Δ Ι Ε Θ Ν Ε Σ Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Τ Η Σ Ε Λ Λ Α Δ Ο Σ

Μ Ε Τ Α Π Τ Υ Χ Ι Α Κ Ο Σ Τ Η Ρ Ο Μ Π Ο Τ Ι Κ Η

ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΠΟΛΛΑΠΛΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

Διπλωματική Εργασία του

ΜΥΡΩΝΙΔΗ ΑΘΑΝΑΣΙΟΥ

Επιβλέπων:ΝΙΚΟΛΑΪΔΗΣ ΑΘΑΝΑΣΙΟΣ

ΣΕΡΡΕΣ, ΦΕΒΡΟΥΑΡΙΟΣ 2023

Υπεύθυνη Δήλωση *: Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην διπλωματική 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 διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του μεταπτυχιακού ρομποτικής του Διεθνούς Πανεπιστημίου της Ελλάδας.*

Περίληψη

Αποτελεί μια πρόκληση στον οπτικό υπολογιστή η παρακολούθηση των τυχαίων αντικειμένων σε φυσικά περιβάλλοντα. Η ανάγκη να προσαρμοστεί σε μεταβαλλόμενες εμφανίσεις κάτω από έντονη απόφραξη και μεταμόρφωση, αποτελεί ένα κεντρικό πρόβλημα. Ταυτόχρονα με τις τυχόν βελτιστοποιήσεις, για να αναπτυχθεί ένα πλαίσιο σε κατανεμημένες συσκευές και ανθρωποειδή ρομπότ, κάτι τέτοιο μπορεί να δείξει πόσο βιώσιμη μπορεί να είναι η έρευνα στην αναπτυξιακή ρομποτική. Ο διαχωρισμός των εργασιών της συνεργατικής χειραγώγησης μπορεί να χωριστεί στα επιμέρους καθήκοντα της διατήρησης της αντίληψης και της παρακολούθησης της τροχιάς αντικειμένου. Οι δύο αυτές εργασίες συνήθως θέτουν μεμονωμένες απαιτήσεις σχετικές με τον έλεγχο του υποκείμενου στόχου, σε σχέση με την ακρίβεια και την ανθεκτικότητα στις παρεμβολές. Στις περιοχές οι οποίες απαιτούν να εκτελεστεί ένα ευρύ φάσμα διαφορετικών εργασιών, η χρήση αυτόματων ρομπότ, επί του παρόντος, παρεμποδίζεται από την υψηλή πολυπλοκότητα του λογισμικού, το οποίο προσαρμόζει τον ελεγκτή σε διαφορετικές καταστάσεις που θα αντιμετωπίσει το ρομπότ . Δε μπορεί να γίνει διευκόλυνση της υλοποίησης των ελεγκτών με τα τρέχοντα πλαίσια λογισμικού ρομποτικής για μεμονωμένες εργασίες, οι οποίες παρουσιάζουν κάποια μεταβλητότητα. Οι δυνατότητες των ελεγκτών τους, ωστόσο, στο να προσαρμοστούν στις απαιτήσεις κατά τη διάρκεια του χρόνου εκτέλεσης θεωρούνται περιορισμένες, ενώ δεν μπορούν να κλιμακωθούν με τις απαιτήσεις ενός ευέλικτου αυτόνομο ρομπότ. Προτείνεται λοιπόν ένα πλαίσιο παρακολούθησης το οποίο χρησιμοποιεί τη δύναμη των Συνελικτικών Νευρωνικών Δικτύων για τη δημιουργία ενός προσαρμοστικού και ισχυρού μοντέλου του αντικειμένου από δεδομένα εκπαίδευσης, τα οποία παράγονται κατά την παρακολούθηση. Μειώνεται το υπολογιστικό κόστος και παρέχεται αυξημένη απόδοση από έναν μηχανισμό σταδιακής ενημέρωσης, επιτρέποντας με αυτό τον τρόπο την παρακολούθηση σε πραγματικό χρόνο με κορυφαίες επιδόσεις.

Περιεχόμενα

Πίνακας εικόνων

Εισαγωγή

Στη σημερινή εποχή, το πρόβλημα του εντοπισμού αντικειμένων και της αναγνώρισής τους μπορεί να επιλυθεί με τη χρήση Νευρωνικά Δίκτυα Συνέλιξης. Τα δίκτυα αυτά συνδυαστικά με τις κάρτες γραφικών, μπορούν να δώσουν την δυνατότητα να επιλυθούν προβλήματα εντοπισμού και αναγνώρισης αντικειμένων σε σχετικά μικρό χρόνο, καθιστώντας ταυτόχρονα ικανή τη χρήση τους σε εφαρμογές πραγματικού χρόνου.

Περιγραφή του προβλήματος

Όταν ένας άνθρωπος κοιτάξει μια εικόνα, γνωρίζει αμέσως τι αντικείμενα υπάρχουν μέσα, πως μπορούν να αλληλοεπιδράσουν μεταξύ τους, αλλά και ποια είναι η θέση τους. Το οπτικό ανθρώπινο σύστημα είναι ακριβές και γρήγορο επιτρέποντας την εκτέλεση πολύπλοκων εργασιών. Σε αντίθεση, το πρόβλημα του να εντοπιστούν αντικείμενα από έναν υπολογιστή δεν θεωρείται σχετικά απλό. Κάποιοι υπολογιστές, οι οποίοι θα διέθεταν αλγορίθμους οι οποίοι θα ήταν γρήγοροι και ακριβείς, θα μπορούσαν να επιτρέψουν σε αυτούς διάφορες λειτουργίες όπως είναι το ξεκλείδωμα των δυνατοτήτων γενικής χρήσης για ανταποκρινόμενα ρομποτικά συστήματα, να επιτρέπουν σε διάφορες βοηθητικές συσκευές τη μεταφορά πληροφοριών πραγματικού χρόνου σε ανθρώπινους χρήστες, την οδήγηση αυτοκινήτων χωρίς ειδικούς αισθητήρες και πολλά ακόμη.

Συνεισφορά στόχος της παρούσας μεταπτυχιακής διατριβής

Ο στόχος της παρούσας μεταπτυχιακής εργασίας είναι να γίνει μια γενικευμένη ανάλυση για τους επικρατέστερους αλγορίθμους που υπάρχουν στη μηχανική μάθηση για την ανίχνευση αντικειμένων, η εις βάθος μελέτη της χρήσης νευρωνικών δικτύων συνέλιξης μέσα από αλγόριθμους μηχανικής μάθησης σε εφαρμογές εντοπισμού και ανίχνευσης αντικειμένων σε βίντεο, εικόνες και εφαρμογές, οι οποίες μπορεί να απαιτήσουν ανίχνευση αντικειμένων σε πραγματικό χρόνο.

Διάρθρωση της διατριβής

Η διάρθρωση της παρούσας μεταπτυχιακής διατριβής είναι η εξής: Στο κεφάλαιο 1 αναπτύσσονται οι έννοιες της τεχνητής νοημοσύνης, της μηχανικής και βαθιάς μάθησης, της υπολογιστικής όρασης, ενώ γίνεται και μια εισαγωγή στα νευρωνικά δίκτυα, τα τεχνητά νευρωνικά δίκτυα και τα συνελικτικά νευρωνικά δίκτυα. Στο κεφάλαιο 2 αναπτύσσεται η συνεργατική ρομποτική σε επιτήρηση από πάνω προβολή, όπου αφορά ένα πλαίσιο για να παρακολουθηθούν πολλαπλά αντικείμενα

με ανίχνευση χρησιμοποιώντας βαθιά μάθηση. Γίνονται αναφορές στη συνεργατική ρομποτική, τη βαθιά μάθηση και τους αλγόριθμους που χρησιμοποιούνται. Στο κεφάλαιο 3 παρουσιάζονται τα MOT, δηλαδή η τρισδιάστατη παρακολούθηση πολλαπλών αντικειμένων με τη χρήση νευρωνικών δικτύων. Στο κεφάλαιο 4 παρουσιάζονται τα DOT, δηλαδή η δυναμική παρακολούθηση αντικειμένων για Visual SLAM, όπου αναλύονται βασικά στοιχεία όπως η τμηματοποίηση στιγμιότυπου, αν το αντικείμενο βρίσκεται σε κίνηση, propagation mask, η παρακολούθηση αντικειμένων και κάμερας, καθώς και η ποιότητα παρακολούθησης και τα ακραία σημεία και οι αποφράξεις. Στο κεφάλαιο 5 παρουσιάζονται οι πιο σύγχρονοι αλγόριθμοι εντοπισμού και αναγνώρισης αντικειμένων Faster R-CNN, SSD, EfficentDet, YOLOv3 και YOLOv4. Στο κεφάλαιο 6 παρουσιάζεται μια πειραματική μελέτη και τα συμπεράσματα της παρούσας μεταπτυχιακής διατριβής μετά και το πέρας της πρακτικής εφαρμογής, καθώς και οι δυνατότητες που μπορεί η συγκεκριμένη διατριβή να προσφέρει σε ένα ευρύ φάσμα επιστημών και εφαρμογών.

1.Γενικές έννοιες

1.1 Τεχνητή νοημοσύνη

Από πολλές απόψεις, ο σημερινός κόσμος μοιάζει με μια χώρα των θαυμάτων, παρόμοια με αυτή που ο βρετανός μαθηματικός Charles Lutwidge Dodgson, περισσότερο γνωστός με το όνομα Lewis Carroll, περιγράφει στα διάσημα μυθιστορήματά του. Τα αυτοοδηγούμενα αυτοκίνητα, τα έξυπνα ηχεία και η αναγνώριση εικόνας, είναι δυνατά εξαιτίας της προόδου στην τεχνητή νοημοσύνη AI (Artiicial Intelligence), η οποία ορίζεται ως η ικανότητα που έχει ένα σύστημα να ερμηνεύει σωστά τα εξωτερικά δεδομένα, να μαθαίνει από αυτά και να χρησιμοποιεί τις εξαγόμενες γνώσεις για να επιτευχθούν συγκεκριμένοι στόχοι και καθήκοντα μέσα από ευέλικτη προσαρμογή [1]

Τη δεκαετία του 1950 έγινε και η καθιέρωσή της ως ακαδημαϊκή επιστήμη. Η τεχνητή νοημοσύνη έμεινε μια περιοχή περιορισμένου ενδιαφέροντος πρακτικής και σχετικής επιστημονικής αφάνειας για πάνω από μισό αιώνα. Στη σημερινή εποχή, εξαιτίας της ανόδου των Big Data και των βελτιώσεων στην υπολογιστική ισχύ, εισήλθε στην κοινή συνομιλία και στο επιχειρηματικό περιβάλλον. Μπορεί να ταξινομηθεί ως ανθρωποποιημένη τεχνητή νοημοσύνη, ανθρώπινης έμπνευσης και αναλυτική νοημοσύνη, ανάλογα με τους τύπους νοημοσύνης οι οποίοι εμφανίζονται δηλαδή είτε γνωστική, είτε κοινωνική, είτε συναισθηματική νοημοσύνη ή σε Artiicial Narrow, General, και Super Intelligence από το εξελικτικό της στάδιο. [2]

Ένα από τα κοινά που έχουν όλοι αυτοί οι τύποι είναι ότι όταν η τεχνητή νοημοσύνη φτάνει σε μια γενική χρήση και δεν χρησιμοποιείται πλέον ως τέτοια. Το φαινόμενο αυτό περιγράφεται ως το φαινόμενο AI, όπως συμβαίνει με την έκπτωση της συμπεριφοράς ενός προγράμματος AI από τους θεατές, υποστηρίζοντας ότι δεν είναι πραγματική νοημοσύνη. Από τη δεκαετία του 1950 και σε τακτά χρονικά διαστήματα, υπήρχε μια πρόβλεψη από τους ειδικούς ότι θα χρειαστούν πολύ λίγα χρόνια ώστε να φτάσουν μέχρι την τεχνητή γενική νοημοσύνη, θέματα τα οποία δείχνουν συμπεριφορά η οποία δεν διακρίνεται από τον άνθρωπο σε όλες τις πτυχές της ενώ έχει και κοινωνική, και συναισθηματική και γνωστική νοημοσύνη. Για να μπορέσει να γίνει όμως καλύτερα αντιληπτό το τι είναι εφικτό, η τεχνητή νοημοσύνη μπορεί να εξεταστεί από δύο πτυχές, από το δρόμο τον οποίο έχει ήδη διανύσει και αυτόν που βρίσκεται ακόμη μπροστά της.

1.2 Μηχανική μάθηση

Οι άνθρωποι από την εξέλιξή τους χρησιμοποιούν πολλούς τύπους εργαλείων για να εκτελέσουν διάφορες εργασίες με τον πιο απλό τρόπο. Η δημιουργικότητα του ανθρώπινου εγκεφάλου οδήγησε στο να εφευρεθούν διαφορετικές μηχανές. Τα μηχανήματα αυτά έκαναν πιο εύκολη την ανθρώπινη ζωή δίνοντας στους ανθρώπους τη δυνατότητα να ανταποκριθούν σε διάφορες ανάγκες ζωής, ανάμεσα στις οποίες είναι οι υπολογιστές, τα ταξίδια και οι βιομηχανίες. Η μηχανική μάθηση είναι και αυτή ανάμεσά τους. Ορίζεται από τον Arthur Samuel, ως το πεδίο σπουδών το οποίο μπορεί να δώσει στους υπολογιστές τη δυνατότητα μάθησης χωρίς να υπάρχει ρητός προγραμματισμός. Ο Arthur Samuel ήταν διάσημος για το πρόγραμμα το οποίο παίζει πούλια.

Μηχανική μάθηση

Η ML (Machine Learning) χρησιμοποιείται για να διδάξει στις μηχανές τον τρόπο χειρισμού των δεδομένων έτσι ώστε να είναι περισσότερο αποτελεσματικά. Κάποιες φορές αφού προβληθούν τα δεδομένα, δεν μπορούν να ερμηνευθούν οι πληροφορίες εξαγωγής από αυτά. Στην περίπτωση αυτή εφαρμόζεται η μηχανική μάθηση. Καθώς είναι διαθέσιμη μια αφθονία συνόλου δεδομένων, αυξάνεται και η ζήτηση για μηχανική μάθηση. Πολλές βιομηχανίες την εφαρμόζουν για να εξαχθούν τα σχετικά δεδομένα. Ο σκοπός της είναι να μπορέσει να μάθει κάποιος από τα δεδομένα. Έχουν γίνει πολλές μελέτες για τον τρόπο με τον οποίο κατασκευάζονται οι μηχανές και πώς μαθαίνουν μόνες τους χωρίς καν να είναι ρητά προγραμματισμένες. Πολλοί προγραμματιστές και μαθηματικοί εφάρμοσαν αρκετές προσεγγίσεις για να βρουν τη λύση του προβλήματος αυτού, που εμφανίζεται όταν υπάρχουν τεράστια σύνολα δεδομένων.

Για να επιλυθούν τα προβλήματα δεδομένων, η μηχανική μάθηση βασίζεται σε διαφορετικούς αλγόριθμους. Οι επιστήμονες των δεδομένων επιθυμούν να επισημάνουν ότι δεν υπάρχει ένας ενιαίος τύπος αλγόριθμου ο οποίος να ταιριάζει και να λύνει όλα τα προβλήματα. Το είδος του αλγορίθμου το οποίο θα χρησιμοποιηθεί εξαρτάται, ανάλογα με το είδος του προβλήματος προς λύση, το είδος του μοντέλου που θα ταίριαζε καλύτερα, τον αριθμό των μεταβλητών και ούτω καθεξής. [3]

Η μηχανική μάθηση χωρίζεται στις 3 εξής κατηγορίες :

Στην εποπτευόμενη μάθηση η οποία προσαρμόζει το σύστημα έτσι ώστε για κάποια συγκεκριμένα δεδομένα εισόδου να παραχθεί μια έξοδος στόχος. Τα μαθησιακά δεδομένα αποτελούνται από πλειάδες δηλαδή ετικέτα και χαρακτηριστικά, όπου η ετικέτα αντιπροσωπεύει την έξοδο στόχο, ενώ τα χαρακτηριστικά αντιπροσωπεύουν τα δεδομένα εισόδου. Σε αυτή την περίπτωση, ο στόχος είναι να προσαρμοστεί το σύστημα έτσι ώστε για κάποια νέα είσοδο να μπορεί το σύστημα να προβλέψει την έξοδο στόχο. Η εποπτευόμενη μάθηση έχει τη δυνατότητα να χρησιμοποιεί τόσο διακριτούς όσο και συνεχείς τύπους εισροών δεδομένων.

Στην μάθηση χωρίς επίβλεψη περιλαμβάνονται τα δεδομένα τα οποία αποτελούνται από διανύσματα εισόδου χωρίς να υπάρχει συγκεκριμένη έξοδος. Στην μάθηση χωρίς επίβλεψη υπάρχουν διαφορετικοί στόχοι όπως είναι η οπτικοποίηση, η εκτίμηση πυκνότητας και η ομαδοποίηση. Ο στόχος στην οπτικοποίηση είναι τα δεδομένα να προβάλλονται προς τα κάτω από το χώρο υψηλών διαστάσεων σε 2 ή 3 διαστάσεις, έτσι ώστε να προβάλλονται παρόμοια στοιχεία δεδομένων. Ο σκοπός της εκτίμησης πυκνότητας είναι ο προσδιορισμός της κατανομής των δεδομένων εντός του χώρου εισόδου, ενώ ο στόχος της ομαδοποίησης είναι να ανακαλυφθούν οι ομάδες παρόμοιων στοιχείων δεδομένων με βάση τις αντιληπτές ή μετρημένες ομοιότητες μεταξύ των στοιχείων δεδομένων.

Στην ημι-εποπτευόμενη μάθηση χρησιμοποιούνται πρώτα τα δεδομένα χωρίς ετικέτα έτσι ώστε να μαθευτεί μια αναπαράσταση χαρακτηριστικών των δεδομένων εισόδου, ενώ στη συνέχεια χρησιμοποιείται η αναπαράσταση της εκμάθησης χαρακτηριστικών για να επιλυθεί η εποπτευόμενη εργασία. Το σύνολο δεδομένων εκπαίδευσης μπορεί να χωριστεί σε δύο μέρη: τα δείγματα δεδομένων όπου οι ετικέτες δεν είναι γνωστές και τα δείγματα δεδομένων με αντίστοιχες ετικέτες. Η ημι-εποπτευόμενη μάθηση έχει τη δυνατότητα της μη παροχής μια ρητής μορφής σφάλματος σε κάθε χρόνο, αλλά λαμβάνεται μόνο μια γενικευμένη ενίσχυση η οποία μπορεί να δίνει ένδειξη για το πώς το σύστημα θα πρέπει να αλλάξει τη συμπεριφορά του, κάτι που μερικές φορές αναφέρεται ως ενισχυτική μάθηση. Η ενισχυτική μάθηση ήταν επιτυχής σε διαφορετικές εφαρμογές όπως είναι ο εργοστασιακός έλεγχος και η αποτελεσματική ευρετηρίαση ιστοσελίδων, η επιλογή στρατηγικής μάρκετινγκ, η δρομολόγηση δικτύου κινητής τηλεφωνίας, η κίνηση με πόδια ρομπότ και η αυτόνομη πτήση ελικοπτέρου [4]

1.3 Βαθιά μάθηση

Η βαθιά μάθηση αποτελεί ένα νέο τομέα της μηχανικής μάθησης ο οποίος έχει αποκτήσει δημοτικότητα στο πρόσφατο παρελθόν. Αναφέρεται στις αρχιτεκτονικές οι οποίες περιέχουν πολλά κρυφά επίπεδα, τα βαθιά δίκτυα, για να γίνουν γνωστά διαφορετικά χαρακτηριστικά με πολλαπλά επίπεδα αφαίρεσης. Οι αλγόριθμοι βαθιάς μάθησης έχουν την επιδίωξη να εκμεταλλευτούν την άγνωστη δομή στην κατανομή εισόδου προκειμένου να ανακαλυφθούν καλές αναπαραστάσεις συνήθως σε πολλαπλά επίπεδα, με τα γνωστά χαρακτηριστικά σε υψηλότερο επίπεδο τα οποία ορίζονται από την άποψη των χαρακτηριστικών του κατώτερου επιπέδου. Οι συμβατικές τεχνικές μηχανικής μάθησης περιορίζονται στον τρόπο επεξεργασίας των φυσικών δεδομένων στην ακατέργαστη μορφή τους.

Για πολλές δεκαετίες το να κατασκευαστεί μια αναγνώριση προτύπων με το σύστημα της μηχανικής εκμάθησης, απαιτούνταν σημαντική τεχνογνωσία στον τομέα και προσεκτικοί μηχανικοί ώστε να καταλήξουν σε έναν εξαγωγέα χαρακτηριστικού, ο οποίος μεταμόρφωνε τα ακατέργαστα δεδομένα, όπως είναι οι τιμές των εικονοστοιχείων μιας εικόνας, σε μια κατάλληλη εσωτερική αναπαράσταση, ή ένα διάνυσμα χαρακτηριστικών από πού το σύστημα εκμάθησης, όπως για παράδειγμα ένας ταξινομητής, θα μπορούσε να ταξινομήσει ή να ανιχνεύσει μοτίβα στην είσοδο. [5] Η βαθιά εκμάθηση επιτρέπει να εισαχθούν ακατέργαστα δεδομένα όπως είναι τα pixels στην περίπτωση των δεδομένων εικόνας στον αλγόριθμο εκμάθησης, χωρίς να γίνει πρώτα εξαγωγή των χαρακτηριστικών ή να οριστεί ένα χαρακτηριστικό διάνυσμα. Οι αλγόριθμοι της βαθιάς μάθησης έχουν τη δυνατότητα να μάθουν το σωστό σύνολο χαρακτηριστικών, κάνοντάς το με πολύ καλύτερο τρόπο από την εξαγωγή των χαρακτηριστικών αυτών, χρησιμοποιώντας χειροκίνητη κωδικοποίηση.

Αντί να δημιουργηθεί ένα σύνολο αλγορίθμων και κανόνων για να εξαχθούν τα χαρακτηριστικά από τα ακατέργαστα δεδομένα, η βαθιά μάθηση περιλαμβάνει την αυτόματη εκμάθηση των χαρακτηριστικών αυτών κατά τη διάρκεια της εκπαιδευτικής διαδικασίας. Στη βαθιά μάθηση, γίνεται συνειδητοποίηση ενός προβλήματος ως προς την ιεραρχία των εννοιών, με κάθε έννοια να χτίζεται πάνω από τις άλλες. Μια βασική αναπαράσταση του προβλήματος κωδικοποιείται από τα κατώτερα στρώματα του μοντέλου, ενώ τα υψηλότερα επίπεδα βασίζονται σε αυτά τα κατώτερα στρώματα, έτσι ώστε να σχηματίσουν πιο σύνθετες έννοιες. Οι τιμές της έντασης των εικονοστοιχείων με δεδομένη μια εικόνα, τροφοδοτούνται ως οι είσοδοι στο σύστημα της βαθιάς μάθησης [5]

Εν συνεχεία, εξάγονται τα χαρακτηριστικά από ένα αριθμό κρυφών επιπέδων, από την εικόνα εισόδου. Τα κρυφά αυτά στρώματα χτίζονται το ένα πάνω στο άλλο με ιεραρχικό τρόπο. Αρχικά, τα χαμηλότερα επίπεδα του δικτύου εντοπίζουν τις περιοχές οι οποίες μοιάζουν με άκρα. Στη συνέχεια χρησιμοποιούνται οι περιοχές αυτές για να οριστούν οι γωνίες όπου τέμνονται οι άκρες και τα περιγράμματα των αντικειμένων. Τα στρώματα του υψηλότερου επιπέδου συνδυάζουν περιγράμματα και γωνίες έτσι ώστε να οδηγήσουν σε πιο αφηρημένα τμήματα αντικειμένου στο επόμενο στρώμα.

Τέλος, το επίπεδο εξόδου αποκτά την ετικέτα κλάσης εξόδου και ταξινομεί την εικόνα. Η λαμβανόμενη έξοδος στο επίπεδο εξόδου επηρεάζεται άμεσα από κάθε άλλο διαθέσιμο κόμβο στο δίκτυο. Η διαδικασία αυτή μπορεί να θεωρηθεί ως ιεραρχική μάθηση, με κάθε επίπεδο στο δίκτυο να χρησιμοποιεί την έξοδο των προηγούμενων επιπέδων ως δομικό στοιχείο, για να κατασκευαστούν όλο και πιο περίπλοκες έννοιες στα υψηλότερα επίπεδα. Στην τρέχουσα βαθιά μάθηση συχνά περιλαμβάνονται η εκμάθηση εκατοντάδων ή δεκάδων διαδοχικών στρωμάτων αναπαράστασης από τα αυτόματα δεδομένα εκπαίδευσης. Στη μηχανική μάθηση οι συμβατικές προσεγγίσεις επικεντρώνονται συχνά στην εκμάθηση μόνο ενός ή 2 επιπέδων αναπαραστάσεων δεδομένων. Τέτοιες προσεγγίσεις κατηγοριοποιούνται συχνά ως ρηχή μάθηση. Η μηχανική μάθηση και η βαθιά μάθηση είναι υποτομείς της Τεχνητής Νοημοσύνης (AI).

Αν και η βαθιά μάθηση υπάρχει από τη δεκαετία του 1980, δεν ήταν δημοφιλής για πολλά χρόνια καθώς η υπολογιστική υποδομή, το λογισμικό και το υλικό δεν ήταν επαρκή και τα διαθέσιμα σύνολα δεδομένων ήταν πολύ μικρά. Όταν άρχισε η πτώση της δημοτικότητας των συμβατικών νευρωνικών δικτύων, τότε μόνο τα βαθιά δίκτυα έκαναν τη μεγάλη επανεμφάνιση πετυχαίνοντας θεαματικά αποτελέσματα σε εργασίες αναγνώρισης ομιλίας και ρομποτικής όρασης. [5]

1.4 Μηχανική όραση

Τα συστήματα μηχανικής όρασης (MVS) έχουν τη δυνατότητα προσομοίωσης των ανθρώπινων οπτικών λειτουργιών για την αναγνώριση και ανάλυση εικόνων και βίντεο. [6][7]. Η χρήση τους μπορεί να επεκταθεί στην αυτόνομη οδήγηση, την ασφάλεια, τη ρομποτική, την παραγωγή, την αναγνώριση εικόνας και χαρακτηριστικών, την επεξεργασία εικόνας την απόκτηση εικόνας και άλλους τομείς. Οι αυξανόμενες ποσότητες δεδομένων και τα νέα σενάρια εφαρμογών, δημιούργησαν μια ζήτηση για τα MVS με χαμηλότερη τιμή, μικρότερο όγκο, υψηλότερη ενεργειακή απόδοση και ταχύτερη παράλληλη επεξεργασία. [8]

1.5 Νευρωνικά δίκτυα

Το νευρωνικό δίκτυο (NN) [9] ορίζεται ως ένας μαζικά παράλληλα κατανεμημένος επεξεργαστής, ο οποίος αποτελείται από μια απλή μονάδα επεξεργασίας, οι οποία έχει μια φυσική τάση αποθήκευσης της βιωματικής γνώσης κατασκευάζοντάς την διαθέσιμη προς χρήση. Τα NN έχουν χρήση από πολλούς ερευνητές σε πολλές διαφορετικές εφαρμογές όπως είναι τα οικονομικά [10], η μεταποίηση [11], ιατρικές εφαρμογές [12], αναγνώριση ανθρωπίνου προσώπου [13], αναγνώριση ομιλίας [14], και ρομποτική [15][16]. Τα NN μπορούν να χρησιμοποιηθούν και σε διάφορους τομείς της κατασκευής. Οι Sukthomya and Tannock [11] πρότειναν μεθόδους για να εκπαιδευτούν τα νευρωνικά δίκτυα για τη μοντελοποίηση πολύπλοκων διαδικασιών παραγωγής. Τα νευρωνικά δίκτυα παρουσιάζουν μια καλή διακριτική ικανότητα έχοντας γενικά καλύτερα αποτελέσματα συγκριτικά με την παραδοσιακή διακριτή ανάλυση.

Τα μοντέλα νεύρων

Ο νευρώνας αποτελεί τη μονάδα επεξεργασίας πληροφοριών η οποία είναι θεμελιώδης για να λειτουργήσει το νευρωνικό δίκτυο [9]. Αποδείχθηκε ότι το νευρωνικό δίκτυο μπορεί κατά προσέγγιση να πλησιάσει οποιαδήποτε σύνθετη μεγάλης κλίμακας ή γραμμική συνάρτηση με την κατάλληλη εκπαίδευση δεδομένων [9][17]. Μπορεί επίσης να προσφέρει χρήσιμες δυνατότητες και ιδιότητες όπως είναι η γενίκευση, η προσέγγιση συναρτήσεων, η χαρτογράφηση εισόδου εξόδου ,η προσαρμοστικότητα και η μη γραμμικότητα [9].

Έχουν ζωτικό ρόλο στο να αναγνωριστούν τα δυναμικά συστήματα και να ανιχνευθούν σφάλματα αφού μπορούν να χρησιμοποιηθούν όχι μόνο για την ανίχνευση της εμφάνισης του σφάλματος, αλλά και για την παροχή ενός μεταγενέστερου μοντέλου του ρομποτικού χειριστή. Το μοντέλο αυτό μετά το σφάλμα μπορεί να χρησιμοποιηθεί αποτελεσματικά για να απομονώσει και να εντοπίσει το σφάλμα και αν είναι δυνατόν να διευθετήσει την αποτυχία [18].

1.6 Τεχνητά νευρωνικά δίκτυα

Τα τεχνητά νευρωνικά δίκτυα ή συνδετικά μοντέλα, παρέχουν ένα μέσο για να αντιμετωπιστούν σύνθετα προβλήματα ανάλυσης τάσεων και κατηγοριοποίησης. Τα νευρωνικά δίκτυα έχουν μια μη παραμετρική φύση η οποία επιτρέπει στα μοντέλα να αναπτυχθούν χωρίς να υπάρχει προηγούμενη γνώση της κατανομής του πληθυσμού

δεδομένων ή κάποια πιθανή αλληλεπίδραση ανάμεσα στις μεταβλητές όπως απαιτείται από τις πιο γνωστές χρησιμοποιούμενες παραμετρικές στατιστικές μεθόδους. Η στατιστική τεχνική όπου χρησιμοποιείται συχνά για να εκτελεστεί η κατηγοριοποίηση είναι η διακριτική ανάλυση.

Βιολογική Βάση Τεχνητών Νευρωνικών δικτύων

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μια τεχνολογία που είναι βασισμένη σε μελέτες του εγκεφάλου και του νευρικού συστήματος. Τα δίκτυα αυτά μιμούνται ένα βιολογικό νευρωνικό δίκτυο χρησιμοποιώντας ένα μειωμένο σύνολο εννοιών από βιολογικά νευρικά συστήματα. Συγκεκριμένα, τα μοντέλα ANN προσομοιώνουν την ηλεκτρική δραστηριότητα που υπάρχει στο νευρικό σύστημα και τον εγκέφαλο. Επεξεργάζονται στοιχεία το οποία είναι επίσης γνωστά ως perceptron, και συνδέονται με άλλα στοιχεία επεξεργασίας. Τυπικά οι νευρώνες έχουν μια διάταξη διανύσματος ή στρώματος, με την έξοδο ενός στρώματος να χρησιμεύει ως είσοδος στο επόμενο στρώμα και πιθανώς και σε άλλα στρώματα. Ένας νευρώνας μπορεί να συνδεθεί είτε σε όλα είτε σε ένα υποσύνολο των νευρώνων στο επόμενο στρώμα με τις συνδέσεις αυτές, να προσομοιώνουν τις συνδέσεις του εγκεφάλου.

Σε μια νευρώδη προσομοίωση της ηλεκτρικής διέγερσης ενός νευρικού κυττάρου, εισέρχονται σταθμισμένα σήματα δεδομένων, έχοντας ως συνέπεια τη μεταφορά πληροφοριών εντός του δικτύου ή του εγκεφάλου. Σε ένα στοιχείο επεξεργασίας οι τιμές εισόδου, in, πολλαπλασιάζονται με ένα βάρος σύνδεσης, wn,m, το οποίο προσομοιώνει την ενίσχυση των νευρικών οδών στον εγκέφαλο. Η μάθηση προσομοιώνεται στα ANN έχοντας περάσει από την προσαρμογή των βαρών σύνδεσης.

Από τη στιγμή που θα υπολογιστεί η τιμή της εισόδου, χρησιμοποιείται στη συνέχεια το στοιχείο επεξεργασίας σε μια συνάρτηση μεταφοράς ώστε να παραχθεί στην έξοδο της και συνεπώς στα σήματα εισόδου για το επόμενο στρώμα επεξεργασίας. Η τιμή εισόδου των νευρώνων μετασχηματίζεται από τη συνάρτηση μεταφοράς. Συνήθως ο μετασχηματισμός αυτός περιλάμβανε τη χρήση κάποιας γραμμικής συνάρτησης όπως υπερβολικής εφαπτομένης ή της χρήσης του σιγμοειδούς. Ανάμεσα στις στρώσεις των στοιχείων επεξεργασίας, επαναλαμβάνεται η διαδικασία μέχρι την τελική τιμή εξόδου, ώστε το διάνυσμα των τιμών να παραχθεί από το νευρωνικό δίκτυο. Θεωρητικά για να προσομοιωθεί η σύγχρονη δραστηριότητα του ανθρώπινου νευρικού συστήματος, τα στοιχεία επεξεργασίας του τεχνητού νευρωνικού δικτύου θα πρέπει να ενεργοποιηθούν με τη χρήση σταθμισμένους σήματος εισόδου με ασύγχρονο τρόπο [19].

1.7 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα (CNN ή ConvNets) αποτελούν έναν αλγόριθμο βαθιάς μηχανικής μάθησης ο οποίος λαμβάνει μια εικόνα εισόδου, εκχωρεί σημασία δηλαδή βάρη σε διάφορες πτυχές ή αντικείμενα της εικόνας και διαφοροποιεί το ένα από το άλλο. Η ονομασία τους προέρχεται από την ομώνυμη μαθηματική πράξη της συνέλιξης. Σε ένα CNN απαιτείται προεπεξεργασία η οποία είναι πολύ λιγότερη συγκριτικά με άλλους αλγορίθμους κατηγοριοποίησης. Η αρχιτεκτονική τους είναι ανάλογη με τη συνδεσιμότητα των νευρώνων στον ανθρώπινο εγκέφαλο, ενώ είναι εμπνευσμένη από την οργάνωση οπτικού φλοιού του εγκεφάλου.

Το Δεκτικό Επίπεδο (Receptive Field), αποτελεί μια περιορισμένη περιοχή του οπτικού πεδίου όπου γίνεται μια ανταπόκριση σε ερεθίσματα. Έγινε αλληλοεπικάλυψη μιας συλλογής τέτοιων πεδίων, ώστε να καλυφθεί ολόκληρη η οπτική περιοχή. Τα συνελικτικά νευρωνικά δίκτυα στα νευρωνικά δίκτυα, αποτελούν μία από τις 3 κύριες κατηγορίες για να αναγνωριστούν, κατηγοριοποιηθούν εικόνες και πρόσωπα, και φυσικά να ανιχνευτούν αντικείμενα. Όσον αφορά τον υπολογιστή, μια εικόνα είναι απλά μια σειρά τιμών, όπου συνήθως είναι ένας πίνακας τρισδιάστατων (RGB) τιμών pixel. Το CNN επεξεργάζεται τις εικόνες με τη χρήση πινάκων βαρών τα οποία ονομάζονται φίλτρα ή χαρακτηριστικά, τα οποία ανιχνεύουν συγκεκριμένα χαρακτηριστικά όπως είναι τα οριζόντια ή κάθετα άκρα κλπ. Ο απώτερος στόχος του CNN είναι ο εντοπισμός του τι συμβαίνει στη σκηνή. Διάφοροι τύποι επιπέδων αποτελούν ένα συνελικτικό νευρωνικό δίκτυο τα οποία περιγράφονται παρακάτω.

1.7.1 Συνελικτικό επίπεδο / Convolutional layer

Το συνελικτικό επίπεδο αποτελεί το βασικό δομικό στοιχείο ενός συνελικτικού νευρωνικού δικτύου. Σε ένα συνελικτικό επίπεδο κάθε νευρώνας θεωρείται υπεύθυνος για να ενεργοποιήσει ένα συγκεκριμένο χαρακτηριστικό όταν το βλέπει στην είσοδο του. Στη δισδιάστατη μεριά του βάθους, όλοι οι νευρώνες μοιράζονται τις ίδιες παραμέτρους δηλαδή πόλωσης και βάρη, κάτι που σημαίνει ότι αναζητούν το ίδιο χαρακτηριστικό σε διαφορετικές τοποθεσίες μιας εικόνας. Καθώς το ίδιο φίλτρο εφαρμόζεται σε όλες τις τοποθεσίες εικόνας για να δοθεί ένας χάρτης χαρακτηριστικών, κάτι τέτοιο μπορεί να θεωρηθεί και ως συνελικτική λειτουργία.

1.7.2 Συναρτήσεις ενεργοποίησης

Σε ένα νευρωνικό δίκτυο ένα συνελικτικό επίπεδο ακολουθείται από μια συνάρτηση ενεργοποίησης προσθέτοντας την απαιτούμενη μη γραμμικότητα στο δίκτυο. Αποτελούν πρακτικά έναν κόμβο ο οποίος τοποθετείται είτε ανάμεσα είτε στο τέλος

των νευρωνικών δικτύων, και βοηθά στο να ληφθεί απόφαση για το αν θα ενεργοποιηθεί όχι ένας νευρώνας. Μια λειτουργία ενεργοποίησης μπορεί να πάρει μια είσοδο εκτελώντας μια συγκεκριμένη μαθηματική λειτουργία σε αυτή. Κάποιες από αυτές τις λειτουργίες ενεργοποίησης είναι οι παρακάτω :

•Sigmoid : Μια σιγμοειδής συνάρτηση η οποία παίρνει μια είσοδο και τη συμπιέζει στην περιοχή 0 έως 1.

•Υπερβολική Εφαπτομένη : Αυτή η μη γραμμικότητα συμπιέζει μια τιμή εισόδου στην περιοχή -1 έως 1. Η έξοδος της tanh είναι μηδενική, ωστόσο θεωρείται κορεσμένη σε πολύ χαμηλές και πολύ υψηλές τιμές με την κλίση να είναι σχεδόν μηδενική.

•ReLU : Η Ανορθωμένη Γραμμική Συνάρτηση Ράμπας (Rectified linear unit) μπορεί να πάρει ως είσοδο επιστρέφοντας μηδέν στην περίπτωση που είναι αρνητική, αλλά για θετικές τιμές επιστρέφει την ίδια τιμή πίσω δίνοντας έτσι μια έξοδο η οποία κυμαίνεται από μηδέν έως άπειρο. Μπορεί να υπολογιστεί γρήγορα συγκριτικά με την σιγμοειδή συνάρτηση και την tanh οι οποίες περιλαμβάνουν εκθετικά. Επιπλέον, δεν έχει πρόβλημα διαβάθμισης κορεσμού, κάτι που την κάνει την πιο συχνά χρησιμοποιούμενη λειτουργία ενεργοποίησης.

•Συνάρτηση Softmax : Η συνάρτηση Softmax αποτελεί μια συνάρτηση ενεργοποίησης, η οποία μετατρέπει τους αριθμούς σε πιθανότητες με συνολικό άθροισμα 1, και εξάγει ένα διάνυσμα το οποίο αντιπροσωπεύει τις πιθανότητες από ένα σύνολο πιθανών αποτελεσμάτων.

•Συνάρτηση Mish : Η συνάρτηση ενεργοποίησης Mish είναι μια ομαλή, μη μονοτονική συνάρτηση ενεργοποίησης [20]. Οι λόγοι για τους οποίους χρησιμοποιείται είναι εξαιτίας του χαμηλού κόστους και των διαφόρων ιδιοτήτων της, όπως είναι η μονοτονική και ομαλή φύση της, δεν υπάρχει άνω όριο, ενώ οριοθετείται κάτω από την ιδιότητα βελτιώνοντας έτσι την απόδοσή της συγκριτικά με άλλες δημοφιλείς συναρτήσεις ενεργοποίησης όπως είναι η συνάρτηση ReLU, ενώ βοηθά και στο να επιτευχθούν ισχυρά αποτελέσματα κανονικοποίησης ταιριάζοντας σωστά το μοντέλο.

1.7.3 Επίπεδα χωρικής υποδειγματοληψίας / Pooling layers

Συνήθως μετά τη συνάρτηση και τη λειτουργία ενεργοποίησης σε ένα CNN εφαρμόζεται ένα στρώμα συγκέντρωσης. Το στρώμα αυτό μειώνει το μέγεθος του

όγκου εισόδου χωρικά, μειώνοντας έτσι συνεπώς και τον αριθμό των παραμέτρων οι οποίες απαιτούνται για να εκπαιδευτεί το CNN. Βοηθά επίσης στο να αποφευχθεί η υπερεκπαίδευση (overfitting), η περίπτωση δηλαδή που ένα μοντέλο μαθαίνει τον θόρυβο και τις λεπτομέρειες τα δεδομένα της εκπαίδευσης, κάτι που μπορεί να επηρεάσει αρνητικά την απόδοση του μοντέλου στα νέα δεδομένα [21]. Η μέγιστη συγκέντρωση είναι η πιο χρησιμοποιούμενη λειτουργία συγκέντρωσης όπου αντιστοιχείται και λαμβάνεται στην έξοδο το μέγιστο μιας περιοχής $n * n$, όπου n είναι το μέγεθος του φίλτρου συγκέντρωσης.

1.7.4 Πλήρως συνδεδεμένα επίπεδα

Η εφαρμογή ενός πλήρως συνδεδεμένου πεδίου γίνεται συνήθως μετά από μια σειρά επιπέδων χωρικής υποδειγματοληψίας και μιας σειράς συνελικτικών επιπέδων σε ένα CNN. Όταν το επίπεδο είναι πλήρως συνδεδεμένο, η σύνδεση του κάθε νευρώνα γίνεται με όλους τους νευρώνες στο προηγούμενο επίπεδο, κάνοντας έτσι το επίπεδο υπεύθυνο για να συσσωρευτούν όλες οι πληροφορίες από τα χαρακτηριστικά των χαμηλότερων επιπέδων. Μετά από ένα πλήρως συνδεδεμένο επίπεδο σε ένα CNN, δεν μπορεί να εφαρμοστεί κάποιο συνελικτικό επίπεδο.

1.7.5 Επίπεδο μαζικής κανονικοποίησης

Οι [22] εξήγαγαν το επίπεδο κανονικοποίησης για να αντιμετωπίσουν το πρόβλημα της εσωτερικής μεταβολής σε βαθιά νευρωνικά δίκτυα. Η εσωτερική μεταβλητή μεταβολή αποτελεί την κατανομή των αλλαγών της εισόδου σε εσωτερικά επίπεδα του δικτύου, έχοντας μεγαλύτερο αντίκτυπο σε βαθύτερα επίπεδα. Καθώς η είσοδος σε οποιαδήποτε επίπεδο μπορεί να επηρεαστεί από τις παραμέτρους όλων των προηγούμενων επιπέδων, στα αρχικά επίπεδα κάποιες μικρές αλλαγές σε βαθύτερα στρώματα, μπορούν να οδηγήσουν στην αλλαγή στην κατανομή. Το επίπεδο μαζικής κανονικοποίησης μπορεί να κανονικοποιήσει τις εισόδους μιας παρτίδας με εκ νέου κλιμάκωση και κεντράρισμα.

1.7.6 Επίπεδο κανονικοποίησης

Για να αποφευχθεί το πρόβλημα της υπερεκπαίδευσης στα νευρωνικά δίκτυα, γίνεται κανονικοποίηση. Καθώς υπάρχει ένας μεγάλος αριθμός παραμέτρων στα νευρωνικά δίκτυα, μπορεί να γίνει εύκολη εκπαίδευση ενός μεγάλου συνόλου δεδομένων χωρίς όμως να είναι σε θέση να γίνει καλή γενίκευση. Από τις πιο γνωστές μεθόδους

κανονικοποίησης αποτελούν η κανονικοποίηση L1, η κανονικοποίηση L2 και η απόσυρση (dropout).

Κανονικοποίηση L2 : Η κανονικοποίηση L2 προσθέτει στη συνάρτηση απόκλισης έναν όρο κανονικοποίησης, ώστε τα βάρη τα οποία τροποποιούνται κατά τη διάρκεια της εκπαίδευσης να παραμείνουν σε χαμηλές τιμές.

Κανονικοποίηση L1 : η μέθοδος αυτή είναι παρόμοια με την κανονικοποίηση L2, χωρίς όμως ο όρος κανονικοποίησης της να χρησιμοποιεί την ύψωση στο τετράγωνο, αλλά το άθροισμα απόλυτων βαρών.

1.7.7 Απόσυρση

Μια τεχνική νευρωνικών δικτύων βαθιάς μάθησης είναι η απόσυρση (dropout) η οποία έχει στόχο την αντιμετώπιση του προβλήματος της υπερβολικής εκπαίδευσης [23]. Η βασική η ιδέα της απόσυρσης είναι να γίνει τυχαία απόσυρση μερικών νευρώνων μαζί με τις συνδέσεις τους καθώς διαρκεί η εκπαίδευση. Αποσύρονται τυχαίοι νευρώνες σε κάθε επανάληψη της εκπαίδευσης, εκπαιδεύοντας έτσι σε ένα διαφορετικό νευρωνικό δίκτυο κάθε φορά ένα πιο γενικευμένο δίκτυο τη στιγμή της δοκιμής. Μπορεί ακόμη να θεωρηθεί και ως εκπαίδευση συνόλου όπου ο μεγαλύτερος αριθμός των κατηγοριοποιητών, εκπαιδεύονται ξεχωριστά μαθαίνοντας έτσι διαφορετικές πτυχές των δεδομένων. Τα αποτελέσματα των αδύναμων κατηγοριοποιητών, όσο διαρκεί η δοκιμή συνδυάζονται για να δώσουν ένα πιο γενικευμένο τελικό κατηγοριοποιητή.

1.7.8 Επίπεδα εξόδου

Η έξοδος από συγκεντρωτικά και συνελικτικά επίπεδα αντιπροσωπεύει τα χαρακτηριστικά υψηλού επιπέδου της εικόνας εισόδου. Αφού εξαχθούν τα χαρακτηριστικά, τα δεδομένα κατηγοριοποιούνται σε διάφορες κατηγορίες. Κάτι τέτοιο μπορεί να γίνει με τη χρήση ενός πλήρως συνδεδεμένου επιπέδου, το οποίο ενεργεί με τον ίδιο τρόπο όπως ένα κανονικό νευρωνικό δίκτυο, καθώς υπάρχει πλήρης σύνδεση με όλες τις ενεργοποιήσεις των προηγούμενων επιπέδων.

1.7.9 Συναρτήσεις απόκλισης

Η αναπροσαρμογή ενός νευρωνικού δικτύου μπορεί να γίνει μετά από μια συνάρτηση απόκλισης κατά τη στιγμή της εκπαίδευσης. Μια συνάρτηση απόκλισης ή αλλιώς συνάρτηση κόστους, μπορεί να μετρήσει πόσο καλά μαθαίνει το δίκτυο συγκρίνοντας την πραγματική με την αναμενόμενη έξοδο. Μπορεί να δώσει μια μεμονωμένη τιμή η οποία θα λέει πόσο καλό είναι το δίκτυο. Οι απαιτήσεις της είναι οι ακόλουθες :

Πρέπει να μπορεί να καταγράψει τη συνολική απόκλιση ως ένα μέσο όρο της απόκλισης του δείγματος. Η καταγραφή του πρέπει να είναι δυνατή μόνο σε συνάρτηση των εξόδων του δικτύου χωρίς να χρησιμοποιηθούν τα έξοδα των εσωτερικών επιπέδων. Υπάρχουν κάποιες πιο συχνά χρησιμοποιούμενες συναρτήσεις απόκλισης οι οποίες είναι :

•Τετραγωνική απόκλιση : Η τετραγωνική απόκλιση (Square Loss) αποτελεί μια απόκλιση για τη διαφορά στις αναμενόμενες και προβλεπόμενες εξόδους η οποία χρησιμοποιείται κυρίως στην παλινδρόμηση.

•Απόκλιση Hinge : Η χρήση της γίνεται κυρίως στην κατηγοριοποίηση στις μηχανές υποστήριξης διανυσμάτων.

•Λογαριθμική απόκλιση : Η χρήση της γίνεται κυρίως στην κατηγοριοποίηση.

•Λογιστική απόκλιση : Η λογιστική απόκλιση είναι στην ουσία η λογαριθμική απόκλιση συνδυαστικά με την σιγμοειδή συνάρτηση.

1.8 Οπισθοδιάδοση

Η πιο γνωστή μέθοδος σήμερα για να εκπαιδευτεί ένα νευρωνικό δίκτυο το οποίο αποτελείται από πολλά επίπεδα και χρησιμοποιείται στις περισσότερες εφαρμογές, είναι η μέθοδος οπισθοδιάδοσης (backpropagation) του λάθους. Η μέθοδος αυτή ακολουθεί έναν αλγόριθμο βελτιστοποίησης ο οποίος βασίζεται στην κλίση η οποία εκμεταλλεύεται τον κανόνα της αλυσίδας [24]. Ένα από τα κύρια χαρακτηριστικά της είναι η αποδοτική, αναδρομική και επαναληπτική μέθοδος, για να υπολογίσει τις ανανεώσεις των βαρών για τη βελτίωση του δικτύου, έως ότου να μπορέσει να εκτελέσει το έργο για το οποίο εκπαιδεύεται. Θεωρείται στενά συνδεδεμένη με τον αλγόριθμο Gauss-Newton. Σαν κεντρική ιδέα θεωρείται αρκετά απλή όπου το δίκτυο ξεκινά τη διαδικασία της μάθησης από τις τυχαίες τιμές των βαρών του. Σε περίπτωση που δοθεί μια λάθος απάντηση που είναι και το πιθανότερο, τότε γίνεται διόρθωση των βαρών ώστε λάθος να γίνει μικρότερο. Η διαδικασία αυτή επαναλαμβάνεται αρκετές φορές έτσι ώστε να ελαττώνεται σταδιακά το λάθος μέχρι να γίνει μικρό έως ανεκτό. Σε αυτό το σημείο το δίκτυο έχει μάθει από τα παραδείγματα που του δόθηκαν για να εκπαιδευτεί με την ακρίβεια την οποία πρέπει.

2. Συνεργατική ρομποτική σε επιτήρηση από πάνω προβολή: Ένα πλαίσιο για παρακολούθηση πολλαπλών αντικειμένων με ανίχνευση με χρήση της βαθιάς μάθησης

2.1 Συνεργατική ρομποτική

Η συνεργατική ρομποτική κέρδισε την προσοχή πολλών ερευνητών ενώ έχει αναδειχθεί ως η βασική τεχνολογία σε πολλούς τομείς όπως είναι ο εντοπισμός, η πλοήγηση, οι υπηρεσίες υγειονομικής περίθαλψης, η ψυχαγωγία, ο χειρισμός οικιακών εργασιών, οι μεταφορές, η μεταποίηση και η βιομηχανία. Καθώς αυξάνεται μέρα με τη μέρα η ανάγκη για έξυπνη επιτήρηση συστημάτων, εγκαταστάθηκαν σε κοινούς χώρους για λόγους παρακολούθησης και ασφάλειας, αρκετές οπτικές συσκευές όπως είναι αισθητήρες και κάμερες. Στο μεγαλύτερό τους μέρος τα συστήματα επιτήρησης αποτελούνται από μια δομή κεντρικής παρακολούθησης, δηλαδή ένα ενιαίο δωμάτιο που καταγράφονται βίντεο από πολλαπλές κάμερες ενώ παρατηρούνται ή παρακολουθούνται από ανθρώπους.

Μπορεί ωστόσο η παρακολούθηση πολλαπλών ροών βίντεο να είναι αρκετά κουραστική για τους χειριστές ασφαλείας. Συνεπώς, είναι επιθυμητή η χρήση της συνεργατικής ρομποτικής ώστε να παραχθεί ένα τέτοιο ευφυές αυτοματοποιημένο συνεργατικό ρομποτικό σύστημα, το οποίο θα παρακολουθεί και θα αναλύει πολλαπλές ροές από βίντεο βοηθώντας τους χειριστές το περισσότερο δυνατό. Ένας άλλος τρόπος που μπορεί να βοηθήσει η συνεργατική ρομποτική, είναι στο να επεκταθούν οι δυνατότητες του συστήματος της επιχείρησης χρησιμοποιώντας τεχνολογία οπτικής επεξεργασίας και έξυπνες συσκευές κάμερας. Οι πρωτεύοντες στόχοι μιας τέτοιας συνεργατικής επιτήρησης η οποία είναι βασισμένη σε ρομποτικά συστήματα είναι η παροχή χρήσιμων πληροφοριών σε ένα συγκεκριμένο περιβάλλον ή σκηνή για διάφορες δραστηριότητες.

Παρέχοντας τις πληροφορίες μπορεί να βοηθήσει στην ανάλυση δραστηριότητας, γεγονότων και συμπεριφοράς, να παρακολουθεί αντικείμενα και να ανιχνεύει πρότυπα κίνησης. Όλα αυτά θεωρούνται σημαντικά έχοντας ένα ευρύ φάσμα πραγματικών εφαρμογών όπως είναι η πλοήγηση και η τοποθεσία, η ρομποτική αλληλεπίδραση ανθρώπου υπολογιστή(HCI) [25][26], η αναγνώριση προσώπου [27], τα αυτόνομα οχήματα [28] και η ανάλυση ασφάλειας [29][30]. Οι εφαρμογές αυτές ενδέχεται να βρουν δυσκολίες από αρκετούς παράγοντες συμπεριλαμβανομένων της στενής αλληλεπίδρασης αντικειμένων, τις απότομες μεταβολές στην κίνηση, τις ακατάστατες σκηνές και τις απόψεις κάμερας, τις συνθήκες φωτισμού, διαφορετικά

υπόβαθρα, τις παραλλαγές στην εμφάνιση των αντικειμένων όπως είναι οι πόζες, προσανατολισμοί σώματος και τα μεγέθη.

Για να μπορέσουν να αντιμετωπιστούν όλα αυτά χρησιμοποιήθηκε μια σειρά από μεθόδους μηχανικής μάθησης, υπολογιστικής μάθησης και βαθιάς μάθησης, που παρέχουν αποτελεσματικές και ισχυρές λύσεις [28]. Ο μεγαλύτερος αριθμός των αναπτυγμένων προσεγγίσεων θεωρούνται κυρίως τα παραδοσιακά χειροποίητα χαρακτηριστικά [30] μαζί με διαφορετικούς ταξινομητές μηχανικής μάθησης [30]. Μια πρόσφατη πρόοδος στα μοντέλα βαθιάς μάθησης, μπορεί να κάνει τις μεθόδους παρακολούθησης αντικειμένων [31] και τον εντοπισμό αντικειμένων [30], πιο αποτελεσματικούς όσον αφορά την ακρίβεια και την ταχύτητα υπολογισμού. Από τα βασικά πλεονεκτήματα των μοντέλων αυτών, είναι η επιλογή των πιο σημαντικών χαρακτηριστικών αντικειμένων, η αυτόματη λειτουργία καθώς και μια μεγαλύτερη πιθανότητα σωστής ταξινόμησης συγκριτικά με τα χειροποίητα χαρακτηριστικά τα οποία απαιτούν επιπλέον εκπαίδευση εικόνων [32].

Επιπρόσθετα, τα μοντέλα αυτά συνήθως έχουν περισσότερη διακριτική δύναμη όσον αφορά την ταξινόμηση αντικειμένων πολλαπλών κλάσεων ανεξαρτήτως κλίμακας, απόφραξης, κατάστασης υποβάθρου, φωτισμού, εμφάνισης σχετικά με την κάμερα, τη θέση τους, τη στάση και το μέγεθος τους. Ο μεγαλύτερος αριθμός των τεχνικών συνήθως βασίζεται στη βαθιά μάθηση η οποία αναπτύσσεται γενικά για ανίχνευση ή οριζόντια παρακολούθηση αντικειμένων μετωπικής όψης, χωρίς να χρησιμοποιείται συλλογική εγκατάσταση ρομποτικής. Διαφορετικοί ερευνητές ανέπτυξαν μεθόδους παρακολούθησης αντικειμένων όπως είναι οι [31][33][34][35], βασισμένοι στην ασύμμετρη με τοπική προοπτική κάμερας .

Με τη χρήση όμως της κάτοψης της προοπτικής παρέχεται μεγαλύτερη ορατότητα του αντικειμένου και της σκηνής στην κάμερα σύμφωνα με την παρακάτω εικόνα.

Εικόνα 1 : Ανίχνευση και παρακολούθηση αντικειμένων μετωπικής όψης: δείγματα εικόνων που παρουσιάζουν επίσης διακύμανση στην εμφάνιση του αντικειμένου (κλίμακα, μέγεθος, στάση).

2.2 Μέθοδοι βασισμένες σε χαρακτηριστικά

Έχουν σχεδιαστεί εξελιγμένες μέθοδοι από διάφορους ερευνητές οι οποίες είναι βασισμένες σε χαρακτηριστικά όπως είναι βάσει σχήματος, κλίμακας μετασχηματισμών αμετάβλητων χαρακτηριστικών (SIT) [36], χαρακτηριστικά που μοιάζουν με Haar [37], τοπικά δυαδικά μοτίβα (LBP) [36], παραδοσιακά χειροποίητα χαρακτηριστικά συμπεριλαμβανομένων των ιστογραμμάτων προσανατολισμένης κλήσης (HOG) [37] και έγχρωμα ιστογράμματα [38] και χαρακτηριστικά [39]. Οι μέθοδοι αυτές εξήγαγαν κυρίως το πιο σημαντικό αντικείμενο των χαρακτηριστικών που χρησιμοποιούνται περαιτέρω για δοκιμή και εκπαίδευση των αλγορίθμων μηχανικής μάθησης όπως είναι η δομική μάθηση [40], το δάσος Hough [41], το τυχαίο δάσος [42], η ενίσχυση [43] και η υποστήριξη διανυσματικής μηχανής (SVM) [44].

Οι παραδοσιακές μέθοδοι παρακολούθησης αντικειμένων εκτός από το αντικείμενο της ανίχνευσης μπορούν να κατηγοριοποιηθούν και για διαφοροποίηση βάσει χαρακτηριστικών μεθόδων, της οπτικής ροής και καρέ. Οι παραδοσιακοί αλγόριθμοι παρακολούθησης είναι επικεντρωμένες στη θέση του στόχου με μέθοδο πρόβλεψης με χρήση φίλτρου και ακολουθίες βίντεο όπως είναι το φίλτρο Kalman . Κάποιοι

ερευνητές χρησιμοποίησαν ορισμένα χαρακτηριστικά εμφάνισης όπως είναι η υφή, το χρώμα και το σχήμα, ώστε να μπορέσουν να παρακολουθήσουν διαφορετικά αντικείμενα σε όλο το πλαίσιο [45]. Κάποιοι άλλοι ερευνητές υποστήριξαν ένα πρότυπο αραιών μεθόδων [46] το οποίο βασίζεται στην αναπαράσταση για να παρακολουθούν αντικείμενα, επικεντρώνοντας στην περιοχή αναζήτησης παρόμοια με το στόχο παρακολούθησης.

Για να μπορέσει να γίνει η διαφοροποίηση του προσκήνιο και του φόντου, κάποιοι ερευνητές ανέπτυξαν αλγορίθμους βασισμένους στη διακριτική μάθηση χαρακτηριστικών [47]. Πολλοί από τους αλγόριθμους αυτούς χρησιμοποίησαν παρόμοια χαρακτηριστικά αντικειμένων σε μεθόδους ανίχνευσης αντικειμένων. Οι Duffner and Garcia [48] παρουσίασαν μια μέθοδο παρακολούθησης αντικειμένων συνδυάζοντας πολλαπλή ανίχνευση χαρακτηριστικών με την τεχνική της πιθανολογικής κατάτμησης. Οι αναπτυγμένες τεχνικές παρακολούθησης και ανίχνευσης, βασίζονται πλέον σε σύνολα δεδομένων τα οποία καταγράφονται από μετωπική όψη και που μπορεί να υποφέρουν από το πρόβλημα της απόφραξης. Για να μπορέσει να ξεπεραστεί το πρόβλημα αυτό, αρκετοί ερευνητές χρησιμοποίησαν και πρότειναν προοπτική κάμερας κάτοψης.

2.3 Μέθοδοι που βασίζονται στη βαθιά μάθηση

Έχουμε μοντέλα βαθιάς μάθησης ανίχνευσης ενός σταδίου και δύο σταδίων. Οι [49] πρότειναν ένα αντικείμενο μοντέλο ανίχνευσης δύο σταδίων με τη χρήση ενός συγκεκριμένου δικτύου συγκέντρωσης πυραμίδων για αναγνώριση αντικειμένου. Το μοντέλο αποτελείται από επίπεδα CNN, τα οποία επιτρέπουν να δημιουργηθεί αναπαράσταση χαρακτηριστικών σταθερού μήκους ανεξάρτητα από την αλλαγή κλίμακας της εικόνας. Μια άλλη μέθοδος ανίχνευση αντικειμένων δύο σταδίων προτείνεται από τους [50], στην οποία επιτρέπεται η εκπαίδευση του ανιχνευτή και αναδρομέα οριοθέτησης πλαισίου ταυτόχρονα στο ίδιο δίκτυο με τη χρήση του συνόλου δεδομένων Pascal VOC.

Οι Ren et al. [51] ανέπτυξαν το πρώτο από άκρο σε άκρο μοντέλο σε πραγματικό χρόνο ώστε να ανιχνεύσει έναν στόχο. Οι συγγραφείς στο Faster-RCNN, κινούνται σε μεμονωμένα μπλοκ ανίχνευσης της περιοχής πρότασης, της ενσωματωμένης εξαγωγής χαρακτηριστικών δικτύου από άκρο σε άκρο, του πλαισίου οριοθέτησης παλινδρόμηση,ς καθιστώντας με αυτό τον τρόπο το δίκτυο να γίνεται πιο γρήγορα συγκριτικά με τα προηγούμενα μοντέλα. Το COCO [52] χρησιμοποιήθηκε για να εκπαιδευτεί και να δοκιμάσει το αναπτυγμένο μοντέλο. Οι [53] ανέπτυξαν βασισμένοι στους [51] ένα άλλο μοντέλο ανίχνευσης αντικειμένων δύο σταδίων,

δίνοντας το πλεονέκτημα για να ταξινομηθούν αντικείμενα με παραλλαγές μεγάλου εύρους κλίμακας. Το πρώτο πλαίσιο ανίχνευσης ενός σταδίου, αναπτύχθηκε από τους [54] προτείνοντας περιοχές που είχαν χαρακτηριστικά CNN. Κάθε πρόταση περιοχής κλιμακώνεται στην εικόνα σταθερού μεγέθους ενώ τροφοδοτείται εκπαιδεύοντας το μοντέλο CNN.

Οι [55] παρουσίασαν ένα μοντέλο ανίχνευσης αντικειμένου ενός σταδίου το οποίο ονομάζεται YOLO, και είναι ένα ενιαίο νευρωνικό δίκτυο από χρησιμοποιείται σε ολόκληρη την εικόνα εξάγοντας περιοχές και προβλέποντας πιθανότητες και οριοθέτηση για κάθε πρόταση περιοχής. Οι Redmon and Farhadi a[56] και Redmon and Farhadi b[57] βελτίωσαν το προηγούμενο μοντέλο ενισχύοντας περαιτέρω την ακρίβεια ανίχνευσης του. Οι Liu et al. [58] πρότειναν ένα άλλο μοντέλο ανίχνευσης ενός σταδίου που ονομάστηκε ανιχνευτής πολλαπλών κουτιών μονής βολής (SSD). Εισήγαγαν επίσης μια ανίχνευση δομής πολλαπλών αναφορών και ανάλυσης. Τα μοντέλα βαθιάς μάθησης χρησιμοποιήθηκαν επίσης από τους Fortino et al. [36] Gidaris and Komodakis [59] Dai et al. [60] δείχνοντας την ακρίβεια, την αποτελεσματικότητα και τη στιβαρότητα τους σε διαφορετικές εφαρμογές ανίχνευσης αντικειμένων.

Οι Wang et al. [61] ανέπτυξαν μια χαρακτηριστική επανασχεδιασμένη οπτική μέθοδο παρακολούθησης αντικειμένων χρησιμοποιώντας κυρίως ένα προεκπαιδευμένο ιεραρχικό δίκτυο ανίχνευσης.

Με όμοιο τρόπο εκμεταλλευόμενοι το δίκτυο προτάσεων περιοχής οι Ning et al [62] χρησιμοποίησαν ένα μοντέλο δικτύου επαναλαμβανόμενης συνέλιξης για να παρακολουθήσουν αντικείμενα. Οι Wang and Yeung [63] ανέπτυξαν μια ηλεκτρονική τεχνική παρακολούθησης αντικειμένων βασισμένη σε νευρωνικό δίκτυο με τη χρήση συνόλου δεδομένων μετωπικής όψης. Οι Fan et al. [64] χρησιμοποίησαν ένα προ εκπαιδευμένο δίκτυο βαθιών στρωμάτων, ώστε να παρακολουθήσουν τον άνθρωπο μέσα από τις εικόνες μετωπικής όψης. Επιπλέον, οι Wang and Yeung [65] ανέπτυξαν ένα γενικό χαρακτηριστικό το οποίο είναι βασισμένο στα χαρακτηριστικά των μεθόδων παρακολούθησης αντικειμένων που είναι ανθεκτικά σε πολλές παραλλαγές εμφάνισης των αντικειμένων. Μια άλλη παρακολούθηση αντικειμένων η οποία είναι βασισμένη στο CNN αναπτύχθηκε από τους Zhu et al. [66], Kuen et al.[67], παρόμοια με τους Gidaris and Komodakis [68], με τη χρήση πληροφοριών προτάσεις για την παρακολούθηση κατοικίδιων ζώων.

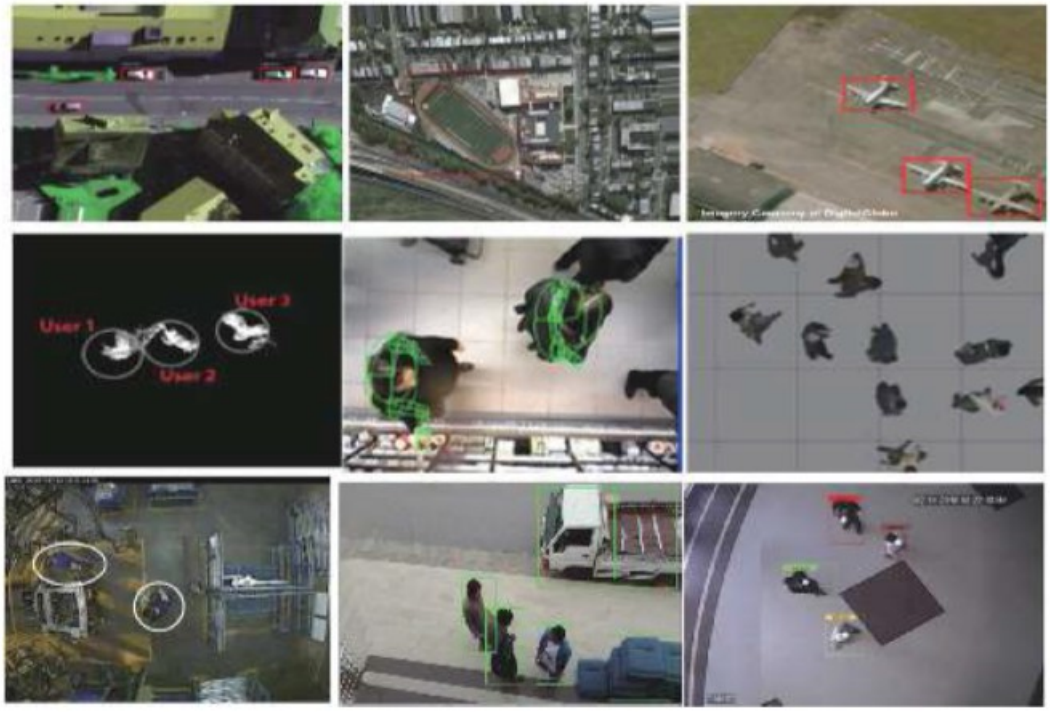
Το μοντέλο CNN εγκρίθηκε από τους Hong et al. [69], παρήγαγε διακριτικούς χάρτες εξέχουσας σημασίας οι οποίοι συνδυάστηκαν περαιτέρω για να

παρακολουθήσουν το αντικείμενο από τη μετωπική άποψη με SVM. Για να εξαχθούν τα χαρακτηριστικά των αντικειμένων, υιοθετήθηκε ο ταξινομητής DLT [65] και CNNSVM [69]. Οι Cui et al. [70] πρότειναν μια μέθοδο παρακολούθησης αντικειμένων βασισμένη σε RNN με τη χρήση φίλτρων συσχέτισης. Επίσης αναπτύχθηκαν από τους Kuen et al. [67], Cui et al. [70], ιχνηλάτες οι οποίοι βασίζονται στο CNN. Τα περισσότερα από τα ανεπτυγμένα μοντέλα βαθιάς μάθησης χρησιμοποιούν μετωπική προβολή εικόνων. Κάποιοι ερευνητές εκτελούν αντικείμενα εργασιών παρακολούθησης και ανίχνευσης χρησιμοποιώντας δορυφορικές και εναέριες εικόνες [71], [72].

Κάποιοι άλλοι χρησιμοποίησαν τη βαθιά μάθηση για να δουν την κορυφή και να ανιχνεύσουν και να παρακολουθήσουν αντικείμενα αλλά το έργα τους ήταν κυρίως εστιασμένο σε ένα μεμονωμένο αντικείμενο κλάσης ή πρόσωπο [71], [73]. Οι [74] εφάρμοσαν μοντέλα ανίχνευσης δύο σταδίων για την κάτοψη αντικειμένων πολλαπλών κλάσεων. Οι Ahmed et al. [74] χρησιμοποίησαν τα Mask-RCNN και Faster-RCNN για να τμηματοποίησουν αντικείμενα από κάτοψη.

2.4 Αλγόριθμοι παρακολούθησης αντικειμένων

Στην παρακάτω εικόνα απεικονίζεται ότι για σκοπούς παρακολούθησης η έξοδος του κάθε μοντέλου ανίχνευσης αποθηκεύεται με τη μορφή λίστας.



Εικόνα 2 : Ανίχνευση και παρακολούθηση αντικειμένου κάτοψης: δείγματα εικόνων που δείχνουν διαφοροποίηση στην εμφάνιση του αντικειμένου (κλίμακα, μέγεθος, στάση) σε προοπτική κάτοψης χειρίζονται επίσης το πρόβλημα απόφραξης.

Η προετοιμασία του ιχνηλάτη γίνεται με τη χρήση διαφορετικών μεθόδων παρακολούθησης. Μετά την αρχικοποίηση του tracker, επιλέγεται χειροκίνητα ο αλγόριθμος παρακολούθησης ο οποίος θα ελέγχει το αντικείμενο για πληροφορίες στη λίστα αρχίζοντας να τις παρακολουθεί από την επάνω προβολή. Στη λειτουργική μονάδα παρακολούθησης αντικειμένων, στην περίπτωση που η τιμή του αντικειμένου ανίχνευσης της λίστας είναι μεγαλύτερη από μηδέν, γίνεται συνεχόμενη ενημέρωση της παρακολούθησης του αντικειμένου. Παρακάτω παρουσιάζονται 6 διαφορετικοί αλγόριθμοι παρακολούθησης οι : GOTURN, MEDIANLOW, TLD, MIL, KC και το BOOSTING το οποίο είναι υλοποιημένο στο OpenCV. Οι αλγόριθμοι αυτοί αρχικά προτάθηκαν για την παρακολούθηση αντικειμένων μετωπικής όψης. ανιχνεύτηκαν πληροφορίες οριοθέτησης οι οποίες εξάγονται από το αντικείμενο, ενώ τα μοντέλα ανίχνευσης χρησιμοποιούνται για να δημιουργηθεί μια λίστα παρακολούθησης. Ο ιχνηλάτης αποθηκεύει στη λίστα τις πληροφορίες αυτές και παρακολουθεί το σύνολο δεδομένων κορυφαίας προβολής των αντικειμένων. Τα αποτελέσματα οπτικοποίησης της παρακολούθησης επεξεργάζονται από τους αλγόριθμους στην ενότητα των πειραματικών αποτελεσμάτων

•BOOSTING Tracker: Αποτελεί μια τεχνική ισχυρής παρακολούθησης σε πραγματικό χρόνο η οποία αναπτύχθηκε από τους Grabner et al. [75] με τη χρήση 20 ps (καρέ ανά δευτερόλεπτο). Στους περιορισμούς της τεχνικής αυτής λήφθηκε υπόψη η παρακολούθηση ενός αντικειμένου ως πρόβλημα δυαδικής ταξινόμησης υπό την έννοια του αντικειμένου και του φόντου. Για την παρακολούθηση θεωρούνται τα περισσότερα διακριτικά χαρακτηριστικά. Είναι σχετικά αργός και δεν λειτουργεί καλά συγκριτικά με τους άλλους αλγορίθμους.

•MIL Tracker: οι. Babenko et al. [76] ανέπτυξαν μια μέθοδο για να παρακολουθούν αντικείμενα η οποία είναι γνωστή ως μάθηση πολλαπλών περιπτώσεων (MIL), η οποία μπορεί να επιλύσει το πρόβλημα μάθησης του μοντέλου προσαρμοστικής εμφάνισης. Χρησιμοποιεί ένα διακριτικό ταξινομητή για να ταξινομήσει το αντικείμενο από το φόντο. Επιτρέπει να ενημερωθεί το μοντέλο εμφάνισης το οποίο χρησιμοποιεί patches εικόνας χωρίς να έχει γνώση ποια ενημερωμένη έκδοση κώδικα εικόνας θεωρείται υπεύθυνη για τη λήψη του αντικείμενου ενδιαφέροντος. Η ακρίβεια είναι καλύτερη συγκριτικά με τον αλγόριθμο παρακολούθησης BOOSTING

.

•KC Tracker: Για να μπορέσει να ξεπεραστεί η υπολογιστική επιβάρυνση του διακριτικού ταξινομητή, οι Henriques et al. [77] ανέπτυξαν ένα μοντέλο γρήγορης ανίχνευσης και εκμάθησης με τη χρήση μιας γρήγορης μεταμόρφωσης Fourier. Οι Henriques et al. [77] χρησιμοποίησαν μια μηχανή με πυρήνα χώρου με γραμμικούς ταξινομητές. Οι συνέπειες αναλύθηκαν από το μοντέλο με τη χρήση πυκνής δειγματοληψίας στην παρακολούθηση. Θεωρείται πιο γρήγορος και από τους δύο προηγούμενους αλγορίθμους, αλλά δεν χειρίζεται την απόφραξη ενώ μπορεί να υποφέρει από αποτυχία όταν υπάρχει διακύμανση της θέσης και του μεγέθους ενός αντικειμένου.

•TLD Tracker: Οι Kalal et al. [68], εξέταζαν μακροπρόθεσμος αλγόριθμους παρακολούθησης σε διαφορετικά αντικείμενα σε ακολουθίες βίντεο. Το ανεπτυγμένο πλαίσιο για τη μάθηση και παρακολούθηση καθώς και την ανίχνευση (TLD) αποσυνθέτει τη μακροπρόθεσμη εργασία παρακολούθησης. Ο ιχνηλάτης εντοπίζει μέσω παρατήρησης την εμφάνιση του αντικειμένου μέσα από ακολουθίες βίντεο, παρακολουθώντας κάθε καρέ, ενώ πάσχει πάρα πολύ από ψευδώς θετικά αποτελέσματα.

•MEDIANLOW Tracker: οι Kalal et al. [78] ανέπτυξαν μια μέθοδο η οποία είναι βασισμένη σε σφάλμα Forward-Backward. Μέτρησαν τις εμπρός και προς τα πίσω διαφορές ανάμεσα στις δύο τροχιές του στόχου. Ο προτεινόμενος αλγόριθμος έκανε ανίχνευση των σφαλμάτων βοηθώντας να ανακαλυφθούν οι αστοχίες παρακολούθησης και να επιλεχθούν οι σωστές διαδρομές παρακολούθησης. Είναι κατάλληλος για αναφορά αποτυχιών αλλά εξαντλείται κάθε φορά που υπάρχει ένα τεράστιο άλμα ή κάποια μεταβολή στην κίνηση κάποια ξαφνική αλλαγή στην εμφάνιση και γρήγορη κίνηση.

•GOTURN Tracker: Η γενική παρακολούθηση αντικειμένων μπορεί να γίνει με τη χρήση δικτύων παλινδρόμησης GOTURN [79] βασισμένη σε επίπεδα CNN. Η αρχιτεκτονική του κατά κύριο λόγο είναι εκπαιδευμένη κυρίως στις ακολουθίες βίντεο μετωπικής όψης σε χιλιάδες καρέ περικοπής. Από τον αλγόριθμο παρέχονται εξαιρετικά αποτελέσματα ενώ μπορεί να χειριστεί πολλές παραλλαγές όπως είναι η παραμόρφωση, οι αλλαγές φωτισμού και η άποψη κατά τη διάρκεια της παρακολούθησης χωρίς όμως να μπορεί να χειριστεί καλά την απόφραξη.

3. Τρισδιάστατη παρακολούθηση ανίχνευσης πολλαπλών αντικείμενων

3.1 Βασική γραμμή για τρισδιάστατη παρακολούθηση πολλαπλών αντικειμένων

Το MOT, δηλαδή η παρακολούθηση πολλαπλών αντικειμένων, αποτελεί μια από τις βασικές τεχνολογίες στοιχείων για πολλές εφαρμογές όρασης, όπως είναι η ευθυγράμμιση προσώπου βίντεο [80], η πρόβλεψη σύγκρουσης ρομπότ [81] και η αυτόνομη οδήγηση [82]. Εξαιτίας ότι σημειώθηκε σημαντική πρόοδος στην ανίχνευση αντικειμένων, σημειώνεται μεγάλη πρόοδος και στο MOT. Στην κατηγορία αυτοκινήτων για παράδειγμα, στο σημείο αναφοράς KITTI [83] MOT, το MOTA δηλαδή η ακρίβεια παρακολούθησης πολλαπλών αντικειμένων βελτιώθηκε από 57,03 σε 84,24 σε δύο χρόνια. Αν και βελτιώθηκε σημαντικά η ακρίβεια, υπάρχει το υπολογιστικό κόστος .

3.2 Μια βασική γραμμή και νέες μετρήσεις αξιολόγησης

Το MOT αποτελεί ένα απαραίτητο συστατικό για πολλές εφαρμογές σε πραγματικό χρόνο όπως είναι η υποβοήθηση στη ρομποτική και η αυτόνομη οδήγηση. Καθώς υπάρχουν μεγάλες εξελίξεις στην ανίχνευση αντικειμένων, σημειώνεται μεγάλη πρόοδος στο MOT. Καθώς υπάρχει ενθάρρυνση από την πρόοδο αυτή, παρατηρείται ότι η εστίαση στην ακρίβεια για την καινοτομία, έχει ως κόστος πρακτικούς παράγοντες όπως είναι η απλότητα του συστήματος και η υπολογιστική απόδοση. Οι μέθοδοι αιχμής απαιτούν συνήθως ένα μεγάλο υπολογιστικό κόστος [84], και έτσι η απόδοση σε πραγματικό χρόνο αποτελεί μια πραγματική πρόκληση.

3.3 Κοινή ανίχνευση αντικειμένων και παρακολούθηση πολλαπλών αντικειμένων με νευρωνικά δίκτυα γραφημάτων

Η συσχέτιση δεδομένων [85] και η ανίχνευση αντικειμένων [86] αποτελούν δύο στοιχεία της παρακολούθησης πολλαπλών αντικειμένων MOT, τα οποία θεωρούνται απαραίτητα για να υπάρχει αντίληψη σε ρομποτικά συστήματα. Ενώ σε προηγούμενες εργασίες τα MOT προσεγγίζονται συχνά με διαδικτυακό τρόπο, αν χρησιμοποιηθεί εντοπισμός με ανίχνευση αγωγού, όπου ο ανιχνευτής εξάγει ανίχνευσης ακολουθούμενες από μια μονάδα συσχέτισης δεδομένων, η οποία αντιστοιχίζει τις ανιχνεύσεις με παρελθοντικά tracklets, έχοντας στόχο τη δημιουργία σχηματισμού νέων tracklets μέχρι το τρέχον πλαίσιο. Οι μονάδες συσχέτισης

δεδομένων και ανιχνευτή, εκπαιδεύονται χωριστά στις εργασίες αυτές. Εάν υπάρξει ωστόσο, μια ξεχωριστή διαδικασία βελτιστοποίησης, δεν μπορεί να υπάρξει επαναδιάδοση των σφαλμάτων μέσα από ολόκληρο το σύστημα MOT.

Δηλαδή, καθένα βελτιστοποιείται μόνο ως προς το δικό τους τοπικό βέλτιστο αλλά όχι ως προς τον συνολικό στόχο του MOT. Κάτι τέτοιο μπορεί να χωρίσει τη διαδικασία βελτιστοποίησης η οποία χρησιμοποιείται σε προηγούμενες εργασίες και συχνά αποδίδει μια υποβέλτιστη εκτέλεση. Για τη βελτίωση της απόδοσης διερευνείται το πώς μπορεί να βελτιστοποιηθούν από κοινού η συσχέτιση δεδομένων και η ανίχνευση αντικειμένων που αναφέρονται στο κοινό πλαίσιο MOT, και εντός της άρθρωσης του πλαισίου MOT, πώς μπορεί να γίνουν περισσότερο διακριτικά τα χαρακτηριστικά. Για να μπορέσει να αντιμετωπιστεί το κοινό πρόβλημα MOT, εξερευνήθηκαν διαφορετικές κατευθύνσεις. Προτείνεται να ενοποιηθεί ο ανιχνευτής αντικειμένων με ένα ανιχνευτή ενός αντικειμένου χωρίς μοντέλο, στον οποίο ο ιχνηλάτης υποχωρεί απευθείας στη θέση του κάθε αντικείμενο το οποίο εντοπίστηκε στο προηγούμενο πλαίσιο, στο τρέχον πλαίσιο.

Η ανεξάρτητη παρακολούθηση κάθε αντικειμένου μπορεί να ωθήσει τη συσχέτιση δεδομένων λύνοντας και το πρόβλημα φυσικά εντός του τρέχοντος πλαισίου. Προτείνεται η επέκταση ενός ανιχνευτή αντικειμένων με την πρόσθεση ενός κλάδου επαναπροσδιορισμού Re-ID [87] ή αλλιώς κλάδου επαλήθευσης ταυτότητας, ο οποίος θα εξάγει χαρακτηριστικά αντικειμένων για να αντιστοιχηθούν μεταξύ των πλαισίων. Οι [88], πρότειναν επίσης έναν σωλήνα αγκύρωσης. Ο σωλήνας αγκύρωσης διαφορετικά από το κουτί αγκύρωσης το οποίο χρησιμοποιείται σε ανιχνευτές βάσης αγκύρωσης, μπορεί να αντιπροσωπεύσει μια ακολουθία οριοθέτησης κουτιών σε μια λίστα πλαισίων, δηλαδή ένα κουτί ανά πλαίσιο. Ως είσοδος μπορεί να δοθεί ένα βίντεο κλιπ, ενώ αληθινά θετικά σωληνάρια μπορούν να χρησιμοποιηθούν αφού βρεθούν ως έξοδοι tracklet, κάτι που μπορεί να λύσει το πρόβλημα της άρθρωσης MOT σε μία μόνο βολή.

Παρόλο που οι προηγούμενες κοινές μέθοδοι MOT είχαν εντυπωσιακή απόδοση, παρατηρείται ότι εξήχθη χαρακτηριστικό για το μεμονωμένο αντικείμενο σωλήνα ή τροχιά, το οποίο είναι ανεξάρτητο το ένα από το άλλο, ενώ αγνοούνται οι σχέσεις αντικειμένων. Τέτοιες σχέσεις αντικειμένων θεωρούνται χρήσιμες τόσο για τη σχέση των δεδομένων, αλλά όσο και για να ανιχνευθούν τα αντικείμενα. Για την ανίχνευση αντικειμένων στο MOT, για παράδειγμα, στην περίπτωση συνύπαρξης σχετικών αντικειμένων αν π χ δύο πεζοί περπατούν μαζί, υπάρχει πιθανότητα συνύπαρξης στο

τελευταίο καρέ καθώς και αυτών στο τρέχον πλαίσιο μιας κοντινής τοποθεσίας. Αν αυξηθεί η βαθμολογία ομοιότητας δύο αντικειμένων στα πλαίσια που έχουν υψηλή αυτοπεποίθηση, δηλαδή πιθανολογείται ότι τα δυο αυτά αντικείμενα έχουν την ίδια ταυτότητα, τότε για τη συσχέτιση δεδομένων, η βαθμολογία ομοιότητας ανάμεσα σε οποιαδήποτε από αυτά τα δύο αντικείμενα καθώς και άλλα αντικείμενα θα πρέπει να αποσιωπηθεί ώστε να αποφευχθεί η σύγχυση στη συσχέτιση δεδομένων.

Για να μπορέσει να επιτευχθεί μια καλύτερη απόδοση για το κοινό πλαίσιο MOT, η μέθοδος πρέπει να σχεδιαστεί ώστε να αξιοποιούνται οι σχέσεις αντικειμένου. Καθώς οι [89] εκμεταλλεύτηκαν τις σχέσεις αντικειμένου στη συσχέτιση δεδομένων, περιορίστηκαν σε ασυνάρτητα MOT, όπου βελτιστοποιείται ξεχωριστά ο ανιχνευτής. Συνεπώς, οι σχέσεις αντικειμένων δεν μπορούν να χρησιμοποιηθούν για έναν ανιχνευτή που δεν είναι βέλτιστος. Για να αντιμετωπιστεί το παραπάνω θέμα προτείνεται ένα νέο παράδειγμα κοινής προσέγγισης MOT, το οποίο έχει τη δυνατότητα μοντελοποίησης των σχέσεων αντικειμένου, τόσο για να ανιχνευτούν αντικείμενα, όσο και για να συσχετιστούν δεδομένα. Συγκεκριμένα, για να μπορέσουν να αποκτηθούν πιο διακριτικά χαρακτηριστικά χρησιμοποιούνται δίκτυα Graph Neural Networks (GNN), για να μπορέσουν να εκμεταλλευτούν τις σχέσεις ανάμεσα στα αντικείμενα.

Τα αποκτηθέντα χαρακτηριστικά χρησιμοποιούνται στη συνέχεια για να μπορέσουν να θεωρηθούν σχέσεις αντικειμένων τόσο για εργασίες ανίχνευσης, όσο και για να συσχετιστούν δεδομένα. Το χαρακτηριστικό που εξάγεται με τη χρήση GNN, για κάθε αντικείμενο δεν θεωρείται πλέον απομονωμένο, αλλά αντιθέτως έχει τη δυνατότητα προσαρμογής μέσα από χαρακτηριστικά των σχετικών αντικειμένων του τόσο σε χρονικό όσο και χωρικό πεδίο.

4. DOT – Δυναμική παρακολούθηση αντικειμένων για Visual SLAM

4.1 Επισκόπηση Συστήματος DOT

Όταν εισάγεται ένα DOT, είναι είτε στερεοφωνικές εικόνες είτε RGB-D, είτε ένας συγκεκριμένος αριθμός βίντεο, ενώ η έξοδος του αποτελείται από μια μάσκα η οποία κωδικοποιεί τα δυναμικά και στατικά στοιχεία της σκηνής που μπορούν να χρησιμοποιηθούν άμεσα με συστήματα οδομετρίας ή SLAM. Η τμηματοποίηση παρουσιών που αποτελεί και το πρώτο μπλοκ, αντιστοιχεί σε ένα CNN το οποίο πρέπει να τμηματοποιήσει σε pixel όλα τα δυνητικά δυναμικά αντικείμενα. Από τη στιγμή που το DOT παρακολουθεί τη μάσκα από πλαίσιο σε πλαίσιο, η λειτουργία αυτή δεν χρειάζεται να γίνεται σε κάθε πλαίσιο. Ο διαχωρισμός και η εξαγωγή των σημείων όπου βρίσκονται τα δυναμικά αντικείμενα, γίνεται από το μπλοκ επεξεργασίας εικόνας. Η παρακολούθηση της πόζας της κάμερας γίνεται με τη χρήση μόνο του στατικού μέρους της σκηνής. Λαμβάνοντας υπόψη την κίνηση και τη στάση της κάμερας, καθώς και από το συγκεκριμένο μπλοκ για καθένα από τα τμηματοποιημένα αντικείμενα, γίνεται ανεξάρτητη εκτίμηση, δηλαδή παρακολούθηση αντικειμένων.

Στο επόμενο μπλοκ όπου αναφέρεται αν το αντικείμενο είναι σε κίνηση ή όχι γίνεται καθορισμός με τη χρήση γεωμετρικών κριτηρίων, εάν γίνεται επισήμανση των αντικειμένων ως δυνητικά δυναμικά από το δίκτυο εάν πράγματι κινούνται. Οι πληροφορίες αυτές χρησιμοποιούνται για να ενημερωθούν οι μάσκες, οι οποίες κωδικοποιούν τις δυναμικές και στατικές περιοχές κάθε πλαισίου, καθώς και να τροφοδοτηθούν τα συνδεδεμένα οπτικά συστήματα οδομετρίας ή SLAM. Τέλος, μπορούν να δημιουργηθούν από το DOT νέες μάσκες μέσα από τις εκτιμήσεις της κίνησης των αντικειμένων (Mask Propagation), κάτι που σημαίνει ότι κάθε πλαίσιο δεν είναι απαραίτητο να τμηματοποιείται από το δίκτυο. Δεδομένου ότι υπάρχει σημαντικός υπολογιστικός φόρτος στην τμηματοποίηση των παρουσιών, κάθε τέτοιο μπορεί να θεωρηθεί ένα σχετικό πλεονέκτημα του DOT συγκριτικά με άλλες μεθόδους τελευταίας τεχνολογίας.

4.2 Τμηματοποίηση περιπτώσεων

Με τη χρήση του Detectron2 [90] σε βάθος δικτύου, γίνεται τμηματοποίηση όλων των δυνητικά κινητών περιπτώσεων, οι οποίες είναι παρούσες σε μια εικόνα. Για να μπορέσει να ληφθεί σε μια ενιαία εικόνα όλη η κατάτμηση των μασκών, έγινε

τροποποίηση της εξόδου του δικτύου. Οι περιοχές της εικόνας οι οποίες δεν μπορούσαν να ταξινομηθούν στις πιθανές κινούμενες κατηγορίες, έλαβαν μια ετικέτα φόντου, κάτι που μπορεί να θεωρηθεί στατικό στα επόμενα μπλοκ, με τη χρήση της γραμμής βάσης τμηματοποίησης παρουσίας COCO σε μοντέλο με μάσκα R-CNN R50-FPN 3x [91]. Οι τάξεις περιορίστηκαν σε αυτές οι οποίες θεωρούνται δυνητικά κινητές όταν εξαιρεθούν οι άνθρωποι, καθώς η παρακολούθηση ατόμων θεωρείται πέρα από το πεδίο της ενότητας αυτής. Σε άλλες περιπτώσεις κατηγοριών, θεωρήθηκε απαραίτητο ότι το δίκτυο μπορούσε να βελτιωθεί χρησιμοποιώντας αυτά τα βάρη ως προπόνηση ή σημείο εκκίνησης από την αρχή με το δικό του σύνολο δεδομένων. Για την συνεπή παρακολούθηση αντικειμένων σε πολλά καρέ, έγινε συμπερίληψη ενός βήματος αντιστοίχισης ανάμεσα στις μάσκες, οι οποίες υπολογίζονται από το DOT και αυτών που παρέχονται από το δίκτυο. Κάποιες νέες ανιχνεύσεις οι οποίες δεν μπορούσαν να αντιστοιχιστούν με κανένα υπάρχον αντικείμενο, χρησιμοποιήθηκαν για να προετοιμαστούν οι νέες παρουσίες.

4.3. Παρακολούθηση κάμερας και αντικειμένων

Στο προηγούμενο βήμα παρουσιάστηκε η τμηματοποίηση παρουσίας και σε αυτό στοχεύεται να εκτιμηθεί η κίνηση της κάμερας και των δυναμικών αντικειμένων. Θεωρώντας δεδομένο ότι η κίνηση των αντικειμένων και της κάμερας συνδέονται στις εικόνες, γίνεται μια εκτίμηση δύο σταδίων. Πρώτα βρίσκεται η πόζα της κάμερας ως ένας σχετικός μετασχηματισμός , ο οποίος μετα αφαιρείται ώστε να υπολογιστεί η κίνηση του αντικειμένου. Η βελτιστοποίηση αυτή σχετίζεται με τις πρόσφατες προσεγγίσεις της άμεσης οπτικής οδομετρίας και SLAM [92], έχοντας στόχο την εύρεση της κίνησης η οποία ελαχιστοποιεί ένα φωτομετρικό σφάλμα επαναπροβολής.

4.4. Ποιότητα παρακολούθησης και ακραίες τιμές

Η κατάτμηση των σφαλμάτων και οι αλλαγές στις συνθήκες φωτισμού , μπορούν να επιδράσουν σημαντικά στην ακρίβεια των αντικειμένων και τις πόζες της κάμερας. Αναπτύχθηκαν αρκετές στρατηγικές οι οποίες εφαρμόστηκαν μετά την παρακολούθηση αντικειμένων σε βήματα, για να μειωθούν οι επιπτώσεις τους. Όσον αφορά την ποιότητα παρακολούθησης, η εμφάνιση των δυναμικών αντικειμένων μπορεί να αλλάξει σημαντικά προκαλώντας υψηλά σφάλματα παρακολούθησης.

5. Ανίχνευση αντικειμένων

Η Ανίχνευση αντικειμένων (object detection) αποτελεί μια τεχνική που επιτρέπει τον εντοπισμό και την ανίχνευση αντικειμένων σε ένα βίντεο ή μια εικόνα. Με το είδος αυτό του εντοπισμού και της ταυτοποίησης, μπορεί να χρησιμοποιηθεί η ανίχνευση των αντικειμένων για να καταμετρηθούν αντικείμενα σε μια σκηνή και για να προσδιοριστεί η παρακολούθηση των ακριβών τοποθεσιών τους. Συνήθως υπάρχει σύγχυση της ανίχνευσης αντικειμένων με την αναγνώριση εικόνων, συνεπώς θεωρείται σημαντικό να διευκρινιστούν οι διαφορές μεταξύ τους. Η αναγνώριση εικόνας μπορεί να εκχωρήσει μια ετικέτα σε μια εικόνα. Για παράδειγμα, μια φωτογραφία που παρουσιάζει ένα σκύλο λαμβάνει την ετικέτα ”σκύλος”. Στην περίπτωση που οι σκύλοι είναι 2, εξακολουθείται να λαμβάνεται η ετικέτα ”σκύλος”. Από την άλλη πλευρά, στην ανίχνευση αντικειμένων, σχεδιάζεται ένα πλαίσιο οριοθέτησης γύρω από κάθε σκύλο, επισημαίνοντας το πλαίσιο με την ετικέτα ”σκύλος”. Το μοντέλο μπορεί να προβλέψει πόσο είναι κάθε αντικείμενο και ποια ετικέτα πρέπει να εφαρμοστεί. Με τον τρόπο αυτό, παρέχονται περισσότερες πληροφορίες από την ανίχνευση αντικειμένων σχετικά με την αναγνώριση εικόνας.

5.1 Λειτουργίες και τύποι ανίχνευσης αντικειμένων

Η ανίχνευση των αντικειμένων σε γενικές γραμμές μπορεί να αναλυθεί σε προσεγγίσεις οι οποίες βασίζονται στη μηχανική μάθηση, καθώς και προσεγγίσεις οι οποίες βασίζονται στη βαθιά μάθηση. Οι πρώτες αποτελούν πιο παραδοσιακές προσεγγίσεις, με τις τεχνικές μηχανικής όρασης να χρησιμοποιούνται για την εξέταση διαφόρων χαρακτηριστικών μιας εικόνας, όπως είναι τα άκρα ή το χρωματικό ιστόγραμμα, όταν προσδιοριστούν οι ομάδες εικονοστοιχείων, οι οποίες μπορεί να ανήκουν σε ένα αντικείμενο. Τα χαρακτηριστικά αυτά, μπορούν στη συνέχεια να τροφοδοτηθούν σε ένα άλλο μοντέλο παλινδρόμησης το οποίο προβλέπει τη θέση που θα έχει το αντικείμενο, καθώς και την ετικέτα του. Από την άλλη πλευρά, ο δεύτερος τρόπος προσέγγισης, της βαθιάς μάθησης, χρησιμοποιεί συνελικτικά νευρωνικά δίκτυα (CNNs) για να πραγματοποιηθεί η ανίχνευση αντικειμένων χωρίς επιτήρηση. Τα χαρακτηριστικά αυτά δεν είναι απαραίτητο να εξαχθούν και να οριστούν χωριστά.

5.2 Αλγόριθμοι Προτάσεων Περιοχής

Χρησιμοποιώντας τη βαθιά μάθηση, γίνεται εμφάνιση περιοχών, τα χαρακτηριστικά των οποίων αποτελούν πρωτοποριακή προσέγγιση στο να ανιχνευθούν αντικείμενα.

Σε αυτή την υποενότητα παρουσιάζεται το R-CNN καθώς και μια σειρά βελτιώσεων που έγιναν σε αυτό: Fast R-CNN , Faster R-CNN και Mask R-CNN.

5.2.1 R-CNN

Η εποχή της βαθιάς μάθησης ξεκινά με τη χρήση του αλγορίθμου R-CNN [93]. Τα μοντέλα έχουν τη δυνατότητα να επιλέξουν πρώτα μέσα από πολλές προτεινόμενες περιοχές από μία εικόνα, ενώ τα πλαίσια αγκύρωσης αποτελούν έναν τύπο μεθόδου επιλογής για παράδειγμα, ενώ στη συνέχεια μπορούν να επισημάνουν τα πλαίσια οριοθέτησης καθώς και τις κατηγορίες. Στη συνέχεια, χρησιμοποιούν ένα νευρωνικό δίκτυο συνέλιξης για την εκτέλεση υπολογισμών , ώστε να γίνει εξαγωγή χαρακτηριστικών από κάθε προτεινόμενη περιοχή. Τέλος, τα χαρακτηριστικά κάθε προτεινόμενης περιοχής, χρησιμοποιούνται για την πρόβλεψη των κατηγοριών και των πλαισίων οριοθέτησης τους. Το R-CNN μπορεί να δημιουργήσει προτάσεις περιοχής ή πλαίσια οριοθέτησης με τη χρήση μιας διαδικασίας η οποία ονομάζεται Επιλεκτική Αναζήτηση (Selective search) [94]. Πιο αναλυτικά η αναζήτηση αυτή, μπορεί να εξετάσει την εικόνα μέσα από παράθυρα διαφορετικού μεγέθους ενώ για κάθε μέγεθος, ομαδοποιούνται γειτονικά εικονοστοιχεία ανά ένταση, χρώμα ή υφή, ώστε να γίνει η αναγνώριση των αντικειμένων. Εν συνεχεία ο R-CNN:

1.Δημιουργεί ένα σύνολο προτάσεων για πλαίσια οριοθέτησης.

2.Περνάει τις εικόνες με πλαίσια οριοθέτησης μέσα από ένα προ εκπαιδευμένο μοντέλο χρησιμοποιώντας μια μηχανή υποστήριξης διανυσμάτων (SVM), όταν αναγνωριστεί το καθένα από τα αντικείμενα στα πλαίσια οριοθέτησης.

3.Τελικά, ελέγχεται το κάθε πλαίσιο μέσα από ένα μοντέλο γραμμικής παλινδρόμησης, ώστε να γίνει εξαγωγή αυστηρότερων συντεταγμένων από τη στιγμή που θα αναγνωριστεί το αντικείμενο.

Κάποια τυχόν προβλήματα που μπορεί να παρουσιάσει το R-CNN είναι :

- Αρκετός χρόνος στο να εκπαιδευτεί το δίκτυο καθώς οφείλουν να κατηγοριοποιηθούν 2000 προτάσεις περιοχής ανά εικόνα.
- Δε μπορεί να γίνει η εφαρμογή τους σε πραγματικό χρόνο, καθώς χρειάζονται περίπου 47 δευτερόλεπτα για κάθε εικόνα
- Ο αλγόριθμος επιλεκτικής αναζήτησης θεωρείται ένας σταθερός αλγόριθμος και έτσι δεν μπορεί να υπάρξει μάθηση σε αυτό το επίπεδο. Κάτι τέτοιο μπορεί να οδηγήσει στο να δημιουργηθούν κακές υποψήφιες προτάσεις για περιοχές.

5.2.2 Fast R-CNN

Κάποια από τα μειονεκτήματα του R-CNN λύθηκαν με στόχο τη δημιουργία ενός ταχύτερου αλγόριθμου ανίχνευσης αντικειμένων του Fast R-CNN [95]. Η προσέγγισή του είναι παρόμοια με τον αλγόριθμο R-CNN. Αντί όμως να τροφοδοτούνται προτάσεις περιοχής στο CNN, τροφοδοτείται η εικόνα εισόδου στο CNN, ώστε να δημιουργηθεί ένας χάρτης χαρακτηριστικών. Από το χάρτη αυτόν εντοπίζεται η περιοχή των προτάσεων, η οποία παραμορφώνεται σε τετράγωνα και με τη χρήση ενός στρώματος ομαδοποίησης RoI, αναδιαμορφώνονται σε σταθερό μέγεθος, έτσι ώστε να μπορούν να τροφοδοτηθούν σε ένα πλήρως συνδεδεμένο επίπεδο. Από αυτό το διάνυσμα χαρακτηριστικών, χρησιμοποιείται ένα επίπεδο Softmax για την πρόβλεψη της κλάσης της προηγούμενης περιοχής, καθώς και των τιμών μετατόπισης του πλαισίου οριοθέτησης.

Ο λόγος για τον οποίο θεωρείται ταχύτερο από το R-CNN, είναι ότι δεν χρειάζεται η τροφοδότηση του συνελικτικού νευρωνικού δικτύου με 2.000 προτάσεις κάθε φορά. Αντί να γίνει αυτό, η λειτουργία συνέλιξης πραγματοποιείται μόνο μία φορά ανά εικόνα από την οποία δημιουργείται ένας χάρτης χαρακτηριστικών. Διαπιστώνεται ότι ο συγκεκριμένος αλγόριθμος είναι πολύ πιο γρήγορος τόσο σε ανίχνευση όσο και σε εκπαίδευση σχετικά με το R-CNN. Κατά τον έλεγχο της απόδοσης του κατά τη διάρκεια του χρόνων δοκιμής, συμπεριλαμβανομένων και των προτάσεων περιοχής, παρουσιάζεται σημαντική επιβράδυνση του αλγορίθμου, συγκριτικά με τη μη χρήση προτάσεων περιοχής. Συνεπώς, οι προτάσεις περιοχής μπορεί να σταθούν εμπόδια στον αλγόριθμο αυτό, καθώς επηρεάζουν την απόδοσή του.

Κάποια προβλήματα που μπορεί να παρουσιάσει, είναι η χρήση της επιλεκτικής αναζήτησης ως μέθοδο προτάσεων, ώστε να βρεθούν οι περιοχές ενδιαφέροντος κάτι που αποτελεί μια χρονοβόρα και αργή διαδικασία. Για να εντοπιστούν τα αντικείμενα χρειάζονται περίπου 2 δευτερόλεπτα ανά εικόνα, κάτι που θεωρείται συγκριτικά καλύτερο με τον R-CNN, αλλά όταν εξετάζονται μεγάλα σύνολα δεδομένων σε πραγματικό χρόνο τότε ακόμη και ο Fast R-CNN δεν παρουσιάζεται πλέον τόσο γρήγορος.

5.2.3 Faster-RCNN

Και οι δύο προηγούμενοι αλγόριθμοι χρησιμοποιούν την επιλεκτική αναζήτηση για να μπορέσουν να μάθουν τις προτάσεις της περιοχής, η οποία είναι μια χρονοβόρα και αργή διαδικασία επηρεάζοντας την απόδοση του δικτύου. Συνεπώς, οι Shaoqing Ren, Kaiming He, Ross Girshick και Jian Sun πρότειναν έναν αλγόριθμο ανίχνευσης

αντικειμένων, ο οποίος μπορεί να εξαλείψει τον αλγόριθμο επιλεκτικής αναζήτησης, επιτρέποντας στο δίκτυο να μάθει τις προτάσεις περιοχής [97]. Όπως και με το Fast R-CNN, η παροχή της εικόνας γίνεται ως είσοδος σε ένα συνελικτικό δίκτυο, το οποίο παρέχει έναν χάρτη συνελικτικών χαρακτηριστικών. Στο χάρτη δυνατοτήτων, για να προσδιοριστούν οι προτάσεις περιοχής, αντί να χρησιμοποιηθεί ο αλγόριθμος επιλεκτικής αναζήτησης, χρησιμοποιείται ένα ξεχωριστό δίκτυο για να προβλεφθούν οι προτάσεις της περιοχής.

Στη συνέχεια, οι προτάσεις αυτές να διαμορφώνονται με τη χρήση ενός επιπέδου συγκέντρωσης περιοχών ενδιαφέροντος (RoI), το οποίο χρησιμοποιείται στη συνέχεια για να κατατάξει την εικόνα μέσα στην προτεινόμενη περιοχή και να προβλέψει τις τιμές για τα πλαίσια οριοθέτησης. Η υλοποίηση του αλγορίθμου αυτή είναι σαφώς πιο γρήγορη από τους δύο προηγούμενους και μπορεί επομένως να χρησιμοποιηθεί για να ανιχνευθούν αντικείμενα σχεδόν σε πραγματικό χρόνο, καθώς ο χρόνος ανίχνευσης του σε μια εικόνα είναι περίπου στα 0,2 δευτερόλεπτα. Παρουσιάζει όμως και κάποια προβλήματα. Ένα από αυτά είναι ότι απαιτεί πολλά περάσματα σε κάθε εικόνα για να μπορέσει να εξάγει όλα τα αντικείμενα. Καθώς αποτελείται από διαφορετικά συστήματα τα οποία λειτουργούν διαδοχικά, η απόδοση των συστημάτων που βρίσκονται πιο μπροστά εξαρτάται από την απόδοση των συστημάτων που προηγούνται.

5.2.4 Mask R-CNN

Το Mask R-CNN αποτελεί ένα υπερσύγχρονο μοντέλο τμηματοποίησης αντικειμένων σε βίντεο και εικόνα, το οποίο αναπτύχθηκε μετά το Faster R-CNN. Έχει τη δυνατότητα διαχωρισμού διαφορετικών αντικειμένων σε ένα βίντεο ή εικόνα. Ως είσοδο δέχεται μια εικόνα, δίνοντας στη συνέχεια πλαίσια οριοθέτησης αντικειμένων, και τις μάσκες και κλάσεις πάνω από τα αντικείμενα [97]. Τα στάδια του είναι δύο. Πρώτα δημιουργεί προτάσεις οι οποίες είναι σχετικές με τις περιοχές στις οποίες μπορεί να υπάρξει ένα αντικείμενο βασισμένο στην εικόνα εισαγωγής. Δεύτερον, μπορεί να προβλέψει την κλάση του αντικειμένου, να δημιουργήσει μια μάσκα του αντικείμενου σε επίπεδο pixel, να βελτιώσει το πλαίσιο οριοθέτησης βασισμένο στην πρόταση του πρώτου σταδίου. Και τα δύο επίπεδα συνδέονται με τη δομή ενός δικτύου backbone.

Το Backbone είναι ένα βαθύ νευρωνικό δίκτυο τύπου FPN [98]. Αποτελείται από πλευρικές συνδέσεις, ένα πάνω κάτω μονοπάτι και ένα κάτω προς τα πάνω μονοπάτι. Μια κατώτατη διαδρομή μπορεί να είναι οποιοδήποτε CNN, συνήθως VGG ή ResNet, όπου εξάγονται λειτουργίες από ακατέργαστες εικόνες. Το από πάνω προς τα

κάτω μονοπάτι μπορεί να δημιουργήσει ένα χάρτη πυραμίδας χαρακτηριστικών (FPN), κάτι που είναι παρόμοιο σε μέγεθος με το μονοπάτι από κάτω προς τα πάνω. Οι συνελίξεις οι οποίες προσθέτουν λειτουργίας ανάμεσα σε δύο αντίστοιχα επίπεδα των δύο διαδρομών, είναι οι πλευρικές συνδέσεις. Το FPN ξεπερνά τα άλλα μεμονωμένα CNN, καθώς μπορεί να διατηρήσει ισχυρά σημασιολογικά χαρακτηριστικά σε διάφορες κλίμακες ανάλυσης.

5.3 Ανιχνευτής Πολλαπλών Θυρίδων μιας Λήψης (SSD)

Το SSD: Single Shot MultiBox Detector [99] κυκλοφόρησε στα τέλη του Νοεμβρίου 2016,, από τους Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu και Alexander C. Berg, όπου έφτασε σε νέα ρεκόρ αφορώντας την ακρίβεια και την απόδοση για εργασίες εντοπισμού αντικειμένων, σημειώνοντας ως μέση ακρίβεια πάνω από 74% mAP στα 59 καρέ ανά δευτερόλεπτο σε τυπικά σύνολα δεδομένων, όπως COCO και Pascal VOC. Για την καλύτερη κατανόηση του SSD, αναλύεται το όνομα του:

Single Shot: αυτό σημαίνει ότι οι διαδικασίες ταξινόμησης και εντοπισμού αντικειμένων, μπορούν να εκτελεστούν με μία μόνο κίνηση προς τα εμπρός του δικτύου MultiBox, κάτι που αποτελεί και το όνομα της τεχνικής για την παλινδρόμηση του πλαισίου οριοθέτησης.

Detector: Το δίκτυο αποτελεί έναν ανιχνευτή αντικειμένων κατηγοριοποιώντας τα αντικείμενα που εντοπίστηκαν.

Οι SSD ανιχνευτές, αντίθετα με τον Faster R-CNN στον οποίο χρησιμοποιείται ένα υποδίκτυο για την πρόταση περιοχών, είναι βασισμένοι σε ένα σύνολο από προκαθορισμένες περιοχές. Πάνω από την εικόνα εισόδου και σε κάθε σημείο πολλαπλών μεγεθών και σχημάτων και αγκύρωσης, τοποθετείται ένα πλέγμα σημείων αγκύρωσης, όπου χρησιμεύει ως περιοχές. Το μοντέλο εξάγει μια πρόβλεψη για κάθε πλαίσιο σε κάθε σημείο αγκύρωσης για το αν το αντικείμενο υπάρχει όχι, μέσα σε μια περιοχή και αν μπορεί να τροποποιηθεί η θέση και το μέγεθος του πλαισίου, ώστε να ταιριάζει πιο κοντά στο αντικείμενο. Καθώς σε κάθε σημείο γύρω μπορεί να υπάρχουν πολλά πλαίσια, ενώ τα σημεία αυτά μπορεί να είναι κοντά το ένα με το άλλο, οι ανιχνευτές SSD μπορούν να παράγουν πολλές επικαλυπτόμενες πιθανές ανιχνεύσεις. Τελικά, γίνεται η εκτέλεση μιας μετα-επεξεργασίας στην έξοδό του, έχοντας στόχο την απομάκρυνση των περισσότερων από αυτών των προβλέψεων και την επιλογή της καλύτερης.

Αρχιτεκτονική του δικτύου

Η αρχιτεκτονική του είναι βασισμένη στην αρχιτεκτονική του VGG-16 [100], χωρίς όμως να περιέχει τα πλήρως συνδεδεμένα επίπεδα. Ο λόγος που γίνεται αυτό, είναι εξαιτίας της ισχυρής απόδοσης του VGG-16 στη βελτίωση των αποτελεσμάτων και στην υψηλής ποιότητας εργασίας κατηγοριοποίησης εικόνων σε κλάσεις. Αντί να χρησιμοποιηθούν τα πλήρως συνδεδεμένα επίπεδα VGG, προστέθηκε από το conv6 και μετά, ένα σύνολο βοηθητικών συνελικτικών στρωμάτων, επιτρέποντας έτσι, να εξαχθούν χαρακτηριστικά σε πολλαπλές κλίμακες μειώνοντας σταδιακά το μέγεθος που θα έχει η είσοδος στο κάθε επόμενο στάδιο.

MultiBox

Η τεχνική παλινδρόμησης οριοθέτησης του αλγορίθμου SSD είναι εμπνευσμένη από την εργασία του Szegedy στο MultiBox, η οποία αποτελεί μια μέθοδο για γρήγορες προτάσεις οριοθετημένων πλαισίων [101]. Για να αναπτυχθεί το MultiBox, γίνεται χρήση ενός συνελικτικού δικτύου τύπου Inception [102]. Οι συνελίξεις 1x1 βοηθούν στο να μειωθούν οι διαστάσεις του δικτύου, καθώς μειώνεται ο αριθμός των διαστάσεων, αλλά το ύψος και το πλάτος παραμένουν ίδια. Επίσης, η συνάρτηση απόκλισης του, μπορεί να συνδυάσει δύο κρίσιμα στοιχεία τα οποία μπήκαν στο SSD: την απόκλιση εμπιστοσύνης, όπου υπολογίζει πόσο σίγουρο μπορεί να είναι για το δίκτυο η αντικειμενικότητα του υπολογισμένου πλαισίου οριοθέτησης. Για την υπολογιστική αυτή απώλεια, χρησιμοποιείται η κατηγοριοποιημένη εγκάρσια εντροπία (cross-entropy).

Η απόκρυψη της τοποθεσίας μπορεί να υπολογίσει πόσο μακριά είναι τα προβλεπόμενα πλαίσια που οριοθετούν το δίκτυο από αυτά τα οποία δόθηκαν στο σύνολο δεδομένων της εκπαίδευσης. Εδώ χρησιμοποιείται η κανονικοποίηση L2. Για τη μέτρηση του πόσο μακριά βρέθηκε από την πρόβλεψη, χρησιμοποιείται μια μαθηματική έκφραση για την απόκλιση, η οποία είναι ο όρος Alpha, που βοηθά στο να εξισορροπηθεί η συμβολή της απόκλισης τοποθεσίας. Ο στόχος στη βαθιά μάθηση είναι να βρεθούν οι τιμές των παραμέτρων οι οποίες μπορούν να μειώσουν βέλτιστα τη συνάρτηση απώλειας, φέρνοντας έτσι πιο κοντά στις πραγματικές τις προβλέψεις του μοντέλου.

Χάρτες χαρακτηριστικών

Οι χάρτες χαρακτηριστικών, ή αλλιώς τα αποτελέσματα των συνελικτικών επιπέδων, αποτελούν μια αναπαράσταση σε διαφορετικές κλίμακες των κυρίαρχων χαρακτηριστικών της εικόνας, και έτσι συνεπώς, μία εκτέλεση του MultiBox σε πολλούς χάρτες χαρακτηριστικών, την πιθανότητα οποιουδήποτε αντικειμένου για ανίχνευση, τοποθετώντας κατάλληλα πλαίσια οριοθέτησης γύρω του και

εντοπίζοντας σωστά την κλάση του. Η διαδικασία της πρόβλεψης θεωρείται σχετικά απλή. Τροφοδοτώντας την εικόνα στο δίκτυο, κάθε prior θα έχει ένα σύνολο ετικετών και πλαισίων οριοθέτησης. Υπάρχουν ωστόσο πολλές προβλέψεις για το ίδιο αντικείμενο.

Για να μπορέσουν να αφαιρεθούν τα διπλότυπα priors πάνω από κάθε αντικείμενο το οποίο έχει ήδη ανιχνευθεί, χρησιμοποιείται η μη μέγιστη καταστολή (Non-Maximum Suppression - NMS). Σε αυτήν, το NMS κρατά μόνο τα κουτιά οριοθέτησης τα οποία έχουν τις μεγαλύτερες πιθανότητες καθώς και τα μεγαλύτερα IoU, αφαιρώντας τα κουτιά οριοθέτησης, τα οποία έχουν χαμηλότερες πιθανότητες σχετικά με τα διατηρημένα. Κάποια από τα μειονεκτήματά του SSD είναι τα ρηχά επίπεδα στο νευρωνικό του δίκτυο, τα οποία ενδέχεται να μην δημιουργήσουν αρκετά χαρακτηριστικά υψηλού επιπέδου για να γίνουν προβλέψεις για μικρά αντικείμενα. Συνεπώς, δεν γίνονται καλές προβλέψεις για μικρότερα αντικείμενα σχετικά με τα μεγαλύτερα.

5.4 EfficientDet

Πρόκειται για μια σειρά από 8 αλγόριθμος οι οποίοι δημιουργήθηκαν από την Google, και είναι πολύ πιο γρήγοροι, ακριβείς και αποδοτικοί σχετικά με τους προϋπάρχοντες ανιχνευτές [103]. Μπορεί να αποδίδονται καλά αποτελέσματα τα οποία είναι πάντα συγκρίσιμα σε ένα ευρύ φάσμα περιορισμένων πόρων σε διαφορετικές συσκευές. Ειδικά στην περίπτωση όπου μπορεί να υπάρχει ένα μοντέλο μιας κλίμακας, το EfficientDet-D7 έφτασε στα 52,2% mAP στο σύνολο δοκιμής COCO, με 52M παραμέτρους και 325B FLOPs, συγκριτικά με τον προηγούμενο αλγόριθμο, υπάρχει μείωση της ποσότητας των παραμέτρων κατά 4 έως 9 φορές, ενώ τα FLOPs μειώθηκαν κατά 13 έως 42 φορές. Για να αναπτυχθεί αυτός ο ανιχνευτής αντικειμένων προτάθηκε από την Google, ένα σταθμισμένο δίκτυο διπλής κατεύθυνσης πυραμίδας χαρακτηριστικών (BiFPN), όπου προτείνεται η γρήγορη και απλή συγχώνευση πολλαπλών κλιμάκων, μια μέθοδος κλιμάκωσης των χαρακτηριστικών πυραμίδας δικτύου, όπου εμφανίζει μία ομοιομορφία κλίμακας της ανάλυσης στο πλάτος και του βάθους του δικτύου προβλέψεων box/class καθώς και το δίκτυο δυνατοτήτων όλων των δικτύων backbone.

5.5 YOLOv3

Όλοι οι προηγούμενοι αλγόριθμοι ανίχνευσης αντικειμένων κάνουν χρήση των περιοχών για τον εντοπισμό αντικειμένων μέσα στην εικόνα. Η πλήρης εικόνα δεν είναι ορατή από το δίκτυο παρά μόνο μερικά τμήματά της, τα οποία έχουν μεγάλες

πιθανότητες να περιέχουν αντικείμενο. Το YOLO ή You Only Look Once [104] αποτελεί έναν αλγόριθμο ανίχνευσης αντικειμένων, ο οποίος παρουσιάζει μεγάλες διαφορές από τους αλγορίθμους, οι οποίοι βασίζονται στις περιοχές. Στον αλγόριθμο αυτό, τα πλαίσια οριοθέτησης και οι πιθανότητες αντικειμένου προβλέπονται από ένα ενιαίο συνελικτικό δίκτυο. Καθώς η διαδικασία ανίχνευσης αποτελεί ολόκληρη ένα μεμονωμένο δίκτυο, μπορεί να γίνει πλήρης βελτιστοποίησης της απόδοσης της ανίχνευσης. Ο αλγόριθμος αυτός, καθώς διαθέτει 75 συνελικτικά επίπεδα, μπορεί να θεωρηθεί ως ένα πλήρες συνελικτικό δίκτυο (FCN).

Χρησιμοποιείται ένα συνελικτικό στρώμα με βήμα (stride), καθώς δεν υπάρχει καμία μορφή ομαδοποίησης και έτσι μπορεί να υποδειχθούν οι χάρτες των χαρακτηριστικών. Κάτι τέτοιο μπορεί να βοηθήσει στο να προληφθεί η απώλεια των χαρακτηριστικών χαμηλού επιπέδου, οι οποίες αποδίδονται συχνά στην ομαδοποίηση. Καθώς πρόκειται για FCN, ο αλγόριθμος YOLO θεωρείται ότι δεν μεταβάλλεται από το μέγεθος της εικόνας εισόδου. Στην πράξη ωστόσο, πρέπει να παραμείνει σε ένα σταθερό μέγεθος εισόδου των εικόνων, καθώς διαπιστώνονται διάφορα προβλήματα κατά τη διάρκεια εφαρμογής του.

Ένα από τα προβλήματα αυτά, είναι ότι σε περίπτωση που πρέπει να γίνει επεξεργασία των εικόνων σε παρτίδες έτσι ώστε να αξιοποιηθούν και οι δυνατότητες επεξεργασίας μιας GPU, όλες οι εικόνες πρέπει να είναι σε σταθερό πλάτος και ύψος. Κάτι τέτοιο μπορεί να απαιτήσει να συνενωθούν πολλαπλές εικόνες σε μια μεγάλη παρτίδα. Η εικόνα υποδειγματοποιείται από το δίκτυο με ένα παράγοντα, ο οποίος ονομάζεται βήμα του δικτύου. Εάν για παράδειγμα το βήμα είναι 32, τότε για μια εικόνα εισόδου μεγέθους 416 x 416 αποδίδεται έξοδος μεγέθους 13 x 13.

Ερμηνεία της εξόδου

Όπως συμβαίνει σε όλους τους ανιχνευτές αντικειμένων, τα χαρακτηριστικά τα οποία μαθαίνουν τα συνελικτικά επίπεδα, μπορούν να μεταφερθούν σε έναν παλινδρομητή (regressor), οποίος εκτελεί την πρόβλεψη ανίχνευσης. Στο YOLO, η πρόβλεψη αυτή εκτελείται με τη χρήση ενός συνελικτικού επίπεδο το οποίο χρησιμοποιεί 1 x 1 συνελίξεις. Η έξοδος του είναι ένας χάρτης χαρακτηριστικών. Έχοντας ως δεδομένο ότι χρησιμοποιήθηκαν 1 x 1 συνελίξεις, το μέγεθος του χάρτη προβλέψεων θα έχει ακριβώς το ίδιο μέγεθος με το χάρτη χαρακτηριστικών πριν από αυτόν. Ο τρόπος με τον οποίο γίνεται η ερμηνεία του χάρτη πρόβλεψης, είναι ότι κάθε κελί μπορεί να προβλέψει ένα καθορισμένο αριθμό από πλαίσια οριοθέτησης. Αν και ο τεχνικός σωστός όρος για να περιγραφεί μία μονάδα στο χαρτί χαρακτηριστικών είναι ο

νευρώνας, όταν αυτό αποκαλείται ως κελί, το καθιστά πιο διαισθητικό στην περίπτωση του αλγορίθμου.

Το YOLO έχει τρεις άγκυρες, οι οποίες φέρουν ως αποτέλεσμα να προβλεφθούν 3 πλαίσια οριοθέτησης ανά κελί. Αυτό το οποίο θεωρείται υπεύθυνο για να ανιχνευθεί ένα αντικείμενο σε μια εικόνα, είναι εκείνο στο οποίο η άγκυρα έχει το υψηλότερο IoU με το χαμηλότερο πλαίσιο.

Συντεταγμένες κέντρων

Για να μπορέσουν να προβλεφθούν οι συντεταγμένες ενός κέντρου του αντικείμενου, χρησιμοποιείται η σιγμοειδή συνάρτηση. Κάτι τέτοιο αναγκάζει την τιμή της εξόδου να κυμαίνεται ανάμεσα στο μηδέν και στο ένα. Κανονικά, δεν μπορεί να γίνει πρόβλεψη των απόλυτων συντεταγμένων του κέντρου του πλαισίου οριοθέτησης. Οι αντισταθμίσεις που προβλέπονται είναι:

•συνθήκες με την πάνω αριστερή γωνία του κελιού πλέγματος το οποίο μπορεί να προβλέψει το αντικείμενο.

•η ομαλοποίηση από τις διαστάσεις του κελιού από το χάρτη χαρακτηριστικών δηλαδή το ένα.

Διαστάσεις του πλαισίου οριοθέτησης

Οι διαστάσεις του πλαισίου οριοθέτησης προβλέπονται με την εφαρμογή ενός μετασχηματισμού του χώρου καταγραφής στην έξοδο, ο οποίος θα πολλαπλασιάζεται μετά με μια άγκυρα. Οι προβλέψεις που προκύπτουν, bw και bh, ομαλοποιούνται από το πλάτος και το ύψος της εικόνας, όπως δηλαδή επιλέγονται και οι ετικέτες εκπαίδευσης.

Βαθμολογία πιθανότητας αντικειμένου

Η πιθανότητα ότι ένα αντικείμενο θα περιέχεται μέσα σε ένα πλαίσιο οριοθέτησης, αντιπροσωπεύεται από τη βαθμολογία πιθανότητας του αντικειμένου. Για το πλέγμα και τα γειτονικά πλέγματα θα πρέπει να είναι σχεδόν ένα, ενώ για το πλέγμα στις γωνίες θα πρέπει να είναι σχεδόν μηδέν. Καθώς θα πρέπει να ερμηνευτεί ως πιθανότητα, η βαθμολογία πιθανότητας αντικειμένου θα περάσει επίσης μέσα από ένα σιγμοειδές.

Εμπιστοσύνη κλάσης

Οι πιθανότητας του αντικειμένου το οποίο ανιχνεύεται ώστε να ανήκει σε μια συγκεκριμένη τάξη, αντιπροσωπεύεται από τις εμπιστευτικές τάξεις. Το YOLO, πριν από το v3, δεν ήθελε να χρησιμοποιεί τη συνάρτηση Softmax για να βαθμολογήσει τις κλάσεις. Ωστόσο, σε αυτή την έκδοση έχει αφαιρεθεί η συγκεκριμένη επιλογή

σχεδιασμού καθώς οι συγγραφείς επέλεξαν τη χρήση της σιγμοειδούς συνάρτησης. Ο λόγος είναι ότι χρησιμοποιώντας τη Softmax, θεωρείται ότι οι κλάσεις αποκλείονται αμοιβαία μεταξύ τους. Το αποτέλεσμα αυτού είναι ότι αν το αντικείμενο ανήκει σε μια κλάση, τότε εγγυημένα δεν μπορεί να ανήκει σε άλλη. Κάτι τέτοιο ισχύει και για το σύνολο δεδομένων COCO [105], στο οποίο είναι βασισμένο το προεκπαιδευμένο μοντέλο YOLOv3. Οι υποθέσεις αυτές μπορεί ωστόσο να μην ισχύουν όταν υπάρχουν κλάσεις όπως ένα το πρόσωπο και ο άνθρωπος. Για αυτό το λόγο αποφεύχθηκε και η χρήση μιας συνάρτησης Softmax.

Πρόβλεψη σε διαφορετικές κλίμακες

Το YOLO v3 μπορεί να προβλέπει σε 3 διαφορετικές κλίμακες. Για να ανιχνευθούν οι χάρτες των χαρακτηριστικών 3 διαφορετικών μεγεθών, με βήματα 32, 16 και 8 αντίστοιχα, χρησιμοποιείται το επίπεδο ανίχνευσης. Γίνεται υποδειγματοποίηση της εικόνας εισόδου από το δίκτυο μέχρι το πρώτο επίπεδο ανίχνευσης, όπου μπορεί να γίνει με τη χρήση των χαρτών χαρακτηριστικών ενός επιπέδου με βήμα 32. Τα επίπεδα προστίθενται σε ένα δείγμα έχοντας συντελεστή 2, ενώ οι χάρτες χαρακτηριστικών των προηγούμενων επιπέδων, οι οποίοι έχουν ίδιο μέγεθος με το χάρτη των χαρακτηριστικών συνδυάζονται. Μια άλλη ανίχνευση η οποία μπορεί να γίνει και στο επίπεδο γίνεται με το βήμα 16.

Γίνεται επανάληψη της ίδιας διαδικασίας δειγματοληψίας με την τελική ανίχνευση να γίνεται στο επίπεδο του βήματος 8. Κάθε κελί προβλέπει τρία πλαίσια οριοθέτησης σε κάθε κλίμακα, με τη χρήση τριών αγκυρών, μετατρέποντας έτσι το συνολικό αριθμό αγκυρών χρήση σε 9. Κάθε κλίμακα έχει διαφορετική άγκυρα. Κάτι τέτοιο μπορεί να βοηθήσει την εκδοχή αυτή στο να βελτιώσει τον εντοπισμό μικρών αντικειμένων, κάτι το οποίο αποτελούσε ένα συχνό παράπονο με προηγούμενες εκδόσεις του YOLO. Το δίκτυο μπορεί να βοηθηθεί από τη δειγματοληψία ώστε να μάθει τις λεπτομέρειες από τις λειτουργίες, οι οποίες καθορίζουν τον εντοπισμό μικρών αντικειμένων.

Μη μέγιστη καταστολή (Non-maximum Suppression)

Η μη μέγιστη καταστολή μπορεί να ελέγξει το πρόβλημα όταν υπάρχουν πολλαπλές ανιχνεύσεις ενός αντικειμένου στην ίδια εικόνα. Αν για παράδειγμα και τα 3 πλαίσια οριοθέτησης του κελιού του πλέγματος έχουν τη δυνατότητα ανίχνευσης των παρακείμενων κελιών ή ενός πλαισίου, μπορεί να γίνει ανίχνευση του ίδιου αντικειμένου.

Η αρχιτεκτονική του YOLO

ΤΟ YOLO είναι ένα δίκτυο το οποίο είναι βασισμένο στη μάθηση χαρακτηριστικών το οποίο υιοθετεί ως το πιο ισχυρό εργαλείο του 75 συνελικτικά επίπεδα. Δεν μπορεί να γίνει χρήση του σε κανένα πλήρως συνδεδεμένο επίπεδο. Η συγκεκριμένη δομή μπορεί να καταστήσει δυνατή την αντιμετώπιση εικόνων με οποιοδήποτε μέγεθος. Το μεγαλύτερο μειονέκτημα και ο περιορισμός του ανιχνευτή αντικειμένων YOLO, είναι ότι δεν μπορεί να εντοπίσει πάντα τα μικρά αντικείμενα και ιδίως δεν μπορεί να χειριστεί καλά τα αντικείμενα τα οποία είναι ομαδοποιημένα μεταξύ τους.

Βελτιώσεις του YOLO

Μια επόμενη έκδοση του αλγορίθμου YOLO είναι το YOLOv4 [106] από τους συγγραφείς Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao.

YOLOv4

Το YOLOv4 αποτελεί μια σημαντική βελτίωση του YOLOv3. Εφαρμόστηκε μια νέα αρχιτεκτονική στο Backbone και τροποποιήθηκε το Neck, οδηγώντας σε εξαιρετικά αποτελέσματα με ταχύτητα ανίχνευσης σε πραγματικό χρόνο. Η έκδοση του αλγορίθμου αυτού, μπορεί να επιτύχει μέση ακρίβεια ίση με 43,5% AP και ταχύτητα στα 65 FPS σε μια GPU Tesla V100 [106] στο σύνολο δεδομένων MS COCO. Για να γίνει αυτό συνεργάζονται δυνατότητες όπως είναι η απόκλιση CIoU, η κανονικοποίηση DropBlock, οι επαυξήσεις δεδομένων, οι Mish ενεργοποιήσεις, το Self-Adversial-training (SAT), οι Cross-Stage-Partial-Connections (CSP) και οι σταθμισμένες συνδέσεις. Μπορεί να διακριθεί ότι το EfficientDet D4-D3, μπορεί να επιτύχει καλύτερο AP από τα μοντέλα YOLO v4, λειτουργώντας όμως με ταχύτητα < 30 FPS σε GPU V100. Από την άλλη πλευρά, το YOLO μπορεί να τρέχει με πολύ μεγαλύτερη ταχύτητα (60+ FPS) με πολύ καλή ακρίβεια.

Η αρχιτεκτονική του backbone του YOLOv4 αποτελείται από τρία βασικά μέρη:

- Bag of freebies
- Bag of specials
- Και το CSPDarknet53

Αρχιτεκτονική του YOLOv4

Backbone: ως μοντέλο εξαγωγής χαρακτηριστικών για την έκδοση GPU χρησιμοποιείται το CSPDarknet53, όπου αποτελεί ένα νέο backbone, το οποίο έχει τη δυνατότητα βελτίωσης της μαθησιακής ικανότητας του CNN. Γίνεται η προσθήκη του μπλοκ χωρικής πυραμίδας μέσα από το CSPDarknet53, ώστε να αυξηθεί το δεκτικό πεδίο διαχωρίζοντας τα πιο σημαντικά χαρακτηριστικά του περιβάλλοντος.

Αντί να γίνει χρήση του FPN όπως στο YOLOv3, γίνεται χρήση του PANet ως μια μέθοδος συγκέντρωσης παραμέτρων για τα διαφορετικά επίπεδα του ανιχνευτή.

Neck: Το YOLOv4 χρησιμοποιεί Path Aggregation Network (PAN) και Spatial pyramid pooling (SPP). Το PAN αποτελεί μια τροποποιημένη έκδοση του αρχικού που αντικαθιστά την προσθήκη μιας συνένωσης (concatenation).

Head: Το YOLOv4 χρησιμοποιεί την ίδια κεφαλή YOLO με το YOLOv3 για να μπορέσει να κάνει ανίχνευση με τα βήματα ανίχνευσης, βασισμένο στην αγκύρωση και τα 3 επίπεδα ανίχνευσης.

Πρόσθετες βελτιώσεις

Στο YOLOv4 γίνεται εισαγωγή μιας νέας μεθόδου επαύξησης δεδομένων, η οποία ονομάζεται μωσαϊκό. Η μέθοδος αυτή μπορεί να συνδυάσει 4 εικόνες συνόλου δεδομένων εκπαίδευσης σε μια εικόνα. Έτσι, η ομαλοποίηση της παρτίδας μπορεί να υπολογίζει στατιστικά στοιχεία ενεργοποίησης από 4 διαφορετικές εικόνες σε κάθε επίπεδο [106]. Με αυτό τον τρόπο, μπορεί να μειωθεί σημαντικά η ανάγκη να επιλεχθεί μεγάλο μέγεθος mini παρτίδας για εκπαίδευση. Επίσης, χρησιμοποιείται η αυτοαντιπαραθετική εκπαίδευση (Self-Adversarial Training - SAT), η οποία λειτουργεί σε δύο στάδια, προς τα εμπρός και προς τα πίσω. Το νευρωνικό δίκτυο στο πρώτο στάδιο, κάνει τροποποίηση της αρχικής εικόνας αντί για τα βάρη του δικτύου. Με τον τρόπο αυτό, μπορεί να εκτελέσει μια εχθρική επίθεση στον εαυτό του αλλάζοντας την αρχική εικόνα, ώστε να δημιουργηθεί εξαπάτηση ότι δεν υπάρχει επιθυμητό αντικείμενο στην εικόνα. Το νευρωνικό δίκτυο στο δεύτερο στάδιο, εκπαιδεύεται ώστε να μπορεί να ανιχνεύσει ένα αντικείμενο σε μια τροποποιημένη εικόνα με τον κανονικό τρόπο.

Tiny YOLO

Παράλληλα με το YOLO, εμφανίζεται και ο αλγόριθμος Tiny-YOLO, ο οποίος αποτελεί στην ουσία μια μικρότερη έκδοση του συνελικτικού νευρωνικού δικτύου του YOLO ενώ είναι εξαιρετικά γρήγορος. Η αρχιτεκτονική του Tiny-YOLO είναι περίπου 442% ταχύτερη από το YOLO, ενώ μπορεί να επιτύχει πάνω από 244 FPS σε μία μόνο GPU. Κατέχει, ωστόσο, σχεδόν τη μισή ακρίβεια από το YOLO. Κάτι τέτοιο συμβαίνει επειδή χρησιμοποιεί ένα ελαφρύτερο μοντέλο, το οποίο περιέχει λιγότερα επίπεδα συγκριτικά με το κανονικό YOLO, με την εικόνα εισόδου να διατηρεί το ίδιο μέγεθος με αυτή του YOLO. Η γρήγορη ταχύτητα συμπερασμάτων και το μικρό μέγεθος του μοντέλου (<50MB), καθιστούν τον ανιχνευτή αντικειμένων Tiny-YOLO κατάλληλο για συσκευές υπολογιστικής όρασης και βαθιάς μάθησης

υπολογιστή οι οποίες είναι ενσωματωμένες, όπως είναι το NVIDIA Jetson Nano, το Google Coral και το Raspberry Pi.[107]

6.Πειραματική σύγκριση αλγορίθμων και συμπεράσματα

Το πρόβλημα

Τα τελευταία χρόνια το πρόβλημα εντοπισμού και αναγνώρισης αντικειμένων απασχολεί χιλιάδες επιστήμονες στον χώρο της τεχνητής νοημοσύνης και ιδιαίτερα στον τομέα της υπολογιστικής όρασης.Η εξέλιξη των συνελικτικών νευρωνικών δικτύων και η ανάπτυξη νέων αλγορίθμων οδήγησε στην εύρεση αρκετά ικανοποιητικών λύσεων πάνω στο πρόβλημα.Κάποιες από αυτές θα εξετάσουμε παρακάτω.

Python

Η python είναι πλέον μια πολύ δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου η οποία συνήθως χρησιμοποιείται για την ανάπτυξη μοντέλων νευρωνικών δικτύων. Δημιουργήθηκε από τον Ολλανδό Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Η Python διαθέτει παρά πολλά πακέτα που ο κάθε προγραμματιστής μπορεί να τα κατεβάσει από τα αποθετήρια πακέτων και να τα εισάγει στον κώδικά του. [108]

TensorFlow

Το TensorFlow είναι μαθηματική βιβλιοθήκη την οποία ανέπτυξε η Google Brain, ομάδα τεχνητής νοημοσύνης της Google, αρχικά για εσωτερική χρήση. Σκοπός της δημιουργίας της ήταν η διευκόλυνση διαδικασιών, όπως ο προγραμματισμός ροής δεδομένων, ενώ βρήκε χρήση και στην μηχανική μάθηση έχοντας ως παράδειγμα τα νευρωνικά δίκτυα. Η δημόσια διανομή του TensorFlow έγινε τον Νοέμβριο του 2015 κάτω από την άδεια ανοιχτού λογισμικού της Apache (Apache 2.0 Open Source License). Αρχικά, η ομάδα της Google δημιούργησε το λογισμικό DistBelief ως ιδιόκτητο σύστημα μηχανικής μάθησης που βρήκε απήχηση σε διάφορες εταιρίες τόσο για έρευνα όσο και για εμπορική χρήση. Στην πορεία ανέθεσε την απλοποίηση και την αναδιάταξη του κώδικα του DistBelief σε διάφορους αναγνωρισμένους επιστήμονες της πληροφορικής, συμπεριλαμβανόμενου του Jeff Dean (επικεφαλής του Google.ai), στοχεύοντας σε μια γρηγορότερη και ισχυρότερη βιβλιοθήκη, με αποτέλεσμα τη γέννηση του TensorFlow.

Το TensorFlow εμφανίζεται σε 2 βασικές εκδόσεις, την έκδοση TensorFlow 1.x και TensorFlow 2.x. Η ευέλικτη αρχιτεκτονική του TensorFlow καθιστά δυνατή την εύκολη ανάπτυξη υπολογισμών σε μια ποικιλία πλατφορμών, από υπολογιστές έως και κινητά. Μπορεί να εκμεταλλευτεί την υπολογιστική ισχύ πολλαπλών επεξεργαστών και καρτών γραφικών . Είναι διαθέσιμο σε λογισμικά 64-bit όπως το Linux και τα Windows. Οι υπολογισμοί του TensorFlow εκφράζονται ως στατικά διαγράμματα ροής δεδομένων και το όνομά του προκύπτει από τις διαδικασίες που εκτελούν τα νευρωνικά δίκτυα στους πολυδιάστατους πίνακες δεδομένων. Αυτοί οι πίνακες αναφέρονται ως "tensors".[109]

Tensorboard

Το TensorBoard είναι ένα εργαλείο για την παροχή των μετρήσεων και των οπτικοποιήσεων που απαιτούνται κατά τη ροή εργασίας της μηχανικής μάθησης. Επιτρέπει την παρακολούθηση μετρήσεων πειράματος, όπως απώλεια και ακρίβεια, οπτικοποίηση του γραφήματος μοντέλου καθώς αλλάζει κατά την διάρκεια του χρόνου, προβολή ενσωματώσεων σε χώρο χαμηλότερης διάστασης, εμφάνιση εικόνων, κειμένου και δεδομένων ήχου και πολλά άλλα .[110]

PyTorch

Το PyTorch είναι ένα πλαίσιο μηχανικής μάθησης ανοιχτού κώδικα (ML) που βασίζεται στη γλώσσα προγραμματισμού Python και στη βιβλιοθήκη Torch. Το Torch είναι μια βιβλιοθήκη Machine Learning ανοιχτού κώδικα που χρησιμοποιείται για τη δημιουργία νευρωνικών δικτύων και είναι γραμμένη στη γλώσσα δέσμης ενεργειών Lua. Είναι μια από τις προτιμώμενες πλατφόρμες για έρευνα βαθιάς μάθησης. Το PyTorch είναι εύκολο να χρησιμοποιηθεί λόγω της απλότητας του. [111]

Google Colab

Τα notebook μας επιτρέπουν να συνδυάζουμε εκτελέσιμο κώδικα και εμπλουτισμένο κείμενο σε ένα έγγραφο με εικόνες, HTML και άλλα. Τα σημειωματάρια αυτά μπορούν να αποθηκευτούν στο λογαριασμό μας στο google drive. Με το colab μπορούμε να αξιοποιήσουμε πλήρως την ισχύ των δημοφιλών βιβλιοθηκών Python για την ανάλυση και οπτικοποίηση δεδομένων. Σημαντικό είναι πως στα σημειωματάρια colab μπορούμε να εισάγουμε δεδομένα από το github καθώς και από άλλες πηγές. Στο colab μπορούμε να εισάγουμε και σύνολα δεδομένων εικόνων, έπειτα να εκπαιδεύσουμε έναν ταξινομητή εικόνων και να αξιολογήσουμε το μοντέλο αυτό με λίγες γραμμές κώδικα. Το σημαντικότερο όμως είναι πως μπορούμε να αξιοποιήσουμε την δύναμη του εξοπλισμού της Google, δηλαδή των GPU και TPU, ανεξάρτητα από την ισχύ του δικού μας μηχανήματος.[112]

Δημιουργία συνόλου δεδομένων εικόνων για εκπαίδευση

Η επιτυχία της μηχανικής μάθησης εξαρτάται και από τα δεδομένα πάνω στα οποία θα εκπαιδευτεί. Τα κατάλληλα δεδομένα θα επιτρέψουν στο νευρωνικό δίκτυο να μάθει να κατηγοριοποιεί και να επιλύει σωστά το πρόβλημα. Επομένως, η επιλογή των δεδομένων προς εκπαίδευση του μοντέλου είναι πολύ σημαντική. Για την αποτελεσματική εκπαίδευση ενός μοντέλου είναι απαραίτητη η δημιουργία ενός συνόλου δεδομένων για εκπαίδευση (train), ενός συνόλου δεδομένων για τεστ (test) και ενός συνόλου δεδομένων για επαλήθευση (validation).

Σύνολα δεδομένων για εκπαίδευση

Υπάρχουν διαθέσιμα μεγάλα σύνολα δεδομένων όπου μπορούμε να εντοπίσουμε εικόνες αντικειμένων για τα οποία θέλουμε να εκπαιδεύσουμε το μοντέλο. Τα πιο δημοφιλή σύνολα δεδομένων είναι τα COCO Dataset, PASCAL VOC και το Google Open Images Dataset. [113]

Στο πείραμά μας χρησιμοποιήθηκαν εικόνες για την εκπαίδευση του μοντέλου που επιλέχθηκαν από το σύνολο δεδομένων Open Images Dataset v4 της Google (OIDv4)

Μέθοδος Μέτρησης

Η πιο γνωστή μέθοδος αξιολόγησης των συνελικτικών νευρωνικών δικτύων για προβλήματα εντοπισμού και ανίχνευσης αντικειμένων είναι η Average Precision και συγκεκριμένα το ποσοστό Mean Average Precision. Η μέθοδος βασίζεται σε δύο μετρικές την precision και την recall. Η precision μετρά πόσο ακριβείς είναι οι προβλέψεις του δικτύου, δηλαδή το ποσοστό των προβλέψεων κατηγοριοποίησης που είναι σωστές και δίνεται από τον τύπο:

$$precision = TP / (TP + FP)$$

Η recall μετρά την επιτυχία της εύρεσης όλων των θετικών περιπτώσεων δηλαδή πόσα από τα αντικείμενα που υπήρχαν βρέθηκαν σωστά και δίνεται από τον τύπο:

$$recall = TP / (TP + FN)$$

TP = όσα ανήκουν στην κλάση i και ταξινομήθηκαν στην i

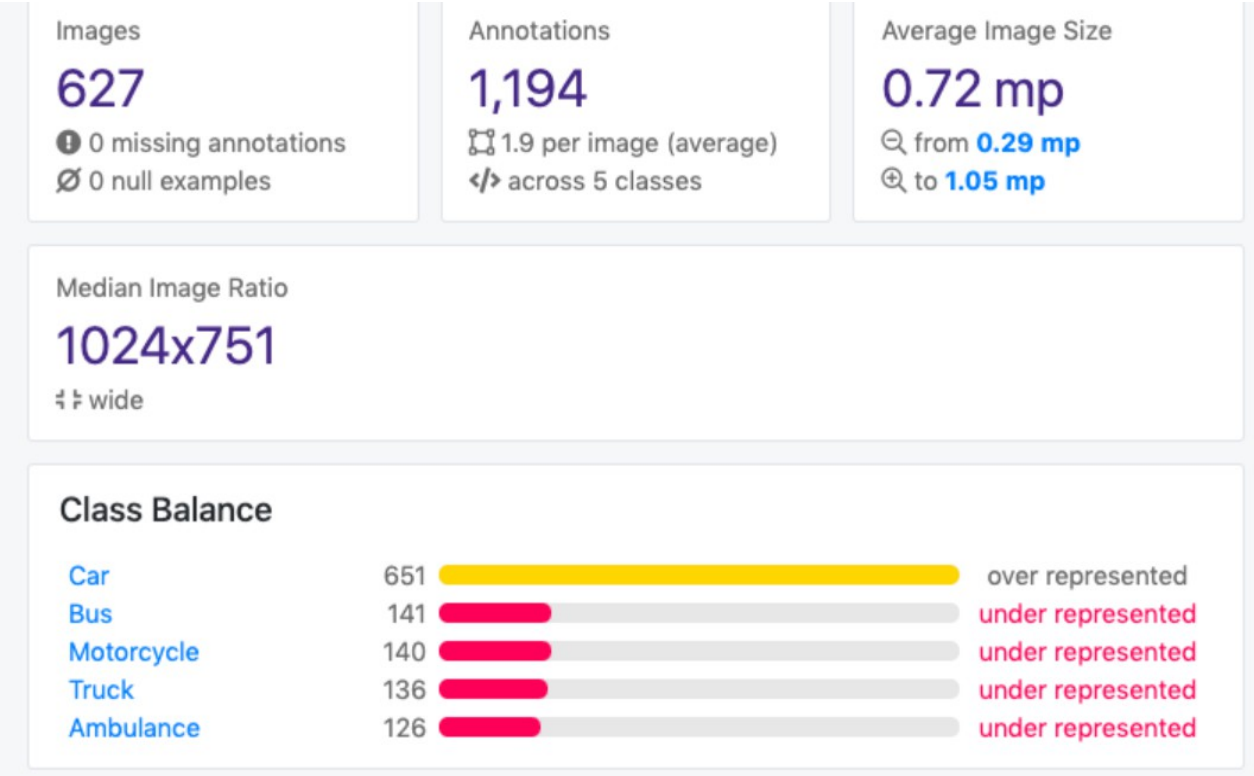
FP = όσα δεν ανήκουν στην κλάση i και ταξινομήθηκαν στην i

FN = όσα ανήκουν στην κλάση i αλλά δεν ταξινομήθηκαν στην i

Αφού υπολογιστούν οι τιμές precision και recall τοποθετούνται σε έναν δισδιάστατο άξονα όπου σχηματίζεται ένα γράφημα. Το εμβαδό του γραφήματος αποτελεί την τιμή του AP.

Το τελικό ποσοστό επιτυχίας του μοντέλου καθορίζεται από τον δείκτη mAP (Mean average Precision) που υπολογίζεται από τον μέσο όρο όλων των τιμών AP. Για την εξαγωγή έγκυρων συμπερασμάτων γίνεται υπολογισμός του ποσοστού mAP εφαρμόζοντας διάφορες τιμές κατωφλίου. Έτσι σχηματίζεται πιο ολοκληρωμένη άποψη σχετικά με την ακρίβεια του εντοπισμού της θέσης των αντικειμένων από το μοντέλο. Ο υπολογισμός πραγματοποιείται συνολικά για το μοντέλο και για τις επιμέρους κλάσεις των αντικειμένων ώστε να γίνει αντιληπτό πόσο καλά μπορεί να εντοπίσει την κάθε κλάση το νευρωνικό δίκτυο.

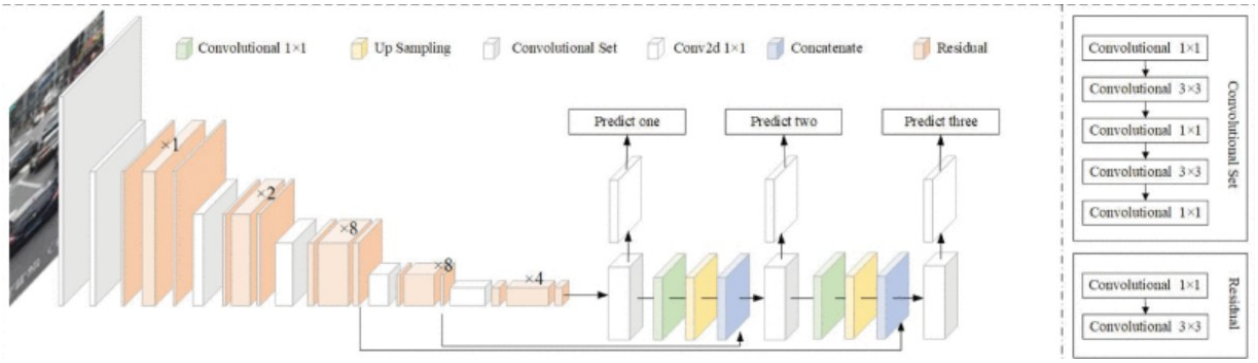
Dataset από roboflow



Εκπαίδευση μοντέλου YOLOv3

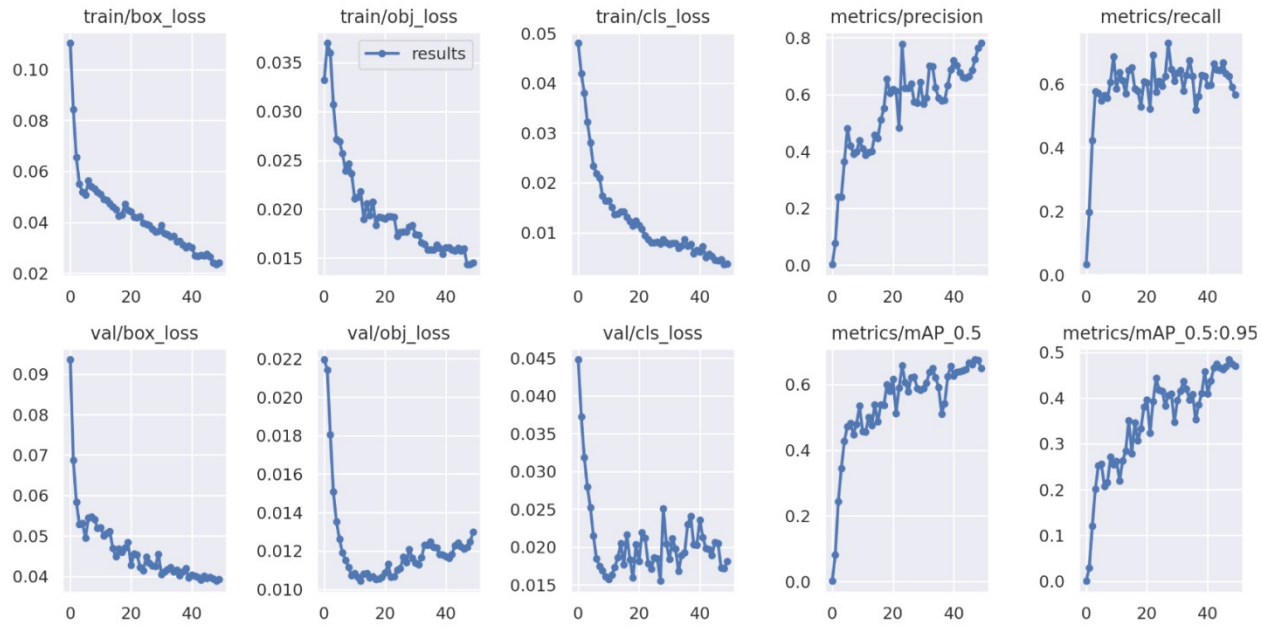
Το YOLOv3 είναι ένας ανιχνευτής αντικειμένων που εκτελεί τη διαδικασία ανίχνευσης ως εργασία παλινδρόμησης. Αυτή η μέθοδος αυξάνει την ταχύτητα ανίχνευσης και δέχεται εικόνες εισόδου διαφορετικών μεγεθών. Το YOLOv3 χρησιμοποιεί το Darknet-53 για την εκτέλεση εξαγωγής χαρακτηριστικών. Το YOLOv3 χρησιμοποιεί πρόβλεψη πολλαπλής κλίμακας και, για τον λόγο αυτόν, η ακρίβεια της ανίχνευσης στόχου βελτιώνεται. Το YOLO v3 μπορεί να επιτύχει

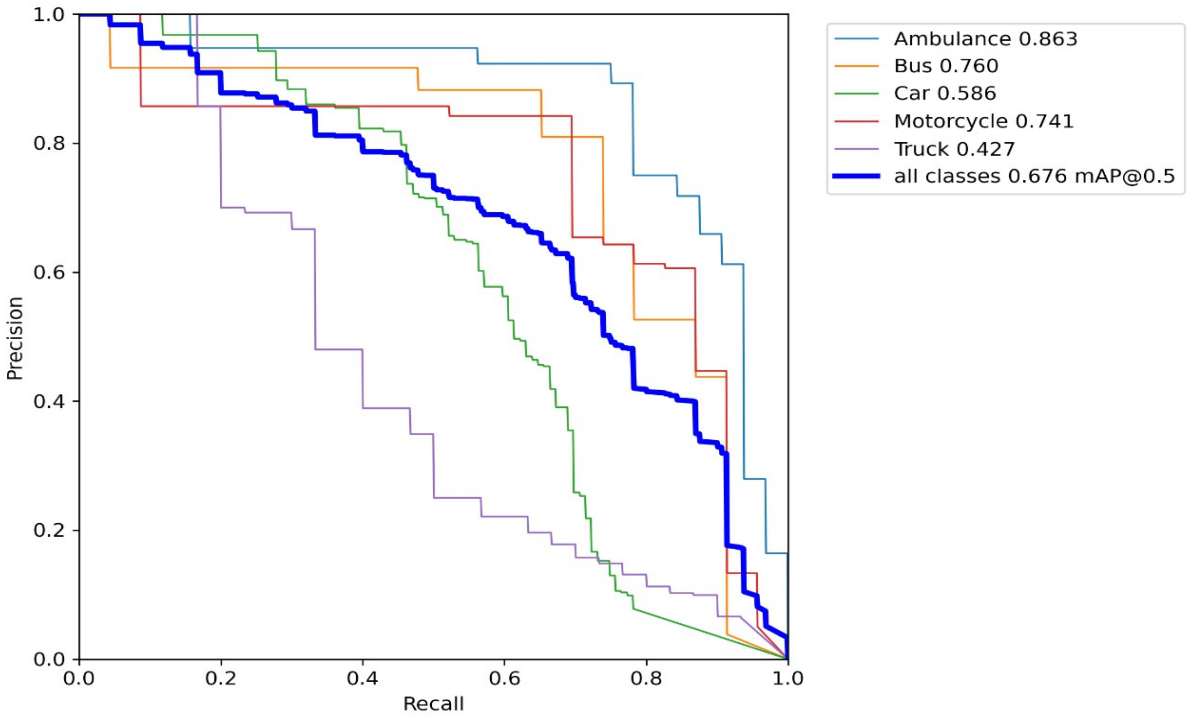
ανίχνευση σε πραγματικό χρόνο μέσω της ισχυρής υπολογιστικής ικανότητας της GPU.[114]



Η εκπαίδευση έγινε σε 125 φωτογραφίες και για 50 εποχές

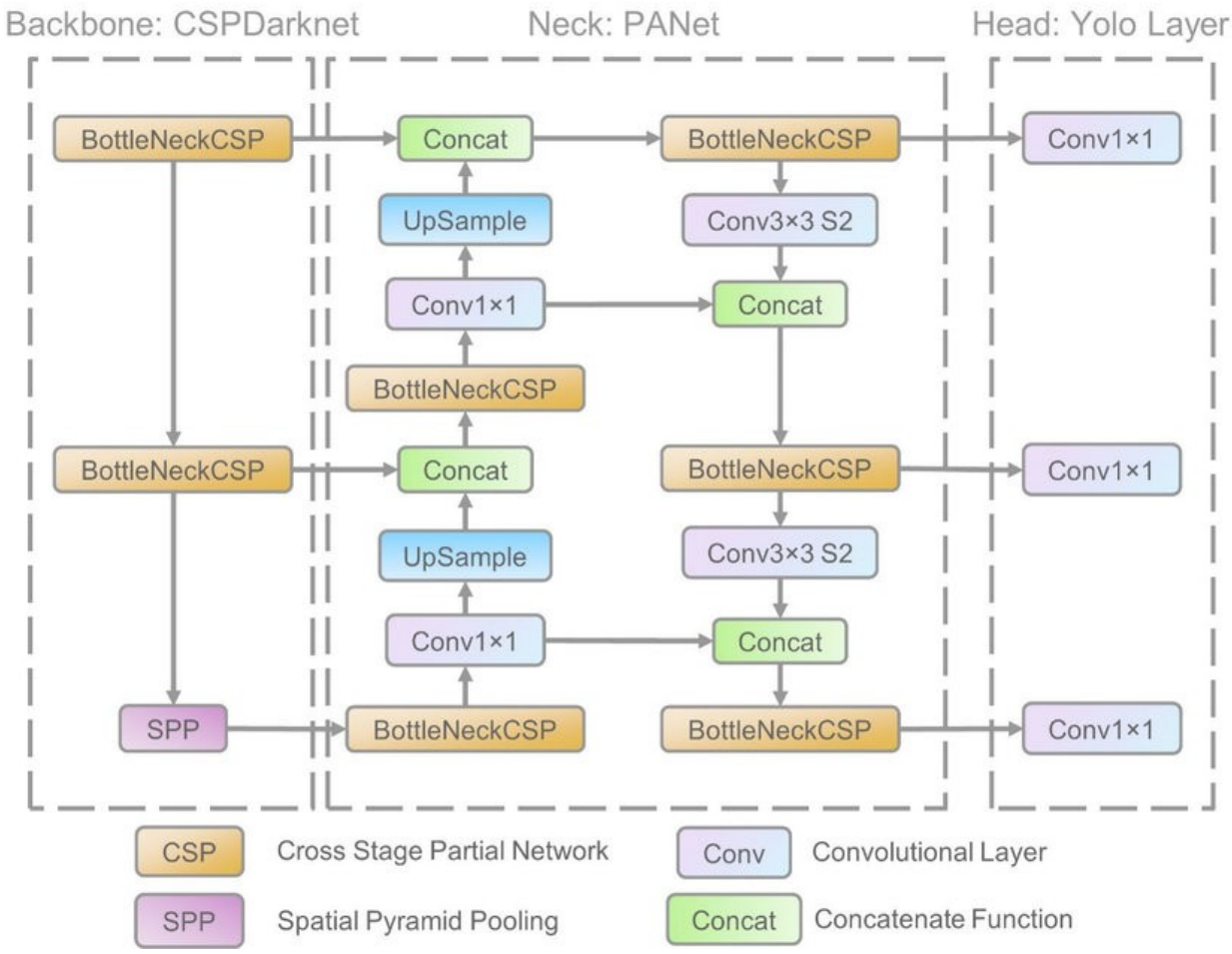
```
50 epochs completed in 0.389 hours.
Optimizer stripped from runs/train/results_1/weights/last.pt, 123.6MB
Optimizer stripped from runs/train/results_1/weights/best.pt, 123.6MB

Validating runs/train/results_1/weights/best.pt...
Fusing layers...
Model Summary: 261 layers, 61518970 parameters, 0 gradients, 154.6 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 4/4 [00:04<00:00,  1.08s/it]
                 all        125        227      0.725      0.624      0.676      0.483
           Ambulance        125         32       0.75      0.844      0.863       0.69
                 Bus        125         23      0.796      0.739       0.76      0.572
                 Car        125        119      0.699      0.507      0.586      0.383
          Motorcycle        125         23      0.815      0.696      0.741      0.457
               Truck        125         30      0.564      0.333      0.427      0.316
Results saved to runs/train/results_1
```



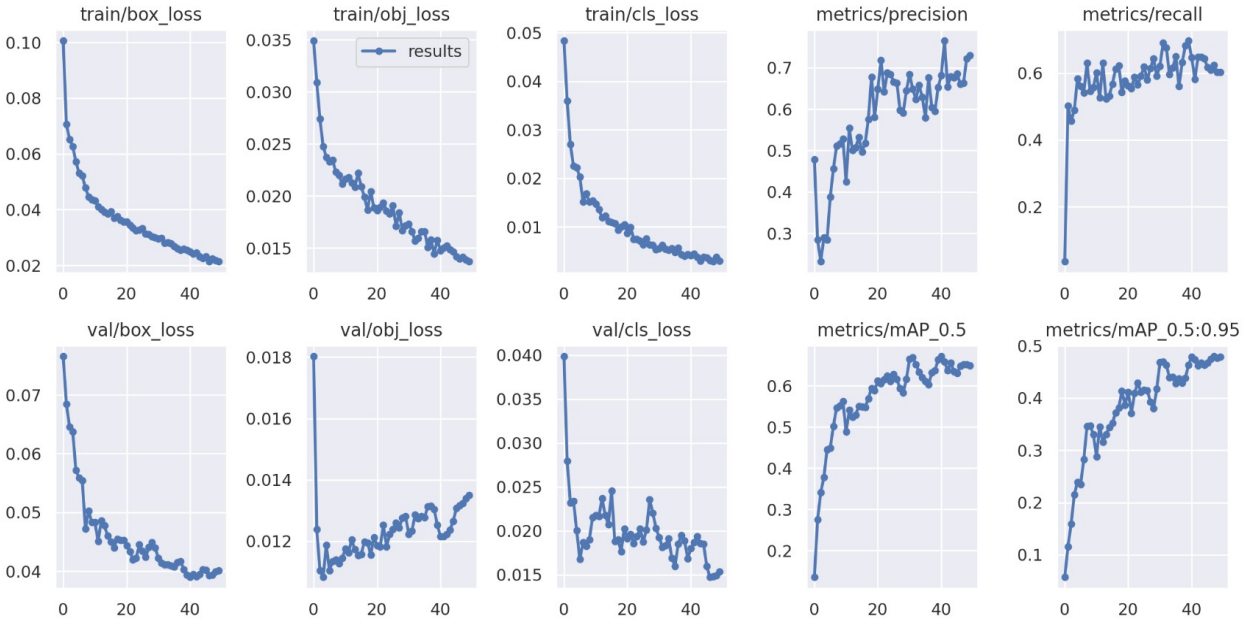


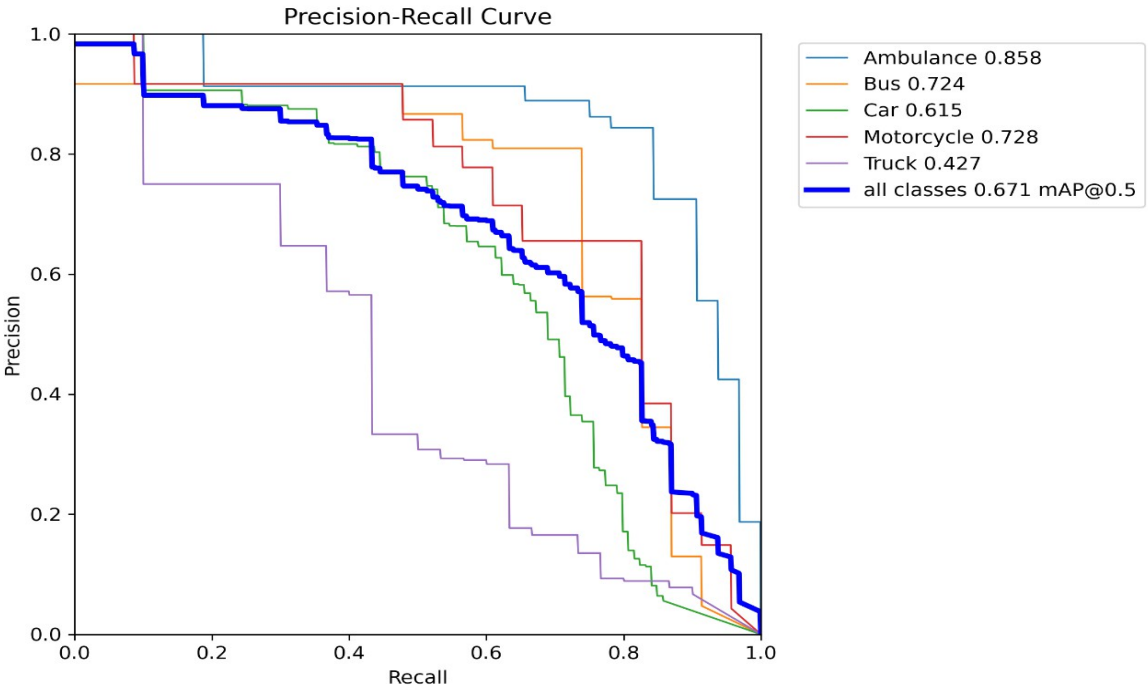
Εκπαίδευση μοντέλου YOLOv5

Καθώς το YOLOv5 είναι ένας ανιχνευτής αντικειμένων ενός σταδίου, έχει τρία σημαντικά μέρη όπως οποιοσδήποτε άλλος ανιχνευτής αντικειμένων ενός σταδίου:α) το Model Backbone που χρησιμοποιείται κυρίως για την εξαγωγή σημαντικών χαρακτηριστικών από τη δεδομένη εικόνα εισόδου.β) το Model Neck που χρησιμοποιείται κυρίως για τη δημιουργία πυραμίδων χαρακτηριστικών. Οι πυραμίδες χαρακτηριστικών βοηθούν στον εντοπισμό του ίδιου αντικειμένου με διαφορετικά μεγέθη και κλίμακες. Υπάρχουν μοντέλα που χρησιμοποιούν διαφορετικούς τύπους τεχνικών πυραμίδας χαρακτηριστικών, όπως FPN, BiFPN, PANet κ.λπ. Στο YOLO v5, το PANet χρησιμοποιείται ως neck για την κατασκευή πυραμίδων χαρακτηριστικών.γ) το Model Head χρησιμοποιείται κυρίως για την εκτέλεση του τελικού τμήματος ανίχνευσης. Εφαρμόζει πλαίσια αγκύρωσης σε χαρακτηριστικά και δημιουργεί τελικά διανύσματα εξόδου με πιθανότητες κλάσεων, βαθμολογίες αντικειμενικότητας και οριοθετημένα πλαίσια. Στο μοντέλο YOLO v5, η κεφαλή είναι ίδια με την προηγούμενη έκδοση YOLO v3.[115]



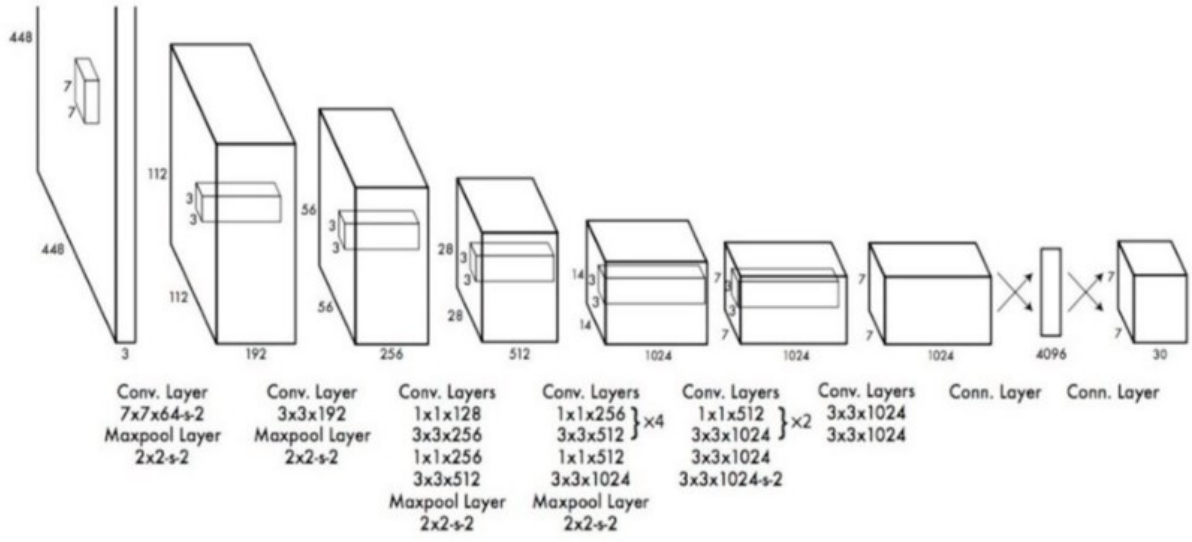
Η εκπαίδευση έγινε σε 125 φωτογραφίες και για 50 εποχές





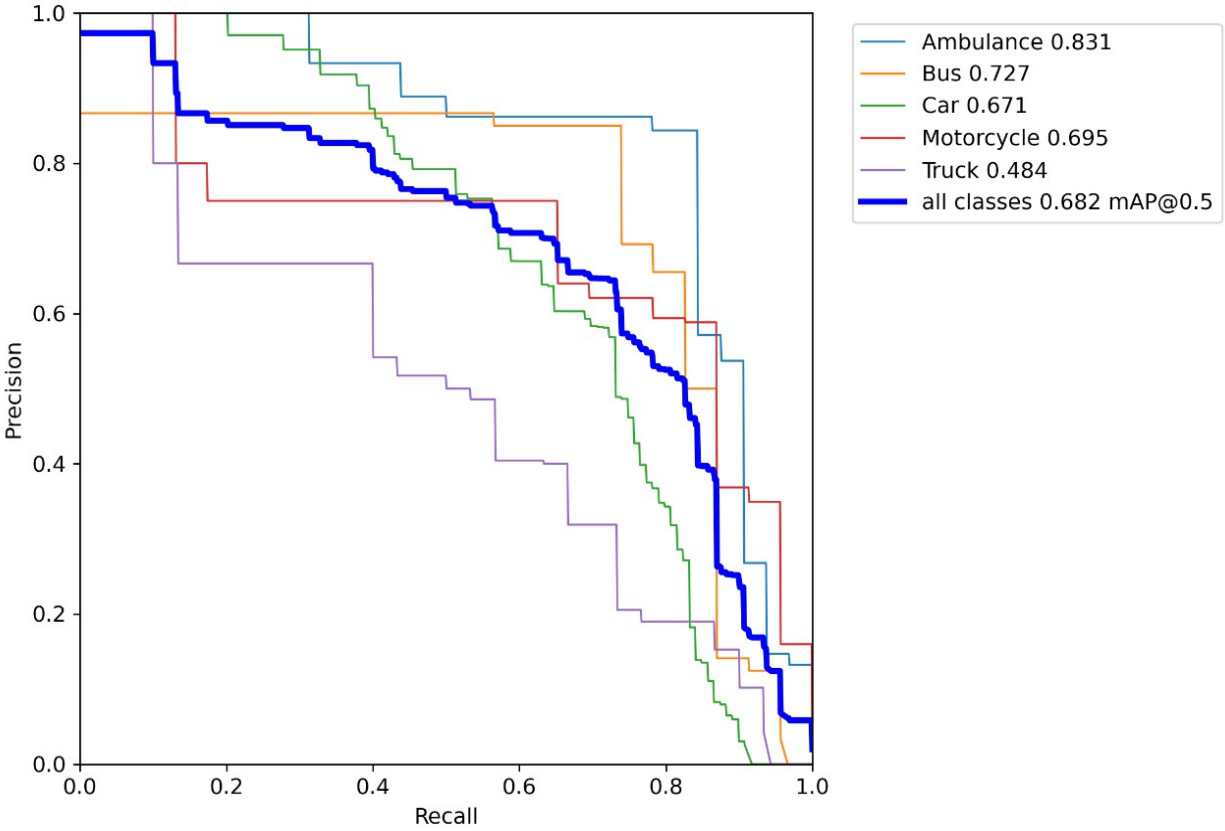
Εκπαίδευση μοντέλου YOLOv7

Το YOLOv7 είναι ένας ανιχνευτής αντικειμένων σε πραγματικό χρόνο ενός σταδίου. Παρουσιάστηκε στην οικογένεια YOLO τον Ιούλιο του ‘22**.** Η αρχιτεκτονική YOLO βασίζεται στο FCNN (Fully Connected Neural Network).Το E-ELAN είναι το υπολογιστικό μπλοκ στη ραχοκοκαλιά του YOLOv7. Το προτεινόμενο E-ELAN χρησιμοποιεί επέκταση, ανακατεύθυνση και συγχώνευση για να επιτύχει τη δυνατότητα συνεχούς βελτίωσης της ικανότητας εκμάθησης του δικτύου χωρίς να καταστρέφει την αρχική διαδρομή κλίσης. Με απλά λόγια, η αρχιτεκτονική E-ELAN επιτρέπει στο πλαίσιο να μαθαίνει καλύτερα.[116]



Η εκπαίδευση έγινε σε 125 φωτογραφίες και για 50 εποχές

```
             Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 4/4 [00:03<00:00,  1.10it/s]
               all         125         227       0.681       0.658       0.682       0.502
         Ambulance         125          32        0.77       0.844       0.831       0.693
               Bus         125          23       0.669       0.783       0.727        0.54
               Car         125         119       0.664       0.614       0.671       0.453
         Motorcycle         125          23       0.634       0.652       0.695       0.497
             Truck         125          30       0.666         0.4       0.484       0.329
50 epochs completed in 0.409 hours.
```



ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρατηρούμε πως η αναγνώριση του φορτηγού έχει το χειρότερο αποτέλεσμα οπότε θα χρειαστούν περισσότερες εικόνες με φορτηγά για εκπαίδευση και περισσότερες εποχές εκπαίδευσης σε όλα τα μοντέλα. Παρατηρήσαμε πως και τα 3 μοντέλα έχουν mAP παρόμοιο με λίγο καλύτερο το YOLOV7 το οποίο τα πήγε και λίγο καλύτερα και στην αναγνώριση φορτηγών. Γρηγορότερο μοντέλο είναι εμφανώς το YOLOV5 όπου διάλεξα για εκπαίδευση το medium μοντέλο και πέτυχε ταχύτητα (0.213 hours) σχεδόν μισή σε σχέση με τα άλλα 2 μοντέλα που είχαν ταχύτητες 0.389 και 0.409 ώρες αντίστοιχα.

ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ

Αρχικά εγκαθίστανται τα βασικά πακέτα

```
import os
import glob as glob
import matplotlib.pyplot as plt
import cv2
import requests
import random
import numpy as np

np.random.seed(42)
```

Στη συνέχεια ορίζουμε τις υπερπαραμέτρους τι σταθερές και για πόσες εποχές θα εκπαιδεύσουμε το μοντέλο. Όταν το TRAIN = False, τότε το τελευταίο εκπαιδευμένο μοντέλο θα χρησιμοποιηθεί για να εξάγουμε συμπεράσματα.

```
TRAIN = True
# Number of epochs to train for.
EPOCHS = 25
```

Έπειτα κάνουμε λήψη και προετοιμασία του συνόλου δεδομένων που θα

χρησιμοποιήσουμε για την εκπαίδευση του ανιχνευτή αντικειμένων YOLO.

```
if not os.path.exists('train'):
    !curl -L "https://public.roboflow.com/ds/xKLV14HbTF?key=aJzo7msVta" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip

    dirs = ['train', 'valid', 'test']

    for i, dir_name in enumerate(dirs):
        all_image_names = sorted(os.listdir(f"{dir_name}/images/"))
        for j, image_name in enumerate(all_image_names):
            if (j % 2) == 0:
                file_name = image_name.split('.jpg')[0]
                os.remove(f"{dir_name}/images/{image_name}")
                os.remove(f"{dir_name}/labels/{file_name}.txt")
```

Το σύνολο δεδομένων είναι δομημένο με τον ακόλουθο τρόπο:

```
├── data.yaml
├── README.dataset.txt
├── README.roboflow.txt
├── test
│   ├── images
│   └── labels
├── train
│   ├── images
│   └── labels
└── valid
    ├── images
    └── labels
```

Το αρχείο δεδομένων YAML περιέχει τη διαδρομή προς τις εικόνες και τις ετικέτες εκπαίδευσης . Αυτό το αρχείο περιέχει επίσης τα ονόματα κλάσεων από το σύνολο δεδομένων. Το σύνολο δεδομένων περιέχει 5 κατηγορίες: 'Ασθενοφόρο', 'Λεωφορείο', 'Αυτοκίνητο', 'Μοτοσικλέτα', 'Φορτηγό'.

```
train: ../train/images
val: ../valid/images

nc: 5
names: ['Ambulance', 'Bus', 'Car', 'Motorcycle', 'Truck']
```

Δημιουργία συνάρτησης που θα τη μετατρέψει από μορφή [x_center, y_center, width, height] σε μορφή [x_min, y_min, x_max, y_max] και εμφάνιση μερικών εικόνων.

```
class_names = ['Ambulance', 'Bus', 'Car', 'Motorcycle', 'Person']
colors = np.random.uniform(0, 255, size=(len(class_names), 3))
```

```
# Function to convert bounding boxes in YOLO format to xmin, ymin, xmax, ymax.
def yolo2bbox(bboxes):
    xmin, ymin = bboxes[0]-bboxes[2]/2, bboxes[1]-bboxes[3]/2
    xmax, ymax = bboxes[0]+bboxes[2]/2, bboxes[1]+bboxes[3]/2
    return xmin, ymin, xmax, ymax
```

```
def plot_box(image, bboxes, labels):
    # Need the image height and width to denormalize
    # the bounding box coordinates
    h, w, _ = image.shape
    for box_num, box in enumerate(bboxes):
        x1, y1, x2, y2 = yolo2bbox(box)
        # denormalize the coordinates
        xmin = int(x1*w)
        ymin = int(y1*h)
        xmax = int(x2*w)
        ymax = int(y2*h)
        width = xmax - xmin
        height = ymax - ymin

        class_name = class_names[int(labels[box_num])]
```

```
def plot(image_paths, label_paths, num_samples):
    all_training_images = glob.glob(image_paths)
    all_training_labels = glob.glob(label_paths)
    all_training_images.sort()
    all_training_labels.sort()

    num_images = len(all_training_images)

    plt.figure(figsize=(15, 12))
    for i in range(num_samples):
        j = random.randint(0,num_images-1)
        image = cv2.imread(all_training_images[j])
        with open(all_training_labels[j], 'r') as f:
            bboxes = []
            labels = []
            label_lines = f.readlines()
            for label_line in label_lines:
                label = label_line[0]
                bbox_string = label_line[2:]
                x_c, y_c, w, h = bbox_string.split(' ')
                x_c = float(x_c)
                y_c = float(y_c)
                w = float(w)
                h = float(h)
                bboxes.append([x_c, y_c, w, h])
                labels.append(label)
        result_image = plot_box(image, bboxes, labels)
        plt.subplot(2, 2, i+1)
        plt.imshow(result_image[:, :, ::-1])
        plt.axis('off')
    plt.subplots_adjust(wspace=0)
    plt.tight_layout()
    plt.show()
```

```
# Visualize a few training images.
plot(
    image_paths='train/images/*',
    label_paths='train/labels/*',
    num_samples=2,
)
```

Μετά γράφουμε τις βοηθητικές συναρτήσεις που χρειαζόμαστε για την καταγραφή των αποτελεσμάτων στο notebook κατά την εκπαίδευση των μοντέλων και εκπαιδεύουμε το μοντέλο μας.

```
def set_res_dir():
    # Directory to store results
    res_dir_count = len(glob.glob('runs/train/*'))
    print(f"Current number of result directories: {res_dir_count}")
    if TRAIN:
        RES_DIR = f"results_{res_dir_count+1}"
        print(RES_DIR)
    else:
        RES_DIR = f"results_{res_dir_count}"
    return RES_DIR
```

```
def monitor_tensorboard():
    %load_ext tensorboard
    %tensorboard --logdir runs/train
```

```
if not os.path.exists('yolov5'):
    !git clone https://github.com/ultralytics/yolov5.git
```

```
%cd yolov5/
!pwd
```

```
!pip install -r requirements.txt
```

```
monitor_tensorboard()
```

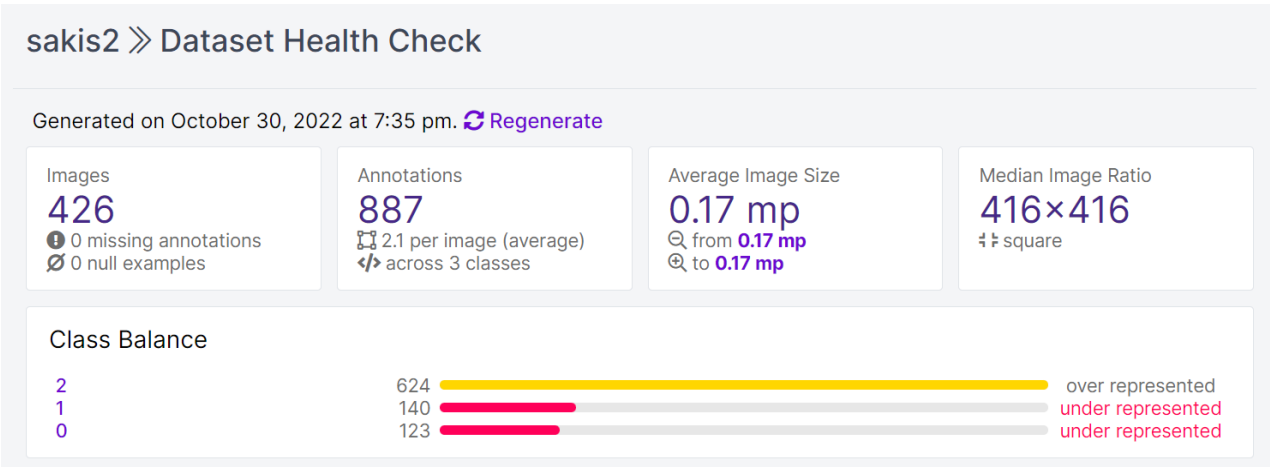
```
RES_DIR = set_res_dir()
if TRAIN:
    !python train.py --data ../data.yaml --weights yolov5m.pt \
    --img 640 --epochs {EPOCHS} --batch-size 16 --name {RES_DIR}
```

Με την παρακάτω συνάρτηση θα ελέγξουμε τις προβλέψεις των εικόνων που αποθηκεύτηκαν κατά τη διάρκεια της εκπαίδευσης και οπτικοποιούμε τις εικόνες αυτές.

```
def visualize(INFER_DIR):
# Visualize inference images.
    INFER_PATH = f"runs/detect/{INFER_DIR}"
    infer_images = glob.glob(f"{INFER_PATH}/*.jpg")
    print(infer_images)
    for pred_image in infer_images:
        image = cv2.imread(pred_image)
        plt.figure(figsize=(19, 16))
        plt.imshow(image[:, :, ::-1])
        plt.axis('off')
        plt.show()
```

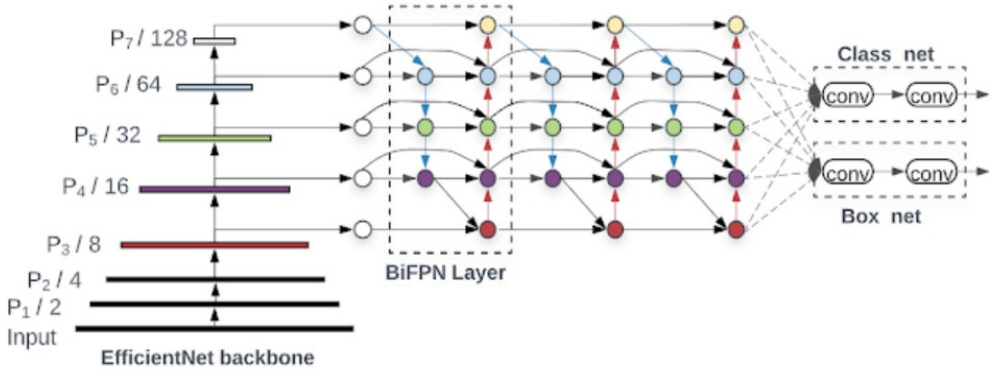
```
show_valid_results(RES_DIR)
```

Διαμορφώνοντας το προηγούμενο dataset στο παρακάτω



όπου 0=ambulance 1=bus 2=car

Και δοκιμάζοντας **efficientdet-d0** που έχει την παρακάτω αρχιτεκτονική



EfficientDet architecture. EfficientDet uses EfficientNet as the backbone network and a newly proposed BiFPN feature network.

Οι ανιχνευτές EfficientDet είναι ανιχνευτές μίας λήψης όπως το SSD και το RetinaNet. Το backbone δικτύου του EfficientDet είναι το EfficientNet προ-εκπαιδευμένο στο σύνολο δεδομένων ImageNet .Το δίκτυο EfficientDet εμπνέεται σε μεγάλο βαθμό από τη δουλειά που έγινε στα μοντέλα EfficientNet. Το συγκεκριμένο μοντέλο προτείνει ένα σταθμισμένο δίκτυο χαρακτηριστικών διπλής κατεύθυνσης και μια προσαρμοσμένη μέθοδο σύνθετης κλιμάκωσης, ώστε να βελτιώσει την ακρίβεια και την αποδοτικότητα. Πράγματι, πολλές μετρήσεις δείχνουν πως η ακρίβεια και η απόδοση του μοντέλου έχει βελτιωθεί σε σύγκριση με άλλα σύγχρονα μοντέλα ανίχνευσης αντικειμένων. [118]

Λάβαμε τα παρακάτω αποτελέσματα

```
DONE (t=0.03s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.541
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.749
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.621
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.109
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.096
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.646
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.488
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.599
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.640
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.300
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.207
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.727
INFO:tensorflow:Eval metrics at step 10000
I1102 00:23:07.107204 139940484757376 model_lib_v2.py:1015] Eval metrics at step 10000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP: 0.541068
I1102 00:23:07.115260 139940484757376 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP: 0.541068
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.50IOU: 0.749389
I1102 00:23:07.116733 139940484757376 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP@.50IOU: 0.749389
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.75IOU: 0.621153
I1102 00:23:07.118157 139940484757376 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP@.75IOU: 0.621153
```

Ακόμη έγινε δοκιμή σε **ssd-mobilenet-v1** που έχει την παρακάτω αρχιτεκτονική

Το MobileNet είναι ένα από τα πολλά μοντέλα βαθιάς συνέλιξης που έχουμε στη διάθεσή μας. Το MobileNet είναι ένα μοντέλο αρχιτεκτονικής νευρωνικού δικτύου συνέλιξης (CNN) που εστιάζει στην ταξινόμηση εικόνας για εφαρμογές σε κινητές συσκευές. Αντί να χρησιμοποιεί τα τυπικά επίπεδα συνέλιξης, χρησιμοποιεί διαχωρίσιμα επίπεδα συνέλιξης κατά βάθος. Αυτό που κάνει αυτό το μοντέλο να ξεχωρίζει είναι ότι η αρχιτεκτονική του μειώνει το υπολογιστικό κόστος και απαιτεί πολύ χαμηλή υπολογιστική ισχύ. Ο ssd Multibox ανιχνευτής απλής λήψης είναι ένας αλγόριθμος που παίρνει μόνο μία λήψη για να ανιχνεύσει πολλά αντικείμενα στην εικόνα χρησιμοποιώντας το multibox. Χρησιμοποιεί ένα ενιαίο βαθύ νευρωνικό δίκτυο για να το πετύχει αυτό. Αυτός ο ανιχνευτής λειτουργεί σε διαφορετικές κλίμακες, επομένως είναι σε θέση να ανιχνεύει αντικείμενα διαφόρων μεγεθών στην εικόνα. Από τον συνδυασμό αυτών προκύπτει το Mobilenet SSD που είναι ένα μοντέλο ανίχνευσης αντικειμένων που υπολογίζει το πλαίσιο οριοθέτησης εξόδου και την κλάση αντικειμένου από την εικόνα εισόδου. Αυτό το μοντέλο ανίχνευσης αντικειμένων Single Shot Detector (SSD) χρησιμοποιεί το Mobilenet ως βάση και μπορεί να επιτύχει γρήγορη ανίχνευση αντικειμένων βελτιστοποιημένη για κινητές συσκευές.[119]

Λάβαμε τα παρακάτω αποτελέσματα

```
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.12s).
Accumulating evaluation results...
DONE (t=0.03s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.555
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.747
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.609
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.005
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.117
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.642
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.510
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.618
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.650
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.400
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.216
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.732
INFO:tensorflow:Eval metrics at step 10000
I1119 16:41:49.869290 140668355159936 model_lib_v2.py:1015] Eval metrics at step 10000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP: 0.554773
I1119 16:41:49.877404 140668355159936 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP: 0.554773
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.50IOU: 0.747045
I1119 16:41:49.879258 140668355159936 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP@.50IOU: 0.747045
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.75IOU: 0.608729
I1119 16:41:49.880795 140668355159936 model_lib_v2.py:1018]     + DetectionBoxes_Precision/mAP@.75IOU: 0.608729
```

ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ

Αρχικά εγκαθιστώ τα πακέτα που θα χρειαστούν

```
import os
import pathlib

# Clone the tensorflow models repository if it doesn't already exist
if "models" in pathlib.Path.cwd().parts:
  while "models" in pathlib.Path.cwd().parts:
    os.chdir('..')
elif not pathlib.Path('models').exists():
  !git clone --depth 1 https://github.com/tensorflow/models
```

```
# Install the Object Detection API
%%bash
cd models/research/
protoc object_detection/protos/*.proto --python_out=.
cp object_detection/packages/tf2/setup.py .
python -m pip install .
```

```
import matplotlib
import matplotlib.pyplot as plt

import os
import random
import io
import imageio
import glob
import scipy.misc
import numpy as np
from six import BytesIO
from PIL import Image, ImageDraw, ImageFont
```

```
from PIL import Image, ImageDraw, ImageFont
from IPython.display import display, Javascript
from IPython.display import Image as IPyImage

import tensorflow as tf
from object_detection.utils import label_map_util
from object_detection.utils import config_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.utils import colab_utils
from object_detection.builders import model_builder
%matplotlib inline
```

```
#run model builder test
!python /content/models/research/object_detection/builders/model_builder_tf2_test.py
```

```
def load_image_into_numpy_array(path):
  """Load an image from file into a numpy array.
  Puts image into numpy array to feed into tensorflow graph.
  Note that by convention we put it into a numpy array with shape
  (height, width, channels), where channels=3 for RGB.
  Args:
    path: a file path.
  Returns:
    uint8 numpy array with shape (img_height, img_width, 3)
  """
  img_data = tf.io.gfile.GFile(path, 'rb').read()
  image = Image.open(BytesIO(img_data))
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape(
```

```
  return np.array(image.getdata()).reshape(
      (im_height, im_width, 3)).astype(np.uint8)

def plot_detections(image_np,
                    boxes,
                    classes,
                    scores,
                    category_index,
                    figsize=(12, 16),
                    image_name=None):
  """Wrapper function to visualize detections.
  Args:
    image_np: uint8 numpy array with shape (img_height, img_width, 3)
    boxes: a numpy array of shape [N, 4]
    classes: a numpy array of shape [N]. Note that class indices are 1-based,
      and match the keys in the label map.
    scores: a numpy array of shape [N] or None.  If scores=None, then
      this function assumes that the boxes to be plotted are groundtruth
      boxes and plot all boxes as black with no classes or scores.
    category_index: a dict containing category dictionaries (each holding
      category index `id` and category name `name`) keyed by category indices.
    figsize: size for the figure.
    image_name: a name for the image file.
  """
  image_np_with_annotations = image_np.copy()
  viz_utils.visualize_boxes_and_labels_on_image_array(
      image_np_with_annotations,
      boxes,
      classes,
      scores,
      category_index,
      use_normalized_coordinates=True,
      min_score_thresh=0.8)
  if image_name:
    plt.imsave(image_name, image_np_with_annotations)
  else:
    plt.imshow(image_np_with_annotations)
```

Προετοιμασία των δεδομένων εκπαίδευσης ανίχνευσης. Το Roboflow δημιουργεί αυτόματα τα αρχεία TFRecord και label_map που χρειαζόμαστε .

```
#follow the link below to get your download code from from Roboflow
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(model_format="tfrecord", notebook="roboflow-tf2-od")
```

```
#Downloading data from Roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="ZLDYIpYSxuzuY3gvA6n2")
project = rf.workspace().project("sakis2")
dataset = project.version("2").download("tfrecord")
```

```
# NOTE: Update these TFRecord names from "cells" and "cells_label_map" to your files!
test_record_fname = dataset.location + '/test/vehicles.tfrecord'
train_record_fname = dataset.location + '/train/vehicles.tfrecord'
label_map_pbtxt_fname = dataset.location + '/train/vehicles_label_map.pbtxt'
```

Ορισμός μοντέλου που θα εκπαιδεύσουμε με το tensorflow.

```
##change chosen model to deploy different models available in the TF2 object detection zoo
MODELS_CONFIG = {
    'efficientdet-d0': {
        'model_name': 'efficientdet_d0_coco17_tpu-32',
        'base_pipeline_file': 'ssd_efficientdet_d0_512x512_coco17_tpu-8.config',
        'pretrained_checkpoint': 'efficientdet_d0_coco17_tpu-32.tar.gz',
        'batch_size': 16
    },
     'efficientdet-d1': {
        'model_name': 'efficientdet_d1_coco17_tpu-32',
        'base_pipeline_file': 'ssd_efficientdet_d1_640x640_coco17_tpu-8.config',
        'pretrained_checkpoint': 'efficientdet_d1_coco17_tpu-32.tar.gz',
        'batch_size': 16
    },
    'faster_rcnn_resnet50_v1': {
        'model_name': 'faster_rcnn_resnet50_v1_640x640_coco17_tpu-8',
        'base_pipeline_file': 'faster_rcnn_resnet50_v1_640x640_coco17_tpu-8.config',
        'pretrained_checkpoint': 'faster_rcnn_resnet50_v1_640x640_coco17_tpu-8.tar.gz',
        'batch_size': 16
    },
    'ssd_mobilenet_v1': {
        'model_name': 'ssd_mobilenet_v1_fpn_640x640_coco17_tpu-8',
        'base_pipeline_file': 'ssd_mobilenet_v1_fpn_640x640_coco17_tpu-8.config',
        'pretrained_checkpoint': 'ssd_mobilenet_v1_fpn_640x640_coco17_tpu-8.tar.gz',
        'batch_size': 16
    }
}
#in this tutorial we implement the lightweight, smallest state of the art efficientdet model
#if you want to scale up tot larger efficientdet models you will likely need more compute!
chosen_model = 'efficientdet-d1'
num_steps = 10000 #The more steps, the longer the training. Increase if your loss function is still decreasing and validation metrics are increasing.
num_eval_steps = 500 #Perform evaluation after so many steps
model_name = MODELS_CONFIG[chosen_model]['model_name']
pretrained_checkpoint = MODELS_CONFIG[chosen_model]['pretrained_checkpoint']
base_pipeline_file = MODELS_CONFIG[chosen_model]['base_pipeline_file']
batch_size = MODELS_CONFIG[chosen_model]['batch_size'] #if you can fit a large batch in memory, it may speed up your training
```

```
#download pretrained weights
%mkdir /content/models/research/deploy/
%cd /content/models/research/deploy/
import tarfile
download_tar = 'http://download.tensorflow.org/models/object_detection/tf2/20200711/' + pretrained_checkpoint

!wget {download_tar}
tar = tarfile.open(pretrained_checkpoint)
tar.extractall()
tar.close()
```

```
#download base training configuration file
%cd /content/models/research/deploy
download_config = 'https://raw.githubusercontent.com/tensorflow/models/master/research/object_detection/configs/tf2/' + base_pipeline_file
!wget {download_config}
```

```
#prepare
pipeline_fname = '/content/models/research/deploy/' + base_pipeline_file
fine_tune_checkpoint = '/content/models/research/deploy/' + model_name + '/checkpoint/ckpt-0'

def get_num_classes(pbtxt_fname):
    from object_detection.utils import label_map_util
    label_map = label_map_util.load_labelmap(pbtxt_fname)
    categories = label_map_util.convert_label_map_to_categories(
        label_map, max_num_classes=90, use_display_name=True)
    category_index = label_map_util.create_category_index(categories)
    return len(category_index.keys())
num_classes = get_num_classes(label_map_pbtxt_fname)
```

```
#write custom configuration file by slotting our dataset, model checkpoint, and training parameters into the base pipeline
import re
%cd /content/models/research/deploy
print('writing custom configuration file')
with open(pipeline_fname) as f:
    s = f.read()
with open('pipeline_file.config', 'w') as f:
    s = re.sub('fine_tune_checkpoint: ".*?"',
               'fine_tune_checkpoint: "{}"'.format(fine_tune_checkpoint), s)
    s = re.sub(
        '(input_path: ".*?)(PATH_TO_BE_CONFIGURED/train)(.*?")', 'input_path: "{}"'.format(train_record_fname), s)
    s = re.sub(
        '(input_path: ".*?)(PATH_TO_BE_CONFIGURED/val)(.*?")', 'input_path: "{}"'.format(test_record_fname), s)
    s = re.sub(
        'label_map_path: ".*?"', 'label_map_path: "{}"'.format(label_map_pbtxt_fname), s)
    s = re.sub('batch_size: [0-9]+',
               'batch_size: {}'.format(batch_size), s)
    s = re.sub('num_steps: [0-9]+',
               'num_steps: {}'.format(num_steps), s)
    s = re.sub('num_classes: [0-9]+',
               'num_classes: {}'.format(num_classes), s)
    s = re.sub(
        'fine_tune_checkpoint_type: "classification"', 'fine_tune_checkpoint_type: "{}"'.format('detection'), s)
    f.write(s)
```

```
%cat /content/models/research/deploy/pipeline_file.config
```

```
pipeline_file = '/content/models/research/deploy/pipeline_file.config'
model_dir = '/content/training/'
```

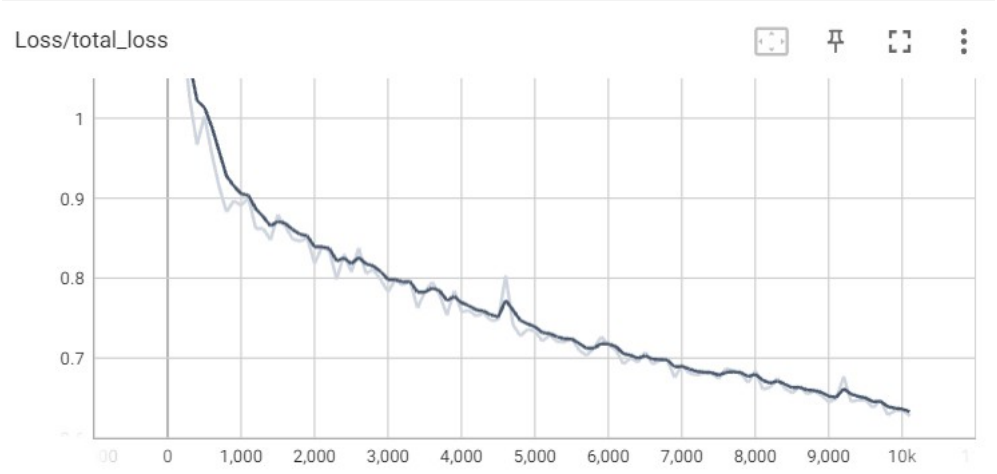
Έναρξη εκπαίδευσης και ενεργοποίηση tensorboard

```
!python /content/models/research/object_detection/model_main_tf2.py \
    --pipeline_config_path={pipeline_file} \
    --model_dir={model_dir} \
    --alsologtostderr \
    --num_train_steps={num_steps} \
    --sample_1_of_n_eval_examples=1 \
    --num_eval_steps={num_eval_steps}
```

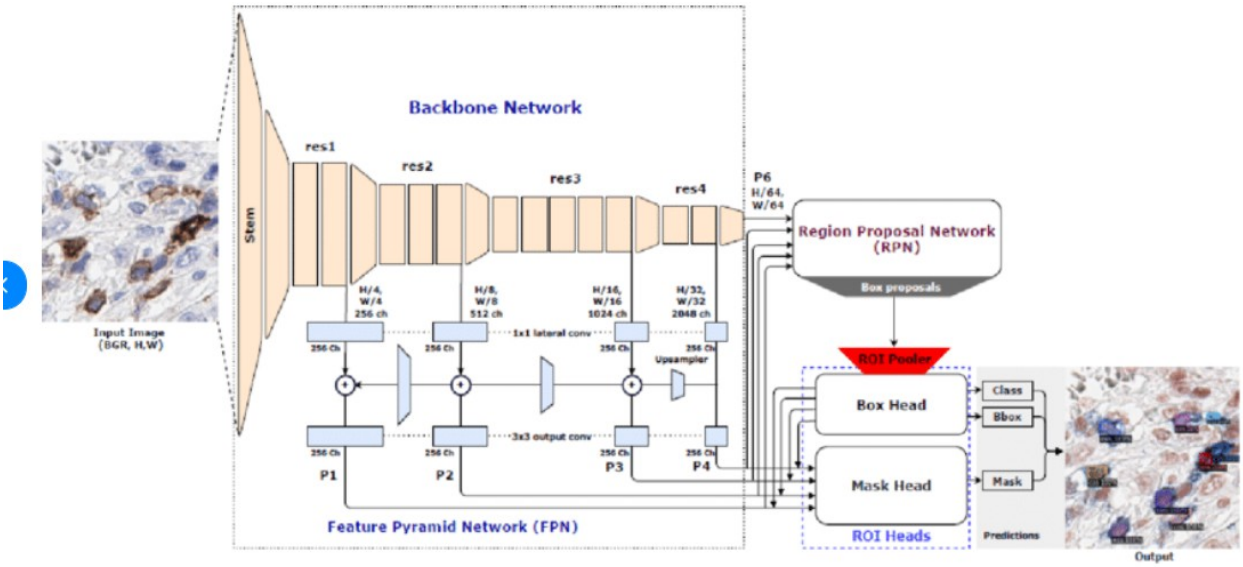
```
#run model evaluation to obtain performance metrics
!python /content/models/research/object_detection/model_main_tf2.py \
    --pipeline_config_path={pipeline_file} \
    --model_dir={model_dir} \
    --checkpoint_dir={model_dir} \
#Not yet implemented for EfficientDet
```

```
%load_ext tensorboard
%tensorboard --logdir '/content/training/train'
```

Γράφημα μείωσης συνολικής απώλειας



Με το ίδιο dataset έκανα δοκιμή σε **detectron2 (Mask R-CNN)** με την παρακάτω αρχιτεκτονική



Το Mask R-CNN είναι ένα μοντέλο ανίχνευσης αντικειμένων που βασίζεται σε βαθιά συνελικτικά νευρωνικά δίκτυα (CNN) και αναπτύχθηκε από μια ομάδα ερευνητών του Facebook το 2017. Το μοντέλο μπορεί να επιστρέψει τόσο το πλαίσιο οριοθέτησης όσο και μια μάσκα για κάθε ανιχνευμένο αντικείμενο σε μια εικόνα. Το Mask R-CNN είναι μια δημοφιλής τεχνική βαθιάς μάθησης που εκτελεί τμηματοποίηση σε επίπεδο pixel σε αντικείμενα που έχουν εντοπιστεί. Ο αλγόριθμος Mask R-CNN μπορεί να φιλοξενήσει πολλαπλές κλάσεις και επικαλυπτόμενα αντικείμενα. Το Mask R-CNN επεκτείνει το Faster R-CNN για την επίλυση εργασιών τμηματοποίησης. Αυτό το επιτυγχάνει προσθέτοντας έναν κλάδο για την πρόβλεψη μιας μάσκας αντικειμένου παράλληλα με τον υπάρχοντα κλάδο για την αναγνώριση του πλαισίου οριοθέτησης. Η αρχιτεκτονική του αποτελείται από το ResNet, ένα δίκτυο πυραμίδας χαρακτηριστικών (FPN) και τέσσερα δομικά μπλοκ συνέλιξης στο ResNet που αποτελούν τέσσερις χάρτες χαρακτηριστικών και αντιπροσωπεύουν διαφορετικά επίπεδα σημασιολογικής πληροφορίας.[120]

Λάβαμε τα παρακάτω αποτελέσματα

```
DONE (t=0.03s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.606
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.842
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.698
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.134
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.256
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.681
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.558
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.703
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.710
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.350
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.402
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.767
[10/30 20:30:56 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl   |
|:------:|:------:|:------:|:------:|:------:|:------:|
| 60.623 | 84.152 | 69.780 | 13.393 | 25.556 | 68.135 |
[10/30 20:30:56 d2.evaluation.coco_evaluation]: Per-category bbox AP:
| category   | AP     | category   | AP     | category   | AP     |
|:-----------|:-------|:-----------|:-------|:-----------|:-------|
| vehicles   | nan    | 0          | 63.781 | 1          | 62.800 |
| 2          | 55.288 |            |        |            |        |
OrderedDict([('bbox',
              {'AP': 60.62304359559022,
               'AP50': 84.15211140458398,
               'AP75': 69.78033089008153,
```

ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ

Εγκαθιστώ τα βασικά πακέτα για το detectron2.

```
# install dependencies: (use cu101 because colab has CUDA 10.1)
!pip install -U torch==1.5 torchvision==0.6 -f https://download.pytorch.org/whl/cu101/torch_stable.html
!pip install cython pyyaml==5.1
!pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'
import torch, torchvision
print(torch.__version__, torch.cuda.is_available())
!gcc --version
# opencv is pre-installed on colab
```

```
# install detectron2:
!pip install detectron2==0.1.3 -f https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.5/index.html
```

```
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

import numpy as np
import cv2
import random
from google.colab.patches import cv2_imshow

from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog
from detectron2.data.catalog import DatasetCatalog
```

Εισαγωγή και καταχώρηση των δεδομένων

```
!curl -L "https://app.roboflow.com/ds/t2oUjgOIId?key=Dd13rnIzuE" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
```

```
from detectron2.data.datasets import register_coco_instances
register_coco_instances("my_dataset_train", {}, "/content/train/_annotations.coco.json", "/content/train")
register_coco_instances("my_dataset_val", {}, "/content/valid/_annotations.coco.json", "/content/valid")
register_coco_instances("my_dataset_test", {}, "/content/test/_annotations.coco.json", "/content/test")
```

```
#visualize training data
my_dataset_train_metadata = MetadataCatalog.get("my_dataset_train")
dataset_dicts = DatasetCatalog.get("my_dataset_train")

import random
from detectron2.utils.visualizer import Visualizer

for d in random.sample(dataset_dicts, 3):
    img = cv2.imread(d["file_name"])
    visualizer = Visualizer(img[:, :, ::-1], metadata=my_dataset_train_metadata, scale=0.5)
    vis = visualizer.draw_dataset_dict(d)
    cv2_imshow(vis.get_image()[:, :, ::-1])
```

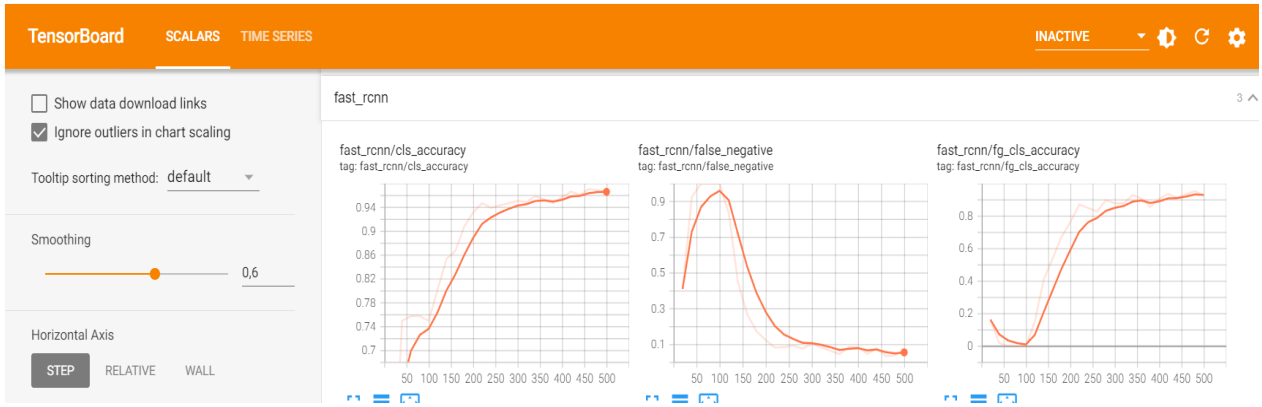
Εκπαίδευση μοντέλου

```
from detectron2.engine import DefaultTrainer
from detectron2.evaluation import COCOEvaluator

class CocoTrainer(DefaultTrainer):

  @classmethod
  def build_evaluator(cls, cfg, dataset_name, output_folder=None):

    if output_folder is None:
        os.makedirs("coco_eval", exist_ok=True)
        output_folder = "coco_eval"

    return COCOEvaluator(dataset_name, cfg, False, output_folder)
```

```python
from detectron2.config import get_cfg
import os

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("my_dataset_train",)
cfg.DATASETS.TEST = ("my_dataset_val",)

cfg.DATALOADER.NUM_WORKERS = 4
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 4
cfg.SOLVER.BASE_LR = 0.001

cfg.SOLVER.WARMUP_ITERS = 1000
cfg.SOLVER.MAX_ITER = 500 #adjust up if val mAP is still rising, adjust down if overfit
cfg.SOLVER.STEPS = (1000, 1500)
cfg.SOLVER.GAMMA = 0.05

cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 64
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 4 #your number of classes + 1

cfg.TEST.EVAL_PERIOD = 500

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = CocoTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Ενεργοποίηση tensorboard

```
# Look at training curves in tensorboard:
%load_ext tensorboard
%tensorboard --logdir output
```



Αξιολόγηση δοκιμής

```
#test evaluation
from detectron2.data import DatasetCatalog, MetadataCatalog, build_detection_test_loader
from detectron2.evaluation import COCOEvaluator, inference_on_dataset

cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.85
predictor = DefaultPredictor(cfg)
evaluator = COCOEvaluator("my_dataset_test", cfg, False, output_dir="./output/")
val_loader = build_detection_test_loader(cfg, "my_dataset_test")
inference_on_dataset(trainer.model, val_loader, evaluator)
```

Συμπεράσματα με τα αποθηκευμένα βάρη

```
%ls ./output/
```

```
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.DATASETS.TEST = ("my_dataset_test", )
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.7   # set the testing threshold for this model
predictor = DefaultPredictor(cfg)
test_metadata = MetadataCatalog.get("my_dataset_test")
```

```
from detectron2.utils.visualizer import ColorMode
import glob

for imageName in glob.glob('/content/test/*jpg'):
  im = cv2.imread(imageName)
  outputs = predictor(im)
  v = Visualizer(im[:, :, ::-1],
                 metadata=test_metadata,
                 scale=0.8
                  )
  out = v.draw_instance_predictions(outputs["instances"].to("cpu"))
  cv2_imshow(out.get_image()[:, :, ::-1])
```

Οπτικοποίηση αποτελεσμάτων

Πρίν περάσουμε στο τελεύταιο μας πείραμα θα κάνουμε μια αναφορά στις μετρικές MOTA και MOTP

MOTA-MOTP

Το MOTA (Multiple Object Tracking Accuracy) και το MOTP (Multiple Object Tracking Precision) είναι δύο μετρικές που χρησιμοποιούνται στο πρόβλημα της παρακολούθησης πολλαπλών αντικειμένων (Multiple Object Tracking - MOT) σε εικόνες ή βίντεο.

Το MOTA μετρά την ακρίβεια της παρακολούθησης πολλαπλών αντικειμένων, λαμβάνοντας υπόψη τα σφάλματα που συμβαίνουν κατά τη διάρκεια της παρακολούθησης. Υπολογίζεται ως η απόλυτη διαφορά μεταξύ του πραγματικού αριθμού αντικειμένων που παρακολουθήθηκαν και του αριθμού αντικειμένων που ανιχνεύθηκαν λανθασμένα, προστίθενται με το πραγματικό αριθμό των αντικειμένων που χάθηκαν, διαιρούνται με το πραγματικό αριθμό των αντικειμένων και αυτό το αποτέλεσμα αφαιρείται από τη μονάδα. Η τελική τιμή του MOTA εκφράζεται σε ποσοστό.

Το MOTP μετρά την ακρίβεια του εντοπισμού θέσης των αντικειμένων κατά τη διάρκεια της παρακολούθησης. Υπολογίζεται ως ο μέσος όρος της ευκλείδειας

απόστασης μεταξύ του κεντρού της πραγματικής θέσης του αντικειμένου και του κεντρού της προβλεπόμενης θέσης του αντικειμένου. Δηλαδή, το MOTP μετράει πόσο κοντά βρίσκεται η προβλεπόμενη θέση του αντικειμένου στην πραγματική θέση του αντικειμένου. Η τελική τιμή του MOTP εκφράζεται συνήθως σε μέσο όρο απόστασης.

Για τον υπολογισμό του MOTA και του MOTP στο πρόβλημα της παρακολούθησης πολλαπλών αντικειμένων, ένας απλός αλγόριθμος περιλαμβάνει τα παρακάτω βήματα:

•Ανίχνευση των αντικειμένων: Οι αλγόριθμοι ανίχνευσης αντικειμένων χρησιμοποιούνται για την εντοπισμό των αντικειμένων σε μια σειρά από εικόνες ή βίντεο. Αυτό δημιουργεί μια αρχική λίστα των αντικειμένων.

•Παρακολούθηση αντικειμένων: Στη συνέχεια, οι αλγόριθμοι παρακολούθησης αντικειμένων χρησιμοποιούνται για να αντιστοιχίσουν τα αντικείμενα μεταξύ των διαφορετικών εικόνων ή βίντεο και να παρακολουθήσουν την κίνησή τους στο χρόνο.

•Υπολογισμός MOTA και MOTP: Στο τέλος της παρκολούθησης, οι τιμές του MOTA και MOTP υπολογίζονται από τη σύγκριση των πραγματικών και προβλεπόμενων θέσεων των αντικειμένων.

Συγκεκριμένα, για τον υπολογισμό του MOTA, πρέπει να αξιολογηθούν τα αποτελέσματα της παρακολούθησης αντικειμένων σε σχέση με τα πραγματικά αντικείμενα. Κατά την αξιολόγηση, τα αντικείμενα χωρίζονται σε τρεις κατηγορίες: α) σωστά ανιχνευμένα (true positive - TP), δηλαδή αντικείμενα που έχουν ανιχνευθεί και παρακολουθηθεί σωστά, β) λανθασμένα ανιχνευμένα (false positive - FP), δηλαδή αντικείμενα που έχουν ανιχνευθεί αλλά δεν υπάρχουν στην πραγματικότητα και γ) αντικείμενα που δεν ανιχνεύθηκαν (false negative - FN), δηλαδή αντικείμενα που υπάρχουν στην πραγματικότητα αλλά δεν ανιχνεύθηκαν.

Για τον υπολογισμό του MOTP, χρειάζεται να υπολογιστεί η μέση απόσταση των προβλεπόμενων θέσεων αντικειμένων από τις πραγματικές τους θέσεις. Αυτό υπολογίζεται ως η μέση απόσταση του κέντρου της πραγματικής θέσης του αντικειμένου και του κέντρου της προβλεπόμενης θέσης του αντικειμένου. Στη συνέχεια, υπολογίζεται το MOTP ως ο μέσος όρος αυτών των αποστάσεων για όλα τα αντικείμενα.

Συνοπτικά, οι τύποι υπολογισμού για το MOTA και το MOTP είναι οι εξής:

$$MOTA = 1 - (\Sigma FP + \Sigma FN + \Sigma ID) / \Sigma GT$$

$$MOTP = \Sigma d / \Sigma c$$

όπου ΣFP, ΣFN, ΣID, ΣGT είναι ο αριθμός των λανθασμένα ανιχνευμένων, των αντικειμένων που δεν ανιχνεύθηκαν, των αντικειμένων που έχουν λανθασμένο ID και των πραγματικών αντικειμένων αντίστοιχα. Το Σd αντιστοιχεί στο άθροισμα των αποστάσεων των προβλεπόμενων θέσεων αντικειμένων από τις πραγματικές τους θέσεις, ενώ το Σc αν αντιστοιχεί στο συνολικό αριθμό των αντικειμένων που εντοπίστηκαν.

Συνολικά, οι δείκτες αυτοί μας δίνουν μια καλή εικόνα της απόδοσης του συστήματος παρακολούθησης αντικειμένων και μπορούν να χρησιμοποιηθούν για τη σύγκριση διαφορετικών αλγορίθμων ή τη βελτίωση της απόδοσης του συστήματος μέσω βελτιστοποίησης των παραμέτρων και των μεθόδων παρακολούθησης αντικειμένων.

ΤΕΛΙΚΟ ΠΕΙΡΑΜΑ

Στο τελευταίο μας πείραμα θα εξετάσουμε το YOLOv4_DeepSort.Αυτό είναι ένα σύστημα ανίχνευσης και παρακολούθησης αντικειμένων που συνδυάζει δύο αλγορίθμους, το YOLOv4 και το DeepSORT, και χρησιμοποιεί το TensorFlow framework για την υλοποίηση τους. Το YOLOv4 είναι ένας αλγόριθμος ανίχνευσης αντικειμένων που χρησιμοποιεί βαθιά συνελικτικά νευρωνικά δίκτυα για να εντοπίσει αντικείμενα σε εικόνες και βίντεο. Αφού εντοπίσει τα αντικείμενα, τα στέλνει στον αλγόριθμο DeepSORT για παρακολούθηση και αναγνώριση τους σε πραγματικό χρόνο. Η συνδυασμένη χρήση αυτών των δύο αλγορίθμων δημιουργεί ένα πολύ ακριβές και αξιόπιστο σύστημα ανίχνευσης και παρακολούθησης αντικειμένων.

Στην υλοποίηση του κώδικα χρησιμοποιήθηκε το COCO (Common Objects in Context) ,το οποίο είναι ένα σύνολο δεδομένων για την αναγνώριση αντικειμένων και την αξιολόγηση αλγορίθμων αναγνώρισης αντικειμένων. Περιλαμβάνει μια μεγάλη

συλλογή εικόνων με ετικέτες αντικειμένων, καθώς και ένα πρότυπο αξιολόγησης απόδοσης αλγορίθμων αναγνώρισης αντικειμένων. Το COCO αποτελεί ένα δημοφιλές benchmark στον χώρο της αναγνώρισης αντικειμένων και χρησιμοποιείται συχνά για την αξιολόγηση και σύγκριση αλγορίθμων αναγνώρισης αντικειμένων.

Κάναμε επίσης χρήση των αρχείων του MOTChallenge. Το MOTChallenge είναι ένας διαγωνισμός που διοργανώνεται κάθε χρόνο από την κοινότητα της υπολογιστικής όρασης (computer vision) και ασχολείται με το πρόβλημα της ανίχνευσης και παρακολούθησης αντικειμένων (Multiple Object Tracking - MOT) σε βίντεο. Ο διαγωνισμός αποτελείται από δύο μέρη: το μέρος της ανίχνευσης αντικειμένων (Detection) και το μέρος της παρακολούθησης αντικειμένων (Tracking). Στο μέρος της ανίχνευσης, ο στόχος είναι να εντοπιστούν όλα τα αντικείμενα σε ένα βίντεο και να τους ανατεθεί μια μοναδική ταυτότητα. Στο μέρος της παρακολούθησης, ο στόχος είναι να παρακολουθηθούν τα αντικείμενα από καρέ σε καρέ και να διατηρηθεί η ταυτότητα τους κατά τη διάρκεια του βίντεο. Ο διαγωνισμός αυτός έχει ως στόχο τη βελτίωση της απόδοσης των αλγορίθμων ανίχνευσης και παρακολούθησης αντικειμένων σε βίντεο και την προώθηση της έρευνας στον τομέα της υπολογιστικής όρασης.

Στη συνέχεια θα προσπαθήσουμε να συγκρίνουμε τα DPM, Faster R-CNN και SDP που είναι τρεις αλγόριθμοι ανίχνευσης αντικειμένων .

Το DPM (Deformable Part-based Model) είναι ένας αλγόριθμος ανίχνευσης αντικειμένων που βασίζεται σε ένα μοντέλο που αποτελείται από παραμετροποιημένα μέρη, τα οποία μπορούν να παραμορφωθούν για να προσαρμοστούν στη μορφολογία των αντικειμένων. Αυτό το μοντέλο χρησιμοποιείται για να εντοπιστούν τα αντικείμενα σε μια εικόνα, καθώς αναζητά τα μέρη που αντιστοιχούν στα αντικείμενα αυτά και στη συνέχεια συνδυάζει αυτές τις πληροφορίες για να εντοπίσει το συνολικό αντικείμενο.

Το Faster R-CNN (Faster Region-based Convolutional Neural Network) είναι ένας αλγόριθμος ανίχνευσης αντικειμένων που βασίζεται σε ένα νευρωνικό δίκτυο συνελικτικών επιπέδων (CNN). Χρησιμοποιεί ένα δίκτυο CNN για να εξάγει χαρακτηριστικά από την εικόνα και ένα δίκτυο R-CNN για να εντοπίσει τα αντικείμενα. Ο Faster R-CNN είναι πολύ αποτελεσματικός στην ανίχνευση αντικειμένων, καθώς χρησιμοποιεί μια προηγμένη στρατηγική που επιτρέπει την αναζήτηση περιοχών της εικόνας που πιθανόν να περιέχουν αντικείμενα, αντί να εξετάζει κάθε πιθανή περιοχή σε όλη την εικόνα.

Το SDP (Structured Deformable Part Model) είναι μια εξέλιξη του DPM που χρησιμοποιεί μια δομημένη προσέγγιση για την ανίχνευση αντικειμένων. Στο SDP, οι παράμετροι του μοντέλου DPM αντικαθίστανται από ένα σύνολο κανόνων, οι οποίοι εκτελούνται σε μια δομημένη αναζήτηση που λαμβάνει υπόψη τις συνδέσεις μεταξύ των μερών των αντικειμένων. Αυτό επιτρέπει στο SDP να αναλύει καλύτερα τη μορφή των αντικειμένων και να εντοπίζει με μεγαλύτερη ακρίβεια τα αντικείμενα σε μια εικόνα.

Τα αποτελέσματα που πήραμε σε διάφορα βίντεο που δοκιμάσαμε χρησιμοποιώντας τον tracker MPNTrack είναι τα παρακάτω.

CLEAR: MPNTrack-pedestrian	MOTA	MOTP	MODA	CLR_Re	CLR_Pr	MTR	PTR	MLR	sMOTA	CLR_TP	CLR_FN	CLR_FP	IDSW
MOT17-02-DPM	39.04	91.754	39.137	39.653	98.714	17.742	35.484	46.774	35.77	7368	11213	96	18
MOT17-02-FRCNN	47.296	91.158	47.43	48.146	98.535	24.194	46.774	29.032	43.039	8946	9635	133	25
MOT17-02-SDP	53.662	90.655	53.888	55.487	97.2	27.419	50	22.581	48.477	10310	8271	297	42
MOT17-04-DPM	65.656	90.874	65.685	65.797	99.831	34.94	36.145	28.916	59.651	31291	16266	53	14
MOT17-04-FRCNN	65.288	90.372	65.303	65.448	99.779	38.554	34.94	26.506	58.987	31125	16432	69	7
MOT17-04-SDP	76.205	89.412	76.235	76.672	99.433	54.217	26.506	19.277	68.087	36463	11094	208	14
MOT17-05-DPM	55.906	85.711	56.166	61.746	91.711	24.06	48.872	27.068	47.083	4271	2646	386	18
MOT17-05-FRCNN	56.6	85.133	56.889	62.383	91.906	31.579	42.857	25.564	47.325	4315	2602	380	20
MOT17-05-SDP	62.137	85.628	62.455	69.018	91.316	36.842	45.113	18.045	52.217	4774	2143	454	22
MOT17-09-DPM	74.742	89.968	74.836	76.469	97.908	53.846	38.462	7.6923	67.07	4072	1253	87	5
MOT17-09-FRCNN	70.16	90.088	70.329	73.484	95.883	53.846	38.462	7.6923	62.876	3913	1412	168	9
MOT17-09-SDP	75.117	89.982	75.192	77.146	97.531	57.692	38.462	3.8462	67.389	4108	1217	104	4
MOT17-10-DPM	62.349	85.934	62.567	65.597	95.585	42.105	42.105	15.789	53.122	8422	4417	389	28
MOT17-10-FRCNN	72.155	85.026	72.529	76.236	95.362	61.404	33.333	5.2632	60.74	9788	3051	476	48
MOT17-10-SDP	73.799	84.853	74.359	79.679	93.741	68.421	28.07	3.5088	61.73	10230	2609	683	72
MOT17-11-DPM	64.985	91.968	65.091	66.384	98.09	24	48	28	59.653	6264	3172	122	10
MOT17-11-FRCNN	70.157	91.481	70.263	71.81	97.891	40	36	24	64.039	6776	2660	146	10
MOT17-11-SDP	75.318	91.221	75.456	77.596	97.315	48	34.667	17.333	68.505	7322	2114	202	13
MOT17-13-DPM	51.048	87.713	51.254	53.066	96.697	27.273	40	32.727	44.528	6178	5464	211	24
MOT17-13-FRCNN	70.718	86.805	71.053	76.284	93.583	58.182	31.818	10	60.652	8881	2761	609	39
MOT17-13-SDP	67.463	86.178	67.806	71.603	94.965	52.727	24.545	22.727	57.566	8336	3306	442	40
COMBINED	64.399	89.143	64.543	66.239	97.503	39.621	38.4	21.978	57.208	223153	113738	5715	482

ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρατηρούμε πως στα περισσότερα βίντεο ο αλγόριθμος DPM υστερεί σε σχέση με τους άλλους δύο ,όσον αφορά το MOTA. Καλύτερη απόδοση σε αυτή την μετρική δείχνει να έχει ο SDP. Αντίθετα στην μετρική MOTP ο αλγόριθμος DPM δείχνει να υπερέχει των άλλων 2 που εμφανίζουν παρόμοια αποτελέσματα.

Βελτιώσεις που μπορούν να γίνουν για να πάρουμε πιο αξιόπιστα αποτελέσματα είναι:

α)Να εκτελεστούν περισσότερες δοκιμές για κάθε αλγόριθμο, έτσι ώστε να έχουμε μια καλύτερη εκτίμηση της απόδοσής τους.

β)Να εκτελέσουμε τους αλγόριθμους σε διαφορετικά σετ δεδομένων για να δούμε πώς ανταποκρίνονται σε διαφορετικές συνθήκες.

γ) Να χρησιμοποιήσουμε διαφορετικές παραμέτρους που μπορείτε να ρυθμίσετε για κάθε αλγόριθμο και να βρούμε τις βέλτιστες.

ΚΩΔΙΚΑΣ

```
[ ]  # clone repository for deepsort with yolov4
     !git clone https://github.com/theAIGuysCode/yolov4-deepsort
```

```
[ ]  # step into the yolov4-deepsort folder
     %cd yolov4-deepsort/
```

```
# download yolov4 model weights to data folder
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.weights -P data/
```

```
# run DeepSort with YOLOv4 Object Detections as backbone (enable --info flag to see info about tracked objects)
!python object_tracker.py --video ./data/video/test.mp4 --output ./outputs/tracker.avi --model yolov4 --dont_show --info
```

```
[ ]  # define helper function to display videos
     import io
     from IPython.display import HTML
     from base64 import b64encode
     def show_video(file_name, width=640):
       # show resulting deepsort video
       mp4 = open(file_name,'rb').read()
       data_url = "data:video/mp4;base64," + b64encode(mp4).decode()
       return HTML("""
       <video width="{0}" controls>
             <source src="{1}" type="video/mp4">
       </video>
       """.format(width, data_url))
```

```
# convert resulting video from avi to mp4 file format
import os
path_video = os.path.join("outputs","tracker.avi")
%cd outputs/
!ffmpeg -y -loglevel panic -i tracker.avi output.mp4
%cd ..

# output object tracking video
path_output = os.path.join("outputs","output.mp4")
show_video(path_output, width=960)
```

```
]  %cd /content
   !git clone https://github.com/JonathonLuiten/TrackEval
```

```
[ ]  %cd TrackEval
     !wget https://omnomnom.vision.rwth-aachen.de/data/TrackEval/data.zip
```

```
! unzip data.zip
```

```
!python scripts/run_mot_challenge.py --BENCHMARK MOT17 --TRACKERS_TO_EVAL MPNTrack --METRICS CLEAR --USE_PARALLEL False --NUM_PARALLEL_CORES 1
```

Βιβλιογραφία

[1]Andreas M. Kaplan and Michael Haenlein, “Siri, Siri, in My Hand: Who’s the airest in the Land? On the Interpretations, Illustrations, and Implications o Artiicial Intelligence,” Business Horizons, 62/1 (January/ebruary 2019): 15-25.

[2]Ibid

[3]International Journal o Science and Research (IJSR) ISSN: 2319-7064 ResearchGate Impact actor (2018): 0.28 | SJI (2018): 7.426

[4]Wang, S., Manning, C.. 2013. ast dropout training. In: International Conerence on Machine Learning, pp. 118–126

[5]Ari M. Wani arooq Ahmad Bhat Sadu Azal Asi Iqbal Khan. 2020. Advances in Deep Learning. Studies in Big Data Volume 57 Series editor Janusz Kacprzyk, Polish Academy o Sciences, Warsaw, Poland https://doi.org/10.1007/978-981-13-6794-6

[6]Mead C. Neuromorphic electronic systems. Proc IEEE. 1990;78: 1629-1636

[7]Kyuma K, Lange E, Ohta J, Hermanns A, Banish B, Oita M. Artificial retinas — fast, versatile image processors. Nature. 1994;372:197-198

[8]Wang S, Wang C-Y, Wang P, et al. Networking retinomorphic sensor with memristive crossbar for brain-inspired visual perception. Natl Sci Rev. 2021;8(2):nwaa172

[9]Haykin S. Neural Networks and Learning Machines, Third Edit. Pearson, 2009.

[10]Falat L, Pancikova L. Quantitative Modelling in Economics with Advanced Artificial Neural Networks. Procedia Econ Financ. 2015; 34: 194-201.

[11]Sukthomya W, Tannock J. The training of neural networks to model manufacturing processes. J Intell Manuf. 2005; 16(1): 39-51.

[12]Moreno-Escobar JA, Gallegos-Funes FJ, Ponomaryov VI. Rank M-type radial basis functions network for medical image processing applications. in Image Processing: Algorithms and Systems V, 2007; 6497: 1-12.

[13]Fadzil MHA, Bakar HA. Human face recognition using neural networks. in Proceedings of 1st International Conference on Image Processing, 1994

[14]Palaz D, Magimai-Doss M, Collobert R. Convolutional Neural Networks-Based Continuous Speech Recognition Using Raw Speech Signal. in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015; 4295-4299.

[15]Sharkawy A-N, Aspragathos N. Human-Robot Collision Detection Based on Neural Networks. Int J Mech Eng Robot Res, 2018;. 7(2): 150-157

[16]Sharkawy A-N, Koustoumpardis PN, Aspragathos N. Manipulator Collision Detection and Collided Link Identification based on Neural Networks. in Advances in Service and Industrial Robotics RAAD 2018 Mechanisms and Machine Science, A. Nikos, K. Panagiotis, and M. Vassilis, Eds. Springer, Cham, 2018; pp. 3-12.

[17]Most T. Approximation of complex nonlinear functions by means of neural networks. in 2nd Weimar Optimization and Stochastic Days 2005; 2005

[18]Vemuri AT, Polycarpou MM. Neural-Network-Based Robust Fault Diagnosis in Robotic Systems. IEEE Trans Neural Networks 1997; 8(6): 1410-1420

[19]Steven Walczak, Narciso Cerpa. Artificial Neural Networks.

[20]M, Jena D, Zhang H. Two-dimensional semiconductors for transistors. Nat. Rev. Mater. 2016;1:15

[21]J. Brownlee. (2016) Overfitting and underfitting with machine learning algorithms. [Online]. Available: https://machinelearningmastery.com/ overfitting-and-underfitting-with-machine-learning-algorithms

[22]S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.

[23]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[24]D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” , vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[25]Jeong Y, S. Son, E. Jeong, and B. Lee, “An integrated self-diagnosis system for an autonomous vehicle based on an IoT gateway and deep learning,” Appl. Sci., vol. 8, no. 7, Article No. 1164, Jul. 2018.

[26]Chen M., Y. Cao, R. Wang, Y. Li, D. Wu, and Z. C. Liu, “Deepfocus: Deep encoding brainwaves and emotions with multi-scenario behavior analytics for human attention enhancement,” IEEE Netw., vol. 33, no. 6, pp. 70–77, Nov.–Dec. 2019.

[27]M. Haghighat and M. Abdel-Mottaleb, “Low resolution face recognition in surveillance systems using discriminant correlation analysis,” in Proc. 12th IEEE Int. Conf. Automatic Face & Gesture Recognition, Washington, USA, 2017, pp. 912–917.

[28]N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, San Diego, USA, 2005

[29]L. G. Clift, J. Lepley, H. Hagras, and A. F. Clark, “Autonomous computational intelligence-based behaviour recognition in security and surveillance,” in Proc. SPIE 10802, Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies II, Berlin, Germany, 2018, pp. 108020L

[30]P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” arXiv preprint arXiv: 1312.6229, 2013

[31]L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional siamese networks for object tracking,” in Proc. European Conf. Computer Vision, Amsterdam, The Netherlands, 2016, pp. 850–865

[32]T. Kong, A. B. Yao, Y. R. Chen, and F. C. Sun, “Hypernet: Towards accurate region proposal generation and joint object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 845–853

[33]A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1442–1468, Jul. 2014

[34]Y. Wu, J. Lim, and M. H. Yang, “Object tracking benchmark,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1834–1848, Sept. 2015.

[35]G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” Neurocomputing, vol. 381, pp. 61–88, Mar. 2020

[36]G. Fortino, W. Russo, C. Savaglio, W. M. Shen, and M. C. Zhou, “Agent-oriented cooperative smart objects: From IoT system design to implementation,” IEEE Trans. Syst. Man Cybern. Syst., vol. 48, no. 11, pp. 1939–1956, Nov. 2018

[37]X. Li, A. Dick, C. H. Shen, A. Van den Hengel, and H. Z. Wang, “Incremental learning of 3D-DCT compact representations for robust visual tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 4, pp. 863–881, Apr. 2013.

[38]D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 5, pp. 564–577, May 2003.

[39]M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, “Adaptive color attributes for real-time visual tracking,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 1090–1097.

[40]N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, San Diego, USA, 2005.

[41]F. Yang, H. Lu, W. Zhang, and G. Yang, “Visual tracking via bag of features,” IET Image Process., vol. 6, no. 2, pp. 115–128, Mar. 2012.

[42]S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[43]R. Yao, Q. F. Shi, C. H. Shen, Y. N. Zhang, and A. van den Hengel, “Part-based visual tracking with online latent structural learning,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Portland, USA, 2013, pp. 2363–2370.

[44]J. Gall, A. Yao, N. Razavi, L. van Gool, and V. Lempitsky, “Hough forests for object detection, tracking, and action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 11, pp. 2188–2202, Nov. 2011.

[45]J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “Prost: Parallel robust online simple tracking,” in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, San Francisco, USA, 2010, pp. 723–730

[46]Y. C. Bai and M. Tang, “Robust tracking via weakly supervised ranking SVM,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, USA, 2012, pp. 1854–1861.

[47]J. Kwon and K. M. Lee, “Tracking by sampling and integratingmultiple trackers,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1428–1441, Jul. 2014

[48]D. Wang, H. C. Lu, and M. H. Yang, “Online object tracking with sparse prototypes,” IEEE Trans. Image Process., vol. 22, no. 1, pp. 314–325, Jan. 2012

[49]K. Ullah, I. Ahmed, M. Ahmad, A. U. Rahman, M. Nawaz, and A. Adnan, “Rotation invariant person tracker using top view,” J. Ambient Intell. Humaniz. Comput., 2019. DOI: 10.1007/s12652-019-01526-5.

[50]I. Ahmed, M. Ahmad, M. Nawaz, K. Haseeb, S. Khan, and G. Jeon, “Efficient topview person detector using point based transformation and lookup table,” Comput. Commun., vol. 147, pp. 188–197, Nov. 2019.

[51]K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sept. 2015.

[52]R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 1440–1448

[53]S. Q. Ren, K. M. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. 28th Int. Conf. Neural Information Processing Systems, Montreal, Canada, 2015, pp. 91–99.

[54]T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. 13th European Conf. Computer Vision, Zurich, Switzerland, 2014, pp. 740–755.

[55]T. Y. Lin, P. Dollár, R. Girshick, K. M. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, USA, 2017, pp. 936–944.

[56]R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 580–587.

[57]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp.779–788.

[58]J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 6517–6525.

[59]J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv: 1804.02767, 2018.

[60]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. 14th European Conf. Computer Vision, Amsterdam, The Netherlands, 2016, pp. 21–37.

[61]S. Gidaris and N. Komodakis, “Object detection via a multi-region and semantic segmentation-aware CNN model,” in Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 1134–1142.

[62]J. F. Dai, Y. Li, K. M. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. 30th Int. Conf. Neural Information Processing Systems, Barcelona, Spain, 2016, pp. 379–387.

[63]N. Y. Wang, S. Y. Li, A. Gupta, and D. Y. Yeung, “Transferring rich feature hierarchies for robust visual tracking,” arXiv preprint arXiv: 1501.04587, 2015.

[64]G. H. Ning, Z. Zhang, C. Huang, X. B. Ren, H. H. Wang, C. H. Cai, and Z. H. He, “Spatially supervised recurrent convolutional neural networks for visual object tracking,” in Proc. IEEE Int. Symp. Circuits and Systems, Baltimore, USA, 2017, pp. 1–4.

[65]N. Y. Wang and D. Y. Yeung, “Ensemble-based tracking: Aggregating crowdsourced structured time series data,” in Proc. 31st Int. Conf. Machine Learning, Beijing, China, 2014, pp. 1107–1115.

[66]J. L. Fan, W. Xu, Y. Wu, and Y. H. Gong, “Human tracking using convolutional neural networks,” IEEE Trans. Neural Netw., vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[67]N. Y. Wang and D. Y. Yeung, “Learning a deep compact image representation for visual tracking,” in Proc. 26th Int. Conf. Neural Information Processing Systems, Lake Tahoe, USA, 2013, pp. 809–817.

[68]G. Zhu, F. Porikli, and H. D. Li, “Robust visual tracking with deep convolutional neural network based object proposals on pets,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, Las Vegas, USA, 2016, pp. 1265–1272.

[69]J. Kuen, K. M. Lim, and C. P. Lee, “Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle,” Pattern Recognit., vol. 48, no. 10, pp. 2964–2982, Oct. 2015.

[70]S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” in Proc. 32nd Int. Conf. Machine Learning, Lille, France, 2015, pp. 597–606.

[71]C. Migniot and F. Ababsa, “Hybrid 3D–2D human tracking in a top view,” J. Real-Time Image Process., vol. 11, no. 4, pp. 769–784, Dec. 2016.

[72]D. W. Du, Y. K. Qi, H. Y. Yu, Y. F. Yang, K. W. Duan, G. R. Li, W. G. Zhang, Q. M. Huang, and Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in Proc. 15th European Conf. Computer Vision, Munich, Germany, 2018, pp. 375–391.

[73]M. Ahmad, I. Ahmed, and A. Adnan, “Overhead view person detection using YOLO,” in Proc. IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conf., New York City, USA, 2019, pp. 627–633.

[74]I. Ahmed, S. Din, G. Jeon, and F. Piccialli, “Exploring deep learning models for overhead view multiple object detection,” IEEE Internet Things J., vol. 7, no. 7, pp. 5737–5744, Jul. 2020.

[75]H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via online boosting,” in Proc. British Machine Vision Conf., Edinburgh, UK, 2006, pp. 6.

[76]B. Babenko, M. H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, USA, 2009, pp. 983–990.

[77]J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in Proc. 12th European Conf. Computer Vision, Florence, Italy, 2012, pp. 702–715.

[78]Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[79]D. Held, S. Thrun, and S. Savarese, “Learning to track at 100 fps with deep regression networks,” in Proc. 14th European Conf. Computer Vision, Amsterdam, The Netherlands, 2016, pp. 749–765.

[80]X. Weng and W. Han. CyLKs: Unsupervised Cycle Lucas-Kanade Network for Landmark Tracking. arXiv:1811.11325, 2018. URL http://arxiv.org/abs/1811.11325

[81]S. Kayukawa and K. Kitani. BBeep: A Sonic Collision Avoidance System for Blind Travellers and Nearby Pedestrians. CHI, 2019.

[82]S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NIPS, 2015

[83]A. Geiger, P. Lenz, and R. Urtasun. AreWe Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. CVPR, 2012.

[84]S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna. Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking. ICRA, 2018

[85]Y. Xiang, A. Alahi, and S. Savarese, “Learning to Track: Online Multi- Object Tracking by Decision Making,” ICCV, 2015.

[86]R. Girshick, “Fast R-CNN,” ICCV, 2015

[87]T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint Detection and Identification Feature Learning for Person Search,” CVPR, 2017.

[88]B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, “TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model,” CVPR, 2020.

[89]S. Sun, N. Akhtar, X. Song, H. Song, A. Mian, and M. Shah, “Simultaneous Detection and Tracking with Motion Modelling for Multiple Object Tracking,” ECCV, 2020.

[90]Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019

[91]Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollr. Microsoft COCO: Common Objects in Context, 2014.

[92]Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. IEEE transactions on pattern analysis and machine intelligence, 40(3):611–625, 2017

[93]R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2013.

[94]J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," International Journal of Computer Vision, 2013. [Online]. Available: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

[95]R. Girshick, "Fast r-cnn," 2015.

[96]S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[97]K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017.

[98]T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017.

[99]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," Lecture Notes in Computer Science, p. 21–37,2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[100]D. Wang, H. C. Lu, and M. H. Yang, "Online object tracking with sparse prototypes," IEEE Trans. Image Process., vol. 22, no. 1, pp. 314–325, Jan. 2012

[101]K. Simonyan and A. Zisserman, "Very deep convolutional networks for largescale image recognition," 2015.

[102]C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," 2015.

[103]T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[104]M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 2020

[105]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified,real-time object detection," 2015

[106]T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” 2015.

[107]A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” 2020.

[108] https://el.wikipedia.org/wiki/Python

[109] https://el.wikipedia.org/wiki/Tensor_Flow

[110] https://www.tensorflow.org/tensorboard

[111] https://en.wikipedia.org/wiki/PyTorch

[112] https://colab.research.google.com/#scrollTo=lSrWNr3MuFUS

[113] https://arxiv.org/pdf/1811.00982.pdf

[114]https://openaccess.thecvf.com/content_CVPR_2020/papers/Tan_EfficientDet_Scalable_and_Efficient_Object_Detection_CVPR_2020_paper.pdf

[115] https://iq.opengenus.org/ssd-mobilenet-v1-architecture/

[116]https://ieeexplore.ieee.org/document/8839032

[117]https://www.researchgate.net/figure/The-architecture-of-YOLO-7_fig1_326535574

[118]https://towardsai.net/p/computer-vision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-and-demystified

[119]https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/

[120] https://arxiv.org/pdf/1703.06870.pdf