ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

“ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ”

Κατεύθυνση: ‘Μεγάλα Δεδομένα και Αναλυτική’

Μεταπτυχιακή Διατριβή

Ανίχνευση Αντικειμένων για Εφαρμογές Αυτόνομης Οδήγησης με χρήση Μοντέλων YOLO και RCNN

Ψωνιτής Αθανάσιος

Πειραιάς,2026

Επιβλέπων: Μιχαήλ Φιλιππάκης, Καθηγητής

Πανεπιστήμιο Πειραιώς. Κάτοχος όλων των δικαιωμάτων

University of Piraeus,. All rights reserved.

Συγγραφέας / Author : Ψωνιτής Αθανάσιος

ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

Ονοματεπώνυμο Φοιτητή: Ψωνιτής Αθανάσιος

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: Ανίχνευση Αντικειμένων για Εφαρμογές Αυτόνομης Οδήγησης με χρήση Μοντέλων YOLO και RCNN

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών "Πληροφοριακά Συστήματα & Υπηρεσίες" του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις 21/05/2026 από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή

Επιβλέπων (Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς): Μιχαήλ Φιλιππάκης, Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς

Μέλος Εξεταστικής Επιτροπής: Μαρία Χαλκίδη, Καθηγήτρια, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς

Μέλος Εξεταστικής Επιτροπής: Δημοσθένης Κυριαζής, Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Ο Ψωνιτής Αθανάσιος, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Ανίχνευση Αντικειμένων για Εφαρμογές Αυτόνομης Οδήγησης με χρήση Μοντέλων YOLO και RCNN», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Επιπλέον *δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.*

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας.

Ο ΔΗΛΩΝ

Ονοματεπώνυμο: Ψωνιτής Αθανάσιος

Αριθμός Μητρώου: ΜΕ2232

Υπογραφή: ΨΑ

Για την Μαρία Γιαννοπούλου

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως βασικό στόχο την αναγνώριση αντικειμένων για εφαρμογές αυτόνομης οδήγησης. Για να επιτευχθεί αυτό, εκπαιδεύσαμε και αξιολογήσαμε δύο διαδεδομένες αρχιτεκτονικές ανίχνευσης δεδομένων: του μονοσταδιακού ανιχνευτή YOLO (You Only Look Once) και του δι-σταδιακού ανιχνευτή Faster R-CNN.

Η εκπαίδευση και η αξιολόγηση των μοντέλων μας πραγματοποιήθηκε στο KITTI Vision Benchmark Suite, ένα από τα πλέον καθιερωμένα σύνολα δεδομένων στον τομέα της υπολογιστικής όρασης για εφαρμογές αυτόνομης οδήγησης. Το σύνολο δεδομένων περιλαμβάνει εικόνες πραγματικών οδικών συνθηκών με αναλυτικές επισημάνσεις για αντικείμενα όπως οχήματα και πεζούς οι οποίες ελήφθησαν στην Καρλσρούη της Γερμανίας.

Με την ολοκλήρωση της εκπαίδευσης και της αξιολόγησης των δύο μοντέλων συγκρίνουμε τις επιδόσεις που πέτυχαν με την βοήθεια μετρικών και δίνουμε έμφαση στα σημεία που πέτυχαν αλλά και σε αυτά που χρήζουν βελτίωσης.

Abstract

The main goal of this thesis is to identify objects for autonomous driving applications. To achieve this, we trained and evaluated two popular data detection architectures: the single-stage YOLO (You Only Look Once) detector and the two-stage Faster R-CNN detector.

The training and evaluation of our models was performed on the KITTI Vision Benchmark Suite, one of the most established datasets in the field of computer vision for autonomous driving applications. The dataset includes images of real road conditions with detailed annotations for objects such as vehicles and pedestrians, which were acquired in Karlsruhe, Germany.

Upon completion of the training and evaluation of the two models, we compare the performances achieved using metrics and emphasize the points that were achieved but also those that need improvement.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Μιχαήλ Φιλιππάκη του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς για την σημαντική και ανθρώπινη στήριξη που μου παρείχε, ακόμα και σε ιδιαίτερα δύσκολες στιγμές, σε όλη την διάρκεια του μεταπτυχιακού αλλά και της εκπόνησης της παρούσας εργασίας καθώς και την οικογένειά μου για την αμέριστη συμπαράστασή τους.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Ο άνθρωπος έχει την ικανότητα να αντιλαμβάνεται την τρισδιάστατη δομή του κόσμου με φαινομενική ευκολία. Μπορεί για παράδειγμα να αντιληφθεί τις τρεις διαστάσεις σε ένα αυτοκίνητο που περνά δίπλα του. Πιο συγκεκριμένα, διακρίνει το σχήμα και το μέγεθος του οχήματος, την διαφάνεια των τζαμιών, το χρώμα της επιφάνειας και τον αριθμό πινακίδας. Επιπλέον, είναι σε θέση να αναγνωρίσει την ύπαρξη οδηγού και άλλων επιβατών αλλά και τα βασικά χαρακτηριστικά τους όπως το φύλλο, την ηλικία (σε ένα εύλογο εύρος) αλλά και βασικές εκφράσεις συναισθημάτων. Η ικανότητα αυτή του εγκεφάλου και του οπτικού μας συστήματος είναι εντυπωσιακή και στο μεγαλύτερό της ποσοστό έχει αποκωδικοποιηθεί.

Η Υπολογιστική όραση (computer vision) είναι ένα πεδίο της τεχνητής νοημοσύνης που ασχολείται με την ανάπτυξη τεχνικών που επιτρέπουν στους υπολογιστές να επεξεργάζονται, να κατανοούν και να αναλύουν εικόνες και βίντεο όπως το κάνουν τα ανθρώπινα μάτια. Στόχος της υπολογιστικής όρασης είναι να εξάγει πληροφορίες ή να προβεί σε αποφάσεις βασιζόμενη σε εικόνες ή βίντεο.

Τα τελευταία χρόνια, οι έρευνες πάνω στο Computer Vision έχουν προχωρήσει ραγδαία δίνοντάς μας συστήματα αναγνώρισης αντικειμένων με μεγάλη ακρίβεια. Μια από τις μεγαλύτερες εφαρμογές του Computer Vision τα τελευταία χρόνια είναι στο πεδίο της οδήγησης. Για παράδειγμα, έχουμε συστήματα βοήθειας οδήγησης τα οποία χρησιμοποιούνται για την παροχή βοήθειας στον οδηγό σε διάφορες διαδικασίες όπως σε αυτή του παρκαρίσματος. Επιπλέον έχουν αναπτυχθεί συστήματα πρόβλεψης κίνησης βοηθώντας με αυτό τον τρόπο στον σχεδιασμό αποτελεσματικότερων συστημάτων κυκλοφορίας. Τέλος, το Computer Vision χρησιμοποιείται στην ανάπτυξη των αυτόνομων συστημάτων οδήγησης.

Στην παρούσα εργασία θα ασχοληθούμε με την Αυτόνομη Οδήγηση, ένα πεδίο που εξελίσσεται συνεχώς και έχει ακόμα πολλά περιθώρια βελτίωσης. Αρχικά, θα μιλήσουμε για την Μηχανική Μάθηση και για τις κατηγορίες μάθησης. Έπειτα, θα αναλύσουμε τους βασικότερους αλγορίθμους Κατηγοριοποίησης και Συσταδοποίησης. Στην συνέχεια, θα αναφερθούμε στην Βαθιά Μάθηση και πιο συγκεκριμένα θα ασχοληθούμε με τα Νευρωνικά Δίκτυα. Τέλος, θα αναλύσουμε τους αλγορίθμους YOLO και RCNN και θα δούμε τις παραλλαγές τους επισημαίνοντας την βελτίωση κάθε νέας έκδοσης στην Αναγνώριση Αντικειμένων.

Στο πρακτικό κομμάτι της εργασίας θα πάρουμε το σύνολο δεδομένων KITTI και θα εφαρμόσουμε αλγορίθμους αναγνώρισης αντικειμένων με σκοπό την εύρεση του καλύτερου μοντέλου που θα μπορούσε να σταθεί πιο αποτελεσματικά σε ένα σύστημα αυτόνομης οδήγησης.

ΚΕΦΑΛΑΙΟ 2

DATA MINING

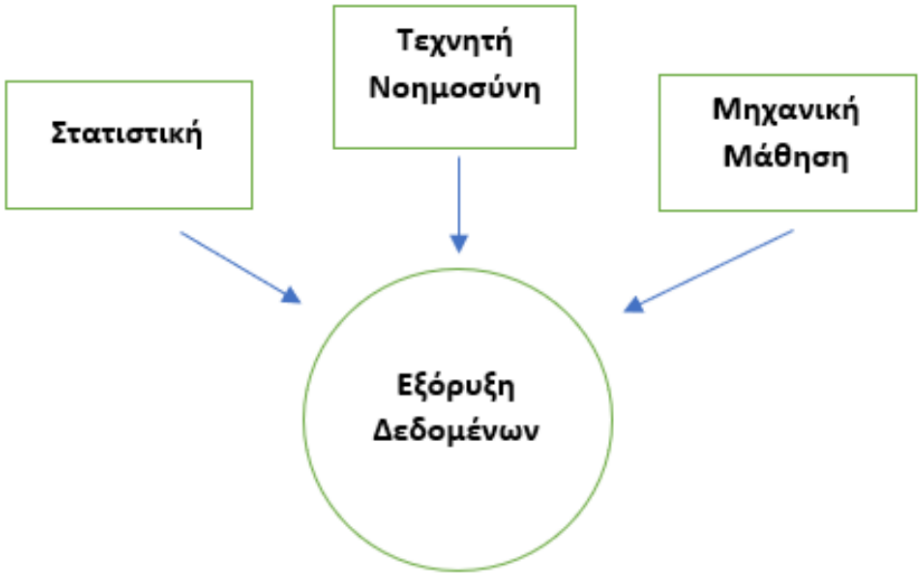
Οι άνθρωποι αναζητούν πρότυπα στα δεδομένα από τότε που ξεκίνησε η ανθρώπινη ζωή. Οι προϊστορικοί άνθρωποι έβρισκαν πρότυπα στις κινήσεις των ζώων, στα περάσματα που επιλέγουν καθώς και στην μεταναστευτική συμπεριφορά τους. Στα πιο σύγχρονα χρόνια, οι αγρότες αναζητούν πρότυπα στην ανάπτυξη των καλλιεργειών τους και πιο συγκεκριμένα ποιοι παράγοντες οδηγούν σε μεγαλύτερες σοδειές.

Τα τελευταία χρόνια η αναζήτηση προτύπων έχει εξελιχθεί σε έναν τομέα πολύ σημαντικό για διάφορους επιστημονικούς κλάδους αλλά και επιχειρήσεις λόγω της ικανότητάς του να διαχειρίζεται μεγάλο όγκο δεδομένων και να δημιουργεί γνώση. Ο επιστήμονας θα πρέπει να κατανοεί τα δεδομένα, να ανακαλύπτει τα πρότυπα που διέπουν τον τρόπο λειτουργίας του φυσικού κόσμου και να τα ενσωματώνουν σε θεωρίες που μπορούν να χρησιμοποιηθούν για την εξέλιξη της επιστήμης τους. Αντίστοιχα, ένας επιχειρηματίας θα πρέπει να εντοπίζει ευκαιρίες και κινδύνους με σκοπό να κάνει τις σωστές κινήσεις που θα του φέρουν την κερδοφορία.

Μπορούμε εύκολα να αντιληφθούμε πως οι ενέργειες που περιγράψαμε παραπάνω είναι δύσκολο να γίνουν στο σύνολό τους από τον άνθρωπο. Φυσικά, δεν μιλάμε για τις περιπτώσεις των προϊστορικών ανθρώπων και το κυνήγι όπου εκεί μέσω παρατήρησης μπορούσαν να βγουν συμπεράσματα. Στα προβλήματα του σύγχρονου κόσμου ο όγκος των δεδομένων για επεξεργασία είναι τις περισσότερες φορές τόσο μεγάλος που θα πρέπει να υπάρξει η συμβολή ενός κλάδου της επιστήμης των Μεγάλων Δεδομένων για να δώσει την λύση.

Η εξόρυξη δεδομένων ορίζεται ως η διαδικασία ανακάλυψης προτύπων σε δεδομένα. Τα πρότυπα αυτά θα πρέπει να οδηγούν σε κάποιο πλεονέκτημα όπως για παράδειγμα στην ανακάλυψη γνώσης, κάποιο οικονομικό όφελος ,την πρόβλεψη τάσεων , την λήψη αποφάσεων και την βελτίωση της απόδοσης.

Οι βασικότεροι επιστημονικοί κλάδοι στους οποίους βασίζεται η Εξόρυξη Δεδομένων είναι:



Εικόνα 1. Σχήμα πεδίων για την Εξόρυξη Δεδομένων

Στατιστική

Στατιστική είναι η επιστήμη η οποία ασχολείται με τον σχεδιασμό πειραμάτων, τη συλλογή και ανάλυση αριθμητικών δεδομένων και την εξαγωγή συμπερασμάτων. Τα συμπεράσματα αυτά αναφέρονται σε άγνωστα χαρακτηριστικά ή ιδιότητες πληθυσμών και εξάγονται με την βοήθεια των πληροφοριών που εμπεριέχονται σε δείγματα από τους πληθυσμούς αυτούς.

Τεχνητή Νοημοσύνη

Τεχνητή Νοημοσύνη είναι ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζουμε με την νοημοσύνη στην ανθρώπινη συμπεριφορά(—Barr,Feigenbaum). Πρόκειται για μια από τις νεότερες επιστήμες και αποτελεί σημείο τομής πολλών επιστημών όπως η πληροφορική, η επιστήμη μηχανικών , η μαθηματική λογική και η φιλοσοφία.

Μηχανική Μάθηση

Μηχανική μάθηση αποτελεί το επιστημονικό πεδίο που ασχολείται με την δημιουργία μοντέλων ή προτύπων από ένα σύνολο δεδομένων. Πρόκειται δηλαδή, για την μελέτη υπολογιστικών μεθόδων για την απόκτηση νέας γνώσης, νέων δεξιοτήτων και νέων τρόπων οργάνωσης της υπάρχουσας γνώσης ενός υπολογιστικού συστήματος. Μπορούμε να διακρίνουμε την μηχανική μάθηση σε δύο κατηγορίες.

Η Εξόρυξη Δεδομένων βρίσκει εφαρμογές σε διάφορους τομείς. Παρακάτω θα δούμε κάποια χαρακτηριστικά παραδείγματα τέτοιων εφαρμογών.

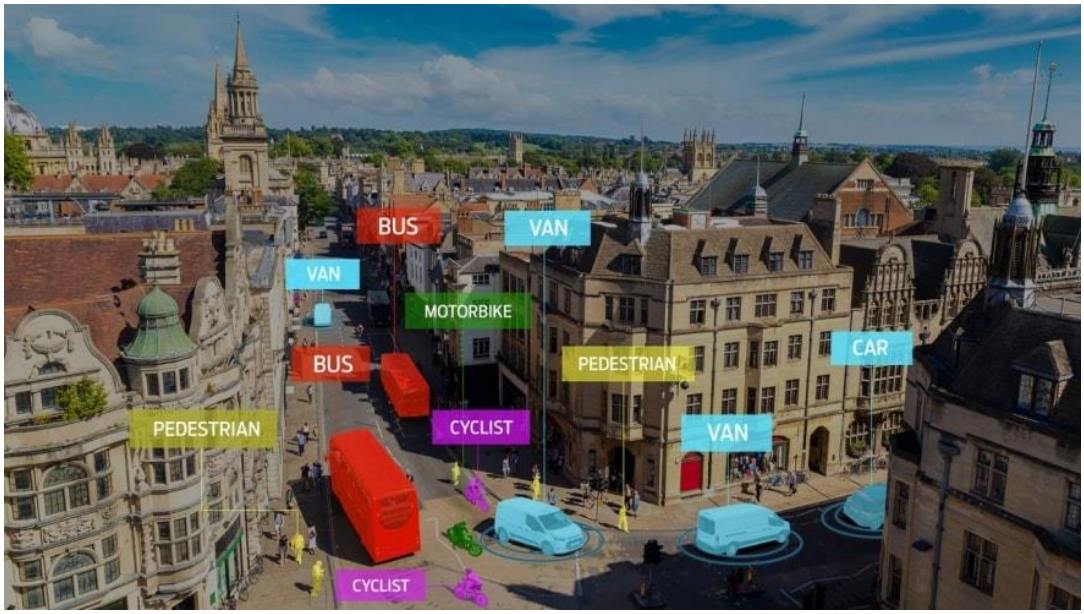
Στις εμπορικές επιχειρήσεις πρωταρχικός στόχος είναι η αύξηση των πωλήσεων και του κέρδους. Η Ανάλυση Καλαθιών (Basket Analysis) είναι μία εφαρμογή της Εξόρυξης Δεδομένων στο λιανικό εμπόριο. Βασικός της σκοπός είναι να ανιχνεύσει τους πιο συνήθεις συνδυασμούς προϊόντων που αγοράζονται μαζί από τους καταναλωτές. Τα αποτελέσματα αυτής της ανάλυσης βοηθά τις επιχειρήσεις να προωθούν τα κατάλληλα προϊόντα με προωθητικές ενέργειες και να διαμορφώνουν κατάλληλα τα ράφια τους ώστε να βελτιωθεί το layout των καταστημάτων. Επιπλέον, η Εξόρυξη Δεδομένων μπορεί να διαχειριστεί μεγάλο όγκο δημογραφικών στοιχείων των πελατών γεγονός που βοηθά στον καλύτερο σχεδιασμό διαφημιστικών καμπανιών.

Στο τομέα των τηλεπικοινωνιών, οι εταιρείες μπορούν να χρησιμοποιήσουν την Εξόρυξη Δεδομένων για να προβλέψουν την κίνηση δεδομένων στα δίκτυά τους. Αυτό μπορεί να βοηθήσει στο σχεδιασμό και την βελτίωση των υποδομών τους καθώς και τον καθορισμό των απαιτήσεων επιπέδου υπηρεσιών για τις μελλοντικές τους κινήσεις. Επιπλέον, με τα δεδομένα χρήσης των πελατών οι εταιρείες τηλεπικοινωνιών είναι σε θέση να προσφέρουν κατάλληλα προγράμματα στους πελάτες τους.

Στον τομέα της υγείας έχουμε πολλές εφαρμογές Εξόρυξης Δεδομένων. Για παράδειγμα, σε ιατρικά δεδομένα μπορούμε να ανακαλύψουμε πρότυπα που αντιστοιχούν σε σοβαρές ασθένειες βοηθώντας έτσι στην πρόβλεψη και την πρόληψη νέων περιστατικών.

Στον χώρο της βιομηχανίας τα δεδομένα που παράγονται από την εξόρυξη δίνουν στοιχεία που βοηθούν στην στοχευμένη αύξηση παραγωγής σε αγαθά με περισσότερη ζήτηση. Επιπλέον βοηθά στην διασφάλιση ποιότητας και στην έγκαιρη διάγνωση προβλημάτων. Στο κομμάτι αυτό, η Ευρωπαϊκή Επιτροπή ζητά από τους κατασκευαστές αυτοκινήτων να εφοδιάσουν τα νέα μοντέλα τους με καινοτόμα συστήματα ασφαλείας, όπως οι περιοριστές ταχύτητας και οι οθόνες παρακολούθησης που ανιχνεύουν την υπνηλία καθώς και την απόσπαση της προσοχής των οδηγών. Για παράδειγμα, η Ford , στις αρχές του 2020, ετοίμασε ένα project με σκοπό την πρόληψη των ατυχημάτων. Με την χρήση 700 οχημάτων για 18 μήνες συνέλλεξε δεδομένα τηλεματικής όπως για παράδειγμα τη χρήση φρένου και γκαζιού και την γωνία περιστροφής τιμονιού . Έτσι, εκπαίδευσε αισθητήρες που χρησιμοποιούν αλγορίθμους μηχανικής μάθησης με σκοπό να προειδοποιούν για ατυχήματα που προβλέπουν ότι θα συμβούν με άλλα διασυνδεδεμένα οχήματα καθώς και με το περιβάλλον τους όπως τους πεζούς και τα φυσικά εμπόδια. Οι πληροφορίες θα επιτρέψουν στις πόλεις να λάβουν προληπτικά μέτρα για την

αντιμετώπιση των ατυχημάτων σε δρόμους και κόμβους που ενέχουν τον υψηλότερο κίνδυνο για τους οδηγούς.



Εικόνα 2. Παράδειγμα αναγνώρισης αντικειμένων

ΚΕΦΑΛΑΙΟ 3

3.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης που εστιάζει στην ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στα συστήματα υπολογιστών να μαθαίνουν και να βελτιώνουν την απόδοσή τους από την εμπειρία. Την μηχανική μάθηση την συναντάμε σε πολλές πτυχές της κοινωνίας μας. Για παράδειγμα, στον τομέα της υγείας και της ιατρικής οι επιστήμονες μπορούν να αναλύσουν μεγάλο όγκο δεδομένων ασθενών και να προβλέψουν την εξέλιξη μίας ασθένειας ή και να βγάλουν εξατομικευμένο πρόγραμμα θεραπείας για έναν ασθενή. Στο τομέα των μεταφορών μπορούμε να έχουμε συστήματα ελέγχου της κίνησης, βελτίωση της απόδοσης της ασφάλειας των μεταφορών καθώς και την ανάπτυξη των αυτόνομων οχημάτων. Τέλος, εφαρμογές μηχανικής μάθησης μπορούμε να έχουμε με σκοπό την πρόβλεψη φυσικών φαινομένων αλλά και για την πρόληψη κοινωνικών προβλημάτων όπως η ηλεκτρονική απάτη χρηστών του διαδικτύου.

Σε έναν πιο επίσημο ορισμό του Tom M. Mitchell :

Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από την εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E.

Για να το πούμε με πιο απλά λόγια, ένας υπολογιστής που έχει σκοπό να παίζει σκάκι, μπορεί να βελτιώσει την απόδοσή του, που μετριέται από την ικανότητά του να κερδίζει στην κατηγορία που αφορά παιχνίδι σκάκι, αποκτώντας εμπειρία παίζοντας παιχνίδια με τον εαυτό του.

Στο πρόβλημα της αναγνώρισης αντικειμένων από εικόνες η μηχανική μάθηση παίζει πολύ σημαντικό ρόλο στην δημιουργία ενός μοντέλου που θα μπορεί αξιόπιστα να μας δίνει τα αποτελέσματα που ζητάμε. Τα βήματα που θα ακολουθούσαμε , για παράδειγμα για το πρόβλημα της αυτόνομης οδήγησης, είναι τα εξής:

Αρχικά, θα πρέπει να συλλέξουμε τα δεδομένα για την διαδικασία της εκπαίδευσης. Αυτό περιλαμβάνει εικόνες με οχήματα από δρόμους, από χώρους στάθμευσης κ.α. Επιπλέον, θα μπορούσαμε να χρησιμοποιήσουμε εικόνες ανθρώπων, ζώων, ποδηλάτων, φωτεινών σηματοδοτών και αναπηρικών αμαξιδίων.

Στη συνέχεια, θα πρέπει να κάνουμε προεπεξεργασία των δεδομένων μας. Αυτό περιλαμβάνει τον καθορισμό συγκεκριμένων διαστάσεων και χρωματικών μοντέλων, την κανονικοποίηση των εικόνων αλλά και την αύξηση του συνόλου δεδομένων με την περιστροφή, την αναστροφή και την μεγέθυνση των εικόνων.

Ακολούθως θα πρέπει να επιλέξουμε ένα προεκπαιδευμένο μοντέλο, κατάλληλο για την αναγνώριση αντικειμένων. Σε αυτό το μοντέλο θα πρέπει να γίνει και ανάλογη προσαρμογή ώστε να ταιριάζει με το πρόβλημα που καλούμαστε να λύσουμε. Αφού το προσαρμόσουμε κατάλληλα, μπορούμε να το τροφοδοτήσουμε με το σύνολο δεδομένων μας, κρατώντας ένα μέρος του εκτός διαδικασίας, ώστε να ξεκινήσει η διαδικασία της εκπαίδευσης. Σε αυτή τη φάση καλό θα είναι να εφαρμόσουμε τεχνικές βελτιστοποίησης με σκοπό την καλύτερη απόδοση του συστήματος.

Έχοντας εκπαιδεύσει το μοντέλο μας, θα πρέπει να το δοκιμάσουμε και σε δεδομένα με τα οποία δεν έχει την εμπειρία. Του δίνουμε έτσι το μέρος του συνόλου δεδομένων που δεν συμπεριλάβαμε στο προηγούμενο στάδιο, με σκοπό να τεστάρουμε την απόδοση και τα παραγόμενα αποτελέσματα. Με το τέλος της διαδικασίας αυτής καλούμαστε να αξιολογήσουμε τα αποτελέσματα που έβγαλε το σύστημα μας με τις νέες εικόνες. Αυτό το καταφέρνουμε κάνοντας χρήση μετρικών όπως το precision, το recall, και το F1 score.

Όταν καταφέρουμε να πετύχουμε υψηλές τιμές στις μετρικές μας , είμαστε έτοιμοι να χρησιμοποιήσουμε το μοντέλο μας σε προβλήματα του πραγματικού κόσμου.

3.2 ΚΑΤΗΓΟΡΙΕΣ ΜΑΘΗΣΗΣ

Μάθηση με επίβλεψη

Κατά τη διαδικασία της μάθησης με επίβλεψη, τροφοδοτούμε το σύστημά μας με ένα σύνολο δεδομένων εκπαίδευσης. Το σύνολο αυτό περιέχει παρατηρήσεις με ετικέτα. Με βάση τις ετικέτες μπορεί να κατηγοριοποιεί σε σύνολα και αυτός είναι και ο τρόπος με τον οποίο μαθαίνει το σύστημα. Στην αυτόνομη οδήγηση για παράδειγμα, ένα σύνολο που θα μπορούσαμε να δώσουμε στο προς εκπαίδευση σύστημα είναι παρατηρήσεις στις οποίες κάθε παράδειγμα αποτελεί και μία κατάσταση (π.χ. κώνος στον δρόμο μπροστά) και σαν ετικέτα να δώσουμε την σωστή ενέργεια που θα πρέπει να εκτελέσει το σύστημα (π.χ. αποφυγή αυτού του δρόμου και επιλογή εναλλακτικής διαδρομής). Η μάθηση με επίβλεψη αποτελεί ίσως τον πιο γνωστό τρόπο μάθησης στην μηχανική μάθηση, καθώς με ένα σωστό σύνολο εκπαίδευσης πετυχαίνουμε καλά αποτελέσματα εκμάθησης.

Χαρακτηριστικά παραδείγματα αλγορίθμων βασισμένοι σε μάθηση με επίβλεψη είναι η Γραμμική Παλινδρόμηση, τα Support Vector Machines (SVM) και ο Random Forest.

Μάθηση χωρίς επίβλεψη

Σε αντίθεση με την προηγούμενη κατηγορία, η μάθηση χωρίς επίβλεψη παίρνει σύνολα δεδομένων εκπαίδευσης χωρίς ετικέτες. Έτσι, το σύστημα καλείται να ανακαλύψει από μόνο του μοτίβα και σχέσεις μεταξύ των δεδομένων χωρίς να έχουν συγκεκριμένες απαιτήσεις από τους σχεδιαστές. Στο παράδειγμα της αυτόνομης οδήγησης, δίνοντας στο σύστημα ένα σύνολο δεδομένων που περιέχει πληροφορίες για την κατάσταση του δρόμου σε διάφορες καιρικές συνθήκες, θα μπορούσαμε να πάρουμε σαν αποτέλεσμα την μείωση ταχύτητας σε κατάσταση βροχής ή καταιγίδας. Αυτή την πληροφορία θα την εντόπιζε το σύστημα από μόνο του βλέποντας σαν μοτίβο την συχνότητα τροχαίων ατυχημάτων λόγω ταχύτητας και ολισθηρότητας του εδάφους σε μέρες βροχής. Όπως και πριν, έτσι και σε αυτή την κατηγορία καταλαβαίνουμε τον σημαντικό ρόλο που παίζει η δημιουργία ενός σωστού και πλήρους συνόλου δεδομένων εκπαίδευσης από τους σχεδιαστές και αναλυτές του συστήματος.

Οι πιο γνωστοί αλγόριθμοι της κατηγορίας αυτής είναι ο k-means και το Principal Component Analysis (PCA).

Ενισχυτική μάθηση

Σε αυτό το είδος μάθησης, σκοπός μας είναι να μάθουμε στο σύστημα τι να κάνει ,μέσα από χαρτογράφηση καταστάσεων σε πράξεις, έτσι ώστε να μεγιστοποιηθεί

ένα αριθμητικό σήμα ανταμοιβής. Συγκεκριμένα, το σύστημα προς εκπαίδευση δεν ενημερώνεται για τα actions που πρέπει να κάνει, αλλά αντίθετα πρέπει να ανακαλύψει μόνο του τις ενέργειες που αποφέρουν την μεγαλύτερη ανταμοιβή μέσα από δοκιμές. Σε κάποιες περιπτώσεις τα actions αυτά μπορούν να επηρεάσουν όχι μόνο την ανταμοιβή της τρέχουσας κατάστασης αλλά και επόμενων. Έτσι, το σύστημα θα πρέπει να προβεί σε actions που προσφέρουν μακρυπρόθεσμη θετική ανταμοιβή για την επίτευξη του τελικού στόχου, και όχι απαραίτητα την μέγιστη ανταμοιβή για κάποιον βραχυπρόθεσμο στόχο. Η διαδικασία αυτή αναφέρεται συχνά ως καθυστερημένη ανταμοιβή. Η δοκιμή και το λάθος αλλά και η καθυστερημένη ανταμοιβή αποτελούν τα δύο βασικότερα στοιχεία της ενισχυτικής μάθησης. Για παράδειγμα, σε ένα σύστημα αυτόνομης οδήγησης ,που βρίσκεται στη διαδικασία εκπαίδευσης, δεν δίνουμε καμία πληροφορία εξ' αρχής. Όταν δούμε για παράδειγμα ότι αποφεύγει επιτυχώς έναν πεζό ή ένα εμπόδιο τότε δίνουμε μεγάλη ανταμοιβή, ενώ όταν παραβιάζει τον κόκκινο σηματοδότη του δίνουμε χαμηλή. Με αυτό το τρόπο το σύστημα καταλαβαίνει ποια ενέργεια είναι σωστή και ποια όχι.

3.3 KDD

Το kdd (Ανακάλυψη Γνώσης από Βάσεις Δεδομένων) αποτελεί ένα από τα πιο δημοφιλή μοντέλα διεργασιών εξόρυξης δεδομένων. Σε αυτή τη διαδικασία γίνεται ανακάλυψη γνώσης μέσα από την ανάλυση των δεδομένων. Με τον όρο kdd αναφερόμαστε στο σύνολο της διαδικασίας εξόρυξης δηλαδή από την συλλογή των δεδομένων μας έως την παραγωγή της γνώσης.

Τα βήματα που ακολουθεί η διαδικασία KDD είναι:

1.Συλλογή/Επιλογή Δεδομένων

Αρχικά, στην διαδικασία KDD έχουμε την Συλλογή Δεδομένων καθώς και την αποθήκευσή τους . Τα δεδομένα αυτά μπορούμε να τα λάβουμε από αισθητήρες, από κάμερες κίνησης, από ραντάρ αλλά και από αρχεία και φόρμες συμπλήρωσης. Όπως μπορούμε να αντιληφθούμε τα δεδομένα που προκύπτουν από αυτή την διαδικασία είναι μη επεξεργασμένα επομένως εμπεριέχουν ελλείπεις τιμές και θόρυβο.

2.Προεπεξεργασία Δεδομένων

Αποτελεί ίσως το σημαντικότερο μέρος της διαδικασίας KDD. Σε αυτό το στάδιο καλούμαστε να φέρουμε τα δεδομένα μας σε μορφή κατάλληλη για επεξεργασία στα επόμενα βήματα. Αυτό γίνεται με τον καθαρισμό των δεδομένων και την επεξεργασία των ελλιπών και λανθασμένων παρατηρήσεων. Πολλές φορές αυτή η διαδικασία καταλαμβάνει το μεγαλύτερο μέρος της συνολικής διαδικασίας KDD. Το γεγονός αυτό δεν μας κάνει εντύπωση καθώς θα πρέπει να έχουμε την σωστή δομή για τα δεδομένα μας για να μπορούμε να συνεχίσουμε με ασφάλεια στα επόμενα

βήματα. Για την προεπεξεργασία των δεδομένων χρησιμοποιούμε στατιστικές μεθόδους και αλγορίθμους από το πεδίο της εξόρυξης δεδομένων.

3.Μετασχηματισμός Δεδομένων

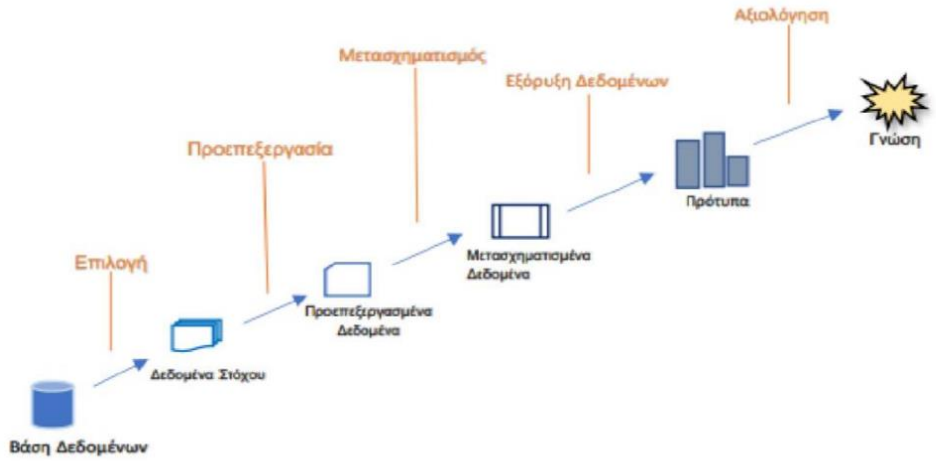
Τρίτο στάδιο αποτελεί ο Μετασχηματισμός Δεδομένων. Εδώ γίνεται η μετατροπή των δεδομένων με τρόπο κατάλληλο ώστε να περάσουν τα δεδομένα στα στάδια της Επεξεργασίας. Συγκεκριμένα , εδώ γίνεται η εξομάλυνση των δεδομένων και η απομάκρυνση θορύβου καθώς και η κανονικοποίηση.

4.Εξόρυξη Δεδομένων

Σε αυτό το στάδιο προκύπτει κάποιο μοντέλο μετά την εφαρμογή κάποιου αλγορίθμου. Συγκεκριμένα, ο αλγόριθμος που θα επιλεγεί, θα πάρει τα ποιοτικά δεδομένα που προέκυψαν στα προηγούμενα στάδια και θα δημιουργήσει κάποιο μοντέλο(συνήθως κατηγοριοποίησης ή πρόβλεψης). Το μοντέλο αυτό στην συνέχεια θα είναι σε θέση να μας δώσει απάντηση για την τιμή ενός χαρακτηριστικού-μεταβλητής στόχου για νέα και άγνωστα δεδομένα.

5. Αξιολόγηση/Διερμηνεία

Στο τελικό στάδιο γίνεται διερμηνεία και αξιολόγηση των αποτελεσμάτων που παρήχθησαν από τα παραπάνω στάδια. Επιπλέον, αξιολογείται το στάδιο της Προεπεξεργασίας σε σχέση με την επίδρασή του στα αποτελέσματα του αλγορίθμου της εξόρυξης δεδομένων.



Εικόνα 3. Σχεδιάγραμμα της διαδικασίας KDD

Με την διαδικασία του KDD μας παρέχεται μία δομημένη προσέγγιση για την ανάλυση και εξόρυξη δεδομένων. Μπορούμε λοιπόν να έχουμε ανακάλυψη προτύπων και τάσεων, σημαντικές πληροφορίες για Ανάλυση Κινδύνου (Risk

Analysis) για τις επιχειρήσεις και τους οργανισμούς αλλά και βελτιστοποίηση των αποφάσεων και των διαδικασιών.

3.4 ΤΕΧΝΙΚΕΣ ΠΑΛΙΝΔΡΟΜΗΣΗΣ

Όταν έχουμε δύο ή περισσότερες μεταβλητές σε ένα πρόβλημα συχνά υπάρχει συσχέτιση μεταξύ τους και η τιμή της μίας καθορίζει λίγο ή πολύ την τιμή των άλλων. Έχουμε λοιπόν δύο είδη μεταβλητών:

- την εξαρτημένη μεταβλητή που προσπαθούμε να προβλέψουμε και
- την ανεξάρτητη μεταβλητή που επηρεάζει την εξαρτημένη

Ας δούμε μερικά παραδείγματα τέτοιων μεταβλητών:

1. Τα κέρδη μίας επιχείρησης από το νέο τους προϊόν σε σχέση με τα χρήματα που καταναλώθηκαν για την προώθησή του από το τμήμα μάρκετινγκ
2. Η τιμή ενοικίου ενός διαμερίσματος σε σχέση με το έτος κατασκευής της πολυκατοικίας, την περιοχή, το μέγεθός του και την εύκολη και γρήγορη πρόσβαση σε Μέσα Μαζικής Μεταφοράς
3. Η βαθμολογία των μαθητών ενός σχολείου σε σχέση με τον χρόνο που αφιερώνουν καθημερινά για μελέτη, την παρουσία τους στα μαθήματα, την κοινωνική αλλά και την οικονομική τους κατάσταση και τέλος την ψυχολογία τους

Στο πρώτο παράδειγμα η εξαρτημένη μεταβλητή είναι τα κέρδη ενώ η ανεξάρτητη είναι τα χρήματα που ξόδεψε η επιχείρηση στην προώθηση του προϊόντος. Συνήθως συμβολίζουμε με Χ την ανεξάρτητη μεταβλητή και με Υ την εξαρτημένη. Για να μπορέσουμε να περιγράψουμε τις παραπάνω σχέσεις και άλλων παρόμοιων θα πρέπει να κάνουμε χρήση στατιστικών μοντέλων παλινδρόμησης. Τα μοντέλα παλινδρόμησης μας προσφέρουν εξισώσεις που μπορούν να περιγράψουν τέτοιου είδους σχέσεις και να κάνουν προβλέψεις για νέες τιμές των μεταβλητών.

3.4.1 ΑΠΛΗ ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ

Ένα από τα πιο γνωστά μοντέλα παλινδρόμησης είναι η Απλή Γραμμική Παλινδρόμηση. Η σχέση που την περιγράφει είναι η παρακάτω:

$$y_i = b_0 + b_1 * x_i + e_i$$

Όπου

- x είναι η ανεξάρτητη μεταβλητή

-y η εξαρτημένη μεταβλητή

-b_0 και b_1 οι παράμετροι του μοντέλου ή αλλιώς οι συντελεστές παλινδρόμησης.

-e_i είναι το τυχαίο σφάλμα για την i-οστή παρατήρηση.

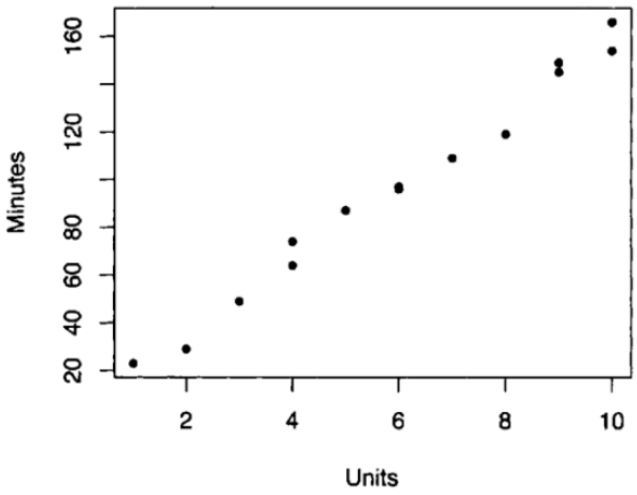
Όπως μπορούμε να καταλάβουμε βάζοντας τιμές στα b0 και b1 μπορούμε να πάρουμε έναν πολύ μεγάλο αριθμό ευθείων που θα περιγράφουν την σχέση μεταξύ των δύο μεταβλητών. Σκοπός μας είναι να καταφέρουμε να βρούμε τον συνδυασμό εκείνο που περιγράφει καλύτερα την σχέση της εξαρτημένης μεταβλητής με τις ανεξάρτητες. Για να δώσουμε εκτίμηση στις παραμέτρους κάνουμε χρήση της μεθόδου ελαχίστων τετραγώνων. Με βάση αυτή την μέθοδο θα πρέπει να βρούμε την ευθεία που περνά κατά μέσο όρο πιο κοντά από όλες τις παρατηρήσεις ώστε να μειωθεί η πιθανότητα λάθους εκτίμησης για μία νέα παρατήρηση.

Ας δούμε παρακάτω ένα παράδειγμα εφαρμογής της απλής γραμμικής παλινδρόμησης. Έχουμε μία εταιρεία που επιδιορθώνει προβλήματα σε ηλεκτρονικούς υπολογιστές απομακρυσμένα μέσω τηλεφώνου. Θέλουμε να δούμε την σχέση μεταξύ διάρκειας κλήσης από τους πελάτες με τους τεχνικούς της εταιρείας και τον αριθμό προβλημάτων που αναφέρουν οι πρώτοι και θα πρέπει να επιλυθούν από τους δεύτερους. Τα δεδομένα μας φαίνονται παρακάτω:

Row	Minutes	Units
1	23	1
2	29	2
3	49	3
4	64	4
5	74	4
6	87	5
7	96	6
8	97	6
9	109	7

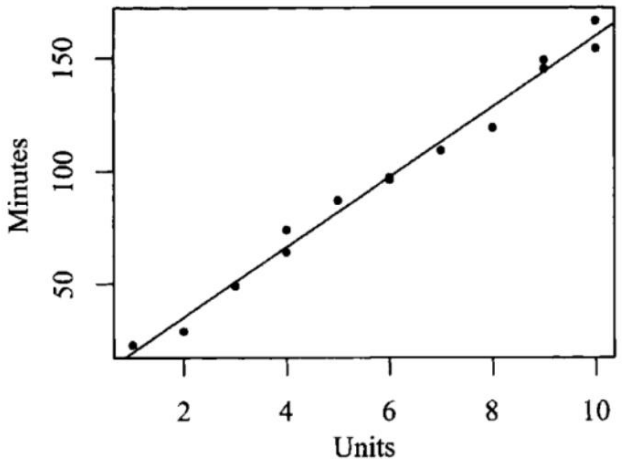
10	119	8
11	149	9
12	145	9
13	154	10
14	166	10

Η οπτικοποίηση των δεδομένων μας σχηματικά φαίνεται στο ακόλουθο διάγραμμα:



Εικόνα 4. Οπτικοποίηση δεδομένων παραδείγματος

Εφαρμόζοντας την μέθοδο ελαχίστων τετραγώνων βρίσκουμε τα b0 και b1 που περιγράφουν καλύτερα την σχέση των δύο μεταβλητών. Η ευθεία που σχηματίζεται φαίνεται στο παρακάτω σχήμα :



Εικόνα 5. Διάγραμμα εφαρμογής μεθόδου ελαχίστων τατραγώνων

3.4.2 ΠΟΛΛΑΠΛΗ ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ

Στις περιπτώσεις που ένα πρόβλημα για να περιγραφεί χρειάζεται παραπάνω από μία ανεξάρτητες μεταβλητές, όπως στο παράδειγμα που δώσαμε στο προηγούμενο κεφάλαιο με την τιμή ενοικίου, η απλή γραμμική παλινδρόμηση δεν μπορεί να εφαρμοστεί. Πιο συγκεκριμένα, στο παράδειγμά μας η τιμή του ενοικίου που είναι η εξαρτημένη μεταβλητή εξαρτάται από πολλές ανεξάρτητες μεταβλητές όπως το έτος κατασκευής της πολυκατοικίας, η περιοχή, το μέγεθός του κ.α. Όπως και στην απλή γραμμική παλινδρόμηση , έτσι και στην πολλαπλή προσπαθούμε να βρούμε την εξίσωση που περιγράφει καλύτερα την σχέση των ανεξάρτητων μεταβλητών με την εξαρτημένη. Στις περιπτώσεις λοιπόν, που έχουμε n ανεξάρτητες μεταβλητές το γραμμικό μοντέλο που χρησιμοποιούμε έχει την παρακάτω μορφή:

$$y_i = b_0 + b_1 * x_1 + b_2 * x_{12} + \cdots + b_n * x_n + e_i$$

Όπου

- x_i είναι οι ανεξάρτητες μεταβλητές

-y_i η εξαρτημένη μεταβλητή

-b_i οι παράμετροι του μοντέλου.

-e_i είναι το τυχαίο σφάλμα για την i-οστή παρατήρηση.

3.5 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ

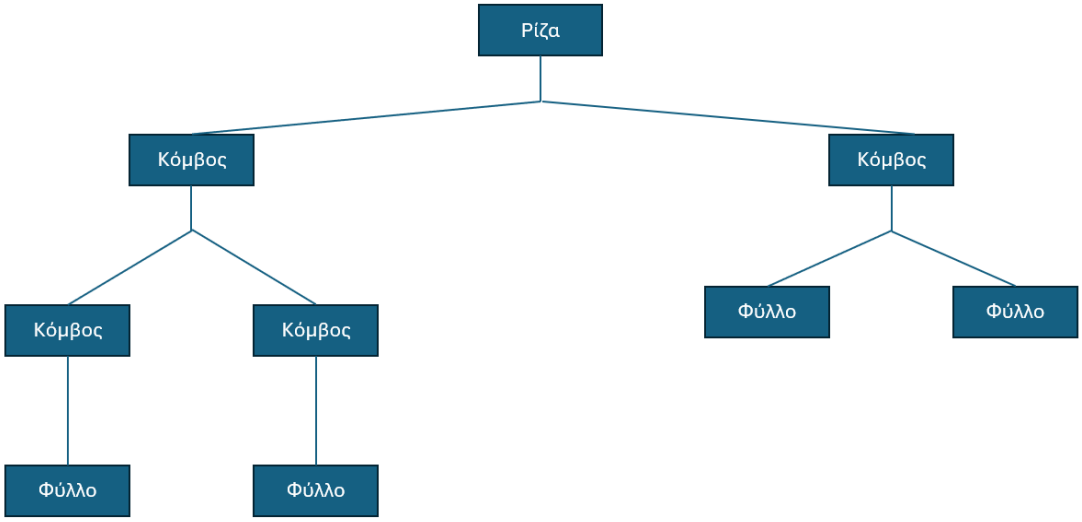
Κατηγοριοποίηση είναι η διαδικασία εκμάθησης μιας συνάρτησης στόχου f που αντιστοιχίζει κάθε σύνολο χαρακτηριστικών x σε μία από τις προκαθορισμένες κλάσης ετικέτας y. Αποτελεί μία από τις γνωστότερες τεχνικές εξόρυξης δεδομένων.

Η συνάρτηση στόχος είναι γνωστή και ως μοντέλο ταξινόμησης. Τα μοντέλα ταξινόμησης είναι σημαντικά για δύο βασικούς λόγους. Πρώτον, βοηθά στην περιγραφική μοντελοποίηση. Είναι δηλαδή ένα χρήσιμο εργαλείο για τη διάκριση αντικειμένων σε διαφορετικές κλάσεις. Δεύτερον, ένα μοντέλο ταξινόμησης μπορεί να χρησιμοποιηθεί για την πρόβλεψη ετικετών σε νέες παρατηρήσεις που δεν έχουν κάποια ετικέτα ακόμα.

Ας δούμε παρακάτω κάποιες από τις πιο γνωστές τεχνικές κατηγοριοποίησης.

3.5..1 ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ

Μια από τις πιο γνωστές μεθόδους κατηγοριοποίησης αποτελούν τα δέντρα απόφασης. Το δέντρο απόφασης αποτελείται από κόμβους οι οποίοι αποτελούν μία ιεραρχική δομή. Κόμβος είναι το κάθε σημείο του δέντρου απόφασης όπου λαμβάνεται μία απόφαση κατάτμησης. Στο άνω μέρος βρίσκεται ο κόμβος ρίζα. Ο κόμβος αυτός δεν έχει καμία ακμή που να εισέρχεται σε αυτόν. Αποτελεί δηλαδή την αρχή του δέντρου. Οι υπόλοιποι κόμβοι του δέντρου έχουν ακριβώς μία ακμή που εισέρχεται προς αυτούς και μία ή παραπάνω ακμές που εξέρχονται από αυτούς. Αν ένας κόμβος δεν έχει ακμές που να εξέρχονται από αυτόν ονομάζεται φύλλο του δέντρο.



Εικόνα 6. Παράδειγμα δομής δέντρου απόφασης

Τα δέντρα απόφασης λειτουργούν με κριτήρια διαχωρισμού. Πιο συγκεκριμένα, κατά τη διαδικασία της κατάτμησης (partitioning) το σύνολο δεδομένων διαχωρίζεται σε υποσύνολα με βάση τα κριτήρια που έχουμε ορίσει σε κάθε κόμβο. Γνωστοί αλγόριθμοι που εφαρμόζουν δέντρα απόφασης είναι ο ID3 , ο C4.5 και ο Random Forest.

Γενικά προτιμούμε δέντρα με μικρή πολυπλοκότητα ώστε να είναι εύκολα κατανοητά και να είναι ακριβή. Οι μετρικές που χρησιμοποιούμε για να προσδιορίσουμε την πολυπλοκότητα των δέντρων αποφάσεων είναι :

-ο συνολικός αριθμός κόμβων

-ο συνολικός αριθμός φύλλων

-το βάθος του δέντρου(μεγαλύτερο μονοπάτι, δηλαδή η μεγαλύτερη διαδρομή ακμών από την ρίζα του δέντρου ως τα φύλλα)

-ο αριθμός των κριτηρίων κατάτμησης που χρησιμοποιούνται

Ο αλγόριθμος ID3 χρησιμοποιεί σαν κριτήριο για τον προσδιορισμό του καλύτερου χαρακτηριστικού διάσπασης το κέρδος πληροφορίας (information gain). Το κέρδος πληροφορίας μετριέται ποσοτικά με την Εντροπία. Η Εντροπία εκφράζει το μέγεθος της ανομοιογένειας σε ένα σύνολο δεδομένων. Για παράδειγμα αν όλα τα δεδομένα μας ανήκουν σε μία κλάση τότε η εντροπία είναι 0. Το ζητούμενο σε ένα Δέντρο Αναζήτησης είναι ο διαχωρισμός του συνόλου εκπαίδευσης , με έναν επαναληπτικό τρόπο, σε υποσύνολα μηδενικής εντροπίας. Παρακάτω βλέπουμε τον τύπο που εκφράζει την εντροπία: Δεδομένων των πιθανοτήτων $p1$, $p2$, … , pk όπου ∑ pi = 1, η εντροπία Ε ορίζεται ως εξής:

$$E(S) = E(p_1, p_2, \dots, p_k) = \sum \left(p_i \, log \frac{1}{p} \right)$$

Το κέρδος πληροφορίας δίνεται από τον τύπο :

$$G(S, X) = E(\mathrm{S}) - \sum \left(\frac{|S_i|}{|S|} \right) E(S_i)$$

Όπου S_i υποσύνολο του S, που περιέχει τα παραδείγματα του S με τιμή x_i για το χαρακτηριστικό X. Ο ID3 προτιμά τα μικρότερα δέντρα από τα μεγαλύτερα και τοποθετεί χαρακτηριστικά με υψηλό κέρδος πληροφορίας πιο κοντά στην ρίζα. Πρόκειται για έναν hill climbing αλγόριθμο που δεν κάνει οπισθοδρόμηση(backtracking) με αποτέλεσμα πολλές φορές να οδηγηθεί σε τοπικό βέλτιστο (και όχι στην βέλτιστη λύση του προβλήματος)

Παρακάτω βλέπουμε ένα από τα πιο γνωστά παραδείγματα εκπαιδευτικού σκοπού στα δέντρα απόφασης. Πρόκειται για ένα δέντρο απόφασης για το ερώτημα PlayTennis με δοθείσες συγκεκριμένες καιρικές συνθήκες.

Έχουμε τα παρακάτω δεδομένα:

Outlook	Temperature	Humidity	Wind	Play Tennis
SUNNY	HOT	HIGH	WEAK	NO
SUNNY	HOT	HIGH	STRONG	NO
OVERCAST	HOT	HIGH	WEAK	YES
RAIN	MILD	HIGH	WEAK	YES
RAIN	COOL	NORMAL	WEAK	YES
RAIN	COOL	NORMAL	STRONG	NO
OVERCAST	COOL	NORMAL	STRONG	YES
SUNNY	MILD	HIGH	WEAK	NO
SUNNY	COOL	NORMAL	WEAK	YES
RAIN	MILD	NORMAL	WEAK	YES
SUNNY	MILD	NORMAL	STRONG	YES
OVERCAST	MILD	HIGH	STRONG	YES
OVERCAST	HOT	NORMAL	WEAK	YES
RAIN	MILD	HIGH	STRONG	NO

Αρχικά, και για απλότητα, θα δώσουμε τις παρακάτω ονομασίες στα χαρακτηριστικά. Outlook: O = { sunny, overcast, rain }

Temperature: T= {hot, mild, cool }

Humidity: H = { high, normal }

Wind: W= {weak, strong }

Χαρακτηριστικό στόχου αποτελεί η συνάρτηση

PlayTennis: PT(yes,no)

η οποία έχει δύο διακριτές τιμές. Στην πρώτη επανάληψη, πρέπει να γνωρίζουμε ποιο είναι το καλύτερο χαρακτηριστικό για να επιλέξουμε ως ρίζα στο δέντρο αποφάσεών μας. Για να γίνει αυτό, το ID3 θα βρει το καλύτερο χαρακτηριστικό που έχει το μέγιστο κέρδος πληροφορίας.

attributes = [O, H, W, T]

G(x, O) = 0.246

G(x, H) = 0.151

G(x, W) = 0.048

G(x, T) = 0.029

Έτσι με βάση τα κέρδη πληροφοριών που υπολογίσαμε παραπάνω, θα επιλέξουμε για ρίζα του δέντρου μας το χαρακτηριστικό Outlook το οποίο θα έχει τρεις κλάδους : Sunny, Rain και Overcast.Για τις επόμενες επαναλήψεις το Outlook φεύγει από την λίστα των χαρακτηριστικών. Κατά την δεύτερη επανάληψη θέλουμε να εξετάσουμε ποιο είναι το καλύτερο χαρακτηριστικό για το κλάδο του Sunny. Το νέο x περιέχει τιμές του

Sunny. x = [x[0], x[1], x[7], x[8], x[10]]

attribute= [H, W, T] G(x, H) = .970 - (3/5) * 0 - (2/5) * 0 = .970 24

G(x, T) = .970 - (2/5) * 0 - (2/5) * 1 = .570

G(x, W) = .970 - (2/5) * 1 - (3/5) * .918 = .019

Έτσι, με βάση τα κέρδη πληροφοριών που υπολογίστηκαν παραπάνω, επιλέγουμε το χαρακτηριστικό Humidity ως χαρακτηριστικό στον κλάδο Sunny. Για τις επόμενες επαναλήψεις το Humidity φεύγει από την λίστα των χαρακτηριστικών. Για την τρίτη επανάληψη, ο επόμενος κόμβος είναι το χαρακτηριστικό Humidity που έχει δύο πιθανές τιμές {High, Normal}. Το κλαδί High έχει ετικέτα No και το κλαδί Normal έχει ετικέτα Yes. Στην τέταρτη επανάληψη του αλγορίθμου το χαρακτηριστικό Outcast κυριαρχείται από μόνο μία ετικέτα (Yes). Επομένως η επανάληψη τελειώνει βάζοντας στο χαρακτηριστικό Outcasttο φύλλο Yes. Κατά την πέμπτη επανάληψη θέλουμε να εξετάσουμε το κλάδο Rain.

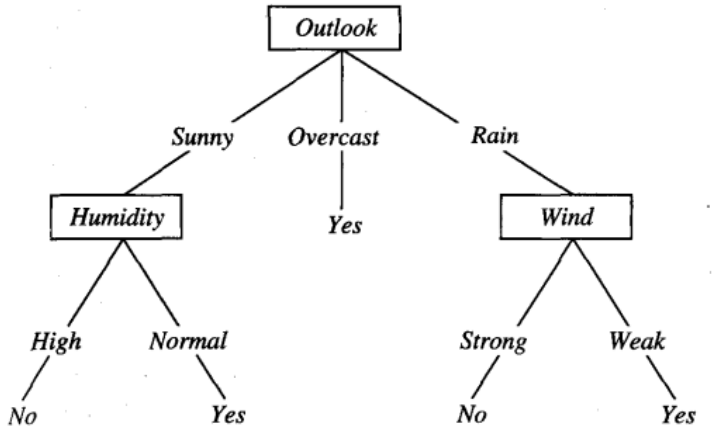
x = [x[3], x[4], x[5], x[9], x[13]]

attribute = [W, T]

G(x, W) = .970 - (2/5) * 0 - (3/5) * 0 = .970

G(x, T) = .970 - (0/5) * 0 - (3/5) * .918 - (2/5) * 1.0 = .019

Έτσι, με βάση τα κέρδη πληροφοριών που υπολογίστηκαν παραπάνω, επιλέγουμε το χαρακτηριστικό Wind ως χαρακτηριστικό στον κλάδο του Rain. Στην έκτη επανάληψη του ID3 ο κόμβος είναι το attribute Wind που έχει δύο πιθανές τιμές {Weak, Strong}. Το κλαδί Strong που κυριαρχείται από μία ετικέτα που είναι No και το κλαδί Weak που κυριαρχείται από την ετικέτα Yes. Στην έβδομη και τελευταία επανάληψη, όλα τα κλαδιά στο δέντρο αποφάσεων μας τελείωσαν με φύλλα. Με άλλα λόγια, κλαδεύουμε το χαρακτηριστικό Temperature από το δέντρο αποφάσεών μας. Το δέντρο που δημιούργησε ο αλγόριθμος ID3 είναι :



Εικόνα 7. Αποτελέσματα ID3 αλγορίθμου

Σε κάθε κόμβο μπορούμε να δούμε να αναγράφεται και το κριτήριο διαχωρισμού. Στις ακμές των κόμβων αναγράφονται οι τιμές που πρέπει να τηρούνται για το κριτήριο ώστε να ακολουθηθεί το αντίστοιχο μονοπάτι. Κάθε αταξινόμητη παρατήρηση ταξινομείται μέσω του δέντρου στον κατάλληλο κόμβο επιστρέφοντας τελικά την ετικέτα που σχετίζεται με το τελικό φύλλο. Στο παράδειγμά μας η ετικέτα μπορεί να πάρει δύο τιμές : Ναι ή Όχι.

Αν έχουμε λοιπόν την παρατήρηση :

<Outlook=Rain , Humidity=High , Wind=Strong >

τότε το δέντρο θα μας επιστρέψει για ετικέτα την τιμή Όχι.

3.5..2 KNN

Ο KNN αποτελεί έναν από τους πιο γνωστούς αλγορίθμους ταξινόμησης της εποπτευόμενης μάθησης. Ανήκει στην κατηγορία των lazy αλγορίθμων. Με τον όρο lazy εννοούμε τους αλγορίθμους που βασίζονται σε παραδείγματα. Οι περισσότεροι αλγόριθμοι ταξινόμησης εποπτευόμενης μάθησης δημιουργούν ένα μοντέλο κατά τη διάρκεια της εκπαίδευσης. Αντίθετα, ο KNN απομνημονεύει το σύνολο δεδομένων εκπαίδευσης και αναβάλει την επεξεργασία του έως ότου απαιτηθεί μια πρόβλεψη για μία νέα παρατήρηση

Αυτή η διαδικασία μπορεί να ακούγεται μη αξιόπιστη τεχνική αλλά δεν είναι. Σε δυναμικά περιβάλλοντα και σύνολα δεδομένων, όπου οι αναλυτές θέλουν να προσθαφαιρούν παρατηρήσεις από το σύνολο δεδομένων , οι αλγόριθμοι αυτοί μπορούν να φανούν εξαιρετικά χρήσιμοι. Αυτό το συμπέρασμα απορρέει από το γεγονός πως η οποιαδήποτε αλλαγή στο σύνολο δεδομένων εκπαίδευσης δείχνει την επιρροή της απευθείας στα αποτελέσματα των νέων παρατηρήσεων. Βέβαια, η διαδικασία αυτή κάνει τον knn να έχει υψηλό κόστος ταξινόμησης νέων περιπτώσεων.

Τα βασικά σημεία του knn που θα πρέπει να θέσουμε για την εκτέλεση του αλγορίθμου είναι

1. το σύνολο των αντικειμένων με ετικέτα που θα χρησιμοποιηθεί για την αξιολόγηση μίας παρατήρησης δοκιμής
2. ο ορισμός του μέτρου απόστασης ή ομοιότητας για τον υπολογισμό της εγγύτητας των αντικειμένων
3. η τιμή του k. Η τιμή αυτή υποδηλώνει τον αριθμό των πλησιέστερων γειτόνων που θα χρησιμοποιήσει η νέα παρατήρηση για να ταξινομηθεί
4. η μέθοδος για τον προσδιορισμό της κλάσης για μία παρατήρηση δοκιμής

Ο αλγόριθμος knn υποθέτει πως όλες οι παρατηρήσεις είναι σημεία στον n-διάστατο χώρο R^n . Η απόσταση δύο παρατηρήσεων υπολογίζεται συνήθως από την Ευκλείδια απόσταση. Μπορούμε όμως να χρησιμοποιήσουμε και άλλες αποστάσεις όπως για παράδειγμα την απόσταση Μανχάταν.

Η απόσταση που μεταξύ δύο παρατηρήσεων X,Y θεωρείται η Ευκλείδεια απόσταση, συμβολίζεται με d(X,Y) και υπολογίζεται σύμφωνα με τον παρακάτω τύπο:

$$d(X,Y) = \sqrt{((x_1 - y_1)^2 - (x_2 - y_2)^2)}$$

με x_1, y_1 και x_2, y_2 οι συντεταγμένες των δύο σημείων. . Κατ' αντιστοιχία ο αλγόριθμος μπορεί να λειτουργήσει και για παρατηρήσεις με n αριθμητικές διαστάσεις. Τότε η Ευκλείδεια απόσταση υπολογίζεται σύμφωνα με τον παρακάτω τύπο:

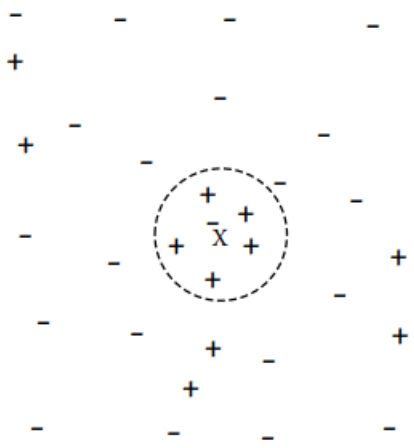
$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Για να εφαρμοστεί ο knn θα πρέπει να επιλέξουμε μία κατάλληλη τιμή για το k. Η επιτυχία ή μη της εκτέλεσης του αλγορίθμου εξαρτάται σε πολύ μεγάλο βαθμό από την επιλογή αυτή. Υπάρχουν διάφοροι τρόποι για να επιλέξουμε τιμή για το k. Ένας απλός και αποτελεσματικός τρόπος είναι να εκτελέσουμε πολλές φορές τον αλγόριθμο για διαφορετικές τιμές του k και να επιλέξουμε την τιμή που μας δίνει τα καλύτερα αποτελέσματα.

Ας δούμε τα βήματα που θα ακολουθούσαμε σε μία εκδοχή του knn :

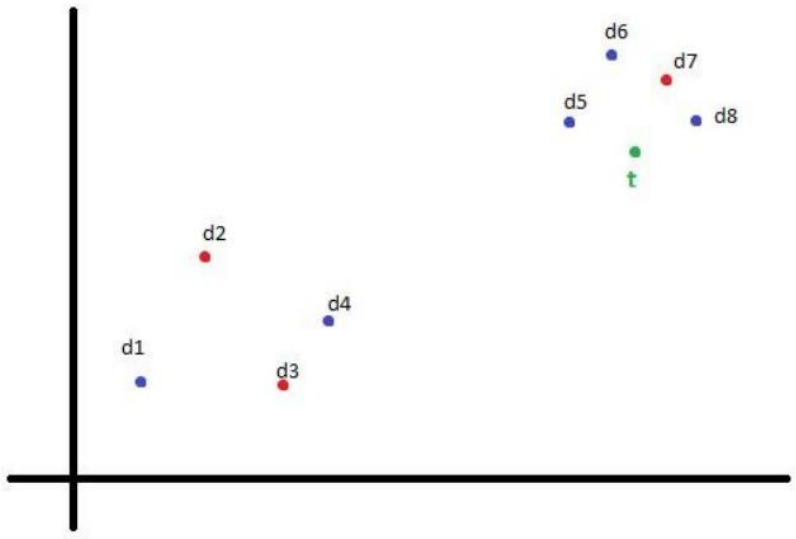
Ο αλγόριθμος λαμβάνει ως είσοδο ένα σύνολο εκπαίδευσης D και ένα αντικείμενο δοκιμής x, το οποίο είναι ένα διάνυσμα τιμών χαρακτηριστικών και άγνωστης ετικέτας κλάσης. Έχοντας δώσει τιμές στις παραμέτρους που είδαμε παραπάνω είμαστε σε θέση να εκτελέσουμε τον αλγόριθμο.

Πιο συγκεκριμένα, ο αλγόριθμος υπολογίζει την απόσταση(ή την ομοιότητα) μεταξύ του x και όλων των παρατηρήσεων του συνόλου εκπαίδευσης για να προσδιορίσει την λίστα των πλησιέστερων γειτόνων. Ακολούθως , ταξινομούμε τις αποστάσεις αυτές με αύξουσα σειρά και κρατάμε τις k πρώτες από αυτές. Βλέποντας πλειοψηφικά την κλάση που υπερτερεί δίνουμε κλάση και στη παρατήρηση δοκιμής x.



Εικόνα 8. Παράδειγμα παρατηρήσεων γύρω από ένα αντικείμενο δοκιμής x

Στην Εικόνα παρακάτω , βλέπουμε ένα παράδειγμα



Εικόνα 9. Παράδειγμα παρατηρήσεων γύρω από ένα αντικείμενο δοκιμής t

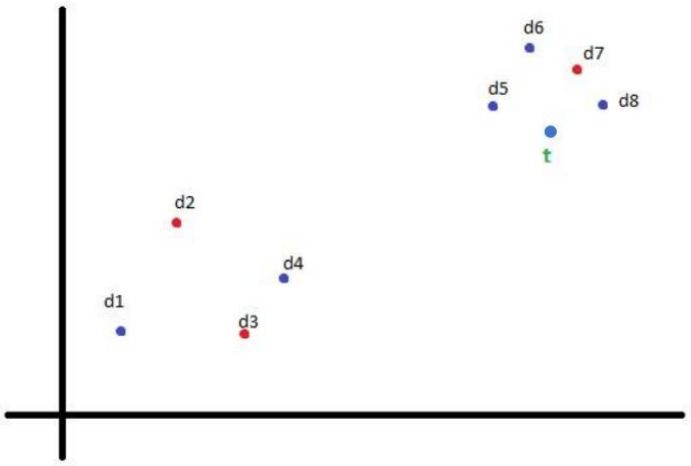
Αρχικά, διακρίνουμε δύο κλάσεις, οι οποίες συμβολίζονται με διαφορετικό χρώμα (κόκκινο, μπλε). Με πράσινο συμβολίζουμε την νέα παρατήρηση. Έστω, ότι έχουμε ορίσει k=4. Για αρχή μετράμε τις αποστάσεις των παρατηρήσεων του χώρου από την νέα παρατήρηση και τις τοποθετούμε σε έναν πίνακα.

d1t	d2t	d3t	d4t	d5t	d6t	d7t	d8t

Έπειτα, ταξινομούμε τον πίνακα σε αύξουσα σειρά.

d8t	d5t	d7t	d6t	d4t	d3t	d2t	d1t

Από τον νέο πίνακα που προέκυψε παίρνουμε τις 4 πρώτες καταχωρήσεις .(d8t ,d5t,d7t,d6t) και βλέπουμε τις κλάσεις-χρώμα στις οποίες ανήκουν. Οι τρεις παρατηρήσεις έχουν μπλε χρώμα και η τέταρτη κόκκινο. Αυτό σημαίνει ότι η νέα παρατήρηση θα εκχωρηθεί στην ΄΄μπλε΄΄ κλάση.



Εικόνα 10. Εκχώρηση κλάσης στο αντικείμενο δοκιμής t

Η πολυπλοκότητα του αλγορίθμου είναι O(n) καθώς για κάθε παρατήρηση θα πρέπει να υπολογίσουμε την απόστασή της από κάθε παρατήρηση δοκιμής του συνόλου εκπαίδευσης. Από την άλλη πλευρά, δεν απαιτείται χρόνος κατασκευής του μοντέλου ταξινόμησης(για παράδειγμα ένα δέντρο απόφασης ή ένα διαχωριστικό επίπεδο). Έτσι, ο knn διαφέρει από τις περισσότερες τεχνικές ταξινόμησης, οι οποίες έχουν ακριβά στάδια κατασκευής μοντέλων αλλά πολύ φθηνά βήματα ταξινόμησης της τάξεως O(σταθερά).

3.5.3 ΛΟΓΙΣΤΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ

Η λογιστική παλινδρόμηση είναι ένας αλγόριθμος εποπτευόμενης μάθησης που πραγματοποιεί ταξινόμηση προβλέποντας την πιθανότητα ενός αποτελέσματος ή γεγονότος. Η λογιστική παλινδρόμηση αναλύει τη σχέση μεταξύ μιας ή περισσότερων ανεξάρτητων μεταβλητών και ταξινομεί τα δεδομένα σε διακριτές κλάσεις. Χρησιμοποιείται συνήθως στην προγνωστική μοντελοποίηση, όπου το μοντέλο εκτιμά τη μαθηματική πιθανότητα εάν ένα στιγμιότυπο ανήκει σε μια συγκεκριμένη κατηγορία ή όχι.

Σε αντίθεση με τη γραμμική παλινδρόμηση, η λογιστική παλινδρόμηση χρησιμοποιεί τη λογιστική συνάρτηση, γνωστή και ως σιγμοειδής συνάρτηση, για να μοντελοποιήσει την πιθανότητα ότι ένα στιγμιότυπο ανήκει σε μια συγκεκριμένη κλάση. Η σιγμοειδής συνάρτηση αναφέρεται σε μια καμπύλη σχήματος S που μετατρέπει οποιαδήποτε πραγματική τιμή σε μια περιοχή μεταξύ 0 και 1 και αποτελεί την συνάρτηση ενεργοποίησης για την λογιστική παλινδρόμηση. Έτσι, σε αυτό το μοντέλο η μεταβλητή απόκρισης Y έχει συνήθως δυαδικό χαρακτήρα.

Η σημαντικότερη διαφορά της λογιστικής παλινδρόμησης σε σχέση με την γραμμική είναι ο τύπος της μεταβλητής απόκρισης Y. Πιο συγκεκριμένα, στην λογιστική παλινδρόμηση η Y είναι κατηγορική ενώ στην γραμμική παλινδρόμηση είναι ποσοτική. Όπως θα δούμε σε λίγο άλλη μία διαφορά των δύο μεθόδων βρίσκεται στον τρόπο εκτίμησης των παραμέτρων α και b_i.

Η λογιστική παλινδρόμηση διακρίνεται σε τρεις κατηγορίες ανάλογα με τον τύπο της εξαρτημένης κατηγορικής μεταβλητής. Μπορούμε λοιπόν να έχουμε:

1. Δίτιμη ή δυαδική (Binary) εξαρτημένη μεταβλητή
   Σε αυτή την περίπτωση έχουμε δύο κατηγορίες. Για παράδειγμα, 0 ή 1 , ΝΑΙ ή ΟΧΙ, Επιτυχία ή Αποτυχία. Η μεταβλητή Y μπορεί να πάρει 0μόνο μία από τις δύο τιμές.

2. Τακτική (Ordinal) μεταβλητή
   Σε αυτή την κατηγορία η μεταβλητή μπορεί να πάρει τιμή από τρεις ή και περισσότερες κατηγορίες. Για παράδειγμα σε ένα ερώτημα το οποίο δίνει επιλογές ‘’Συμφωνώ απόλυτα’’, ‘’Συμφωνώ’’, ‘’Ούτε συμφωνώ, ούτε διαφωνώ’’, ‘’Διαφωνώ’’ και ‘’Διαφωνώ απόλυτα’’ η απάντηση θεωρείται τακτική. Άλλα παραδείγματα θα μπορούσαν να είναι το εύρος μίας βαθμολογίας από το 0 έως το 10 ή η κατάταξη ενός υλικού ως λεπτό , μεσαίο ή παχύ.
3. Ονομαστική ή πολυωνυμική μεταβλητή
   Αναφέρεται στις μεταβλητές οι οποίες δεν έχουν φυσική διαβάθμιση όπως ο καθορισμός μίας χρωματικής απόχρωσης ενός αντικειμένου η οποία μπορεί να περιέχει τόνους από πολλά βασικά χρώματα και να αποτελεί συνδυασμό αυτών.

Εφαρμογές της λογιστικής παλινδρόμησης έχουμε σε πολλούς τομείς της επιστήμης:

1. Μάρκετινγκ
   Στο τομέα του Μάρκετινγκ, θα μπορούσαμε να εφαρμόσουμε λογιστική παλινδρόμηση ώστε να προβλέψουμε την επιτυχία ή μη ενός προϊόντος με βάση τα χρήματα που επενδύθηκαν σε συνδυασμό με την τάση των αγορών για αντίστοιχα προϊόντα αλλά και την οικονομική κατάσταση του target group του προϊόντος.

2. Υγεία
   Στον τομέα της υγείας θα μπορούσαμε να εφαρμόσουμε λογιστική παλινδρόμηση για να δούμε αν ένας ασθενής πρόκειται να εμφανίσει ή όχι μία νόσο με βάση τα διάφορα χαρακτηριστικά του όπως η ηλικία, το φύλλο, οι διατροφικές συνήθειες, το ιατρικό ιστορικό και οι εξετάσεις του.

3. Περιβάλλον
   Στο περιβάλλον θα μπορούσαμε να χρησιμοποιήσουμε την λογιστική παλινδρόμηση ώστε να προβλέψουμε την ρύπανση της Αθήνας, μετά τις φωτιές στα δάση του νομού τα τελευταία χρόνια, την αύξηση των οχημάτων στους δρόμους αλλά και την βιομηχανική δραστηριότητα στις παραπλήσιες περιοχές.

4. Εκπαίδευση
   Τέλος, στον τομέα της εκπαίδευσης μπορούν να αναπτυχθούν μοντέλα πρόβλεψης επιδόσεων των μαθητών σε σχέση με το προφίλ τους βασισμένα στην λογιστική παλινδρόμηση. Πιο συγκεκριμένα, θα μπορούσαμε να δούμε πόσο επηρεάζουν τις επιδόσεις των μαθητών οι κοινωνικές συναναστροφές, το κάπνισμα, ο ύπνος, η έλλειψη ελεύθερου χρόνου, η οικογενειακή αλλά και η οικονομική τους κατάσταση. Με αυτόν τον τρόπο οι καθηγητές θα έχουν σημαντικά στοιχεία στα χέρια τους και θα μπορούν πιο εύκολα να καθορίσουν το εκπαιδευτικό πλαίσιο της τάξης τους για μία εκπαιδευτική χρονιά.

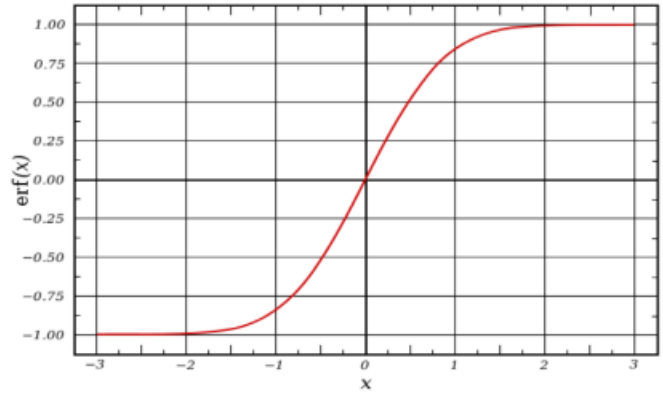
Σε αυτό το σημείο να αναφέρουμε πως κατά την εκτέλεση του αλγορίθμου καθορίζεται και ένα κατώφλι το οποίο μας δείχνει την τιμή στην οποία γίνεται ο διαχωρισμός των κλάσεων. Για παράδειγμα, έστω ότι θέλουμε να εξετάσουμε αν ένας φοιτητής έχει περάσει την εξέταση ενός μαθήματος. Το πρόβλημα αυτό έγκειται σε δυαδικό πρόβλημα με τις κλάσεις Επιτυχία (1) ή Αποτυχία(0). Έστω πως θέτουμε σαν όριο επιτυχίας το 0,5. Έτσι, αν ο φοιτητής πετύχει επίδοση η οποία

μεταφράζεται σε τιμή 0,74 τότε η κλάση που θα τοποθετηθεί η συγκεκριμένη παρατήρηση είναι η Επιτυχία (1).

Η σιγμοειδής συνάρτηση ορίζεται ως :

$$f(z) = \frac{1}{1 + e^{-z}}$$

Σχηματικά, η σιγμοειδής συνάρτηση φαίνεται παρακάτω:



Εικόνα 11. Διάγραμμα σιγμοειδούς συνάρτησης

Έστω λοιπόν πως έχουμε μία μεταβλητή z η οποία δίνεται από την γραμμική σχέση:

$$z = a + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Όπου X_i είναι ανεξάρτητες μεταβλητές και τα α και β_i είναι σταθεροί όροι που αντιπροσωπεύουν παραμέτρους της μεταβλητής z. Εφαρμόζοντας την σιγμοειδή συνάρτηση στην παραπάνω γραμμική σχέση θα πάρουμε το τύπο του μοντέλου της λογιστικής παλινδρόμησης που θα είναι :

$$f(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(a+\sum \beta_i X_i)}}$$

3.5.4 SUPPORT VECTOR MACHINE

Τα Support Vector Machines (SVM) είναι ένας αλγόριθμος εποπτευόμενης μηχανικής μάθησης που χρησιμοποιείται για παλινδρόμηση, ταξινόμηση αλλά και σε άλλες εφαρμογές. Η αρχή για την θεωρία πίσω από τα SVM έγινε από τους Vapnik και Lerner το 1963. Βασιζόμενοι στην λογική ότι δύο κατηγορίες παραδειγμάτων εκπαίδευσης μπορούν να διαχωριστούν από ένα υπερεπίπεδο πρότειναν ως βέλτιστο υπερεπίπεδο αυτό που διαχωρίζει τα παραδείγματα εκπαίδευσης και τους δίνει το μεγαλύτερο περιθώριο. Το υπερεπίπεδο αυτό διαχωρίζει τις κλάσεις και λειτουργεί ως συνάρτηση απόφασης με τις νέες παρατηρήσεις να κατηγοριοποιούνται ανάλογα από ποια πλευρά του υπερεπιπέδου βρίσκονται. Το υπερεπίπεδο ορίζεται ως εξής :

$$w^T x + b = 0$$

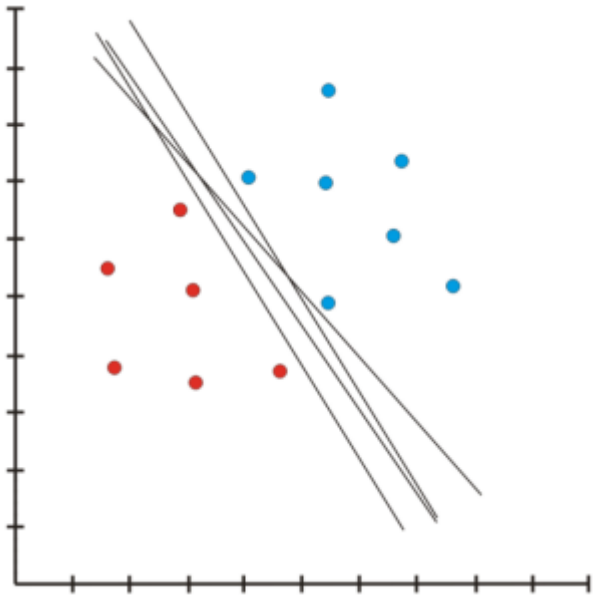
όπου:

-w είναι το διάνυσμα βαρών το οποίο είναι κάθετο στο επίπεδο και ορίζει τον προσανατολισμό του

-b είναι το κατώφλι και ως τιμή καθορίζει την παράλληλη μετατόπιση του επιπέδου

-και x είναι η υπό εξέταση παρατήρηση

Όπως μπορούμε να δούμε στο παρακάτω σχήμα στον χώρο μπορεί να υπάρχουν πολλά υπερεπίπεδα

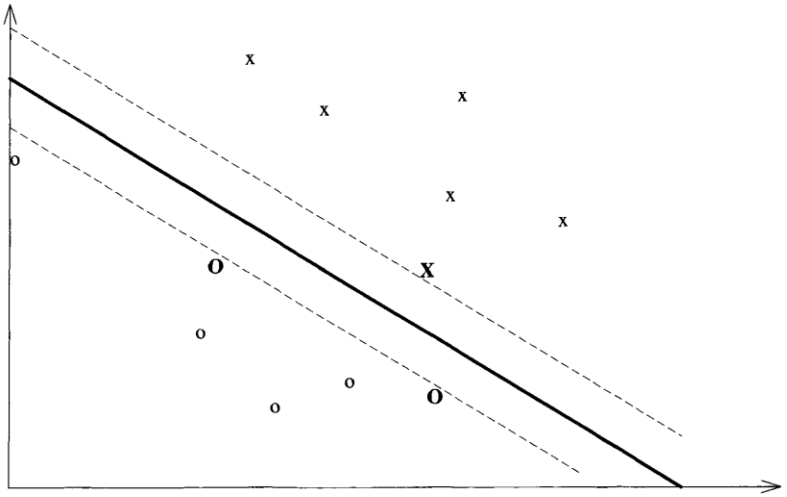


Εικόνα 12. Παράδειγμα διαγράμματος με υπερεπίπεδα

Αυτό που προσπαθούμε να κάνουμε στην πραγματικότητα είναι να βρούμε το υπερεπίπεδο εκείνο το οποίο μεγιστοποιεί το περιθώριο (margin) .

Επομένως, τα SVM δεν προσπαθούν μόνο να κάνουν σωστή ταξινόμηση στα δεδομένα, αλλά έχουν ως στόχο και την μεγιστοποίηση του περιθωρίου ώστε να πετύχει την καλύτερη απόδοση γενίκευσης. Το περιθώριο είναι η μικρότερη απόσταση μίας παρατήρησης από το υπερεπίπεδο. Η γενίκευση αναφέρεται στο γεγονός ότι ένας ταξινομητής δεν έχει μόνο καλή απόδοση ταξινόμησης (π.χ. ακρίβεια) στα δεδομένα εκπαίδευσης, αλλά εγγυάται επίσης υψηλή προγνωστική ακρίβεια για τα μελλοντικά δεδομένα από την ίδια κατανομή με τα δεδομένα εκπαίδευσης.

Ας δούμε την παρακάτω εικόνα σαν παράδειγμα

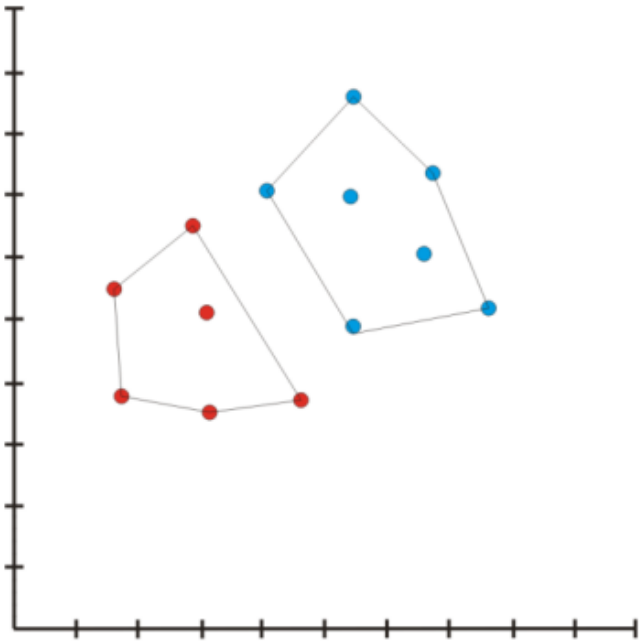


Εικόνα 13. Παράδειγμα επιλογής υπερεπιπέδου

Έχουμε λοιπόν ένα σύνολο δεδομένων με δύο κατηγορίες x και ο. Για να διαχωριστούν οι δύο κατηγορίες, υπάρχουν πολλά πιθανά υπερεπίπεδα που θα μπορούσαν να επιλεγούν. Ο στόχος μας είναι να βρούμε ένα επίπεδο που έχει το μέγιστο περιθώριο, δηλαδή τη μέγιστη απόσταση μεταξύ των σημείων δεδομένων και των δύο κατηγοριών. Σε αυτή την περίπτωση, τα δεδομένα κάθε κατηγορίας δημιουργούν νοητά δύο χώρους. Το ερώτημα που προκύπτει εύλογα είναι τι θα γινόταν αν μία παρατήρηση της κατηγορίας x βρισκόταν ανάμεσα σε παρατηρήσεις κατηγορίας ο. Σε μία τέτοια περίπτωση ο αλγόριθμος SVM θα αναγνώριζε αυτή την αποκλείνουσα παρατήρηση σας outlier και το αποτέλεσμα δεν θα άλλαζε.

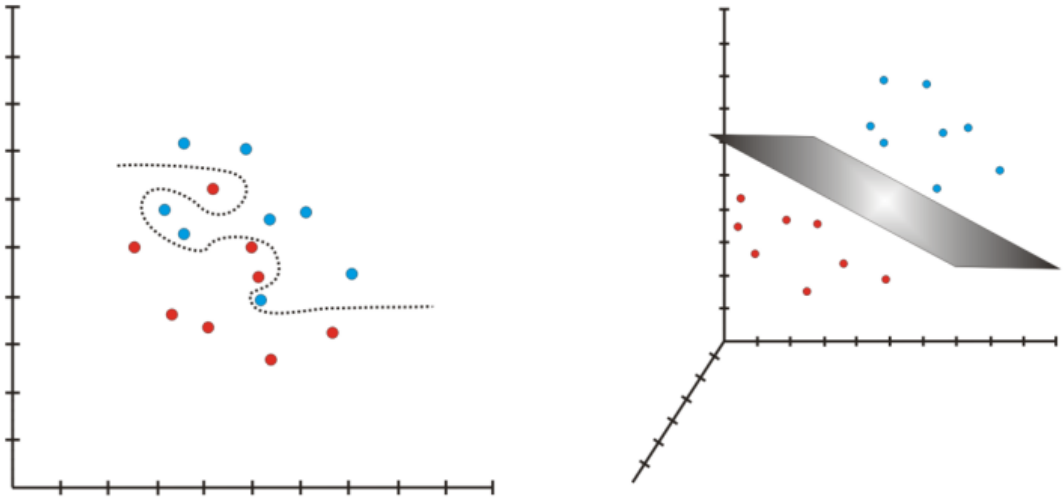
Υπάρχουν δύο βασικά είδη SVM

1. Γραμμικά SVM
   Τα γραμμικά SVM χρησιμοποιούνται για την ταξινόμηση γραμμικά διαχωρίσιμων δεδομένων. Ένα σύνολο δεδομένων λέμε ότι είναι γραμμικά διαχωρίσιμο όταν μπορεί να χωριστεί σε κατηγορίες ή κλάσεις με την χάραξη μίας ευθείας γραμμής αν μιλάμε για τον 2D χώρο ή από ένα υπερεπίπεδο αν μιλάμε για υψηλότερες διαστάσεις. Στην παρακάτω εικόνα βλέπουμε παρατηρήσεις που δημιουργούν οπτικά την έννοια των ομάδων και είναι εμφανές πως αποτελούν παρατηρήσεις γραμμικά διαχωρίσιμες.



Εικόνα 14. Παράδειγμα γραμμικού SVM

2. Μη γραμμικά SVM
   Τα μη γραμμικά SVM δεν μπορούν να διαχωρίσουν τα δεδομένα σε διακριτές κατηγορίες με την βοήθεια μίας μόνο γραμμής. Για να μπορέσουμε σε αυτή την περίπτωση να κάνουμε τον διαχωρισμό θα πρέπει να χρησιμοποιήσουμε τις συναρτήσεις πυρήνα (kernel functions). Μέσα από αυτές τις συναρτήσεις τα αρχικά δεδομένα εισόδου μετασχηματίζονται προσθέτοντάς τους χαρακτηριστικά σε υψηλότερες διαστάσεις κάνοντάς τα γραμμικά διαχωρίσιμα. Όπως βλέπουμε στην παρακάτω εικόνα αρχικά οι παρατηρήσεις μας στον δισδιάστατο χώρο δεν είναι γραμμικά διαχωρίσιμες καθώς δεν χωρίζονται με μία ευθεία γραμμή. Όταν όμως τα μετασχηματίζουμε και τα προβάλουμε στον τρισδιάστατο χώρο τότε μπορούμε με ένα υπερεπίπεδο να τα διαχωρίσουμε.



Εικόνα 15. Παράδειγμα μη γραμμικού SVM

Τα SVM παρουσιάζουν ένα βασικό μειονέκτημα και αυτό είναι η δαπανηρή υπολογιστική πολυπλοκότητά τους στην φάση της εκπαίδευσης. Το γεγονός αυτό οδηγεί σε δυσκολίες εφαρμογής για μεγάλα σύνολα δεδομένων. Λύση σε αυτό το πρόβλημα βελτιστοποίησης θα μπορούσε να είναι το σπάσιμο του σε μια σειρά από μικρότερα προβλήματα με καθένα από αυτά να περιλαμβάνει ένα μέρος του αρχικού προβλήματος. Ολοκληρώνοντας τα επιμέρους ΄κομμάτια΄ προβλημάτων θα έχουμε καταφέρει να επιλύσουμε και το βασικό μας πρόβλημα.

3.6 ΣΥΣΤΑΔΟΠΟΙΗΣΗ

Η συσταδοποίηση (clustering) αποτελεί μια περιγραφική μέθοδο. Στόχος της είναι να δημιουργεί συστάδες (clusters), δηλαδή ομάδες οι οποίες θα αποτελούνται από παρατηρήσεις με παρόμοια χαρακτηριστικά. Σημαντική διαφορά με την κατηγοριοποίηση είναι πως οι συστάδες δεν είναι γνωστές από την αρχή, όπως οι κατηγορίες (ή κλάσεις), αλλά δημιουργούνται κατά την διαδικασία ανάλυσης του εκάστοτε συνόλου δεδομένων.

Σε ένα σύνολο αντικειμένων, το να υπάρχει διαχωρισμός σε ομάδες είναι πολύ σημαντικός παράγοντας για την κατανόηση των αντικειμένων αυτών από τους ανθρώπους που καλούνται να τα αναλύσουν . Παρακάτω αναφέρουμε ενδεικτικά κάποιους τομείς στους οποίους φαίνεται έντονα η χρησιμότητα αυτή.

Βιολογία

Στην επιστήμη της βιολογίας οι επιστήμονες έχουν δημιουργήσει μία ταξινόμηση κατά την οποία μπορούμε να διαχωρίσουμε το ζωικό βασίλειο σε τάξεις, οικογένειες και είδη. Πιο πρόσφατα από αυτό, οι επιστήμονες κατάφεραν να αναλύσουν δεδομένα από μεγάλες ποσότητες γενετικών πληροφοριών που πλέον μας είναι

διαθέσιμες με απώτερο σκοπό την εύρεση ομάδων γονιδίων με παρόμοιες λειτουργίες.

Business

Οι επιχειρήσεις έχουν ως βασικό τους σκοπό την αύξηση των κερδών. Αυτό θα το επιτύχουν με την αύξηση των πελατών τους και την σωστή επιλογή συνεργατών. Με το clustering μπορούν να διαχωρίζουν τους πελάτες τους σε μικρότερα κομμάτια όπου το καθένα περιέχει μία ομάδα από αυτούς με κοινά χαρακτηριστικά και παρόμοιες ανάγκες. Έτσι, γίνεται ευκολότερη η δημιουργία υπηρεσιών για κάθε υπό ομάδα και επομένως οι επιχειρήσεις πετυχαίνουν και τον βασικό τους σκοπό που αναφέραμε παραπάνω.

Ανάκτηση πληροφοριών

Ο Παγκόσμιος Ιστός αποτελείται από δισεκατομμύρια ιστοσελίδες. Όταν ένας χρήστης κάνει μία αναζήτηση συχνά του επιστρέφονται χιλιάδες αποτελέσματα. Με το clustering μπορούμε να ομαδοποιήσουμε τις σελίδες σε μικρά clusters με το καθένα να εξυπηρετεί ένα κομμάτι του ερωτήματος του χρήστη. Για παράδειγμα, αν ένας χρήστης κάνει αναζήτηση με τον όρο ‘τένις’ θα του εμφανιστούν αποτελέσματα που έχουν ομαδοποιηθεί σε κατηγορίες όπως άθλημα, προγνωστικά, γήπεδο, αγώνες. Τα clusters αυτά μπορούν να ‘σπάσουν΄ σε υπό ομάδες. Για παράδειγμα, το clusters ‘τουρνουά΄ μπορεί να περιλαμβάνει τα clusters ‘Wimbledon’ , ‘Roland Garros’, ‘Australian Open’. Με αυτό τον τρόπο δημιουργείται μία ιεραρχική δομή στις ομάδες με αποτέλεσμα την καλύτερη εμπειρία για τον χρήστη.

3.6.1 KMEANS

Ο αλγόριθμος k-means είναι από τους πιο δημοφιλείς αλγορίθμους συσταδοποίησης. Παρακάτω θα δούμε τα βήματα που τον αποτελούν δίνοντας και ένα σύντομο παράδειγμα.

Έστω $X=\{x_i, i=1,\ldots,n\}$ ένα σύνολο n σημείων d-διαστάσεων που θα πρέπει να χωριστούν σε K clusters.

Ο αλγόριθμος εκτελείται στα παρακάτω τρία βήματα:

1 Αρχικά, ο αλγόριθμος κάνει μία τυχαία πρώτη εκτίμηση των K κεντροειδών.

2. Στην συνέχεια, κάθε νέο σημείο από το σύνολο δεδομένων τοποθετείται στο cluster με το πιο κοντινό κεντροειδές σημείο. Η απόσταση μεταξύ των σημείων καθορίζεται από το τετράγωνο της Ευκλείδειας απόστασης.

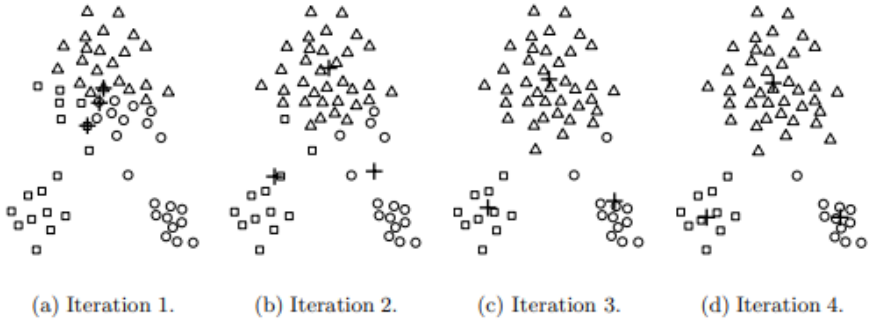
3.Έπειτα, υπολογίζουμε τα νέα κεντροειδή με βάση τον μέσο όρο των σημείων κάθε cluster. Ο υπολογισμός για το i-οστό cluster δίνεται από τον παρακάτω τύπο:

$$c_i = \frac{1}{m_i} \sum_{x \epsilon C_i} x$$

όπου Σx το σύνολο των παρατηρήσεων που βρίσκονται στο cluster και mi το πλήθος αυτών των παρατηρήσεων.

Τα βήματα 2 και 3 επαναλαμβάνονται έως ότου το κριτήριο τερματισμού εκπληρωθεί.

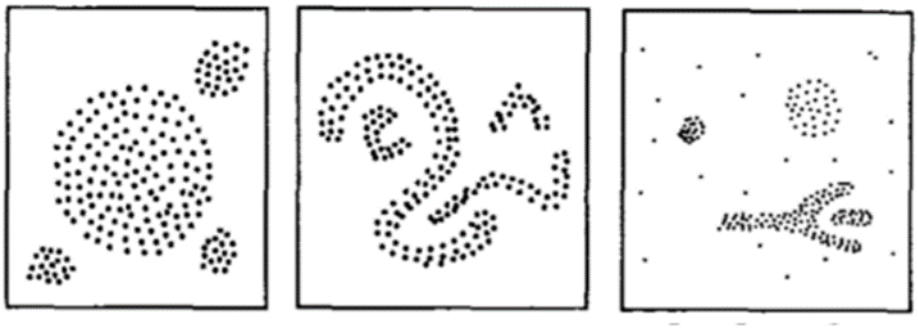
Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα εκτέλεσης του αλγορίθμου. Στο πρώτο βήμα βλέπουμε τα αρχικά σημεία αλλά και την πρώτη εκτίμηση των κεντροειδών. Για την εκτίμηση αυτή παίρνουμε τον μέσο όρο των παρατηρήσεων ως κριτήριο καθορισμού των αρχικών κεντροοειδών. Στις επόμενες επαναλήψεις, τα κεντροειδή ενημερώνονται και παίρνουν τις νέες τους τιμές. Όπως παρατηρούμε το σχήμα βλέπουμε την τάση των κεντροειδών να φεύγουν από τον μέσο όρο που τα καθόρισε στην αρχή και να κινούνται προς τις δύο μικρότερες ομάδες από παρατηρήσεις στα άκρα του συνόλου. Στην τελευταία επανάληψη ο αλγόριθμος έχει φτάσει σε ένα σημείο που δεν κάνει πλέον αλλαγές. Αυτό σημαίνει πως η ομαδοποίηση έχει ολοκληρωθεί.



Εικόνα 16. Οπτικοποίηση kmeans αλγορίθμου

Ο kmeans αποτελεί έναν γρήγορο και αποτελεσματικό αλγόριθμο. Παρόλα αυτά ο αλγόριθμος μπορεί εύκολα να εγκλωβιστεί σε τοπικά ελάχιστα κατά την διάρκεια ενημέρωσης των κεντροειδών.

3.6.2 DBSCAN



Εικόνα 17. Διαφορετικοί τύπου παρατηρήσεων

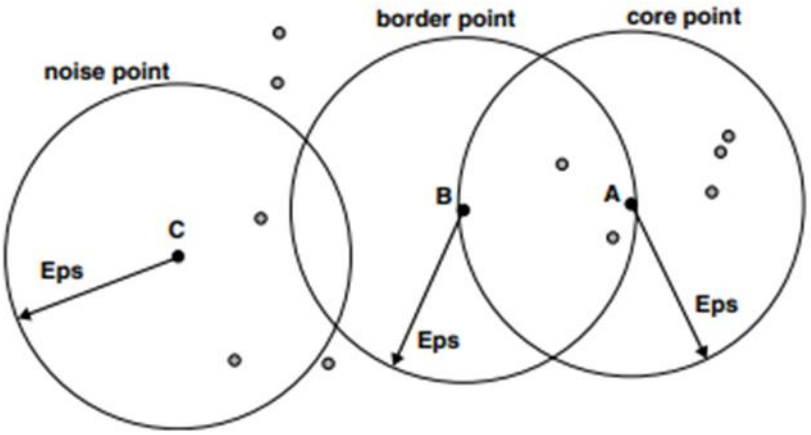
Στην εικόνα βλέπουμε σύνολα παρατηρήσεων διαφορετικού τύπου , πυκνότητας και σχήματος. Πυκνότητα είναι ο αριθμός των παρατηρήσεων σε μία ακτίνα Eps. Ο βασικότερος λόγος που μπορούμε να δούμε εύκολα τα σύνολα αυτά είναι η υψηλότερη πυκνότητα που παρουσιάζεται σε εντός συμπλεγμάτων σε σχέση με τις παρατηρήσεις εκτός αυτών. ορισμός.

Στην συνέχεια θα ορίσουμε την Eps-γειτονιά για μία παρατήρηση x. Γειτονιά, ονομάζουμε ένα σύνολο της μορφής:

με D το σύνολο των παρατηρήσεων και dist την συνάρτηση απόστασης.

Το σχήμα για μία γειτονιά δεν είναι συγκεκριμένο. Βασικός παράγοντας που το καθορίζει είναι η συνάρτηση απόστασης που θα χρησιμοποιήσουμε. Για παράδειγμα, με την απόσταση Manhattan στον δυσδιάστατο χώρο θα πάρουμε γειτονιές σε σχήμα ορθογωνίου, ενώ με την Ευκλείδεια απόσταση οι γειτονιές θα έχουν κυκλικό σχήμα.

Υπάρχουν τρία είδη παρατηρήσεων:



Εικόνα 18. Είδη παρατηρήσεων

-Κεντρικές παρατηρήσεις : Οι παρατηρήσεις αυτές έχουν πυκνότητα μεγαλύτερη ή ίση από μία τιμή MinPts. Πιο συγκεκριμένα, αν μία παρατήρηση περιβάλλεται από ένα συγκεκριμένο αριθμό παρατηρήσεων και πάνω, εντός μιας απόστασης που υπολογίζεται από την συνάρτηση απόστασης, τότε θεωρούμε πως είναι κεντρική. Οι κεντρικές παρατηρήσεις βρίσκονται στο εσωτερικό του συμπλέγματος που έχει οριστεί με βάση την πυκνότητα.

-Συνοριακές παρατηρήσεις: Οι παρατηρήσεις αυτές έχουν πυκνότητα μικρότερη από MinPts αλλά βρίσκονται εντός της απόστασης που έχει οριστεί από την αντίστοιχη συνάρτηση.

-Θορυβώδεις παρατηρήσεις: Σε αυτή την κατηγορία περιλαμβάνονται οι παρατηρήσεις που δεν ανήκουν σε κάποια από τις παραπάνω κατηγορίες.

Ο αλγόριθμος DBSCAN ακολουθεί τα παρακάτω βήματα:

1. Χαρακτηρίστε κάθε σημείο-παρατήρηση σε κεντρική , συνοριακή ή θορυβώδης

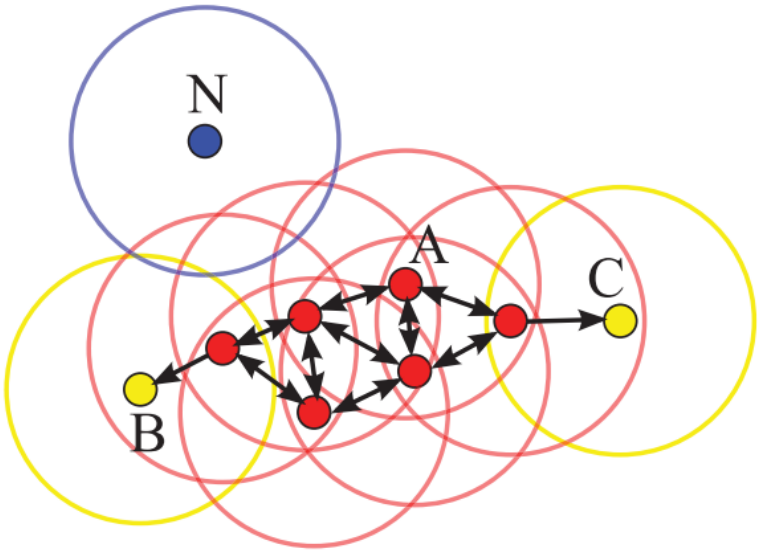
2. Αγνοήστε τα θορυβώδη σημεία

3. Δημιουργήστε έναν γράφο με μία κορυφή για κάθε παρατήρηση και στη συνέχεια τοποθετήστε μία ακμή μεταξύ των κεντρικών σημείων σε απόσταση έως Eps μεταξύ τους.

4. Θέστε κάθε ομάδα συνδεδεμένων κεντρικών σημείων ως μία διαφορετική συστάδα-σύμπλεγμα

5. Αναθέστε κάθε συνοριακό σημείο σε μία από τις συστάδες που περιέχει το πιο κοντινό του κεντρικό σημείο

Παρακάτω βλέπουμε μία εφαρμογή του DBSCAN αλγορίθμου:



Εικόνα 19. Οπτικοποίηση αλγορίθμου DBSCAN

Η παράμετρος minPts είναι 4 και η Eps υποδεικνύεται από τους κύκλους που περιβάλουν τις παρατηρήσεις.

-Το N είναι ένα σημείο θορύβου, το A είναι ένα κεντρικό σημείο και τα B και C είναι σημεία συνόρων.

-Τα βέλη υποδεικνύουν τις ακμές σύνδεσης των σημείων με βάση την πυκνότητα.

- Όπως μπορούμε να δούμε τα σημεία B και C συνδέονται με το κεντρικό σημείο A ενώ το σημείο N όχι. Αυτός είναι και ο λόγος που το σημείο N είναι σημείο θορύβου.

ΚΕΦΑΛΑΙΟ 4

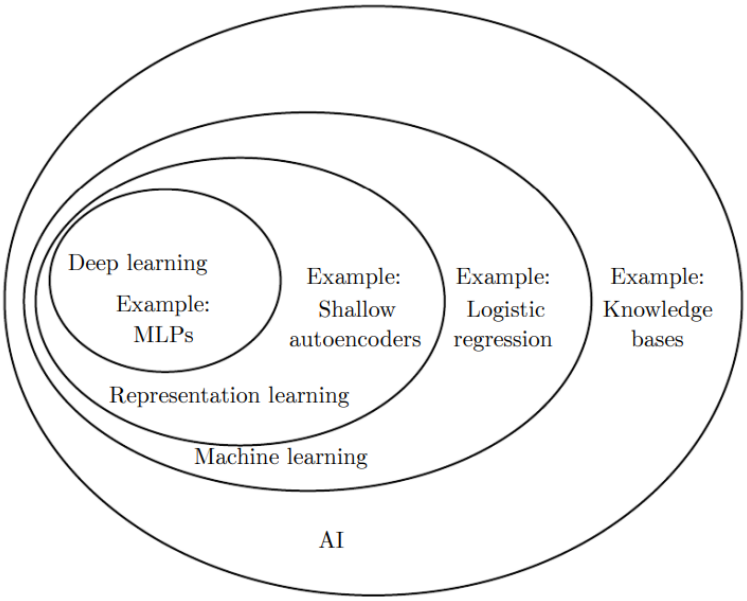
4.1 ΒΑΘΕΙΑ ΜΑΘΗΣΗ

Όπως αναφέραμε και σε προηγούμενη ενότητα, η μηχανική μάθηση έχει εφαρμογές σε πολλούς τομείς της κοινωνίας μας. Τα συστήματα μηχανικής μάθησης χρησιμοποιούνται για την αναγνώριση αντικειμένων σε εικόνες, την μετατροπή της ομιλίας σε κείμενο, την αντιστοίχιση ιστοσελίδων αλλά και προτάσεων αγορών σχετικά με την αναζήτηση των χρηστών. Οι εφαρμογές αυτές, αλλά και πολλές άλλες χρησιμοποιούν μία κατηγορία τεχνικών που ονομάζεται βαθιά μάθηση. Παλαιότερα γινόταν χρήση τεχνικών που περιόριζαν τους επιστήμονες στην επεξεργασία μεγάλων δεδομένων σε ακατέργαστη μορφή.

Με την βαθιά μάθηση παρέχονται τεχνικές που μας επιτρέπουν την δημιουργία εργαλείων εξαγωγής χαρακτηριστικών τα οποία μετασχηματίζουν ακατέργαστα δεδομένα όπως τα pixel μίας εικόνας ή ένα ηχητικό απόσπασμα σε διανύσματα χαρακτηριστικών τα οποία μπορούν να αποτελέσουν είσοδο σε ένα μοντέλο

ταξινόμησης. Οι τεχνικές αυτές αποτελούν μεθόδους μάθησης μέσω αναπαράστασης (Representation-learning methods) ,δηλαδή μεθόδων που επιτρέπουν σε μία μηχανή να λαμβάνει δεδομένα χωρίς επεξεργασία και να μπορεί να ανακαλύψει τις απαραίτητες αναπαραστάσεις που απαιτούνται για την ανίχνευση και την ταξινόμηση.

Οι μέθοδοι βαθιάς μάθησης έχουν πολλά επίπεδα αναπαράστασης. Συγκεκριμένα, τα δεδομένα εισόδου περνούν μέσα από πολλαπλά επίπεδα επεξεργασίας. Κάθε επίπεδο επεξεργάζεται την είσοδο με μετασχηματισμούς που αφαιρούν περιττές λεπτομέρειες και αναδεικνύουν τις σημαντικές λεπτομέρειες των δεδομένων. Όσο προχωράμε στα επίπεδα τόσο πιο αποτελεσματική γίνεται και η αναπαράσταση. Στο τέλος, καταλήγουμε να έχουμε μία πιο αφηρημένη αναπαράσταση που είναι κατάλληλη για την επίλυση του προβλήματός μας σε χαμηλότερα επίπεδα. Στο παρακάτω σχήμα βλέπουμε ένα διάγραμμα Venn στο οποίο βλέπουμε πως η βαθιά μάθηση αποτελεί κομμάτι του Representation learning, το οποίο αποτελεί μία μέθοδο μηχανικής μάθησης. Όλες αυτές οι μέθοδοι και οι τεχνικές βρίσκονται κάτω από την 'ομπρέλα' της επιστήμης της Τεχνητής Νοημοσύνης.

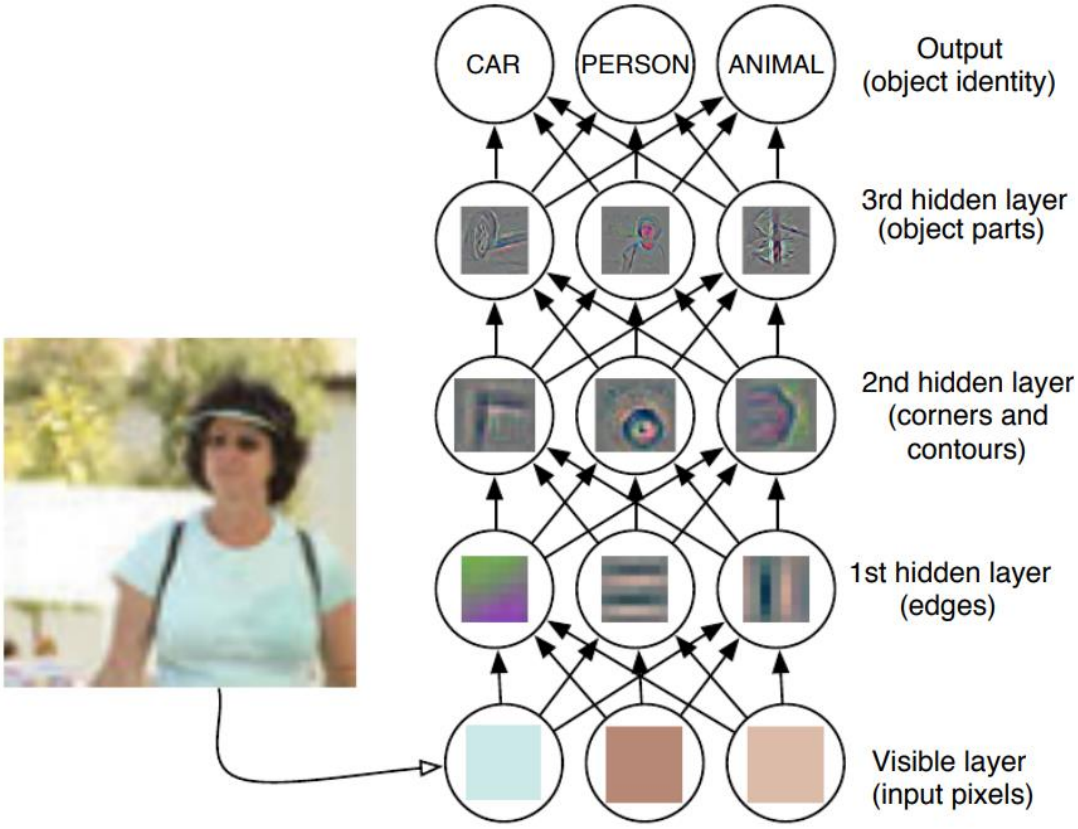


Εικόνα 20. Διάγραμμα Venn για Τεχνητή Νοημοσύνη

Παρακάτω θα προσπαθήσουμε να κάνουμε μία απεικόνιση ενός μοντέλου βαθιάς μάθησης σε δεδομένα εικόνας. Αρχικά, η είσοδος παρουσιάζεται στο πρώτο και ορατό επίπεδο. Ονομάζουμε το πρώτο επίπεδο ορατό επειδή περιέχει μεταβλητές που μπορούμε να παρατηρήσουμε. Στο πρώτο στρώμα μπορούμε να αναγνωρίσουμε τις άκρες συγκρίνοντας την φωτεινότητα σε γειτονικά pixel. Στη συνέχεια, ακολουθούν μια σειρά από κρυμμένα επίπεδα.

Το δεύτερο επίπεδο λαμβάνοντας ως είσοδο τα δεδομένα που έχουν περάσει από το πρώτο επίπεδο, μπορεί να αναγνωρίσει γωνίες και περιγράμματα που συντελούν

συλλογές άκρων. Έπειτα, ακολουθεί το τρίτο επίπεδο το οποίο μπορεί να ανιχνεύσει ολόκληρα μέρη αντικειμένων με την βοήθεια των συλλογών άκρων, περιγραμμάτων και γωνιών. Τέλος, οι συλλογές αυτές μπορούν να δώσουν συμπεράσματα ως προς την φύση, το σχήμα, το μήκος και άλλων χαρακτηριστικών των αντικειμένων.

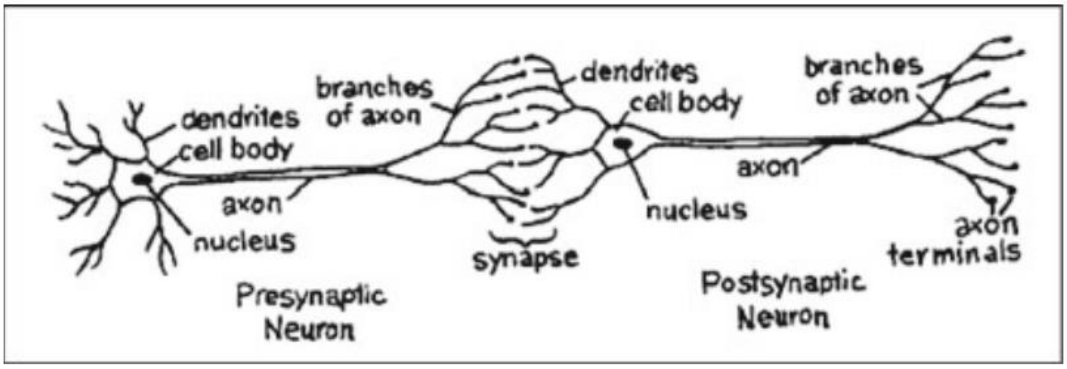


Εικόνα 21. Απεικόνιση μοντέλου βαθιάς μάθησης

Βασικό χαρακτηριστικό της βαθιάς μάθησης είναι η μη επέμβαση του ανθρώπου σε αυτά τα στρώματα χαρακτηριστικών. Μέσα από τα πολλά στρώματα που περνάει το σύνολο δεδομένων καταφέρνουμε να έχουμε αποτελεσματικές ταξινομήσεις και ανιχνεύσεις. Είναι γνωστό πως οι μέθοδοι βαθιάς μάθησης κατάφεραν να λύσουν προβλήματα τεχνητής νοημοσύνης που για χρόνια έμεναν άλυτα.

4.2 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

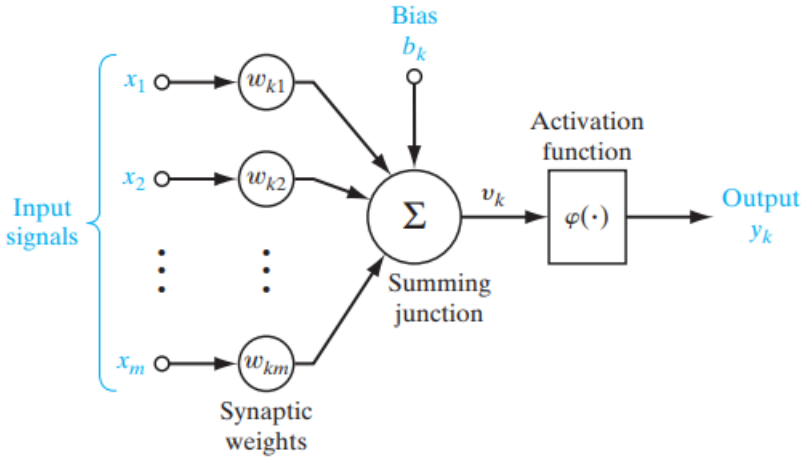
Τα τεχνητά νευρωνικά δίκτυα είναι μία δημοφιλής τεχνική μηχανικής μάθησης που προσομοιώνει τον μηχανισμό μάθησης του ανθρώπινου εγκεφάλου. Συγκεκριμένα, το ανθρώπινο νευρικό σύστημα περιέχει κύτταρα, τα οποία αναφέρονται ως νευρώνες.



Εικόνα 22.Απεικόνιση ανθρώπινου νευρικού συστήματος

Στα τεχνητά νευρωνικά δίκτυα τα βασικά στοιχεία που χαρακτηρίζουν έναν νευρώνα είναι :

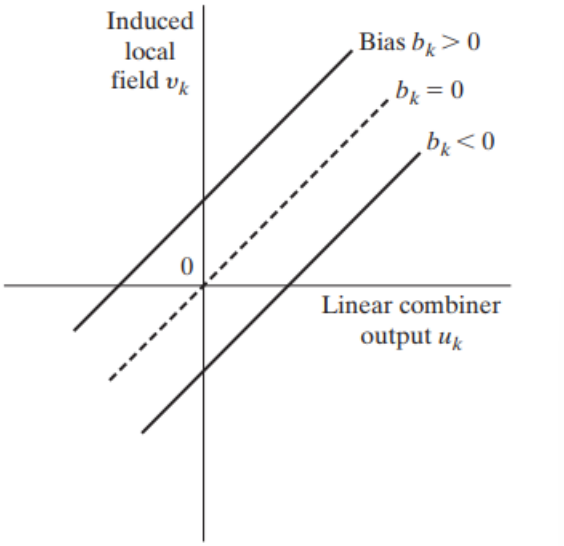
1. Ένα σύνολο συνάψεων που συνδέουν τους νευρώνες μεταξύ τους. Καθεμία από τις συνάψεις χαρακτηρίζεται από ένα βάρος. Συγκεκριμένα, ένα σήμα xj στην είσοδο της σύναψης συνδέεται με έναν νευρώνα k όπου πολλαπλασιάζεται με ένα βάρος w_{kj}. Σε αυτό το σημείο αξίζει να αναφέρουμε πως το βάρος w έχει δύο δείκτες, έναν για τον νευρώνα που αφορά και έναν για το άκρο του νευρώνα που αναφέρεται το βάρος.
2. Έναν αθροιστή ο οποίος αθροίζει όλα τα σήματα εισόδου συνυπολογίζοντας τα βάρη του καθενός.
3. Μία συνάρτηση ενεργοποίησης φ που καθορίζει την έξοδο του δικτύου. Βασικός της σκοπός είναι η εισαγωγή μη γραμμικότητας στο δίκτυο επιτρέποντάς του να μαθαίνει πολύπλοκα μοτίβα στα δεδομένα εισόδου. Στην παρακάτω εικόνα βλέπουμε σχηματικά ένα νευρωνικό δίκτυο



Εικόνα 23. Απεικόνιση νευρωνικού δικτύου

Το uk αποτελεί τον γραμμικό συνδυασμό των εισόδων του δικτύου. Όπως βλέπουμε στο σχήμα έχει προστεθεί και η πρόσθετη παράμετρος bias bk. Η παράμετρος αυτή επιτρέπει στο νευρωνικό δίκτυο να προσαρμόζει καλύτερα σύνθετα δεδομένα εισόδου αφού μετατοπίζει την συνάρτηση ενεργοποίησης. Το αποτέλεσμα είναι η

συνάρτηση ενεργοποίησης να μην περνά από την αρχή των αξόνων όπως φαίνεται και στην παρακάτω εικόνα



Εικόνα 24. Συνάρτηση ενεργοποίησης

Η έξοδος για ένα νευρωνικό δίκτυο με είσοδο $x_1, x_2, \dots, x_n$, βάρη $w_1, w_2, \dots, w_n$ και bias b θα είναι:

$$y = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n + b)$$

όπου f συνάρτηση ενεργοποίησης.

Ένα τεχνητό νευρωνικό δίκτυο υπολογίζει συναρτήσεις με τα βάρη να αποτελούν τις ενδιάμεσες παραμέτρους μεταξύ εισόδου και εξόδου. Η διαδικασία μάθησης περιλαμβάνει την εναλλαγή των βαρών μεταξύ των νευρώνων. Κατά την διαδικασία της εκπαίδευσης το σύστημα παρέχει ανατροφοδότηση σχετικά με την ορθότητα των προβλεπόμενων εξόδων. Έτσι, ο στόχος με την εναλλαγή βαρών είναι να τροποποιηθεί η συνάρτηση ώστε να παρέχει πιο σωστές προβλέψεις.

Convolutional Neural Networks (CNNs)

Τα συνελικτικά νευρωνικά δίκτυα είναι ένα είδος νευρωνικού δικτύου για την επεξεργασία δεδομένων που έχουν γνωστή τοπολογία με την μορφή πλέγματος. Για παράδειγμα, δεδομένα χρονοσειρών μπορούν να θεωρηθούν ως ένα πλέγμα 1D που λαμβάνει δεδομένα σε τακτά χρονικά διαστήματα. Σε ένα άλλο παράδειγμα , δεδομένα εικόνας μπορούν να θεωρηθούν ως ένα πλέγμα 2D pixel.

Οι περισσότερες εφαρμογές των συνελικτικών νευρωνικών δικτύων επικεντρώνονται σε δεδομένα εικόνας, αν και μπορεί κανείς να χρησιμοποιήσει

επίσης αυτά τα δίκτυα για όλους τους τύπους χρονικών, χωρικών και χωροχρονικών δεδομένων.

Το σημαντικότερο χαρακτηριστικό των συνελικτικών νευρωνικών δικτύων είναι μια λειτουργία, που αναφέρεται ως συνέλιξη. Για να έχουμε συνελικτικό νευρωνικό δίκτυο θα πρέπει να γίνεται χρήση της συνέλιξης σε τουλάχιστον ένα επίπεδο, αν και τα περισσότερα συνελικτικά νευρωνικά δίκτυα χρησιμοποιούν αυτή τη λειτουργία σε περισσότερα από ένα επίπεδα.

Η έμπνευση για τα συνελικτικά νευρωνικά δίκτυα προήλθε από τα πειράματα που πραγματοποίησαν οι Hubel και Wiesel στον οπτικό φλοιό των γατών και τα οποία δημοσιεύτηκαν το 1959. Οι Hubel και Wiesel αποκάλυψαν ότι ο οπτικός φλοιός περιέχει νευρώνες με μικρά δεκτικά πεδία που ανταποκρίνονται επιλεκτικά σε συγκεκριμένα χαρακτηριστικά του οπτικού πεδίου, όπως ακμές, σχήματα και προσανατολισμούς.

Οι νευρώνες στον οπτικό φλοιό βρέθηκαν να είναι οργανωμένοι με ιεραρχικό τρόπο, με διαφορετικά στρώματα να ανταποκρίνονται σε όλο και πιο πολύπλοκα χαρακτηριστικά. Για παράδειγμα, ορισμένοι νευρώνες ήταν ευαίσθητοι σε βασικά χαρακτηριστικά όπως οι ακμές, ενώ άλλοι ανταποκρίνονταν σε πιο αφηρημένα και πολύπλοκα οπτικά μοτίβα. Αυτή η πολυεπίπεδη αρχιτεκτονική ανέδειξε το γεγονός πως το οπτικό σύστημα επεξεργάζεται τις οπτικές πληροφορίες ιεραρχικά, εξάγοντας πρωτόγονα χαρακτηριστικά στα πρώιμα στρώματα και συνδυάζοντάς τα για να αναγνωρίσει πιο περίπλοκα μοτίβα σε μεταγενέστερα επίπεδα.

Τα συνελικτικά νευρωνικά δίκτυα, εμπνευσμένα από αυτή τη βιολογική γνώση, σχεδιάστηκαν με πολυεπίπεδες αρχιτεκτονικές που μιμούνται την ιεραρχική οργάνωση του οπτικού φλοιού. Στα CNN, όπως θα δούμε και παρακάτω, τα πρώτα επίπεδα καταγράφουν απλά χαρακτηριστικά, όπως άκρες και υφές, ενώ τα βαθύτερα στρώματα συνδυάζουν προοδευτικά αυτά τα χαρακτηριστικά για να αναγνωρίσουν πιο περίπλοκα και αφηρημένα μοτίβα.

Παρακάτω θα αναλύσουμε τα τέσσερα βήματα λειτουργίας των CNN, δηλαδή:

α) Το επίπεδο Συνέλιξης (Convolution operation) και εισαγωγή μη-γραμμικότητας (ReLU layer)

β) Το επίπεδο Συγκέντρωσης (Pooling)

γ) και τέλος το Πλήρες Συνδεδεμένο (Full Connection) Επίπεδο

Επίπεδο Συνέλιξης

Το επίπεδο συνέλιξης αποτελεί ένα πολύ σημαντικό επίπεδο στα νευρωνικά δίκτυα. Είναι το επίπεδο που είναι υπεύθυνο για την ανίχνευση χαρακτηριστικών σε εισόδους δεδομένων όπως οι εικόνες. Η συνέλιξη, μπορεί να εφαρμοστεί

αποτελεσματικά και σε άλλου τύπου δεδομένα όπως δεδομένα ήχου αλλά και χρονοσειρών. Συγκεκριμένα, ένα διάνυσμα λαμβάνεται ως είσοδος και πολλαπλασιάζεται με έναν πίνακα ώστε να παραχθεί μία έξοδος.

Παρακάτω θα δούμε ένα παράδειγμα συνέλιξης. Μία εικόνα μπορεί να θεωρηθεί ένα σύνολο από πίξελ με κάθε πίξελ να λαμβάνει μία αριθμητική τιμή. Έστω ότι έχουμε έναν πίνακα 5Χ5 που αναπαριστά μία εικόνα ως είσοδο. Επίσης έχουμε και έναν δεύτερο πίνακα 3Χ3 που αντιπροσωπεύει το φίλτρο μας. Σκοπός μας είναι να δημιουργήσουμε έναν νέο πίνακα μικρότερης διάστασης από την αρχική εικόνα. Το φίλτρο αρχικά ‘’τοποθετείται’’ στην πάνω αριστερή γωνία του πίνακα εισόδου και συνεχίζουμε μέχρι το φίλτρο να καλύψει όλο τον πίνακα. Σε κάθε θέση φίλτρου πραγματοποιούμε την πράξη της συνέλιξης. Στην συνέχεια, οι πολλαπλασιασμένες τιμές προστίθενται για να δημιουργήσουν έναν νέο αριθμό που θα αντιστοιχεί και σε ένα στοιχείο του νέου πίνακα. Έπειτα, μετακινούμε το φίλτρο κατά ένα βήμα προς τα δεξιά μέχρι να φτάσουμε στο τέλος της εικόνας.

Έστω ο αρχικός πίνακας εισόδου και το φίλτρο

3	4	5	6	1
2	0	0	1	3
4	6	0	5	4
5	4	4	3	1
1	2	1	0	1

1	0	1
0	2	0
0	1	0

Ακολουθώντας την διαδικασία που περιγράψαμε παραπάνω θα έχουμε αρχικά την πρώτη πράξη συνέλιξης του πίνακα εισόδου με το φίλτρο.

3	4	5	6	1
2	0	0	1	3
4	6	0	5	4
5	4	4	3	1
1	2	1	0	1

1	0	1
0	2	0
0	1	0

Η πράξη αυτή θα ‘’γεμίσει’’ την πρώτη θέση του νέου πίνακα που θα δημιουργηθεί κάνοντας την πράξη (3*1)+(4*0)+(5*1)+(2*0)+(0*2)+(0*0)+(4+0)+(6*1)+(0*0)

14		

Στο δεύτερο βήμα το φίλτρο μετακινείται μία θέση δεξιά ως εξής :

3	4	5	6	1
2	0	0	1	3
4	6	0	5	4
5	4	4	3	1
1	2	1	0	1

1	0	1
0	2	0
0	1	0

και μετά τις απαραίτητες πράξεις ο τελικός πίνακας θα διαμορφωθεί ως εξής:

14	10	

Ολοκληρώνοντας την διαδικασία ο πίνακας θα έχει τις ακόλουθες τιμές

14	10	13
14	5	16
14	20	10

Ο πίνακας που δημιουργείται ονομάζεται ‘’Ενεργοποίηση’’ ή ‘’Χάρτης Χαρακτηριστικών’’. Κατά την διάρκεια εκπαίδευσης το CNN ανιχνεύει το φίλτρο εκείνο το οποίο καταφέρνει να μας δώσει καλύτερα αποτελέσματα. Αυτό σημαίνει ότι θα πρέπει να δοκιμάσουμε πολλά φίλτρα ώστε να καταλήξουμε σε αυτό που μας κάνει καλύτερη ανίχνευση χαρακτηριστικών.

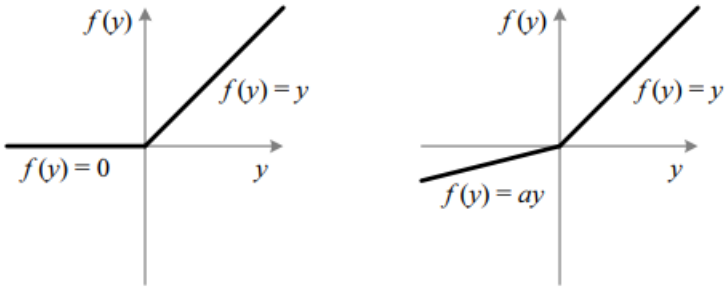
Πολύ συχνά, κατά την διαδικασία της συνέλιξης κάνουμε χρήση του padding (ή zero-padding), μιας τεχνικής που μας βοηθά να διατηρήσουμε την διαστάση των δεδομένων εισόδου. Η συνέλιξη μπορεί να οδηγήσει σε μείωση διαστάσεων, όταν το φίλτρο που εφαρμόζουμε είναι μικρότερης διάστασης από του πίνακα εισόδου. Το padding μας λύνει αυτό το πρόβλημα προσθέτοντας επιπλέον στοιχεία με τιμή μηδέν στον πίνακα εισόδου. Με αυτό το τρόπο εξασφαλίζουμε πως οι έξοδοι της συνέλιξης θα έχουν ίδιες διαστάσεις με εκείνες των πινάκων εισόδου. Επιπλέον εξασφαλίζουμε την διατήρηση της πληροφορίας στα άκρα της εικόνας εισόδου, βελτιώνοντας την απόδοσή του.

Έχοντας πάρει τον χάρτη χαρακτηριστικών θα πρέπει να του εφαρμόσουμε μία συνάρτηση ενεργοποίησης μέσα από την οποία ενεργοποιείται ένας νευρώνας. Βασικός σκοπός των συναρτήσεων ενεργοποίησης είναι η εισαγωγή μη γραμμικότητας στο δίκτυό μας. Αυτό γίνεται διότι τα δεδομένα που καλείται να επεξεργαστεί ένα συνελικτικό νευρωνικό δίκτυο , όπως τα δεδομένα εικόνας ή ήχου, είναι συνήθως μη γραμμικά. Τα προηγούμενα χρόνια γινόταν χρήση συναρτήσεων όπως η σιγμοειδής και η tanh (Hyberbolic Tangent) . Πλέον , μία πρόσφατη εξέλιξη στα νευρωνικά δίκτυα, έφερε στο προσκήνιο την συνάρτηση ενεργοποίησης ReLU (Rectified Linear Unit). Η ReLU επιστρέφει την τιμή εισόδου της εάν αυτή είναι θετική ή μηδέν. Στην περίπτωση που η είσοδος έχει αρνητική τιμή τότε η συνάρτηση επιστρέφει και πάλι μηδέν. Υπάρχουν και παραλλαγές αυτής της συνάρτησης.

Παραλλαγή της ReLU αποτελεί η PReLU(Parametric ReLU). Μαθηματικά, η συνάρτηση αυτή φαίνεται παρακάτω:

$$f(y_i) = \begin{cases} y_i, & if\ y_i > 0 \\ a_i y_i, & if\ y_i \leq 0 \end{cases}$$

Όπου yi η είσοδος της συνάρτησης και αi μία παράμετρος. Η Parametric ReLU επιτρέπει την ανάθεση ενός παραμετρικού βάρους α στην κλίση για αρνητικές τιμές της εισόδου. Αυτό σημαίνει ότι το α είναι μια παράμετρος που μπορεί να μάθει το μοντέλο κατά την εκπαίδευση, για να προσαρμοστεί καλύτερα στα δεδομένα. Έτσι, αντί να είναι μια σταθερή τιμή, το α μπορεί να προσαρμοστεί για κάθε είσοδο ξεχωριστά. Στην περίπτωση όμως που το αi είναι σταθερό και συνήθως κοντά στην τιμή 0,01 τότε έχουμε την παραλλαγή LReLU (Leaky ReLU). Η LReLU επιτρέπει μια μικρή, αρνητική κλίση για αρνητικές τιμές της εισόδου. Με αυτό τον τρόπο αποφεύγουμε δυσκολίες που μπορεί να μας φέρει μία είσοδος με μεγάλη αρνητική τιμή. Στο παρακάτω σχήμα βλέπουμε την διαφορά μεταξύ ReLU και PReLU.



Εικόνα 25. Σχήμα απεικόνισης διαφορών ReLU και PReLU

Επίπεδο Συγκέντρωσης (Pooling)

Το επίπεδο συγκέντρωσης έπεται της συνέλιξης και χρησιμοποιείται στα συνελικτικά νευρωνικά δίκτυα για την μείωση των διαστάσεων του χάρτη χαρακτηριστικών. Η μείωση αυτή γίνεται χρησιμοποιώντας μία συνάρτηση η οποία μπορεί να είναι η μέση τιμή ή η μέγιστη τιμή μίας χωρικής γειτονιάς που ορίζουμε από τον χάρτη χαρακτηριστικών. Με αυτό τον τρόπο καταφέρνουμε να μειώσουμε τις διαστάσεις του χάρτη χαρακτηριστικών . Έτσι μειώνεται και η πολυπλοκότητα βοηθώντας το δίκτυο να εστιάσει στην ανίχνευση σημαντικών παραμέτρων και χαρακτηριστικών.

Στην αναγνώριση αντικειμένων, και συγκεκριμένα στην αυτόνομη οδήγηση, η χρησιμότητα του polling είναι πολύ μεγάλη. Ας υποθέσουμε πως λαμβάνουμε δεδομένα από κάμερες αυτοκινήτων. Για να μάθει το δίκτυό μας να αναγνωρίζει τις πινακίδες του STOP θα πρέπει να εκπαιδευτεί σε εικόνες STOP από διαφορετικές οπτικές γωνίες, με διαφορετικό μέγεθος και φωτισμό. Με την χρήση του pooling καταφέρνουμε να παρακάμψουμε αυτές τις δυσκολίες και να εστιάσουμε στα πιο σημαντικά χαρακτηριστικά μίας εικόνας.

Παρακάτω θα δούμε ένα παράδειγμα εφαρμογής pooling σε έναν χάρτη χαρακτηριστικών. Για το παράδειγμα αυτό θα κάνουμε χρήση της συνάρτησης της μέγιστης τιμής.

Έστω ότι έχουμε τον ακόλουθο χάρτη χαρακτηριστικών:

0	1	1	2	0
0	2	2	2	0
1	4	4	5	0
2	3	3	2	1
0	1	4	2	0

Έστω ότι έχουμε επιλέξει σαν χώρο εφαρμογής της συνάρτησης ένα πλέγμα 2X2.

Η πρώτη εφαρμογή θα γίνει στον μαρκαρισμένο χώρο του χάρτη χαρακτηριστικών που φαίνεται παρακάτω.

0	1	1	2	0
0	2	2	2	0
1	4	4	5	0
2	3	3	2	1
0	1	4	2	0

Η πρώτη τιμή που θα εισάγουμε στον νέο πίνακα είναι η μέγιστη των 0,1,0,2.

Έτσι, ο νέος πίνακας θα έχει για την ώρα την παρακάτω μορφή:

2		

Στο επόμενο βήμα (stride) μετακινούμαστε κατά μία θέση δεξιά. Επομένως, η νέα περιοχή εφαρμογής της συνάρτησης είναι η παρακάτω:

0	1	1	2	0
0	2	2	2	0
1	4	4	5	0
2	3	3	2	1
0	1	4	2	0

η οποία θα μας συμπληρώσει την δεύτερη τιμή στον νέο πίνακα με την μέγιστη τιμή των 1,1,2,2.

Ο νέος πίνακας θα έχει τώρα την μορφή:

2	2	

Παρακάτω βλέπουμε με διαφορετικό χρώμα όλες τις περιοχές που θα εξεταστούν καθώς και τον τελικό πίνακα που θα δημιουργηθεί:

0	1	1	2	0
0	2	2	2	0
1	4	4	5	0
2	3	3	2	1
0	1	4	2	0

2	2	0
4	5	1
1	4	0

Πλήρως Συνδεδεμένο Επίπεδο (Full Connected Layer)

Το πλήρως συνδεδεμένο επίπεδο ή perceptron όπως αναφέρεται συχνά, είναι το επίπεδο όπου κάθε νευρώνας στο στρώμα συνδέεται με κάθε νευρώνα του προηγούμενου στρώματος . Εφαρμόζεται μετά από τα επίπεδα συνέλιξης και συγκέντρωσης και είναι υπεύθυνα για την επεξεργασία χαρακτηριστικών υψηλού επιπέδου.

Τα πλήρως συνδεδεμένα επίπεδα είναι ιδιαίτερα αποτελεσματικά στην αναγνώριση προτύπων και σχέσεων σε δεδομένα. Παρ’ όλα αυτά εισάγουν έναν πολύ μεγάλο αριθμό παραμέτρων, οι οποίες μπορεί και να οδηγήσουν σε overfitting εάν το δίκτυο δεν έχει τις σωστές ρυθμίσεις. Τεχνικές όπως η dropout, η batch normalization και η regularization εφαρμόζονται συχνά για την αντιμετώπιση αυτού του προβλήματος.

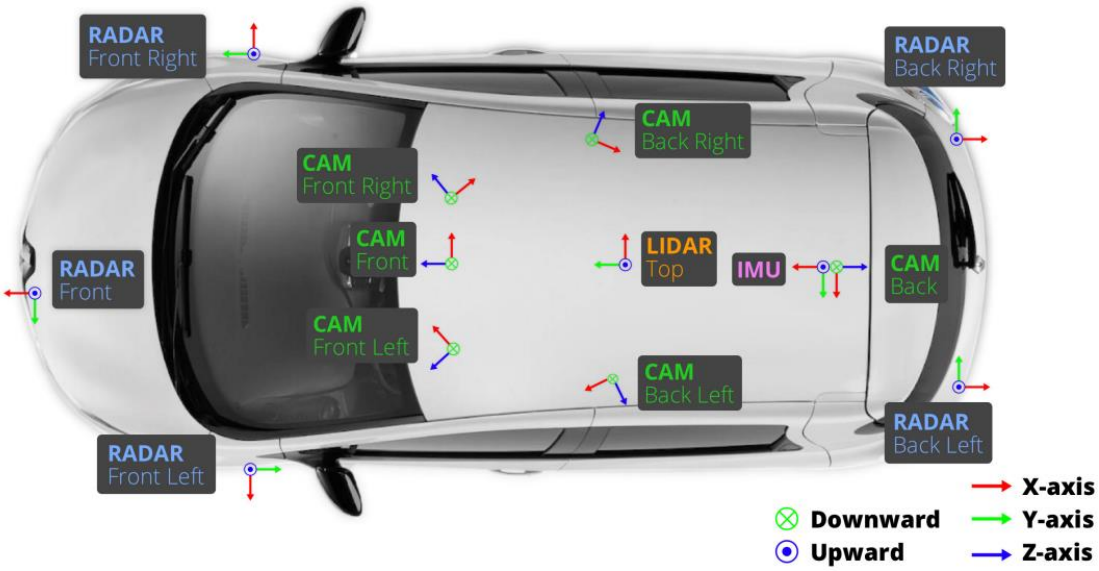
ΚΕΦΑΛΑΙΟ 5

5.1 Αναγνώριση αντικειμένων και Αυτόνομη Οδήγηση

Η όραση υπολογιστών αποτελεί έναν από τους πιο ενεργούς τομείς της βαθιάς μάθησης με πολλές εφαρμογές όπως η αναγνώριση αντικειμένων σε εικόνες. Τα τελευταία χρόνια, ο τομέας του computer vision έχει σημειώσει σημαντικά βήματα προόδου, επιτρέποντας στις μηχανές να κατανοούν και να ερμηνεύουν τον οπτικό κόσμο. Μία από τις βασικές εφαρμογές του computer vision βρίσκεται στην αυτοκινητοβιομηχανία οπού η ανίχνευση αντικειμένων παίζει καθοριστικό ρόλο στη διασφάλιση της ασφάλειας και της αυτονομίας των οχημάτων. Η ανίχνευση αντικειμένων στο αυτοκίνητο περιλαμβάνει την αναγνώριση και τον εντοπισμό διαφόρων αντικειμένων, όπως πεζών, οχημάτων, πινακίδων κυκλοφορίας και εμποδίων στο περιβάλλον του αυτοκινήτου. Αποτελεί τη βάση για τα προηγμένα συστήματα υποστήριξης οδηγού (ADAS) και τις τεχνολογίες αυτόνομης οδήγησης.

Ο τομέας της αυτόνομης οδήγησης αποτελεί μία εφαρμογή του Computer Vision που θα αλλάξει τις μεταφορές στην κοινωνία μας. Σύμφωνα με μία έκθεση της National Highway Traffic Safety Administration (NHTSA) του 2018, το 94% των ατυχημάτων στους δρόμους οφείλεται σε ανθρώπινο λάθος με το υπόλοιπο 6% να μοιράζεται ισόποσα σε οχήματα, περιβάλλον και σε άγνωστους παράγοντες. Καταλαβαίνουμε λοιπόν πως η ένταξη αυτόνομων οχημάτων στους δρόμους θα μείωνε σημαντικά το ποσοστό των ατυχημάτων που μπορεί να προκαλέσουν υλικές ζημιές, τραυματισμούς αλλά και θανάτους.

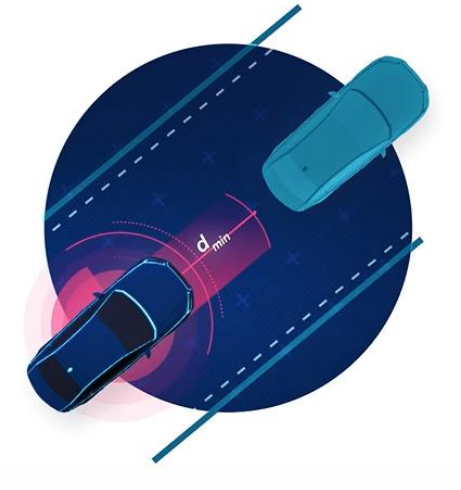
Στην παρακάτω εικόνα βλέπουμε το βασικό setup αισθητήρων που χρειάζεται ένα όχημα για να μπορέσει να υποστηρίξει αυτόνομη οδήγηση. Όπως βλέπουμε περιλαμβάνει κάμερες προς όλες τις κατευθύνσεις. Οι κάμερες χρησιμοποιούνται για την ανίχνευση οχημάτων , πεζών , πινακίδων οδικής κυκλοφορίας και άλλων αντικειμένων που βρίσκονται στο περιβάλλον του οχήματος . Οι κάμερες αυτές τροφοδοτούν το σύστημα αυτόνομης οδήγησης με δεδομένα εικόνας. Στη συνέχεια, μπορούμε να δούμε πως στην κορυφή του οχήματος τοποθετείται ένας αισθητήρας LIDAR. Οι αισθητήρες LIDAR χρησιμοποιούν το φως για να μετρήσουν αποστάσεις των αντικειμένων σε σχέση με το περιβάλλον. Ο αισθητήρας αυτός βοηθά στην κατανόηση της τοπολογίας του περιβάλλοντος, στην ανίχνευση αντικειμένων και στην πρόβλεψη της κίνησης άλλων αντικειμένων. Ακόμα βλέπουμε πως έχουν τοποθετηθεί περιμετρικά του οχήματος αισθητήρες ραντάρ οι οποίοι μετρούν την απόσταση, την ταχύτητα και την κατεύθυνση των αντικειμένων γύρω από το όχημα. Τέλος, βλέπουμε την ύπαρξη ενός συστήματος IMU(Inertial Measurement Unit-Ενότητα Μέτρησης Αδράνειας) το οποίο αποτελείται από διάφορους αισθητήρες που μετρούν μεγέθη όπως η ταχύτητα και η επιτάχυνση.



Εικόνα 26. Εικόνα απεικόνισης βασικού εξοπλισμού οχήματος για αυτόνομη οδήγηση

Για να μπορέσουν όμως οι άνθρωποι να εμπιστευτούν τα συστήματα αυτόνομης οδήγησης για την μεταφορά τους θα πρέπει να νιώθουν και την απαραίτητη ασφάλεια μέσα στα οχήματα. Επομένως, οι βιομηχανίες που είναι υπεύθυνες για την δημιουργία αυτών των συστημάτων, οι κυβερνήσεις αλλά και το κοινό θα πρέπει να ακολουθούν κάποια πρότυπα ασφάλειας. Ένα τέτοιο πρότυπο είναι και το RSS Safety Rules (Roadway Safety Solutions), το οποίο αποτελείται από πέντε κανόνες:

1. Τα συστήματα αυτόνομης οδήγησης θα πρέπει να είναι προγραμματισμένα ώστε να διατηρούν αποστάσεις ασφαλείας από τα προπορευόμενα οχήματα. Επομένως, θα πρέπει να υπάρχει μία ελάχιστη απόσταση μεταξύ των οχημάτων που δεν θα μπορεί να παραβιαστεί.

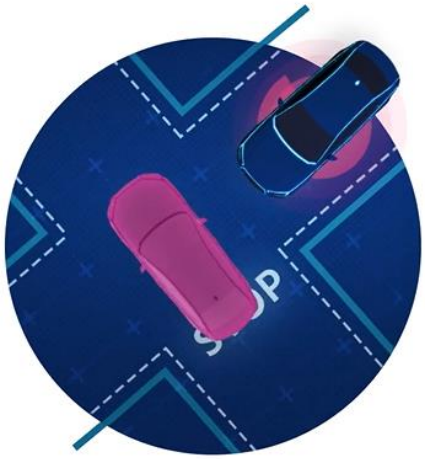


Εικόνα 27. Πρώτος κανόνας ασφάλειας για αυτόνομη οδήγηση

2. Δεύτερος κανόνας είναι η ασφαλής αλλαγή λωρίδας. Ένα σύστημα αυτόνομης οδήγησης θα πρέπει να μπορεί να διασφαλίσει την διατήρηση της ασφαλούς πλευρικής απόστασης από τα διπλανά οχήματα κατά την αλλαγή λωρίδας. Επιπλέον, όταν ένα αμάξι 'καταλάβει' πως ένα άλλο όχημα προσπαθεί να μπει στην λωρίδα του με απότομο κόψιμο προς αυτό τότε θα πρέπει να φρενάρει για να γλυτώσει την σύγκρουση.

Εικόνα 28. Δεύτερος κανόνας ασφάλειας για αυτόνομη οδήγηση

3. Ένας τρίτος και πολύ βασικός κανόνας είναι η σωστή λήψη απόφασης όταν ένας άλλος οδηγός παραβιάζει την προτεραιότητα του οχήματος αυτόνομης οδήγησης. Για παράδειγμα, υπάρχουν περιπτώσεις που οδηγοί παραβιάζουν μία σήμανση STOP. Σε αυτή την περίπτωση, θα έχουμε ενστικτωδώς απότομο φρενάρισμα του οδηγού που θα έπρεπε θεωρητικά να έχει προτεραιότητα αφήνοντας το όχημα που παρανόμησε να περάσει πρώτο.



Εικόνα 29. Τρίτος κανόνας ασφάλειας για αυτόνομη οδήγηση

4. Τέταρτος κανόνας ασφάλειας είναι η προσαρμογή της οδήγησης σε περιβάλλοντα με χαμηλή ορατότητα. Τα συστήματα αυτόνομης οδήγησης θα πρέπει να λαμβάνουν υπόψιν τους το περιβάλλον, τις συνθήκες και την επικινδυνότητα του σημείου οδήγησης και να προσαρμόζουν ανάλογα τις κινήσεις τους. Για παράδειγμα, όταν οδηγούμε δίπλα από ένα μεγάλο όχημα, θα πρέπει να είμαστε πολύ προσεκτικοί καθώς χάνουμε μέρος της ορατότητας του δρόμου και μπορεί να μην εντοπίσουμε έγκαιρα έναν πεζό που θέλει να διασχίσει τον δρόμο.

Εικόνα 30. Τέταρτος κανόνας ασφάλειας για αυτόνομη οδήγηση

5. Τέλος, αναφέρεται ένας πολύ σημαντικός κανόνας ασφάλειας που αφορά το σενάριο στο οποίο ένας ξαφνικός κίνδυνος εμφανίζεται στον δρόμο. Εάν για παράδειγμα, ένα αντικείμενο εμφανιστεί ξαφνικά στο οπτικό πεδίο του συστήματος αυτόνομης οδήγησης τότε το σύστημα θα πρέπει να είναι σε θέση να δει τις εναλλακτικές του (όπως για παράδειγμα το ΄πάτημα΄ σε άλλη λωρίδα), και να ακολουθήσει μία από αυτές χωρίς όμως να προκαλέσει άλλη σύγκρουση.

Εικόνα 31. Πέμπτος κανόνας ασφάλειας για αυτόνομη οδήγηση

Αντιλαμβανόμαστε λοιπόν πόσο σημαντικός είναι ο σχεδιασμός αλγορίθμων ανίχνευσης αντικειμένων με μεγάλα ποσοστά επιτυχίας ανίχνευσης. Οι αλγόριθμοι αυτοί θα πρέπει να

παραμετροποιηθούν και να εκπαιδευτούν κατάλληλα ώστε να ελαχιστοποιούν τις λανθασμένες προβλέψεις.

5.2 Αλγόριθμοι για αναγνώριση αντικειμένων

Οι αλγόριθμοι αναγνώρισης αντικειμένων αποτελούν έναν κρίσιμο τομέα στην υπολογιστική όραση και την τεχνητή νοημοσύνη. Ο βασικός στόχος αυτών των αλγορίθμων είναι να εντοπίσουν και να ταξινομήσουν αντικείμενα σε εικόνες ή βίντεο, καθιστώντας δυνατή την αυτόματη ανίχνευση αντικειμένων και την εξαγωγή σημαντικών πληροφοριών από τα περιεχόμενα τους.

Υπάρχουν δύο βασικές προϋποθέσεις που θα πρέπει να πληρούνται κατά την κατασκευή αλγορίθμων αναγνώρισης αντικειμένων:

-πρώτον, θα πρέπει να διατηρείται ένα υψηλό επίπεδο ακρίβειας στην αναγνώριση αντικειμένων καθώς λάθος ανιχνεύσεις θα μπορούσαν να οδηγήσουν σε επιζήμιες επιπτώσεις.
-δεύτερον, θα πρέπει ο αλγόριθμος που θα σχεδιάσουμε να είναι ικανός να αναγνωρίζει αντικείμενα με καλό ρυθμό ώστε να πετύχουμε την ανίχνευση σε πραγματικό χρόνο.

Ένας από τους πιο δημοφιλείς αλγορίθμους αναγνώρισης αντικειμένων είναι το YOLO (You Only Look Once). Ο YOLO είναι γνωστός για την ταχύτητα και την αποτελεσματικότητά του, καθώς μπορεί να εντοπίσει αντικείμενα σε πραγματικό χρόνο σε εικόνες και βίντεο.

Άλλος ένας σημαντικός αλγόριθμος είναι το RCNN (Regions with Convolutional Neural Networks). Ο RCNN χρησιμοποιεί ένα σύνολο προεπεξεργασμένων περιοχών ενδιαφέροντος και εκπαιδευμένων CNNs για την αναγνώριση αντικειμένων, επιτρέποντας την αναγνώριση αντικειμένων με μεγαλύτερη ακρίβεια.

Παρακάτω θα αναλύσουμε διεξοδικά τους αλγορίθμους αυτούς και θα αναφερθούμε στις βασικές εκδόσεις τους αλλά και σε επόμενες εκδόσεις που τους βελτίωσαν.

5.2.1 YOLO

Ο YOLO (You Only Look Once) είναι ένας από τους βασικότερους αλγορίθμους για ανίχνευση αντικειμένων σε εικόνες ή βίντεο. Αυτό που κάνει τον YOLO διαφορετικό σε σχέση με άλλες μεθόδους ανίχνευσης αντικειμένων είναι ο τρόπος που επεξεργάζεται την εικόνα. Πιο συγκεκριμένα, η επεξεργασία της εικόνας γίνεται μία φορά στο σύνολό της , σε αντίθεση με άλλες μεθόδους που διαιρούν την εικόνα σε πολλά τμήματα εξετάζοντας το κάθε ένα ξεχωριστά.

Ο αλγόριθμος ακολουθεί τα παρακάτω βασικά βήματα:

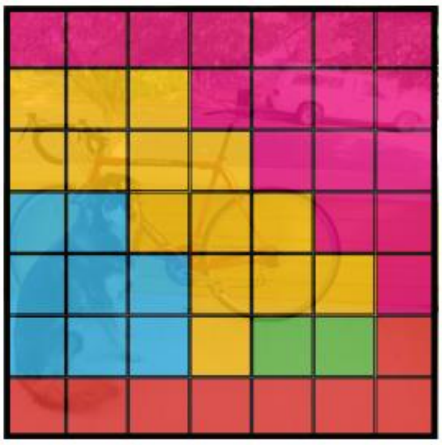
1. Αρχικά, εισάγουμε την εικόνα που μας ενδιαφέρει σε ένα συνελικτικό νευρωνικό δίκτυο (CNN)

2. Έπειτα, διαιρούμε την εικόνα σε έναν ορισμένο αριθμό κελιών δημιουργώντας ένα πλέγμα

Εικόνα 32. Διαίρεση εικόνας σε κελιά

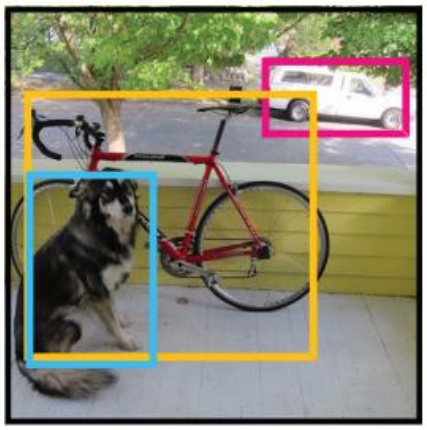
3. Στην συνέχεια, για κάθε κελί του πλέγματος το δίκτυο προβλέπει τα bounding boxes και τις αντίστοιχες πιθανότητες να ανήκουν σε κάποια κλάση

Εικόνα 33. Πρόβλεψη bounding boxes



Εικόνα 34. Πιθανότητες πιθανών κλάσεων

4. Τέλος, γίνεται έλεγχος για την επιλογή του bounding box με την μεγαλύτερη πιθανότητα σε περίπτωση που υπάρχουν επικαλυπτόμενα bounding boxes σε κάποια περιοχή.



Εικόνα 35. Πρόβλεψη κλάσεων

Ένα από τα μεγάλα πλεονεκτήματα του αλγορίθμου είναι η απλότητά του. Ένα single convolutional network προβλέπει ταυτόχρονα πολλαπλά πλαίσια οριοθέτησης (bounding boxes) και τις πιθανότητες κλάσης για αυτά τα πλαίσια. Επιπλέον, ο YOLO είναι εξαιρετικά γρήγορος αλγόριθμος. Η ανίχνευση πλαισίου ανάγεται σε πρόβλημα παλινδρόμησης και έτσι δεν χρειαζόμαστε ένα πολύπλοκο pipeline. Το μόνο που χρειάζεται να κάνουμε είναι να τρέξουμε το νευρωνικό δίκτυο σε μία νέα test image (εικόνα δοκιμής) για πρόβλεψη ανιχνεύσεων. Ο αλγόριθμος YOLO μπορεί να επεξεργαστεί έως και 45 καρέ ανά δευτερόλεπτο αλλά και να επεξεργαστεί ροή βίντεο σε πραγματικό χρόνο με λιγότερο από 25 χιλιοστά του δευτερολέπτου σε καθυστερήσεις.

Ακόμα, ο αλγόριθμος YOLO επιτυγχάνει μεγαλύτερη από την διπλάσια μέση ακρίβεια άλλων αντίστοιχων συστημάτων επεξεργασίας σε πραγματικό χρόνο. Επιπρόσθετα, ο YOLO καταφέρνει να κάνει λιγότερο από το μισό αριθμό σφαλμάτων φόντου σε σύγκριση με τον Fast R-CNN. Σφάλματα φόντου μπορούμε να έχουμε για

παράδειγμα όταν το φόντο μίας εικόνας είναι σύνθετο και περίπλοκο οδηγώντας τους αλγορίθμους σε λανθασμένη ανίχνευση αντικειμένων.

Τέλος, ο YOLO έχει την ικανότητα να γενικεύει τις αναπαραστάσεις των αντικειμένων. Εκπαιδεύοντας τον αλγόριθμο με φυσικές εικόνες αντικειμένων και εφαρμόζοντάς τον έπειτα σε εικόνες έργων τέχνης ο YOLO καταφέρνει να ανιχνεύσει τα αντικείμενα πολύ καλύτερα σε σχέση με άλλες μεθόδους ανίχνευσης.

Εικόνα 36. Εντοπισμός αντικειμένων με χρήση YOLO σε έργα τέχνης

Αξίζει να αναφέρουμε και ένα βασικό μειονέκτημα του αλγορίθμου που είναι η ανίχνευση μικρών αντικειμένων σε μία εικόνα. Ακόμα, ο YOLO επιβάλλει ισχυρούς χωρικούς περιορισμούς στην πρόβλεψη των bounding boxes αφού κάθε κελί πλέγματος προβλέπει μόνο δύο bounding boxes και μπορεί να έχει μόνο μία κλάση. Το γεγονός αυτό καθιστά δύσκολη την αναγνώριση αντικειμένων που βρίσκονται σε κοντινή απόσταση. Επιπλέον, αναλύσεις σφαλμάτων έχουν δείξει πως ο YOLO παράγει πολλά περισσότερα σφάλματα εντοπισμού σε σχέση με τον Fast R-CNN.

YOLOv2 – YOLO9000
Ο αλγόριθμος YOLO έχει πολλές εκδόσεις ανά τα χρόνια με την καθεμία να βελτιώνει την απόδοση στην αναγνώριση αντικειμένων. Η έκδοση YOLOv2 είναι αρκετά πιο γρήγορη σε σχέση με άλλα συστήματα ανίχνευσης καθώς μπορεί με ευκολία να εκτελεστεί σε μεγάλη ποικιλία μεγεθών εικόνας.

Μία έκδοση που έφερε σημαντικές βελτιώσεις είναι επίσης η YOLO9000. Αποτελεί ένα real-time framework που μπορεί να ανιχνεύσει πάνω από 9000 κατηγορίες αντικειμένων. Ο κύριος στόχος της είναι να ανιχνεύει αντικείμενα από ένα μεγάλο σύνολο κλάσεων σε αντίθεση με τις προηγούμενες εκδόσεις που ήταν αρκετά περιορισμένες σε αυτό το κομμάτι. Σε αυτή την έκδοση έγινε χρήση του δικτύου WordTree για τον συνδυασμό δεδομένων από διαφορετικές πηγές για την διαδικασία της εκπαίδευσης.

YOLOv3

Το 2018 δημοσιεύτηκε η έκδοση YOLO v3. Η έκδοση αυτή παρουσιάστηκε από τους Joseph Redmon και Ali Farhadi. Για την πρόβλεψη κλάσεων, σε πολλά συστήματα ανίχνευσης γίνεται χρήση της συνάρτησης softmax στο τέλος του δικτύου για την

κατηγοριοποίηση των προβλεπόμενων κλάσεων αντικειμένων. Στο YOLOv3 παρατηρήθηκε πως η χρήση της συνάρτησης αυτής δεν βοηθάει στην καλύτερη απόδοση του αλγορίθμου. Έτσι, έγινε χρήση ανεξάρτητων λογιστικών ταξινομητών (logistic classifiers) για κάθε κλάση που προσπαθούμε να ανιχνεύσουμε. Κάθε αντικείμενο προς εξέταση αντιμετωπίζεται ως πρόβλημα δυαδικής ταξινόμησης (αν υπάρχει ή όχι στο προς εξέταση πλαίσιο).

Επιπλέον, ο YOLOv3 χρησιμοποιεί τρία επίπεδα ανίχνευσης σε διαφορετικές κλίμακες εικόνας σε τρία διαφορετικά σημεία του δικτύου. Έτσι βελτιώνεται αισθητά η ανίχνευση αντικειμένων σε διαφορετικά μεγέθη. Τέλος, να χρησιμοποιεί προ-εκπαιδευμένα δίκτυα όπως το VGG ή το ResNet, ο YOLOv3 χρησιμοποιεί το Darknet-53 ως backbone, το οποίο είναι εξαιρετικά βαθύ και διατηρεί την αποτελεσματικό .Το Darknet-53 είναι σημαντικό για το YOLOv3 επειδή παρέχει την απαιτούμενη ισχύ και ευελιξία για να αντιμετωπίσει την πολυπλοκότητα της ανίχνευσης αντικειμένων σε διαφορετικές κλίμακες και περιβάλλοντα.

YOLOv4

Το 2020 ήρθε μία νέα έκδοση για τον YOLO, η YOLOv4 από τους Alexey Bochkovskiy, Chien-Yao Wang και Hong-Yuan Mark Liao. Ο αλγόριθμος YOLOv4 χρησιμοποιεί το δίκτυο CSPDarknet53 ως backbone, το οποίο αποτελεί μία βελτιωμένη έκδοση του Darknet53 έχοντας προσθέσει την τεχνική Cross-Stage Partial Connections για καλύτερη εκμάθηση και απόδοση. Επιπλέον, εισάγει νέες μέθοδο για data augmentation, τις n Mosaic, and Self-Adversarial Training (SAT). Η Mosaic Data Augmentation συνδυάζει τέσσερεις τυχαίες εικόνες εκπαίδευσης σε ένα μεγάλο πλαίσιο (mosaic) και στην συνέχεια εκπαιδεύει το μοντέλο να αναγνωρίζει αντικείμενα σε αυτό το μεγάλο πλαίσιο. Αυτή η τεχνική βοηθά στην εκπαίδευση του μοντέλου να μάθει πώς να αντιμετωπίζει διάφορες συνθήκες φωτισμού, σκίασης, αντικειμένων σε διαφορετικές θέσεις, και άλλες μεταβαλλόμενες συνθήκες. Η τεχνική Self-Adversarial Training (SAT) είναι μια μέθοδος για την αύξηση της ανθεκτικότητας του μοντέλου. Κατά τη διάρκεια της εκπαίδευσης, προστίθενται παραλλαγές των εικόνων εκπαίδευσης, που είναι ελαφρώς παραμορφωμένες από την αρχική τους μορφή, με σκοπό να δημιουργήσουν δυσκολότερα παραδείγματα. Αυτό βοηθά το μοντέλο να μάθει να αντιμετωπίζει ακραίες συνθήκες ή παραλλαγές που μπορεί να συναντήσει στο πραγματικό κόσμο.
Τέλος, στον YOLOv4 γίνεται χρήση τεχνικών που βελτιώνουν την ακρίβεια του μοντέλου. Το ένα βασικό είδος τέτοιων τεχνικών είναι οι Bag of Freebies (BoF). Οι BoF αποτελούν τεχνικές βελτιστοποίησης που εφαρμόζονται στο μοντέλο χωρίς να προσθέτουν επιπλέον κόστος κατά την εκπαίδευση ή τον υπολογισμό αποτελεσμάτων. Το άλλο βασικό είδος τεχνικών βελτιστοποίησης είναι οι Bag of Specials (BoS). Οι BoS προσφέρουν βελτιωμένη απόδοση μέσα από διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων. Σε αυτή την περίπτωση όμως έχουμε αυξημένο υπολογιστικό κόστος.

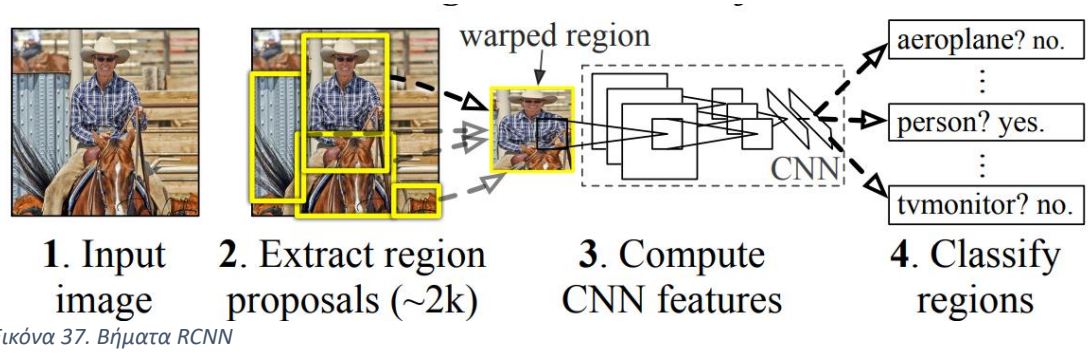
5.2.2 RCNN

Το 2014 προτάθηκε από τους Ross Girshick, Jeff Donahue, Trevor Darrell και Jitendra Malik ένα νέο συνελικτικό νευρωνικό δίκτυο που βασίζεται σε περιοχές γνωστό και ως R-CNN (Region-based Convolutional Neural Network).

Ο R-CNN αποτελείται από τρία βασικά βήματα :

1. Αρχικά, είναι η δημιουργία προτάσεων περιοχών. Οι προτάσεις περιοχών είναι περιοχές σε μία εικόνα που ενδέχεται να εμπεριέχουν αντικείμενα. Αλγόριθμοι που εκτελούν αυτή τη διαδικασία είναι ο Selective Search και ο EdgeBoxes.
2. Έπειτα, έχουμε την εξαγωγή χαρακτηριστικών. Με την βοήθεια ενός προ-εκπαιδευμένου συνελικτικού δικτύου εξάγουμε διανύσματα χαρακτηριστικών σταθερού μήκους από την κάθε περιοχή. Τα διανύσματα αυτά περιγράφουν τα αντικείμενα που έχουμε προς εξέταση. Στην περίπτωση αναγνώρισης εικόνας, ένα διάνυσμα χαρακτηριστικών μπορεί να περιέχει πληροφορίες για το χρώμα, τη φωτεινότητα και το σχήμα ενός αντικειμένου. Για να γίνει αυτό χρησιμοποιούμε περιγραφές εικόνας όπως για παράδειγμα το Histogram of Oriented Gradients (HOG).
3. Τέλος, τα διανύσματα χαρακτηριστικών χρησιμοποιούνται για την εκχώρηση κάθε προτεινόμενης περιοχής σε ‘φόντο’ ή σε ‘αντικείμενο’ με την βοήθεια linear SVMs.

Τα βήματα που αναφέραμε παραπάνω φαίνονται και σχηματικά στην παρακάτω εικόνα:



Εικόνα 37. Βήματα RCNN

Ο R-CNN έχει μερικά σημαντικά πλεονεκτήματα. Για παράδειγμα μπορεί να ανιχνεύσει πολλαπλά αντικείμενα σε μία εικόνα, ακόμα και αν αυτά επικαλύπτονται. Ακόμα, ο R-CNN είναι ευέλικτος και μπορεί να προσαρμοστεί ώστε να ανιχνεύει αντικείμενα σε εικόνες αλλά και σε βίντεο.

Από την άλλη, ο R-CNN είναι αρκετά αργός στην ανίχνευση σε σχέση με άλλες μεθόδους όπως ο YOLO. Ο βασικός λόγος που τον καθιστά αργό είναι η ανάγκη για ανεξάρτητη επεξεργασία κάθε προτεινόμενης περιοχής. Επιπλέον, η εμπλοκή αλγορίθμων για την δημιουργία των περιοχών όπως ο Selective Search αυξάνει και αυτή τον απαιτούμενο χρόνο για την ολοκλήρωση της ανίχνευσης. Έχει μετρηθεί πως η ανίχνευση με δίκτυο VGG16 διαρκεί 47 δευτερόλεπτα ανά εικόνα (σε μία GPU). Το γεγονός αυτό κάνει ιδιαίτερα δύσκολη την εκτέλεση του αλγορίθμου σε πραγματικό χρόνο. Τέλος, ο R-CNN απαιτεί μεγάλο χώρο αποθήκευσης. Συγκεκριμένα, για την εκπαίδευση των SVM χρειάζεται να εξάγονται χαρακτηριστικά από κάθε πρόταση περιοχής και να γραφούν στο δίσκο. Με πολύ βαθιά δίκτυα όπως το VGG16 η διαδικασία αυτή παίρνει πολύ χρόνο (περίπου 2,5GPU-ημέρες) και χρειάζεται εκατοντάδες gigabyte χώρου αποθήκευσης.

Fast RCNN

Ο Fast R-CNN αποτελεί μία βελτιωμένη έκδοση του R-CNN η οποία προτάθηκε από τον Ross Girshick για την έρευνα της Microsoft. Παρέχει αρκετές καινοτομίες για την βελτίωση της ταχύτητας στα στάδια του training και του testing. Συγκεκριμένα, η διαδικασία εκπαίδευσης του δικτύου γίνεται εννέα φορές ταχύτερα και η διαδικασία της δοκιμής διακόσιες δεκατρείς φορές ταχύτερα σε σχέση με τον R-CNN. Παράλληλα προσφέρει μεγαλύτερη ακρίβεια ανίχνευσης αφού τα ποσοστά ακρίβειας στο σύνολο δεδομένων PASCAL VOC 2012 με χρήση της μετρικής mean Average Precision (mAP) είναι στο 66% έναντι 62% του RCNN.

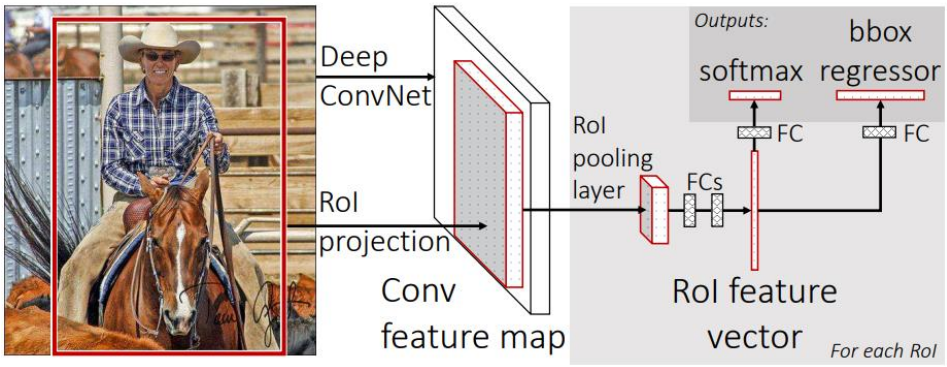
Βασικό σημείο αναφοράς για τον Fast R-CNN είναι η διαδικασία εκπαίδευσης η οποία γίνεται σε ένα μόνο στάδιο, δηλαδή δεν υπάρχουν ξεχωριστά στάδια για την εξαγωγή χαρακτηριστικών και την ανίχνευση αντικειμένων όπως στον RCNN. Αντίθετα, χρησιμοποιείται μια μοναδική συνάρτηση απώλειας για την μέτρηση της απόδοσης που ενσωματώνει πολλαπλά κριτήρια, όπως η πρόβλεψη του τύπου του αντικειμένου, την εκτίμηση του bounding box και την ανίχνευση της παρουσίας του αντικειμένου.

Ακόμα, κατά τη διάρκεια της εκπαίδευσης, όλα τα επίπεδα του νευρωνικού δικτύου μπορούν να ενημερωθούν. Αυτό επιτρέπει στο δίκτυο να εκπαιδευτεί να αναγνωρίζει χαρακτηριστικά σε όλα τα επίπεδα του, από την είσοδο μέχρι την έξοδο, και να προσαρμόζει τα βάρη του ώστε να βελτιώσει την ακρίβειά της ανίχνευσης αντικειμένων.

Τέλος, αξίζει να αναφέρουμε πως ο Fast R-CNN βελτιώνει και το πρόβλημα του αποθηκευτικού χώρου καθώς δεν απαιτείται η αποθήκευση των χαρακτηριστικών στον δίσκο κατά την διάρκεια της εκπαίδευσης.

Παρακάτω θα δούμε την αρχιτεκτονική του Fast R-CNN:

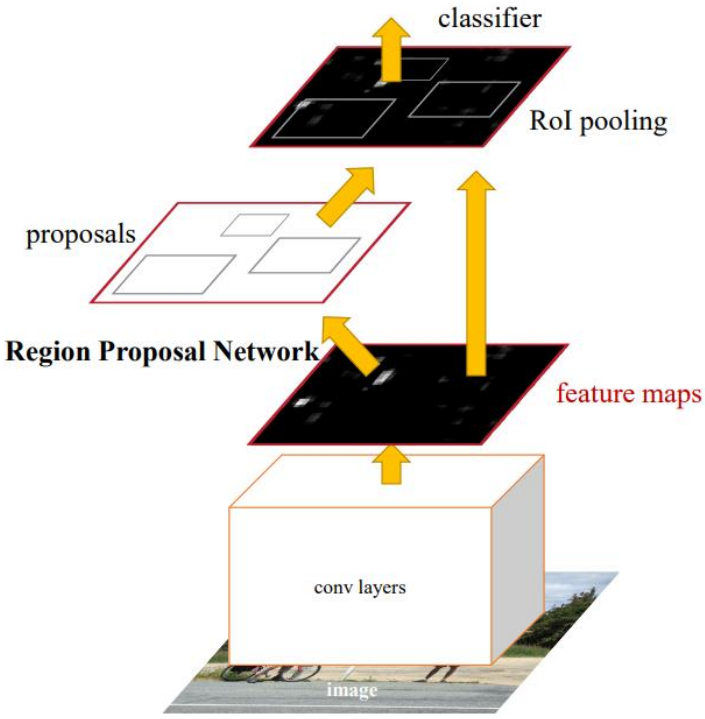
1. Η είσοδος του δικτύου Fast R-CNN αποτελείται από μια ολόκληρη εικόνα και ένα σύνολο από προτάσεις αντικειμένων, που παράγονται διαδικασίες προτάσεων, όπως το Selective Search ή το EdgeBoxes.
2. Η είσοδος επεξεργάζεται από μια σειρά συνελικτικών επιπέδων και max pooling επιπέδων, προκειμένου να παράξει ένα χάρτη χαρακτηριστικών (feature map). Αυτά τα επίπεδα επεξεργάζονται ολόκληρη την εικόνα, ανεξάρτητα από τις προτάσεις αντικειμένων.
3. Για κάθε πρόταση αντικειμένου, ένα επίπεδο "Region of Interest (RoI) pooling" εξάγει ένα σταθερού μήκους διάνυσμα χαρακτηριστικών από τον χάρτη χαρακτηριστικών που προέκυψε από τα προηγούμενα στάδια. Αυτά τα χαρακτηριστικά αντιπροσωπεύουν την περιοχή της εικόνας που είναι ενδιαφέρουσα για το κάθε αντικείμενο.
4. Το διάνυσμα χαρακτηριστικών κάθε πρότασης αντικειμένου περνάει μέσα από μια σειρά από πλήρως συνδεδεμένα επίπεδα (fully connected layers), τα οποία τελικά καταλήγουν σε δύο εξόδους:
   α) Ένα επίπεδο που παράγει πιθανότητες softmax για κάθε κατηγορία αντικειμένου, συμπεριλαμβανομένης μιας "catch-all" κατηγορίας "φόντου" (background).
   β) Ένα επίπεδο που εξάγει τέσσερις πραγματικές τιμές για κάθε κατηγορία αντικειμένου, τις οποίες χρησιμοποιείται για την πρόβλεψη του πλαισίου περιοχής (bounding box) γύρω από το αντικείμενο.



Εικόνα 38. Αρχιτεκτονική Fast RCNN

Faster RCNN

Με την εμφάνιση του Fast RCNN μειώθηκε σημαντικά ο χρόνος εκτέλεσης για τα δίκτυα ανίχνευσης. Έφερε όμως στην επιφάνεια το πρόβλημα στον υπολογισμό των προτάσεων περιοχών. Έτσι, η προσοχή των ερευνητών στράφηκε προς τα εκεί καθώς αποτελούσε το βασικό υπολογιστικό σημείο συμφόρησης στα συστήματα ανίχνευσης. Οι Shaoqing Ren, Kaiming He, Ross Girshick και Jian Sun πρότειναν τον Faster R-CNN το 2016. Πρόκειται για μία επέκταση του Fast R-CNN. Παρακάτω βλέπουμε πως λειτουργεί:



Εικόνα 39. Τρόπος λειτούργιας Faster RCNN

1. Αρχικά, η εικόνα εισόδου διαβιβάζεται μέσω ενός συνελικτικού νευρωνικού δικτύου , όπως το VGG16 ή το ResNet, για την εξαγωγή χαρακτηριστικών.

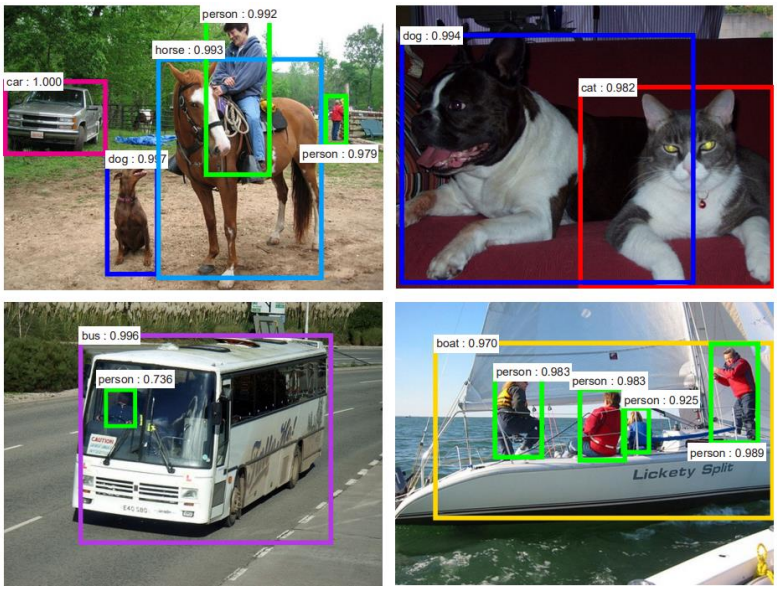
2. Στην συνέχεια, έχουμε την ενσωμάτωση του δικτύου πρότασης περιοχής (Region Proposal Network - RPN) μέσα στο CNN. Το RPN είναι ένα πλήρως συνελικτικό δίκτυο που δημιουργεί προτάσεις περιοχής με διάφορες κλίμακες και αναλογίες διαστάσεων. Βασικό σημείο αναφοράς για το RPN είναι η μέθοδος κουτιών αγκύρωσης (anchor boxes). Κάθε ένα από τα κουτιά αγκύρωσης αντιστοιχεί σε ένα συγκεκριμένο μέγεθος και αναλογία. Έτσι με την χρήση πολλών τέτοιων κουτιών καλύπτουμε διαφορετικές αναλογίες και μεγέθη για ένα αντικείμενο. Καταλήγουμε λοιπόν να έχουμε μία πυραμίδα κουτιών αγκύρωσης. Το RPN λειτουργεί προκαταρκτικά πάνω στο feature map που παράγεται από τα συνελικτικά επίπεδα του δικτύου και παράγει προτάσεις για περιοχές που πιθανόν να περιέχουν αντικείμενα.

3. Αυτές οι προτάσεις χρησιμοποιούνται στη συνέχεια από έναν αλγόριθμο εξαγωγής χαρακτηριστικών (RoI pooling) για την ανάλυση και την εξαγωγή χαρακτηριστικών από κάθε πρόταση περιοχής. Δημιουργούμε με αυτόν τον τρόπο έναν χάρτη χαρακτηριστικών σταθερού μήκους.

4. Έπειτα από την εξαγωγή των χαρακτηριστικών, ακολουθεί ένα νευρωνικό δίκτυο πλήρως συνδεδεμένων επιπέδων (fully connected layers) για την ταξινόμηση των αντικειμένων και την εκτίμηση των πλαισίων περιοχής τους.

Η προσθήκη του RPN έφερε πολλά πλεονεκτήματα σε σχέση με άλλες μεθόδους όπως την Selective Search. Πρώτον, το RPN είναι ένα δίκτυο που μπορεί να εκπαιδευτεί επιτρέποντας έτσι την βελτίωση της απόδοσης των προτάσεων περιοχών. Ακόμα, το RPN είναι πολύ πιο γρήγορο από την επιλεκτική αναζήτηση. Αντί να αναζητά σε όλη την εικόνα για προτάσεις περιοχών που πιθανόν να περιέχουν αντικείμενα, το RPN χρησιμοποιεί ένα νευρωνικό δίκτυο για να προτείνει περιοχές με πιθανά αντικείμενα απευθείας από το χάρτη χαρακτηριστικών, μειώνοντας έτσι τον αριθμό των προτάσεων που πρέπει να εξετάσει το σύστημα ανίχνευσης

Στην επόμενη εικόνα φαίνονται παραδείγματα ανίχνευσης χρησιμοποιώντας RPN στο σύνολο δεδομένων PASCAL VOC 2007.



Εικόνα 40. Παραδείγματα ανίχνευσης αντικειμένων με χρήση RPN

ΚΕΦΑΛΑΙΟ 6

6.1 Σύνολο Δεδομένων

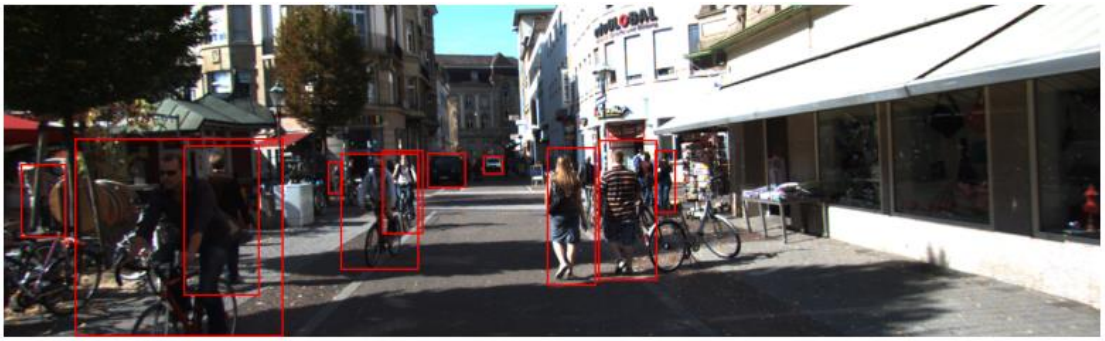
Το σύνολο δεδομένων με το οποίο θα ασχοληθούμε στην παρούσα εργασία είναι το KITTI dataset. Το KITTI dataset είναι ένα από τα πιο γνωστά σύνολα δεδομένων πάνω στην αυτόνομη οδήγηση. Δημιουργήθηκε από το Karlsruhe Institude of Technology και το Toyota Technological Institute του Σικάγο. Στο KITTI παρέχονται σύνολα για έρευνα πάνω σε πολλούς τομείς όπως την αναγνώριση αντικειμένων σε 2D και 3D εικόνες.

Το KITTI dataset αποτελείται από 7481 εικόνες εκπαίδευσης και 7518 εικόνες για testing και συνολικά διαθέτει 80256 αντικείμενα με ετικέτα. Τα αρχεία εικόνων είναι έγχρωμες και τύπου .png. Οι εικόνες συλλέχθηκαν από την πόλη Καρλσρούη της Γερμανίας με όχημα που διέθετε εξοπλισμό σύμφωνα με την παρακάτω εικόνα:

Εικόνα 41. Εξοπλισμός οχήματος συλλογής εικόνων για το KITTI dataset

Οι κάμερες κατέγραφαν 10 καρέ ανά δευτερόλεπτο και η ανάλυση των εικόνων είναι 1224 X 370 . 48

Παρακάτω βλέπουμε κάποιες από τις εικόνες εκπαίδευσης



Όπως βλέπουμε, το σύνολο δεδομένων διαθέτει εικόνες από μικρούς δρόμους, από λεωφόρους αλλά και από πεζόδρομους. Οι συνθήκες φωτισμού ποικίλουν όπως επίσης και οι γωνίες και οι αποστάσεις καταγραφής των αντικειμένων ενδιαφέροντος. Το γεγονός αυτό κάνει το σύνολο δεδομένων κατάλληλο για την εκμάθηση ενός μοντέλου ανίχνευσης αντικειμένων.

Οι κατηγορίες που ανιχνεύει το KITTI dataset είναι οι παρακάτω:

Αριθμός κλάσης	Αντικείμενο ανίχνευσης
0	Car
1	Pedestrian
2	Van
3	Cyclist
4	Truck
5	Misc
6	Tram
7	Person sitting
8	Don’t care

Στην κατηγορία Don’t Care υπάρχουν περιοχές της εικόνας τις οποίες δεν πρέπει να λάβουμε υπόψιν κατά την αξιολόγηση των αποτελεσμάτων του μοντέλου μας. Οι περιοχές αυτές μπορεί να περιλαμβάνουν αντικείμενα που είναι πολύ δύσκολο να ανιχνευτούν με ακρίβεια και πιθανό να μην έκανε καλό στην διαδικασία της εκπαίδευσης. Τα αντικείμενα αυτά μπορεί να έχουν ασαφή όρια, να είναι μερικώς κρυμμένα ή ακόμα και να μην ανήκουν σε κάποια από τις βασικές κατηγορίες.

6.2 Αποτελέσματα

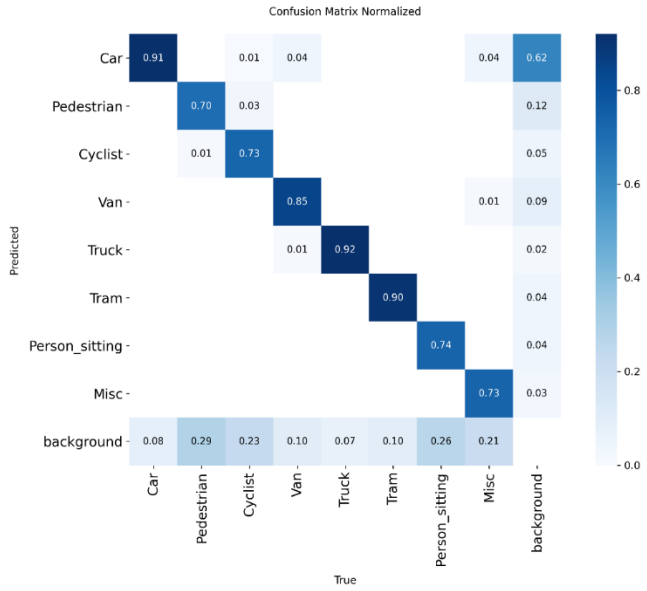
Για το YOLO χρησιμοποιήθηκε το μοντέλο YOLOv8 της βιβλιοθήκης Ultalytics YOLO. Η εκπαίδευση βασίστηκε στην έκδοση YOLOv8n, η οποία αποτελεί την ελαφρύτερη παραλλαγή της αρχιτεκτονικής καθώς δουλεύουμε σε περιβάλλον CPU.

Επιπλέον, κάναμε χρήση προεκπαιδευμένων βαρών από το COCO dataset. Έτσι, το δίκτυο δεν ξεκινάει με τυχαία βάρη καθώς έχει ήδη μάθει βασικά οπτικά χαρακτηριστικά όπως edges, shapes και textures.
Το YOLO είναι one-stage detector δηλαδή προβλέπει ταυτόχρονα bounding boxes και κατηγορίες.

Η εκπαίδευση έτρεξε σε δύο στάδια. Στο πρώτο στάδιο τρέξαμε για 30 εποχές με learning rate 0,01 (default βιβλιοθήκης). Στο δεύτερο στάδιο ολοκληρώσαμε με άλλες 15 εποχές ρίχνοντας σημαντικά το learning rate σε 0,0005 ώστε να παρέμβουμε διακριτικά στην εκπαίδευση του πρώτου σταδίου. Στην ουσία, κατά το δεύτερο στάδιο εφαρμόσαμε τεχνική fine-tuning κατά την οποία επιτρέπουμε την βελτιωτική προσαρμογή του ήδη εκπαιδευμένου μοντέλου, επιτυγχάνοντας καλύτερη εξειδίκευση στην αναγνώριση αντικειμένων.

Παρακάτω, παρουσιάζουμε τα αποτελέσματα της εκπαίδευσης του μοντέλου μας μετά την ολοκλήρωση των 45 εποχών:

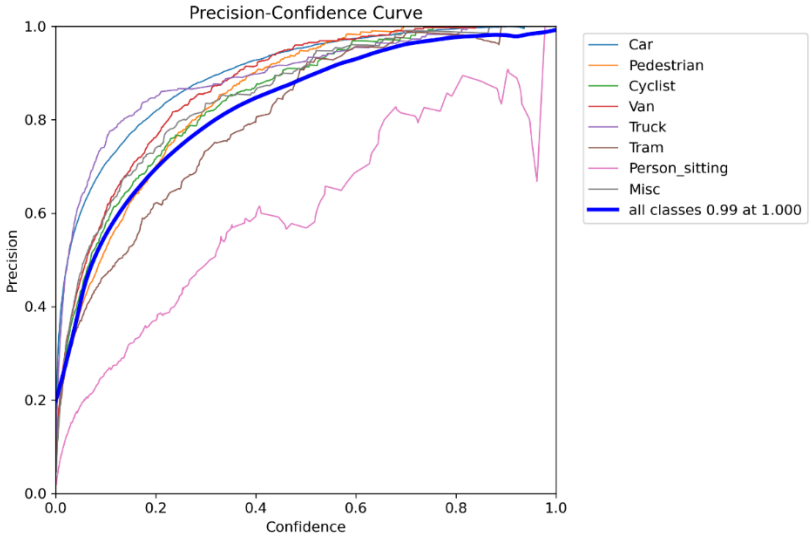
Αρχικά, παραθέτουμε το normalized confusion matrix:



που μας δείχνει ότι στις περισσότερες κατηγορίες τα αντικείμενα κατηγοριοποιούνται σωστά.

Στο παρακάτω γράφημα βλέπουμε την καμπύλη για το Precision το οποίο δίνεται από την σχέση:

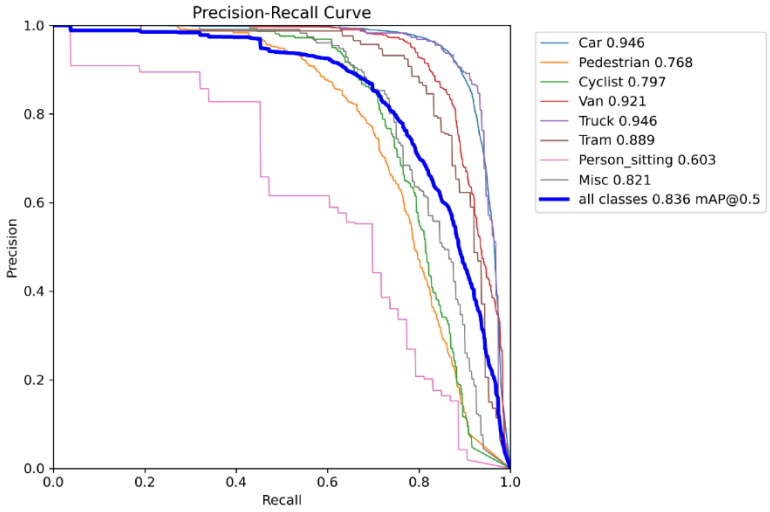
Precision=(TP/(TP+FP))



Όπως βλέπουμε, η καμπύλη είναι ομαλή και σταθερή γεγονός που μας δείχνει καλή εκπαίδευση με αποτελεσματικό fine-tuning. Παρατηρώντας την κάθε κατηγορία ξεχωριστά βλέπουμε πολύ καλά αποτελέσματα σε όλες τις κλάσεις με εξαίρεση την κλάση Person sitting που παρουσιάζει χαμηλότερη σταθερότητα. Αυτό το αποτέλεσμα είναι λογικό καθώς οι παρατηρήσεις για αυτή την κατηγορία είναι λιγότερες λόγω σπανιότητας εμφάνισης στους δρόμους.

Στην συνέχεια, βλέπουμε την καμπύλη για το Recall που δίνεται από την σχέση:

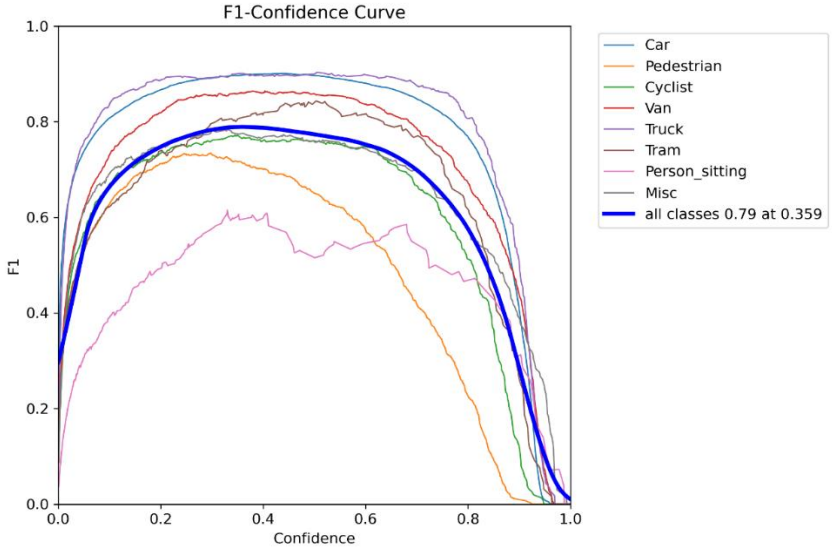
Recall =(TP/(TP+FN))



Όπως βλέπουμε, το Recall που προκύπτει από όλες τις κατηγορίες είναι 0,836 που αποτελεί μία ισχυρή επίδοση για το μοντέλο. Πιο συγκεκριμένα, στις κατηγορίες με μεγαλύτερη συχνότητα εμφάνισης στο σύνολο δεδομένων όπως για παράδειγμα τα αυτοκίνητα, η επίδοση είναι εξαιρετική. Σε μικρότερες κατηγορίες όπως αυτή του Person sitting η επίδοση είναι χαμηλότερη.

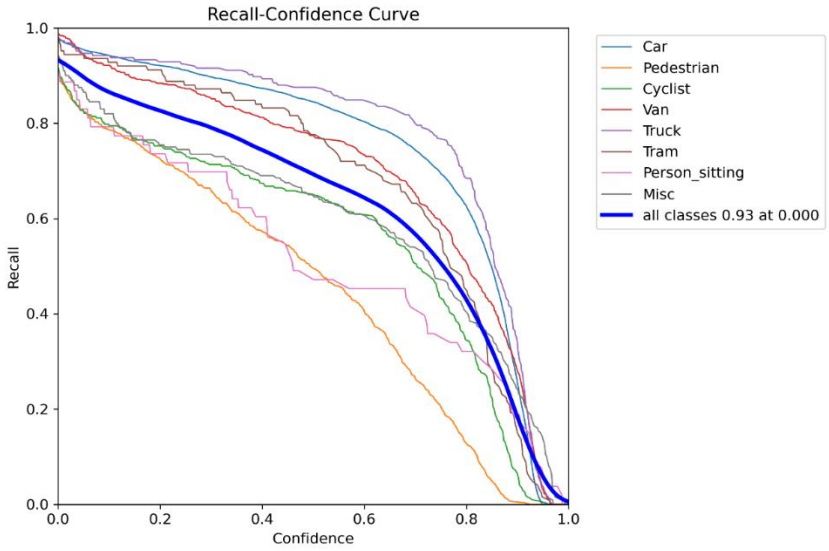
Στο παρακάτω διάγραμμα βλέπουμε την καμπύλη για το f1-score που δίνεται από την σχέση:

F1=2*((Precision*Recall)/(Precision+Recall))



Το γράφημα μας δείχνει μέγιστο f1 0,79 και βέλτιστο confidence threshold 0,36. Αναλυτικά, σε χαμηλό threshold (<0,2) έχουμε υψηλό recall, πολλά false positives και χαμηλό precision. Αντίθετα, σε υψηλό threshold (>0,7) έχουμε υψηλό percision και χαμηλό recall με αποτέλεσμα το f1 να πέφτει απότομα.

Τέλος, βλέπουμε το διάγραμμα που μας δείχνει πως μειώνεται το recall όσο αυξάνεται το confidence threshold.



Παρατηρούμε, πως μετά το 0,6 το threshold πέφτει απότομα.

Για το RCNN χρησιμοποιήθηκε το μοντέλο Faster R-CNN. Η υλοποίηση βασίστηκε στην αρχιτεκτονική fasterrcnn_resnet50_fpn της βιβλιοθήκης torchvision. Το Faster R-CNN αποτελεί ένα two-stage detector, δηλαδή η ανίχνευση πραγματοποιείται σε δύο στάδια. Αρχικά, έχουμε το Region Proposal Network κατά το οποίο προτείνονται υποψήφιες περιοχές (bounding boxes) όπου ενδέχεται να υπάρχουν αντικείμενα και στην συνέχεια το δεύτερο στάδιο κατά το οποίο για κάθε μία από τις περιοχές αυτές ταξινομείται το αντικείμενο που ανιχνεύεται.

Ως backbone χρησιμοποιούμε το ResNet-50 σε συνδυασμό με Feature Pyramid Network(FPN), επιτρέποντας την ανίχνευση αντικειμένων σε πολλαπλές κλίμακες. Το μοντέλο αρχικοποιήθηκε με προεκπαιδευμένα βάρη από το COCO dataset.

Η εκπαίδευση έγινε σε δύο στάδια. Στο πρώτο στάδιο κάναμε εκπαίδευση για 10 εποχές με χρήση 400 τυχαίων εικόνων για κάθε εποχή. Έχοντας κάνει έλεγχο στο τέλος των δέκα εποχών διαπιστώσαμε πως το μοντέλο μας έχει ήδη αρχίσει να μας δίνει κάποια ικανοποιητικά αποτελέσματα με περιθώρια βελτίωσης. Έτσι συνεχίσαμε την εκπαίδευση στο δεύτερο στάδιο με άλλες 15 εποχές με 600 εικόνες

ανά εποχή και ρίχνοντας το learning rate από 0,005 σε 0,003 για μικρότερη παρέμβαση στη διαμόρφωση των βαρών.

Μετά την εκπαίδευση συγκρίναμε τις επιδόσεις για δύο Confidence threshold. Αρχικά είδαμε για 0,5 και στην συνέχεια αυξήσαμε σε 0,6. Οι μετρικές που χρησιμοποιήσαμε είναι :

-True Positives

- False Positives

- False Negatives

- Precision

- Recall

-F1-score

Λάβαμε λοιπόν τα εξής αποτελέσματα:

Confidence threshold	0,5	0,6
TP	7092	6974
FP	1930	1441
FN	853	971
Precision	0.7861	0.8288
Recall	0.8926	0.8778
F1-score	0.8360	0.8526

Αρχικά, παρατηρούμε πως ενώ αυξήσαμε το όριο από 0,5 σε 0,6 τα True Positives παραμένουν υψηλά. Επομένως οι περισσότερες προβλέψεις έχουν confidence>0.6. Επιπλέον, με την αύξηση του ορίου είχαμε μείωση των False Positives σε 489 περιπτώσεις (μείωση περίπου 25%). Αυτό αποτελεί μία πολύ καλή βελτίωση του μοντέλου. Τέλος, παρατηρούμε μία αύξηση κατά 118 περιπτώσεων στα False Negatives. Όσο αυξάνουμε το όριο τόσο θα αυξάνεται και το FN. Ωστόσο, η αύξηση είναι μικρή συγκριτικά με την μείωση των FP.

Στην συνέχεια θα αξιολογήσουμε την μετρική precision. Παρατηρούμε, πως αυξάνοντας το Confidence threshold από 0,5 σε 0,6 το Precision βελτιώνεται και φτάνει στην τιμή 82,9%. Αυτό πρακτικά σημαίνει πως το 82,9% των προβλέψεων που έκανε το μοντέλο βγήκαν σωστό. Η τιμή αυτή είναι πολύ καλή για ένα μοντέλο προβλέψεων.

Έπειτα, πάμε στην μετρική Recall. Σε αυτή την μετρική το μοντέλο μας με Confidence threshold 0,6 πιάνει ποσοστό 87,8% που σημαίνει πως εντοπίζει μεγάλο ποσοστό των πραγματικών αντικειμένων. Βλέπουμε λοιπόν μια καλή επίδοση για two-stage

detector σε multi-class dataset και σε αυτή την μετρική.Υπάρχει μείωση συγκριτικά με την χρήση Confidence threshold 0,5 αλλά η μείωση αυτή είναι μικρή.

Τέλος, θα δούμε την επίδοση στο F1 score. Πρόκειται στην ουσία για την αρμονική μέση τιμή των δύο προηγούμενων μετρικών. Συγκριτικά με τα δύο όρια στο 0,6 βλέπουμε μεγαλύτερο ποσοστό στο F1
και αυτό οφείλεται στο γεγονός πως η αύξηση του precision ήταν μεγαλύτερη από την μείωση του recall.

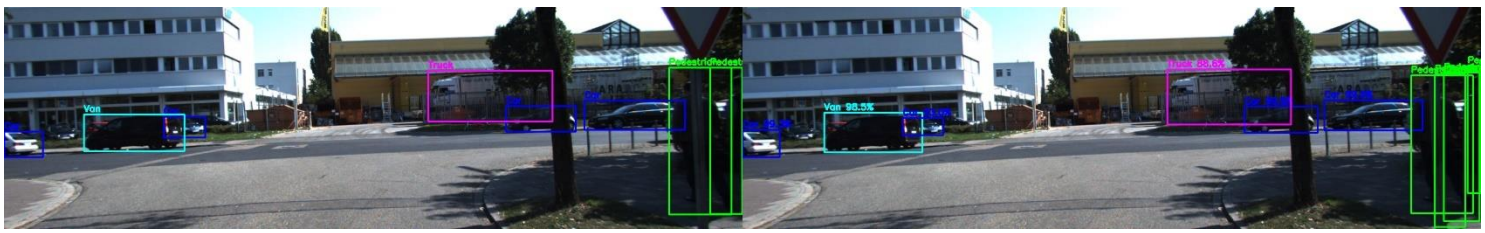
Μετά την εκπαίδευση εφαρμόσαμε validation στα μοντέλα μας με εικόνες από το dataset που χρησιμοποιήσαμε για την εκπαίδευση . Ενδεικτικά παρουσιάζουμε κάποια αποτελέσματα του validation. Στην αριστερή πλευρά βλέπουμε την εικόνα από το dataset καθώς και τα κουτιά ορίων όπως δίνονται στα αρχεία label. Στην δεξιά πλευρά βλέπουμε το αποτέλεσμα του μοντέλου για την ίδια εικόνα με την αντίστοιχη πιθανότητα πρόβλεψης.

Τα αποτελέσματα για τα δύο μοντέλα φαίνονται παρακάτω:

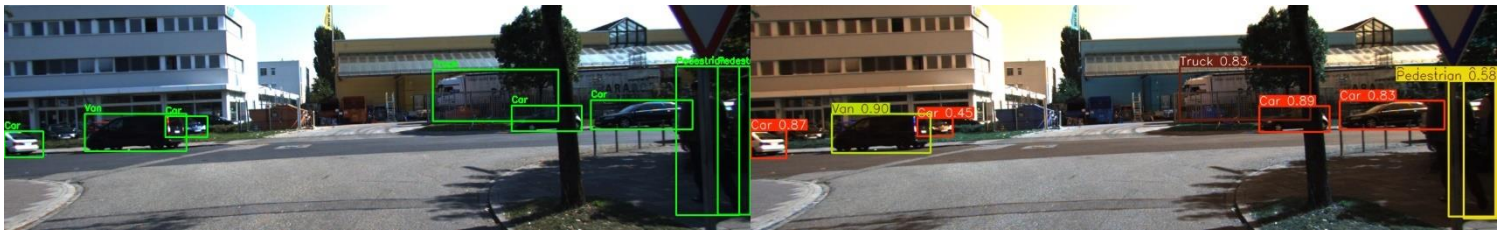
RCNN

YOLO

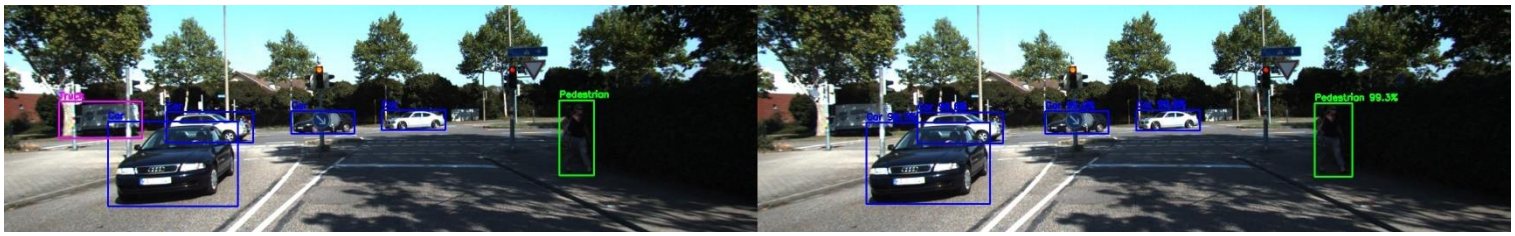
RCNN



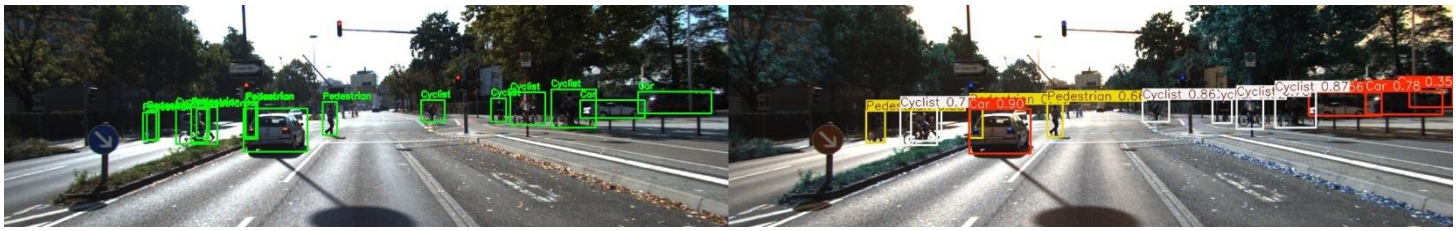
YOLO



RCNN



YOLO



RCNN

YOLO

RCNN

YOLO

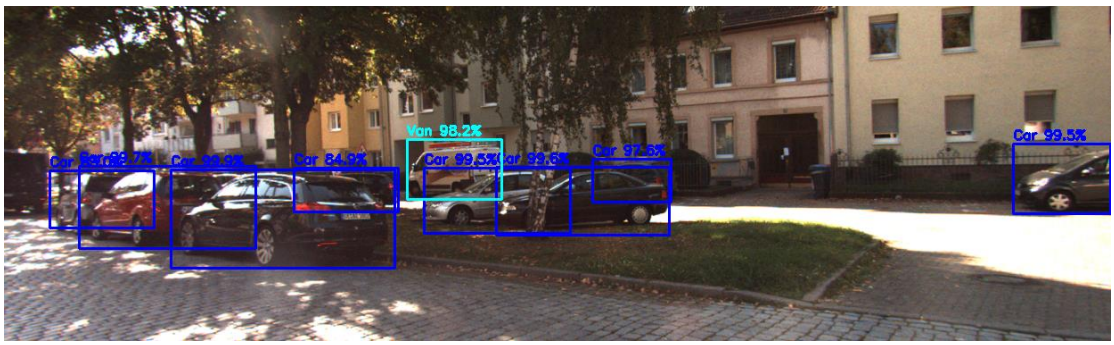
RCNN

YOLO

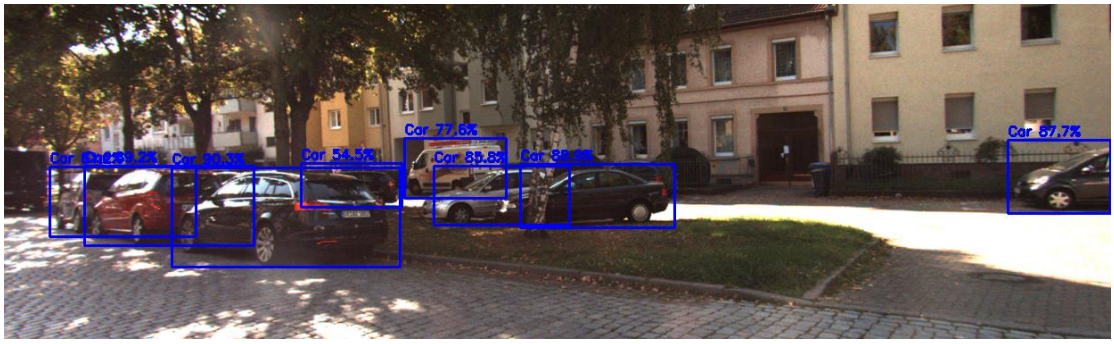
Έχοντας ολοκληρώσει την διαδικασία εκπαίδευσης και validation των δύο μοντέλων μας προχωρήσαμε στην εφαρμογή τους πάνω σε νέες εικόνες στις οποίες δεν έχουν εκπαιδευτεί.

Παρακάτω παρουσιάζουμε τα αποτελέσματα με κοινές εικόνες και για τα δύο μοντέλα.

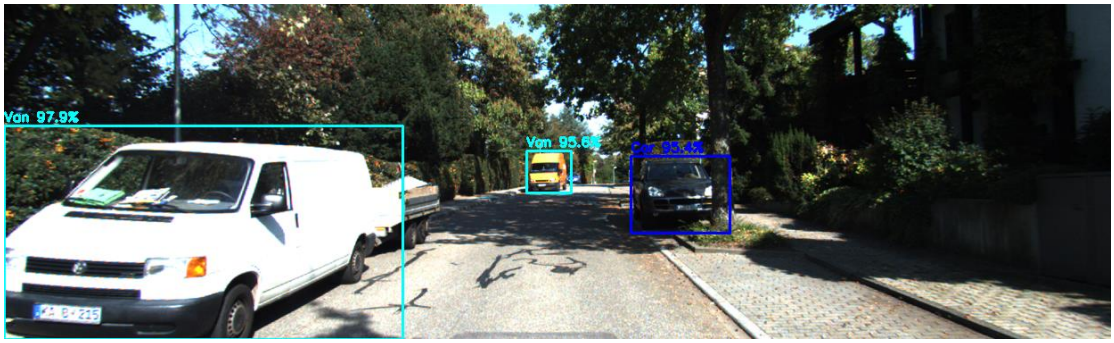
RCNN



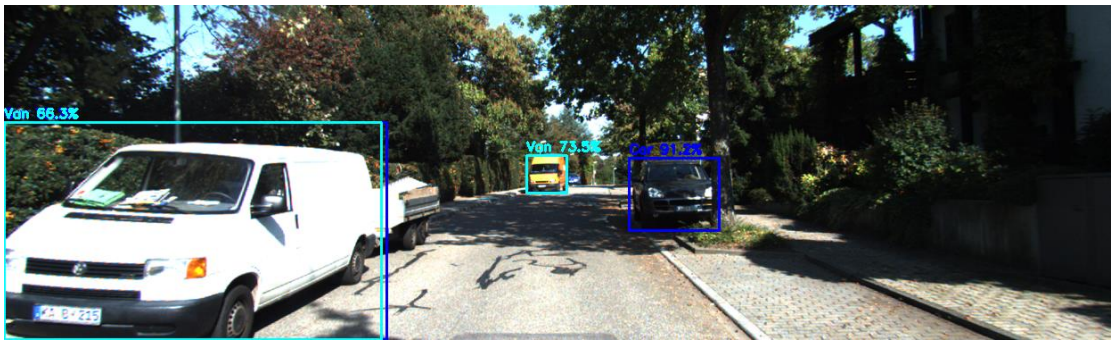
YOLO



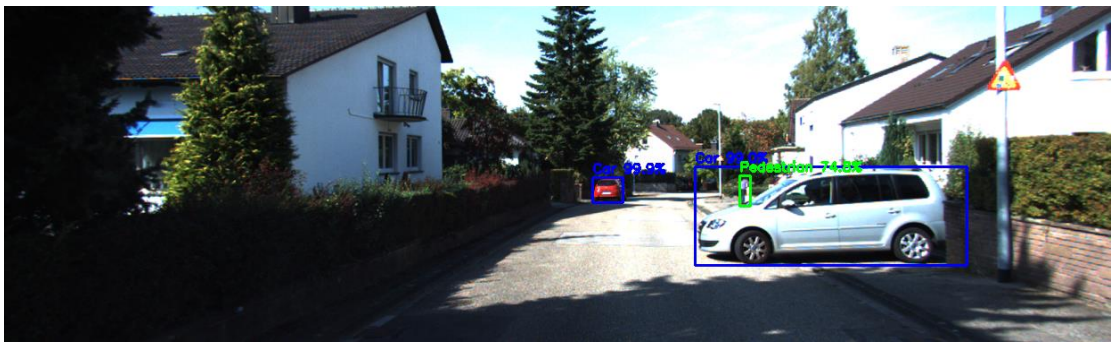
RCNN



YOLO



RCNN



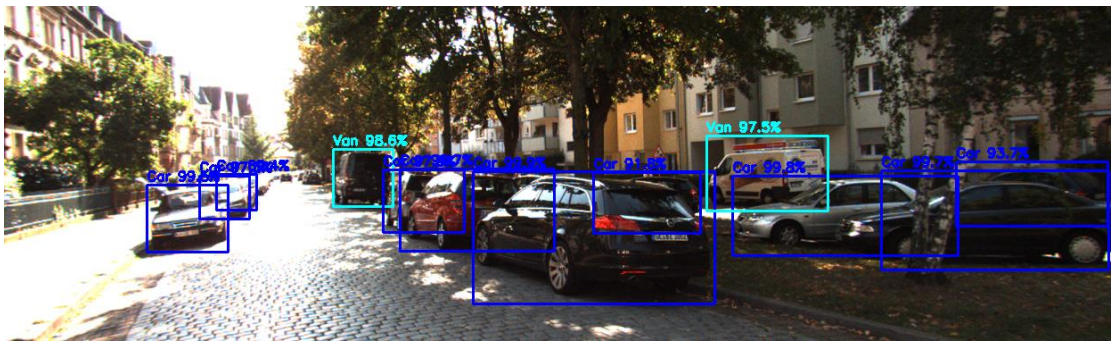
YOLO

Σε αυτό το παράδειγμα βλέπουμε πως ο RCNN ανταποκρίνεται με εντυπωσιακό τρόπο στην αναγνώριση μικρών αντικειμένων που και με γυμνό μάτι δυσκολευόμαστε να εντοπίσουμε.

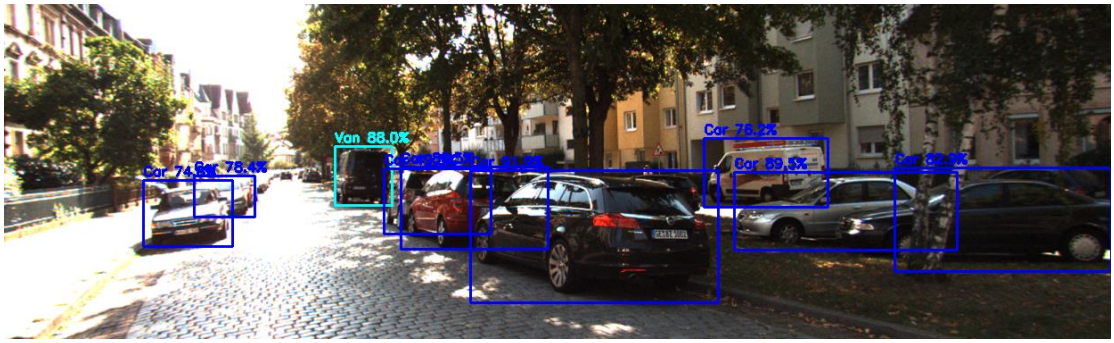
RCNN

YOLO

RCNN

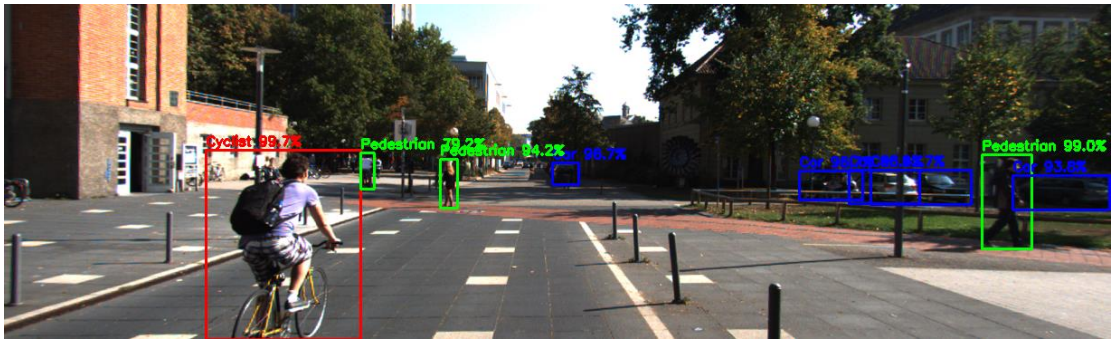


YOLO



Σε αυτή την εικόνα βλέπουμε ικανοποιητικά αποτελέσματα και από τα δύο μοντέλα. Αξίζει βέβαια να σχολιάσουμε πως το YOLO μοντέλο παρά τον σωστό εντοπισμό του αντικειμένου πέφτει συχνά σε λανθασμένη εκτίμηση κλάσης όπως στην περίπτωση του βαν.

RCNN



YOLO

Όπως αναφέραμε και παραπάνω το RCNN μοντέλο ανταποκρίνεται καλύτερα στον εντοπισμό μικρών αντικειμένων όπως οι πεζοί.

RCNN

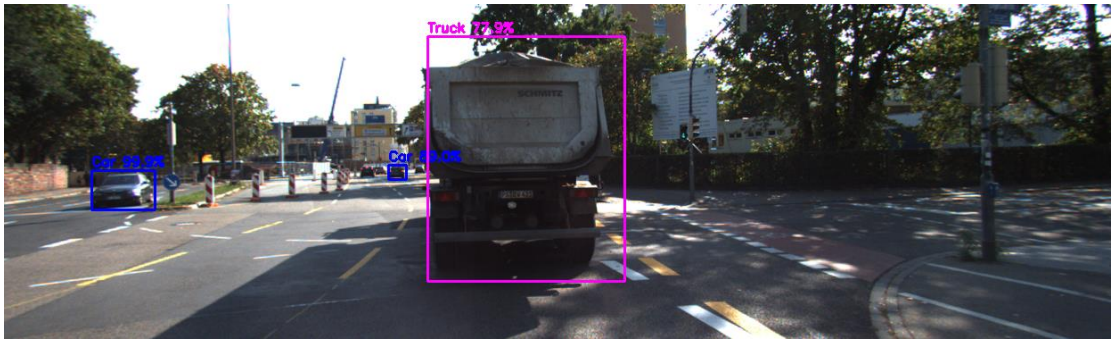
YOLO

Σε αυτή την περίπτωση, ενδεχομένως λόγω της αντίθεσης στο φως λόγω έντονης ηλιοφάνειας, το YOLO μοντέλο έχασε τον εντοπισμό ενός μεγάλου φορτηγού.

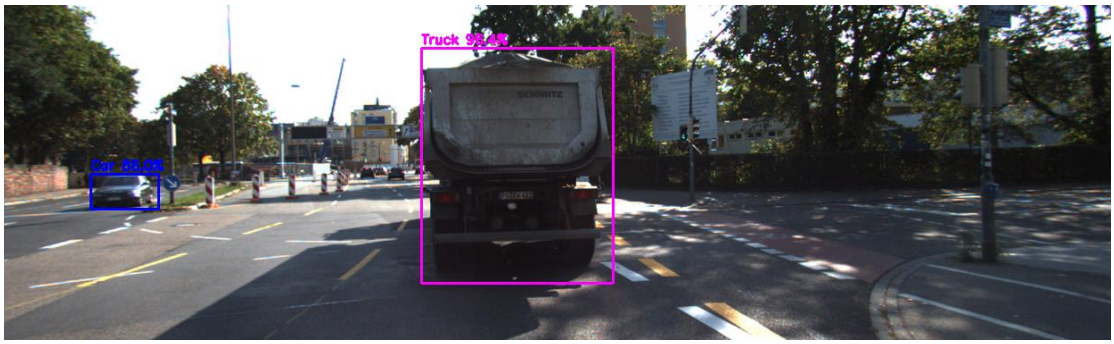
RCNN

YOLO

RCNN

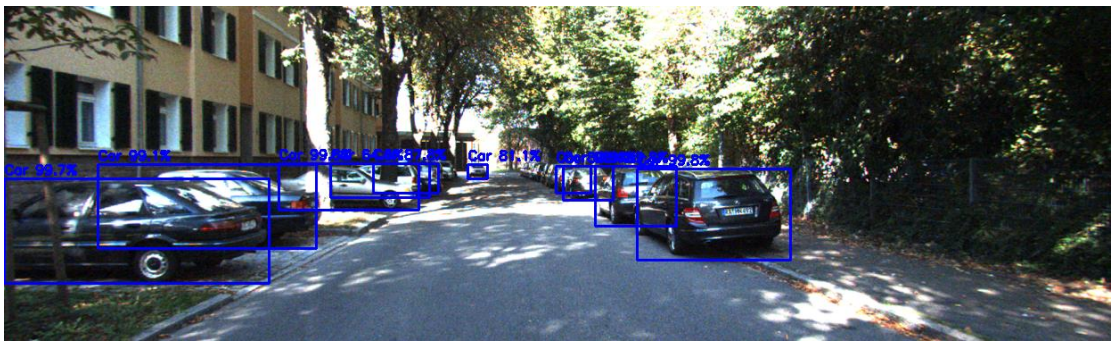


YOLO

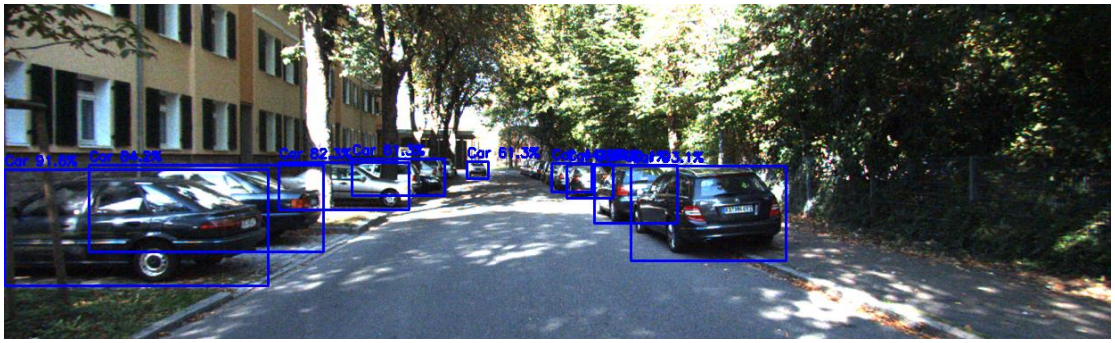


Στην παραπάνω εικόνα έχουμε τον σωστό εντοπισμό των βασικών οχημάτων και από τα δύο μοντέλα, με το YOLO να εντοπίζει με μεγαλύτερη πιθανότητα το φορτηγό στο κέντρο του δρόμου, χάνοντας όμως ξανά το μικρό όχημα στο βάθος.

RCNN

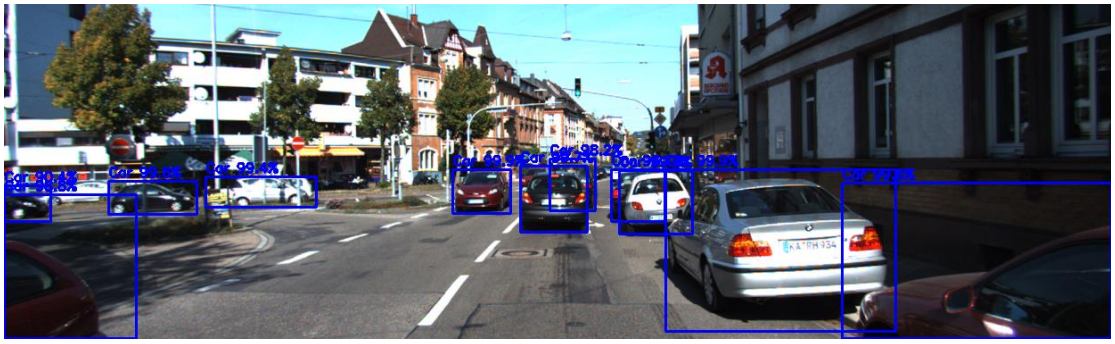


YOLO

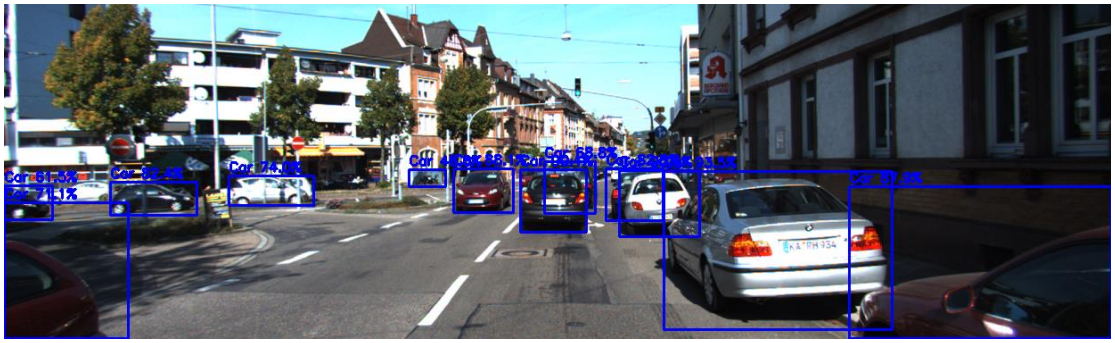


Η αναγνώριση οχημάτων στην παραπάνω περίπτωση έγινε επιτυχώς και από τα δύο μοντέλα παρά τις έντονες σκιές που θα μπορούσαν να τα ‘μπερδέψουν’.

RCNN



YOLO



6.3 Συμπεράσματα

Τα αποτελέσματα της παρούσας εργασίας μας δείχνουν πως:

- το Faster R-CNN παρουσίασε ελαφρώς υψηλότερη συνολική precision-recall (F1=0,8526).

-το YOLO εμφάνισε πολύ καλή συνολική απόδοση (mAP 0.836), με ιδιαίτερα υψηλή ακρίβεια σε κατηγορίες που εμφανίζονται συχνά όπως οι ‘’Cars’’ και ‘’Truck’’.

-το R-CNN εμφάνισε καλύτερο recall, ειδικά στις πιο σπάνιες κατηγορίες καθώς κα ιστις κατηγορίες που περιλαμβάνουν μικρότερα αντικείμενα.

Οι διαφορές αυτές οφείλονται κυρίως σε μια θεμελιώδη διαφορά στην αρχιτεκτονική των δύο μοντέλων. Το YOLO είναι one-stage detector που δίνει ιδιαίτερη στην ταχύτητα. Από την άλλη πλευρά, το Fater R-CNN είναι two-stage detector και πραγματοποιεί πρώτα region proposal και στην συνέχεια ταξινόμηση, γεγονός που οδηγεί σε υψηλότερο recall.

Καταλήγοντας, και τα δύο μοντέλα απέδωσαν ικανοποιητικά στο dataset KITTI, επιβεβαιώνοντας την καταλληλότητά τους για εφαρμογές ανίχνευσης αντικειμένων στο πεδίο της αυτόνομης οδήγησης.

Το Faster R-CNN παρουσίασε μεγαλύτερη ακρίβεια, ενώ το YOLO κατάφερε καλύτερες επιδόσεις στην ταχύτητα κρατώντας πολύ καλή επίδοση κυρίως στις πιο συχνές και μεγάλες κατηγορίες. Η τελική επιλογή μοντέλου εξαρτάται τελικά από τις απαιτήσεις της εφαρμογής που θέλουμε να υλοποιήσουμε. Εάν η προτεραιότητά μας είναι η μέγιστη ακρίβεια και η μείωση ψευδών ανιχνεύσεων τότε το Faster R-CNN αποτελεί την πιο κατάλληλη επιλογή. Εάν όμως θέλουμε να στηριχτούμε στην ταχύτητα σε συνδυασμό με υψηλή απόδοση τότε το YOLO αποτελεί καλύτερη επιλογή.

Βιβλιογραφία

1 Tom M. Mitchell, Machine Learning, 1997

2 Richard S. Sutton, Andrew G. Barto, Reinforcement Learning, 2018

3. Pang-Ning Tan, Introduction to Data Mining,2014

4. David G. Kleinbaum, Mitchel Klein, Logistic Regression: A Self-Learning Text, 2010

5. Claude Sammut, Geoffrey I. Webb, Encyclopedia of Machine Learning, 2010

6. Nello Cristianini, John Shawe-Taylor, An Introduction to Support Vector Machines, 2000

7. Βερύκιος Βασίλειος, Καγκλής Βασίλειος, Σταυρόπουλος Ηλίας, Η επιστήμη των δεδομένων μέσα από τη γλώσσα R, 2015

8. Anil K. Jain, Pattern Recognition Letters, 2010

9. Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 1996

10..Charu C. Aggarwal , Neural Networks and Deep Learning, 2018

11.Simon Haykin, Neural networks and learning machines, 1999

12.Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016

13.D. H. Hubel, T. N. Wiesel, Receptive Fields of Single Neurones in the Cat's Striate Cortex, 1959

14. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015

15. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, 2016

16.Joseph Redmon, Ali Farhadi, YOLO9000: Better, Faster, Stronger, 2016

17.Joseph Redmon, Ali Farhadi, YOLOv3: An Incremental Improvement, 2018

18.Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020

19.Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 2014

20. Ross Girshick, Fast R-CNN, 2015

21.Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2016

22.Shai Shalev-Shwartz, Shaked Shammah, Amnon Shashua, On a Formal Model of Safe and Scalable Self-driving Cars, 2018

23.Computer Vision, Algorithms and Applications, Richard Szeliski, 2022

24.Mobileye official site: Responsibility- Sensitive Safety, A mathematical model for automated vehicle safety https://www.mobileye.com/technology/responsibility-sensitive-safety/

25.Robert Meersman, Zahir Tari, Douglas C. Schmidt et al. (Eds.), On The Move to Meaningful Internet Systems 2023: CoopIS, DOA, and ODBASE, 2003

26.Xindong Wu, Vipin Kumar, The Top Ten Algorithms in Data Mining, 2009

27.Δημήτριος Πετρίδης, Ανάλυση Πολυμεταβλητών Τεχνικών, Εφαρμογές Περιπτώσεων, 2015

28.Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep Learning, 2015

29.Yoshua Bengio, Ian Goodfellow, Aaron Courville, Deep Learning, 2015

30.Singh, S., Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey, 2018

31.Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, Oscar Beijbom, nuScenes: A multimodal dataset for autonomous driving, 2020

32Abhishek Sarda, Dr.Shubhra Dixit, Dr. Anupama Bhan, Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm, 2021

33.Erich Schubert, Jörg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu, DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN, 2015

34.Umair Shafique, A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA), 2014

35.Ian H. Witten, Eibe Frank, Data Mining, Practical Machine Learning Tools and Techniques, 2005

36.Ford official site: Ford Trials Tech to Help Foresee Traffic Incidens; Connects Cars and Sensors to Improve Road Safety, 2020, https://media.ford.com/content/fordmedia/feu/en/news/2020/08/20/ford-trials-tech-to-help-foresee-traffic-incidents--connects-car.html

37.Ευστάθιος Κύρκος ,Επιχειρηματική Ευφυΐα και Εξόρυξη Δεδομένων, 2015

38.Οικονόμου Πολυχρόνης, Μαλεφάκη Σόνια, Μπατσίδης Απόστολος, Πιθανότητες – Στατιστική, 2022

39.Samprit Chatterjee, Ali S. Hadi, Regression Analysis By Example, 2012

40.The KITTI Vision Benchmark Suite official site:
https://www.cvlibs.net/datasets/kitti/index.php