



Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Βιομηχανικής Σχεδίασης και Παραγωγής

Διπλωματική εργασία

Τεχνητή νοημοσύνη και αυτόνομα οχήματα

Τζούλιο Μπ. Χαμίτι
71446176

Επιβλέπων:

κ. Γρηγόρης Νικολάου
Επίκουρος Καθηγητής

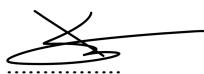
Αθήνα, Οκτώβριος, 2021

Εγκρίθηκε από την εξεταστική επιτροπή τον Οκτώβριο 2021.

Γρηγόρης Νικολάου

Σουλτάνα Βασιλειάδου

Χρήστος Δρόσος



Χαμίτι Τζούλιο

Τμήμα Βιομηχανικής Σχεδίασης και Παραγωγής

Πανεπιστήμιο Δυτικής Αττικής Copyright ©Χαμίτι Τζούλιο, 2021 Με επιφύλαξη παντός δικαιώματος. All rights reserved. Ο κάτωθι υπογεγραμμένος Χαμίτι Τζούλιο του Μπεσνίκ, με αριθμό μητρώου 71446176 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι: «Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Στους γονείς και την αδελφή μου.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Λέκτορα κο. Γρηγόρη Νικολάου, για την παρότρυνση, την ενθάρρυνση, την προθυμία του για συνεχείς συζητήσεις, την καθοδήγηση και τη βοήθεια του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Θα ήθελα επιπλέον να εκφράσω την εκ των προτέρων εκτίμησή μου στα υπόλοιπα μέλη της Επιτροπής για τις παρατηρήσεις τους για τη διπλωματική μου εργασία.

Αθήνα, Οκτώβριος 2021
Τζούλιο Χαμίτι

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η χρήση αλγορίθμων τεχνητής νοημοσύνης στον τομέα της αυτόνομης οδήγησης. Η είσοδος των αυτόνομων οχημάτων στο περιβάλλον οδήγησης αναμένεται να επιφέρει σημαντικές μεταβολές στην κυκλοφορία, βελτιώνοντας την ασφάλεια, την αποδοτικότητα και την άνεση των οδικών μεταφορών, καθιερώνοντας έναν νέο τρόπο μετακίνησης. Η παρούσα εργασία επικεντρώνεται στα συστήματα αντίληψης περιβάλλοντος με μεθόδους μηχανικής όρασης. Συγκεκριμένα μέσα σε μια ειδικά διαμορφωμένη πίστα θα χρησιμοποιήσουμε ένα όχημα/ρομπότ, το οποίο θα κινείται εντός των ορίων της πίστας και με έναν αυτόνομο πράκτορα θα μπορεί να αναγνωρίζει πεζούς καθώς και τις γραμμές στον δρόμο. Ως αποτέλεσμα το όχημα μας θα μπορεί να κινείται στην πόλη τελείως αυτόνομα χωρίς να χρειαστεί καμία ανθρώπινη επέμβαση.

Λέξεις Κλειδιά: Μηχανική Όραση, Συνελκτικά Νευρωνικά Δίκτυα, Yolov5, Duckiebot, Ανίχνευση Αντικειμένων

Abstract

The purpose of this thesis is the use of artificial intelligence algorithms in the field of autonomous driving. Research in autonomous vehicles is expected to bring about significant changes in traffic, improvement of safety, efficiency and convenience in road transport, introducing a new way of transportation. The present work focuses on environment perception systems with methods of computer vision. Specifically inside a specially designed track we will use a vehicle / robot, which will move within the limits of the track and using an autonomous agent that will be able to recognize pedestrians as well as the lines on the road. As a result our vehicle will be able to move in the city completely independently without the need for any human interaction.

Keywords: Computer Vision, Convolution Neural Networks, Yolov5, Duckiebot, Object Detection

Πίνακας περιεχομένων

Ευχαριστίες	iii
Περίληψη	v
Abstract	vii
Πίνακας περιεχομένων	ix
Πίνακας σχημάτων	xii
1 Εισαγωγή	1
1.1 Σκοπός και διάρθρωση της εργασίας	1
2 Θεωρητικό μέρος	3
2.1 Τεχνητή νοημοσύνη	3
2.1.1 Μηχανική Μάθηση	4
2.1.2 Βαθιά Μάθηση	7
2.2 Συνελκτικά Νευρωνικά Δίκτυα (CNNs)	10
2.2.1 Διαδικασία συνέλιξης	12
2.2.2 Συναρτήσεις Ενεργοποίησης	15
2.2.3 Πλήρες συνδεδεμένο στρώμα (Fully connected layer FC)	17
2.2.4 Συνάρτηση απώλειας (Loss Function)	18
2.2.5 Διαδικασία εκπαίδευσης Συνελκτικών νευρωνικών δικτύων	20
2.2.6 Συνελκτικά νευρωνικά δίκτυα για ταξινόμηση εικόνων	24
2.2.7 Συνελκτικά νευρωνικά δίκτυα και ανίχνευση αντικειμένων	24
2.2.8 Κριτήρια επίδοσης για ανιχνευτές αντικειμένων	27
3 Πρακτικό Μέρος	29
3.1 Duckiebot	29
3.2 Visual Lane Servoing	30
3.3 Σύνολο Δεδομένων	31
3.4 Μοντέλο Εκπαίδευσης	33
4 Αποτελέσματα	35
4.1 Αποτελέσματα του Yolov5	35
5 Επίλογος	39
Βιβλιογραφικές Αναφορές	41
Πρόσθετη Βιβλιογραφία	43

Πίνακας σχημάτων

Εικόνα 1.	Σχέση Τεχνητής νοημοσύνης, Μηχανικής Μάθησης και Βαθιάς Μάθησης[2]	3
Εικόνα 2.	Γράφημα Ακρίβειας-πολυπλοκότητας[6]	6
Εικόνα 3.	Πιθανά αποτελέσματα εκπαίδευσης[7]	6
Εικόνα 4.	Απεικόνιση ενός μοντέλου Perceptron[8]	8
Εικόνα 5.	Απεικόνιση της συνάρτησης XOR[8]	8
Εικόνα 6.	Στοχαστική Βαθμωτή Κατάβαση[9]	9
Εικόνα 7.	Παράδειγμα ενός φίλτρου 2x2[10]	10
Εικόνα 8.	Διαδικασία υποδειγματοληψίας[10]	11
Εικόνα 9.	Μαθηματική απεικόνιση διαδικασίας υποδειγματοληψίας[10]	11
Εικόνα 10.	Γκριζα εικόνα και τυχαίο φίλτρο[10]	12
Εικόνα 11.	Διαδικασία Συνέλιξης[10]	12
Εικόνα 12.	Τελικός χάρτης χαρακτηριστικών[10]	13
Εικόνα 13.	Μαθηματικός τύπος για την εύρεση του χάρτη χαρακτηριστικών[10]	13
Εικόνα 14.	Διαδικασία συνέλιξης με μηδενικό padding[10]	14
Εικόνα 15.	Μαθηματική αναπαράσταση της σιγμοειδής συνάρτησης[10]	15
Εικόνα 16.	Μαθηματική αναπαράσταση της σιγμοειδής συνάρτησης[10]	15
Εικόνα 17.	Μαθηματική εξίσωση συνάρτησης ReLU[10]	16
Εικόνα 18.	Μαθηματική εξίσωση συνάρτησης Leaky ReLU[10]	16
Εικόνα 19.	Μαθηματική εξίσωση Noisy ReLU[10]	16
Εικόνα 20.	Γραφική αναπαράσταση Noisy ReLU[10]	16
Εικόνα 21.	Πλήρη συνδεδεμένα στρώματα[10]	17
Εικόνα 22.	Εξίσωση για να βρεθεί η πιθανότητα για κάθε εξόδο[10]	18
Εικόνα 23.	Μαθηματική εξίσωση της συνάρτησης Cross Entropy[10]	18
Εικόνα 24.	Μαθηματική εξίσωση της Ευκλείδειας συνάρτησης[10]	19
Εικόνα 25.	Μαθηματική εξίσωση της συνάρτησης Hinge[10]	19
Εικόνα 26.	Η διαδικασία μέσης αφαίρεσης[10]	20
Εικόνα 27.	Μαθηματική εξίσωση της κανονικοποίησης[10]	21
Εικόνα 28.	Αύξηση δεδομένων[11]	22
Εικόνα 29.	Η τεχνική σύνδεσης Dropout[10]	23
Εικόνα 30.	Χρονοδιάγραμμα διαφορετικών μοντέλων	24
Εικόνα 31.	Το μοντέλο R-CNN[12]	24
Εικόνα 32.	Το μοντέλο Fast R-CNN[13]	25
Εικόνα 33.	Μοντέλο Faster R-CNN[14]	26
Εικόνα 34.	Το μοντέλο Yolo[15]	26
Εικόνα 35.	Υπολογισμός IoU[16]	27
Εικόνα 36.	Παραδειγμά με διαφορετικά IoU[16]	28
Εικόνα 37.	Ενα Duckiebot[17]	29
Εικόνα 38.	Εικόνα απο την κάμερα του Duckiebot	30
Εικόνα 39.	Η εικόνα μετά τα φίλτρα	30

Εικόνα 40.	Το σύνολο των δεδομένων μας	31
Εικόνα 41.	Πως βλέπει ο αλγόριθμος τα δεδομένα εκπαίδευσης 1	32
Εικόνα 42.	Πως βλέπει ο αλγόριθμος τα δεδομένα εκπαίδευσης 2	32
Εικόνα 43.	Τελικά αποτελέσματα του εκπαιδευμένου δικτύου	33
Εικόνα 44.	Γραφήματα απώλειας του Yolov5	35
Εικόνα 45.	Γράφημα ευστοχίας και ανάκλασης	36
Εικόνα 46.	Καμπύλη Precision-Recall	36
Εικόνα 47.	Πίνακας σύγκυσης	37
Εικόνα 48.	Στιγμιότυπο αποτο πείραμα 1	38
Εικόνα 49.	Στιγμιότυπο αποτο πείραμα 2	38

Κεφάλαιο 1

Εισαγωγή

Η παρούσα διπλωματική εργασία συντάχθηκε χρησιμοποιώντας τον αλγόριθμο Yolo, συγκεκριμένα την 5η έκδοση του, σε ένα ρομπότ (Duckiebot) εντός μιας ειδικά διαμορφωμένης πίστας. Στο πλαίσιο της εργασίας συλλέξαμε ένα σύνολο δεδομένων τα οποία ήταν εικόνες από πλαστικά πατάκια που υποδήλωναν τους πεζούς στον δρόμο και κόνους που υποδήλωναν εμπόδια στον δρόμο. Στην συνέχεια με χρήση των παραπάνω δεδομένων προπονήσαμε τον αλγόριθμο YoloV5 ο οποίος χρησιμοποιεί νευρωνικά δίκτυα για να ανιχνεύσει αντικείμενα (εν προκειμένω πατάκια και κόνους) σε πραγματικό χρόνο. Παράλληλα με το YoloV5, δούλεψε ένας ειδικός αλγόριθμος ανίχνευσης λωρίδας προκειμένου το ρομπότ να καθοδηγείται αυτόνομα.

1.1 Σκοπός και διάρθρωση της εργασίας

Η εργασία διαρθρώνεται σε τρία κεφάλαια με έμφαση στην εκπαίδευση του χρησιμοποιηθέντος αλγορίθμου (YoloV5). Ειδικότερα ιδιαίτερη αναφορά γίνεται στο θεωρητικό υπόβαθρο της μηχανικής μάθησης καθώς και των νευρωνικών δικτύων. Εν συνεχεία, η εργασία επικεντρώνεται στην εφαρμογή των παραπάνω σε πρακτικό επίπεδο, καταλήγοντας σε αναλυτική παρουσίαση των εξαχθέντων αποτελεσμάτων με την μορφή γραφημάτων και εικόνων.

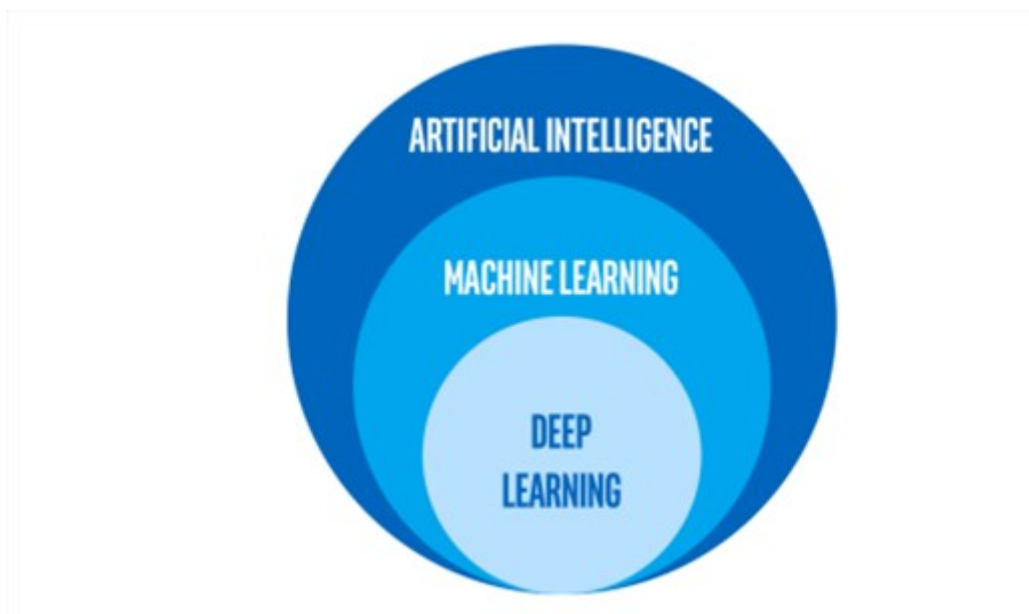
Κεφάλαιο 2

Θεωρητικό μέρος

2.1 Τεχνητή νοημοσύνη

Η τεχνητή νοημοσύνη είναι ένας κλάδος της επιστήμης των υπολογιστών που μελετά την σχέση μεταξύ υπολογισμού και γνώσης. Η έρευνα στην τεχνητή νοημοσύνη περιλαμβάνει τη συγγραφή προγραμμάτων που προσπαθούν να επιτύχουν κάποιο είδος ευφυούς συμπεριφοράς. Πολλές ανθρώπινες δραστηριότητες, όπως η σύνταξη προγραμμάτων υπολογιστών, τα μαθηματικά, ο συλλογισμός κοινής λογικής, η κατανόηση της γλώσσας αλλά ακόμη και η οδήγηση ενός αυτοκινήτου λέγεται ότι απαιτούν «νοημοσύνη». Τις τελευταίες δεκαετίες, έχουν κατασκευαστεί διάφορα συστήματα υπολογιστών που μπορούν να εκτελούν εργασίες όπως αυτές. Πιο συγκεκριμένα, υπάρχουν συστήματα υπολογιστών που μπορούν να διαγνώσουν ασθένειες, να σχεδιάσουν τη σύνθεση πολύπλοκων οργανικών χημικών ενώσεων, να επιλύσουν διαφορικές εξισώσεις σε συμβολική μορφή, να αναλύσουν ηλεκτρονικά κυκλώματα, να κατανοήσουν περιορισμένες ποσότητες κειμένου φυσικής γλώσσας και ανθρώπινης ομιλίας ή να αποκτήσουν «όραση» ξεχωρίζοντας διάφορα χαρακτηριστικά σε μια εικόνα, το οποίο αποτελεί και θέμα της παρούσης εργασίας.[1]

Με το πέρασμα του χρόνου δημιουργήθηκαν υποπεδία στην τεχνητή νοημοσύνη, η μηχανική μάθηση που είναι η ικανότητα μάθησης χωρίς να προγραμματίζεται κάτι ρητά, και η βαθιά μάθηση η οποία μιμείται τους νευρώνες του ανθρώπινου εγκεφάλου και μαθαίνει με τον ίδιο τρόπο.



Εικόνα 1. Σχέση Τεχνητής νοημοσύνης, Μηχανικής Μάθησης και Βαθιάς Μάθησης[2]

2.1.1 Μηχανική Μάθηση

Η μηχανική μάθηση μπορεί να οριστεί ευρέως ως μέθοδοι υπολογισμού χρησιμοποιώντας μαθηματικά μοντέλα και δεδομένα προκειμένου να γίνει μια ακριβής πρόβλεψη. Αυτά τα δεδομένα θα μπορούσαν να έχουν τη μορφή ψηφιοποιημένων σετ εκπαίδευσης με ανθρώπινη σήμανση ή άλλων τύπων πληροφοριών που λαμβάνονται μέσω αλληλεπίδρασης με το περιβάλλον. Σε όλες τις περιπτώσεις, η ποιότητα και το μέγεθός τους είναι ζωτικής σημασίας για την επιτυχία των προβλέψεων που κάνει ο αλγόριθμος. Εφόσον η επιτυχία αυτών των αλγορίθμων βασίζεται τόσο πολύ στα δεδομένα που χρησιμοποιούνται, η μηχανική μάθηση είναι συσχετιζόμενη με την ανάλυση δεδομένων και την στατιστική.[3] Γενικότερα συνδυάζονται ιδέες της στατιστικής, των πιθανοτήτων και της βελτιστοποίησης με έννοιες της επιστήμης των υπολογιστών, προκειμένου να δημιουργηθεί ένας ανθεκτικός αλγόριθμος, ο οποίος θα είναι αποτελεσματικός και σε παραδείγματα που δεν έχει ξαναδεί στα δεδομένα εκπαίδευσης του. Αυτό ονομάζεται γενίκευση και είναι ο στόχος που προσπαθούμε να πετύχουμε για όλους του αλγορίθμους μηχανικής μάθησης. Θεωρητικά θα μπορούσαμε να δώσουμε στον αλγόριθμο μας έναν τεράστιο όγκο δεδομένων και να πετύχουμε την γενίκευση, ωστόσο πρακτικά αυτό από μόνο του δεν είναι αρκετό, διότι υπάρχει το πρόβλημα της υπερπροσαρμογής στα δεδομένα, δηλαδή υφίσταται ο κίνδυνος ο αλγόριθμος να απομνημονεύει τα δεδομένα με τα οποία εκπαιδεύεται. Προκειμένου να αποφύγουμε την υπερπροσαρμογή αλλά και να επιτύχουμε την γενίκευση χρειαζόμαστε μεγάλο όγκο δεδομένων τα οποία είναι διαφορετικά μεταξύ τους. Χωρίζουμε τους τρόπους μάθησης σε υποκατηγορίες ανάλογα με τον τρόπο που μαθαίνει ο αλγόριθμος από τα δεδομένα σε:

- **Επιβλεπόμενη μάθηση:** Ο χρήστης παρέχει στον αλγόριθμο ζευγάρια δεδομένων εισόδου και επιθυμητής εξόδου και ο αλγόριθμος βρίσκει τον τρόπο να παράγει την επιθυμητή έξοδο ανάλογα με την εκάστοτε είσοδο. Με την πάροδο του χρόνου εκπαίδευσης ο αλγόριθμος μαθαίνει την σχέση μεταξύ των δεδομένων με στόχο στο τέλος της εκπαίδευσης να είναι σε θέση να παράγει μια έξοδο για ένα δεδομένο που δεν έχει ξαναδεί, χωρίς ανθρώπινη παρέμβαση. Για παράδειγμα η αναγνώριση ενός λουλουδιού σε ποιά οικογένεια ανήκει. Έχουμε ως είσοδο την εικόνα του λουλουδιού και ως επιθυμητή έξοδο το είδος του. Με αυτό τον τρόπο δημιουργούμε μια βάση δεδομένων γεμάτη με εικόνες λουλουδιών και ετικέτες για το είδος που ανήκουν και τις φορτώνουμε στον αλγόριθμο. Ένα άλλο παράδειγμα είναι η πρόβλεψη για την τιμή ενός σπιτιού προς πώληση. Στην συγκεκριμένη περίπτωση τα δεδομένα εισόδου μπορεί να είναι οι τιμές των γύρο σπιτιών, η απόστασή τους από σταθμούς συγκοινωνιών, ο βαθμός εγκληματικότητας κ.ο.κ. ο αλγόριθμος με την σειρά του χρησιμοποιώντας μαθηματικά μοντέλα και την «γνώση» που έχει για τις τιμές των ήδη υπάρχοντων σπιτιών κάνει μια πρόβλεψη για ένα καινούργιο σπίτι.
- **Μη επιβλεπόμενη μάθηση:** Σε αυτή την ομάδα αλγορίθμων μόνο τα δεδομένα εισόδου είναι γνωστά χωρίς να δίνετε κανένα δεδομένο εξόδου στον αλγόριθμο. Υπάρχουν φυσικά πολλές επιτυχημένες εφαρμογές αυτών των αλγορίθμων αλλά ο τρόπος λειτουργίας τους, κάνει πιο δύσκολη την κατανόηση και την αξιολόγηση τους. Ένα παράδειγμα χρήσης τους είναι η ομαδοποίηση καταναλωτών σε ομάδες ανάλογα με τις προτιμήσεις τους. Δεδομένου μιας βάσης δεδομένων με προτιμήσεις πελατών μπορεί κάποια εταιρία να θέλει να χωρίσει τους πελάτες της ανάλογα με προηγούμενες αγορές τους, προκειμένου να μάθει περισσότερα πράγματα και να οργανώσει μελλοντικά το πλάνο λειτουργίας της. Επειδή δεν γνωρίζει κανείς εξαρχής ποιες θα είναι αυτές οι ομάδες ή πόσες θα είναι αυτό κάνει άγνωστα για τον αλγόριθμο τα δεδομένα εξόδου. Μία άλλη χρήση της μη επιβλεπόμενης μάθησης είναι επιλογή σημαντικότερων δεδομένων. Ας υποθέσουμε ότι θέλουμε να προβλέψουμε πόσο ικανός είναι ένας υποψήφιος να αποπληρώσει ένα δάνειο από την προοπτική μιας τράπεζας. Εδώ, πρέπει να βοηθήσουμε την τράπεζα να δημιουργήσει ένα σύστημα μηχανικής μάθησης έτσι ώστε κάθε δάνειο να μπορεί να δοθεί σε αιτούντες που μπορούν να το εξοφλήσουν.

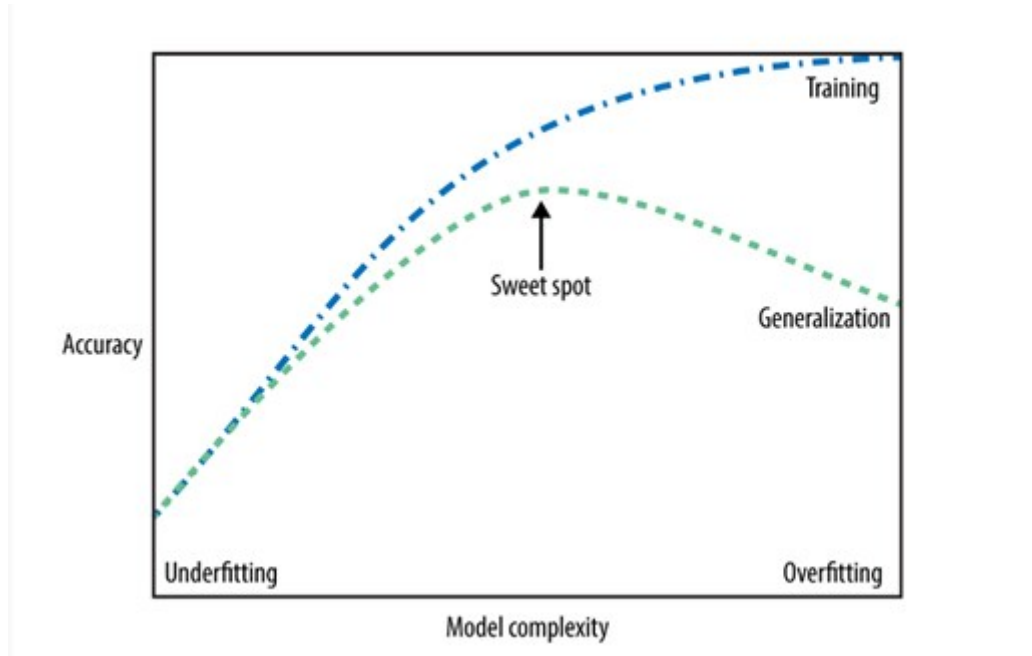
Χρειαζόμαστε πολλές πληροφορίες για κάθε αίτηση για να κάνουμε προβλέψεις. Μερικά σημαντικά χαρακτηριστικά σχετικά με τους αιτούντες είναι το μέσο μηνιαίο εισόδημα, το χρέος, το πιστωτικό ιστορικό και ούτω καθεξής. Ωστόσο, οι τράπεζες συλλέγουν πολλές περισσότερες πληροφορίες από τους αιτούντες κατά τη λήψη των αιτήσεων τους, πράγμα που κάνει πολλές από αυτές λιγότερο χρήσιμες. Δεδομένα όπως, η ηλικία του αιτούντος, η εθνικότητα του και η διεύθυνση κατοικίας του είναι λιγότερο σημαντικά σε σχέση με το μηνιαίο του εισόδημα και το ιστορικό του με την συγκεκριμένη τράπεζα. Έτσι, η μη επιβλεπόμενη μάθηση καταφέρνει και απορρίπτει τέτοιου είδους δεδομένα, φροντίζοντας ο αλγόριθμος να χρησιμοποιήσει τα καλύτερα από αυτά.

- **Ενισχυτική μάθηση:** Η ενισχυτική μάθηση αφορά την εκπαίδευση μοντέλων μηχανικής μάθησης με σκοπό την σωστή λήψη μιας ακολουθίας αποφάσεων. Ο πράκτορας μαθαίνει να επιτυγχάνει έναν στόχο σε ένα αβέβαιο, περίπλοκο για αυτόν περιβάλλον και ο τρόπος με τον οποίο καταφέρνει να βρει λύση είναι αντιμετωπίζοντας το πρόβλημα ως ένα παιχνίδι, με την μέθοδο δοκιμής-σφάλματος. Προκειμένου να δείξουμε στον αλγόριθμο τι θέλουμε να κάνει σχεδιάζουμε ένα σύστημα ανταμοιβών και ποινών έτσι ώστε, κάθε φορά που κάνει κάτι σωστό ή λάθος να μαθαίνει από αυτό αντίστοιχα. Παρότι δίνονται στον αλγόριθμο ανταμοιβές και ποινές, που αποτελούν ουσιαστικά τους κανόνες του παιχνιδιού, δεν δίνεται καμία βοήθεια ή πρόταση για να κερδίσει το παιχνίδι. Εναπόκειται στο μοντέλο να καταλάβει πώς να εκτελέσει την εργασία για να μεγιστοποιήσει την ανταμοιβή, ξεκινώντας από εντελώς τυχαίες δοκιμές και τελειώνοντας με εξελεγμένες τακτικές και υπεράνθρωπες δεξιότητες. Συγκεντρώνοντας εμπειρία από χιλιάδες επαναλαμβανόμενες δοκιμές και παιχνίδια που εκτελούνται παράλληλα, η ενισχυτική μάθηση είναι ο πιο αποτελεσματικός τρόπος να δούμε την δημιουργικότητα ενός αλγορίθμου.[4]

Υπάρχουν 2 βασικές κατηγορίες προβλημάτων επιβλεπόμενης μάθησης, οι οποίες ονομάζονται ταξινόμηση και παλινδρόμηση. Στην ταξινόμηση, ο αλγόριθμος λειτουργεί σαν μια συνάρτηση (f), η οποία αντιστοιχεί κάθε δεδομένου εισόδου της (x) σε μια διακριτή έξοδο (y). Αυτές οι έξοδοι ονομάζονται κλάσης και βρίσκονται σε μια προκαθορισμένη γνωστή λίστα. Πρακτικά, ο αλγόριθμος μας δίνει ως έξοδο την πιθανότητα που έχει το κάθε δεδομένο να ανήκει στην οποιαδήποτε κλάση. Για παράδειγμα, αν δεχθεί ως είσοδο ένα ηλεκτρονικό μήνυμα(e-mail) και πρέπει να προβλέψει αν θα μεταφερθεί στον φάκελο ανεπιθύμητα ή σημαντικά ο αλγόριθμος θα μας δώσει μια πιθανότητα 0.1 για τα σημαντικά και 0.9 για τα ανεπιθύμητα, στην συνέχεια μετατρέπουμε αυτές τις πιθανότητες σε κλάσεις εξόδου. Τα προβλήματα παλινδρόμησης λειτουργούν ως μια συνάρτηση (f), η οποία αντιστοιχεί κάθε δεδομένου εισόδου της (x) σε μια συνεχή έξοδο. Αναλυτικότερα, στα προβλήματα παλινδρόμησης οι επιθυμητές τιμές είναι πραγματικοί αριθμοί για τους οποίους απαιτούνται κριτήρια μέτρησης της απόστασης μεταξύ της εξόδου του μοντέλου και της επιθυμητής τιμής.[5]

Οι πιο επιτυχημένοι αλγόριθμοι μηχανικής μάθησης είναι εκείνοι που αυτοματοποιούν διαδικασίες λήψης αποφάσεων γενικεύοντας από γνωστά παραδείγματα. Συνήθως χτίζουμε μοντέλα με τέτοιο τρόπο ώστε να κάνουν ακριβείς προβλέψεις στα δεδομένα εκπαίδευσης και ελέγχουμε την ακρίβεια τους σε δεδομένα δοκιμής τα οποία δεν έχει ξαναδεί ο αλγόριθμος. Εάν τα δεδομένα εκπαίδευσης και δοκιμής έχουν αρκετά κοινά χαρακτηριστικά περιμένουμε το μοντέλο να είναι το ίδιο επιτυχημένο και στα δεδομένα δοκιμής. Ωστόσο, σε πολλές περιπτώσεις δεν είναι τόσο εύκολο. Πρακτικά μπορούμε να χρησιμοποιήσουμε έναν πολύ περίπλοκο αλγόριθμο ο οποίος θα μπορεί να έχει τεράστιο ποσοστό ακρίβειας στα δεδομένα εκπαίδευσης του. Αυτό ωστόσο, κάνει κακό στον αλγόριθμο αφού παρατηρείται ότι μαθαίνει απέξω τα δεδομένα εκπαίδευσης και ως αποτέλεσμα δεν αποδίδει καλά στα δεδομένα δοκιμής. Το παραπάνω φαινόμενο ονομάζεται υπερπροσαρμογή. Η υπερπροσαρμογή συμβαίνει όταν το μοντέλο μας επικεντρώνεται σε λεπτομέρειες των δεδομένων εκπαίδευσης και έτσι δεν μπορεί να γενικεύει σε δεδομένα που δεν έχει ξαναδεί. Αντιθέτως αν το μοντέλο είναι πολύ απλό τότε παρατηρούμε ότι δεν αποδίδει καλά ούτε στα δε-

δομένα εκπαίδευσης. Όσο πιο πολύπλοκο κάνουμε το μοντέλο μας, τόσο καλύτερα μαθαίνει τα δεδομένα εκπαίδευσης του αλλά, η μεγάλη πολυπλοκότητα οδηγεί στην υπερπροσαρμογή και το μοντέλο δεν γενικεύει σε άλλα δεδομένα. Δουλεία μας είναι να βρίσκουμε το ηδυσημείο ανάμεσα στην πολυπλοκότητα και την υπερπροσαρμογή για να αποκτήσουμε ένα επιτυχημένο μοντέλο μηχανικής μάθησης.[6]



Εικόνα 2. Γράφημα Ακρίβειας-πολυπλοκότητας[6]

Στην παραπάνω εικόνα παρατηρούμαι ότι όσο ανεβαίνει η πολυπλοκότητα, τόσο ανεβαίνει και η ακρίβεια του μοντέλου μας αρκεί να μην περάσουμε το ηδυσημείο πράγμα που μειώνει την γενίκευση και ωθεί στην υπερπροσαρμογή.



Εικόνα 3. Πιθανά αποτελέσματα εκπαίδευσης[7]

Παραπάνω βλέπουμε αριστερά ότι η γνώση του μοντέλου είναι ανεπαρκής για τα δεδομένα, δεξιά το φαινόμενο της υπερπροσαρμογής και ανάμεσα τους το ιδανικό.

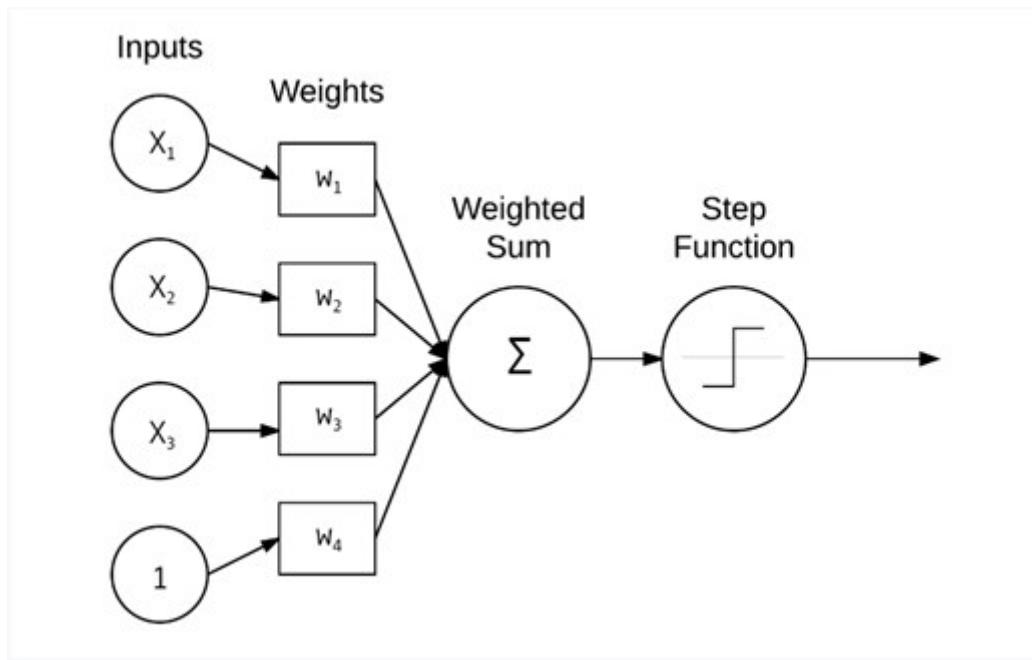
2.1.2 Βαθιά Μάθηση

Ο κεντρικός στόχος της τεχνητής νοημοσύνης είναι να παρέχει ένα σύνολο αλγορίθμων και τεχνολογιών, οι οποίες μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων που οι άνθρωποι εκτελούν διαισθητικά και αυτόματα. Ένα παράδειγμα τέτοιων προβλημάτων αποτελεί η διερμηνεία και η κατανόηση των περιεχομένων μιας εικόνας. Η παραπάνω διαδικασία αν και είναι κάτι φυσικό για τους ανθρώπους που δεν απαιτεί καμία προσπάθεια, έχει αποδειχτεί ότι συνιστά μεγάλο πρόβλημα για τους υπολογιστές. Ενώ η τεχνητή νοημοσύνη ενσωματώνει ένα ποικίλο σύνολο εργασιών που σχετίζονται με την αυτόματη λογική (όπως είναι για παράδειγμα να βγάξει συμπεράσματα, να σχεδιάζει την λύση ενός προβλήματος με βελτιστοποίηση, να βρίσκει μια μέθοδο για πιο γρήγορη λύση σε ένα πρόβλημα, κ.κ.), το υποπεδίο της μηχανικής μάθησης τείνει να εστιάζει στην αναγνώριση μοτίβων και στην εκμάθηση από δεδομένα. Τα νευρωνικά δίκτυα αποτελούν ένα είδος αλγορίθμων μηχανικής μάθησης εμπνευσμένων από την λειτουργία του εγκεφάλου, τα οποία μαθαίνουν από δεδομένα και ειδικεύονται στην αναγνώριση μοτίβων. Βασικό δομικό στοιχείο του εγκεφάλου είναι οι νευρώνες, δηλαδή τα νευρικά κύτταρα τα οποία δημιουργούν ένα πυκνό δίκτυο επικοινωνίας μεταξύ τους. Η μελέτη των νευρώνων γίνεται προκειμένου να ανακαλυφθεί ένα νέο υπολογιστικό μοντέλο βασισμένο σε μια δικτυακή δομή παρόμοια με αυτή του εγκεφάλου η οποία αποσκοπεί στο να είναι κατάλληλη για ανάπτυξη ευφυών αλγορίθμων και γενικότερα διαδικασιών σχετιζόμενων με τη νοημοσύνη, όπως η μάθηση η μνήμη, η γενίκευση κ.λπ.[5]

Η ιστορία των νευρωνικών δικτύων και της βαθιάς μάθησης ξεκινάει από το 1940, αλλάζοντας αρχικά πολλά ονόματα όπως κυβερνητικά συστήματα, συστήματα διασύνδεσης και καταλήγοντας στο πιο γνωστό, τα τεχνητά νευρωνικά δίκτυα. Παρότι είναι εμπνευσμένα από τον ανθρώπινο εγκέφαλο και τον τρόπο που αντιδράνε οι νευρώνες του, τα νευρωνικά δίκτυα δεν αποτελούν ρεαλιστικά μοντέλα του εγκεφάλου μας. Αντιθέτως, αντλούν έμπνευση από αυτόν, γεγονός που μας επιτρέπει να οδηγούμαστε σε παραλληλισμούς ανάμεσα σε ένα πολύ βασικό μοντέλο του εγκεφάλου και τον τρόπο που μπορούμε να μιμηθούμε την συμπεριφορά του μέσω των νευρωνικών δικτύων.[8] Το πρώτο νευρωνικό μοντέλο ήρθε από τους McCulloch και Pitts το 1943. Ήταν μοντέλο δυαδικής ταξινόμησης που αναγνώριζε 2 διαφορετικές καταστάσεις ανάλογα την είσοδο του. Η κατάσταση του νευρώνα περιγράφεται από έναν αριθμό y όπου $y = 0 \rightarrow$ ο νευρώνας είναι αδρανής (δεν πυροδοτεί παλμούς) $y=1 \rightarrow$ ο νευρώνας πυροδοτεί παλμούς στη μέγιστη συχνότητα. Οι συνάψεις περιγράφονται από τα συναπτικά βάρη (synaptic weights) w_i , τα οποία είναι πραγματικοί αριθμοί και, πιο συγκεκριμένα, θετικοί για τις ενισχυτικές συνάψεις και αρνητικοί για τις ανασταλτικές συνάψεις. Έστω ότι x_1, x_2, \dots, x_n είναι οι είσοδοι του νευρώνα. Ελέγχουμε αν το άθροισμα $w_1x_1 + \dots + w_nx_n$ του φορτίου που δέχεται ο νευρώνας είναι μεγαλύτερο από ένα κατώφλι θ :

- Αν ισχύει $w_1x_1 + \dots + w_nx_n > \theta$, ο νευρώνας πυροδοτεί παλμούς.
- Διαφορετικά, ο νευρώνας παραμένει αδρανής.

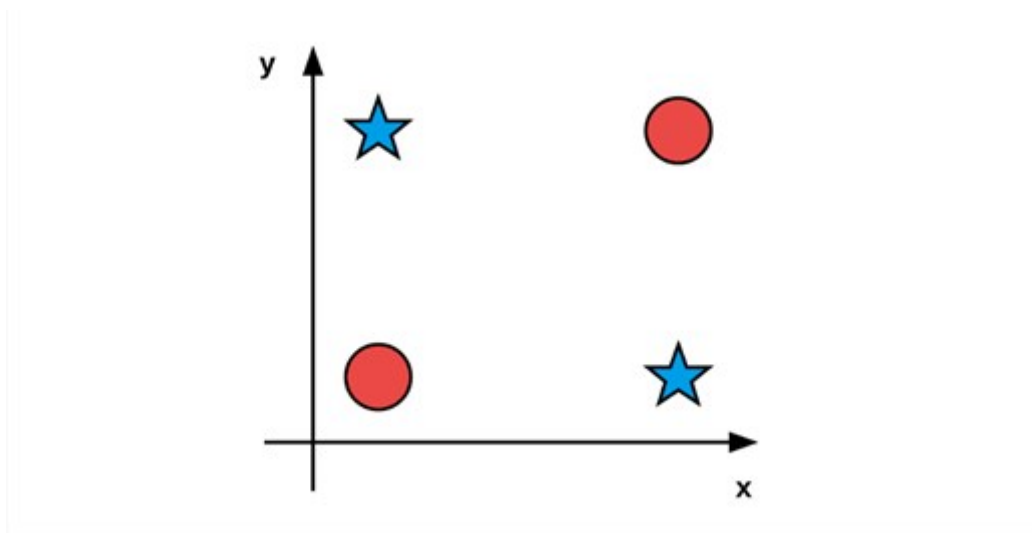
Το κύριο πρόβλημα αυτού του μοντέλου ήταν ότι τα συναπτικά βάρη που χρησιμοποιούνταν για να καθορίσουμε την κατάσταση του νευρώνα, έπρεπε να συντονιστούν χειροκίνητα από έναν άνθρωπο, γεγονός που κάνει το μοντέλο να χάνει τον σκοπό του εάν απαιτείται ανθρώπινος χειριστής. Έπειτα, το 1950 ο αλγόριθμος Perceptron δημοσιεύτηκε από τον Rosenblatt. Αυτό το μοντέλο μπορούσε αυτόματα να μάθει τα συνοπτικά βάρη προκειμένου να ταξινομήσει μια είσοδο χωρίς ανθρώπινη παρέμβαση.[8]



Εικόνα 4. Απεικόνιση ενός μοντέλου Perceptron[8]

Ένα παράδειγμα ενός απλού μοντέλου Perceptron που δέχεται εισόδους, υπολογίζει το άθροισμα μαζί με τα συναπτικά βάρη και εφαρμόζει μια συνάρτηση ενεργοποίησης του νευρώνα.

Κατά την διάρκεια εκείνης της περιόδου, οι τεχνικές βασισμένες στο μοντέλο Perceptron ήταν οι πιο δημοφιλείς όσον αφορά την κοινότητα των νευρωνικών δικτύων. Ωστόσο, το 1969 μια δημοσίευση από τους Minsky και Papert άφησε στάσιμο το πεδίο της έρευνας στα νευρωνικά δίκτυα. Η δημοσίευση τους αποδείκνυε ότι ένα μοντέλο Perceptron με μία γραμμική συνάρτηση ενεργοποίησης ήταν ένας απλός αλγόριθμος γραμμικής ταξινόμησης που χρησιμοποιούνταν και στην απλή μηχανική μάθηση, ο οποίος αδυνατούσε να λύσει μη γραμμικά προβλήματα. Ένα παράδειγμα τέτοιου προβλήματος ήταν η συνάρτηση XOR που απεικονίζεται παρακάτω.[8]

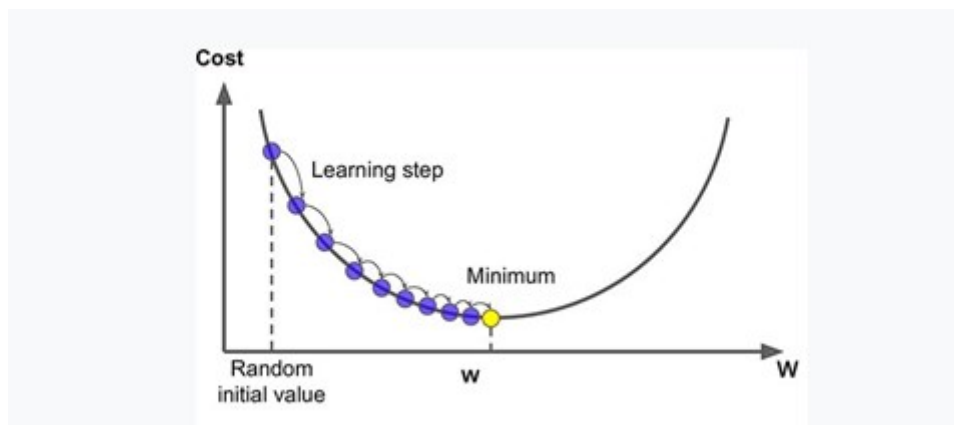


Εικόνα 5. Απεικόνιση της συνάρτησης XOR[8]

Παρατηρείται ότι δεν υπάρχει τρόπος να χωρίσουμε με μία γραμμή τις εξόδους της συνάρτησης γεγονός που καθιστά το πρόβλημα μη γραμμικό. Επιπλέον, η δημοσίευση υποστήριζε ότι εκείνη

την περίοδο δεν υπήρχαν αρκετοί υπολογιστικοί πόροι προκειμένου να δημιουργηθούν μεγάλα, βαθιά νευρωνικά δίκτυα.

Έπειτα όμως από την δημιουργία του αλγορίθμου ανάστροφης διάδοσης και την έρευνα των Werbos (1974) και Rumelhart (1986) τα νευρωνικά δίκτυα ξαναήρθαν στην επιφάνεια. Συνδυάζοντας τον αλγόριθμο ανάστροφης διάδοσης μαζί με μη γραμμικές συναρτήσεις ενεργοποίησης, οι ερευνητές μπορούσαν πλέον να λύσουν μη γραμμικά προβλήματα λύνοντας την εξίσωση XOR και ανοίγοντας δρόμο για ένα τελείως καινούργιο πεδίο έρευνας στα νευρωνικά δίκτυα. Πρακτικά η ανάστροφη διάδοση μας επιτρέπει να εκπαιδεύουμε τον νευρωνικό δίκτυο και να «μαθαίνει» από τα λάθη του. Αναλυτικότερα όταν ξεκινάει η εκπαίδευση του νευρωνικού δικτύου τα βάρη παίρνουν τυχαίες τιμές ή την τιμή 0. Κάθε φορά που περνάει μια εποχή εκπαίδευσης, δηλαδή ένας πλήρης κύκλος εκπαίδευσης, η ανάστροφη διάδοση ενημερώνει όλα τα προηγούμενα βάρη και σφάλματα προκειμένου να φτάσει μια επιθυμητή τιμή. Η μέθοδος κατά την οποία το νευρωνικό δίκτυο ενημερώνει μετά από κάθε εποχή εκπαίδευσης τα βάρη του και τα συγκρίνει στο τέλος με μια επιθυμητή τιμή, υπολογίζει την διαφορά και επαναλαμβάνει τα ίδια βήματα ονομάζεται στοχαστική βαθμωτή κατάβαση.[5]



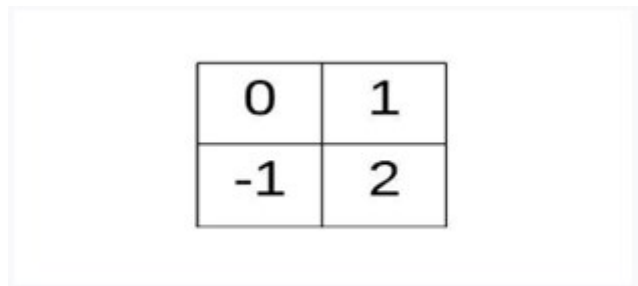
Εικόνα 6. Στοχαστική Βαθμωτή Κατάβαση[9]

Σημερινή εξέλιξη των νευρωνικών δικτύων αποτελεί η βαθιά μάθηση. Το χαρακτηριστικό που την ξεχωρίζει από όλες τις προηγούμενες μεθόδους που αναφέραμε είναι ότι έχουμε γρηγορότερα, πιο ειδικευμένα μηχανήματα και πολλά περισσότερα δεδομένα για εκπαίδευση. Μπορούμε πλέον να εκπαιδεύουμε νευρωνικά δίκτυα με περισσότερα κρυφά στρώματα τα οποία, έχουν την δυνατότητα να μαθαίνουν ιεραρχικά δηλαδή, οι απλές έννοιες μαθαίνονται στα κατώτερα στρώματα και τα πιο αφηρημένα χαρακτηριστικά μαθαίνονται στα υψηλότερα στρώματα του δικτύου. Ίσως το βασικότερο παράδειγμα της εφαρμοσμένης βαθιάς μάθησης όσον αφορά την εκμάθηση χαρακτηριστικών είναι τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks – CNNs). Ξεκίνησαν να εφαρμόζονται το 1988 για την αναγνώριση χειρόγραφων χαρακτήρων και αυτόματα μάθαιναν διακριτικά χαρακτηριστικά μιας εικόνας στοιβάζοντας διαδοχικά στρώματα το ένα μετά το άλλο και ως αποτέλεσμα κατάφεραν να αναγνωρίζουν γωνίες και άκρες στα κατώτερα στρώματα του δικτύου ενώ τα ανώτερα στρώματα χρησιμοποιούσαν αυτές τις γωνίες προκειμένου να μάθουν πιο αφηρημένα μοτίβα, χρήσιμα για την διαφοροποίηση μεταξύ κλάσεων ταξινόμησης. Σε πολλές εφαρμογές, τα συνελκτικά νευρωνικά δίκτυα θεωρούνται οι πιο δυνατοί ταξινομητές εικόνων και είναι σήμερα υπεύθυνοι για την εξέλιξη στο πεδίο της όρασης του υπολογιστή άλλα και τα υποπεδία του που χρησιμοποιούν μηχανική μάθηση.[5]

2.2 Συνελκτικά Νευρωνικά Δίκτυα (CNNs)

Τα συνελκτικά νευρωνικά δίκτυα (convolutional neural networks –CNNs) είναι δίκτυα πολλών στρώματων κατάλληλα κυρίως για λειτουργίες αναγνώρισης εικόνας. Αποτελούνται από εναλλασσόμενα στρώματα συνέλιξης και υποδειγματοληψίας, των οποίων η δομή περιγράφεται παρακάτω. Μετά το τελευταίο στρώμα συνέλιξης ή υποδειγματοληψίας ακολουθούν ένα ή περισσότερα πλήρως συνδεδεμένα στρώματα τα οποία λειτουργούν ως ταξινομητές. Η αλληλουχία των στρώματων συνέλιξης και υποδειγματοληψίας υλοποιεί ουσιαστικά έναν μετασχηματισμό της εικόνας εισόδου σε χάρτη χαρακτηριστικών. Κατόπιν, ο χάρτης αυτός εισάγεται ως είσοδος σε έναν ταξινομητή, ο οποίος με την σειρά του μας δίνει πιθανότητες για κάθε ξεχωριστή κλάση που έχουμε ορίσει στο δίκτυο. Τα στρώματα ενός συνελκτικού νευρωνικού δικτύου και οι διαδικασίες που λαμβάνουν μέρος σε αυτό περιγράφονται παρακάτω:

- **Στρώμα συνέλιξης:** Είναι το πιο σημαντικό στοιχείο οποιουδήποτε συνελκτικού δικτύου. Περιέχει ένα σύνολο από πυρήνες (kernels) ή αλλιώς φίλτρα, τα οποία χρησιμοποιούνται στην εικόνα εισόδου και στην συνέχεια παράγουν έναν χάρτη χαρακτηριστικών ως έξοδο.
- **Πυρήνας (kernel):** Ένας πυρήνας αποτελείται από ένα πλέγμα διακριτών τιμών ή αριθμών, στο οποίο κάθε τιμή είναι γνωστή ως το βάρος εκείνου του πυρήνα. Στην αρχή της εκπαίδευσης σε ένα συνελκτικό νευρωνικό δίκτυο, όλα τα βάρη του πυρήνα ανατίθενται με τυχαία νούμερα. Στην συνέχεια, με κάθε εποχή εκπαίδευσης, τα βάρη συντονίζονται και ο πυρήνας μαθαίνει να εξάγει σημαντικά χαρακτηριστικά. Παρακάτω φαίνεται ένα φίλτρο 2 διαστάσεων.

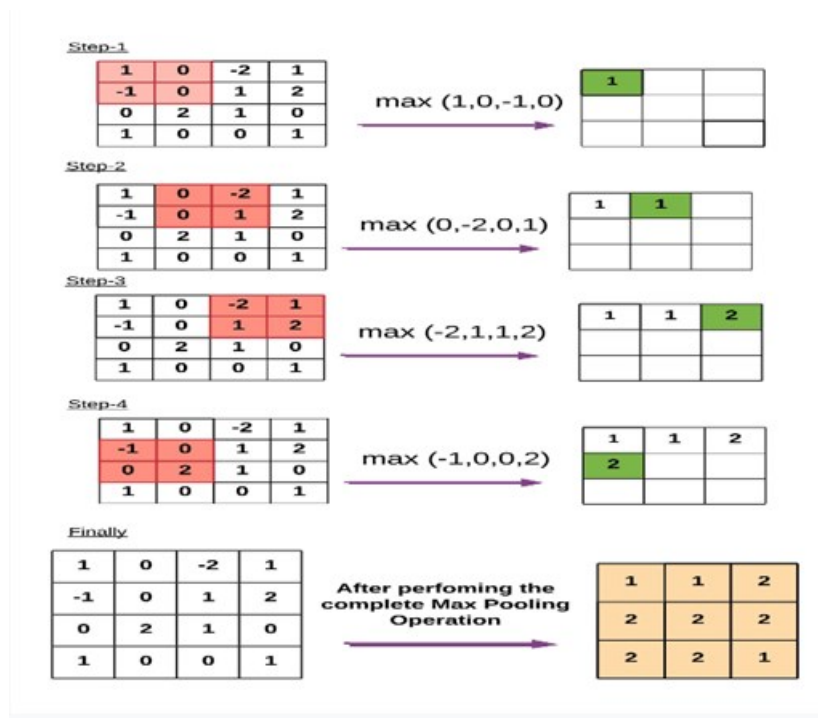


0	1
-1	2

Εικόνα 7. Παράδειγμα ενός φίλτρου 2x2[10]

- **Στρώμα υποδειγματοληψίας:** Τα στρώματα υποδειγματοληψίας (pooling layers) χρησιμοποιούνται για να παίρνουν υποδείγματα από τον χάρτη χαρακτηριστικών (παράγεται από το στρώμα συνέλιξης) και ουσιαστικά δέχονται χάρτες χαρακτηριστικών μεγάλου μεγέθους και τους συστέλλουν σε χάρτες χαρακτηριστικών με μικρότερο μέγεθος. Ενώ συστέλλονται, οι χάρτες χαρακτηριστικών διατηρούν τα πιο σημαντικά χαρακτηριστικά δηλαδή, τις πιο σημαντικές πληροφορίες τους σε κάθε στρώμα. Η διαδικασία υποδειγματοληψίας είναι παρόμοια με την διαδικασία συνέλιξης αφού, δηλώνουμε το μέγεθος της περιοχής που θα λάβουμε ως είσοδο αλλά και το stride για να προχωρήσουμε στην επόμενη περιοχή της εικόνας. Υπάρχουν διαφορετικές τεχνικές υποδειγματοληψίας όπως η max pooling, min pooling, average pooling, gated pooling, tree pooling, κ.α. Η τεχνική max pooling είναι η πιο γνωστή και αυτή που χρησιμοποιείται πιο συχνά όσον αφορά τις τεχνικές υποδειγματοληψίας. Πρακτικά ο τρόπος που λειτουργεί είναι ότι ελέγχει όλα τα pixel στην περιοχή που δηλώσαμε ως είσοδο και μεταφέρει στο επόμενο στρώμα το pixel με την μεγαλύτερη τιμή πριν περάσει στην επομένη περιοχή με το βήμα stride.

Το βασικό μειονέκτημα των στρωμάτων υποδειγματοληψίας είναι ότι κάποιες φορές μειώνουν συνολικά την απόδοση του συνελκτικού νευρωνικού δικτύου. Αυτό γίνεται επειδή, το στρώμα υποδειγματοληψίας βοηθάει το συνελκτικό νευρωνικό δίκτυο να βρει αν υπάρχει ένα συγκεκριμένο χαρακτηριστικό ή όχι, χωρίς να δίνει σημασία στην σωστή τοποθεσία του συγκεκριμένου χαρακτηριστικού στην εικόνα. Βλέπουμε ένα παράδειγμα στρώματος υποδειγματοληψίας όπου η περιοχή εισόδου είναι της τιμής 2x2 (εμφανίζεται με πορτοκαλί χρώμα), stride της τιμής 1 και την εκάστοτε τιμή εξόδου του χάρτη χαρακτηριστικών (εμφανίζεται με πράσινο).



Εικόνα 8. Διαδικασία υποδειγματοληψίας[10]

Η εξίσωση για να βρεθεί ο χάρτης χαρακτηριστικών μετά την διαδικασία υποδειγματοληψίας εμφανίζεται παρακάτω: Όπου το h' δηλώνει το ύψος του χάρτη χαρακτηριστικών της

$$h' = \left\lfloor \frac{h - f}{s} \right\rfloor$$

$$w' = \left\lfloor \frac{w - f}{s} \right\rfloor$$

Εικόνα 9. Μαθηματική απεικόνιση διαδικασίας υποδειγματοληψίας[10]

εξόδου, το w' δηλώνει το πλάτος του χάρτη χαρακτηριστικών της εξόδου, το h δηλώνει το ύψος του χάρτη χαρακτηριστικών της εισόδου, το w δηλώνει το πλάτος του χάρτη χαρακτηριστικών της εισόδου, το f είναι η περιοχή δειγματοληψίας και το s δηλώνει το stride της διαδικασία δειγματοληψίας.

2.2.1 Διαδικασία συνέλιξης

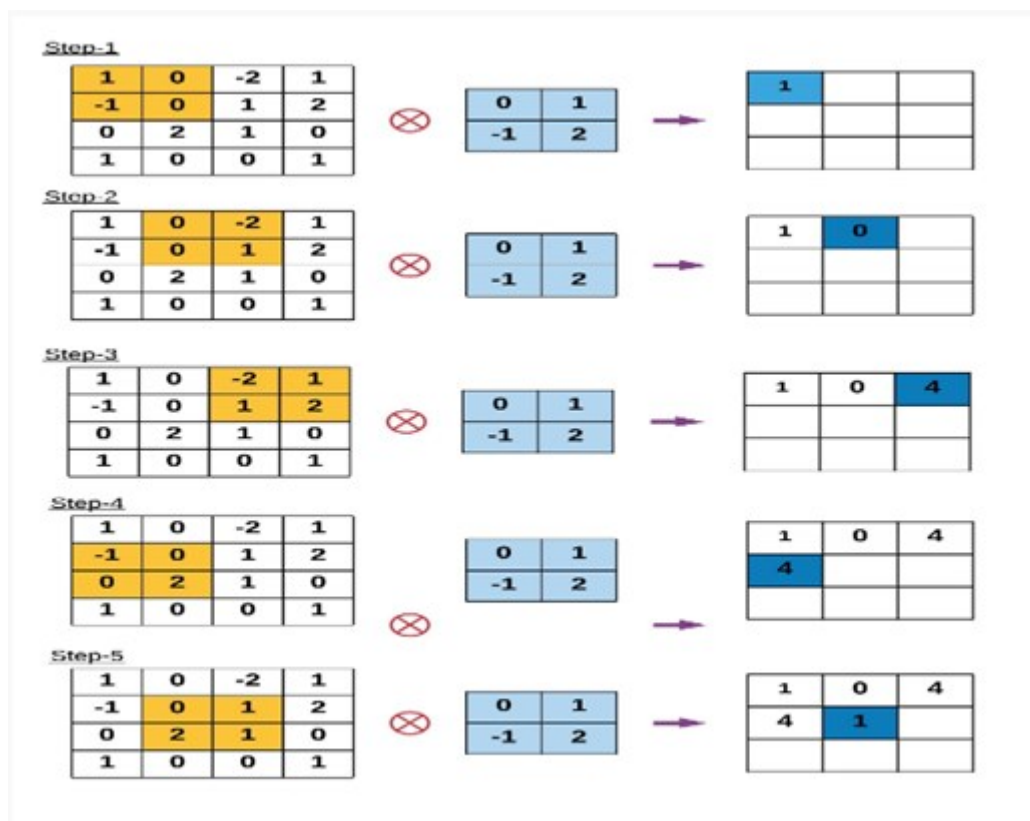
Σε αντίθεση με τα κλασσικά νευρωνικά δίκτυα όπου η είσοδος δεδομένων γίνεται σε μορφή διανυσμάτων, στα συνελικτικά νευρωνικά δίκτυα η είσοδος είναι μια εικόνα πολλών καναλιών (π.χ. στις RGB εικόνες έχουμε μια εικόνα με 3 κανάλια, ενώ στις γκριζες εικόνες έχουμε 1 μόνο κανάλι). Για να καταλάβουμε την διαδικασία συνέλιξης θα πάρουμε μια γκριζα εικόνα με διαστάσεις 4x4 και ένα φίλτρο με ένα τυχαίο αρχικό βάρος όπως δείξαμε παραπάνω.

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

0	1
-1	2

Εικόνα 10. Γκριζα εικόνα και τυχαίο φίλτρο[10]

Στην διαδικασία συνέλιξης, παίρνουμε το 2x2 φίλτρο και το περνάμε οριζόντια και κάθετα πάνω από όλη την 4x4 εικόνα και στην πορεία πολλαπλασιάζουμε όλες τις τιμές μεταξύ τους και στο τέλος υπολογίζουμε το συνολικό άθροισμα. Στον παρακάτω πίνακα φαίνονται τα βήματα την διαδικασίας όπου το φίλτρο (εμφανίζεται με μπλέ χρώμα) πολλαπλασιάζεται με την περιοχή της εικόνας (εμφανίζεται με κίτρινο χρώμα) μέσα στην 4x4 εικόνα και τα αποτελέσματα αθροίζονται στον χάρτη χαρακτηριστικών με κάθε βήμα συνέλιξης.



Εικόνα 11. Διαδικασία Συνέλιξης[10]

Αφού ολοκληρωθεί η διαδικασία έχουμε το τελικό χάρτη χαρακτηριστικών.

1	0	4
4	1	1
1	1	2

Εικόνα 12. Τελικός χάρτης χαρακτηριστικών[10]

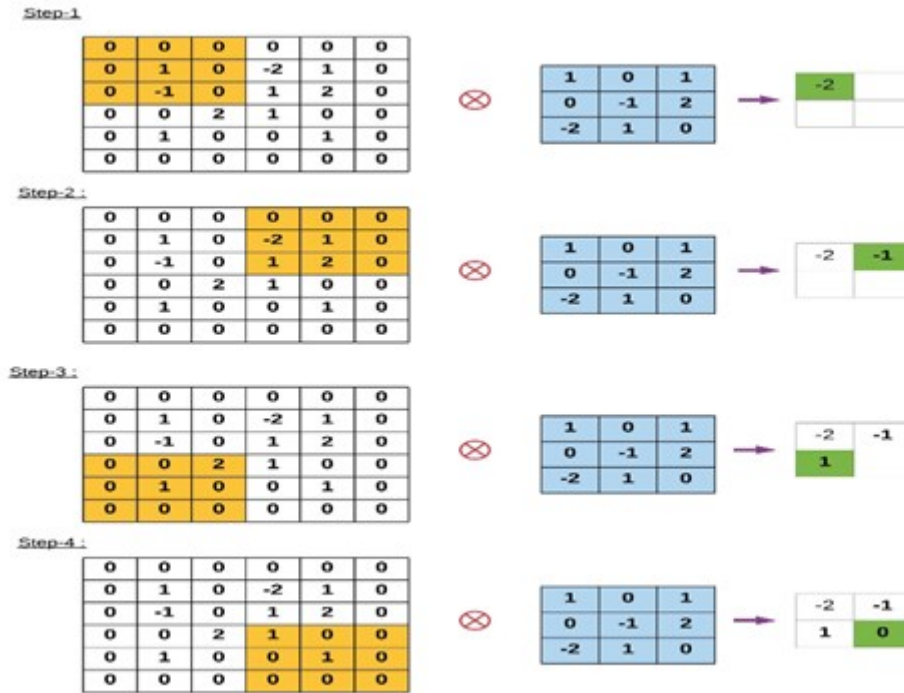
Στο παραπάνω παράδειγμα, εφαρμόσαμε την διαδικασία συνέλιξης χωρίς padding στην εικόνα εισόδου και με stride (το βήμα με το οποίο θα κινείται οριζόντια ή κάθετα το φίλτρο) της τιμής 1. Μπορούμε να αλλάξουμε την τιμή του stride αντί για 1 σε μια διαδικασία συνέλιξης. Σε αυτή την περίπτωση παρατηρούμε ότι όσο αυξάνεται το stride της διαδικασίας συνέλιξης, τόσο μειώνονται οι διαστάσεις του χάρτη χαρακτηριστικών. Το padding είναι εξίσου σημαντικό για να δίνει περισσότερη σημασία στις πληροφορίες που βρίσκονται στο περίγραμμα της εικόνας. Χωρίς αυτό, οι πληροφορίες στο περίγραμμα θα εξαφανιζόντουσαν πολύ γρήγορα. Η φόρμουλα για να βρεθεί το μέγεθος του χάρτη χαρακτηριστικών φαίνεται παρακάτω:

$$h' = \left\lfloor \frac{h - f + p}{s} + 1 \right\rfloor$$

$$w' = \left\lfloor \frac{w - f + p}{s} + 1 \right\rfloor$$

Εικόνα 13. Μαθηματικός τύπος για την εύρεση του χάρτη χαρακτηριστικών[10]

Όπου το h' δηλώνει το ύψος του χάρτη χαρακτηριστικών, το w' δηλώνει το πλάτος του χάρτη χαρακτηριστικών, το h δηλώνει το ύψος της εικόνας εισόδου, το w δηλώνει το πλάτος της εικόνας εισόδου, το f είναι το μέγεθος του φίλτρου, το p δηλώνει το padding του στρώματος συνέλιξης και το s δηλώνει το stride της διαδικασίας συνέλιξης.



Εικόνα 14. Διαδικασία συνέλιξης με μηδενικό padding[10]

Στην από πάνω εικόνα δίνουμε ένα παράδειγμα μιας διαδικασίας συνέλιξης με μηδενικό padding και τιμή stride 3. Κατά την διαδικασία συνέλιξης στην οποία πολλαπλασιάζουμε το 3x3 φίλτρο (εμφανίζεται με μπλε χρώμα) με την περιοχή που εμφανίζεται με κίτρινο χρώμα, παρατηρούμε ότι η αρχική εικόνα μεγέθους 4x4 μετατράπηκε σε εικόνα μεγέθους 6x6 λόγω μηδενικού padding καθώς και ο χάρτης χαρακτηριστικών ως έξοδος είναι μικρότερου μεγέθους από τον αρχικό λόγο stride.

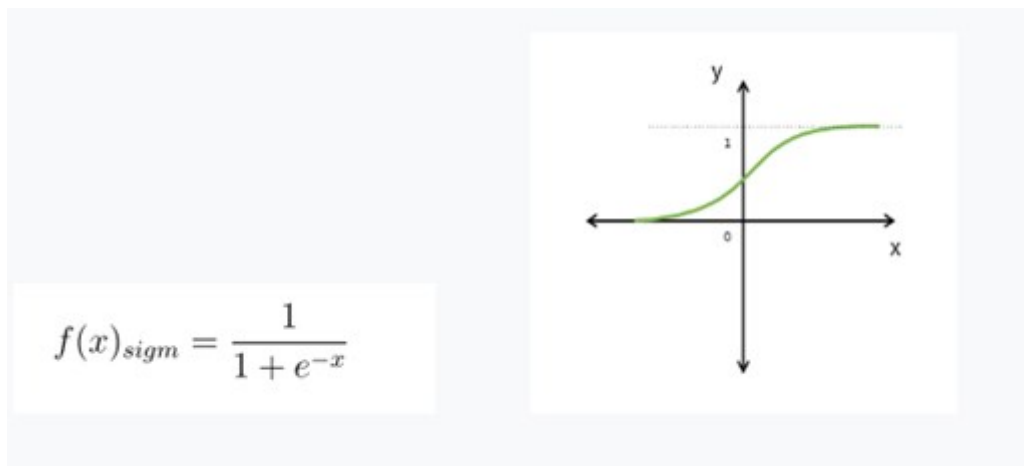
Τέλος τα βασικά πλεονεκτήματα των στρωμάτων συνέλιξης είναι:

- Αραιή συνδεσιμότητα: Σε ένα πλήρες συνδεδεμένο νευρωνικό δίκτυο, κάθε νευρώνας ενός στρώματος συνδέεται με κάθε νευρώνα του επόμενου στρώματος, αλλά στα συνελκτικά νευρωνικά δίκτυα λόγω μικρού αριθμού βαρών μεταξύ δύο στρωμάτων, ο αριθμός των συνδέσεων που χρειαζόμαστε είναι μικρός και η ποσότητα μνήμης για την αποθήκευση αυτών των βαρών είναι ελάχιστη, επομένως είναι αποδοτικά στη μνήμη.
- Μοίρασμα βαρών: Γνωρίζοντας ότι ένα συνελκτικό νευρωνικό δίκτυο έχει πολλά στρώματα, το μοίρασμα βαρών συμβαίνει όταν έχουμε ένα φίλτρο και περνάει μέσα από κάθε σημείο της εικόνας, αυτό δεν αλλάζει τα βάρη του. Αντί να μαθαίνει το νευρωνικό δίκτυο καινούργια βάρη για κάθε είσοδο της εικόνας σε κάθε διαφορετικό στρώμα, όταν βρίσκεται σε ένα στρώμα χρησιμοποιεί τα ίδια βάρη στο φίλτρο του για όλες τις εισόδους της εικόνας. Ως αποτέλεσμα, μειώνεται ο χρόνος εκπαίδευσης.

2.2.2 Συναρτήσεις Ενεργοποίησης

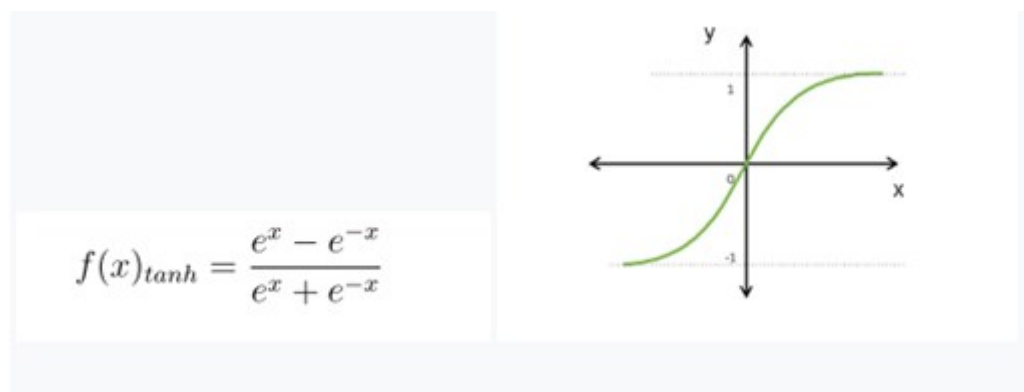
Το κύριο καθήκον κάθε συνάρτησης ενεργοποίησης σε οποιοδήποτε νευρωνικό δίκτυο είναι να αντιστοιχεί την είσοδο στην έξοδο, όπου η είσοδος λαμβάνεται υπολογίζοντας το άθροισμα της εισόδου του νευρώνα πολλαπλασιάζοντας το με ένα βάρος και προσθέτοντας και κάποιο σφάλμα (εάν υπάρχει). Με άλλα λόγια, η συνάρτηση ενεργοποίησης αποφασίζει εάν ένας νευρώνας θα ενεργοποιηθεί ή όχι για μια δεδομένη είσοδο που αντιστοιχεί σε μια έξοδο. Στην αρχιτεκτονική των συνελκτικών νευρωνικών δικτύων, μετά από κάθε στρώμα μάθησης (στρώματα που έχουν βάρη) για παράδειγμα, συνελκτικά στρώματα ή πλήρες συνδεδεμένα στρώματα χρησιμοποιείται μια συνάρτηση ενεργοποίησης. Παρακάτω δείχνουμε τι συναρτήσεις ενεργοποίησης που χρησιμοποιούνται πιο συχνά σε συνελκτικά νευρωνικά δίκτυα.

- Σιγμοειδής συνάρτηση: Η σιγμοειδής συνάρτηση ενεργοποίησης δέχεται πραγματικούς αριθμούς ως είσοδο και παράγει έξοδο στο εύρος $[0,1]$. Η μαθηματική αναπαράσταση της σιγμοειδής συνάρτησης είναι:



Εικόνα 15. Μαθηματική αναπαράσταση της σιγμοειδής συνάρτησης[10]

- Υπερβολική εφαπτομένη (tanh): Η υπερβολική εφαπτομένη συνάρτηση (tanh) δέχεται πραγματικούς αριθμούς και παράγει έξοδο στο εύρος $[-1,1]$. Η μαθηματική αναπαράσταση της υπερβολικής εφαπτομένης είναι:



Εικόνα 16. Μαθηματική αναπαράσταση της σιγμοειδής συνάρτησης[10]

- ReLU: Η συνάρτηση Rectifier Linear Unit (ReLU) είναι η πιο χρησιμοποιημένη συνάρτηση ενεργοποίησης στα συνελκτικά νευρωνικά δίκτυα. Μετατρέπει όλες τις τιμές εισόδου σε θετικούς αριθμούς. Τα πλεονεκτήματα της είναι ότι χρειάζεται ελάχιστη υπολογιστική ισχύ σε σύγκριση με άλλες συναρτήσεις. Η μαθηματική της αναπαράσταση της συνάρτησης ReLU είναι:

$$f(x)_{ReLU} = \max(0, x)$$

Εικόνα 17. Μαθηματική εξίσωση συνάρτησης ReLU[10]

- Leaky ReLU: Αντίθετα με την κανονική συνάρτηση ReLU, η συνάρτηση ενεργοποίησης Leaky ReLU δεν αγνοεί τελείως τις αρνητικές τιμές, αντί αυτού μικραίνει την κλίμακα τους. Η μαθηματική της αναπαράσταση είναι:

$$f(x)_{LeakyReLU} = \begin{cases} x, & \text{if } x > 0 \\ mx, & x \leq 0 \end{cases}$$

Εικόνα 18. Μαθηματική εξίσωση συνάρτησης Leaky ReLU[10]

- Noisy ReLU: Η συνάρτηση ενεργοποίησης Noise ReLU χρησιμοποιείται στην κατανομή κατά Gauss (Gaussian distribution) για να δώσει έναν θόρυβο στην κανονική ReLU. Η μαθηματική της αναπαράσταση είναι:

$$f(x)_{NoisyReLU} = \max(x + Y), \text{ with } Y \sim N(0, \sigma(x))$$

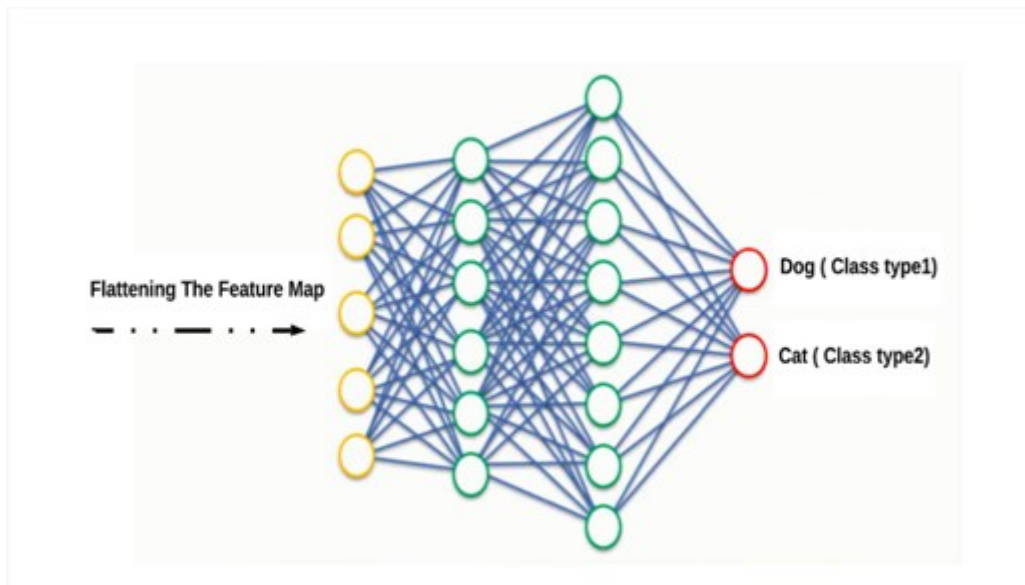
Εικόνα 19. Μαθηματική εξίσωση Noisy ReLU[10]



Εικόνα 20. Γραφική αναπαράσταση Noisy ReLU[10]

2.2.3 Πλήρες συνδεδεμένο στρώμα (Fully connected layer FC)

Συνήθως το τελευταίο κομμάτι κάθε συνελκτικού νευρωνικού δικτύου αποτελείται από ένα πλήρες συνδεδεμένο στρώμα όπου, κάθε νευρώνας μέσα στο στρώμα είναι συνδεδεμένος με κάθε νευρώνα του προηγούμενου στρώματος. Τα πλήρες συνδεδεμένα στρώματα είναι ένα είδος νευρωνικού δικτύου μproστινής τροφοδότησης και ακολουθούν το παραδοσιακό δίκτυο Perceptron απλά με περισσότερα από ένα στρώμα δηλαδή ένα MLP (Multi layer perceptron). Δέχονται είσοδο από το τελικό στρώμα συνέλιξης ή υποδειγματοληψίας η οποία, είναι σε μορφή ενός χάρτη χαρακτηριστικών ο οποίος μετατρέπεται σε ένα διάνυσμα που με την σειρά του τροφοδοτείται στο πλήρες συνδεδεμένο στρώμα για να παραχθεί το τελικό αποτέλεσμα του συνελκτικού νευρωνικού δικτύου όπως φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 21. Πλήρη συνδεδεμένα στρώματα[10]

2.2.4 Συνάρτηση απώλειας (Loss Function)

Όπως προαναφέραμε το τελευταίο στρώμα ενός συνελκτικού νευρωνικού δικτύου είναι ένα στρώμα εξόδου στο οποίο, γίνεται η τελική ταξινόμηση. Στα περισσότερα στρώματα εξόδου, υπολογίζουμε και το σφάλμα πρόβλεψης που δημιουργήθηκε από το συνελκτικό δίκτυο όταν του δώσαμε τα δεδομένα εκπαίδευσης χρησιμοποιώντας μια συνάρτηση απώλειας. Το σφάλμα πρόβλεψης μας δείχνει πόσο «έξω» έχει πέσει το μοντέλο στην πρόβλεψη του από την πραγματική έξοδο που θέλαμε, και στην συνέχεια χρησιμοποιείται με σκοπό την βελτιστοποίηση του μοντέλου κατά την διάρκεια της διαδικασίας εκπαίδευσης του νευρωνικού δικτύου. Η συνάρτηση απώλειας ουσιαστικά χρησιμοποιεί 2 παραμέτρους για να υπολογίσει το σφάλμα. Η πρώτη παράμετρος είναι η τιμή της προβλεπομένης εξόδου (η πρόβλεψη του δικτύου δηλαδή) και η δεύτερη παράμετρος είναι η πραγματική έξοδος (δηλαδή η ετικέτα εξόδου). Υπάρχουν διαφορετικές συναρτήσεις απώλειας ανάλογα το είδος του προβλήματος, παρακάτω αναφέρουμε τις πιο γνωστές.[10]

- Cross-Entropy ή Soft-Max Loss Function: Η συνάρτηση απώλειας Cross-entropy, γνωστή και ως λογαριθμική συνάρτηση απώλειας, χρησιμοποιείται για τον υπολογισμό της απόδοσης ενός συνελκτικού νευρωνικού δικτύου του οποίου, η έξοδος είναι μια πιθανότητα $p \in [0, 1]$. Χρησιμοποιείται αντί για την συνάρτηση μέσης τετραγωνικής απόκλισης (squared error loss function) στα προβλήματα ταξινόμησης πολλών κλάσεων. Έχει μια ενεργοποίηση softmax στο στρώμα εξόδου προκειμένου να παράγει την έξοδο μέσω μιας κατανομής πιθανοτήτων, $p, y \in \mathbb{R}^N$, όπου το p είναι η πιθανότητα για κάθε διαφορετική έξοδο και το y δηλώνει την επιθυμητή έξοδο. Η πιθανότητα για κάθε κλάση εξόδου μπορεί να βρεθεί από την παρακάτω εξίσωση όπου N είναι ο αριθμός

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}}$$

Εικόνα 22. Εξίσωση για να βρεθεί η πιθανότητα για κάθε έξοδο[10]

των νευρώνων στο στρώμα εξόδου και το e^{a_i} δηλώνει κάθε έξοδο του προηγούμενου στρώματος στο δίκτυο πριν γίνει η μετατροπή του σε διάνυσμα. Γενικά η συνάρτηση cross-entropy έχει την παρακάτω μορφή όπου $i \in [1, N]$

$$H(p, y) = - \sum_i y_i \log(p_i)$$

Εικόνα 23. Μαθηματική εξίσωση της συνάρτησης Cross Entropy[10]

- Euclidean Loss Function: Η ευκλείδεια συνάρτηση απώλειας, γνωστή και ως συνάρτηση μέσης τετραγωνικής απόκλισης, χρησιμοποιείται ευρέως σε προβλήματα παλινδρόμησης. Το μέσω τετραγωνικό σφάλμα ανάμεσα στην προβλεπόμενη έξοδο $p \in \mathbb{R}^N$ και στην πραγματική έξοδο $y \in \mathbb{R}^N$ σε κάθε νευρώνα στο στρώμα εξόδου ενός συνελκτικού νευρωνικού δικτύου ορίζεται ως $H(p,y) = (p - y)^2$. Οπότε, εάν υπάρχουν N νευρώνες σε ένα στρώμα εξόδου τότε, η εκτιμώμενη ευκλείδεια απώλεια ορίζεται ως:

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^N (p_i - y_i)^2$$

Εικόνα 24. Μαθηματική εξίσωση της Ευκλείδειας συνάρτησης[10]

- Hinge Loss Function: Η συνάρτηση απώλειας Hinge χρησιμοποιείται ευρέως σε προβλήματα δυαδικής ταξινόμησης. Γίνεται χρήση της, κυρίως σε αλγορίθμους SVMs (Support Vector Machines) και προσπαθούν να μεγιστοποιήσουν το περιθώριο ανάμεσα σε 2 επιθυμητές κλάσεις. Ορίζεται ως:

$$H(p, y) = \sum_{i=1}^N \max(0, m - (2y_i - 1)p_i)$$

Εικόνα 25. Μαθηματική εξίσωση της συνάρτησης Hinge[10]

2.2.5 Διαδικασία εκπαίδευσης Συνελκτικών νευρωνικών δικτύων

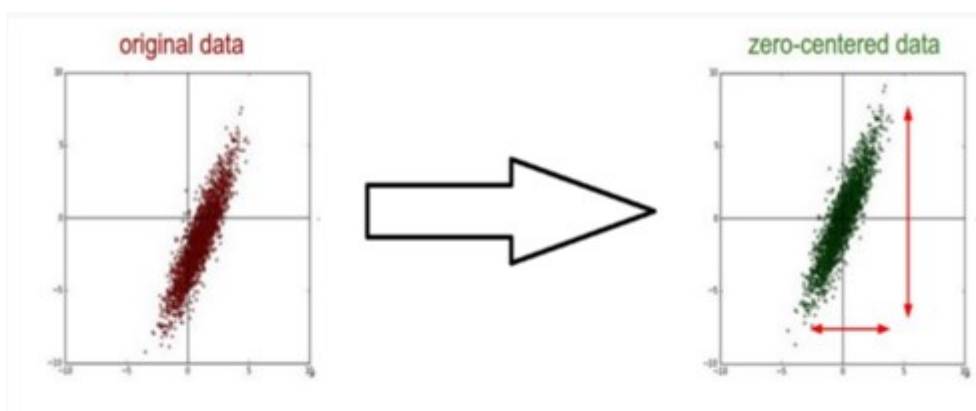
Προηγουμένως περιγράψαμε τις βασικές έννοιες και το θεωρητικό υπόβαθρο πίσω από την λειτουργία των συνελκτικών νευρωνικών δικτύων, καθώς και τα πιο σημαντικά χαρακτηριστικά στην αρχιτεκτονική τους. Παρακάτω θα αναφέρουμε την διαδικασία εκπαίδευσης και μάθησης ενός συνελκτικού μοντέλου με ορισμένες μεθόδους προκειμένου να μειωθεί ο απαραίτητος χρόνος εκπαίδευσης και να βελτιωθεί η ακρίβεια του μοντέλου μας. Η διαδικασία εκπαίδευσης αποτελείται από τα παρακάτω βήματα:

- Προεπεξεργασία και αύξηση δεδομένων
- Προετοιμασία βαρών
- Regularization
- Προεπεξεργασία και αύξηση δεδομένων Η προεπεξεργασία δεδομένων αναφέρεται σε διαφορετικούς μετασχηματισμούς που γίνονται στο ακατέργαστο σύνολο δεδομένων που έχουμε (περιλαμβάνει τα δεδομένα εκπαίδευσης και ελέγχου), προκειμένου να γίνουν τα δεδομένα πιο «καθαρά», πιο λειτουργικά, πιο ομοιόμορφα και γενικά πιο εύκολα στην μάθηση. Η προεπεξεργασία γίνεται πριν περάσουμε τα δεδομένα από το συνελκτικό νευρωνικό δίκτυο. Γνωρίζουμε ότι, η απόδοση του συνελκτικού νευρωνικού δικτύου είναι ανάλογη με τον όγκο των δεδομένων που χρησιμοποιούνται για παράδειγμα, μια μεγάλη βάση δεδομένων που έχει περάσει από καλή προεπεξεργασία θα αυξήσει την ακρίβεια του μοντέλου ενώ, μια κακή προεπεξεργασία μπορεί επίσης να μειώσει την απόδοση του μοντέλου. Οι πιο συχνές τεχνικές προεπεξεργασίας αναφέρονται παρακάτω.

* Μέση-αφαίρεση/Μηδενικό κεντράρισμα (Mean-subtraction/Zero centering): Είναι ένας γραμμικός μετασχηματισμός που μετατοπίζει τα δεδομένα έτσι ώστε να επικεντρώνονται στην αρχή των αξόνων. Συνήθως αυτό γίνεται αφαιρώντας τον μέσο όρο από κάθε δεδομένο μεμονωμένα, προκειμένου να βρεθεί το "φυσικό σύστημα συντεταγμένων" για τα δεδομένα. Η διαδικασία αναπαριστάτε μαθηματικά παρακάτω:

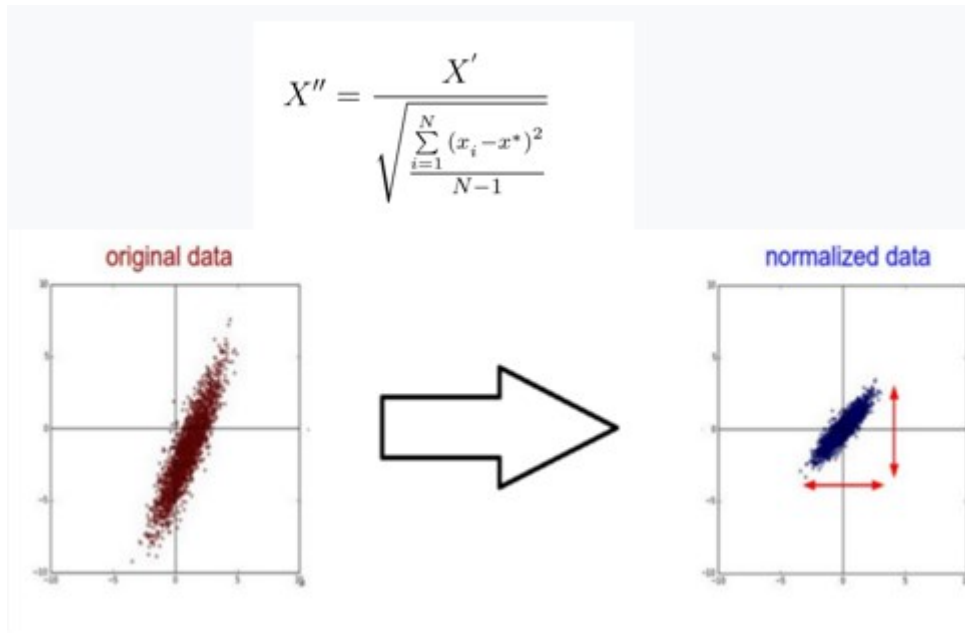
$$X' = X - \chi^* \quad \text{και} \quad \chi^* = \frac{1}{N} \sum_{i=1}^N \chi^i$$

όπου N δηλώνει το σύνολο των δεδομένων εκπαίδευσης.



Εικόνα 26. Η διαδικασία μέσης αφαίρεσης[10]

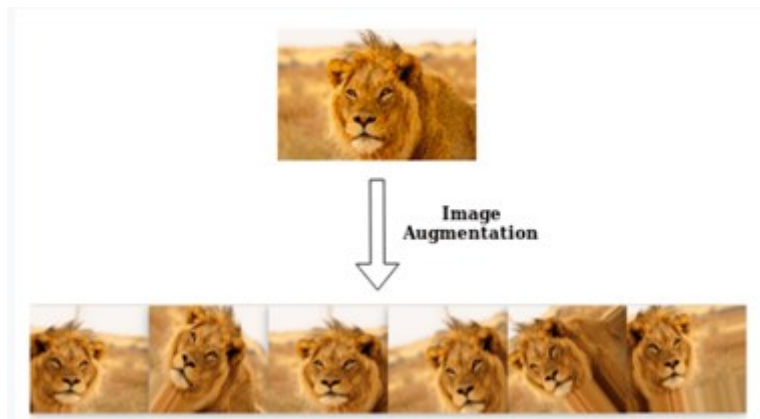
- * Κανονικοποίηση (Normalization): Είναι μια διαδικασία διαίρεσης η οποία αλλάζει το εύρος τιμών έντασης των pixel της εικόνας στην βάση δεδομένων μας. Με την κανονικοποίηση καταφέρνουμε να ομαδοποιήσουμε τα δεδομένα μας στο ίδιο εύρος τιμών. Ο λόγος που το κάνουμε αυτό είναι επειδή κατά την διαδικασία εκπαίδευσης του δικτύου μας, θα πολλαπλασιάσουμε βάρη και θα προσθέσουμε σφάλματα στις αρχικές εισόδους, προκειμένου να προκαλέσουμε συναρτήσεις ενεργοποίησης που στη συνέχεια θα αντιστρέψουμε με την μέθοδο στοχαστικής βαθμωτής κατάβασης για να εκπαιδεύσουμε το μοντέλο. Θέλουμε κατά την διάρκεια αυτής της διαδικασίας τα δεδομένα μας να έχουν το ίδιο εύρος, ώστε οι κλίσεις κατά την βαθμωτή κατάβαση να μην βγαίνουν εκτός ελέγχου. Η μαθηματική αναπαράσταση είναι: [10]



Εικόνα 27. Μαθηματική εξίσωση της κανονικοποίησης [10]

- * Αύξηση δεδομένων: Η τεχνική ονομάζεται image augmentation και χρησιμοποιείται για να αυξήσουμε το μέγεθος του συνόλου των δεδομένων μας. Εφαρμόζουμε διαφορετικές λειτουργίες σε δείγματα από τα δεδομένα εκπαίδευσης και τα μεταμορφώνουμε σε καινούργια δεδομένα τα οποία επίσης θα χρησιμοποιηθούν στην διαδικασία εκπαίδευσης. Η αύξηση δεδομένων είναι σημαντική επειδή σε πολλές περιπτώσεις έχουμε πολύ περιορισμένο αριθμό δεδομένων εκπαίδευσης και όπως προαναφέραμε, περισσότερα δεδομένα μπορούν να δημιουργήσουν ένα δυνατότερο και ακριβές μοντέλο.

Κάποιες από τις τεχνικές αύξησης δεδομένων είναι η περικοπή, η περιστροφή, η αναστροφή, η κλιμάκωση η προσθήκη θορύβου κ.α. Μπορούμε να τις εφαρμόσουμε ξεχωριστά ή σε συνδυασμό, για να δημιουργήσουμε νέες εικόνες από ένα μόνο δείγμα των δεδομένων. Ένας άλλος λόγος που χρησιμοποιούμε την αύξηση δεδομένων είναι για να αποφύγουμε προβλήματα υπερπροσαρμογής. [11]

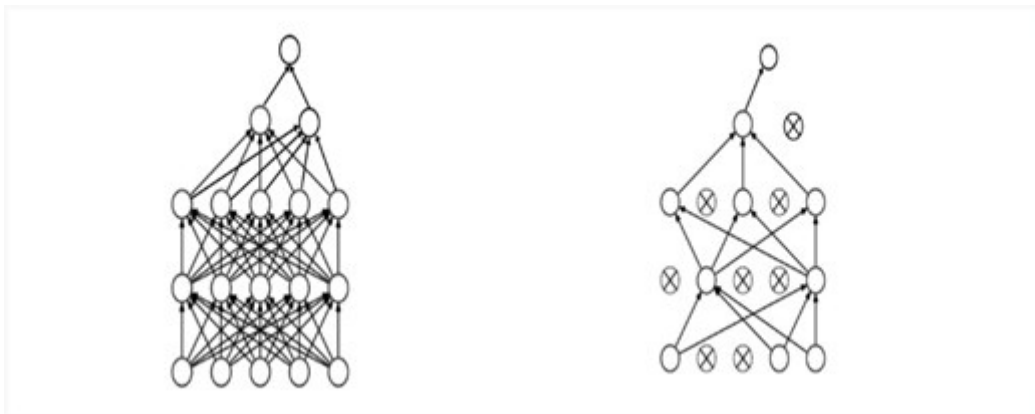


Εικόνα 28. Αύξηση δεδομένων[11]

- Προετοιμασία βαρών: Ένα βαθύ συνελκτικό νευρωνικό δίκτυο αποτελείται από εκατομμύρια ή δισεκατομμύρια βάρη. Φροντίζουμε να προετοιμάσουμε αυτά τα βάρη στην αρχή της διαδικασίας εκπαίδευσης, δεδομένου ότι τα βάρη αυτά καθορίζουν το πόσο γρήγορα το συνελκτικό δίκτυο θα «μάθει» και πόσο ακριβές θα είναι. Στην συνέχεια θα αναφέρουμε τις πιο δημοφιλείς τεχνικές προετοιμασίας βαρών στα συνελκτικά μοντέλα: Ο πιο εύκολος τρόπος θα ήταν να δώσουμε σε όλα τα βάρη την τιμή 0. Ωστόσο, αυτό θα ήταν ένα λάθος, αφού εάν όλα τα βάρη άρχιζαν με τον αριθμό 0 η έξοδοι τους καθώς και οι κλίσεις (από την στοχαστική βαθμωτή κατάβαση) που θα υπολογιζόντουσαν από κάθε νευρώνα στο δίκτυο θα ήταν οι ίδιες. Ως αποτέλεσμα, θα ενημερωνόντουσαν με την ίδια τιμή, δεν θα υπήρχε καμία διαφορά μεταξύ των νευρώνων και το δίκτυο δεν θα μάθαινε κανένα χρήσιμο χαρακτηριστικό. Για να το αποφύγουμε αυτό αποφεύγουμε να δίνουμε την τιμή μηδέν και χρησιμοποιούμε μια από τις τεχνικές που ακολουθούν:

- * Τυχαία προετοιμασία παραμέτρων: Όπως αναφέρει και το όνομα, δίνουμε στο κάθε βάρη μια τυχαία τιμή χρησιμοποιώντας πίνακες, των οποίων τα περιεχόμενα λαμβάνονται από κάποια κατανομή (Gaussian, Uniform, Orthogonal) με μικρή απόκλιση από το 0 (0.1, 0.01) και μηδενικό μέσο όρο. Το κύριο πρόβλημα αυτών των μεθόδων είναι ότι ενδέχεται να οδηγήσουν σε προβλήματα όπως η εξασθένιση διορθώσεων (vanishing gradients).
- * Προετοιμασία Xavier (Xavier Initialization): Η τεχνική χρησιμοποιήθηκε από τον Xavier Glorot και τον Yoshua Bengio το 2010, ο στόχος είναι να αρχικοποιηθούν τα βάρη έτσι ώστε η διακύμανση των συναρτήσεων ενεργοποίησης να είναι η ίδια σε κάθε στρώμα. Αυτή η σταθερή διακύμανση βοηθά στην αποφυγή των προβλημάτων της έκρηξης ή της εξαφάνισης διορθώσεων. Αποδεικνύεται ότι η τεχνική είναι πολύ χρήσιμη όταν χρησιμοποιούμε συναρτήσεις ενεργοποίησης tanh ή σιγμοειδείς, αλλά δεν αποδίδει καλά όταν χρησιμοποιούμε συναρτήσεις ReLU. Αργότερα, η τεχνική του Kaiming He η οποία, βασιζόταν στην ίδια ιδέα λειτουργούσε και στις συναρτήσεις ενεργοποίησης ReLU.

- * Το πρόβλημα των εξασθενουσών διορθώσεων (Vanishing Gradient Problem): Κατά την διάρκεια της εκπαίδευσης ενός νευρωνικού δικτύου, έχουμε επισημάνει ότι τα βάρη ενημερώνονται μέσω της ανάστροφης διάδοσης (Back-propagation) συγκρίνοντας την τρέχουσα και την επιθυμητή τιμή τους, το σφάλμα δηλαδή, και μέσω της στοχαστικής βαθμωτής κατάβασης αφού τα δεδομένα περάσουν από όλους τους νευρώνες, τα βάρη ενημερώνονται προς τα πίσω. Πρακτικά αυτό γίνεται υπολογίζοντας την παράγωγο τον σφαλμάτων και λαμβάνοντας μια κλίση. Ουσιαστικά όμως, όταν έχουμε ένα νευρωνικό δίκτυο με πολλά χιλιάδες στρώματα, όσο πιο πίσω πηγαίνουμε για να ενημερωθούν τα αρχικά βάρη, λόγω της φύσης της παραγώγου, τόσο περισσότερο θα μικραίνει η τιμή του σφάλματος και ως αποτέλεσμα τα αρχικά βάρη θα λαμβάνουν πολύ μικρές τιμές (πολύ μικρή κλίση) και θα ενημερώνονται ελάχιστα. Έτσι θα μαθαίνουν πολύ αργά και αναποτελεσματικά. Λύση αυτού του προβλήματος αποτελεί η χρήση συναρτήσεων ενεργοποίησης της οικογένειας ReLU όπως η leaky ReLU, τα Residual νευρωνικά δίκτυα τα οποία αγνοούν κάποια βάρη και συνδέουν κατευθείαν την είσοδο με την συνάρτηση ενεργοποίησης και με αλγορίθμους κανονικοποίησης στα δεδομένα εισόδου.
- Regularization Η βασική πρόκληση αλγορίθμων βαθιάς μάθησης, όπως και κάθε αλγόριθμος μηχανικής μάθησης, είναι η προσαρμογή σε νέα δεδομένα δηλαδή η γενίκευση. Όπως προαναφέραμε το πρόβλημα που δεν επιτρέπει την γενίκευση ονομάζεται υπερπροσαρμογή. Το Regularization βοηθά το δίκτυο να αποφύγει την υπερπροσαρμογή χρησιμοποιώντας ιδέες που θα δούμε παρακάτω:
 - Dropout Η τεχνική Dropout είναι μια από τις πιο χρησιμοποιημένες προσεγγίσεις για Regularization. Εδώ επιλέγονται τυχαία κάποιοι νευρώνες και σταματάνε την λειτουργία τους κατά την διάρκεια κάθε εποχής εκπαίδευσης. Σταματώντας την λειτουργία τυχαίων νευρώνων προσπαθούμε να μοιράσουμε ίση πληροφορία σε όλους τους υπόλοιπους νευρώνες και ως αποτέλεσμα αναγκάζουμε το μοντέλο να μάθει πολλά διαφορετικά χαρακτηριστικά από τα δεδομένα του. Στην παρακάτω εικόνα φαίνονται η διαφοράς ενός κανονικού δικτύου και ενός που έχει εφαρμοστεί Dropout.[10]



Εικόνα 29. Η τεχνική σύνδεσης Dropout[10]

2.2.6 Συνελκτικά νευρωνικά δίκτυα για ταξινόμηση εικόνων

Η αναγνώριση εικόνων αποτελεί ένα από τα πλέον δημοφιλή και εκτενώς μελετημένα προβλήματα τεχνητής νοημοσύνης. Μια από τις πρώτες αρχιτεκτονικές συνελκτικών νευρωνικών δικτύων χρησιμοποιούνταν για την αναγνώριση χειρόγραφων αριθμών σε τραπεζικές επιταγές. Μετά το 1998, υπήρξε μια περίοδος κατά την οποία δεν έγιναν πολλές πρόοδοι λόγω του περιορισμού των υλικών εκείνης της εποχής. Το 2012 όμως ο Alex Krizhevsky παρουσίασε μια αρχιτεκτονική ονόματι AlexNet η οποία, κατέκτησε την πρώτη θέση στον τότε διαγωνισμό ImageNet (που περιείχε 14.000.000 εικόνες προς αναγνώριση) και θεωρείται από πολλούς ως η μεγαλύτερη πρόοδος στα συνελκτικά νευρωνικά δίκτυα. Μετά το AlexNet υπήρξαν και άλλες αρχιτεκτονικές σχεδιασμένες για την ταξινόμηση εικόνων όπως τα δίκτυα ZFNet και VGG16 οι οποίες, χρησιμοποιήθηκαν αργότερα ως βασική πυλώνας στις αρχιτεκτονικές με σκοπό την ανίχνευση αντικειμένων.

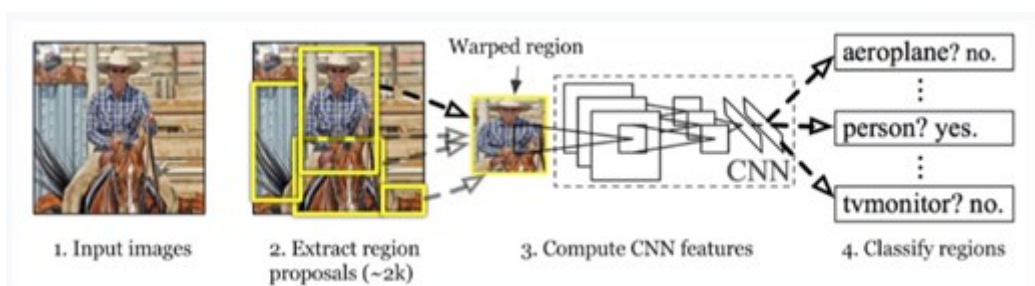
2.2.7 Συνελκτικά νευρωνικά δίκτυα και ανίχνευση αντικειμένων

Όσον αφορά την ανίχνευση αντικειμένων σε εικόνες, έχει γίνει τεράστια δουλειά και σε πολλές αρχιτεκτονικές συνελκτικών νευρωνικών δικτύων αλλά και σε διαφορετικούς τρόπους χρήσης τους. Το παρακάτω είναι ένα χρονοδιάγραμμα το οποίο απεικονίζει την εξέλιξη στην ανίχνευση αντικειμένων.



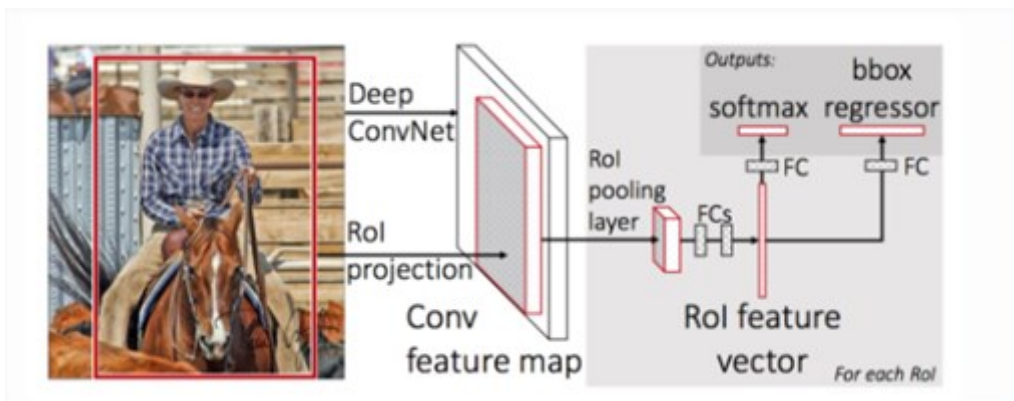
Εικόνα 30. Χρονοδιάγραμμα διαφορετικών μοντέλων

- R-CNN (Region-based Convolutional Neural Networks) Τα R-CNN (Συνελκτικά νευρωνικά δίκτυα βασισμένα σε περιοχές) αποτελούν μια αρχιτεκτονική για ανίχνευση αντικειμένων η οποία, βασίζεται στον συνδυασμό απλών συνελκτικών δικτύων με τον αλγόριθμο Selective Search ο οποίος προτείνει κομμάτια τις εικόνες που νομίζει ότι έχουν κάποιο αντικείμενο. Συγκεκριμένα, η αρχική μας εικόνα χωρίζεται σε έναν αριθμό υποψήφιων μικρότερων εικόνων οι οποίες έχουν οριοθετημένα αντικείμενα μέσα τους. Στην συνέχεια, οι υποψήφιες εικόνες περνάνε μέσα από το συνελκτικό δίκτυο το οποίο κάνει μια απλή ταξινόμηση. Αυτό αποτελεί και ένα μειονέκτημα της αρχιτεκτονικής αυτής αφού είναι υπολογιστικά ακριβή.[12]



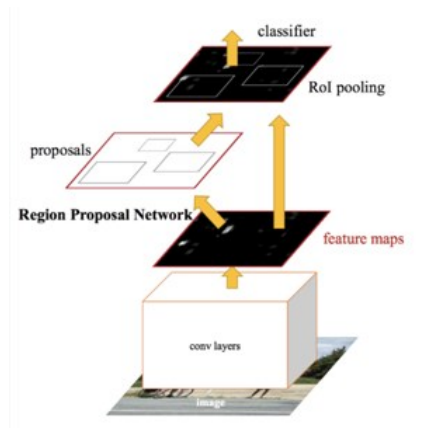
Εικόνα 31. Το μοντέλο R-CNN[12]

- Fast R-CNN Προκειμένου να αυξηθεί η ταχύτητα των R-CNN, οι ερευνητές παρουσίασαν μια μέθοδο στην οποία ένωσαν τρία ανεξάρτητα μοντέλα σε ένα κοινό εκπαιδευμένο πλαίσιο. Συγκεκριμένα, πρώτα πέρναγαν την αρχική εικόνα από ένα συνελκτικό νευρωνικό δίκτυο για να πάρουν έναν χάρτη χαρακτηριστικών και στην συνέχεια από αυτόν τον χάρτη λαμβάνανε διανύσματα. Αυτά τα διανύσματα, περνούσαν μέσα από ειδικά στρώματα και στο τέλος συνδέονταν με μια ενεργοποίηση Softmax με σκοπό να γίνει η ταξινόμηση τους και με ένα στρώμα παλινδρόμησης για να βρεθεί το πλαίσιο στο οποίο βρισκόντουσαν τα αντικείμενα. Ως εκ τούτου, το μοίρασμα των διαδικασιών υπολογισμού αύξησε την ταχύτητα του R-CNN.[13]



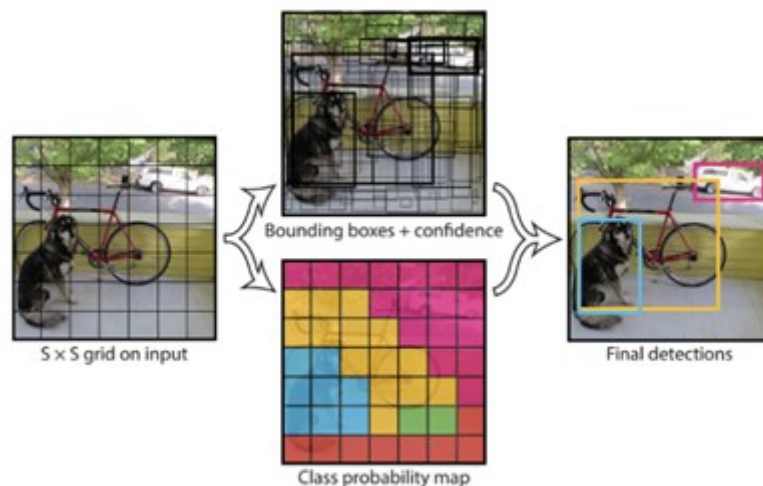
Εικόνα 32. Το μοντέλο Fast R-CNN[13]

- Faster R-CNN Και οι δύο παραπάνω μέθοδοι (R-CNN & Fast R-CNN) χρησιμοποιούν τον αλγόριθμο Selective Search για να προτείνουν μικρότερες περιοχές της αρχικής εικόνας. Ωστόσο η μέθοδος Selective Search είναι μια αργή και χρονοβόρα διαδικασία που επηρεάζει την απόδοση του μοντέλου. Επομένως, οι ερευνητές κατέληξαν σε έναν αλγόριθμο ανίχνευσης αντικειμένων που προσπερνάει τον αλγόριθμο Selective Search και επιτρέπει στο νευρωνικό δίκτυο να μάθει τις προτάσεις περιοχής. Παρόμοια με το Fast R-CNN, η εικόνα παρέχεται ως είσοδος σε ένα συνελκτικό νευρωνικό δίκτυο που παρέχει έναν χάρτη χαρακτηριστικών. Αντί να χρησιμοποιήσετε Selective Search στο χάρτη χαρακτηριστικών για τον προσδιορισμό των προτάσεων περιοχής, χρησιμοποιείται ξεχωριστό δίκτυο για την πρόβλεψη των προτάσεων περιοχής. Οι προτεινόμενες μικρότερες περιοχές της εικόνας στη συνέχεια αναδιαμορφώνονται χρησιμοποιώντας ένα ειδικό στρώμα RoI(Region of Interest Pooling), το οποίο στη συνέχεια χρησιμοποιείται για την ταξινόμηση της εικόνας εντός της προτεινόμενης περιοχής και την πρόβλεψη για τα πλαίσια του αντικειμένου.[14]



Εικόνα 33. Μοντέλο Faster R-CNN[14]

- You Only Look Once (YOLO) Το YOLO ή You Only Look Once είναι ένας αλγόριθμος ανίχνευσης αντικειμένων πολύ διαφορετικός από τους αλγόριθμους που βασίζονται στο Selective Search που αναφέρονται παραπάνω. Στο YOLO ένα ενιαίο συνεκτικό δίκτυο προβλέπει τα πλαίσια των αντικειμένων και τις πιθανότητες κλάσης για αυτά τα πλαίσια. Ο τρόπος που λειτουργεί το YOLO είναι ότι παίρνουμε μια εικόνα και τη χωρίζουμε σε ένα πλέγμα $S \times S$, μέσα σε κάθε ένα από τα πλέγματα παίρνουμε m πλαίσια. Για κάθε ένα από τα πλαίσια οριοθέτησης, το δίκτυο εξάγει τιμές πιθανότητας για το πλαίσιο οριοθέτησης. Τα πλαίσια οριοθέτησης με πιθανότητα κλάσης πάνω από μια τιμή κατωφλίου επιλέγονται και χρησιμοποιούνται για τον εντοπισμό του αντικειμένου εντός της εικόνας. Το YOLO είναι πολύ γρηγορότερο (45 καρέ ανά δευτερόλεπτο) από άλλους αλγόριθμους ανίχνευσης αντικειμένων. Ο περιορισμός του αλγορίθμου YOLO είναι ότι δυσκολεύεται με μικρά αντικείμενα μέσα στην εικόνα, για παράδειγμα μπορεί να έχει δυσκολίες στην ανίχνευση ενός κοπαδιού πουλιών.[15]



Εικόνα 34. Το μοντέλο Yolo[15]

2.2.8 Κριτήρια επίδοσης για ανιχνευτές αντικειμένων

Προκειμένου να μπορούμε να αξιολογήσουμε την επίδοση ενός ανιχνευτή αντικειμένων πρέπει πρώτα να προσδιορίσουμε εάν μια ανίχνευση είναι έγκυρη ή όχι. Για αυτό τον λόγο χρησιμοποιούμε δυο δημοφιλή κριτήρια, που ονομάζονται:

Ευστοχία(Precision):

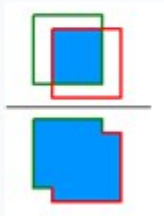
$$\text{Precision} = \frac{TP}{\text{POSITIVE}} = \frac{TP}{TP+FP}$$

Ανάκλαση(Recall):

$$\text{Recall} = \frac{TP}{\text{κλάση 1}} = \frac{TP}{TP+FN}$$

όπου TP, FP και FN είναι ο αριθμός των true positive (πραγματικά θετικά δηλαδή μια έγκυρη ανίχνευση), false positive (εσφαλμένα θετικά δηλαδή μια εσφαλμένη ανίχνευση) και false negatives (εσφαλμένα αρνητικά δηλαδή το μοντέλο δεν βρήκε την τιμή ground-truth η οποία εξηγείτε παρακάτω).

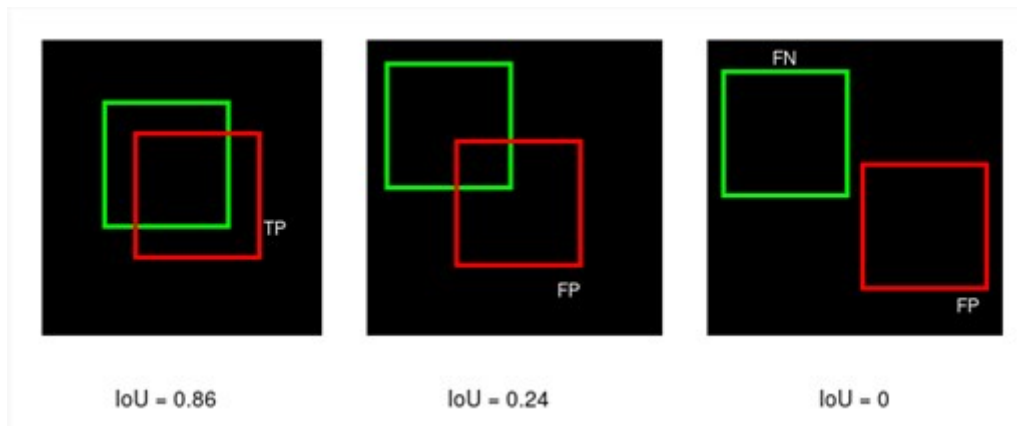
Στα πλαίσια της εγκυρότητας μιας ανίχνευσης απαιτείται και μια επιπλέον μέτρηση που ονομάζεται Intersection over Union. Αυτή αξιολογεί την σχέση μεταξύ των προβλεπόμενων πλαισίων οριοθέτησης και των ground-truth τιμών τους, δηλαδή των πραγματικών πλαισίων οριοθέτησης του εκάστοτε αντικειμένου που δώσαμε εμείς στον αλγόριθμο για εκπαίδευση. Υπολογίζεται διαγραμματικά ως:

$$\text{IoU} = \frac{\text{περιοχή επικάλυψης}}{\text{περιοχή ένωσης}} =$$


Εικόνα 35. Υπολογισμός IoU[16]

Η τιμή της μέτρησης IoU κυμαίνεται από 0 μέχρι 1, με το 0 να δηλώνει καμία επικάλυψη και το 1 να δηλώνει τέλεια επικάλυψη. Γενικότερα με την μέτρηση IoU ορίζουμε μια τιμή κατωφλιού (α) που χρησιμοποιείται για να διακρίνουμε μια έγκυρη ανίχνευση. Μπορούμε έτσι να επαναπροσδιορίσουμε τους ορισμούς των όρων TP σε ανίχνευση για την οποία το $\text{IoU} \geq \alpha$ και FP σε ανίχνευση όπου $\text{IoU} \leq \alpha$. Το FN είναι μια ground-truth τιμή που δεν μπόρεσε να πετύχει το μοντέλο.

Για παράδειγμα, με μια τιμή $\text{IoU} \alpha = 0.5$ (50%), ορίζουμε τις τιμές TP, FP, FN, στο παρακάτω διάγραμμα:



Εικόνα 36. Παραδειγμά με διαφορετικά IoU[16]

Ανακεφαλαιώνοντας, το κριτήριο Precision είναι η ικανότητα ενός ταξινομητή να αναγνωρίζει τα σχετικά αντικείμενα δηλαδή αυτά που ψάχνουμε να βρούμε. Από την άλλη πλευρά, το κριτήριο Recall είναι η ικανότητα του ταξινομητή να βρίσκει το ποσοστό των σχετικών αντικειμένων που βρίσκονται μέσα στις τιμές ground-truth. Ένα καλό μοντέλο μπορεί να εντοπίσει το ποσοστό των αντικειμένων στις τιμές ground-truth, ενώ παράλληλα βρίσκει τα σχετικά αντικείμενα που μας ενδιαφέρουν. Ένα τέλειο μοντέλο έχει τιμές FN = 0 (recall = 1) και FP = 0 (precision = 1). Το πρώτο μοντέλο είναι συνήθως ο στόχος μας, το δεύτερο μοντέλο είναι πρακτικά ανέφικτο.

- Καμπύλη Precision-Recall (PR Curve)

Η καμπύλη ευστοχίας-ανάκλησης (PR) είναι μια γραφική παράσταση της ευστοχίας ως συνάρτηση της ανάκλησης. Δείχνει την αντιστάθμιση μεταξύ των δύο μετρήσεων για διαφορετικές τιμές IoU για τις ανιχνεύσεις μοντέλων. Εάν τα FP είναι χαμηλά, η ευστοχία είναι υψηλή, αλλά μπορεί να παραλειφθούν περισσότερες ανιχνεύσεις αντικειμένων αποδίδοντας υψηλά FN δηλαδή χαμηλή ανάκληση. Αντίστροφα, εάν κάποιος δεχτεί περισσότερα αντικείμενα μειώνοντας το όριο IoU, η ανάκληση θα αυξηθεί αλλά τα FP μπορεί επίσης να αυξηθούν, μειώνοντας την τιμή της ευστοχίας. Για ένα καλό μοντέλο, τόσο η ευστοχία όσο και η ανάκληση θα πρέπει να παραμένουν υψηλές ακόμη και αν το όριο IoU ποικίλλει.

$$AP@α = \int_0^1 p(r) dr$$

Το σύμβολο AP@α ή APα συμβολίζει την μέση τιμή της ευστοχίας σε μία τιμή κατωφλιού IoU α. Έτσι, τα σύμβολα AP50 και AP75 δηλώνουν την μέση τιμή ευστοχίας (Average Precision) σε τιμή IoU 50% και 75% αντίστοιχα.[16]

Κεφάλαιο 3

Πρακτικό Μέρος

Το παρακάτω κεφάλαιο χωρίζεται σε διαφορετικές ενότητες που περιγράφουν το πρακτικό κομμάτι αυτής της διπλωματικής εργασίας. Στην πρώτη ενότητα αναφέρουμε τον εξοπλισμό που χρησιμοποιήσαμε, στην δεύτερη τον αλγόριθμο που έτρεχε κατά την διάρκεια της πειραματικής διαδικασίας, στην τρίτη το σύνολο των δεδομένων που χρησιμοποιήσαμε και τέλος στο τέταρτο το μοντέλο νευρωνικού δικτύου που διαλέξαμε και η διαδικασία εκπαίδευσης του.

3.1 Duckiebot



Εικόνα 37. Ένα Duckiebot[17]

Το ρομπότ που χρησιμοποιήσαμε για το πείραμα μας ονομάζεται duckiebot και είναι συγκεκριμένα το μοντέλο DB21. Αποτελείται κυρίως από μια μπαταρία, δυο ρόδες, ένα jetson nano, μια πλακέτα HUT, δυο σερβοκινητήρες, μία κάμερα, έναν αισθητήρα IMU και έναν αισθητήρα ToF.

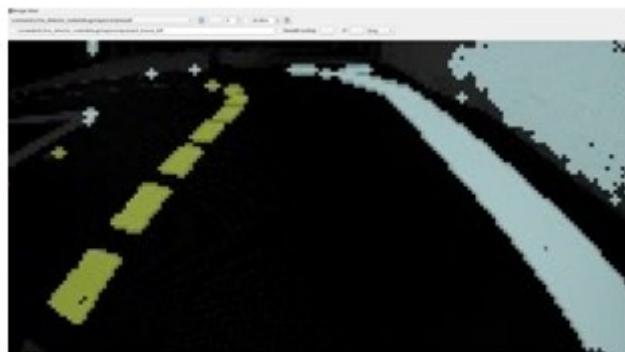
3.2 Visual Lane Servoing

Συνδυάζοντας διαφορετικά εργαλεία επεξεργασίας εικόνας καταφέρνουμε να δημιουργήσουμε μια διαδικασία ακολουθίας λωρίδας, η οποία δίνει στο duckiebot την δυνατότητα αυτόνομης οδήγησης στην ειδικά διαμορφωμένη πίστα του. Η διαδικασία ελέγχου του duckiebot είναι η εξής: Λαμβάνοντας την εικόνα της κάμερας χρησιμοποιούμε τεχνικές επεξεργασίας εικόνας για να ενισχύσουμε κομμάτια της (τις λωρίδες πιο συγκεκριμένα), προκειμένου να ξεχωρίζει το ρομπότ τις κίτρινες διακεκομμένες γραμμές και την λευκή γραμμή και να συνεχίζει της πορείας του.



Εικόνα 38. Εικόνα απο την κάμερα του Duckiebot

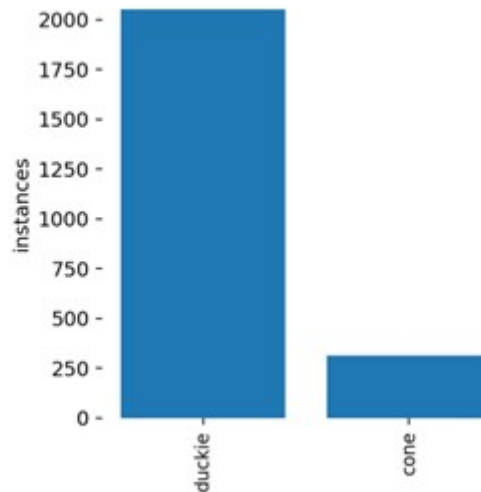
Ως εκ τούτου, αν υποθέσουμε ότι το ρομπότ ξεκινάει από το κέντρο της λωρίδας ταξιδεύοντας προς τα εμπρός με μια a ταχύτητα, χρησιμοποιούμε τη θέση των διακεκομμένων κίτρινων γραμμών και των λευκών γραμμών για να μεταβάλλουμε τις ταχύτητες περιστροφής του αριστερού και του δεξιού τροχού για να παραμείνει το ρομπότ στην λωρίδα. Για να γίνει αυτό εφαρμόζουμε ένα φίλτρο που κάνει τις λωρίδες να εμφανίζονται πιο έντονα στην εικόνα ενώ ταυτόχρονα καταπιέζει όσο περισσότερο το υπόλοιπο της εικόνας για να μην αποσπάται η προσοχή του ρομπότ. Παρακάτω φαίνεται η εικόνα που βλέπει το ρομπότ μετά από όλες τις επεξεργασίες και τα φίλτρα που δέχεται:



Εικόνα 39. Η εικόνα μετά τα φίλτρα

3.3 Σύνολο Δεδομένων

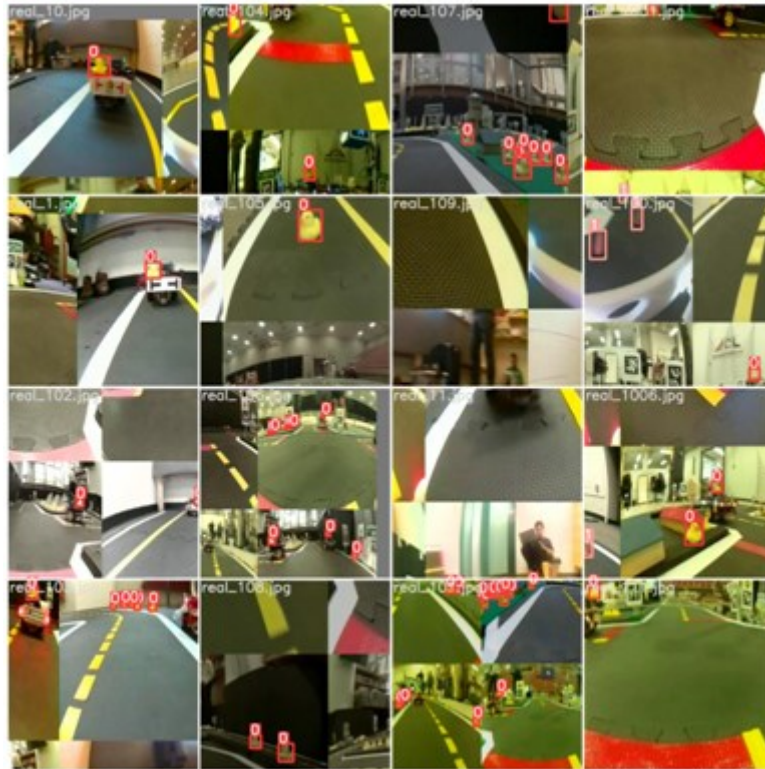
Ένα μοντέλο ανίχνευσης αντικειμένων είναι όσο καλό όσο τα δεδομένα εκπαίδευσης του. Για αυτόν τον λόγο, χρησιμοποιήσαμε δεδομένα που προσφέρει η πλατφόρμα του duckiebot και προσθέσαμε φίλτρα θορύβου και θολούρας εικόνας προκειμένου να αυξήσουμε και άλλο τα δεδομένα.



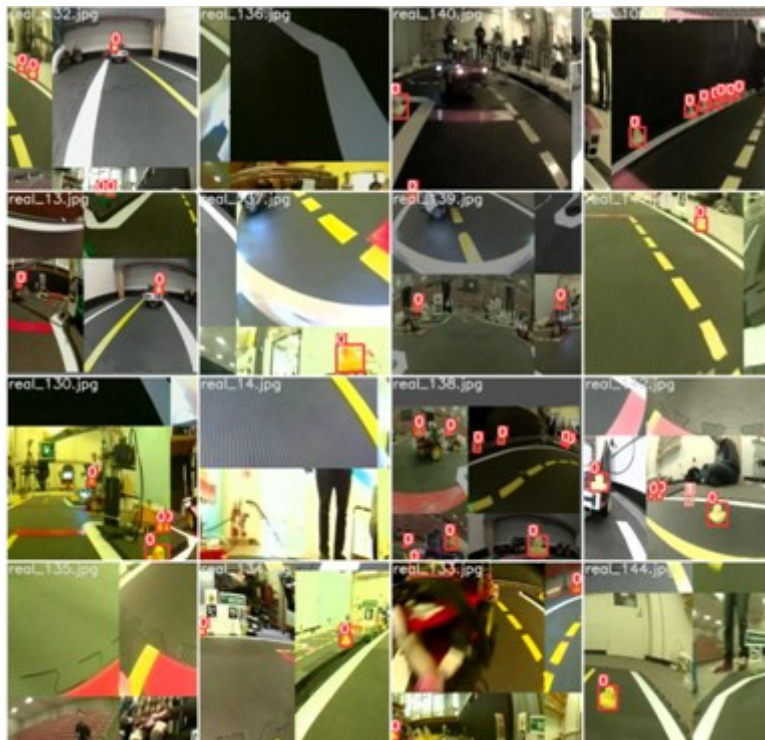
Εικόνα 40. Το σύνολο των δεδομένων μας

Ορίζουμε δυο κλάσεις για ταξινόμηση με ονόματα `duckie` και `cone`. Θέλουμε να ανιχνεύσουμε τα `duckies` και προσθέσαμε την δεύτερη κλάση μαζί με κάποιες κενές εικόνες (δεν είχαν κανένα αντικείμενο) προκειμένου να αποφύγουμε υπερπροσαρμογή. Η κάθε εικόνα συνοδεύεται από ένα `.txt` αρχείο της μορφής `C, x, y, h, w` όπου `C` είναι ο αριθμός 0 ή 1 για την εκάστοτε κλάση, `x` και `y` η συντεταγμένες της πάνω δεξιάς ή της κάτω αριστερά γωνίας του πλαισίου οριοθέτησης του αντικειμένου και `h, w` το ύψος και το πλάτος του πλαισίου οριοθέτησης. Όλες οι τιμές περνάνε από κανονικοποίηση προκειμένου να μην επηρεάζονται οι συντεταγμένες από τυχόν αλλαγές μεγέθους της εικόνας.

Παρακάτω φαίνονται κάποιες εικόνες εκπαίδευσης που δώσαμε στο μοντέλο μας:



Εικόνα 41. Πως βλέπει ο αλγόριθμος τα δεδομένα εκπαίδευσης 1



Εικόνα 42. Πως βλέπει ο αλγόριθμος τα δεδομένα εκπαίδευσης 2

3.4 Μοντέλο Εκπαίδευσης

Χρησιμοποιήσαμε το νευρωνικό δίκτυο Yolo που αναφέραμε παραπάνω και συγκεκριμένα την 5η και τελευταία έκδοση του με σκοπό την βέλτιστη ταχύτητα εκπαίδευσης και καλύτερες ανιχνεύσεις σε πραγματικό χρόνο. Το δίκτυο εκπαιδευτικό για 100 epochs, ορίσαμε το μέγεθος που θα έχουν οι εικόνες πριν την είσοδο τους στο δίκτυο στα 416x416 pixel εφόσον είναι το μέγεθος που «βλέπε» η κάμερα μας και με αυτόν τον τρόπο λαμβάνουμε καλύτερα αποτελέσματα. Σε περίπτωση που μια εικόνα δεν έχει το απαιτούμενο μέγεθος το yolo αυτόματα την μετατρέπει στην πρόπευσα τιμή και αυτός είναι ο λόγος που περνάμε τις συντεταγμένες των πλαισίων οριοθέτησης από κανονικοποίηση. Στην συνέχεια ορίζουμε εμπειρικά την τιμή batch size = 16 που δηλώνει το πόσες εικόνες θα περνάνε κάθε φορά μέσα από το δίκτυο μας. Τέλος χρησιμοποιήσαμε προεκπαιδευμένα βάρη, ειδικά για δεδομένα μικρού μεγέθους όπως τα δικά μας με σκοπό την γρηγορότερη εκπαίδευση του μοντέλου μας. Παρακάτω εμφανίζεται η τελευταία ένδειξη του μοντέλου μας αφού ολοκληρώθηκε η εκπαίδευση του και δείχνει τις τελικές πληροφορίες που μας ενδιαφέρουν.

Epoch	gpu_mem	box	obj	cls	labels	img_size		
99/99	1.34G	0.04475	0.01565	0.001049	14	416: 100%		
	Class	Images	Labels	P	R	mAP@.5	mAP@	
	all	202	571	0.706	0.675	0.729	0.357	
	duckie	202	513	0.815	0.747	0.841	0.462	
	cone	202	58	0.598	0.603	0.616	0.252	

Εικόνα 43. Τελικά αποτελέσματα του εκπαιδευμένου δικτύου

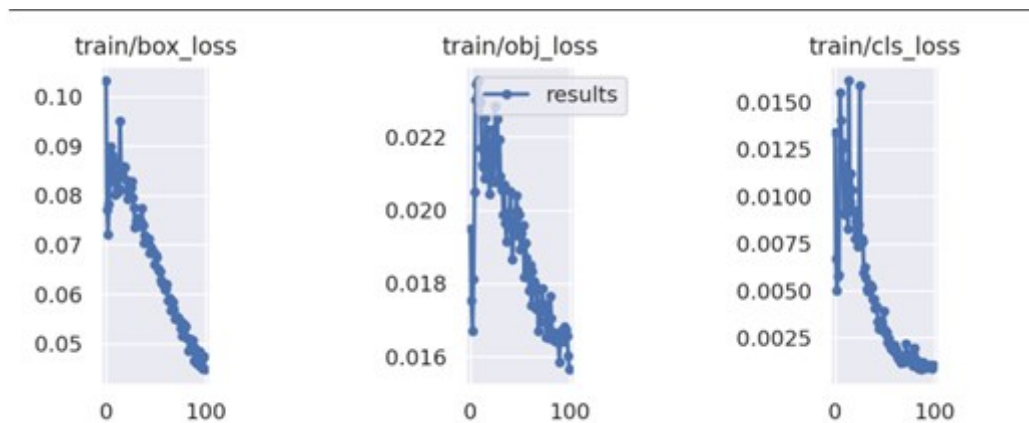
Επικεντρώνουμε την προσοχή μας στην τιμή mAP@5 η οποία όπως αναφέραμε πιο πάνω δηλώνει την μέση τιμή ευστοχίας για τιμή κατωφλιού (IoU) 50%. Με αυτόν τον τρόπο καταλαβαίνουμε ότι το μοντέλο μας είχε μια επιτυχής διαδικασία εκπαίδευσης.

Κεφάλαιο 4

Αποτελέσματα

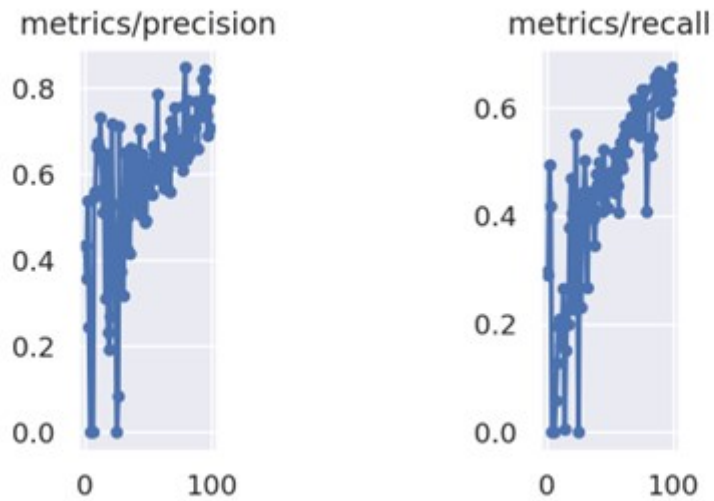
Στο παρακάτω κεφάλαιο δείχνουμε τα αποτελέσματα των πειραμάτων μας με διάφορα γραφήματα και εικόνες.

4.1 Αποτελέσματα του Yolov5

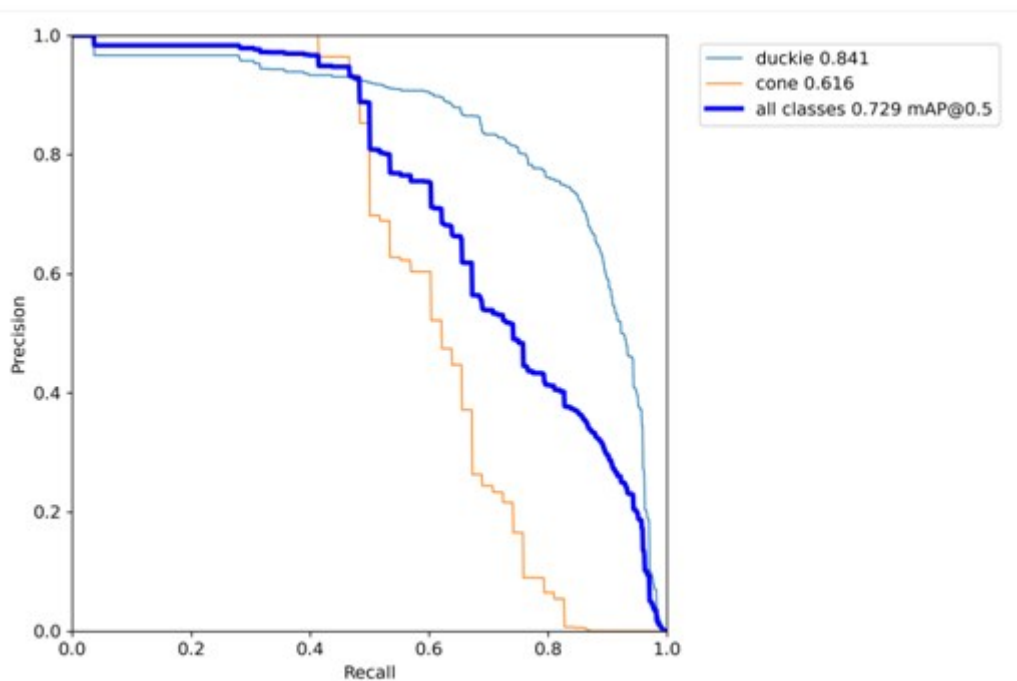


Εικόνα 44. Γραφήματα απώλειας του Yolov5

Παραπάνω παρατηρείται πως μειώνεται το σφάλμα στα πλαίσια οριοθέτησης και στην ταξινόμηση των αντικειμένων ανίχνευσης όσο αυξάνονται τα epochs. Παρακάτω βλέπουμε πως αυξάνεται η ευστοχία και η ανάκλαση αντίστοιχα

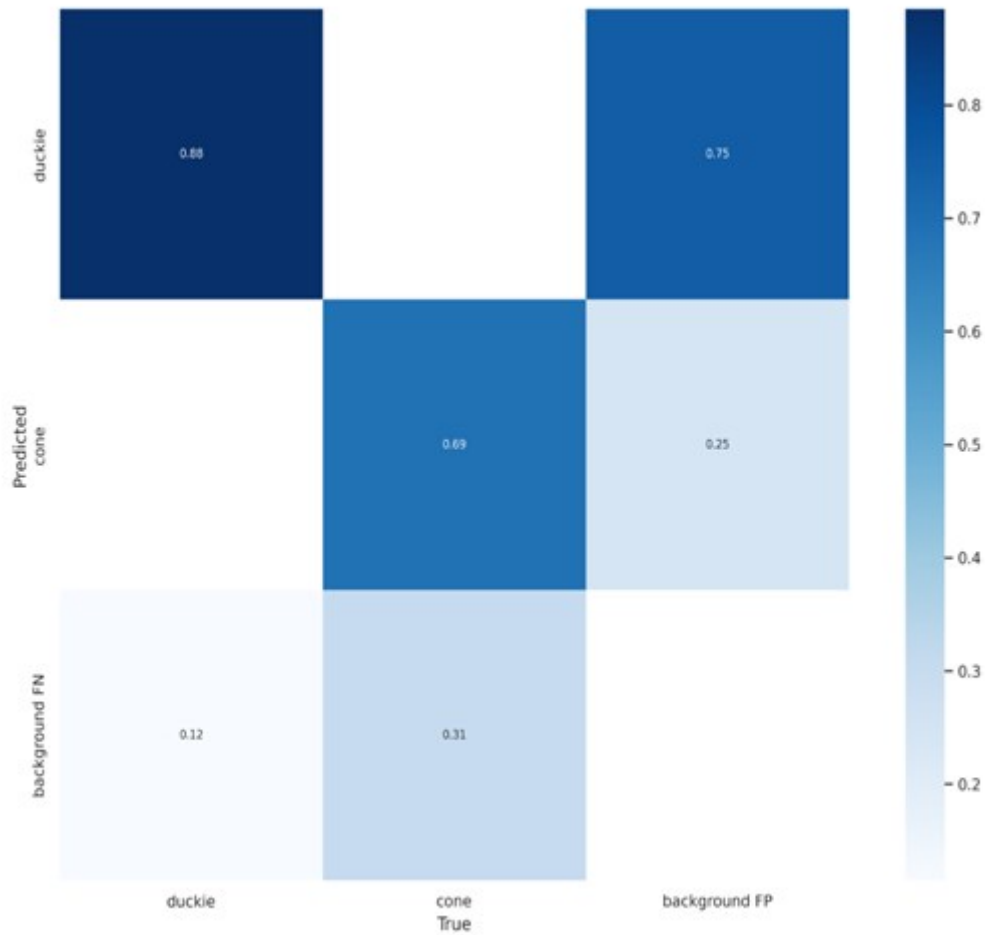


Εικόνα 45. Γράφημα ευστοχίας και ανάκλασης



Εικόνα 46. Καμπύλη Precision-Recall

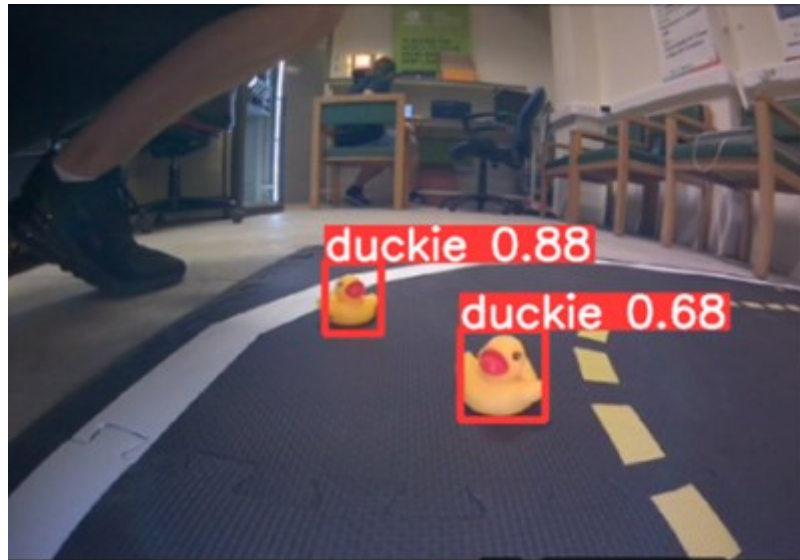
Παραπάνω φαίνεται η καμπύλη Precision-Recall στην οποία εστιάζουμε στην μπλε γραμμή (δηλώνει την τιμή $mAP@0.5$ για όλες τις κλάσεις) και συγκεκριμένα στο σημείο όπου το Precision και το Recall έχουν τις μεγαλύτερες δυνατές τιμές. Αυτά τα βάρη θα χρησιμοποιήσουμε.



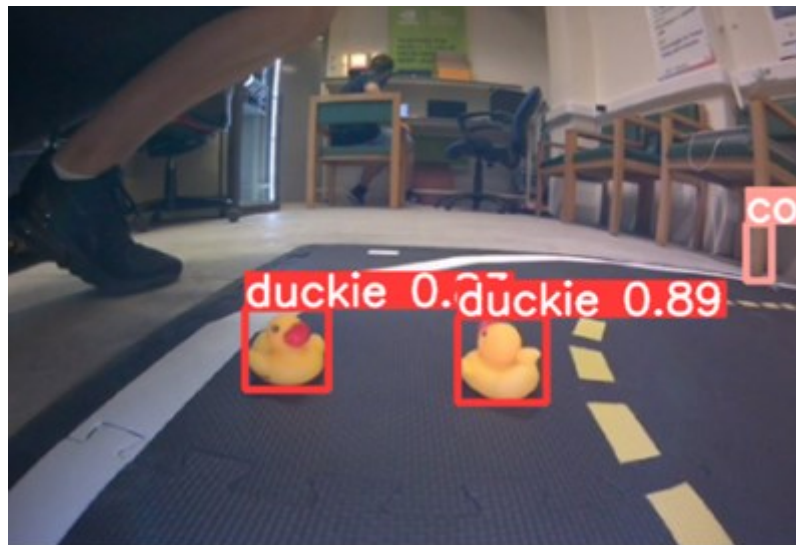
Εικόνα 47. Πίνακας σύγχυσης

Παραπάνω φαίνεται ο πίνακας που ονομάζεται Confusion Matrix και δηλώνει πάνω αριστερά τα TP πάνω δεξιά τα FP κάτω αριστερά τα FN και κάτω δεξιά τα TN. Πάντα κοιτάμε τα τετράγωνα τα οποία αντιστοιχούν σε μία κλάση και το background.

Τέλος βλέπουμε κάποια στιγμιότυπα από το Υolo ενώ τρέχει στο ρομπότ μας μέσα στην ειδικά διαμορφωμένη πίστα.



Εικόνα 48. Στιγμιότυπο αποτο πείραμα 1



Εικόνα 49. Στιγμιότυπο αποτο πείραμα 2

Κεφάλαιο 5

Επίλογος

Απο την ανάλυση και την πειραματική διαδικασία που πηγήθηκαν, παρατηρούμε καταρχάς ότι το ποσοστό της ευστοχίας αναφορικά με την ανίχνευση «ανθρώπων», ήτοι παπάκια, στο οδικό μας δίκτυο ανέρχεται σε 84%. Αντιθέτως, όσον αναφορά τους κώνους το ποσοστό ευστοχίας ανέρχεται σε 61%. Συνολικά δε, ο μέσος όρος του ποσοστού ευστοχίας για όλες τις κλάσεις αγγίζει το 73%. Απο τα προαναφερθέντα καθίσταται σαφές ότι, προκειμένου να αυξηθούν τα ποσοστά ευστοχίας, χωρίς παράλληλα να φτάσουμε στην υπερπροσαρμογή, θα πρέπει να παρέχουμε στο μοντέλο μας μεγαλύτερη ποικιλία δεδομένων και ενδεχομένως να αυξήσουμε και τον χρόνο εκπαίδευσης του μοντέλου.

Ωστόσο, παρά τα αυξημένα ποσοστά αυστοχίας στο πείραμα μας, αντιλαμβανόμαστε ότι οι συνθήκες δεν είναι ακόμη κατάλληλες για την εφαρμογή της τεχνητής νοημοσύνης στο δικό μας οδικό δίκτυο, δεδομένου ότι οι εξωγενείς παράγοντες σε συνδυασμό με την υπάρχουσα αβεβαιότητα των μοντέλων ανίχνευσης αντικειμένων καθιστούν ιδιαίτερος δύσκολο το παραπάνω εγχείρημα.

Παρόλα αυτά εκτιμούμε ότι με την ραγδαία εξέλιξη και βελτιστοποίηση της τεχνητής νοημοσύνης η τελείως αυτόνομη και ασφαλής οδήγηση θα είναι εφικτή στο μέλλον, πόσω μάλλον αν λάβουμε υπόψη την δουλειά που γίνεται στο επίπεδο αυτό, θέτοντας ήδη τα κατάλληλα θεμέλια.

Βιβλιογραφικές Αναφορές

- [1] J. McDermott, *Principles of artificial intelligence: Nils J. Nilsson*, Tioga Publishing Co., Palo Alto, CA, 1980, 476 pages, \$24.50, ISBN 0-935382-01, 1980.
- [2] *The Difference Between Artificial Intelligence, Machine Learning and Deep Learning*, 2021. διεύθυν.: <https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html>.
- [3] M. Mohri, *Foundations of Machine Learning*. 2012, σ. 432, ISBN: 9780262018258.
- [4] K. Budek, *What is reinforcement learning? The complete guide*, 2018. διεύθυν.: <https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/>.
- [5] Κ. Διαμανταράς, *Μηχανική Μάθηση*. Αθήνα: εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 2019, σ. 792, ISBN: 9789604619955.
- [6] A. Muller, *Introduction to with Python Learning Machine*. 2017, σ. 376, ISBN: 9781449369415.
- [7] *What is Underfitting?* 2021. διεύθυν.: <https://www.ibm.com/cloud/learn/underfitting>.
- [8] D. A. Rosebrock, *Deep Learning for Computer Vision – Starter Edition*. 2019, σ. 330.
- [9] *What Is Gradient Descent In Machine Learning?* 2018. διεύθυν.: <https://saugatbhattarai.com/np/what-is-gradient-descent-in-machine-learning/gradient-descent-2/>.
- [10] A. Ghosh, *Fundamental concepts of convolutional neural network*. 2019, σ. 567, ISBN: 9783030326449.
- [11] *What is data augmentation in CNN?* 2020. διεύθυν.: <https://www.quora.com/What-is-data-augmentation-in-CNN>.
- [12] R. Girshick, J. Donahue, T. Darrell και J. Malik, «Rich feature hierarchies for accurate object detection and semantic segmentation,» στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, σσ. 580–587.
- [13] R. Girshick, «Fast R-CNN,» στο *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Δεκ. 2015.
- [14] S. Ren, K. He, R. Girshick και J. Sun, «Faster r-cnn: Towards real-time object detection with region proposal networks,» *Advances in neural information processing systems*, τόμ. 28, σσ. 91–99, 2015.
- [15] J. Redmon, S. Divvala, R. Girshick και A. Farhadi, «You only look once: Unified, real-time object detection,» στο *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, σσ. 779–788.
- [16] *Object Detection Metrics With Worked Example*, 2020. διεύθυν.: <https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e>.
- [17] *Duckiebot MOOC Founder’s Edition (DB21M)*, 2020. διεύθυν.: <https://get.duckietown.com/products/duckiebot-db21-m>.

Πρόσθετη Βιβλιογραφία

- [18] J. Redmon και A. Farhadi, «Yolov3: An incremental improvement,» *arXiv preprint arXiv:1804.02767*, 2018.
- [19] A. Bochkovskiy, C.-Y. Wang και H.-Y. M. Liao, «Yolov4: Optimal speed and accuracy of object detection,» *arXiv preprint arXiv:2004.10934*, 2020.
- [20] K. Simonyan και A. Zisserman, «Very deep convolutional networks for large-scale image recognition,» *arXiv preprint arXiv:1409.1556*, 2014.

