

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



UNIVERSITY of WEST ATTICA

FACULTY OF ENGINEERING

Department of Electrical & Electronics Engineering

Πρόγραμμα Μεταπτυχιακών Σπουδών

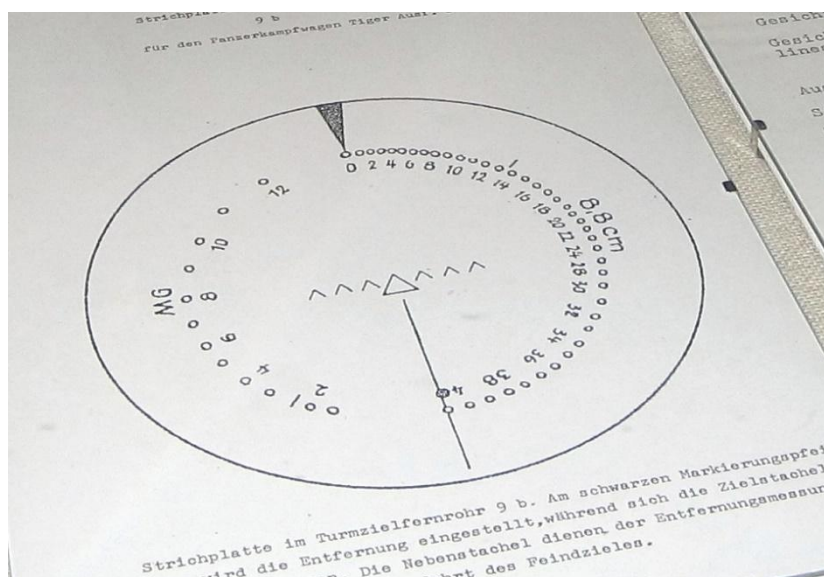
Διαδικτυωμένα Ηλεκτρονικά Συστήματα

Master of Science in

Internetworked Electronic Systems

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μηχανική όραση – εφαρμογή βλητικού υπολογιστή



Μεταπτυχιακός Φοιτητής : Κοσμίδης Γεώργιος, Α.Μ. IES-0028

Επιβλέπων καθηγητής : Μετάφας Δημήτριος, Επίκ. Καθηγητής

ΑΙΓΑΛΕΩ, ΝΟΕΜΒΡΙΟΣ 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



UNIVERSITY of WEST ATTICA

FACULTY OF ENGINEERING

Department of Electrical & Electronics Engineering

Πρόγραμμα Μεταπτυχιακών Σπουδών

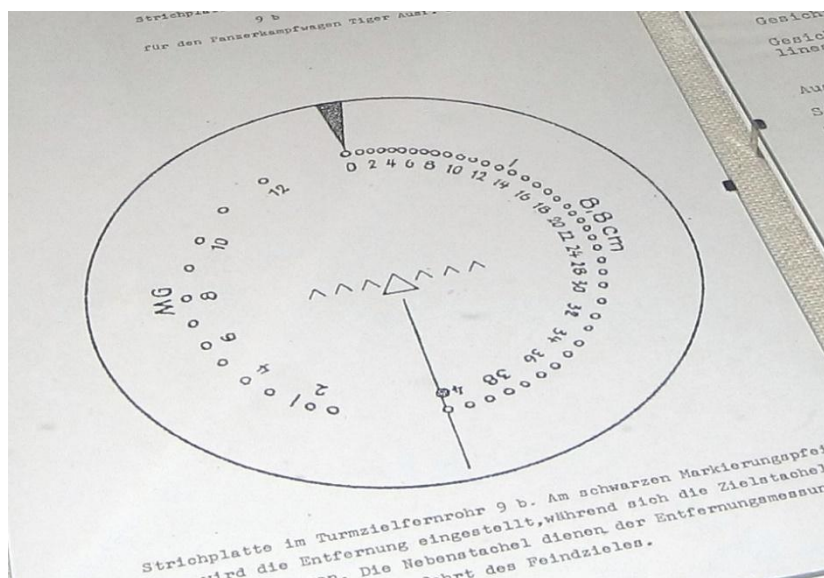
Διαδικτυωμένα Ηλεκτρονικά Συστήματα

Master of Science in

Internetworked Electronic Systems

MSc Thesis

MACHINE VISION – BALLISTIC COMPUTER



Student: KOSMIDIS GEORGIOS, Reg. Nr. IES-0028

MSc Thesis Supervisor: METAFAS DIMITRIOS, Assist. Prof.

ATHENS-EGALEO, NOVEMBER 2019

Περίληψη

Σκοπός της εργασίας είναι να αναδείξει σε πρακτικό επίπεδο τις δυνατότητες των νέων τεχνολογιών και συγκεκριμένα της μηχανικής μάθησης με χρήση νευρωνικών δικτύων σε απλά καθημερινά ή όχι, προβλήματα.

Είναι μια εφαρμογή μηχανικής μάθησης σχεδίασης σκοπευτικού συστήματος. Αν και η χρήση του θα μπορούσε να επεκταθεί σε πολλούς τομείς, εδώ επικεντρώνεται ως βοηθητικός βλητικός υπολογιστής άρματος. Για λόγους πρακτικής επίδειξης χρησιμοποιεί ορολογία και γραφικό περιβάλλον όμοιο με αυτό των αρμάτων. Η εφαρμογή ολοκληρώνεται με ένα σύστημα σερβοκινητήρων για την επίδραση με το περιβάλλον, ενός από τα πολλά αισθητήρια/ενεργοποιητές που θα χρειαζόνταν για να γίνει ρεαλιστικό το σύνολο.

Στην εργασία αυτή δεν γίνεται σφαιρική ανάλυση όλων των τεχνολογιών που χρησιμοποιούνται ή που θα μπορούσαν να αποτελούν εναλλακτικές λύσεις τους. Δεν είναι ένα εγχειρίδιο νευρωνικών δικτύων ή μηχανικής μάθησης. Βαδίζοντας προς το πρακτικό μέρος της εφαρμογής, γίνεται μια αναφορά στο περιβάλλον από το οποίο προέρχεται η κάθε τεχνολογία στο βαθμό που χρειάζεται για να γίνει κατανοητή η χρήση της.

Οι τεχνικές μηχανικής μάθησης αναπτύσσονται ραγδαία τα τελευταία χρόνια. Είναι χαρακτηριστικό ότι στους μήνες που διήρκεσε ο σχεδιασμός και η ανάπτυξη αυτής της εργασίας, καινούργια προγραμματιστικά εργαλεία δημιουργήθηκαν ή έστω σταθεροποιήθηκαν, ενώ παρατηρήθηκαν και μεγάλες μετακινήσεις μηχανικών από μια πλατφόρμα σε άλλη με μορφή ρεύματος. Κριτήρια για αυτά είναι η υποστήριξη που παρέχουν μεγάλες εταιρίες κατασκευής hardware στις διάφορες πλατφόρμες καθώς οι απαιτήσεις σε υπολογιστική δύναμη είναι μεγάλες και η βελτιστοποίηση του hardware για συγκεκριμένους αλγόριθμους οδηγεί σε αύξηση ζήτησης.

ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ :

αναγνώριση αντικειμένων, βλητικός υπολογιστής, ιχνηλάτης

Abstract

The purpose of this work is to bring to light the potential of new technologies and in particular of machine learning using neural networks in simple everyday or not problems.

It is a machine learning application of targeting system. Although its use could be extended to many areas, it focuses here as a ballistic computer for battle tanks. For practical purposes, it uses terminology and a graphical interface similar to the battle tank environment. The application is completed with a servomotor system for environmental impact, one of the many sensors / actuators that would be needed to make the whole app realistic.

This work does not provide a comprehensive analysis of all the technologies used or could be used as alternatives. It is not a neural network or machine learning manual. Moving on to the practical part of the application, a reference is made to the environment from which each technology originates, up to the point that is necessary to understand its use.

Machine learning techniques have grown rapidly in recent years. It is noteworthy that in the months leading up to the design and development of this work, new programming tools were created or at least stabilized, and large movements of engineers were observed from one platform to another in the form of wave. Criteria for this are the support provided by major hardware manufacturers across platforms as computing requirements are high and hardware optimization for specific algorithms leads to increased demand.

KEYWORDS:

ballistic computer, object detection, tracker, YOLO

Περιεχόμενα

Περίληψη.....	3
Abstract	4
OpenCV.....	7
Χρωματοχώρος – Color space	7
Βαθμιαίο διάνυσμα κλίσης εικόνας - Image Gradient Vector	8
HOG (histogram of oriented gradients).....	9
Τεχνητή νοημοσύνη	12
Μηχανική μάθηση – machine learning	14
Εποπτευόμενη μάθηση - supervised learning	14
Μη εποπτευόμενη μάθηση - unsupervised learning.....	14
Ενισχυτική μάθηση - reinforcement learning	15
Μάθηση δια της μεταφορά – transfer learning.....	15
Αλγόριθμοι λήψης αποφάσεων	16
Deep Learning – Βαθιά εκμάθηση	27
Τεχνητό Νευρικό Δίκτυο – ANN	28
Βαθύ νευρωνικό δίκτυο - Deep Neural Network	31
Δομή νευρωνικού δικτύου	33
Συναρτήσεις ενεργοποίησης.....	40
CNN – Convolutional NN (ConvNet)	44
RNN - Recurrent Neural Networks	47
Αναγνώριση αντικειμένου	50
Παραδοσιακές τεχνικές.....	50
Δίκτυο βάσης ενός πλαισίου ανίχνευσης αντικειμένων.....	51
YOLO (You only look once) Εδώ η εικόνα τροφοδοτείται μόνο μια φορά στο cnn δίκτυο	54
Μετρήσεις Αξιολόγησης μοντέλων (Evaluation Metrics)	58
Παρακολούθηση αντικειμένου.....	68
Boosting tracker	68
MIL tracker	69
KCF tracker.....	69
TLD tracker	69
Medianflow tracker	69
MOSSE tracker.....	69
CSRT tracker	69
Πειραματικό μέρος.....	70

Υλικό συστήματος	71
Περιβάλλον διασύνδεσης χρήστη	72
Εκπαίδευση Μοντέλου	75
Google Colab	76
Δοκιμή μοντέλου	83
Επίλογος	88
Επίπεδα του Yolo	89
Βιβλιογραφία – πηγές	93

OpenCV

Η OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη που συγκεντρώνει εκατοντάδες αλγόριθμους μηχανικής όρασης. Αρχικά αναπτύχθηκε από την intel (1999) και παραχωρήθηκε το 2012 στην μη κερδοσκοπική εταιρεία OpenCV.org . Ολοκληρώνεται σε πολλές γλώσσες προγραμματισμού και διακρίνεται για την ευκολία χρήσης της αλλά και την υποστήριξη που προσφέρει σε ειδικό hardware. Σε αυτή την εργασία χρησιμοποιήθηκε OpenCV με Python σε περιβάλλον Windows (PC) και Linux (Raspberry).

Οι εικόνες και κατ' επέκταση κάθε στιγμιότυπο (frame) βίντεο αντιμετωπίζονται ως πίνακες δεδομένων. Παρακάτω θα αναφερθούν κάποιες πολύ βασικές έννοιες που μας βοηθούν να καταλάβουμε τον τρόπο επεξεργασίας της εικόνας σε έναν υπολογιστή.

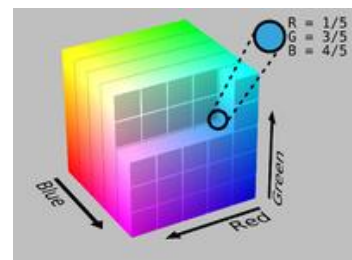
Χρωματόχωρος – Color space

Ως χρωματόχωρος νοείτε ένα εργαλείο που μας επιτρέπει να ξεχωρίζουμε και αποτυπώνουμε τα χρώματα. . Ο τρόπος απόδοσης ενός χρώματος ποικίλει. Υπάρχουν μέθοδοι για χρήση από ηλεκτρονικά κυκλώματα , αλλά και άλλοι που επικεντρώνονται στην αντίληψη του ανθρώπου. Στην πρώτη κατηγορία ανήκουν οι χώροι RGB,CMY,YUV ενώ στη δεύτερη οι HSV,HLS,Lab,Luv κτλ.

Αλγοριθμικά υπάρχει μεγάλο ενδιαφέρον να καταλάβουμε πως λειτουργεί το καθένα , καθώς μια συνήθης λειτουργία στην επεξεργασία εικόνας είναι ο κατακερματισμός μιας εικόνας σε περιοχές ενδιαφέροντος , ομαδοποιώντας κάποια τους χαρακτηριστικά σε επίπεδο pixel ή block από pixel.

RGB

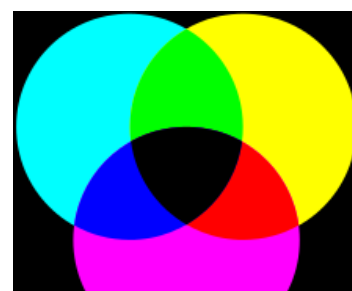
Στο χρωματόχωρο RGB ,το σύνολο των αποχρώσεων μπορεί να αναπαρασταθεί ως ένα σετ τριών τιμών , μια τιμή για κάθε βασικό χρώμα. Αν και είναι το πλέον διαδεδομένο σύστημα στα ψηφιακά ηλεκτρονικά, δεν εξυπηρετεί την επεξεργασία εικόνας.



Εικόνα 1 Χρωματόχωρος RGB

CMY

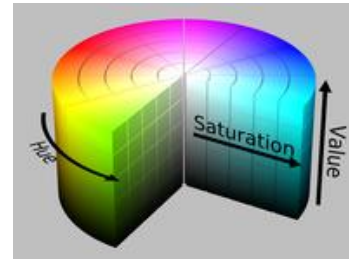
Σε αυτό το χρωματόχωρο (Cyan, Magenta, Yellow) το κάθε χρώμα είναι το αντίθετο των βασικών, δηλαδή το κυανό απορροφά το κόκκινο (άρα δεν το εκπέμπει), η ματζέντα το αντίθετο του πράσινου και το κίτρινο του μπλε. Έχει εφαρμογή κυρίως σε τυπογραφεία αφού εξυπηρετεί τις μεθόδους της.



Εικόνα 2 Χρωματόχωρος CMY

HSV

Αυτός ο χρωματοχώρος (Hue,Saturation,Value) έχει καλή εφαρμογή στην ανθρώπινη αντίληψη. Με τον όρο hue νοούμε το χρωματικό τόνο (πχ κόκκινο), με το saturation τη πυκνότητα (πχ φωτεινό απαλό) και με το value ή lightness τη ποσότητα φωτός (σκούρο ,ανοικτό). Μπορεί να αποδοθεί στο τρισδιάστατο επίπεδο στη παρακάτω εικόνα.



Εικόνα 3 Χρωματοχώρος HSV

YUV

Σε αυτό το χρωματοχώρο ,όπου χρησιμοποιείται κατά κόρον στη ψηφιακή τηλεόραση, το Y αναπαριστά τη φωτεινότητα ενώ τα UV χρωματική πληροφορία. Υπάρχει μέθοδος μετατροπής από RGB σε YUV .

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Βαθμιαίο διάνυσμα κλίσης εικόνας - Image Gradient Vector

Στην επεξεργασία εικόνας μας ενδιαφέρει η κατεύθυνση που αλλάζουν τα χρώματα . Σε κάθε πίξελ μπορούμε να βρούμε την «κλίση» του ,δηλαδή προς τα που αλλάζει περισσότερο. Ορίζουμε ένα διάνυσμα για κάθε πίξελ που μας δείχνει τόσο την κλίση όσο και την ένταση της αλλαγής.

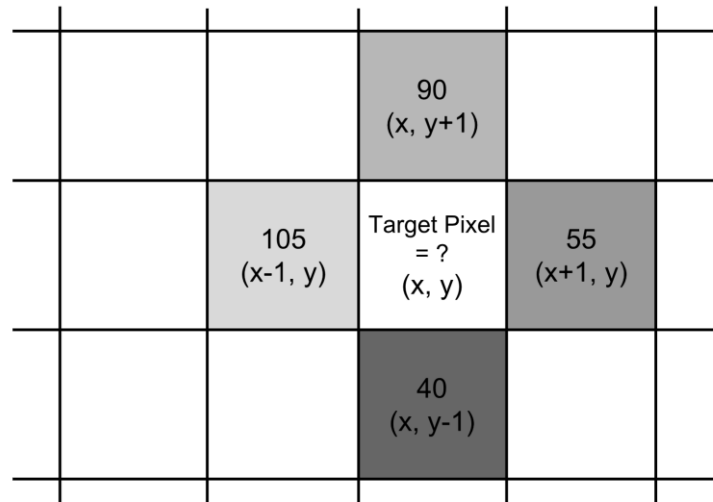
Έστω $f(x,y)$ η πληροφορία του χρώματος του πίξελ στη θέση x,y . Ορίζουμε το βαθμιαίο διάνυσμα ως

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1,y) - f(x-1,y) \\ f(x,y+1) - f(x,y-1) \end{bmatrix}$$

Το μέτρο του διανύσματος θα είναι $g = \sqrt{g_x^2 + g_y^2}$

Και η κατεύθυνση θα ορίζεται από τη γωνία $\theta = \arctan \frac{g_x}{g_y}$

Για παράδειγμα το βαθμιαίο διάνυσμα του παρακάτω πίξελ



Εικόνα 4 Image Gradient Vector

Θα είναι

$$\nabla f = \begin{bmatrix} f(x+1, y) - f(x-1, y) \\ f(x, y+1) - f(x, y-1) \end{bmatrix} = \begin{bmatrix} 55 - 105 \\ 90 - 40 \end{bmatrix} = \begin{bmatrix} -50 \\ 50 \end{bmatrix}$$

Με μέτρο $\sqrt{50^2 + (-50)^2} = 70.71$ και γωνία $\arctan\left(\frac{-50}{50}\right) = -45^\circ$

Διαφορετικά, μπορούμε να χρησιμοποιήσουμε οπτικό φίλτρο και να σαρώσουμε όλη την εικόνα ώστε να βρούμε όλα τα διανύσματα. Αυτό γίνεται με το φίλτρο συνέλιξης $k_y = [1, 0, -1]^T$ και $k_x = [-1, 0, 1]$ όπως θα δούμε παρακάτω. Αυτό θα είχε ως αποτέλεσμα

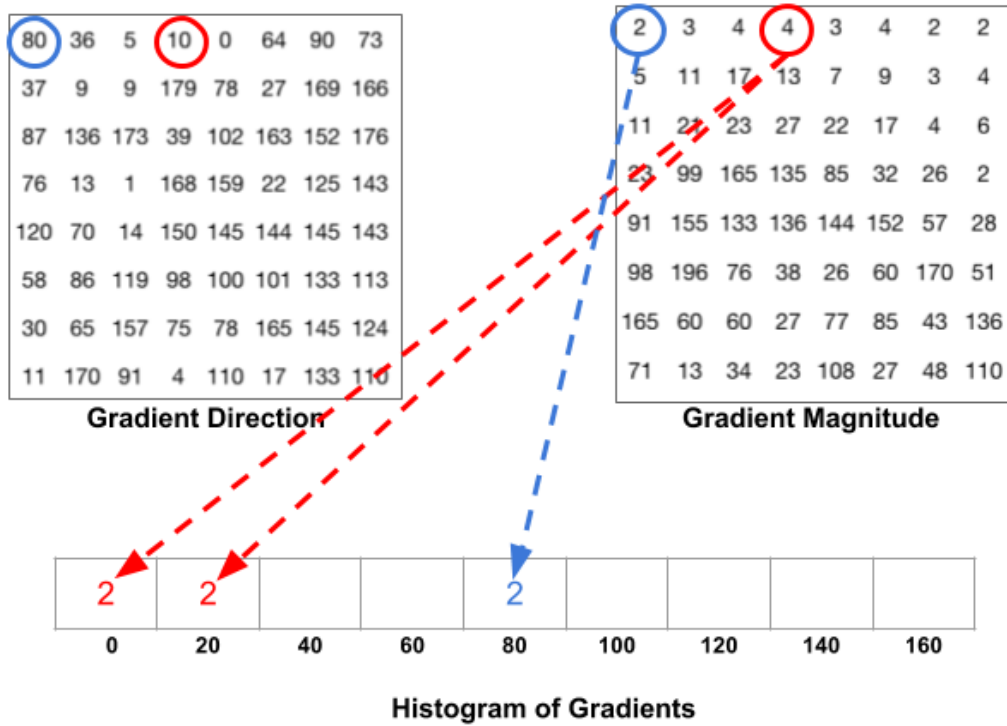
$$G_x = [-1, 0, 1] * [105, 255, 55] = -105 + 0 + 55 = -50$$

Και

$$G_y = [1, 0, -1] * \begin{bmatrix} 90 \\ 255 \\ 40 \end{bmatrix} = 90 + 0 - 40 = 50$$

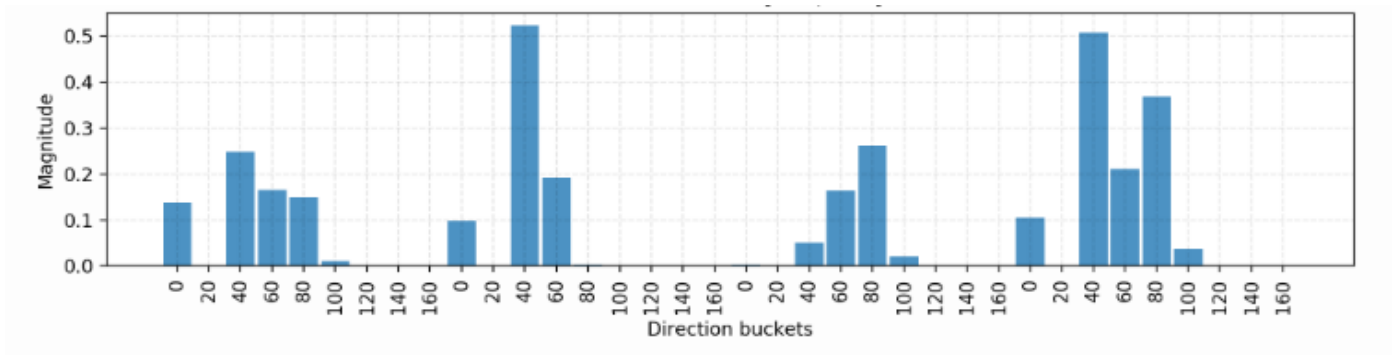
HOG (histogram of oriented gradients)

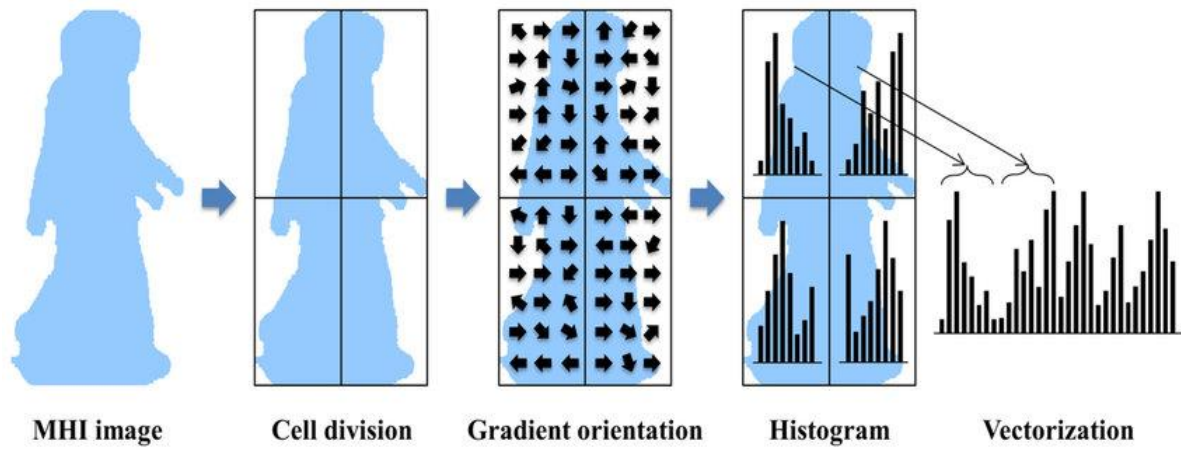
Πρόκειται για μέθοδο εξαγωγής χαρακτηριστικών από μια εικόνα βασισμένη στο βαθμιαίο διάνυσμα του κάθε πίξελ. Αφού υπολογιστούν όλα τα διανύσματα και τα μεγέθη τους, η εικόνα διαιρείται σε μπλοκ από πίξελ 8x8. Κατόπιν δημιουργούμε 9 μετρητές, ένα για κάθε υποδιαίρεση των 180° (0, 20, 40, ..., 160). Οι μετρητές αυξάνουν τιμή ανάλογα με τα μέτρα των διανυσμάτων που έχουν την προκαθορισμένη γωνία τους. Διανύσματα με γωνία ενδιάμεση των προκαθορισμένων τιμών, μοιράζουν το μέτρο τους στους κοντινούς τους μετρητές.



Εικόνα 5 HOG από IGV

Στο τελευταίο βήμα ενώνονται τα ιστογράμματα ανά 4 block δημιουργώντας έτσι ένα μονοδιάστατο πίνακα 36 στοιχείων. Η αλληλουχία αυτών των πινάκων αποτελεί το τελικό HOG διάνυσμα χαρακτηριστικών της εικόνας . Θα έχει δηλαδή τη παρακάτω μορφή .



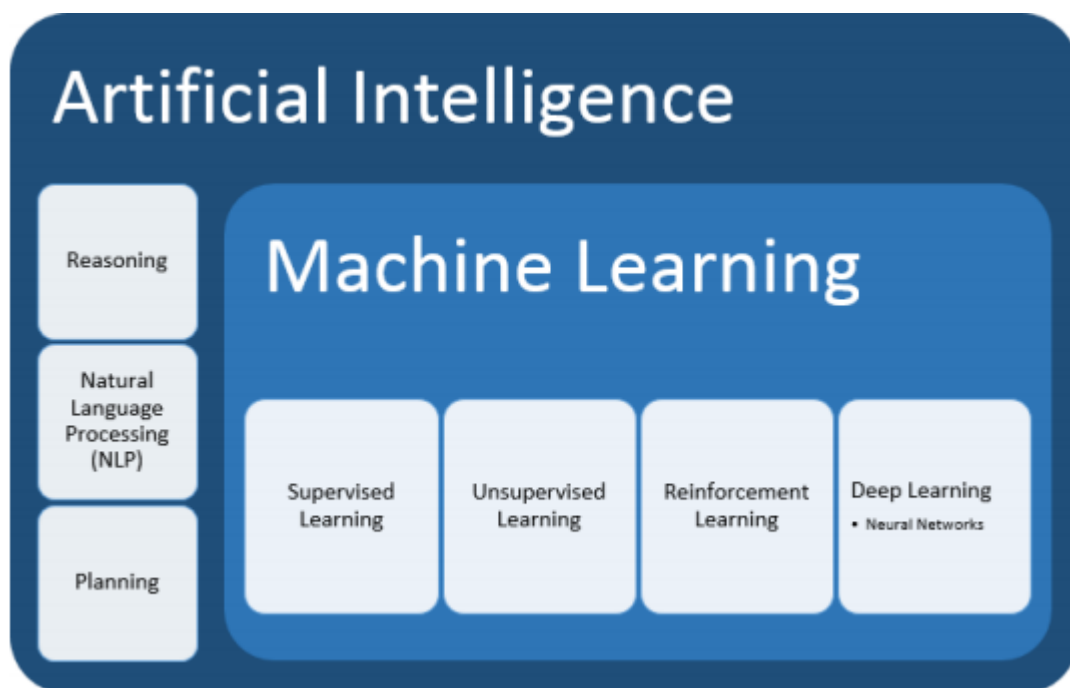


Εικόνα 6

HOG γραφική απεικόνιση

Τεχνητή νοημοσύνη

Τεχνητή ή μηχανική νοημοσύνη, είναι η νοημοσύνη που μπορεί να φέρει μια μηχανή, σε αντίθεση με την φυσική νοημοσύνη των έμβιων όντων. Ανάλογα με την επιστήμη που την μελετά, δίνονται διαφορετικοί ορισμοί και ακολουθούνται διαφορετικές προσεγγίσεις. Εδώ πάντως θα ακολουθήσουμε την επιστήμη των υπολογιστών, η οποία εξάλλου είναι και αυτή που την αναπτύσσει σε πρακτικό επίπεδο. Βάση αυτής της προσέγγισης ακολουθεί ο ορισμός. «Η Τεχνητή Νοημοσύνη είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζουμε με τη νοημοσύνη στην ανθρώπινη συμπεριφορά» (Barr και Feigenbaum)



Εικόνα 7

Πλαίσιο τεχνητής νοημοσύνης

Θα μπορούσαμε να χωρίσουμε την τεχνητή νοημοσύνη στους παρακάτω άξονες. Μάθηση, αιτιολόγηση – reasoning, επίλυση προβλημάτων - planning, αντίληψη, γλωσσική επεξεργασία.

Υπάρχουν πολύ τρόποι μάθησης μιας μηχανής ώστε να αντιδρά σωστά. Από απλές προγραμματιστικές μεθόδους απομνημόνευσης αντιδράσεων, ως έξυπνες μεθόδους γενίκευσης όπου ο αλγόριθμος έχει τη δυνατότητα να εξάγει συμπεράσματα για δεδομένα που δεν έχει ξανασυναντήσει, έχοντας προηγουμένως αποκτήσει εμπειρία σε συναφή προβλήματα.

Αιτιολόγηση είναι να συναχθούν συμπεράσματα κατάλληλα για την κατάσταση. Τα συμπεράσματα ταξινομούνται ως αφαιρετικά ή επαγωγικά. Τα αφαιρετικά συμπεράσματα είναι πιο ισχυρά ενώ τα επαγωγικά εξάγονται με παραδοχές. Επιτρέπει τη δημιουργία

συμπερασμάτων βασισμένα στα δεδομένα εισόδου. Συνδέει τα δεδομένα συμπληρώνοντάς τα κενά στη ροή σκέψης.

Για την **επίλυση προβλημάτων** η γενική μέθοδος είναι η σταδιακή μείωση της απόστασης μεταξύ της παρούσας και της επιθυμητής κατάστασης . Αυτό προϋποθέτει ικανότητα ενός συστήματος να δρα αυτόνομα . Δεν εννοούμε να ακολουθεί μια συγκεκριμένη διαδικασία αλλά να μπορεί να επιλέξει τα βήματα που πρέπει να ακολουθήσει για να πετύχει το στόχο του.

Η **αντίληψη** και επίγνωση του περιβάλλοντος δράσης είναι πολύ σημαντικός παράγοντας για την επίλυση οποιουδήποτε προβλήματος καθώς το περιβάλλον αποτελεί μέρος των δεδομένων εισόδου. Η δυσκολία έγκειται στο ότι το περιβάλλον θα πρέπει να ψηφιοποιηθεί και να αντιστοιχηθεί σε γνωστές καταστάσεις. Για παράδειγμα η αναγνώριση ενός ανθρώπου από έναν αλγόριθμο αυτόνομης οδήγησης δεν θα πρέπει να επηρεάζεται από το φωτισμό το προσανατολισμό η το μέγεθος του.

Ένας ξεχωριστός κλάδος της τεχνητής νοημοσύνης είναι η **αναγνώριση και η σύνθεση ομιλίας** . Η φύση της ανθρώπινης ομιλίας και του γραπτού λόγου, οι αμέτρητοι τρόποι και παραλλαγές να εκφράσει κάποιος κάτι, περιέχει πολλές δυσκολίες στο να γίνει κατανοητή από μια μηχανή. Για τον λόγο αυτό έχουν δημιουργηθεί ειδικοί αλγόριθμοι για την επίλυση τέτοιων προβλημάτων .

Τις βάσεις για την ανάπτυξη της τεχνητής νοημοσύνης έθεσε το 1956 μια ομάδα εργασίας στο κολέγιο Dartmouth στις ΗΠΑ. 17 χρόνια αργότερα ωστόσο διακόπηκε η χρηματοδότηση του προγράμματος και αμφισβητήθηκε η βιωσιμότητα του όλου εγχειρήματος. Σε εκείνο το σημείο ξεκινάει ο «χειμώνας» στην τεχνητή νοημοσύνη καθώς οι προσπάθειες που γίνονταν κατά διαστήματα για επανεκκίνηση στην έρευνα προσέκρουε στο υψηλό κόστος και την αδυναμία των υπολογιστικών συστημάτων να υποστηρίξουν τους απαιτούμενους αλγόριθμους. Ορόσημο αποτελεί η δημιουργία του υπολογιστικού συστήματος Deep Blue το 1997 από την IBM , το οποίο κατόρθωσε να νικήσει τον παγκόσμιο πρωταθλητή σκάκι Kasparov . Έκτοτε υπήρξε αναθέρμανση του ενδιαφέροντος ενώ τα τελευταία χρόνια η εξέλιξη της έχει φέρει θεαματικά αποτελέσματα. Σαν έννοια πάντως, η τεχνητή νοημοσύνη απασχόλησε τους ανθρώπους για χιλιάδες χρόνια και θα μπορούσαμε να εντάξουμε σε αυτή την αναζήτηση τα αρχαία δημιουργήματα Τάλω και Πανδώρα.

Μηχανική μάθηση – machine learning

Με τον όρο machine learning, αναφερόμαστε στην επιστήμη και τις τεχνικές της ,που διερευνά τη μελέτη και κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν σχετικές προβλέψεις. Βρίσκει εφαρμογή σε εργασίες αναγνώρισης spam mail, computer vision κλπ. όπου η δημιουργία αλγορίθμων από τον προγραμματιστή θα ήταν ανεπαρκής για να καλύψουν την έκταση των δεδομένων εισόδου.

Ένα μοντέλο μηχανικής μάθησης είναι η έξοδος που παράγεται όταν εκπαιδεύετε τον αλγόριθμο μηχανικής μάθησης με δεδομένα. Μετά την εκπαίδευση, όταν παρέχετε σε ένα μοντέλο με μια είσοδο, θα σας δοθεί μια έξοδος. Για παράδειγμα, ένας αλγόριθμος πρόβλεψης θα δημιουργήσει ένα πρότυπο πρόβλεψης. Αν τροφοδοτήσουμε το μοντέλο πρόβλεψης με δεδομένα, θα λάβουμε μια πρόβλεψη με βάση τα δεδομένα που εκπαίδευσαν το μοντέλο. Η μηχανική μάθηση είναι πλέον απαραίτητη για τη δημιουργία μοντέλων ανάλυσης.

Η εκμάθηση μηχανών επιτρέπει στα μοντέλα να εκπαιδεύονται σε σύνολα δεδομένων προτού αυτά αναπτυχθούν πλήρως. Ορισμένα μοντέλα μηχανικής μάθησης (on line) προσαρμόζονται συνεχώς καθώς εισάγονται νέα δεδομένα. Από την άλλη, άλλα μοντέλα (offline) προέρχονται από αλγόριθμους μηχανικής μάθησης που αφού εκπαιδευτούν δεν αλλάζουν. Αυτή η επαναληπτική διαδικασία των on line μοντέλων οδηγεί σε βελτίωση στους τύπους συσχετίσεων που γίνονται μεταξύ των δεδομένων.

Τρεις είναι οι κύριες κατηγορίες machine learning ,συναρτήσε των δεδομένων εισόδου.

Εποπτευόμενη μάθηση - supervised learning

Η εποπτευόμενη μάθηση συνήθως γίνεται στο πλαίσιο της ταξινόμησης , όταν θέλουμε να αντιστοιχίσουμε τα αντικείμενα εισόδου σε μια ετικέτα , ή της παρεμβολής (regression) για συνεχή τιμές εξόδου. Και στην ταξινόμηση και στην παρεμβολή ,ο στόχος είναι να βρεθεί στα αντικείμενα εισόδου μια σχέση ή δομή που θα μας επιτρέψει να τα αναγνωρίσουμε. Η δομή που αναζητούμε έχει δημιουργηθεί από τα δεδομένα εισόδου.

Κατά την εποπτευόμενη μάθηση θα πρέπει να λαμβάνονται υπόψη η πολυπλοκότητα του μοντέλου που θα δημιουργηθεί αλλά και η σχέση bias-variance σφάλμα-διακύμανση. Η αύξηση της πολυπλοκότητας από την μια μας δημιουργεί ένα πολύ πιστό μοντέλο του τι ζητάμε, αλλά ενέχει τον κίνδυνο του overfit (υπερπροσαρμογή) κυρίως αν τα δείγματα μας δεν είναι πολλά, δηλαδή δεν μπορεί να χρησιμοποιηθεί γενικότερα για πραγματικά αντικείμενα που διαφέρουν από τα δείγματα μας. Η σχέση bias-variance παρομοίως υποδηλώνει τη διακύμανση του σφάλματος της πρόβλεψης(ταξινόμησης), και είναι αντιστρόφως ανάλογες έννοιες συνήθως. Δηλαδή θα μπορούσε ένα μοντέλο να έχει σφάλμα (bias) 20% με διακύμανση 10%, οπότε συνολικά 10-30% ή σε μια άλλη υλοποίηση να έχει $25\% \pm 2\%$,δηλαδή 23-27% .

Μη εποπτευόμενη μάθηση - unsupervised learning

Χρησιμοποιείται κυρίως για την ομαδοποίηση αντικειμένων. Χωρίς τη χρήση ετικετών , ανακαλύπτει κοινά χαρακτηριστικά ανάμεσα σε αντικείμενα και μπορεί να χρησιμοποιηθεί για την ελαχιστοποίηση των κρίσιμων χαρακτηριστικών τους ή την κατηγοριοποίηση τους.

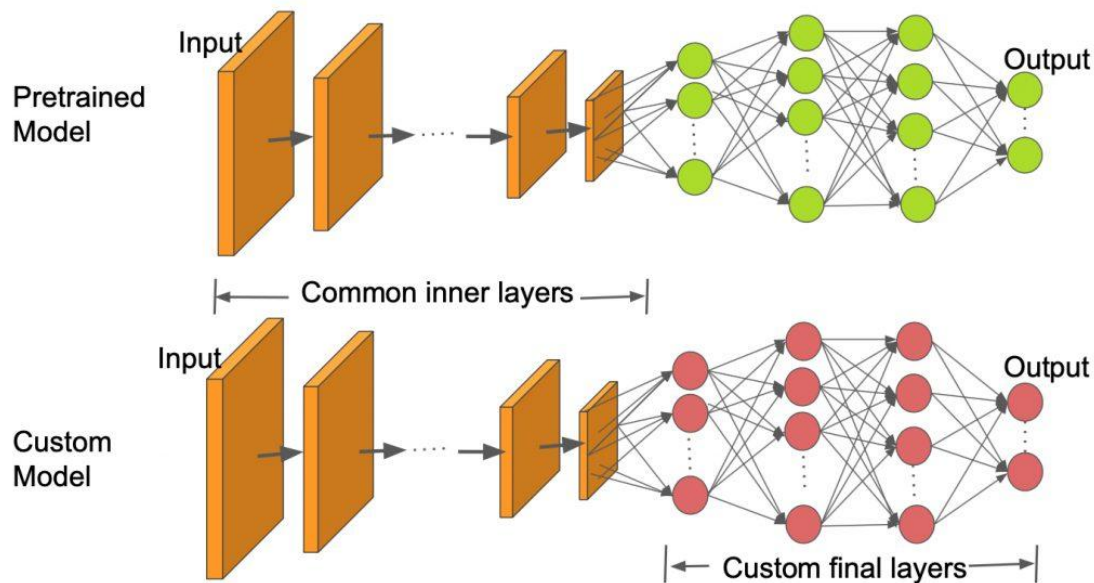
Είναι δηλαδή ένα εργαλείο που μπορεί να χρησιμοποιηθεί από αναλυτές δεδομένων ,και όχι για να μας παράσχει τελικά αποτελέσματα.

Ενισχυτική μάθηση - reinforcement learning

Η χρήση του δεν προϋποθέτει δεδομένα εκπαίδευσης. Με αυτή τη μέθοδο το μοντέλο δημιουργείται και μαθαίνει μέσα από τα λάθη του. Χρησιμοποιείται κατά κόρον σε ρομποτικά συστήματα και συστήματα αυτόνομης οδήγησης . Το σύστημα υιοθετεί μια συμπεριφορά και εάν αυτή δεν οδηγήσει στο επιθυμητό αποτέλεσμα το σύστημα την τροποποιεί μέχρι να επιτύχει το στόχο του. Η Ενισχυτική μάθηση ασχολείται με το πώς ένα υποκείμενο θα πρέπει να δράσει σε ένα περιβάλλον , έτσι ώστε να μεγιστοποιηθεί κάποια έννοια μακροπρόθεσμης ανταμοιβής. Οι αλγόριθμοι ενισχυτικής μάθησης προσπαθούν να βρουν μια πολιτική που αντιστοιχίζει τις καταστάσεις του περιβάλλοντος με τις ενέργειες που ο αλγόριθμος θα πρέπει να επιτελέσει σε αυτές τις καταστάσεις. Η ενισχυτική μάθηση διαφέρει από τα προβλήματα επιτηρούμενης μάθησης αφού τα σωστά ζεύγη δεδομένων εισόδου/εξόδου ζεύγη όπως αναφέραμε δεν παρουσιάστηκαν ποτέ, ούτε οι βέλτιστες δυνατές ενέργειες έχουν ρητά διορθωθεί.

Μάθηση δια της μεταφοράς – transfer learning

Η εκμάθηση μεταφοράς είναι μια μέθοδος εκμάθησης μηχανής όπου ένα μοντέλο που αναπτύχθηκε για μια εργασία επαναχρησιμοποιείται ως αφετηρία για ένα μοντέλο σε μια δεύτερη εργασία. Είναι μια δημοφιλής προσέγγιση στη βαθιά εκμάθηση, όπου τα προ-εκπαιδευμένα μοντέλα χρησιμοποιούνται ως σημείο εκκίνησης για εφαρμογές μηχανικής όρασης και επεξεργασίας φυσικής γλώσσας, δεδομένου του μεγάλου αριθμού υπολογιστών και των πόρων που απαιτούνται για την ανάπτυξη νέων μοντέλων νευρωνικών δικτύων για αυτά τα προβλήματα . Σε αυτή τη διαδικασία χρησιμοποιούμε ένα μοντέλο προ-εκπαιδευμένο σε ένα μεγάλο όγκο δεδομένων που δεν θα μπορούσαμε να υποστηρίξουμε εμείς, χρησιμοποιούμε από αυτό τα αρχικά layers για την εξαγωγή των βασικών χαρακτηριστικών αλλά απορρίπτουμε τις συνάψεις των τελικών επιπέδων ώστε να ξαναπάρουν καινούργιες τιμές οι ενεργοποιητές και τα βάρη από την δική μας εκπαίδευση στα δικά μας αντικείμενα. Κατά την εκπαίδευση του μοντέλου με τις δικές μας εικόνες μόνο στα νέα επίπεδα γίνονται οι υπολογισμοί με τις forward και backward συναρτήσεις.



Εικόνα 8 Transfer learning

Ένα απλό παράδειγμα είναι οι συλλογές εικόνων (datasets) . Έχουν δημιουργηθεί συλλογές με χιλιάδες αντικείμενα ομαδοποιημένα , έτοιμες να χρησιμοποιηθούν από οποιοδήποτε δίκτυο για την εκπαίδευση του (Image datasets). Για παράδειγμα η συλλογή CalTech256 περιέχει 30607 εικόνες κατηγοριοποιημένες σε 256 κατηγορίες αντικειμένων . Περιέχει από ζώα μέχρι μουσικά όργανα και έχει μέγεθος 1.2 gb . Η επιλογή μιας αξιόπιστης συλλογής είναι πολύ σημαντική για την εκπαίδευση ενός δικτύου . Θα πρέπει ιδανικά να είναι μικρή σε μέγεθος, να περιέχει ικανό αλλά και αντιπροσωπευτικό όγκο δεδομένων ώστε να μην υπάρξει overfit, με αποτέλεσμα το ίδιο δίκτυο να παρουσιάζει διακυμάνσεις στην επιτυχία του αναλόγως της συλλογής που χρησιμοποίησε. Μπορούμε να χρησιμοποιήσουμε τα βάρη ενός μοντέλου που έχει ήδη δημιουργηθεί από αυτό το dataset σε ένα συγκεκριμένο δίκτυο, και να το χρησιμοποιήσουμε ως βάση για το δικό μας που θέλουμε να εκπαιδεύσουμε με το ίδιο δίκτυο αλλά διαφορετικά αντικείμενα.

Αλγόριθμοι λήψης αποφάσεων

Οι κύριες προσεγγίσεις , αναλόγως την κατηγορία δεδομένων είναι οι παρακάτω.

Για μάθηση με ή χωρίς επίβλεψη

Ο τρόπος χρήσης κάθε αλγορίθμου μπορεί να διαφέρει. Κάποιοι εξυπηρετούν τη μάθηση χωρίς επίβλεψη και κάποιοι με επίβλεψη. Ωστόσο η σύνθετη δομή κάποιων μας επιτρέπει να τους κατατάξουμε και στους δύο τρόπους. Ακόμα και όσοι κατατάσσονται σε μια

κατηγορία δε σημαίνει ότι δεν θα μπορούσαν να εφαρμοστούν και στην άλλη, ωστόσο μια γενική ταξινόμηση μπορεί να γίνει ως εξής.

Βαθιά μάθηση – νευρωνικά δίκτυα

(Deep learning - neural networks)

Είναι ο τρόπος εκμάθησης που βασίζεται σε πολλαπλά κρυφά επίπεδα (layers) μεταξύ εισόδου δεδομένων και εξόδου, μέσα σε ένα νευρωνικό δίκτυο. Με τον όρο νευρωνικό δίκτυο αποδίδουμε τη προσπάθεια να μιμηθεί τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου, με έως εκατομμύρια συνάψεις (μικρές διεργασίες) που αποσκοπούν στην υπερκέρωση των όποιων απρόσμενων δυσκολιών δεν είχαν αναγνωριστεί προηγουμένως και τη επαναληπτική ανατροφοδότηση των δεδομένων μεταξύ των επιπέδων. Η λειτουργία του βασίζεται σε συνδυασμούς εποπτευόμενων και μη, αλγορίθμων. Τα σύγχρονα νευρωνικά δίκτυα είναι εργαλεία μη γραμμικής στατιστικής μοντελοποίησης δεδομένων. Συνήθως χρησιμοποιούνται για τη μοντελοποίηση σύνθετων σχέσεων μεταξύ δεδομένων εισόδου και εξόδου, για την ανακάλυψη προτύπων στα δεδομένα, ή για τον εντοπισμό στατιστικής δομής σε μία άγνωστη κοινή κατανομή πιθανότητας μεταξύ των παρατηρούμενων μεταβλητών.

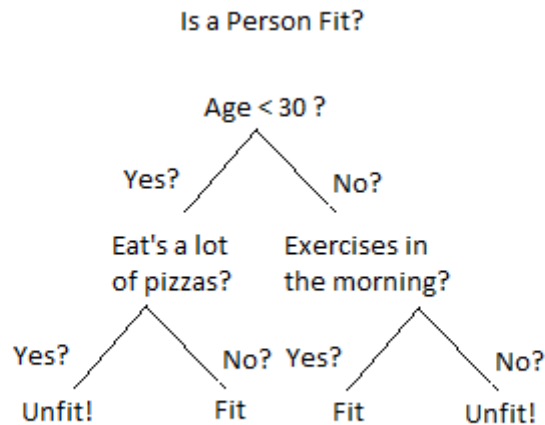
Η πτώση των τιμών του υλικού των τελευταίων ετών καθώς και η ανάπτυξη των GPU για προσωπική χρήση, οδήγησε στην ανάπτυξη της ιδέας της Βαθιάς Μάθησης. Αυτή η προσέγγιση προσπαθεί να μοντελοποιήσει τον τρόπο που ο ανθρώπινος εγκέφαλος επεξεργάζεται το φως και τον ήχο και τα μετατρέπει σε όραση και ακοή. Ορισμένες επιτυχείς εφαρμογές της Βαθιάς μάθησης είναι η μηχανική όραση και η αναγνώριση ομιλίας. Όπως και τα νευρωνικά δίκτυα που είναι κατηγορία της Βαθιάς μάθησης, κατατάσσονται και στις δύο κατηγορίες.

Για εποπτευόμενη μάθηση

Εκμάθηση με δέντρο απόφασης

(Decision tree)

Η εκμάθηση με δέντρο απόφασης χρησιμοποιεί ένα δέντρο απόφασης ως προγνωστικό μοντέλο, το οποίο αντιστοιχίζει παρατηρήσεις σχετικά με ένα στοιχείο σε συμπεράσματα σχετικά με την τιμή στόχο του αντικειμένου. Οι αλγόριθμοι των δένδρων αποφάσεων χρησιμοποιούν μια δομή διακλάδωσης για την απεικόνιση του αποτελέσματος της απόφασης. Τα δέντρα απόφασης μπορούν να χρησιμοποιηθούν για να χαρτογραφηθούν τα πιθανά αποτελέσματα μιας απόφασης. Κάθε κόμβος ενός δέντρου αποφάσεων αντιπροσωπεύει ένα πιθανό αποτέλεσμα. Στους κόμβους αντιστοιχίζονται ποσοστά (ή βάρη) με βάση την πιθανότητα εμφάνισης του αποτελέσματος.



Εικόνα 9

Decision tree

Επαγωγικός λογικός προγραμματισμός

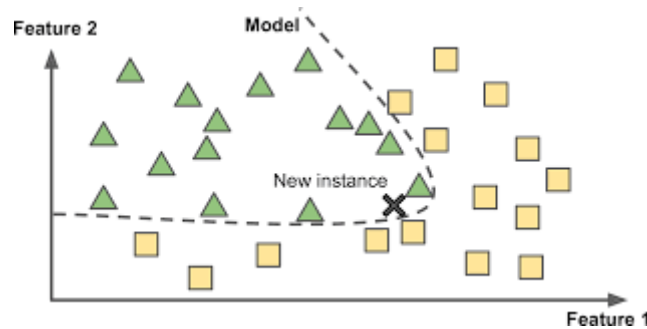
(inductive logic programming)

Ο Επαγωγικός λογικός προγραμματισμός (ILP) είναι μια προσέγγιση που διέπει την μάθηση και χρησιμοποιεί λογικό προγραμματισμό ως τρόπο παρουσίασης των παραδειγμάτων εισόδου, του γνωστικού υποβάθρου και των υποθέσεων. Δεδομένης μιας κωδικοποίησης του γνωστικού υποβάθρου και ενός συνόλου παραδειγμάτων που παρουσιάζονται σαν λογική βάση γεγονότων, το σύστημα ΕΛΠ παράγει το υποτιθέμενο λογικό πρόγραμμα που περιέχει όλα τα θετικά και κανένα αρνητικό παράδειγμα. Ο επαγωγικός προγραμματισμός είναι ένας σχετικός τομέας που λαμβάνει υπόψιν κάθε είδος προγραμματιστικής γλώσσας για την αναπαράσταση υποθέσεων (και όχι μόνο λογικό προγραμματισμό), όπως τα συναρτησιακά προγράμματα.

Εκμάθηση κατά περίπτωση

(Instance-based Learning)

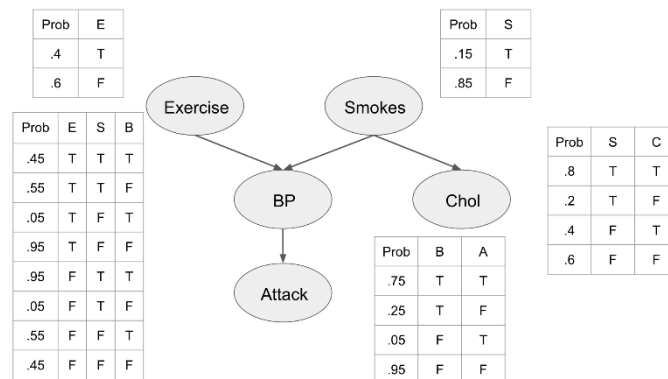
Αλγόριθμοι που βασίζονται σε δείκτες χρησιμοποιούνται όταν θέλουμε να κατηγοριοποιήσουμε νέα σημεία δεδομένων που βασίζονται σε ομοιότητες με δεδομένα κατάρτισης. Αυτά το σετ των αλγορίθμων αναφέρονται μερικές φορές lazy learning επειδή δεν υπάρχει φάση κατάρτισης. Αντ' αυτού, αλγόριθμοι με βάση τα στιγμιότυπα απλά ταιριάζουν νέα δεδομένα με δεδομένα εκπαίδευσης και κατηγοριοποιούν τα νέα σημεία δεδομένων που βασίζονται στην ομοιότητα με τα δεδομένα εκπαίδευσης. Η μάθηση βασισμένη σε instances δεν είναι κατάλληλη για τα σύνολα δεδομένων που έχουν τυχαία παραλλαγή, άσχετα δεδομένα ή δεδομένα με ελλείπουσες τιμές. Οι αλγόριθμοι με βάση τα στιγμιότυπα μπορούν να είναι πολύ χρήσιμοι στην αναγνώριση προτύπων. Για παράδειγμα, η εκμάθηση των περιπτώσεων χρησιμοποιείται σε χημικά και ανάλυση βιολογικών δομών και χωρική ανάλυση. Ανάλυση βιολογικών, φαρμακευτικών, χημικών και μηχανολογικών πεδίων συχνά χρησιμοποιεί διάφορους αλγόριθμους που βασίζονται σε στιγμιότυπα .



Εικόνα 10 Instance based learning

Δίκτυα Bayes

Ένα δίκτυο Bayes, ένα δίκτυο εμπιστοσύνης ή ένα άκυκλο γραφικό μοντέλο είναι ένα πιθανοθεωρητικό γραφικό μοντέλο που απεικονίζει ένα σύνολο τυχαίων μεταβλητών και την μεταξύ τους υποθετική ανεξαρτησία διαμέσου ενός κατευθυνόμενου άκυκλου γράφου. Για παράδειγμα, ένα δίκτυο Bayes μπορεί να αναπαραστήσει την πιθανοθεωρητική σχέση μεταξύ ασθενειών και συμπτωμάτων. Δεδομένων των συμπτωμάτων, το δίκτυο μπορεί να χρησιμοποιηθεί για να υπολογίσει τις πιθανότητες παρουσίας διαφόρων ασθενειών. Αυτοί οι αλγόριθμοι είναι ιδιαίτερα χρήσιμοι όταν δεν διαθέτουμε τεράστια ποσά δεδομένων για να εκπαιδευσουμε με σιγουριά.

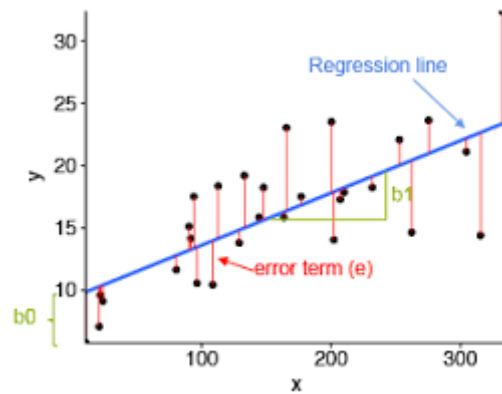


Εικόνα 11 Bayes network

Γραμμικής παλινδρόμησης

(Linear Regression)

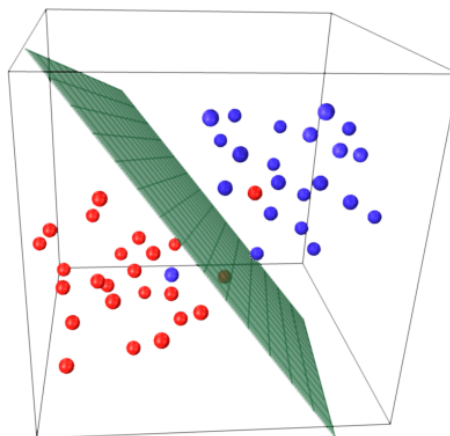
Αλγόριθμοι παλινδρόμησης χρησιμοποιούνται συνήθως για στατιστική ανάλυση και είναι βασικοί αλγόριθμοι χρήσης στη μηχανική μάθηση. Βοηθούν τους αναλυτές να διαμορφώνουν σχέσεις μεταξύ των σημείων δεδομένων. Οι αλγόριθμοι παλινδρόμησης μπορούν να ποσοτικοποιήσουν την ισχύ της συσχέτισης μεταξύ των μεταβλητών σε ένα σύνολο δεδομένων. Η ανάλυση με παλινδρόμηση μπορεί να είναι χρήσιμη για την πρόβλεψη των μελλοντικών τιμών των δεδομένων με βάση ιστορικές τιμές. Ωστόσο, είναι σημαντικό στον συσχετισμό να ισχύει αιτιώδης συνάφεια. Χωρίς την κατανόηση του πλαισίου γύρω από τα δεδομένα, η ανάλυση παλινδρόμησης μπορεί να οδηγήσει σε ανακριβείς προβλέψεις.



Εικόνα 12 Linear regression

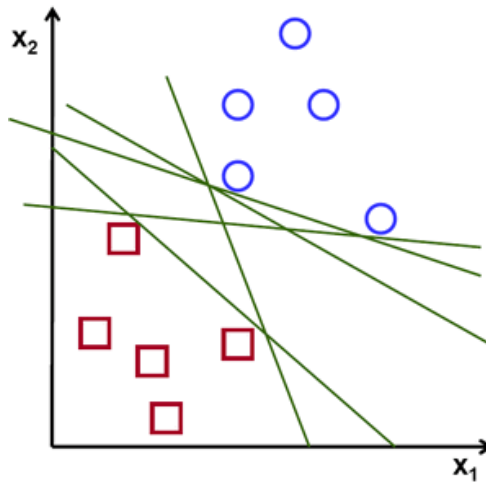
Μηχανές διανυσμάτων υποστήριξης (SVM)

Οι μηχανές διανυσμάτων υποστήριξης είναι ένα σύνολο μεθόδων επιτηρούμενης μάθησης που χρησιμοποιούνται για την ταξινόμηση και την παρεμβολή. Σ' αυτήν την περίπτωση δίνεται ένα σύνολο παραδειγμάτων εκπαίδευσης και κάθε φορά δηλώνεται σε ποια από τις δύο κατηγορίες ανήκει το παράδειγμα. Μία μηχανή διανυσμάτων υποστήριξης κατασκευάζει ένα μοντέλο που προβλέπει αν το νέο παράδειγμα εμπίπτει στην μία κατηγορία ή την άλλη. Λειτουργεί δημιουργώντας διαχωριστικά επίπεδα (hyperplanes) ανάμεσα σε συστάδες αντικειμένων, τα οποία βρίσκονται συνήθως στον ίδιο χώρο εφόσον ανήκουν στην ίδια κατηγορία. Τα αντικείμενα που χρησιμοποιούνται για την εκπαίδευση του αλγόριθμου είναι αντικείμενα που γνωρίζουμε την κατηγορία τους (εποπτευόμενη μάθηση). Αφού τελειώσει η εκπαίδευση και δημιουργηθούν τα διαχωριστικά επίπεδα, τα νέα αντικείμενα ταξινομούνται σε σχέση με τη θέση που καταλαμβάνουν στο χώρο.



Εικόνα 13 SVM 3d

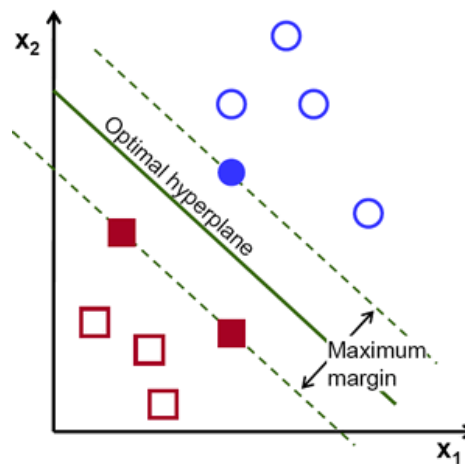
Λόγο της σπουδαιότητας αυτού του αλγορίθμου και τις ευρύτατης χρήσης του ας το δούμε παρακάτω πιο αναλυτικά με ένα απλό παράδειγμα σε χώρο δύο διαστάσεων.



Εικόνα 14 SVM 2d

Από τις δύο κατηγορίες αντικειμένων που φαίνονται παραπάνω μπορούν να οριστούν διάφορες διαχωριστικές γραμμές που θα μπορούσαν να χρησιμοποιηθούν για τον ορισμό του χώρου που καταλαμβάνει η κάθε κατηγορία. Ωστόσο ο αλγόριθμος θα πρέπει να διαλέξει μια, την πιο αποδοτική που θα κάνει τα λιγότερα λάθη στην ταξινόμηση νέων αντικειμένων.

Ο γενικός κανόνας είναι να μεγιστοποιήσουμε την ελάχιστη απόσταση των αντικειμένων με την διαχωριστική γραμμή.



Εικόνα 15 Optimal hyperplane

Τα διαχωριστικά επίπεδα ορίζονται γενικά από τη σχέση

$$f(x) = \beta_0 + \beta^T x,$$

όπου β_0 το bias και β διάνυσμα βάρους (weight vector).

Θεωρούμε ως ιδανικό αυτό που ορίζεται από $|\beta_0 + \beta^T x| = 1$ όπου το x συμβολίζει τα αντικείμενα εκπαίδευσης πλησιέστερα στο επίπεδο διαχωρισμού. Αυτά τα αντικείμενα ονομάζονται support vectors (διανύσματα υποστήριξης) και η απεικόνιση του ονομάζεται κανονικό επίπεδο υπερεπίπεδο (canonical hyperplane).

Βρίσκουμε την απόσταση μεταξύ των σημείων x και του διαχωριστικού επιπέδου

$$distance = \frac{|\beta_0 + \beta^T x|}{\|\beta\|}$$

ενώ για το κανονικό υπερεπίπεδο

$$distance(sv) = \frac{1}{\|\beta\|}$$

ενώ το περιθώριο Margin

$$M = \frac{2}{\|\beta\|}$$

Το πρόβλημα της μεγιστοποίησης του M ανάγεται σε πρόβλημα εύρεσης του ελάχιστου $L(\beta)$

$$\min L(\beta)_{\beta, \beta_0} = \frac{1}{2} \|\beta\|^2$$

για κάθε $y_i (\beta^T x_i + \beta_0) \geq 1$

όπου x_i τα αντικείμενα που χρησιμοποιήθηκαν για την εκπαίδευση και y_i κάθε κατηγορία τους.

Παράδειγμα χρήσης SVM σε περιβάλλον python με χρήση OpenCV

Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε 4 αντικείμενα εκπαίδευσης. Τα 3 θα ανήκουν στην ίδια κατηγορία (την -1) και το ένα στη κατηγορία 1.

```
import numpy as np
import cv2 as cv
image = np.zeros((400, 800, 3), np.uint8)
labels = np.array([1,-1,-1,-1])
trainingData = np.matrix([[501,10],[255,10],[501,255],[10,501]], dtype=np.float32)
```

```
svm = cv.ml.SVM_create()           'δημιουργία αλγορίθμου
svm.setType(cv.ml.SVM_C_SVC)      'τύπος μηχανής (μη γραμμικός)
svm.setKernel(cv.ml.SVM_LINEAR)   'αναδιάταξη δεδομένων (όχι)
svm.setTermCriteria((cv.TERM_CRITERIA_MAX_ITER, 100, 1e-6)) 'αριθμός
επαναλήψεων ή ανοχή σφάλματος (όποιο έρθει πρώτο)
```

και εκτελείτε η ρουτίνα εκπαίδευσης.

```
svm.train(trainingData, cv.ml.ROW_SAMPLE, labels)
```

Τώρα θα δημιουργήσουμε ένα βρόχο όπου θα οπτικοποιηθούν όλα τα πιθανά αντικείμενα στο επίπεδο μας. Στην περίπτωση μας είναι εύκολο καθώς τα αντικείμενα είναι τα pixel της εικόνας που θα δημιουργήσουμε και έχουν διακριτό αριθμό.

```

green = (0,255,0)           'BGR τιμές
blue = (255,0,0)
for i in range(image.shape[0]):      'για κάθε τιμή x του pixel(x,y)
    for j in range(image.shape[1]):  'για κάθε τιμή y του pixel(x,y)
        sampleMat = np.matrix([[j,i]], dtype=np.float32)
        response = svm.predict(sampleMat)[1] 'ταξινόμηση του κάθε σημείου
        if response == 1:
            image[i,j] = green       'αντιστοίχιση χρώματος
        elif response == -1:
            image[i,j] = blue

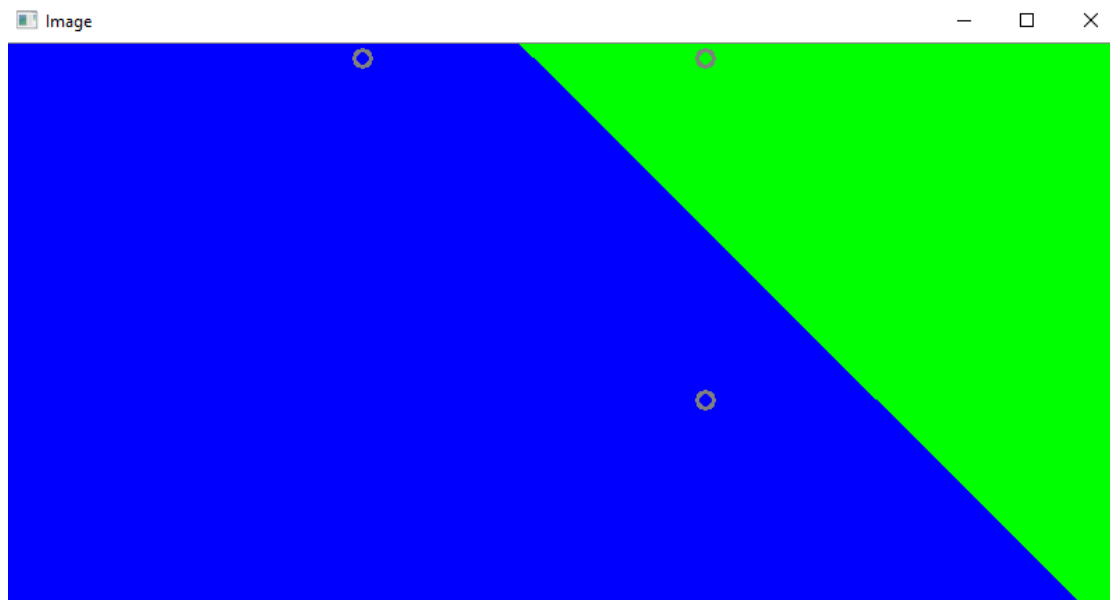
```

Και τέλος στιγματίζουμε τα support vectors

```

thickness = 2
radius = 6
sv = svm.getUncompressedSupportVectors()
for i in range(sv.shape[0]):
    cv.circle(image, (sv[i,0], sv[i,1]), radius, (128, 128, 128), thickness)
cv.imshow("Image", image)

```



Εικόνα 16

SVM - OpenCV

Για μη εποπτευόμενη μάθηση

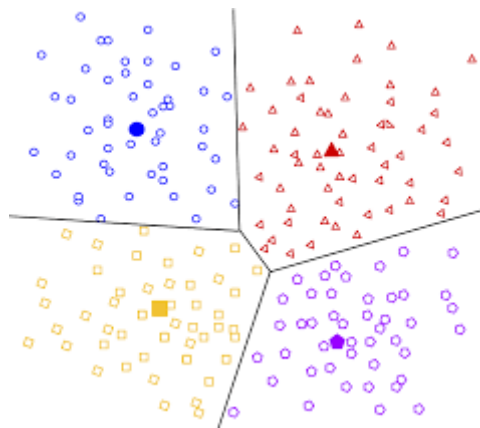
Εκμάθηση με μέτρο ομοιότητας ή κανόνες συσχέτισης

Σε αυτή την κατηγορία προβλημάτων δίνονται στην μηχανή μάθησης ζεύγη παραδειγμάτων που θεωρούνται όμοια και ζεύγη που θεωρούνται ανόμοια. Τότε η μηχανή μάθησης πρέπει να κατασκευάσει μια συνάρτηση ομοιότητας (ή μια συνάρτηση μετρικής απόστασης), που μπορεί να προβλέψει αν δύο καινούρια αντικείμενα είναι όμοια. Πρόκειται για μια τεχνική που χρησιμοποιείται σε συστήματα σύστασης.

Ομαδοποίηση

(Clustering)

Αντικείμενα με παρόμοιες παραμέτρους ομαδοποιούνται (σε μια ομάδα). Όλα τα αντικείμενα σε μια ομάδα είναι πιο όμοια μεταξύ τους από αντικείμενα σε άλλες ομάδες. Η ομαδοποίηση είναι ένας τύπος μη επιτηρούμενης μάθησης επειδή τα δεδομένα δεν έχουν επισημανθεί. Ο αλγόριθμος ερμηνεύει τις παραμέτρους που απαρτίζουν κάθε στοιχείο και στη συνέχεια τις ομαδοποιεί αναλόγως. Διαφορετικές τεχνικές κατηγοριοποίησης οδηγούν σε διαφορετικές υποθέσεις σχετικά με τη δομή των δεδομένων, οι οποίες συχνά καθορίζονται από κάποιο μέτρο ομοιότητας και αξιολογούνται για παράδειγμα ως προς την εσωτερική συνοχή (ομοιότητα μεταξύ των μελών του ίδιου cluster) και το διαχωρισμό ανάμεσα σε διαφορετικές ομάδες. Άλλες μέθοδοι βασίζονται στην εκτιμώμενη πυκνότητα και την συνεκτικότητα των γραφημάτων. Η ομαδοποίηση είναι μία μέθοδος μη επιτηρούμενης μάθησης και μία τεχνική η οποία χρησιμοποιείται επίσης στην στατιστική ανάλυση δεδομένων.



Εικόνα 17

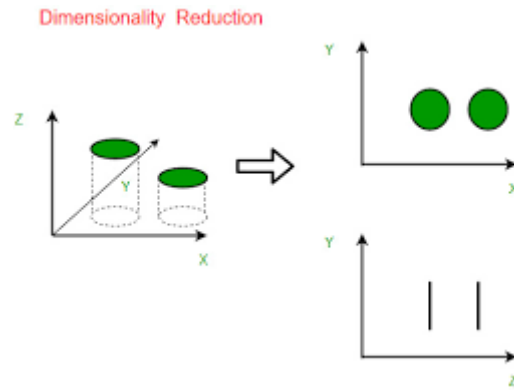
Clustering

Τεχνικές μείωσης διαστάσεων

(Dimensionality reduction)

Η μείωση των διαστάσεων βοηθά τα συστήματα να απομακρύνουν δεδομένα που δεν είναι χρήσιμα για ανάλυση. Η μείωση διαστάσεων μπορεί να είναι χρήσιμη κατά την ανάλυση δεδομένων από αισθητήρες και του Διαδίκτυο των πραγμάτων (IoT). Στα συστήματα IoT,

μπορεί να υπάρχουν χιλιάδες σημεία επαναλαμβανόμενων δεδομένων. Για παράδειγμα η κατάσταση ενός αισθητήρα (ενεργοποιημένος ή όχι). Η αποθήκευση και η ανάλυση αυτών των δεδομένων δεν είναι χρήσιμη και θα καταλάμβανε σημαντικό αποθηκευτικό χώρο. Αφαιρώντας αυτά τα πλεονάζοντα δεδομένα, οι επιδόσεις μιας μηχανής εκμάθησης μπορεί να βελτιωθεί, αλλά και η οπτικοποίηση των δεδομένων θα είναι πιο φιλική.



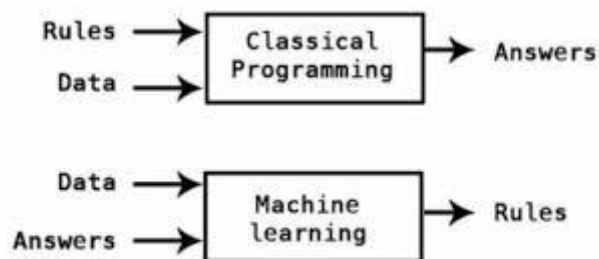
Εικόνα 18

Dimensionality reduction

Εκμάθηση μηχανών βασισμένη σε κανόνες συσχέτισης

(Rule-based machine learning)

Η εκμάθηση με κανόνες συσχέτισης είναι μια μέθοδος ανακάλυψης ενδιαφερουσών σχέσεων μεταξύ των μεταβλητών σε μεγάλες βάσεις δεδομένων. Σε γενικές γραμμές, τα συστήματα βασισμένα σε κανόνες είναι πολύ εύκολα κατανοητά: Εάν εισαχθούν δεδομένα X τότε πράξε Y . Ωστόσο, καθώς τα συστήματα αρχίζουν να λειτουργούν η προσέγγιση της μηχανικής μάθησης μπορεί να γίνει πολύ σύνθετη. Για παράδειγμα, ένα σύστημα μπορεί να περιλαμβάνει 100 προκαθορισμένους κανόνες. Όσο το σύστημα συναντά όλο και περισσότερα δεδομένα και εκπαιδεύεται, είναι πιθανό να προκύψουν εκατοντάδες εξαιρέσεις από τους κανόνες.



Εικόνα 19

Rule based ML

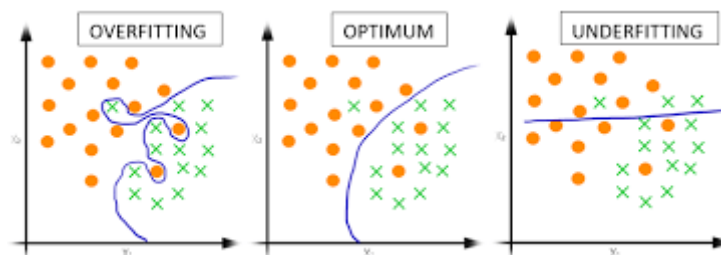
Άλλοι μέθοδοι

Γενετικοί αλγόριθμοι

Ένας γενετικός αλγόριθμος (GA) είναι μια ευρετική αναζήτηση που μιμείται τη διαδικασία της φυσικής επιλογής, και χρησιμοποιεί μεθόδους όπως αυτή της μετάλλαξης και της διασταύρωσης προκειμένου να δημιουργήσει καινούρια γονότυπα με την ελπίδα εύρεσης αποτελεσματικών λύσεων σε ένα συγκεκριμένο πρόβλημα. Στη μηχανική μάθηση, γενετικοί αλγόριθμοι χρησιμοποιήθηκαν τη δεκαετία του 1980 και του 1990. Αντίστροφα, τεχνικές μηχανικής μάθησης έχουν χρησιμοποιηθεί για την βελτίωση της απόδοσης γενετικών και εξελικτικών αλγορίθμων.

Ρυθμίσεις για να αποφευχθεί η υπερπροσαρμογή (overfit)

Η τακτοποίηση είναι μια τεχνική που τροποποιεί τα μοντέλα για να αποφευχθεί το πρόβλημα της υπερπροσαρμογής. Μπορούμε να εφαρμόσουμε τακτοποίηση σε οποιοδήποτε μηχανή εκμάθησης. Για παράδειγμα, μπορείτε να τροποποιήσετε ένα μοντέλο δέντρου απόφασης ώστε να απλοποιηθεί. Η υπερφόρτωση εμφανίζεται όταν δημιουργείται ένα μοντέλο για συγκεκριμένα δεδομένα αλλά δεν μπορεί να ανταποκριθεί σε λίγο διαφορετικά, να γίνει πιο γενικευμένο δηλαδή.



Εικόνα 20 Overfit

Deep Learning – Βαθιά εκμάθηση

Η βαθιά δομημένη μάθηση ή η ιεραρχική μάθηση ή η βαθιά εκμάθηση εν συντομία είναι μέρος της οικογένειας μεθόδων μηχανικής μάθησης, οι οποίες είναι ένα υποσύνολο του ευρύτερου πεδίου της Τεχνητής Νοημοσύνης.

Η βαθιά εκμάθηση είναι μια κλάση αλγορίθμων μηχανικής μάθησης που χρησιμοποιούν διάφορα στρώματα μη γραμμικών μονάδων επεξεργασίας για την εξαγωγή και τον μετασχηματισμό χαρακτηριστικών. Κάθε διαδοχική στρώση χρησιμοποιεί την έξοδο από το προηγούμενο επίπεδο ως είσοδο.

Τα πεδία νευρωνικών δικτύων, τα δίκτυα βαθιών πεποιθήσεων (deep belief networks) και τα επαναλαμβανόμενα νευρωνικά δίκτυα έχουν εφαρμοστεί σε πεδία όπως η μηχανική όραση, η αναγνώριση ομιλίας, η επεξεργασία της φυσικής γλώσσας, η αναγνώριση ήχου, το φιλτράρισμα κοινωνικών δικτύων, η μηχανική μετάφραση και η βιοπληροφορική, όπου παρήγαγαν αποτελέσματα συγκρίσιμα με και σε ορισμένες περιπτώσεις καλύτερα από τους ανθρώπινους εμπειρογνώμονες.

Αλγόριθμοι βαθιάς εκμάθησης και δίκτυα βασίζονται στην ανεξέλεγκτη εκμάθηση πολλαπλών επιπέδων χαρακτηριστικών ή αναπαραστάσεων των δεδομένων. Οι λειτουργίες υψηλότερου επιπέδου προέρχονται από χαρακτηριστικά κατώτερου επιπέδου για να σχηματίσουν μια ιεραρχική αναπαράσταση.

βαθιά εκμάθηση και παραδοσιακή μάθηση μηχανών

Μία από τις μεγαλύτερες προκλήσεις που αντιμετωπίζουν τα παραδοσιακά μοντέλα μηχανικής μάθησης είναι μια διαδικασία που ονομάζεται εξαγωγή χαρακτηριστικών. Ο προγραμματιστής πρέπει να είναι συγκεκριμένος και να λείει στον υπολογιστή τις δυνατότητες που πρέπει να αναζητηθούν. Αυτά τα χαρακτηριστικά θα βοηθήσουν στη λήψη αποφάσεων.

Η είσοδος ακατέργαστων δεδομένων στον αλγόριθμο σπάνια λειτουργεί, οπότε η εξαγωγή χαρακτηριστικών είναι ένα κρίσιμο μέρος της παραδοσιακής ροής εργασιών της μηχανής.

Αυτό θέτει μια τεράστια ευθύνη στον προγραμματιστή και η απόδοση του αλγορίθμου εξαρτάται σε μεγάλο βαθμό από το πόσο εφευρετικός είναι ο προγραμματιστής. Για πολύπλοκα προβλήματα όπως η αναγνώριση αντικειμένων ή η αναγνώριση χειρόγραφου, αυτό είναι ένα τεράστιο ζήτημα.

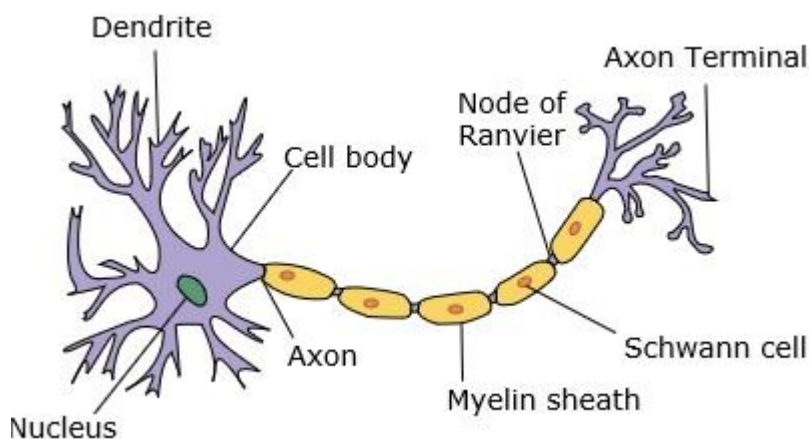
Η βαθιά εκμάθηση, με την ικανότητα να μαθαίνει πολλαπλά στρώματα αναπαράστασης, είναι μια από τις λίγες μεθόδους που μας βοηθούν με την αυτόματη εξαγωγή χαρακτηριστικών. Τα χαμηλότερα στρώματα μπορούν να θεωρηθούν ότι εκτελούν αυτόματη εξαγωγή χαρακτηριστικών, απαιτώντας μικρή ή καθόλου καθοδήγηση από τον προγραμματιστή.

Τεχνητό Νευρικό Δίκτυο – ANN

Το Τεχνητό Νευρικό Δίκτυο, ή απλά νευρωνικό δίκτυο ως ιδέα υπάρχει περίπου 80 χρόνια.

Το 2011 όμως, τα Deep Neural Networks έγιναν δημοφιλή με τη χρήση νέων τεχνικών, την τεράστια διαθεσιμότητα δεδομένων και ισχυρών υπολογιστών.

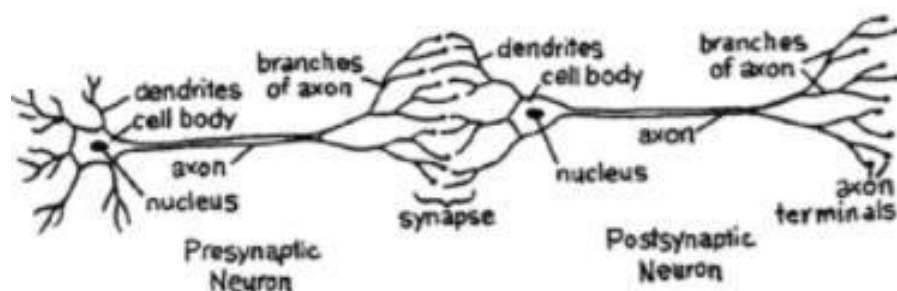
Ένα νευρωνικό δίκτυο μιμείται έναν νευρώνα, ο οποίος έχει δενδρίτες, έναν πυρήνα, έναν άξονα και ένα τελικό άξονα.



Εικόνα 21

Νευρώνας

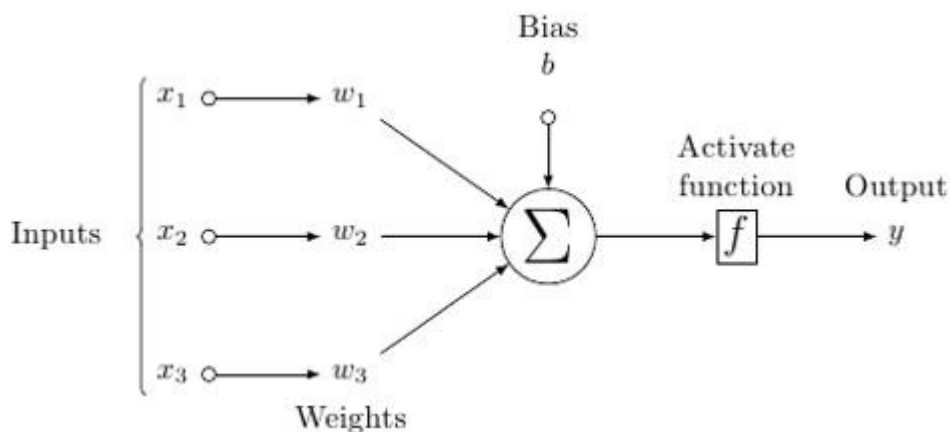
Για ένα δίκτυο, χρειαζόμαστε τουλάχιστον δύο νευρώνες. Αυτοί οι νευρώνες μεταφέρουν πληροφορίες μέσω συνάψεως μεταξύ των αρχικών δενδριτών ενός και του τελικού άξονα άλλου.



Εικόνα 22

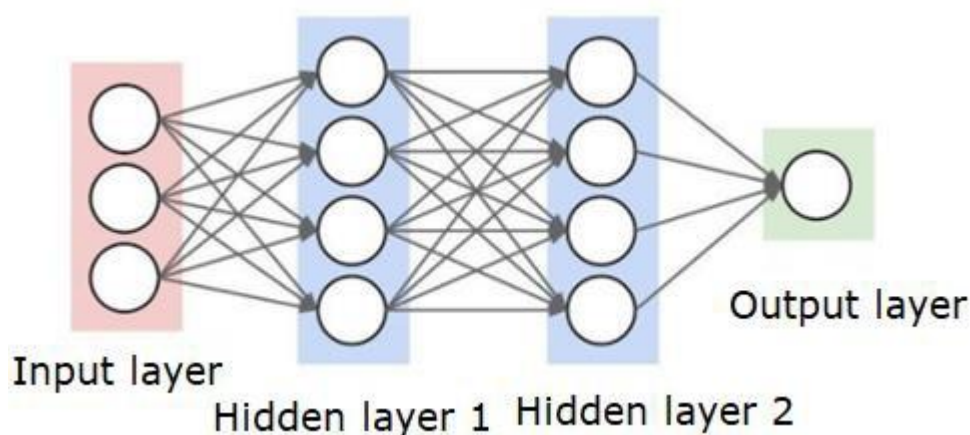
Σύναψη νευρώνων

Η ψηφιοποίηση αυτού του μοντέλου απεικονίζεται ως εξής



Εικόνα 23

Ψηφιοποίηση νευρώνων



Εικόνα 24

Νευρωνικό δίκτυο

Οι κύκλοι είναι νευρώνες ή κόμβοι, με τις λειτουργίες τους στα δεδομένα και οι γραμμές που τις συνδέουν είναι οι πληροφορίες που διαβιβάζονται.

Κάθε στήλη είναι ένα επίπεδο (layer). Το πρώτο επίπεδο των δεδομένων είναι το επίπεδο εισόδου. Στη συνέχεια, όλα τα επίπεδα μεταξύ του εισόδου και της εξόδου είναι τα κρυμμένα επίπεδα.

Αν υπάρχουν ένα ή μερικά κρυμμένα στρώματα, τότε έχουμε ένα ρηχό νευρωνικό δίκτυο. Εάν υπάρχουν πολλά κρυμμένα στρώματα, τότε έχουμε ένα βαθύ νευρωνικό δίκτυο.

Σε αυτό το μοντέλο, έχουμε δεδομένα εισόδου, τα διαμορφώνουμε αναλόγως του βάρους τους και τα περνάμε μέσα από τη λειτουργία στον νευρώνα που ονομάζεται λειτουργία κατωφλίου ή λειτουργία ενεργοποίησης (threshold/activation function).

Βασικά, είναι το άθροισμα όλων των τιμών μετά τη σύγκρισή τους με μια ορισμένη τιμή. Εάν πυροδοτήσετε ένα σήμα, τότε το αποτέλεσμα θα είναι 1 ή 0. Αυτό στη συνέχεια σταθμίζεται και μεταφέρεται στον επόμενο νευρώνα, όπου ακολουθείτε η ίδια διαδικασία με τον επόμενο.

Τα βάρη, είναι τυχαία κατά την εκκίνηση, και είναι μοναδικά ανά είσοδο στον κόμβο / νευρώνα.

Σε ένα τυπικό "feed forward", τον πιο βασικό τύπο νευρωνικού δικτύου οι πληροφορίες θα περάσουν κατευθείαν μέσω του δικτύου που δημιουργήσαμε και θα συγκριθεί με τα αντίστοιχα αποτελέσματα από τα δείγματα δεδομένων.

Από εδώ, θα πρέπει να ρυθμίσουμε τα βάρη για να έχουμε την επιθυμητή έξοδο .

Η αποστολή δεδομένων κατευθείαν μέσω ενός νευρωνικού δικτύου καλείται νευρωνικό δίκτυο τροφοδοσίας προς τα εμπρός (feed forward). Τα δεδομένα μας πηγαινούν από την είσοδο, στα επόμενα επίπεδα με τη σειρά και στη συνέχεια στην έξοδο.

Όταν πηγαινουμε προς τα πίσω και αρχίζουμε να προσαρμόζουμε τα βάρη για να ελαχιστοποιούμε την απώλεια , αυτό ονομάζεται ανάστροφη διάδοση (back propagation).

Αυτό είναι ένα πρόβλημα βελτιστοποίησης. Με το νευρικό δίκτυο, στην πράξη, πρέπει να αντιμετωπίσουμε εκατοντάδες χιλιάδες μεταβλητές, ή ακόμα και εκατομμύρια.

Η πρώτη λύση ήταν να χρησιμοποιηθεί η στοχαστική κλίση ως μέθοδος βελτιστοποίησης. Τώρα, υπάρχουν επιλογές όπως το AdaGrad, το Adam Optimizer και ούτω καθεξής. Είτε έτσι είτε αλλιώς, είναι μια τεράστια υπολογιστική λειτουργία. Αυτός είναι ο λόγος για τον οποίο τα Νευρικά Δίκτυα παρέμειναν ως επί το πλείστον στη θεωρία για πάνω από μισό αιώνα. Μόλις πρόσφατα είχαμε τη δύναμη και την αρχιτεκτονική στα μηχανήματά μας να εξετάσουμε ακόμη και να κάνουμε αυτές τις λειτουργίες αλλά και να αξιοποιήσουμε μεγάλα μεγέθη δεδομένων.

Για απλά καθήκοντα ταξινόμησης, το νευρικό δίκτυο είναι σχετικά κοντά σε επιδόσεις σε άλλους απλούς αλγόριθμους, όπως ο K Nearest Neighbors. Η πραγματική χρησιμότητα των νευρωνικών δικτύων πραγματοποιείται όταν έχουμε πολύ μεγαλύτερα δεδομένα και πολύ πιο περίπλοκα ερωτήματα, τα οποία και ξεπερνούν τα υπόλοιπα μοντέλα μηχανικής μάθησης.

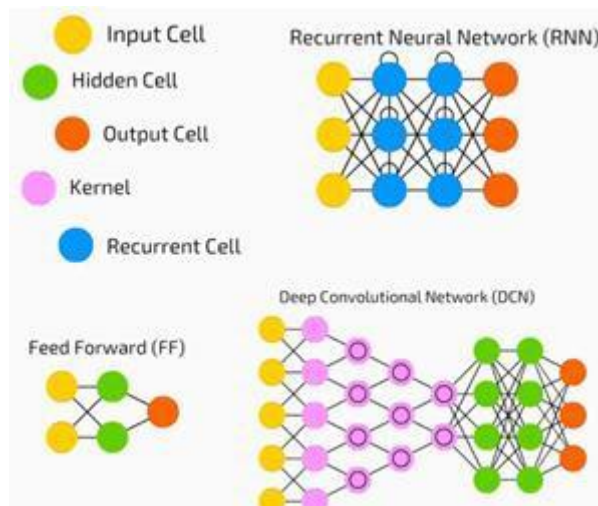
Και στους αλγόριθμους βαθμωτής κλίση (GD) και στοχαστικής βαθμωτής κλίση (SGD), ενημερώνουμε ένα σύνολο παραμέτρων με έναν επαναληπτικό τρόπο για να ελαχιστοποιήσουμε μια λειτουργία σφάλματος. Ενώ στο GD, πρέπει να περάσουμε μέσα από όλα τα δείγματα του εκπαιδευτικού σετ για να γίνει μια μόνο ενημέρωση για μια παράμετρο σε μια συγκεκριμένη επανάληψη, στο SGD, χρησιμοποιούμε μόνο ένα δείγμα εκπαίδευσης ή υποσύνολο του (Minibatch Stochastic gradient Descent), για να κάνουμε την ενημέρωση για μια παράμετρο σε μια συγκεκριμένη. Έτσι, εάν ο αριθμός των δειγμάτων εκπαίδευσης είναι μεγάλος, τότε η χρήση βαθμωτής κλίσης μπορεί να διαρκέσει πάρα πολύ, διότι σε κάθε επανάληψη, όταν ενημερώνετε τις τιμές των παραμέτρων, τρέχετε μέσα από το πλήρες σετ εκπαίδευσης. Από την άλλη πλευρά, η χρήση του SGD θα είναι ταχύτερη επειδή χρησιμοποιείτε μόνο ένα δείγμα εκπαίδευσης και αρχίζει να βελτιώνεται αμέσως από το πρώτο δείγμα. Το SGD συχνά συγκλίνει πολύ πιο γρήγορα σε σύγκριση με το GD αλλά η λειτουργία σφάλματος δεν ελαχιστοποιείται όσο στην περίπτωση της GD. Συχνά, στις περισσότερες περιπτώσεις, η στενή προσέγγιση που λαμβάνετε στο SGD για τις τιμές των παραμέτρων είναι αρκετή επειδή φθάνουν στις βέλτιστες τιμές και διατηρούν ταλαντώσεις εκεί.

Βαθύ νευρωνικό δίκτυο - Deep Neural Network

Ένα βαθύ νευρωνικό δίκτυο (DNN) είναι ένα ANN με πολλαπλά κρυμμένα επίπεδα/ στρώματα μεταξύ των επιπέδων εισόδου και εξόδου. Παρόμοια με τα ρηχά ANNs, τα DNN μπορούν να μοντελοποιήσουν πολύπλοκες μη γραμμικές σχέσεις.

Ο κύριος σκοπός ενός νευρικού δικτύου είναι να λαμβάνει ένα σύνολο εισροών, να εκτελεί προοδευτικά πολύπλοκους υπολογισμούς πάνω σε αυτά και να παρέχει έξοδο για την επίλυση πραγματικών προβλημάτων όπως η ταξινόμηση. Ο ρόλος του χρήστη περιορίζεται στο να τροφοδοτεί τα νευρωνικά δίκτυα.

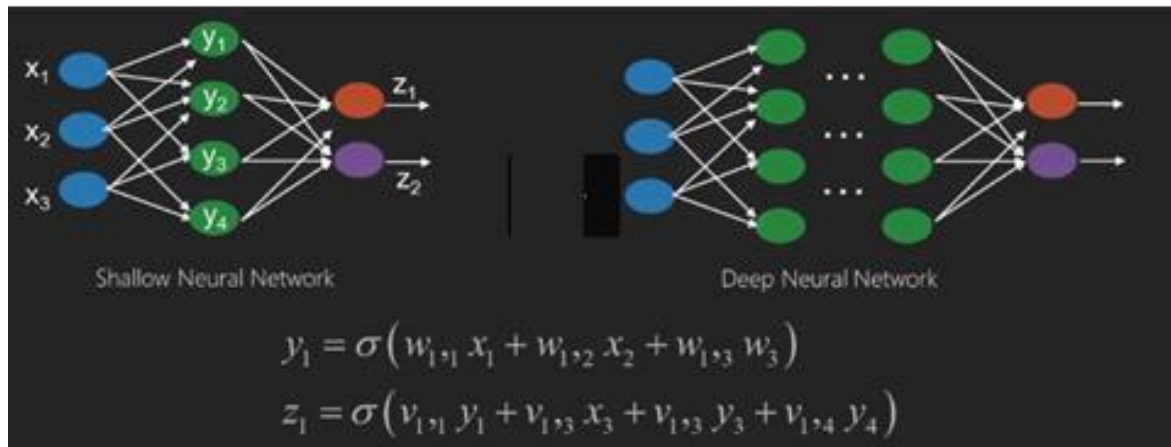
Έχουμε μια είσοδο, μια έξοδο και μια ροή διαδοχικών δεδομένων σε ένα βαθύ δίκτυο.



Εικόνα 25 DNN

Στη βαθιά μάθηση, ο αριθμός των κρυμμένων στρωμάτων, ως επί το πλείστο μη γραμμικών, μπορεί να είναι μεγάλος. πχ 1000 στρώματα. Τα μοντέλα DL παράγουν πολύ καλύτερα αποτελέσματα από τα κανονικά δίκτυα ML. Χρησιμοποιούμε ως επί το πλείστο τη μέθοδο βαθμωτής κλίσης (gradient descent) για βελτιστοποίηση του δικτύου και ελαχιστοποίηση της λειτουργίας απώλειας. Η κατάρτιση των συνόλων δεδομένων αποτελεί σημαντικό μέρος των μοντέλων Deep Learning. Επιπλέον, ο Back propagation είναι ο κύριος αλγόριθμος στην εκπαίδευση μοντέλων DL. Το DL ασχολείται με την εκπαίδευση μεγάλων νευρωνικών δικτύων με σύνθετους μετασχηματισμούς εξόδου εισόδου.

Τα νευρωνικά δίκτυα είναι λειτουργίες που έχουν εισόδους όπως $x_1, x_2, x_3 \dots$ που μετατρέπονται σε εξόδους όπως z_1, z_2, z_3 και ούτω καθεξής σε δύο (ρηχά δίκτυα) ή μερικές ενδιάμεσες πράξεις που ονομάζονται επίσης επίπεδα (βαθιά δίκτυα).

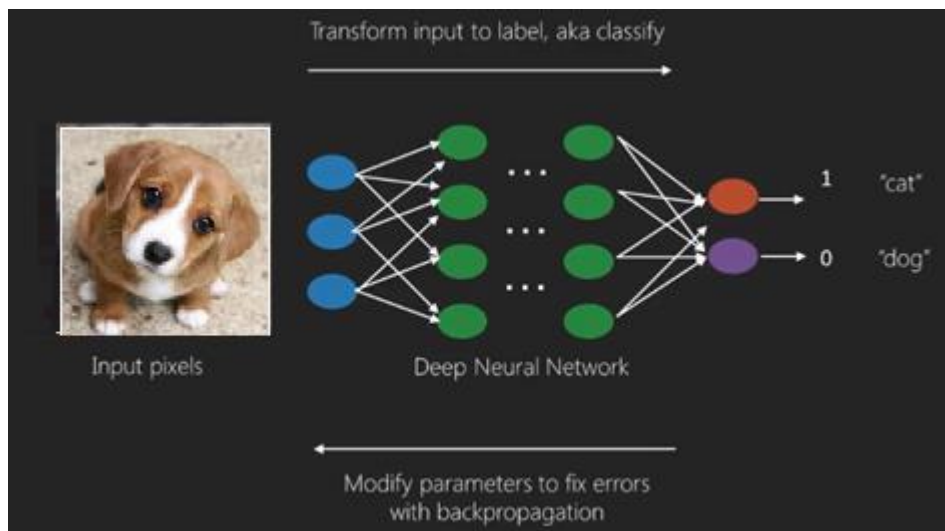


Εικόνα 26

DNN weights

Τα βάρη και οι αρχικές τιμές αλλάζουν από επίπεδο σε επίπεδο. 'W' και 'v' είναι τα βάρη ή συνάψεις των επιπέδων των νευρωνικών δικτύων.

Η αποδοτικότερη περίπτωση χρήσης της βαθιάς μάθησης είναι η εποπτευόμενη μάθηση. Εδώ, έχουμε μεγάλο σύνολο δεδομένων εισόδου με ένα επιθυμητό σύνολο εξόδων.



Εικόνα 27

DNN backpropagation

Το πιο βασικό σύνολο δεδομένων βαθιάς μάθησης είναι το MNIST, ένα σύνολο δεδομένων χειρόγραφων ψηφίων. Μπορούμε να εκπαιδύσουμε βαθιά ένα (Convolutional NN) Συνελκτικό Νευρωνικό Δίκτυο με Keras για να ταξινομήσουμε εικόνες από τα χειρόγραφα ψηφία από αυτό το σύνολο δεδομένων. Παρομοίως μπορούμε να εκπαιδύσουμε ένα δίκτυο σε αντικείμενα από imagenet ή άλλες βάσεις ψηφιοποίησης και κατηγοριοποίησης αντικειμένων.

Η πυροδότηση ή ενεργοποίηση ενός νευρωνικού ταξινομητή παράγει ένα σκορ. Για παράδειγμα, για να ταξινομηθούν οι ασθενείς ως άρρωστοι και υγιείς, εξετάζουμε

παραμέτρους όπως το ύψος, το βάρος και η θερμοκρασία του σώματος, η αρτηριακή πίεση κλπ. Ένα υψηλό σκορ σημαίνει ότι ο ασθενής είναι άρρωστος και ένα χαμηλό σκορ σημαίνει ότι είναι υγιής. Κάθε κόμβος σε επίπεδα εξόδου και κρυφά έχει τους δικούς του ταξινομητές. Το επίπεδο εισόδου λαμβάνει εισόδους και μεταδίδει τις βαθμολογίες του στο επόμενο κρυμμένο στρώμα για περαιτέρω ενεργοποίηση και αυτό συνεχίζεται έως ότου επιτευχθεί η έξοδος. Αυτή η πρόοδος από την είσοδο στην έξοδο από τα αριστερά προς τα δεξιά στην προς τα εμπρός κατεύθυνση ονομάζεται προώθηση προς τα εμπρός forward propagation.

Η πορεία εκχώρησης πίστωσης (CAP) σε ένα νευρικό δίκτυο είναι η σειρά μετασχηματισμών που αρχίζουν από την είσοδο στην έξοδο. Οι CAP επεξεργάζονται πιθανές αιτιώδεις συνδέσεις μεταξύ εισόδου και εξόδου. Το βάθος CAP για ένα δεδομένο νευρωνικό δίκτυο τροφοδοσίας προς τα εμπρός (forward propagation) είναι ο αριθμός των κρυμμένων στρώσεων συν ένα δεδομένου ότι περιλαμβάνεται το στρώμα εξόδου. Για τα επαναλαμβανόμενα νευρικά δίκτυα, όπου ένα σήμα μπορεί να πολλαπλασιάζεται μέσω ενός στρώματος (front/back propagation), το βάθος της CAP μπορεί να είναι δυνητικά απεριόριστο.

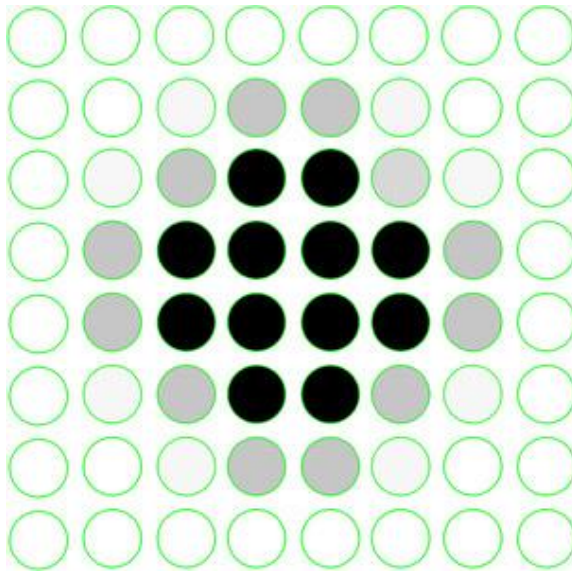
Δομή νευρωνικού δικτύου

Θα εξετάσουμε παρακάτω ένα πολύ απλό νευρωνικό δίκτυο που θα έχει ως σκοπό την αναγνώριση τριών απλών σχημάτων. Τετράγωνο, τρίγωνο και κύκλο. Θεωρούμε ότι οι εικόνες μας είναι διαστάσεων 8*8 πιξελ. Παράλληλα θα βλέπουμε την ίδια εικόνα με αριθμητικά δεδομένα. Οι αριθμοί έχουν πεδίο ορισμού 0 έως 1, με το 0 να υποδηλώνει το μαύρο ενώ το 1 να υποδηλώνει το λευκό.

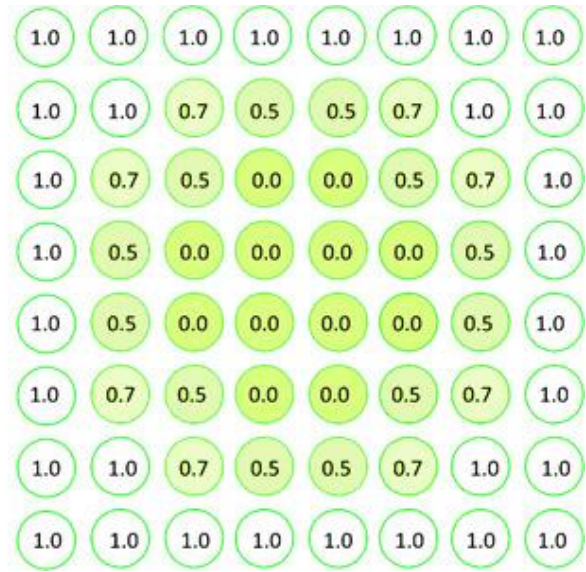


Εικόνα 28

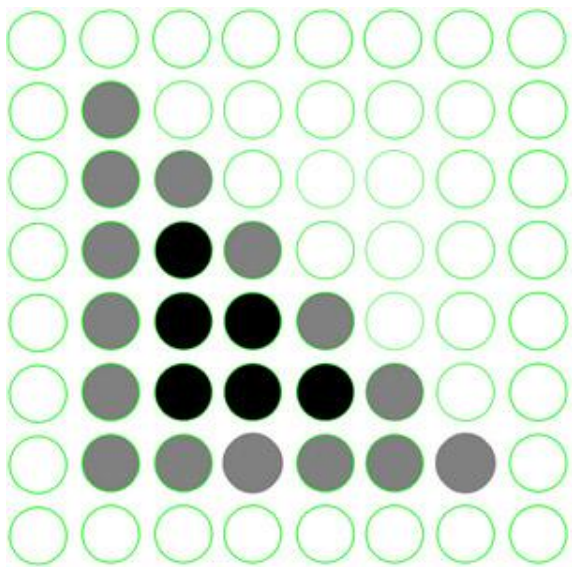
Τετράγωνο



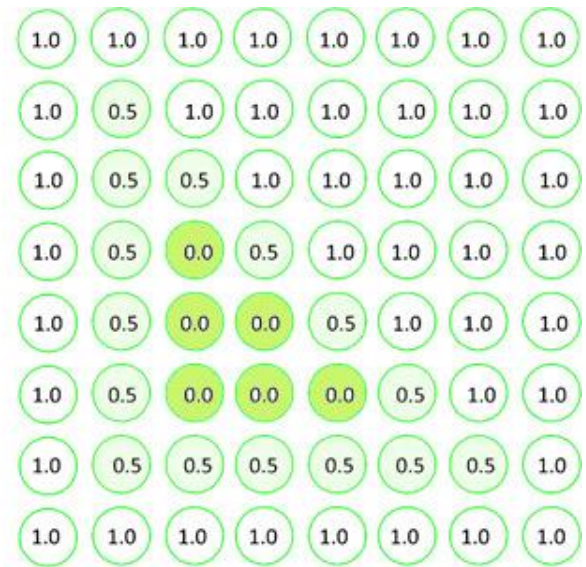
Εικόνα 29



Κύκλος



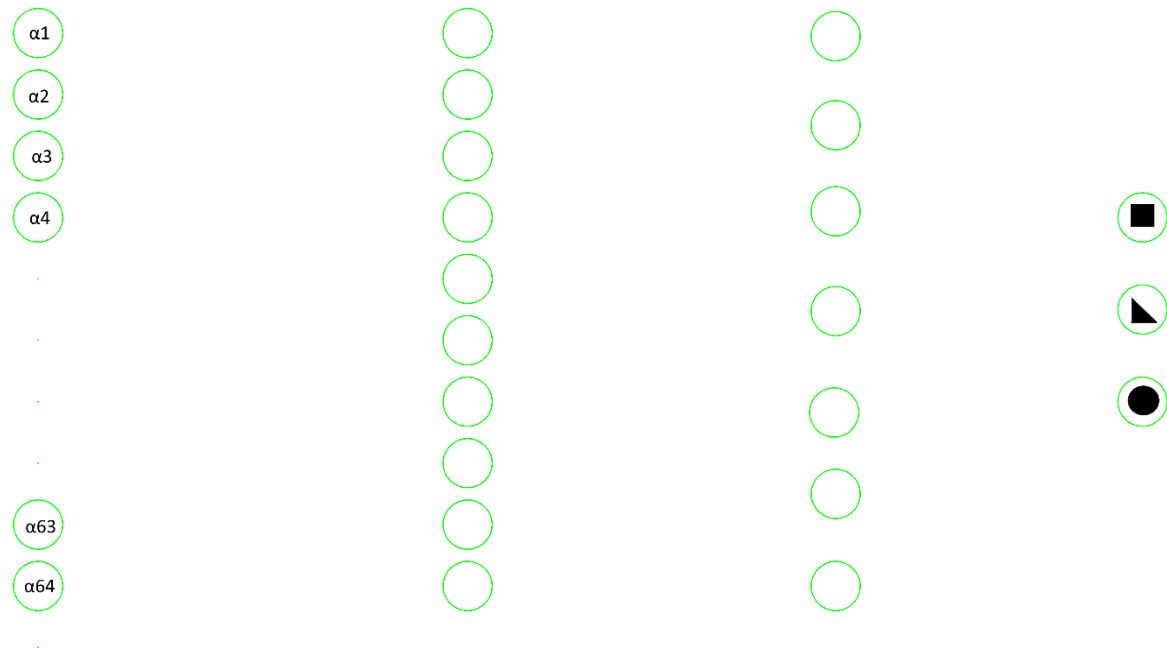
Εικόνα 30



Τρίγωνο

Στην περίπτωση μας , όταν εξετάζουμε μια εικόνα, έχουμε ως είσοδο 64 διαφορετικά αριθμητικά στοιχεία, όσα δηλαδή είναι και τα pixel της εικόνας. Κάθε στοιχείο είναι και

έναν νευρώνα. Μπορούμε να σκεφτούμε κάθε νευρώνα ως ένα κύκλο με έναν αριθμό μέσα. Ο αριθμός (στην περίπτωση μας η φωτεινότητα του pixel) θα το ονομάζουμε ενεργοποιητή. Το σύνολο των 64 στοιχείων αποτελεί και το πρώτο μας επίπεδο (input layer).

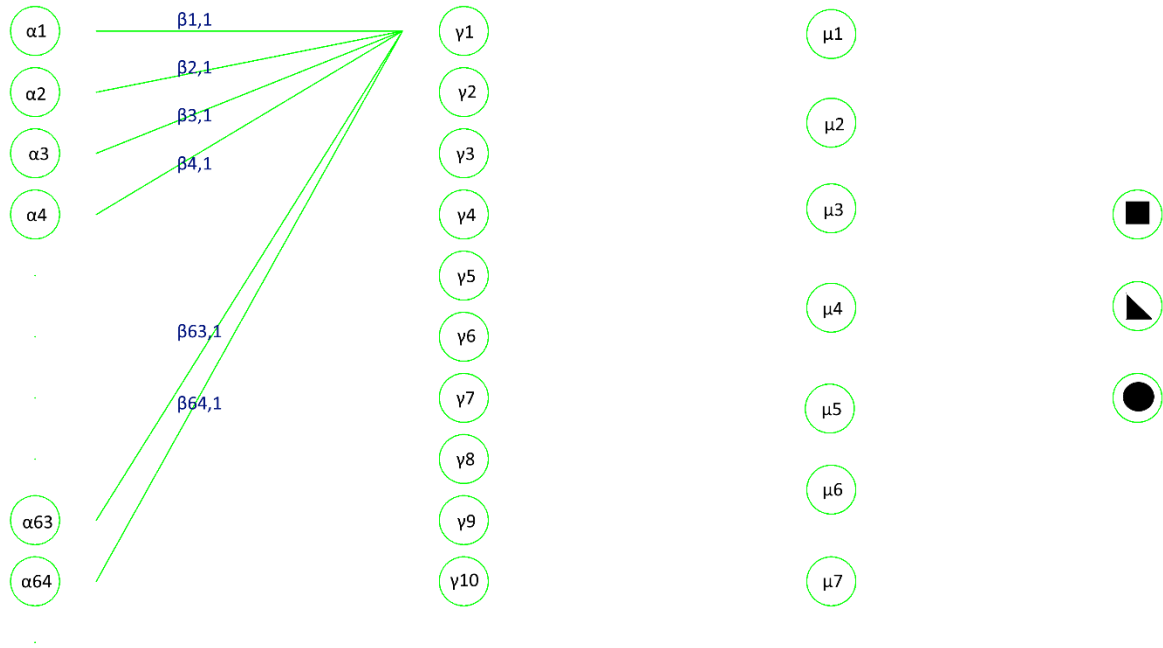


Εικόνα 31

δομή δικτύου 1

Ακολουθούν άλλα 3 επίπεδα, τα 2 ενδιάμεσα τα ονομάζουμε κρυφά επίπεδα και το τελευταίο επίπεδο εξόδου. Στο επίπεδο εξόδου, ο κάθε νευρώνας αντιστοιχεί και σε ένα από τα σχήματα που μπορεί να αναγνωρίσει το δίκτυο μας, με τον ενεργοποιητή του να καθορίζει την πιθανότητα που έχει το σχήμα στην είσοδο του δικτύου να ταυτίζεται μαζί του.

Κάθε νευρώνας του πρώτου επιπέδου, ενώνεται με κάθε νευρώνα του δεύτερου επιπέδου. Το ίδιο ισχύει και για τα επόμενα layers. Η κάθε ένωση υποκρύπτει και μια μαθηματική πράξη, συνήθως κοινή πράξη για όλες τις ενώσεις μεταξύ δύο επιπέδων αλλά με διαφορετικούς συντελεστές σε κάθε νευρώνα. Έτσι το δίκτυο μας γίνεται ως εξής



Εικόνα 32

δομή δικτύου 2

Όπου τελικά ο νευρώνας γ_1 διαμορφώνεται ως εξής

$$\gamma_1 = f(a_1\beta_{1,1} + a_2\beta_{2,1} + a_3\beta_{3,1} + \dots + a_{64}\beta_{64,1} - bias)$$

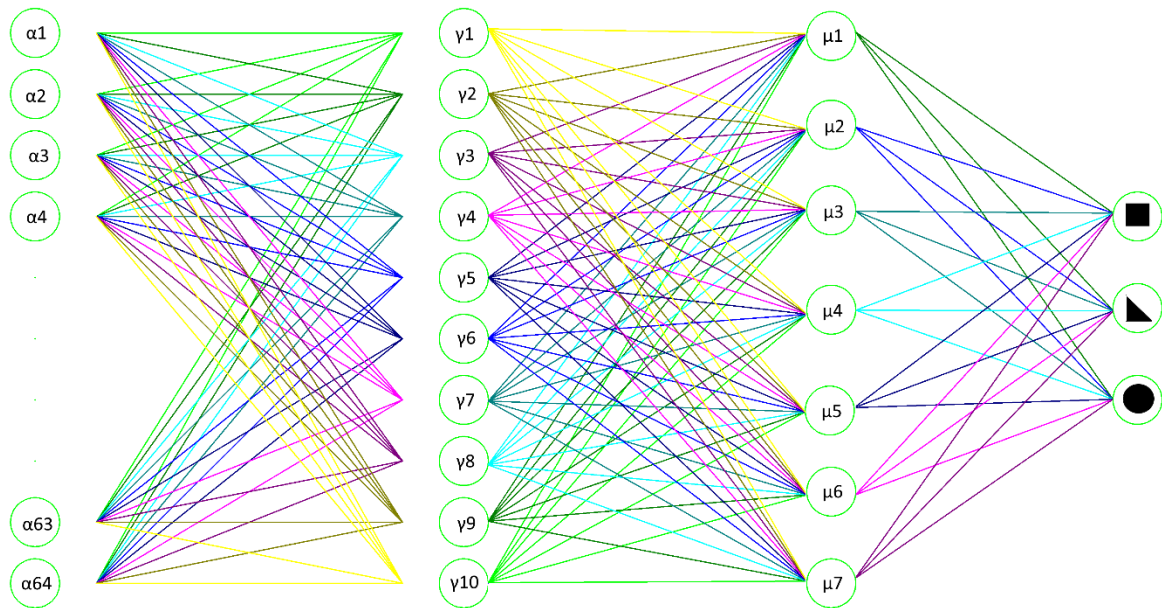
Το bias μπορεί να είναι μια τιμή offset για τις ανάγκες της εφαρμογής μας. Όλοι οι συντελεστές έχουν μια αρχική τυχαία τιμή η οποία αλλάζει και διορθώνεται κατά την εκπαίδευση του δικτύου. Η συνάρτηση f μπορεί να είναι οποιαδήποτε υποστηρίζει το δίκτυο μας. Μια συνηθισμένη είναι η ReLU ($=\max(0,x)$) οπότε τελικά έχουμε συνάρτηση νευρώνων την

$$\gamma_1 = ReLU(a_1\beta_{1,1} + a_2\beta_{2,1} + a_3\beta_{3,1} + \dots + a_{64}\beta_{64,1} - bias)$$

Θα αναφερθούμε παρακάτω για τις συνηθισμένες συναρτήσεις .

Τα β_n στις συνδέσεις ονομάζονται βάρη (weights) .

Με παρόμοιο τρόπο συνδέονται όλοι οι νευρώνες και έτσι έχουμε μια μορφή



Εικόνα 33 δομή δικτύου 3

και έχουμε τη παρακάτω μαθηματική απεικόνιση.

$$\gamma_1 = f \left(\begin{bmatrix} a_1^1 \\ a_2^1 \\ \vdots \\ a_{64}^1 \end{bmatrix} \cdot \begin{bmatrix} \beta_1^1 & \beta_2^1 & \beta_{64}^1 \\ \beta_1^2 & \beta_2^2 & \beta_{64}^2 \\ \vdots & \vdots & \vdots \\ \beta_1^{10} & \beta_2^{10} & \beta_{64}^{10} \end{bmatrix} + \begin{bmatrix} bias1 \\ bias2 \\ \vdots \\ bias64 \end{bmatrix} \right)$$

Ή
$$\gamma = f(\beta \cdot \alpha + b)$$

Τι μπορεί όμως να αντιπροσωπεύουν τα κρυφά επίπεδα; Όσο περισσότερα είναι τόσο πιο αναλυτική είναι η διαδικασία για την ταυτοποίηση, όσο περισσότεροι νευρώνες τόσο αυξάνει η ικανότητα του δικτύου για πιο πολύπλοκους υπολογισμούς.

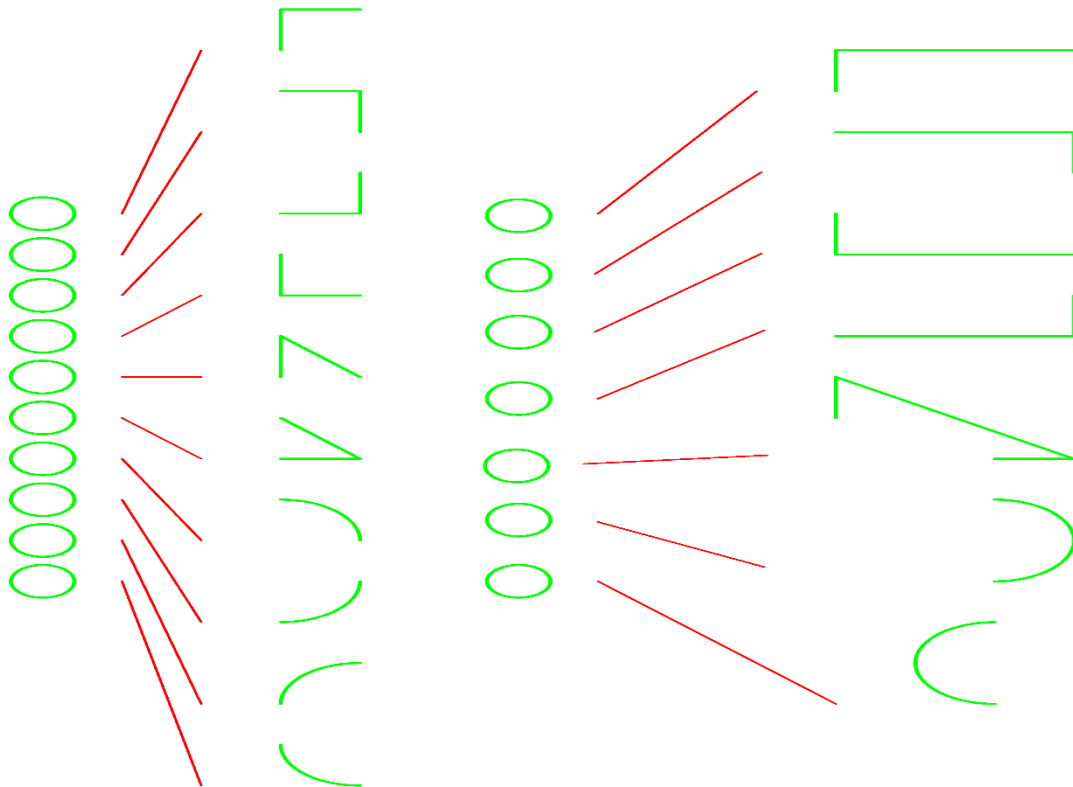
Στις εικόνες ψάχνουμε τα κρίσιμα στοιχεία που θα μας βοηθήσουν να αντιστοιχήσουμε την εικόνα μας με κάτι γνωστό.

Για παράδειγμα στην εικόνα με το δελφίνι μπορούμε να τις αφαιρέσουμε πολλά δεδομένα – λεπτομέρειες και να τονίσουμε τις ακμές ή τις διαχωριστικές γραμμές. Και πάλι μας είναι εύκολο να αναγνωρίσουμε το υποκείμενο.



Εικόνα 34 Ανίχνευση ακμών

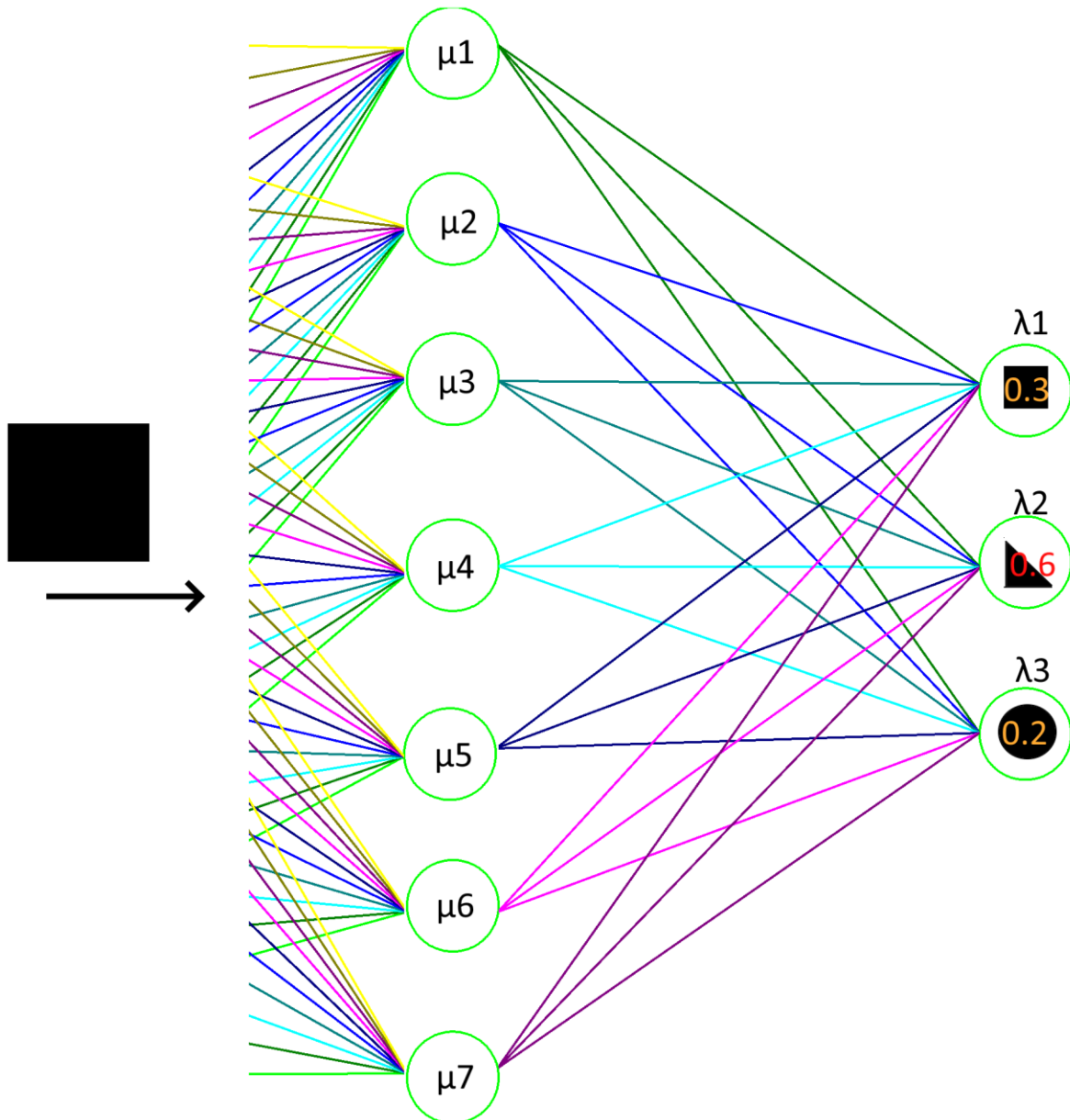
Το δίκτυο μας αντίστοιχα μπορεί να χρησιμοποιεί ένα ή περισσότερα επίπεδα για την ανίχνευση ακμών και άλλο επίπεδο για μοτίβα. Έτσι το πρώτο κρυφό επίπεδο θα μπορούσε να ανιχνεύει τις παρακάτω 10 γωνίες και το δεύτερο τα παρακάτω 7 μοτίβα. Θα μπορούσαμε να χρησιμοποιήσουμε και άλλα επίπεδα όπου για παράδειγμα θα συμπλήρωναν το πρώτο κρυφό επίπεδο, αναδεικνύοντας μας και την κατεύθυνση της γωνίας που βρέθηκε.



Εικόνα 35 Ανίχνευση ακμών 2

Πως όμως έγινε η εκπαίδευση και πήραν τιμές οι ενεργοποιητές τα βάρη και τα biases; Θα πρέπει να κάνουμε την ανάποδη διαδρομή.

Έστω ότι εισάγουμε στο μοντέλο προς εκπαίδευση ένα τετράγωνο και λαμβάνουμε τυχαίες τιμές στην έξοδο 0.3 (πιθανότητα για τετράγωνο) , 0.5 (πιθανότητα για τρίγωνο) και 0.2 (πιθανότητα για κύκλο).



Αν το δίκτυο ήταν ιδανικό θα είχαμε $\lambda_1=1.0$ $\lambda_2=0.0$ και $\lambda_3=0.0$. Τα λ υπολογίζονται τυχαία από τις αρχικές τιμές των βαρών.

$$\lambda_1 = f(\beta_{1,1}^3 \cdot \mu_1 + \beta_{2,1}^3 \cdot \mu_2 + \dots + \beta_{7,1}^3 \cdot \mu_7 - bias)$$

Μπορούμε να δημιουργήσουμε την έννοια της συνάρτησης κόστους για αυτό το παράδειγμα, η οποία θα έχει μεταβλητές τα βάρη και τις σταθερές (bias) του δικτύματος .

$$C(\beta, b) = (0.3 - 1)^2 + (0.5 - 0)^2 + (0.2 - 0)^2 = 1.17$$

Δηλαδή το άθροισμα των τετραγώνων των διαφορών μεταξύ των ενεργοποιητών και του ιδανικού τους αποτελέσματος. Όσο αυξάνει το κόστος του κάθε παραδείγματος εκπαίδευσης τόσο αυξάνει το σφάλμα της πρόβλεψης.

Ο μέσος όρος όλων των συναρτήσεων κόστους από όλα τα παραδείγματα εκπαίδευσης θα μας δώσει το τελικό κόστος του μοντέλου.

Ο στόχος μας μέσω της εκπαίδευσης του μοντέλου είναι να βρεθούν οι τιμές της συνάρτησης C που θα μας δώσουν την ελάχιστη τιμή της C. Κάτι αρκετά χειροπιαστό! Μαθηματικά μπορεί να εκφραστεί ως (κλίση κατάβασης)

$$-\nabla C(\vec{\beta}) = \begin{bmatrix} \frac{dC}{d\beta_1} \\ \frac{dC}{d\beta_2} \\ \vdots \\ \frac{dC}{d\beta_7} \end{bmatrix}$$

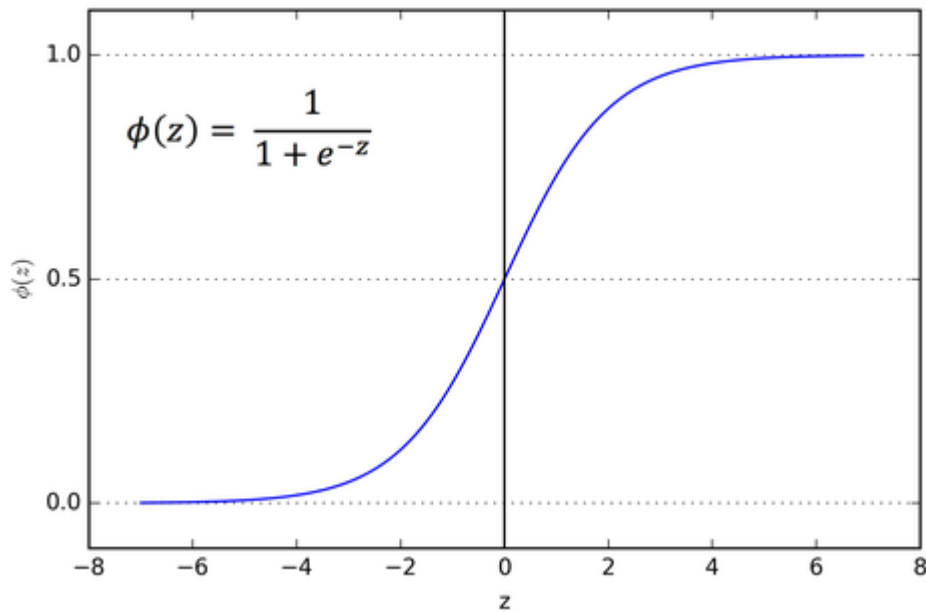
Με αυτό το τρόπο και γνωρίζοντας φυσικά τις συναρτήσεις ενεργοποίησης , σε κάθε κύκλο εκπαίδευσης , το μοντέλο διορθώνει τα βάρη και πλησιάζει το ιδανικό.

Ο αλγόριθμος αυτός όπου αλλάζουμε τα βάρη του τελευταίου επιπέδου και κατόπιν για να αλλάξουν οι τελευταίοι ενεργοποιητές , αλλάζουμε τα βάρη στο αμέσως προηγούμενο επίπεδο κ.ο.κ λέγεται αντίστροφης διάδοσης (back propagation).

Συναρτήσεις ενεργοποίησης

Η χρήση μη γραμμικών συναρτήσεων βοηθάει την βελτιστοποίηση των δικτύων. Εξυπηρετεί το σύνολο τιμών να είναι στο διάστημα 0 έως 1, τόσο για ομοιομορφία μεταξύ των επιπέδων αλλά κυρίως διότι αναφερόμαστε σε πιθανότητες που ορίζονται επίσης στο 0..1 . Ωστόσο η κανονικοποίηση μπορεί να γίνεται και σε επόμενο επίπεδο. Με αυτά τα δεδομένα οι παρακάτω συναρτήσεις μπορούν να χρησιμοποιηθούν.

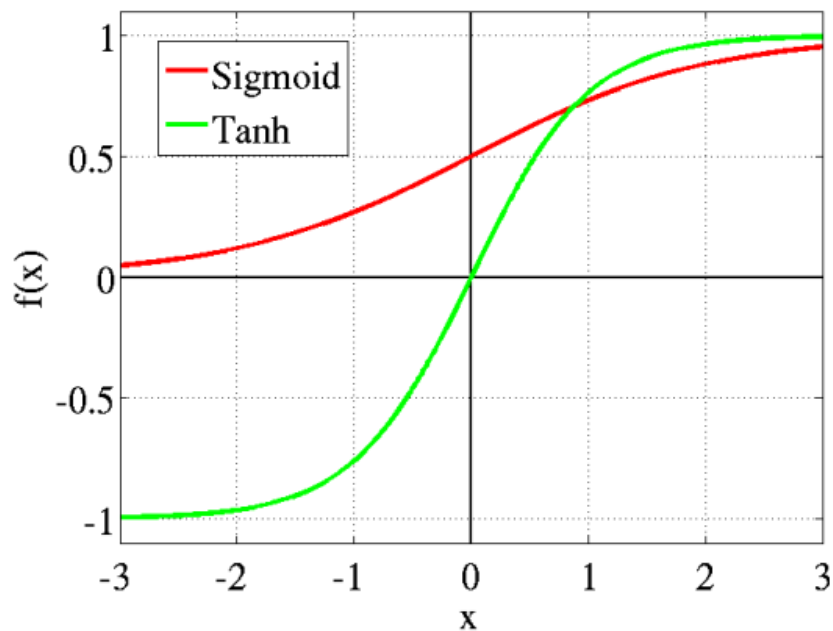
1. Σιγμοειδής (sigmoid or logistic)



Εικόνα 36 sigmoid

Πρόκειται για μονοτονική διαφορική εξίσωση ,δηλαδή είναι συνεχώς αύξουσα και ορίζεται η παράγωγος της σε όλη την καμπύλη.

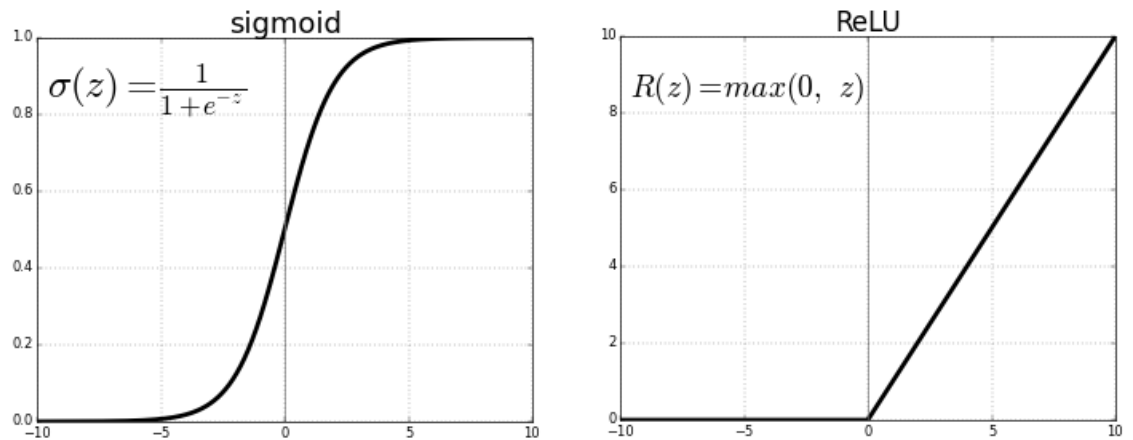
2. Εφαπτόμενη tanh (Tanh or hyperbolic tangent)



Εικόνα 37 tanh

Είναι επίσης σιγμοειδής και αναδεικνύει περισσότερο τις αρνητικές τιμές. Επίσης η μηδενική είσοδος μας δίνει μηδενική έξοδο.

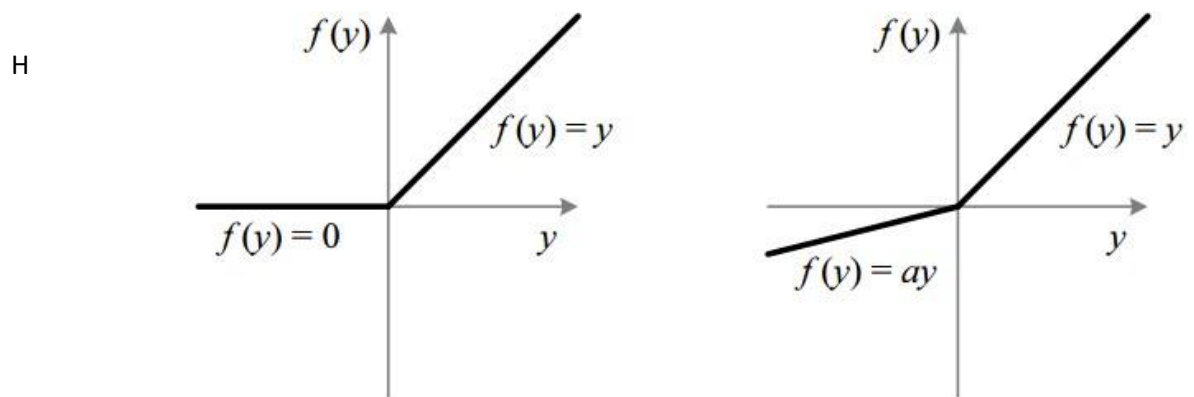
3. ReLU (Rectified Linear Unit)



Εικόνα 38 ReLU

Πρόκειται για μονοτονική συνάρτηση . Η πλέον ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης σε νευρωνικά δίκτυα. Ωστόσο ο μηδενισμός της για κάθε αρνητική είσοδο δημιουργεί κάποια προβλήματα στην απόδοση της.







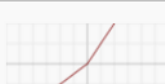


4. Διαρρέων ReLU (Leaky ReLU)



Εικόνα 39 Leaky ReLU

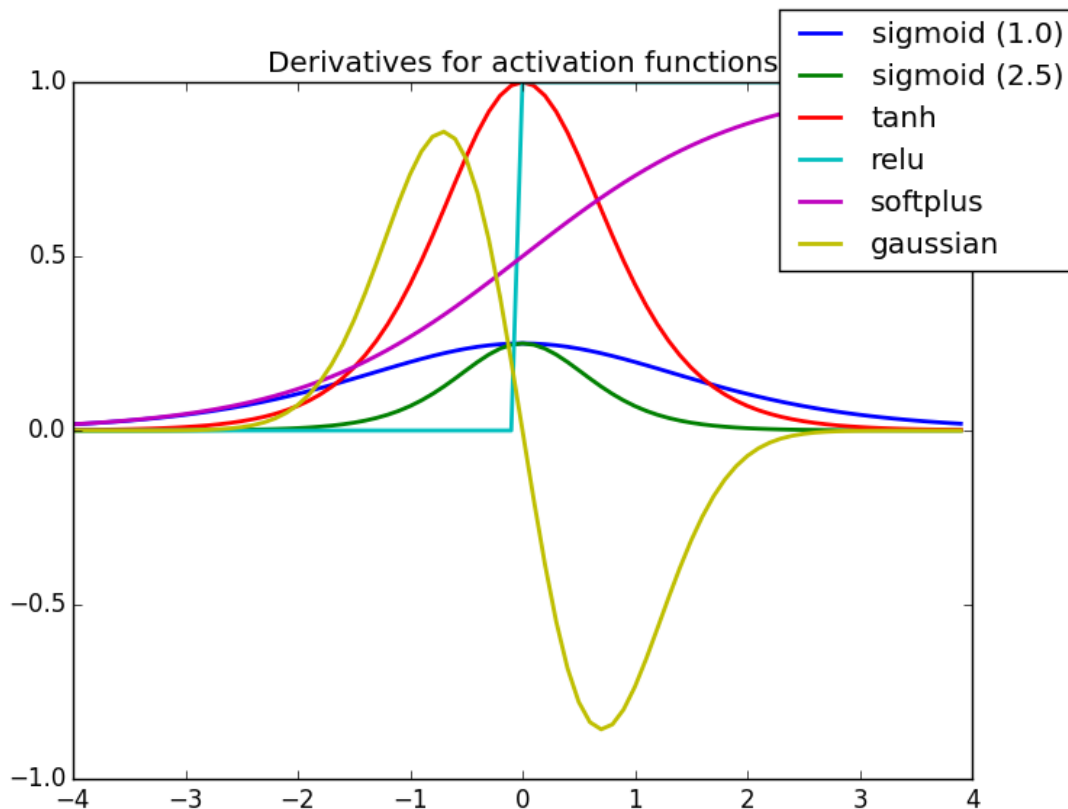
χρήση της διορθώνει κάποια προβλήματα της απλής ReLU. Το a είναι 0.01 , σε κάθε άλλη περίπτωση ονομάζεται τυχαία (random) ReLU

Παρακάτω φαίνονται μερικές σύνηθες συναρτήσεις και οι παράγωγοι τους

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Εικόνα 40

Παράγωγοι συναρτήσεων ενεργοποίησης



Εικόνα 41

Παράγωγοι συναρτήσεων ενεργοποίησης (γραφική αναπαράσταση)

CNN – Convolutional NN (ConvNet)

Συνελικτικά νευρωνικά δίκτυα

Σε σχέση με την απλούστερη και παλαιότερη μορφή εμπροσθοτροφοδοτούμενου (feed-forward) νευρωνικού δικτύου, υπάρχουν διάφορες υλοποιήσεις που εστιάζουν σε συγκεκριμένες εφαρμογές. Τα δίκτυα CNN που θα δούμε παρακάτω έχουν επικρατήσει στις εφαρμογές διαχείρισης/ταξινόμησης εικόνας. Έχουν την ιδιότητα να εξάγουν αυτόματα τα κρίσιμα χαρακτηριστικά μιας εικόνας και να τα χρησιμοποιούν για την εκπαίδευση του δικτύου. Ας δούμε τα επίπεδα που χρησιμοποιεί.

Στην ουσία, είναι φίλτρα τα οποία χρησιμοποιούνται για να δημιουργηθούν νέα συμπυκνωμένα νευρωνικά επίπεδα, τα οποία σε αντίθεση με τα παραδοσιακά όπου είναι πλήρως διασυνδεδεμένα, περιέχουν μόνο τις συνδέσεις που επηρεάζονται και είναι πολύ μικρότερα σε μέγεθος παραμέτρων.

Συνελικτικό επίπεδο - Πυρήνας (kernel)

Σε αυτό το επίπεδο εξάγονται τα πρωτογενή χαρακτηριστικά της εικόνας που μπορεί να μας ενδιαφέρουν.

Αυτό γίνεται αντιμετωπίζοντας τις εικόνες ως πίνακες και εφαρμόζοντας φίλτρα πάνω τους. Τα φίλτρα/kernels ή K είναι πίνακες μικρότερης διάστασης .

Για παράδειγμα έστω $K = \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$ και μια εικόνα 5×5 $I = \begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{matrix}$

Πρακτικά θα ψάχνουμε στην εικόνα το μοτίβο «X» και όσο μεγαλύτερος αριθμός είναι το αποτέλεσμα του φίλτρου τόσο μεγαλύτερη η ομοιότητα του block με το φίλτρο.

Η εφαρμογή του φίλτρου γίνεται με το παρακάτω τρόπο

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Εικόνα

$$\begin{array}{r}
 1x1 \\
 1x0 \\
 1x1 \\
 0x0 \\
 1x1 \\
 1x0 \\
 0x1 \\
 0x0 \\
 + 1x1 \\
 \hline
 4
 \end{array}$$

4		

Χαρακτηριστικό

1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	1	1	0	0

Εικόνα

$$\begin{array}{r}
 1x1 \\
 1x0 \\
 0x1 \\
 1x0 \\
 1x1 \\
 1x0 \\
 0x1 \\
 1x0 \\
 + 1x1 \\
 \hline
 3
 \end{array}$$

4	3	

Χαρακτηριστικό

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Εικόνα

4	3	4
2	4	3
2	3	4

Χαρακτηριστικό

Εικόνα 42

Kernels

Παρόλου που στο παράδειγμα είδαμε μια μονόχρωμη εικόνα 1bit ,ακριβώς ίδια είναι η διαδικασία για εικόνα με μεγαλύτερο βάθος με όποιο χρωματοχώρο και αν τη χρησιμοποιήσουμε. Το μέγεθος του πίνακα που εξάγεται μπορεί να είναι μικρότερο ,ίσο ή και μεγαλύτερο με τον πίνακα της εικόνας. Αυτό έχει να κάνει τον τρόπο κίνησης του φίλτρου επάνω στην εικόνα.

Μπορούν να εφαρμοστούν πολλά παράλληλα φίλτρα για εξαγωγή πολλών βασικών χαρακτηριστικών , ή και σε σειρά , ώστε με το συνδυασμό των φίλτρων να εξαχθούν σύνθετα χαρακτηριστικά όπως ακμές.

Pooling layer – Επίπεδο συμπίκνωσης

Σε αυτό το επίπεδο η εικόνα μικραίνει ώστε να ελαττωθεί η υπολογιστική δύναμη που θα απαιτηθεί στα επόμενα επίπεδα. Αυτό μπορεί να γίνει κρατώντας από ένα block τη μέγιστη τιμή ή το μέσο όρο των τιμών. Επίσης δρα και ως φίλτρο αφαίρεσης θορύβου.

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

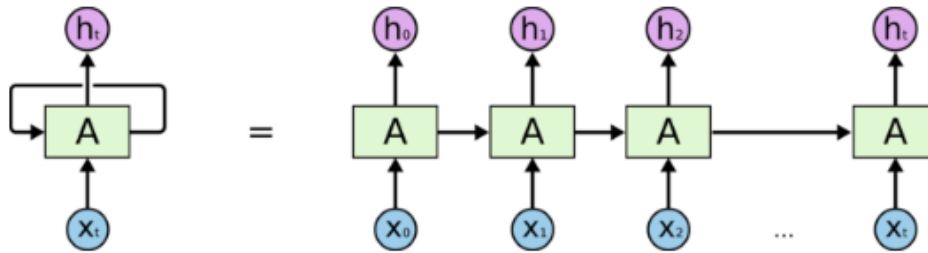
Εικόνα 43 – Συμπύκνωση βάση μεγίστων

Μετά το πέρας αυτών των επιπέδων , η εικόνα είναι έτοιμη να τροφοδοτηθεί στο νευρωνικό δίκτυο για να ταξινομηθεί.

RNN - Recurrent Neural Networks

Επαναλαμβανόμενα νευρωνικά δίκτυα

Η διαφορά αυτών των αλγορίθμων είναι ότι διαθέτουν μνήμη. Δηλαδή εισέρχεται η έννοια του χρόνου και της μεταβολής ενός φαινομένου μέσα στο χρόνο. Αντίθετα με στιγμιότυπα δεδομένων που χειρίζονται άλλοι αλγόριθμοι , οι rnn για την πρόβλεψη μια μελλοντικής κατάστασης συνυπολογίζουν και τις προηγούμενες καταστάσεις. Για αυτό το λόγο είναι ιδιαίτερως χρήσιμοι για εφαρμογές αναγνώρισης φωνής, μετάφρασης , σύνθεσης φωνής κτλ.



An unrolled recurrent neural network.

Εικόνα 44

RNN

Παράδειγμα μοντελοποίησης γλώσσας σε επίπεδο χαρακτήρων.

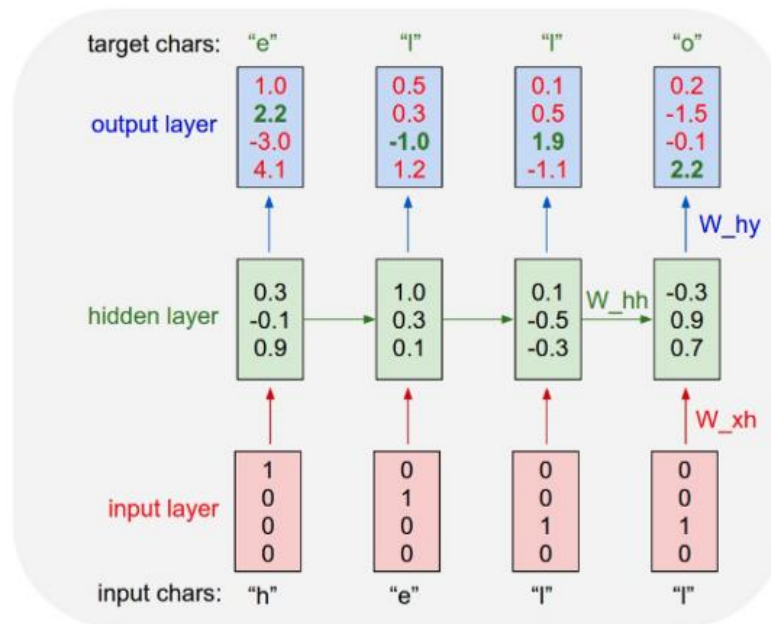
Στο παρακάτω παράδειγμα θα εκπαιδεύσουμε ένα δίκτυο ώστε να προβλέπει τον επόμενο χαρακτήρα μιας λέξης. Ας υποθέσουμε ότι τα διαθέσιμα γράμματα είναι μόλις 4, τα “h” “e” “l” “o” και θέλουμε να εκπαιδεύσουμε το δίκτυο στη λέξη hello.

Για τη σύνθεση της λέξης hello

1. Η πιθανότητα να ακολουθεί το γράμμα e είναι αυξημένη αν προηγείτο το h .
2. Η πιθανότητα του “l” είναι αυξημένη μετά από το “he” .
3. Η πιθανότητα του “l” είναι αυξημένη μετά από το “hel” .
4. Η πιθανότητα τέλος να ακολουθεί το “o” είναι αυξημένη μετά την αναγνώριση του “hell” .

Αρχικά κωδικοποιούμε κάθε χαρακτήρα σε δυαδική μορφή . $h=1000$, $e=0100$, $l=0010$, $o=0001$ και τροφοδοτούμε το δίκτυο βηματικά ένα τη φορά.

Παρακάτω βλέπουμε το αποτέλεσμα.



Εικόνα 45

RNN 2

Στα διανύσματα εξόδου (γαλάζια) μας επιστρέφει το δίκτυο το διάνυσμα της πιθανότητας του επόμενου χαρακτήρα. Δεδομένου ότι κατά την αρχικοποίηση του δικτύου, τα βάρη στα ενδιάμεσα επίπεδα είναι τυχαία, βλέπουμε ότι τροφοδοτώντας το δίκτυο με το "h", αυτό προβλέπει τυχαία ότι το επόμενο γράμμα θα είναι μάλλον το "o" με διάνυσμα πιθανότητας 4.1. Αυτό φυσικά είναι λάθος. Κατά την εκπαίδευση του δικτύου, γνωρίζοντας τη σωστή διαδοχή, με τεχνικές ανάδρασης θα διορθώσει τα βάρη των επιπέδων του ώστε η έξοδος του να προσεγγίζει την επιθυμητή, δηλαδή τα διανύσματα της εξόδου όπου είναι μαρκαρισμένα πράσινα, να αυξάνουν σταδιακά την τιμή τους. Ο αριθμός των ανατροφοδοτήσεων μπορεί να είναι μεγάλος ώστε να επέλθει το επιθυμητό αποτέλεσμα, ενώ όπως καταλαβαίνουμε υπάρχει η έννοια του χρόνου καθώς η έξοδος μετά από ένα "l" εξαρτάται και από το n προηγούμενο χαρακτήρα ("l" για "e" και "o" για "l").

Αναγνώριση αντικειμένου

Object detection

Με τον όρο «αναγνώριση αντικειμένου» περιλαμβάνουμε την ταξινόμηση του – βάση των αντικειμένων που μπορεί να αναγνωρίσει ο αλγόριθμος και προσδιορισμός της θέσης του μέσα στην εικόνα. Στην πρώτη εικόνα που ακολουθεί έχουμε ταξινόμηση εικόνας ενώ στη δεύτερη αναγνώριση αντικειμένου.



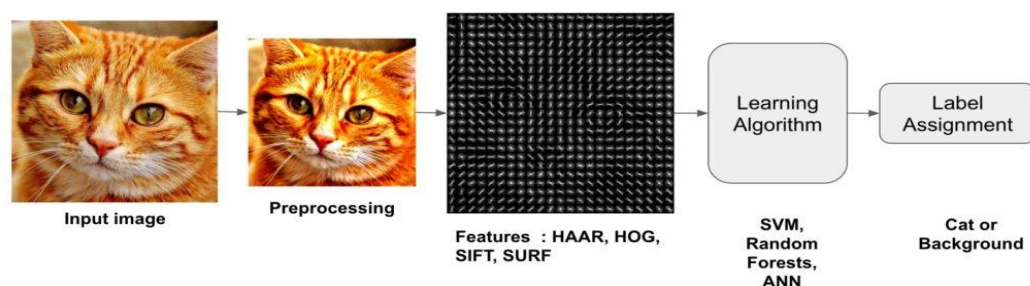
Εικόνα 46

Ταξινόμηση εικόνας – Αναγνώριση αντικειμένου

Κατά τη διαδικασία της ταξινόμησης μιας εικόνας , εισάγουμε την εικόνα στο δίκτυο και αυτό μας εξάγει μια ετικέτα που χαρακτηρίζει όλη την εικόνα. Η αναγνώριση αντικειμένου ακολουθεί της ταξινόμησης και προσπαθεί να εντοπίσει την ακριβή του θέση. Αυτό το κάνει είτε με τεχνικές βαθιάς εκμάθησης είτε με άλλες τεχνικές μηχανικής όρασης.

Για την αναγνώριση αντικειμένου μπορούμε να χρησιμοποιήσουμε είτε τεχνικές μη βαθιάς μάθησης ,όπως σάρωση περιοχής με φίλτρο και χρήση πυραμίδας αναλογιών -όπως ο συνδυασμός HOG με γραμμικούς SVM , είτε με χρήση ενός εκπαιδευμένου δικτύου ως βάση και αλγόριθμους βαθιάς εκμάθησης όπως R-CNN , SSD, YOLO).

Παραδοσιακές τεχνικές

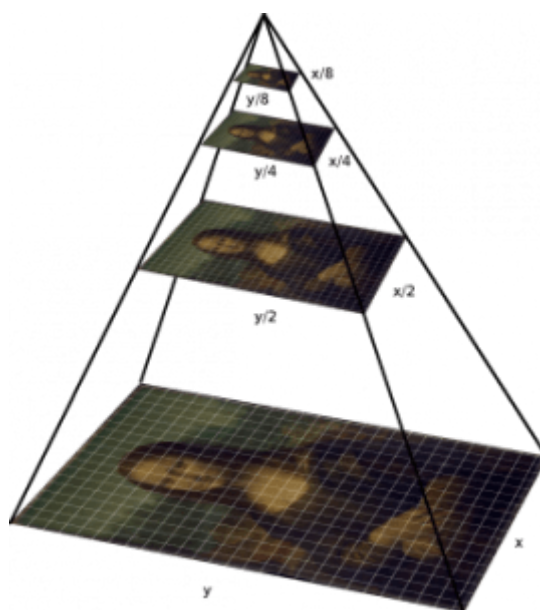


Εικόνα 47

Ταξινόμηση εικόνας με παραδοσιακούς μεθόδους μηχανικής όρασης.

Στην πρώτη κατηγορία χρησιμοποιούμε έναν εξαγωγέα χαρακτηριστικών (feature extraction) όπως το HOG και τροφοδοτούμε τα αποτελέσματα σε έναν αλγόριθμο ταξινόμησης όπως ο SVM. Αυτές οι μέθοδοι χρησιμοποιούνται και για την ιχνηλάτηση αντικειμένων λόγω της μεγάλης τους ταχύτητας.

Για την αναγνώριση του αντικειμένου , η εικόνα μας αλλάζει κλίμακα και κάθε φορά σαρώνεται από την αρχή.



Εικόνα 48

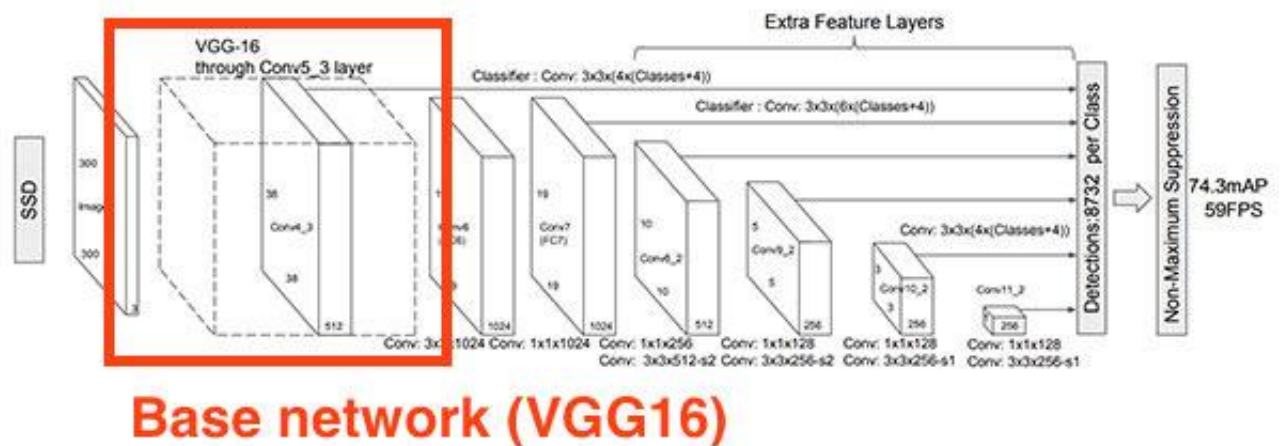
Κλίμακα πυραμίδας

Η μείωση του μεγέθους της εικόνας ,που έχει μορφή πυραμίδας, γίνεται μέχρι κάποιο όριο -ελαχίστου μεγέθους αντικειμένου προς αναγνώριση, το οποίο ορίζουμε εμείς.

Δίκτυο βάσης ενός πλαισίου ανίχνευσης αντικειμένων

Στη δεύτερη μέθοδο που αναφερθήκαμε ακολουθούνται τεχνικές βαθιάς εκμάθησης. Τα δύο βασικά υποσυστήματα εδώ , είναι το πλαίσιο αναγνώρισης αντικειμένων - object detection framework (R-CNN, SSD, YOLO κτλ) και το δίκτυο βάσης (base network) που ενσωματώνεται στο παραπάνω.

Τα δίκτυα βάσης είναι CNN αρχιτεκτονικές ταξινόμησης όπως τα VGGNet ,ResNet ,MobileNet ,DenseNet . Πρόκειται για δίκτυα που έχουν εκπαιδευτεί πάνω σε μεγάλες βάσεις εικόνων για την δημιουργία φίλτρων εξαγωγής κρίσιμων χαρακτηριστικών αντικειμένων.



Εικόνα 49

SSD

Η κάθε μέθοδος αναγνώρισης έχει τα πλεονεκτήματά της. Η επιλογή γίνεται βάσει ακρίβειας που θέλουμε, της επεξεργαστικής ισχύος που διαθέτουμε αλλά και της εφαρμογής. Η κάθε μέθοδος μπορεί να έχει πολλά κοινά στοιχεία με μια άλλη αλλά να διαφέρει σε κάποιο επίπεδο μόνο, ή να είναι απλά μια ελαφρώς βελτιωμένη εκδοχή μιας άλλης. Μερικές είναι οι παρακάτω.

R-CNN (Region based CNN). Στην μέθοδο αυτή δημιουργείτε έναν πεπερασμένο αριθμό περιοχών, έως 2000 (ROI – region of interest, bounding box – το τετράγωνο που το περιβάλλει). Η κάθε περιοχή επιλέγεται βάσει συνάφειας του περιεχομένου της, έτσι έχει πολλές πιθανότητες να περιέχει κάποιο αντικείμενο ή κάποιο χαρακτηριστικό ενός αντικειμένου. Οι περιοχές αυτές αφού αποκτήσουν τη σωστή διάσταση, τροφοδοτούνται σε δίκτυο CNN προς κατηγοριοποίηση. Βελτιωμένες εκδόσεις είναι οι Fast R-CNN και Faster R-CNN.

SSD (Single shot detector)

Semantic segmentation (Σημασιολογική κατάτμηση)

Σε αυτή τη διαδικασία κάθε pixel μιας εικόνας ταυτοποιείται σε μια κλάση. Παρακάτω βλέπουμε μια εικόνα και κατόπιν τις δύο κλάσεις (μοτοσυκλέτα, άνθρωπος) που περιέχει αλλά και την Τρίτη με μαύρο χρώμα, δηλαδή το υπόβαθρο.



Εικόνα 50

Σημασιολογική κατάτμηση

Αυτή η τεχνική χρησιμοποιείται στην ανάπτυξη της αυτόνομης οδήγησης.



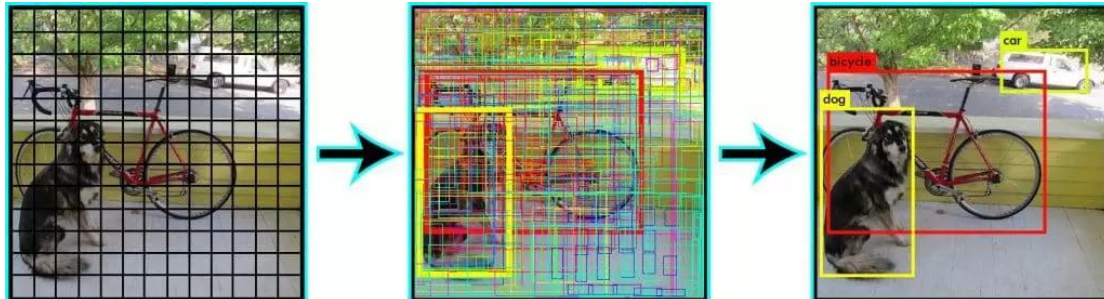
Εικόνα 51

Αυτόνομη οδήγηση

Η αυτόνομη οδήγηση είναι από τις πλέον απαιτητικές εφαρμογές, και το λογισμικό θα πρέπει να είναι σε θέση να αντιλαμβάνεται την ακριβή θέση κάθε αντικειμένου μέσα στο οπτικό του πεδίο.

YOLO (You only look once)

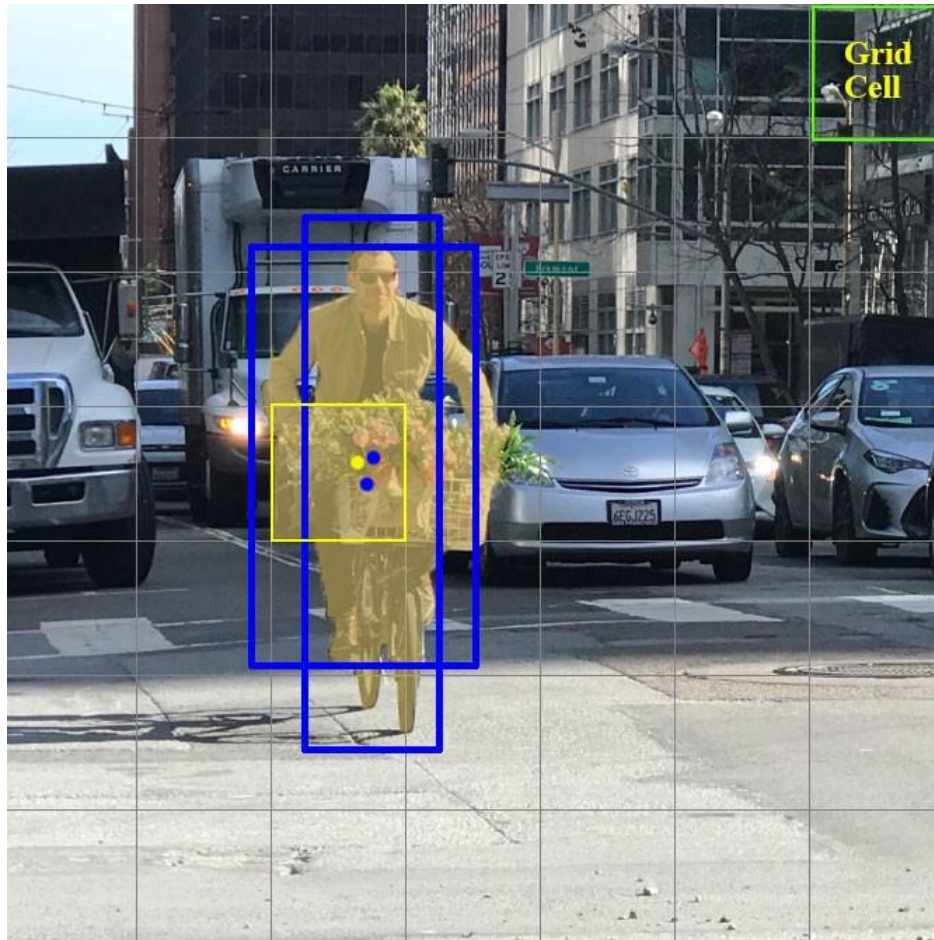
Εδώ η εικόνα τροφοδοτείται μόνο μια φορά στο CNN δίκτυο



Εικόνα 52

Yolo

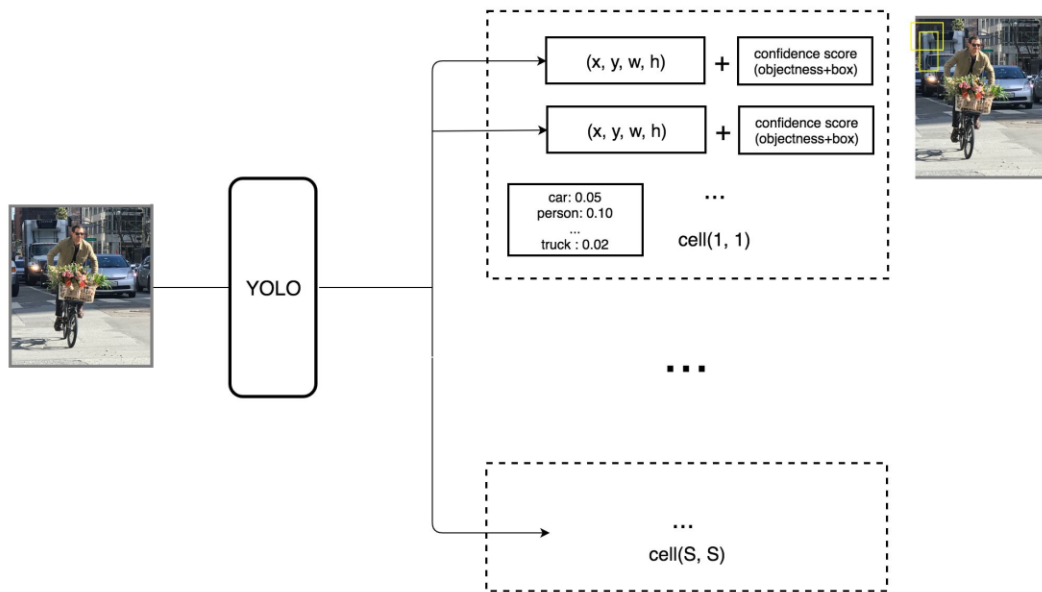
Αντίθετα με άλλους αλγόριθμους που λειτουργούν σε δύο στάδια (επιλογή ROI και ταξινόμηση τους), στην υλοποίηση YOLO υπάρχει μόνο ένα στάδιο. Η εικόνα χωρίζεται σε πλέγμα $S \times S$. Γίνεται συσταδοποίηση των γειτονικών κελιών που μάλλον φέρουν κάποιο αντικείμενο και απορρίπτονται τα κελιά όπου μάλλον δεν υπάρχει αντικείμενο ενδιαφέροντος αλλά πιθανότατα είναι μέρος του υπόβαθρου. Στις δύο τελευταίες εκδόσεις του YOLO γίνεται χρήση και των anchor boxes. Αυτά είναι ορθογώνια πλαίσια όπου οι αναλογίες τους χρησιμοποιούνται στα bounding boxes κατά την ομαδοποίηση. Έτσι στη περίπτωση μας 3 κουτιά δημιουργούνται γύρω από κάθε πιθανό αντικείμενο και στην συνέχεια διορθώνονται και απορρίπτονται όλα εκτός ενός. Αυτό βέβαια γίνεται τρεις φορές σε τρεις διαφορετικές κλίμακες της εικόνας, οπότε μιλάμε για 9 κουτιά στην περίπτωση του YOLO v3. Η αναλογία των anchor boxes δημιουργείτε πριν την εκπαίδευση του δικτύου και είναι σχετική με τα αντικείμενα εκπαίδευσης. Δηλαδή είναι κουτιά που διατηρούν την αναλογία ενός ανθρώπου στο παράδειγμα παρακάτω. Το αρνητικό της μεθόδου είναι ότι δεν ανιχνεύει μικρά αντικείμενα καθώς σε κάθε πλαίσιο μπορεί να αντιστοιχήσει μόνο ένα τύπο αντικειμένου.



Εικόνα 53

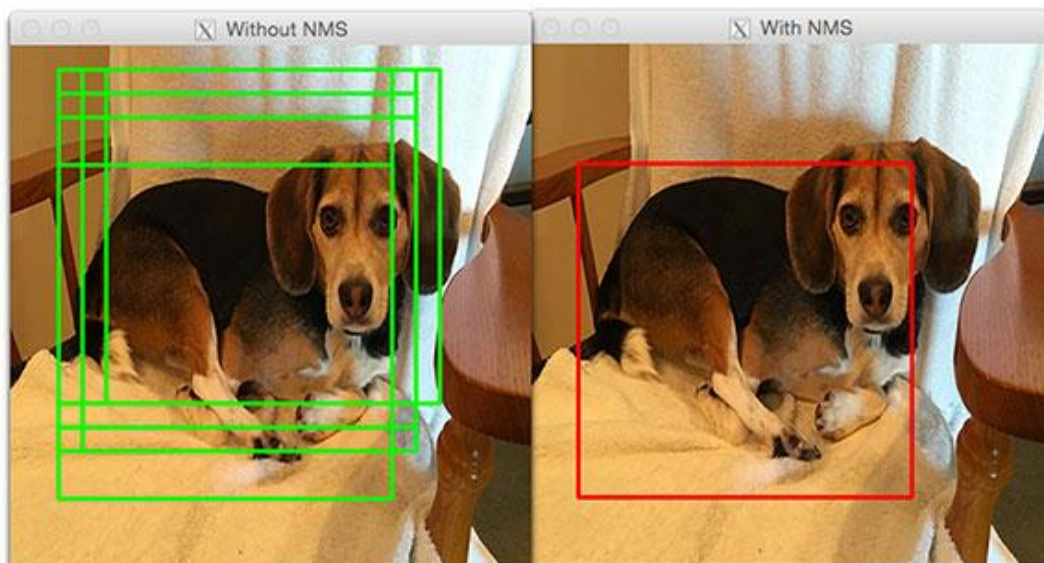
Anchor boxes

Όπως βλέπουμε παραπάνω , αφού βρέθηκε μέσω συσταδοποίησης ότι πιθανότατα υπάρχει ένα αντικείμενο του οποίου το κέντρο είναι στο κελί 3,4 , δημιουργούνται τα τρία anchor boxes με κέντρο αυτό το κελί. Οι περαιτέρω ενέργειες θα επικεντρωθούν σε αυτές τις περιοχές -όπως και σε όποια άλλα anchor boxes έχουν βρεθεί , και έτσι θα αποφορτιστεί αρκετά ο αλγόριθμος.



Εικόνα 54

Yolo προγνώσεις αντικειμένων



Εικόνα 55

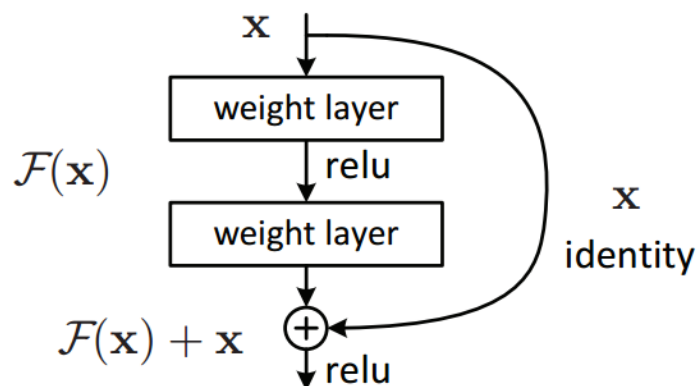
maxima suppression

Για την επιλογή του τελικού πλαισίου που επιλέγεται σε κάθε στιγμιότυπο γίνεται εφαρμογή τεχνικής non-maxima suppression, κατά την οποία από τα πλαίσια όπου επαληθεύουν ότι ανήκουν στο ίδιο αντικείμενο, επιλέγεται αυτό στο οποίο αποδόθηκε η μεγαλύτερη πιθανότητα αναγνώρισης (μεγαλύτερο IoU).

Για την εξαγωγή χαρακτηριστικών χρησιμοποιείτε ως δίκτυο βάσης το DarkNet-53. Αυτό περιέχει 53 συνελκτικά επίπεδα και αρκετά σημεία ανατροφοδότησης (residual). Τα τελευταία μεταφέρουν πληροφορία μεταξύ blocks επιπέδων καθώς έχει αποδειχθεί ότι όσο το δίκτυο γίνεται πιο βαθύ, έχει την τάση να υποβαθμίζει τη συνεισφορά των αρχικών επιπέδων.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Εικόνα 56 Darknet-53 layers



Εικόνα 57 Residual block

Μετρήσεις Αξιολόγησης μοντέλων (Evaluation Metrics)

Υπάρχουν πολύ τρόποι αξιολόγησης ενός μοντέλου . Ωστόσο αν δεν γίνει συνδυασμός τους είναι πιθανό τα αποτελέσματα τους να είναι παραπλανητικά. Παρακάτω θα δούμε μερικούς.

Classification Accuracy	- Ακρίβεια ταξινόμησης
Logarithmic Loss	- Λογαριθμική απώλεια
Confusion Matrix	- Μήτρα σύγχυσης
Area under Curve	- Περιοχή κάτω από καμπύλη
F1 Score	- Βαθμολογία F1
Mean Absolute Error	- Μέσο απόλυτο σφάλμα
Mean Squared Error	- Μέσο σφάλμα τετραγώνου
Mean Average precision – mAP	Ακρίβεια μέσου όρου

Classification Accuracy

Ακρίβεια ταξινόμησης ή απλά ακρίβεια, είναι η αναλογία του αριθμού των σωστών προβλέψεων στο συνολικό αριθμό των δειγμάτων εισόδου.

$$Accuracy = \frac{\text{σωστές προβλέψεις}}{\text{σύνολο προβλέψεων}}$$

Λειτουργεί καλά μόνο αν υπάρχει ίσος αριθμός δειγμάτων που ανήκουν σε κάθε κλάση. Για παράδειγμα, θεωρήστε ότι υπάρχουν 98% δείγματα της κλάσης α και 2% των δειγμάτων της κλάσης β στο εκπαιδευτικό μας σετ. Στη συνέχεια, το μοντέλο μας μπορεί εύκολα να πάρει 98% ακρίβεια εκπαίδευσης, προβλέποντας απλώς κάθε δείγμα εκπαίδευσης που ανήκει στην κλάση α. Όταν το ίδιο μοντέλο δοκιμαστεί σε ένα σετ δοκιμών με 60% δείγματα των δειγμάτων κλάσης α και 40% της κατηγορίας β, τότε η ακρίβεια της δοκιμής θα πέσει στο 60%. Η ακρίβεια της ταξινόμησης είναι μεγάλη, αλλά μας δίνει την ψεύτικη αίσθηση ότι επιτυγχάνουμε υψηλή ακρίβεια. Το πραγματικό πρόβλημα προκύπτει, όταν το κόστος της εσφαλμένης κατάταξης των δειγμάτων δευτερεύουσας κλάσης είναι πολύ υψηλό. Εάν αντιμετωπίζουμε μια σπάνια αλλά μοιραία ασθένεια, το κόστος της αποτυχίας για τη διάγνωση της ασθένειας ενός άρρωστου είναι πολύ υψηλότερο από το κόστος αποστολής ενός υγιούς ατόμου για περισσότερες εξετάσεις.

Logarithmic Loss

Η λογαριθμική απώλεια, λειτουργεί τιμωρώντας τις ψευδείς ταξινομήσεις. Λειτουργεί καλά για ταξινόμηση πολλαπλών κλάσεων. Κατά την εργασία με την λογαριθμική απώλεια, η

ταξινόμηση πρέπει να εκχωρεί την πιθανότητα σε κάθε κλάση για όλα τα δείγματα. Ας υποθέσουμε ότι υπάρχουν N δείγματα που ανήκουν σε τάξεις M , τότε η λογαριθμική απώλεια υπολογίζεται ως εξής:

$$\text{Λογαριθμική απώλεια} = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij})$$

Όπου

y_{ij} , υποδεικνύει αν το δείγμα i ανήκει στην κλάση j ή όχι

p_{ij} , υποδεικνύει την πιθανότητα του δείγματος i να ανήκει στην κλάση j

Η Log loss δεν έχει ανώτερο όριο και υπάρχει στην περιοχή $[0, \infty)$. Η log loss όσο πιο πολύ πλησιάζει το 0 υποδηλώνει μεγαλύτερη ακρίβεια, ενώ αν απομακρύνεται τότε υποδεικνύει χαμηλότερη ακρίβεια. Γενικά, η ελαχιστοποίηση της λογαριθμικής απώλειας δίνει μεγαλύτερη ακρίβεια στην ταξινόμηση

Confusion Matrix

Το Confusion Matrix μας αποτυπώνει την απόδοση ενός μοντέλου μέσω ενός πίνακα. Ας υποθέσουμε ότι έχουμε ένα δυαδικό πρόβλημα ταξινόμησης. Έχουμε μερικά δείγματα που ανήκουν σε δύο τάξεις: Ναι ή όχι. Επίσης, έχουμε έναν ταξινομητή που προβλέπει σε πια κλάση ανήκει ένα δεδομένο δείγμα εισόδου. Όταν τεστάρουμε το μοντέλο μας σε 100 δείγματα, έχουμε το ακόλουθο αποτέλεσμα.

n = 100	Πρόβλεψη : ΟΧΙ	Πρόβλεψη : ΝΑΙ
Πραγματικότητα : ΟΧΙ	30	6
Πραγματικότητα : ΝΑΙ	4	60

Εικόνα 58 Confusion Matrix

Υπάρχουν 4 σημαντικοί όροι:

True Positives : οι περιπτώσεις στις οποίες προβλέψαμε Ναι και η πραγματική κατάσταση ήταν επίσης Ναι.

True Negatives : οι περιπτώσεις στις οποίες προβλέψαμε Όχι και η πραγματική κατάσταση ήταν Όχι.

False Positives : οι περιπτώσεις στις οποίες προβλέψαμε Ναι και η πραγματική έξοδος ήταν Όχι.

False Negatives : οι περιπτώσεις στις οποίες προβλέψαμε Όχι και η πραγματική παραγωγή ήταν Ναι.

Η ακρίβεια για τη μήτρα μπορεί να υπολογιστεί με τη λήψη του μέσου όρου των τιμών που βρίσκονται κατά μήκος του "κύριου διαγώνιου".

$$Accuracy = \frac{\text{σωστές προβλέψεις}}{\text{σύνολο προβλέψεων}} = \frac{TruePositives + TrueNegatives}{\text{σύνολο προβλέψεων}}$$

$$Accuracy = \frac{60 + 30}{100} = 0.9$$

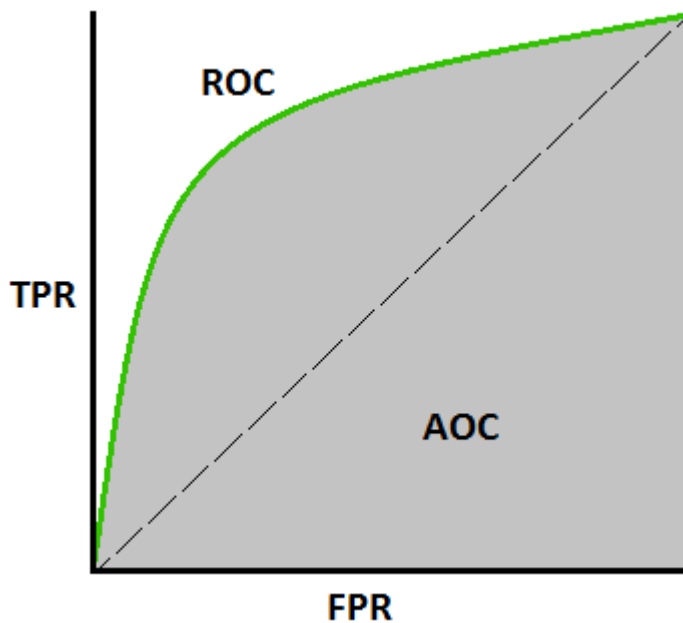
Ο πίνακας σύγκρισης αποτελεί τη βάση για τους άλλους τύπους μετρήσεων.

Area under Curve - Receiver Operating Characteristics

Η AUC-ROC, AUROC.

Χρησιμοποιείται για να οπτικοποιήσουμε την απόδοση ενός ταξινομητή. Η καμπύλη AUC-ROC είναι μια μέτρηση απόδοσης σε διάφορες ρυθμίσεις threshold. Το ROC είναι μια καμπύλη πιθανότητας και η AUC αντιπροσωπεύει το βαθμό ή το μέτρο της δυνατότητας διαχωρισμού. Όσο υψηλότερη η AUC, τόσο καλύτερα το μοντέλο προβλέπει τα 0 ως 0 και τα 1 ως 1 σε ένα δυαδικό σύστημα

Η καμπύλη ROC σχεδιάζεται με TPR έναντι του FPR όπου ο TPR είναι στον άξονα y και ο FPR είναι στον άξονα x.



Εικόνα 59 AUROC

Ας δούμε συγκεντρωμένους τους όρους που χρησιμοποιούνται.

- TP - True Positive
- FP - False Positive
- TN - True Negative

FN - False Negative

TPR - True Positive Rate / Recall / Sensitivity

$$TPR = \frac{TP}{TP+FN}$$

Specificity

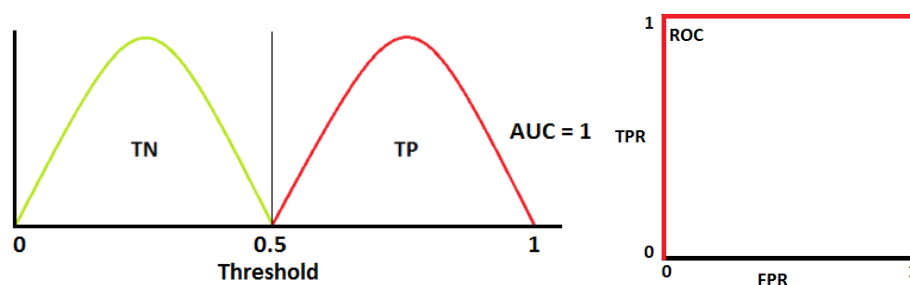
$$Specificity = \frac{TN}{TN+FP}$$

FPR - False Positive Rate

$$FPR = 1 - Specificity = \frac{FP}{TN+FP}$$

Ένα ιδανικό μοντέλο έχει AUC με τιμή 1, πράγμα που σημαίνει ότι έχει άριστη δυνατότητα διαχωρισμού. Ένα φτωχό μοντέλο έχει AUC κοντά στο 0 που σημαίνει ότι έχει λάθος δυνατότητα διαχωρισμού. Δηλαδή κάνει ακριβώς την ανάποδη ταξινόμηση από ότι θα έπρεπε. Προβλέπει τα 0 ως 1 και τα 1 ως 0. Όταν η AUC είναι 0,5, σημαίνει ότι το μοντέλο δεν έχει καθόλου ικανότητα διαχωρισμού κλάσης.

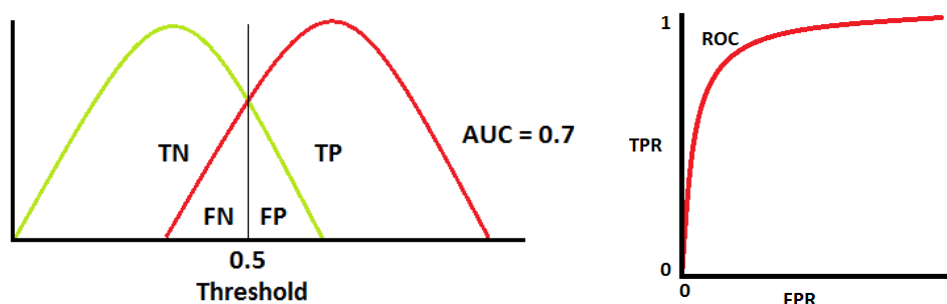
Σχηματικά υπάρχουν οι περιπτώσεις



Εικόνα 60 AUC = 1

όπου ο διαχωρισμός είναι ιδανικός.

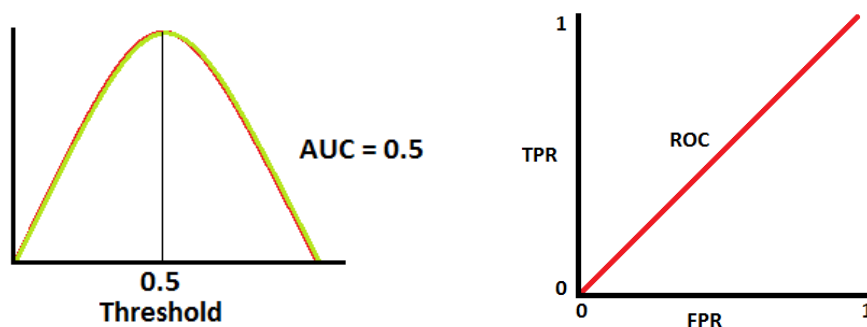
Μια πιο ρεαλιστική κατάσταση είναι η παρακάτω



Εικόνα 61 AUC = 0.7

Αυτό σημαίνει ότι το μοντέλο μας ,έχει πιθανότητα 70% να ταξινομήσει σωστά ένα δείγμα.

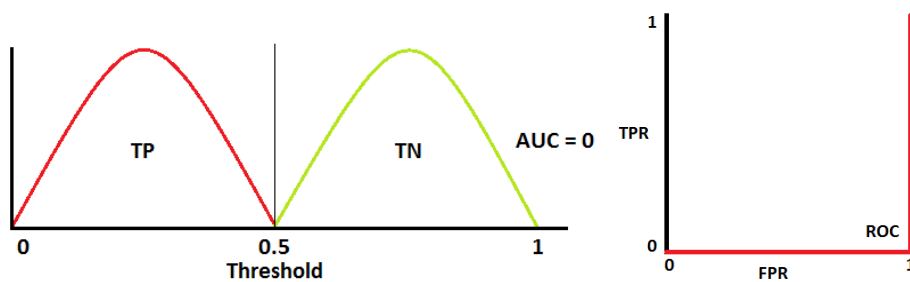
Η παρακάτω κατάσταση είναι η χειρότερη .



Εικόνα 62

AUC = 0.5

Ενώ παρακάτω είναι η κατάσταση ανάποδης ταξινόμησης.



Εικόνα 63

AUC = 0

Σε περίπτωση που έχουμε περισσότερες κλάσεις από δύο, μπορούμε να υπολογίζουμε το AUC ξεχωριστά για κάθε δύο κλάσεις.

F1 Score

Η βαθμολογία F1 χρησιμοποιείται για τη μέτρηση της ακρίβειας μιας δοκιμής. Η βαθμολογία F1 είναι ο αρμονικός μέσος μεταξύ ακρίβειας (precision) και ανάκλησης (recall). Το εύρος για τη βαθμολογία F1 είναι [0,1]. Όσο μεγαλύτερη είναι η βαθμολογία F1, τόσο το καλύτερο είναι η απόδοση του μοντέλου μας. Μαθηματικά, μπορεί να εκφραστεί ως:

$$f1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Precision (ακρίβεια) : Είναι ο αριθμός των σωστών θετικών αποτελεσμάτων διαιρούμενο με τον αριθμό των θετικών αποτελεσμάτων που μας επιστρέφει ο ταξινομητής.

$$Precision(p) = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall (ανάκληση) : είναι ο αριθμός των σωστών θετικών αποτελεσμάτων διαιρούμενο με τον αριθμό όλων των σχετικών δειγμάτων (όλα τα δείγματα που θα έπρεπε να έχουν προσδιοριστεί ως θετικά).

$$Recall(r) = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Θα την συναντήσουμε και ως TPR (true positive rate) αλλά και ως ευαισθησία (sensitivity) .

Mean Absolute Error

Το μέσο απόλυτο σφάλμα – MAE ,είναι ο μέσος όρος της διαφοράς μεταξύ των πραγματικών τιμών και των προβλεπόμενων τιμών. Μας δίνει το μέτρο του πόσο μακριά ήταν οι προβλέψεις από την πραγματικότητα. Ωστόσο, δεν μας φαναιρώνει την κατεύθυνση του σφάλματος. Μαθηματικά, αντιπροσωπεύεται ως:

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - x_j|$$

Όπου y_j η πρόβλεψη και x_j η πραγματική τιμή.

Mean Squared Error

Το μέσο τετραγωνικό σφάλμα (MSE) είναι αρκετά παρόμοιο με το μέσο απόλυτο σφάλμα, η μόνη διαφορά είναι ότι το MSE λαμβάνει το μέσο όρο του τετραγώνου της διαφοράς μεταξύ των αρχικών τιμών και των προβλεπόμενων τιμών. Το πλεονέκτημα του MSE είναι ότι είναι ευκολότερο να υπολογιστεί η διαβάθμιση, ενώ το μέσο απόλυτο σφάλμα απαιτεί περίπλοκα γραμμικά εργαλεία προγραμματισμού για τον υπολογισμό της διαβάθμισης.

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - x_j)^2$$

Καθώς, παίρνουμε το τετράγωνο του σφάλματος, η επίδραση των μεγαλύτερων σφαλμάτων γίνεται πιο έντονη, ως εκ τούτου, η ανάλυση του μοντέλο μπορεί τώρα να επικεντρωθεί περισσότερο για τα μεγαλύτερα σφάλματα.

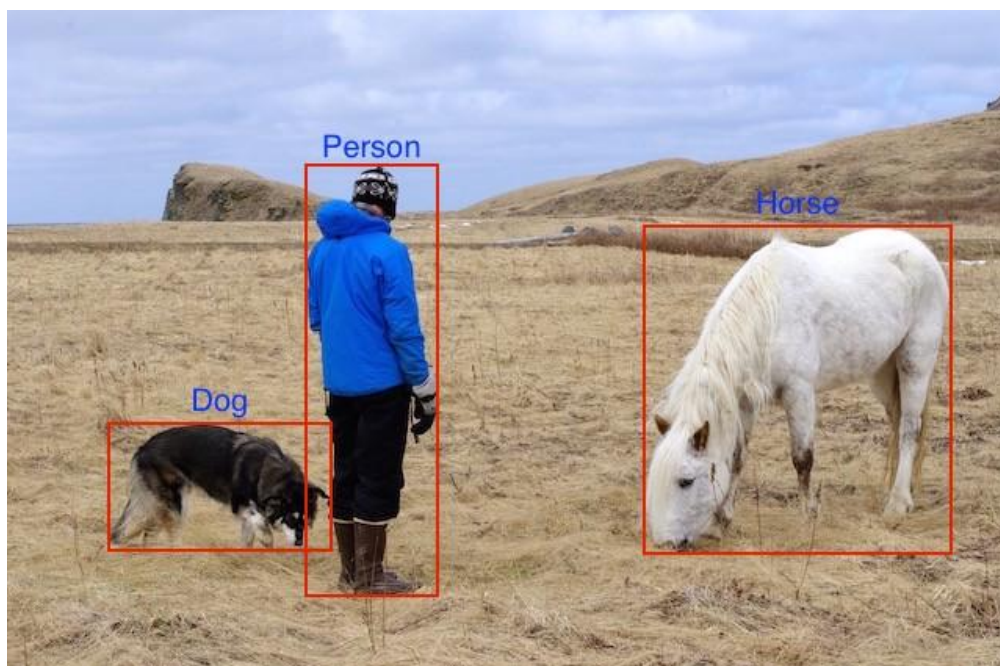
mAP

Αποτελεί ίσως τον πιο διαδεδομένο δείκτη αποτελεσματικότητας μοντέλων αναγνώρισης αντικειμένων. Η απόδοση ενός τέτοιου μοντέλου όπου δεν αρκεί η ταξινόμηση μιας εικόνας αλλά και η εύρεση της θέσης των αντικειμένων μέσα σε αυτήν, είναι πιο σύνθετη από δυαδικά μοντέλα και απαιτεί μερικές νέες έννοιες.

Για την αξιολόγηση των μοντέλων χρησιμοποιούνται οι πληροφορίες Ground Truth των αντικειμένων, που είναι γνωστές για datasets εκπαίδευσης δοκιμής και αξιολόγησης. Η πληροφορία ground truth περιλαμβάνει την εικόνα, τις κλάσεις των αντικειμένων και τα κουτιά που τα περιβάλλουν.

Για παράδειγμα στην παρακάτω εικόνα διαστάσεων 1000x800 η πληροφορία αλήθειας θα είναι

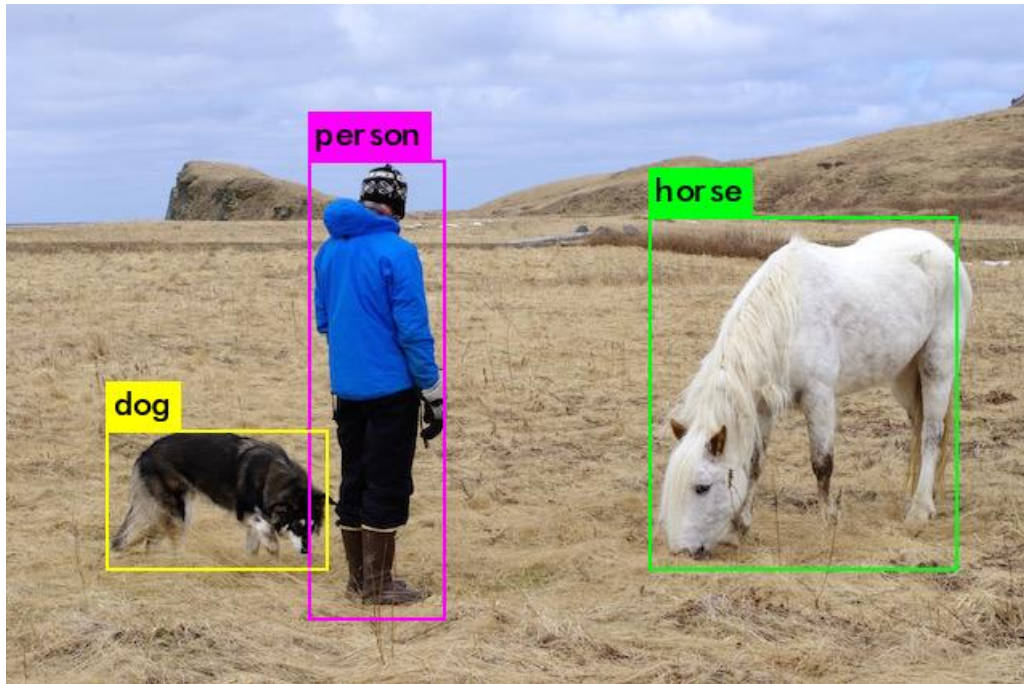
Class	X coordinate	Y coordinate	Box Width	Box Height
Dog	100	600	150	100
Horse	700	300	200	250
Person	400	400	100	500



Εικόνα 64

mAP – Ground truth

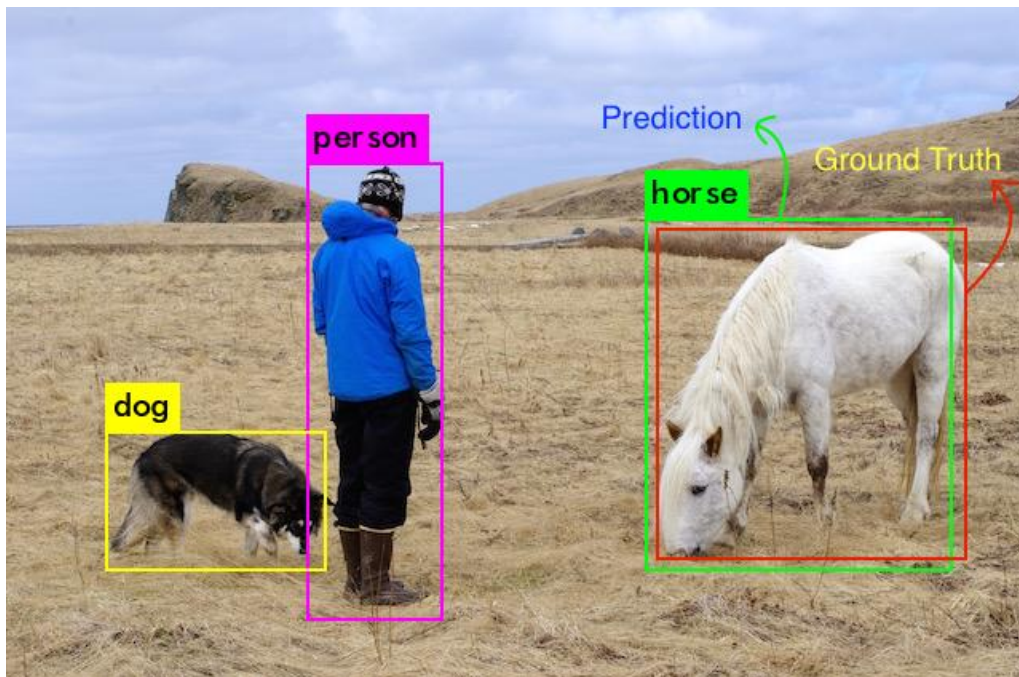
Τροφοδοτώντας μια εικόνα στο μοντέλο μας, αυτό θα επιστρέψει διάφορες προβλέψεις αλλά θα διατηρήσουμε μόνο αυτές που έχουν πιθανότητα πάνω από κάποιο όριο. Για παράδειγμα μας επιστρέφει το παρακάτω αποτέλεσμα



Εικόνα 65

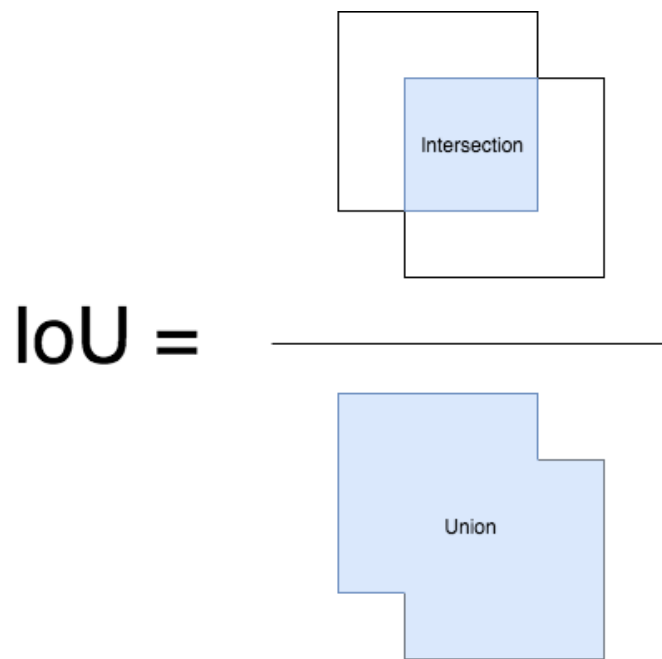
mAP - Πρόβλεψη

Αν και εκ πρώτης όψεως φαίνεται απόλυτη επιτυχία, η αλήθεια διαφέρει λίγο. Θα εισάγουμε τον όρο IoU – intersection over union δηλαδή επιφάνεια τομής. Ως IoU ορίζεται ο λόγος της αλληλοεπικάλυψης με την ένωση των πλαισίων από τη πρόβλεψη και αυτών του ground truth.



Εικόνα 66

mAP - IoU



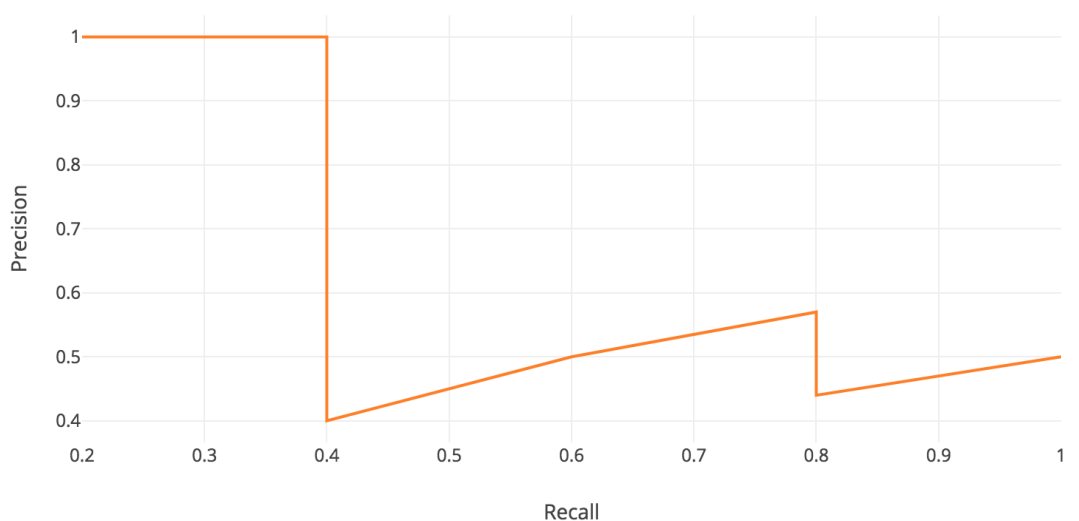
Εικόνα 67

mAP – IoU (2)

Για να μπορέσουμε να χρησιμοποιήσουμε και όπως προηγούμενες έννοιες που είδαμε ως δυαδικές, όπως το Recall , χρησιμοποιούμε το IoU μαζί με έναν δείκτη threshold. Για παράδειγμα αν θέσουμε το threshold 0.5 ,σημαίνει ότι για $\text{IoU} > 0.5$ θα θεωρούμε το δείγμα ως True Positive ενώ σε αντίθετη περίπτωση False Positive .

Για τον υπολογισμό των AP (average precision) για κάθε κλάση του μοντέλου δημιουργούμε το γράφημα precision – Recall και βρίσκουμε το εμβαδό του . Για το mAP συνδυάζουμε τα γραφήματα AP όλων των κλάσεων για να βγει ο μέσος όρος.

Πχ



Εικόνα 68

Average Precision

$$AP = \int_0^1 p(r)dr$$

Ωστόσο να σημειωθεί ότι δημιουργοί διαφόρων dataset ,προσαρμόζουν τους δείκτες εφαρμόζοντας μικρές παραλλαγές στον τρόπο υπολογισμού τους.

Παρακολούθηση αντικειμένου

(Object tracking)

Η παρακολούθηση αντικειμένων αναφέρεται στο κλάδο της μηχανικής όρασης όπου το αντικείμενο μας μετακινείται στο χώρο και πρέπει να έχουμε επίγνωση της θέσης του. Σε σχέση με την διαδοχική αναγνώριση αντικειμένου σε ένα βίντεο έχει κάποια προτερήματα.

Είναι πολύ πιο γρήγορη διαδικασία. Ο αλγόριθμος γνωρίζει τη προηγούμενη θέση του αντικειμένου αλλά και πως μοιάζει αυτό το αντικείμενο ,χωρίς να χρειάζεται να ανατρέξει σε ένα γενικευμένο νευρωνικό δίκτυο.

Επίσης κάποιοι αλγόριθμοι παρακολούθησης μπορούν να κάνουν πρόβλεψη της μελλοντικής θέσης του αντικειμένου και έτσι να συνεχίσουν την παρακολούθηση ακόμα και όταν έχει κρυφτεί πίσω από κάποιο εμπόδιο, κάτι που δεν μπορεί να γίνει με την αναγνώριση αντικειμένου.

Η παρακολούθηση είναι συνυφασμένη με την ταυτότητα του αντικειμένου. Αντίθετα η αναγνώριση δεν διατηρεί την ταυτότητα των αντικειμένων που βρίσκει.

Όλα αυτά βέβαια είναι σχετικά και αναφέρονται στους αλγόριθμους ως έχουν. Σε ένα ολοκληρωμένο πρόγραμμα αυτοί οι περιορισμοί μπορούν εν μέρει να καμφθούν. Το καλύτερο αποτέλεσμα έρχεται με συνδυασμό τεχνικών.

Με βάση τα παραπάνω ορίζονται δύο νέοι όροι που χρησιμοποιούνται στην παρακολούθηση αντικειμένων. Αυτοί είναι το «μοντέλο κίνησης» ,δηλαδή το ιστορικό ταχύτητας και κατεύθυνσης ενός αντικειμένου, και το «μοντέλο εμφάνισης» όπου κωδικοποιεί την εμφάνιση ενός αντικειμένου σε πολύ γρήγορο χρόνο και χρησιμοποιώντας ελάχιστη πληροφορία, πχ την εικόνα του από ένα και μόνο frame. Σε πολλούς αλγόριθμους το μοντέλο εμφάνισης είναι ένας ταξινομητής όπου εκπαιδεύεται on line. Μαθαίνει στον χρόνο εκτέλεσης του ποιο είναι το περιβάλλον και ποιο είναι το αντικείμενο ενδιαφέροντος.

Στο περιβάλλον OpenCV υπάρχουν κάποιοι αλγόριθμοι παρακολούθησης, τους οποίους και χρησιμοποιούμε στο κομμάτι επίδειξης. Παρακάτω υπάρχει περιγραφή τους.

Boosting tracker

Ο εν λόγω tracker είναι αρκετά παλιός. Βασίζεται στον αλγόριθμο Adaboost. Αφού δοθεί από τον χρήστη το περίγραμμα που περιέχει το αντικείμενο, δημιουργούνται ένα θετικό παράδειγμα αλλά και αρνητικά από την υπόλοιπη εικόνα που δεν περικλείεται στο αρχικό περίγραμμα. Σε κάθε νέο frame ο ταξινομητής συγκρίνει τα προηγούμενα θετικά παραδείγματα με τα γειτονικά pixels και βρίσκει τη νέα θέση του αντικειμένου. Από τη νέα θέση εξάγει νέο θετικό παράδειγμα όπου το χρησιμοποιεί στο επόμενο frame κ.ο.κ . Αρνητικό χαρακτηριστικό του είναι ότι πάντα επιστρέφει την καλύτερη του πρόγνωση χωρίς να χρησιμοποιεί κάποιο threshold , έτσι δεν γνωρίζουμε πότε έχει αποτύχει .

MIL tracker

Βασίζεται στον Boosting με τη διαφορά ότι από το πρώτο frame λαμβάνει πολλαπλά θετικά παραδείγματα, από γειτονικές περιοχές του περιγράμματος. Έχει βρεθεί ότι έχει καλύτερα αποτελέσματα ενώ δεν επιστρέφει αποτέλεσμα αν χαθεί η παρακολούθηση, ωστόσο όχι αξιόπιστα.

KCF tracker

Ή Kernelized Correlation Filters. Αποτελεί την μαθηματική εξέλιξη του MIL. Θεωρείται από τους πιο σταθερούς και αξιόπιστους.

TLD tracker

TLD σημαίνει tracking, learning and detection. Κάνει αρκετά σύνθετους υπολογισμούς και διορθώνει το μοντέλο εμφάνισης συνεχώς. Επίσης δημιουργεί και νέα μοντέλα σε διαφορετικές κλίμακες και τα χρησιμοποιεί με μερική επιτυχία όταν το αντικείμενο μας πλησιάζει ή απομακρύνεται. Ψάχνει για τη νέα θέση του αντικειμένου σε μεγαλύτερη έκταση και όχι μόνο στα γειτονικά ριχελ του αρχικού μας περιγράμματος. Αν και σε κάποιες εφαρμογές θα μπορούσε να έχει θετικά αποτελέσματα, γενικά είναι ασταθής αλγόριθμος.

Medianflow tracker

Αυτός ο αλγόριθμος αξιοποιεί περισσότερο το μοντέλο κίνησης. Δημιουργεί εικονικές τροχιές του αντικειμένου προς διάφορες κατευθύνσεις και προσπαθεί να τις επαληθεύσει στη πορεία. Λειτουργεί καλά για ομαλές και προβλέψιμες τροχιές, ενώ αντιλαμβάνεται και τότε έχει αποτύχει.

MOSSE tracker

Χρησιμοποιεί φίλτρα συσχέτισης με πολύ καλά αποτελέσματα σε αλλαγές φωτισμού και κλίμακας.

CSRT tracker

Ή Discriminative Correlation Filter with Channel and Spatial Reliability. Χρησιμοποιεί χαρακτηριστικά από HOG και Colornames για τη δημιουργία χάρτη χαρακτηριστικών τα οποία και χρησιμοποιεί για την παρακολούθηση.

Πειραματικό μέρος

Στην εφαρμογή έχουν ενσωματωθεί οι αλγόριθμοι παρακολούθησης (για συντομία tracker) που προαναφέραμε καθώς και ο αλγόριθμος αναγνώρισης αντικειμένων Yolo v3 (για συντομία detector). Οι συνδυασμοί και οι επιλογές που θα μπορούσαν να αναπτυχθούν είναι απεριόριστες. Η ανάπτυξη της εφαρμογής έγινε έως το σημείο όπου δεν θα είναι ασκόπως περίπλοκη και με γνώμονα την διαθέσιμη υπολογιστική ισχύ τόσο κατά την ανάπτυξη της όσο και για την εκτέλεση της. Ιδανικά φυσικά θα ήταν προτιμότερη η χρήση μόνο αλγόριθμου αναγνώρισης αντικειμένων μαζί με τεχνικές ταυτοποίησης και καταμέτρησης αντικειμένων ωστόσο κάτι τέτοιο θα ήταν εξαιρετικά αργό και μας ωθεί στην αναζήτηση εναλλακτικών επιλογών.

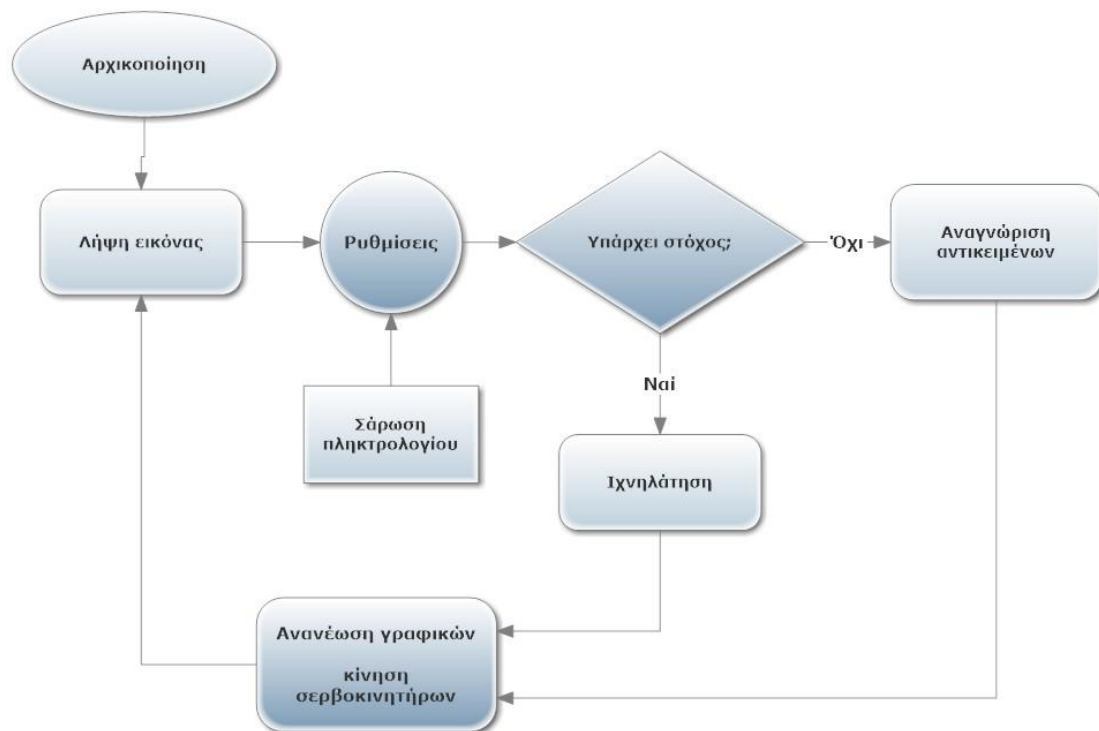
Η ροή του προγράμματος είναι η εξής. Το πρόγραμμα ξεκινάει μην έχοντας στόχο οπότε και σαρώνει την εικόνα για εύρεση ενός στόχου. Το μοντέλο που εκπαίδευσα μπορεί να αναγνωρίσει άρματα μάχης καθώς και τετρακόπτερα drones. Όταν βρεθεί στόχος ,αυτός παρακολουθείτε με κάποιο αλγόριθμο παρακολούθησης μέχρις ότου χαθεί , οπότε ξαναρχίζει η σάρωση για αναγνώριση αντικειμένου.

Μεταξύ του detector και του tracker κατά την εναλλαγή , ανταλλάσσονται στοιχεία θέσης αντικειμένου καθώς και της κλάσης του (άρμα ή drone) ως ένα στοιχείο ταυτοποίησης.

Ο αλγόριθμος αναγνώρισης επιστρέφει λίστα ως δέκα αντικειμένων που έχει αναγνωρίσει , αλλά στον tracker τροφοδοτείτε αυτός με το μεγαλύτερο ποσοστό σιγουριάς (confidence score) που ανήκει στην ίδια κλάση με το τελευταίο αντικείμενο που είχε κλειδώσει. Όλα αυτά βέβαια είναι μερικός παραμετροποίησημα .

Εναλλακτικά μπορεί να παρακαμφθεί ο detector και ο χρήστης να εισάγει απευθείας στον tracker με χρήση του cursor ένα οποιοδήποτε αντικείμενο βρίσκεται στο οπτικό πεδίο της κάμερας – με το πλήκτρο m(anual) .

Ροή προγράμματος

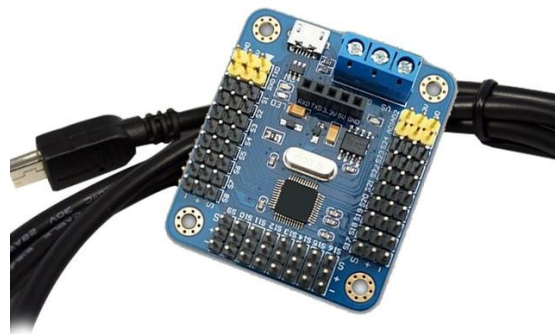


Εικόνα 69

Ροή προγράμματος

Υλικό συστήματος

Η εφαρμογή του πειραματικού μέρους δημιουργήθηκε σε γλώσσα Python 3.7.4 με εγκατεστημένη τη βιβλιοθήκη OpenCV 4.1.0 . Για την λειτουργία της χρειάζεται μια camera usb (ή ενσωματωμένη σε laptop), η οποία ανιχνεύεται αυτόματα , και προαιρετικά ένας adaptor usb to pwm . Εμπορικά μπορεί να βρεθεί ως «16CH Channel Way Servo Controller» . Παρέχει 16 εξόδους pwm για χρήση σε servo ή οτιδήποτε άλλο. Ο χειρισμός του είναι απλός καθώς γίνεται με αποστολή σειριακών εντολών μέσω την εικονικής σειριακής θύρας που προσθέτει στο σύστημα μας.



Εικόνα 70 Servo controller

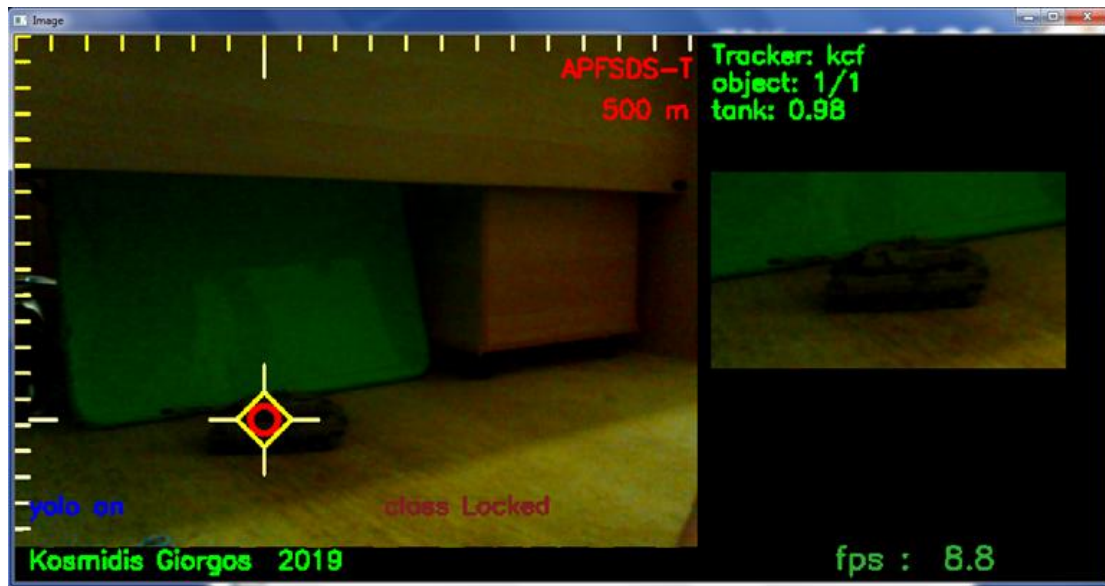
Στην παρούσα εφαρμογή χρησιμοποιούνται τα 2 κανάλια για χειρισμό pan-tilt ενός βραχίονα δύο αξόνων. Στον βραχίονα μπορεί να στηριχθεί η κάμερα ή κάποιο άλλο ενδεικτικό πχ φακός. Στην πρώτη περίπτωση με την κίνηση της κάμερας ο στόχος παραμένει κλειδωμένος στο κέντρο της οθόνης, ενώ αν η κάμερα είναι ανεξάρτητη ο στόχος μετακινείται σε όλο το οπτικό πεδίο της.

Με μικρή τροποποίηση του προγράμματος μπορεί να χρησιμοποιηθεί και η κάμερα υψηλής ανάλυσης /υψηλών επιδόσεων που συνδέεται στην ειδική θύρα του raspberry. Ωστόσο λόγω την μικρής υπολογιστικής του ισχύος, το raspberry (3B+) εγκαταλείφθηκε και όλες οι δοκιμές έγιναν σε laptop /desktop .

Περιβάλλον διασύνδεσης χρήστη

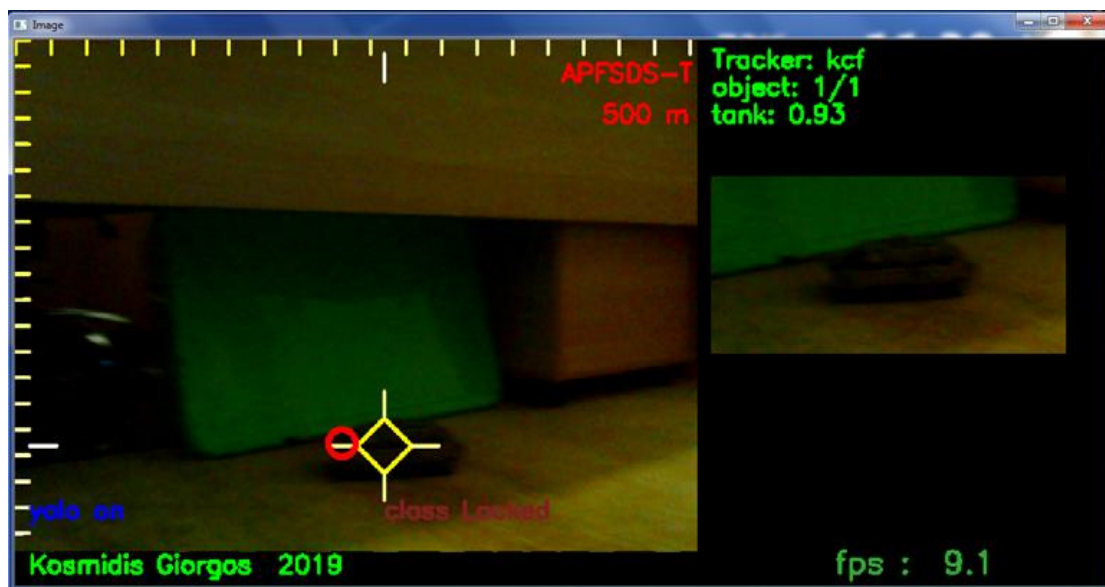
Κατά την εκτέλεση της εφαρμογής ανοίγουν δύο παράθυρα. Ένα παράθυρο κελύφους όπου καταγράφει διάφορες λειτουργίες που συμβαίνουν και έχει χαρακτήρα αποσφαλμάτωσης/πληροφόρησης και το κύριο γραφικό παράθυρο. Στην γραφική απεικόνιση υπάρχουν δύο κυρίαρχες περιοχές.

Αριστερά είναι εικόνα που μας δίνει η κάμερα. Στην εικόνα υπάρχει μια κλίμακα στον οριζόντιο και μια κλίμακα στον κάθετο άξονα καθώς και δύο δρομείς που αποτυπώνουν πάνω στις κλίμακες τη θέση του στόχου. Πάνω στην εικόνα υπάρχουν 4 ενδείξεις και το σκοπευτικό νήμα. Όταν υπάρχει κλειδωμένος στόχος εμφανίζεται πλησίον του νήματος και ένας κόκκινος κύκλος που μας δείχνει τη προπορεία που θα πρέπει να έχει το πυροβόλο.



Εικόνα 71

GUI – Ακίνητος στόχος



Εικόνα 72

GUI – Κινούμενος στόχος

Οι 4 ενδείξεις είναι :

Τύπος πυρομαχικού – αλλάζει με το πλήκτρο a(mmo) . Μπορεί να πάρει 4 προεπιλεγμένες τιμές. Αυτές είναι Laser ,APFSDS-T , HEAT-T και WP . Ο τύπος πυρομαχικού επηρεάζει τη προπορία του πυροβόλου καθώς το βλήμα κινητικής ενέργειας (APFSDS-T) κινείται με μεγαλύτερη ταχύτητα από τα εκρηκτικά , ενώ η σκόπευση με laser μηδενίζει τη προπορία.

Απόσταση. Εδώ φαίνεται η απόσταση του στόχου η οποία επίσης επηρεάζει τη προπορία του πυροβόλου. Για αξιόπιστη μέτρηση θα ήθελε εξωτερικό αισθητήρα ή χρήση δεύτερης κάμερας. Αν και υπήρχε η σκέψη για δεύτερη κάμερα και έγιναν τροποποιήσεις στο πρόγραμμα για να την υποστηρίξει , η ανεπαρκής υπολογιστική ισχύς το απέτρεψε. Τελικά γίνεται υπολογισμός από τη διάσταση του αντικειμένου με μια κάμερα, ενώ ο χρήστης μπορεί και χειροκίνητα να αυξομειώσει την απόσταση στόχου με τα πλήκτρα +/- .

Μεταπτυχιακή Διπλωματική Εργασία, Γεώργιου Κοσμίδη , AM IES-0028

Χρήση YOLO – αλλάζει με το πλήκτρο y. Μπορεί να πάρει 3 τιμές.

Yolo on όταν γίνεται χρήση του αλγόριθμου αναγνώρισης αντικειμένου και όταν βρεθεί ένα, γυρνάει σε αλγόριθμο παρακολούθησης.

Yolo only όταν γίνεται μόνο χρήση αλγόριθμου αναγνώρισης αντικειμένου.

Ανενεργό όταν λειτουργεί μόνο ως tracker.

Class Locked – αλλάζει με το πλήκτρο k(leidoma) . Όταν είναι ενεργό , ο αλγόριθμος αναγνώρισης αντικειμένων δίνει προτεραιότητα στην επιστροφή αντικειμένου ίδιας κλάσης με αυτή που είχε αναγνωριστεί τελευταία.

Στο δεξί μέρος αναγράφεται :

Tracker - αλλάζει με το πλήκτρο t(racker) . Ο αλγόριθμος παρακολούθησης που χρησιμοποιείται.

Object. Μας αναφέρει το σύνολο των αντικειμένων που βρέθηκαν τελευταία από τον detector καθώς και ποιο έχει επιλεγεί ως στόχος. Αν και αρχικά τα αναγνωρισμένα αντικείμενα τίθενται με σειρά σιγουριάς, με την πάροδο του χρόνου , το αντικείμενο που παρακολουθούμε ενδέχεται να έχει αλλάξει confidence score και σειρά προτεραιότητας . Αυτή η αλλαγή μπορεί να φανεί εδώ.

Λίστα αντικειμένων. Κάτω από το object i/x αναγράφεται η λίστα με τα αντικείμενα μαζί με στο confidence score. Το αντικείμενο στόχος θα έχει πιο έντονο χρωματισμό γραμματοσειράς.

Εικόνα στόχου. Παρακάτω φαίνεται σε καλύτερη ανάλυση (πριν την υποβάθμιση της για την επεξεργασία που ακολουθεί) το αντικείμενο στόχος σε πρώτο πλάνο.

Τέλος αναγράφεται το framerate.



Εικόνα 73

GUI – 2 αναγνωρισμένα αντικείμενα

Συνοπτικά τα πλήκτρα ελέγχου είναι :

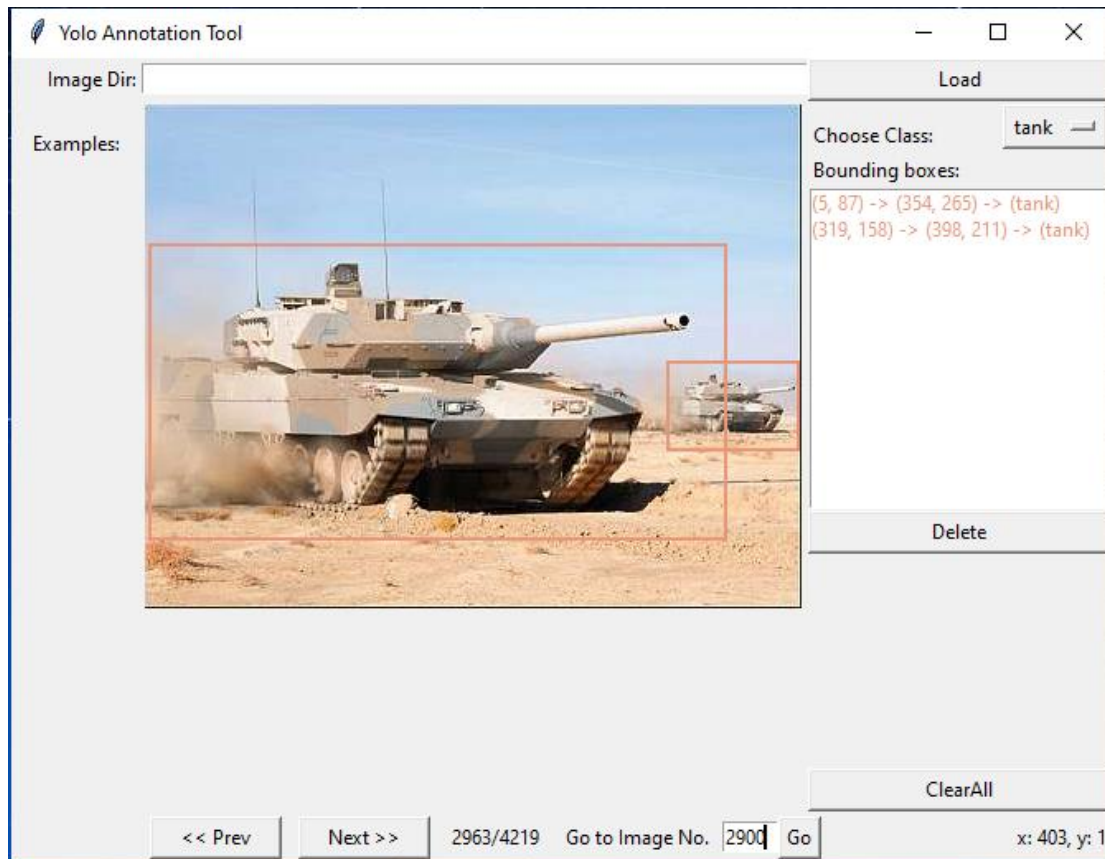
- a (αλλαγή τύπου πυρομαχικών)
- k (κλείδωμα στόχου)
- γ (αλλαγή λειτουργίας)
- t (αλλαγή tracker)
- m (χειροκίνητη στόχευση)
- i (εκτέλεση αναγνώρισης)
- + (αύξηση απόστασης ανά 100μ)
- (μείωση απόστασης ανά 100μ)
- c (επόμενη σειριακή θύρα)
- r (εκτέλεση detector κάθε 100 frames)
- d (αυτόματος υπολογισμός απόστασης)

Εκπαίδευση Μοντέλου

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν 2883 φωτογραφίες από drones οι οποίες ήταν ήδη χαρακτηρισμένες (annotated) και διαθέσιμες στο github και 1226 φωτογραφίες από άρματα leopard διαφόρων τύπων , m48 και m60. Το υλικό βρέθηκε σε μεγάλο βαθμό στο διαδίκτυο, κατά ένα μέρος είναι φωτογραφίες που τράβηξα από άρμα μινιατούρα και κατά ένα μέρος από υλικό που μου στάλθηκε από τον διοικητή του ΚΕΘ ταξίαρχο κ.Μαυροειδή σε μορφή βίντεο το οποίο και μετέτρεψα σε φωτογραφίες.

Για να χρησιμοποιηθούν οι φωτογραφίες ως υλικό εκπαίδευσης , θα πρέπει να συνοδεύονται από ένα αρχείο όπου περιλαμβάνει τη θέση του αντικειμένου. Αυτό γίνεται δίνοντας τις συντεταγμένες x_1, y_1 και x_2, y_2 του πλαισίου που το περιλαμβάνει μέσα στην εικόνα . Σε κάθε εικόνα μπορούν να υπάρχουν πολλά αντικείμενα και μάλιστα διαφορετικής κλάσης.

Το εργαλείο που χρησιμοποιήθηκε για τη δημιουργία των αρχείων αυτών , ήταν το Yolo Annotation Tool. Ένα στιγμιότυπο του φαίνεται παρακάτω.



Εικόνα 74

Yolo Annotation Tool

Ο χαρακτηρισμός όλων των εικόνων γίνεται χειροκίνητα και είναι μια χρονοβόρα διαδικασία. Υπάρχουν και έτοιμες συλλογές με χαρακτηρισμένα αντικείμενα, μια από αυτές ήταν η συλλογή με drone που χρησιμοποιήθηκε. Χρειάζεται πάντως προσοχή καθώς περιείχε μερικά λάθη τα οποία διορθώθηκαν πριν χρησιμοποιηθεί.

Google Colab

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε το framework του Darknet. Η εκπαίδευση ήταν αδύνατη σε οικιακό υπολογιστή καθώς μετά από μερικές εποχές εκπαίδευσης το framework σταμάταγε να λειτουργεί λόγω ανεπάρκειας μνήμης και άλλα προβλήματα που σχετίζονταν με την απόδοση του επεξεργαστή. Πάντως και να λειτουργούσε, οι χρόνοι ήταν απογοητευτικοί. Θέλοντας περίπου 3 ώρες για 50 επαναλήψεις (iterations) κατά την εκπαίδευση μπορούμε να αναλογιστούμε ότι για 10.000 επαναλήψεις θα χρειαζόνταν 300 ώρες.

Τελικά επιλέχθηκε η λύση Google Colab. Πρόκειται για εικονική μηχανή όπου προσφέρεται διαδικτυακά από την google για τέτοιες περιπτώσεις. Τρέχει CLI linux με ενσωματωμένη την Pythοn και πλήρη ελευθερία να εγκαταστήσεις ότι άλλο θες (όπως το darknet). Μάλιστα προσφέρει CUDA GPU με 12 GB μνήμη. Σε αυτό το σύστημα χρειάζονται λιγότερες από 40 ώρες για την εκπαίδευση 10.000 επαναλήψεων. Το αρνητικό είναι ότι είναι διαθέσιμο μονάχα για 12 ώρες εργασίας. Επίσης το ότι είναι διαδικτυακό σημαίνει ότι κάθε φορά που

το εκκινείς θα πρέπει να υπολογίζεις ότι μια ώρα τουλάχιστο θα χαθεί σε μεταφορές αρχείων κτλ. Ένα δεύτερο μειονέκτημα που αντιμετώπισα ήταν ότι σου περιορίζει τους πόρους αν αντιληφθεί ότι το χρησιμοποιείς εντατικά. Έτσι μετά από δύο-τρία δωδεκάωρα είναι πολύ πιθανό να μην σου διαθέτει GPU στην εικονική μηχανή που χρησιμοποιείς, για κάποια 24ωρα. Δεδομένο ότι μέχρι τη δημιουργία του τελικού μοντέλου γίναν άλλες 4 δοκιμές με διαφορετικά dataset και ρυθμίσεις, το πρόβλημα της αναμονής ήταν μεγάλο.

Λέγοντας επανάληψη, εννοούμε τη διαδικασία εκπαίδευσης ενός batch αρχείων, δηλαδή μιας δέσμης 64 αρχείων. Αυτό είναι το νούμερο εικόνων που επιλέξαμε για κάθε υποδιαίρεση των 3798 αρχείων που χρησιμοποιήσαμε για την εκπαίδευση (από τα 4220 – τα υπόλοιπα για δοκιμή). Αυτό γίνεται για να μπορέσει να διαχειριστεί το σύστημα μας το μεγάλο όγκο δεδομένων. Επομένως κάθε εποχή εκπαίδευσης (εκπαίδευση όλων των δεδομένων) απαιτεί 60 επαναλήψεις.

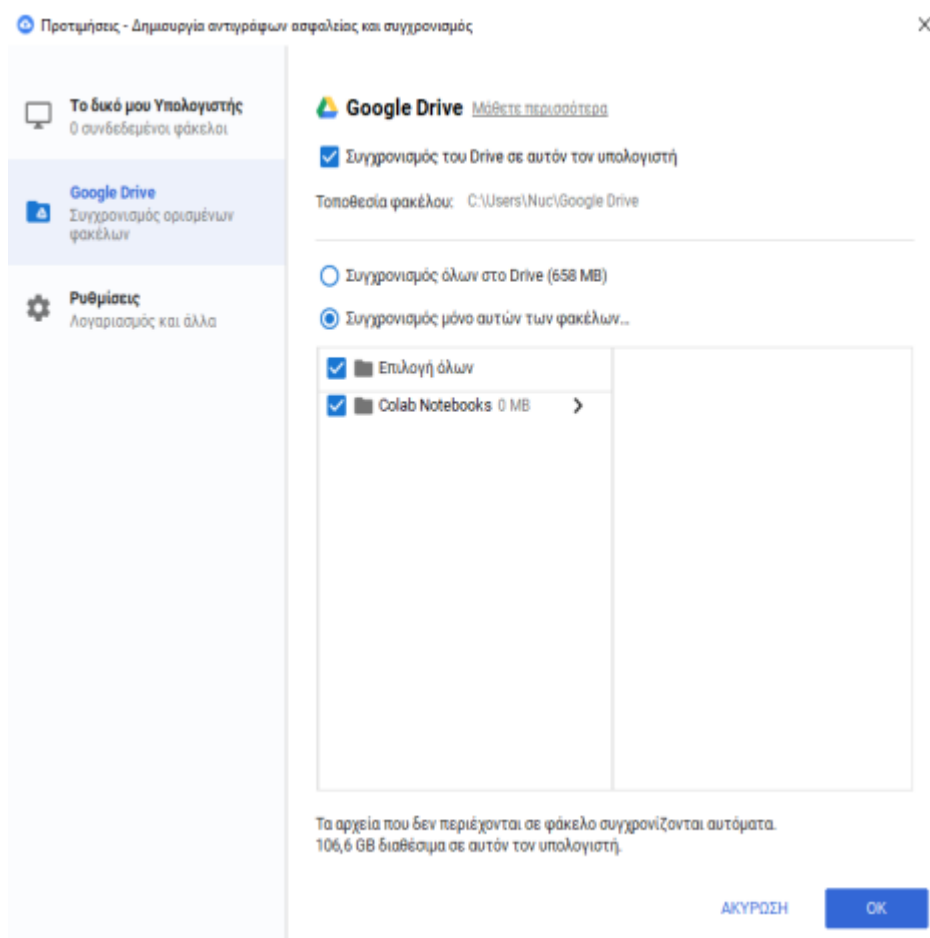
Παρακάτω θα δούμε τον τρόπο χρήσης του για τις δοκιμές μας.

Καταρχάς ρυθμίζουμε το Colab notebook που θα χρησιμοποιήσουμε ώστε να χρησιμοποιεί τη διαθέσιμη από τη google GPU.

Menu > Runtime > Configure Runtime Type και επιλέγουμε GPU

Ενεργοποιούμε το συγχρονισμό του google drive με έναν τοπικό φάκελο τον οποίο δημιουργούμε και ονομάζουμε πχ Yolo train μέσα σε ένα φάκελο Colab Notebooks (τον οποίο επίσης δημιουργήσαμε).





Τώρα μπορούμε να εκτελέσουμε τον παρακάτω κώδικα μέσα στο Colab.

```
# Συνδέουμε το φάκελο που διατηρούμε τοπικά στο google
drive με το Colab Notebook για εύκολο διαμοιρασμό αρχείων.
Ο φάκελος αυτός θα φαίνεται στο notebook στο φάκελο

from google.colab import drive
drive.mount('/content/gdrive')
```

Θα πρέπει να κατεβάσουμε από την nvidia τις βιβλιοθήκες νευρωνικών δικτύων που παρέχει για χρήση τους με τις gru. Πριν το κάνουμε αυτό θα πρέπει να δούμε πια έκδοση CUDA χρησιμοποιεί το Colab notebook .

```
! /usr/local/cuda/bin/nvcc --version
```

Η απόκριση του συστήματος είναι

Μεταπτυχιακή Διπλωματική Εργασία, Γεώργιου Κοσμίδη , AM IES-0028

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005–2018 NVIDIA Corporation
Built on Sat_Aug_25_21:08:01_CDT_2018
Cuda compilation tools, release 10.0, V10.0.130
```

Οπότε θα πρέπει να κατεβάσουμε το cuDNN για CUDA 10.

Τοπικά στο φάκελο *Yolo train* δημιουργούμε το φάκελο *cuDNN* και κατεβάζουμε από την nvidia το αρχείο *cuda-10.0-linux-x64-v7.5.0.56.tgz*

(https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v7.5.0.56/prod/10.0_20190219/cudnn-10.0-linux-x64-v7.5.0.56.tgz)

Κατόπιν το αντιγράφουμε στο Colab Notebook.

```
!tar -xzf "gdrive/My Drive/Colab Notebooks/Yolo train/cuDNN/cudnn-10.0-
linux-x64-v7.5.0.56.tgz" -C /usr/local/
!chmod a+r /usr/local/cuda/include/cudnn.h
```

Ακολουθεί η εγκατάσταση του darknet στο vm. Μετά κατεβάζουμε το δίκτυο

```
!git clone https://github.com/kriyeng/darknet/
%cd darknet
!git checkout feature/google-colab
!make
!wget https://pjreddie.com/media/files/darknet53.conv.74
```

darknet53.conv.74 .

Το περιβάλλον εργασίας έχει ολοκληρωθεί.

Τώρα θα μεταφέρουμε το dataset μας με τις ρυθμίσεις μαζί στο Colab Netbook για να αρχίσει η εκπαίδευση.

Τοποθετούμε τον φάκελο(*tank_e*) που περιέχει το dataset και τις ρυθμίσεις στο google drive στο φάκελο *Yolo train* που είχαμε δημιουργήσει. Κατόπιν το αντιγράφουμε στο vm μέσα στο φάκελο *darknet*. Προσέχουμε να εκτελέσουμε την εντολή βρισκόμενοι μέσα στο φάκελο *darknet* του vm.

```
!cp -r "/content/gdrive/My Drive/Colab Notebooks/Yolo train/tank_e" ./
```

Μεταπτυχιακή Διπλωματική Εργασία, Γεώργιου Κοσμίδη , AM IES-0028

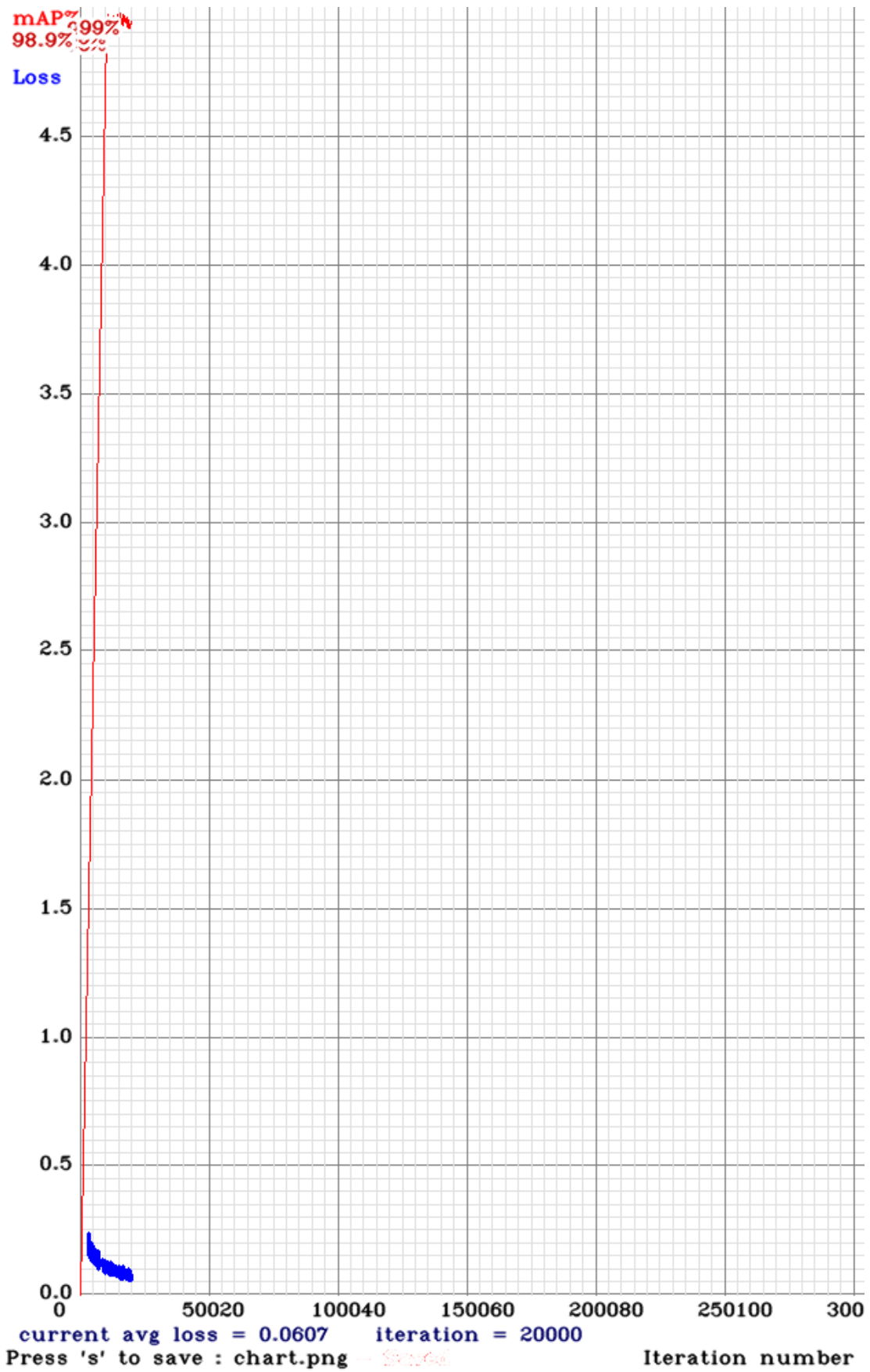
```
!./darknet detector train "/content/gdrive/My Drive/Colab Notebooks/Yolo train/tank_e/cfg/tank_e.data"  
"/content/gdrive/My Drive/Colab Notebooks/Yolo train/tank_e/cfg/yolov3.cfg" /darknet/darknet53.conv.74 > mylog.txt -  
dont_show -map
```

Ή στην περίπτωση που θέλουμε να συνεχίσουμε την εκπαίδευση έχοντας ήδη αρχείο με βάρη:

```
!./darknet detector train "/content/gdrive/My Drive/Colab Notebooks/Yolo train/tank_e/cfg/tank_e.data"  
"/content/gdrive/My Drive/Colab Notebooks/Yolo train/tank_e/cfg/yolov3.cfg"  
/darknet/tank_e/cfg/backup/yolov3_5l_last.weights /darknet/darknet53.conv.74 > mylog.log -dont_show -map
```

Από το dataset tank_e δημιουργήθηκαν δύο μοντέλα. Το ένα χρησιμοποίησε τις προκαθορισμένες ρυθμίσεις που συνοδεύουν το framework (106 layers) και το άλλο, λίγο μεγαλύτερο, χρησιμοποιεί επιπλέον blocks layer με έμφαση στην αναγνώριση αντικειμένων σε περισσότερες κλίμακες (130 layers). Αναλυτικά τα layers κάθε σετ ρυθμίσεων υπάρχουν στο τέλος της εργασίας.

Δυστυχώς λόγω της περιορισμένης πρόσβασης σε σύστημα εκπαίδευσης, κάποιες ρυθμίσεις που θα έπρεπε να γίνουν κατά τη διάρκεια της εκπαίδευσης και έχουν σχέση με γραφικές παραστάσεις απόκρισης του μοντέλου, δεν έγιναν σωστά και τα αποτελέσματα δεν είναι ιδανικά. Επίσης η επανεκπαίδευση μόνο και μόνο για να παρθούν καλύτερες παραστάσεις θα ήταν χρονικά ασύμφορη. Παρακάτω βλέπουμε την συνάρτηση απώλειας του μοντέλου με τις προκαθορισμένες ρυθμίσεις.



Εικόνα 75

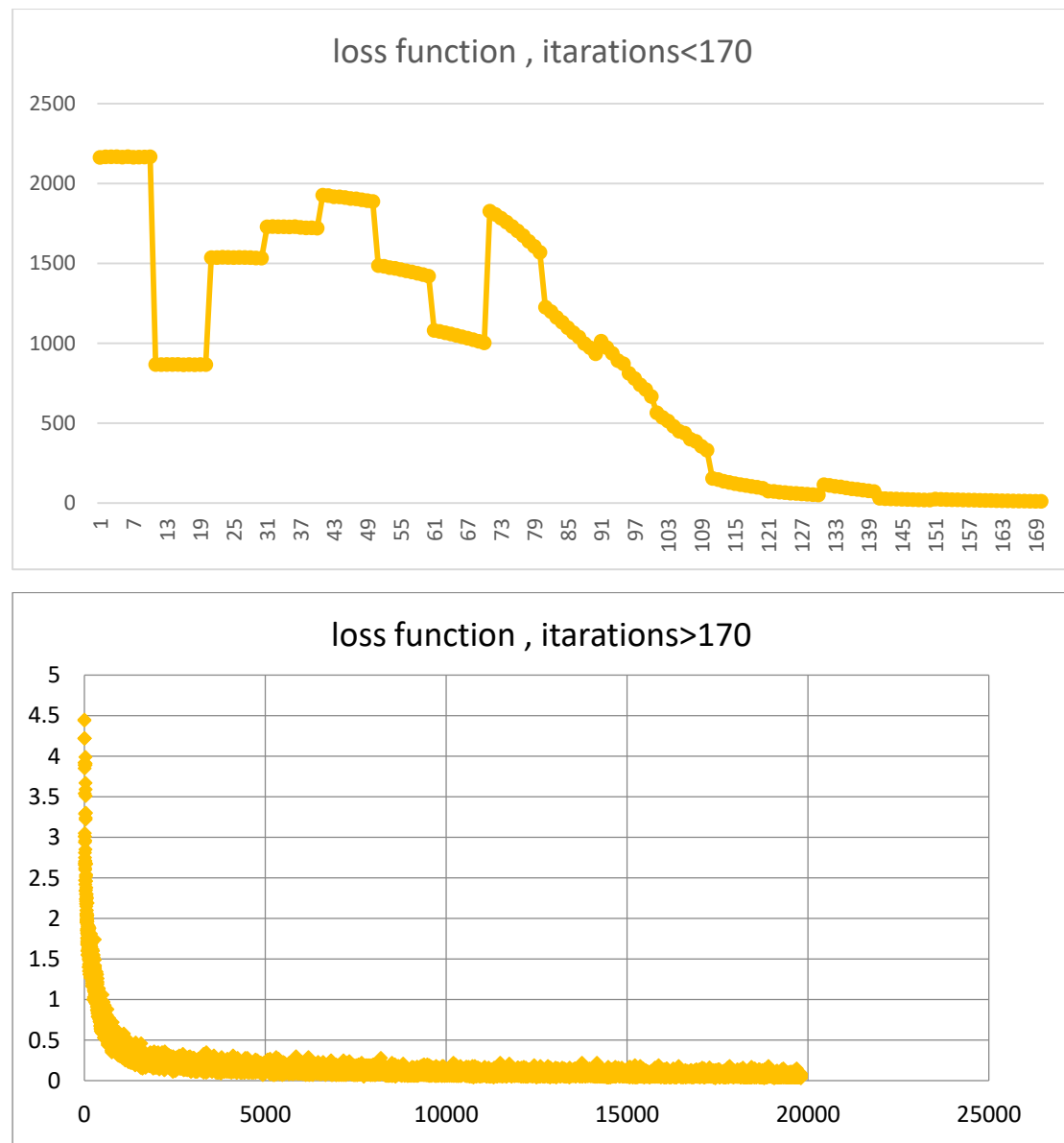
loss function , προκαθορισμένες ρυθμίσεις 106 layers

Βλέπουμε ότι στις 20000 επαναλήψεις που έγιναν (333 εποχές) η αποτυχία περιορίστηκε στο 0.061 .

Για το μεγαλύτερο μοντέλο δεν εξάχθηκε η συνάρτηση από το framework την ώρα της εκπαίδευσης αλλά μόνο η τελική τιμή η οποία ήταν για τις 200 εποχές 0.053 .

Ωστόσο πέρα από τα διάφορα γραφήματα που θα μπορούσε να μας δώσει το framework, έγινε ανάκτηση των logs και από εκεί δημιουργήθηκαν τα loss functions (MSE) .

Για το προκαθορισμένο μοντέλο :

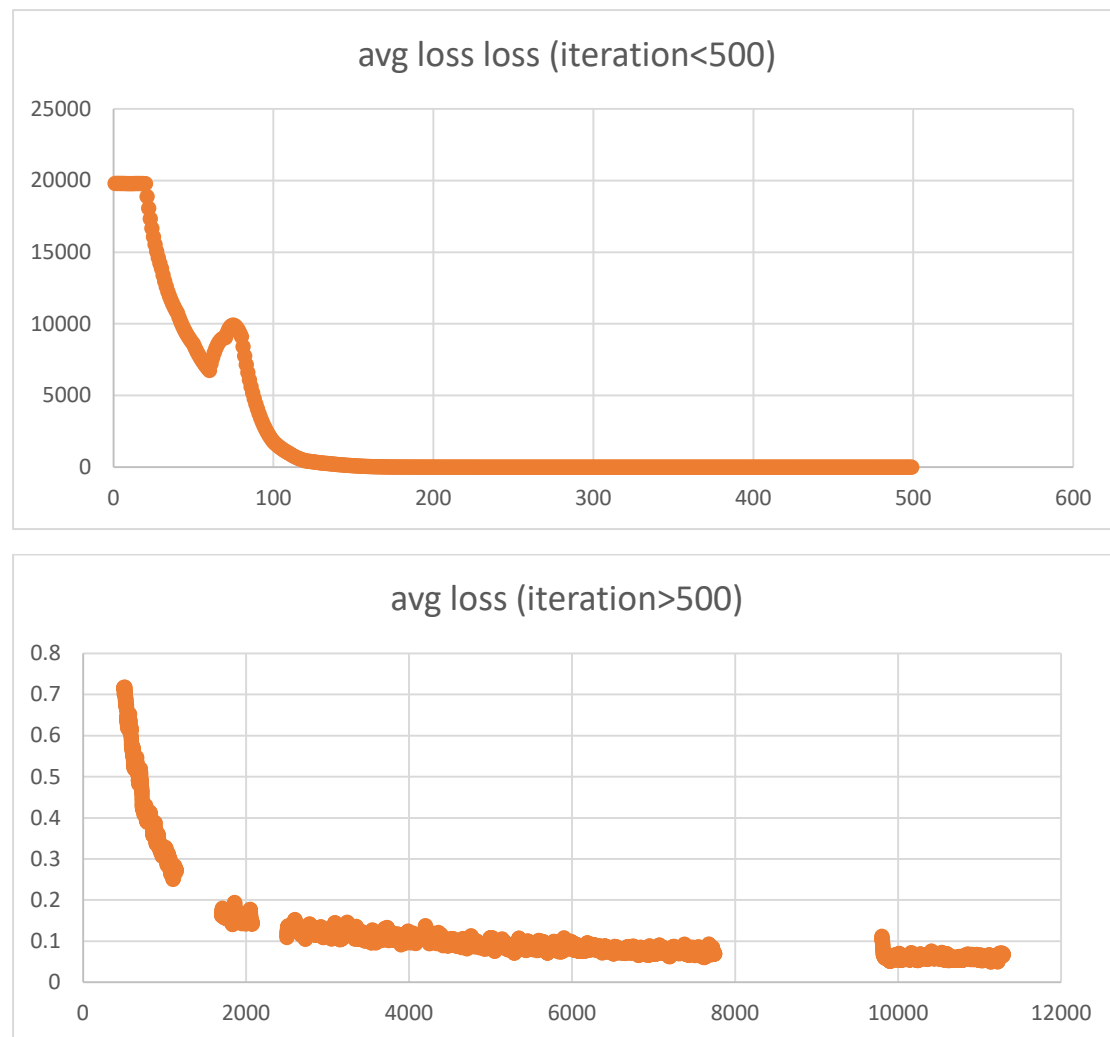


Εικόνα 76

loss function 106 layers

Με [mAP@0.5](#) = 98.85

Για το μεγαλύτερο μοντέλο :



Εικόνα 77

loss function 130 layers

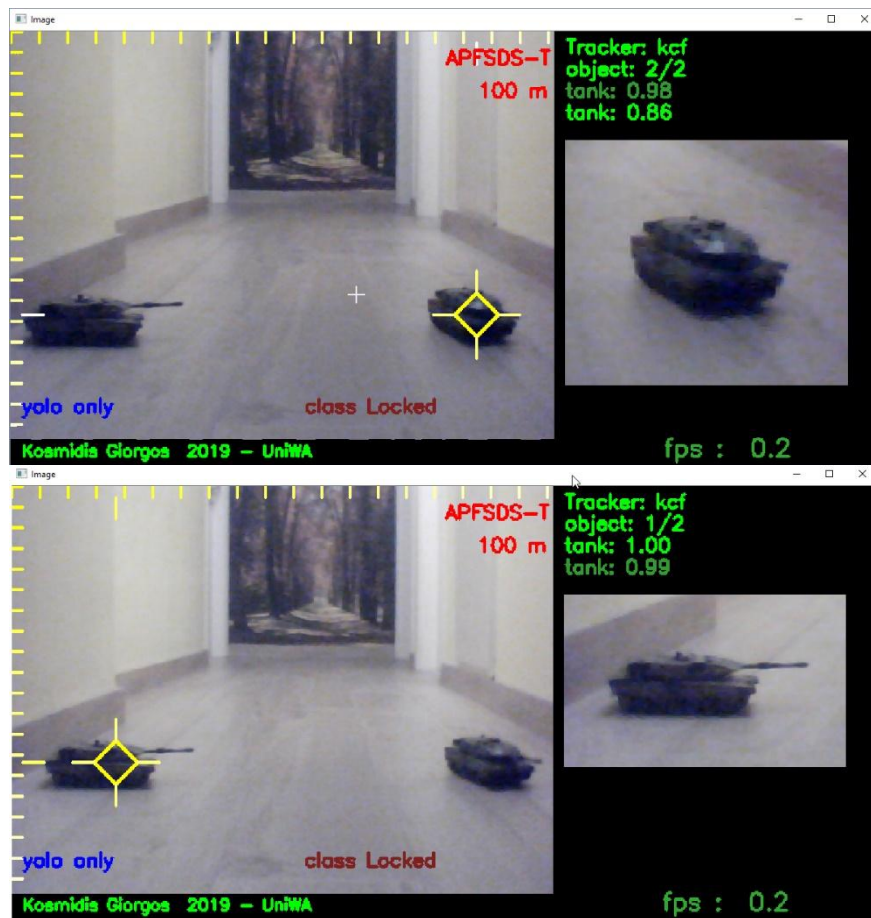
Με [mAP@0.5](#) = 98.89

Δοκιμή μοντέλου

Για τη δοκιμή του μοντέλου χρησιμοποίησα 2 παιχνίδια -αντίγραφα αρμάτων – σε αποστάσεις από 1 έως 5 μέτρα , σε χώρο και με φωτισμό που δεν υπήρχε στο υλικό εκπαίδευσης. Σε κάθε δοκιμή πάρθηκε ένα χαρακτηριστικό στιγμιότυπο της εφαρμογής όπου φαίνεται το σκορ σιγουριάς των αντικειμένων. Αντικείμενα με σκορ χαμηλότερο του 15% απορρίπτονταν αμέσως από το πρόγραμμα για να αποφευχθούν false positives

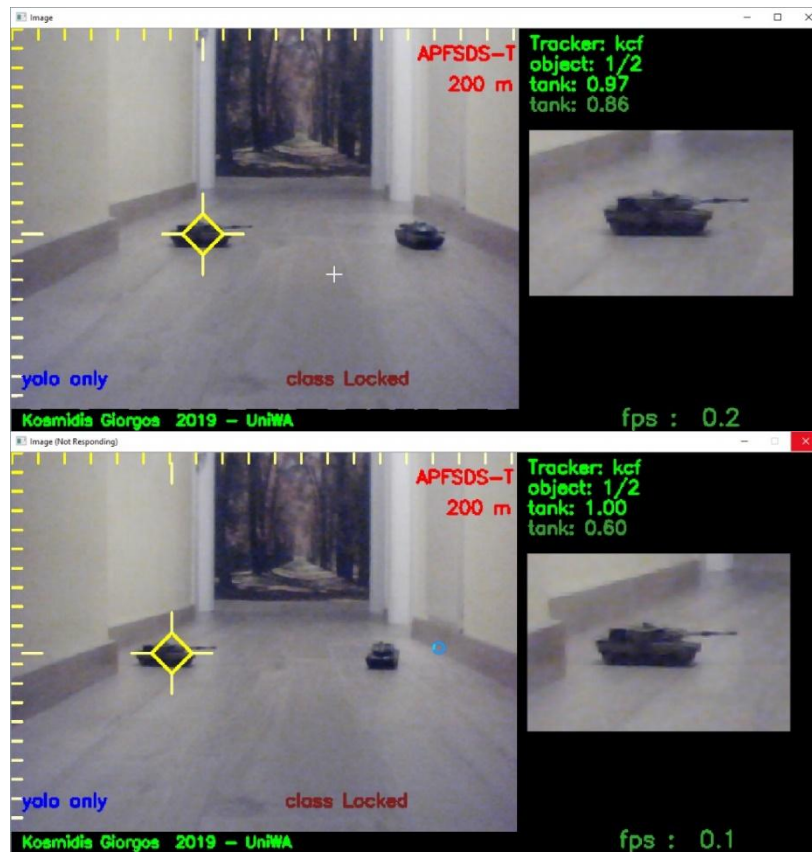
Μεταπτυχιακή Διπλωματική Εργασία, Γεώργιου Κοσμίδη , AM IES-0028

αναγνώρισης. Παρακάτω βλέπουμε σε αντιπαραβολή τα δύο μοντέλα. Πρώτα είναι το μικρότερο με τα 106 layers και μετά με τα 130.

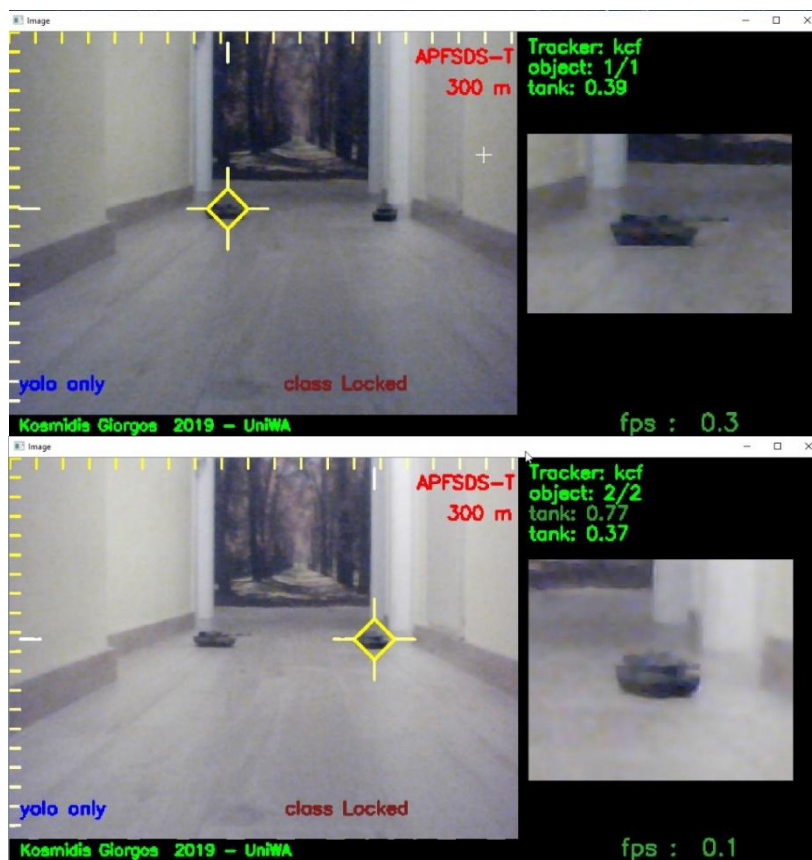


Εικόνα 78

Αναγνώριση στο 1 μέτρο



Εικόνα 79 Αναγνώριση στα 2 μέτρα



Εικόνα 80 Αναγνώριση στα 3 μέτρα



Εικόνα 81 Αναγνώριση στα 4 μέτρα



Εικόνα 82 Αναγνώριση στα 5 μέτρα

Στις μεγαλύτερες αποστάσεις όπου τα αντικείμενα μας μικραίνουν αρκετά, φαίνεται η ανωτερότητα του δεύτερου μοντέλου . Ωστόσο αυτό έχει αντίκτυπο στην ταχύτητα αναγνώρισης .

Επίλογος

Η τεχνητή νοημοσύνη είναι εδώ και καθημερινά χρησιμοποιούμε εκφάνσεις πολλών εφαρμογών της. Αυτό που πρέπει να γίνει κατανοητό είναι ότι εκπαιδεύεται από ανθρώπους και παραδείγματα και από αυτά αντλεί την ευφυΐα της. Αν το υλικό εκπαίδευσης είναι σωστό, τότε οι αποφάσεις της είναι σωστές. Αυτό φυσικά ισχύει όχι μόνο για εφαρμογές μηχανικής όρασης, ομιλίας κτλ αλλά και για εφαρμογές που πολλές φορές διεγείρουν τη αντίληψη μας για ηθική ή σωστό και λάθος, όπως τον αν μπορεί να αντικαταστήσει έναν δικαστή ή ακόμα και να αρθρογραφήσει αυτόνομα. Αυτό που προσφέρει είναι ταχύτητα στην λήψη αποφάσεων, βάση της εκπαίδευσης της πάντα, με τη βοήθεια της στατιστικής.

Στο μέλλον αναμένεται ραγδαία ανάπτυξη σε αυτόν τον τομέα και σε αυτό βοηθάει ότι είναι μια επιστήμη «open source». Η διασφάλιση της ακεραιότητας και της ομαλής λειτουργίας του κλάδου αναμένεται να γνωρίσει μεγάλη άνθηση παράλληλα. Για παράδειγμα στην προσπάθεια εύρεσης αδυναμιών των νευρωνικών δικτύων που κυριαρχούν σήμερα, έχει βρεθεί ότι η παρακάτω εικόνα δεν μπορεί να αναγνωριστεί.



Εικόνα 83

AI και ασφάλεια

Αρκούν 4 αυτοκόλλητα για να αχρηστέψουν ένα σύστημα αυτόνομης οδήγησης! Το πιθανότερο είναι το εν λόγω bug να καταπολεμήθηκε εν τη γενέσει του από τις εταιρείες που δραστηριοποιούνται στο χώρο της αυτόνομης οδήγησης, αλλά καταδεικνύει τις τάσεις που θα υπάρξουν πάνω στον κλάδο.

ΠΑΡΑΡΤΗΜΑ: Επίπεδα του Υολο

Τα layers του προκαθορισμένου σετ ρυθμίσεων yolov3.cfg

layer	filters	size	input	output
0 conv	32	3 x 3 / 1	416 x 416 x 3	416 x 416 x 32 0.299 BF
1 conv	64	3 x 3 / 2	416 x 416 x 32	208 x 208 x 64 1.595 BF
2 conv	32	1 x 1 / 1	208 x 208 x 64	208 x 208 x 32 0.177 BF
3 conv	64	3 x 3 / 1	208 x 208 x 32	208 x 208 x 64 1.595 BF
4	Shortcut Layer: 1			
5 conv	128	3 x 3 / 2	208 x 208 x 64	104 x 104 x 128 1.595 BF
6 conv	64	1 x 1 / 1	104 x 104 x 128	104 x 104 x 64 0.177 BF
7 conv	128	3 x 3 / 1	104 x 104 x 64	104 x 104 x 128 1.595 BF
8	Shortcut Layer: 5			
9 conv	64	1 x 1 / 1	104 x 104 x 128	104 x 104 x 64 0.177 BF
10 conv	128	3 x 3 / 1	104 x 104 x 64	104 x 104 x 128 1.595 BF
11	Shortcut Layer: 8			
12 conv	256	3 x 3 / 2	104 x 104 x 128	52 x 52 x 256 1.595 BF
13 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
14 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
15	Shortcut Layer: 12			
16 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
17 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
18	Shortcut Layer: 15			
19 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
20 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
21	Shortcut Layer: 18			
22 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
23 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
24	Shortcut Layer: 21			
25 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
26 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
27	Shortcut Layer: 24			
28 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
29 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
30	Shortcut Layer: 27			
31 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
32 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
33	Shortcut Layer: 30			
34 conv	128	1 x 1 / 1	52 x 52 x 256	52 x 52 x 128 0.177 BF
35 conv	256	3 x 3 / 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
36	Shortcut Layer: 33			
37 conv	512	3 x 3 / 2	52 x 52 x 256	26 x 26 x 512 1.595 BF
38 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
39 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
40	Shortcut Layer: 37			
41 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
42 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
43	Shortcut Layer: 40			
44 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
45 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
46	Shortcut Layer: 43			
47 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
48 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
49	Shortcut Layer: 46			
50 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
51 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
52	Shortcut Layer: 49			
53 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
54 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
55	Shortcut Layer: 52			
56 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
57 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
58	Shortcut Layer: 55			
59 conv	256	1 x 1 / 1	26 x 26 x 512	26 x 26 x 256 0.177 BF
60 conv	512	3 x 3 / 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
61	Shortcut Layer: 58			
62 conv	1024	3 x 3 / 2	26 x 26 x 512	13 x 13 x 1024 1.595 BF
63 conv	512	1 x 1 / 1	13 x 13 x 1024	13 x 13 x 512 0.177 BF
64 conv	1024	3 x 3 / 1	13 x 13 x 512	13 x 13 x 1024 1.595 BF
65	Shortcut Layer: 62			

```

66 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
67 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
68 Shortcut Layer: 65
69 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
70 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
71 Shortcut Layer: 68
72 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
73 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
74 Shortcut Layer: 71
75 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
76 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
77 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
78 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
79 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
80 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
81 conv 21 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 21 0.007 BF
82 yolo
83 route 79
84 conv 256 1 x 1 / 1 13 x 13 x 512 -> 13 x 13 x 256 0.044 BF
85 upsample 2x 13 x 13 x 256 -> 26 x 26 x 256
86 route 85 61
87 conv 256 1 x 1 / 1 26 x 26 x 768 -> 26 x 26 x 256 0.266 BF
88 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
89 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
90 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
91 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
92 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
93 conv 21 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 21 0.015 BF
94 yolo
95 route 91
96 conv 128 1 x 1 / 1 26 x 26 x 256 -> 26 x 26 x 128 0.044 BF
97 upsample 2x 26 x 26 x 128 -> 52 x 52 x 128
98 route 97 36
99 conv 128 1 x 1 / 1 52 x 52 x 384 -> 52 x 52 x 128 0.266 BF
100 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
101 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
102 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
103 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
104 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
105 conv 21 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 21 0.029 BF
106 yolo
Total BFLOPS 65.297

```

Τα layers του ειδικού σετ ρυθμίσεων yolov3L5.cfg

layer	filters	size	input	output
0 conv	32	3 x 3 / 1	416 x 416 x 3	-> 416 x 416 x 32 0.299 BF
1 conv	64	3 x 3 / 2	416 x 416 x 32	-> 208 x 208 x 64 1.595 BF
2 conv	32	1 x 1 / 1	208 x 208 x 64	-> 208 x 208 x 32 0.177 BF
3 conv	64	3 x 3 / 1	208 x 208 x 32	-> 208 x 208 x 64 1.595 BF
4	Shortcut Layer: 1			
5 conv	128	3 x 3 / 2	208 x 208 x 64	-> 104 x 104 x 128 1.595 BF
6 conv	64	1 x 1 / 1	104 x 104 x 128	-> 104 x 104 x 64 0.177 BF
7 conv	128	3 x 3 / 1	104 x 104 x 64	-> 104 x 104 x 128 1.595 BF
8	Shortcut Layer: 5			
9 conv	64	1 x 1 / 1	104 x 104 x 128	-> 104 x 104 x 64 0.177 BF
10 conv	128	3 x 3 / 1	104 x 104 x 64	-> 104 x 104 x 128 1.595 BF
11	Shortcut Layer: 8			
12 conv	256	3 x 3 / 2	104 x 104 x 128	-> 52 x 52 x 256 1.595 BF
13 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
14 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
15	Shortcut Layer: 12			
16 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
17 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
18	Shortcut Layer: 15			
19 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
20 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
21	Shortcut Layer: 18			
22 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
23 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
24	Shortcut Layer: 21			
25 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
26 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
27	Shortcut Layer: 24			
28 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
29 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
30	Shortcut Layer: 27			
31 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
32 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
33	Shortcut Layer: 30			
34 conv	128	1 x 1 / 1	52 x 52 x 256	-> 52 x 52 x 128 0.177 BF
35 conv	256	3 x 3 / 1	52 x 52 x 128	-> 52 x 52 x 256 1.595 BF
36	Shortcut Layer: 33			
37 conv	512	3 x 3 / 2	52 x 52 x 256	-> 26 x 26 x 512 1.595 BF
38 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
39 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
40	Shortcut Layer: 37			
41 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
42 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
43	Shortcut Layer: 40			
44 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
45 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
46	Shortcut Layer: 43			
47 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
48 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
49	Shortcut Layer: 46			
50 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
51 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
52	Shortcut Layer: 49			
53 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
54 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
55	Shortcut Layer: 52			
56 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
57 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
58	Shortcut Layer: 55			
59 conv	256	1 x 1 / 1	26 x 26 x 512	-> 26 x 26 x 256 0.177 BF
60 conv	512	3 x 3 / 1	26 x 26 x 256	-> 26 x 26 x 512 1.595 BF
61	Shortcut Layer: 58			
62 conv	1024	3 x 3 / 2	26 x 26 x 512	-> 13 x 13 x 1024 1.595 BF
63 conv	512	1 x 1 / 1	13 x 13 x 1024	-> 13 x 13 x 512 0.177 BF
64 conv	1024	3 x 3 / 1	13 x 13 x 512	-> 13 x 13 x 1024 1.595 BF
65	Shortcut Layer: 62			
66 conv	512	1 x 1 / 1	13 x 13 x 1024	-> 13 x 13 x 512 0.177 BF
67 conv	1024	3 x 3 / 1	13 x 13 x 512	-> 13 x 13 x 1024 1.595 BF
68	Shortcut Layer: 65			
69 conv	512	1 x 1 / 1	13 x 13 x 1024	-> 13 x 13 x 512 0.177 BF

```

70 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
71 Shortcut Layer: 68
72 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
73 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
74 Shortcut Layer: 71
75 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
76 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
77 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
78 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
79 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
80 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
81 conv 21 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 21 0.007 BF
82 yolo
83 route 79
84 conv 256 1 x 1 / 1 13 x 13 x 512 -> 13 x 13 x 256 0.044 BF
85 upsample 2x 13 x 13 x 256 -> 26 x 26 x 256
86 route 85 61
87 conv 256 1 x 1 / 1 26 x 26 x 768 -> 26 x 26 x 256 0.266 BF
88 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
89 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
90 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
91 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
92 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
93 conv 21 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 21 0.015 BF
94 yolo
95 route 91
96 conv 128 1 x 1 / 1 26 x 26 x 256 -> 26 x 26 x 128 0.044 BF
97 upsample 2x 26 x 26 x 128 -> 52 x 52 x 128
98 route 97 36
99 conv 128 1 x 1 / 1 52 x 52 x 384 -> 52 x 52 x 128 0.266 BF
100 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
101 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
102 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
103 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
104 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
105 conv 21 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 21 0.029 BF
106 yolo
107 route 103
108 conv 128 1 x 1 / 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
109 upsample 2x 52 x 52 x 128 -> 104 x 104 x 128
110 route 109 11
111 conv 64 1 x 1 / 1 104 x 104 x 256 -> 104 x 104 x 64 0.354 BF
112 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BF
113 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
114 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BF
115 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
116 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BF
117 conv 21 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 21 0.058 BF
118 yolo
119 route 115
120 conv 128 1 x 1 / 1 104 x 104 x 64 -> 104 x 104 x 128 0.177 BF
121 upsample 2x 104 x 104 x 128 -> 208 x 208 x 128
122 route 121 4
123 conv 32 1 x 1 / 1 208 x 208 x 192 -> 208 x 208 x 32 0.532 BF
124 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BF
125 conv 32 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 32 0.177 BF
126 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BF
127 conv 32 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 32 0.177 BF
128 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BF
129 conv 21 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 21 0.116 BF
130 yolo
Total BFLOPS 76.902

```

Βιβλιογραφία – πηγές

Νευρωνικά δίκτυα

Suvro Banerjee (medium.com)
Samit Saha (towardsdatascience.com)
SAGAR SHARMA (medium.com)
Grant Sanderson
Jonathan Hui (medium.com)
cc.gatech.edu

Αναγνώριση αντικειμένων

cv-tricks.com
Lil Log (github.com)

Metrics

Tarang Shah (towardsdatascience.com)

Yolo

Manivannan Murugavel (medium.com)

Λοιπά Blogs

Adrian Posebrock (pyimagesearch.com)
Sunita Nayak (learnopencv.com)
Satya Malick (learnopencv.com)