



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΓΝΩΡΙΣΗ ΚΑΙ ΑΠΟΦΥΓΗ ΟΔΙΚΩΝ ΕΜΠΟΔΙΩΝ ΜΕ ΧΡΗΣΗ ΕΙΚΟΝΑΣ ΚΑΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ



Του φοιτητή

Πόπη Νικόλαου

Αρ. Μητρώου: 134035

Επιβλέπων καθηγητής

Διαμαντάρας Κωνσταντίνος

Θεσσαλονίκη 2018

ΠΡΟΛΟΓΟΣ

Την σημερινή εποχή οι ηλεκτρονικοί υπολογιστές έχουν εισχωρήσει σε πολύ μεγάλο βαθμό στην καθημερινή ζωή των ανθρώπων. Σε αντίθεση με το παρελθόν, όπου η κατοχή υπολογιστή για τους ανθρώπους ήταν σπάνια, μπορούμε να αναφέρουμε πως είναι πλέον σπάνια η μη κατοχή ενός υπολογιστή. Η χρήση τους έχει γίνει εύχρηστη και αναγκαία, σε σημείο όπου υπάρχουν διαδικασίες στην καθημερινή ζωή που δεν διεκπαιρέωνονται χωρίς την χρήση υπολογιστή. Υπάρχει γενικά μια προσπάθεια αυτοματοποίησης λειτουργιών χρησιμοποιώντας ηλεκτρονικούς υπολογιστές που αποσκοπούν στην καλύτερη και πιο ποιοτική ζωή των ανθρώπων. Ένας από τους πιο φλέγων κλάδους αυτοματοποίησης λειτουργιών είναι ο κλάδος των αποκαλούμενων αυτόνομων οχημάτων. Όπως σε κάθε κλάδο αυτοματοποίησης έτσι και εδώ, πρέπει να ληφθούν πρώτα τα απαραίτητα μέτρα προστασίας, ώστε να μην τηθούν ζωές σε κίνδυνο. Η συγκεκριμένη πτυχιακή εργασία θα ασχοληθεί με το πώς ένα αυτόνομο όχημα μπορεί να εντοπίσει οδικά εμπόδια.

ΠΕΡΙΛΗΨΗ

Σε αυτή την πτυχιακή εργασία θα μιλήσουμε για την μηχανική μάθηση και πώς αυτή έχει βοηθήσει και συνεχίζει να βοηθάει στην επιστήμη των υπολογιστών. Ξεκινώντας θα αναφερθούμε στην αυτόνομη οδήγηση, στα οφέλη που έχει καθώς και στις κατηγορίες αυτόνομης οδήγησης που υπάρχουν. Έπειτα θα γίνει μια εισαγωγή στην μηχανική μάθηση, την διαδικασία που ακολουθείται, θα αναφέρουμε τα μοντέλα μηχανικής μάθησης που υπάρχουν και θα ασχοληθούμε με ένα από αυτά, το Deep Learning, το οποίο χρησιμοποιήθηκε για την εκπόνηση της συγκεκριμένης πτυχιακής. Θα συνεχίσουμε με την παράλληλη επεξεργασία, το πώς βοηθά στην επιτάχυνση των υπολογισμών των προγραμμάτων και θα αναφέρουμε τις δύο βασικές βιβλιοθήκες παράλληλης επεξεργασίας που χρησιμοποιήθηκαν, τις βιβλιοθήκες CUDA και Tensorflow. Στο τέταρτο κεφάλαιο θα γίνει περιγραφή του συνόλου δεδομένων που χρησιμοποιήθηκε και του συνελκτικού δικτύου SqueezeDet που χρησιμοποιήθηκε για τον εντοπισμό αντικειμένων σε δεδομένα εικόνων. Εκτός από το SqueezeDet θα γίνει και μια αναφορά σε εναλλακτικές μεθόδους που αποσκοπούν στον εντοπισμό αντικειμένων σε εικόνες. Τέλος θα δούμε τα αποτελέσματα που λήφθηκαν από την εκτέλεση της πτυχιακής και θα αναφερθούν τα συμπεράσματα που λήφθηκαν κατά την διάρκεια εκπόνησής της.

ABSTRACT

In this thesis we will talk about machine learning and how it has helped and continues to help in computer science. At first we will refer to autonomous driving, the benefits that it has and the levels of autonomous driving. Then there will be an introduction to machine learning, the following process, we will mention the categories of machine learning and we will deal with one of them, the Deep Learning one, which was used for the development of this thesis. We will go on with parallel processing, how it helps in speeding up calculations and we will mention the two parallel processing libraries that were used, CUDA and Tensorflow. The fourth chapter will describe the data set that was used, and the convolutional neural network SqueezeDet, which detects objects in image data. Finally, we will look at the results obtained from the completion of this thesis and the conclusions drawn during it's preparation.

ΕΥΧΑΡΙΣΤΙΕΣ

Για την υλοποίηση της συγκεκριμένης πτυχιακής εργασίας θα ήθελα να ευχαριστήσω τους καθηγητές της σχολής μου και ιδιαίτερα τον επιβλέποντα καθηγητή μου κύριο Κωνσταντίνο Διαμαντάρα ο οποίος μου έδωσε την ευκαιρία μέσω της συγκεκριμένης πτυχιακής αλλά και πρακτικής άσκησης να διευρύνω τους ορίζοντές μου αλλά και να έρθω σε επαφή με τεχνολογίες και γνωστικά αντικείμενα τα οποία είναι ανωτέρου επιπέδου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΠΕΡΙΕΧΟΜΕΝΑ	6
Ευρετήριο σχημάτων	7
Ευρετήριο πινάκων.....	8
ΕΙΣΑΓΩΓΗ.....	9
ΚΕΦΑΛΑΙΟ 1.....	10
ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΕΝΤΟΠΙΣΜΟΥ ΟΔΙΚΩΝ ΕΜΠΟΔΙΩΝ ΣΤΗΝ ΑΥΤΟΝΟΜΗ ΟΔΗΓΗΣΗ.....	10
ΕΙΣΑΓΩΓΗ.....	10
1.1 ΑΥΤΟΝΟΜΗ ΟΔΗΓΗΣΗ.....	11
1.2 ΜΕΘΟΔΟΙ.....	12
1.3 ΕΠΙΠΕΔΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗΣ ΟΔΗΓΗΣΗΣ	13
1.4 ΠΡΟΒΛΗΜΑ	15
1.5 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	16
ΕΠΙΛΟΓΟΣ.....	17
ΚΕΦΑΛΑΙΟ 2.....	18
ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	18
ΕΙΣΑΓΩΓΗ.....	18
2.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	18
2.2 ΔΙΑΔΙΚΑΣΙΑ ΜΑΘΗΣΗΣ	21
2.3 ΜΟΝΤΕΛΑ ΜΑΘΗΣΗΣ	22
2.4 DEEP LEARNING / ΒΑΘΙΑ ΜΑΘΗΣΗ.....	25
ΕΠΙΛΟΓΟΣ.....	29
ΚΕΦΑΛΑΙΟ 3.....	30
ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ.....	30
ΕΙΣΑΓΩΓΗ.....	30
3.1 Η ΠΛΑΤΦΟΡΜΑ CUDA	31
3.2 Η ΒΙΒΛΙΟΘΗΚΗ TENSORFLOW	33
ΕΠΙΛΟΓΟΣ.....	36
ΚΕΦΑΛΑΙΟ 4.....	37

ΠΕΡΙΓΡΑΦΗ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ ΚΙΤΤΙ ΚΑΙ ΣΥΝΕΛΙΚΤΙΚΟΥ ΔΙΚΤΥΟΥ SQUEEZEDET.....	37
ΕΙΣΑΓΩΓΗ.....	37
4.1 ΤΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ ΚΙΤΤΙ.....	38
4.2 ΤΟ ΣΥΝΕΛΙΚΤΙΚΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ SQUEEZEDET	40
4.3 ΕΝΑΛΛΑΚΤΙΚΑ ΜΟΝΤΕΛΑ ΕΝΤΟΠΙΣΜΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ.....	44
4.4 ΕΚΠΑΙΔΕΥΣΗ ΤΟΥ ΔΙΚΤΥΟΥ	47
ΕΠΙΛΟΓΟΣ.....	55
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	56
ΑΝΑΦΟΡΕΣ	58

Ευρετήριο σχημάτων

Σχήμα 1 "Αναπαράσταση μεθόδου Cross-Validation"	23
Σχήμα 2 "Αναπαράσταση νευρωνικού δικτύου τριών στρωμάτων".....	28
Σχήμα 3 "Αναπαράσταση λειτουργίας παραδείγματος του CIFAR"	29
Σχήμα 4 "Αναπαράσταση κώδικα γλώσσας C με και χωρίς χρήση της πλατφόρμας CUDA"	32
Σχήμα 5 "Αναπαράσταση ροής επεξεργασίας με χρήση της πλατφόρμας CUDA".....	33
Σχήμα 6 "Αναπαράσταση ιεραρχίας εργαλείων της βιβλιοθήκης Tensorflow".....	34
Σχήμα 7 "Αναπαράσταση του ρόλου Πελάτης".....	35
Σχήμα 8 "Αναπαράσταση του ρόλου Ιδιοκτήτης Εργασιών".....	35
Σχήμα 9 "Αναπαράσταση του ρόλου Εργάτης υπηρεσιών".....	36
Σχήμα 10 "Η κινητή πλατφόρμα συλλογής δεδομένων του συνόλου δεδομένων ΚΙΤΤΙ".....	39
Σχήμα 11 "Αναπαράσταση διαδικασίας ανίχνευσης αντικειμένων του συνελικτικού δικτύου SqueezeDet".....	42
Σχήμα 12 "Αναπαράσταση του Fire module".....	43
Σχήμα 13 "Αναπαράσταση της αρχιτεκτονικής του δικτύου SqueezeNet".....	44
Σχήμα 14 "Έξοδος αποτελεσμάτων κατά την διαδικασία εκπαίδευσης του SqueezeDet".....	49
Σχήμα 15 "Έξοδος αποτελεσμάτων κατά την εκτέλεση του αρχείου αξιολόγησης".....	50
Σχήμα 16 "Αναπαράσταση υλοποίησης κώδικα Python συνελικτικού στρώματος και RELU".....	50

Σχήμα 17 "Αναπαράσταση υλοποίησης κώδικα Python συγκεντρωτικού στρώματος ".....	51
Σχήμα 18 "Αναπαράσταση υλοποίησης κώδικα Python του Fire module".....	51
Σχήμα 19 "Αναπαράσταση υλοποίησης κώδικα Python της αρχιτεκτονικής του δικτύου SqueezeDet".....	52
Σχήμα 20 "Αναπαράσταση υλοποίησης κώδικα Python μεταβλητών (Placeholders) εισόδου".	53
Σχήμα 21 "Αναπαράσταση υλοποίησης κώδικα Python συνεδρίας του Tensorflow".....	53
Σχήμα 22 "Παράδειγμα νο.1 εικόνας πριν και μετά τον εντοπισμό αντικειμένων ".....	54
Σχήμα 23 "Παράδειγμα νο.2 εικόνας πριν και μετά τον εντοπισμό αντικειμένων".....	54
Σχήμα 24 "Παράδειγμα νο.3 εικόνας πριν και μετά τον εντοπισμό αντικειμένων".....	55
Σχήμα 25 "Παράδειγμα νο.4 εικόνας πριν και μετά τον εντοπισμό αντικειμένων".....	55

Ευρετήριο πινάκων

Πίνακας 1 "Επίπεδα αυτονομίας SAE".....	15.
---	-----

ΕΙΣΑΓΩΓΗ

Με την πάροδο των χρόνων η επιστήμη των υπολογιστών έχει εξελιχθεί με πολύ γρήγορους και μεγάλους ρυθμούς. Οι νέες τεχνολογικές ανακαλύψεις μας επέτρεψαν να έχουμε υπολογιστικές μηχανές όχι μόνο στα σπίτια μας, αλλά στα κινητά, στα αυτοκίνητα και σχεδόν σε οτιδήποτε υπάρχει γύρω μας. Καθώς οι τεχνολογίες που υπάρχουν μας επιτρέπουν να λύνουμε προβλήματα και να εκτελούμε διαδικασίες με μεγαλύτερη ευκολία, δημιουργούνται καινούργιες ιδέες που αποσκοπούν στο να κάνουν τη ζωή μας καλύτερη. Με την εισαγωγή της μηχανικής μάθησης στον τομέα της επιστήμης των υπολογιστών ολοένα και περισσότερες λειτουργίες τείνουν να αυτοματοποιούνται ώστε ο άνθρωπος να κάνει λιγότερες λειτουργίες. Ένας τέτοιος μεγάλος κλάδος, ο οποίος είναι και πάρα πολύ φλέγων την τελευταία δεκαετία, είναι η αυτόνομη οδήγηση.

Η αυτόνομη οδήγηση παρόλο που είναι ένας από τους πιο φλέγων κλάδους των τελευταίων ετών έχει παρουσιαστεί σαν έννοια πολλά χρόνια νωρίτερα. Με την πάροδο των χρόνων και τις αυξανόμενες κατά τα χρόνια δυνατότητες που αποκτούν οι υπολογιστικές μηχανές, η αυτόνομη οδήγηση μεταβαίνει από το ένα στάδιο στο επόμενο, κάνοντας μικρά βήματα κάθε φορά με στόχο την πλήρη αυτοματοποίηση, δηλαδή ένα όχημα 100% αυτόνομο που θα είναι σε θέση να εκτελεί όλες τις επιθυμητές λειτουργίες χωρίς την παραμικρή ανθρώπινη επέμβαση.

Το ερώτημα όμως είναι γιατί είναι σημαντική η αυτόνομη οδήγηση και επιθυμούν πολλοί να ασχοληθούν με αυτή; Η απάντηση είναι ότι η αυτόνομη οδήγηση έχει πολλά οφέλη σε αρκετούς τομείς. Μερικά από αυτά είναι :

- Ασφάλεια. Μελέτες έχουν δείξει ότι το 90% των τροχαίων ατυχημάτων οφείλεται στους οδηγούς. Τα αυτόνομα οχήματα θα μειώσουν σε μεγάλο βαθμό το ποσοστό αυτό, με ιδανικό στόχο να μην προκαλούνται καθόλου τροχαία ατυχήματα από λάθη των οδηγών.
- Ταχύτητα. Με τα αυτόνομα οχήματα θα μειωθούν κατά πολύ οι αποστάσεις των εν κινήσει οχημάτων μειώνοντας ως αποτέλεσμα τις κυκλοφοριακές συμφορές.
- Ιδιοκτησία οχημάτων. Για μεγάλο ποσοστό των ανθρώπων, ο χρόνος που δαπανάται οδηγώντας είναι πολύ μικρός. Τον περισσότερο χρόνο τα οχήματα παραμένουν σταυθμεσμένα. Αυτός είναι ο λόγος για τον οποίο υπηρεσίες όπως το Uber και το Lyft είναι τόσο δημοφιλείς. Φανταστείτε τις συγκεκριμένες υπηρεσίες αλλά με αυτόνομα οχήματα. Αυτό θα μειώσει το κόστος, θα βελτιώσει την ασφάλεια των επιβατών και θα μειώσει την ανάγκη για προσωπική ιδιοκτησία αυτοκινήτων.

- Ευκολότερο παρκάρισμα. Τα αυτόνομα οχήματα θα μπορούσαν να αλλάξουν τη συμπεριφορά κινητικότητας των καταναλωτών, μειώνοντας έτσι την ανάγκη για χώρο κατά πολλά τετραγωνικά μέτρα. Για παράδειγμα, τα συστήματα αυτόματης στάθμευσης αυτοκινήτων δεν απαιτούν χώρο ανοιχτής πόρτας για την είσοδο και έξοδο των επιβατών όταν σταθμεύουν, μειώνοντας έτσι το χώρο που καταλαμβάνεται από τον ίδιο αριθμό οχημάτων.

Οι παραπάνω λόγοι έχουν οδηγήσει αρκετές βιομηχανίες οχημάτων στην ενασχόληση με την αυτόνομη οδήγηση. Τέτοιες βιομηχανίες είναι οι CMU Navlab, Google Cars, Toyota, Renault, Audi, Volvo, Mercedes-Benz, General Motors, Nissan, Bosch, PSA Peugeot Citroen και ίσως η πιο γνώστη βιομηχανία για τα αυτόνομα αυτοκίνητά της, η Tesla. Ακόμα και το σύστημα αυτόνομης οδήγησης της Tesla μπορεί να χρησιμοποιηθεί σε μεγάλους αυτοκινητόδρομους και μόνο για τον εντοπισμό άλλων οχημάτων.

Είναι λοιπόν κατανοητό ότι στην παρούσα φάση υπάρχουν αυτόνομα οχήματα αλλά η παρουσία του ανθρώπινου παράγοντα είναι αναγκαία. Θα είναι άραγε τα αυτόνομα οχήματα πλήρως αυτόνομα μέσα στην επόμενη δεκαετία; Η απαντησή είναι άγνωστη. Ελπίζουμε με την συγκεκριμένη πτυχιακή εργασία να κατανοήσουμε καλύτερα το τι απαιτείται ώστε να έρθουμε πιο κοντά στην ύπαρξη αυτόνομων οχημάτων.

ΚΕΦΑΛΑΙΟ 1

ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΕΝΤΟΠΙΣΜΟΥ ΟΔΙΚΩΝ ΕΜΠΟΔΙΩΝ ΣΤΗΝ ΑΥΤΟΝΟΜΗ ΟΔΗΓΗΣΗ

ΕΙΣΑΓΩΓΗ

Οδικό εμπόδιο θεωρείται οποιοδήποτε αντικείμενο που υπάρχει σε ένα δρόμο το οποίο μπορεί να αναγκάσει τον οδηγό ενός οχήματος σε οποιαδήποτε ενέργεια που αποσκοπεί στην αποφυγή σύγκρουσης του οχήματος με το αντικείμενο. Οδικά εμπόδια μπορεί να είναι αντικείμενα που βρέθηκαν τυχαία σε ένα δρόμο, άλλα οχήματα, ακόμα και άνθρωποι. Σε κάθε περίπτωση η αποφυγή τους είναι αναγκαία διαφορετικά μπορούν να υπάρξουν σοβαρά ατυχήματα που μπορούν να κοστίσουν υλικές ζημιές αλλά και ανθρώπινες ζωές. Ο άνθρωπος εκ φύσεως εντοπίζει αυτόματα οτιδήποτε μπορεί να υπάρχει στον δρόμο και προβαίνει στις ανάλογες ενέργειες. Τι συμβαίνει όμως στην περίπτωση των αυτόνομων αυτοκινήτων; Σε αυτό το κεφάλαιο θα δούμε τι είναι τα αυτόνομα οχήματα και η αυτόνομη οδήγηση, τις κατηγορίες αυτόνομης οδήγησης και θα δούμε το πρόβλημα που έχουν να αντιμετωπίσουν τα αυτόνομα αυτοκίνητα, τον εντοπισμό των οδικών εμποδίων.

1.1 ΑΥΤΟΝΟΜΗ ΟΔΗΓΗΣΗ

Αυτόνομη οδήγηση είναι ο τομέας της μηχανικής μάθησης που ασχολείται με την αυτοματοποίηση της διαδικασίας της οδήγησης. Ο όρος χρήζει ανάλυση καθώς υπάρχουν πολλά στάδια και καταστάσεις στα οποία μπορεί κάποιος να θεωρήσει ότι υπάρχει αυτόνομη οδήγηση. Για παράδειγμα, το αυτόματο κιβώτιο ταχυτήτων είναι μία μορφή αυτονομίας που υπάρχει εδώ και αρκετά χρόνια, όμως αφορά μία μεμονωμένη λειτουργία του αυτοκινήτου. Στην σημερινή εποχή, και αυτό το οποίο θέλουν όλοι να επιτύχουν είναι να αυτοματοποιήσουν πλήρως τη διαδικασία της οδήγησης και όχι μερικά κομμάτια αυτής.

Τα δυνητικά οφέλη των αυτόνομων αυτοκινήτων περιλαμβάνουν μειωμένη κινητικότητα και κόστος υποδομής, αυξημένη ασφάλεια, αυξημένη κινητικότητα, αυξημένη ικανοποίηση του πελάτη και μειωμένη πιθανότητα συγκρούσεων. Συγκεκριμένα, από την μείωση των συγκρούσεων κυκλοφορίας και των τραυματισμών, μειώνονται και τα κόστη που μπορεί να προκύπτουν, συμπεριλαμβανομένης της μικρότερης ανάγκης για ασφάλιση. Η αυτόνομη οδήγηση μπορεί επίσης να βοηθήσει στην κυκλοφοριακή ροή των δρόμων, καθώς και άτομα τα οποία δεν έχουν τη δυνατότητα να οδηγήσουν από μόνοι τους κάποιο όχημα, όπως για παράδειγμα παιδιά, ηλικιωμένοι, άτομα με αναπηρίες.

1.2 ΜΕΘΟΔΟΙ

Τα αυτόνομα αυτοκίνητα προκειμένου να οδηγηθούν μόνα τους πρέπει να κάνουν μια πολύ σημαντική ενέργεια την οποία ίδια ενέργεια κάνει και ο άνθρωπος, και πρέπει να γίνεται συνεχώς. Αυτή η ενέργεια είναι η παρατήρηση. Πρέπει να βλέπει συνεχώς τον δρόμο που έχει μπροστά του, να βλέπει τα πιθανά εμπόδια που υπάρχουν, καθώς και τις αλλαγές που μπορούν να συμβούν σε ένα σενάριο οδήγησης. Πρέπει δηλαδή να συλλέγει δεδομένα. Και ο άνθρωπος συλλέγει δεδομένα και το επιτυγχάνει μέσω της όρασης και της ακοής. Τα αυτοκίνητα συλλέγουν τα δεδομένα με χρήση διαφόρων αισθητήρων. Οι αισθητήρες που χρησιμοποιούνται μπορεί να είναι κάμερες, ραντάρ, λέιζερ φως (LiDAR), παγκόσμια συστήματα τοποθεσίας (GPS), συστήματα μέτρησης οδομετρίας.

Τα δεδομένα που συλλέγονται από τους αισθητήρες χρησιμοποιούνται είτε ξεχωριστά, είτε συνδυάζονται από προηγμένα συστήματα έλεγχου για τον εντοπισμό των κατάλληλων διαδρομών πλοήγησης, καθώς και τον εντοπισμό οδικών εμποδίων και της οδικής σήμανσης. Τα προηγμένα συστήματα έλεγχου, προκειμένου να λειτουργήσουν σωστά, πέρα από τον κλασικό προγραμματισμό πρέπει να κάνουν χρήση μεθόδων μηχανικής μάθησης, καθώς με τον κλασικό προγραμματισμό είναι αδύνατη η εκ των πρότερων γνώση όλων των πιθανών οδικών σεναρίων.

1.3 ΕΠΙΠΕΔΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗΣ ΟΔΗΓΗΣΗΣ

Όπως αναφέρθηκε και νωρίτερα, υπάρχουν διάφορα επίπεδα στα οποία μπορεί κάποιος να θεωρήσει ότι υπάρχει αυτόνομη οδήγηση. Το 2014, δημοσιεύτηκε από την SAE(Society of Automotive Engineers) International ένα σύστημα ταξινόμησης των επιπέδων αυτοματοποιημένης οδήγησης. Αυτό το σύστημα ταξινόμησης βασίζεται στο ποσό της απαιτούμενης παρέμβασης και προσοχής που πρέπει να καταβάλει ο οδηγός του οχήματος. Δεν λαμβάνει υπ όψιν τις δυνατότητες που έχει το κάθε όχημα, παρόλο που μπορεί να διαφέρουν από όχημα σε όχημα.

Με τον ορό κατάσταση οδήγησης σημαίνει ένα σενάριο οδήγησης με χαρακτηριστικές απαιτήσεις δυναμικής οδήγησης. Μερικά παραδείγματα είναι

κυκλοφοριακή συμφόρηση, συγχώνευση ταχείας κυκλοφορίας, μετάβαση από υψηλή σε χαμηλή ταχύτητα. Τα επίπεδα κατηγοριοποίησης είναι τα εξής:

- Επίπεδο 0: Στο πρώτο επίπεδο, το όχημα μπορεί να προειδοποιεί τον οδηγό του οχήματος ακόμα και να παρέμβει προσωρινά σε κάποια κατάσταση που κρίνει αναγκαία, δεν έχει όμως σταθερό έλεγχο του οχήματος.
- Επίπεδο 1: Σε αυτή τη κατάσταση ο οδηγός και το όχημα μοιράζονται τον έλεγχο του οχήματος, με την απαίτηση ο οδηγός να είναι πάντα σε ετοιμότητα να αναλάβει τον πλήρη έλεγχό του. Ένα παράδειγμα είναι το Adaptive Cruise Control (ACC), όπου ο οδηγός ελέγχει το τιμόνι και το αυτοματοποιημένο σύστημα ρυθμίζει την ταχύτητα του οχήματος.
- Επίπεδο 2: Το αυτοματοποιημένο σύστημα σε αυτή τη περίπτωση ελέγχει πλήρως το όχημα (επιτάχυνση, πέδηση και σύστημα διεύθυνσης). Ο οδηγός πρέπει να παρακολουθεί την οδήγηση και να είναι έτοιμος να παρέμβει αμέσως ανά πάσα στιγμή, εάν το αυτοματοποιημένο σύστημα δεν ανταποκριθεί σωστά.
- Επίπεδο 3: Στο τρίτο επίπεδο ο οδηγός μπορεί με ασφάλεια να στρέψει την προσοχή του μακριά από τα καθήκοντα οδήγησης. Το όχημα θα χειριστεί καταστάσεις που απαιτούν άμεση ανταπόκριση, όπως πέδηση έκτακτης ανάγκης. Και πάλι βέβαια ο οδηγός πρέπει να είναι έτοιμος να παρέμβει, όμως σε αυτή τη περίπτωση εντός ενός ορισμένου χρονικού διαστήματος, το οποίο καθορίζεται από τον κατασκευαστή, όταν προειδοποιηθεί από το όχημα.
- Επίπεδο 4: Παρόμοιο με το τρίτο επίπεδο, αλλά σε αυτή την κατάσταση δεν απαιτείται ποτέ προσοχή από τον οδηγό για την ασφάλεια, δηλαδή ο οδηγός μπορεί να πάει με ασφάλεια ή να βγει από το κάθισμα του οδηγού. Η αυτοκίνηση υποστηρίζεται μόνο σε περιορισμένους χώρους (γεωγραφικά) ή υπό ειδικές συνθήκες, όπως κυκλοφοριακή συμφόρηση. Εκτός αυτών των περιοχών ή περιστάσεων, το όχημα πρέπει να είναι σε θέση να διακόψει με ασφάλεια το ταξίδι, δηλαδή να σταθμεύσει το αυτοκίνητο, αν ο οδηγός δεν αναλάβει τον έλεγχο του αυτοκινήτου.
- Επίπεδο 5: Στο πέμπτο και τελευταίο επίπεδο η παρέμβαση του ανθρώπινου παράγοντα δεν απαιτείται σε κανένα επίπεδο. Θα μπορούσε να παρομοιαστεί με ένα ταξί το οποίο έχει σαν οδηγό το αυτοκίνητο και σαν επιβάτη τον ιδιοκτήτη του αυτοκινήτου.

Πτυχιακή εργασία του φοιτητή Πόπη Νικόλαου

Πίνακας 1 "Επίπεδα αυτονομίας SAE"

SAE Level	Name	Narrative definition		Execution of steering and acceleration/ deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)
Human driver monitors the driving environment							
0	No Automation	The full-time performance by the human driver of all aspects of the dynamic driving task, even when "enhanced by warning or intervention systems"		Human driver	Human driver	Human driver	n/a
1	Drive Assistance	The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration"	using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task	Human driver and system		Some driving modes	
2	Partial Automation	The driving mode-specific execution by one or more driver assistance systems of <i>both steering and acceleration/deceleration</i>		System			
Automated driving system monitors the driving environment							
3	Conditional Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task	with the expectation that the <i>human driver will respond appropriately to a request to intervene even if a human driver does not respond appropriately to a request to intervene under all roadway and environmental conditions</i> that can be managed by a human driver	System	System	Human driver	Some driving modes
4	High Automation					System	Many driving modes
5	Full Automation					System	All driving modes

1.4 ΠΡΟΒΛΗΜΑ

Αναφερθήκαμε προηγουμένως στην αυτόνομη οδήγηση καθώς και στις κατηγορίες αυτόνομης οδήγησης που υπάρχουν. Όλα αυτά φυσικά για να καταλήξουμε στο πρόβλημα που έχουμε να αντιμετωπίσουμε. Η πτυχιακή αυτή μελέτησει έναν από τους πολλούς τρόπους μάθησης που χρησιμοποιείται από το όχημα ώστε αυτό να είναι αυτόνομο. Πιο συγκεκριμένα, θέλουμε με την χρήση καμερών σαν αισθητήρες δεδομένων, να συλλέγουμε και να επεξεργαζόμαστε δεδομένα, ώστε το όχημα να είναι σε θέση να εντοπίσει οδικά εμπόδια που μπορεί να υπάρχουν στον δρόμο. Το να μπορέσουμε να το κάνουμε να τα αποφεύγει κιόλας η να προειδοποιεί τον οδηγό είναι μια πολύ μεγάλη διαδικασία με την οποία δεν θα ασχοληθούμε. Αυτό που θέλουμε είναι να μπορούμε σε δεδομένα εικόνων να εντοπίζουμε τι οδικά εμπόδια υπάρχουν καθώς και τη θέση τους. Θα εμβαθύνουμε με λεπτομερές στο 4^ο κεφάλαιο για το πως μπορούμε να επιτύχουμε κάτι τέτοιο.

1.5 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Το πρόβλημα του εντοπισμού αντικειμένων σε δεδομένα εικόνων δεν είναι κάτι πρόσφατο. Πολλές ομάδες επιστημόνων εδώ και χρόνια αναπτύσσουν και ελέγχουν μοντέλα τα οποία είναι σε θέση να εντοπίζουν αντικείμενα αρκετά επιτυχημένα. Τα τελευταία χρόνια όμως πολλές βιοχημανίες οχημάτων προσπαθούν να αυτοματοποιήσουν την διαδικασία της οδήγησης. Λόγο ανταγωνισμού όμως η κάθε βιοχημανία αναπτύσει το δικό της σύστημα το οποίο βέβαια δεν δημοσιεύει. Εταιρίες όπως Volvo, Audi και άλλες προσπαθούν να πετύχουν ότι έχει καταφέρει η Tesla εδώ και καιρό, προκειμένου να κερδίσουν νέους πελάτες. Παρόλα αυτά, ομάδες επιστημόνων συνεχίζουν να αναπτύσσουν δικά τους μοντέλα, κάποια από τα οποία δημοσιεύονται. Είναι φανερό λοιπόν πως το πεδίο του εντοπισμού και της αναγνώρισης αντικειμένων απασχολεί και θα συνεχίσει να απασχολεί τόσο την επιστημονική, όσο και την βιομηχανική κοινότητα.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο μιλήσαμε για την αυτόνομη οδήγηση. Αναφερθήκαμε στα οφέλη που έχει στις ζωές των ανθρώπων, αναλύσαμε τα παγκόσμια επίπεδα ταξινόμησης της και μιλήσαμε για τον τρόπο με τον οποίο συλλέγονται οι πληροφορίες από τα αυτοκίνητα. Αναφέραμε επίσης ότι η μηχανική μάθηση είναι αναγκαία προκειμένου η αυτόνομη οδήγηση να μπορέσει να πραγματοποιηθεί. Στο επόμενο κεφάλαιο θα μιλήσουμε σχετικά με την μηχανική μάθηση, ώστε να καταλάβουμε πως αυτή είναι το κλειδί στο να δημιουργήσουμε αυτόνομα αυτοκίνητα.

ΚΕΦΑΛΑΙΟ 2

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΕΙΣΑΓΩΓΗ

Μηχανική Μάθηση είναι ο κλάδος της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Η μηχανική μάθηση, όπως και κάθε άλλη λειτουργία ενός υπολογιστή, χρησιμοποιεί έναν αλγόριθμο ο οποίος είναι υπεύθυνος για την βελτίωση μιας ήδη υπάρχουσας λειτουργίας του συστήματος. Φυσικά ο αλγόριθμος που είναι υπεύθυνος για αυτή τη λειτουργία χρειάζεται δεδομένα προκειμένου να επιτύχει τον σκοπό του.

2.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η μηχανική μάθηση (machine learning) είναι μια περιοχή της τεχνητής νοημοσύνης η οποία αφορά αλγορίθμους και μεθόδους που επιτρέπουν στους υπολογιστές να " μαθαίνουν ". Η συγκεκριμένη φράση σημαίνει πως οι υπολογιστές έχουν την δυνατότητα να βελτιώνουν την λειτουργία τους με την πάροδο του χρόνου. Με τη μηχανική μάθηση καθίσταται εφικτή η κατασκευή προσαρμοσμένων προγραμμάτων τα οποία λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι την εκ των προτέρων γνώση των μηχανικών που τα προγραμμάτισαν. Η μηχανική μάθηση επικαλύπτεται σημαντικά με τη στατιστική, αφού και τα δύο πεδία μελετούν την ανάλυση δεδομένων. Είναι μία περιοχή της τεχνητής νοημοσύνης που ασχολείται με την ανάπτυξη αλγορίθμων που λαμβάνουν ως είσοδο εμπειρικά δεδομένα όπως μετρήσεις από αισθητήρες ή βάσεις δεδομένων. Ο αλγόριθμος έχει σχεδιαστεί για να εντοπίζει πολύπλοκες σχέσεις που υποτίθεται ότι είναι χαρακτηριστικά του παρόντος μηχανισμού που δημιούργησε τα δεδομένα και να χρησιμοποιεί αυτά τα πρότυπα για να κάνει προβλέψεις πάνω σε καινούρια. Τα δεδομένα μπορούν να θεωρηθούν ως στιγμιότυπα των πιθανών σχέσεων μεταξύ των παρατηρούμενων μεταβλητών. Ο αλγόριθμος ενεργεί σαν ένας μηχανισμός μάθησης που μελετά ένα τμήμα των παρατηρούμενων δεδομένων για να συλλάβει τα χαρακτηριστικά ενδιαφέροντος του αγνώστου δείγματος και να τα χρησιμοποιήσει ως γνώσεις που έχει αποκτήσει έτσι ώστε να λάβει έξυπνες αποφάσεις που βασίζονται σε νέα δεδομένα εισόδου, βελτιώνοντας με αυτόν τον τρόπο την λειτουργία του.

Το 1959, ο Arthur Samuel όρισε τη μηχανική μάθηση ως " πεδίο μελέτης που δίνει στους υπολογιστές τη δυνατότητα να μαθαίνουν χωρίς να είναι ρητά

προγραμματισμένοι ”. Ο Tom M. Mitchell παρείχε, πιο επίσημο ορισμό: “ Ένα πρόγραμμα υπολογιστή πρόκειται να μάθει από την εμπειρία E σε σχέση με κάποια κατηγορία εργασιών T και την μέτρηση της επίδοσης P , αν η P απόδοσή του σε εργασίες T βελτιώνεται με την εμπειρία E ”.

Το πρώτο στάδιο της σχεδίασης ενός συστήματος μηχανικής μάθησης συνίσταται στον προσδιορισμό της γνώσης που θα χρησιμοποιηθεί κατά την εκπαίδευσή του. Θα πρέπει να σημειωθεί πως οι όποιες επιλογές ακολουθηθούν, θα έχουν άμεση επίδραση στην απόδοση του συστήματος. Μια από τις σημαντικότερες επιλογές που μπορούν να γίνουν ως προς το είδος της παρεχόμενης γνώσης είναι το κατά πόσο προσφέρει η τελευταία στο σύστημα μηχανικής μάθησης άμεση πληροφόρηση ως προς την υπό εκμάθηση έννοια (concept), η έμμεση, αναγκάζοντάς το να προβεί το ίδιο στην ανακάλυψή της, εκ του αποτελέσματος του σταδίου της εκπαίδευσης. Επίσης, ιδιαίτερη σημασία στην ποιότητα του σώματος εκπαίδευσης αποδίδεται στην κατανομή των παραδειγμάτων από τα οποία θα αντληθεί η γνώση από το σύστημα μηχανικής μάθησης, καθώς θα πρέπει να είναι όμοια με την αντίστοιχη κατανομή των περιπτώσεων που θα χρησιμοποιηθούν κατά τον έλεγχο της απόδοσης, ακόμα δε περισσότερο κατά τη λειτουργία του. Μια από τις πλέον συνήθεις μορφές αναπαράστασης των παραδειγμάτων είναι η διανυσματική: Κάθε παράδειγμα εκπαίδευσης, ή στιγμιότυπο (instance) όπως έχει επικρατήσει να αποκαλείται κυρίως στα προβλήματα ταξινόμησης, αποτελείται από ένα σύνολο χαρακτηριστικών (attributes ή features), μετρήσιμων δηλαδή ποσοτήτων που έχουν επιλεγεί με τέτοιο τρόπο από τα δεδομένα εκπαίδευσης, ώστε το σύστημα μηχανικής μάθησης να μπορέσει να εκμαιεύσει από τις τιμές τους τη γνώση που χρειάζεται. Ανάλογα με το είδος της πληροφορίας που κωδικοποιεί ένα χαρακτηριστικό, αυτό μπορεί να είναι συνεχές (continuous), για παράδειγμα ένας πραγματικός ή ακέραιος αριθμός, ή ονομαστικό (nominal), το οποίο λαμβάνει ένα προκαθορισμένο σύνολο διακριτών τιμών, αριθμητικών ή συμβολικών. Με αυτόν τον τρόπο, έχοντας επιλέξει N χαρακτηριστικά, απεικονίζουμε το χώρο του προβλήματος μας σε έναν N -διάστατο χώρο, το χώρο των στιγμιότυπων (instance space), αντιστοιχώντας κάθε στιγμιότυπο εκπαίδευσης σε ένα διάνυσμα N διαστάσεων.

Επόμενο στάδιο στη σχεδίαση του συστήματος αποτελεί ο τρόπος με τον οποίο θα διαχειριστεί το τελευταίο τη γνώση που αποκομίζει κατά την εκπαίδευση, με σκοπό την αποδοτικότερη λειτουργία του. Υιοθετώντας το μοντέλο της διανυσματικής αναπαράστασης, αναζητούμε ουσιαστικά μια συνάρτηση f η οποία προσεγγίζει όσο το δυνατόν περισσότερο μια ιδανική συνάρτηση με την οποία δύναται να μοντελοποιηθεί το πρόβλημα, τη συνάρτηση στόχο (target function) f . Οι δύο συναρτήσεις έχουν ελεύθερη μεταβλητή ένα τυχαίο διάνυσμα x , πεδίο ορισμού το χώρο των στιγμιότυπων και σύνολο τιμών το οποίο καθορίζεται από την εκάστοτε εφαρμογή. Έτσι, η επίλυση ενός προβλήματος μηχανικής μάθησης ανάγεται στην επίλυση ενός προβλήματος προσέγγισης των τιμών μιας συνάρτησης (function approximation).

Ανάλογα με το είδος της γνώσης που παρέχεται για εκπαίδευση, διαιρούμε το πεδίο της μηχανικής μάθησης σε τρεις μεγάλες κατηγορίες : στη μάθηση με επίβλεψη (supervised learning), στη μάθηση χωρίς επίβλεψη (unsupervised learning) και στην μάθηση με ενίσχυση(reinforcement learning). Στην πρώτη περίπτωση, η διαδικασία της μάθησης μαθαίνει από τα στιγμιότυπα εκπαίδευσης υπό την έννοια ότι υποδεικνύουν στον αλγόριθμο μάθησης την τιμή της συνάρτησης στόχου που επιδιώκει να προσεγγίσει. Στην κατηγορία αυτή ανήκουν τα λεγόμενα προβλήματα ταξινόμησης (classification), στα οποία το αποτέλεσμα της κατηγοριοποίησης αποτελεί χαρακτηριστικό των στιγμιότυπων εκπαίδευσης και η τιμή του για κάθε στιγμιότυπο συμπεριλαμβάνεται στα δεδομένα εκπαίδευσης. Αντίθετα, στην περίπτωση της μάθησης χωρίς επίβλεψη, ο αλγόριθμος καλείται να ανακαλύψει δομικούς σχηματισμούς στα στιγμιότυπα εκπαίδευσης, βάσει κριτηρίων τα οποία δεν του παρέχονται άμεσα. Παραδείγματα προβλημάτων που εντοπίζονται σε αυτήν την περιοχή είναι: η ομαδοποίηση οντοτήτων (clustering), όπου επιχειρείται ομαδοποίηση των στιγμιότυπων σε κλάσεις τις οποίες θα συμπεράνει ο αλγόριθμος, στηριζόμενος στην ομοιότητα των μελών κάθε ομάδας, η αριθμητική πρόβλεψη (numeric prediction), κατά την οποία δεν αναζητείται η κατηγορία ενός στιγμιότυπου αλλά μια αριθμητική ποσότητα, και η αποκάλυψη συσχετίσεων (association discovery), αντικείμενο της οποίας αποτελεί η αναζήτηση ενδιαφερουσών δομών στα στιγμιότυπα εκπαίδευσης μέσω της ταυτόχρονης πρόβλεψης της τιμής περισσότερων του ενός χαρακτηριστικών. Τέλος, στην μάθηση με ενίσχυση ο αλγόριθμος χρησιμοποιεί τα δεδομένα εισόδου προκειμένου να πάρει μια απόφαση. Αφού λάβει την απόφαση του δίνεται η πληροφορία για το αν πέτυχε τον επιθυμητό στόχο η όχι, και χρησιμοποιώντας αυτό προσπαθεί να βελτιώσει τα αποτελέσματά του. Εφαρμόζεται στον έλεγχο κίνησης ρομπότ, στη βελτιστοποίηση εργασιών σε εργοστάσια, στη μάθηση επιτραπέζιων παιχνιδιών και πολλούς άλλους τομείς.

2.2 ΔΙΑΔΙΚΑΣΙΑ ΜΑΘΗΣΗΣ

Όπως η μάθηση θεωρείται βασικό συστατικό της ανθρώπινης νοημοσύνης, έτσι και η μηχανική μάθηση είναι βασικό συστατικό της τεχνητής νοημοσύνης, προσφέροντος επιπλέον πλεονεκτήματα τα οποία η κλασική τεχνητή νοημοσύνη των κανόνων δεν προσφέρει. Τα 2 κύρια πλεονεκτήματα της μηχανικής μάθησης είναι η ευελιξία και η προσαρμοστικότητα.

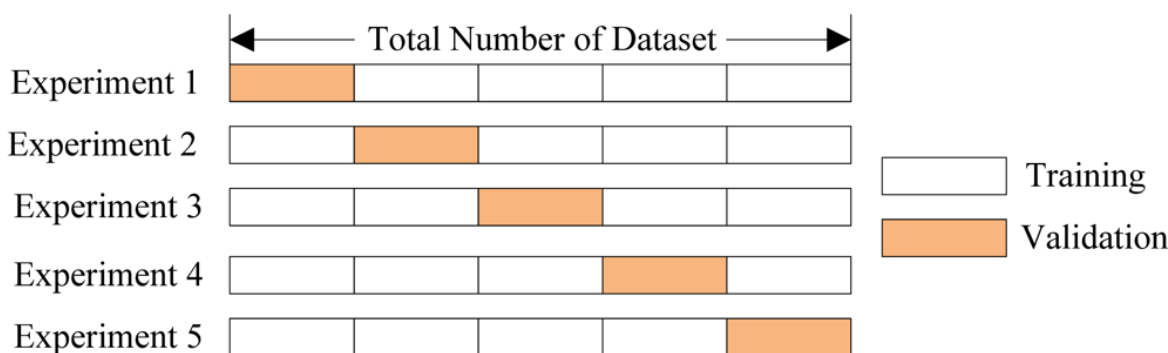
Ένα παράδειγμα είναι η προσπάθεια διαχωρισμού επιβατικών αυτοκινήτων από φορτηγά. Ο διαχωρισμός πρέπει να γίνει με βάση κάποιο χαρακτηριστικό. Σαν χαρακτηριστικό διαχωρισμού μπορεί να χρησιμοποιηθεί το ύψος των οχημάτων. Δημιουργούνται λοιπόν οι εξής κανόνες: αν το αντικείμενο είναι κάτω από 2 μέτρα είναι επιβατικό, διαφορετικά αν είναι πάνω από 3 μέτρα είναι φορτηγό. Τι θα συμβεί όμως σε περίπτωση που έχουμε κάποιο τζίπ; Με χρήση των παραπάνω κανόνων ο κατηγοριοποιητής θα αποφασίσει ότι ανήκει στην κλάση των φορτηγών, αποτέλεσμα το οποίο είναι λάθος. Η αποφυγή δημιουργίας κανόνων για κάθε νέο πρόβλημα αντιμετωπίζεται με χρήση αλγορίθμων μάθησης, οι οποίοι λαμβάνουν σαν δεδομένα πολλά πρότυπα από κάθε κλάση οχημάτων.

Γενικά, ένα πρότυπο είναι ένα διάνυσμα με ψ αριθμό χαρακτηριστικών $P = [X_1, X_2, \dots, X_\psi]$. Στο συγκεκριμένο παράδειγμα το πρότυπο αποτελείται από ένα στοιχείο το οποίο είναι το ύψος του οχήματος : $P = [X]$.

Χρησιμοποιώντας τα πρότυπα ο αλγόριθμος θα "μάθει" να ξεχωρίζει τους διάφορους τύπους αυτοκινήτων με βάση το ύψος. Αυτός άλλωστε είναι και ο απώτερος σκοπός της μηχανικής μάθησης. Η ικανότητα δηλαδή των μοντέλων να χρησιμοποιούν σωστά την γνώση που έχουν αποκτήσει για οποιοδήποτε πρότυπο κληθούν να κατηγοριοποιήσουν, το οποίο δεν έχουν συναντήσει άλλη φορά στο παρελθόν. Η ενέργεια αυτή ονομάζεται γενίκευση. Στον κλάδο της μηχανικής μάθησης ορίζεται ως γενίκευση η ικανότητα ενός αλγορίθμου να είναι αποτελεσματικός σε ένα εύρος εισόδων και εφαρμογών. Το ερώτημα που προκύπτει λοιπόν είναι πως διαπιστώνουμε κατά πόσο ο αλγόριθμος μας είναι ικανός να γενικεύει σωστά ή καλύτερα, με μεγάλο ποσοστό ακριβείας. Για να γίνει η συγκεκριμένη διαπίστωση αυτό πρέπει να ακολουθηθούν δυο πολύ σημαντικές διαδικασίες, αυτή της εκπαίδευσης και αυτή του έλεγχου. Κατά την διαδικασία της εκπαίδευσης στον αλγόριθμο μηχανικής μάθησης εισάγονται δεδομένα και εκπαιδεύεται ο αλγόριθμος σε αυτά. Από αυτή την διαδικασία προκύπτει ένα σφάλμα. Στην διαδικασία έλεγχου εισάγονται δεδομένα τα οποία δεν πρέπει να έχουν χρησιμοποιηθεί κατά την διαδικασία της εκπαίδευσης. Από εδώ θα προκύψει άλλο ένα σφάλμα.

Υπάρχουν βέβαια και μέθοδοι οι οποίοι βοηθάνε στη εύρεση σχετικά με το πόσο καλά γενικεύει ο αλγόριθμος μας. Μια τέτοια μέθοδος είναι η μέθοδος διασταύρωσης η αλλιώς μέθοδος Cross-Validation. Έστω ότι έχουμε ένα σύνολο δεδομένων P . Προκειμένου να χρησιμοποιήσουμε αυτή τη μέθοδο κάνουμε τα εξής βήματα :

- Χωρίζουμε το σύνολο δεδομένων σε N τμήματα ίδιου μεγέθους
- Χρησιμοποιούμε τα $N-1$ τμήματα για να εκπαιδεύσουμε το μοντέλο αφήνοντας το ένα που δεν χρησιμοποιήσαμε για την διαδικασία του ελέγχου
- Υπολογίζουμε το σφάλμα που προκύπτει για το συγκεκριμένο πείραμα
- Εκτελούμε ξανά την προηγούμενη διαδικασία αφήνοντας αυτή την φορά κάποιο άλλο από τα τμήματα για τον έλεγχο υπολογίζοντας ξανά το αντίστοιχο σφάλμα. Επαναλαμβάνουμε την διαδικασία S φορές ώστε να χρησιμοποιηθεί κάθε τμήμα ξεχωριστά για έλεγχο
- Αφού τελειώσουμε υπολογίζουμε τον μέσο ορό σφαλμάτων έλεγχου για όλα τα S πειράματα, τα οποία ονομάζονται "fold"
- Εκτελούμε την παραπάνω διαδικασία για διαφορετικό αλγόριθμο μάθησης
- Επιλεγούμε τον αλγόριθμο με το μικρότερο μέσο ορό σαγμάτων σαν καλύτερο στο να γενικεύει



Σχήμα 1 "Αναπαράσταση μεθόδου Cross-Validation"

2.3 ΜΟΝΤΕΛΑ ΜΑΘΗΣΗΣ

Αναφερθήκαμε προηγουμένως στην μηχανική μάθηση, τα προβλήματα τα οποία μας βοήθα να αντιμετωπίσουμε καθώς και τους τρόπους με τους οποίους ένα μοντέλο μπορεί να μάθει να γενικεύει. Μοντέλα όμως μάθησης υπάρχουν πολλά. Θα ανέρθουμε σε αυτά και θα αναλύσουμε το ένα βασικό μοντέλο το οποίο και χρησιμοποιήθηκε σε αυτή την πτυχιακή εργασία. Τα μοντέλα μάθησης είναι τα εξής :

- Τεχνητά νευρωνικά δίκτυα :
Αλγόριθμοι μάθησης που εμπνέονται από την δομή των βιολογικών νευρώνων. Οι υπολογισμοί βασίζονται στην επικοινωνία εσωτερικών διασυνδεδεμένων τεχνητών νευρώνων. Χρησιμοποιούνται για μοντελοποίηση σχέσεων, ανακάλυψη προτύπων και για εντοπισμό στατιστικών δομών.

- Μηχανές διανυσμάτων υποστήριξης :
Σύνολο μεθόδων επιτηρούμενης μάθησης που χρησιμοποιούνται για ταξινόμηση και παλινδρόμηση.
- Στατιστικά μοντέλα Bayes :
Γραφικά πιθανοθεωρητικά μοντέλα που απεικονίζουν σύνολα τυχαίων μεταβλητών και την μεταξύ τους ανεξαρτησία.
- Δέντρα αποφάσεων :
Τα δέντρα αποφάσεων χρησιμοποιούνται ως προγνωστικά μοντέλα τα οποία αντιστοιχίζουν παρατηρήσεις με συμπεράσματα όσο αφορά την τιμή στόχο του αντικειμένου.
- Εκμάθηση με κανόνες συσχέτισης :
Η εκμάθηση με κανόνες συσχέτισης είναι μια μέθοδος ανακάλυψης ενδιαφερουσών σχέσεων μεταξύ των μεταβλητών σε μεγάλες βάσεις δεδομένων.
- Επαγωγικός λογικός προγραμματισμός :
Χρησιμοποιεί λογικό προγραμματισμό ως τρόπο παρουσίασης δεδομένων εισόδου και με την χρήση αυτού και μιας δεδομένης κωδικοποίησης παράγεται το λογικό πρόγραμμα που περιέχει όλα τα θετικά δεδομένα.
- Ομαδοποίηση :
Διαδικασία κατά την οποία ένα σύνολο παρατηρήσεων χωρίζεται σε υποσύνολα έτσι ώστε οι παρατηρήσεις που ανήκουν στην ίδια ομάδα (cluster) είναι όμοιες, σύμφωνα με κάποιο ή κάποια προκαθορισμένα κριτήρια, ενώ οι παρατηρήσεις που προέρχονται από διαφορετικά υποσύνολα είναι ανόμοιες.
- Ενισχυτική μάθηση :
Η Ενισχυτική μάθηση ασχολείται με το πώς ένα υποκείμενο θα πρέπει να δράσει σε ένα περιβάλλον ώστε να μεγιστοποιηθεί η απόδοσή του. Η ενισχυτική μάθηση διαφέρει από τα προβλήματα επιτηρούμενης μάθησης αφού τα σωστά ζεύγη δεδομένων εισόδου/εξόδου δεν παρουσιάστηκαν ποτέ, ούτε οι βέλτιστες δυνατές ενέργειες έχουν δοθεί.
- Εκμάθηση με μετρό ομοιότητας :
Σε αυτή την περίπτωση η μηχανή μάθησης λαμβάνει όμοια και ανόμοια ζεύγη παραδειγμάτων και προσπαθεί να εντοπίσει μια

συνάρτηση ομοιότητας που μπορεί να κατατάσσει σωστά ένα άγνωστο ζεύγος αντικειμένων.

- Γενετικοί αλγόριθμοι :
Ένας γενετικός αλγόριθμος είναι μια ευρετική αναζήτηση που μιμείται τη διαδικασία της φυσικής επιλογής, και χρησιμοποιεί μεθόδους όπως αυτή της μετάλλαξης και της διασταύρωσης προκειμένου να δημιουργήσει καινούρια γονότυπα με την ελπίδα εύρεσης αποτελεσματικών λύσεων σε ένα συγκεκριμένο πρόβλημα
- Βαθιά μάθηση / Deep Learning :
Η βαθιά μάθηση προσπαθεί να μοντελοποιήσει τον τρόπο που διαχειρίζεται τις πληροφορίες ο ανθρώπινος εγκέφαλος. Θα μιλήσουμε αναλυτικά για αυτή στο επόμενο κεφάλαιο, καθώς χρησιμοποιείται σε αυτή την πτυχιακή εργασία.

2.4 DEEP LEARNING / ΒΑΘΙΑ ΜΑΘΗΣΗ

Η βαθιά μάθηση είναι ένα μοντέλο μηχανικής μάθησης το οποίο είναι εμπνευσμένο από τον τρόπο που λειτουργεί ο ανθρώπινος εγκέφαλος. Ο όρος Deep Learning εισήχθη στη κοινότητα της μηχανικής μάθησης από τη Rina Dechter το 1986, και στα Τεχνητά Νευρωνικά Δίκτυα από τον Igor Aizenberg και τους συναδέλφους του το 2000(, στο πλαίσιο των δυαδικών νευρώνων κατωφλίου).

Στο Deep Learning, κάθε επίπεδο νευρώνων μαθαίνει να μετασχηματίζει τα δεδομένα εισόδου του σε μια ελαφρώς πιο αφηρημένη και σύνθετη αναπαράσταση. Το μεγάλο και σημαντικό κομμάτι του είναι ότι ένα βαθύ δίκτυο μπορεί να μάθει "από μόνο το" ποια χαρακτηριστικά να τοποθετήσει στο σωστό επίπεδο ώστε να εκλάβει το βέλτιστο αποτέλεσμα. Τα βαθιά δίκτυα πήραν το όνομα του από το πλήθος των στρωμάτων νευρώνων στα οποία γίνεται ο μετασχηματισμός των δεδομένων. Τα συστήματα βαθιάς μάθησης έχουν ένα βάθος διαδρομής εκχώρησης πίστωσης (credit assignment path – CAP), το οποίο είναι η αλυσίδα μετασχηματισμού από την είσοδο των δεδομένων μέχρι την έξοδο. Για να γίνουν όλα τα παραπάνω πιο κατανοητά ας κάνουμε σε αυτό το σημείο ένα παράδειγμα : έστω ότι στο βαθύ δίκτυό μας εισάγουμε μια εικόνα ενός αυτοκινήτου. Η είσοδος προφανώς δεν μπορεί να είναι η εικόνα καθαυτή αλλά μια αναπαράσταση της, δηλαδή ένας πίνακας με τα εικονοστοιχεία της. Στο πρώτο στρώμα αναπαραστάσεων μπορεί να γίνει αφαίρεση των εικονοστοιχείων και κωδικοποίηση των ακρών. Στο δεύτερο στρώμα μπορεί να γίνει σύνθεση και κωδικοποίηση διατάξεων ακμών. Στο τρίτο στρώμα μπορεί να κωδικοποιηθούν οι ροδές. Τέλος, στο τέταρτο στρώμα μπορεί να αναγνωρίσει το δίκτυο ότι η εικόνα περιέχει ένα αυτοκίνητο.

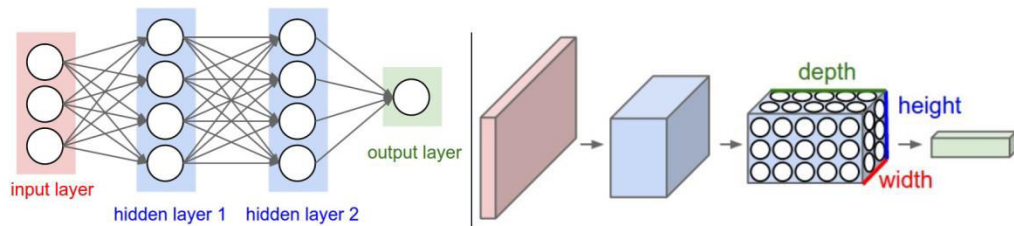
Υπάρχουν όμως και υπολογιστικά συστήματα τα οποία είναι εμπνευσμένα από τους βιολογικούς νευρώνες που βρίσκονται στους εγκεφάλους των ζώων. Αυτά τα συστήματα ονομάζονται τεχνητά νευρωνικά δίκτυα ή αλλιώς artificial neural networks (ANNs). Τα συστήματα αυτά βελτιώνουν τη λειτουργία τους εκτελώντας κάποια γενικά παραδείγματα χωρίς να χρειάζεται κάποιου είδους προγραμματισμός για συγκεκριμένες διεργασίες. Τα συστήματα αυτά μπορεί να αποτελέσουν σημαντική βοήθεια σε εφαρμογές όπου η κλασική αλγοριθμική δεν μπορεί ή είναι δύσκολο να λύσει το εκάστοτε πρόβλημα. Ένα τεχνητό νευρωνικό δίκτυο βασίζεται σε ένα σύνολο διασυνδεδεμένων μονάδων οι οποίες ονομάζονται τεχνητοί νευρώνες. Οι συνδέσεις μεταξύ των νευρώνων ονομάζονται συνάψεις και μπορούν να μεταφέρουν ένα σήμα από νευρώνα σε νευρώνα. Ο νευρώνας που δέχεται ένα σήμα μπορεί να εκτελέσει διαφορές πράξεις σε αυτό και στη συνέχεια να το μεταδώσει σε έναν άλλο νευρώνα. Οι τιμές των σημάτων μπορεί να είναι 0 και 1. Υπάρχουν επίσης τόσο σε νευρώνες όσο και συνάψεις βάρη τα οποία διαφέρουν ανάλογα με την αντίστοιχη διαδικασία εκμάθησης. Η διάταξη των νευρώνων γίνεται σε στρώματα και τα σήματα μεταδίδονται από την είσοδο προς την έξοδο.

Αφού λοιπόν εξηγήσαμε τι είναι ένα τεχνητό νευρωνικό δίκτυο είμαστε σε θέση να καταλάβουμε τι είναι τα βαθιά νευρωνικά δίκτυα. Ένα βαθύ νευρωνικό δίκτυο – Deep Neural Network (DNN), είναι ένα τεχνητό νευρωνικό δίκτυο το οποίο έχει πολλαπλά κρυφά στρώματα νευρώνων ανάμεσα στο στρώμα εισόδου και στο στρώμα εξόδου. Αυτό που επιτυγχάνεται με την χρήση των επιπλέον κρυφών στρωμάτων είναι η δυνατότητα μοντελοποίησης δεδομένων πολύπλοκων μορφών που είναι δύσκολα να γίνει με χρήση απλών δικτύων. Το ερώτημα που μπορεί να θέσει κάποιος σε αυτό το σημείο είναι πιο είναι τα κατάλληλο πλήθος στρωμάτων. Η απάντηση στο παραπάνω ερώτημα είναι ότι ο αριθμός αυτός διαφέρει από πρόβλημα σε πρόβλημα. Υπάρχουν αρχιτεκτονικές δικτύων που επιτυγχάνουν να αντιμετωπίσουν συγκεκριμένου είδους προβλήματα. Οι ίδιες αρχιτεκτονικές όμως μπορεί σε κάποιο άλλο τομέα να μην τα καταφέρουν το ίδιο καλά. Γενικά στα βαθιά νευρωνικά δίκτυα δεν υπάρχει κάποια συγκεκριμένη αρχιτεκτονική, τόσο γενικά, όσο και ανά κατηγορίες προβλημάτων και εφαρμογών. Συνέχεια προκύπτουν νέες αρχιτεκτονικές μέσω πειραμάτων οι οποίες προσπαθούν να επιτύχουν καλύτερα αποτελέσματα σε σχέση με άλλες υπάρχουσες αρχιτεκτονικές. Ο μονός τρόπος για να συγκριθούν δυο η περισσότερες αρχιτεκτονικές είναι να συγκριθούν τα αποτελέσματα που έχουν σε ίδια σύνολα δεδομένων.

Έχοντας λοιπόν εξηγήσει τι είναι ένα βαθύ νευρωνικό δίκτυο μπορούμε σε αυτό το σημείο να μιλήσουμε για τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks – CNN ή ConvNet). Τα συνελκτικά δίκτυα αποτελούνται από νευρώνες οι οποίοι έχουν βάρη που αυτοπροσαρμόζονται ανάλογα με το πρόβλημα. Η λειτουργία τους είναι ίδια με την λειτουργία που εκτελούν τα βαθιά νευρωνικά δίκτυα. Τι το διαφορετικό υπάρχει όμως στα συνελκτικά δίκτυα; Οι αρχιτεκτονικές των συνελκτικών δικτύων κάνουν την εξής παραδοχή: τα δεδομένα που εισέρχονται στο δίκτυο είναι δεδομένα εικόνων. Έχοντας λοιπόν εικόνες σαν είσοδο είναι δυνατή η κωδικοποίηση ορισμένων ιδιοτήτων όσο αφορά στην αρχιτεκτονική των δικτύων. Ως αποτέλεσμα η εμπρόσθια λειτουργία είναι πιο αποτελεσματική και η ποσότητα των παραμέτρων των δικτύων μειώνεται σημαντικά.

Όπως έχουμε αναφέρει, ένα απλό συνελκτικό δίκτυο αποτελείται από μια ακολουθία στρωμάτων, και σε κάθε επίπεδο αλλάζει η μορφή των δεδομένων με την ενεργοποίηση μιας διαφορετικής σε κάθε επίπεδο συνάρτησης διαμόρφωσης. Για την δημιουργία διαφόρων αρχιτεκτονικών συνελκτικών δικτύων χρησιμοποιούνται τρία βασικά είδη στρωμάτων. Αυτά τα είδη είναι :

1. Συνελκτικό επίπεδο – Convolutional layer
2. Συγκεντρωτικό επίπεδο – Pooling layer
3. Πλήρως συνδεδεμένο επίπεδο – Fully Connected Layer



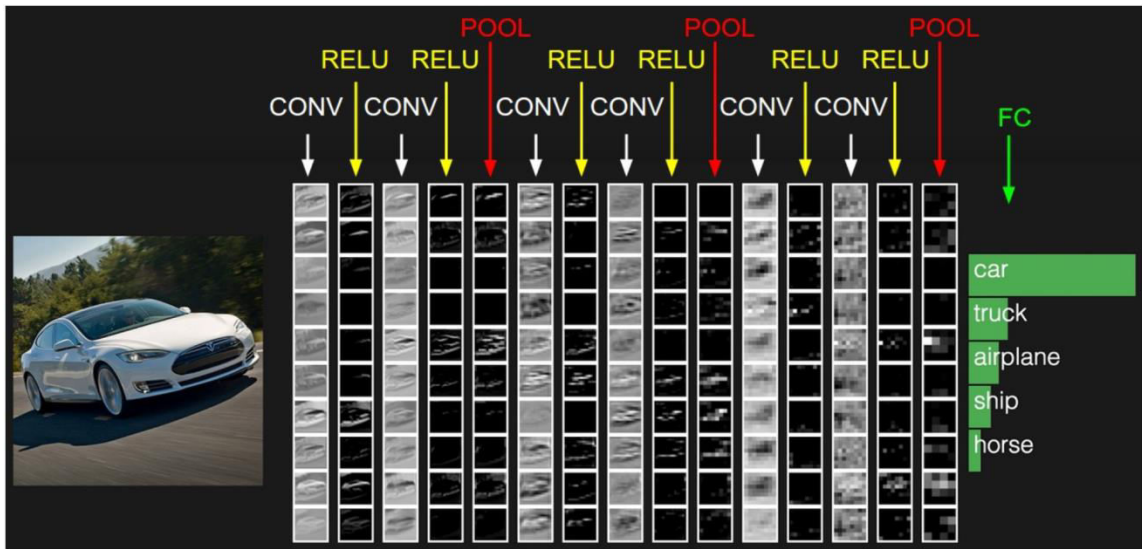
Σχήμα 2 "Αναπαράσταση νευρωνικού δικτύου τριών στρωμάτων"

Τοποθετώντας αυτά τα επίπεδα το ένα μετά το άλλο δημιουργούμε μια πλήρη αρχιτεκτονική συνελκτικού δικτύου. Επιπλέον υπάρχει και το επίπεδο RELU. Για να καταλάβουμε καλύτερα πως λειτουργούν αυτά τα στρώματα θα το εξηγήσουμε με ένα παράδειγμα. Έστω ότι θέλουμε να κάνουμε κατηγοριοποίηση των εικόνων του συνόλου δεδομένων του CIFAR[1], με διαστάσεις 32×32 .

- Στρώμα Εισόδου : Το στρώμα εισόδου θα έχει τις τιμές των εικονοστοιχείων της εικόνας. Πιο συγκεκριμένα θα έχει διαστάσεις που θα είναι το πλάτος της εικόνας, το ύψος της, καθώς και τα τρία κανάλια των χρωμάτων RGB. Δηλαδή θα έχουμε ένα πίνακα διάστασης $32 \times 32 \times 3$.
- Συνελκτικό Στρώμα : Εδώ θα γίνει υπολογισμός των εξόδων των νευρώνων που συνδέονται με την είσοδο. Κάθε υπολογισμός είναι ένας πολλαπλασιασμός μεταξύ των βαρών των νευρώνων και μιας περιοχής που είναι συνδεδεμένη με την είσοδο. Το πλήθος των φίλτρων που θα χρησιμοποιηθούν καθορίζουν και το μέγεθος του πίνακα, για παράδειγμα με πλήθος φίλτρων 12 θα καταλήξουμε σε πίνακα διάστασης $32 \times 32 \times 12$.
- Στρώμα RELU : Εάν χρησιμοποιηθεί αυτό το στρώμα, θα εφαρμόσει μια στοιχειώδη συνάρτηση ενεργοποίησης, όπως η συνάρτηση μεγίστου (max). Το μέγεθος του πίνακα παραμένει ως έχει.
- Συγκεντρωτικό επίπεδο : Αυτό το επίπεδο θα εκτελέσει μια διαδικασία δειγματοληψίας κατά μήκος των χωρικών διαστάσεων (πλάτος, ύψος). Αυτό θα μειώσει τις διαστάσεις του πίνακα από $32 \times 32 \times 12$ σε $16 \times 16 \times 12$.

- Πλήρως συνδεδεμένο επίπεδο : Στο τελευταίο επίπεδο θα γίνει υπολογισμός των αποτελεσμάτων των κλάσεων, καταλήγοντας σε πινάκα διαστάσεων $1 \times 1 \times 10$, όπου κάθε ένας από τους 10 αριθμούς αντιστοιχεί σε βαθμολογία κλάσης, για τις 10 κλάσεις που έχει το CIFAR.

Μέσω της παραπάνω διαδικασίας, το συνολικό δίκτυο μετασχηματίζει το αρχικό επίπεδο που αποτελείται από εικονοστοιχεία σε βαθμολογίες κλάσεων.



Σχήμα 3 "Αναπαράσταση λειτουργίας παραδείγματος του CIFAR"

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο μιλήσαμε για την μηχανική μάθηση. Αναλύσαμε τον τρόπο λειτουργίας, τους σκοπούς της και για ποιο λόγο είναι απαραίτητη για την λύση συγκεκριμένων προβλημάτων. Αναφέραμε επίσης τομείς της μηχανικής μάθησης και εμβαθύναμε στον τομέα της Βαθιάς Μάθησης καθώς είναι πολύ σημαντικό η κατανόησή του προκειμένου να εκπονήσουμε την συγκεκριμένη πτυχιακή εργασία. Στο επόμενο κεφάλαιο θα αναφερθούμε σε επιπλέον τεχνολογίες που είναι πολύ χρήσιμες και βοηθούν στην εκπόνηση της πτυχιακής εργασίας.

ΚΕΦΑΛΑΙΟ 3

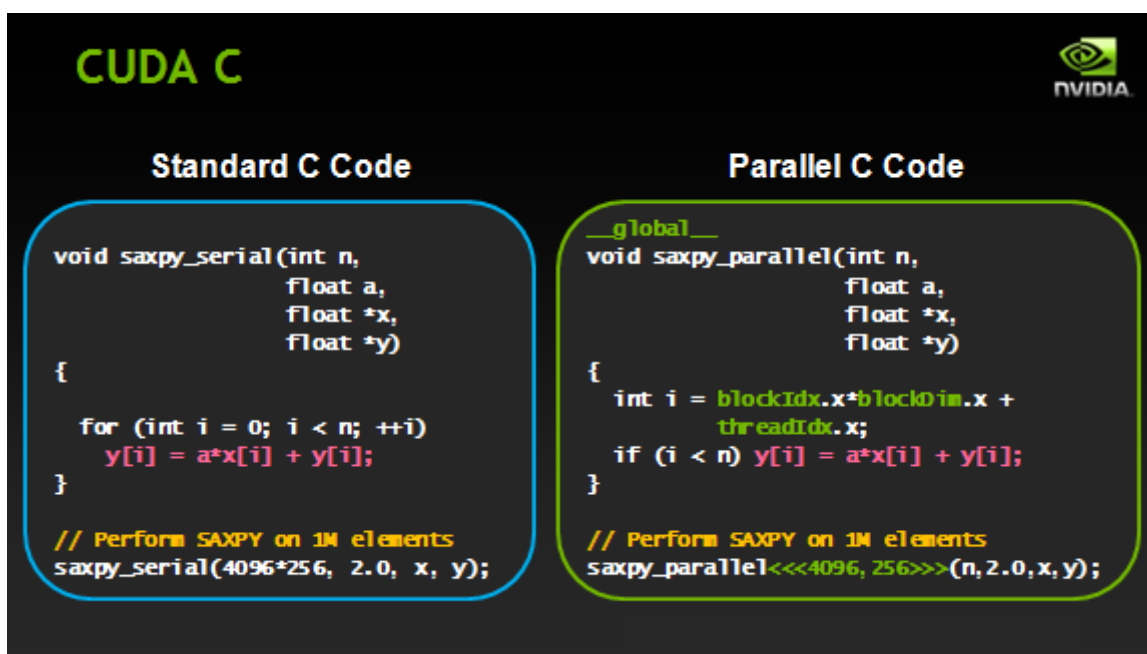
ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΕΙΣΑΓΩΓΗ

Η παράλληλη επεξεργασία είναι μια μέθοδος ταυτόχρονης διάσπασης και εκτέλεσης εργασιών του προγράμματος σε πολλαπλούς μικροεπεξεργαστές, με σκοπό την αύξηση των υπολογιστικών επιδόσεων και την μείωση του απαιτούμενου χρόνου εκτέλεσης ενός προγράμματος. Η παράλληλη επεξεργασία μπορεί να πραγματοποιηθεί μέσω ενός υπολογιστή με δύο ή περισσότερους επεξεργαστές ή μέσω ενός δικτύου υπολογιστών. Οι απαιτήσεις που υπάρχουν πλέον όσο αφορά την ταχύτητα αλλά και το μέγεθος των προγραμμάτων κάνουν την παράλληλη επεξεργασία να είναι μονόδρομος σε πολλές περιπτώσεις εκτέλεσης προγραμμάτων.

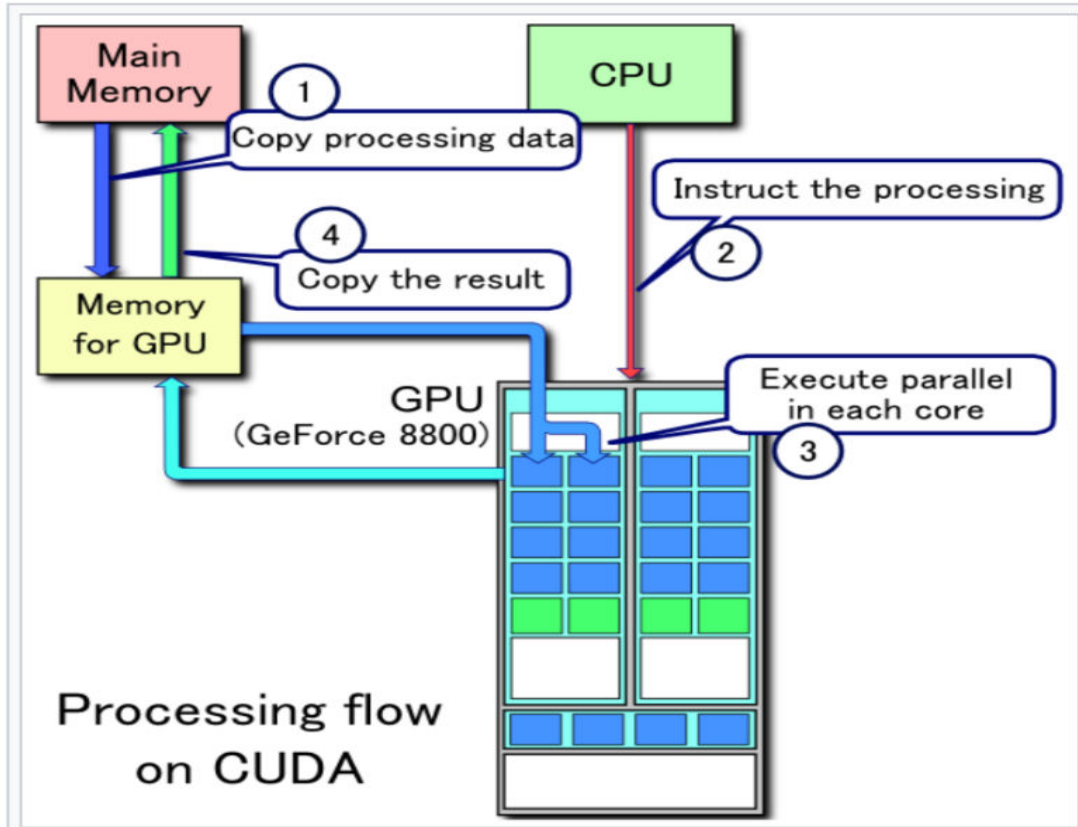
3.1 Η ΠΛΑΤΦΟΡΜΑ CUDA

Η CUDA είναι μια παράλληλη υπολογιστική πλατφόρμα και μοντέλο διεπαφής προγραμματισμού εφαρμογών που δημιουργήθηκε από τη εταιρία σχεδιασμού μονάδων επεξεργασίας γραφικών (Graphics Processing Unit – GPU) Nvidia. Με τη χρήση αυτής της πλατφόρμας είναι δυνατή η χρήση των μονάδων επεξεργασίας γραφικών της Nvidia ως μια επιπλέον πηγή υπολογιστικής ισχύος. Πιο συγκεκριμένα η πλατφόρμα αυτή είναι ένα στρώμα λογισμικού το οποίο παρέχει άμεση πρόσβαση στο σύνολο εντολών της κάρτας γραφικών για εκτέλεση ενεργειών σε αυτή. Ένα μεγάλο πλεονέκτημα της πλατφόρμας CUDA είναι ο σχεδιασμός της να δουλεύει με άλλες γλώσσες προγραμματισμού όπως C, C++, το οποίο δίνει τη δυνατότητα σε χρήστες να την χρησιμοποιούν χωρίς να απαιτείται η εξειδικευμένη γνώση προγραμματισμού γραφικών στοιχείων.



Σχήμα 4 "Αναπαράσταση κώδικα γλώσσας C χωρίς και με χρήση της πλατφόρμας CUDA"

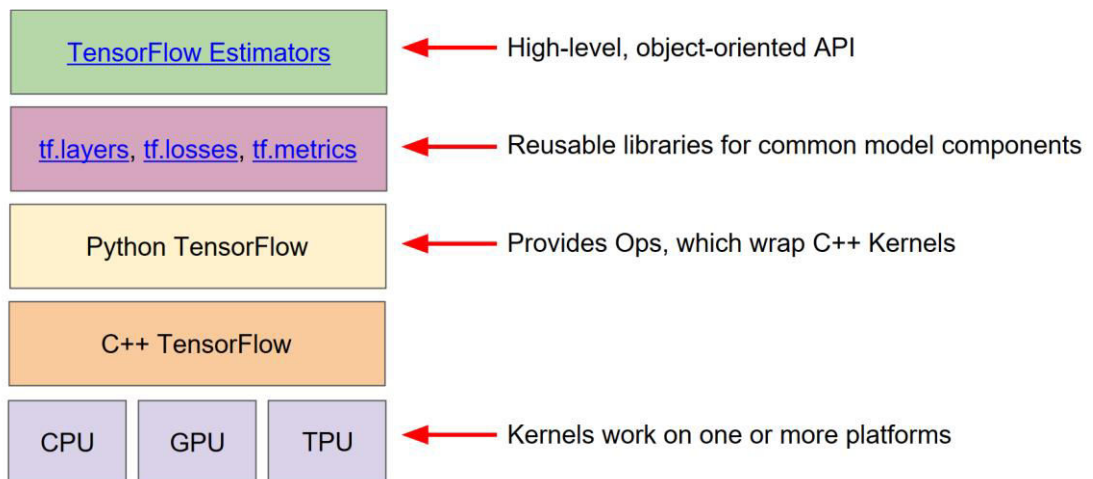
Η χρήση της συγκεκριμένης πλατφόρμας μπορεί να ενισχύσει σε μεγάλο βαθμό τόσο τον χρόνο εκτέλεσης των προγραμμάτων, όσο και να διευκολύνει το έργο της κύριας μονάδας επεξεργασίας (Central Processing Unit – CPU). Η βελτίωση αυτή μπορεί να παρατηρηθεί σε μεγάλα προβλήματα, που τόσο το μέγεθος του προγράμματος αλλά και των δεδομένων είναι πολύ μεγάλα.



Σχήμα 5 "Αναπαράσταση ροής επεξεργασίας με χρήση της πλατφόρμας CUDA"

3.2 Η ΒΙΒΛΙΟΘΗΚΗ TENSORFLOW

Το Tensorflow είναι μια συμβολική βιβλιοθήκη μαθηματικών ανοιχτού κώδικα η οποία χρησιμοποιείται για προγραμματισμό ροής δεδομένων (Dataflow Programming) καθώς και σε ένα εύρος εφαρμογών μηχανικής μάθησης όπως για παράδειγμα τα βαθιά νευρωνικά δίκτυα. Το Tensorflow μπορεί να χρησιμοποιείται σε άπλες κεντρικές υπολογιστικές μονάδες αλλά το μεγάλο του πλεονέκτημα είναι ότι μπορεί επίσης να εκτελείται σε πολλαπλές κεντρικές υπολογιστικές μονάδες αλλά και σε πολλαπλές μονάδες επεξεργασίας γραφικών. Μπορεί δηλαδή να χρησιμοποιηθεί σε συνδυασμό με την πλατφόρμα CUDA αλλά και με άλλες υπολογιστικές πλατφόρμες γενικού σκοπού βασισμένες τόσο γραφικές μονάδες επεξεργασίας αλλά και κεντρικές.

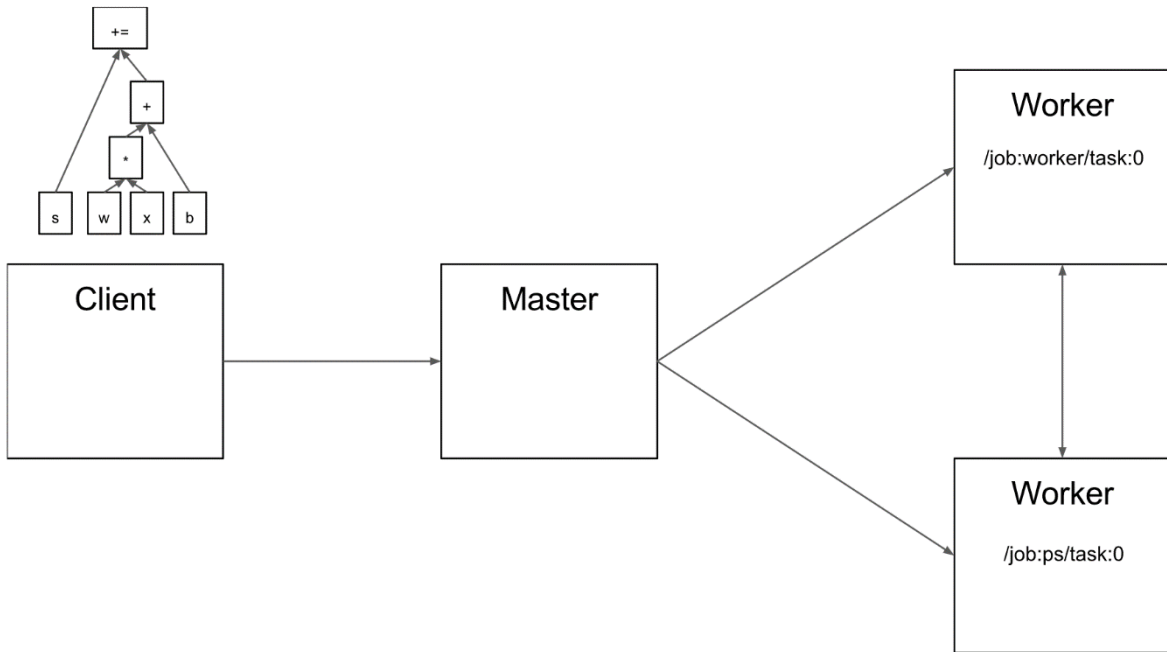


Σχήμα 6 "Αναπαράσταση ιεραρχίας εργαλείων της βιβλιοθήκης Tensorflow"

Οι υπολογισμοί που γίνονται με το Tensorflow αναπαριστούνται σαν γράφοι σταδιακής ροής δεδομένων. Το όνομα του προέρχεται από τους πινάκες που χρησιμοποιούνται για την αναπαράσταση δεδομένων, οι οποίοι πινάκες ονομάζονται τένσορες. Παρέχει επίσης μια μεγάλη ποικιλία εργαλείων με τα οποία είναι δυνατή η κατασκευή μοντέλων και αναπαραστάσεων με μεγάλη δυνατότητα αφαιρετικότητας, το οποίο είναι επιθυμητό στην κατασκευή αρχιτεκτονικών νευρωνικών δικτύων.

Ο τρόπος λειτουργίας του Tensorflow χαρακτηρίζεται από τρεις βασικούς ρόλους :

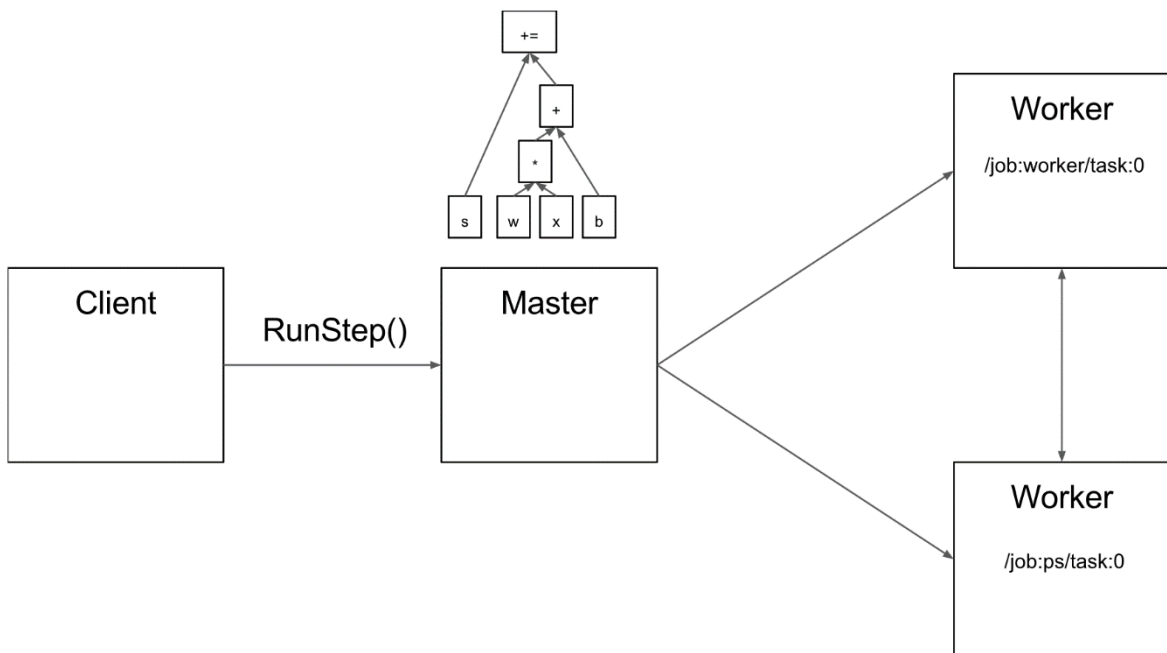
1. Πελάτης – Client



Σχήμα 7 "Αναπαράσταση του ρόλου Πελάτης"

Εδώ ο προγραμματιστής σχεδιάζει τον υπολογιστικό γράφο τον οποίο θέλει να εκτελέσει. Αφού γίνει ο σχεδιασμός του γράφου δημιουργείται από τον προγραμματιστή μια συνεδρία (session) στην οποία στέλνεται ο γράφος στον ιδιοκτήτη εργασιών.

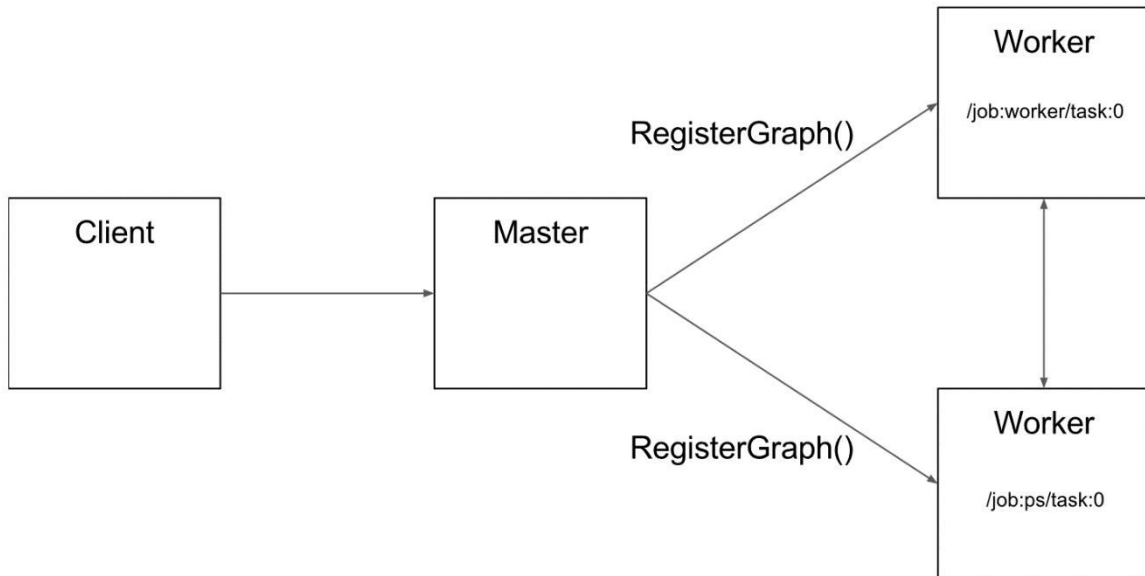
2. Ιδιοκτήτης Εργασιών – Distributed Master



Σχήμα 8 "Αναπαράσταση του ρόλου Ιδιοκτήτης Εργασιών"

Ο ιδιοκτήτης εργασιών εκτελεί με τρεις σημαντικές δουλείες. Πρώτον, από τον κύριο γράφο που έλαβε φτιάχνει τον υπογράφο ο οποίος χρειάζεται για την αξιολόγηση των κόμβων του γράφου. Δεύτερον, τεμαχίζει τον γράφο και κατανέμει τα κομμάτια στις συσκευές που υπάρχουν για να εκτελούν κομμάτια γραφών. Τέλος αποθηκεύει τα κομμάτια που δημιούργησε στο προηγούμενο βήμα ώστε να μπορεί να τα επαναχρησιμοποιήσει σε επόμενα βήματα.

3. Εργάτης Υπηρεσιών – Worker Services



Σχήμα 9 "Αναπαράσταση του ρόλου Εργάτης υπηρεσιών"

Οι εργάτες για κάθε αίτημα του ιδιοκτήτη εργασιών εκτελούν τις εξής ενέργειες : διαχειρίζονται τα αιτήματα του, χρονοπρογραμματίζει τις λειτουργίες των πυρήνων που αποτελούν έναν υπογράφο και μεσολαβεί στην επικοινωνία μεταξύ διεργασιών

ΕΠΙΛΟΓΟΣ

Σε αυτό το κέφαλο αναφερθήκαμε στο κομμάτι της παράλληλης επεξεργασίας. Αναφερθήκαμε στην πλατφόρμας CUDA και στην βιβλιοθήκη Tensorflow και κατανοήσαμε τον τρόπο και το όφελος που έχουν όταν καλούμαστε να εκτελέσουμε μεγάλου μεγέθους προγράμματα. Αυτές οι δυο τεχνολογίες χρησιμοποιήθηκαν σε συνδυασμό για την εκτέλεση του προγράμματος για την εκπαίδευση του νευρωνικού δικτύου που χρησιμοποιήθηκε για την εκπόνηση αυτής της πτυχιακής εργασίας. Στο επόμενο κεφάλαιο θα μιλήσουμε για το πρόγραμμα, το δίκτυο και όλα όσα είναι απαραίτητα για αυτή την πτυχιακή εργασία.

ΚΕΦΑΛΑΙΟ 4

ΠΕΡΙΓΡΑΦΗ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ ΚΙΤΤΙ ΚΑΙ ΣΥΝΕΛΙΚΤΙΚΟΥ ΔΙΚΤΥΟΥ SQUEEZEDET

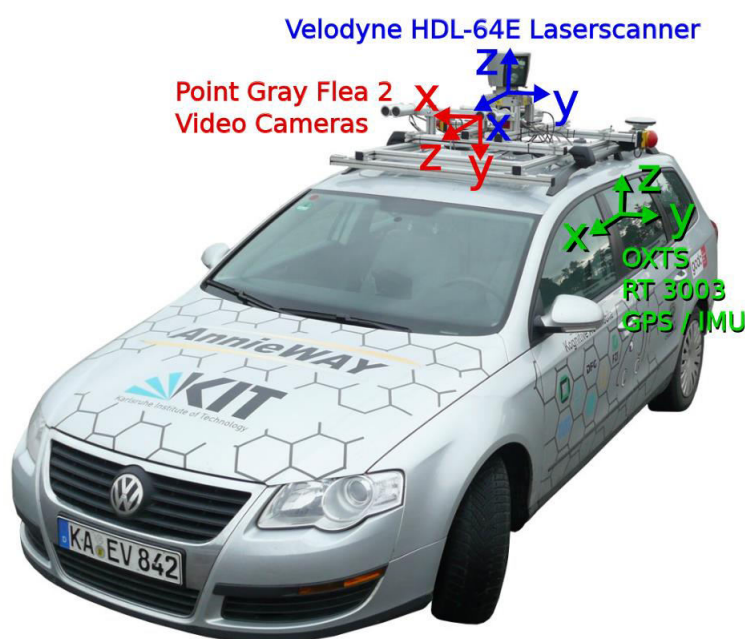
ΕΙΣΑΓΩΓΗ

Σε αυτό το σημείο έχουμε κατανοήσει πως λειτουργεί η μηχανική μάθηση στην προσπάθεια των ανθρώπων να λύσουν πιο σύνθετα προβλήματα. Επίσης έχουμε κατανοήσει πως για να λύσουμε το πρόβλημα του εντοπισμού και αποφυγής οδικών εμποδίων πρέπει να εκπαιδεύσουμε ένα νευρωνικό δίκτυο. Για να γίνει αυτό όμως είναι απαραίτητη η χρήση δεδομένων που θα χρησιμοποιηθούν για την διαδικασία της εκπαίδευσης του δικτύου. Θα μιλήσουμε λοιπόν για το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση καθώς και την αρχιτεκτονική του νευρωνικού δικτύου.

4.1 ΤΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ ΚΙΤΤΙ

Σύμφωνα με τους Andreas Geiger, Philip Lenz, Christoph Stiller και Raquel Urtasun [2] το σύνολο δεδομένων ΚΙΤΤΙ μπορεί να χρησιμοποιηθεί για έρευνα πάνω στους τομείς της αυτόνομης οδήγησης και της κινητής ρομποτικής. Είναι ένα αρκετά μεγάλο σύνολο δεδομένων καθώς χρησιμοποιήθηκαν διάφοροι αισθητήρες για την συλλογή των δεδομένων. Συγκεκριμένα η πλατφόρμα συλλογής των δεδομένων είχε τους εξής αισθητήρες συλλογής δεδομένων :

- Δυο κάμερες λήψης εικόνων στην κλίμακα του χρώματος γκρι, τύπου PointGray Flea2, τοποθετημένες η μια στο δεξί και η άλλη στο αριστερό μέρος του αυτοκινήτου.
- Δυο κάμερες λήψης έγχρωμων εικόνων, τύπου PointGray Flea2, τοποθετημένες η μια στο δεξί και η άλλη στο αριστερό μέρος του αυτοκινήτου.
- Τέσσερις φακοί με δυνατότητα λήψης εικόνων 90 μοιρών οριζοντίως και 35 μοιρών κατακόρυφα τύπου Edmund Optic lenses.
- Ένα περιστρεφόμενο σαρωτή λέιζερ τύπου Velodyne HDL-64E τοποθετημένο στη κορυφή του οχήματος.
- Ένα σύστημα πλοήγησης τύπου OXTS RT3003.



Σχήμα 10 "Η κινητή πλατφόρμα συλλογής δεδομένων του συνόλου δεδομένων ΚΙΤΤΙ"

Για κάθε λήψη φωτογραφίας υπάρχουν τα αντίστοιχα δεδομένα που συλλέχτηκαν από τους υπόλοιπους αισθητήρες. Τα δεδομένα είναι χωρισμένα ανά κατηγορίες αλλά δεν είναι απαραίτητο να χρησιμοποιούνται μεμονωμένα.

Για κάθε ακολουθία δεδομένων εκτός από τα καθαρά δεδομένα παρέχονται και σχολιασμοί αντικειμένων σε μορφή δισδιάστατων αλλά και τρισδιάστατων οροθετικών κουτιών (bounding box), καθώς αρχεία βαθμονόμησης που μεταφράζουν τις ρυθμίσεις ενός αισθητήρα σε αυτές ενός άλλου. Για παράδειγμα υπάρχουν οι συντεταγμένες του αυτοκινήτου τόσο από την θέση της κάμερας αλλά και από το λείζερ ραντάρ. Γίνεται επίσης διαχωρισμός κλάσεων αντικειμένων τα οποία υπάρχουν στο σύνολο δεδομένων. Οι κλάσεις αυτές είναι :

- Αυτοκίνητο
- Μικρό φορτηγό – Βαν
- Φορτηγό
- Πεζός
- Άνθρωπος καθιστός
- Ποδηλάτης
- Τραμ
- Διάφορα (πχ τρίλερ και αλλά)

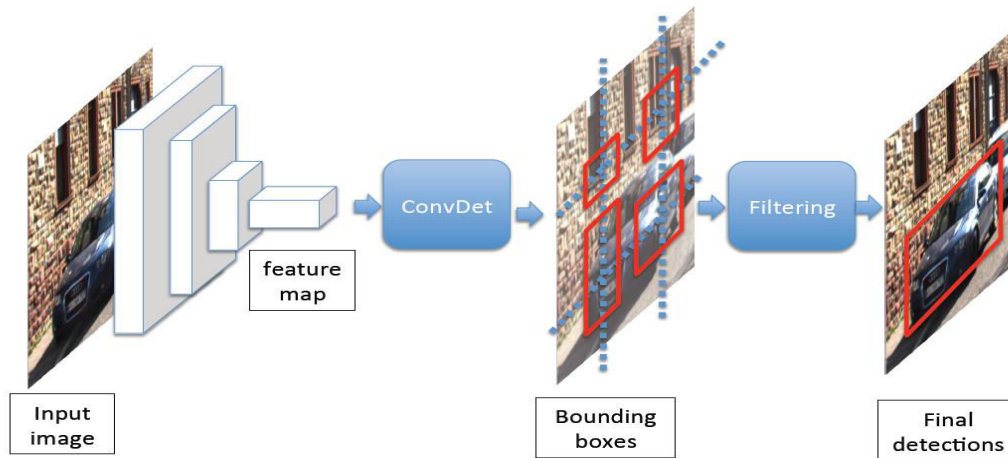
Για την πτυχιακή αυτή χρησιμοποιήθηκαν τα δεδομένα εικόνων καθώς και τα αρχεία βαθμονόμησης διότι περιγράφουν τι υπάρχει σε κάθε εικόνα αλλά και σε ποιες συντεταγμένες.

4.2 ΤΟ ΣΥΝΕΛΙΚΤΙΚΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ SQUEEZEDET

Έχοντας λοιπόν και τα απαραίτητα δεδομένα που χρειάζονται για την εκπαίδευση του δικτύου μας λείπει ένα τελευταίο συστατικό το οποίο είναι το ίδιο το δίκτυο. Για αυτή την εργασία χρησιμοποιήσαμε μια ήδη υπάρχουσα αρχιτεκτονική συνελικτικού δικτύου. Το δίκτυο αυτό ονομάστηκε SqueezeDet από τους δημιουργούς του Bichen Wu, Alvin Wan, Forrest Landola, Peter H. Jin και Kurt Keutzer [3]. Οι λόγοι που οδήγησαν στην επιλογή της συγκεκριμένης αρχιτεκτονικής είναι οι εξής:

- Ακρίβεια (Accuracy) : Σύμφωνα με τους δημιουργούς του η αρχιτεκτονική του δικτύου αποτελεί τεχνολογία αιχμής (state of the art) όσο αφορά τα αποτελέσματα τα οποία συλήχθηκαν μετά από ελέγχους στο σύνολο δεδομένων KITTI που χρησιμοποιείται στην συγκεκριμένη πτυχιακή.
- Μικρό μέγεθος μοντέλου : Το μέγεθος του μοντέλου που παράγει το SqueezeDet είναι μικρότερο σε σχέση με άλλες υπάρχουσες αρχιτεκτονικές δικτύων. Οι λόγοι για τους οποίους είναι επιθυμητή η ύπαρξη μικρού μεγέθους μοντέλου είναι η δυνατότητα αποδοτικής καταναλωμένης εκπαίδευσης του δικτύου, κατανάλωση λιγότερων πόρων επικοινωνίας όταν επρόκειται να γίνει μια εξ αποστάσεως αναβάθμιση στα μοντέλα των πελατών, λιγότερη κατανάλωση ενέργειας και πιο εύκολη ανάπτυξη ενός ενσωματωμένου συστήματος για οχήματα.
- Ενεργειακή απόδοση : Η εκπαίδευση του μοντέλου μπορεί να γίνεται σε υπολογιστές που καταναλώνουν μεγάλη υπολογιστική ισχύ αλλά το μοντέλο που παράγεται πρέπει να λειτουργεί σε ενσωματωμένους επεξεργαστές οχημάτων οι οποίοι διαθέτουν πολύ λιγότερη ισχύ.
- Ταχύτητα : Η αρχιτεκτονική του δικτύου έχει δομή που η ταχύτητες ανίχνευσης αντικειμένων χαρακτηρίζονται πραγματικού χρόνου. Οι απαιτούμενες ταχύτητες ανίχνευσης οδικών εμποδίων για αυτόνομα οχήματα πρέπει να είναι πραγματικού χρόνου ώστε να λαμβάνονται σωστές αποφάσεις από μεριάς του οχήματος και να εξασφαλίζεται η ασφάλεια.

Το δίκτυο αυτό είναι ικανό να αναγνωρίζει αντικείμενα και την θέση του κάθε αντικείμενου σε αρχεία εικόνων και είναι ιδανικό για χρήση στον τομέα της αυτόνομης οδήγησης.



Σχήμα 11 "Αναπαράσταση διαδικασίας ανίχνευσης αντικειμένων του συνελκτικού δικτύου SqueezeDet"

Ξεκινώντας, το δίκτυο αποτελείται από στρώματα συνελκτικού δικτύου τα οποία είναι :

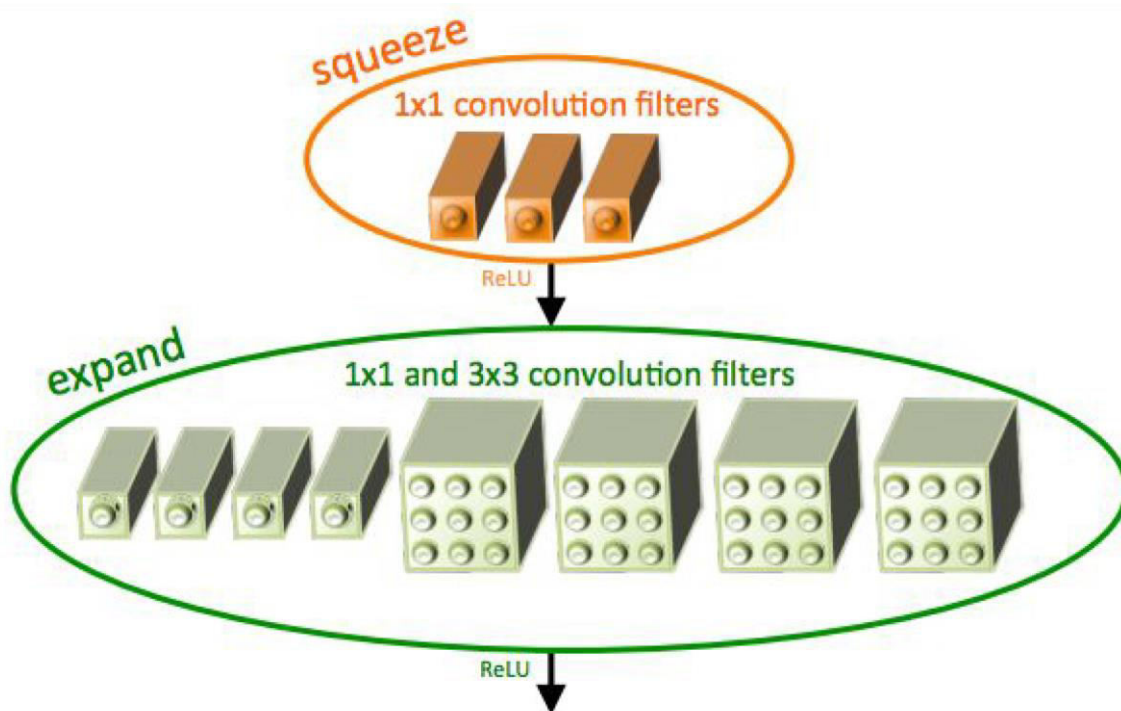
- Στρώμα εισόδου
- Συνελκτικό στρώμα
- Στρώμα RELU
- Συγκεντρωτικό στρώμα
- Πλήρως συνδεδεμένο

Να υπενθυμίσουμε ότι τα παραπάνω στρώματα μπορούν να χρησιμοποιηθούν είτε όλα είτε κάποια από αυτά με βασικά στρώματα το συνελκτικό, το συγκεντρωτικό και το πλήρως συνδεδεμένο. Η έξοδος αυτού του δίκτυο είναι ένας χάρτης χαρακτηριστικών (feature map) της εικόνας. Αυτός ο χάρτης έπειτα εισάγεται σε ένα συνελκτικό στρώμα το οποίο ονομάζεται ConvDet, το οποίο χρησιμοποιώντας τον χάρτη χαρακτηριστικών σαν είσοδο υπολογίζει ένα μεγάλο πλήθος οριοθετικών κουτιών (bounding box) καθώς και προσπαθούν να προβλέψουν την κλάση των αντικειμένων που βρίσκεται στο κάθε ένα από αυτά.

Αφού γίνουν οι υπολογισμοί χρησιμοποιείται το συνελκτικό δίκτυο SqueezeNet για το φιλτράρισμα των προβλέψεων. Το δίκτυο SqueezeNet [4] είναι μια έτοιμη αρχιτεκτονική συνελκτικού νευρωνικού δικτύου των Forrest N. Iandola, Song Han,

Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, για τον εντοπισμό αντικειμένων σε εικόνες.

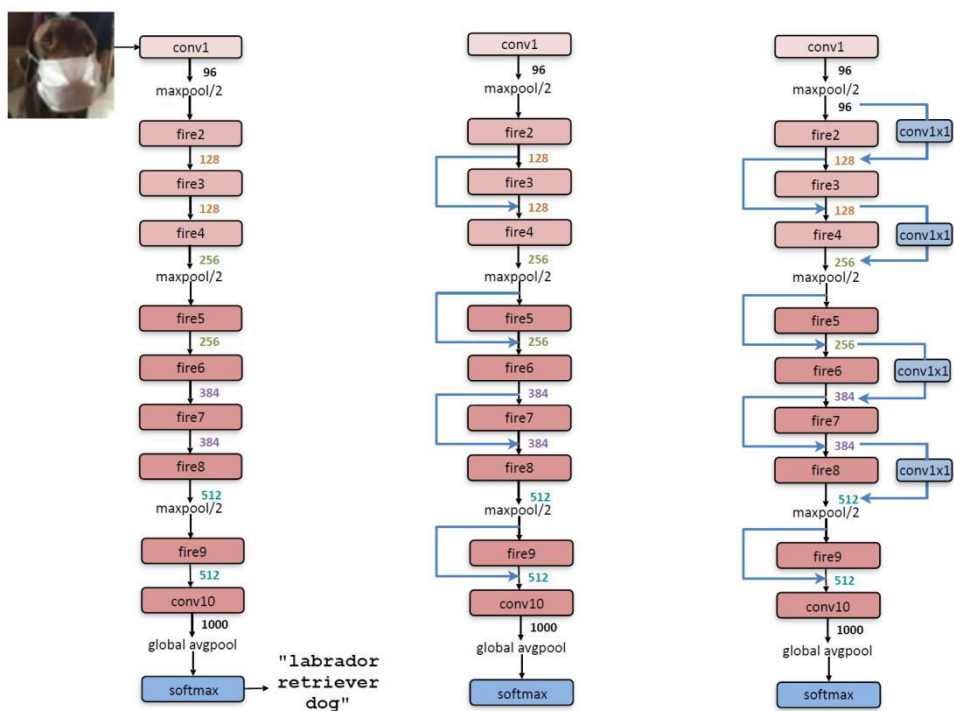
Πριν δούμε την αρχιτεκτονική του SqueezeNet είναι σημαντικό να μιλήσουμε για ένα συστατικό που χρησιμοποιείται σε αυτή, το ονομαζόμενο Fire module. Ένα Fire module αποτελείται από ένα συνελικτικό στρώμα το οποίο αποτελείται από φίλτρα / νευρώνες διάστασης 1×1 και συνεχίζει με ένα συνελικτικό στρώμα επέκτασης το οποίο αποτελείται από φίλτρα διάστασης 1×1 και 3×3 .



Σχήμα 12 "Αναπαράσταση του Fire Module"

Έχοντας εξηγήσει το Fire module μπορούμε πλέον να αναφερθούμε στην αρχιτεκτονική του δικτύου SqueezeNet. Το SqueezeNet αποτελείται από :

- Ένα συνελικτικό στρώμα
- Ένα συγκεντρωτικό επίπεδο
- Τρία Fire modules
- Ένα συγκεντρωτικό επίπεδο
- Τέσσερα Fire modules
- Ένα συγκεντρωτικό επίπεδο
- Ένα Fire module
- Ένα συνελικτικό στρώμα
- Ένα συγκεντρωτικό επίπεδο



Σχήμα 13 "Αναπαράσταση αρχιτεκτονικής του δικτύου SqueezeNet"

Αφού ολοκληρωθεί και αυτή η διαδικασία καταλήγουμε με τις εικόνες που έχουν μέσα τους τις τελικές προβλέψεις των κουτιών οριοθέτησης με τις πιθανότητες των κλάσεων των αντικειμένων που υπάρχουν σε αυτά. Όπως και σε ένα πραγματικό σενάριο έτσι και άδω περά μπορεί σε μια εικόνα να έχουμε παραπάνω από ένα αντικείμενο άρα και παραπάνω προβλέψεις.

4.3 ΕΝΑΛΛΑΚΤΙΚΑ ΜΟΝΤΕΛΑ ΕΝΤΟΠΙΣΜΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ

Το δίκτυο SqueezeDet δεν είναι η μόνη αρχιτεκτονική δικτύου που αποσκοπεί στον εντοπισμό αντικειμένων. Υπάρχουν και άλλες αρχιτεκτονικές τις οποίες θα δούμε αναφορικά σε αυτό το κεφάλαιο.

- R-CNN [5] : Το δίκτυο R-CNN είναι η πρώτη αρχιτεκτονική δικτύου που δημιουργήθηκε από τους Ross Girshick, Jeff Donahue, Trevor Darrell και Jitendra Malik για τον τομέα του εντοπισμού και αναγνώρισης αντικειμένων σε δεδομένα εικόνων. Ο αλγόριθμος του δικτύου αποτελείται από τρία βασικά βήματα :
 1. Σάρωση της εικόνας που εισάγεται χρησιμοποιώντας τον αλγόριθμο Επιλεκτικής Αναζήτησης (Selective Search), ο οποίος δημιουργεί περίπου 2000 προτάσεις περιοχής.
 2. Εκτέλεση ενός συνελκτικού δικτύου για κάθε μια από τις προτάσεις περιοχής που δημιουργήθηκαν.
 3. Εισαγωγή τις κάθε εξόδου του συνελκτικού δικτύου σε μια Μηχανή Διανυσμάτων Υποστήριξης (Support Vector Machine – SVM) για ταξινόμηση της περιοχής και σε έναν γραμμικό παλινρομητή για καλύτερη οριοθέτηση των αντικειμένων που βρίσκονται στο συγκεκριμένο πλαίσιο.
- Fast R-CNN [6] : Αποτελεί συνέχεια της προηγούμενης αρχιτεκτονικής, από έναν εκ των δημιουργών της, Ross Girshick, η οποία όμως είναι βελτιωμένη όσο αφορά την ταχύτητα. Αυτό οφείλεται σε δύο παράγοντες :
 1. Αντί για δημιουργία προτάσεων περιοχών το Fast R-CNN κάνει εξαγωγή χαρακτηριστικών για κάθε εικόνα και έτσι η εκτέλεση του συνελκτικού δικτύου γίνεται μια φορά για ολόκληρη την εικόνα.
 2. Αντικατάσταση της Μηχανής Διανυσμάτων Υποστήριξης με ένα συνελκτικό στρώμα Softmax που έχει ως αποτέλεσμα την επέκταση του συνελκτικού δικτύου όσο αφορά τις προβλέψεις.
- Faster R-CNN [7] : Το τελευταίο και καλύτερο δίκτυο της οικογένειας R-CNN, το Faster R-CNN, το οποίο δημοσιεύτηκε από τους Shaoqing Ren, Kaiming He, Ross Girshick και Jian Sun, προσθέτει στην αρχιτεκτονική του προηγούμενου μοντέλου ένα Δίκτυο Προτάσεων Περιοχής (Region Proposal Network – RPN). Το RPN λειτουργεί ως εξής :
 1. Στο τελευταίο στρώμα ενός αρχικού συνελκτικού δικτύου ένα συρόμενο παράθυρο (Sliding Window) διαστάσεως 3x3 μετακινείται στον χάρτη χαρακτηριστικών και τον χαρτογραφεί σε μία μικρότερη διάσταση.
 2. Για κάθε θέση του συρόμενου παραθύρου δημιουργεί πολλαπλές πιθανές περιοχές που βασίζονται σε κουτία οριοθέτησης σταθερής αναλογίας.

3. Κάθε πρόταση περιοχής αποτελείται από μια βαθμολογία "αντικειμενότητας" (Objectness Score), δηλαδή το πόσο πιθανή είναι η ύπαρξη αντικειμένου στην συγκεκριμένη περιοχή, καθώς και από 4 συντεταγμένες του πλαισίου οριοθέτησης της περιοχής.

Το Faster R-CNN κατάφερε να βελτιώσει την ταχύτητα αλλά και τα ποσοστά επιτυχίας των προηγούμενων αρχιτεκτονικών. Αν και δεν είναι η απλούστερη ούτε η γρηγορότερη μέθοδος για εντοπισμό αντικειμένων, παραμένει μέχρι και σήμερα μία από τις καλύτερες αρχιτεκτονικές στον τομέα της επίδοσης.

- R-FCN [8] : Ο τρόπος λειτουργίας του Faster R-CNN, δηλαδή αύξηση της ταχύτητας μεγιστοποιώντας τους κοινόχρηστους υπολογισμούς, ήταν και η πηγή έμπνευσης για τους Jifeng Dai, Yi Li, Kaiming He, Jian Sun και το δίκτυο που ονομάστηκε R-FCN (Region-based Fully Convolutional Network). Ο αλγόριθμος λειτουργίας του δικτύου είναι ο εξής :
 1. Εκτέλεση ενός Συνελικτικού Νευρωνικού Δικτύου (CNN) στην εικόνα εισόδου.
 2. Προσθήκη ενός συνελικτικού στρώματος με σκοπό την δημιουργία βαθμολογιών για τις περιοχές των χαρτών χαρακτηριστικών.
 3. Εκτέλεση ενός πλήρως συνδεδεμένου δικτύου προτάσεων περιοχής (Region Proposal Network – RPN) για την δημιουργία περιοχών ενδιαφέροντος (Regions of Interest – RoI).
 4. Διαίρεση της κάθε RoI σε αριθμό ίσο με τον αριθμό των χαρτών βαθμολογίας.
 5. Για κάθε περιοχή γίνεται έλεγχος των βαθμολογιών που δημιουργήθηκαν στο δεύτερο βήμα με σκοπό τον εντοπισμό θέσεων που ταιριάζουν με την θέση καποιου αντικειμένου. Για παράδειγμα, αν είμαι στο "πάνω αριστερό" κελί, θα τραβήξω τους χάρτες βαθμολογίας που αντιστοιχούν στην "πάνω αριστερή" γωνία ενός αντικειμένου και θα μετρήσω αυτές τις τιμές στην περιοχή RoI. Αυτή η διαδικασία επαναλαμβάνεται για κάθε κατηγορία.
 6. Μόλις κάθε από κυψέλη έχει τιμή "αντιστοίχισης αντικειμένου" για κάθε κλάση, υπολογίζεται ο μέσος όρος για τα κελιά για να υπολογιστεί ένα μοναδικό αποτέλεσμα ανά κατηγορία.
 7. Κατηγοριοποίηση των RoI με χρήση μιας συνάρτησης softmax.

Παρόλο που ο αλγόριθμος είναι περίπλοκος, το R-FCN είναι αρκετές φορές ταχύτερο από το Faster R-CNN και επιτυγχάνει συγκρίσιμη ακρίβεια.

- YOLO [9]: Το δίκτυο YOLO που δημιουργήθηκε από τους Joseph Redmon, Santosh Divvala, Ross Girshick και Ali Farhadi, πρότεινε μια απλή συνελικτική προσέγγιση νευρωνικών δικτύων, η οποία έχει τόσο μεγάλα

αποτελέσματα όσο και υψηλή ταχύτητα, επιτρέποντας για πρώτη φορά την ανίχνευση αντικειμένων σε πραγματικό χρόνο. Τα βήματα του δικτύου είναι τα εξής :

1. Διαχωρισμός της εικόνας εισόδου σε πολλαπλά πλέγματα, για παράδειγμα 7×7 .
2. Χαρακτηρισμός των δεδομένων εκπαίδευσης ως εξής: αν για παράδειγμα έχουμε K αριθμό ξεχωριστών αντικειμένων για εντοπισμό στα δεδομένα και η εικόνα έχει χωριστεί σε $N \times N$ πλέγματα, τότε ο φορέας εξόδου (vector) θα είναι μήκους $N \times N \times (K+5)$.
3. Δημιουργία ενός συνελκτικού νευρωνικού δικτύου στο οποίο ορίζεται σαν συνάρτηση απώλειας (loss function) η διαφορά μεταξύ ενεργοποιήσεων εξόδου και φορέων ετικετών (label vector). Βασικά, το μοντέλο προβλέπει την έξοδο όλων των πλεγμάτων σε ένα μόνο πέρασμα της εικόνας εισόδου μέσω του ConvNet.

Στην ουσία το δίκτυο για κάθε κελί υπολογίζει βαθμολογίες βεβαιότητας για ύπαρξη αντικειμένων σε αυτό. Όλα τα βήματα του δικτύου γίνονται με μία εκτέλεση του δικτύου, κάνοντας το YOLO δίκτυο εκτέλεσης πραγματικού χρόνου.

- SSD [10]: Τελευταίο μοντέλο, το οποίο προέρχεται από την ομάδα των Wei Liu, Dragomir Anguelov, Dimitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu και Alexander C. Berg, είναι το μοντέλο Ανίχνευσης Μονής Λήψης (Single-Shot Detector – SSD). Σε σχέση με προηγούμενα μοντέλα, τα οποία πρώτα δημιουργούν προτάσεις περιοχής και έπειτα κατηγοριοποιούν αυτές τις προτάσεις, το SSD εκτελεί αυτές τις λειτουργίες συγχρόνως. Πιο συγκεκριμένα:
 1. Πέρασμα της εικόνας εισόδου μέσω μιας σειράς συνελκτικών στρωμάτων, έχοντας ως αποτέλεσμα διάφορα σύνολα χαρτών χαρακτηριστικών σε διαφορετικές κλίμακες.
 2. Για κάθε θέση σε κάθε έναν από αυτούς τους χάρτες χαρακτηριστικών, γίνεται χρήση ενός συνελκτικού φίλτρου διάστασης 3×3 για την αξιολόγηση ενός μικρού συνόλου προεπιλεγμένων πλαισίων οριοθέτησης. Αυτά τα προκαθορισμένα πλαίσια οριοθέτησης είναι ισοδύναμα με τα κιβώτια αγκύρωσης του Faster R-CNN.
 3. Για κάθε κιβώτιο γίνεται ταυτόχρονα πρόβλεψη της μετατόπισης του κουτιού οριοθέτησης και των πιθανοτήτων των κλάσεων.
 4. Κατά την διάρκεια εκπαίδευσης του δικτύου, γίνεται ταίριασμα των “πραγματικών” κουτιών με τα κουτιά που εντοπίζονται βάσει του IoU. Οι καλύτερες προβλέψεις χαρακτηρίζονται ως “θετικές”, όπως και όλες οι προβλέψεις που έχουν $\text{IoU} > 0,5$.

Τελικώς το SSD δεν διαφέρει πολύ από τα R-CNN και R-FCN. Απλώς παραβλέπει το βήμα της δημιουργίας προτάσεων περιοχής. Σε συνδιασμό με την εκτέλεση ταυτόχρονη εκτέλεση των λειτουργιών, το δίκτυο είναι ταχύτερο από τα προαναφερθέντα δίκτυα.

4.4 ΕΚΠΑΙΔΕΥΣΗ ΤΟΥ ΔΙΚΤΥΟΥ

Αφού είδαμε όλα τα απαραίτητα συστατικά του δικτύου που χρησιμοποιείται σε αυτή την πτυχιακή θα αναφέρουμε τα χαρακτηριστικά και τα αποτελέσματα του πειράματος που διεξήχθη με την αρχιτεκτονική SqueezeDet και τις εικόνες του συνόλου δεδομένων του KITTI. Οι εικόνες του συνόλου δεδομένων είναι διαφορετικού μεγέθους, για αυτό τον λόγο μετασχηματίζονται όλες σε διαστάσεις ίσες με αυτές της μέγιστης σε διαστάσεις εικόνα οι οποίες είναι 1248 πλάτος και 384 σε ύψος. Στο πείραμά μας προσπαθήσαμε να εκπαιδεύσουμε το δίκτυο στον εντοπισμό αντικειμένων από τέσσερις κλάσεις του KITTI. Οι κλάσεις που επιλέχθηκαν είναι οι :

- Αυτοκίνητο
- Πεζός
- Ποδηλάτης
- Φορητό

Ο λόγος για τον οποίο επιλέξαμε τέσσερις και όχι όλες τις κλάσεις είναι επειδή είναι πολύ δύσκολο να εκπαιδεύσει κάποιος ένα δίκτυο να ξεχωρίζει πολλές κατηγορίες αντικειμένων όταν το εκπαιδεύει σε έναν τοπικό υπολογιστή και γενικά σε υπολογιστή με μικρή υπολογιστική ισχύ, τόσο όσο αφορά τις δυνατότητες της κύριας μονάδας επεξεργασίας αλλά και της μονάδας επεξεργασίας γραφικών. Επιλέξαμε το πείραμα να εκτελέσει διακόσιες χιλιάδες βήματα εκπαίδευσης και λάβαμε τα αποτελέσματα της εκπαίδευσης τα οποία είναι τα εξής :

- Loss
- Confidence loss
- Bounding Box loss


```
conf_loss: 25.304452896118164, bbox_loss: 0.8102636933326721, class_loss: 2.2493419647216797
2018-06-10 18:07:30.649876: step 0, loss = 28.37 (1.4 images/sec; 11.071 sec/batch)
conf_loss: 21.944080352783203, bbox_loss: 1.1044272184371948, class_loss: 2.2415249347686768
2018-06-10 18:07:48.588562: step 10, loss = 25.30 (2.5 images/sec; 6.007 sec/batch)
conf_loss: 10.46895980834961, bbox_loss: 0.7749626040458679, class_loss: 2.2028138637542725
2018-06-10 18:08:03.823038: step 20, loss = 13.45 (2.4 images/sec; 6.132 sec/batch)
conf_loss: 1.948520302772522, bbox_loss: 0.8473010659217834, class_loss: 1.5552387237548828
2018-06-10 18:08:19.026466: step 30, loss = 4.36 (2.5 images/sec; 6.102 sec/batch)
conf_loss: 1.906261920928955, bbox_loss: 1.4476215839385986, class_loss: 1.7453432083129883
2018-06-10 18:08:34.632964: step 40, loss = 5.11 (2.3 images/sec; 6.418 sec/batch)
conf_loss: 1.9913723468780518, bbox_loss: 0.9889236688613892, class_loss: 0.9069072604179382
2018-06-10 18:08:50.086050: step 50, loss = 3.89 (2.4 images/sec; 6.331 sec/batch)
```

Σχήμα 14 "Έξοδος αποτελεσμάτων κατά την διαδικασία εκπαίδευσης του δικτύου SqueezeDet"

Τα αποτελέσματα αυτά υπολογίζονταν για κάθε βήμα της εκπαίδευσης. Θέλουμε οι αριθμοί αυτοί να είναι όσο μικρότεροι το δυνατό, ιδανικά θα θέλαμε να είναι ίσοι με μηδέν. Το loss μεταφράζεται ως η διαφορά του εντοπισμού αντικειμένου στην εικόνα με την πραγματική θέση του αντικειμένου. Σαν εντοπισμό σε αυτό το σημείο εννοούμε και τον προσδιορισμό των συντεταγμένων της θέσης ενός αντικειμένου μέσα στην εικόνα αλλά και τον σωστό προσδιορισμό της κλάσης που ανήκει το κάθε αντικείμενο. Τα παραπάνω νούμερα λαμβάνονται εκτελώντας το αρχείο train.py

Για να καταλάβουμε το πόσο επιτυχημένη η μη ήταν η διαδικασία εκπαίδευσης πρέπει να εκτελέσουμε το αρχείο eval.py το οποίο είναι το αρχείο που θα αξιολογήσει το μοντέλο που παρήχθηκε από την εκπαίδευση του δικτύου. Από την διαδικασία αξιολόγησης λαμβάνουμε δύο μετρικές αξιολόγησης του μοντέλου :

- Recall - Ανάκληση
- Average Precision – Μέση ακρίβεια

Για να καταλάβουμε καλύτερα τι αντιπροσωπεύουν αυτοί οι αριθμοί θα χρησιμοποιήσουμε ένα παράδειγμα : Σε ένα πρόβλημα ταξινόμησης αντικειμένων σε δύο κλάσεις υπάρχουν οι παρακάτω περιπτώσεις

1. Αντικείμενα που ταξινομήθηκαν ως κλάση 1 ($\kappa=1$) και ανήκουν πράγματι στην κλάση 1 ($\tau = 1$) – Διαφορετικά, True Positive
2. Αντικείμενα που ταξινομήθηκαν ως κλάση 0 ($\kappa=0$) και ανήκουν πράγματι στην κλάση 0 ($\tau = 0$) – Διαφορετικά, True Negative
3. Αντικείμενα που ταξινομήθηκαν ως κλάση 1 ($\kappa=1$) και ανήκουν όμως στην κλάση 0 ($\tau = 0$) – Διαφορετικά, False Positive
4. Αντικείμενα που ταξινομήθηκαν ως κλάση 0 ($\kappa=0$) και ανήκουν όμως στην κλάση 1 ($\tau = 1$) – Διαφορετικά, False Negative

Σαν ανάκληση ορίζεται ορίζεται το ποσοστό των προτύπων που ανήκουν στην κλάση 1 και κατηγοριοποιούνται ως θετικά. Μαθηματικά η ανάκληση αναπαριστάται ως $\text{True Positive} / (\text{True Positive} + \text{False Negative})$. Ακρίβεια είναι το ποσοστό των προτύπων που κατηγοριοποιήθηκαν ως θετικά και είναι όντως θετικά. Η ακρίβεια αναπαριστάται ως $\text{True Positive} / (\text{True Positive} + \text{False Positive})$. Εκτελώντας το αρχείο eval.py τα αποτελέσματα που λαμβάνουμε είναι 0 για Average Precision και 0.0031959629941127 για Recall. Τα αποτελέσματα αυτά είναι πολύ κακά και μπορούμε να πούμε με σιγουριά ότι το μοντέλο που δημιουργήθηκε δεν είναι σε θέση να εντοπίσει κανένα από τα επιθυμητά αντικείμενα.

```
Mean average precision: 0.000
Analyzing detections...
Detection Analysis:
  Number of detections: 17835.0
  Number of objects: 17835.0
  Percentage of correct detections: 0.0031959629941127
  Percentage of localization error: 0.07216148023549201
  Percentage of classification error: 0.02012895991028876
  Percentage of background error: 0.9044575273338941
  Percentage of repeated detections: 5.6069526212503504e-05
  Recall: 0.0031959629941127
```

Σχήμα 15 "Έξοδος αποτελεσμάτων κατά την εκτέλεση του αρχείου αξιολόγησης"

Οι δημιουργοί του μοντέλου εκπαιδύσαν το δίκτυο για τον εντοπισμό τριών κλάσεων, αυτοκίνητο, ποδηλάτης και πεζός, και είχαν σαν Average Precision ποσοστό ίσο με 76,6% . Στην ιστοσελίδα του KITTI μπορούμε να δούμε τα αποτελέσματα που κατατάθηκαν για άλλα μοντέλα. Το μοντέλο Faster R-CNN έχει Average Precision ίσο με 70.2%. Η δεύτερη έκδοση του δικτύου YOLO [11] έχει 14.41 Average Precision εκπαιδευμένο με χρήση GPU των 3.5Ghz αλλά έχει και Average Precision ίσο με 62.78 με χρήση Titan X GPU και εκπαιδευμένο μόνο για την ανίχνευση αυτοκινήτων.

Σε αυτό το σημείο θα παραθέσουμε την υλοποίηση ορισμένων κομματιών του κώδικα στη γλώσσα προγραμματισμού Python.

```
conv = tf.nn.conv2d(
    inputs, kernel, [1, stride, stride, 1], padding=padding,
    name='convolution')
conv_bias = tf.nn.bias_add(conv, biases, name='bias_add')

if relu:
    out = tf.nn.relu(conv_bias, 'relu')
else:
    out = conv_bias
```

Σχήμα 16 "Αναπαράσταση υλοποίησης κώδικα Python συνελκτικού στρώματος και RELU"

```
def _pooling_layer(  
    self, layer_name, inputs, size, stride, padding='SAME'):  
    """Pooling layer operation constructor.  
  
    Args:  
        layer_name: layer name.  
        inputs: input tensor  
        size: kernel size.  
        stride: stride  
        padding: 'SAME' or 'VALID'. See tensorflow doc for detailed description.  
    Returns:  
        A pooling layer operation.  
    """  
  
    with tf.variable_scope(layer_name) as scope:  
        out = tf.nn.max_pool(inputs,  
                              ksize=[1, size, size, 1],  
                              strides=[1, stride, stride, 1],  
                              padding=padding)  
        activation_size = np.prod(out.get_shape().as_list()[1:])  
        self.activation_counter.append((layer_name, activation_size))  
    return out
```

Σχήμα 17 "Αναπαράσταση υλοποίησης κώδικα Python συγκεντρωτικού στρώματος"

```
def _fire_layer(self, layer_name, inputs, s1x1, e1x1, e3x3, stddev=0.01,  
               freeze=False):  
    """Fire layer constructor.  
  
    Args:  
        layer_name: layer name  
        inputs: input tensor  
        s1x1: number of 1x1 filters in squeeze layer.  
        e1x1: number of 1x1 filters in expand layer.  
        e3x3: number of 3x3 filters in expand layer.  
        freeze: if true, do not train parameters in this layer.  
    Returns:  
        fire layer operation.  
    """  
  
    s1x1 = self._conv_layer(  
        layer_name+'/squeeze1x1', inputs, filters=s1x1, size=1, stride=1,  
        padding='SAME', stddev=stddev, freeze=freeze)  
    e1x1 = self._conv_layer(  
        layer_name+'/expand1x1', s1x1, filters=e1x1, size=1, stride=1,  
        padding='SAME', stddev=stddev, freeze=freeze)  
    e3x3 = self._conv_layer(  
        layer_name+'/expand3x3', s1x1, filters=e3x3, size=3, stride=1,  
        padding='SAME', stddev=stddev, freeze=freeze)  
  
    return tf.concat([e1x1, e3x3], 3, name=layer_name+'/concat')
```

Σχήμα 18 "Αναπαράσταση υλοποίησης κώδικα Python του Fire module"

```
conv1 = self._conv_layer(  
    'conv1', self.image_input, filters=64, size=3, stride=2,  
    padding='SAME', freeze=True)  
pool1 = self._pooling_layer(  
    'pool1', conv1, size=3, stride=2, padding='SAME')  
  
fire2 = self._fire_layer(  
    'fire2', pool1, slx1=16, elx1=64, e3x3=64, freeze=False)  
fire3 = self._fire_layer(  
    'fire3', fire2, slx1=16, elx1=64, e3x3=64, freeze=False)  
pool3 = self._pooling_layer(  
    'pool3', fire3, size=3, stride=2, padding='SAME')  
  
fire4 = self._fire_layer(  
    'fire4', pool3, slx1=32, elx1=128, e3x3=128, freeze=False)  
fire5 = self._fire_layer(  
    'fire5', fire4, slx1=32, elx1=128, e3x3=128, freeze=False)  
pool5 = self._pooling_layer(  
    'pool5', fire5, size=3, stride=2, padding='SAME')  
  
fire6 = self._fire_layer(  
    'fire6', pool5, slx1=48, elx1=192, e3x3=192, freeze=False)  
fire7 = self._fire_layer(  
    'fire7', fire6, slx1=48, elx1=192, e3x3=192, freeze=False)  
fire8 = self._fire_layer(  
    'fire8', fire7, slx1=64, elx1=256, e3x3=256, freeze=False)  
fire9 = self._fire_layer(  
    'fire9', fire8, slx1=64, elx1=256, e3x3=256, freeze=False)  
  
# Two extra fire modules that are not trained before  
fire10 = self._fire_layer(  
    'fire10', fire9, slx1=96, elx1=384, e3x3=384, freeze=False)  
fire11 = self._fire_layer(  
    'fire11', fire10, slx1=96, elx1=384, e3x3=384, freeze=False)  
dropout11 = tf.nn.dropout(fire11, self.keep_prob, name='drop11')  
  
num_output = mc.ANCHOR_PER_GRID * (mc.CLASSES + 1 + 4)  
self.preds = self._conv_layer(  
    'conv12', dropout11, filters=num_output, size=3, stride=1,  
    padding='SAME', xavier=False, relu=False, stddev=0.0001)
```

Σχήμα 19 "Αναπαράσταση υλοποίησης κώδικα Python της αρχιτεκτονικής του δικτύου SqueezeDet"

```
class ModelSkeleton:
    """Base class of NN detection models."""

    def __init__(self, mc):
        self.mc = mc
        # a scalar tensor in range (0, 1]. Usually set to 0.5 in training phase and
        # 1.0 in evaluation phase
        self.keep_prob = 0.5 if mc.IS_TRAINING else 1.0

        # image batch input
        self.ph_image_input = tf.placeholder(
            tf.float32, [mc.BATCH_SIZE, mc.IMAGE_HEIGHT, mc.IMAGE_WIDTH, 3],
            name='image_input'
        )
        # A tensor where an element is 1 if the corresponding box is "responsible"
        # for detection an object and 0 otherwise.
        self.ph_input_mask = tf.placeholder(
            tf.float32, [mc.BATCH_SIZE, mc.ANCHORS, 1], name='box_mask')
        # Tensor used to represent bounding box deltas.
        self.ph_box_delta_input = tf.placeholder(
            tf.float32, [mc.BATCH_SIZE, mc.ANCHORS, 4], name='box_delta_input')
        # Tensor used to represent bounding box coordinates.
        self.ph_box_input = tf.placeholder(
            tf.float32, [mc.BATCH_SIZE, mc.ANCHORS, 4], name='box_input')
        # Tensor used to represent labels
        self.ph_labels = tf.placeholder(
            tf.float32, [mc.BATCH_SIZE, mc.ANCHORS, mc.CLASSES], name='labels')

        # IOU between predicted anchors with ground-truth boxes
        self.iou = tf.Variable(
            initial_value=np.zeros((mc.BATCH_SIZE, mc.ANCHORS)), trainable=False,
            name='iou', dtype=tf.float32
        )
    )
```

Σχήμα 20 "Αναπαράσταση υλοποίησης κώδικα Python μεταβλητών (Placeholders) εισόδου"

```
sess = tf.Session(config=tf.ConfigProto(allow_soft_placement=True))

saver = tf.train.Saver(tf.global_variables())
summary_op = tf.summary.merge_all()

ckpt = tf.train.get_checkpoint_state(FLAGS.train_dir)
if ckpt and ckpt.model_checkpoint_path:
    saver.restore(sess, ckpt.model_checkpoint_path)

summary_writer = tf.summary.FileWriter(FLAGS.train_dir, sess.graph)

init = tf.global_variables_initializer()
sess.run(init)

coord = tf.train.Coordinator()
```

Σχήμα 21 "Αναπαράσταση υλοποίησης κώδικα Python συνεδρίας του Tensorflow"

Σε αυτό το σημείο θα παρουσιαστούν μερικές εικόνες από ένα ήδη εκπαιδευμένο μοντέλο του SqueezeDet το οποίο είναι διαθέσιμο σαν demo από τους δημιουργούς του.



Σχήμα 22 "Παράδειγμα νο.1 εικόνας πριν και μετά τον εντοπισμό αντικειμένων"



Σχήμα 23 "Παράδειγμα νο.2 εικόνας πριν και μετά τον εντοπισμό αντικειμένων"



Σχήμα 24 "Παράδειγμα νο.3 εικόνας πριν και μετά τον εντοπισμό αντικειμένων"



Σχήμα 25 "Παράδειγμα νο.4 εικόνας πριν και μετά τον εντοπισμό αντικειμένων"

Όπως βλέπουμε ακόμα και στο ήδη έτοιμο μοντέλο το δίκτυο μπορεί να μην ανιχνεύσει σωστά κάποιο αντικείμενο, να εντοπίσει αντικείμενο με λανθασμένη

οριοθέτηση, να εντοπίσει κάποιο μη υπαρκτό αντικείμενο, αλλά μπορεί και να μην εντοπίσει καθόλου κάποιο πιθανό αντικείμενο.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο μιλήσαμε για τι πείραμα που διεξήχθη για την εκπόνηση αυτής της πτυχιακής χρησιμοποιώντας το σύνολο δεδομένων KITTI και το συνελκτικό νευρωνικό δίκτυο SqueezeDet. Εκπαιδεύσαμε το δίκτυο και μπορέσαμε να ανιχνεύσουμε αντικείμενα των κλάσεων που ορίσαμε για την εκπαίδευση σε εικόνες που το δίκτυο δεν είχε συναντήσει προηγουμένως κατά την εκπαίδευση του.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή την πτυχιακή εργασία ασχοληθήκαμε με το πώς μπορεί το ευρύ φάσμα της μηχανικής μάθησης να συμβάλει στην ανάπτυξη συστημάτων ικανά να εντοπίζουν αντικείμενα σε εικόνες. Δείξαμε πως μπορεί να γίνει αυτό χρησιμοποιώντας συνελκτικά νευρωνικά δίκτυα αλλά το να φτάσει στο σημείο ένα αυτοκίνητο να είναι σε θέση να οδηγεί χωρίς την επέμβαση του ανθρώπινου παράγοντα απέχει πολύ από το σημείο που είμαστε με αυτή την πτυχιακή. Ακολουθούν μερικά συμπεράσματα που προσκομίστηκαν με την παραπάνω πτυχιακή εργασία :

- Ο εξοπλισμός για την εκτέλεση πειραμάτων παίζει πολύ σημαντικό ρόλο. Το συγκεκριμένο δίκτυο εκπαιδεύτηκε σε δυο διαφορετικά μηχανήματα με διαφορετικές μονάδες γραφικών επεξεργασίας. Κατά την πρώτη διεξαγωγή του πειράματος, που έγινε σε υπολογιστή της εταιρίας Zenuity, χρησιμοποιώντας την κάρτα γραφικών της Nvidia Quadro 2000M η οποία έχει 192 πυρήνες CUDA το πρόγραμμα χρειάστηκε 6 ημέρες για να εκτελέσει 90.000 βήματα εκπαίδευσης. Την δεύτερη φορά χρησιμοποιώντας την κάρτα της Nvidia GeForce GTX 1050 με 640 πυρήνες CUDA το πρόγραμμα εκτέλεσε 200.000 επαναλήψεις σε 5 ημέρες. Μπορούμε να καταλάβουμε ότι η κάρτα γραφικών παίζει σημαντικό ρόλο στην εκπαίδευση δικτύων. Για αυτό και σε μεγάλες εταιρίες και ερευνητικά κέντρα χρησιμοποιούνται εξυπηρετητές που μπορεί να έχουν μέχρι και 40.000 πυρήνες CUDA.
- Ο αριθμός των κλάσεων επίσης παίζει πολύ σημαντικό ρόλο. Όσοι περισσότερες κλάσεις αναθέτεις σε ένα δίκτυο να μάθει κατά την διαδικασία εκπαίδευσης τόσο πιο δύσκολο είναι για αυτό το να έχει καλά αποτελέσματα. Για αυτό και το δίκτυο SqueezeDet εκπαιδεύτηκε από τους δημιουργούς του σε τρεις μόνο κλάσεις από τις οχτώ που έχει το σύνολο δεδομένων KITTI.
- Για καλύτερη απόδοση θα μπορούσαν να χρησιμοποιηθούν διαφορετικά δίκτυα το κάθε ένα από αυτά εκπαιδευμένο σε μια ή και δύο κλάσεις αντικειμένων. Ακόμα θα μπορούσαν να συνδυαστούν όχι απλά διαφορετικά δίκτυα, αλλά και δίκτυα που εκπαιδεύονται σε διαφορετικά δεδομένα όπως για παράδειγμα τα δεδομένα εικόνας και τα δεδομένα από φως λέιζερ που έχει το KITTI.
- Ο κλάδος του Deep Learning έχει βοηθήσει σημαντικά στον τομέα της μηχανικής μάθησης και είναι ικανός να λύσει προβλήματα που ο κλασικός προγραμματισμός δεν είναι σε θέση να λύσει.

- Το Deep Learning δεν εγκυιάται την επιτυχημένη λύση ενός προβλήματος. Στο πείραμα που εκτελέστηκε οι 200000 επαναλήψεις δεν ήταν αρκετές ώστε το δίκτυο να μπορέσει να ρυθμίσει τις εσωτερικές του παραμέτρους ώστε να μπορεί να εντοπίζει επιτυχημένα τα επιθυμητά αντικείμενα. Οι δημιουργοί του εκπαίδευσαν το δίκτυο για 1000000 επαναλήψεις, χωρίς να είναι ούτε γνωστό ούτε σίγουρο σε πιο στάδιο το δίκτυο θα εντοπίσει τις κατάλληλες ρυθμίσεις. Αυτό καθιστά τον κλάδο του Deep Learning ως μη εγκυήσιμο τρόπο επίλυσης προβλημάτων.
- Υπάρχουν πολλά ακόμα βήματα που πρέπει να γίνουν ώστε από την εκπαίδευση δικτύων να φτάσουμε στο σημείο να έχουμε όχημα το οποίο μπορεί να ανιχνεύει αντικείμενα που έχει στο δρόμο. Πρέπει να είναι σε θέση το όχημα να ανιχνεύει όχι απλές εικόνες που χρησιμοποιούμε σε αυτό το πείραμα, αλλά κινούμενες εικόνες όπως συμβαίνει σε ένα πραγματικό σενάριο οδήγησης. Οι ενέργειες που θα εκτελεί το όχημα μετά τον εντοπισμό είναι και αυτές ένα πολύ μεγάλο ξεχωριστό κεφάλαιο.

ΑΝΑΦΟΡΕΣ

- [1] <https://www.cs.toronto.edu/~kriz/cifar.html>
- [2] Andreas Geiger, Philip Lenz, Christoph Stiller and Raquel Urtasun. Vision meets Robotics : The KITTI Dataset
- [3] Bichen Wu, Alvin Wan, Forrest Iandola, Peter H. Jin, Kurt Keutzer, UC Berkeley, DeepScale. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving
- [4] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, DeepScale & UC Berkeley Stanford University. SQUEEZENET: ALEXNET-LEVEL ACCURACY WITH 50X FEWER PARAMETERS AND <0.5MB MODEL SIZE
- [5] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, UC Berkeley. Rich feature hierarchies for accurate object detection and semantic segmentation, Tech report (v5)
- [6] Ross Girshick. Fast R-CNN
- [7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
- [8] Jifeng Dai, Yi Li, Kaiming He, Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks
- [9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection
- [10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector
- [11] Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger