



**Τμήμα Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Πανεπιστήμιο Πελοποννήσου**

Διπλωματική Εργασία

**Αυτόματη ανίχνευση σκουπιδιών σε εσωτερικό χώρο με τη
χρήση αλγορίθμων μηχανικής μάθησης.**

Ευθυμία Μεθοδιου

Αριθμός Μητρώου 2680

Επιβλέπων Καθηγητής: Νικόλαος Βώρος

Συνεπιβλέπουσα: Ευανθία Φαλιάγκα

ΠΑΤΡΑ 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 12/07/2022

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Νικόλαος Βώρος, Υπογραφή
2. Χρήστος Αντωνόπουλος, Υπογραφή
3. Νίκος Πετρέλλης, Υπογραφή

Υπεύθυνη Δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτητριας Ευθυμίας Μεθοδίου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης η συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας της συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Η συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Περιεχόμενα

Περίληψη	7
Κεφάλαιο 1: Εισαγωγή	9
1.1 Γενική εισαγωγή	9
1.2 Ορισμός του προβλήματος	9
1.3 Εισαγωγή στην Τεχνητή Νοημοσύνη	10
1.4 Εισαγωγή στην υπολογιστική όραση	12
1.5 Εισαγωγή στα έξυπνα σπίτια	14
Κεφάλαιο 2: Μηχανική Μάθηση	16
2.1 Εισαγωγή στην μηχανική μάθηση	16
2.2 Εξέλιξη της μηχανικής μάθησης.	17
2.3 Εφαρμογές	19
2.4 Τύποι μηχανικής μάθησης.	20
2.4.1 Μάθηση με επίβλεψη.	21
2.4.2 Μάθηση χωρίς επίβλεψη.	22
2.4.3 Μάθηση με ημι-επίβλεψη.	23
2.4.4 Μάθηση με ενίσχυση.	24
2.5 Αλγόριθμοι μηχανικής μάθησης.	25
2.5.1 K-Nearest Neighbours	25
2.5.2 Support Vector Machine	26
2.5.3 Decision Trees	26
2.5.4 Naive Bayes Classifier	27
2.5.5 Gaussian Process Classifier	27
Κεφάλαιο 3: Τεχνολογίες	28
3.1 Εισαγωγή	28
3.2 Roboflow	28
3.3 Google Colab	30
3.4 YOLO	31
3.5 TensorBoard	32
3.6 PyTorch	33
3.7 Anaconda Navigator	34
Κεφάλαιο 4: Υλοποίηση	36
4.1 Δημιουργία συνόλου δεδομένων	36
4.1.1 Προεπεξεργασία δεδομένων	38
4.1.2 Δημιουργία συνόλου δεδομένων προς ανάλυση	39
4.2 Περιβάλλον υλοποίησης	42

4.3 Εκπαίδευση συνόλου δεδομένων	44
4.4 Βήματα υλοποίησης	45
Κεφάλαιο 5: Αποτελέσματα	46
5.1 Αποτελέσματα αρχικών μοντέλων	46
5.2 Αποτελέσματα από YOLOv5	50
5.3 Αποτελέσματα από TensorBoard	53
5.4 Ανίχνευση αντικειμένων	55
5.5 Μελλοντικές κατευθύνσεις	58
Κεφάλαιο 6:	60
Βιβλιογραφία	60

Κατάλογος εικόνων

Εικόνα 1: Τεχνητή νοημοσύνη και τα υποσύνολα της.....	12
Εικόνα 2: Δυνατότητες ενός έξυπνου σπιτιού.....	15
Εικόνα 3: Τύποι μηχανικής μάθησης.....	22
Εικόνα 4: Roboflow και οι δυνατότητες του.....	29
Εικόνα 5: Περιμετρικό πλαίσιο.....	32
Εικόνα 6: Δείγμα εικόνων με χαρτόνι.....	36
Εικόνα 7: Δείγμα εικόνων με γυαλί.....	36
Εικόνα 8: Δείγμα εικόνων με μέταλλο.....	37
Εικόνα 9: Δείγμα εικόνων με χαρτί.....	37
Εικόνα 10: Δείγμα εικόνων με πλαστικό.....	37
Εικόνα 11: Δείγμα εικόνων με σκουπίδια.....	37
Εικόνα 12: Παράδειγμα εικόνων με ετικέτες.....	39
Εικόνα 13: Προεργασίες στο Roboflow.....	40
Εικόνα 14: Παράδειγμα αναστροφής και περιστροφής από Roboflow.....	41
Εικόνα 15: Παράδειγμα διάτμησης και θορύβου από Roboflow.....	41
Εικόνα 16: Κώδικας εγκατάστασης απαραίτητων βιβλιοθηκών.....	43
Εικόνα 17: Κώδικας ένταξης του συνόλου δεδομένων στο Google Colab.....	43
Εικόνα 18: Παράμετροι εκπαίδευσης.....	44
Εικόνα 19: Κώδικας για φόρτιση γράφων από το TensorBoard.....	44
Εικόνα 20: Κώδικας για ανίχνευση στις εικόνες.....	45
Εικόνα 21: Κώδικας για εμφάνιση αποτελεσμάτων ανίχνευσης.....	45
Εικόνα 22: Γραφική αναπαράσταση F1 με Confidence.....	51
Εικόνα 23: Γραφική αναπαράσταση Precision με Confidence.....	51
Εικόνα 24: Γραφική αναπαράσταση Precision με Recall.....	52
Εικόνα 25: Προβλέψεις του YOLO.....	52
Εικόνα 26: Προβλέψεις του YOLO.....	53
Εικόνα 27: Γραφήματα μετρήσεων TensorBoard.....	54
Εικόνα 28: Γραφήματα εκπαίδευσης TensorBoard.....	54
Εικόνα 29: Γραφήματα επικύρωσης TensorBoard.....	55
Εικόνα 30: Αποτελέσματα YOLO από εικόνες. (1).....	57
Εικόνα 31: Αποτελέσματα YOLO από εικόνες. (2).....	57
Εικόνα 32: Αποτελέσματα YOLO από βίντεο.....	58

Κατάλογος πινάκων

Πίνακας 1: Μοντέλο με 80 εικόνες ανά κατηγορία.....	46
Πίνακας 2: Μοντέλο με 100 εικόνες ανά κατηγορία.....	47
Πίνακας 3: Μοντέλο με 130 εικόνες ανά κατηγορία.....	48
Πίνακας 4: Μοντέλο με 200 εικόνες ανά κατηγορία.....	49
Πίνακας 5: Μοντέλο με 50 εικόνες ανά κατηγορία.....	49

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε με σκοπό τη μελέτη του προβλήματος επεξεργασίας, οπτικής αναγνώρισης και ταξινόμησης αντικειμένων με τεχνικές μηχανικής μάθησης. Αρχικά γίνεται μια εισαγωγή στην έννοια της τεχνητής νοημοσύνης, στην υπολογιστική όραση και στα έξυπνα σίτια. Στην συνέχεια γίνεται μια εισαγωγή στην μηχανική μάθηση, αναφέρεται η εξέλιξη και εφαρμογές της και αναλύονται περαιτέρω οι τύποι της όπως η μάθηση με επίβλεψη και η μάθηση χωρίς επίβλεψη αλλά και κάποιοι αλγόριθμοι της μαθητικής μάθησης όπως τα δέντρα αποφάσεων και ο αλγόριθμος K-πλησιέστερων γειτόνων. Έπειτα αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν για την πραγματοποίηση του πρακτικού κομματιού της διπλωματικής εργασίας όπως το YOLOv5 και το Google Colab. Ακολούθως αναλύονται τα βήματα που έγιναν για την κατασκευή της εφαρμογής που αναγνωρίζει έξι διαφορετικά είδη από σκουπίδια. Ξεκινώντας από την δημιουργία του συνόλου των δεδομένων (dataset), την προετοιμασία και προεπεξεργασία αυτών των δεδομένων πριν την χρήση τους στην εκπαίδευση του αλγορίθμου. Περιγράφεται το περιβάλλον υλοποίησης της και ο αλγόριθμος που χρησιμοποιήθηκε για την εκπαίδευση. Τέλος, παρουσιάζονται τα αποτελέσματα τα από διάφορα πειράματα που πραγματοποιήθηκαν και παρατίθενται τα αποτελέσματα της τελικής εφαρμογής και οι μελλοντικές κατευθύνσεις αυτής της εφαρμογής.

Abstract

The present thesis was prepared with the aim of studying the problem of processing, visual recognition and classification of objects with machine learning techniques. Initially, an introduction to the concept of artificial intelligence, computer vision and smart houses is made. Then, there is an introduction to machine learning, its evolution and its applications. Also there are mentioned, its types such as supervised learning and unsupervised learning, as well as some learning algorithms such as decision trees and the K-Nearest neighbour algorithm. Next the technologies used to accomplish the practical part of the thesis are mentioned, such as YOLOv5 and Google Collab. Subsequently the steps that took place to build the application that identifies six different types of garbage. Then, the implementation environment and the algorithm that was used for the training of the application are described. Finally, the results of the various experiments performed are presented, the results of the final application and the future directions of this application are presented.

Κεφάλαιο 1: Εισαγωγή

1.1 Γενική εισαγωγή

Στην σύγχρονη εποχή η ανάπτυξη της τεχνολογίας και η μαζική χρήση των ηλεκτρονικών υπολογιστών έχει αλλάξει ριζικά πολλούς παράγοντες της καθημερινότητας των ανθρώπων. Η τεχνολογίες και οι διάφορες εφαρμογές της έχουν κατακλύσει όλες τις πτυχές της ζωής μας, όπως η υγεία, η ψυχαγωγία και η εκπαίδευση. Όλες αυτές οι εφαρμογές βασίζονται στην συγκομιδή δεδομένων. Τα δεδομένα αυτά μπορεί να υπάρχουν γύρω μας έτοιμα προς συλλογή, όμως η σωστή ανάλυση και διερμηνεία αυτών, είναι που δημιουργεί την πληροφορία που θα μετατραπεί σε γνώση. Η αδυναμία του ανθρώπου να διαχειριστεί αυτόν τον μεγάλο όγκο δεδομένων, οδήγησε στην ανάπτυξη του κλάδου της πληροφορικής που ονομάζεται μηχανική μάθηση και έχει ως στόχο την αυτόματη εξαγωγή συμπερασμάτων.

Η μηχανική μάθηση αποτελεί έναν από τους πιο εξελισσόμενους τομείς της επιστήμης της πληροφορικής. Έχει προσφέρει πληθώρα μεθόδων και τεχνικών για την επίλυση προβλημάτων και την πρόβλεψη αποτελεσμάτων σε ένα μεγάλο εύρος ζητημάτων. Η ταχύτητα και η ακρίβεια των αποτελεσμάτων που προσφέρει, καθιστούν την χρήση της πολύτιμη και δίνουν ένα ισχυρό πλεονέκτημα σε όσους επιλέξουν να την εφαρμόσουν. Αν και βρίσκεται ακόμα σε στάδια έρευνας και ανάπτυξης οι εφαρμογές της έχουν βρει ήδη χώρο στη βιομηχανία και στην καθημερινότητα των ανθρώπων με ποικίλους τρόπους. [1]

1.2 Ορισμός του προβλήματος

Στην παρούσα διπλωματική εργασία το πρόβλημα το οποίο καλούμαστε να επιλύσουμε είναι η οπτική αναγνώριση σκουπιδιών τα οποία βρίσκονται στον εσωτερικό χώρο ενός σπιτιού όπως για παράδειγμα η κουζίνα. Η εργασία εντάσσεται στην ευρύτερη περιοχή της τεχνητής νοημοσύνης και επί το πλείστον της μηχανικής μάθησης. Η ανάλυση και η επεξεργασία δεδομένων εκφρασμένων σε μορφή εικόνων θεωρείται σήμερα ένας από τους πιο σημαντικούς κλάδους της μηχανικής μάθησης. Τα αντικείμενα τα οποία εμπεριέχονται σε

αρχεία εικόνων και βίντεο μπορούν να εντοπιστούν και να αναγνωριστούν αυτόματα. Η εν λόγω διαδικασία ονομάζεται αναγνώριση αντικειμένων και αποτελεί ένα από τα μεγαλύτερα προβλήματα του computer vision (υπολογιστική όραση). Κάθε κλάση αντικειμένων έχει τις δικές της μοναδικές ιδιότητες, γεγονός που βοηθάει στην ταξινόμηση της κλάσης.

Για την επίλυση αυτού του προβλήματος αναπτύξαμε ένα καινούργιο μοντέλο μηχανικής μάθησης προσαρμοσμένο στο πρόβλημα μας, χρησιμοποιώντας νευρωνικά δίκτυα μαζί με τεχνικές επιβλεπόμενης μάθησης. Στην διαδικασία της ανάπτυξης του συγκεκριμένου μοντέλου, αντιμετωπίσαμε το πρόβλημα της ακατέργαστης μορφής των δεδομένων. Για την εκπαίδευση του μοντέλου και την ταξινόμηση των εικόνων στις έξι κατηγορίες μας, έπρεπε να γίνουν ειδικά βήματα προεπεξεργασίας τα οποία μειώνουν την ποσότητα πληροφορίας και το μέγεθος κάθε εικόνας, φτάνοντας έτσι στην μορφή η οποία περιέχει μόνο την πληροφορία που χρειαζόμαστε, που σε αυτήν την περίπτωση είναι τα σκουπίδια. Κάναμε την εκπαίδευση του μοντέλου αποθηκεύοντας τα τελικά βάρη (weights) από το δικό μας μοντέλο εκπαιδευμένο πάνω στο δικό μας dataset. Έπειτα χρησιμοποιώντας τα βάρη αυτά, πήραμε αποτελέσματα από εικόνες που το μοντέλο μας δεν είχε ξανά ανιχνεύσει καθώς και αποτελέσματα από βίντεο μη πραγματικού χρόνου και βίντεο πραγματικού χρόνου (live) μέσω κάμερας υπολογιστή (webcam). [2]

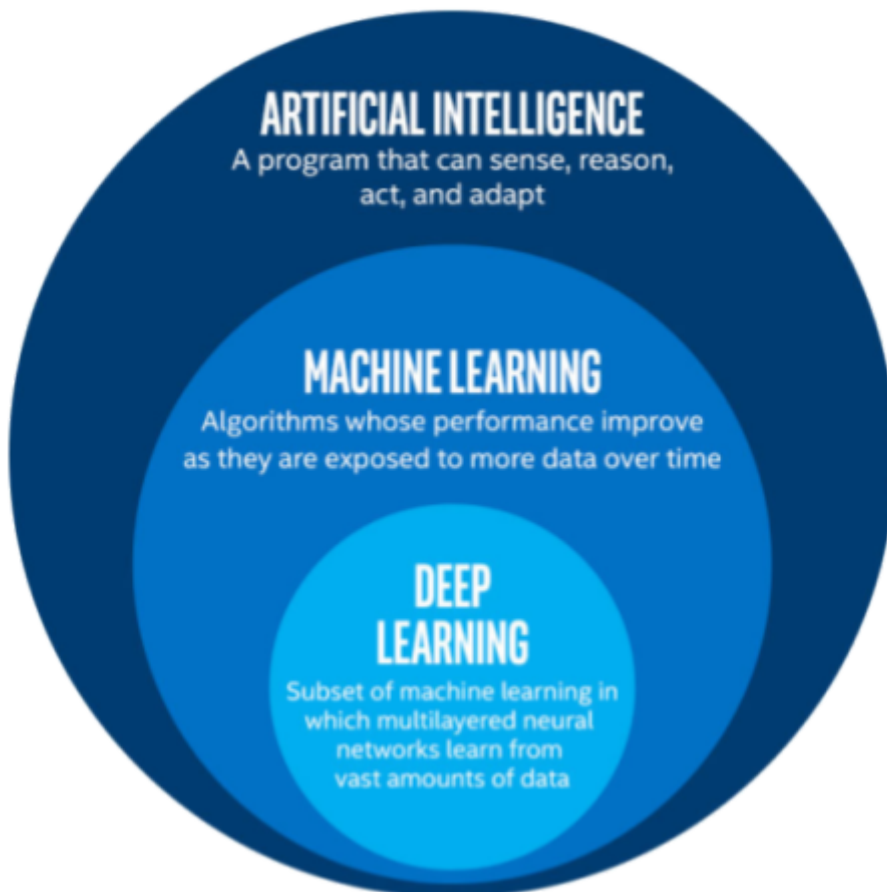
1.3 Εισαγωγή στην Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη (AI) είναι η επιστήμη του κλάδου της πληροφορικής η οποία ασχολείται με τη σχεδίαση και τη δημιουργία υπολογιστικών συστημάτων τα οποία μπορούν στοιχειωδώς να σκεφτούν και να μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς όπως η μάθηση, η εξαγωγή συμπερασμάτων, η αντιμετώπιση προβλημάτων, η προσαρμοστικότητα κ.α.

Πατέρας της τεχνητής νοημοσύνης θεωρείται ο Τζον Μακάρθι, ο οποίος και την όρισε ως “την επιστήμη της δημιουργίας νοήμωνων μηχανών”. Ο επίσημος όρος “τεχνητή νοημοσύνη” επινοήθηκε το 1956 όταν οι Μάρβιν Μίνσκι και Τζον Μακάρθι, οι οποίοι ήταν επιστήμονες πληροφορικής στο πανεπιστήμιο Στάνφορντ στις ΗΠΑ, φιλοξένησαν το πρόγραμμα τεχνητής νοημοσύνης σε μια έρευνα του κολλεγίου Ντάρτμουθ. Οι ραγδαίες εξελίξεις στον τομέα της πληροφορικής, της συλλογής και αποθήκευσης δεδομένων έχουν καταστήσει την τεχνητή νοημοσύνη έναν ιδιαίτερα ελκυστικό και σημαντικό τομέα της επιστήμης της

πληροφορικής τόσο για ερευνητές όσο και για επαγγελματίες, βρίσκοντας εφαρμογή σε πολλούς επιστημονικούς και επιχειρηματικούς κλάδους.

Η τεχνητή νοημοσύνη χρησιμοποιείται για την βελτίωση πολύπλοκων διεργασιών μέσω της αυτοματοποίησης τους, βελτιώνοντας έτσι και την απόδοση των συστημάτων. Οι διεργασίες περιλαμβάνουν συλλογιστική για επίλυση προβλημάτων, αναπαράσταση γνώσης για μοντελοποίηση, ικανότητα αντίληψης, όπως επεξεργασία εικόνας και ήχου, σχεδιασμό και επικοινωνία, όπως είναι η επεξεργασία γλώσσας καθώς επίσης και αυτόνομα συστήματα όπως τα ρομπότ. Η τεχνητή νοημοσύνη διακρίνεται σε συμβολική και σε μη συμβολική. Η συμβολική τεχνητή νοημοσύνη επιχειρεί την εξομοίωση της ανθρώπινη νοημοσύνης αλγοριθμικά με χρήση συμβόλων και λογικών κανόνων υψηλού επιπέδου. Σε αυτή συμπεριλαμβάνονται η αναπαράσταση γνώσης (knowledge representation), συλλογιστική (reasoning) και η πρόβλεψη/προγραμματισμός (prediction/planning). Η μη συμβολική τεχνητή νοημοσύνη προσπαθεί να προσομοιώσει την ανθρώπινη ευφυΐα, χρησιμοποιώντας στοιχειώδη αριθμητικά μοντέλα που συνθέτουν επαγωγικά νοήμονες συμπεριφορές, μέσω της διαδοχικής αυτοοργάνωσης απλούστερων δομικών συστατικών. Με αυτόν τον τρόπο είναι εφικτή η προσομοίωση πραγματικών βιολογικών διαδικασιών, όπως για παράδειγμα η λειτουργία του ανθρώπινου εγκεφάλου. Μια από τις υποκατηγορίες που ανήκουν στην μη συμβολική τεχνητή νοημοσύνη είναι η μηχανική μάθηση, η οποία ασχολείται με την βελτίωση της γνώσης και της απόδοσης της μηχανής μέσω δεδομένων, αλληλεπιδράσεων με το περιβάλλον κ.α. Όπως βλέπουμε στην παρακάτω εικόνα, η μηχανική μάθηση αποτελεί αν όχι το μεγαλύτερο, ένα από τα μεγαλύτερα σύνολα/κατηγορία της τεχνητής νοημοσύνης και αποτελείται από αλγορίθμους που συνεχώς βελτιώνουν την απόδοσή τους καθώς με την πάροδο του χρόνου εκπαιδεύονται πάνω σε περισσότερα δεδομένα. Επίσης ένα υποσύνολο της μηχανικής μάθησης είναι η βαθιά μάθηση στην οποία πολυεπίπεδα νευρωνικά δίκτυα μαθαίνουν από πολύ μεγάλες ποσότητες δεδομένων. Αν και η ανοδική πορεία της τεχνητής νοημοσύνης είναι αδιαμφισβήτητη, παρόλα αυτά εξακολουθεί να αποτελεί ένα ανοικτό και πρόσφορο πεδίο προς περαιτέρω διερεύνηση και έρευνα. [3]



Εικόνα 1: Τεχνητή νοημοσύνη και τα υποσύνολα της.

1.4 Εισαγωγή στην υπολογιστική όραση

Ένας από τους βασικότερους τομείς της τεχνητής νοημοσύνης είναι η υπολογιστική όραση (computer vision), κατά την οποία οι μηχανές προσπαθούν να προσομοιώσουν την αίσθηση της ανθρώπινης όρασης. Χρησιμοποιώντας την υπολογιστική όραση, δίνεται η δυνατότητα σε αυτόνομα υπολογιστικά συστήματα να πραγματοποιούν λειτουργίες που βασίζονται στην όραση, όπως η αντίληψη αντικειμένων σε έναν χώρο και η κατηγοριοποίηση τους. Η όραση των υπολογιστών είναι πλέον ένα από τα σημαντικότερα εργαλεία του ανθρώπου στον τομέα της υγείας, της ψυχαγωγίας και γενικότερα για την βελτίωση των συνθηκών διαβίωσης του. Η πρώτη προσπάθεια έγινε το 1958 από τον Φρανκ Ρόζενμπλατ, ο οποίος παρουσίασε τον πρώτο νευρώνα (perceptron) στην προσπάθεια προσομοίωσης της εκμάθησης του

ανθρώπινου εγκεφάλου. Η ιδέα του χρησιμοποιήθηκε για την ταξινόμηση εικόνων με καμουφλαρισμένα τανκ σε δάση, σε σχέση με εικόνες που είχαν μόνο δάση. Σχεδόν δέκα χρόνια αργότερα το 1969, οι Μάρβιν Μίνσκι και Σέιμουρ Πέιπερτ με το βιβλίο τους *Perceptrons*, έδειξαν τρόπους με τους οποίους τα νευρωνικά δίκτυα (multi-layer perceptrons) μπορούν να λύσουν σε γραμμικό χρόνο προβλήματα πολλών επιπέδων, με αποτέλεσμα να είναι πλέον εφικτή η πολυδιάστατη ταξινόμηση αντικειμένων μιας εικόνας.

Χρησιμοποιώντας μια κάμερα ως μάτια, η υπολογιστική όραση προσομοιώνεται με τεχνικές επεξεργασίας εικόνων, μέσα από την επεξεργασία των καρέ ενός βίντεο ή των εικόνων που δέχεται ως είσοδο. Μια ψηφιακή εικόνα μπορεί να παραχθεί από διαφορετικές συσκευές, οι οποίες παρέχουν διαφορετικές πληροφορίες προς επεξεργασία. Πληροφορίες όπως, οι εικόνες μπορεί να είναι δισδιάστατες ή τρισδιάστατες, οι τιμές των pixels μπορεί να αντιστοιχούν στην ένταση του φωτός ανάλογα με τη φασματική ζώνη της εικόνας, δηλαδή το αν είναι έγχρωμη ή ασπρόμαυρη, όπως επίσης και τα φυσικά μεγέθη της εικόνας όπως βάθος και απορρόφηση. Μετά την λήψη της εικόνας από την συσκευή, πραγματοποιούνται τα εξής βήματα για την αναγνώριση των αντικειμένων. Προεπεξεργασία της εικόνας η οποία εφαρμόζεται ώστε να είναι δυνατή η εξαγωγή των απαιτούμενων πληροφοριών από την εικόνα. Μορφές προεπεξεργασίας είναι η μείωση του θορύβου της εικόνας, αύξηση της αντίθεσης καθώς και κλιμάκωση των διαστάσεων της με σκοπό την εστίαση σε συγκεκριμένα σημεία. Εξαγωγή των πληροφοριών από την εικόνα αναφέρεται στον εντοπισμό των χαρακτηριστικών της εικόνας με σκοπό τον επιτυχή διαχωρισμό των αντικειμένων από το υπόβαθρο τους. Τα χαρακτηριστικά αυτά εστιάζονται κυρίως σε γραμμές, ακμές και κορυφογραμμές. Επόμενο βήμα είναι ο εντοπισμός και η κατάτμηση, στο σημείο αυτό γίνεται μια διαλογή των περιεχομένων που θα χρησιμοποιηθούν για περαιτέρω επεξεργασία, ανάλογα με το πρόβλημα που ζητείται να επιλυθεί. Μετά από αυτό γίνεται η επεξεργασία υψηλού επιπέδου, όπου εδώ η είσοδος είναι ένα μικρό σύνολο δεδομένων που περιέχει τα σημεία προς αναγνώριση, με σκοπό να επιτευχθεί η επαλήθευση τους ως προς την ικανοποίηση των υποθέσεων που είναι τεθεί για το πρόβλημα. Τελικό βήμα είναι η λήψη των αποφάσεων, αυτό το στάδιο αναφέρεται στις διαδικασίες που ακολουθούνται μετά την ταξινόμηση του αντικειμένου, όπως ενέργειες μετά την θετική ή την αρνητική αναγνώριση ενός αντικειμένου.[4] [5]

1.5 Εισαγωγή στα έξυπνα σπίτια

Τα πρώτα έξυπνα σπίτια ήταν ιδέες, όχι πραγματικές κατασκευές. Για δεκαετίες η επιστημονική φαντασία έχει εξερευνήσει την ιδέα του αυτοματισμού στο σπίτι. Το έξυπνο σπίτι ή αλλιώς το σπίτι του μέλλοντος δεν αποτελεί πλέον μια καινούργια έννοια ή ιδέα. Τα έξυπνα είναι πραγματικότητα τόσο στο εξωτερικό όσο και στο εσωτερικό. Αν και πριν μερικά χρόνια φαινόταν σενάριο επιστημονικής φαντασίας, τώρα πια ο αυτοματισμός και η ευφυΐα των σπιτιών αποτελούν πραγματικότητα.

Το ενδιαφέρον για την έξυπνη οικιακή τεχνολογία έχει αυξηθεί ραγδαία κατά τη διάρκεια της τελευταίας δεκαετίας. Πρόσφατα η ώθηση προς την συνεχή βελτίωση της τεχνολογίας στην ζωή μας, μεταφέρθηκε σταδιακά στις οικιακές συσκευές, αλλάζοντας με αυτόν τον τρόπο τις απαιτήσεις του παραδοσιακού σπιτιού όπως το ξέραμε. Δηλαδή, οι προσδοκίες των ανθρώπων για τα οικιακά περιβάλλοντα μετατοπίζονται όλο και περισσότερο από την ενεργό ώθηση των κουμπιών στην εναλλαγή των διακοπών στο σπίτι, οι οποίες δημιουργούνται από την αυτοματοποιημένη και πανταχού παρούσα τεχνολογία των υπολογιστών. Οι συσκευές δεν χρειάζεται να εκτελούν απλές απομονωμένες εργασίες, αλλά μπορούν να αποτελούν μέρος ενός καταναλωμένου τεχνολογικού συστήματος. Όμως, η αγορά έχει ακόμα πολύ δρόμο για να κατανοήσει και, κατα συνέπεια, να προσαρμόσει αυτό το είδος αυτοματισμού σε αυτό που οι καταναλωτές αντιλαμβάνονται ως έξυπνο.

Ένω τα έξυπνα σπίτια αναμένεται να διαδραματίσουν σημαντικό ρόλο στην ζωή μας στο μέλλον, σήμερα εξακολουθούν να βρίσκονται σε μια πρώιμη φάση και αγωνίζονται να επιτύχουν την μαζική υιοθεσία από τους καταναλωτές. Η ζήτηση είναι ακόμα περιορισμένη καθώς οι υπάρχουσες λειτουργίες συνήθως περιορίζονται σε αυτόνομες συσκευές που προσθέτουν πολυπλοκότητα και όχι φιλικότητα προς τον χρήστη τους. Οι έξυπνες οικιακές συσκευές περιορίζονται από ένα σύνολο χαρακτηριστικών που ορίζουν οι κατασκευαστές και η αλληλεπίδραση των καταναλωτών με αυτές εξακολουθεί να μοιάζει αδέξια και μη φυσική. Παρόλα αυτά, τα έξυπνα σπίτια που είναι εξοπλισμένα με την τεχνολογία των αντικειμένων (Internet of Things - IoT) κερδίζουν γρήγορα δημοτικότητα, χάρης στην άνεση και την ασφάλεια που προσφέρουν. Με σχεδόν όλους τους χώρους των σπιτιών να αλληλοσυνδέονται και να ελέγχονται από απομακρυσμένο έλεγχο, σε αυτόν τον έλεγχο συμπεριλαμβάνονται ο φωτισμός, η θερμοκρασία και η ασφάλεια. Ο έξυπνος οικιακός αυτοματισμός αναγνωρίζεται ευρέως ως μια καινοτομία που ενισχύει σημαντικά τη ζωή των ανθρώπων που τον χρησιμοποιεί. Υπο το πρίσμα των σχεδόν άπειρων δυνατοτήτων που

προσφέρει η έξυπνη οικιακή τεχνολογία με το IoT, κάθε όψη ενός σπιτιού μπορεί να ενισχυθεί με λύσεις αιχμής, αξιοποιώντας τη δύναμη του IoT. [6] [7]



Εικόνα 2: Δυνατότητες ενός έξυπνου σπιτιού.

Κεφάλαιο 2: Μηχανική Μάθηση

2.1 Εισαγωγή στην μηχανική μάθηση

Η μάθηση είναι μια από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Είναι η διαδικασία κατά την οποία ο υποβαλλόμενος σε αυτή αποκτά γνώσεις, δεξιότητες, συμπεριφορές και αξίες μέσα από την παρατήρηση του περιβάλλοντος, την σύγκριση και εφαρμογή γνωστικών διαδικασιών. Οι αλγόριθμοι μάθησης χρησιμοποιούνται ευρέως σε εφαρμογές μηχανικής.

Η μηχανική μάθηση (machine learning) αποτελεί έναν βασικό και πολλά υποσχόμενο τομέα της τεχνητής νοημοσύνης. Μπορεί να θεωρηθεί ως μια συλλογή από μεθόδους που μπορούν αυτόματα να αναγνωρίσουν διάφορα μοτίβα σε δεδομένα και στη συνέχεια με βάση αυτά τα μοτίβα να προβλέψουν μελλοντικά αποτελέσματα ή να πάρουν αποφάσεις κάτω από συγκεκριμένες καταστάσεις. Η μηχανική μάθηση ασχολείται με τη μελέτη αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους από προηγούμενες εκτελέσεις της ίδιας ή παραπλήσιας εργασίας σε διαφορετικά δεδομένα. Όλα αυτά υλοποιούνται χρησιμοποιώντας διάφορους αλγορίθμους που επιτρέπουν στις μηχανές να καταλαβαίνουν καταστάσεις και βασισμένες σε αυτές, να παίρνουν αποφάσεις. Ο αλγόριθμος μαθαίνει από πειραματικά δεδομένα και με αυτόν τον τρόπο βελτιώνει την επίδοσή του. Από την εκπαίδευση αυτή προκύπτει το μοντέλο, το οποίο μπορεί να κάνει προβλέψεις ή να πάρει αποφάσεις για τα νέα δεδομένα που του δίνονται ως προς το πρόβλημα που του έχει ανατεθεί. Η ικανότητα των υπολογιστικών συστημάτων να αποθηκεύουν μεγάλο όγκο πληροφοριών και να μπορούν να τον επεξεργαστούν ή να εκτελέσουν πράξεις σε ελάχιστο χρόνο, βοήθησε την ταχεία εξέλιξη της μηχανικής μάθησης. Η μηχανική μάθηση βρίσκει χρήση σε πολλούς τομείς, αφού παρέχει πολλά πλεονεκτήματα όπως τα γρήγορα και καλά αποτελέσματα τα οποία συνεχίζουν να βελτιώνονται, προσαρμοστικότητα στο περιβάλλον και διορατικότητα.

Οι επιστήμονες του χώρου της πληροφορικής και της τεχνητής νοημοσύνης ορίζουν την μηχανική μάθηση ως “το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να

προγραμματιστεί εκ νέου”. Με άλλα λόγια, την δημιουργία υπολογιστικών συστημάτων που είναι ικανά να μάθουν και να βελτιώσουν την απόδοση τους, κάνοντας κατάλληλη χρήση των αποτελεσμάτων κάποιας εργασίας. Ένας σχετικός ορισμός της μηχανικής μάθησης δίνεται από τον Τομ Μίτσελ το 1997, “ ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μετρό απόδοσης P , αν η απόδοση του σε εργασίες από το T , όπως μετρείται από το P , βελτιώνεται μέσω της εμπειρίας E ”.

Όσον αφορά τη σχεδίαση των συστημάτων μηχανικής μάθησης, για τα συστήματα που ανήκουν στη συμβολική τεχνητή νοημοσύνη, η δυνατότητα μάθησης προσδιορίζεται ως η ικανότητα πρόσκτησης επιπλέον γνώσης, που επιφέρει μεταβολές στην υπάρχουσα καταχωρημένη γνώση είτε αλλάζοντας χαρακτηριστικά της είτε με αυξομείωση της. Ενώ στην περίπτωση των συστημάτων που ανήκουν στη μη συμβολική τεχνητή νοημοσύνη, ως μάθηση προσδιορίζεται η δυνατότητα που διαθέτουν τα συστήματα στο να μετασχηματίζουν την εσωτερική τους δομή, παρά στο να μεταβάλλουν κατάλληλα τη γνώση που έχει καταχωρηθεί μέσα σε αυτά κατά τον σχεδιασμό τους. Αν και απέχουμε πάρα πολύ από τη δημιουργία μηχανών που μαθαίνουν τόσο καλά όσο ο άνθρωπος, για συγκεκριμένες περιοχές μάθησης έχουν αναπτυχθεί αλγόριθμοι οι οποίοι έχουν επιτρέψει την εμφάνιση σύγχρονων εμπορικών εφαρμογών. Επιπλέον, τα αποτελέσματα από τις εφαρμογές της τεχνητής νοημοσύνης αρχίζουν ήδη να είναι ορατά και να δίνουν απαντήσεις σε απάντησα εως τώρα ερωτήματα άλλων κλάδων που διερευνούν την ικανότητα του ανθρώπου να μαθαίνει. Εκτός της ίδιας της τεχνητής νοημοσύνης, μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της μηχανικής μάθησης συγκαταλέγονται οι εξόρυξη δεδομένων, πιθανότητες και στατιστική, θεωρία της πληροφορικής, αριθμητική βελτιστοποίηση, θεωρία της πολυπλοκότητας, θεωρία ελέγχου, ψυχολογία, νευρολογία και γλωσσολογία. [8]

2.2 Εξέλιξη της μηχανικής μάθησης.

Οι έννοιες της τεχνητής νοημοσύνης και της μηχανικής μάθησης δεν είναι πρόσφατες στην επιστημονική κοινότητα. Έχουν μελετηθεί, χρησιμοποιηθεί και ακόμα ερευνώνται από επιστήμονες υπολογιστών, μηχανικούς, ερευνητές και επαγγελματίες για δεκαετίες. Αφετηρία στην ανάπτυξη της μηχανικής μάθησης αποτέλεσε η συμβολή ερευνητών όπως ο

Άλαν Τούρινγκ, Τζον Μακκάρθι, Άρθουρ Σάμιουελς, Άλαν Νιούελ και Φρανκ Ρόζενμπλατ κατά την δεκαετία του 1950 και του 1960.

Το 1950 ο Τούρινγκ δημιούργησε το “Test Turing” για να διαπιστώσει αν μια μηχανή είναι πραγματικά έξυπνη. Για να περάσει το τεστ, η μηχανή πρέπει να είναι ικανή να κάνει έναν άνθρωπο να πιστεύει ότι είναι η ίδια άνθρωπος αντί για υπολογιστής. Το 1952 ο Σάμιουελς πρότεινε το πρώτο μοντέλο μηχανικής μάθησης για την βελτιστοποίηση ενός προγράμματος εφαρμογής σε παιχνίδι ντάμας (checkers). Το 1958 ο Ρόζενμπλατ δημιούργησε το Perceptron, έναν δημοφιλή αλγόριθμο μηχανικής μάθησης ο οποίος πηγάζει από τους ανθρώπινους βιολογικούς νευρώνες και με αυτόν τον τρόπο έθεσε τις βάσεις δημιουργίας του τεχνητού νευρωνικού δικτύου. Η δεκαετία του 80 είναι γνωστή για την γέννηση των “expert systems” ειδικών συστημάτων βασισμένα σε κανόνες. Τα συστήματα αυτά υιοθετήθηκαν από τον επαγγελματικό τομέα, δημιουργώντας με αυτόν τον τρόπο, νέο ενδιαφέρον για τη μηχανική μάθηση. Το 1981 ο Τζέραλντ Ντετζόνγκ εισάγει την έννοια της εκμάθησης βασισμένη στη μάθηση (explanation based learning), στην οποία ένας υπολογιστής αναλύει τα δεδομένα εκπαίδευσης και δημιουργεί γενικούς κανόνες που επιτρέπουν την απόρριψη των λιγότερο σημαντικών δεδομένων. Το 1985 ο Τέρι Σεϊνόφσκι εφευρίσκει το NETtalk, ένα τεχνητό νευρικό δίκτυο το οποίο μαθαίνει να προφέρει λέξεις με τον ίδιο τρόπο εκμάθησης ενός παιδιού. Την δεκαετία του 90 η τεχνολογία της μηχανικής μάθησης μετακινείται από την εστίαση στην ανάπτυξη μοντέλων βασισμένων στην γνώση σε μοντέλα που βασίζονται στην μάθηση μέσω επεξεργασίας μεγάλων ποσοτήτων δεδομένων. Οι επιστήμονες αρχίζουν να δημιουργούν προγράμματα που αναλύουν μεγάλες ποσότητες δεδομένων και εξάγουν συμπεράσματα από τα αποτελέσματά τους. Το 1997 ο υπολογιστής Deep Blue της IBM, νικά τον πρωταθλητή σκακιού Γκάρι Κασπάροφ. Από το 2006 έως και σήμερα έχει σημειωθεί μία έκρηξη στην εμπορευματοποίηση της τεχνητής νοημοσύνης. Το 2006 ο Τζέφρι Χίντον εφηύρε τη φράση “Deep Learning” για να εξηγήσει τις νέες αρχιτεκτονικές των βαθιών νευρωνικών δικτύων ικανών να μάθουν πολύ καλύτερα από άλλα μοντέλα. Η αύξηση της υπολογιστικής δύναμης των υπολογιστών μαζί με την μεγάλη αφθονία των διαθέσιμων δεδομένων ξαναζωντάνεψαν το πεδίο της μηχανικής μάθησης. Πολλές επιχειρήσεις στρέφουν τις εταιρείες τους προς την συλλογή δεδομένων και ενσωματώνουν τη μηχανική μάθηση στις διαδικασίες, τα προϊόντα και τις υπηρεσίες τους, προκειμένου να αποκτήσουν πλεονεκτήματα έναντι του ανταγωνισμού τους. [9]

2.3 Εφαρμογές

Η μηχανική μάθηση υποστηρίζει ήδη μια σειρά εφαρμογών ενώ η δυναμική της φαίνεται να επεκτείνεται πολύ περισσότερο. Η εξέλιξη που έχει τα τελευταία χρόνια έχει δώσει τη δυνατότητα να επεκταθούν οι εφαρμογές της σε πάρα πολλούς τομείς. Δεν είναι τυχαίο το γεγονός ότι πλέον πολλές επιχειρήσεις και οργανισμοί συλλέγουν δεδομένα είτε για να τα αξιοποιήσουν με κάποιον τρόπο οι ίδιοι, είτε για να τα διαθέσουν σε τρίτους για αντίστοιχους λόγους, ανάμεσα σε αυτούς τους λόγους βρίσκεται και η δημιουργία μοντέλων μηχανικής μάθησης. Η μηχανική μάθηση είναι ικανή να συντελέσει στην βελτίωση της αποτελεσματικότητας των προϊόντων και των υπηρεσιών, μιας εταιρίας ή ενός οργανισμού, μέσω της αυξημένης ακρίβειας και των ουσιαστικότερων προσαρμοστικών λύσεων που θα αναδείξει μέσω των προβλέψεων της. Λόγω των διαταραχών που επιφέρει η τεχνολογία σε πολλά επιχειρηματικά πεδία, εξαιτίας των ταχύτατων αλλαγών της, η μηχανική μάθηση θα μπορούσε να ανταποκριθεί είτε αλλάζοντας τον τρόπο με τον οποίο οι επιχειρήσεις είναι οργανωμένες, είτε επηρεάζοντας τα διάφορα επιχειρηματικά μοντέλα που χρησιμοποιούν.

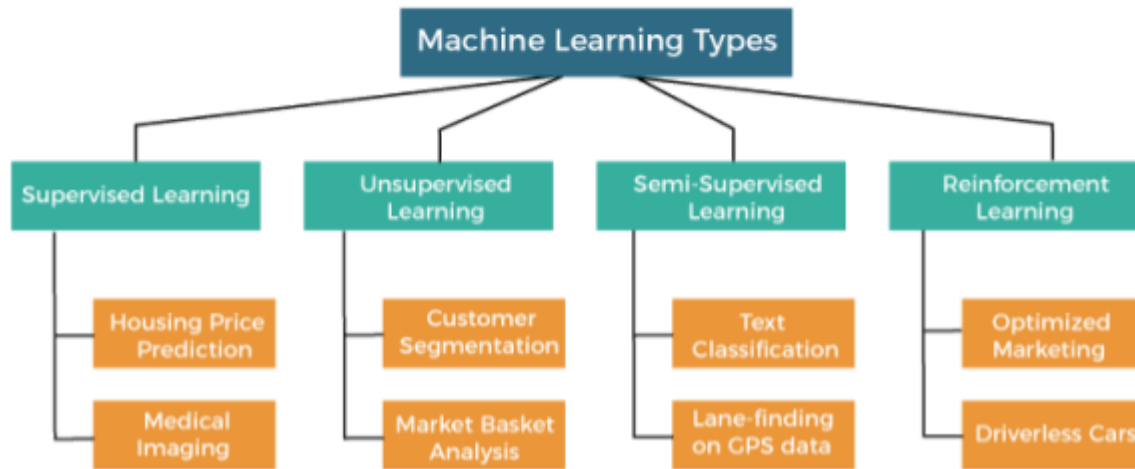
Στην υγειονομική περίθαλψη, η μηχανική μάθηση θα μπορούσε να βοηθήσει παρέχοντας ακριβέστερες διαγνώσεις και πολλές άλλες αποτελεσματικές υπηρεσίες υγειονομικής περίθαλψης, μέσω προηγμένης ανάλυσης που βελτιώνει τη λήψη αποφάσεων. Χαρακτηριστικό παράδειγμα αποτελεί η περίπτωση της διάγνωσης ή εξάπλωσης καρκινικών κυττάρων σε αρκετά πρώιμο στάδιο μέσω “image recognition”. Επίσης, άλλες τεχνικές μηχανικής μάθησης, όπως η επεξεργασία φυσικής γλώσσας (natural language processing), μπορούν να συμβάλλουν στη καλύτερη λήψη αποφάσεων από πλευράς ιατρικού προσωπικού, επιτρέποντας στους υπολογιστές να επεξεργάζονται γραπτές ή προφορικές πληροφορίες από διαφορετικές πηγές. Στην εκπαίδευση, η μηχανική μάθηση μπορεί να υποστηρίξει νέους τρόπους παράδοσης διδακτικού υλικού, ειδικά στο διαδικτυακό περιβάλλον και μπορεί να βοηθήσει τους εκπαιδευτικούς να παρέχουν εξατομικευμένη μάθηση σε μαθητές. Στο τομέα των μεταφορών και της εφοδιαστικής αλυσίδας, αναδεικνύεται η παρουσία των αυτόνομων οχημάτων, τα οποία πρέπει να είναι σε θέση να αναγνωρίζουν ένα μεγάλο μέρος των περιβαλλοντικών χαρακτηριστικών που τα περιβάλλουν όπως εμπόδια, οδικές πινακίδες, πεζούς και άλλα οχήματα. Το εύρος και η μεταβλητότητα των παραπάνω χαρακτηριστικών υποδηλώνει ότι δεν είναι μόνο εφικτό αλλά και αναγκαίο να επινοηθούν αυστηροί κανόνες κωδικοποίησης, οι οποίοι να καθορίζουν τον τρόπο συμπεριφοράς των οχημάτων. Στην αυτόνομη οδήγηση οι εφαρμογές της μηχανικής μάθησης

μπορούν να βοηθήσουν με ανίχνευση των εμποδίων και αποφυγή τους. Μέσω αλγορίθμων που επεξεργάζονται τα στοιχεία που συλλέγουν οι κάμερες, οι αισθητήρες και ότι άλλο χρησιμοποιείται για των αναγνώριση του περιβάλλοντος του οχήματος. Με την χρήση της μηχανικής μάθησης στο μάρκετινγκ μπορούμε να αναγνωρίσουμε το προφίλ των καταναλωτών ως προς τις συνθήκες τους και τα ενδιαφέροντα τους σε συνδυασμό με διάφορα άλλα χαρακτηριστικά τους. Με αυτόν τον τρόπο η ικανοποίηση των πελατών μπορεί να βελτιωθεί και αυτό να αποφέρει τα ανάλογα στις επιχειρήσεις. Αλλά και στον τομέα της διαφήμισης, με την εξατομίκευση των διαφημίσεων καταφέρνουμε να προσελκύσουμε πιθανούς πελάτες πιο στοχευμένα. Η μηχανική μάθηση εφαρμόζεται ήδη στο τραπεζικό σύστημα και στην οικονομία γενικότερα. Παράδειγμα αποτελούν τραπεζικές εφαρμογές εντοπισμού ασυνήθιστων ή ύποπτων συναλλαγών καθώς και συστημάτων χειρόγραφης αναγνώρισης, που επιτρέπει σε αυτοματοποιημένες ταμειακές μηχανές την αναγνώριση επιταγών που έχουν κατατεθεί. Όσον αφορά την ενέργεια η μηχανική μάθηση μπορεί να χρησιμοποιηθεί για την βελτιστοποίηση των ενεργειακών υποδομών. Μπορεί να αναλύσει μοτίβα ενεργειακής χρήσης καθώς και να τα αξιοποιεί κατάλληλα για το σχεδιασμό συστημάτων τα οποία δύναται να ανταποκριθούν πιο αποτελεσματικά σε υψηλότερα αιτήματα του τομέα αυτού. Στο τομέα της βιομηχανοποίησης, προσφέρει ευκαιρίες αυτοματοποιημένων διαδικασιών και βελτιώνει την αποδοτικότητα τους. Επιπλέον δημιουργεί εξατομικευμένα προϊόντα και εφαρμογές προληπτικής συντήρησης. [9]

2.4 Τύποι μηχανικής μάθησης.

Η μηχανική μάθηση δεν είναι απλά ένα πρόβλημα διαχείρισης μεγάλου όγκο δεδομένων, είναι κομμάτι της τεχνητής νοημοσύνης. Για να είναι ένα σύστημα νοήμων, πρέπει να έχει την ικανότητα να μαθαίνει μέσα σε ένα διαρκώς μεταβαλλόμενο περιβάλλον. Εάν το σύστημα έχει αυτήν την ικανότητα να μαθαίνει και να προσαρμόζεται σε αυτές τις μεταβολές, τότε ο σχεδιαστής του συστήματος δεν χρειάζεται να προβλέπει και να παρέχει συνεχώς λύσεις για αυτές τις μεταβαλλόμενες καταστάσεις. Η μηχανική μάθηση είναι η εκτέλεση ενός προγράμματος με τη χρήση υπολογιστή με σκοπό τη μεγιστοποίηση των παραμέτρων ενός μοντέλου χρησιμοποιώντας δεδομένα εκπαίδευσης ή προηγούμενη εμπειρία και διαχωρίζεται ανάλογα με το είδος των δεδομένων του συνόλου εκπαίδευσης στις παρακάτω κατηγορίες:

μάθηση με επίβλεψη, μάθηση χωρίς επίβλεψη, μάθηση με ημι-επίβλεψη και μάθηση με ενίσχυση. [10]



Εικόνα 3: Τύποι μηχανικής μάθησης.

2.4.1 Μάθηση με επίβλεψη.

Η μάθηση με επίβλεψη (Supervised Learning) είναι μία από τις πιο κοινές και επιτυχημένες μεθόδους που χρησιμοποιούνται στη μηχανική μάθηση, η οποία χρησιμοποιείται όταν θέλουμε να προβλέψουμε ένα σίγουρο αποτέλεσμα από μια δεδομένη είσοδο. Ονομάζεται μάθηση με επίβλεψη ή επιβλεπόμενη μάθηση γιατί το μοντέλο μαθαίνει από ένα σύνολο εκπαίδευσης δημιουργώντας έτσι ένα άλλο μοντέλο, όπου με βάση αυτό προσαρμόζεται στο νέο σύνολο δεδομένων για να προβλέψει τα αποτελέσματα. Στη μάθηση με επίβλεψη εισάγονται στο υπολογιστικό πρόγραμμα οι τιμές των πειραματικών δεδομένων εισόδου μαζί με τις τιμές που αντιστοιχούν στα αποτελέσματα τους. Το πρόγραμμα προσαρμόζει τις συναρτήσεις έτσι ώστε τα δεδομένα εισόδου να είναι αντίστοιχα των τιμών των αποτελεσμάτων που αρχικά δόθηκαν. Από τη διαδικασία αυτή προκύπτει ένας γενικός κανόνας (μοντέλο) το οποίο μπορεί να εφαρμοστεί και σε δεδομένα εισόδου με άγνωστη έξοδο. Υπάρχουν δύο υποκατηγορίες επιβλεπόμενης μάθησης, η κατηγοριοποίηση και η παλινδρόμηση. Αλγόριθμοι οι οποίοι εφαρμόζουν μάθηση με επίβλεψη είναι η γραμμική παλινδρόμηση (Linear Regression), τα νευρωνικά δίκτυα (Neural Networks), η μάθηση κατά Bayes (Bayesian Learning), τα δέντρα απόφασης (Decision Trees), ο k πλησιέστεροι γείτονες (k Nearest Neighbors), η λογιστική παλινδρόμηση (Logistic Regression), τα τυχαία δάση (Random Forest) κ.α.

Η κατηγοριοποίηση είναι μια από τις βασικότερες εργασίες της μηχανικής μάθησης. Κύριος στόχος της κατηγοριοποίησης είναι η πρόβλεψη μιας κατηγοριοποιημένης ετικέτας κατηγοριών βασισμένη σε προηγούμενες παρατηρήσεις. Στην περίπτωση της κατηγοριοποίησης, αρχικά δίνονται τα δεδομένα εισόδου ταξινομημένα ή με διακεκριμένες τιμές εξόδου. Αφού το υπολογιστικό σύστημα κατασκευάσει το ζητούμενο μοντέλο, μπορεί να κατηγοριοποιήσει τα δεδομένα εισόδου σε δύο ή περισσότερες κατηγορίες (κλάσεις). Ως επί το πλείστον, η κατηγοριοποίηση είναι χωρισμένη σε δυαδική κατηγοριοποίηση, όπου ο αλγόριθμος μαθαίνει μια σειρά από κανόνες για τη διάκριση των ετικετών μεταξύ δύο πιθανών κατηγοριών, και σε πολλαπλή κατηγοριοποίηση, η οποία κατηγοριοποιεί τα δεδομένα σε περισσότερες από δύο κατηγορίες. Η κατηγοριοποίηση μπορεί να περιγραφεί ως μια διαδικασία με δύο στάδια, την εκμάθηση και την κατηγοριοποίηση/ εκπαίδευση. Ο δεύτερος τύπος της μάθησης με επίβλεψη είναι η μέθοδος της παλινδρόμησης (Regression), η οποία χρησιμοποιείται για την πρόβλεψη συνεχών αποτελεσμάτων. Στόχος αυτής της μεθόδου είναι να προβλεφθεί ένας συνεχής αριθμός, αφού αρχικά δοθεί ένας αριθμός από τον παράγοντα της μεταβλητής καθώς και μια συνεχής μεταβλητή αποτελέσματος. Στην περίπτωση της παλινδρόμησης τα πειραματικά δεδομένα εισόδου έχουν ως τιμές (στόχο) μη διακριτικές τιμές. Με βάση τα παραπάνω προσπαθούμε να βρούμε μια σχέση μεταξύ των μεταβλητών, η οποία θα μας επιτρέψει να προβλέψουμε το αποτέλεσμα. Η χρήση των μοντέλων που προκύπτουν από τέτοιου τύπου αλγόριθμους χρησιμοποιούνται για την πρόβλεψη αριθμητικών τιμών.

2.4.2 Μάθηση χωρίς επίβλεψη.

Ο δεύτερος τύπος μηχανικής μάθησης είναι η μάθηση χωρίς επίβλεψη ή μη επιβλεπόμενη μάθηση (Unsupervised Learning). Σε αντίθεση με την μάθηση με επίβλεψη όπου γνωρίζαμε εξ αρχής τη σωστή απάντηση, στην μη επιβλεπόμενη μάθηση δεν είναι γνωστές οι τιμές στόχος και ερχόμαστε αντιμέτωποι με μη κατηγοριοποιημένα δεδομένα ή με δεδομένα που έχουν άγνωστη δομή. Χρησιμοποιώντας αυτή την τεχνική, μπορούμε να εξερευνήσουμε τη δομή των δεδομένων και να εξάγουμε σημαντική πληροφορία χωρίς την βοήθεια ενός γνωστού αποτελέσματος. Στη μάθηση χωρίς επίβλεψη ο αλγόριθμος δέχεται απλά τα δεδομένα και ζητείται η εξαγωγή γνώσης από αυτά. Αλγόριθμοι που εφαρμόζουν μάθηση χωρίς επίβλεψη είναι η ανάλυση κυρίων συνιστωσών (Principal Components Analysis), η τυποποίηση (Standardization), η κανονικοποίηση (Normalization) κ.α. Η μάθηση χωρίς επίβλεψη έχει δύο υποκατηγορίες, τον μετασχηματισμό δεδομένων και τη συσταδοποίηση.

Η χρήση αλγορίθμων για τη δημιουργία νέων αναπαραστάσεων του συνόλου δεδομένων ονομάζεται μετασχηματισμός (Transformation) δεδομένων. Οι αναπαραστάσεις αυτές είναι πιο εύκολες να κατανοηθούν από τους ανθρώπους ή τους αλγορίθμους μηχανικής μάθησης, έτσι ώστε να είναι εφικτή η καλύτερη σύγκριση των δεδομένων. Οι πιο κοινή εφαρμογή αλγορίθμων μετασχηματισμού είναι η μείωση διαστάσεων (Dimensionality Reduction). Αυτό οφείλεται στο ότι τα αρχικά δεδομένα τα οποία χρησιμοποιούνται και επεξεργάζονται έχουν αρκετές διαστάσεις, πράγμα το οποίο μπορεί να παρουσιάσει πρόκληση όταν υπάρχει περιορισμένος χώρος αποθήκευσης και υπολογιστικής απόδοσης των αλγορίθμων μηχανικής μάθησης. Η χρήση της μείωσης διαστάσεων γίνεται κυρίως κατά τη διαδικασία της προεπεξεργασίας όπου αφαιρείται ο θόρυβος από τα δεδομένα, καθώς και η συμπίεση των δεδομένων σε ένα μικρότερο διάστημα τιμών, διατηρώντας όμως το μεγαλύτερο κομμάτι της πληροφορίας. Η μέθοδος της συσταδοποίησης (Clustering) είναι μια διερευνητική τεχνική ανάλυσης δεδομένων, που μας επιτρέπει να οργανώσουμε μια στοίβα από πληροφορίες σε υποομάδες (clusters), χωρίς όμως να υπάρχει κάποια προηγούμενη γνώση των μελών της ομάδας στην οποία οργανώνονται. Κάθε υποομάδα η οποία μπορεί να προκύψει κατά τη διάρκεια της ανάλυσης, ορίζεται από μια ομάδα αντικειμένων που έχουν κάποια κοινά χαρακτηριστικά. Ανάλογα με τη μέθοδο που θα χρησιμοποιηθεί για τον καθορισμό των υποομάδων, οι αλγόριθμοι κατηγοριοποιούνται σε έξι είδη. Διαιρετική συσταδοποίηση (Partitional Clustering), ιεραρχική συσταδοποίηση (Hierarchical Clustering), συσταδοποίηση βασισμένη στην πυκνότητα (Density-based Clustering), συσταδοποίηση βασισμένη σε πλέγμα (Grid-based Clustering), συσταδοποίηση υποχώρων (Subspace Clustering) και ασαφής συσταδοποίηση (Fuzzy Clustering).

2.4.3 Μάθηση με ημι-επίβλεψη.

Η μάθηση με ημι-επίβλεψη (Semi-supervised Learning) είναι ένας συνδυασμός των δύο προηγούμενων τύπων μάθησης, όπου κάποιες από τις τιμές στόχος (κλάσεις) των προτύπων δεν είναι γνωστές. Το σύστημα μάθησης λαμβάνει ένα σύνολο δεδομένων εκπαίδευσης που αποτελείται από μικρό πλήθος δεδομένων με γνωστές κλάσεις και μεγάλο πλήθος δεδομένων χωρίς γνωστές κλάσεις και στη συνέχεια παράγει προβλέψεις για νέα δεδομένα. Τα δεδομένα χωρίς κλάση, όταν χρησιμοποιούνται σε συνδυασμό με ένα μικρό πλήθος δεδομένων με κλάση μπορούν να επιφέρουν σημαντική βελτίωση στην ακρίβεια της μάθησης. Με αυτόν τον τύπο μάθησης μπορεί να μειωθεί ο χρόνος και το κόστος υλοποίησης, καθώς επίσης σε μεγάλο όγκο δεδομένων η επισήμανση με κλάσεις στα δεδομένα που δεν έχουν κλάση,

μπορεί να αποφευχθεί. Η ημι-επιβλεπόμενη μάθηση συχνά εφαρμόζεται σε προβλήματα όπου είναι εύκολη η συλλογή δεδομένων χωρίς να είναι γνωστές οι κλάσεις, ενώ αντίθετα τα δεδομένα με γνωστές κλάσεις είναι δύσκολο να αποκτηθούν, είτε λόγω του μεγάλου χρόνου που απαιτείται, είτε λόγω του μεγάλου κόστους. Διάφοροι τύποι προβλημάτων όπως η ταξινόμηση, η πρόβλεψη τιμής και η ταξινόμηση με βάση το κριτήριο μπορούν να αντιμετωπιστούν ως προβλήματα μάθησης με ημι-επίβλεψη.

2.4.4 Μάθηση με ενίσχυση.

Άλλο ένα είδος μηχανικής μάθησης είναι η μάθηση με ενίσχυση (Reinforcement Learning). Σκοπός του συστήματος είναι η μεγιστοποίηση της αριθμητικής τιμής της ανταμοιβής. Δεν υπάρχει αρχική γνώση για το πως θα επιτευχθεί ο στόχος. Το μοντέλο αλληλεπιδρά μέσα σε ένα δυναμικό περιβάλλον προσπαθώντας να φτάσει στην κατάσταση στόχο μαθαίνοντας συνεχώς από τις τιμές ανταμοιβής και τιμωρίας. Η μάθηση με ενίσχυση χρησιμοποιεί τρεις μεθόδους (components), τον πράκτορα (agent), το περιβάλλον (environment) και την ενέργεια (action). Ο πράκτορας είναι η οντότητα η οποία μαθαίνει και παίρνει διάφορες αποφάσεις, ενώ οτιδήποτε άλλο είναι το περιβάλλον. Μεταξύ του πράκτορα και του περιβάλλοντος υπάρχει συνεχής αλληλεπίδραση, όπου ο πράκτορας επιλέγει διάφορες ενέργειες που θα πραγματοποιήσει το περιβάλλον, παρουσιάζοντας του με αυτόν το τρόπο καινούργιες καταστάσεις στον πράκτορα. Το περιβάλλον δίνει στον πράκτορα διάφορες ανταμοιβές, όπου με βάση και το σκοπό του συστήματος, ο πράκτορας προσπαθεί να μεγιστοποιήσει μακροπρόθεσμα. Πιο συγκεκριμένα, ο πράκτορας βρίσκεται σε ένα περιβάλλον το οποίο περιγράφεται από διάφορες πιθανές καταστάσεις (states) και κάθε φορά που εκτελεί μια ενέργεια σε αυτή την κατάσταση, λαμβάνει μια ανταμοιβή (reward). Αυτός ο τύπος μάθησης χρησιμοποιείται σε προβλήματα σχεδιασμού ή για να μάθει ένα σύστημα να παίζει κάποιο παιχνίδι.

2.5 Αλγόριθμοι μηχανικής μάθησης.

Η κατηγοριοποίηση αποτελεί μια από τις βασικότερες τεχνικές εξόρυξης δεδομένων, η οποία εξετάζοντας τα γνωρίσματα ενός στιγμιότυπου το αντιστοιχεί σε μια προκαθορισμένη κλάση. Ένα μοντέλο κατηγοριοποίησης κατασκευάζεται λαμβάνοντας ως σύνολο εκπαίδευσης ένα πλήθος ταξινομημένων δεδομένων και χρησιμοποιώντας κατάλληλους αλγορίθμους καταφέρνει να κατηγοριοποίηση σωστά τις εισόδους με άγνωστα δεδομένα. Κάποιοι από τους βασικούς αλγόριθμους κατηγοριοποίησης της μηχανικής μάθησης είναι ο αλγόριθμος K-πλησιέστερων γειτόνων (K-Nearest Neighbors), η μηχανή υποστήριξης διανύσματος (Support Vector Machine), τα δέντρα αποφάσεων (Decision Trees), απλοϊκός κατηγοριοποιητής Bayes (Naive Bayes Classifier) και η διαδικασία ταξινόμησης με Gauss (Gaussian Process Classifier). [11]

2.5.1 K-Nearest Neighbours

Ο αλγόριθμος K-Nearest Neighbors ή σε συντομία K-NN, ανήκει στην ευρύτερη οικογένεια αλγορίθμων μάθησης που βασίζονται στα παραδείγματα (instance based learning), και αντί να εκτελεί ρητή γενίκευση, συγκρίνει τις νέες περιπτώσεις δεδομένων με περιπτώσεις που εμφανίζονται στην εκπαίδευση (training data) και έχουν αποθηκευτεί στην μνήμη. Ο K-NN αποτελεί τον απλούστερο αλγόριθμο μηχανικής μάθησης. Κατά την εφαρμογή του, το μοντέλο αποθηκεύει στη μνήμη όλα τα δεδομένα εκπαίδευσης. Για την πρόβλεψη ενός νέου δεδομένου (data point), ο αλγόριθμος βρίσκει με το κοντινότερο σημείο από τα δεδομένα εκπαίδευσης σε σύγκριση με το νέο εισαγόμενο παράδειγμα (data instance), δηλαδή τον κοντινότερο του γείτονα. Για να πραγματοποιηθεί η διαδικασία αυτή χρειάζεται να προσδιορίσουμε το k , δηλαδή των αριθμό των κοντινότερων γειτόνων που θα χρησιμοποιήσουμε για την κατηγοριοποίηση, αλλά και την μετρική απόσταση για τον υπολογισμό της απόστασης από αυτούς. Ένα από τα πλεονεκτήματα αυτού του αλγορίθμου είναι ότι μπορεί να εφαρμοστεί ακόμη και αν ο αριθμός των δειγμάτων του συνόλου εκπαίδευσης είναι πολύ μικρός. Επίσης, λόγω της απλής σχετικά διαδικασίας που ακολουθεί, προγραμματίζετε πιο εύκολα. Ωστόσο υπάρχουν και μερικά μειονεκτήματα, όπως το ότι τα δεδομένα αντιμετωπίζονται ως ισόβαρα ενώ το πιο πιθανό είναι κάθε δεδομένο να έχει την δική του κλίμακα και επομένως να είναι απαραίτητη η κανονικοποίηση του. Δυσκολία

επισης αντιμετωπιζουμε οταν τα δεδομενα μας εχουν κατηγορικες τιμες, αλλα και στην επιλογη του καταλληλου αριθμου k των γειτονων.

2.5.2 Support Vector Machine

Ο αλγοριθμος Support Vector Machine ή σε συντομία SVM αποτελεί μια σύγχρονη αποτελεσματική προσέγγιση της επίλυσης ζητημάτων κατηγοριοποίησης. Ουσιαστικά αποτελεί έναν αλγόριθμο για μάθηση χωρίς επίβλεψη. Ο SVM μπορεί να χρησιμοποιηθεί είτε για προβλήματα παλινδρόμησης, είτε κατηγοριοποίησης. Λειτουργεί με βάση τον υπολογισμό περιθωρίων (margin calculation). Στον αλγόριθμο αυτό, κάθε δεδομένο αναπαρίσταται ως ένα σημείο μέσα σε ένα χώρο πολλαπλών διαστάσεων, όπου n είναι ο αριθμός των χαρακτηριστικών-μεταβλητών που έχουν στο σετ δεδομένων. Η αξία του κάθε επιμέρους χαρακτηριστικού ορίζεται από την θέση του μέσα στο χώρο αυτό. Ο τρόπος κατηγοριοποίησης των δεδομένων σε κλάσεις βασίζεται στη δημιουργία μιας γραμμής (hyper plane) και δουλεύει μεγιστοποιώντας τις αποστάσεις μεταξύ του κοντινότερου σημείου, αυτές οι αποστάσεις καλούνται περιθώρια. Ο SVM προσπαθεί να βρει μια υπερεπιφάνεια (hypersurface), ώστε να μπορέσει να διαχωρίσει στον χώρο τα θετικά και τα αρνητικά δεδομένα. Έτσι μπορεί να κατηγοριοποιήσει περιπτώσεις που είναι παρόμοιες και όχι πανομοιότυπες με κάποιο δεδομένο εκπαίδευσης.

2.5.3 Decision Trees

Τα δέντρα απόφασης (Decision Trees) είναι συναρτήσεις κατηγοριοποίησης οι οποίες αναπαρίστανται ως δέντρα. Αυτή η δεντρική δομή καθιστά πιο εύκολα κατανοητή την διαδικασία κατηγοριοποίησης και γι' αυτό το λόγο χρησιμοποιείται πολύ συχνά σε προβλήματα κατηγοριοποίησης. Οι κόμβοι του δέντρου αναπαριστούν τα χαρακτηριστικά και ο κόμβος που βρίσκεται στο υψηλότερο επίπεδο ονομάζεται ρίζα του δέντρου. Οι κόμβοι συνδέονται μεταξύ τους με κλαδιά τα οποία αναπαριστούν τις δυνατές τιμές του κόμβου “πατέρα” τους. Οι κόμβοι από τους οποίους δεν ξεκινά κάποιο κλαδί λεγονται φύλλα και παίρνουν το όνομα κάποιας κλάσης. Ένα δεδομένο κατηγοριοποιείται ξεκινώντας από την ρίζα του δέντρου, η οποία αναπαριστά ένα χαρακτηριστικό και αφού ελέγξουμε ποια τιμή έχει το αντίστοιχο χαρακτηριστικό του δεδομένου προχωράμε στο κατάλληλο κλαδί. Συνεχίζουμε την ίδια διαδικασία και για τον κόμβο στον οποίο καταλήγει το κλαδί που ακολουθήσαμε. Η παραπάνω διαδικασία σταματάει όταν φτάσουμε σε κάποιο φύλλο, το

οποίο μας υποδεικνύει την κλάση του δεδομένου. Έχοντας ένα σύνολο απόφασης μπορούμε να δημιουργήσουμε διαφορετικά δέντρα. Το κριτήριο που τα διαφοροποιεί είναι σε ποιους κόμβους θα βάλουμε τα γνωρίσματα.

2.5.4 Naive Bayes Classifier

Ο απλοϊκός κατηγοριοποιητής βασίζεται στο θεώρημα του Bayes και επιπλέον υποθέτει ότι τα γνωρίσματα είναι ανεξάρτητα μεταξύ τους. Το θεώρημα αυτό υπολογίζει την δεσμευμένη πιθανότητα ενός γεγονότος A , λαμβάνοντας υπόψη ένα προηγούμενο γεγονός. Έτσι προκύπτει ένας πιο απλός κατηγοριοποιητής αλλά αρκετά αποτελεσματικός. Έστω ένα στιγμιότυπο X που περιγράφεται από τα γνωρίσματα (x_1, x_2, \dots, x_n) και αναζητούμε την κλάση στην οποία ανήκει. Οι κατηγορίες της κλάσης συμβολίζονται με (k_1, k_2, \dots, k_m) . Με βάση το θεώρημα του Bayes αρκεί να υπολογίσουμε για κάθε κατηγορία της κλάσης τη πιθανότητα ένα στιγμιότυπο X να ανήκει στην κατηγορία αυτή. Όποια κατηγορία έχει μεγαλύτερη πιθανότητα $P(k_d | X = (x_1, x_2, \dots, x_n))$ για $d \in (1, \dots, m)$, τότε λέμε ότι το στιγμιότυπο ανήκει σε αυτή. Ο συγκεκριμένος αλγόριθμος έχει ως πλεονεκτήματα ότι η εκπαίδευση μπορεί να πραγματοποιηθεί ακόμη και αν έχουν μικρό αριθμό δειγμάτων. Έχει ως μειονέκτημα ότι θεωρεί τα γνωρίσματα ανεξάρτητα μεταξύ τους ενώ στην πραγματικότητα πολλές φορές συναντάμε δεδομένα που είναι αλληλένδετα.

2.5.5 Gaussian Process Classifier

Οι Γκαουσιανές μέθοδοι είναι μια γενίκευση της κατανομής πιθανοτήτων του Gauss και μπορούν να χρησιμοποιηθούν ως βάση για εξελιγμένους, μη παραμετρικούς αλγόριθμους μηχανικής μάθησης για ταξινόμηση και παλινδρόμηση. Είναι ένας τύπος μοντέλου πυρήνα (kernel), όπως ο Support Vector Machine και σε αντίθεση με αυτόν, είναι σε θέση να προβλέψει σε ικανοποιητικό βαθμό τις πιθανότητες της κάθε κλάσης. Η σωστή επιλογή και η διαμόρφωση του πυρήνα θεωρείται η καρδιά της μεθόδου. Οι Γκαουσιανές διαδικασίες μπορούν να οριστούν απευθείας στο χώρο της συνάρτησης, δεν είναι τόσο περιοριστικές όσο τα παραμετρικά μοντέλα σχετικά με τον υποθετικό χώρο στον οποίο η μάθηση λαμβάνει μέρος. Ένας αλγόριθμος μηχανικής μάθησης που περιλαμβάνει μια Γκαουσιανή διαδικασία χρησιμοποιεί τη lazy learning μέθοδο και ένα μέτρο της ομοιότητας μεταξύ των σημείων (kernel function), για να προβλέψει την τιμή για ένα άορατο σημείο από τα δεδομένα εκπαίδευσης. Οι συναρτήσεις κατανομής πιθανοτήτων του Gauss συνοψίζουν την κατανομή τυχαίων μεταβλητών, ενώ οι διαδικασίες συνοψίζουν τις ιδιότητες των συναρτήσεων. Ως εκ τούτου, οι Γκαουσιανές διαδικασίες μπορούν να νοηθούν ως ένα επίπεδο αφαίρεσης πάνω στις Γκαουσιανές συναρτήσεις. [12]

Κεφάλαιο 3: Τεχνολογίες

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δοθεί μια μικρή περιγραφή/ εισαγωγή στις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής εργασίας. Το πρώτο βήμα της εργασίας είναι η περισυλλογή, επεξεργασία και προετοιμασία των εικόνων πριν τις χρησιμοποιήσουμε για την εκπαίδευση του αλγορίθμου. Αυτό το βήμα πραγματοποιήθηκε στο Roboflow και μέσω αυτού του framework δημιουργήθηκε ένα dataset. Αφού ολοκληρώθηκε αυτό το βήμα, χρησιμοποίησα αυτό το dataset για την εκπαίδευση του αλγορίθμου YOLOv5 ώστε να αναγνωρίζει τις έξι διαφορετικές κλάσεις (κατηγορίες) εικόνων που δημιούργησα στο Roboflow. Το τρέξιμο του αλγορίθμου και η εκπαίδευση του έγινε στο environment (περιβάλλον) του Google Colab. Μέσω του Google Colab και σε συνδυασμό με το Tensorboard μπορούσα δω και να καταγράψω τις απώλειες της εκπαίδευσης και την απόδοση της σε γράφους και βάση αυτών να βγάλω συμπεράσματα σχετικά με το τι μπορώ να αλλάξω ώστε ο αλγόριθμος να αναγνωρίζει και να κατηγοριοποιεί μεγαλύτερο ποσοστό εικόνων. Αφού τελείωσε η εκπαίδευση πήρα τον εκπαιδευμένο πλέον αλγόριθμο και τον έτρεξα τοπικά στον υπολογιστή μου ώστε να πραγματοποιήσω διάφορα τεστ για την αποτελεσματικότητα του αλγορίθμου. Για να γίνει αυτό έπρεπε πρώτα να εγκαταστήσω στον υπολογιστή το PyTorch και όλες τις βιβλιοθήκες που χρειάζεται ώστε να μπορεί να τρέξει το YOLOv5 τοπικά. Μετά την εγκατάσταση όλων των απαραίτητων εργαλείων, ενεργοποίησα το PyTorch μέσω του CMD.exe Prompt το οποίο είναι εφαρμογή του Anaconda Navigator.

3.2 Roboflow

Το Roboflow είναι μια πλατφόρμα υπολογιστικής όρασης που επιτρέπει στους χρήστες να δημιουργούν μοντέλα υπολογιστικής όρασης γρηγορότερα και με μεγαλύτερη ακρίβεια μέσω της παροχής καλύτερων τεχνικών συλλογής δεδομένων, προεπεξεργασίας και εκπαίδευσης

μοντέλων. Παρέχει δημόσια σύνολα δεδομένων, καθώς επίσης επιτρέπει στους χρήστες να ανεβάζουν δικά τους προσαρμοσμένα σύνολα δεδομένων, να βάζουν annotations (σχολιασμούς) στις εικόνες, να τροποποιούν των προσανατολισμό των εικόνων, να αλλάζουν το μέγεθος τους, να τροποποιούν την αντίθεση στα χρώματα της εικόνας και να εκτελούν αύξηση δεδομένων (data augmentation). Το Roboflow δέχεται διάφορες μορφές annotation.

Επίσης ο χρήστης μπορεί να χρησιμοποιήσει το Roboflow για την εκπαίδευση και την ανάπτυξη (deploy) του μοντέλου, αυτή η δυνατότητα παρέχεται στην επί πληρωμή έκδοση της πλατφόρμας.



Εικόνα 4: Roboflow και οι δυνατότητες του.

Μια εργασία μπορεί να συντονιστεί και διαμοιραστεί σε ομάδες εντός της πλατφόρμας. Για την εκπαίδευση των μοντέλων, υπάρχουν ήδη πολλές βιβλιοθήκες μοντέλων, όπως οι EfficientNet, MobileNet, YOLO, TensorFlow, PyTorch κ.λπ. Επιπλέον υπάρχουν διάφορες επιλογές ανάπτυξης και οπτικοποίησης του μοντέλου, καλύπτοντας το σύνολο των τελευταίων τεχνολογιών. Το Roboflow χρησιμοποιείται σε διάφορες βιομηχανίες όρασης υπολογιστών σε περιπτώσεις χρήσης όπως, για ανίχνευση διαρροής αερίου, ανίχνευση αγριόχορτων σε φυτά, συντήρηση αεροπλάνων, δορυφορικές εικόνες, αυτοοδηγούμενα αυτοκίνητα, μετρητές κυκλοφορίας και πολλά άλλα. Τα βήματα που πραγματοποιούνται στο Roboflow για την ανίχνευση αντικειμένων είναι φόρτωση του συνόλου δεδομένων (dataset loading), προσθήκη ετικετών στις εικόνες (labeling), οργάνωση των δεδομένων (organise), διεργασία τους (process), εκπαίδευση (train), ανάπτυξη (deploy) και εμφάνιση των αποτελεσμάτων (display). Όπως αναφέρθηκε και παραπάνω, ο χρήστης μπορεί να φορτώσει το δικό του σύνολο δεδομένων ή να χρησιμοποιήσει κάποιο από τα δημόσια σύνολα δεδομένων που προσφέρει η πλατφόρμα. Τα δημόσια σύνολα δεδομένων συνοδεύονται με τα annotations τους, ενώ αν ο χρήστης επιθυμεί να χρησιμοποιήσει το δικό του σύνολο δεδομένων θα πρέπει να πραγματοποιήσει το labeling ο ίδιος. Labeling στην ουσία είναι η

διαδικασία πρόσθεσης των annotations δηλαδή η προσθήκη πλαισίων οριοθέτησης γύρω από το αντικείμενο της εικόνας που θέλουμε να κατηγοριοποιήσουμε. Στην διεργασία του συνόλου δεδομένων εμπλέκονται διάφορα βήματα, όπως ο έλεγχος υγείας (health check) του συνόλου κατά των οποίου ελέγχετε αν το σύνολο δεδομένων δεν είναι αρκετά ισορροπημένο για την δημιουργία μοντέλου κατηγοριοποίησης, και άλλες τέτοιες πληροφορίες σχετικά με το σύνολο των δεδομένων. Για την εκπαίδευση το Roboflow παρέχει αναλογία διαχωρισμού του συνόλου δεδομένων σε υποσύνολο εκπαίδευσης, υποσύνολο επικύρωσης και υποσύνολο τεστ (train-valid-test split ratio). Η βιβλιοθήκη μοντέλων που παρέχει μας επιτρέπει να επιλέξουμε ανάμεσα σε διάφορα μοντέλα εκπαίδευσης και είναι διαθέσιμα για χρήση απευθείας στο σημειωματάριο (notebook) του Google Colab. Κατά την ανάπτυξη μπορούμε να εκτελέσουμε το μοντέλο στο σύννεφο (cloud) ή στο σύστημα και τα βάρη (weights) που παρείχαν τα καλύτερα αποτελέσματα αποθηκεύονται αυτόματα. Τελος, κατά την εμφάνιση των αποτελεσμάτων ο χρήστης μπορεί να δει τα αποτελέσματα της εκπαίδευσης απευθείας πάνω στις εικόνες που είχε επιλέξει ως το υποσύνολο του τεστ. [13]

3.3 Google Colab

Το Google Colab αποτελεί έργο της Google Research, είναι ένα δωρεάν περιβάλλον (environment) που βασίζεται στο Jupyter και μας επιτρέπει να δημιουργήσουμε σημειωματάρια προγραμματισμού (programming notebooks) και να εκτελούμε κώδικα Python σε ένα πρόγραμμα περιήγησης. Μπορεί επίσης να εκτελεί άλλα εργαλεία τρίτων που βασίζονται στην Python και σε πλαίσια (frameworks) μηχανικής μάθησης, όπως τα Pandas, PyTorch, Tensorflow, Keras, Monk, OpenCV και άλλα. Ένα notebook προγραμματισμού είναι ένας τύπος shell ή kernel με την μορφή επεξεργαστή κειμένου όπου μπορούμε να γράψουμε και να εκτελέσουμε κώδικα. Τα δεδομένα που απαιτούνται για την επεξεργασία στο Google Colab μπορούν να προστεθούν στο Google Drive ή να εισαχθούν από οποιαδήποτε πηγή στο διαδίκτυο. Το Jupyter είναι ένας οργανισμός λογισμικού ανοικτού κώδικα που αναπτύσσει και υποστηρίζει Jupyter notebooks για διαδραστικούς υπολογιστές. Το Google Colab δεν απαιτεί διαμόρφωση για να χρησιμοποιηθεί και παρέχει δωρεάν πρόσβαση σε GPU. Μία από τις κύριες λειτουργίες του Google Colab είναι ότι επιτρέπει σε οποιονδήποτε να μοιράζεται κώδικα, μαθηματικές εξισώσεις, οπτικοποιήσεις δεδομένων (data visualization), επεξεργασία δεδομένων (καθαρισμός και μετασχηματισμός τους), αριθμητικές προσομοιώσεις, μοντέλα μηχανικής μάθησης και πολλά άλλα έργα.

Το Google Colab έχει μοναδικά και κρίσιμα χαρακτηριστικά. Παρέχει ένα δωρεάν περιβάλλον Jupyter, έχει προεγκατεστημένες βιβλιοθήκες, φιλοξενείται εξ ολοκλήρου στο Google Cloud, οι χρήστες δεν χρειάζεται να το εγκαταστήσουν σε διακομιστές, τα

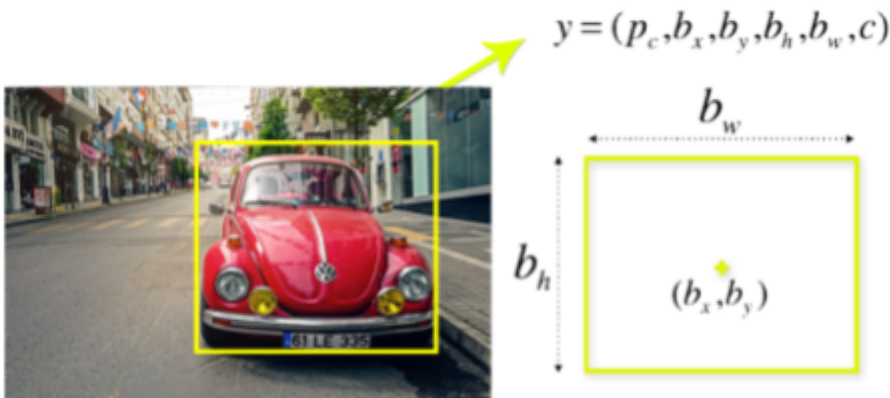
σημειωματάρια αποθηκεύονται αυτόματα στο Google Drive του χρήστη, παρέχει Jupyter notebooks που βασίζονται σε προγράμματα περιήγησης (browser based), είναι εντελώς δωρεάν και προσφέρει ισχύ GPU (Graphical Processing Unit) και TPU (Tensor Processing Unit), επίσης σε περίπτωση που ο χρήστης χρειάζεται περισσότερους πόρους τότε μπορεί να χρησιμοποιήσει την επαγγελματική έκδοση Colab Pro. Τέλος παρέχει ομαλή ενσωμάτωση με το GitHub. [14]

3.4 YOLO

Το YOLO είναι ένας αλγόριθμος που χρησιμοποιεί νευρωνικά δίκτυα για να παρέχει αναγνώριση αντικειμένων σε πραγματικό χρόνο. Ο αλγόριθμος αυτός είναι γνωστός λόγω της ταχύτητας και της ακρίβειας του και έχει χρησιμοποιηθεί σε ποικίλες εφαρμογές για τον εντοπισμό πινακίδων, ανθρώπων κ.α.

YOLO είναι η συντομογραφία του όρου “You Only Look Once”. Είναι ένας αλγόριθμος που εντοπίζει και αναγνωρίζει διάφορα αντικείμενα σε μια εικόνα. Η αναγνώριση των αντικειμένων στο YOLO αντιμετωπίζεται σαν ένα πρόβλημα παλινδρόμησης και ως αποτέλεσμα δίνει την πιθανότητα που έχει το αντικείμενο προς αναγνώριση να ανήκει σε μια κατηγορία. Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί συνελκτικά νευρωνικά δίκτυα (convolutional neural networks) ή CNN σε συντομογραφία, για την αναγνώριση αντικειμένων σε πραγματικό χρόνο. Όπως φαίνεται και από το όνομα ο αλγόριθμος απαιτεί μόνο ένα πέρασμα μέσω ενός νευρωνικού δικτύου για τον εντοπισμό των αντικειμένων. Το οποίο σημαίνει ότι η πρόβλεψη σε μια ολόκληρη εικόνα γίνεται με έναν μόνο κύκλο του αλγορίθμου. Το CNN χρησιμοποιείται για την πρόβλεψη ποικίλων πιθανοτήτων μιας κατηγορίας και περιμετρικών πλαισίων ταυτόχρονα. Ο αλγόριθμος YOLO είναι σημαντικός λόγω της ταχύτητας, της υψηλής ακρίβειας και των μαθησιακών δυνατοτήτων που παρέχει. Πιο συγκεκριμένα, ο χρόνος εντοπισμού βελτιώνεται επειδή μπορεί να κάνει προβλέψεις σε πραγματικό χρόνο, το μοντέλο YOLO αποτελεί μια τεχνική πρόβλεψης που παρέχει ακριβή αποτελέσματα με ελάχιστα λάθη στο παρασκήνιο. Επίσης ο αλγόριθμος έχει εξαιρετικές μαθησιακές ικανότητες που του επιτρέπουν να μάθει το παρουσιαστικό των αντικειμένων και να εφαρμόζει αυτήν την γνώση στην αναγνώριση αντικειμένων.

Ο συγκεκριμένος αλγόριθμος δουλεύει χρησιμοποιώντας τις εξής τεχνικές: residual blocks, bounding box regression και intersection over union. Στην τεχνική residual blocks, η εικόνα διαχωρίζεται σε διάφορα πλέγματα. Κάθε πλέγμα εντοπίζει αντικείμενα που εμφανίζονται μέσα σε αυτό. Στην τεχνική bounding box regression, ένα περιμετρικό πλαίσιο είναι μια γραμμή που τονίζει ένα αντικείμενο στην εικόνα.



Εικόνα 5: Περιμετρικό πλαίσιο.

Κάθε τέτοιο πλαίσιο αποτελείται από πλάτος, ύψος, κατηγορία και το κέντρο του. Στην παραπάνω εικόνα παρουσιάζεται ένα παράδειγμα περιμετρικού πλαισίου, το οποίο παρουσιάζεται με κίτρινο περίγραμμα. Το YOLO χρησιμοποιεί μια παλινδρόμηση περιμετρικού πλαισίου για να προβλέψει το ύψος, πλάτος, κέντρο και κατηγορία του αντικειμένου. Η τεχνική Intersection over union ή IOU είναι ένα φαινόμενο στην αναγνώριση αντικειμένων που περιγράφει το πως το πραγματικό πλαίσιο και το πλαίσιο πρόβλεψης αλληλεπικαλύπτονται. Το YOLO χρησιμοποιεί το IOU για να παρέχει ένα πλαίσιο το οποίο καλύπτει τέλεια περιμετρικά το αντικείμενο. Κάθε πλεγματοειδές κελί είναι υπεύθυνο για να προβλέψει τα περιμετρικά πλαίσια και το ποσοστό σιγουριάς τους. Το IOU είναι ίσο με 1 έναν το περιμετρικό πλαίσιο που βρέθηκε είναι το ίδιο με το πραγματικό. Αυτή η τεχνική μπορεί να απορρίψει περιμετρικά πλαίσια που δεν είναι ίδια ή ίσα με τα πραγματικά.

Το YOLOv5 είναι η πρώτη έκδοση της οικογένειας YOLO που γράφτηκε σε PyTorch αντί για Darknet. Το Darknet είναι μια αρκετά ευέλικτη πλατφόρμα, αλλά δεν έχει χτιστεί με γνώμονα τα περιβάλλοντα παραγωγής. Επίσης, έχει μικρότερο αριθμό χρηστών. Επειδή το YOLOv5 εφαρμόζεται αρχικά στο PyTorch, επωφελείται από το καθιερωμένο οικοσύστημα του. Έτσι η υποστήριξη του είναι απλούστερη και η ανάπτυξη του ευκολότερη. Επιπλέον, ως ευρύτερα γνωστό ερευνητικό πλαίσιο μπορεί να κάνει την χρήση του YOLOv5 ευκολότερη λόγω της ευρύτερης ερευνητικής κοινότητας. Τέλος, το YOLOv5 είναι εκπληκτικά πιο γρήγορο. Σε πείραμα που χρησιμοποιήθηκε διαπιστώθηκαν χρόνοι συμπερασμάτων έως 0.007 δευτερόλεπτα ανά εικόνα, δηλαδή 140 frames (περιμετρικά πλαίσια) το δευτερόλεπτο. Αντίθετα η έκδοση YOLOv4 πέτυχε 50 frames το δευτερόλεπτο. [15] [16]

3.5 TensorBoard

Το TensorBoard είναι ένα εργαλείο που επιτρέπει την απεικόνιση οποιονδήποτε στατιστικών στοιχείων ενός νευρωνικού δικτύου, όπως οι εικόνες, οι γράφοι και οι παράμετροι εκπαίδευσης, απώλεια (loss), ακρίβεια (accuracy) και βάρη (weights). Και αυτό είναι πολύ χρήσιμο για την κατανόηση της ροής των tensors στο γράφημα, και κατά συνέπεια να οδηγήσει στον εντοπισμό σφαλμάτων και στη βελτιστοποίηση του μοντέλου. Το TensorBoard λειτουργεί διαβάζοντας τα αρχεία συμβάντων (event files), στα οποία το

TensorFlow γράφει τα δεδομένα σύνοψης, δηλαδή τα δεδομένα που πρόκειται να οπτικοποιηθούν. Συνήθως είναι εγκατεστημένο στο TensorFlow και για να εκτελεστεί απλά πρέπει να εκτελεστεί πρώτα η ακόλουθη εντολή: `tensorboard --logdir=[dir]`, όπου το `[dir]` είναι η θέση των αρχείων συμβάντος. Η βασική ιδέα πίσω από το TensorBoard είναι ότι τα νευρωνικά δίκτυα μπορεί να είναι κάτι γνωστό ως μαύρο κουτί και χρειαζόμαστε ένα εργαλείο για να επιθεωρήσουμε τι υπάρχει μέσα σε αυτό το κουτί. Αυτό το εργαλείο βοηθά στην κατανόηση των εξαρτήσεων μεταξύ των λειτουργιών και του τρόπου που υπολογίζονται τα βάρη, επίσης εμφανίζει τη συνάρτηση απώλειας και πολλές άλλες χρήσιμες πληροφορίες. [17]

3.6 PyTorch

Η PyTorch είναι μια βιβλιοθήκη ανοιχτού κώδικα που έχει σχεδιαστεί με γνώμονα την Python και έχει κατασκευαστεί για έργα μηχανικής μάθησης. Αναπτύχθηκε από το ινστιτούτο Idiap Research στο πανεπιστήμιο της Νέας Υόρκης, ΗΠΑ. Ειδικεύεται στην αυτόματη διαφοροποίηση, στους υπολογισμούς για tensor (tensor computations), στην επιτάχυνση της GPU και στα βαθιά νευρωνικά δίκτυα. Αυτά τα χαρακτηριστικά την καθιστούν ιδιαίτερα κατάλληλη για εφαρμογές μηχανικής μάθησης όπως η βαθιά μάθηση (deep learning). Χρησιμοποιείται τόσο για κατασκευή πρωτότυπων για έρευνες όσο και στην ανάπτυξη και παραγωγή τους.

Η PyTorch βασίζεται στο Torch, το οποίο είναι ένα πρώιμο framework για βαθιά μάθηση, από το οποίο παίρνει τις δυνατότητες βαθιάς μάθησης και τις μεταφέρει στο περιβάλλον της Python. Η πρώτη προσπάθεια ενσωμάτωσης της βαθιάς μάθησης στην Python έγινε με την Keras. Η οποία αναπτύχθηκε το 2015 και παρουσίασε ένα API (Application Programming Interface) για την εκπαίδευση νευρωνικών δικτύων με Python. Η TensorFlow δημιουργήθηκε επίσης το 2015 και έγινε η πλευρά του διακομιστή (server-side) της Keras. Όμως το πρώιμο API της TensorFlow δεν ήταν ιδιαίτερα κατάλληλο για την Python. Το PyTorch κυκλοφόρησε τον Σεπτέμβριο του 2016 και ήταν λύση στα προβλήματα που είχαν οι ερευνητές με την Keras και την TensorFlow. Η PyTorch διόρθωσε μερικές ελλείψεις που υπήρχαν στην Keras και παρείχε ένα πιο διαισθητικό API. Σε αυτό το σημείο, η PyTorch είναι έτοιμη για παραγωγή, επιτρέποντας στον χρήστη να μεταβεί με ευκολία μεταξύ των καταστάσεων ανυπομονησίας (eager mode) και γράφων με το TorchScript και να επιταχύνει τη διαδρομή προς την παραγωγή με το TorchServer. Το torch.distributed backend επιτρέπει κλιμακούμενη κατανεμημένη εκπαίδευση και βελτιστοποίηση απόδοσης στην έρευνα και την παραγωγή, επίσης ένα πλούσιο οικοσύστημα εργαλείων και βιβλιοθηκών επεκτείνει το PyTorch και υποστηρίζει την ανάπτυξη στην υπολογιστική όραση, την επεξεργασία φυσικής γλώσσας και πολλά άλλα. Τέλος η PyTorch υποστηρίζεται από μεγάλες πλατφόρμες cloud, όπως οι Amazon Web Services (AWS), Google Cloud Platform (GCP) και Microsoft Azure. Μερικές σημαντικές ιδιότητες της PyTorch είναι, ότι είναι γραμμένη σε Python και μπορεί να ενσωματωθεί με δημοφιλείς βιβλιοθήκες της Python όπως η NumPy για επιστημονικούς υπολογισμούς, η SciPy και η Cython για την μεταγλώττιση της Python σε C για καλύτερη

απόδοση. Επειδή η σύνταξη και η χρήση της είναι παρόμοια με την Python, η PyTorch είναι σχετικά εύκολη για προγραμματιστές Python. Η γλώσσα scripting, που ονομάζεται TorchScript, είναι εύκολη στη χρήση και ευέλικτη όταν βρίσκεται σε κατάσταση ανυπομονησίας (eager mode). Υποστηρίζει CPU, GPU και παράλληλη επεξεργασία, καθώς και κατανεμημένη εκπαίδευση. Αυτό σημαίνει ότι η υπολογιστική εργασία μπορεί να κατανεμηθεί μεταξύ πολλών πυρήνων CPU σε πολλαπλές μηχανές. Η PyTorch υποστηρίζει δυναμικά υπολογιστικά γραφήματα, επιτρέποντας την αλλαγή της συμπεριφοράς του δικτύου κατά το χρόνο εκτέλεσης. Αυτό παρέχει ένα σημαντικό πλεονέκτημα ευελιξίας σε σχέση με τη πλειονότητα των frameworks μηχανικής μάθησης, τα οποία απαιτούν τα νευρωνικά δίκτυα να ορίζονται ως στατικά αντικείμενα πριν το χρόνο εκτέλεσης. Τέλος, η PyTorch έχει ένα καλά αναγνωρισμένο σύνολο API που μπορούν να χρησιμοποιηθούν για την επέκταση της βασικής λειτουργικότητας. [18] [19]

3.7 Anaconda Navigator

Το Anaconda Navigator είναι ένα γραφικό περιβάλλον εργασίας για την χρήση κοινών προγραμμάτων Python χωρίς να χρειάζεται ο χρήστης να χρησιμοποιήσει γραμμές εντολών, για την εγκατάσταση πακέτων και τη διαχείριση των περιβαλλόντων του. Επιτρέπει στον χρήστη να ανοίξει εφαρμογές και να διαχειρίζεστε εύκολα πακέτα conda, περιβάλλοντα και κανάλια επίσης χωρίς να χρησιμοποιήσει την γραμμή εντολών. Το Navigator μπορεί να αναζητήσει πακέτα στο Anaconda Cloud ή σε ένα τοπικό Anaconda repository και είναι διαθέσιμο για Windows, macOS και Linux. Το Navigator είναι ένας εύκολος τρόπος εργασίας με πακέτα και περιβάλλοντα χωρίς ο χρήστης να χρειάζεται να πληκτρολογήσει εντολές conda. Επιπλέον, ο χρήστης μέσα από τον Navigator μπορεί να το χρησιμοποιήσει για να βρει τα πακέτα που θέλει, να τα εγκαταστήσει σε ένα περιβάλλον, να εκτελέσει τα πακέτα και να τα ενημερώσει. Αυτό βοηθά τους επιστήμονες δεδομένων να διασφαλίζουν ότι κάθε έκδοση κάθε πακέτου έχει όλες τις εξαρτήσεις που χρειάζεται για να λειτουργήσει σωστά. Το Anaconda είναι ένα πακέτο λογισμικού ανοιχτού κώδικα που περιλαμβάνει Jupyter, Spyder και άλλα εργαλεία για επεξεργασία δεδομένων μεγάλης κλίμακας, ανάλυση δεδομένων και επιστημονικούς υπολογισμούς. Η R και η Python υποστηρίζονται από την Anaconda. Το Jupyter Notebook είναι ένα δημοφιλές σύστημα που μας επιτρέπει να ενσωματώνουμε τον κώδικα, το περιγραφικό κείμενο, την έξοδο, τα γραφικά και τις διαδραστικές διεπαφές σε ένα ενιαίο αρχείο σημειωματάριου που μπορούμε να επεξεργαστούμε, να προβάλουμε και να χρησιμοποιήσουμε σε ένα πρόγραμμα περιήγησης ιστού. Η Python υλοποιείται χρησιμοποιώντας το Spyder. Στο Spyder μπορούμε να

χρησιμοποιήσουμε `openv` για Python. Το σύστημα διαχείρισης πακέτων `anaconda` παρακολουθεί τις εκδόσεις πακέτων. Η `Anaconda` στοχεύει στο να διευκολύνει τη διαχείριση και την ανάπτυξη πακέτων σε επιστημονικά πεδία όπως η επιστήμη δεδομένων, οι εφαρμογές μηχανικής μάθησης, η επεξεργασία δεδομένων μεγάλης κλίμακας και η προγνωστική ανάλυση. Η κύρια διαφορά μεταξύ του `conda` και του `pip` είναι ο τρόπος με τον οποίο διατηρούνται οι εξαρτήσεις των πακέτων, κάτι που αποτελεί βασική δυσκολία για την επιστήμη δεδομένων της Python, αλλά και την βάση της ύπαρξης του `conda`. Η `conda` εξετάζει το τρέχον περιβάλλον και μαζί με τυχόν περιορισμούς έκδοσης, οι οποίοι είναι καθορισμένοι, καθορίζει τον τρόπο εγκατάστασης ενός συμβατού συνόλου εξαρτήσεων και εκδίδει μια προειδοποίηση εάν αυτό δεν είναι δυνατό. Τέλος, τα προσαρμοσμένα πακέτα μπορούν να δημιουργηθούν χρησιμοποιώντας την εντολή `conda build` και να μοιραστούν με άλλους μέσω του `Anaconda Cloud`, του `PyPI` ή άλλων `repositories`.

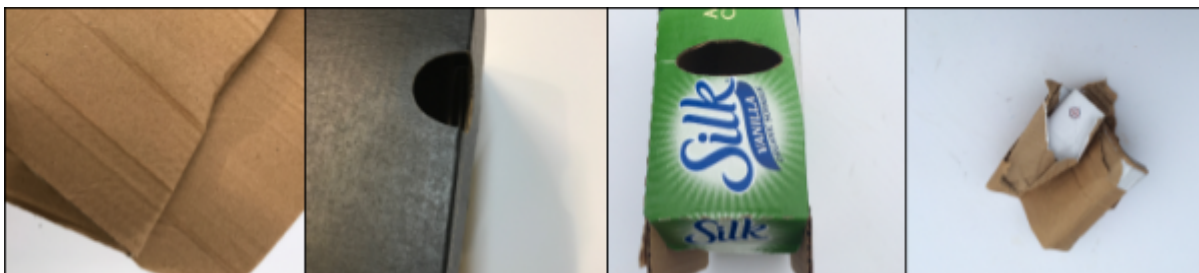
[20][21]

Κεφάλαιο 4: Υλοποίηση

4.1 Δημιουργία συνόλου δεδομένων

Στην παρούσα εργασία το αρχικό σύνολο δεδομένων (dataset) που δημιουργήθηκε αποτελείται από τρεις χιλιάδες εικόνες έξι διαφορετικών κατηγοριών και η κάθε μια κατηγορία αποτελεί το ένα έκτο του dataset. Αρχικά δημιούργησα το dataset με εικόνες βρήκα στο διαδίκτυο και από άλλα dataset που βρήκα στο Kaggle [22] το οποίο είναι μια διαδικτυακή κοινότητα για επιστήμονες δεδομένων και λάτρεις της μηχανικής μάθησης. Στην συνέχεια, χώρισα τις εικόνες σε φακέλους ανά κατηγορία στον υπολογιστή μου και τις ανέβασα στο Roboflow, το οποίο έχει αναφερθεί στο παραπάνω κεφάλαιο. Στο Roboflow οι εικόνες ανέβηκαν επίσης ανά κατηγορία ώστε να είναι πιο εύκολο το κομμάτι της προσθήκης ετικετών στις εικόνες. Για καλύτερη κατανόηση του συνόλου δεδομένων παρακάτω παρατίθενται εικόνες με δείγματα από τις κατηγορίες μας, οι οποίες περιλαμβάνουν χαρτόνι, γυαλί, μέταλλο, χαρτί, πλαστικό και σκουπίδια.

Στην εικόνα 6 βλέπουμε τα διαφορετικά είδη χαρτονιών, αλλά και τις διαφορετικές μορφές και οπτικές γωνίες που χρησιμοποιήθηκαν για την καλύτερη εκπαίδευση του αλγορίθμου.



Εικόνα 6: Δείγμα εικόνων με χαρτόνι.

Στην εικόνα 7 βλέπουμε τα διαφορετικά είδη γυαλιού τα οποία μπορούμε να βρούμε σε ένα σπίτι, αλλά και τις διαφορετικές οπτικές γωνίες που χρησιμοποιήθηκαν για την καλύτερη εκπαίδευση του αλγορίθμου.



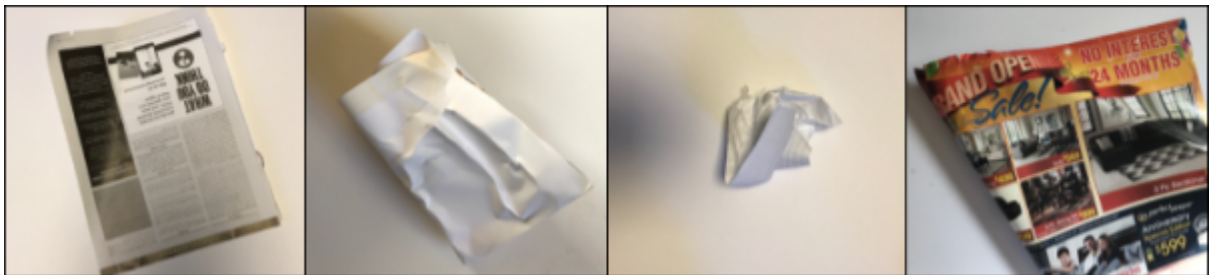
Εικόνα 7: Δείγμα εικόνων με γυαλί.

Στην εικόνα 8 βλέπουμε τα διαφορετικά είδη μέταλλο, τις διαφορετικές μορφές και τις οπτικές γωνίες που χρησιμοποιήθηκαν για την καλύτερη εκπαίδευση του αλγορίθμου.



Εικόνα 8: Δείγμα εικόνων από μέταλλο.

Στην εικόνα 9 βλέπουμε τα διαφορετικά είδη χαρτιού, τα διαφορετικά σχήματα που μπορεί να έχει και τις διαφορετικές οπτικές γωνίες που χρησιμοποιήθηκαν για την καλύτερη εκπαίδευση του αλγορίθμου.



Εικόνα 9: Δείγμα εικόνων από χαρτί.

Στην εικόνα 10 βλέπουμε τα διαφορετικά είδη πλαστικού που μπορούμε να εντοπίσουμε σε ένα σπίτι, και τις διαφορετικές οπτικές γωνίες που χρησιμοποιήθηκαν για την καλύτερη εκπαίδευση του αλγορίθμου.



Εικόνα 10: Δείγμα εικόνων από πλαστικό.

Στην εικόνα 11 βλέπουμε τα διαφορετικά είδη σκουπιδιών, δηλαδή τα μη ανακυκλώσιμα οικιακά απόβλητα. Η εκπαίδευση του αλγορίθμου για να αναγνωρίζει αυτήν την κατηγορία είναι ιδιαίτερα δύσκολη καθώς υπάρχουν πάρα πολλά διαφορετικά είδη αποβλήτων.



Εικόνα 11: Δείγμα εικόνων από σκουπίδια.

4.1.1 Προεπεξεργασία δεδομένων

Η διαδικασία της προεπεξεργασίας δεδομένων (data preprocessing) είναι ένα πολύ σημαντικό βήμα πριν μεταβούμε στο στάδιο της εφαρμογής ενός αλγορίθμου μηχανικής μάθησης. Είναι ένα αναγκαίο βήμα καθώς τα πραγματικά δεδομένα που λαμβάνουμε από τον πραγματικό κόσμο δεν είναι τέλεια αλλά ούτε και ιδανικά για να μπορέσουν να εφαρμοστούν απευθείας σε ένα μοντέλο κατηγοριοποίησης. Σχεδόν πάντα τα δεδομένα τα οποία συλλέγουμε και πρέπει να διαχειριστούμε μπορεί να έχουν πολλά προβλήματα που πρέπει να διορθωθούν και ώστε να μπορέσουν να εξυπηρετήσουν τις δικές μας ανάγκες. Ένα από αυτά τα προβλήματα είναι ο θόρυβος, με την έννοια ότι τα δεδομένα μας μπορεί να περιέχουν λάθη ή ακραίες τιμές. Γι αυτό τον λόγο η προεπεξεργασία των δεδομένων είναι μια απαραίτητη διαδικασία που εξασφαλίζει την εγκυρότητα των αποτελεσμάτων. Η διαδικασία αυτή πραγματοποιείται με σκοπό τη παραγωγή καλύτερης ποιότητας δεδομένων, ώστε η εκπαίδευση να δημιουργήσει αποδοτικότερα συστήματα για την πρόβλεψη των νέων δεδομένων.

Κατά την προεπεξεργασία, αρχικά πραγματοποιείται ένας καθαρισμός των δεδομένων. Σε αυτό το σημείο ελέγχονται τα δεδομένα για σφάλματα τα οποία είτε δημιουργούνται κατά την καταγραφή τους είτε προέρχονται από λανθασμένη ταξινόμηση άλλων ταξινομητών. Στη πρώτη περίπτωση τα δεδομένα ελέγχονται για τυπογραφικά λάθη στις τιμές των μεταβλητών, το διάστημα που ανήκουν και την τυπική απόκλιση των τιμών των μεταβλητών. Στη δεύτερη περίπτωση προκαλεί μεγάλο πρόβλημα στην απόδοση του συστήματος και για το λόγο αυτό πρέπει να εξαλείφονται περιπτώσεις με λανθασμένη ταξινόμηση. Ένα άλλο πρόβλημα για την καθαρότητα των δεδομένων είναι η απουσία τιμών σε μεταβλητές. Υπάρχουν διάφορες μέθοδοι για την λύση αυτού του προβλήματος και η καθεμία, ανάλογα τα δεδομένα, έχει διαφορετικό βαθμό αποδοτικότητας. Επόμενο βήμα είναι η διακριτοποίηση συνεχών μεταβλητών, με σκοπό την κατασκευή διακριτών κατηγοριών ώστε να παράγονται συγκεκριμένοι κανόνες ταξινόμησης για τις συνεχές μεταβλητές. Δύο απλές μέθοδοι διακριτοποίησης μιας συνεχούς μεταβλητής είναι η κατάτμηση, είτε σε συγκεκριμένο πλήθος διαστημάτων ίσου μεγέθους, είτε σε διαστήματα που περιέχουν ίδιο αριθμό περιπτώσεων. Κάθε μέθοδος ενδείκνυται ανάλογα το σύνολο των δεδομένων. Επόμενο βήμα στην προεπεξεργασία των δεδομένων είναι η μείωση του πλήθους των μεταβλητών. Όταν χρησιμοποιούνται πολλές μεταβλητές, έχει ως αποτέλεσμα τον εξαναγκασμό του συστήματος σε πολλούς συναρτησιακούς υπολογισμούς, με αποτέλεσμα να αυξάνεται η υπολογιστική πολυπλοκότητα στη διαδικασία της εκπαίδευσης. Επίσης, αν οι μεταβλητές είναι περισσότερες από το αναγκαίο πλήθος, εμφανίζεται το φαινόμενο του υπερταϊριάσματος, δηλαδή η δημιουργία κανόνων ταξινόμησης που χρησιμοποιούν και περιττές μεταβλητές, με αποτέλεσμα την λάθος ταξινόμηση περιπτώσεων. Για το λόγο αυτό πρέπει να πραγματοποιείται μια επιλογή στις μεταβλητές που θα χρησιμοποιηθούν για την εκπαίδευση. Έτσι, γίνεται ένας διαχωρισμός των μεταβλητών σε σχετικές, άσχετες και περιττές. Οι σχετικές μεταβλητές είναι εκείνες που έχουν μεγάλη επιρροή στην ακρίβεια του ταξινομητή και δεν μπορούν να αποκατασταθούν από καμία άλλη μεταβλητή του συνόλου εκπαίδευσης. Ως άσχετες αναφέρονται εκείνες οι μεταβλητές που δεν έχουν καμία επιρροή στην ακρίβεια του ταξινομητή. Οι περιττές, είναι αυτές που έχουν μεγάλη επιρροή στην ακρίβεια του ταξινομητή αλλά ο ρόλος τους μπορεί να υποκατασταθεί από άλλες μεταβλητές του συνόλου εκπαίδευσης.

Τέλος, στην περίπτωση μεγάλων συνόλων δεδομένων, είναι σκόπιμο να γίνει επιλογή ενός μικρότερου αριθμού των περιπτώσεων με σκοπό την κατασκευή ενός συστήματος όσο το δυνατόν ίδιο σε απόδοση με το αρχικό σύνολο δεδομένων, κάνοντας το όμως αρκετά γρηγορότερο. [23]

Στο δικό μας σύνολο δεδομένων, συλλέξα ίσο αριθμό εικόνων για κάθε κατηγορία που ήθελα να δημιουργήσω και αφού έλεγξα όλες τις εικόνες για θέματα όπως η ποιότητα και ο θόρυβος, αναίβασα τις εικόνες στο RoboFlow ανά κατηγορία. Όταν πλέον όλες οι εικόνες της κατηγορίας είχαν αναίβει, ξεκίνησα την διαδικασία του annotation ή labeling δηλαδή έβαζα ετικέτες χειροκίνητα σε κάθε εικόνα αυτής της κατηγορίας και μετά τις πρόσθετα στο νέο σύνολο δεδομένων. Επαναλάβα αυτήν την διαδικασία και για τις έξι κατηγορίες μου και στο τέλος αυτής της διαδικασίας το σύνολο δεδομένων μου αποτελούνταν από τρεις χιλιάδες εικόνες οι οποίες ήταν έτοιμες για να χρησιμοποιηθούν από τον αλγόριθμο επιβλεπόμενης μάθησης. Στην παρακάτω εικόνα βλέπουμε την μορφή των εικόνων μετά την ολοκλήρωση της διαδικασίας του annotation.



Εικόνα 12: Παράδειγμα εικόνων με ετικέτες.

4.1.2 Δημιουργία συνόλου δεδομένων προς ανάλυση

Μετά την ολοκλήρωσή της διαδικασίας προσθήκης ετικετών στις εικόνες, πρέπει να χωρίσουμε το σύνολο δεδομένων που δημιουργήσαμε σε τρία υποσύνολα, στο κομμάτι της εκπαίδευσης (training set), στο κομμάτι της επικύρωσης (validation set) και στο κομμάτι της δοκιμής (test set).

Ένα σύνολο δεδομένων που χρησιμοποιείται για τη μηχανική μάθηση πρέπει να χωριστεί στα τρία υποσύνολα που αναφέρθηκαν παραπάνω. Τα δεδομένα εκπαίδευσης αξιοποιούνται για την εκπαίδευση του μοντέλου συμβάλλοντας στον καθορισμό των βέλτιστων παραμέτρων του με σκοπό τη μάθηση από τα δεδομένα. Τα δεδομένα δοκιμών απαιτούνται για την αξιολόγηση του εκπαιδευμένου μοντέλου και την ικανότητα του για γενίκευση. Δηλαδή, η ικανότητα ενός μοντέλου να αναγνωρίζει μοτίβα σε νέα αγνωστά σε αυτό δεδομένα, έχοντας εκπαιδευτεί αποκλειστικά πάνω σε δεδομένα εκπαίδευσης. Η αξιοποίηση διαφορετικών

υποσυνόλων εκπαίδευσης και δοκιμής είναι ζωτικής σημασίας για να αποφευχθεί η υπερβολική μάθηση του μοντέλου (overfitting), η οποία κάνει δύσκολη την γενίκευση του μοντέλου. Τα δεδομένα επικύρωσης παρουσιάζουν καταλυτικό ρόλο στην ρύθμιση και επιλογή του κατάλληλου μοντέλου βάσει των δεδομένων που διαθέτουμε και του προβλήματος που καλούμαστε να επιλύσουμε. Σκοπός ενός συνόλου δεδομένων επικύρωσης είναι η πραγματοποίηση τροποποιήσεων στις υπερπαραμέτρους (hyperparameters) ενός μοντέλου, οι οποίες αφορούν δοκιμές και ρυθμίσεις υψηλού επιπέδου που το μοντέλο δεν μπορεί να μάθει άμεσα από τα δεδομένα εκπαίδευσης. Αυτές οι ρυθμίσεις μπορούν να εκφράσουν το πόσο περίπλοκο είναι ένα μοντέλο και το πόσο γρήγορα εντοπίζει μοτίβα στα δεδομένα. Έτσι τα αποτελέσματα ενός μοντέλου πάνω στα δεδομένα επικύρωση μας δίνουν μια πρώτη εκτίμηση για την απόδοση του μοντέλου. Στην παρούσα εργασία ο διαχωρισμός του συνόλου δεδομένων έγινε με ποσοστό εκπαίδευσης 70% , ποσοστό επικύρωσης 20% και ποσοστό δοκιμής 10%. Τέλος, είναι χαρακτηριστικό ότι όσο μεγαλύτερο είναι το ποσοστό των δεδομένων εκπαίδευσης, τόσο καλύτερη θα είναι και η απόδοση του μοντέλου.

Έπειτα από βήμα του διαχωρισμού του συνόλου δεδομένων το Roboflow δίνει στον χρήστη την δυνατότητα να αυξήσει την ακρίβεια του μοντέλου εφαρμόζοντας κάποιες συγκεκριμένες τεχνικές προεργασίας. Εφάρμοσα επαναπροσδιορισμό μεγέθους σε όλες τις εικόνες ώστε να έχουν όλες το ίδιο μέγεθος 416×416 και ένα “φίλτρο” αυτόματου προσανατολισμού το οποίο απορρίπτει τις περιστροφές των ετικετών και τυποποιεί τη σειρά των εικονοστοιχείων. Το σύνολο δεδομένων μας έχει την ιδιομορφία ότι οι εικόνες του αποτελούνται από εικόνες που έχουν τραβηχτεί σε εσωτερικό χώρο με καλό φωτισμό. Γι’αυτό κρίνεται κατάλληλο να χρησιμοποιηθούν κάποια βήματα αύξησης (augmentation steps) στα οποία κοινοποιούνται οι εικόνες του dataset μας και ξανά προσθέτονται σε αυτό με επιπλέον χαρακτηριστικά. Στην παρακάτω εικόνα εμφανίζονται η προεργασίες που έγιναν όπως αυτές εμφανίζονται στο Roboflow.

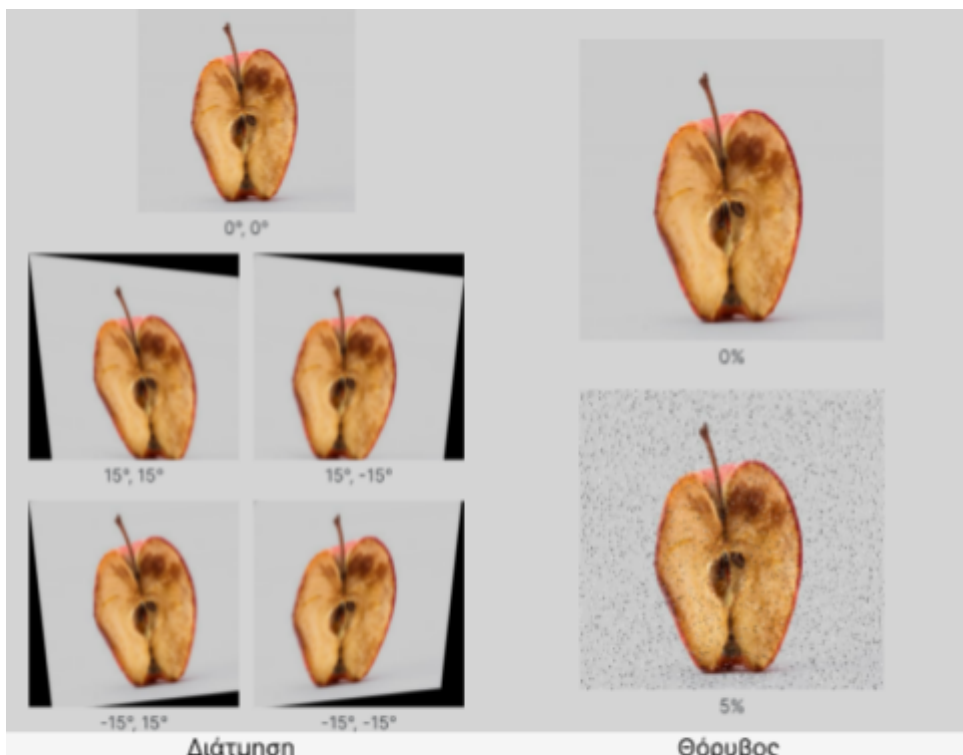
✓ Source Images	Images: 3,000 Classes: 6 Unannotated: 0
✓ Train/Test Split	Training Set: 2.1k images Validation Set: 599 images Testing Set: 308 images
✓ Preprocessing	Auto-Orient: Applied Resize: Stretch to 416×416
✓ Augmentation	Flip: Horizontal Rotation: Between -15° and +15° Shear: ±15° Horizontal, ±15° Vertical Noise: Up to 5% of pixels

Εικόνα 13: Προεργασίες στο Roboflow.

Τα augmentation steps που χρησιμοποίησα είναι η οριζόντια αναστροφή (flip horizontal), περιστροφή -15° και $+15^\circ$, διάτμηση (shear) $\pm 15^\circ$ οριζόντια και $\pm 15^\circ$ κάθετα και θόρυβος εώς 5% των pixel. Στην παρακάτω εικόνα βλέπουμε αυτά τα augmentation steps όπως αυτά εμφανίζονται σε παραδείγματα του Roboflow.



Εικόνα 14: Παράδειγμα αναστροφής και περιστροφής από Roboflow.



Εικόνα 15: Παράδειγμα διάτμησης και θορύβου από Roboflow.

Μετά την ολοκλήρωση αυτών των βημάτων και τις επιπλέον προσθήκες εικόνων, μέσω τις κλωνοποίησης που αναφέρθηκε παραπάνω, το τελικό σύνολο δεδομένων είναι έτοιμο για εκπαίδευση. Το τελευταίο βήμα της δημιουργίας του συνόλου δεδομένων στο Roboflow, είναι η παράγωγή (generate) της έκδοσης του συνόλου δεδομένων με τα χαρακτηριστικά που έχουμε επιλέξει αλλά το Roboflow μας δίνει και την δυνατότητα να πολλαπλασιάσουμε το σύνολο εκπαίδευσης του dataset μας ώστε το τελικό dataset να είναι πιο μεγάλο και έτσι το μοντέλο μας να έχει καλύτερη απόδοση. Μετά την επιλογή του βαθμού πολλαπλασιασμού που επιθυμούμε, μπορούμε πλέον να εξάγουμε (export) το σύνολο δεδομένων μας, εγώ το πολλαπλασίασα επί τρεις και το τελικό dataset μου αποτελείται από 7488 εικόνες. Τέλος το Roboflow δίνει την δυνατότητα στον χρήστη να εξάγει το σύνολο δεδομένο σε τέσσερις διαφορετικές μορφές, ο χρήστης μπορεί είτε να κατεβάσει ολόκληρο το σύνολο δεδομένων σε μορφή zip στον υπολογιστή του, είτε να το αποθηκεύσει σε μορφή url, είτε σε μορφή κώδικα για Jupyter notebook ή για terminal. Εγώ το εξήγαγα σε μορφή κώδικα για Jupyter. [24]

4.2 Περιβάλλον υλοποίησης

Για την υλοποίηση του αλγορίθμου YOLOv5 που χρησιμοποιήθηκε στην εργασία ήταν απαραίτητες οι βιβλιοθήκες Matplotlib, Numpy, OpenCV-Python, SciPy, Torch και Pandas οι οποίες είναι κάποιες από τις αρκετές βιβλιοθήκες που απαιτεί το YOLOv5 και από τις πιο γνωστές. Πιο αναλυτικά στο αρχείο requirements.txt που βρίσκεται στον φάκελο του YOLOv5 στο GitHub αναφέρονται όλες οι βιβλιοθήκες που είναι απαραίτητες για την χρήση αυτού του αλγορίθμου και βρίσκεται στον σύνδεσμο <https://github.com/ultralytics/yolov5/blob/master/requirements.txt>. Οι βιβλιοθήκες αυτές είναι από τις πιο βασικές βιβλιοθήκες της γλώσσας προγραμματισμού Python στον τομέα της μηχανικής μάθησης. Το περιβάλλον προγραμματισμού το οποίο χρησιμοποιήθηκε για την υλοποίηση του πρακτικού μέρους της εργασίας είναι το Google Colab το οποίο προαναφέρθηκε πιο πάνω και μπορεί κανείς να το χρησιμοποιήσει μέσω του συνδέσμου του Google Research https://colab.research.google.com/?utm_source=scs-index.

Η βιβλιοθήκη Matplotlib είναι η βασική βιβλιοθήκη για σχεδιασμό στην Python. Μέσω αυτής της βιβλιοθήκης δίνεται η δυνατότητα να κατασκευαστούν γραφήματα, ιστογράμματα, γραφήματα διασποράς, γραμμικά διαγράμματα κ.α. [25] Η βιβλιοθήκη Numpy είναι ανοιχτή βιβλιοθήκη για την Python. Περιέχει διάφορες έτοιμες για πολυδιάστατους πίνακες και μαθηματικές συναρτήσεις. Επιπρόσθετα, έχει τη δυνατότητα να παράγει ψευδοτυχαίους αριθμούς. Χρησιμοποιεί επίσης αυθαίρετους τύπους δεδομένων επιτρέποντας την ενσωμάτωση τους χωρίς προβλήματα και με ταχύτητα. [26] Το OpenCV-Python είναι μια βιβλιοθήκη της Python που έχει σχεδιαστεί για την επίλυση προβλημάτων υπολογιστικής όρασης. [27] Η βιβλιοθήκη SciPy είναι μια συλλογή μαθηματικών αλγορίθμων και λειτουργιών που βασίζεται στη βιβλιοθήκη Numpy. [28] Το Torch είναι μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα, επίσης είναι και ένα επιστημονικό υπολογιστικό

πλαίσιο με ευρεία υποστήριξη για αλγορίθμους μηχανικής μάθησης που βάζει τις GPU στην πρώτη θέση. [29] Η βιβλιοθήκη Pandas είναι μια ανοιχτή βιβλιοθήκη υψηλής απόδοσης, η οποία είναι βασισμένη στη βιβλιοθήκη NumPy και βοηθάει την Python στην ανάλυση και την μοντελοποίηση των δεδομένων. Είναι βασισμένη πάνω σε μια δομή δεδομένων που ονομάζεται DataFrame και η οποία μοιάζει με πίνακα. Έτσι μπορούν να εκτελεστούν διάφορα ερωτήματα της γλώσσας προγραμματισμού SQL. [30]

Από το GitHub του YOLOv5 μπορούμε να μεταβούμε κατευθείαν στο περιβάλλον του Google Colab με τον βασικό κώδικα της Python που χρειαζόμαστε να είναι ήδη αποθηκευμένος σε αυτό. Για να δημιουργήσουμε δικό μας notebook ώστε να μπορούμε να αποθηκεύσουμε τις αλλαγές που κάνουμε στον κώδικα αλλά και την πρόοδο μας πρέπει πρώτα να αποθηκεύσουμε την έτοιμη έκδοση του Google Colab στο Google Drive μας και αυτόματα στο πρόγραμμα περιήγησης μας, ανοίγει μια νέα καρτέλα με το δικό μας πλέον περιβάλλον του Colab. Αφού ανοίξουμε το δικό μας περιβάλλον πρέπει να συνδεθούμε, στην ουσία να ενεργοποιήσουμε, την GPU ώστε να μπορούμε να τρέξουμε τις εντολές. Πρώτο βήμα είναι να κλωνοποιήσουμε τα αρχεία του YOLOv5 από το GitHub και μετά να εγκαταστήσουμε τις βιβλιοθήκες που απαιτεί. Στην παρακάτω εικόνα βλέπουμε τον κώδικα για αυτά τα δύο πρώτα βήματα.

```
#clone YOLOv5 and
git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
!pip install -qr requirements.txt # install dependencies
!pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images

print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")
```

Εικόνα 16: Κώδικας εγκατάστασης απαραίτητων βιβλιοθηκών.

Αμέσως μετά εισάγουμε την βιβλιοθήκη του Roboflow στο περιβάλλον μας, δημιουργούμε ένα περιβάλλον για το σύνολο δεδομένων που δημιουργήσαμε σε αυτό και τέλος ανεβάζουμε το σύνολο δεδομένων μας στο Google Colab μέσω του κώδικα για Jupyter που πήραμε από το Roboflow όταν εξαγάγαμε το σύνολο δεδομένων μας από αυτό. Αφού ολοκληρωθεί αυτό το βήμα είμαστε πλέον έτοιμοι να ξεκινήσουμε την εκπαίδευση. Στην παρακάτω εικόνα βλέπουμε τον κώδικα για την ένταξη του συνόλου δεδομένων μας στο Google Colab.

```
[ ] from roboflow import Roboflow
rf = Roboflow(model_format="yolov5", notebook="ultralytics")

upload and label your dataset, and get an API KEY here: https://app.roboflow.com/?model=yolov5&ref=ultralytics

[ ] # set up environment
os.environ["DATASET_DIRECTORY"] = "/content/datasets"

[ ] from roboflow import Roboflow
rf = Roboflow(api_key="Jrg5p5n3CwQDdlj2rKSH")
project = rf.workspace("new-workspace-mxcjb").project("trash_detect_500")
dataset = project.version(1).download("yolov5")
```

Εικόνα 17: Κώδικας ένταξης του συνόλου δεδομένων στο Google Colab.

4.3 Εκπαίδευση συνόλου δεδομένων

Μετά την ολοκλήρωση της προετοιμασίας του περιβάλλοντος, σειρά έχει η εκπαίδευση του μοντέλου πάνω στο δικό μας σύνολο δεδομένων. Πριν την εκκίνηση της εκπαίδευσης θα πρέπει πρώτα να ορίσουμε κάποιες παραμέτρους. Οι παράμετροι αυτοί είναι το μέγεθος της εικόνας των δεδομένων μας, το μέγεθος του batch (παρτίδα) το οποίο είναι μια υπερ παράμετρος που ορίζει τον αριθμό των δειγμάτων που πρέπει να επεξεργαστούν πριν την ενημέρωση των εσωτερικών παραμέτρων του μοντέλου. Άλλη μια παράμετρος είναι ο αριθμός των epochs (εποχών) εκπαίδευσης, το οποίο είναι μια υπερ παράμετρος που ορίζει τον αριθμό φορών που ο αλγόριθμος μάθησης θα επεξεργαστεί ολόκληρο το σύνολο δεδομένων μάθησης. [31] Οι υπόλοιποι παράμετροι είναι το data που καθορίζει την τοποθεσία που αποθηκεύεται το σύνολο δεδομένων, το weights (βάρη) που καθορίζει μια διαδρομή προς τα weights για να ξεκινήσει τη μεταφορά μάθησης, το cache το οποίο “κρύβει” εικόνες για πιο γρήγορη εκπαίδευση, επίσης σε αυτό το στάδιο της διαδικασίας πρέπει να επιλέξουμε το μοντέλο του YOLOv5 που θέλουμε να χρησιμοποιήσουμε, τα μοντέλα είναι τα εξής YOLOv5s (s-small-μικρό), YOLOv5m (m-medium-μεσαίο), YOLOv5l(l-large-μεγάλο) και YOLOv5x (x-xlarge-πολύ μεγάλο), τα οποία αντιστοιχούν στο μέγεθος του συνόλου δεδομένων που θέλουμε να χρησιμοποιήσουμε στην εκπαίδευση μας. Από την σωστή επιλογή μοντέλου του YOLOv5 εξαρτάται η ταχύτητα της εκπαίδευσης, αλλά μερικές φορές και τα αποτελέσματα της. Επιπλέον όταν ξεκινάει η εκπαίδευση αρχίζει να μετράει ένας timer έτσι ώστε να ξέρουμε τον συνολικό χρόνο που έκανε το μοντέλο μας να τελειώσει την εκπαίδευση. Στην παρακάτω εικόνα βλέπουμε τον κώδικα με τις παραμέτρους που χρησιμοποίησα για την εκπαίδευση του μοντέλου.

```
[ ] !python train.py --img 416 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --weights yolov5m.pt --cache
```

Εικόνα 18: Παράμετροι εκπαίδευσης.

Λόγω του μεγέθους του συνόλου δεδομένων μας, αποφάσισα να χρησιμοποιήσω του μοντέλο YOLOv5m για την εκπαίδευση του αλγορίθμου. Το μέγεθος των εικόνων στο dataset είχε οριστεί στο Roboflow 416×416 και γι'αυτό το λόγο το ορίσα με το ίδιο μέγεθος και για την εκπαίδευση. Τέλος εκπαίδευσα αυτό μοντέλο για 16 batch και 150 epoch. Μετά την ολοκλήρωση της εκπαίδευσης τρέχουμε την εντολή που βλέπουμε στην εικόνα 19, για να φορτώσουμε απο το TensorBoard τους γράφουν που μας δείχνουν την αξιολόγηση της απόδοσης της εκπαίδευσης και τις απώλειες που είχε.

```
[ ] %load_ext tensorboard
      %tensorboard --logdir runs
```

Εικόνα 19: Κώδικας για φόρτιση γράφων από το TensorBoard.

Αμέσως μετά εκτελούμε το αρχείο detect.py το οποίο κάνει ανίχνευση στις εικόνες που είχαμε ορίσει ως test και αποθηκεύει τα αποτελέσματα, δηλαδή τις εικόνες με τις ετικέτες τους σε έναν φάκελο. Η ανίχνευση γίνεται με βάση τα weights που πήραμε από την ολοκλήρωση της εκπαίδευσης και είναι αποθηκευμένα στο αρχείο best.pt. Κατά το τρέξιμο αυτής της εντολής μπορούμε επίσης να ορίσουμε το μέγεθος της εικόνας που των

αποτελεσμάτων, με την παράμετρο `img` και το ποσοστό σιγουρίας του αποτελέσματος με την παράμετρο `conf`. Η εικόνες που έχουν ποσοστό σιγουρίας μικρότερο από αυτό που έχουμε θέσει, θα αποθηκευτούν χωρίς ετικέτες. Στην παρακάτω εικόνα βλέπουμε την εντολή ανίχνευσης.

```
!python detect.py --weights runs/train/exp/weights/best.pt --img 416 --conf 0.5 --source {dataset.location}/test/images
```

Εικόνα 20: Κώδικας για ανίχνευση στις εικόνες.

Τέλος τρέχουμε την εντολή της εικόνας 21 για να δούμε τα αποτελέσματα μας. Τα αποτελέσματα των γράφων με τις αποδόσεις του μοντέλου και τα αποτελέσματα με τις ανιχνεύσεις στις εικόνες θα τα δούμε πιο αναλυτικά στο κεφάλαιο 5.

```
[ ] import glob
    from IPython.display import Image, display

    for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'):
        display(Image(filename=imageName))
        print("\n")
```

Εικόνα 21: Κώδικας για εμφάνιση αποτελεσμάτων ανίχνευσης.

4.4 Βήματα υλοποίησης

Για την υλοποίηση της εργασίας πραγματοποιήθηκαν τα παρακάτω βήματα, τα οποία αναλύθηκαν σε αυτό το κεφάλαιο. Πιο συγκεκριμένα, πρώτο βήμα ήταν η ανεύρεση και δημιουργία ενός ακατεργαστου συνόλου δεδομένων που κάλυπτε τις ανάγκες μας, σχετικά με την ποικιλία και την ποιότητα των έξι κατηγοριών μας οι οποίες είναι χαρτόνι, γυαλί, μέταλλο, χαρτί, πλαστικό και σκουπίδια. Δεύτερο βήμα, ανέβασμα των εικόνων στο Roboflow και προσθήκη ετικετών σε αυτές. Τρίτο βήμα ήταν η προεπεξεργασία και προετοιμασία του συνόλου δεδομένων μας ώστε να φτάσει στην τελική μορφή η οποία είναι έτοιμη για εκπαίδευση. Τέταρτο βήμα, εξαγωγή του συνόλου δεδομένων απο το Roboflow. Πέμπτο βήμα, εξαγωγή του αλγορίθμου YOLO στο περιβάλλον υλοποίησης Google Colab και εγκατάσταση όλων των απαραίτητων βιβλιοθηκών. Έκτο βήμα, εισαγωγή της βιβλιοθήκης του Roboflow στο Colab και ανέβασμα του συνόλου δεδομένων μας σε αυτό. Έβδομο βήμα, επιλογή παραμέτρων, εκπαίδευση μοντέλου και έλεγχος αποτελεσμάτων. Σε αυτό το βήμα πραγματοποιήθηκαν πολλά διαφορετικά πειράματα ώστε να βρεθεί ο αριθμός των εικόνων ανά κατηγορία αλλά και οι παράμετροι που θα μας έδιναν το πιο αποτελεσματικό μοντέλο. Ογδοο βήμα ήταν η αποθήκευση του μοντέλου στον υπολογιστή και πραγματοποίηση σε αυτόν περαιτέρω πειραμάτων ανίχνευσης σε εικόνες και βίντεο, με την βοήθεια του Anaconda Navigator και του Pytorch. Ένατο βήμα, σύγκριση της απόδοσης των διαφορετικων μοντέλων και επιλογή του πιο αποτελεσματικού.

Κεφάλαιο 5: Αποτελέσματα

5.1 Αποτελέσματα αρχικών μοντέλων

Σε αυτό το κεφάλαιο θα γίνει αναφορά των αποτελεσμάτων ανίχνευσης και των αποδόσεων των διάφορων μοντέλων που εκπαιδεύσαμε, με σκοπό την εύρεση του μοντέλου που έχει την καλύτερη απόδοση στην ανίχνευση των αντικειμένων μας στις εικόνες. Οι αποδόσεις των μοντέλων δίνονται από το TensorBoard και από το YOLO σε μορφή γραφημάτων τα οποία θα αναλυθούν περαιτέρω στα κεφάλαια 5.2 και 5.3. Σε αυτό το στάδιο της εργασίας επικεντρωνόμαστε στο ποσοστό σιγουριάς για το αποτέλεσμα και στον μέσο όρο ακριβείας (mean Average Precision) ή mAP.

Στα εκπαιδευμένα μοντέλα μας τρέξαμε τον κώδικα ανίχνευσης, στο ποσοστό των εικόνων που είχαν οριστεί ως υποσύνολο δοκιμών στο Roboflow, δύο φορές. Την πρώτη φορά με ποσοστό σιγουριάς 0.2 δηλαδή 20% και την δεύτερη φορά με ποσοστό σιγουριάς 0.5 και στις δύο ανιχνεύσεις αν το ποσοστό σιγουριάς σε μια εικόνα ήταν μικρότερο από αυτό που είχαμε θέση τότε η εικόνα δεν αποκτούσε ετικέτα. Το mAP είναι μια μέτρηση για τις μετρήσεις ακριβείας των ανιχνευτών αντικειμένων (object detectors) και υπολογίζει την μέση τιμή ακριβείας για την τιμή ανάκλησης από 0 έως 1. Η τιμή ανάκλησης μετράει το πόσο σωστά είναι τα θετικά μας αποτελέσματα, δηλαδή τα αποτελέσματα που ανίχνευσαν κάποια από τις κατηγορίες μας στην εικόνα. [32]

Εικόνες ανά κατηγορία	80
Μέγεθος batch	16
Αριθμός epochs	150
metrics/mAP_0.5	0.82
metrics/mAP_0.5:0.95	0.62
metrics/precision	0.74
metrics/recall	0.78
train/box_loss	0.016
train/cls_loss	0.001
train/obj_loss	0.0096
Ποσοστό σιγουριάς: conf	Αποτελέσματα ανίχνευσης:
0.2	4/5
0.5	2/5

Πίνακας 1: Μοντέλο με 80 εικόνες ανά κατηγορία.

Το πρώτο μοντέλο μας είχε μόνο ογδόντα εικόνες ανά κατηγορία και εκπαιδεύτηκε με παραμέτρους batch ίσο με 16 και epoch ίσο με 150, αυτά τα νούμερα για τα batch και epoch είναι τα πρότυπα νούμερα που προτείνει το Google Colab για εκπαίδευση με YOLO, με την υποσημείωση ότι αυτά τα νούμερα συνήθως έχουν και τις καλύτερες αποδόσεις. Στον πίνακα 1 βλέπουμε τις αποδόσεις του πρώτου μοντέλου μας, το mAP είναι 0.82 και το μοντέλο ανίχνευσε τέσσερις από τις πέντε εικόνες σωστά, με παράμετρο σιγουριάς 0.2, αλλά μόνο δύο στις πέντε όταν η παράμετρος σιγουριάς ήταν 0.5, το οποίο καθιστά στο μοντέλο μας μη αποδοτικό.

Εικόνες ανά κατηγορία	100	100	100	100	100	100	100	100
Μέγεθος batch	10	12	13	15	16	16	16	18
Αριθμός epochs	200	200	200	200	150	180	200	200
metrics/mAP_0.5	0.9	0.93	0.92	0.89	0.91	0.89	0.88	0.91
metrics/mAP_0.5:0.95	0.63	0.62	0.62	0.62	0.62	0.61	0.61	0.63
metrics/precision	0.85	0.9	0.86	0.84	0.88	0.83	0.83	0.8
metrics/recall	0.83	0.83	0.86	0.82	0.8	0.83	0.81	0.84
train/box_loss	0.012	0.013	0.013	0.014	0.015	0.015	0.014	0.015
train/cls_loss	0.007	0.008	0.007	0.008	0.01	0.008	0.008	0.009
train/obj_loss	0.0088	0.0089	0.009	0.0093	0.0097	0.0094	0.0092	0.0092
Ποσοστό σιγουριάς: conf	Αποτελέσματα ανίχνευσης:							
0.2	7/8	6/8	7/8	7/8	6/8	7/8	7/8	7/8
0.5	7/8	6/8	7/8	7/8	6/8	7/8	7/8	6/8

Πίνακας 2: Μοντέλο με 100 εικόνες ανά κατηγορία

Στο δεύτερο μοντέλο, το σύνολο δεδομένων στο οποίο εκπαιδεύτηκε αποτελείται από εκατό εικόνες ανά κατηγορία. Στον πίνακα 2 βλέπουμε τις αποδόσεις οκτώ εκ δώδεκα εκδοχών αυτού του μοντέλου, οι οποίες ήταν και οι πιο αξιοσημείωτες. Το πιο υψηλό mAP που έχουμε σε αυτό το μοντέλο είναι 0.93 και εμφανίζεται στην εκδοχή με τα 12 batch και 200 epoch, εδώ αν και το mAP είναι το πιο υψηλό σε σχέση με τις άλλες εκδοχές αυτού του μοντέλου, η ανίχνευση δεν έχει από τα μικρότερα θετικά αποτελέσματα του μοντέλου και αναγνωρίζει έξι στις οκτώ εικόνες άσχετα με την παράμετρο ποσοστού σιγουριάς. Αλλά μια αξιοσημείωτη λεπτομέρεια είναι ότι τα ποσοστά ανίχνευσης σχεδόν σε όλες τις εκδόσεις αυτού του μοντέλου είναι ίδια και στις δύο τιμές της παραμέτρου, δηλαδή άσχετα με το αν η παράμετρος σιγουριάς έχει τεθεί ίση με 0.2 είτε με 0.5, η ανίχνευση της κάθε έκδοσης μας δίνει τα ίδια αποτελέσματα.

Εικόνες ανά κατηγορία	130	130	130	130	130	130	130	130
Μέγεθος batch	14	15	16	16	16	18	24	24
Αριθμός epochs	200	200	300	340	380	300	300	370
metrics/mAP_0.5	0.85	0.88	0.9	0.87	0.87	0.88	0.87	0.88
metrics/mAP_0.5:0.95	0.58	0.62	0.63	0.6	0.6	0.61	0.59	0.61
metrics/precision	0.86	0.82	0.82	0.85	0.88	0.85	0.85	0.8
metrics/recall	0.82	0.81	0.81	0.77	0.8	0.81	0.79	0.83
train/box_loss	0.013	0.0014	0.013	0.013	0.013	0.014	0.016	0.015
train/cls_loss	0.008	0.007	0.007	0.007	0.007	0.008	0.09	0.08
train/obj_loss	0.0092	0.0091	0.0087	0.0086	0.0085	0.0089	0.009	0.0087
Ποσοστό σιγουριάς: conf	Αποτελέσματα ανίχνευσης:							
0.2	12/12	12/12	11/12	11/12	11/12	11/12	11/12	11/12
0.5	10/12	10/12	11/12	9/12	9/12	10/12	10/12	10/12

Πίνακας 3: Μοντέλο με 130 εικόνες ανά κατηγορία

Στο τρίτο μοντέλο, το σύνολο δεδομένων στο οποίο εκπαιδεύτηκε αποτελείται από εκατό τριάντα εικόνες ανά κατηγορία. Στον πίνακα 3 βλέπουμε τις αποδόσεις οκτώ εκ των είκοσι δύο εκδοχών αυτού του μοντέλου, οι οποίες ήταν και οι πιο αξιοσημείωτες. Το πιο υψηλό mAP που έχουμε σε αυτό το μοντέλο είναι 0.9 και εμφανίζεται στην εκδοχή με τα 16 batch και 300 epoch. Σε αυτήν την έκδοση ο αριθμός των θετικών αποτελεσμάτων είναι ίδιος και στις δύο παραμέτρους του ποσοστού σιγουριάς. Επίσης, αν και σε άλλες εκδόσεις έχουν σημειωθεί πιο υψηλά θετικά αποτελέσματα, σε αυτήν την έκδοση ο αριθμός παραμένει ο ίδιος ενώ στις άλλες όταν ανεβαίνει το ποσοστό σιγουριάς, ο αριθμός των σωστών αποτελεσμάτων τους πέφτει.

Στο τέταρτο μοντέλο, το σύνολο δεδομένων στο οποίο εκπαιδεύτηκε αποτελείται από διακόσιες εικόνες ανά κατηγορία. Στον πίνακα 4 βλέπουμε τις αποδόσεις των έξι εκδοχών αυτού του μοντέλου. Το πιο υψηλό mAP που έχουμε σε αυτό το μοντέλο είναι 0.84 και εμφανίζεται στην εκδοχή με τα 20 batch και 180 epoch. Σε αυτήν την έκδοση ο αριθμός των θετικών αποτελεσμάτων είναι ο πιο υψηλός ανάμεσα στις εκδόσεις αυτού του μοντέλου. Σε αυτό το μοντέλο όλες οι εκδόσεις έχουν, πάνω κάτω, το ίδιο mAP και κανένας από τους αριθμούς σωστών αποτελεσμάτων ανίχνευσης δεν είναι αξιοσημείωτος.

Εικόνες ανά κατηγορία	200	200	200	200	200	200
Μέγεθος batch	16	16	20	20	20	20
Αριθμός epochs	150	200	180	200	220	250
metrics/mAP_0.5	0.84	0.78	0.84	0.83	0.84	0.84
metrics/mAP_0.5:0.95	0.57	0.52	0.57	0.55	0.58	0.57
metrics/precision	0.81	0.74	0.84	0.82	0.81	0.82
metrics/recall	0.8	0.75	0.78	0.76	0.78	0.79
train/box_loss	0.015	0.016	0.015	0.015	0.015	0.014
train/cls_loss	0.008	0.008	0.007	0.007	0.008	0.007
train/obj_loss	0.011	0.011	0.010	0.010	0.010	0.0098
Ποσοστό σιγουριάς: conf	Αποτελέσματα ανίχνευσης:					
0.2	15/19	16/19	17/19	14/19	15/19	15/19
0.5	13/19	14/19	15/19	14/19	15/19	14/19

Πίνακας 4: Μοντέλο με 200 εικόνες ανά κατηγορία

Εικόνες ανά κατηγορία	500	500	500	500
Μέγεθος batch	16	16	16	20
Αριθμός epochs	150	180	200	180
metrics/mAP_0.5	0.77	0.77	0.78	0.78
metrics/mAP_0.5:0.95	0.48	0.48	0.48	0.47
metrics/precision	0.8	0.8	0.78	0.8
metrics/recall	0.72	0.72	0.74	0.72
train/box_loss	0.017	0.017	0.016	0.016
train/cls_loss	0.005	0.005	0.005	0.005
train/obj_loss	0.013	0.012	0.012	0.012
Ποσοστό σιγουριάς: conf	Αποτελέσματα ανίχνευσης:			
0.2	20/20	19/20	20/20	20/20
0.5	19/20	18/20	17/20	18/20

Πίνακας 5: Μοντέλο με 500 εικόνες ανά κατηγορία

Στο πέμπτο μοντέλο, το σύνολο δεδομένων στο οποίο εκπαιδεύτηκε αποτελείται από πεντακόσιες εικόνες ανά κατηγορία. Στον πίνακα 5 βλέπουμε τις αποδόσεις των τεσσάρων εκδοχών αυτού του μοντέλου. Το πιο υψηλό mAP που έχουμε σε αυτό το μοντέλο είναι 0.78 και εμφανίζεται σε δύο εκδοχές η μία είναι αυτή με τα 20 batch και 180 epoch και η άλλη είναι αυτή με τα 16 batch και 200 epoch. Αν και η εκδοχή με τα 20 batch και 180 epoch έχει το μεγαλύτερο αριθμό θετικών αποτελεσμάτων ανίχνευσης ανάμεσα σε αυτές τις δύο, δεν είναι αυτή με τους πιο υψηλούς αριθμούς. Η εκδοχή με τα 16 batch και τα 150 epoch είναι αυτή που παρουσιάζει τους πιο μεγάλους αριθμούς θετικών αποτελεσμάτων και στις δύο παραμέτρους ποσοστού σιγουριάς. Επίσης πρέπει να σημειωθεί ότι αν και σε σχέση με τα υπόλοιπα μοντέλα μας, αυτό το μοντέλο δεν έχει πολύ υψηλό αποτέλεσμα στο mAP, έχει όμως από τα καλύτερα αποτελέσματα ανίχνευσης και στα δύο ποσοστά σιγουριάς που έχουμε θέσει.

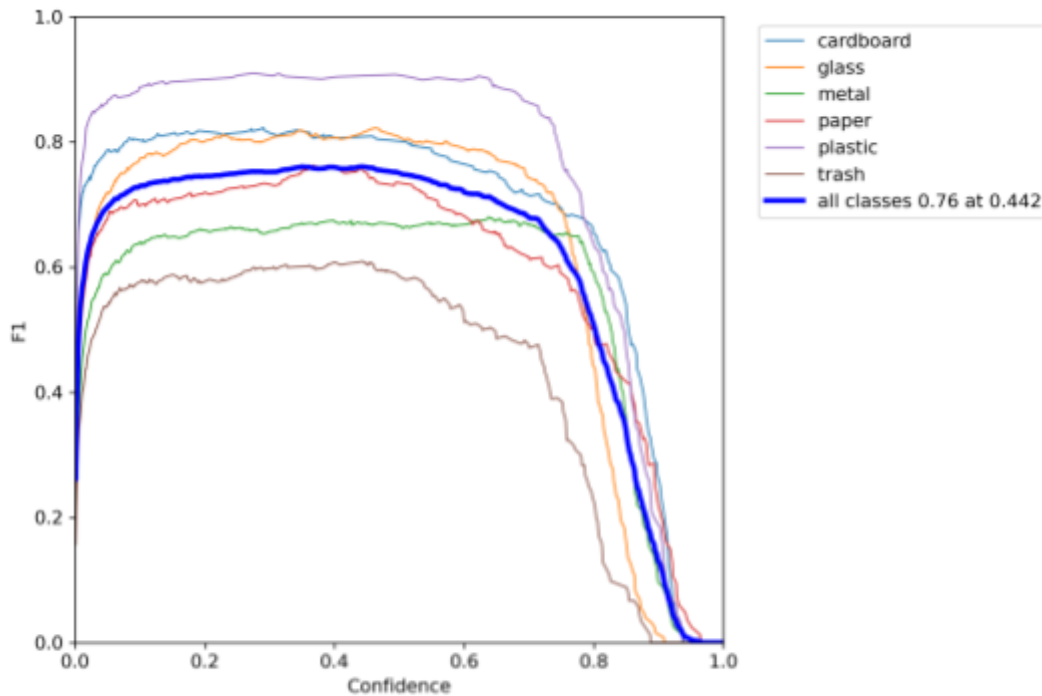
Λόγω των παραπάνω αποτελεσμάτων και λαμβάνοντας υπόψη το μέγεθος του σύνολο δεδομένων που χρησιμοποιήθηκε σε αυτό το μοντέλο, καταλήξαμε στην επιλογή επιλογή του πέμπτου μοντέλου ως το πιο κατάλληλο για την εργασία μας και πιο συγκεκριμένα στην έκδοση με τα 16 batch και τα epoch.

5.2 Αποτελέσματα από YOLOv5

Αρχικά θα ήθελα να αναφέρω ότι η εκπαίδευση των παραπάνω μοντέλων αλλά και η εκπαίδευση του τελικού μοντέλου που επιλέξαμε ως πιο κατάλληλο, έγιναν στην επαγγελματική έκδοση (pro version) του Google Colab. Στην δωρεάν έκδοση του Google Colab αντιμετωπίσα αρκετά προβλήματα, όπως ότι κάποιες φορές κατά την εκπαίδευση του μοντέλου η GPU αποσυνδέοταν λόγω του φόρτου εργασίας της πλατφόρμας, η πρόοδος χάνονταν και έπρεπε να ξεκινήσω πάλι από το μηδέν. Άλλο ένα βασικό πρόβλημα ήταν το μεγάλο χρονικό διάστημα που χρειάζεται για να ολοκληρωθεί η εκπαίδευση ακόμα και του πιο “μικρού” μοντέλου. Επίσης λόγω του φόρτου εργασίας κάποιες μέρες δεν μπορούσα να συνδεθώ με την GPU, αυτό γίνεται γιατί κάθε χρήστης της δωρεάν έκδοσης παίρνει συγκεκριμένο χώρο στο δίσκο και στην RAM (Random Access Memory). Με το Google Colab Pro παρέχεται μεγαλύτερος χώρος στο δίσκο και στην RAM και γι αυτό το λόγο η εκπαίδευση ολοκληρωνόταν πολύ πιο γρήγορα. Η εκπαίδευση του τελικού μοντέλου μας χρειάστηκε πέντε ώρες για να ολοκληρωθεί και με την ολοκλήρωση της λάβαμε τις αποδόσεις του μοντέλου τις οποίες και θα αναλύσουμε παρακάτω.

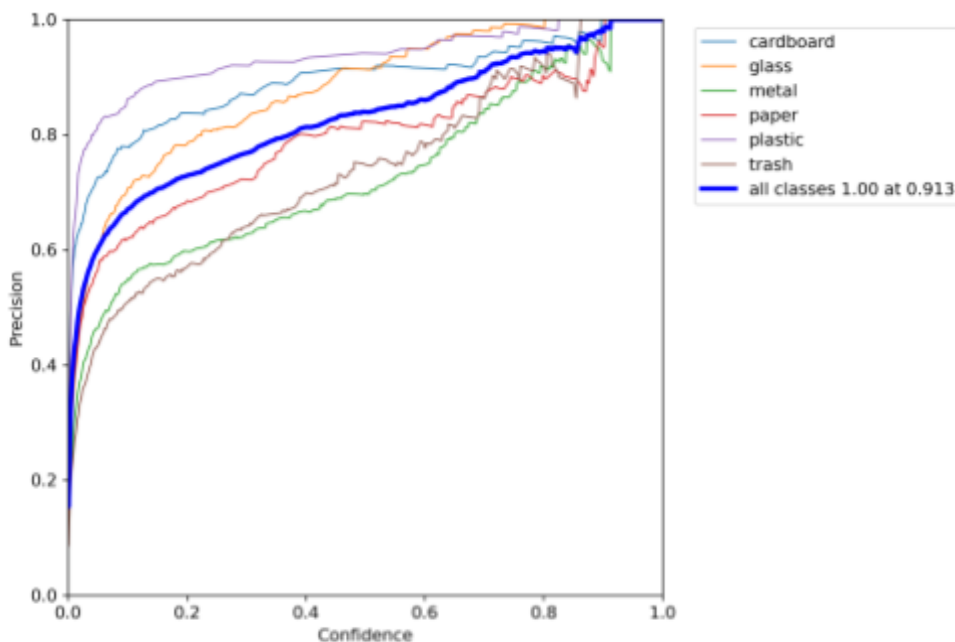
Πριν ξεκινήσουμε την ανάλυση των αποτελεσμάτων πρέπει να αναλύσουμε ορισμένες ορολογίες που χρησιμοποιούνται. Το Recall μας δείχνει την ικανότητα του μοντέλου να βρίσκει αντικείμενα που μοιάζουν στα αντικείμενα της εκπαίδευσης και ιδανικά η τιμή του για ένα τέλειο μοντέλο είναι ίση με 1. Με άλλα λόγια είναι ο αριθμός των σωστών προβλέψεων που έγιναν. Το Precision είναι ο αριθμός που αντιπροσωπεύει πόσες από τις συνολικές προβλέψεις ήταν τελικά σωστές. Τέλος η F1_score είναι μια μέτρηση που αφορά την ακρίβεια του μοντέλου και υπολογίζεται από τον τύπο $2*((precision*recall)/(precision+recall))$.

Στην παρακάτω εικόνα βλέπουμε την γραφική αναπαράσταση της εξέλιξης της F1_score σε σχέση με το Confidence, δηλαδή την σιγουριά του μοντέλου για την κάθε κατηγορία. Όπως φαίνεται στο γράφημα στις πιο πολλές κλάσεις μας όσο το confidence αυξάνεται το F1_score μειώνεται.



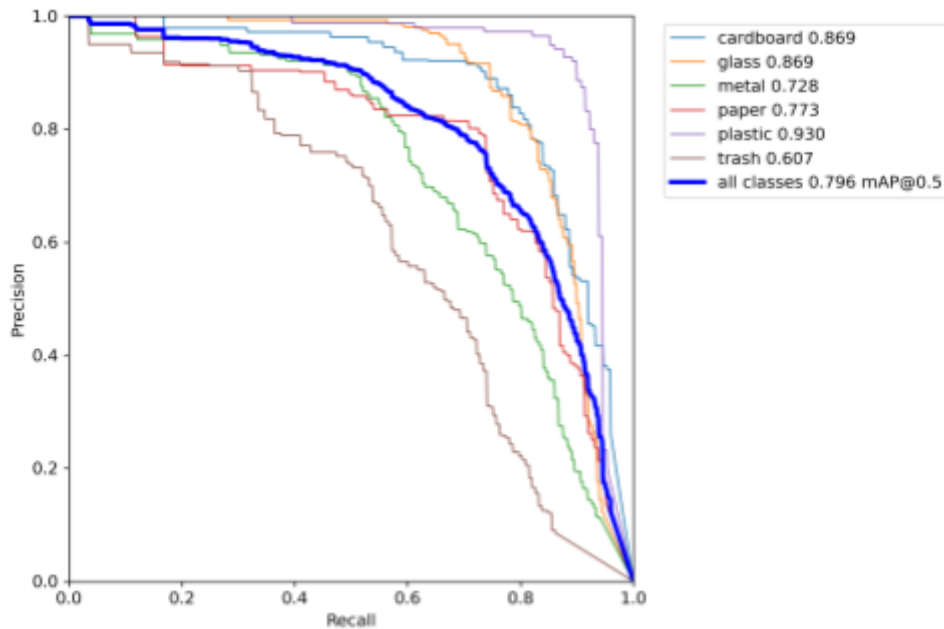
Εικόνα 22: Γραφική αναπαράσταση F1 με Confidence.

Στην εικόνα 23 βλέπουμε την γραφική παράσταση της καμπύλης Precision και παρατηρούμε ότι όσο αυξάνεται το confidence του μοντέλου τόσο ανεβαίνει και το precision, το οποίο σημαίνει ότι το σύνολο των σωστών ανιχνεύσεων αυξάνεται.

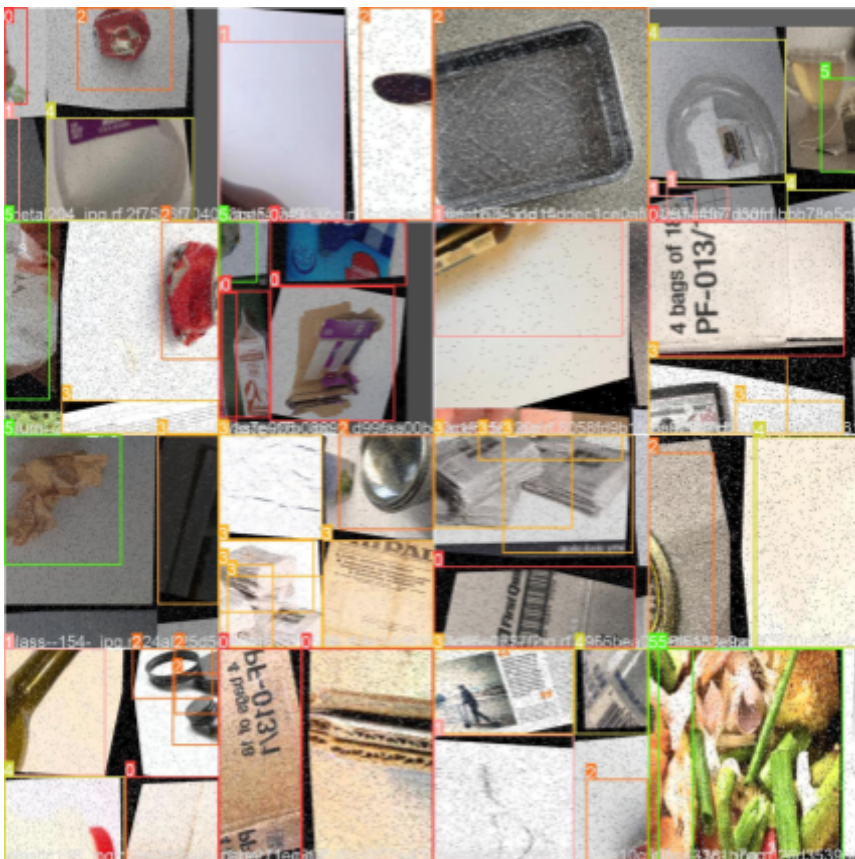


Εικόνα 23: Γραφική αναπαράσταση Precision με Confidence.

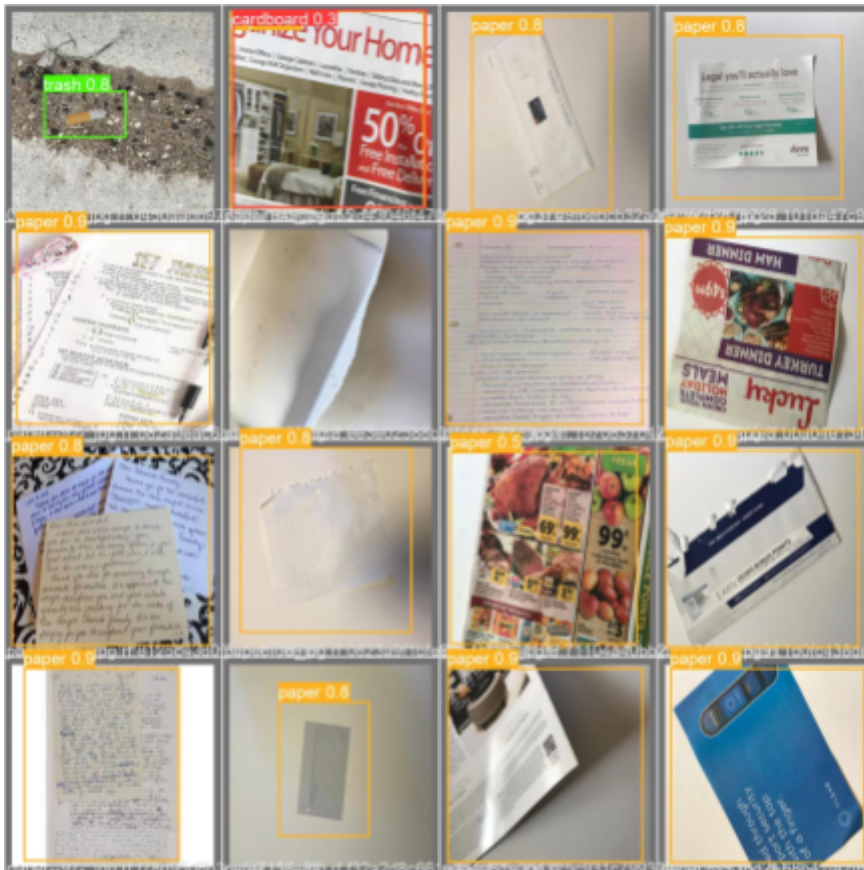
Στην παρακάτω γραφική παράσταση αναπαριστάται η σχέση μεταξύ Recall και Precision μετά την εκπαίδευση του μοντέλου. Σύμφωνα με το γράφημα, με την πάροδο της εκπαίδευσης το Recall αυξάνεται με σταθερούς ρυθμούς το οποίο είναι πολύ θετικό, γιατί όσο μεγαλύτερο είναι το Recall μας τόσο μικρότερη η πιθανότητα το μοντέλο να ανιχνεύσει λανθασμένα κάποιο αντικείμενο.



Εικόνα 24: Γραφική αναπαράσταση Precision με Recall.



Εικόνα 25: Προβλέψεις του YOLO.



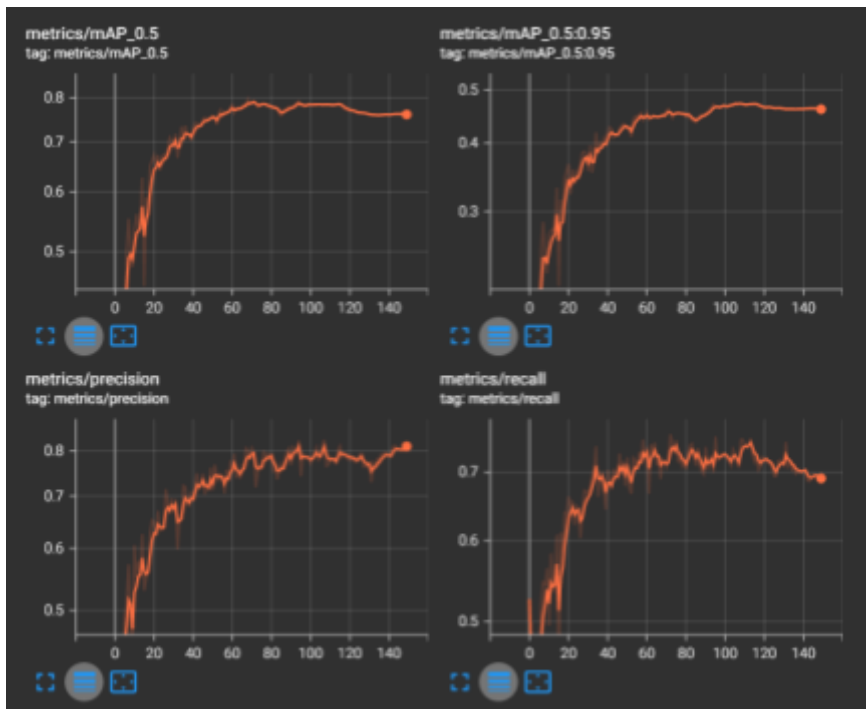
Εικόνα 26: Προβλέψεις YOLO.

Τέλος, στις παραπάνω εικόνες βλέπουμε τις προβλέψεις που έκανε το μοντέλο μετά την εκπαίδευση. Στην εικόνα 25 τα νούμερα στις ετικέτες αντιπροσωπεύουν τις κατηγορίες μας με την σειρά που αυτές ορίστηκαν κατά την προετοιμασία του συνόλου δεδομένων. Οι κατηγορίες αντιπροσωπεύονται ως εξής: η κατηγορία 0 (μηδέν) είναι το χαρτόνι, 1 (ένα) είναι το γυαλί, 2 (δύο) είναι το μέταλλο, 3 (τρία) είναι το χαρτί, 4 (τέσσερα) είναι το πλαστικό και 5 (πέντε) είναι τα σκουπίδια. Στην εικόνα 26 οι ετικέτες εμφανίζονται με την κατηγορία που έχουν ανιχνεύσει ολογράφως και το ποσοστό σιγουριάς για την σωστή κατηγοριοποίηση αυτού του αντικειμένου. Επίσης, κάθε κατηγορία μας έχει το δικό της χρώμα πλαισίου οριοθέτησης (bounding box) και η ετικέτα της παίρνει το ίδιο χρώμα.

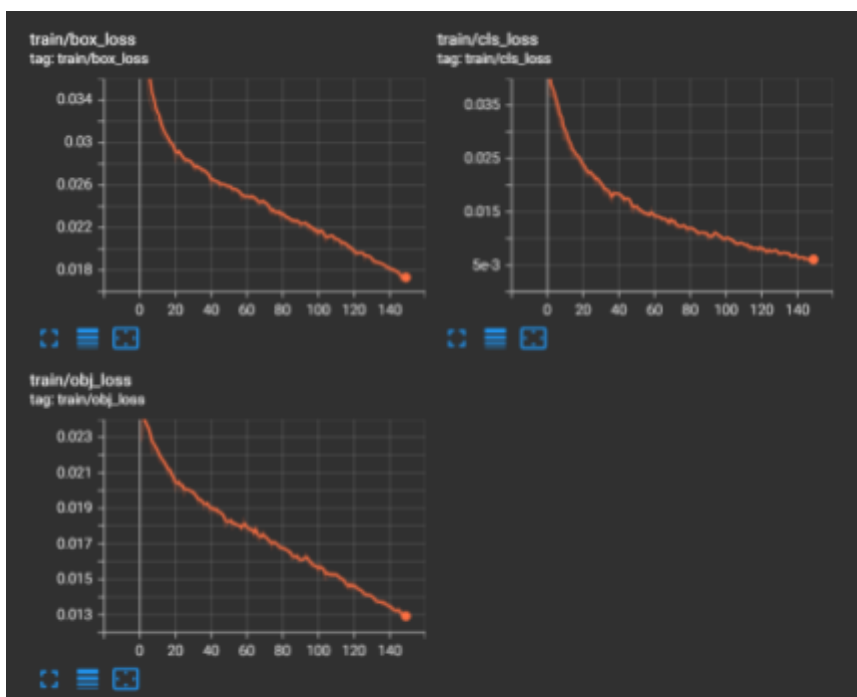
5.3 Αποτελέσματα από TensorBoard

Στο περιβάλλον του Google Colab όπως έχει προαναφερθεί, μετά τον κώδικα για την εκπαίδευση του μοντέλου και αφού αυτή έχει ολοκληρωθεί, τρέχουμε την εντολή για λάβουμε τις αποδόσεις της εκπαίδευσης του μοντέλου σε γραφήματα μέσω του TensorBoard. Πριν την ανάλυση των γραφημάτων του TensorBoard, πρέπει να αναλύσουμε κάποιες επιπλέον ορολογίες που δεν έχουν περιγραφεί στα παραπάνω κεφάλαια. Το `val_loss` αντιπροσωπεύει την ακρίβεια του μοντέλου μας σε σχέση με το υποσύνολο επικύρωσης (validation) του συνόλου δεδομένων μας. Το `mAP` αναλύθηκε σε παραπάνω κεφάλαιο αλλά

δεν δικαιολογήθηκε η μορφή του ως $mAP_{0.5}$ και $mAP_{0.5:95}$. Το mAP υπολογίζεται με την βοήθεια του IoU (Intersection over union) το οποίο μετρά την επικάλυψη μεταξύ 2 ορίων των ετικετών, και τα 0.5 και 0.5:95 αντιπροσωπεύουν τα διαφορετικά κατώφλια του IoU.[32] Τα συγκεκριμένα γραφήματα παρουσιάζουν την πρόοδο της εκπαίδευσης καθώς οι “κύκλοι” εκπαίδευσης των epoch ολοκληρώνονται. Στην παρακάτω εικόνα βλέπουμε της μετρήσεις του mAP , του Precision και της Recall καθώς ο αλγόριθμος εκπαιδεύονταν. Παρατηρούμε ότι και στα τέσσερα γραφήματα υπάρχει μια γρήγορη άνοδος μέχρι τα 70 epoch και μετά η πρόοδος γίνεται πιο μικρή αλλά σταθερή.

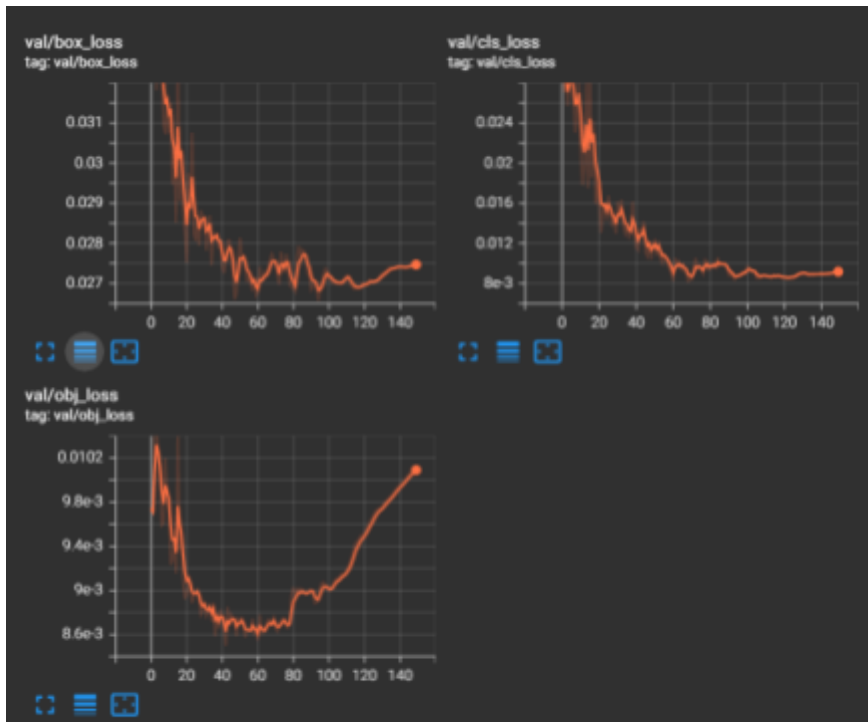


Εικόνα 27: Γραφήματα μετρήσεων TensorBoard.



Εικόνα 28: Γραφήματα εκπαίδευσης TensorBoard.

Στην εικόνα 28 παρατηρούμε ότι κατά την εκπαίδευση σημειώνεται γρήγορη πτώση του loss, η μεταβλητή αυτή αντιπροσωπεύει το πόσο καλές καλές είναι οι ανιχνεύσεις του μοντέλου κατά την διάρκεια της εκπαίδευσης όσο πιο κοντά στο μηδέν βρίσκεται το loss τόσο πιο ακριβές είναι το μοντέλο. Σε αυτά τα γραφήματα βλέπουμε ότι η τιμή του loss είναι από την αρχή της εκπαίδευσης μηδενική αλλά και κατά την διάρκεια της, συνεχίζει πτωτική πορεία.



Εικόνα 29: Γραφήματα επικύρωσης TensorBoard.

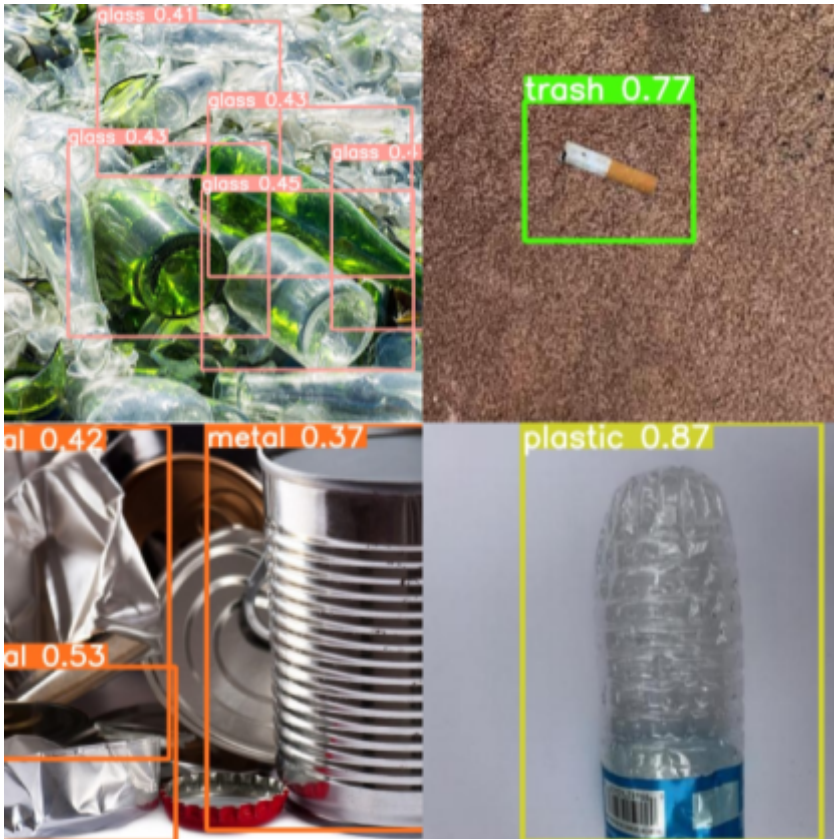
Τα γραφήματα της εικόνα 29 αναφέρονται στο loss του μοντέλου κατά την εκπαίδευση του υποσυνόλου της επικύρωσης του dataset μας. Σε αυτό παρατηρούμε ότι και εδώ το loss είναι μηδενικό από την αρχή της εκπαίδευσης. Αν και η πορεία τους δεν έχουν αρκετά σταθερή καθοδική πορεία, οι τιμές τους συνεχίζουν να είναι μικρότερες του μηδέν και κατά την ολοκλήρωση της εκπαίδευσης.

5.4 Ανίχνευση αντικειμένων

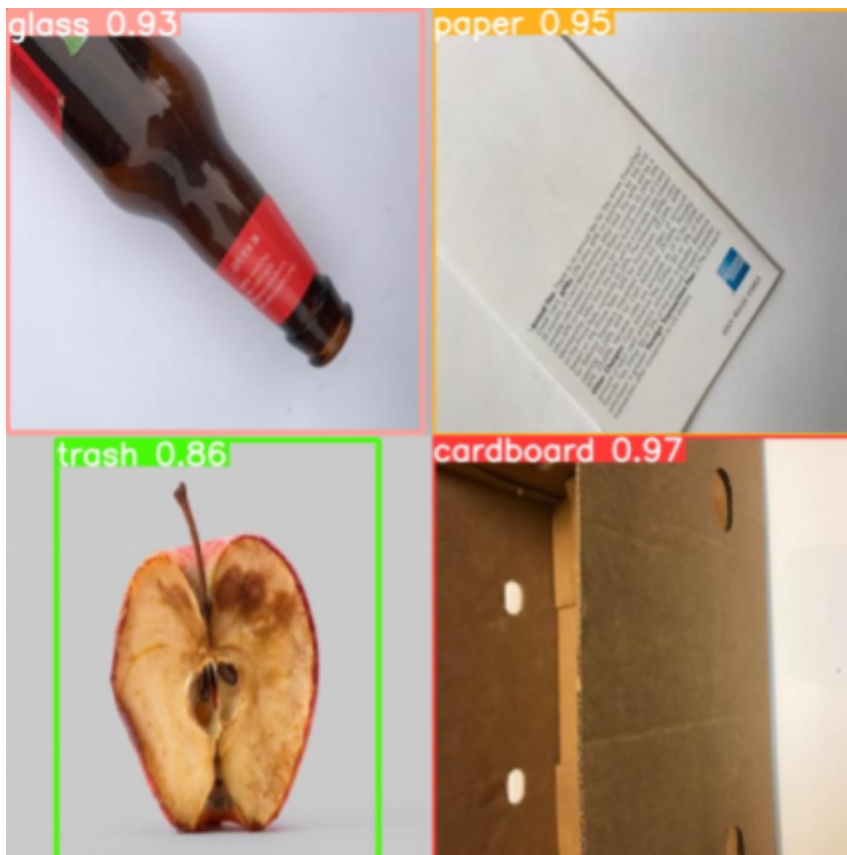
Η τεχνική της ανίχνευσης αντικειμένων (object detection) εμπεριέχει δύο επιμέρους τεχνικές, τον εντοπισμό της θέσης ενός αντικειμένου (object localization) και την κατηγοριοποίηση αντικειμένων (object classification). Ο εντοπισμός της θέσης των αντικειμένων είναι η τεχνική κατά την οποία εντοπίζονται όλα τα αντικείμενα στην εικόνα προς κατηγοριοποίηση. Υπάρχουν αρκετοί αλγόριθμοι για την ανίχνευση αντικειμένων και μπορούν να χωριστούν σε δύο κατηγορίες. Στη πρώτη κατηγορία είναι οι αλγόριθμοι που βασίζονται στην ταξινόμηση και εφαρμόζονται με δύο βήματα. Πρώτα επιλέγουν τις περιοχές ενδιαφέροντος για μια εικόνα και μετά ταξινομούν αυτές τις περιοχές

χρησιμοποιώντας συνελκτικά νευρωνικά δίκτυα. Αυτοί οι αλγόριθμοι είναι πιο αργοί λόγω του ότι πρέπει να εκτελέσουν προβλέψεις για κάθε επιλεγμένη περιοχή. Στη δεύτερη κατηγορία είναι οι αλγόριθμοι που βασίζονται στην παλινδρόμηση. Αυτοί οι αλγόριθμοι, προβλέπουν κλάσεις και πλαίσια οριοθέτησης (bounding boxes) για ολόκληρη την εικόνα με μια εκτέλεση του αλγορίθμου. Σε αυτήν την κατηγορία ανήκει ο αλγόριθμος YOLO. Επίσης οι αλγόριθμοι αυτής της κατηγορίας, δίνουν την δυνατότητα ανίχνευσης αντικειμένων σε πραγματικό χρόνο χωρίς να υπάρχουν σημαντικά μειονεκτήματα στην απόδοση του αλγορίθμου, καθώς θυσιάζονται μικρά ποσοστά ακρίβειας για μεγάλες βελτιώσεις στην ταχύτητα. [33]

Κατά την ανίχνευση αντικειμένων το μοντέλο μας εντοπίζει το αντικείμενο, εστιάζει σε αυτό μέσω του πλαισίου οριοθέτησης, υπολογίζει σε ποιά κλάση ανήκει το αντικείμενο και παρουσιάζει το ποσοστό σιγουριάς (confidence score). Στην εργασία μας πραγματοποιήσαμε ανίχνευση σε τρεις διαφορετικές καταστάσεις, ανίχνευση σε εικόνες, ανίχνευση σε βίντεο μη πραγματικού χρόνου σε μορφή mp4 και ανίχνευση βίντεο πραγματικού χρόνου (live) μέσω της κάμερας του υπολογιστή. Η ανίχνευση σε εικόνες εξηγήθηκε σε παραπάνω κεφάλαιο, η διαφορά της ανίχνευσης σε βίντεο η πραγματικού χρόνου από ότι σε εικόνες είναι απλά μια μικρή αλλαγή του κώδικα της εικόνας 20, στο τέλος της εντολής μπορούμε απλά να αλλάξουμε το μονοπάτι (path), από το φάκελο των εικόνων στον φάκελο των βίντεο και αν θέλουμε να κάνουμε ανίχνευση μόνο σε ένα βίντεο τότε προσθέσουμε το όνομα του αρχείου μαζί με το .mp4, στο path. Αυτές οι δύο εντολές για ανίχνευση είναι οι ίδιες στο Google Colab αλλά και όταν γίνεται ανίχνευση τοπικά στον υπολογιστή μέσω του Anaconda Navigator. Για ανίχνευση σε βίντεο πραγματικού χρόνου χρησιμοποιούμε το Anaconda Navigator. Αφού το ενεργοποιήσουμε ανοίγουμε το CMD.exe Prompt και η πρώτη εντολή που δίνουμε στο terminal είναι να ενεργοποιήσει το περιβάλλον της PyTorch με την εντολή `conda activate pytorch` μετά συμπληρώνουμε το path για τον φάκελο yolo που έχουμε αποθηκεύσει τα αρχεία του YOLO και της εκπαίδευσής μας. Σε αυτό το σημείο αν θέλουμε να κάνουμε ανίχνευση σε εικόνες ή βίντεο μη πραγματικού χρόνου, απλά πληκτρολογούμε την ανάλογη εντολή όπως στην εικόνα 20. Για ανίχνευση σε βίντεο πραγματικού χρόνου όμως η εντολή αλλάζει και γίνεται `python detect.py --weights runs/train/exp/weights/best.pt --img 416 --conf 0.5 --source 0` το source 0 αντιπροσωπεύει την κάμερα του υπολογιστή μας. Μολίς τρέξει η εντολή, η κάμερα μας ανοίγει και στο παράθυρο που ανοίγει για να βλέπουμε την εικόνα της κάμερας, βλέπουμε την ανίχνευση των αντικειμένων που κάνει το μοντέλο μας. Για να σταματήσουμε την ανίχνευση μέσω της webcam μας, πατάμε το Q στο πληκτρολόγιο μας, επίσης μετά το κλείσιμο της κάμερας ένα αρχείο .mp4 αποθηκεύεται στον φάκελο detect του φακέλου yolo. Στις παρακάτω εικόνες παρουσιάζονται παραδείγματα από τις ανιχνεύσεις του μοντέλου σε εικόνες και βίντεο. Στην εικόνα 32 αναπαρίστανται στιγμιότυπα από τα βίντεο και παρατηρούμε ότι το μοντέλο αναγνωρίζει το γυαλί με ποσοστό σιγουριάς 0.83, το μέταλλο με ποσοστό 0.80, το χαρτόνι με ποσοστό 0.64 και το χαρτί με ποσοστό 0.85.



Εικόνα 30: Αποτελέσματα YOLO από εικόνες. (1)



Εικόνα 31: Αποτελέσματα YOLO από εικόνες. (2)



Εικόνα 32: Αποτελέσματα YOLO από βίντεο.

5.5 Μελλοντικές κατευθύνσεις

Σε αυτήν την εργασία ασχοληθήκαμε με το πρόβλημα της οπτικής αναγνώρισης σκουπιδιών τα οποία βρίσκονται σε έναν εσωτερικό χώρο ενός σπιτιού. Για να το λύσουμε δημιουργήσαμε ένα μοντέλο επιτηρούμενης μάθησης. Τα αποτελέσματα αυτού του μοντέλου είναι αρκετά ενθαρρυντικά καθώς καταφέρνει να αναγνωρίζει έξι διαφορετικές κατηγορίες σκουπιδιών με το ποσοστό σιγουριάς πολλές φορές να φτάνει το 0.9 δηλαδή το 90%.

Ο τομέας της μηχανικής μάθησης διαρκώς αναπτύσσεται και έχει πολλές προοπτικές, γι' αυτό το λόγο είναι σίγουρο ότι μπορέσουν να γίνουν μελλοντικές βελτιώσεις και επεκτάσεις. Θα μπορούσαμε να το εκπαιδεύσουμε σε ένα μεγαλύτερο σύνολο δεδομένων και να αποκτήσει καλύτερη απόδοση μέσω της εκπαίδευσης του σε εικόνες μεγαλύτερης ανάλυσης. Βέβαια και οι δύο αυτές βελτιώσεις απαιτούν μεγαλύτερη υπολογιστική ισχύ. Θα μπορούσε να γίνει μεγαλύτερη μελέτη στον τομέα της προεργασίας του συνόλου δεδομένων για την επίτευξη μιας ακόμα πιο αποτελεσματικής εκπαίδευσης, η οποία θα οδηγήσει στην μεγαλύτερη ευστοχία του μοντέλου. Επιπλέον το μοντέλο θα μπορούσε να επεκτείνει την γνώση του και να εκπαιδευτεί ώστε να αναγνωρίζει και άλλες κατηγορίες ή το μοντέλο να εκπαιδευτεί από διαφορετικούς αλγορίθμους όπως ο αλγόριθμος SSD (Single Shot Multibox Detector) και ο RCNN (Region-based Convolutional Neural Network). Τέλος, θα μπορούσε να αναπτυχθεί μια ολοκληρωμένη εφαρμογή για κινητά τηλέφωνα, με την οποία ο/η χρήστης της να μπορεί να χρησιμοποιεί την κάμερα του κινητού του/της για να αναγνωρίσει τι είδους σκουπίδια έχει στο σπίτι του/της και η εφαρμογή να τον/την βοηθάει τα κατηγοριοποιήσει αυτά τα σκουπίδια σε απόβλητα ή ανακυκλώσιμα και με αυτόν τον τρόπο να γίνεται πιο ακριβής και αξιόπιστη η ανακύκλωση στα σπίτια.

Κεφάλαιο 6:

Βιβλιογραφία

- [1] Shengzhao Long, Balbir S. Dhillon - Man-Machine-Environment System Engineering (2020)
- [2] K. Simonyan, A. Zisserman - Very Deep Convolutional Networks for Large-Scale Image Recognition (2014)
- [3] Stuart Russell, Peter Norvig - Artificial Intelligence a modern approach (Third Edition) (2010)
- [4] E. R. Davies: Machine Vision - Theory, Algorithms, Practicalities (2005)
- [5] Rostyslav Demush - A Brief History of Computer Vision and Convolutional Neural Networks (2019) Ανάκτηση από: <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-network-s-8fe8aacc79f3>
- [6] C. Wilson, T. Hargreaves, R. Hauxwell-Baldwin - Smart homes and their users: a systematic analysis and key challenges. (2015)
- [7] Y. Heetae, L. Hwansoo, Z. Hangjung - User acceptance of smart home services: an extension of the theory of planned behavior. (2017)
- [8] Κ. Διαμαντάρας, Δ. Μπότσης - Μηχανική Μάθηση (2019)
- [9] Royal Society - Machine learning: The power and promise of computers that learn by example (2017) Ανάκτηση από: <https://royalsociety.org/~media/policy/projects/machine-learning/publications/machine-learning-report.pdf>
- [10] Αριστείδης Λύκας - Τεχνητά Νευρωνικά Δίκτυα-Εφαρμογές (2008)
- [11] J. Alzubi, A. Nayyar, A. Kumar - Machine Learning from Theory to Algorithms: An Overview (2018) Ανάκτηση από: <https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012>
- [12] J. Brownlee - Gaussian Processes for Classification with Python (2020) Ανάκτηση από: <https://machinelearningmastery.com/gaussian-processes-for-classification-with-python/>
- [13] J. Bhattacharyya - Step by step guide to object detection using Roboflow (2020) Ανάκτηση από: <https://analyticsindiamag.com/step-by-step-guide-to-object-detection-using-roboflow/>
- [14] S. Parveez, R. Iriondo - Google Colab 101 Tutorial with Python Tips, Tricks and FAQ (2020) Ανάκτηση από: <https://towardsai.net/p/programming/google-colab-101-tutorial-with-python-tips-tricks-and-faq-7689bd4d24b4>
- [15] G. Karimi - Introduction to YOLO Algorithm for Object Detection (2021) Ανάκτηση από: <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/#:~:text=YOLO%20is%20an%20abbreviation%20for,probabilities%20of%20the%20detected%20images.>

- [16] J.Nelson, J. Solawetz - YOLOv5 is Here: Stat-of-the-Art Object Detection at 140 FPS (2020) Ανάκτηση από: <https://blog.roboflow.com/yolov5-is-here/>
- [17] B. E. Nishimoto - Using TensorBoard (2018) Ανάκτηση από: <https://medium.com/neuronio/using-tensorboard-e3906a5798e6>
- [18] Oracle - What is PyTorch: A Complete Guide Ανάκτηση από: <https://developer.oracle.com/what-is-pytorch/>
- [19] Nvidia - PyTorch Ανάκτηση από: <https://www.nvidia.com/en-us/glossary/data-science/pytorch/>
- [20] P. S. Kalchuri - Anaconda Navigator | An Overview (2020) Ανάκτηση από: <https://medium.com/analytics-vidhya/anaconda-navigator-an-overview-4e5d27ca8047>
- [21] Educba - Anaconda Navigator Ανάκτηση από: <https://www.educba.com/anaconda-navigator/>
- [22] Caglar Uslu - What is Kaggle? (2022) Ανάκτηση από: <https://www.datacamp.com/blog/what-is-kaggle>
- [23] S. Raschka - Python Machine Learning (2015)
- [24] Altexsoft - Machine Learning Project Structure: Stages, Roles and Tools (2018) Ανάκτηση από: <https://www.altexsoft.com/blog/datascience/machine-learning-project-structure-stages-roles-and-tools/>
- [25] Matplotlib - Matplotlib: Visualization with Python Ανάκτηση από: <https://matplotlib.org/>
- [26] Numpy - Numpy documentation Ανάκτηση από: <https://numpy.org/doc/stable/>
- [27] OpenCV - Introduction to OpenCV-Python Tutorials Ανάκτηση από: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [28] SciPy - SciPy documentation Ανάκτηση από: <https://docs.scipy.org/doc/scipy/>
- [29] Torch - What is Torch? Ανάκτηση από: <http://torch.ch/>
- [30] Pandas - Pandas documentation Ανάκτηση από: <https://pandas.pydata.org/docs/>
- [31] Jason Brownlee - Difference Between a Batch and an Epoch in a Neural Network (2018) Ανάκτηση από: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>
- [32] Jonathan Hui - mAP(mean Average Precision) for Object Detection (2018) Ανάκτηση από: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
- [33] D. Scherer, A. C. Muller, S. Behnke - Evaluation of Pooling Operations in Convolutional Architectures for Object Detection (2010) Ανάκτηση από: https://www.researchgate.net/publication/221080312_Evaluation_of_Pooling_Operations_in_Convolutional_Architectures_for_Object_Recognition