0) = \{z \in X : x_0 \rightarrow z\}$.

■ Αν Θ είναι η περιοχή με ικανοποιητική επίδοση, το αδιέξοδο σύνολο ορίζεται ως $N(S, \Theta) = \{z \in X : R^S(z) \cap \Theta\}$.

Στον παραπάνω ορισμό το αδιέξοδο σύνολο είναι δηλαδή οι καταστάσεις από τις οποίες δεν είναι εφικτή η μετάβαση στο σύνολο Θ . Έτσι μπορούμε να θέσουμε ότι η μετάδοση γνώσης είναι ή ο μετασχηματισμός του χώρου εισόδου, ώστε το προσβάσιμο σύνολο να εμπεριέχει περιοχή με ικανοποιητικό σφάλμα ή η μετακίνηση της αρχικής θέσης x_0 έτσι ώστε το καινούριο προσβάσιμο σύνολο να εμπεριέχει περιοχή με ικανοποιητικό σφάλμα.

Στην κλασική μηχανική μάθηση όπου η εκπαίδευση γίνεται από την αρχή στο καινούριο

πρόβλημα, δημιουργούνται τα εξής προβλήματα:

- Έστω ότι γίνεται τυχαία επιλογή των παραμέτρων του μοντέλου S με βάση κάποια συνάρτηση αθροιστικής κατανομής P . Αν η περιοχή με ικανοποιητική επίδοση είναι η Θ , τότε η αρχική κατάσταση των παραμέτρων μπορεί να βρεθεί στο αδιέξοδο σύνολο $N(S, \theta)$, διότι εν γένει δεν ισχύει $P(N(S, \Theta)) = 0$. Οπότε, μπορεί να χρειαστούν αρκετές επαναλήψεις της εκπαίδευσης ώστε η αρχική κατάσταση να μη βρεθεί σε αυτό το σύνολο.
- Έστω ότι η επιλογή γίνεται από τον ερευνητή, τότε αυτός θα πρέπει να δοκιμάσει αρκετές τιμές, ώστε το σύνολο των αρχικών καταστάσεων που δοκίμασε να μην βρίσκεται εξολοκλήρου στο αδιέξοδο σύστημα.

Κεφάλαιο 3

Πειραματισμοί και αποτελέσματα

3.1 Το πρόβλημα και η προτεινόμενη λύση

Οι αλγόριθμοι και τα παραδείγματα που αναφέρθηκαν σε ολόκληρο το κεφάλαιο 2, μέχρι στιγμής αφορούσαν αλγορίθμους οι οποίοι δεν ήταν πραγματικού χρόνου και εμπεριείχαν μεγάλο πλήθος παραμέτρων και άρα μπορούσαν να χειριστούν περισσότερα χαρακτηριστικά χαμηλής διάστασης των πεδίων ορισμού. Σε αντίθεση οι αλγόριθμοι πραγματικού χρόνου έχουν μικρότερο πλήθος χαρακτηριστικών και αυτό δεν βοηθάει καθόλου στη μεταφερσιμότητά τους.

Επιπλέον, τα νευρωνικά δίκτυα εντοπισμού αντικειμένων χρειάζεται να εντοπίζουν ένα αντικείμενο σε οποιαδήποτε θέση βρίσκεται πάνω στην εικόνα. Στα δίκτυα πραγματικού χρόνου υπεύθυνος για τον εντοπισμό ενός αντικειμένου σε μία περιοχή είναι ένας νευρώνας. Αυτό επιταχύνει τον χρόνο εκτέλεσης του δικτύου, ωστόσο κάνει πιο δύσκολη την εκπαίδευση, αφού όλοι σχεδόν οι νευρώνες θα πρέπει να μπορούν να εντοπίσουν ένα αντικείμενο στην περιοχή για την οποία είναι υπεύθυνοι. Ταυτόχρονα δυσχεραίνεται και η μεταφερσιμότητα και σχεδόν γίνεται αδύνατη, στα τελευταία επίπεδα του νευρωνικού.

Δεν υπάρχει κάποια αναφορά (εν γνώση του συγγραφέα) σε δοκιμές για τη μεταφερσιμότητα των παραμέτρων νευρωνικών δικτύων εντοπισμού αντικειμένων που λειτουργούν σε πραγματικό χρόνο. Οπότε πριν μπορέσει να υπάρξει οποιαδήποτε μελλοντική εφαρμογή θα πρέπει να γίνουν πειραματισμοί.

3.2 Το πείραμα

Οι πειραματισμοί που θα πρέπει να γίνουν είναι η χρήση ενός νευρωνικού δικτύου εντοπισμού αντικειμένων πραγματικού χρόνου για την πραγματοποίηση πρώτα του πειράματος που περιγράφεται στην ενότητα 2.7 και έπειτα του πειράματος που περιγράφεται στην ενότητα 2.9.

Με το πρώτο πείραμα μπορεί να παρουσιαστεί η συμπεριφορά της απλής επαγωγικής μετάδοσης γνώσης σε αυτού του τύπου δίκτυα. Επίσης, μέσω της συμπεριφοράς αυτής θα γίνει σαφές αν θα πρέπει να προτιμάται η μετάδοση γνώσης ή η μάθηση πολλαπλών έργων. Η μάθηση πολλαπλών έργων ωστόσο δεν μπορεί να γίνει για έναν αφηρημένο αριθμό κλάσεων αντικειμένων, διότι το δίκτυο τότε μπορεί να σταματήσει να συμπεριφέρεται ως δίκτυο πραγματικού χρόνου.

Επιπρόσθετα, ήδη ο μεγάλος αριθμός κλάσεων δημιουργεί πρόβλημα στο SqueezeDet από άποψη της μνήμης που χρησιμοποιεί. Ο σκοπός δημιουργίας του δικτύου είναι αυτό να έχει μικρή μνήμη, ώστε να μπορεί να χρησιμοποιηθεί από μικρής ισχύος ενσωματωμένες συσκευές. Αν αυξηθεί ο αριθμός των κλάσεων κατά πολύ, η αρχιτεκτονική του δικτύου οδηγείται

στην καταστροφή αυτής της ικανότητας. Ο τύπος του μεγέθους του τελευταίου επιπέδου του squeezeDet παρουσιάζεται στην παρακάτω εξίσωση:

$$final_layer_shape = 3 \times 3 \times K * (C + 1 + 4)$$

με K τον αριθμό των υπεύθυνων νευρώνων ανά περιοχή πλέγματος και C τον αριθμό των κλάσεων.

Η επιρροή της μικρής μνήμης είναι πολύ μεγαλύτερη στο δεύτερο πείραμα της εργασίας. Οι λίγοι παράμετροι όμως το κάνουν να μην μπορεί να επαναχρησιμοποιηθεί. Με αυτό το πείραμα μπορεί να εξεταστεί κατά πόσο “πλαστικό” είναι το δίκτυο στη μετάδοση γνώσης.

Για το πρώτο κομμάτι αυτού του πειράματος ακολουθούνται τα εξής βήματα:

1. Μετάδοση γνώσης μέσω της μεταφοράς των παραμέτρων από το εκπαιδευμένο squeezeNet στο ImageNet στο SqueezeDet.
2. Εκπαίδευση του δικτύου SqueezeDet στις μισές κλάσεις του PASCAL VOC 2012.
3. Μετάδοση γνώσης μέσω της μεταφοράς των παραμέτρων από όλα τα επίπεδα πλην τα τρία τελευταία από το εκπαιδευμένο squeezeDet στο PASCAL VOC2012 στο SqueezeDet.
4. Εκπαίδευση του δικτύου SqueezeDet στις άλλες μισές κλάσεις του PASCAL VOC 2012.

Η εκπαίδευση γίνεται με συμπίληψη της μετρικής που αναφέρεται στην ενότητα 2.9.

Για το δεύτερο κομμάτι αυτού του πειράματος ακολουθούνται τα εξής βήματα:

1. Μετάδοση γνώσης μέσω της μεταφοράς των παραμέτρων από το εκπαιδευμένο squeezeNet στο ImageNet στο SqueezeDet.
2. Εκπαίδευση γνώσης απευθείας στις δεύτερες μισές κλάσεις του PASCAL VOC2012

Στο τέλος αυτών των δύο κομματιών του πειράματος συγκρίνουμε τα αποτελέσματα μεταξύ τους.

3.3 Αυτόματη επιλογή υπερπαραμέτρων

Κατά την εκπαίδευση δεν υπάρχει βεβαιότητα ότι η επιλογή των υπερπαραμέτρων ήταν η βέλτιστη ή τουλάχιστον αρκετά κοντά σε αυτή. Για αυτό το σκοπό πέρα από την αναζήτηση τύπου δοκιμή/ σφάλμα, χρησιμοποιήθηκε και μία άλλη μέθοδος βελτιστοποίησης των υπερπαραμέτρων χωρίς τη χρήση παραγώγων (derivative free). Η μέθοδος αυτή αναφέρεται στο [40] και πρόκειται για μέθοδο καθολικής βελτιστοποίησης.

Ο μόνος περιορισμός που λαμβάνει υπόψη αυτή η μέθοδος είναι ότι η συνάρτηση για την οποία γίνεται η αναζήτηση της βέλτιστης τιμής θα πρέπει να είναι τύπου *Lipschitz*. Βεβαίως, αυτό είναι ήδη γνωστό για τα νευρωνικά δίκτυα [?] και αυτό επιτρέπει την χρήση αυτής της μεθόδου. Οποιαδήποτε άλλη ιδιότητα και αν έχει το νευρωνικό ως μαύρο κουτί ο αλγόριθμος βελτιστοποίησης των παραμέτρων το χρησιμοποιεί ως μαύρο κουτί.

Προκειμένου να εξετασθεί η σωστή λειτουργία αυτού του αλγορίθμου, δοκιμάσαμε μία εκπαίδευση με μία μόνο κλάση του PASCAL VOC χρησιμοποιώντας το squeezeDet. Επίσης για σύγκριση δοκιμάσαμε έναν αλγόριθμο τυχαίας αναζήτησης (*random search*) και έναν γενετικό. Η σύγκριση των τριών αλγορίθμων φαίνεται στον πίνακα [??]. Τα ίδια αποτελέσματα φαίνονται οπτικοποιημένα στο Σχήμα [3.1].

[εικόνα αποτελεσμάτων]

Πίνακας 3.1: My caption

[HTML]DAE8FC Όνομα βελτιστοποιητή	mAP	Αριθμός βημάτων

Τέλος, για να δικαιολογηθεί περαιτέρω η χρήση τέτοιων μεθόδων, επισημαίνεται ότι με αυτές δίνεται η δυνατότητα ο ερευνητής να έχει μία εποπτεία των περιοχών στις οποίες οι υπερπαραμέτροι κάνουν το μοντέλο να αποκτά πιο θεμιτή συμπεριφορά. Ταυτόχρονα, η αναζήτηση αυτή γίνεται πιο γρήγορα χωρίς κενά χρόνου στα οποία οι υπολογιστικοί πόροι μένουν ανεχμετάλλευτοι.

3.4 Επιτάχυνση του δικτύου SqueezeDet

3.5 Σύνοψη & Συμπεράσματα

Κεφάλαιο 4

Υλικό & λογισμικό

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά εργαλεία που χρησιμοποιήθηκαν τόσο για τις υλοποιήσεις όσο και για τα πειράματα. Η αναφορά αυτή γίνεται για τον ερευνητή ο οποίος θέλει να εξετάσει λεπτομέρειες της υλοποίησης. Όλα τα σχετικά αρχεία και ο κώδικας βρίσκεται στο [?]. Στο υποκεφάλαιο 4.1 παρουσιάζεται το υλικό πάνω στο οποίο βασίστηκε η εργασία, ενώ στο υποκεφάλαιο 4.2 παρουσιάζεται η υλοποίηση του λογισμικού και τα εργαλεία που χρησιμοποιήθηκαν.

4.1 Περιγραφή του υλικού

Η παρούσα διπλωματική έγινε στην υπολογιστική συστοιχία του εργαστηρίου Αρχιτεκτονικής Υπολογιστών και Συστημάτων. Τα κύρια μέρη της συστοιχίας που αφορούν είναι 1) η CPU 2) η GPU 3) RAM και ο σκληρός δίσκος. Για περισσότερες λεπτομέρειες, επειδή σε μελλοντικά πειράματα μπορεί να υπάρχουν επιπλέον παράμετροι υλικού, ολόκληρη η ανάλυση του υλικού της συστοιχίας παρατίθεται σε έναν υπερσύνδεσμο¹.

1. CPU: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz ²
2. GPU: NVIDIA GEFORCE GTX 1080 Ti ³
3. RAM: 2× DIMM Synchronous 2400 MHz, 8GiB, 64 bits
4. hard drive: ATA Disk HDD(για τα σύνολα δεδομένων και τα αποτελέσματα) + ATA Disk SSD (για αποθήκευση του κώδικα)

4.2 Βασικά στοιχεία λογισμικού υλοποίησης

Σε αυτή την ενότητα παρουσιάζονται τα δύο βασικότερα κομμάτια της υλοποίησης και τα πακέτα λογισμικού τρίτων.

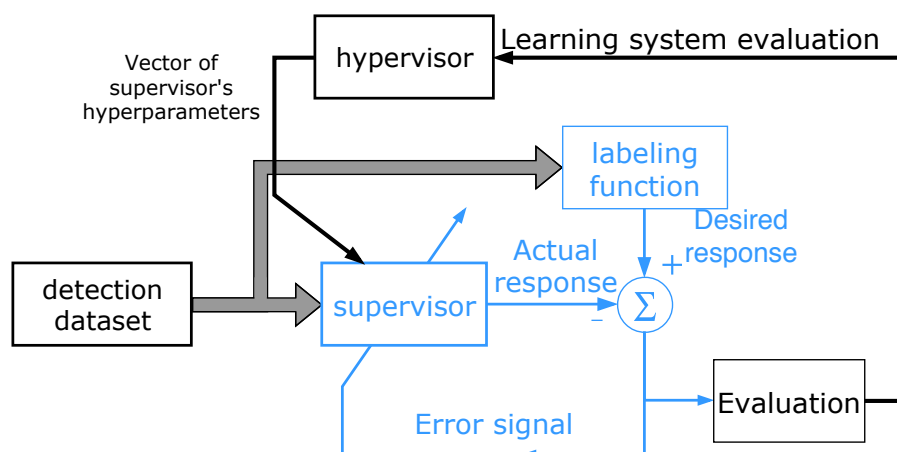
4.3 Περιγραφή της υλοποίησης

Η υλοποίηση είναι εμπνευσμένη από το σχήμα 24 στο βιβλίο νευρωνικών δικτύων του Haykin [24]. Η εικόνα αυτή διαφοροποιημένη φαίνεται στο Σχήμα [4.1]. Εκεί παρουσιάζεται

¹<https://drive.google.com/file/d/1c7IuY3qEbZ3VoLnWwF4ZrtjjNa501Rf/view?usp=sharing>

²https://ark.intel.com/products/97129/Intel-Core-i7-7700K-Processor-8M-Cache-up-to-4_50-GHz

³<https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/>



Σχήμα 4.1: Η λειτουργία της υλοποίησης όπως είναι κατά την πορεία της εκπαίδευσης. Ο *supervisor* είναι υπεύθυνος για την εκπαίδευση και βελτιστοποίηση των παραμέτρων του μοντέλου, ενώ ο *hypervisor* είναι υπεύθυνος για τη βελτιστοποίηση των παραμέτρων του *supervisor*.

οπτικά και ο τρόπος λειτουργίας της υλοποίησης. Σύμφωνα με αυτή έχουμε τα δύο μπλοκ *supervisor* και *hypervisor* τα οποία υλοποιούνται στα ομώνυμα πακέτα. Τα δύο αυτά πακέτα αναλύονται παρακάτω.

4.3.1 supervisor

Ο *supervisor* είναι υπεύθυνος για την εκπαίδευση του νευρωνικού δικτύου που επιλέγει ο χρήστης. Δηλαδή λαμβάνει ως παραμέτρους το δίκτυο, το σύνολο δεδομένων και όλες τις υπερπαραμέτρους που απαιτούνται κατά την εκπαίδευση από τον *hypervisor* ή τον χρήστη και αναλαμβάνει την συνεχή εκτέλεση του αλγορίθμου back-propagation για το δίκτυο μέχρι αυτό να συγκλίνει στο επιθυμητό οριακό σφάλμα ορισμένο από τον χρήστη ή να αποκλίνει.

Στην πρώτη περίπτωση, επιστρέφονται τα βάρη του δικτύου και ο γράφος της αρχιτεκτονικής του για την ανακατασκευή του. Στη δεύτερη περίπτωση, επιστρέφεται μήνυμα σφάλματος ότι η εκπαίδευση με αυτές τις υπερπαραμέτρους οδηγεί σε αποκλίνουσα συμπεριφορά.

Το σφάλμα το οποίο εξάγεται κατά την διάρκεια της εκπαίδευσης ορίζεται από μία συνάρτηση που αφορά το δίκτυο. Βέβαια, στη παρούσα περίπτωση πρόκειται για την συνάρτηση κόστους του squeezeDet. Οπότε, δεν αφήνεται περιθώριο για αλλαγές στην εξαγωγή του σφάλματος πέρα από μεταβολή των συντελεστών επιμέρους συντελεστών της συνάρτησης.

Οι υπερπαραμέτροι που μπορούν να εισαχθούν για την εκπαίδευση του δικτύου είναι αρκετές. Ωστόσο προκύπτει ότι είναι πιο εύληπτο να εξηγήσουμε αυτές, παρά τον πολύπλοκο μηχανισμό που τις υλοποιεί. Παρόλα αυτά, όταν δίνονται οι υπερπαραμέτροι σε ένα αρχείο, πιθανόν να περιλαμβάνονται παραπάνω πληροφορίες μέσα στο αρχείο. Οι υπερπαραμέτροι που λαμβάνει υπόψη του το πακέτο *supervisor* είναι οι εξής:

Λίστα υπερπαραμέτρων

- **CLASS_NAMES** : Τα ονόματα των κλάσεων τα οποία θα κατηγοριοποιηθούν τα εντοπισμένα αντικείμενα.
- **CLASSES** : Ο αριθμός των ονομάτων των κλάσεων.
- **LEAKY_COEF** : Παράμετρος διαρροής για τις συναρτήσεις leaky ReLU.

- **KEEP_PROB** : Πιθανότητα να παραμείνει ανεπηρέαστος ένας κόμβος του νευρωνικού δικτύου μετά από εφαρμογή dropout.
- **IMAGE_WIDTH** : Πλάτος εικόνας. Είναι το ίδιο για κάθε εικόνα, αν όχι τότε γίνεται μετατροπή πλάτους της εικόνας εισόδου σε αυτό.
- **IMAGE_HEIGHT** : Ύψος εικόνας. Είναι το ίδιο για κάθε εικόνα, αν όχι τότε γίνεται μετατροπή ύψους της εικόνας εισόδου σε αυτό.
- **ANCHOR_BOX** : Πίνακας ο οποίος εμπεριέχει "anchors" όπως λέγονται στο squeezeDet [61], δηλαδή στοιχεία τύπου (cx, xy, w, h) , όπου cx η τετμημένη κέντρου του anchor, cy η τεταγμένη κέντρου του anchor, w το πλάτος του anchor και h το ύψος του anchor.
- **ANCHORS** : Ο συνολικός αριθμός των anchor
- **ANCHOR_PER_GRID** : Ο αριθμός των anchor ανά σημείο του τελικού πλέγματος (βλέπε [61]).
- **INITIAL_ANCHOR_SHAPES** : Πίνακας με προκαθορισμένες τιμές του πλάτους και του ύψους κάθε anchor.
- **BATCH_SIZE** : Μέγεθος της παρτίδας για ευρωστότερη εκπαίδευση, συνίσταται ο αριθμός αυτός να είναι ίσος με 1 για μετεκπαίδευση σε μικρά σετ δεδομένων.
- **PROB_THRESH** : Κατώφλι το οποίο ορίζει αν ένα υποψήφιο εντοπισμένο αντικείμενο θα θεωρηθεί ως ορθό και θα δοθεί ως είσοδο στον αλγόριθμο NMS.
- **PLOT_PROB_THRESH** : Κατώφλι το οποίο ορίζει αν ένα εντοπισμένο αντικείμενο θα προβληθεί στα αποτελέσματα προς οπτικοποίηση. Χρησιμοποιείται μόνο κατά την οπτικοποίηση των αποτελεσμάτων.
- **NMS_THRESH** : Ανώφλι μετρικής iou μεταξύ των κουτιών που εντοπίστηκαν. Αν δύο κουτιά έχουν μεταξύ τους μεγαλύτερη τιμή iou, τότε κάποιο από αυτά θα αφαιρεθεί όπως προβλέπεται από τον αλγόριθμο NMS.
- **TOP_N_DETECTION** : Ο μέγιστος αριθμός αντικειμένων τα οποία θα εξαχθούν από μία εικόνα στο πέρα του αλγορίθμου του νευρωνικού δικτύου.
- **BGR_MEANS** : Μέση τιμή του χρώματος των pixel στο σετ δεδομένων σε σειρά BGR ως ένας πίνακας (1, 1, 3).
- **LOSS_COEF_CONF** : Συντελεστής σφάλματος εμπιστοσύνης παλινδρόμησης.
- **LOSS_COEF_CLASS** : Συντελεστής σφάλματος παλινδρόμησης της κατηγοριοποίησης.
- **LOSS_COEF_BBOX** : Συντελεστής σφάλματος παλινδρόμησης του περιβλήματος των υποψήφιων εντοπισμένων αντικειμένων.
- **LOSS_COEF_CONF_POS** : Συντελεστής θετικού σφάλματος εμπιστοσύνης για τη παλινδρόμηση του σκορ.
- **LOSS_COEF_CONF_NEG** : Συντελεστής αρνητικού σφάλματος εμπιστοσύνης για τη παλινδρόμηση του σκορ.
- **DECAY_STEPS** : Αριθμός βημάτων κατά τον οποίο γίνεται σμίχρυνση του ρυθμού ανανέωσης βαρών.
- **LR_DECAY_FACTOR** : Αριθμός με τον οποίο πολλαπλασιάζεται ο ρυθμός μάθησης κάθε **DECAY_STEPS** επαναλήψεις.
- **LEARNING_RATE** : Αρχικός ρυθμός μάθησης(ανανέωσης βαρών).

- **MOMENTUM** : Ορμή, Χρησιμοποιείται για τον βελτιστοποιητή ορμής [51].
- **BETA1** : Παράμετρος του βελτιστοποιητή Adam beta1.
- **BETA2** : Παράμετρος του βελτιστοποιητή Adam beta2.
- **WEIGHT_DECAY** : ποσοστό μείωσης της τιμής του μέτρου των παραμέτρων κατά την ανανέωση τους σε κάθε επανάληψη.
- **LOAD_PRETRAINED_MODEL** : Αν τα βάρη θα αρχικοποιηθούν όπως στο [61].
- **PRETRAINED_MODEL_PATH** : Διαδρομή αρχείου για την αρχικοποίηση του αρχείου. Πρόκειται για αρχείο τύπου *pkl*.
- **DEBUG_MODE** : Ενεργοποίηση ή όχι της λειτουργίας αποσφαλμάτωσης.
- **EPSILON** : Πολύ μικρή τιμή για να αποτραπεί η αριθμητική αστάθεια στις πράξεις.
- **EXP_THRESH** : Ανώφλι για τις ασφαλείς εκθετικές συναρτήσεις.
- **MAX_GRAD_NORM** : Ανώφλι μέτρου παραγώγων. Παράγωγοι με τιμή μεγαλύτερη δε λαμβάνονται υπόψη.
- **DATA_AUGMENTATION** : Αν θα γίνει προσαύξηση των δεδομένων με πρόσθεση θορύβου.
- **EXCLUDE_HARD_EXAMPLES** : Υπερπαραμέτρος που αφορά το σύνολο δεδομένων KITTI.
- **BATCH_NORM_EPSILON** : Μικρή τιμή που χρησιμοποιείται στους παρονομαστές για τις διαιρέσεις που απαιτεί η εκπαίδευση κατά παρτίδες. Η προκαθορισμένη τιμή είναι αυτή που χρησιμοποιεί και το Caffe framework [31].
- **FREEZE_LAYERS** : Αντικείμενο που ορίζει για κάθε επίπεδο του νευρωνικού αν θα εκπαιδευτεί ή όχι.
- **IS_TRAINING** : Αν πρόκειται για εκπαίδευση(true) ή απλά για εκτίμηση του μοντέλου(false).
- **NUM_THREAD** : Αριθμός των νημάτων τα οποία εξάγουν δεδομένα από το σετ δεδομένων.
- **MAX_NUM_PARALLEL_NETS** : Μέγιστος αριθμός παράλληλων εκπαιδύσεων.
- **PREPROCESS_DATASET** : Η αληθής τιμή αυτής της υπερπαραμέτρου επιτρέπει την προ-επεξεργασία όλου του σετ δεδομένων και την εξαγωγή των μεταβλητών *aidx* και *deltas* για γρηγορότερη εκπαίδευση αργότερα. Ωστόσο, αν η προ-επεξεργασία ενεργοποιηθεί δεν επιτρέπει μοντέλα με διαφορετική αρχιτεκτονική στο επίπεδο που εξάγονται τα *anchor*.

4.3.2 hypervisor

Το πακέτο αυτό είναι υπεύθυνο για το χειρισμό των υπερπαραμέτρων. Στο κύριο μέρος του υλοποιεί έναν βελτιστοποιητή μαύρου κουτιού όπως αυτός που περιγράφεται στην ενότητα 3.3. Υπάρχει επιλογή μεταξύ τεσσάρων αλγορίθμων όπως περιγράφονται στην παρακάτω λίστα.

1. random search: Τυχαία αναζήτηση η οποία δοκιμάζει κάθε φορά καινούριες τυχαίες τιμές ως τιμές των παραμέτρων, παραγόμενες από μία γεννήτρια τυχαίων αριθμών.
2. LIPO: Ο πρώτος αλγόριθμος που παρουσιάζεται στην εργασία του Nicolas και του Cedric [40]. Οι σταθερές k που χρησιμοποιούνται είναι 1, 10, 100. Επίσης ορίζεται μέγιστος αριθμός επαναλήψεων ίσος με 50.

3. adaLIPO: Ο δεύτερος αλγόριθμος που παρουσιάζεται στην ίδια εργασία. Στη μόνη ρύθμιση που αρκείται είναι στο μέγιστο αριθμό επαναλήψεων (ίσος με 50).
4. genetic: Ένας τυπικός γενετικός αλγόριθμος σύμφωνα με τις προτάσεις στο [4]. Οι ρυθμίσεις του είναι: ??????????.

Την τιμή εκκίνησης και την περιοχή (πλέγμα) στην οποία γίνεται η αναζήτηση της βέλτιστης τιμής των υπερπαραμέτρων την βάζει ο χρήστης για κάθε αλγόριθμο βελτιστοποίησης. Όπως επίσης, εισάγει και τις τιμές ρύθμισης τους αν αυτές απαιτούνται. Εδώ να τονιστεί πως αυτό δεν χρειάζεται για τον αλγόριθμο *adalipo*, αυτό είναι και ένα από τα πλεονεκτήματά του.

Το πακέτο *hypervisor* λαμβάνει ως τιμές τις συνάρτησης βελτιστοποίησης την έξοδο (χύρια μετρική) της διαδικασίας εκτίμησης του μοντέλου. Η μετρική που μελετάται στη συγκεκριμένη περίπτωση είναι η mAP. Η προτίμηση αυτή γίνεται για την καλύτερη σύνδεση με την πρωταρχική εργασία του *SqueezeDet*.

Αν και υπάρχει πληθώρα άλλων πακέτων για αυτή την υλοποίηση, ωστόσο τα υπόλοιπα δεν παρουσιάζονται διότι δεν έχουν μεγάλη σημασία.

4.3.3 Περιγραφή πακέτων λογισμικού τρίτων

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε ήταν η Python 2.7, διότι είναι πιο πλήρως υποστηριζόμενη από το *Tensorflow* (βλέπε παρακάτω). Το λειτουργικό της συστοιχίας ήταν *Ubuntu 16.04.4 LTS (Xenial Xerus)*. Τα πακέτα που χρησιμοποιήθηκαν στη τελική υλοποίηση δεν ήταν πολλά γιατί όλες οι υπολογιστικές ανάγκες αυτής της εργασίας καλύπτονταν από τις βιβλιοθήκες *Tensorflow* και *NumPy*.

- *Tensorflow*⁴ [2]: Ανοικτού κώδικα βιβλιοθήκη για αριθμητικούς υπολογισμούς χρησιμοποιώντας γράφους ροής δεδομένων ανεπτυγμένο από την ερευνητική ομάδα της *Google*. Η ευέλικτη αρχιτεκτονική της επιτρέπει την ανάπτυξη λογισμικού σε μία ή και περισσότερες κεντρικές μονάδες επεξεργασίας CPU ή μονάδες GPU ακόμα και για ενσωματωμένες συσκευές.
- *NumPy*⁵: *NumPy* είναι το βασικό πακέτο για επιστημονικούς υπολογισμούς στην Python. Η χρήση του βοήθησε κυρίως στην υλοποίηση του *hypervisor*, διότι αυτός εκτελείται από τη CPU.
- *Matplotlib*⁶: Βιβλιοθήκη της Python για κατασκευή διαγραμμάτων.
- *Google-Protocol-buffers*(*protoc*)⁷: Επεκτάσιμος μηχανισμός της *Google* για τη σειριοποίηση δομημένων δεδομένων. Ο μηχανισμός αυτός είναι ανεξάρτητος πλατφόρμας, είναι παρόμοιος με αυτόν της XML, ωστόσο μικρότερος και γρηγορότερος. Η κάθε ανάγνωση και συγγραφή γίνεται με συναρτήσεις οι οποίες έχουν παραχθεί από τον μηχανισμό αυτό ειδικά για κάθε περίπτωση.
- *vod-converter*⁸: Εργαλείο για την μετατροπή της μορφής των συνόλων δεδομένων KITTI, PASCAL VOC από το ένα σύνολο δεδομένων στο άλλο. Η παρούσα εργασία συνείσφερε και στην ανάπτυξη αυτού του εργαλείου, πέρα από την χρήση του.

⁴<https://www.tensorflow.org>

⁵<http://www.numpy.org>

⁶<https://matplotlib.org/>

⁷<https://developers.google.com/protocol-buffers>

⁸<https://github.com/umautobots/vod-converter>

Κεφάλαιο 5

Μελλοντικές επεκτάσεις

Βιβλιογραφία

- [1] YOLOv2 low fps on tiny memory devices <https://github.com/thtrieu/darkflow/issues/16>, 2016.
- [2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard και others. Tensorflow: A system for large-scale machine learning. Στο *OSDI*, τόμος 16, σελίδες 265--283, 2016.
- [3] Sam Abrahams. Inception-v3 speed on tensorflow, <https://github.com/samjabrahams/tensorflow-on-raspberry-pi/tree/master/benchmarks/inceptionv3>, 2017.
- [4] Ali Alajmi και Jonathan Wright. Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem. *International Journal of Sustainable Built Environment*, 3(1):18--26, 2014.
- [5] Andreas Argyriou, Theodoros Evgeniou και Massimiliano Pontil. Multi-task feature learning. Στο *Advances in neural information processing systems*, σελίδες 41--48, 2007.
- [6] Andreas Argyriou, Andreas Maurer και Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. Στο *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, σελίδες 71--85. Springer, 2008.
- [7] Andreas Argyriou, Charles A Micchelli, Massimiliano Pontil και Yiming Ying. A spectral regularization framework for multi-task structure learning, nips 20. *Journal Publications on Mathematics (Harmonic Analysis)*, 2008.
- [8] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho και Yoshua Bengio. Attention-based models for speech recognition. Στο *Advances in Neural Information Processing Systems*, σελίδες 577--585, 2015.
- [9] COCO Consortium. COCO DATASET detection evaluation <http://mscoco.org/dataset/detections-eval>, 2017.
- [10] Diane Cook, Kyle D Feuz και Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537--556, 2013.
- [11] Nicolás Cruz, Kenzo Lobos-Tsunekawa και Javier Ruiz-del-Solar. Using convolutional neural networks in robots with limited computational resources: Detecting NAO robots while playing soccer. *CoRR*, abs/1706.06702, 2017.

- [12] Wenyuan Dai, Qiang Yang, Gui Rong Xue και Yong Yu. Boosting for transfer learning. Στο *Proceedings of the 24th international conference on Machine learning*, σελίδες 193--200. ACM, 2007.
- [13] Wenyuan Dai, Qiang Yang, Gui Rong Xue και Yong Yu. Self-taught clustering. Στο *Proceedings of the 25th international conference on Machine learning*, σελίδες 200--207. ACM, 2008.
- [14] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li και Li Fei-Fei. Imagenet: A large-scale hierarchical image database. Στο *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, σελίδες 248--255. IEEE, 2009.
- [15] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [16] Eric Eaton, Terran Lane και others. Modeling transfer relationships between learning tasks for improved inductive transfer. Στο *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, σελίδες 317--332. Springer, 2008.
- [17] Harrison Edwards και Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.
- [18] Dumitru Erhan, Christian Szegedy, Alexander Toshev και Dragomir Anguelov. Scalable object detection using deep neural networks. Στο *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 2147--2154, 2014.
- [19] Stratis Gavves, Thomas Mensink, Tatiana Tommasi, Cees Snoek και Tinne Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. Στο *Proceedings ICCV 2015*, σελίδες 2731--2739, 2015.
- [20] Ross Girshick. Fast r-cnn. Στο *Proceedings of the IEEE international conference on computer vision*, σελίδες 1440--1448, 2015.
- [21] Ross Girshick, Jeff Donahue, Trevor Darrell και Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 580--587, 2014.
- [22] Ross Girshick, Jeff Donahue, Trevor Darrell και Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 580--587, 2014.
- [23] Ian Goodfellow, Yoshua Bengio, Aaron Courville και Yoshua Bengio. *Deep learning*, τόμος 1. MIT press Cambridge, 2016.
- [24] Simon S Haykin, Simon S Haykin, Simon S Haykin και Simon S Haykin. *Neural networks and learning machines*, τόμος 3. Pearson Upper Saddle River, NJ, USA:, 2009.
- [25] Kaiming He, Georgia Gkioxari, Piotr Dollár και Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. Deep residual learning for image recognition. Στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 770--778, 2016.
- [27] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto και Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [28] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama και others. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.
- [29] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally και Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <> 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.
- [30] Max Jaderberg, Andrea Vedaldi και Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.
- [31] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama και Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. Στο *Proceedings of the 22nd ACM international conference on Multimedia*, σελίδες 675--678. ACM, 2014.
- [32] Justin Johnson. cnn benchmarks
<https://github.com/jcjohnson/cnn-benchmarks>, 2017.
- [33] Diederik Kingma και Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [34] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska και others. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521--3526, 2017.
- [35] Alex Krizhevsky, Ilya Sutskever και Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Στο *Advances in neural information processing systems*, σελίδες 1097--1105, 2012.
- [36] Yann LeCun, Léon Bottou, Yoshua Bengio και Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324, 1998.
- [37] Tsung Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan και Serge Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*, 2016.
- [38] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu και Alexander C Berg. Ssd: Single shot multibox detector. Στο *European conference on computer vision*, σελίδες 21--37. Springer, 2016.

- [39] MM Mahmud και Sylvian Ray. Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations. Στο *Advances in neural information processing systems*, σελίδες 985--992, 2008.
- [40] Cédric Malherbe και Nicolas Vayatis. Global optimization of lipschitz functions. *arXiv preprint arXiv:1703.02628*, 2017.
- [41] Mohammad Motamedi, Daniel Fong και Soheil Ghiasi. Fast and energy-efficient cnn inference on iot devices. *arXiv preprint arXiv:1611.07151*, 2016.
- [42] Sinno Jialin Pan και Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345--1359, 2010.
- [43] Adam Paszke, Abhishek Chaurasia, Sangpil Kim και Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.
- [44] David N Perkins, Gavriel Salomon και others. Transfer of learning. *International encyclopedia of education*, 2:6452--6457, 1992.
- [45] David N Perkins, Gavriel Salomon και others. Transfer of learning. *International encyclopedia of education*, 2:6452--6457, 1992.
- [46] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon και Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. Στο *European Conference on Computer Vision*, σελίδες 525--542. Springer, 2016.
- [47] Joseph Redmon. YOLOv2 <http://pjreddie.com/yolo/>, 2017.
- [48] Joseph Redmon, Santosh Divvala, Ross Girshick και Ali Farhadi. You only look once: Unified, real-time object detection. Στο *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 779--788, 2016.
- [49] Joseph Redmon και Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
- [50] Shaoqing Ren, Kaiming He, Ross Girshick και Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Στο *Advances in neural information processing systems*, σελίδες 91--99, 2015.
- [51] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [52] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.
- [53] Jethro Shell. Fuzzy transfer learning. 2013.
- [54] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever και Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929--1958, 2014.
- [55] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke και Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. Στο *AAAI*, σελίδες 4278--4284, 2017.

- [56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke και Andrew Rabinovich. Going deeper with convolutions. Στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 1--9, 2015.
- [57] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers και Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154--171, 2013.
- [58] Elias Vansteenkiste και Patrick Kern. Taming adversarial domain transfer with structural constraints for image enhancement. *arXiv preprint arXiv:1712.00598*, 2017.
- [59] Chang Wang και Sridhar Mahadevan. Manifold alignment using procrustes analysis. Στο *Proceedings of the 25th international conference on Machine learning*, σελίδες 1120--1127. ACM, 2008.
- [60] David H Wolpert και William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67--82, 1997.
- [61] Bichen Wu, Forrest Iandola, Peter H Jin και Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. Στο *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, σελίδες 446--454. IEEE, 2017.
- [62] Jason Yosinski, Jeff Clune, Yoshua Bengio και Hod Lipson. How transferable are features in deep neural networks? Στο *Advances in neural information processing systems*, σελίδες 3320--3328, 2014.
- [63] Qian Yu, Yongxin Yang, Yi Zhe Song, Tao Xiang και Timothy Hospedales. Sketch-a-net that beats humans. *arXiv preprint arXiv:1501.07873*, 2015.
- [64] Wei Zhang, Kazuyoshi Itoh, Jun Tanida και Yoshiki Ichioka. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*, 29(32):4790--4797, 1990.
- [65] Yftah Ziser και Roi Reichart. Neural structural correspondence learning for domain adaptation. *arXiv preprint arXiv:1610.01588*, 2016.

