Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Εντοπισμός Αυτοκινήτων σε Τρισδιάστατα Νέφη Σημείων μέσω Συνελικτικών Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Του

Ευσταθίου Κώτση

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης

Καθηγητής ΕΜΠ

Αθήνα , Αύγουστος 2022

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Εντοπισμός Αυτοκινήτων σε Τρισδιάστατα Νέφη Σημείων μέσω Συνελικτικών Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Του

Ευσταθίου Κώτση

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31η Αυγούστου 2022.

……	……...……	……
Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης	Γεώργιος Στάμου
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ

Αθήνα , Αύγουστος 2022

……..

Κώτσης Ευστάθιος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

Περίληψη

Ο εντοπισμός αντικειμένων αποτελεί έναν από τους πιο δημοφιλείς τομείς της Όρασης Υπολογιστών. Ο τομέας αποσκοπεί στην κατασκευή μοντέλων, ικανά να διακρίνουν την κλάση ενδιαφέροντος από όλα τα υπόλοιπα αντικείμενα που εμφανίζονται. Ενώ ο εντοπισμός αντικειμένων σε δύο διαστάσεις έχει σημειώσει μεγάλες επιτυχίες, ο απευθείας εντοπισμός τρισδιάστατων αντικειμένων αποτελεί πολύ μεγαλύτερη πρόκληση. Ειδικότερα ο εντοπισμός τρισδιάστατων αντικειμένων σε Νέφη Σημείων αποκτά ιδιαίτερο ενδιαφέρον λόγω των εφαρμογών του σε οικιακά ρομπότ, στην εικονική πραγματικότητα και στα συστήματα αυτόνομης οδήγησης. Όσον αφορά τον τελευταίο τομέα, ο εντοπισμός αυτοκινήτων σε Νέφη Σημείων αποτελεί επιτακτική ανάγκη για την πρόοδο της αυτόνομης οδήγησης και την καθιέρωση της ως μία ασφαλή βιώσιμη εναλλακτική πρόταση στην οδήγηση. Για τον εντοπισμό των σημαντικών χαρακτηριστικών που υπάρχουν στα Νέφη Σημείων, ένα μεγάλο μέρος της βιβλιογραφίας στηρίζεται στην χρήση χειροκίνητων αναπαραστάσεων τους. Στην εργασία αυτή γίνεται χρήση μίας ολοκληρωμένης αρχιτεκτονικής εντοπισμού αντικειμένων σε τρεις διαστάσεις [73] καθώς και δοκιμές τροποποίησης της, με στόχο τον εντοπισμό αυτοκινήτων, αντλούμενα από τρισδιάστατα Νέφη Σημείων, τα οποία έχουν συλλεχθεί από συσκευές λέιζερ lidar (Light Detection And Ranging) οι οποίες προσφέρουν την δυνατότητα υπολογισμού του βάθους των αντικειμένων, κρίσιμο στοιχείο που στερούνται οι τεχνικές χειρισμού εικόνων. Η συγκεκριμένη αρχιτεκτονική κάνει χρήση τριών υποσυστημάτων. Πιο συγκεκριμένα, ένα Σύστημα Εκμάθησης Χαρακτηριστικών, το οποίο αφού χωρίσει το Νέφος Σημείων σε τρισδιάστατα Βόξελ, μεταμορφώνει το σύνολο των σημείων κάθε Βόξελ σε μία αραιή αναπαράσταση χαρακτηριστικών, μέσω μίας Αλυσίδας Επιπέδων Κωδικοποίησης. Στην συνέχεια η αραιή αναπαράσταση του Νέφους σημείων τροφοδοτεί επιπλέον Συνελικτικά Επίπεδα και τέλος ένα Δίκτυο Προτάσεων Περιοχής που στηρίζεται σε συνελικτικά νευρωνικά δίκτυα, το οποίο κάνει τις τελικές προβλέψεις για την ύπαρξη ή όχι αυτοκινήτων στην εκάστοτε υποπεριοχή του Νέφους, ενώ ταυτόχρονα επιχειρεί να τελειοποιήσει την θέση και τον προσανατολισμό των επιτυχημένων προβλέψεων. Η εκπαίδευση του μοντέλου είναι επιβλεπόμενη και γίνεται με ετικέτες που δίνονται για συγκεκριμένα χαρακτηριστικά του Cloud, ενώ η συνάρτηση απωλειών αποτελείται από τις απώλειες για την κατηγοριοποίηση σε κλάση ενδιαφέροντος ή όχι και τις απώλειες για την διόρθωση των προβλέψεων. Τα πειράματα, σε κατάλληλο σύνολο δεδομένων αποτελούμενο από Νέφη Σημείων οδικού ιστού, δείχνουν πως το μοντέλο είναι ικανό να κατασκευάζει την αναπαράσταση των αντικειμένων στον χώρο και να μαθαίνει την δομή τους, κατορθώνοντας γενικά προβλέψεις υψηλής ποιότητας, οι οποίες περιορίζονται κατά βάση από την ποικιλομορφία του προσανατολισμού των αυτοκινήτων και την δυσκολία διάκρισης παρόμοιων οχημάτων.

Λέξεις-Κλειδιά: Εντοπισμός Τρισδιάστατων Αντικειμένων, Νευρωνικά Δίκτυα, Συνελικτικά Νευρωνικά Δίκτυα, Μηχανική Μάθηση, Όραση Υπολογιστών, Βαθιά Μάθηση, Αυτόνομη οδήγηση, Lidar, Νέφη Σημείων, Βόξελ.

Abstract

The task of Object Detection is arguably one of the most active tasks in the field of Computer Vision. The task is primarily concerned with the design and implementation of models with the ability to detect instances of specific classes. In contrast to 2d object detection, which has enjoyed great success over the years, 3d detection constitutes a much bigger challenge but also bears great interest due to the numerous applications in household robots, autonomous driving and virtual reality as well. Specifically for the autonomous driving task, the detection of cars in Point Clouds has immense scientific value and is a deciding factor for the success of autonomous driving as a lasting and safe driving alternative. For the encoding of the important features in a Point Cloud, a great number of methods make use of handcrafted features. In this thesis a complete end-to-end 3d object detection architecture is deployed [73], as well as modifications, in order to detect cars from Point Cloud representations, collected from 'lidar' (Light Detection And Ranging) laser devices, which help encode the important element of 'depth' as well, in contrast to image based methods. Specifically, the architecture used makes use of three subsystems. Firstly, a Feature Learning Network is responsible for the partitioning of the Point Cloud into voxels and the encoding of the points of each voxel into feature representations, with the help of a chain of Feature Encoding Layers. This sparse feature representation is then feed to some Convolutional Layers and then a Region Proposal Network based upon convolutional neural networks, which makes the final predictions and regresses the predicted targets to improve the position and the orientation of the predictions. Model training is supervised and is achieved with labels for specific parameters that accompany the Point Clouds, whereas the Loss Function is the sum of the Classification Loss (detected or not) and the Regression Loss. The experiments are conducted on a specific Point Cloud dataset which encodes roads into Point Cloud representations. It is demonstrated that the model is able to construct a high level representation of the objects in the Point Cloud and learn their shape and features, thus achieving high quality predictions, limited mostly by the the multitude of possible car orientations and the existing and inevitable resemblance between many road trucks.

Keywords: 3D Object Detection, Neural Networks, Convolutional Neural Networks, Machine Learning, Computer Vision, Deep Learning, Autonomous Driving, Lidar, Point Clouds, Voxel.

Ευχαριστίες

Με την διπλωματική αυτή εργασία περατώνεται η πολυετής φοίτηση μου στην σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Στο πλαίσιο αυτής της σταδιοδρομίας, στην πορεία της οποίας απέκτησα γνώσεις καίριες για την ανάπτυξη μου τόσο πνευματικά όσο και ηθικά, θεωρώ πως είναι πρέπον να εκφράσω τις θερμές μου ευχαριστίες στα άτομα εκείνα που με στήριξαν στις προπτυχιακές μου σπουδές.

Πρωτίστως θα ήθελα να ευχαριστήσω τον κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, επιβλέποντα καθηγητή στον τομέα της Τεχνολογίας Πληροφορικής και Υπολογιστών, για την δυνατότητα που μου προσέφερε να ασχοληθώ με ένα θέμα που δεν έχει μόνο εξαιρετικό τεχνολογικό ενδιαφέρον αλλά και πολύ μεγάλη πρακτική αξία. Επίσης θα ήθελα να τον ευχαριστήσω για την άμεση ανταπόκριση του σε οποιαδήποτε απορία μου και την εξαίρετη επικοινωνία. Εν συνεχεία θα ήθελα να ευχαριστήσω τον κ. Γεώργιο Σιόλα για την αμεσότητα στην μεταξύ μας επικοινωνία και την καθοδήγηση στην επιλογή και την μελέτη του αντικειμένου. Αναμφίβολα σημαντική είναι επίσης η βοήθεια όλων των καθηγητών μου όλα αυτά τα χρόνια που με βοήθησαν με τις γνώσεις τους και την μεταδοτικότητα τους. Επιπλέον δεν μπορεί να παραληφθεί η σημαντική ενθάρρυνση των φίλων μου και συμφοιτητών μου με τους οποίους μοιράστηκα κοινά προβλήματα και ανησυχίες. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου και τις αδερφές μου για την συμπαράστασή τους όλα αυτά τα χρόνια. Χωρίς εκείνους η σταδιοδρομία αυτή δεν θα ήταν εφικτή.

Κώτσης Ευστάθιος,

Αθήνα, Αύγουστος 2022

Περιεχόμενα

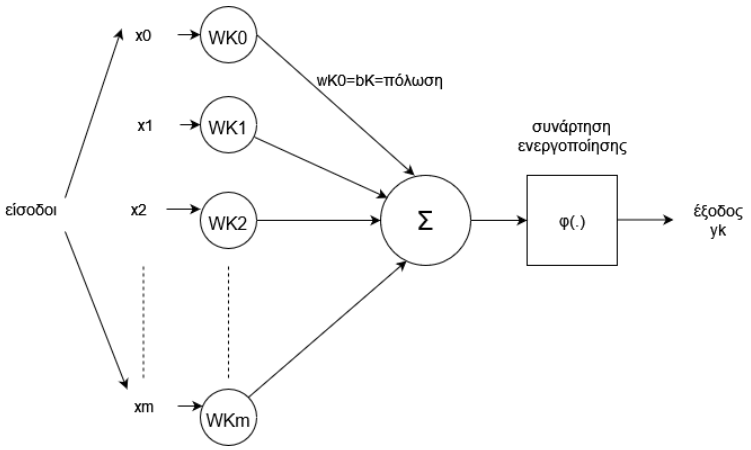
Κεφάλαιο 1

Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια σύντομη παρουσίαση της ιστορίας της μηχανικής μάθησης και των δημοφιλέστερων τομέων της.

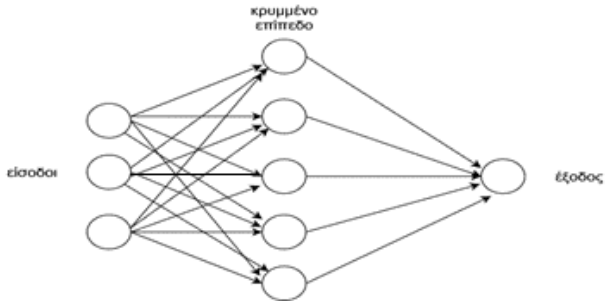
1.1 Μηχανική Μάθηση

Μηχανική μάθηση είναι ο τομέας της Τεχνητής Νοημοσύνης της Επιστήμης Υπολογιστών που ασχολείται με την προσπάθεια κατανόησης και υλοποίησης μεθόδων και αλγορίθμων που «μαθαίνουν», με την έννοια ότι τροφοδοτούνται με δεδομένα τα οποία ονομάζονται δεδομένα εκπαίδευσης και καλούνται να ανακαλύψουν τα υπάρχοντα πρότυπα και χαρακτηριστικά των δεδομένων με στόχο να λάβουν αποφάσεις ή να κάνουν προβλέψεις για μελλοντικά δεδομένα. Καλούνται λοιπόν να παράγουν αποτελέσματα χωρίς τον ρητό προγραμματισμό τους ενώ ταυτόχρονα βελτιώνεται η γνώση τους ανάλογα με τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση τους. Η αναγνώριση αυτών των προτύπων και χαρακτηριστικών των δεδομένων συντελεί στην λεγόμενη «Μάθηση»». Η Μηχανική Μάθηση έχει πλούσια ιστορία εφόσον συνδέεται στενά με την ιστορία και την πρόοδο της Τεχνητής Νοημοσύνης. Από το 1949 και μέσα στα επόμενα χρόνια ο επιστήμονας υπολογιστών Arthur Samuel ανέπτυξε το 1ο υπολογιστικό σύστημα που έπαιζε το παιχνίδι ‘Checkers’. Ο υπολογιστής εφαρμόζοντας τον αλγόριθμο MinMax σε συνδυασμό με το Alpha-Beta Pruning μπορούσε να θυμάται τεχνικές και κινήσεις τις οποίες επαναλάμβανε ή απέρριπτε με βάση μια συνάρτηση ανταμοιβής που λειτουργούσε πιθανοτικά. Η επιλογή κίνησης γινόταν με στόχο την μεγιστοποίηση της ανταμοιβής του. Επίσης ο υπολογιστής αποθήκευε στην μνήμη και μάθαινε κινήσεις τις οποίες είχε επαναλάβει πολλές φορές, αποτελώντας ένα ευφυές υπολογιστικό σύστημα με σχετική ικανότητα μάθησης [1]. Στην συνέχεια το 1957 ο Φρανκ Ρόζενμπλατ υλοποίησε το πρώτο υπολογιστικό σύστημα με την ικανότητα να μαθαίνει , το οποίο προσομοίωνε τον ανθρώπινο εγκέφαλο στην δομή του. Το λεγόμενο ‘Perceptron’ είχε εφευρεθεί το 1943 από τους McCulloch και Pitts και είναι ένας αλγόριθμος ταξινόμησης που μπορεί να αποφασίσει αν ένα δεδομένο εισόδου ανήκει ή όχι σε μία συγκεκριμένη κλάση. Το 1ο Perceptron σχεδιάστηκε για την ανάλυση και αναγνώριση κλάσεων σε εικόνες [2,3,4].



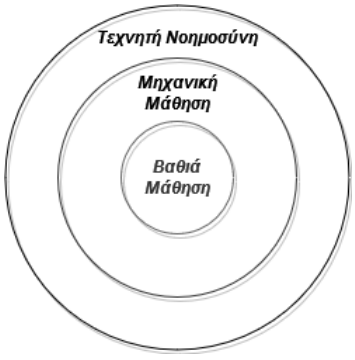
Σχήμα 1.1. *Μοντέλο νευρώνα McCulloch-Pitts.Το μοντέλο νευρώνα προσομοίωνε τους νευρώνες στον εγκέφαλο και αποτελούνταν από τις εισόδους ,τα βάρη με τα οποία πολλαπλασιάζονταν αυτές,το άθροισμα των γινομένων αυτών και μια συνάρτηση ενεργοποίησης η οποία δεχόταν το άθροισμα αυτό ως είσοδο.*

Το 1960 έγινε η ανακάλυψη και χρήση πολυεπίπεδων Perceptron (MLP).Η εισαγωγή 2 ή και περισσότερων επιπέδων προσέφερε μεγαλύτερη δυνατότητα επεξεργασίας των δεδομένων. Τα ενδιάμεσα επίπεδα αποκαλούνται κρυμμένα επίπεδα (hidden layers [5]. Η χρήση των MLP οδήγησε στα feedforward νευρωνικά δίκτυα και στην οπισθοδιάδοση (backpropagation) [6,7].Η οπισθοδιάδοση έδωσε την δυνατότητα στα hidden layers να προσαρμόζουν τα βάρη τους ανάλογα με το λάθος στην πρόβλεψη με στόχο την καλύτερη εκπαίδευση τους.



Σχήμα 1.2. *Μοντέλο MLP.Υπάρχει ένα δεύτερο κρυμμένο επίπεδο πέρα από το επίπεδο εισόδου και εξόδου, ικανό να εκφράσει πιο σύνθετες σχέσεις μεταξύ εισόδου και εξόδου.*

Στην συνέχεια το 1967 η τεχνική του 'Nearest Neighbor' για την ταξινόμηση δεδομένων για τα οποία δεν υπάρχει γνώση ,στην κλάση των κοντινότερων γειτονικών δεδομένων έδειξε ότι η συγκεκριμένη υπόθεση είναι φραγμένη από το διπλάσιο της Bayesian πιθανότητας λάθους [8, 9]. Αυτό αποτέλεσε την βάση για την αναγνώριση προτύπων στα δεδομένα (Pattern Recognition). Τα τελευταία χρόνια των δεκαετιών 1970 και του 1980 η έρευνα στα νευρωνικά δίκτυα εγκαταλείφθηκε για αρκετά χρόνια ενώ άρχισε να αναπτύσσεται ξανά με την ανάπτυξη του διαδικτύου και την όλο και μεγαλύτερη διαθεσιμότητα δεδομένων μετά το 1990. Η έρευνα εστιάστηκε περισσότερο στα διαθέσιμα δεδομένα και στην μάθηση υπολογιστικών συστημάτων μέσω της ανάλυσης τεράστιων ποσοτήτων δεδομένων και εκτίμησης των αποτελεσμάτων. Την δεκαετία του 1990 παρουσιάστηκε η τεχνική του «boosting». Σύμφωνα με αυτή συνδυάζονται πολλοί αδύναμοι ταξινομητές (ταξινομητές που δεν έχουν μεγάλη συσχέτιση με την πραγματική κλάση) με έναν τελικό δυνατό ταξινομητή. Στην συνέχεια γίνεται εκτίμηση των βαρών τους και αναπροσαρμόζονται ανάλογα με τα δεδομένα που κατηγοριοποίησαν λανθασμένα. Το 1997 υπήρχε μεγάλη ανάπτυξη στον τομέα της αναγνώρισης ομιλίας , με τον σχεδιασμό νευρωνικών δικτύων που ονομάζονται LSTM (long short-term memory) [10].Τα δίκτυα αυτά ανήκουν στην κατηγορία των Recurrent Neural Networks, όπου οι συνδέσεις μεταξύ των κόμβων σχηματίζουν γράφο. Τα LSTM μπορούν να κατηγοριοποιούν να επεξεργάζονται και να κάνουν προβλέψεις βασιζόμενα σε δεδομένα διακριτού χρόνου. Το 1998 ο υπερυπολογιστής Deep Blue της IBM νίκησε τον πρωταθλητή στο σκάκι Γκάρι Κασπάροφ αξιοποιώντας την δυνατότητα να προβλέπει τις κινήσεις σε ένα μεγάλο βάθος κινήσεων. Το 2006 η εκτίμηση των αλγορίθμων αναγνώρισης προσώπων στον διαγωνισμό Face Recognition Grand Challenge έδειξε ότι οι νέοι αλγόριθμοι πετύχαιναν 10 φορές μεγαλύτερη ακρίβεια από τους αντίστοιχους αλγόριθμους του 2002. Το 2007 τα LSTM ξεπέρασαν παραδοσιακούς αλγόριθμους για την αναγνώριση ομιλίας. Το 2012 η Google ανέπτυξε έναν αλγόριθμο για τον εντοπισμό βίντεο με γάτες και το 2014 η Facebook ανέπτυξε το DeepFace έναν αλγόριθμο ικανό να αναγνωρίζει άτομα σε φωτογραφίες με την ίδια ακρίβεια με τους ανθρώπους [11].



Εικόνα 1.1

Κατηγορίες Μηχανικής Μάθησης :

Υπάρχουν οι ακόλουθες κατηγορίες αλγορίθμων μηχανικής μάθησης που διακρίνονται σε σχέση με τον τρόπο που γίνεται η εκμάθηση του συστήματος και ανάλογα με το αν στοχεύουν στην πρόβλεψη αποτελεσμάτων για τα οποία γνωρίζουμε την πραγματική τιμή τους, ή στην η κατηγοριοποίηση-συσταδοποίηση των δεδομένων.

- Επιβλεπόμενη μάθηση (Supervised learning): Το μοντέλο δέχεται ως είσοδο ένα σύνολο δεδομένων και τις πραγματικές τιμές (τις ετικέτες) που προσπαθεί να προβλέψει και καλείται να μάθει την συσχέτιση μεταξύ δεδομένων και ετικετών ώστε οι προβλέψεις ιδανικά να ταυτίζονται με τις πραγματικές τιμές [12, 13].
- Μη επιβλεπόμενη μάθηση (Unsupervised learning): Τα δεδομένα που εισάγονται στο μοντέλο δεν είναι κατηγοριοποιημένα (χωρίς ετικέτες) και το μοντέλο καλείται να ανακαλύψει την δομή και τις υπάρχουσες συσχετίσεις τους ώστε να τα κατηγοριοποιήσει. Οι συσχετίσεις και τα χαρακτηριστικά αυτά μπορεί να είναι χρήσιμα για την εκπαίδευση στην συνέχεια μοντέλων επιβλεπόμενης μάθησης που λαμβάνουν υπόψιν την εσωτερική αυτή αναπαράσταση των δεδομένων.
- Ημιεπιβλεπόμενη μάθηση (Semisupervised learning): Σε αυτήν τη μορφή της μάθησης χρησιμοποιούνται: ένα μικρό μέρος από δεδομένα με ετικέτες και ένα μεγάλο μέρος από δεδομένα χωρίς ετικέτες, με βάση τα οποία το μοντέλο πρέπει να μάθει και να κάνει προβλέψεις σε νέα δεδομένα [15, 16].
- Ενισχυτική μάθηση (Reinforcement learning): Ο τομέας αυτός ασχολείται με τον τρόπο με τον οποίον ευφυείς πράκτορες πρέπει να δράσουν σε ένα περιβάλλον ώστε να μεγιστοποιήσουν την συνάρτηση ανταμοιβής τους. Η ενισχυτική μάθηση δεν χρειάζεται ετικετοποιημένα ζευγάρια από δεδομένα εισόδου και εξόδου και δεν αποσκοπεί στην βέλτιστη ακρίβεια παρά στην ισορροπία μεταξύ εξερεύνησης-ανακάλυψης νέας γνώσης και στην αξιοποίηση της υπάρχουσας [17].

1.2 Όραση υπολογιστών

Η όραση υπολογιστών είναι ένας διεπιστημονικός τομέας της μηχανικής μάθησης, που επιχειρεί να δώσει στους υπολογιστές την δυνατότητα να αντλούν πληροφορία από εικόνες, βίντεο και γενικά μορφές οπτικών δεδομένων και να μετασχηματίζουν αυτήν την πληροφορία σε γνώση χρήσιμη για τους υπολογιστές. Στόχος του τομέα αυτού είναι να προσομοιώσει και να αυτοματοποιήσει τις λειτουργίες του ανθρώπινου οπτικού συστήματος. Σημαντικοί τομείς της όρασης υπολογιστών είναι οι παρακάτω [18] :

- Κατηγοριοποίηση εικόνων (image classification): Στον τομέα αυτόν εικόνες δίνονται σε ένα μοντέλο και αυτό καλείται να βρει σε ποια κατηγορία ανήκουν.
- Σημασιολογική κατάτμηση (semantic segmentation): Ο τομέας αυτός ασχολείται με την συσταδοποίηση (clustering) τμημάτων της εικόνας τα οποία ανήκουν στο ίδιο αντικείμενο-στην ίδια κλάση. Κάθε pixel της εικόνας κατηγοριοποιείται ώστε να ανήκει σε μία συγκεκριμένη κατηγορία.
- Εκτίμηση βάθους (depth estimation): Ο τομέας αυτός ασχολείται με την μέτρηση απόστασης κάθε pixel σε σχέση με την κάμερα. Το βάθος αντλείται από εικόνες μιας λήψης ή από πολλές διαφορετικές λήψεις.
- Αναγνώριση οπτικού χαρακτήρα (optical character recognition): Ο τομέας αυτός επιχειρεί την αναγνώριση χειρόγραφων ή εκτυπωμένων χαρακτήρων ή χαρακτήρων από φωτογραφίες και την μετατροπή τους σε μορφές κωδικοποιημένες για τον υπολογιστή.
- Κατάτμηση Στιγμιοτύπων (instance segmentation): Ο τομέας αυτός ασχολείται με τον εντοπισμό και την διάκριση αντικειμένων ενδιαφέροντος που εμφανίζονται σε εικόνες.
- Αναγνώριση Δράσης (action recognition): Στην τρέχουσα έρευνα του τομέα αυτού βρίσκεται η κατανόηση των ανθρωπίνων δραστηριοτήτων μέσα σε βίντεο.
- Εντοπισμός Αντικειμένου (object detection): Σύμφωνα με τον κλάδο αυτό επιχειρείται η αναγνώριση στιγμιοτύπων αντικειμένων συγκεκριμένων κλάσεων μέσα σε μία εικόνα. Οι μέθοδοι αναγνώρισης χωρίζονται σε: ενός σταδίου[19] (YOLO, SSD, RetinaNet) και σε 2 σταδίων οι οποίες είναι πιο αργές αλλά πιο ακριβείς (Faster R-CNN,Mask R-CNN, Cascade R-CNN) [20]. Υποκατηγορία της αναγνώρισης αντικειμένων είναι ο τρισδιάστατος εντοπισμός αντικειμένου, στον οποίο τομέα γίνεται ο υπολογισμός των κυτίων αναγνώρισης αντικειμένων του φυσικού κόσμου με την βοήθεια τρισδιάστατων δεδομένων από αισθητήρες, όπως συσκευές laser (lidar).
- Αυτόνομα Συστήματα Οδήγησης (Autonomous Driving): Ο τομέας αυτός είναι επιφορτισμένος με το έργο της αυτόνομης οδήγησης αυτοκινήτων χωρίς την παρουσία ανθρώπινου οδηγού. Είναι στενά συνδεδεμένος με τον τρισδιάστατο εντοπισμό αντικειμένων και τη σημασιολογική κατάτμηση εφόσον μεγάλη πρόκληση στα αυτόνομα αυτοκίνητα είναι η κατανόηση ενός δυναμικού οδικού περιβάλλοντος και η αναγνώριση όλων των εμποδίων που μπορεί να συναντηθούν (αυτοκίνητα, άλλα οχήματα, πεζοί,

ποδηλάτες). Η μελέτη και εφαρμογή μεθόδων για τον εντοπισμό αυτοκινήτων στο δρόμο αποτελεί και τον στόχο αυτής της μελέτης.

1.3 Αυτόνομα Αυτοκίνητα

Αυτόνομο ονομάζεται το αυτοκίνητο που περιλαμβάνει ένα σύνολο από μηχανικά και ηλεκτρονικά συστήματα τεχνητής νοημοσύνης με στόχο να υπηρετεί τον χειριστή του οχήματος [21]. Το αυτοκίνητο αυτό έχει την δυνατότητα να αντιλαμβάνεται το περιβάλλον του, το οποίο μπορεί να μεταβάλλεται συνεχώς και να κινείται με ασφάλεια έχοντας λίγη η μηδαμινή ανθρώπινη συμμετοχή. Τα αυτόνομα αυτοκίνητα συνδυάζουν μια ποικιλία από αισθητήρες για την κατανόηση του περιβάλλοντος τους. Μερικοί από τους αισθητήρες αυτούς είναι οι θερμικές κάμερες, τα ραντάρ, συσκευές lidar, σόναρ, GPS και IMU συστήματα. Πειράματα σε αυτόνομα συστήματα οδήγησης είχαν ήδη ξεκινήσει από το 1920 και δοκιμές ξεκίνησαν το 1950. Το 1^{ο} ημιαυτόνομο αυτοκίνητο αναπτύχθηκε το 1977 από το μηχανολογικό εργαστήριο Tsukuba'. Στην συνέχεια από το 1960 και έπειτα αναπτύχθηκαν πολλά πλήρως αυτόνομα αυτοκίνητα μικρής ταχύτητας με χαρακτηριστικό επίτευγμα το αυτοκίνητο του πανεπιστημίου ‘Carnegie Mellon’ το 1980 ενώ μέχρι το 1995 τα αυτόνομα αυτοκίνητα απέκτησαν μεγάλες ταχύτητες και μπορούσαν αποφεύγουν εμπόδια. Το 1991 η έρευνα στην Αμερική επικεντρώθηκε στον συνδυασμό «έξυπνων» αυτοκινήτων και οδικού δικτύου με ενσωματωμένες συσκευές επικοινωνίας με αυτά. Το 1995 το αυτοκίνητο Navlab του Πανεπιστημίου Carnegie Mellon οδήγησε 4,584 χιλιόμετρα στην Αμερική και το 98% αυτών αυτόνομα. Το 2015 κάποιες πολιτείες της Αμερικής επέτρεψαν την δοκιμή αυτοματοποιημένων αυτοκινήτων σε δημόσιους δρόμους. Το 2018 η Waymo ανακοίνωσε ότι τα αυτοματοποιημένα αυτοκίνητα της είχαν οδηγήσει αυτόνομα για πάνω από 16.000.000 χιλιόμετρα και ήταν η 1^{η} εταιρεία που αξιοποίησε εμπορικά πλήρως αυτόνομα ταξί στην Φοίνιξ της Αριζόνα. Το 2020 στην Αμερική όλα τα εμπορικά διαθέσιμα αυτοκίνητα απαιτούσαν από τον οδηγό να είναι συνεχώς σε επαγρύπνηση και να συμμετέχει ενεργά στον χειρισμό του αυτοκινήτου. Υπάρχει σύγχυση για τον επακριβή ορισμό ενός πλήρως αυτόνομου αυτοκινήτου. Τα αυτόνομα αυτοκίνητα διακρίνονται από τα αυτοματοποιημένα αυτοκίνητα. Αυτόνομο αυτοκίνητο θεωρείται το αυτοκίνητο που χειρίζεται το ίδιο τα συστήματα του και μπορεί να «οδηγείται» μόνο του σε ένα δυναμικό περιβάλλον, διαθέτοντας την δυνατότητα να διορθώνει λάθη χωρίς εξωτερική βοήθεια. Στην Ευρώπη χρησιμοποιούνται οι φράσεις «αυτοματοποιημένο» και «πλήρως αυτοματοποιημένο» για να γίνει διάκριση μεταξύ αυτοκινήτων που έχουν σχεδιαστεί να κινούνται αυτόνομα για συγκεκριμένες περιόδους χωρίς συνεχή επιτήρηση από τον άνθρωπο οδηγό και των αυτοκινήτων που έχουν σχεδιαστεί με στόχο την οδήγηση με μηδαμινή επιτήρηση από τον οδηγό. Για τον πλήρη προσδιορισμό των αυτόνομων αυτοκινήτων έχουν οριστεί 6 επίπεδα από την «SAE International» [21] :

Level 0 : Καμία αυτοματοποίηση. Το σύστημα απλώς ειδοποιεί τον χειριστή και μπορεί στιγμιαία να δράσει.

Level 1: Μοιραζόμενος έλεγχος του τιμονιού με τον οδηγό. Χαρακτηριστικές εφαρμογές του επιπέδου αυτού είναι τα: Cruise Control, Parking Assistance, Lane Keeping Assistance.

Level 2 : Τα χέρια του οδηγού δεν χρειάζεται να βρίσκονται στο τιμόνι. Το αυτοματοποιημένο σύστημα λαμβάνει πλήρη έλεγχο του αυτοκινήτου, χειρίζεται το γκάζι, το φρένο και το τιμόνι. Ο οδηγός οφείλει να επιτηρεί την διαδικασία και να είναι προετοιμασμένος να επέμβει αν το σύστημα αποτύχει.

Level 3 : Ο οδηγός δεν είναι απαραίτητο να επιτηρεί την οδήγηση. Το σύστημα ειδοποιεί σε περίπτωση ανάγκης. Ο οδηγός και πάλι θα πρέπει να είναι σε ετοιμότητα για να δράσει άμεσα αν ειδοποιηθεί.

Level 4: Ο οδηγός δεν χρειάζεται να ασχολείται με την οδήγηση του αυτοκινήτου. Η πλήρως αυτόνομη οδήγηση βέβαια υποστηρίζεται μόνο σε συγκεκριμένες περιορισμένες περιοχές ή κάτω από ορισμένες προϋποθέσεις. Το αυτοκίνητο υπό άλλες συνθήκες πρέπει να μπορεί να σταματάει την οδήγηση με ασφάλεια εάν ο οδηγός δεν μπορεί να λάβει τον έλεγχο του αυτοκινήτου.

Level 5: Χειρισμός του τιμονιού προαιρετικός. Καμία ανθρώπινη παρέμβαση δεν απαιτείται. Το αυτοκίνητο είναι πλήρως αυτόνομο.

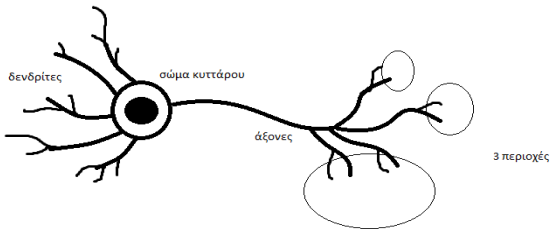
Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά τα βαθιά νευρωνικά δίκτυα και τα συνελικτικά δίκτυα τα οποία χρησιμοποιούνται στην εργασία αυτή.

2.1 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks) αναπτύχθηκαν με βάση την δομή των νευρώνων στον εγκέφαλο του ανθρώπινου οργανισμού με στόχο να εκτελούν πολύπλοκες λειτουργίες [22]. Όπως οι νευρώνες αποτελούνται από τους δενδρίτες, οι οποίοι δέχονται το σήμα από άλλους νευρώνες και από τους άξονες, που μεταδίδουν το σήμα σε άλλους νευρώνες έτσι και οι τεχνητοί νευρώνες αποτελούνται από κόμβους που δέχονται τα σήματα και τα μεταφέρουν στο επόμενο επίπεδο μέσο των αντίστοιχων «συνάψεων» στις οποίες δίνονται βάρη. Επίσης κατ' αντιστοιχία με την διακλάδωση των αξόνων οι κόμβοι ενός τεχνητού νευρωνικού δικτύου μπορούν να μεταδίδουν την πληροφορία σε όλους ή συγκεκριμένους κόμβους του επόμενου επιπέδου του δικτύου. Έτσι κάθε κόμβος μπορεί να δεχτεί σήμα από πολλούς άλλους κόμβους, λειτουργία η οποία στο νευρωνικό δίκτυο του ανθρώπινου εγκεφάλου αποκαλείται 'Σύγκλιση' [23].



Σχήμα 2.1. *Οι νευρώνες στον ανθρώπινο εγκέφαλο. Όπως παρατηρείται, οι άξονες μπορεί να καταλήγουν σε διαφορετικές περιοχές. Έτσι και κάθε κόμβος σε ένα τεχνητό νευρωνικό δίκτυο μπορεί να συνδέεται με διαφορετικούς κόμβους του επόμενου επιπέδου.*

Τα τεχνητά νευρωνικά δίκτυα σχεδιάστηκαν με στόχο την μάθηση με την έννοια της συσσώρευσης γνώσης και τροποποίησης της συμπεριφοράς ώστε να μεγιστοποιείται μια συνάρτηση επιβράβευσης ή να ελαχιστοποιείται μια συνάρτηση «ζημίας ή λάθους». Οφείλουν να διαθέτουν πλαστικότητα και να εξελίσσονται όπως ο ανθρώπινος εγκέφαλος. Ένα νευρωνικό δίκτυο υλοποιείται με ηλεκτρονικά συστατικά ή με λογισμικό στον υπολογιστή. Για βελτίωση της απόδοσης τα νευρωνικά δίκτυα αποτελούνται από μεγάλο αριθμό νευρώνων–μονάδων επεξεργασίας. Τα δίκτυα αυτά λαμβάνουν γνώση από το περιβάλλον τους μέσω της διαδικασίας της μάθησης και τροποποιούν την ισχύ μεταξύ των νευρώνων τους, που αποκαλείται βάρος, για την ενίσχυση ή την αποδυνάμωση των σχέσεων μεταξύ συγκεκριμένων κόμβων του δικτύου.Σ ημαντικές ιδιότητες των Νευρωνικών Δικτύων είναι οι παρακάτω [24] :

Μη γραμμικότητα: Οι νευρώνες μπορεί να είναι γραμμικοί ή και μη γραμμικοί ώστε να μπορούν να χειρίζονται και σήματα εισόδου που εκ φύσεων είναι μη γραμμικά, όπως η ανθρώπινη ομιλία.
Αντιστοιχία Εισόδου και Εξόδου: Τα δεδομένα έρχονται σε ζεύγη όπου κάθε σημείο εισόδου αντιστοιχεί σε έναν στόχο. Κάθε τέτοιο παράδειγμα εισάγεται στο μοντέλο και αυτό τροποποιεί τα βάρη του για να ελαχιστοποιήσει την διαφορά μεταξύ στόχου και απόκρισης.
Δυνατότητα Προσαρμογής: Τα νευρωνικά προσαρμόζουν τα βάρη τους ανάλογα με τα ερεθίσματα του περιβάλλοντος.
Αποθήκευση συνολικής πληροφορίας με την χρήση βαρών: Τα νευρωνικά δίκτυα αποθηκεύουν την τωρινή γνώση τους ως τιμές βαρών των συνάψεων τους και κάθε νευρώνας μπορεί να επηρεάζεται από όλους τους υπόλοιπους.
Δυνατότητα Κατασκευής σε VLSI :Η παραλληλοποιημένη λειτουργία των νευρώνων είναι κατάλληλη για υλοποίηση τους σε VLSI.
Μεγάλη ομοιότητα στην ανάλυση και την υλοποίηση: Οι νευρώνες αποτελούν δομικό συστατικό των νευρωνικών δικτύων και η αρχιτεκτονική και οι διαφορετικές υπομονάδες τους είναι κοινές σε πολλούς τομείς εφαρμογής.
Αρχιτεκτονική βασισμένη στην Νευροφυσιολογία του εγκεφάλου: Η αρχιτεκτονική και η σχεδίαση των νευρωνικών είναι βασισμένες στην φυσιολογία και την δομή του εγκεφάλου και χρησιμοποιούνται εκτενώς για την έρευνα στην νευροβιολογία.
Τα σημαντικότερα μοντέλα νευρώνων παρουσιάζονται παρακάτω :

2.1.1 Perceptron

Το perceptron, όπως αναφέρθηκε και στο 1ο κεφάλαιο, είναι μοντέλο για την ταξινόμηση κλάσεων από δυαδικούς ταξινομητές. Αποτελείται από ένα επίπεδο νευρώνα. Δέχεται ένα διάνυσμα εισόδου x $\in R$ και παράγει έξοδο a $\in$ R.

Κάθε στοιχείο x_i του διανύσματος εισόδου x πολλαπλασιάζεται με το αντίστοιχο συναπτικό βάρος w_i και δημιουργείται ένα άθροισμα. Επίσης στο άθροισμα αυτό ενσωματώνεται και μια εξωτερική πόλωση (bias) $b \in R$ που ορίζει την «προδιάθεση» και έτσι ορίζεται το συνολικό άθροισμα z(x).

$$z(x) = x^T w = \sum_{i=1}^{n} w_i \, x_i + b \qquad (2.1)$$

Στην συνέχεια η τιμή αυτή δίνεται σε μία μη γραμμική συνάρτηση ενεργοποίησης η οποία παράγει την τελική έξοδο. Αν η είσοδος στην συνάρτηση είναι θετική παράγεται έξοδος ίση με +1 αλλιώς αν είναι αρνητική παράγεται έξοδος -1 [25].

$$\alpha = \sigma(\sum_{i=1}^{n} w_i \, x_i + b) \qquad (2.2)$$

Το μοντέλο Perceptron παρουσιάζεται στο ακόλουθο σχήμα :

2.1.2 Multilayer Perceptron

Ένα Multilayer Perceptron [26] αποτελεί ένα πλήρως συνδεδεμένο τεχνητό νευρωνικό δίκτυο. Το πολυεπίπεδο Perceptron αποτελείται από τουλάχιστον 3 επίπεδα κόμβων, ένα εισόδου, ένα ή περισσότερα κρυμμένα επίπεδα και ένα επίπεδο εξόδου. Όλοι οι κόμβοι εκτός των κόμβων εισόδου είναι νευρώνες που χρησιμοποιούν μια μη γραμμική διαφορίσιμη συνάρτηση ενεργοποίησης. Το πολυεπίπεδο Perceptron διακρίνεται από το γραμμικό Perceptron λόγω των πολλαπλών επιπέδων του και της μη γραμμικής συνάρτησης ενεργοποίησης του. Μπορεί να διακρίνει δεδομένα τα οποία δεν μπορούν να διαχωριστούν σε 2 απέναντι πλευρές με μία ευθεία γραμμή σε ένα επίπεδο. Λόγω της κατανεμημένης μορφής της μη γραμμικότητας στην μάθηση

πρέπει να αποφασιστούν τα χαρακτηριστικά της εισόδου που θα περιγράφονται με τους κρυφούς νευρώνες. Η αναζήτηση γίνεται για αυτό τον λόγο σε ένα πολύ μεγαλύτερο χώρο λύσεων. Το διάνυσμα εξόδου σε ένα κρυφό επίπεδο ή στα επίπεδα εξόδου μπορεί να περιγραφεί ως εξής :

$$y_l = \varphi(W_l y_{l-1} + b_l) \tag{2.3}$$

Όπου w_l είναι ένας πίνακας N x M που περιλαμβάνει ένα M-διαστάσεως διανυσματικό βάρος, y_{l-1} είναι το M-διαστάσεως διάνυσμα εξόδου του επιπέδου $l-1$, b_l είναι το N-διαστάσεως διάνυσμα της πόλωσης (bias) και το φ (.) είναι η μη γραμική συνάρτηση ενεργοποίησης. Για κάθε N νευρώνες στο επίπεδο l υπάρχουν M νευρώνες στο επίπεδο l-1.

2.1.3 Back Propagation

Για την μάθηση των MLP χρησιμοποιείται ο αλγόριθμος Back Propagation ή ΒΚ (οπισθοδιάδοση). Ο αλγόριθμος της οπισθοδιάδοσης παρουσιάστηκε για πρώτη φορά στην δεκαετία του 1960 και έγινε γνωστός το 1986 από τον Rumelhart, Hinton και Williams στην σχετική δημοσίευση «Learning representations by back-propagating errors» [6, 7].Ο αλγόριθμος χρησιμοποιεί την μέθοδο της ανάλυσης για την ανάπτυξη-ξεδίπλωμα σύνθετων παραγώγων, γνωστή και ως «κανόνας της αλυσίδας». Η βασική ιδέα του αλγορίθμου είναι η εξής :

- Πρώτα γίνεται το εμπρόσθιο πέρασμα (forward pass) μέσω του νευρωνικού δικτύου στο οποίο το μοντέλο υπολογίζει τα γινόμενα των βαρών με τις εισόδους και μεταδίδει κάθε σήμα από το εκάστοτε επίπεδο στο επόμενο μέχρι να παραχθεί η έξοδος-πρόβλεψη.

- Στην συνέχεια γίνεται πέρασμα από πίσω προς τα εμπρός για τον υπολογισμό της κλίσης της συνάρτησης απωλειών συναρτήσει των βαρών του δικτύου με την χρήση του κανόνα της αλυσίδας.

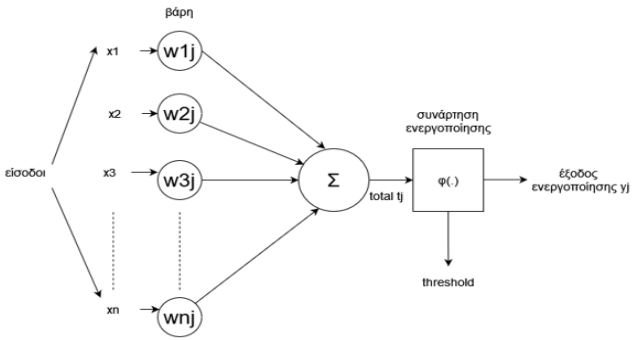
Αν $x \in R^n, z \in R, y = g(x)$ *και* $z = f(y) = f\big(g(x)\big)$, τότε ο κανόνας της αλυσίδας ορίζεται ως:

$$\frac{dz}{dx_i} = \sum_{j=1}^{n} \frac{dz}{dy_j}\frac{dy_j}{dx_i} \tag{2.4}$$

όπου το x είναι το διάνυσμα των χαρακτηριστικών, το y είναι η έξοδος-απόκριση του δικτύου, g(.) η συνάρτηση που αντιστοιχίζει τα χαρακτηριστικά x στις προβλέψεις y ενώ z το κόστος που καθορίζεται από την συνάρτηση σφάλματος f(.) [27]. Ο κανόνας της αλυσίδας εφαρμόζεται είτε σε απλούς μοναδικούς αριθμούς (scalars),ε ίτε σε διανύσματα είτε και σε Τανυστές (Tensors, διαστάσεις >= 3). Τότε η συνάρτηση g(.) αντιστοιχίζει τον Τανυστή Υ σε μία τιμή z. Σε αυτή την περίπτωση ο κανόνας της αλυσίδας ορίζεται ως εξής :

$$\nabla x = \sum_j \nabla x_j \frac{dz}{dY_j} \tag{2.5}$$

Η μέθοδος της οπισθοδιάδοσης συνίσταται στον υπολογισμό της κλίσης της συνάρτησης των απωλειών, συναρτήσει των βαρών του δικτύου. Η οπισθοδιάδοση υπολογίζει αποδοτικά την κλίση της συνάρτησης απωλειών ένα επίπεδο την φορά από το τελευταίο επίπεδο προς το πρώτο αποφεύγοντας υπολογισμούς που έχουν γίνει ήδη και περιττές ενδιάμεσες τιμές. Κάθε βάρος w επηρεάζει την τιμή της συνάρτησης απωλειών μέσω της επίδρασης του στο επόμενο επίπεδο. Όταν γίνει ο υπολογισμός της κλίσης στο επίπεδο l δεν ξαναυπολογίζονται οι παράγωγοι στα μετέπειτα επίπεδα $l+1$, $l+2$, κάθε φορά. Επίσης σε κάθε στάδιο υπολογίζει άμεσα την κλίση των βαρών συναρτήσει της συνολικής συνάρτησης απωλειών και δεν αναλώνεται στον υπολογισμό των παραγώγων των κρυμμένων επιπέδων συναρτήσει των αλλαγών στα βάρη.



$$\frac{dE}{dw_{ij}} = \frac{dE}{dy_j}\frac{dy_j}{dw_{ij}} = \frac{dE}{dy_j}\frac{dy_j}{dt_j}\frac{\partial t_j}{\partial w_{ij}}$$

Σχήμα 2.2. *Εφαρμογή κανόνα αλυσίδας από την έξοδο προς την είσοδο, η παραγώγιση γίνεται ως προς τα βάρη.*

Η συνάρτηση ενεργοποίησης και η συνάρτηση απωλειών που χρησιμοποιεί το εκάστοτε δίκτυο δεν έχουν σημασία για την οπισθοδιάδοση αρκεί να μπορούν να υπολογιστούν αποδοτικά μαζί με τις παραγώγους τους.

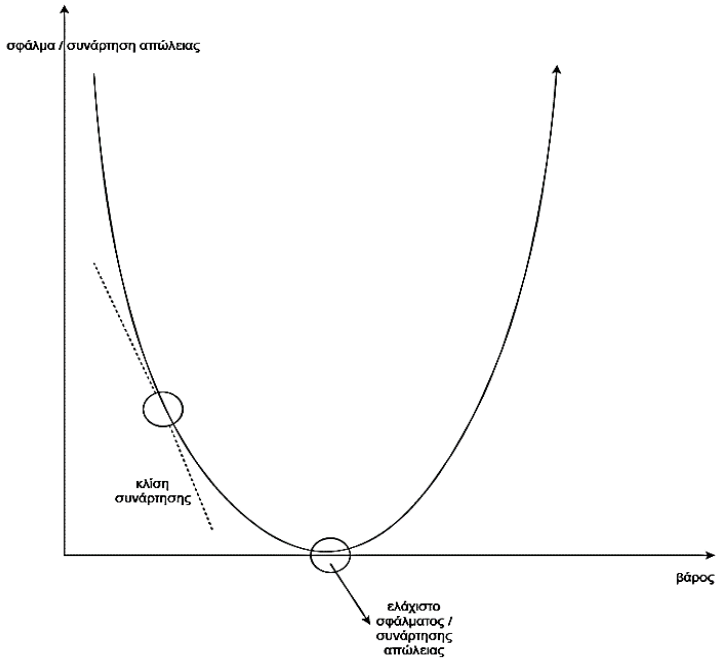
2.1.4 Gradient Descent

Μόλις γίνει ο υπολογισμός των παραγώγων των διαφόρων επιπέδων οι παράγωγοι μπορούν να χρησιμοποιηθούν για την μάθηση του δικτύου. Η μάθηση γίνεται συνήθως με την χρήση του αλγορίθμου της Κατάβασης Κλίσης (Gradient Descent) που αποτελεί επαναληπτικό αλγόριθμο βελτιστοποίησης με στόχο την εύρεση των τοπικών ελαχίστων μιας διαφορίσιμης συνάρτησης [28].Η βασική ιδέα της μεθόδου είναι τα βάρη των «συνάψεων» να μετατοπίζονται κατά την αντίθετη κατεύθυνση της κλίσης της συνάρτησης των απωλειών. Η μετατόπιση γίνεται προς αυτήν την κατεύθυνση εφόσον η κατεύθυνση της κλίσης είναι προς το τοπικό μέγιστο. Συνεπώς η αντίθετη κατεύθυνση είναι προς το τοπικό ελάχιστο της συνάρτησης το οποίο αναζητείται για την ελαχιστοποίηση του λάθους στα νευρωνικά δίκτυα. Η μέθοδος θεωρείται ότι ανακαλύφθηκε πρώτα από τον Augustin-Louis Cauchy το 1847. Η μέθοδος άρχισε να μελετάται εντατικά και να εφαρμόζεται τις δεκαετίες μετά το 1944, όταν ο Haskell Curry μελέτησε τις ιδιότητες σύγκλισης που παρουσίαζε σε μη γραμμικά προβλήματα βελστιστοποίησης. Αν $F(x)$ είναι μια συνάρτηση πολλών μεταβλητών και είναι διαφορίσιμη σε μία γειτονιά ενός σημείου α, τότε η $F(x)$ μειώνεται γρηγορότερα αν ακολουθήσει κανείς την διαδρομή από το α στην κατεύθυνση στης αρνητικής κλίσης της F στο σημείο α,η οποία συμβολίζεται ως $-\nabla F(\alpha)$. Συνεπώς προκύπτει ότι η νέα μετατοπισμένη θέση δίνεται από την εξίσωση

$$a_{n+1} = a_n - \gamma \nabla F(a_n) \qquad (2.6)$$

Όπου το γ είναι ο ρυθμός μάθησης $\gamma \in R_+$ και $F(\alpha_n) \geq F(\alpha_{n+1})$. **Ο ρυθμός μάθησης καθορίζει πόσο γρήγορα ή αργά θα ανανεώνονται τα βάρη** καθώς όπως φαίνεται από την παραπάνω εξίσωση αποτελεί ένα ποσοστό της κλίσης. Είναι μια σημαντική υπερπαράμετρος των νευρωνικών δικτύων καθώς πολύ μεγάλες τιμές του μπορεί να οδηγήσουν σε μη ολικά και μη βέλτιστα ελάχιστα της συνάρτησης απωλειών ενώ πολύ μικρές τιμές του μπορεί να οδηγήσουν σε αργή μάθηση και σύγκλιση του μοντέλου. Συνεπώς ο όρος $\gamma \nabla F(\alpha)$ αφαιρείται με στόχο την προσέγγιση του τοπικού ελαχίστου αντίθετα από την κατεύθυνση της κλίσης. Στον τομέα των νευρωνικών δικτύων μπορεί να υποτεθεί ότι το α αποτελεί το αντίστοιχο βάρος που πρέπει να τροποποιηθεί προς την αντίθετη κατεύθυνση της κλίσης της συνάρτησης απωλειών. Η αφαίρεση

της κλίσης εγγυάται ότι το βάρος θα μετατοπιστεί προς την σωστή κατεύθυνση για την ελαχιστοποίηση του σφάλματος ακόμη και αν οι είσοδοι είναι αρνητικές τιμές. Πιο απλά η κατάβαση κλίσης αποτελεί μία μέθοδο βελτιστοποίησης μίας αντικειμενικής συνάρτησης παραμετροποιημένης από τις παραμέτρους του μοντέλου (βάρη), με την ανανέωση των παραμέτρων προς την αντίθετη κατεύθυνση της κλίσης της αντικειμενικής συνάρτησης.



Σχήμα 2.3. *Η κατεύθυνση αντίθετα από την κατεύθυνση της κλίσης οδηγεί στο τοπικό ελάχιστο του σφάλματος. Στην κατάβαση κλίσης αναζητείται το βάρος αυτό που αντιστοιχεί στο ολικό ή ικανοποιητικό τοπικό ελάχιστο.*

Κατηγορίες Gradient Descent και Optimizers

Ο αλγόριθμος της κατάβασης κλίσης μπορεί να εφαρμοστεί με διαφορετικούς τρόπους με τον οποίο γίνεται η ανανέωση των βαρών [29].

1. Batch Gradient Descent: Η μαζική (batch) κατάβαση κλίσης υπολογίζει το σφάλμα για κάθε δεδομένο παράδειγμα του συνόλου εκπαίδευσης αλλά η τροποποίηση των βαρών γίνεται μόνο μετά την εξέταση όλων των δεδομένων εκπαίδευσης. Η ολοκλήρωση μιας επανάληψης εκπαίδευσης αποκαλείται εποχή. Η μαζική κατάβαση κλίσης είναι υπολογιστικά αποδοτική και παράγει σφάλμα χωρίς διακυμάνσεις ενώ οδηγεί σε σταθερή σύγκλιση. Από την άλλη η τιμή στην οποία συγκλίνει το σφάλμα δεν είναι πάντα η βέλτιστη και η υλοποίηση της απαιτεί την αποθήκευση μεγάλων ποσοτήτων δεδομένων (το σύνολο εκπαίδευσης) στην μνήμη. Αν θ είναι οι παράμετροι όλου του συνόλου εκπαίδευσης τότε η σχέση που περιγράφει την batch gradient descent είναι :

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$$

Όπου $J(\theta)$ είναι η συνάρτηση απωλειών, που μπορεί να οριστεί γενικά σαν το άθροισμα των σφαλμάτων για κάθε στιγμιότυπο, δηλαδή το άθροισμα των διαφορών προβλέψεων και πραγματικών τιμών για όλα τα δεδομένα.

2. Stochastic Gradient Descent (SGD): Στην μέθοδο αυτή η τροποποίηση των βαρών γίνεται μετά από κάθε δεδομένο εκπαίδευσης και όχι μαζικά. Η στοχαστική κατάβαση κλίσης είναι πιο γρήγορη από την μαζική και παρουσιάζει μια καλύτερη εικόνα για τις αλλαγές των βαρών. Παρά ταύτα είναι υπολογιστικά ακριβή λόγω των συνεχών τροποποιήσεων των βαρών και επίσης οδηγεί στην εμφάνιση ταλαντώσεων του σφάλματος χωρίς να δίνει μια καθαρή εικόνα της μείωσης του.

$$\theta_j = \theta_j - \alpha\left(\widehat{y^i} - y^i\right)$$

Στην συγκεκριμένη περίπτωση το i δηλώνει ένα τυχαίο δεδομένο από το σύνολο εκπαίδευσης και η σχέση επαναλαμβάνεται για κάθε δεδομένα από το σύνολο εκπαίδευσης.

3. Mini-Batch Gradient Descent: Η κατάβαση κλίσης μίνι-παρτίδας αποτελεί έναν συνδυασμό της SGD και της μαζικής κατάβασης κλίσης. Η εκπαίδευση-τροποποίηση των βαρών γίνεται σε μικρές παρτίδες (mini batches) του συνόλου εκπαίδευσης πετυχαίνοντας την ισορροπία μεταξύ των δύο μεθόδων. Τα μεγέθη των παρτίδων εξαρτώνται από το πρόβλημα και το μοντέλο. Η κατάβαση κλίσης μίνι-παρτίδας είναι η πιο συχνά χρησιμοποιούμενη στην μηχανική μάθηση. Η εξίσωση της μπορεί να θεωρηθεί παρόμοια με την εξίσωση της SGD όπου όμως η ανανέωση των βαρών γίνεται για μια ομάδα των δεδομένων και όχι για κάθε δεδομένο ξεχωριστά.

Για την επιτάχυνση της σύγκλισης της συνάρτησης των απωλειών των νευρωνικών δικτύων ώστε να βρεθούν πιο γρήγορα τα ελάχιστα της, χρησιμοποιούνται κάποιοι αλγόριθμοι βελτιστοποίησης σε συνδυασμό με τον αλγόριθμο της κατάβασης κλίσης. Οι πιο γνωστοί είναι :

- Momentum

Επειδή η SGD μπορεί να εγκλωβιστεί σε «φαράγγια» της συνάρτησης που καλείται να ελαχιστοποιήσει (αυτά εμφανίζονται συχνά κοντά σε τοπικά ελάχιστα), γίνεται χρήση του όρου Momentum [30], μιας τεχνικής για την επιτάχυνση της σύγκλισης η οποία συσσωρεύει ένα μέρος από τις προηγούμενες ανανεώσεις των βαρών με συνέπεια η ανανέωση των βαρών προς την σωστή κατεύθυνση να επιταχύνεται σε περίπτωση αλλαγών προς την σωστή κατεύθυνση και να επιβραδύνεται στην περίπτωση αλλαγών προς την λανθασμένη κατεύθυνση.

$$\theta = \theta - \left(\gamma u_{t-1} + \eta \nabla J(\theta)\right)$$

Όπου θ είναι η παράμετρος που μεταβάλλεται, γu_{t-1} είναι ο όρος των προηγούμενων ανανεώσεων και ο $\eta \nabla J(\theta)$ ο γνωστός όρος της κατάβασης κλίσης για την ανανέωση των βαρών. Συνήθως το γ τίθεται 0.9.

- Nesterov

Αποτελεί μια βελτίωση της Momentum όπου επιπλέον επιχειρείται να υπολογιστεί και η επόμενη θέση της παραμέτρου και στην συνέχεια διορθώνεται η ανανέωση με βάση την προσέγγιση αυτή ώστε να μην γίνονται μεγάλες «τυφλές» αλλαγές των παραμέτρων [31].

- Adagrad

Με την μέθοδο αυτή σε κάθε χρονική ανανέωση των παραμέτρων ο ρυθμός μάθησης δεν είναι σταθερός αλλά μεταβάλλεται με βάση τις προηγούμενες κλίσεις που υπολογίστηκαν. Με την χρήση της μεθόδου αυτής μπορεί να επιτευχθεί αυτοματοποίηση της προσαρμογής του ρυθμού μάθησης [32].

- Adadelta

Η τεχνική είναι επέκταση της AdaGrad και περιορίζει τις κλίσεις που συσσωρεύονται αντί να χρησιμοποιεί όλες τις προηγούμενες κλίσεις [33].

- RMSprop

Η μέθοδος είναι μη δημοσιευμένη και προτάθηκε από τον Geoff Hinton. Σχεδιάστηκε υπό την παρατήρηση ότι τα μεγέθη των κλίσεων μπορεί να μεταβάλλονται για διαφορετικά βάρη και ορίζει ένα κινούμενο μέσο όρο των τετραγώνων των κλίσεων προσαρμόζοντας τις αλλαγές στα βάρη με βάση αυτόν τον μέσο [34, 35].

- Adam

Η μέθοδος Adaptive Moment Estimation [36] υπολογίζει ρυθμούς μάθησης που αυτοπροσαρμόζονται με βάση και όρους τύπου Momentum αλλά και αποθηκεύοντας έναν εκθετικό μέσο όρο των τετραγώνων των παρελθοντικών κλίσεων. Έτσι είναι ικανός να επιταχύνει αλλά και να επιβραδύνει αναζητώντας ελάχιστα στην επιφάνεια που κινείται.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t$$

Όπου m_t είναι προσέγγιση της μέσης τιμής και v_t είναι προσέγγιση της μη κεντρικοποιημένης διασποράς των κλίσεων. Αν στην συνέχεια υπολογιστούν τα διορθωμένα $\widehat{m_t}$ και $\widehat{v_t}$ τότε η ανανέωση των βαρών σύμφωνα με τον Adam είναι :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon} \widehat{m_t}$$

με τα m_t και $\widehat{v_t}$ ορισμένα ως εξής :

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}$$

$$\widehat{v_t} = \frac{v_t}{1 - \beta_2^t}$$

2.1.5 Σύνολα Εκπαίδευσης, Επαλήθευσης και Test

Στα νευρωνικά δίκτυα και την μηχανική μάθηση χρησιμοποιούνται για την εκπαίδευση και την δοκιμή των κατασκευαζόμενων μοντέλων τρία σύνολα [37].

- Σύνολο εκπαίδευσης: Το σύνολο αυτό αποτελείται από δεδομένα τα οποία δίνονται στο μοντέλο με στόχο τον υπολογισμό των εκπαιδεύσιμων παραμέτρων του μοντέλου, κυριότερα από τα οποία είναι τα βάρη των συνάψεων των νευρώνων.Το μοντέλο εκπαιδεύεται στα δεδομένα του συνόλου εκπαίδευσης χρησιμοποιώντας μια μέθοδο Επιβλεπόμενης μάθησης όπως την στοχαστική κατάβαση κλίσης. Τα δεδομένα εκπαίδευσης αποτελούνται από ζεύγη από εισόδους και εξόδους, όπου οι έξοδοι είναι οι πραγματικές τιμές-στόχοι που το μοντέλο προσπαθεί να προσεγγίσει. Το αποτέλεσμα της διέλευσης των δεδομένων εισόδου εκπαίδευσης μέσα από το μοντέλο είναι η παραγωγή του αποτελέσματος το οποίο συγκρίνεται με τον στόχο και τροποποιείται ανάλογα το αντίστοιχο βάρος με στόχο την ελαχιστοποίηση της διαφοράς τους.
- Στην συνέχεια το μοντέλο καλείται να προβλέψει τις τιμές στόχους σε ένα σύνολο δεδομένων που ονομάζεται σύνολο επαλήθευσης. Οι προβλέψεις στο σύνολο αυτό αποτελούν μία εκτίμηση της απόδοσης του μοντέλου και της ποιότητας της μάθησης που έχει γίνει στα δεδομένα εκπαίδευσης ενώ ταυτόχρονα τροποποιούνται οι υπερπαράμετροι του μοντέλου. Τα σύνολα επαλήθευσης έχουν μεγάλη χρησιμότητα για την πρωϊμότερη διακοπή της μάθησης σε περίπτωση που το σφάλμα στα σύνολα αυτά αυξηθεί, κάτι το οποίο μπορεί να δηλώνει overfitting.
- Μετά την εκπαίδευση του μοντέλου αυτό εφαρμόζεται σε ένα ανεξάρτητο σύνολο δεδομένων που ονομάζεται test σύνολο και χρησιμοποιείται για την εκτίμηση της επιτυχίας του μοντέλου σε δεδομένα που δεν έχει ξανασυναντήσει.

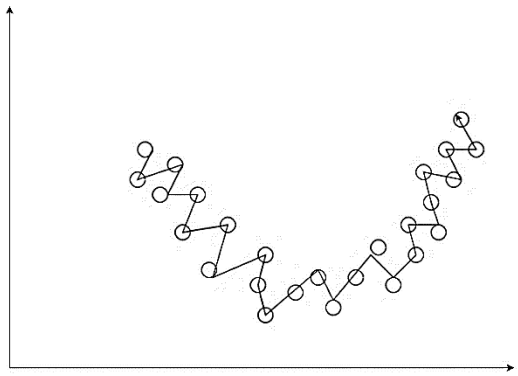
Τα σύνολα επαλήθευσης και test πρέπει να ακολουθούν την ίδια στατιστική κατανομή με το σύνολο εκπαίδευσης. Συνήθως τα σύνολα εκπαίδευσης και επαλήθευσης κατασκευάζονται από την διαίρεση ενός συνόλου δεδομένων σε δύο μέρη ενώ το σύνολο test είναι πάντα ανεξάρτητο. Μπορεί επίσης να γίνει επαναληπτική διαίρεση ενός συνόλου δεδομένων σε σύνολα εκπαίδευσης και επαλήθευσης με στόχο πιο σταθερά αποτελέσματα. Η μέθοδος αυτή ονομάζεται διασταυρούμενη επαλήθευση (cross validation). Στόχος της εκπαίδευσης είναι η υψηλή ακρίβεια πρόβλεψης του μοντέλου σε ανεξάρτητα δεδομένα του προβλήματος που καλείται να επιλύσει.

2.1.6 Υπερπροσαρμογή (Overfitting) και Υποπροσαρμογή(Underfitting)

Υπερπροσαρμογή

Στην στατιστική ανάλυση και την επιστήμη δεδομένων, όταν ένα μοντέλο ταιριάζει απόλυτα με τα δεδομένα εκπαίδευσης, λέγεται ότι συμβαίνει overfitting [38].Στην περίπτωση αυτή το μοντέλο δεν μαθαίνει να γενικεύει και δεν έχει καλή απόδοση σε δεδομένα που δεν έχει ξαναδεί. Το μοντέλο έχει μάθει απλώς στα δεδομένα αυτά εκπαίδευσης να δίνει μια συγκεκριμένη απόκριση και όχι την δομή και τα χαρακτηριστικά που θα οδηγούσαν σε αυτή την απόκριση και

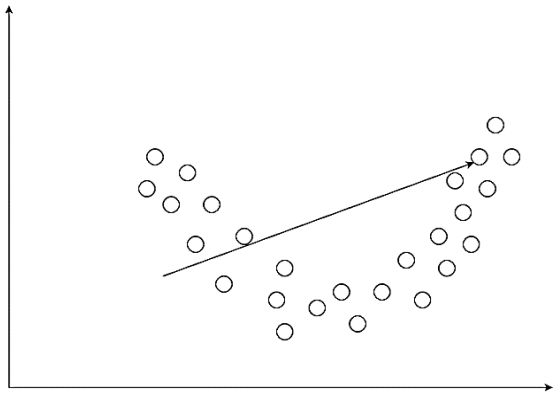
θα μπορούσαν να αναγνωριστούν και σε νέα δεδομένα. Το overfitting συμβαίνει όταν το μοντέλο εκπαιδεύεται για πολύ μεγάλο διάστημα στα δεδομένα εκπαίδευσης ή όταν το μοντέλο είναι πιο σύνθετο, στην οποία περίπτωση μαθαίνει να μοντελοποιεί τον θόρυβο των δεδομένων. Τότε δεν μπορεί να γενικεύσει και να χρησιμοποιηθεί για την πρόβλεψη ή την κατηγοριοποίηση νέων δεδομένων. Δείγματα overfitting είναι συχνά το πολύ μικρό σφάλμα και η υψηλή διασπορά. Για την αποφυγή του, χρησιμοποιούνται τεχνικές ομαλοποίησης όπως dropout επίπεδα για την τυχαία απόρριψη κάποιων κόμβων από την εκπαίδευση σε κάθε βήμα αλλά και η μέθοδος της πρόωρης διακοπής (early stopping) [39]. Σύμφωνα με την τεχνική του early stopping όταν παρατηρηθεί αύξηση του σφάλματος στα δεδομένα επαλήθευσης -δείγμα overfitting- τότε σταματάει η εκπαίδευση και το μοντέλο κρατάει τις παραμέτρους που είχε ακριβώς πριν την αύξηση.



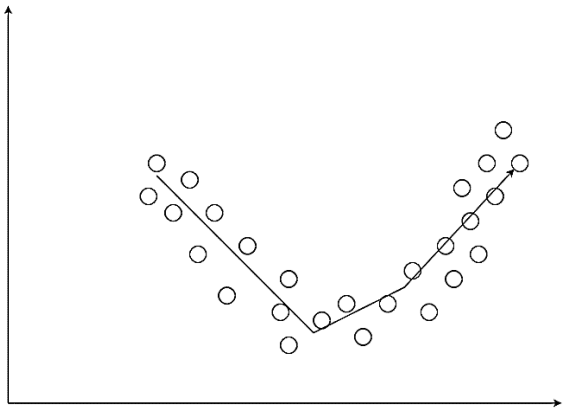
Σχήμα 2.4. *Υπερπροσαρμογή: Το μοντέλο μαθαίνει τον θόρυβο και δεν μοντελοποιεί σωστά τις σχέσεις , τα χαρακτηριστικά των δεδομένων για την κατηγοριοποίηση. Η καμπύλη είναι πολύ υψηλής τάξης.*

Υποπροσαρμογή

Underfitting [40] συμβαίνει όταν το μοντέλο δεν έχει εκπαιδευτεί αρκετά ή τα δεδομένα εκπαίδευσης δεν είναι κατάλληλα για την εύρεση και την μοντελοποίηση της δομής και της σχέσης μεταξύ εισόδου και εξόδου, με συνέπεια μεγάλο σφάλμα αδυναμία του μοντέλου να γενικεύσει. Μία πιθανώς ικανοποιητική καμπύλη για την παραπάνω κατανομή είναι αυτή του σχήματος 2.6.



Σχήμα 2.5. *Το μοντέλο δεν έχει εκπαιδευτεί αρκετά ή τα δεδομένα δεν επαρκούν για να μάθει τις σχέσεις και τα χαρακτηριστικά των δεδομένων.Έτσι η καμπύλη που προβλέπει είναι πολύ απλή και λαναθασμένη.* (ευθεία γραμμή).



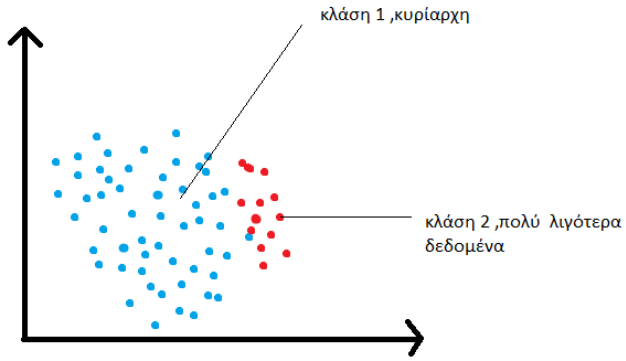
Σχήμα 2.6. *Η προβλεπόμενη καμπύλη μοντελοποιεί σχετικά καλά την υπάρχουσα δομή των δεδομένων.*

(υψηλής τάξης καμπύλη).

2.1.7 Ανισορροπία Κλάσεων

Ανισορροπία κλάσεων εμφανίζεται σε ένα σύνολο δεδομένων όταν μια κλάση αντιπροσωπεύεται σε πολύ μεγαλύτερο βαθμό από μια άλλη, με την εμφάνιση περισσότερων στιγμιότυπων της κλάσης αυτής. Το μοντέλο σε αυτή την περίπτωση μπορεί απλώς να μάθει να προβλέπει την κλάση με την μεγαλύτερη συχνότητα έχοντας μεγάλη ακρίβεια. Η εικόνα αυτή όμως δεν αναπαριστά σωστά την πληροφορία των δεδομένων και δεν είναι ασφαλές το συμπέρασμα ότι το μοντέλο έχει μάθει εφ'όσον σε νέα δεδομένα, όπου η ανισορροπία έχει αντιστραφεί ή υπάρχει ισορροπία, το μοντέλο θα έχει πολύ μεγάλο σφάλμα. Για την αντιμετώπιση της ανισορροπίας μπορούν να εφαρμοστούν τεχνικές που επιχειρούν να επαναφέρουν την ισορροπία στα δεδομένα με τους παρακάτω τρόπους :

- Υποδειγματοληψία της κλάσης με την μεγαλύτερη συχνότητα: Η μέθοδος αυτή κατασκευάζει ένα σύνολο εκπαίδευσης στο οποίο η πιο συχνά εμφανιζόμενη κλάση περιορίζεται από λιγότερα δείγματα με στόχο ένα πιο ισορροπημένο σύνολο.
- Υπερδειγματοληψία της κλάσης με την μικρότερη συχνότητα: Τα στιγμιότυπα της πιο σπάνιας κλάσης αντιγράφονται και πολλαπλασιάζονται και επιπλέον μπορεί να υποστούν διάφορες μορφές μετασχηματισμών (data augmentation), όπως περιστροφές , κατοπτρισμούς, flips, μεταφορές συστημάτων, πρόσθεση θορύβου, διάτμηση, οι οποίοι διατηρούν την υπάρχουσα πληροφορία. Στόχος είναι η δημιουργία πολλών νέων δεδομένων από τα αρχικά λίγα δεδομένα της πιο σπάνιας κλάσης. Η τεχνική μπορεί και να συνδυαστεί με υποδειγματοληψία της κυρίαρχης κλάσης.



Σχήμα 2.7. *Ανισορροπία κλάσεων.Η κλάση 1 στο σχήμα έχει πολύ περισσότερα στιγμιότυπα από την κλάση 2.*

2.1.8 Συναρτήσεις ενεργοποίησης

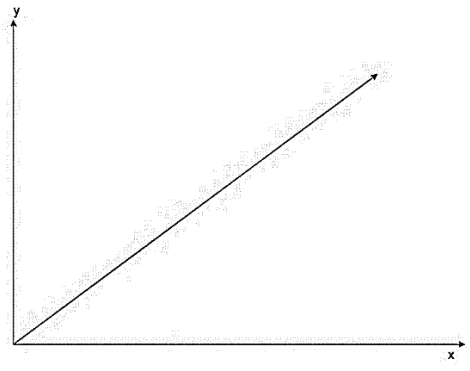
Στα τεχνητά νευρωνικά δίκτυα, συνάρτηση ενεργοποίησης αποκαλείται μια συνάρτηση ενός κόμβου η οποία καθορίζει την έξοδο του κόμβου αυτού, δοθείσης μιας είσοδου ή ενός συνόλου εισόδων. Για την αντιμετώπιση σύνθετων προβλημάτων με την χρήση νευρωνικών δικτύων με μικρό αριθμό κόμβων, πρέπει οι συναρτήσεις ενεργοποίησης να είναι μη γραμμικές. Οι συναρτήσεις αυτές τότε ονομάζονται μη-γραμμικότητες. Είναι απαραίτητες για την κωδικοποίηση μη γραμμικών σχέσεων μεταξύ εισόδων και εξόδων και την μάθηση του εκάστοτε μοντέλου καθώς καμπυλώνονται με αποτέλεσμα να μπορούν να περιγράψουν σύνθετες λειτουργίες. Για να μπορούν να αξιοποιηθούν στους αλγόριθμους υπολογισμού της κλίσης (backpropagation) και ύστερα στην μάθηση, οι συναρτήσεις ενεργοποίησης οφείλουν να είναι διαφορίσιμες. Μια συνάρτηση ενεργοποίησης οδηγείται σε κορεσμό εάν $\lim_{v\to\infty}|\nabla f(v)| = 0$. Πιο συχνά χρησιμοποιούμενες συναρτήσεις ενεργοποίησης είναι οι παρακάτω γραμμικές και μη γραμμικές συναρτήσεις [41, 42, 43].

Γραμμικές

- Γραμμική ενεργοποίηση/ταυτότητα (linear)

$$f(x) = x$$

Η συνάρτηση αυτή δεν είναι κατάλληλη για τον αλγόριθμο backpropagation εφόσον η παράγωγος της είναι σταθερά. Επίσης το τελευταίο επίπεδο είναι στην ουσία μια γραμμική συνάρτηση του πρώτου. Συνεπώς το νευρωνικό καταρρέει σε ένα επίπεδο.

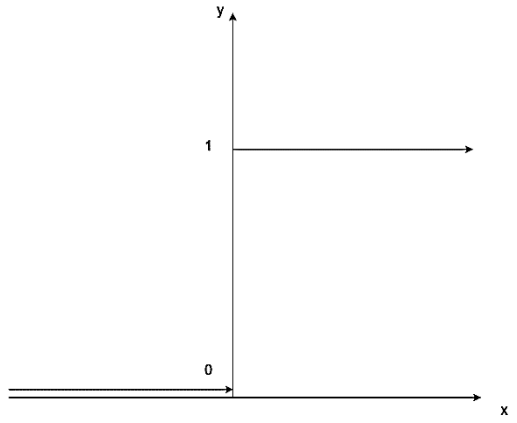


Σχήμα 2.8. *Γραμμική συνάρτηση*

- Συνάρτηση δυαδικού βήματος (step)

$$f(x) = \begin{cases} 0\,, & για\ x\ < 0 \\ 1\,, & για\ x\ \geq 0 \end{cases}$$

Η συνάρτηση αυτή δεν μπορεί να χρησιμοποιηθεί για κατηγοριοποίηση πολλαπλών κλάσεων εφόσον έχει δυαδική τιμή. Λόγω των σταθερών τιμών που δίνει, η παράγωγος είναι 0 γεγονός που δεν επιτρέπει την εφαρμογή του αλγορίθμου backpropagation.



Σχήμα 2.9. *Συνάρτηση Βήματος*

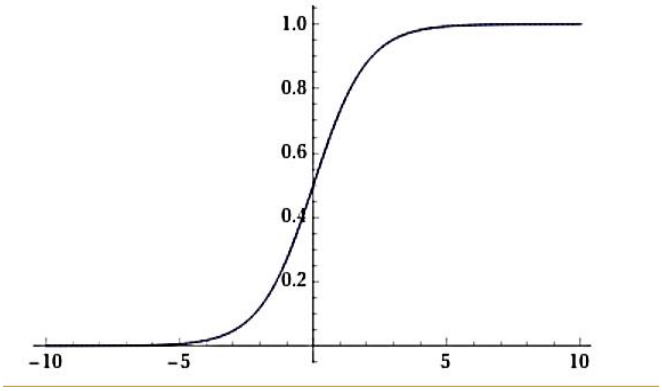
Μη Γραμμικές

- Σιγμοειδής/Λογιστική συνάρτηση (sigmoid)

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

Η συνάρτηση αυτή χρησιμοποιείται συχνότερα για την παραγωγή εξόδου με την μορφή πιθανότητας, δηλαδή το σύνολο τιμών της είναι από 0 έως 1. Οι μεγάλοι αρνητικοί αριθμοί γίνονται 0 ενώ οι μεγάλοι θετικοί γίνονται 1. Είναι διαφορίσιμη και αποτρέπει την εμφάνιση αλμάτων στις εξόδους. Χρησιμοποιείται και για κρυμμένα επίπεδα και για επίπεδα εξόδου.

Σημαντικό πρόβλημα της συνάρτησης είναι από την άλλη ότι η παράγωγος της sigmoid(x)*(1-sigmoid(x)) παίρνει μεγάλες τιμές μόνο στο διάστημα [-3,3] με αποτέλεσμα οι κλίσεις σε τιμές εκτός του διαστήματος να είναι πολύ μικρές, σχεδόν 0. Έτσι το δίκτυο δεν μπορεί να μάθει εφόσον οι παράγωγοι που συντελούν στην τροποποίηση των βαρών είναι μηδενικές και τα βάρη δεν ανανεώνονται. Το πρόβλημα αυτό ονομάζεται «Εξαφανιζόμενη Κλίση» (Vanishing Gradient). Επίσης η σιγμοειδής δεν είναι συμμετρική γύρω από το 0 και όλες οι έξοδοι έχουν το ίδιο πρόσημο με συνέπεια η εκπαίδευση να είναι πιο δύσκολη.

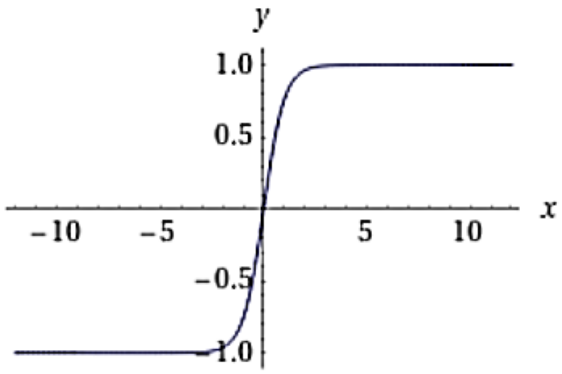


Σχήμα 2.10. *Σιγμοειδής συνάρτηση*

- Υπερβολική εφαπτομένη (tanh)

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

Μοιάζει γραφικά με την σιγμοειδή αλλά το σύνολο τιμών της είναι [-1,1]. Όσο μεγαλύτερη είναι η είσοδος τόσο πιο κοντά θα είναι η έξοδος στο 1.0 ενώ όσο μικραίνει, πλησιάζει το -1.0. Λόγω του ότι το σύνολο τιμών περιλαμβάνει και αρνητικούς αριθμούς μπορεί να κωδικοποιήσει αρνητική συσχέτιση για κρυμμένα επίπεδα. Σε αντίθεση με την σιγμοειδή είναι συμμετρική γύρω από το 0 και πολλές φορές προτιμάται από την σιγμοειδή. Από την άλλη το πρόβλημα της εξαφανιζόμενης κλίσης εμφανίζεται και σε αυτή τη συνάρτηση.

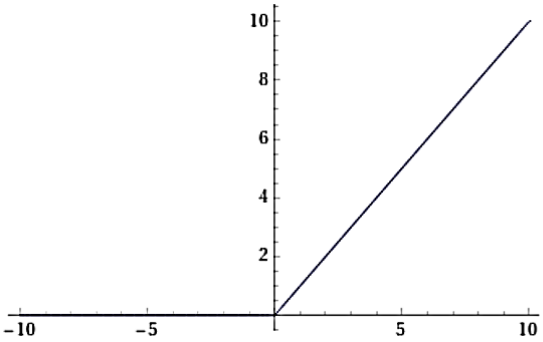


Σχήμα 2.11. *Υπερβολική εφαπτομένη*

- Ανορθωμένη γραμμική μονάδα (ReLu)

$$f(x) = max(0, x)$$

Λόγω του ότι το σύνολο τιμών της κρατάει μόνο τις θετικές τιμές ή το 0 και συνεπώς πολλοί νευρώνες απενεργοποιούνται, είναι μια αποδοτική συνάρτηση. επίσης, επειδή δεν οδηγείται στον κορεσμό, συντελεί στην επιτάχυνση της σύγκλισης της κατάβασης κλίσης στο ολικό ελάχιστο της συνάρτησης απωλειών. Χαρακτηριστικό πρόβλημα της συνάρτησης αυτής είναι ότι στις αρνητικές τιμές η παράγωγος της κάνει την κλίση 0, με αποτέλεσμα στο backpropagation πολλοί νευρώνες να μην ενεργοποιούνται ποτέ ενώ τα βάρη δεν ανανεώνονται. Το φαινόμενο ονομάζεται «Νεκρή ReLU». Επίσης όλες οι αρνητικές τιμές γίνονται 0 από την ReLU με συνέπεια το μοντέλο να μην κωδικοποιεί σωστά όλη την πληροφορία των εισόδων το οποίο μπορεί να είναι πρόβλημα όταν ενδιαφέρει το πρόσημο για την εξαγωγή των χαρακτηριστικών.

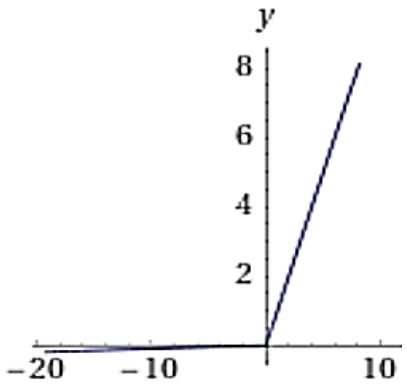


Σχήμα 2.12. *Ανορθωμένη Γραμμική Μονάδα (ReLU)*

- Leaky Ανορθωμένη γραμμική μονάδα (Leaky ReLU)

$$f(x) = max(0.1x, x)$$

Η μη γραμμική αυτή συνάρτηση αποτελεί μια προσπάθεια αντιμετώπισης του προβλήματος της «Νεκρής ReLU». Αντί για μηδενισμό των τιμών στις αρνητικές τιμές εισάγει μια μικρή θετική κλίση έτσι ώστε να αποτρέπονται οι νεκροί νευρώνες για αρνητικές τιμές. Σε αντίθεση με την ReLU επιτρέπει την εφαρμογή του backpropagation ακόμη και για αρνητικές τιμές εισόδου. Βέβαια επειδή η κλίση στις αρνητικές τιμές είναι μικρή, η μάθηση μπορεί να είναι αργή. Παρόμοια ορίζεται και η παραμετρική Relu απλώς αλλάζει ο συντελεστής για $x < 0$ και γίνεται ένας συντελεστής α.



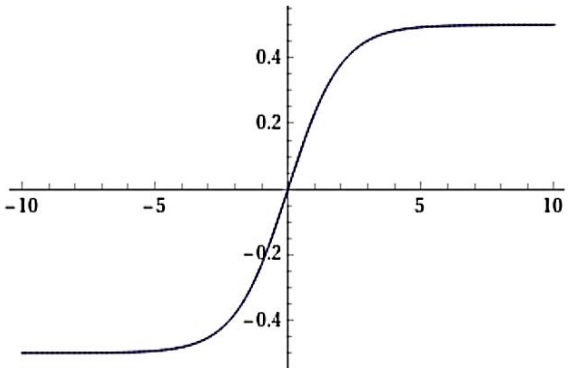
Σχήμα 2.13. *Leaky ReLU*

- Συνάρτηση Softmax

Η συνάρτηση αποτελεί γενίκευση της λογιστικής συνάρτησης σε πολλές διαστάσεις/ κλάσεις .Δέχεται σαν είσοδο ένα διάνυσμα z από K πραγματικούς αριθμούς και εφαρμόζει τον ακόλουθο τύπο:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \; για \;\; i = 1, \dots, K \; και \; z = (z_1, \dots, z_K) \in R^K$$

Σε κάθε z_i εφαρμόζει την συνάρτηση e^x και κανονικοποιεί κάθε τιμή με το άθροισμα όλων των εκθετικών. Η συνάρτηση Softmax μετατρέπει την έξοδο από ένα επίπεδο σε ένα διάνυσμα από πιθανότητες. Μετά την εφαρμογή της συνάρτησης αυτής όλες οι τιμές ανήκουν στο (0,1) και αθροίζονται στο 1 σε αντίθεση με την σιγμοειδή. Στα νευρωνικά χρησιμοποιείται συχνά στο τελευταίο επίπεδο ενός ταξινομητή. Χρησιμοποιείται συχνά για κατηγοριοποίηση πολλών κλάσεων (multiclass classification) καθώς παρουσιάζει 1 για την μεγαλύτερη πιθανότητα και 0 για τις υπόλοιπες.

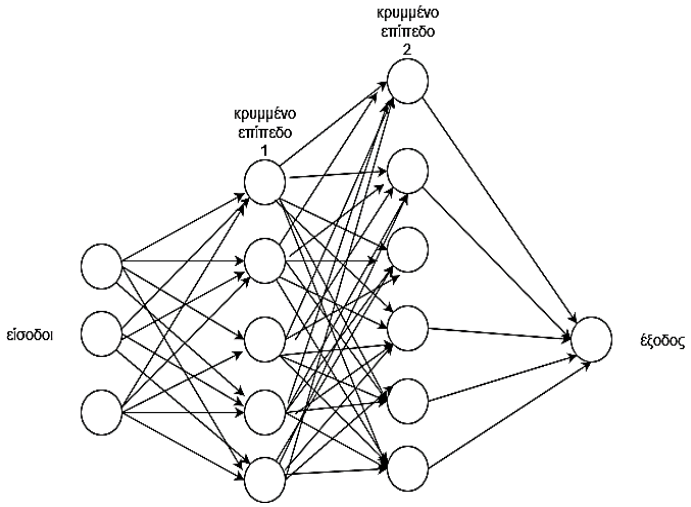


Σχήμα 2.14. *Συνάρτηση Softmax*

2.2 Βαθιά Μάθηση (Deep learning)

Ο όρος Deep learning χρησιμοποιείται επίσημα για να περιγράψει νευρωνικά δίκτυα με τρία ή περισσότερα επίπεδα. Πιο γενικά περιγράφει μεθόδους της μηχανικής μάθησης που περιλαμβάνουν «βαθιές» αρχιτεκτονικές νευρωνικών δικτύων με πολλά κρυμμένα επίπεδα [44]. Με την επαύξηση των δικτύων με πολλά επίπεδα επιχειρείται η προσομοίωση της λειτουργίας του ανθρώπινου εγκεφάλου και προσφέρεται η δυνατότητα βελτιστοποίησης της μάθησης και της ακρίβειας που αποκτάται με απλά νερωνικά δύο επιπέδων. Η βαθιά μάθηση διακρίνεται από την μηχανική μάθηση ως προς τα δεδομένα που δέχονται τα μοντέλα ως είσοδο για να μάθουν και ως προς τις χρησιμοποιούμενες μεθόδους. Συνήθως τα βαθιά νευρωνικά δίκτυα μπορούν να διαχειριστούν πιο «ωμά» (raw) δεδομένα, από τα οποία μπορούν να αντλήσουν χαρακτηριστικά, περιορίζοντας έτσι την ανάγκη ειδικών, οι οποίοι θα ορίσουν χειροκίνητα τα χαρακτηριστικά που πρέπει να βρει το δίκτυο. Αν για παράδειγμα πρέπει να γίνει διάκριση μεταξύ ανθρώπων και ζώων σε φωτογραφίες από ένα μοντέλο τότε ενώ στην μηχανική μάθηση τα σημαντικότερα χαρακτηριστικά πρέπει να αναγνωριστούν και επισημανθούν από μηχανικούς, στη βαθιά μάθηση το ίδιο το μοντέλο μπορεί να τα αναγνωρίσει. Στην συνέχεια το μοντέλο εκπαιδεύεται μέσω του backpropagation και της κατάβασης κλίσης και μαθαίνει να διακρίνει τις κλάσεις με μεγαλύτερη ακρίβεια. Τα επίπεδα εισόδου και εξόδου στα βαθιά νευρωνικά δίκτυα ονομάζονται ορατά επίπεδα ενώ τα ενδιάμεσα, κρυμμένα επίπεδα. Με τα επίπεδα εισόδου διαβάζεται η πληροφορία εισόδου ενώ με τα επίπεδα εξόδου γίνεται η τελική πρόβλεψη ή κατηγοριοποίηση.

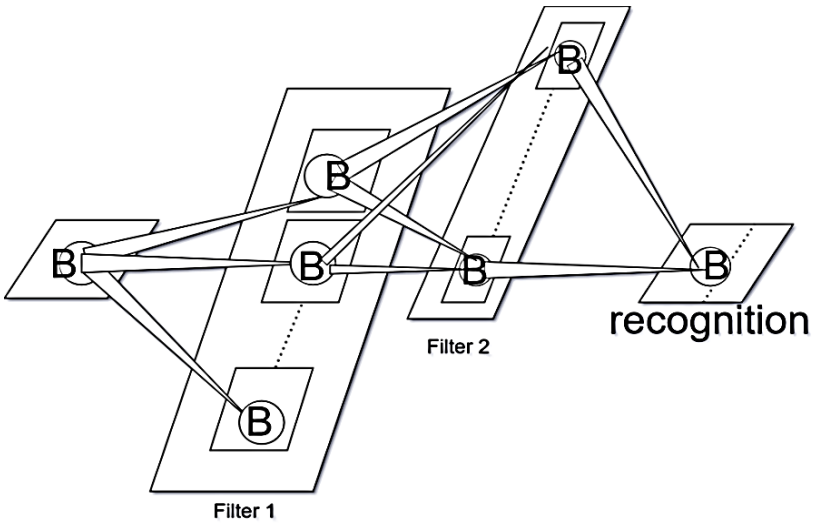
Τα βαθιά νευρωνικά δίκτυα έχουν πολλές εφαρμογές σε πολλούς τομείς της καθημερινότητας των ανθρώπων, όπως στην αστυνόμευση για την ανάλυση και μάθηση επικίνδυνων μοτίβων από την παρακολούθηση βίντεο, την ανάλυση εικόνων ή την επεξεργασία φωνητικών σημάτων, στους τομείς των οικονομικών για την εκτίμηση ρίσκων και την αναγνώριση οικονομικής απάτης και στον τομέα της υγείας, με στόχο την αναγνώριση σε ακτινογραφίες και εικόνες χαρακτηριστικών που αποτελούν σημάδια για σοβαρές ασθένειες, ένας τομέας που γνωρίζει ιδιαίτερη άνθηση. Στον τομέα της τεχνολογίας η βαθιά μάθηση έχει κυρίαρχη θέση εφόσον χρησιμοποιείται εκτενώς στην εικονική πραγματικότητα και την όραση υπολογιστών, στην επεξεργασία φυσικής γλώσσας και την αναγνώριση ομιλίας αλλά και στην αυτοματοποίηση υπηρεσιών με την χρήση chatbots. Χαρακτηριστικό της βαθιάς μάθησης είναι ότι απαιτεί μεγάλη υπολογιστική ισχύ, η οποία καλύπτεται με γραφικές επεξεργαστικές μονάδες (GPU). Οι GPU μπορούν να εκτελούν έναν μεγάλο αριθμό υπολογισμών κατανέμοντας τους σε διαφορετικούς πυρήνες. Οι υπολογιστικοί αυτοί πόροι είναι βέβαια ακριβοί στην κλιμάκωση.



Σχήμα 2.15. *Στα βαθιά νευρωνικά δίκτυα τα κρυμμένα επίπεδα μπορεί να είναι πάρα πολλά.Εδώ παρουσιάζεται ένα δίκτυο με 2 κρυμμένα επίπεδα.*

2.2.1 Συνελικτικά Νευρωνικά δίκτυα (CNN)

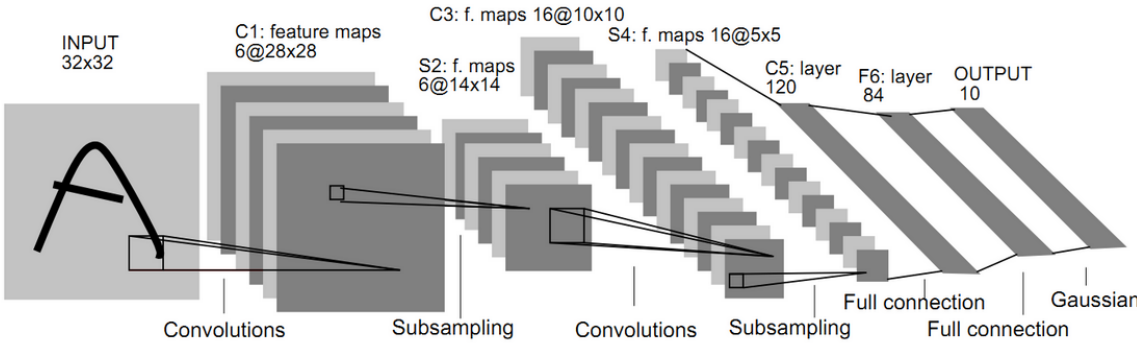
Ένα ευρέως χρησιμοποιούμενο είδος νευρωνικών δικτύων είναι τα συνελικτικά νευρωνικά δίκτυα τα οποία χρησιμοποιούνται εκτενώς στην όραση υπολογιστών, την κατηγοριοποίηση σε εικόνες, βίντεο αλλά και για την αναγνώριση αντικειμένων προσώπων και διαδικασιών.Το πρώτο νευρωνικό δίκτυο που αποτέλεσε τον πρόδρομο των συνελικτικών δικτύων είναι το Neocognitron το οποίο προτάθηκε από τον Kunihiko Fukushima το 1979 και βασίστηκε στο μοντέλο των Hubel και Wiesel του 1959, οι οποίοι μελετώντας τον οπτικό φλοιό πρότειναν ένα μοντέλο για την αναγνώριση προτύπων (pattern recognition tasks). Το neocognitron αποτέλεσε επέκταση των μοντέλων αυτών [45].



Σχήμα 2.16. *Η αρχιτεκτονική του Neocognitron είχε δομή παρόμοια με του σχήματος. Κάθε νευρώνας είχε συγκεκριμένα πεδία υποδοχής και κάθε επόμενο επίπεδο προσέθετε λεπτομέρειες στο εκάστοτε χαρακτηριστικό.*

Το 1998 ο Yann LeCun και οι συνεργάτες του παρουσίασαν το LeNet [46] για την αναγνώριση χειρόγραφων χαρακτήρων (optical character recognition) μέσω του οποίου γεννήθηκε ο όρος Συνελικτικά Νευρωνικά Δίκτυα. Σύμφωνα με την θεωρία που ανέπτυξαν τα συνελικτικά νευρωνικά δίκτυα συνδυάζουν κάποιες αρχιτεκτονικές ιδέες για να εγγυηθούν ότι τα δεδομένα παραμένουν αναλλοίωτα σε κάποιο βαθμό ύστερα από μετατόπιση, αλλαγή κλίμακας, είτε

παραμόρφωση. Αυτές είναι τα τοπικά πεδία υποδοχής (local receptive fields), τα μοιραζόμενα βάρη και χωρική ή χρονική υποδειγματοληψία. Το επίπεδο εισόδου δέχεται εικόνες από χαρακτήρες που είναι κανονικοποιημένοι και κεντρικοποιημένοι. Κάθε κόμβος στο εκάστοτε επίπεδο δέχεται σήμα από ένα σύνολο από κόμβους σε μια μικρή περιοχή του προηγούμενου επιπέδου. Με την χρήση τοπικών πεδίων υποδοχής, δίνεται η δυνατότητα να αντλούν οι νευρώνες οπτικά χαρακτηριστικά, όπως γωνίες και άκρα γραμμών τα οποία αποτελούν τα χαρακτηριστικά των χαρακτήρων. Τα χαρακτηριστικά αυτά συνδυάζονται από μετέπειτα επίπεδα για την αναγνώριση χαρακτηριστικών υψηλότερης τάξης. Οι κόμβοι σε κάθε επίπεδο μοιράζονται τα ίδια βάρη. Το σύνολο των εξόδων των κόμβων σε ένα τέτοιο επίπεδο ονομάστηκε Χάρτης Χαρακτηριστικών (feature map). Οι κόμβοι ενός χάρτη επιτελούν την ίδια λειτουργία σε διαφορετικά μέρη της εικόνας. Ακολούθως ένα πλήρες συνελικτικό επίπεδο αποτελείται από πολλούς χάρτες χαρακτηριστικών. Η άντληση των χαρακτηριστικών μοιάζει με την συνέλιξη καθώς ένας χάρτης χαρακτηριστικών εξετάζει την εικόνα εισόδου με έναν μοναδικό κόμβο και αποθηκεύει τις καταστάσεις του κόμβου στις αντιστοιχιζόμενες θέσεις του χάρτη. Από την παραπάνω λειτουργία προκύπτει και το όνομα συνελικτικά δίκτυα. Η ακριβής τοποθεσία των χαρακτηριστικών είναι άνευ σημασίας [46]. Η μετατόπιση, περιστροφή ή flip των εικόνων μεταβάλλει και τους χάρτες χαρακτηριστικών ανάλογα αλλά η πληροφορία συνεχίζει να βρίσκεται εντοπισμένη σε αυτούς, μια ιδιότητα που προσδίδει στα συνελικτικά δίκτυα ανοχή στις παραμορφώσεις των εισόδων και συνεπώς προσφέρει την δυνατότητα δημιουργίας συνθετικών δεδομένων από τα ήδη υπάρχοντα με τέτοιους απλούς μετασχηματισμούς. Στην σχετική δημοσίευση γίνεται παρουσίαση των επιπέδων υποδειγματοληψίας (sub-sampling layers), τα οποία υπολογίζουν την μέση τιμή και την υποδειγματοληψία, μειώνοντας τις διαστάσεις του χάρτη χαρακτηριστικών. Κάθε κόμβος υπολογίζει τον μέσο των εισόδων, τον πολλαπλασιάζει με έναν συντελεστή που μπορεί να εκπαιδευτεί και εφαρμόζει μία σιγμοειδή συνάρτηση ενεργοποίησης. Καθώς οι χωρικές διαστάσεις μειώνονται ο αριθμός των χαρτών χαρακτηριστικών αυξάνεται. Τα βάρη αποκτούν τιμές μέσω του αλγορίθμου του backpropagation με συνέπεια τα συνελικτικά νευρωνικά δίκτυα να βρίσκουν τα ίδια τα χαρακτηριστικά που απαιτούνται. Επίσης τα μοιραζόμενα βάρη μειώνουν τις ελεύθερες παραμέτρους και το σφάλμα. Οι συγγραφείς του LeNet παρουσιάζουν και την αρχιτεκτονική του LeNet5 που χρησιμοποίησαν στα πειράματά τους.



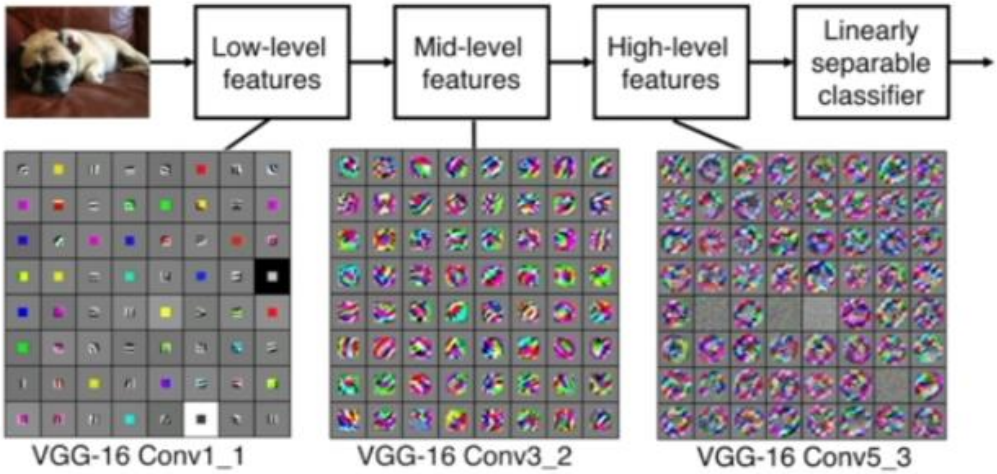
Σχήμα 2.17. *Η αρχιτεκτονική του LeNet 5.Το μοντέλο περιλαμβάνει επίπεδο εισόδου, συνελικτικά επίπεδα, pooling επίπεδα, πλήρως συνδεδεμένα επίπεδα, subsampling επίπεδα, επίπεδο εξόδου, όπου γίνεται η αναγνώριση του συμβόλου. Η εικόνα είναι από την σχετική δημοσίευση του 1998 [46].*

Στην ουσία τα συνελικτικά νευρωνικά δίκτυα βασίζονται στην χρήση μοιραζόμενων βαρών και συνελικτικών πυρήνων ή αλλιώς φίλτρων που ολισθαίνουν στα χαρακτηριστικά εισόδου και παρέχουν αμεταβλητότητα στο μετασχηματισμό της μεταφοράς. Λόγω του downsampling που γίνεται στα δεδομένα εισόδου σε πολλά συνελικτικά δίκτυα η αμεταβλητότητα αυτή βέβαια αναιρείται. Σε αντίθεση με τα Perceptrons πολλαπλών επιπέδων (MLP), τα οποία χρησιμοποιούν πλήρως συνδεδεμένα δίκτυα και είναι ευάλωτα σε overfitting, τα CNN εφαρμόζουν διαδοχικά φίλτρα χτίζοντας επίπεδα αυξανόμενης πολυπλοκότητας με στόχο την αναγνώριση των ζητούμενων στόχων. Σημαντική παρατήρηση είναι ότι σε αντίθεση με άλλα νευρωνικά δίκτυα τα χαρακτηριστικά μαθαίνονται από τα ίδια τα συνελικτικά δίκτυα και δεν είναι κατασκευασμένα με το χέρι. Αν τα συνελικτικά δίκτυα είχαν αντικατασταθεί από πλήρως συνδεδεμένα δίκτυα θα υπήρχε τεράστια ανάγκη αριθμό νευρώνων εφόσον κάθε pixel είναι χρήσιμο χαρακτηριστικό και γι'αυτόν το λόγο τα πλήρως συνδεδεμένα δίκτυα είναι μη πρακτικά για υψηλών διαστάσεων δεδομένα. Τα CNN είναι κατάλληλα για δεδομένα υψηλών διαστάσεων οργανωμένα σε πλέγματα καθώς οι σχέσεις γειτνίασης και γενικά οι χωρικές σχέσεις των χαρακτηριστικών λαμβάνονται υπόψιν στα επίπεδα συνέλιξης και pooling.

Αρχιτεκτονική των Συνελικτικών Δικτύων

Φίλτρα (Filters)

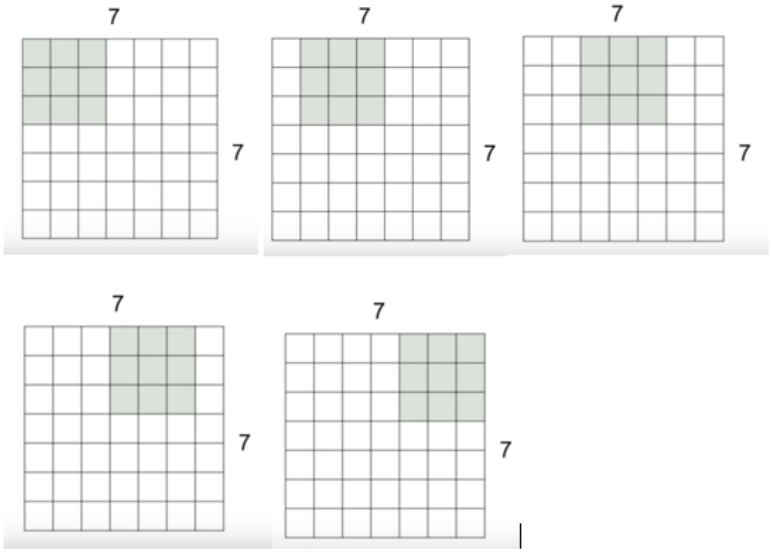
Για την εφαρμογή της συνέλιξης και την άντληση των χαρακτηριστικών των δεδομένων εισόδου χρησιμοποιούνται φίλτρα τα οποία, στον τομέα της επεξεργασίας σήματος αφαιρούν συγκεκριμένα χαρακτηριστικά.Σ τον τομέα της κατηγοριοποίησης (classification) συχνά χρησιμοποιούμενα φίλτρα είναι τα Butterworth, Chebyshev, Bessel, Gaussian. Τα φίλτρα κατατάσσονται σε ποιότητα ανάλογα με την τάξη των πολυωνύμων της συνάρτησης μεταφοράς τους. Τα φίλτρα υψηλότερης τάξης είναι πιο ποιοτικά.



Σχήμα 2.18. *Το φίλτρο σε κάθε κελί αποτελεί κάποιο συγκεκριμένο χαρακτηριστικό που ο κάθε νευρώνας αναζητάει για να δώσει την μέγιστη τιμή της συνάρτησης ενεργοποίησης. Τα φίλτρα στοιβάζονται σε κάθε επίπεδο και σταδιακά εμπλουτίζουν τα χαρακτηριστικά με περισσότερες λεπτομέρειες ως προς το τι αναζητάται. Η εικόνα είναι από τις διαλέξεις Stanford για deep learning [47].*

Συνελικτικά επίπεδα (Convolutional layers)

Βασικό χαρακτηριστικό των συνελικτικών νευρωνικών δικτύων είναι τα συνελικτικά τους επίπεδα.Σ ε ένα CNN, η είσοδος είναι ένας τανυστής με 3 διαστάσεις *: (αριθμός εισόδων) x (ύψος εισόδου) x (πλάτος εισόδου).*Σ την περίπτωση εικόνων η διέλευση της εικόνας μέσα από το συνελικτικό επίπεδο την μετατρέπει στην πιο αυθαίρετη μορφή του χάρτη χαρακτηριστικών, με διαστάσεις: *(αριθμός εισόδων) x (ύψος χάρτη) x (πλάτος χάρτη) x (κανάλια χάρτη) ,* όπου τα κανάλια μπορεί να είναι τα 3 ,σε μία RGB εικόνα. Δηλαδή οι νευρώνες οργανώνονται σε 3 διαστάσεις . Κάθε συνελικτικός νευρώνας επεξεργάζεται μόνο δεδομένα του πεδίου υποδοχής του. Τα συνελικτικά επίπεδα εφαρμόζουν φίλτρα που ονομάζονται πυρήνες, οι οποίοι πολλαπλασιάζονται με τα δεδομένα εισόδου μιας συγκεκριμένης περιοχής των δεδομένων. Στην συνέχεια ολισθαίνουν πάνω στα δεδομένα στο πλάτος και το ύψος και η διαδικασία επαναλαμβάνεται. Τα παραγόμενα γινόμενα αποτελούν τον χάρτη χαρακτηριστικών και αντιστοιχούν σε αναγνωρισμένες ιδιότητες των δεδομένων, όπως γωνίες, διαγώνιες γραμμές, συγκεκριμένο χρώμα κ.α.. Κάθε φίλτρο παράγει έναν ξεχωριστό χάρτη χαρακτηριστικών δύο διαστάσεων. Οι χάρτες αυτοί έχουν συνολικά διάσταση ίση με το βάθος.

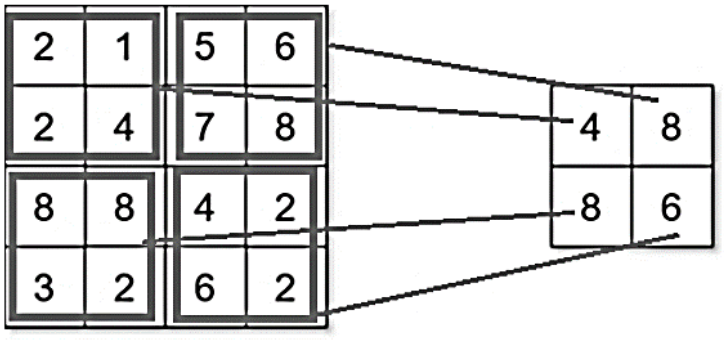


Σχήμα 2.19. Συνέλιξη εισόδου 7x7 με φίλτρο 3x3 (kernel size), βήμα(stride) 1 και μηδενικό γέμισμα (padding). Κάθε τετράγωνο του φίλτρου πολλαπλασιάζεται με κάθε τετράγωνο της εισόδου στις αντίστοιχες θέσεις. Στην συνέχεια το φίλτρο ολισθαίνει κατά τον άξονα τον x κατά το βήμα=1 και η διαδικασία επαναλμβάνεται. Το ίδιο γίνεται και ως προς τον άξονα τον y, εδώ παραλείπεται. Η εικόνα είναι από τις διαλέξεις Stanford για deep learning [47]

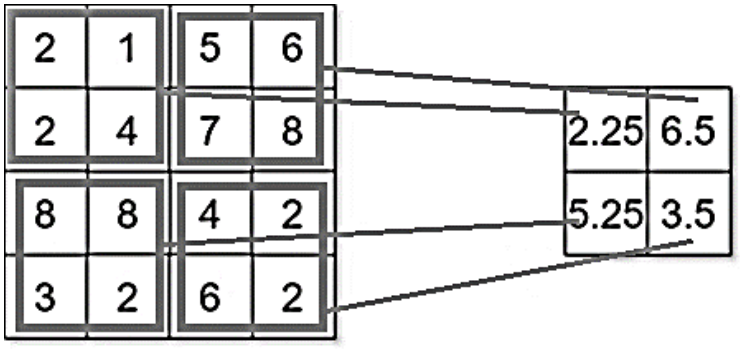
.

Pooling επίπεδα

Τα επίπεδα pooling χρησιμοποιούνται για την μείωση των διαστάσεων των δεδομένων συνδυάζοντας τις εξόδους από ομάδες ή συστάδες (clusters) νευρώνων ενός επιπέδου σε λιγότερους νευρώνες ή και σε έναν μόνο νευρώνα στο επόμενο επίπεδο. Το τοπικό pooling συνδυάζει μικρές συστάδες όπως 2x2 .Το global pooling δρα σε όλους τους νευρώνες του χάρτη χαρακτηριστικών. Πιο πολυσύχναστα είδη pooling είναι το max pooling και το average pooling. Το max pooling χρησιμοποιεί την μέγιστη τιμή από κάθε συστάδα νευρώνων και την τοποθετεί στον χάρτη χαρακτηριστικών ενώ το average pooling λαμβάνει και τοποθετεί στον χάρτη τον μέσο όρο κάθε συστάδας. Το max pooling εγγυάται την συλλογή των σημαντικών χαρακτηριστικών ύστερα από μετατοπίσεις, περιστροφές, αλλαγές κλίμακας του δεδομένου εισόδου. Τα επίπεδα pooling τοποθετούνται συνήθως μετά από τα συνελικτικά επίπεδα και δρουν πάνω στους χάρτες χαρακτηριστικών με στόχο την μέιωση των διαστάσεων του χάρτη που εισέλθει στο επόμενο επίπεδο. Οι pooled χάρτες χαρακτηριστικών αποτελούν πιο συνεπτυγμένες περιγραφές του χάρτη χαρακτηριστικών.



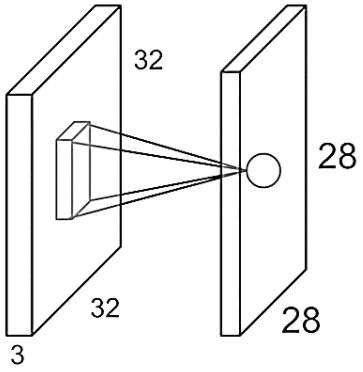
Σχήμα 2.20. *Max pooling με φίλτρο 2 x 2 και βήμα = 2. Από κάθε τετράδα αριθμών εισόδου 4 x 4 συλλέγεται ο μέγιστος και η έξοδος έχει μικρότερες διαστάσεις 2 x 2.*



Σχήμα 2.21. *Average pooling με φίλτρο 2x2 και βήμα=2. Από κάθε τετράδα αριθμών εισόδου 4x4 συλλέγεται ο μέσος όρος τους και η έξοδος έχει μικρότερες διαστάσεις 2x2.*

Πεδίο Υποδοχής (Receptive field)

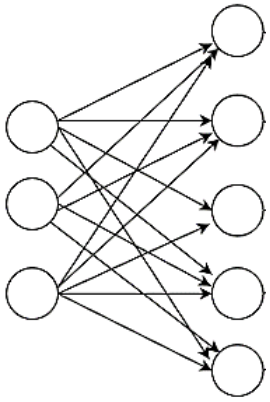
Κάθε νευρώνας δέχεται σήματα από ένα συγκεκριμένο αριθμό από νευρώνες του προηγούμενου επιπέδου, κάτι το οποίο ισχύει τόσο στα τεχνητά όσο και στα εγκεφαλικά νευρωνικά δίκτυα. Στα συνελικτικά επίπεδα κάθε νευρώνας δέχεται την είσοδο από μια συγκεκριμένη περιοχή του προηγούμενου επιπέδου, που ονομάζεται πεδίο υποδοχής. Αυτό έρχεται σε αντιδιαστολή με τα πλήρως συνδεδεμένα επίπεδα όπου κάθε νευρώνας δέχεται τις εξόδους όλων των νευρώνων του προηγούμενου επιπέδου. Με την χρήση των πεδίων υποδοχής και την ιδέα της τοπικής συνδεσιμότητας (local connectivity) τα φίλτρα στοιβάζονται το ένα πάνω στο άλλο ώστε το δίκτυο να δημιουργεί αναπαραστάσεις μικρών τμημάτων της εισόδου και στην συνέχεια φτιάχνει αναπαραστάσεις μεγαλύτερων τμημάτων.



Σχήμα 2.22. Το πεδίο υποδοχής για τον νευρώνα του χάρτη ενεργοποίησης στα δεξιά είναι μια υποπεριοχή της εισόδου. Με άλλα λόγια κάθε νευρώνας ενεργοποιείται από συγκεκριμένα τμήματα της εισόδου. Η εικόνα είναι από τις διαλέξεις Stanford για deep learning [47].

Πλήρως συνδεδεμένα επίπεδα(Fully connected layers)

Πλήρως συνδεδεμένο είναι το επίπεδο όπου κάθε νευρώνας συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου. Η αρχιτεκτονική αυτή ταυτίζεται με το Perceptron πολλαπλών επιπέδων (MLP).



Σχήμα 2.23. *Ένα πλήρως συνδεδεμένο επίπεδο. Κάθε νευρώνας συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου.*

Βάρη(Weights)

Τα βάρη μαζί με τις τιμές των biases ονομάζονται φίλτρα (filters) και εκφράζουν χαρακτηριστικά της εισόδου. Στα συνελικτικά νευρωνικά δίκτυα μπορεί να είναι μοιραζόμενα για πολλούς νευρώνες, κάτι το οποίο μειώνει τις απαιτήσεις σε μνήμη.

Εξισώσεις συνελικτικών και pooling επιπέδων

Συνελικτικό Επίπεδο

Οι χάρτες χαρακτηριστικών των συνελικτικών επιπέδων καθορίζονται από τις υπερπαραμέτρους του βάθους *D1*, του αριθμού των φίλτρων *K*, του μεγέθους πυρήνα *F*, του μεγέθους βήματος (stride) *S* και του μεγέθους γεμίσματος (padding) *P*.

- Το βάθος *D1* ορίζει τον αριθμό των καναλιών και την τρίτη διάσταση του τανυστή *W1 x H1 x D1* που δέχεται το συνελικτικό επίπεδο.
- Η υπερπαράμετρος *K* ορίζει τον αριθμό των φίλτρων που θα εφαρμοστούν στον τανυστή.
- Το βήμα(stride) ορίζει πόσο θα ολισθήσει το φίλτρο ύστερα από κάθε συνέλιξη για την επανάληψη της διαδικασίας στις γειτονικές περιοχές. Με βήμα 1 τα φίλτρα μετακινούνται κατά το ακριβώς διπλανό εικονοστοιχείο ενώ με 2 ή παραπάνω, αγνοούν κάποια γειτονικά εικονοστοιχεία. Έτσι παράγεται διαφορετικό μέγεθος εξόδου ανάλογα με το μέγεθος του βήματος ή και τον συνδυασμό του με το γέμισμα.
- Το padding καθορίζει πόσα μηδενικά ή άλλες προκαθορισμένες τιμές θα γεμίσουν τα άκρα του τανυστή προκειμένου να μπορεί να εφαρμοστεί συνέλιξη με το επιθυμητό βήμα και ο τελικός χάρτης να έχει τις επιθυμητές διαστάσεις.
- Το μέγεθος πυρήνα καθορίζει το ύψος *H* και το πλάτος *W* στο οποίο εκτείνεται το φίλτρο.

Έτσι στον παραπάνω τανυστή *W1 x H1 x D1* αν εφαρμοστεί συνελικτικό επίπεδο με τα παραπάνω χαρακτηριστικά ο νέος τανυστής που προκύπτει θα έχει διαστάσεις *W2 x H2 x D2* που λαμβάνονται από τις ακόλουθες σχέσεις [47]:

$$W2 = (W1 - F + 2P)/S + 1 \quad (2.7)$$

$$H2 = (H1 - F + 2P)/S + 1 \quad (2.8)$$

$$D2 = K \tag{2.9}$$

Αν οι παράμετροι διαμοιράζονται τότε για κάθε φίλτρο τα βάρη είναι *F x F x D1* και συνολικά προκύπτουν *K x (F x F x D1)* βάρη και *K biases.* Στον τελικό χάρτη το *i-οστό κομμάτι μεγέθους W2 x H2* προκύπτει από την συνέλιξη του *i-οστού* φίλτρου με τα δεδομένα εισόδου και έχοντας βήμα *S*, μετατοπισμένης κατά το του *i*-οστό bias.

Pooling Επίπεδο

Όσον αφορά τα pooling επίπεδα,αν ο Τανυστής *W1 x H1 x D1* αποτελεί την είσοδο στο pooling επίπεδο τότε έχοντας:

- μεγέθος βήματος (stride) *S*
- μέγεθος πυρήνα *F*

παράγεται νέος Τανυστής με διαστάσεις *W2 x H2 x D2* ως εξής [47]:

$$W2 = (W1 - F)/S + 1 \tag{2.10}$$

$$H2 = (H1 - F)/S + 1 \tag{2.11}$$

$$D2 = D1 \tag{2.12}$$

2.2.2 Χρησιμοποιούμενα Επίπεδα

Σε αυτήν τη μελέτη γίνεται χρήση της βιβλιοθήκης **Tensorflow** και του deep learning **API Keras** για την κατασκευή διαφορετικών επιπέδων απαραίτητων στην σύνθεση πολύπλοκων αρχιτεκτονικών μοντέλων βαθιάς μάθησης [48].

- Πυκνά (Dense, Densely connected layers), Πλήρως Συνδεδεμένα επίπεδα (FC).

Αποτελεί την απλούστερη μορφή επιπέδου, όπου κάθε νευρώνας συνδέεται με όλους τους νευρώνες στο προηγούμενο επίπεδο.

- Συνελικτικά επίπεδα δύο και 3 διαστάσεων

Χρησιμοποιούνται για να περιγράψουν συνέλιξη σε 2 ή 3 διαστάσεις με τα δεδομένα εισόδου.

- Pooling επίπεδα

Χρησιμοποιούνται για την μείωση των διαστάσεων των δεδομένων που περνάνε από επίπεδο σε επίπεδο και μπορεί να διαιρεθούν περαιτέρω σε max pooling και average pooling που περιγράφτηκαν παραπάνω.

- Batch Normalization επίπεδα

Το Batch Normalization είναι μέθοδος που χρησιμοποείται για την κανονικοποίηση των δεδομένων σε κάθε επίπεδο ώστε να είναι κεντρικοποιημένα και να έχουν την ίδια κλίμακα. Αυτό είναι απαραίτητο επειδή σε κάθε επίπεδο του δικτύου τα δεδομένα μεταβάλλονται. Η μέθοδος επιταχύνει την εκπαίδευση των τεχνητών νευρωνικών δικτύων και τα κάνει πιο σταθερά.

$$\widehat{x^{(k)}} = \frac{x^{(k)} - E[X^{(k)}]}{\sqrt{Var[x^{(k)}]}} \tag{2.13}$$

Η εξίσωση [49] υπολογίζει τον εμπειρικό μέσο και την διασπορά ανεξάρτητα για κάθε διάσταση. Το Batch Normalization επίπεδο συνήθως τοποθετείται μετά από πλήρως συνδεδεμένα επίπεδα ή συνελικτικά επίπεδα και πριν την μη γραμμική συνάρτηση ενεργοποίησης. Η μέθοδος μπορεί να χρησιμοποιηθεί σε διάφορες μορφές δικτύων.

- Activation επίπεδα

Είναι τα επίπεδα στα οποία εφαρμόζεται μια κατάλληλη συνάρτηση ενεργοποίησης στην είσοδο πριν αυτή μεταβεί στο επόμενο επίπεδο

- Ανάστροφα Συνελικτικά επίπεδα

Τα επίπεδα αυτά (transpose convolutional) επιτελούν αύξηση των διαστάσεων του χάρτη χαρακτηριστικών (upsampling) σε αντίθεση με τα τυπικά συνελικτικά επίπεδα και ονομάζονται και deconvolution επίπεδα. Ορίζονται όπως και τα συνελικτικά επίπεδα, με την έννοια ότι χρειάζονται το βήμα και το γέμισμα για να υπολογιστούν. Όμως οι τιμές αυτές είναι οι υποθετικές τιμές που αν χρησιμοποιούνταν στην έξοδο θα έδιναν την επιθυμητή είσοδο. Το βήμα για παράδειγμα αναφέρεται στο επίπεδο εξόδου και όχι εισόδου. Επίσης με την κατάλληλη τιμή γεμίσματος οι διαστάσεις της εξόδου θα είναι είτε ίσες με αυτές τις εισόδου είτε μεγαλύτερες [50, 51, 52].

- Flatten επίπεδα

Τα επίπεδα αυτά δέχονται συνήθως έναν τανυστή με διαστάσεις $(N1\ x\ N2\ x\ \dots Nn)$ και μετασχηματίζει τον τανυστή σε έναν πίνακα μίας διάστασης με το μέγεθος της να είναι ίσο με το γινόμενο όλων των διαστάσεων του τανυστή $Nf\ =\ N1\ x\ N2\ x\ \dots x\ Nn$. Ο νέος πίνακας έχει διάσταση (Nf).

2.2.3 Μετρικές και Συναρτήσεις Απωλειών για Εντοπισμό Αντικειμένων (object detection)

Ο εντοπισμός αντικειμένων αποτελεί έναν δημοφιλή τομέα της όρασης υπολογιστών. Στόχος της είναι να βρεθούν αντικείμενα ενδιαφέροντος σε εικόνες ,βίντεο ή και τρισδιάστατες αναπαραστάσεις, στις οποίες υπάρχουν πολλά διαφορετικά αντικείμενα, δηλαδή αντικείμενα διαφορετικών κλάσεων.

Σημαντικότερες μετρικές που χρησιμοποιούνται στον τομέα αυτό είναι η μέση ακρίβεια (average precision) και ο μέσος όρος της μέσης ακρίβειας (mean average precision).

Για να προσδιορισθούν οι όροι αυτοί πρέπει πρώτα να αναφερθούν οι στατιστικές μετρικές που χρησιμοποιούνται για τον εντοπισμό κλάσεων :

- P (θετικά), τα στιγμιότυπα που ανήκουν σε μία κλάση
- N (αρνητικά), τα στιγμιότυπα που δεν ανήκουν σε μία κλάση

Με βάση τα παραπάνω ορίζονται τα εξής :

- TP (αληθώς θετικά), το σύνολο των πραγματικά θετικών στιγμιότυπων μιας κλάσης, που αναγνώρισε το μοντέλο, δηλαδή τα στιγμιότυπα που υπέθεσε ότι ανήκουν σε μια κλάση και πράγματι ανήκουν σε αυτή.
- FP (ψευδώς θετικά), το σύνολο των στιγμιοτύπων που το μοντέλο θεώρησε λανθασμένα ότι ανήκουν σε μία κλάση αλλά στην πραγματικότητα δεν ανήκουν σε αυτήν
- FN (ψευδώς αρνητικά), το σύνολο των στιγμιότυπων που το μοντέλο απέρριψε ως μη σχετικά της κλάσης ενώ στην πραγματικότητα αποτελούν στιγμιότυπα της κλάσης. Έτσι το μοντέλο χάνει πραγματικά θετικά στιγμιότυπα (ground truths).
- TN (αληθώς αρνητικά), το σύνολο των στιγμιότυπων που το μοντέλο απέρριψε και πράγματι δεν ανήκουν στην κλάση. Συνήθως ονομάζονται background περιοχή και δεν χρησιμοποιούνται στον εντοπισμό αντικειμένων διότι δεν υπάρχει σαφής περιγραφή στις ετικέτες για τις περιοχές που δεν ανήκουν στην κλάση.

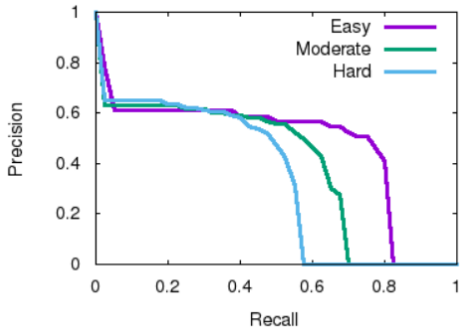
Με βάση τα παραπάνω παρατίθενται οι δύο στατιστικές μετρικές για την εκτίμηση των προβλέψεων του μοντέλου στην κατηγοριοποίηση σε μία κλάση.

$$precision = \frac{TP}{TP+FP} \tag{2.14}$$

$$recall = \frac{TP}{TP+FN} \tag{2.15}$$

Η μετρική precision ουσιαστικά υπολογίζει πόσα από τα στιγμιότυπα που το μοντέλο θεώρησε θετικά είναι πραγματικά θετικά, δηλαδή των ακρίβεια των θετικών προβλέψεων ή αλλιώς την προβλεπόμενη αξία των θετικών (positive predictive value) ενώ η μετρική recall που ονομάζεται και sensitivity υπολογίζει πόσα θετικά το μοντέλο υπολόγισε σε σχέση με όλα τα θετικά που υπάρχουν [53]. Είναι εύκολο να δει κανείς ότι τα θετικά που υπάρχουν συνολικά ταυτίζονται με το άθροισμα των αληθώς θετικών που βρήκε το μοντέλο και αυτών που δεν βρήκε, δηλαδή των ψευδώς αρνητικών.

Με βάση τις δύο παραπάνω μετρικές σχηματίζεται η καμπύλη precision-recall curve που χρησιμοποιείται για την επιλογή του ιδανικού φράγματος που να εξισορροπεί τις 2 μετρικές. Στον εντοπισμό αντικειμένων αυτό είναι το IoU threshold.

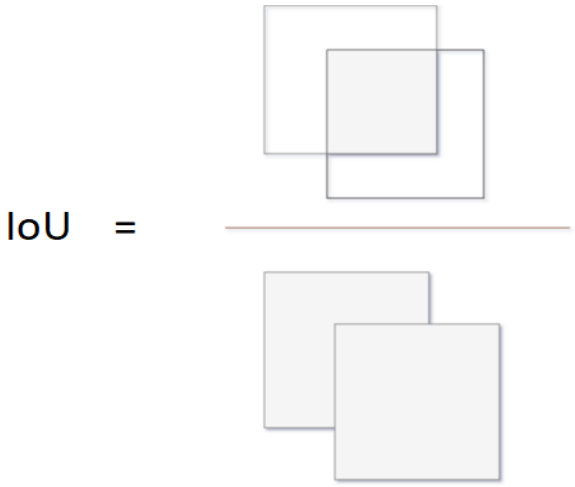


Σχήμα 2.24. *Καμπύλη precision recall. Καθώς αυξάνεται το recall τείνει να φθίνει το precision και αντίστροφα.*

Στην συνέχεια ορίζεται ο τύπος της ***Τομής προς Ένωση.*** Ο τύπος αυτός χρησιμοποιείται στον εντοπισμό αντικειμένων για να εκτιμήσει τον βαθμό της επικάλυψης (overlap) μεταξύ των προβλέψεων και του πραγματικού στιγμιότυπου της κλάσης [54, 55].

$$\boldsymbol{IoU} = \frac{\boldsymbol{\varepsilon\mu\beta\alpha\delta\acute{o}\nu(gt \cap prediction)}}{\boldsymbol{\varepsilon\mu\beta\alpha\delta\acute{o}\nu(gt \cup prediction)}} = \frac{|\boldsymbol{gt \cap prediction}|}{|\boldsymbol{gt}|+|\boldsymbol{prediction}|-|\boldsymbol{gt \cap prediction}|} = J(gt, prediction) \qquad (2.16)$$

Όπου gt είναι το πραγματικό στιγμιότυπο της κλάσης και $prediction$ είναι το προβλεπόμενο στιγμιότυπο. Η σχέση ονομάζεται και Jaccard index ή Jaccard similarity coefficient. Πιο απλά η σχέση οπτικοποιείται ως εξής :

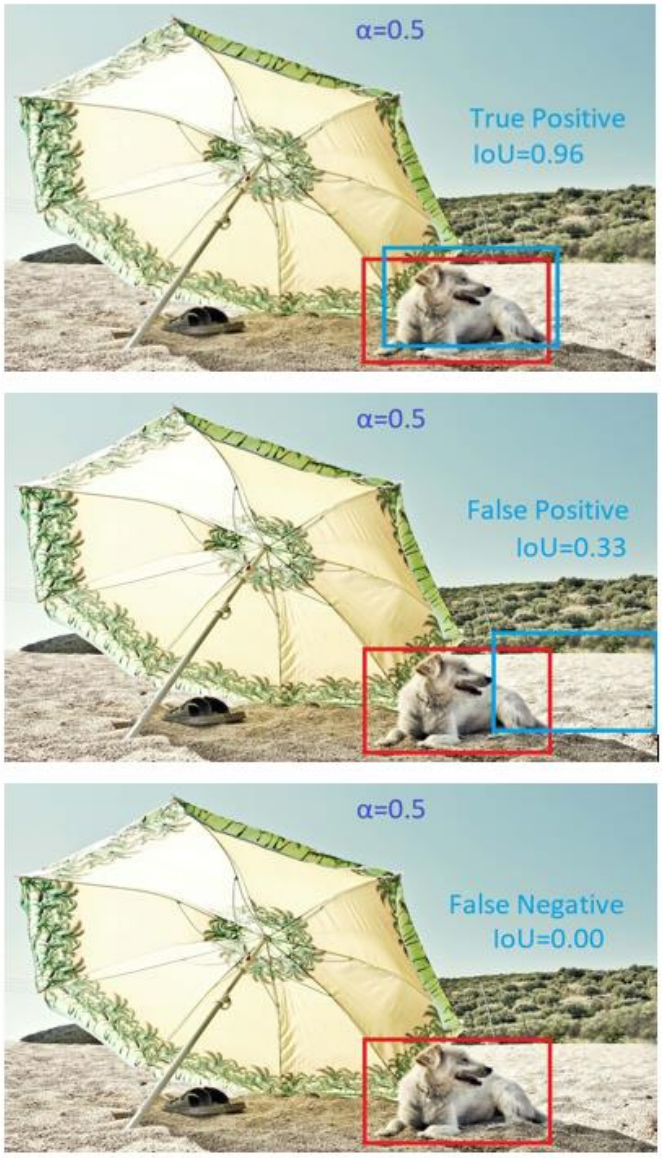


Σχήμα 2.25. *Η IoU ορίζεται ως το εμβαδόν της τομής προς το εμβαδόν της ένωσης των 2 περιοχών.*

Το IoU έχει σύνολο τιμών στο διάστημα (0,1), με το 0 να δηλώνει καμία επικάλυψη και το 1 να δηλώνει πλήρης επικάλυψη, όπου η πρόβλεψη ταυτίζεται με το πραγματικό

στιγμιότυπο (Ground truth). Για την εκτίμηση των προβλέψεων του μοντέλου συνήθως χρησιμοποιείται ένα φράγμα για το IoU (IoU threshold), πάνω από το οποίο η πρόβλεψη θεωρείται επιτυχής και θεωρείται TP και κάτω από το οποίο (μεγαλύτερο όμως το 0) θεωρείται FP. Το κατώφλι αυτό καθορίζεται από τους σχεδιαστές-μηχανικούς και εξαρτάται από το πρόβλημα που πρέπει να αντιμετωπιστεί. Αν το μοντέλο δεν παράξει κυτίο αναγνώρισης για ένα πραγματικό στιγμιότυπο τότε το αποτέλεσμα είναι 0 και θεωρείται ότι το μοντέλο εμφανίζει FN.

$$prediction = \begin{cases} positive, & IoU \geq Threshold \\ negative, & IoU < Threshold \end{cases}$$



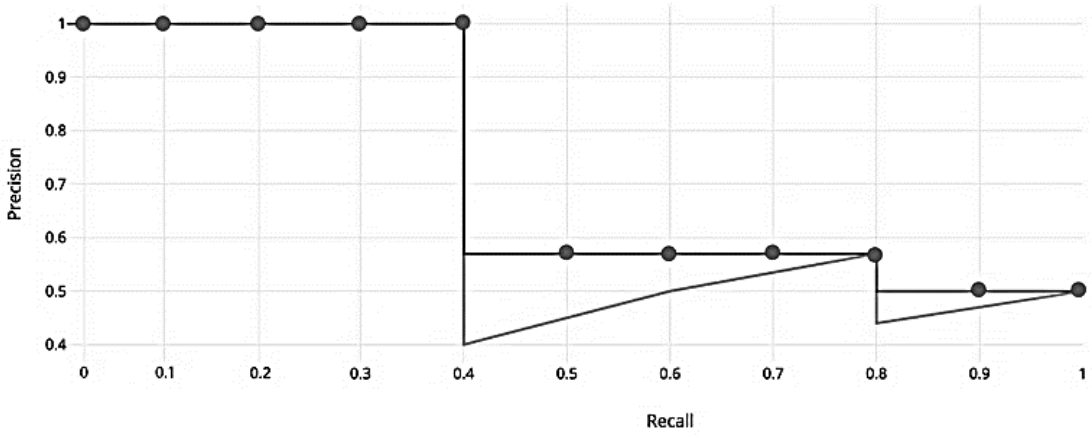
Εικόνα 2.1. *Το IoU threshold είναι 0.5.Στην πάνω εικόνα το IoU είναι μεγαλύτερο από το IoU threshold οπότε η πρόβλεψη λαμβάνεται ως True Positive.Στην μεσαία εικόνα είναι μικρότερο του threshold οπότε λαμβάνεται ως False Positive. Στην κάτω εικόνα το μοντέλο δεν προβλέπει κυτίο για το στιγμιότυπο, συνεπώς «χάνει» το στιγμιότυπο και το IoU =0, False Negative. Φωτογραφία από τον Marko Milivojevic.*

Η μετρική που χρησιμοποιείται συχνά για τον εντοπισμό και την αναγνώριση αντικειμένων είναι η Average Precision η οποία προκύπτει από την καμπύλη precision recall curve και συνοψίζει το σχήμα της. Πιο συγκεκριμένα ορίζεται ως η μέση τιμή της precision σε 11 ομοιόμορφα κατανεμημένα επίπεδα recall *[0,0.1,….,1]*. Σε κάθε ένα από τα 11 επίπεδα recall $r \in [0, 0.1, \dots, 1]$ λαμβάνεται η μέγιστη τιμή του precision που ανήκει σε επίπεδο δεξιά από αυτό που εξετάζεται, στο οποίο το recall είναι μεγαλύτερο από το recall του παρόντος επιπέδου. Στην συνέχεια κάθε όρος precision που λαμβάνεται αθροίζεται με όλους τους υπόλοιπους και το τελικό άθροισμα διαιρείται με το 11 ώστε να υπολογιστεί ο μέσος όρος. Στην ουσία υπολογίζεται το εμβαδόν της καμπύλης αλλά τροποποιημένο έτσι ώστε η καμπύλη να είναι φθίνουσα χωρίς τις αυξομειώσεις του precision [56].

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p_{int}(r) \tag{2.17}$$

$$p_{int}(r) = max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{2.18}$$

Όπου $p_{int}(r)$ είναι η τιμή precision στο επίπεδο με recall $\tilde{r}$. Σχηματικά οι δύο σχέσεις μπορούν να εξηγηθούν με το παρακάτω σχήμα :



Σχήμα 2.26. *Όπου φαίνονται τα 11 επίπεδα recall και σε κάθε επίπεδο λαμβάνεται το precision του δεξιότερου επιπέδου με το μεγαλύτερο recall.Η γραμμή με την μορφή «σκαλοπατιού» ορίζει τα* $p_{int}(r)$ *ενώ η γραμμή «zig-zag» ορίζει την αρχική καμπύλη precision-recall curve. Εικόνα από [57].*

Επιπλέον για την αξιόλογηση του προσανατολισμού των προβλεπόμενων κυτίων αναγνώρισης χρησιμοποιείται συχνά η μετρική Average Orientation Similarity (AOS) [58].

$$AOS = \frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1\}} max_{\tilde{r}:\tilde{r}\geq r} s(\tilde{r}) \qquad (2.19)$$

Η orientation similarity $s \in [0,1]$ στο επίπεδο recall r αποτελεί μία κανονικοποιημένη μορφή της ομοιότητας συνημιτόνων που ορίζεται ως :

$$s(r) = \frac{1}{|D(r)|}\sum_{i\in D(r)} \frac{1+cos\,\Delta_\theta^{(i)}}{2}\delta_i$$

Όπου το $D(r)$ ορίζει το σύνολο όλων των προβλέψεων στο επίπεδο recall r και $\Delta_\theta^{(i)}$ είναι η διαφορά σε γωνίες μεταξύ του πραγματικού στιγμιότυπου και της πρόβλεψης i. Για να ασκηθεί κάποιου είδους ποινή στην περίπτωση που πολλές προβλέψεις έχουν γίνει για ένα συγκεκριμένο αντικείμενο(στιγμιότυπο), το δ_i τίθεται ίσο με 1 αν η πρόβλεψη i έχει αντιστοιχιστεί σε πραγματικό στιγμιότυπο, με την έννοια ότι έχει επικάλυψη πάνω από το δεδομένο φράγμα και τίθενται 0 αν δεν έχει αντιστοιχιστεί.

Στην συγκεκριμένη εργασία η αξιολόγηση των μοντέλων γίνεται στο Kitti Dataset το οποίο αναλύεται στο επόμενο κεφάλαιο και όσον αφορά τον τρισδιάστατο εντοπισμό αντικειμένων χρησιμοποιείται η μετρική AP και AOS .

Συναρτήσεις Απωλειών για τον Εντοπισμό Αντικειμένων

Στον τομέα της θεωρίας πληροφορίας η cross entropy [59] μεταξύ δύο κατανομών πιθανότητας p και q στο ίδιο σύνολο από γεγονότα μετράει τον μέσο αριθμό από bits που απαιτούνται για την αναγνώριση ενός γεγονότος που αντλείται από το σύνολο αυτό, αν η κωδικοποίηση του συνόλου είναι βελτιστοποιημένη για την εκτιμωμένη κατανομή q και όχι για την πραγματική p. Για δύο διακριτές κατανομές πιθανότητας p και q ορίζεται ως :

$$H(p,q) = -\sum_{x \in X} p(x) log\ q(x)$$

Με βάση την cross entropy ορίζεται η cross entropy loss [60] στον τομέα της κατηγοριοποίησης πολλαπλών κλάσεων.

$$L_{CE} = -\sum_{c=1}^{K} y_{o,c}\, log(p_o, c)$$

Όπου K το πλήθος των κλάσεων, $y_{o,c}$ είναι 0 αν η κλάση c είναι η σωστή κλάση για το στιγμιότυπο o ή 1 αν το o δεν ανήκει σε αυτήν την κλάση, p η προβλεπόμενη πιθανότητα το στιγμιότυπο o να ανήκει στην κλάση c .

Το παραπάνω άθροισμα ορίζεται για όλα τα στιγμιότυπα του συνόλου δεδομένων το οποίο μπορεί να είναι σύνολο εκπαίδευσης, επαλήθευσης ή και testing.

Για τον υπολογισμό του σφάλματος κατηγοριοποίησης σε ένα μοντέλο για τον εντοπισμό αντικειμένων χρησιμοποιείται η συνάρτηση Binary Cross Entropy Loss εφ'όσον η αναζήτηση ουσιαστικά περιορίζεται σε δύο κλάσεις, με την μία να είναι η κλάση ενδιαφέροντος, η κλάση στην οποία ανήκουν τα αντικείμενα που επιχειρείται να εντοπιστούν και η άλλη είναι το background που είναι οτιδήποτε άλλο δεν ανήκει στην κλάση αυτή. Τότε η cross entropy loss γίνεται binary cross entropy loss και ορίζεται ως εξής :

$$L_{BCE} = -\sum_{i=1}^{2} t_i \, log(p_i) =$$
$$-[t_1 log(p_1) + t_2 log(p_2)] =$$
$$-[t log(p) + (1-t) log(1-p)] \quad (2.20)$$

Όπου t_i είναι 0 αν ανήκει στην κλάση i το στιγμιότυπο που εξετάζεται και 1 αν δεν ανήκει ,και p_i είναι η softmax πιθανότητα να ανήκει στην i κλάση.Τα p_1 και p_2 είναι συμπληρωματικά ως πιθανότητες οπότε $p1 + p2 = 1,\ p1 = 1 - p2$.

Στο Fast R-CNN [61] ορίζεται συνάρτηση απωλειών κατάλληλη για τον υπολογισμό του σφάλματος μεταξύ των πραγματικών κυτίων που περιβάλλουν τα στιγμιότυπα της κλάσης και των προβλεπόμενων κυτίων από το μοντέλο. Αυτή ονομάζεται Localization Loss , L_{loc} .

$$L_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i^u - v_i) \quad (2.21)$$

Όπου

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & αν\ |x| < 1 \\ |x| - 0.5\,, & οπουδήποτε\ αλλού \end{cases}$$

Η συνάρτηση αυτή χρησιμοποιείται για να αντιμετωπιστεί το πρόβλημα των exploding gradients.

Στην εργασία αυτή χρησιμοποιείται η binary cross entropy loss για τον υπολογισμό του classification loss ενώ επιπλέον χρησιμοποιείται η συνάρτηση , L_{loc} για τον υπολογισμό του regression loss (η απώλεια πραγματικού κυτίου και προβλεπόμενου κυτίου αναγνώρισης).

Η πιο εξειδικευμένη συνάρτηση που χρησιμοποείται στα πειράματα περιγράφεται στο επόμενο κεφάλαιο.

Κεφάλαιο 3

Αρχιτεκτονική

Στο κεφάλαιο αυτό παρουσιάζονται οι σχετικές αρχιτεκτονικές και αναλύεται η εφαρμοζόμενη αρχιτεκτονική του μοντέλου.

3.1 Σχετικές αρχιτεκτονικές

3.1.1 R-CNN

Το R-CNN [62] αποτελεί ένα σημαντικό δίκτυο για την κατηγοριοποίηση αντικειμένων σε εικόνες. Στην αρχή συλλέγει ένα σύνολο από περιοχές (2000 στην δημοσίευση) στις οποίες θα γίνουν οι μετέπειτα προβλέψεις (region proposals) και στην συνέχεια υπολογίζει τα χαρακτηριστικά για την κάθε περιοχή χρησιμοποιώντας βαθιά συνελικτικά δίκτυα (CNN).Στο τέλος κατηγοριοποιεί κάθε περιοχή που έχει συλλέξει χρησιμοποιώντας support vector machines [63] κατάλληλες για κάθε κλάση.

3.1.2 Fast R-CNN

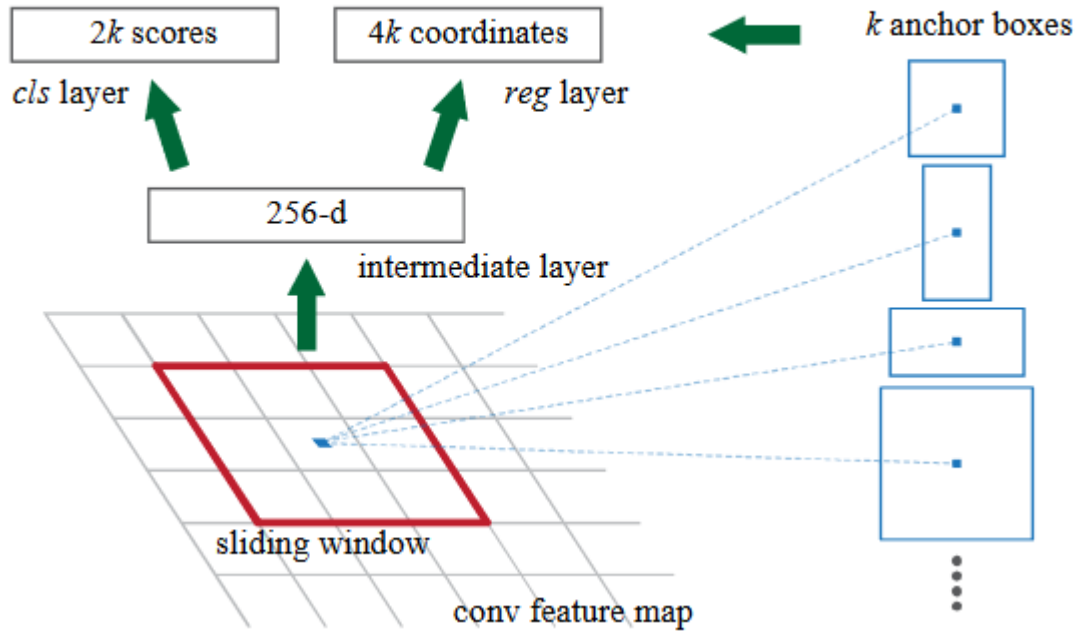
Το Fast-RCNN [61] αποτελεί βελτίωση του R-CNN [62] ενώ διαφοροποιείται ως προς τον τρόπο χρήσης των συνελικτικών δικτύων , τον υπολογισμό των προβλέψεων και την συνάρτηση απώλειας. Πλέον αντί να εφαρμόζονται τα συνελικτικά δίκτυα ξεχωριστά σε κάθε περιοχή ,πρώτα ανακαλύπτονται τα χαρακτηριστικά με την χρήση βαθιών συνελικτικών δικτύων και στην συνέχεια στον προκύπτον χάρτη χαρακτηριστικών γίνονται οι προτάσεις περιοχών , με βάση την αρχική εικόνα.Οι προβλέψεις γίνονται πλέον με επίπεδο softmax και όχι με svm επίπεδα.Τέλος η συνάρτηση απωλειών αποτελεί άθροισμα 2 παραγόντων, απώλεια κατηγοριοποίησης (classification loss) και απώλεια διόρθωσης (regression loss).

3.1.3 Region Proposal Network (RPN)

Οι τεχνικές προτάσεων περιοχής(region proposal methods) βασίζονται τυπικά σε ανέξοδα χαρακτηριστικά και ανέξοδα σχήματα εξαγωγής συμπερασμάτων. Η εκλεκτική αναζήτηση (Selective Search) [64] χρησιμοποιεί άπληστη συγχώνευση των ύπερ-εικονοστοιχείων (superpixels) με βάση χαρακτηριστικά χαμηλού επιπέδου. Παρ' όλα αυτά είναι αργή σε σχέση με αποδοτικά δίκτυα αναγνώρισης [61] .

Το δίκτυο προτάσεων περιοχής (Region Proposal Network ή RPN) από την δημοσίευση Faster-RCNN [65], προτείνει τον υπολογισμό προτάσεων περιοχής(region proposals) με βαθιά συνελικτικά νευρωνικά δίκτυα με στόχο έναν αποδοτικό υπολογισμό προτάσεων. Η αρχιτεκτονική του δικτύου αυτού μοιράζεται συνελικτικά επίπεδα με τα δίκτυα αναγνώρισης αντικειμένων. Συγκεκριμένα οι συνελικτικοί χάρτες χαρακτηριστικών (convolutional feature maps) που χρησιμοποιούνται από τα δίκτυα αναγνώρισης όπως το Fast R-CNN μπορούν επίσης να χρησιμοποιηθούν για την παραγωγή προτάσεων περιοχών. Το RPN κατασκευάζεται πάνω στους χάρτες αυτούς με την πρόσθεση κάποιων επιπλέον συνελικτικών επιπέδων, τα οποία ταυτοχρόνως υπολογίζουν και διορθώνουν τα όρια της προτεινόμενης περιοχής και την βαθμολογία ύπαρξης αντικειμένων σε κάθε περιοχή . Το RPN είναι συνεπώς ένα είδος πλήρως συνελικτικού δικτύου και μπορεί να εκπαιδευτεί από άκρη σε άκρη για τον σκοπό της παραγωγής πιθανών προτάσεων αντικειμένων. Τα RPN είναι σχεδιασμένα να μαντεύουν αποδοτικά μια πληθώρα από προτάσεις διαφορετικών μεγεθών και αναλογίας διαστάσεων. Για τον σκοπό αυτό χρησιμοποιούν τα λεγόμενα κυτία προκαθορισμένων διαστάσεων (anchor boxes), που λειτουργούν ως κυτία αναφοράς σε διαφορετικά μεγέθη και κλίμακες. Τα RPN λαμβάνουν ως είσοδο μία εικόνα οποιουδήποτε μεγέθους και παράγουν σαν έξοδο ένα σύνολο προτάσεων με την μορφή ορθογωνίων. Η παραγωγή προτάσεων περιοχής γίνεται ώς εξής: Ένα μικρό δίκτυο ολισθαίνει πάνω στον συνελικτικό χάρτη χαρακτηριστικών, τον οποίον παράγει το τελευταίο συνελικτικό επίπεδο που διαμοιράζεται μεταξύ του δικτύου αναγνώρισης και του δικτύου προτάσεων περιοχής. Το δίκτυο αυτό λαμβάνει σαν είσοδο ένα $n \; x \; n$ χωρικό παράθυρο - τμήμα του συνελικτικού χάρτη χαρακτηριστικών .Κάθε ολισθαίνον παράθυρο αντιστοιχεί σε ένα χαρακτηριστικό χαμηλών διαστάσεων. Το χαρακτηριστικό αυτό δίνεται ως είσοδο σε 2 πλήρως συνδεδεμένα επίπεδα, ένα επίπεδο οπισθοδρόμησης του κυτίου (box regression layer) και ένα επίπεδο κατηγοριοποίησης του κυτίου (box classification layer). Σημαντικό είναι επίσης, ότι τα 2 πλήρως συνδεδεμένα δίκτυα διαμοιράζονται μεταξύ όλων των τοποθεσιών στον χώρο. Η παραπάνω αρχιτεκτονική εφαρμόζεται με ένα $n \; x \; n$ συνελικτικό επίπεδο ακολουθούμενο από τα 2 1 x 1 προαναφερθέντα συνελικτικά επίπεδα.Σε κάθε τοποθεσία του ολισθαίνοντος παραθύρου γίνεται ταυτόχρονα η πρόβλεψη πολλαπλών προτεινόμενων κυτίων, όπου ο μέγιστος αριθμός προτάσεων δηλώνεται με κ . Έτσι το επίπεδο οπισθοδρόμησης των κυτίων (box regression layer) έχει 4κ εξόδους που κωδικοποιούν τις διαστάσεις των κ προτεινόμενων κυτίων και το επίπεδο κατηγοριοποίησης του κυτίου έχει σαν έξοδο 2κ βαθμολογίες που υπολογίζουν την πιθανότητα το αντικείμενο να υπάρχει ή να μην υπάρχει αντικείμενο σε κάθε πρόταση. Ο όρος «ύπαρξη» αντικειμένου δηλώνει την συμμετοχή του αντικειμένου σε κάποια από τις κλάσεις αντικειμένων που μας ενδιαφέρουν σε αντίθεση με την μη ύπαρξη που δηλώνει υπόβαθρο, κλάσεις που δεν μας ενδιαφέρουν (don’t care areas)/background. Τα κ προτεινόμενα κυτία ρυθμίζονται από κ προκαθορισμένα κυτία. Τα προκαθορισμένα κυτία βρίσκονται στο κέντρο του ολισθαίνοντος παραθύρου και κάθε ένα έχει συγεκριμένο μέγεθος και διαστάσεις. Σημαντική παρατήρηση είναι

ότι η μέθοδος είναι αμετάβλητη ως προς την μετατόπιση όσον αφορά τα προκαθορισμένα κυτία και τις προτάσεις περιοχών σχετικές με αυτά. Η μετατόπιση ή η περιστροφή ενός αντικειμένου δεν επηρεάζει την πρόταση περιοχής.Σχετικά με τα προκαθορισμένα κυτία μπορεί να γίνει χρήση συγκεκριμένων μεγεθών και διαστάσεων.



Σχήμα 3.1. *Αρχιτεκτονική του RPN. Σε κάθε ολίσθηση του παραθύρου παράγονται k προκαθορισμένα κυτία αναγνώρισης. Στο τέλος υπάρχει το επίπεδο κατηγοριοποίησης και το επίπεδο διόρθωσης των κυτίων Εικόνα από το [65].*

Η συνάρτηση απωλειών (loss function) που χρησιμοποιείται για την εκπαίδευση των RPN ορίζεται ως εξής: Σε κάθε προκαθορισμένο κυτίο γίνεται ανάθεση μιας θετικής είτε αρνητικής ταμπέλας. Ένα προκαθορισμένο κυτίο χαρακτηρίζεται θετικό εάν έχει είτε την μέγιστη τομή προς ένωση (IoU) με ένα κυτίο γνωστής κλάσης (ground truth box) είτε τομή προς ένωση μεγαλύτερη ενός συγκεκριμένου ποσοστού με ένα κυτίο γνωστής κλάσης. Αντίστοιχα ,ένα προκαθορισμένο κυτίο χαρακτηρίζεται αρνητικό εάν έχει τομή προς ένωση μικρότερη ενός

συγκεκριμένου ποσοστού με όλα τα κυτία γνωστών κλάσεων. Έτσι ορίζεται η συνάρτηση απωλειών [66] που αποτελεί μια τροποποίηση της συνάρτησης απωλειών του Fast R-CNN [67].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \qquad (3.1)$$

Όπου i είναι ο δείκτης κάθε προκαθορισμένου κυτίου σε μία μίνι-παρτίδα (mini batch) και p_i είναι η προβλεπόμενη πιθανότητα του κυτίου να είναι ένα αντικείμενο. Η μεταβλητή p_i^* που αντιστοιχεί σε κυτίο γνωστής κλάσης είναι 1 εάν το προκαθορισμένο κυτίο είναι θετικό και 0 εάν είναι αρνητικό. Το t_i είναι ένα διάνυσμα που περιγράφει τις 4 παραμετροποιημένες συντεταγμένες του προτεινόμενου κυτίου οριοθέτησης (bounding box) και το t_i^*είναι το διάνυσμα που αντιστοιχεί σε κυτίο γνωστής κλάσης που έχει συσχετιστεί με ένα θετικό προκαθορισμένο κυτίο. Η απώλεια κατηγοριοποίησης L_{cls} είναι λογάριθμος των 2 πιθανών αποτελεσμάτων, να είναι ή να μην είναι αντικείμενο. Η απώλεια διόρθωσης $L_{reg}(ti, ti*) = R(ti - ti*)$ όπου R είναι η συνάρτηση απωλειών smoothL1 [67]. Ο όρος $p_i^* L_{reg}$ φροντίζει η απώλεια διόρθωσης των κυτίων να ενεργοποιείται μόνο για κυτία που πράγματι ήταν επιτυχημένα και αναγνώρισαν κάποιο στιγμιότυπο της κλάσης και να παραμένει απενεργοποιημένη σε όλες τις άλλες περιπτώσεις. Οι έξοδοι των 2 επιπέδων των cls και reg είναι $\{pi\}$ και ti αντίστοιχα. Οι 2 όροι κανονικοποιούνται με το N_{cls}και το N_{reg} αντίστοιχα και πολλαπλασιάζονται με τον συντελεστή βαρύτητας λ. Στην σχετική δημοσίευση ο όρος L_{cls} κανονικοποιείται με το μέγεθος της παρτίδας και ο όρος L_{reg} με τον αριθμό των προκαθορισμένων κυτίων. Το λ λαμβάνει την τιμή 10 και για τους 2 όρους της εξίσωσης. Οι παραμετροποιήσεις για την διόρθωση των προτεινόμενων κυτίων ορίζονται ως εξής :

$$t_x = (x - x_a)/w_a, \qquad t_y = (y - y_a)/h_a,$$

$$t_w = log(w/w_a), \qquad t_h = log(h/h_a),$$

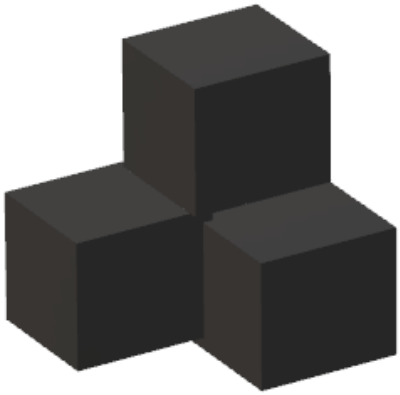
$$t_x^* = (x^* - x_a)/w_a, \qquad t_y^* = (y^* - y_a)/h_a$$

$$t_w^* = log(w^*/w_a), \qquad t_h^* = log(h^*/h_a) \qquad (3.2)$$

Όπου x, y, w, h δηλώνουν τις συντεταγμένες του κέντρου του κυτίου καθώς και το πλάτος και το ύψος του. Στα παραπάνω οι μεταβλητές χωρίς δείκτη αντιστοιχούν στο προβλεπόμενο κυτίο αναγνώρισης ενώ αυτές με δείκτη α δηλώνουν το προκαθορισμένο κυτίο αναγνώρισης. Οι μεταβλητές με δείκτη * ορίζουν κυτίο γνωστής κλάσης (ground truth). Τα χαρακτηριστικά που χρησιμοποιούνται για την διόρθωση είναι ίδιου μεγέθους στους χάρτες χαρακτηριστικών. Για την εκμάθηση διαφορετικών μεγεθών χρησιμοποιούνται κ διορθωτές κυτίων. Κάθε ένας αναλαμβάνει συγκεκριμένο μέγεθος και κλίμακα και δεν μοιράζονται βάρη μεταξύ τους έτσι ώστε να καθίσταται δυνατό να προβλέπονται κυτία διαφορετικών μεγεθών ακόμη και αν τα χαρακτηριστικά είναι συγκεκριμένου μεγέθους και κλίμακας. Για την εκπαίδευση των RPN μπορεί να γίνει χρήση του αλγορίθμου backpropagation σε συνδυασμό με την στοχαστική κατάβαση κλίσης για την ανανέωση των βαρών.

3.1.4 Voxel-Grid Representation of 3d Point Cloud

Πολλές αρχιτεκτονικές εντοπισμού τρισδιάστατων αντικειμένων που κάνουν χρήση συσκευών lidar χρησιμοποιούν συγκεκριμένες αναπαραστάσεις των Point Clouds με την χρήση πλεγμάτων voxel. Στον τομέα των γραφικών υπολογιστών το voxel εκφράζει μία τιμή στον τρισδιάστατο χώρο [68]. Τα voxels ουσιαστικά κωδικοποιούν τις τρισδιάστατες συντεταγμένες στον χώρο με έναν κύβο κατ' αντιστοιχία με την κωδικοποίηση των δισδιάστατων συντεταγμένων με ένα εικονοστοιχείο pixel.Τα voxels συχνά δεν έχουν τις συντεταγμένες τους σαν τιμές αλλά αντί αυτού τα συστήματα παραγωγής εικόνων ή τρισδιάστατων γραφικών συμπεραίνουν την τοποθεσία κάθε voxel με βάση την θέση του με τα υπόλοιπα voxels. Ένα voxel εκφράζει ένα μοναδικό στιγμιότυπο ή σημείο (point) στον χώρο σε ένα τρισδιάστατο πλέγμα. Η τιμή του κάθε voxel μπορεί να εκφράζει πολλές διαφορετικές ιδιότητες και μπορεί να περικλείουν μοναδικούς αριθμούς (scalars) ή διανύσματα ή τανυστές . Τα voxel συστήνονται για την αναπαράσταση κανονικών χώρων στους οποίους τα σημεία είναι ανομοιόμορφα κατανεμημένα.



Εικόνα 3.1. *Τα voxels αναπαρίστανται σαν παραλληλεπίπεδα στον τρισδιάστατο χώρο.*

Το μοντέλο Vote3deep [69] αντιπροσωπεύει κάθε voxel με την χρήση 6 στατιστικών ποσοτήτων που ανασύρονται από τα σημεία του point cloud που περιέχει κάθε voxel. Επίσης το μοντέλο Voting for Voting in online point cloud object detection [70] προτείνει την αναπαράσταση του Point Cloud με ένα τρισδιάστατο πλέγμα το οποίο χωρίζεται σε κελιά. Κάθε κελί περιλαμβάνει έναν αριθμό από σημεία του cloud με μαζί με τις τιμές των reflectance τους και αντιστοιχίζονται σε ένα διάνυσμα χαρακτηριστικών. Τα κελιά που δεν περιλαμβάνουν σημεία αντιστοιχίζονται σε μηδενικά διανύσματα.

3.1.5 Image Based Detection

Πολλές τεχνικές αξιοποιούν την προβολή των 3d clouds σε 2 διαστάσεις και εν συνεχεία χρησιμοποιούν μεθόδους κωδικοποίησης χαρακτηριστικών σε εικόνες [71].

3.1.6 Multi-Modal Fusion Methods

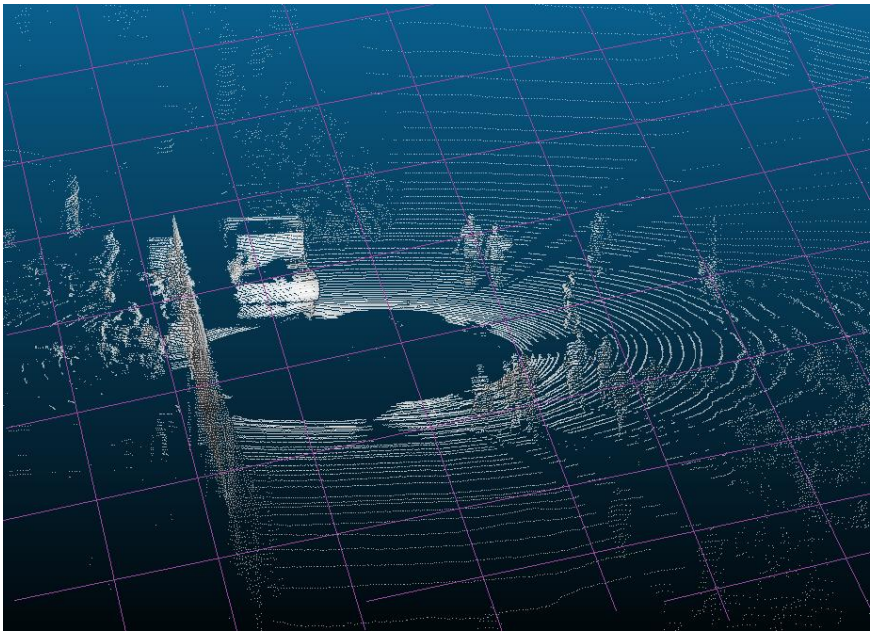
Πολλές μέθοδοι αναγνώρισης συνδυάζουν εικόνες και Lidar δεδομένα για να βελτιώσουν την ακρίβεια των αναγνωρίσεων [72]. Οι μέθοδοι αυτοί παρουσιάζουν βελτιωμένη ακρίβεια σε σχέση με lidar only μεθόδους κυρίως για μικρά αντικείμενα ή για μακρινά αντικείμενα αλλά απαιτούν επιπλέον κάμερα η οποία να είναι σε συγχρονισμό με την συσκευή lidar, πράγμα που τις κάνει ευάλωτες σε σφάλματα των συσκευών.

3.2 Χρησιμοποιούμενη αρχιτεκτονική

Η εφαρμοζόμενη αρχιτεκτονική βασίζεται στην δημοσίευση «Voxelnet: End-to-End Learning for Point Cloud Based 3D Object Detection» [73]. Για την υλοποίηση της αναγνώρισης τρισδιάστατων αντικειμένων γίνεται χρήση ενός δικτύου εκμάθησης χαρακτηριστικών (Feature Learning Network), Συνελικτικών επιπέδων για την βελτίωση των χαρακτηριστικών αυτών και ενός RPN για την παραγωγή anchors και bounding boxes που χρησιμοποιούνται για την αναγνώριση του αντικειμένου .

3.2.1 Feature Learning Network

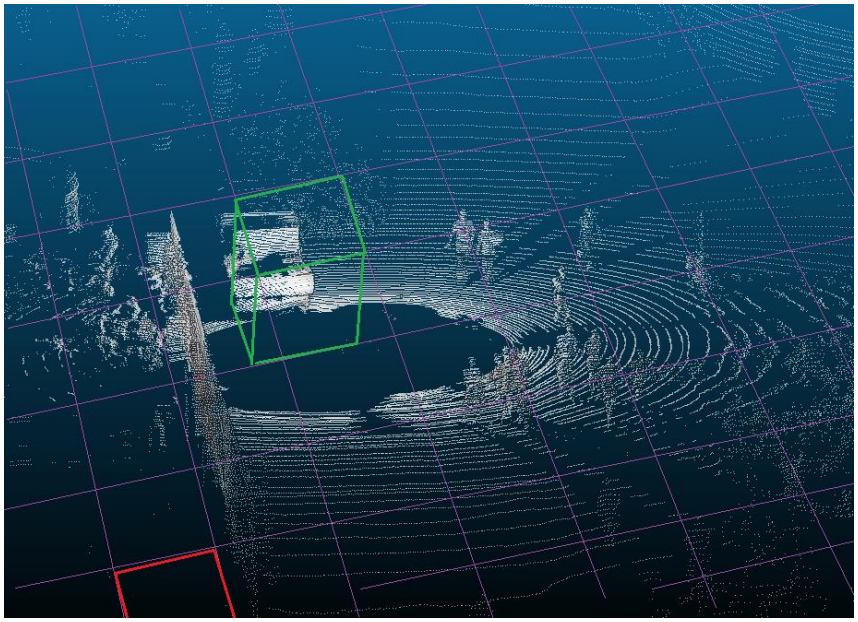
Τα Feature learning Networks χρησιμοποιούνται για την συλλογή των σημαντικών χαρακτηριστικών των δεδομένων τα οποία είναι απαραίτητα για την παραγωγή anchors και bounding boxes σε περιοχές των δεδομένων σε μετέπειτα στάδιο. Δεδομένου ενός 3d Point Cloud γίνεται στην αρχή διαίρεση του Point Cloud σε voxels ίδιου μεγέθους. Εάν χρησιμοποιηθεί voxel διαστάσεων v_D, v_H, v_W τότε ένα Point Cloud με διαστάσεις $\mathrm{D}, \mathrm{H}, \mathrm{W}$ στους άξονες $\mathrm{Z}, \mathrm{Y}, \mathrm{X}$ μετατρέπεται σε ένα voxel grid μεγέθους $\mathrm{D}' = D/v_D, H' = H/v_H, W' = W/v_W$, όπου θεωρείται ότι τα $\mathrm{D}, \mathrm{H}, \mathrm{W}$ είναι πολλαπλάσια των v_D, v_H, v_W .



Εικόνα 3.2. *Το τρισδιάστατο Point Cloud χωρίζεται σε grid μεγέθους D' x H' x W'.Point Cloud από [75].*

Grouping

Στην συνέχεια γίνεται η ομαδοποίηση των points με βάση το voxel στο οποίο ανήκουν. Σε μία 3D αναπαράσταση πολλών αντικειμένων υπάρχουν αντικείμενα περιστραμμένα είτε αντικείμενα με ένα μέρος τους εκτός του 3D cloud (truncated) είτε αντικείμενα με ένα μέρος τους να κρύβεται πίσω από κάποιο άλλο αντικείμενο (occluded). Για τους παραπάνω λόγους ένα point cloud είναι μια αραιά αναπαράσταση αλλά πολυποίκιλη ως προς την πυκνότητα σημείων στις διάφορες περιοχές του cloud. Συνεπώς κατά την ομαδοποίηση των σημείων κάθε voxel θα περιλαμβάνει διαφορετικό αριθμό σημείων.



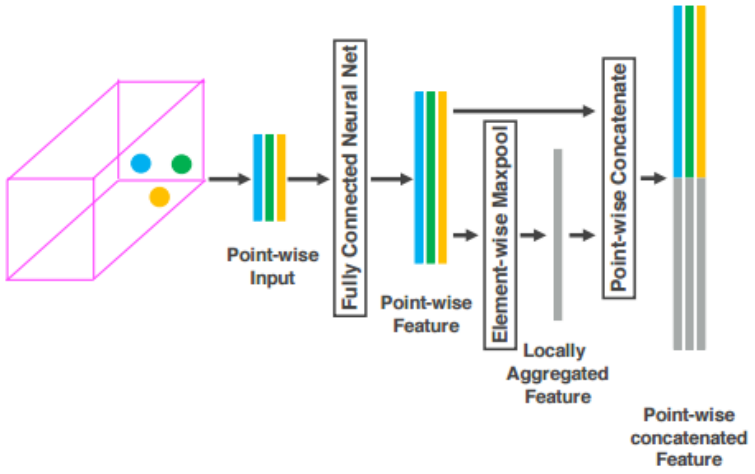
Εικόνα 3.3. *Κάθε voxel θα περιλαμβάνει κάποιον αριθμό ψηφίων.Το σημειωμένο με πράσινο για παράδειγμα περιλαμβάνει τα σημεία που αποτελούν ένα μέρος του φορτηγού στο cloud.Το σημειωμένο με κόκκινο από την άλλη δεν περιλαμβάνει κανένα σημείο .Point Cloud από [75].*

Random Sampling

Τα point clouds συνήθως αποτελούνται από ~100k σημεία .Η άμεση επεξεργασία τόσον πολλών σημείων είναι πολύ ακριβή σε υπολογιστική ισχύ και μνήμη υπολογιστή ενώ ενέχει τον κίνδυνο η διαφορετική πυκνότητα των σημείων στον χώρο να στρέψει την αναγνώριση προς τις πιο πυκνές περιοχές επηρεάζοντας την .Για αυτόν τον λόγο γίνεται δειγματοληψία ενός προκαθορισμένου αριθμού Τ σημείων ,από κάθε voxel που περιλαμβάνει παραπάνω από Τ σημεία.Η δειγματοληψία με αυτόν τον τρόπο οδηγεί σε αποδοτικό υπολογισμό και εξοικονόμηση υπολογιστικής ισχύος ενώ ταυτόχρονα μειώνει την ανισορροπία του αριθμού των σημείων μεταξύ των voxels ,εφόσον δειγματοληπτούνται και πιο αραιές περιοχές .Προστίθεται έτσι ποικιλία στην εκπαίδευση.

Stacked Voxel Feature Encoding

Για την συλλογή των χαρακτηριστικών όπως σχήμα, μορφή, χρώμα, γωνίες που περικλείουν τα voxel γίνεται χρήση μιας ιεραρχικής διαδικασίας κωδικοποίησης για κάθε voxel με την μορφή αλυσίδας. Η διαδικασία αυτή αποτελείται από n επαναλήψεις ενός τμήματος της αλυσίδας το οποίο και ονομάζεται VFE layer (voxel feature encoding layer).

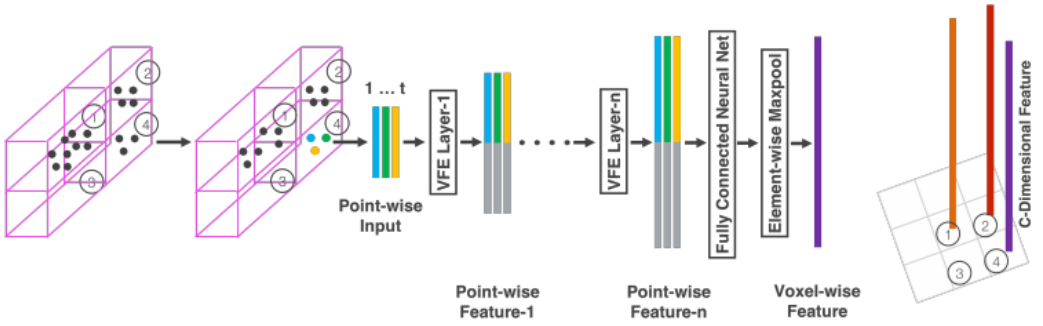


Σχήμα 3.2. *Voxel Feature Encoding Layer. Για κάθε voxel συλλέγονται τα pointwise χαρακτηριστικά και στην συνέχεια τα συνολικά χαρακτηριστικά για κάθε σημείο του voxel ώστε κάθε σημείο να περιλαμβάνει τα σημειακά και τα συνολικά χαρακτηριστικά. Εικόνα από [73].*

Κάθε μη άδειο voxel που περιέχει $t \leq T$ σημεία ορίζεται ως $V = \{pi = [x\_i, y\_i, z\_i, r\_i]^T \in R^4\}_{i=1...t}$ όπου το p_i περιέχει X, Y, Z συντεταγμένες στο point cloud και r_i είναι η λαμβανόμενη reflectance του σημείου. Αρχικά υπολογίζεται ο τοπικός μέσος (centroid) των σημείων που ανήκουν στο voxel και περιγράφεται ως (u_x, u_y, u_z). Στην συνέχεια γίνεται τροποποίηση κάθε

σημείου pi με την σχετική απόσταση από το centroid και λαμβάνεται το input feature set $V_{\text{in}} = \{\hat{p}_i = [x_i, y_i, z_i, r_i, x_i - \upsilon_x, y_i - \upsilon_y, z_i - \upsilon_z]^T \in R^7\}_{i=1\ldots t}$. Στην συνέχεια κάθε σημείο $\hat{p}_i$εισάγεται σε ενα fully connected network (FCN) και τα χαρακτηριστικά του $f_i \in R^m$εισάγονται στο feature space. Έτσι επαυξάνεται και βελτιώνεται η περιγραφή της επιφάνειας που περικλείει το voxel με την εισαγωγή πληροφορίας από κάθε σημείο του. Το FCN αποτελείται από ένα linear layer, ένα batch normalization layer και ένα ReLU layer. Αφού ληφθούν τα χαρακτηριστικά των σημείων και δημιουργηθεί η αναπαράσταση τους γίνεται MaxPooling από όλα τα σημεία $f_i \in R^m$που ανήκουν στο voxel V και δημιουργείται έτσι το συγκεντρωτικό feature $\tilde{f} \in R^m$, η αναπαράσταση για το voxel V που αποτελεί μια σύνθεση χαρακτηρηστικών απ'όλα τα σημεία του. Ακολούθως το feature f_i κάθε σημείου του voxel V συνενώνεται με το συγκεντρωτικό $\tilde{f}$ και λαμβάνεται η τελική αναπαράσταση κάθε σημείου $f_i^{\text{out}} = \left[f_i^T, \tilde{f}^T\right]^T \in R^{2m}$ όπου ο εκθέτης T δηλώνει αναστροφή στο διάνυσμα. Το τελικό σετ χαρακτηριστικών (feature set) κάθε μη άδειου voxel V κωδικοποιείται ως $V_{out} = \{f_i^{out}\}_{i\ldots t}$. Τα κωδικοποιημένα voxels μοιράζονται τις ίδιες παραμέτρους κατά την διέλευση μέσα από το FCN.

Όπως προαναφέρθηκε χρησιμοποιείται μια αλυσίδα από VFE layers στην οποία ορίζεται ως $VFE - i(Cin, cout)$ κάθε VFE layer το οποίο λαμβάνει τα χαρακτηριστικά features διαστάσεων Cin και παράγει σαν έξοδο features διαστάσεων $cout$. Το linear layer συντελεί στην εκμάθηση ενός πίνακα μεγέθους $c_{\text{in}}(\times c_{out}/2)$ και η συνένωση των features των υπόλοιπων σημείων με το feature κάθε σημείου ,όπως γίνεται παραπάνω, οδηγεί σε τελική διάσταση c_{out}. Η τελική αναπαράσταση – έξοδος ενός VFE layer είναι ένας συνδυασμός χαρακτηριστικών από κάθε σημείο και συγκεντρωτικών χαρακτηριστικών από όλα τα σημεία. Συνεπώς η αλυσίδα των VFE layers κωδικοποιεί τις αλληλεπιδράσεις-σχέσεις των σημείων ενός voxel και επιτρέπει στην τελική αναπαράσταση να μαθαίνει το σχήμα και την μορφή της επιφάνειας που περικλείεται από το voxel. Το συγκεντρωτικό feature ενός Voxel λαμβάνεται από την διέλευση της εξόδου του τελικού $VFE - n$ layer μέσω ενός FCN και στην συνέχεια μέσω element-wise MaxPooling, όπου C είναι η διάσταση του voxelwise feature.



Σχήμα 3.3. *Η αλυσίδα VFE.Κάθε voxel διέρχεται από μια αλυσίδα από VFE layers για να κατασκευαστεί το Voxel-wise feature.Εικόνα από [73].*

Sparse Tensor Representation

Λόγω της διαφορετική πυκνότητας σημείων ανά voxel, η επεξεργασία των voxels περιορίζεται μόνο στα μη άδεια voxels για τα οποία λαμβάνεται μία λίστα από χαρακτηριστικά, η οποία έχει ένα προς ένα αντιστοίχιση με τις συντεταγμένες του μοναδικού voxel με το οποίο σχετίζεται. Η λίστα αυτή κάθε voxel μπορεί να εκφραστεί σαν ένας αραιός τανυστής 4 διαστάσεων (Tensor) και πιο συγκεκριμένα $C \times D' \times H' \times W'$. Παρά το μεγάλο μέγεθος των point clouds περισσότερο από το 90% των voxels τους δεν περιέχουν σημεία. Έτσι η κωδικοποίηση των voxel features ως έναν αραιό τανυστή είναι κρίσιμη για την εξοικονόμηση μνήμης και υπολογιστικής ισχύος κατά το backpropagation.

3.2.2 Convolutional Middle Layer

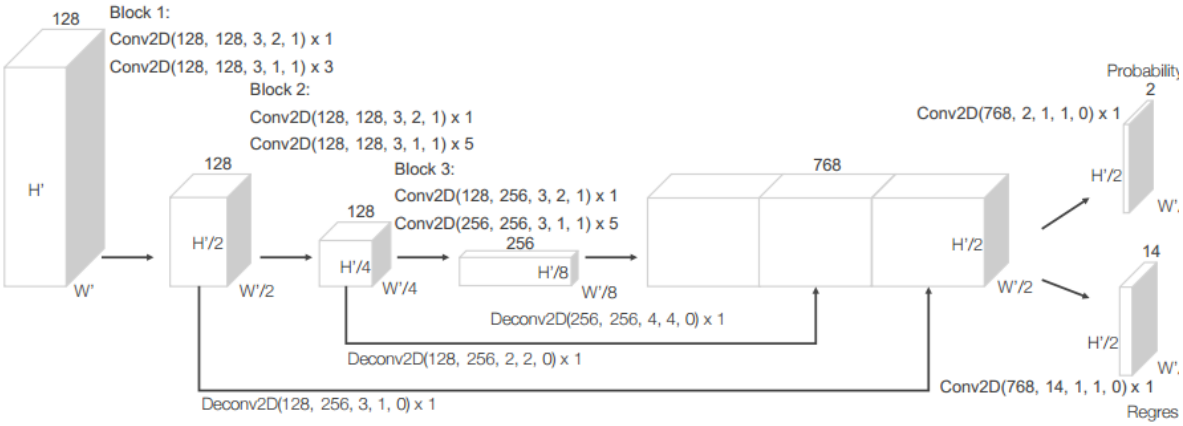
Για την περιγραφή ενός M διαστάσεων συνελικτικού επιπέδου(convolutional layer) χρησιμοποιείται ο παρακάτω όρος: $ConvMD(c_{\text{in}}, c_{\text{out}}, k, s, p)$ όπου το c_{in} είναι ο αριθμός των input channels και c_{out} είναι ο αριθμός των output channels, k είναι διάνυσμα διαστάσεων M που δηλώνει το μέγεθος του kernel (kernel size), s είναι είναι διάνυσμα διαστάσεων M που δηλώνει το μέγεθος του stride (stride size) και p είναι διάνυσμα διαστάσεων M που ορίζει το μέγεθος του padding.

Για παράδειγμα αν το stride είναι διάνυσμα 3 διαστάσεων και έχει μεγέθη 3, 1 και 1 στις 3 διαστάσεις τότε θα εκφραστεί ως $s = (3,1,3)$.Εαν και στις 3 διαστάσεις έχει το ίδιο μέγεθος g τότε μπορεί να περιγραφεί από έναν μόνο αριθμό g .

Κάθε ενδιάμεσο συνελικτικό επίπεδο (convolutional middle layer) αναλύεται σε τρισδιάστατη συνέλιξη, batch normalization επίπεδο και ReLu επίπεδο. Τα convolutional middle layers συσσωρεύουν voxel-wise features ολοένα και διευρυνόμενων πεδίων υποδοχής προσθέτοντας πληροφορίες από γειτνίαζουσες περιοχές στην περιγραφή και την μορφή κάθε σχήματος.

3.2.3 Region Proposal Network

Το Region Proposal Network που χρησιμοποείται, αποτελεί μια τροποποίηση της αρχιτεκτονικής Region Proposal Network του Faster-RCNN [74]. Λαμβάνει ώς είσοδο το feature map που παρήγαγαν τα convolutional middle layers και έχει 3 μπλοκ από πλήρως συνελικτικά επίπεδα. Το 1ο επίπεδο από κάθε μπλοκ εκτελεί downsampling (υποδειγματοληψία) του feature map στο μισό μέγεθος με την χρήση συνέλιξης με βήμα stride 2 ενώ στην συνέχεια ακολουθεί μια αλυσίδα από συνελίξεις με βήμα stride 1. Μετά από κάθε επίπεδο συνέλιξης εφαρμόζεται Batch Normalization και εφαρμόζεται συνάρτηση ενεργοποίησης ReLu. Οι έξοδοι από κάθε μπλοκ στην συνέχεια υπόκεινται σε upsampling (αύξηση των διαστάσεων) σε ένα προκαθορισμένο μέγεθος και συνενώνονται ώστε να κατασκευαστεί το feature map υψηλών διαστάσεων. Αυτό το feature map χρησιμοποιείται για την παραγωγή των πιθανοτήτων των σκορ κάθε κλάσης (εδώ χρησιμοποιούνται 2 κλάσεις, αντικείμενο ή όχι αντικείμενο) αλλά και για τον χάρτη της διόρθωσης (regression map) ώστε να γίνει η διόρθωση των προβλεπόμενων bounding boxes.



Σχήμα 3.5. *Το τελικό Region Proposal Network αποτελείται από 3 μπλοκ από συνελικτικά επίπεδα όπου ύστερα από κάθε επίπεδο εφαρμόζεται batch normalization και ReLU.Στην συνέχεια γίνεται upsampling και ενώνονται οι έξοδοι από το κάθε μπλοκ για τον τελικό χάρτη χαρακτηριστικών, ο οποίος αντιστοιχίζεται σε έναν χάρτη για την κατηγοριοποίηση (υπάρχει ή όχι το αντικείμενο ?) και έναν χάρτη για διόρθωση των κυτίων στις περιπτώσεις που η απάντηση στην ύπαρξη αντικειμένου είναι θετική. Εικόνα από [73] .*

3.3 Loss Function

Η συνάρτηση των απωλειών που ζητείται να ελαχιστοποιηθεί ορίζεται με τον παρακάτω τρόπο:

Εάν $\{a_i^{pos}\}_{i=1\dots N_{pos}}$ είναι το σετ N_{pos} θετικών anchors και $\{a_i^{pos}\}_{i=1\dots N_{pos}}$ είναι το σετ N_{neg}των αρνητικών anchors ,τότε η παραμετροποίηση ενός τρισδιάστατου bounding box γίνεται με το σύνολο των μεταβλητών $\left(x_c^g, y_c^g, z_c^g, l^g, w^g, h^g, \theta^g\right)$ όπου x_c^g, y_c^g, z_c^gεκφράζουν το κέντρο του box και l^g, w^g, h^g δηλώνουν το μήκος ,το πλάτος και το ύψος του box και η θ^g είναι η yaw περιστροφή γύρω από τον Z άξονα. Κάθε anchor περιγράφεται με τις μεταβλητές $(x_c^a, y_c^a, z_c^a, l^a, w^a, h^a, \theta^a)$ που δηλώνουν κατ' αντιστοιχία τις συντεταγμένες του κέντρου, το μήκος, το πλάτος, το ύψος και την yaw περιστροφή γύρω από τον άξονα Z. Για την εύρεση του προτεινόμενου bounding box που «ταιριάζει» με την εκάστοτε anchor ορίζεται το υπολειπόμενο διάνυσμα $u *\in R^7$που περιέχει τους 7 στόχους-διαφορές $\Delta x, \Delta y, \Delta z, \Delta l, \Delta w, \Delta h, \Delta\theta$ ως προς τον άξονα $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ δηλαδή το μήκος, το πλάτος, το ύψος και

την γωνία αντίστοιχα, οι οποίοι πρέπει να υπολογιστούν προκειμένου να εκτιμηθεί η τοποθεσία και οι διαστάσεις του τρισδιάστατου box:

$$\Delta x = \frac{x_c^g - x_c^a}{d^a}, \Delta y = \frac{y_c^g - y_c^a}{d^a}, \Delta z = \frac{z_c^g - z_c^a}{h^a}$$
$$\Delta l = \log\frac{l^g}{l^a}, \Delta w = \log\frac{w^g}{w^a}, \Delta h = \log\frac{h^g}{h^a}$$
$$\Delta\theta = \theta^g - \theta^a \quad (3.3)$$

Οι $\Delta x, \Delta y$ κανονικοποιούνται ομοιόμορφα με την $d^a = \sqrt{(l^a)^2 + (w^a)^2}$, η οποία είναι η διαγώνιος της βάσης του anchor box . Έτσι προκύπτει ο τύπος της συνάρτησης απωλειών:

$$L = a\frac{1}{N_{\text{pos}}}\sum_i L_{\text{cls}}\left(p_i^{\text{pos}}, 1\right) + \beta\frac{1}{N_{\text{neg}}}\sum_j L_{cls}\left(p_j^{\text{neg}}, 0\right) + \frac{1}{N_{\text{pos}}}\sum_i L_{\text{reg}}(u_i, u_i^*) \quad (3.4)$$

όπου το p_i^{pos}, p_j^{neg} εκφράζουν την softmax ενεργοποιημένη έξοδο για την θετική anchor a_i^{pos}και την αρνητική anchor p_j^{neg}αντίστοιχα. Οι όροι $u_i \in R^7$και $u_i^* \in R^7$αποτελούν την κανονικοποιημένη απώλεια για $\{a_i^{\text{pos}}\}_{i=1\ldots N_{\text{pos}}}$ $και$ $\{a_j^{\text{neg}}\}_{j=1\ldots N_{\text{neg}}}$, τις θετικές και τις αρνητικές anchors και το L_{cls} είναι απώλεια της μορφής binary cross entropy. Οι 2 συντελεστές a, β είναι θετικές σταθερές και αποτελούν τους συντελεστές βαρύτητας των 2 πρώτων όρων της συνάρτησης απωλειών. Ο τελευταίος όρος L_{reg} είναι η απώλεια οπισθοδρόμησης για την διόρθωση των bounding boxes . Η L_{reg} είναι της μορφής smoothL1.

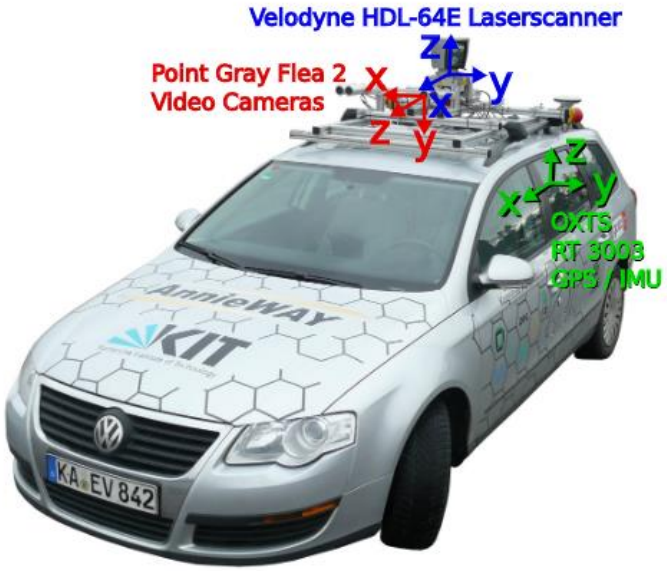
Κεφάλαιο 4

Ανάλυση Δεδομένων και Μοντέλο

Στο κεφάλαιο αυτό δίνονται πληροφορίες για την συλλογή των δεδομένων και την μορφή τους, την απαράιτητη προεπεξεργασία τους και τον τρόπο χρήσης τους και αναλύεται η τροποποιημένη αρχιτεκτονική που θα χρησιμοποιηθεί, η οποία βασίζεται στην αρχιτεκτονική των συνελικτικών δικτύων που περιγράφτηκε στο κεφάλαιο 3, στην ενότητα **3.2 .**

4.1 ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ

Για τον εντοπισμό τρισδιάστατων αντικειμένων απαιτείται η συλλογή δεδομένων, σε κατάλληλη μορφή ώστε να μπορούν να εισαχθούν στο εκάστοτε μοντέλο και να χρησιμοποιηθούν για την εκπαίδευση του. Στην παρούσα μελέτη ζητείται ο εντοπισμός αυτοκινήτων σε ποικίλα πραγματικά σενάρια κυκλοφορίας. Για να επιτευχθεί ο παραπάνω στόχος, γίνεται χρήση ενός υποσυνόλου δεδομένων του Kitti Dataset [75], http://www.cvlibs.net/datasets/kitti/. Καταγράφονται πραγματικά σενάρια κυκλοφορίας 6 ωρών οδήγησης σε 10-100kz, στους δρόμους της πόλης Karlsruhe της Γερμανίας. Το Kitti Dataset αποτελείται από μια πληθώρα μορφών δεδομένων: Πιο συγκεκριμένα, τρισδιάστατες αναπαραστάσεις σε μορφή point clouds οι οποίες συλλέγονται από ένα περιστρεφόμενο Velodyne 3d laser scanner, left color εικόνες που παράγονται από μία color camera στα αριστερά, right color εικόνες που παράγονται από μία δεύτερη δεξιά color camera, grayscale εικόνες από 2 grayscale cameras, μετρήσεις ακριβείας από σύστημα παγκόσμιου εντοπισμού θέσης (GPS) καθώς και οι επιταχύνσεις που μετρούνται από σύστημα μέτρησης αδράνειας (Inertia Measurement Unit, IMU, επιταχυνσιόμετρο) σε συνδυασμό με GPS.



Εικόνα 4.1. *Το σύστημα που χρησιμοποιήθηκε για την συλλογή των δεδομένων από το Kitti Benchmark Suite.Στην εικόνα φαίνονται οι 2 κάμερες ο laser scanner και το σύστημα gps [75], http://www.cvlibs.net/datasets/kitti/* .

Για την εκπαίδευση του αξιοποιούμενου μοντέλου (RPN) είναι απαραίτητη η εκμάθηση των χαρακτηριστικών περιοχών κάθε point cloud (Feature Encoding) ώστε να επιτευχθεί η αναγνώριση του αντικειμένου ενδιαφέροντος, το οποίο στο παρόν πρόβλημα είναι τα αυτοκίνητα. Συνεπώς τα δεδομένα που χρησιμοποιούνται για την αναζήτηση και κωδικοποίηση των features ενός point cloud και στην συνέχεια για την εκπαίδευση του RPN είναι ένα υποσύνολο των point clouds που έχουν καταγραφεί από τον Velodyne laser scanner. Πιο συγκεκριμένα στο Kitti Benchmark Suite είναι διαθέσιμα για άμεση λήψη 7481 point clouds για training και 7518 point clouds για testing, σε δυαδική μορφή .bin , όπου το καθένα περιλαμβάνει τις συντεταγμένες στους άξονες x,y,z και το reflectance value, για κάθε σημείο του. Ταυτοχρόνως δίνονται αντίστοιχες εικόνες(.png)-frames για ομαλότερη οπτικοποίηση, οι πίνακες βαθμονόμησης της κάμερας (.txt) και για τα μεν training αρχεία παρέχονται επιπλέον και τα αρχεία με τις ετικέτες που δηλώνουν την κλάση κάθε στιγμιοτύπου που μπορεί να εμφανιστεί σε ένα point cloud (.txt) για την χρήση στην επιβλεπόμενη μάθηση του μοντέλου.

4.1.1 Ανάλυση Ετικετών

Τα αρχεία των ετικετών περιλαμβάνουν σε κάθε γραμμή χωρισμένα με κενό τα εξής πεδία :

[type | truncation | occlusion | alpha | 2d bbox | dimensions | location | rotation_y | score]

Το **type** δηλώνει την κλάση του αντίστοιχου αντικειμένου που εμφανίζεται στο cloud. Οι πιθανές κλάσεις είναι :

[Car | Van | Truck | Tram | Misc | Pedestrian | Cyclist | Person (sitting) | DontCare]

Το **2d bbox** δίνει τις συντεταγμένες του bounding box που το περικλείει, στο σύστημα συντεταγμένων της εικόνας (σε pixel), $(x1, y2, x2, y1)$ = $(αριστερά, πάνω, δεξιά, κάτω)$. Τα **dimensions** είναι τα ύψος, πλάτος, μήκος του αντικειμένου στις 3 διαστάσεις, δοσμένα σε μέτρα. Το **location** δηλώνει τις συντεταγμένες του στις 3 διαστάσεις στο σύστημα συντεταγμένων της κάμερας. Το **rotation_y** δηλώνει την γωνία περιστροφής γύρω από τον άξονα των y και ανήκει στο διάστημα $[-\pi, \pi]$. Το **truncation** δείχνει πόσο το αντικείμενο ξεπερνάει τα όρια της εικόνας και είναι αριθμός στο διάστημα $[0 = βρίσκεται\ \ ολόκληρο\ εντός\ εικόνας, 1 = μέρος\ του\ είναι\ εκτός\ των\ ορίων\ της\ εικόνας]$.
Αντιστοίχως το **occlusion** δηλώνει εάν το αντικείμενο κρύβεται πίσω από κάποιο άλλο με τιμές στο διάστημα $[0 = πλήρως\ ορατό, 1 = μερικώς\ κρύβεται, 2 = κρύβεται\ μεγάλο\ μέρος\ του, 3 = άγνωστο]$.

Το **alpha** ορίζει την γωνία παρατήρησης του αντικειμένου στο διάστημα $[-\pi, \pi]$. Τέλος το **score** είναι μόνο για τα αποτελέσματα και δηλώνει την εμπιστοσύνη ότι το μοντέλο θα αναγνωρίσει το συγκεκριμένο αντικείμενο . Τα αρχεία που αντιστοιχούν στο testing δεν συνοδεύονται από τις αντίστοιχες ετικέτες εφόσον χρησιμοποιούνται μόνο για εφαρμογή του μοντέλου ύστερα από την εκπαίδευση και πιθανή αξιολόγηση των επιδόσεων.

4.1.2 Συστήματα Συντεταγμένων και Μετατροπή

A. Συστήματα Συντεταγμένων

Τα συστήματα των συντεταγμένων που χρησιμοποιούνται είναι :

Κάμερα :

$$x = \delta\epsilon\xi\iota\acute{\alpha}$$

$$y = \kappa\acute{\alpha}\tau\omega$$

$$z = \mu\pi\rho\sigma\tau\acute{\alpha}$$

Velodyne Laser Scanner :

$$x = \mu\pi\rho\sigma\tau\acute{\alpha}$$

$$y = \alpha\rho\iota\sigma\tau\epsilon\rho\acute{\alpha}$$

$$z = \pi\acute{\alpha}\nu\omega$$

GPS/IMU :

$$x = \mu\pi\rho\sigma\tau\acute{\alpha}$$

$$y = \alpha\rho\iota\sigma\tau\epsilon\rho\acute{\alpha}$$

$$z = \pi\acute{\alpha}\nu\omega$$

Σημειογραφία: Οι τρισδιάστατες rigid body μετατροπές για την μεταφορά ενός σημείου από ένα σύστημα συντεταγμένων α σε ένα σύστημα συντεταγμένων b δηλώνονται με T_{α}^{b} όπου το T δηλώνει την μετατροπή.

Για την ρύθμιση (calibration) της κάμερας χρησιμοποιείται η μέθοδος του [76]. Τα κέντρα όλων των καμερών είναι ευθυγραμμισμένα, με την έννοια ότι βρίσκονται στο ίδιο x/y επίπεδο. Η διόρθωση συνεπώς όλων των εικόνων μπορεί να γίνει ταυτόχρονα. Η μελέτη των γραμμών του *calib_cam_to_cam.txt δίνει τις calibration παραμέτρους για κάθε μέρα της καταγραφής. Πιο συγκεκριμένα παρουσιάζονται οι εξής παράμετροι :*

$s^{(i)} \in N^2$,	το αρχικό μέγεθος της εικόνας (1392 x 512)
$K^{(i)} \in R^{3x3}$,	οι πίνακες calibration (μη διορθωμένοι)
$d^{(i)} \in R^5$,	συντελεστές παραμόρφωσης (μη διορθωμένοι)
$R^{(i)} \in R^{3x3}$,	περιστροφή από την κάμερα 0 στην κάμερα i
$t^{(i)} \in R^{1x3}$,	μεταφορά από την κάμερα 0 στην κάμερα i
$s_{rect}^{(i)} \in N^2$,	μέγεθος εικόνας μετά την διόρθωση
$R_{rect}^{(i)} \in R^{3x3}$,	πίνακας περιστροφής διόρθωσης
$P_{rect}^{(i)} \in R^{3x4}$,	πίνακας προβολής μετά την διόρθωση

όπου το $i \in 0,1,2,3$ είναι ο δείκτης της κάθε κάμερας. Το 0 αντιστοιχεί στην αριστερή grayscale κάμερα, το 1 στην δεξιά grayscale, το 2 στην αριστερή color, το 3 στην δεξιά color κάμερα. Επίσης οι εικόνες έχουν υποστεί περικοπή, με αποτέλεσμα οι διορθωμένες εικόνες να είναι μικρότερες από την ανάλυση 1392 x 512 , που είναι η αρχική ανάλυση.

Β. Μετατροπες Συστηματων Συντεταγμενων και Προβολη από 3D σε 2D

Camera coordinates and calibration

Δεδομένων των παραπάνω παραμέτρων ορίζεται η προβολή ενός σημείου 3 διαστάσεων, που έχει περιστραφεί ώστε να δίνεται στο σύστημα συντεταγμένων της κάμερας με την μορφή $x = (x, y, z, 1)^T$, σε ένα σημείο $y = (u, v, 1)^T$ στην αντίστοιχη εικόνα της i-οστής κάμερας ως εξής :

$$y = P_{rect}^{(i)} x \tag{4.1}$$

όπου

$$P_{rect}^{(i)} x = \begin{pmatrix} f_u^{(i)} & 0 & c_u^{(i)} & -f_u^{(i)} b_x^{(i)} \\ 0 & f_v^{(i)} & c_v^{(i)} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.2}$$

δηλώνει τον Πίνακα Προβολής της i-οστής κάμερας και το $b_x^{(i)}$δηλώνει την κάμερα αναφοράς που αντιστοιχεί σε $i = 0$. Για την προβολή ενός σημείου x 3 διαστάσεων δοσμένου σε συντεταγμένες της κάμερας αναφοράς , σε ένα σημείο y στο επίπεδο εικόνας (2 διαστάσεων) της $i - οστής$ κάμερας, υπεισέρχεται στην εξίσωση μετατροπής και ο πίνακας περιστροφής – διόρθωσης της κάμερας αναφοράς $R_{rect}^{(0)}$για την περαιτέρω μετατροπή από το σύστημα συντεταγμένων της κάμερας αναφοράς στο σύστημα αναφοράς της $i - οστής$ κάμερας.

$$y = P_{rect}^{(i)} R_{rect}^{(0)} x \tag{4.3}$$

Ο πίνακας περιστροφής $R_{rect}^{(0)}$ επεκτείνεται σε έναν πίνακα 4x4 για το «ταίριασμα» των διαστάσεων των πινάκων που απαιτεί ο πολλαπλασιασμός, κάτι το οποίο επιτυγχάνεται με την προσθήκη μιας επιπλέον στήλης και γραμμής και θέτοντας $R_{rect}^{(0)}(4,4) = 1$

Velodyne to camera coordinates

Η μετατροπή συντεταγμένων του Velodyne Scanner στο σύστημα συντεταγμένων της κάμερας δίνεται στο αρχείο *calib_velo_to_cam.txt.*

$$R_{\text{velo}}^{cam} \in R^{3X3}, \quad \text{πινακας περιστροφής: } velodyne \rightarrow camera$$
$$t_{\text{velo}}^{cam} \in R^{1X3}, \quad \text{πινακας μεταφοράς: } velodyne \rightarrow camera$$

Επιπλέον η σχέση

$$T_{\text{velo}}^{cam} = \begin{pmatrix} R_{\text{velo}}^{\text{cam}} & t_{\text{velo}}^{\text{cam}} \\ 0 & 1 \end{pmatrix}$$

εκτελεί περιστροφή και μεταφορά ταυτόχρονα (rigid body transformation) από το ένα σύστημα στο άλλο.

Συνεπώς η προβολή ενός σημείου 3 διαστάσεων, δοσμένου στο σύστημα συντεταγμένων του velodyne scanner σε ένα σημείο 2 διαστάσεων στην εικόνα (στο 2 διαστάσεων επίπεδο) της i-οστής κάμερας γίνεται με την παρακάτω εξίσωση :

$$y = P_{rect}^{(i)} R_{\text{rect}}^{(0)} T_{\text{velo}}^{cam} x \tag{4.4}$$

GPS/IMU to camera coordinates

Ο πίνακας περιστροφής R_{imu}^{velo}και το διάνυσμα μεταφοράς T_{imu}^{velo} για την μεταφορά και την περιστροφή μεταξύ των συστημάτων συντεταγμένων των Velodyne Scanner και του GPS/IMU είναι αποθηκευμένα στο αρχείο calib_imu_to_velo.txt. Ένα σημείο x 3 διαστάσεων, δοσμένο σε συντεταγμένες του IMU/GPS προβάλλεται σε ένα σημείο y στην $i-$ οστή εικόνα με την εξίσωση:

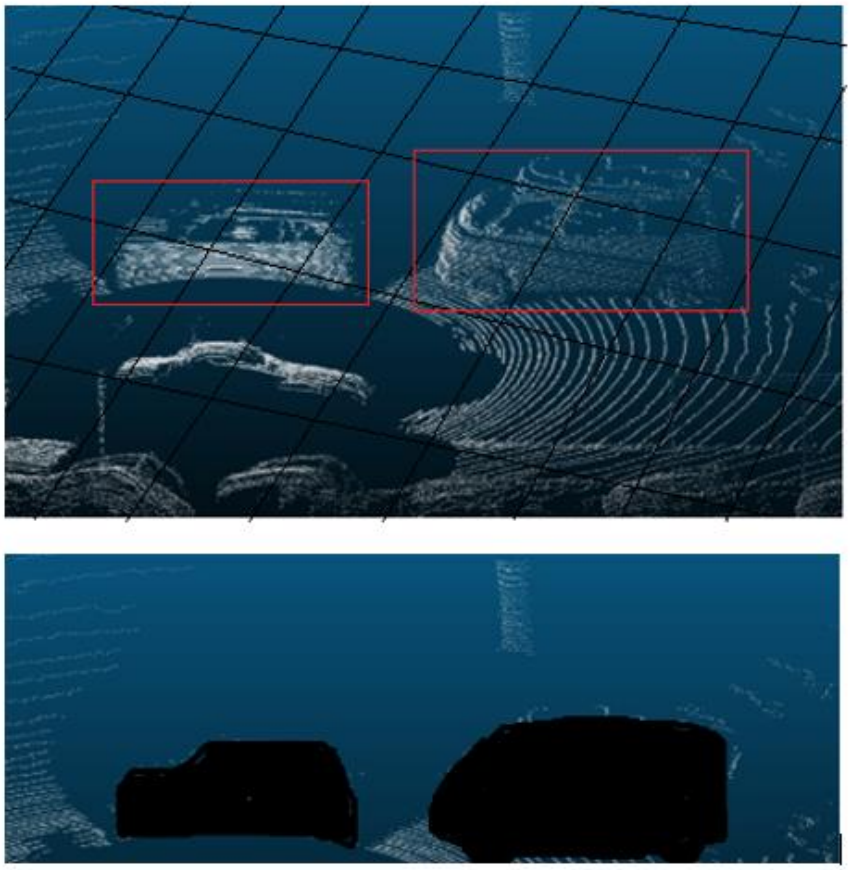
$$y = P_{rect}^{(i)} P_{rect}^{(0)} T_{\text{velo}}^{\text{cam}} T_{\text{imu}}^{\text{velo}} x$$

4.2 Προεπεξεργασία Δεδομένων

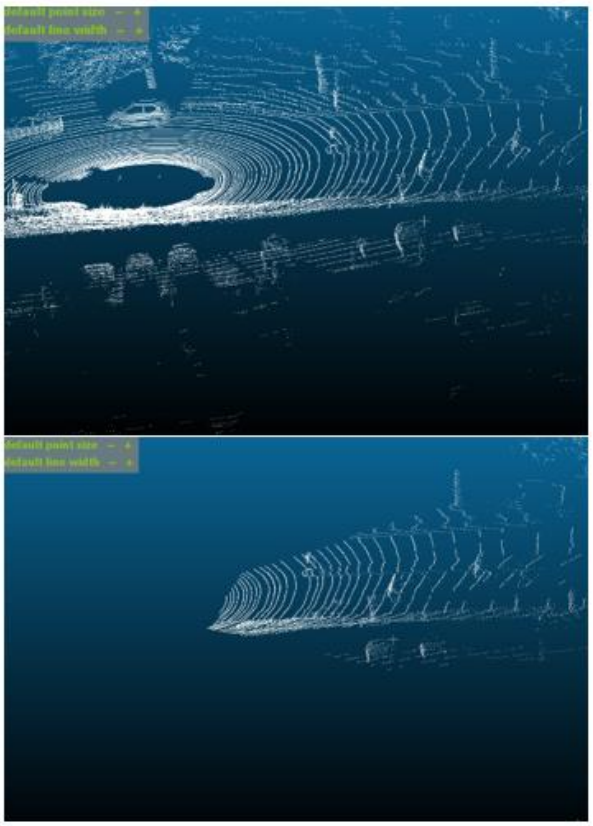
4.2.1 Περικοπή των Point Clouds

Για την κωδικοποίηση των χαρακτηριστικών κάθε point cloud (feature encoding), είναι απαραίτητη προϋπόθεση να δοθεί το cloud σε μορφή τέτοια ώστε το μοντέλο να μπορεί να συλλέξει τα σημεία του cloud (τις συντεταγμένες τους). Επιπλέον είναι κρίσιμο να γίνει χρήση για κάθε cloud μιας μειωμένης σε διαστάσεις αναπαράστασης τους ώστε να είναι πιο ελαφριά σε υπολογιστική ισχύ και μνήμη η επεξεργασία τους. Για τους παραπάνω λόγους γίνεται στην αρχή ανάγνωση κάθε point cloud σε μορφή δισδιάστατου πίνακα N γραμμών-σημείων, [N x 4] όπου οι 3 πρώτες στήλες κάθε γραμμής αντιστοιχούν στις συντεταγμένες του

εκάστοτε σημείου x, y, z και η τελευταία στήλη η παρατηρούμενη τιμή reflectance (ανακλαστικότητα). Επιπλέον διαβάζονται οι πίνακες Μεταφοράς, Περιστροφής και Προβολής (Tr, R, P), που χρησιμοποιούνται για την μεταφορά από το ένα σύστημα συντεταγμένων στο άλλο την περιστροφή συστήματος και την προβολή από τις 3 διαστάσεις στις 2 αντίστοιχα. Οι πίνακες διαβάζονται από τα αντίστοιχα αρχεία «calibration». Ύστερα τα σημεία πρέπει να μεταφερθούν από το τρισδιάστατο σύστημα συντεταγμένων των point clouds στο τρισδιάστατο σύστημα συντεταγμένων της κάμερας, οπότε πολλαπλασιάζονται με τον αντίστοιχο πίνακα $[R|T]$ που μαζί αποτελεί την περιστροφή και μεταφορά που χρειάζεται το αρχικό σύστημα για να εκφραστεί στο σύστημα της κάμερας. Εν συνεχεία για την μελέτη της χρήσιμης πληροφορίας των clouds αγνοούνται τα σημεία με $reflectance < 0$ και στον τελικό πίνακα μπαίνουν μόνο αυτά με $reflectance \geq 0$. Στην συνέχεια πρέπει να ληφθούν υπ'όψιν μόνο τα σημεία που βρίσκονται μπροστά από την κάμερα εφ'όσον η αναγνώριση αντικειμένου γίνεται σε κάθε frame ξεχωριστά και συνεπώς πρέπει να διατηρηθεί το μέρος του point cloud το οποίο θα προβληθεί στην αντίστοιχη εικόνα και όχι ολόκληρο το cloud. Έτσι αγνοούνται τα σημεία με $z < 0$ και τα υπόλοιπα σημεία τοποθετούνται σε έναν πίνακα. Μετά γίνεται η προβολή των σημείων εκφρασμένων στο τρισδιάστατο σύστημα της κάμερας, στο δισδιάστατο πλέον επίπεδο εικόνας της κάμερας. Παράλληλα είναι σημαντικό τα σημεία που έχουν προβληθεί να είναι εντός του ύψους και πλάτους της εικόνας αλλιώς δεν μπορεί να γίνει οπτικοποίηση των αποτελεσμάτων. Γι' αυτόν τον λόγο γίνεται έλεγχος για τα σημεία και συλλέγονται μόνο όσα ικανοποιούν τον περιορισμό αυτόν. Στο τέλος αποθηκεύονται για κάθε σημείο: οι αρχικές συντεταγμενες αν είναι εντός της κάμερας,τα reflectance values, τα χρώματα των points που έχουν προβληθεί πάνω στην εικόνα (3 κανάλια) και οι συντεταγμένες των σημείων που έχουν προβληθεί στο επίπεδο εικόνας (2 διαστάσεις). Για την εξοικονόμηση χώρου και ταχύτητας μπορούν τα επεξεργασμένα point clouds (cropped) να αντικαταστήσουν τα παλιά point clouds και να χρησιμοποιηθούν αυτά τα οποία είναι πυκνά σε πληροφορία αλλά μικρότερα σε μέγεθος αρχεία και η επεξεργασία τους είναι πιο αποδοτική.



Εικόνα 4.2. *Στο point cloud πάνω τα 2 οχήματα σημειωμένα με κόκκινο θα προβληθούν πάνω στο επίπεδο της κάμερας και στην κάτω εικόνα φαίνεται το περίγραμμα τους. Τα σημεία στο point cloud θα σχηματίσουν αυτό το περίγραμμα όταν γίνει η προβολή στις 2 διαστάσεις . Point Cloud από [75].*



Εικόνα 4.3. *Ένα παράδειγμα από περικοπή cloud. Επάνω φαίνεται το αρχικό cloud με όλα τα σημεία του. Κάτω είναι το κομμένο point cloud όπου ουσιαστικά λείπουν όλα τα σημεία αριστερά από τον κυκλικό κόμβο. Προφανώς το συγκεκριμένο cropping δεν είναι καλό εφ'όσον χάνονται αυτοκίνητα από το κομμένο cloud και αποτελεί απλώς ένα παράδειγμα. Point Cloud από [75].*

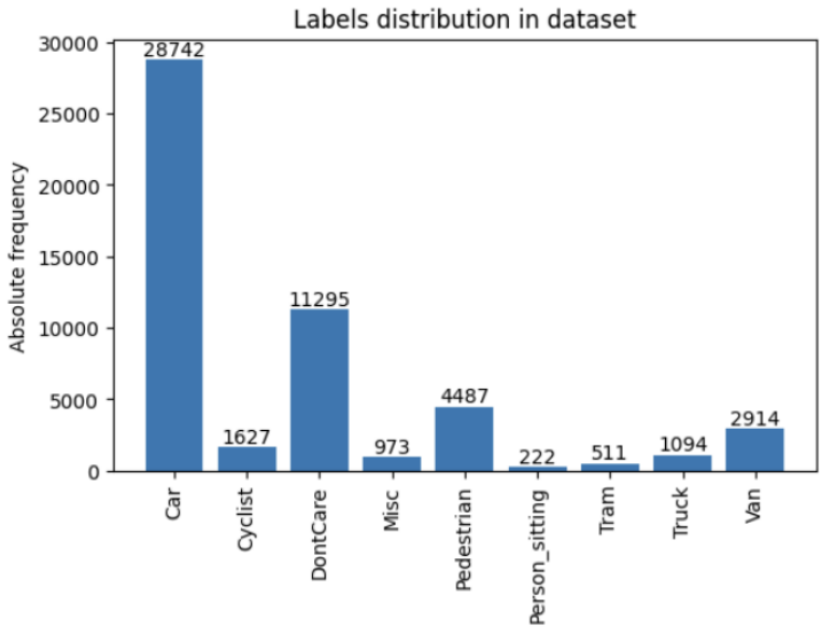
Τα cropped point clouds αποθηκεύονται μαζί με τις εικόνες και τα αρχεία ετικετών και calibration που τους αντιστοιχούν καθώς θα αποτελέσουν την είσοδο στην εφαρμοζόμενη αρχιτεκτονική.

4.2.2 Διαίρεση των δεδομένων

Εν συνεχεία πρέπει να γίνει η διαίρεση των δεδομένων σε δεδομένα εκπαίδευσης και δεδομένα επαλήθευσης. Η διαίρεση στα 2 σετ γίνεται φροντίζοντας να μην καταλήγουν δεδομένα ταυτόχρονα και στα 2 σετ. Πέρα από τα clouds χωρίζονται οι εικόνες, τα αρχεία ετικετών και τα calibration files με τον ίδιο τρόπο ώστε να διατηρείται η αντιστοιχία. Ακολουθείται το σχήμα διαίρεσης 3769 δεδομένα στο σετ επαλήθευσης και 3712 δεδομένα στο σετ εκπαίδευσης. Το σχήμα χρησιμοποιείται σαν αρχικό σχήμα και στα πειράματα δοκιμάζονται επίσης άλλα σχήματα με στόχο την καλύτερη εκπαίδευση του μοντέλου.

4.2.3 Ανισορροπία των Κλάσεων και Augmentation των Δεδομένων

Από την παρατήρηση της συχνότητας των διαφόρων κλάσεων στο Kitti Dataset γίνεται σαφές πως τα δεδομένα των αυτοκινήτων και πεζών ξεπερνάνε όλες τις υπόλοιπες κλάσεις σε συχνότητα εμφάνισης ενώ πολλές κλάσεις όπως τα «Tram» , «Misc» εμφανίζονται με αμελητέα συχνότητα. Για τον παραπάνω μπορεί να γίνει είτε υποδειγματοληψία των δεδομένων των αυτοκινήτων και των πεζών (των κυρίαρχων κλάσεων) είτε να γίνει υπερδειγματοληψία των κλάσεων με την μικρότερη συχνότητα έτσι ώστε να γίνει πιο ισορροπημένο το dataset. Και οι 2 τεχνικές είναι δόκιμες και για τον σκοπό του πειράματος γίνεται δοκιμή και των 2 τεχνικών. Επιπλέον το random sampling από τυχαίες θέσεις του dataset είναι μια μέθοδος που σε συνδυασμό με τις παρακάτω μπορεί να βελτιώσει την εκπαίδευση. Για την υπερδειγματοληψία των δεδομένων των πιο «σπάνιων» κλάσεων εφόσον δεν διατίθενται περισσότερα δεδομένα, είναι δυνατόν να εφαρμοστούν τεχνικές κατασκευής πλασματικών δεδομένων οι οποίες συνίστανται σε δημιουργία αντιγράφων των δεδομένων και τροποποίηση τους με την εφαρμογή γεωμετρικών μετασχηματισμών όπως περιστροφών των γνωστών αντικειμένων -ground truths clouds γύρω από τον άξονα των z ,εφαρμογή μετατοπίσεων (πολλαπλασιασμός με πίνακα R και Translation), ολική μεγέθυνση των clouds και ολική περιστροφή των clouds. Οι τιμές για την εκάστοτε επεξεργασία επιλέγονται από την ομοιόμορφη κατανομή.



Σχήμα 4.1. *Η κατανομή των κλάσεων στο Kitti Dataset. Παρατηρείται μεγάλη ανισορροπία κλάσεων με κυρίαρχη αυτή των αυτοκινήτων και επόμενη την κλάση DontCare. Μετά ακολουθούν όλες οι υπόλοιπες [77].*

Data Augmentation

Το augmentation των δεδομένων βασίζεται στις προτεινόμενες τεχνικές του [73]. Πιο συγκεκριμένα, ορίζεται το σετ $M = \{p_i = [x_i, y_i, z_i, r_i]^T \in R^4\}_{\text{i=1,...,N}}$ ως ολόκληρο το point cloud που αποτελείται από N points. Κάθε τρισδιάστατο κυτίο γνωστής κλάσης b_i ορίζεται ως $(x_c, y_c, z_c, l, w, h, \theta)$, όπου (x_c, y_c, z_c) είναι οι τοποθεσίες των κέντρων των κυτίων και (l, w, h) είναι το μήκος, το πλάτος και το ύψος ενώ θ είναι η γωνία γύρω από τον άξονα των Z. Στην συνέχεια ορίζεται το $\Omega_i = p|x \in [x_c - l2, x_c + l2], y \in [y_c - w2, y_c + w2], z \in [z_c - h2, z_c + h2], p \in M$ ως το σύνολο που περιέχει όλα τα σημεία του point cloud εντός του

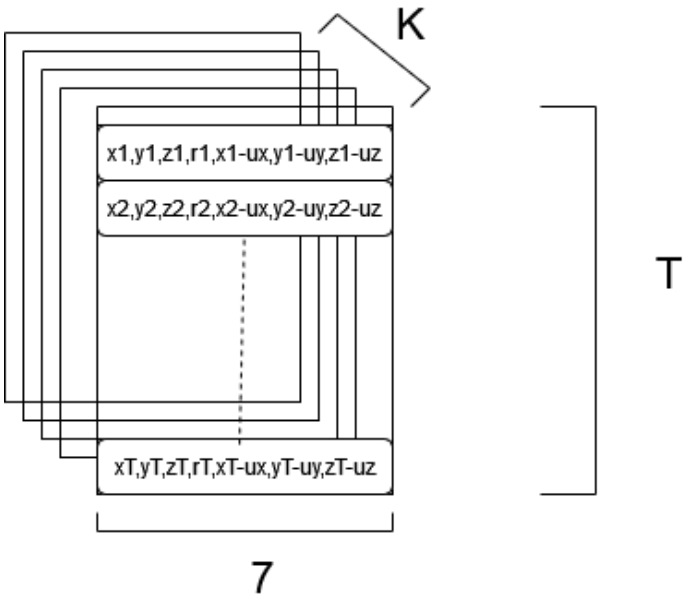
b_i, όπου $p = [x, y, z, r]$ είναι το εκάστοτε σημείο του cloud εντός του συνόλου M. Αρχικά γίνεται περιστροφή και μεταφορά των κυτίων γνωστής κλάσης και των σημείων που περιέχονται εντός αυτών. Η περιστροφή του b_i και του σχετικού Ω_i γίνεται γύρω από τον άξονα Z ως προς το κέντρο (x_c, y_c, z_c), με γωνία $\Delta\theta$ ομοιόμορφα κατανεμημένη που προκύπτει ψευδοτυχαία στο διάστημα $[-\pi/10, \pi/10]$ δηλαδή $\Delta\theta \in [-\pi/10, \pi/10]$. Εν συνεχεία γίνεται μεταφορά του κέντρου (x_c, y_c, z_c) με την προσθήκη μετατόπισης $(\Delta x, \Delta y, \Delta z)$, στο κέντρο και στα σημεία Ω_iπου περιέχει το κυτίο. Οι μετατοπίσεις επιλέγονται και πάλι ψευδοτυχαία από κανονική κατανομή Gauss με μέση τιμή 0 και τυπική απόκλιση 1.0. Επιπλέον, καθώς είναι δυνατόν πολλά κυτία μετά τους μετασχηματισμούς να συγκρούονται, γίνεται έλεγχος αν συμβαίνει κάτι τέτοιο και σε αυτή την περίπτωση αναιρείται ο μετασχηματισμός. Ο συνδυασμός μετασχηματισμός σε κυτία και σημεία δημιουργεί ποικιλία στα δεδομένα και ενισχύει την μάθηση πιο περίπλοκων συνδυασμών. Ύστερα γίνεται αλλαγή κλίμακας και πιο συγκεκριμένα μεγέθυνση στα κυτία των γνωστών κλάσεων b_i και σε ολόκληρο το Point Cloud M. Η μεγέθυνση γίνεται με τον πολλαπλασιασμό των (x_c, y_c, z_c) των κέντρων των κυτίων και και των (l, w, h) καθώς και των συντεταγμένων (x, y, z) των σημείων του M, με τυχαία μεταβλητή που επιλέγεται από ομοιομορφη κατανομή [0.95,1.05]. Η μεγέθυνση δίνει την δυνατότητα στο μοντέλο να αναγνωρίζει αντικείμενα διαφορετικών μεγεθών και αποστάσεων [78, 79, 80]. Ακολούθως εφαρμόζεται περιστροφή στα κυτία γνωστών κλάσεων και σε ολόκληρο το Point Cloud M. Η περιστροφή γίνεται γύρω από τον άξονα των Z και γύρω από το (0,0,0). Η μετατόπιση περιστροφής επιλέγεται από ομοιόμορφη κατανομή με διάστημα $[-\pi/4, \pi/4]$. Η περιστροφή ολόκληρου του Point Cloud δημιουργεί νέα στιγμιότυπα και προσομοιώνει στροφές των οχημάτων αλλά και των άλλων κλάσεων.

4.2.4 Δημιουργία των Voxels

Για την κατασκευή του Feature Encoding Network πρέπει να γίνει διαίρεση του Point Cloud σε Voxels ώστε όλα τα σημεία του cloud να ανήκουν στο voxel grid και να περιέχονται σε κάποιο voxel. Για την αναγνώριση αυτοκινήτων λαμβάνονται υπόψιν τα σημεία των clouds εντός του εύρους $[-3,1] \times [-40,40] \times [0,70.4]$ για τους άξονες Z, Y, X. Εεπειδή τα σημεία κωδικοποιούνται με 1η διάσταση το X, εναλλάσονται οι συντεταγμένες της 3ης με την 1η στήλη $(X, Y, Z \rightarrow Z, Y, X)$. Τα σημεία των οποίων η προβολή στο επίπεδο εικόνας τα τοποθετεί πέρα από τα όρια του αφαιρούνται. Επιλέγεται μέγεθος voxel $\{u_D = 0.4, u_H = 0.2, u_W = 0.2\}$, οπότε προκύπτει $Voxel\ Grid\ με\ μέγεθος = \{D' = D/u_D = 10, H' = H/u_H = 400, W' = W/u_W = 352\}$. Προστίθεται επίσης η μετατόπιση για την μεταφορά από το point cloud στο σύστημα συντεταγμένων του lidar scanner. Πριν γίνει περαιτέρω επεξεργασία των σημείων, αυτά «ανακατεύονται» .Από κάθε σημείο αφαιρείται το reflectance value καθώς δεν χρειάζεται για το

Feature Learning Network. Εν συνεχεία γίνεται η κατανομή των σημείων στα voxels, μόνο για τα σημεία που είναι εντός του voxel grid. Γι'αυτά τα σημεία η διαίρεση με το μέγεθος voxel και αφαίρεση του κλασματικού μέρους αντιστοιχίζει κάθε σημείο σε ένα συγκεκριμένο voxel id. Στην συνέχεια κατασκευάζεται το Coordinate Buffer το οποίο περιέχει όλα τα μοναδικά Voxel Id, τα διαφορετικά Voxel του Cloud. Είναι ένας πίνακας $(K \times 3)$ με K μοναδικά voxel Id και σε κάθε ένα τις συντεταγμένες των αντίστοιχων voxel .

Για την συλλογή της πληροφορίας που βρίσκεται στο point cloud και πρέπει να κωδικοποιηθεί, η οποία είναι οι συντεταγμένες των σημείων του κάθε voxel και οι χωρικές σχέσεις μεταξύ τους, γίνεται δειγματοληψία ενός αριθμού σημείων το πολύ ίσος με T για κάθε voxel. Για την αναγνώριση αυτοκινήτων ο αριθμός ορίζεται ίσος με 35. Για την κωδικοποίηση των χαρακτηριστικών του cloud κατασκευάζεται το Voxel Input Feature Buffer, το οποίο είναι ένας τανυστής 3 διαστάσεων $(K \times T \times 7)$ που περιέχει τα K μοναδικά voxel id και σε κάθε voxel id αντιστοιχίζει τις **z,y,x** συντεταγμένες των T το πολύ σημείων που περιέχει. Ο εντοπισμός του voxel id στο οποίο ανήκει κάθε σημείο μπορεί να γίνει σε O(1) με την δημιουργία ενός λεξικού το οποίο περιέχει σαν κλειδιά της συντεταγμένες του κάθε μοναδικού voxel id και το αντίστοιχο voxel id σαν τιμή (index Buffer). Κάθε σημείο του οποίου οι συντεταγμένες ταυτίζονται με τις συντεταγμένες ενός μοναδικού voxel id θα ανήκει σε αυτό. Η κατασκευή των Voxel Input Feature Buffer και του Coordinate Buffer γίνεται ταυτόχρονα με O(n) πολυπλοκότητα. Επίσης δημιουργείται και πίνακας $(K \times T)$ που για κάθε μοναδικό voxel περιέχει τον αριθμό των σημείων που ελήφθησαν από την δειγματοληψία (number Buffer). Ακολούθως κατασκευάζεται ένα λεξικό που κωδικοποιεί σε κάθε Point cloud, το Voxel Input Feature Buffer του, το Coordinate Buffer του και τον πίνακα του αριθμού των σημείων του κάθε voxel.

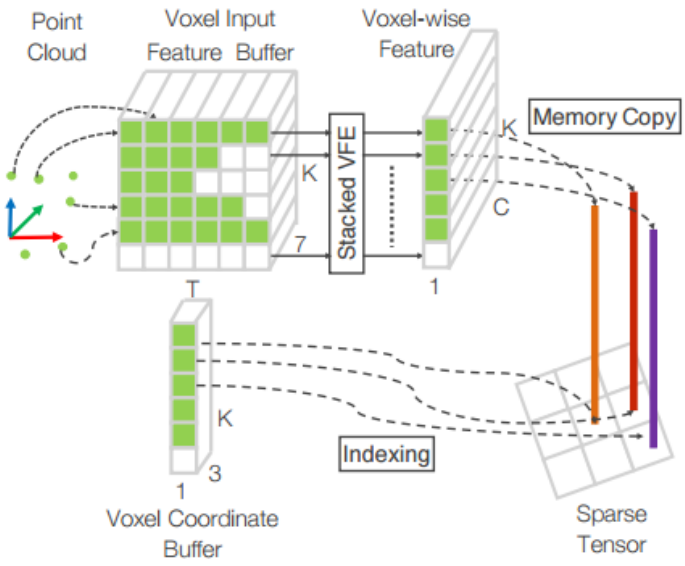


Σχήμα 4.2. *Voxel Input Feature Buffer. Ο τανυστής αυτός έχει σαν 1η διάσταση τα K voxels και κάθε voxel περιλαμβάνει ένα σύνολο από 1 points μέχρι T points (γραμμές), με κάθε point να κωδικοποιείται σαν ένα σύνολο 7 χαρακτηριστικών (στήλες). Τανυστής KxTx7.*

4.3 Κατασκευή των Voxel Feature Encoding Layers

Στην συνέχεια τα voxels που έχουν δημιουργηθεί διέρχονται μέσα από μια αλυσίδα Voxel Feature Layers με στόχο να κωδικοποιηθεί η πληροφορία που περιλαμβάνουν αλλά και η σχέση μεταξύ των σημείων σε κάθε voxel. Αρχικά υπολογίζεται ο τοπικός μέσος (centroid) των σημείων που ανήκουν κάθε voxel και περιγράφεται ως (u_x, u_y, u_z). Στην συνέχεια γίνεται τροποποίηση κάθε σημείου pi με την σχετική απόσταση από το centroid και λαμβάνεται το input feature set του voxel $V_{\text{in}} = \{\hat{p}_i = [x_i, y_i, z_i, r_i, x_i - \upsilon_x, y_i - \upsilon_y, z_i - \upsilon_z]^T \in R^7\}_{i=1\ldots t}$. Στην συνέχεια κάθε σημείο $\hat{p}_i$ εισάγεται σε ενα fully connected network (FCN), με ένα γραμμικό επίπεδο (1), ένα επίπεδο Batch Normalization (2), ένα επίπεδο διορθωμένης γραμμική μονάδας (ReLU) (3).Έτσι λαμβάνονται τα pointwise χαρακτηριστικά, τα χαρακτηριστικά των σημείων που ανήκουν στο εκάστοτε voxel. Στην συνέχεια γίνεται Max Pooling των σημειακών χαρακτηριστικών και γίνεται ένωση του αποτελέσματος με το χαρακτηριστικό κάθε σημείου. Το χαρακτηριστικό Feature κάθε voxel είναι το Voxel Input Feature Buffer ,ο τανυστής $(K \times T \times 7)$ που

αναφέρθηκε παραπάνω, και αυτός ο τανυστής δίνεται ως είσοδος στο Voxel Feature Encoding Network(VFE). Τα Voxel Feature Encoding Layers έχουν την μορφή: $VFE - i(c_{\text{in}}, c_{\text{out}})$, όπου το c_{in} δηλώνει είσοδο διάστασης c_{in} και το c_{out} , ότι το επίπεδο δίνει έξοδο διάστασης c_{out}. Για την αναγνώριση αυτοκινήτων η αλυσίδα των VFE layers αποτελείται από 2 VFE επίπεδα-layers: $VFE - 1(7,32)$ και $VFE - 2(32,128)$. Η έξοδος του $VFE - 2(32,128)$ στην συνέχεια εισάγεται σε ένα πλήρως συνδεδεμένο δίκτυο (FCN) και η έξοδος του FCN ύστερα από ένα επίπεδο element-wise Max Pooling αποτελεί το Voxel-wise χαρακτηριστικό για το εκάστοτε Voxel. Όλα τα voxelwise features μαζί αποτελούν έναν αραιό τετραδιάστατο τανυστή $(K \times T \times 7)$ όπου 10 , 400, 352 είναι οι διαστάσεις το voxel πλέγματος.



Σχήμα 4.3. *Η κωδικοποίηση των χαρακτηριστικών για κάθε voxel. Στο τέλος προκύπτει ένας αραιός τανυστής μεγέθους C x D' x H' x W' ο οποίος κωδικοποιεί τα vowelwise χαρακτηριστικά του point cloud. Εικόνα από [73].*

4.4 Ενδιάμεσα Συνελικτικά επίπεδα

Για την άθροιση των voxel-wise χαρακτηριστικών γίνεται χρήση τριών ενδιάμεσων συνελικτικών επιπέδων. Εδώ χρησιμοποιείται πάλι η έκφραση $ConvMD\ (c_{\text{in}}, c_{\text{out}}, k, s, p)$ για την περιγραφή ενός συνελικτικού επιπέδου και οι εφαρμόζονται οι εξισώσεις του κεφαλαίου 2, ενότητας **2.2.1 *,Εξισώσεις συνελικτικών και pooling επιπέδων*.**

1ο Συνελικτικό Επίπεδο

Το 1ο συνελικτικό επίπεδο $ConvMD\ (128{,}64, 3, (2{,}1{,}1), (1{,}1{,}1))$ δέχεται τον τανυστή $128 \times 10 \times 400 \times 352$ και εφαρμόζει συνέλιξη με μέγεθος πυρήνα (kernel size) $= (3 \times 3 \times 3)$, βήμα (stride) $= (2 \times 1 \times 1)$ και padding $= (1 \times 1 \times 1)$. Το αποτέλεσμα αυτό προκύπτει από την εφαρμογή της εξίσωσης **2.7, 2.8, 2.9** του κεφαλαίου 2.

$H1 = (H - F + 2P)/S + 1 = (400 - 3 + 2 \times 1)/1 + 1 = 400,\ W1 = (W - F + 2P)/S + 1 = (352 - 3 + 2 \times 1)/1 + 1 = 352,\quad D1 = (D - F + 2P)/S + 1 = (10 - 3 + 2 \times 1)/2 + 1 = 5.$
Συνεπώς η έξοδος από το 1ο συνελικτικό επίπεδο είναι Τανυστής με διαστάσεις $(64 \times 5 \times 400 \times 352)$.

2ο Συνελικτικό Επίπεδο

Στην συνέχεια αυτός ο τανυστής διέρχεται από το 2ο συνελικτικό επίπεδο $ConvMD\ (64{,}64, 3, (1{,}1{,}1), (0{,}1{,}1))$ οπότε προκύπτει τανυστής $(64 \times 3 \times 400 \times 352)$ με την εφαρμογή της εξίσωσης όπως παραπάνω :

$H2 = (H1 - F + 2P)/S + 1 = (400 - 3 + 2 \times 1)/1 + 1 = 400, W2 = (W1 - F + 2P)/S + 1 = (352 - 3 + 2 \times 1)/1 + 1 = 352, D2 = (D1 - F + 2P)/S + 1 = (5 - 3 + 2 \times 0)/1 + 1 = 3$

3ο Συνελικτικό Επίπεδο

Το 3ο συνελικτικό επίπεδο είναι της μορφής $ConvMD\ (64{,}64, 3, (2{,}1{,}1), (1{,}1{,}1))$ και παράγει τανυστή της μορφής $(64 \times 2 \times 400 \times 352)$.Η ανάλυση γίνεται όπως και παραπάνω :

$H3 = (H2 - F + 2P)/S + 1 = (400 - 3 + 2 \times 1)/1 + 1 = 400, W3 = (W2 - F + 2P)/S + 1 = (352 - 3 + 2 \times 1)/1 + 1 = 352, D3 = (D2 - F + 2P)/S + 1 = (3 - 3 + 2 \times 1)/2 + 1 = 2$

Εν συνεχεία ο τελικός τανυστής μετασχηματίζεται (flattened) στις διαστάσεις $128 \times 400 \times 352$ με τις διαστάσεις να αντιστοιχούν σε $κανάλια \times ύψος \times πλάτος$.

4.5 Τελικό Region Proposal Network

Για την αναγνώριση των αυτοκινήτων γίνεται χρήση Region Proposal Network (RPN). Το RPN δίκτυο δέχεται σαν είσοδο την έξοδο των ενδιάμεσων συνελικτικών δικτύων $128 \times 400 \times 352$ και αποτελείται από επίπεδα downsampling και upsampling. Πιο συγκεκριμένα αποτελείται από 3 μπλοκ από συνελικτικά επίπεδα 2 διαστάσεων τα οποία συντελούν στο downsampling του Feature Map εισόδου. Το «x» στα παρακάτω δηλώνει πόσες φορές εφαρμόζεται η αντίστοιχη «συνέλιξη». Στο 1ο block εφαρμόζονται $Conv2D$(128,128,3,2,1) x 1 και μετά $Conv2D$(128,128,3,1,1) x 3. Στο 2ο μπλοκ εφαρμόζεται $Conv2D$(128,128,3,2,1) x 1 και $Conv2D$(128,128,3,1,1) x 5. Στο 3ο μπλοκ εφαρμόζεται $Conv2D$(128,256,3,2,1) x 1 και $Conv2D$(256,256,3,1,1) x 5. Σημαντικό είναι ότι μετά από κάθε επίπεδο συνέλιξης εφαρμόζεται Batch Normalization και ενεργοποίηση συνάρτησης ReLu. Η έξοδος από κάθε μπλοκ στην συνέχεια διέρχεται από deconvolutional επίπεδο με στόχο το upsampling των χαρακτηριστικών. Στην έξοδο του 1ου μπλοκ εφαρμόζεται $Deconv2D$ (128,256,3,1,0) x 1. Στην έξοδο του 2ου μπλοκ εφαρμόζεται $Deconv2D$ (128,256,2,2,0) x 1. Στην έξοδο του 3ου μπλοκ εφαρμόζεται $Deconv2D$ (256,256,4,4,0) x 1.Ακολούθως οι έξοδοι των deconvolutions, τανυστές διαστάσεων $128 \times 200 \times 176$ συνενώνονται και συναποτελούν το τελικό Feature Map (χάρτης χαρακτηριστικών) το οποίο έχει διαστάσεις $128 \times 200 \times 176 = (768 \times 200 \times 276)$. Το Feature Map αυτό «σπάει» σε 2 συνσιστώσες: τον χάρτη πιθανοτήτων να είναι ή να μην είναι αντικείμενο η περιοχή που εξετάζεται (Probability Score Map) με διαστάσεις $2 \times 200 \times 176$ και τον χάρτη εντοπισμού και διόρθωσης των bounding boxes, για την παραμετροποίηση των κυτίων αναγνώρισης (Regression map) με διαστάσεις $14 \times 200 \times 176$. Το σχήμα της ενότητας 3.5 δείχνει το τελικό RPN και την αντιστοίχιση σε probability score map και regression map. Για την αναγνώριση αυτοκινήτων γίνεται χρήση προκαθορισμένου κυτίου αναγνώρισης (anchor) 1 μεγέθους με διαστάσεις $l^a = 3.9, w^a = 1.6, h^a = 1.56$ μέτρα με κέντρο το $z_c^a = -1.0$ μέτρα και 2 περιστροφές 0 και 90 μοίρες. Συνεπώς υπάρχουν 2 προκαθορισμένα κυτία αναγνώρισης εξ'ού και η διάσταση 14 (= 2 *x* 7) στην αντιστοίχιση στα Regression Targets παραπάνω. Ένα προκαθορισμένο κυτίο αναγνώρισης θεωρείται θετικό αν το IoU με ένα κυτίο γνωστής κλάσης (ground truth box) είναι $IoU > 0.6$ ή το κυτίο αναγνώρισης έχει την μέγιστη IoU με ένα κυτίο γνωστής κλάσης. Αντίστοιχα το προκαθορισμένο κυτίο αναγνώρισης θεωρείται αρνητικό αν το IoU του με όλα τα κυτία γνωστών κλάσεων είναι $IoU < 0.45$. Αν το IoU του προκαθορισμένου κυτίου αναγνώρισης με οποιοδήποτε κυτίο γνωστής κλάσης είναι $0.45 \leq IoU \leq 0.6$ θεωρείται αδιάφορο. Για την αναγνώριση των αυτοκινήτων το α και β στην συνάρτηση απωλειών τίθενται $a = 1.5, \beta = 1$.

ΚΕΦΑΛΑΙΟ 5
Μοντέλα και Αποτελέσματα

Στο κεφάλαιο παρουσιάζονται τα εφαρμοζόμενα μοντέλα και γίνεται η αξιολόγηση τους. Επίσης παρατίθενται επιπλέον τεχνικές που δοκιμάστηκαν για την δειγματοληψία και κάποιες πιθανές προτάσεις για μελλοντική εργασία. Οι εικόνες του κεφαλαίου (πρόσθια και κάτοψη), στις οποίες γίνονται οι προβολές των προβλέψεων για την οπτικοποίηση της απόδοσης κάθε μοντέλου, βρίσκονται κάτω από τα precision-recall curve κάθε μοντέλου και είναι στιγμιότυπα 'images' από το Kitti Dataset [75]. Όλες οι εικόνες του κεφαλαίου ,στις οποίες οπτικοποιούνται τα αποτελέσματα προβλέψεων με κόκκινο (ground truths) και πράσινο χρώμα (predicted bounding boxes) είναι από το Kitti Dataset [75]. Ο κώδικας της εργασίας θα φιλοξενείται στην προσωπική ιστοσελίδα του συγγραφέα στο github [84].

5.1 Εκπαίδευση και Δοκιμαζόμενα Μοντέλα

5.1.1 Παράμετροι όλων των μοντέλων

Σε όλα τα μοντέλα χρησιμοποιήθηκε για την τροποποίηση των βαρών και την ελαχιστοποίηση της συνάρτησης απωλειών το Stochastic Gradient Descent (SGD). Επίσης επιλέχθηκε ο Adam optimizer σαν αλγόριθμος βελτιστοποίησης του gradient descent εφόσον διαπιστώθηκε ύστερα από δοκιμές ότι η χρήση άλλων αλγορίθμων όπως RMSProp και Momentum δεν οδηγούσαν σε καλύτερα αποτελέσματα από τον Adam. Επιπλέον το μέγεθος παρτίδας τέθηκε ίσο με 2 εφόσον η χρήση μεγέθους 1 οδηγεί σε χειρότερα αποτελέσματα και η χρήση μεγαλύτερων μεγεθών απαιτεί την κατανάλωση υψηλότερης υπολογιστικής ισχύος, η οποία δεν ήταν διαθέσιμη. Τα clouds από τα οποία μπορεί να γίνει διαχωρισμός σε σύνολο εκπαίδευσης και σύνολο επαλήθευσης προέκυψαν με αποκοπή των αρχικών clouds κρατώντας τα points με $reflectance \geq 0$, εφόσον παρατηρήθηκε πως η απόρριψη των points με $reflectance = 0$, οδηγεί σε απώλεια σημαντικής πληροφορίας του κάθε cloud με αποτέλεσμα την επιδείνωση της παρατηρούμενης ακρίβειας. Επιπλέον για τις τιμές των alpha και beta δοκιμάστηκαν οι συνδυασμοί $alpha = \{1.5,1.0,0.9\}$ και $alpha = \{1.5,1,0.9\}$ και τελικά αποφασίστηκε alpha=1.5 και beta =1.0 εφόσον ο συνδυασμός αυτός οδηγεί σε καλύτερη ακρίβεια. Σημαντικό είναι ότι σύμφωνα και με την θεωρία οι συντελεστές αυτοί δεν επηρεάζουν σε μεγάλο βαθμό τα αποτελέσματα , σε ένα μεγάλο διάστημα τιμών τους.

5.1.2 Συμβάσεις που ακολουθήθηκαν και συμβολισμοί

Για την παρουσίαση των παραμέτρων κάθε μοντέλου στους σχετικούς πίνακες χρησιμοποιήθηκαν οι παρακάτω συμβολισμοί:

Mx : Το όνομα κάθε μοντέλου, x ο δείκτης του.

Εποχές : E

Μέγεθος Παρτίδας : BS

Ρυθμός εκμάθησης : lr

Alpha : a

Beta : b

Μέθοδος τροποποίησης των δεδομένων για την αντιμετώπιση της ανισορροπίας κλάσεων και την δημιουργία ποικιλίας δεδομένων : aug, όπου θα αναγράφεται "def" (από το "default") εάν χρησιμοποιήθηκε η τροποποίηση που παρουσιάστηκε στο κεφάλαιο 4, ενότητα 4.2.3, ή θα αναγράφεται "-", που παραπέμπει τον αναγνώστη στην ανάλυση του σχετικού μοντέλου στο κείμενο που το συνοδεύει.

Μέγεθος συνόλου εκπαίδευσης : Tr

Μέγεθος συνόλου επαλήθευσης : Val

Αρχιτεκτονική Συνελικτικών Επιπέδων(Ενδιάμεσων και RPN): $Conv$, όπου θα αναγράφεται "def", εάν χρησιμοποιήθηκε η αρχιτεκτονική που παρουσιάστηκε στο κεφάλαιο 4 ή θα αναγράφεται "-" ,που παραπέμπει τον αναγνώστη στην ανάλυση του σχετικού μοντέλου στο κείμενο που το συνοδεύει.

Αρχιτεκτονική VFE Επιπέδων: VFE , όπου θα αναγράφεται "def" εάν χρησιμοποιήθηκε η αρχιτεκτονική που παρουσιάστηκε στο κεφάλαιο 4 ή θα αναγράφεται "-" ,που παραπέμπει τον αναγνώστη στην ανάλυση του σχετικού μοντέλου στο κείμενο που το συνοδεύει.

Περιγραφή Κλάσεων Δυσκολίας και Αξιολόγηση

Η αναφορά του αποτελέσματος αποτελείται από 3 τιμές για τις 3 κλάσεις δυσκολίας, Easy, Moderate, Hard. Στόχος είναι η μεγιστοποίηση τους. Η σουίτα Kitti χρησιμοποιεί τα παρακάτω κριτήρια για να κατατάξει τα στιγμιότυπα στις κλάσεις δυσκολίας [75].

Difficulty Class	Bounding Box height	Max. occlusion level	Max. truncation
Easy	40 px	Fully visible	15 %
Moderate	25 px	Partly Occluded	30 %
Hard	25 px	Difficult to see	50 %

Πιο συγκεκριμένα η κλάση δυσκολίας moderate ορίζεται υπό τις ακόλουθες συνθήκες:

- Max Truncation = 0.3
- Occlusion = 0 ή 1
- Max bounding box height = 25.0 px, όπου bounding box height = top – bottom,

Με βάση τα παραπάνω οι μετρικές ακρίβειας που χρησιμοποιούνται είναι:

- Μέση ακρίβεια αναγνώρισης κάτοψης σε 2 διαστάσεις για τις 3 κλάσεις δυσκολίας (Bird's Eye View AP) με συμβολισμό: BEV AP
- Μέση ακρίβεια τρισδιάστατης αναγνώρισης για τις 3 κλάσεις δυσκολίας (3D AP) με συμβολισμό: 3DAP

Η αξιολόγηση είναι σύμφωνη με το σύνολο δεδομένων PASCAL και τα κριτήρια που ορίζει [56]. Τα αντικείμενα φιλτράρονται με βάση το bounding box height στο επίπεδο της εικόνας. Στην αξιολόγηση δεν λαμβάνονται υπόψιν σαν False Positives τα αντικείμενα που ανήκουν στην κλάση «Don't Care». Αντιθέτως τα αντικείμενα που δεν είναι ορατά στο επίπεδο της εικόνας αλλά είναι μέρος του point cloud, λαμβάνονται υπ'όψιν και απαιτείται περαιτέρω επεξεργασία ώστε να μην συνυπολογίζονται σαν False Positives.

5.1.3 Απλά Μοντέλα

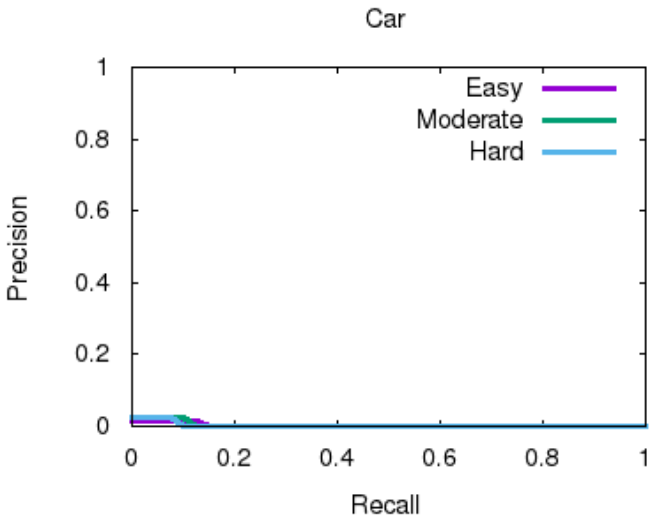
1. Μοντέλο 1

Αρχικά γίνεται τυχαία δειγματοληψία 400 point clouds και των αντίστοιχων εικόνων, calibration πινάκων της κάμερας και των αρχείων ετικετών που τους αντιστοιχούν. Αντίστοιχα το «σπάσιμο» των δεδομένων γίνεται με μέγεθος εκπαίδευσης=0.5 οπότε επιλέγονται και 400 clouds για το σύνολο επαλήθευσης .Το μέγεθος παρτίδας (batch size) επιλέγεται να είναι 2 ενώ ο ρυθμός εκμάθησης είναι σταθερός και ίσος με 0.01. Οι υπεραπαράμετροι alpha (α) και beta (β) στην συνάρτηση απωλειών τίθενται ίσες με 1.5 και 1 αντίστοιχα. Οι εποχές εκπαίδευσης επιλέγονται 10 με στόχο να γίνει μια γρήγορη αξιολόγηση ενός απλού μοντέλου. Τα επίπεδα του Feature Net και του RPN καθώς και τα μεγέθη πυρήνα βήματος και padding παρέμειναν σταθερά στις αρχικές τους τιμές, που παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Οι παράμετροι συνοψίζονται στον παρακάτω πίνακα :

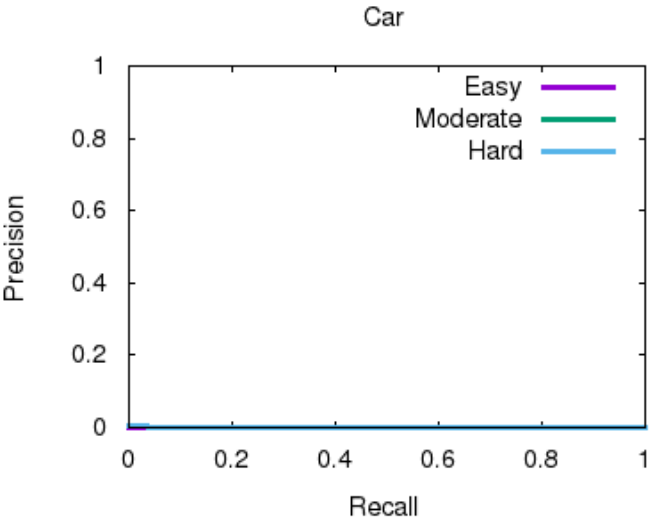
M1	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP			3d AP		
	10	2	0.01	1.5	1	def	400	400	def	def	0.23	0.42	0.22	0.01	0.04	0.04

Πίνακας 5.1.

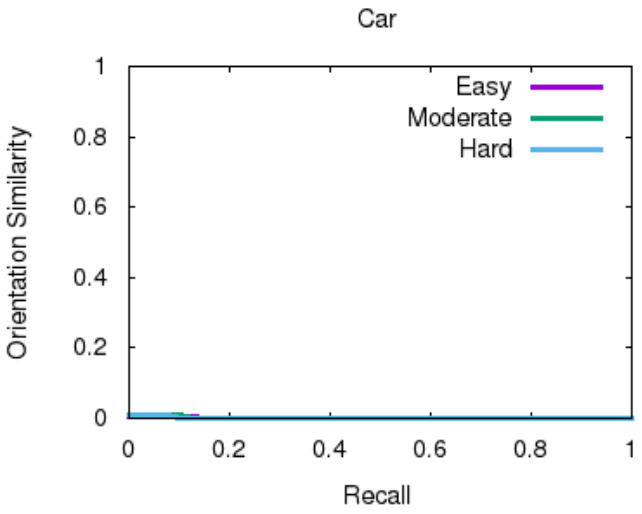
Επίσης δίνεται το διάγραμμα Precision Recall Curve για τον εντοπισμό σε bird’s eye view και σε 3D καθώς και για το orientation similarity μεταξύ ground truth box και predicted bounding box.



Σχημα 5.1. *Precision Recall Curve για bird's eye view detection.*



Σχήμα 5.2. *Precison Recall Curve για 3D Detection (3d):*



Σχήμα 5.3. *Precision Recall Curve για orientation similarity*

Ενδεικτικά παρουσιάζεται επίσης η απόκριση του μοντέλου σε ένα στιγμιότυπο cloud [75] στο οποίο έκανε τις παρακάτω προβλέψεις. Σε κάθε μοντέλο χρησιμοποιείται το ίδιο στιγμιότυπο ,από τα δεδομένα [75]. Τα bounding boxes που αναγνώρισε το μοντέλο ως αυτοκίνητα σημειώνονται με πράσινο χρώμα ενώ με ροζ σημειώνονται τα πραγματικά αυτοκίνητα που υπάρχουν στο cloud όπως είναι σημειώμενα στο αντίστοιχο αρχείο ετικετών του cloud. Η οπτικοποίηση γίνεται σε πρόσοψη και κάτοψη (Bird's Eye View).

Εικόνα 5.1. *Προβλέψεις σε μπροστά όψη*

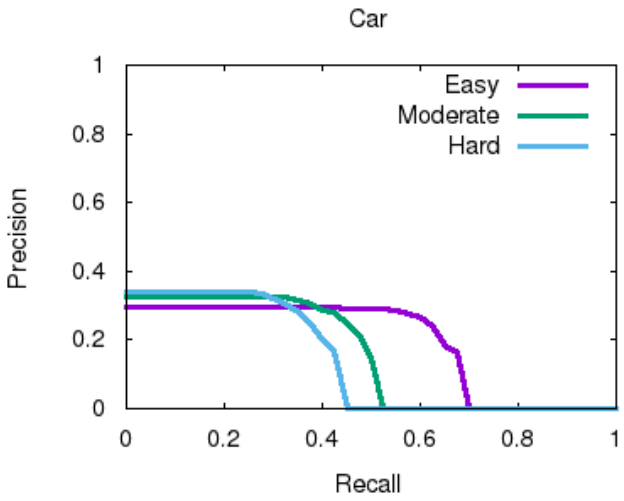
Εικόνα 5.2. *Προβλέψεις σε κάτοψη (bird' eye view).*

2. Μοντέλο 2

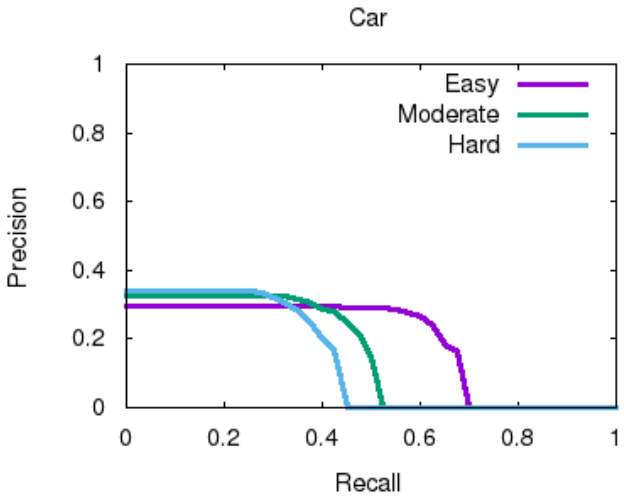
Στο μοντέλο αυτό χρησιμοποιούνται οι ίδιες παράμετροι με το μοντέλο 1 με την διαφορά ότι ο ρυθμός εκμάθησης τίθεται τώρα lr = 0.001 και η εκπαίδευση γίνεται και πάλι σε 400 clouds ενώ η επαλήθευση πάλι σε 400 clouds. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

M2	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP	3d AP
	10	2	0.001	1.5	1	def	400	400	def	def	18.4 15.7 14.02	2.88 2.88 2.26

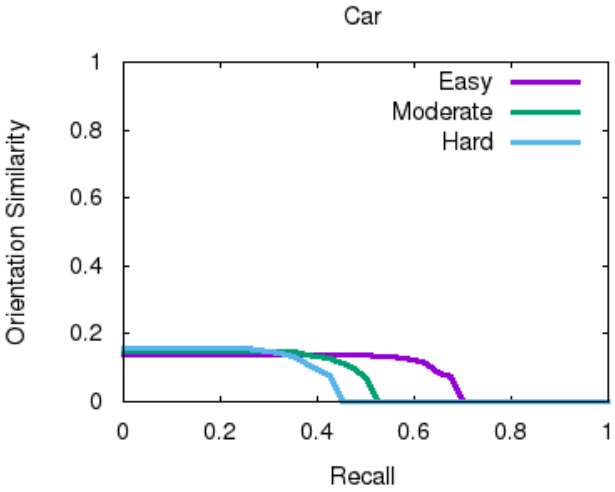
Πίνακας 5.2.



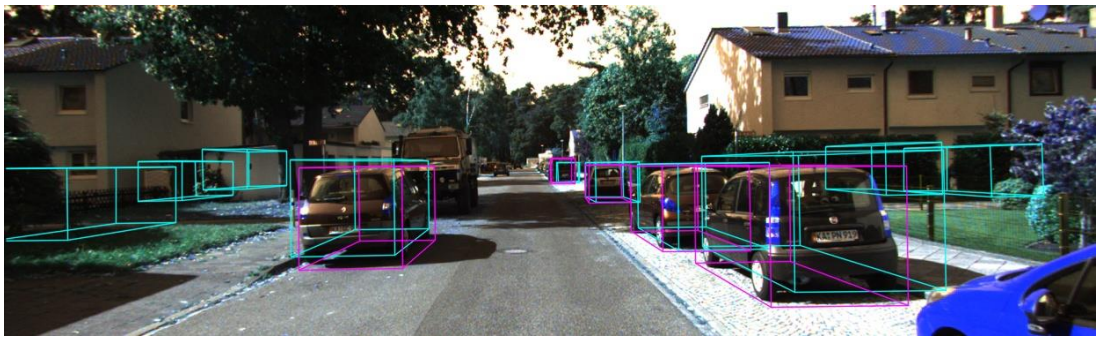
Σχήμα 5.4. *Precision Recall curve για Bird's Eye View Detection*



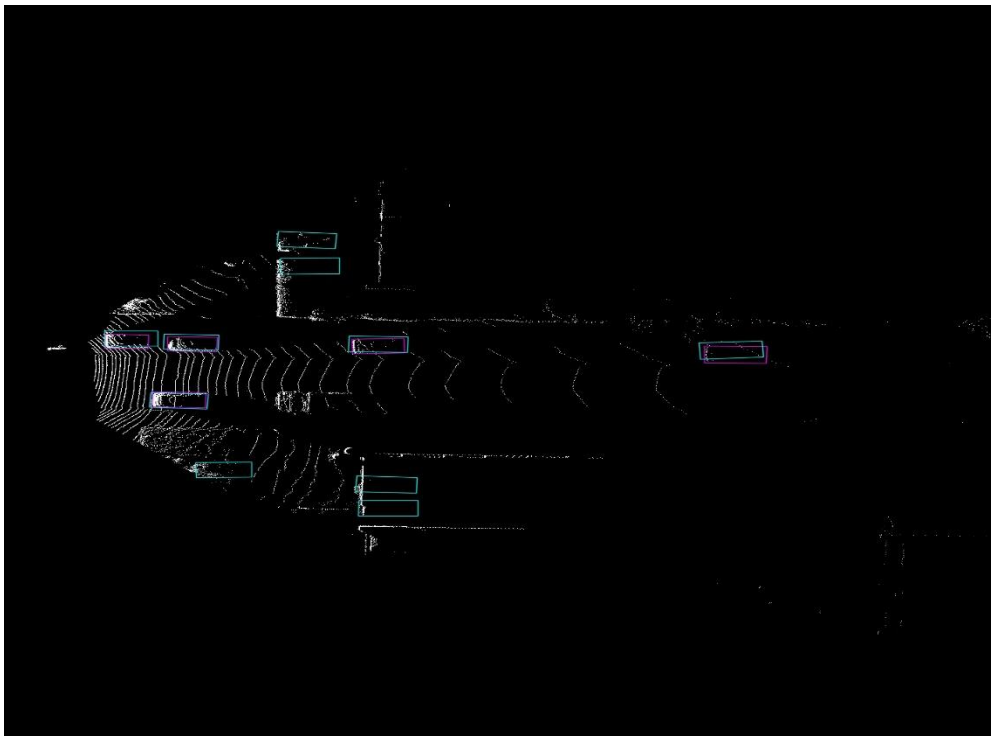
Σχήμα 5.5. *Precision recall curve για 3D detection.*



Σχήμα 5.6. *Precision Recall Curve για orientation similarity*.



Εικόνα 5.3. *Προβλέψεις σε μπροστά όψη*.



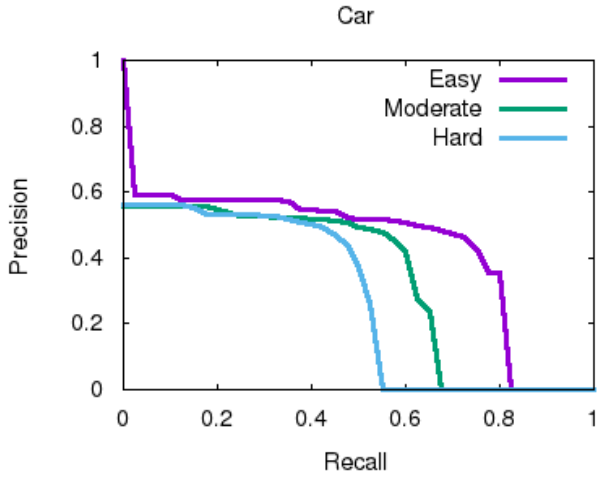
Εικόνα 5.4. *Προβλέψεις σε κάτοψη*

3. Μοντέλο 3

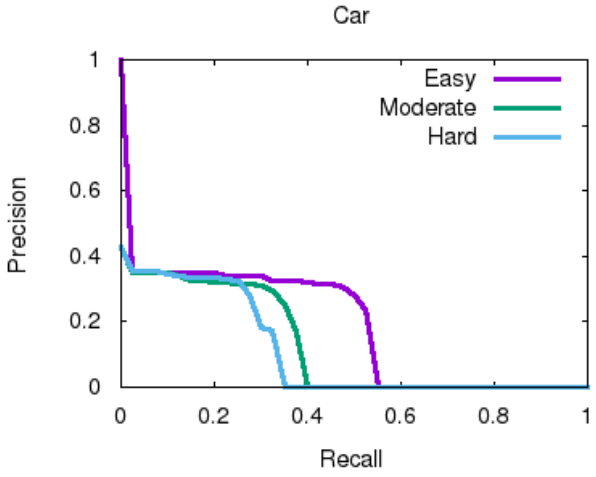
Στην συνέχεια δοκιμάζεται αύξηση του αριθμού των εποχών για την αξιολόγηση του ρόλου που εποχές εκπαίδευσης διαδραματίζουν στην εκπαίδευση του μοντέλου. Ταυτόχρονα υιοθετείται ο ρυθμός εκμάθησης lr =0.0001. Κατά τ'άλλα το μοντέλο 3 είναι πανομοιότυπο με το μοντέλο 2. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

M3	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP			3d AP		
	30	2	0.0001	1.5	1	def	400	400	def	def	46.73	32.8	27.8	23.9	12.7	11.76

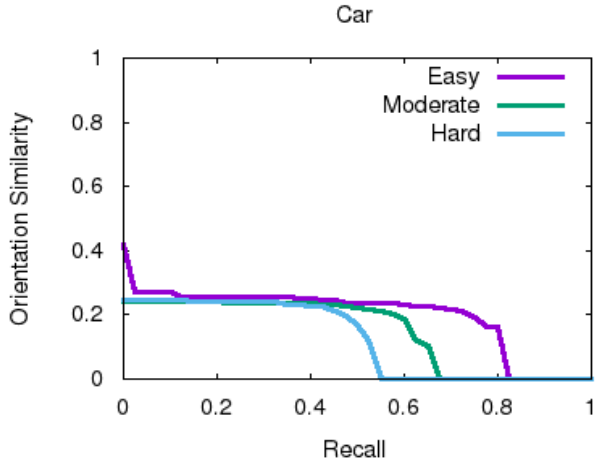
Πίνακας 5.3.



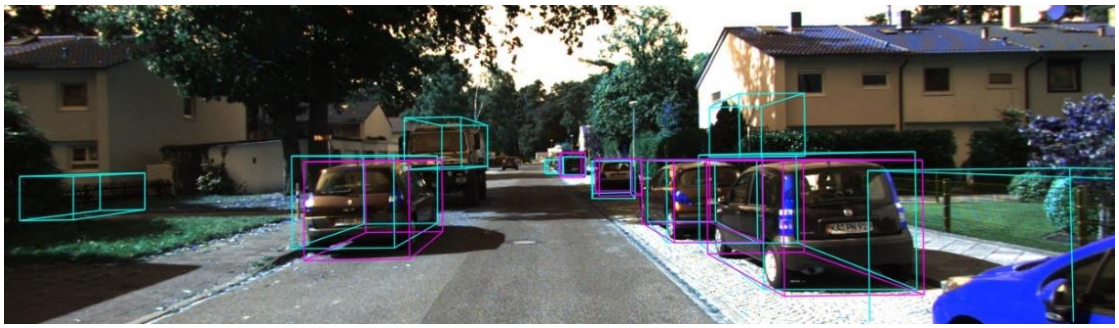
Σχήμα 5.7. *Precision Recall curve για Bird's Eye View Detection.*



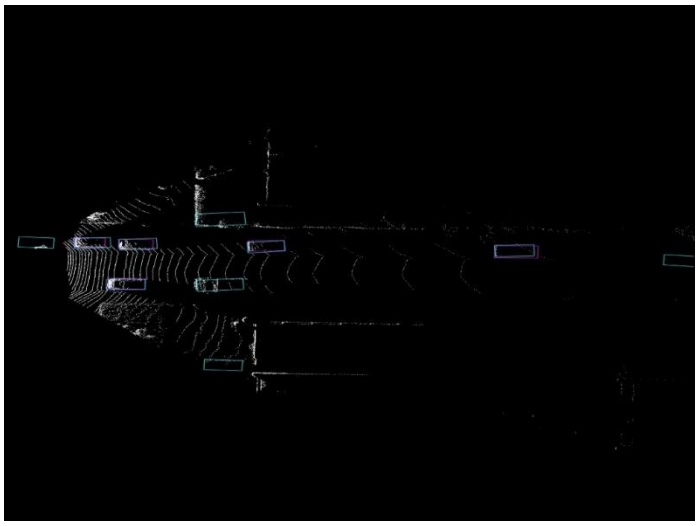
Σχήμα 5.8. *Precision Recall curve για 3D Detection.*



Σχήμα 5.9. *Precision Recall Curve για orientation similarity*.



Εικόνα 5.5. *Προβλέψεις σε πρόσθια όψη.*



Εικόνα 5.6. *Προβλέψεις σε κάτοψη.*

4. Μοντέλο 4-5

Στην συνέχεια δοκιμάζεται περαιτέρω αύξηση των εποχών και μείωση του ρυθμού μάθησης. Συγκεκριμένα δοκιμάζονται 40 εποχές και ρυθμός μάθησης 1e-05. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

M4	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP			3d AP		
	40	2	1e-05	1.5	1	def	400	400	def	def	17.53	14.85	12.07	1.68	1.56	1.37

Πίνακας 5.4

Τα αποτελέσματα του average precision οδηγούν στην υποψία overfittin και γίνεται δοκιμή κατασκευής μοντέλου early stopping στην συνέχεια.

5. Μοντέλο 5 . Χρήση early stopping μεταβλητής

Λαμβάνοντας υπ'όψιν την πιθανότητα το μοντέλο να μην μαθαίνει μετά τις 30 εποχές δοκιμάζεται ακολούθως η χρήση του Early Stoppping έτσι ώστε να συνεχίζεται η εκπαίδευση όσο η τιμή της συνάρτησης απωλειών στο σύνολο επικύρωσης μειώνεται και να σταματάει με την αύξηση. Ο ρυθμός μάθησης έχει και πάλι τιμή 0.0001. Οι εποχές τέθηκαν ίσες με 60.

Για το σκοπό αυτό χρησιμοποιούνται οι εξής μεταβλητές:

es_best: που δηλώνει την χαμηλότερη τιμή της συνάρτησης απωλειών, κάτω από την οποία το μοντέλο συνεχίζει να εκπαιδεύεται με την νέα πλέον χαμηλότερη τιμή σαν es_best ενώ αν το μοντέλο φτάσει σε αυτή την τιμή και δεν βελτιωθεί άλλο ύστερα από έναν αριθμό εποχών τότε η εκπαίδευση σταματά.

es_wait: χρησιμοποιείται για την μέτρηση των εποχών μετά από την 1η φορά που το μοντέλο φτάσει στην τιμή es_best.

es_patience: ο χρόνος που δίνεται στο μοντέλο ύστερα από την 1η φορά που φτάνει στο es_best για να βελτιωθεί. Όταν το es_wait >= es_patience και η τιμή των απωλειών δεν βελτιωθεί άλλο τότε σταματά η εκπαίδευση.

Σημαντικό είναι ότι το es_best το patience αλλά και το πότε θα ξεκινήσει να εφαρμόζεται το Early stopping (μετά από πόσες εποχές) ορίζεται από τον χρήστη και δοκιμάστηκαν αρκετές τιμές. Τελικά το όριο των εποχών που αποφασίστηκε ότι πρέπει να ολοκληρωθούν πριν ξεκινήσει το Early stopping αποφασίστηκε να είναι 30 και σαν es_best τέθηκε η τιμή 0.35 ενώ δόθηκε στο μοντέλο es_patience ίση με 5 εποχές.

Παρατίθεται ο σχετικός ψευδοκώδικας:

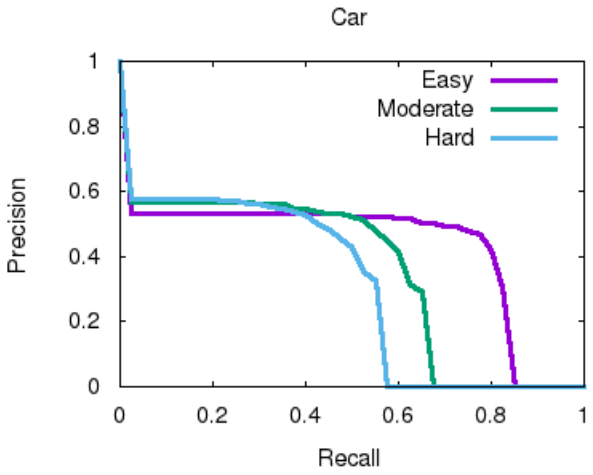
es_patience =5
es_wait = 0
es_best = 0.35

Όσο εποχές >= 30 :
Αύξησε es_wait κατά 1
Αν validation_loss < es_best:
es_best := validation_loss
es_wait := 0
Αν es_wait >= es_patience:
Σταμάτα και τερμάτισε το training

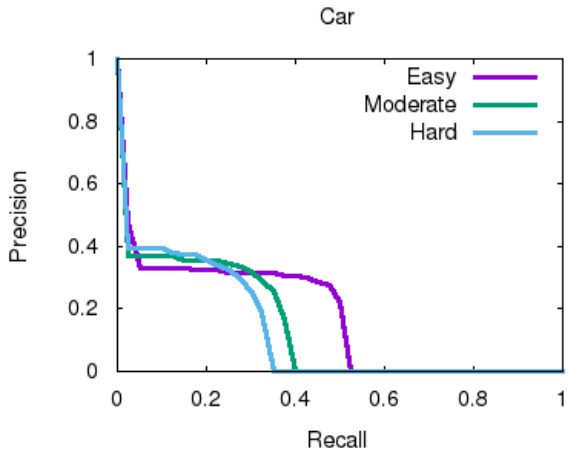
Παρατίθεται ο σχετικός πίνακας των παραμέτρων του μοντέλου.

M5	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP	3d AP
	60	2	0.0001	1.5	1	def	400	400	def	def	46.21 37.96 33.38	22.72 18.60 18.20

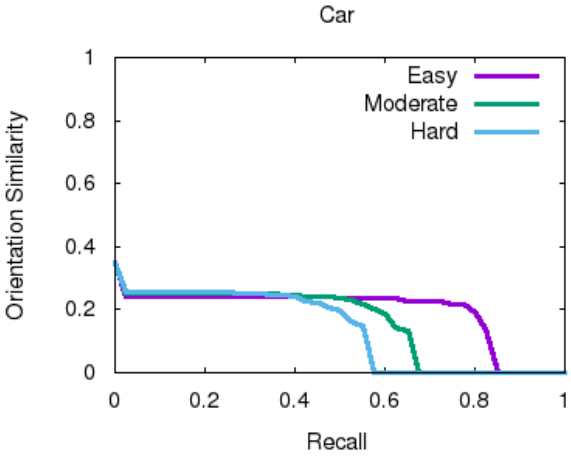
Πίνακας 5.5.



Σχήμα 5.10. *Precision Recall curve για Bird's eye view detection.*



Σχήμα 5.11. *Precision Recall curve για 3D detection.*



Σχήμα 5.12. *Precision Recall Curve για orientation similarity.*

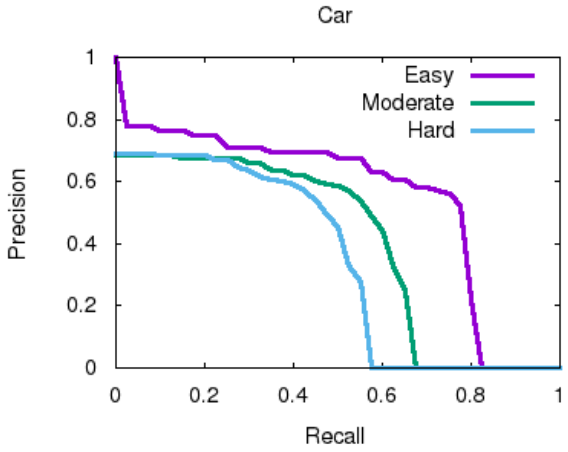
Τα predictions και η οπτικοποίησή τους είναι παρόμοια με το μοντέλο 3 και δεν παρατίθενται εδώ.

6. Μοντέλο 6

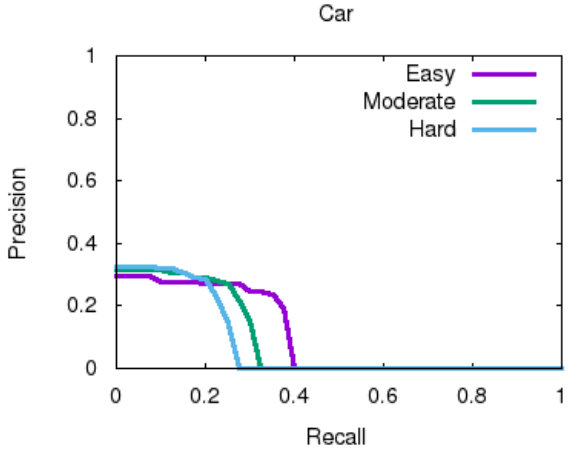
Στην συνέχεια υιοθετήθηκε το μοντέλο 5 στο οποίο επιπλέον τέθηκε εκθετική μείωση του ρυθμού μάθησης με αρχικό ρυθμό ίσο με 0.0001 και ρυθμό μείωσης ίσο με 0.73 ενώ τα βήματα τέθηκαν 100000.

M6	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP			3d AP		
	60	2	-	1.5	1	def	400	400	def	def	54.74	39.56	33.99	9.90	9.75	8.46

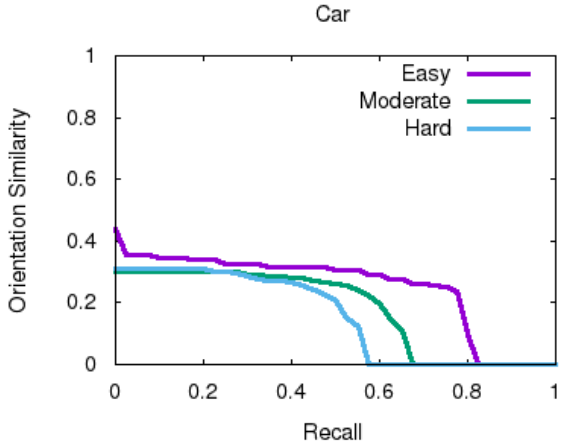
Πίνακας 5.6.



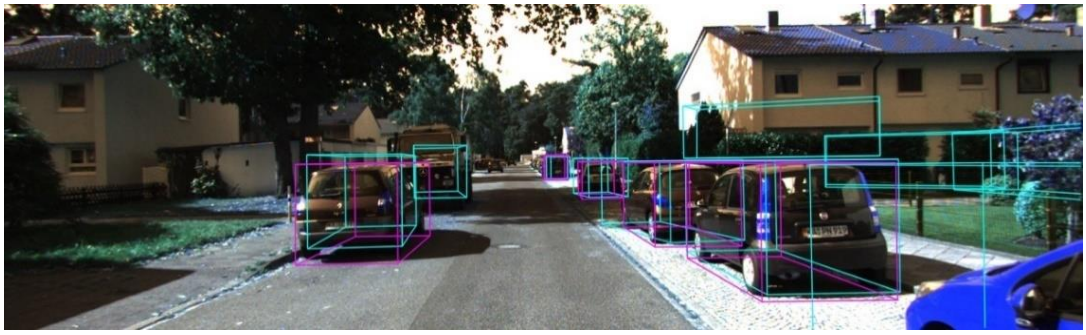
Σχήμα 5.13. *Precision Recall curve για Bird's eye view detection.*



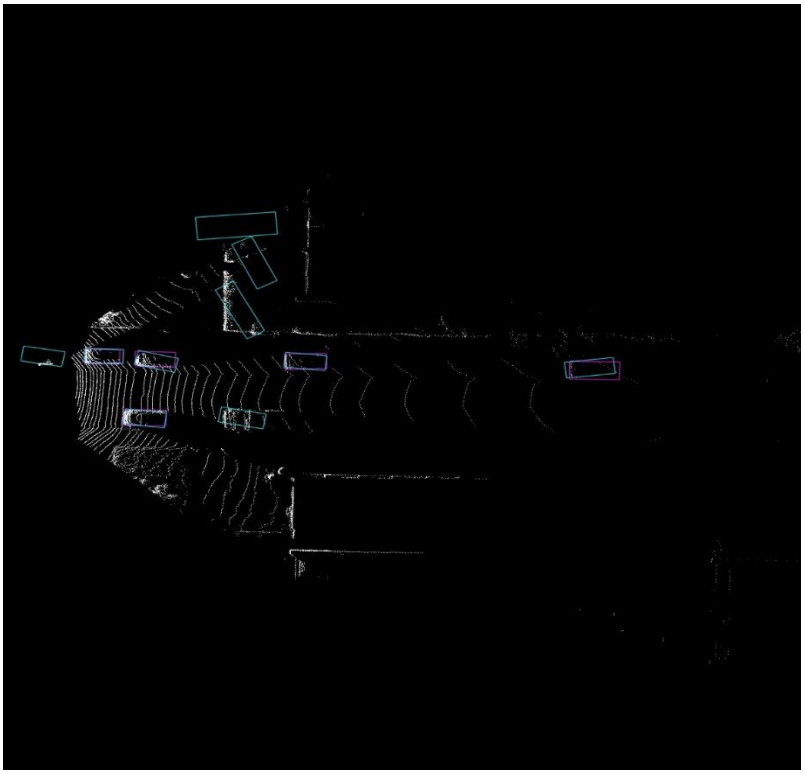
Σχήμα 5.14. *Precision Recall curve για 3D detection.*



Σχήμα 5.15. *Precision Recall Curve για orientation similarity.*



Εικόνα 5.7. *Προβλέψεις σε πρόσθια όψη.*



Εικόνα 5.8. *Προβλέψεις σε κάτοψη.*

Κρίνεται σκόπιμο στην επόμενη ενότητα να γίνει εκπαίδευση σε περισσότερα δεδομένα και με διαφορετικές τεχνικές επεξεργασίας – εμπλουτισμού του συνόλου εκπαίδευσης.

5.1.4 Σύνθετα Μοντέλα

7. Μοντέλο 7

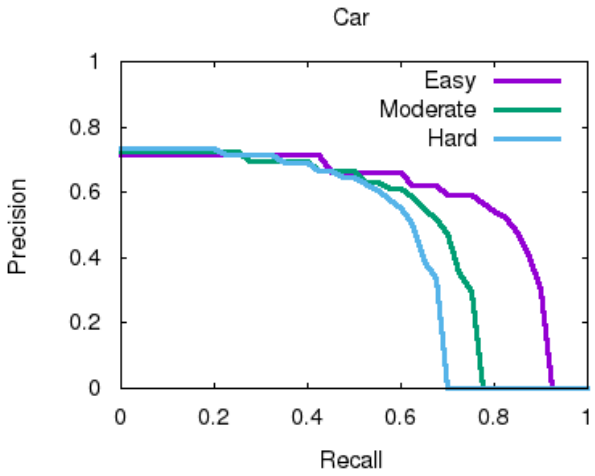
Για την κατασκευή ενός μοντέλου με μεγαλύτερη δυνατότητα γενίκευσης και για να αποφευχθούν φαινόμενα overfitting επιλέγεται στην συνέχεια σαν τροποποίηση των δεδομένων και δημιουργία συνθετικών δεδομένων να προστίθενται δύο μετασχηματισμοί στα Point clouds. Σε αντίθεση με τους αρχικούς μετασχηματισμούς του κεφαλαίου 4 (4.2.2) αφαιρείται εδώ ο μετασχηματισμός μεταφοράς και διατηρούνται μόνο οι μετασχηματισμοί περιστροφής με τροποποιημένο διάστημα περιστροφής [-π/2,π/2] και οι μετασχηματισμοί κλίμακας, η μεγέθυνση από ομοιόμορφη κατανομή όπως στο κεφάλαιο 4. Τα δεδομένα αυξάνονται σε 2000 στο σύνολο εκπαίδευσης και επαλήθευσης και ο ρυθμός μάθησης τίθεται ίσος με 0.0002. Ο Adam optimizer αποκτά πλέον beta1 = 0.9 και beta2 = 0.85 ενώ το e = 1e-06 στην προσπάθεια να αποφευχθεί το overshoot. Οι εποχές είναι 30. Ο ρυθμός μάθησης μειώνεται με σύμφωνα με το εξής σχήμα:

$$lr = 0.0002, \text{εποχές} \in [0,5]\ lr = lr \times 0.1, \text{εποχές} \in (5,10]\ lr = lr \times 0.01, \text{εποχές} \in (10,12]\ lr = lr \times 0.01, \text{εποχές} \in (12,30]$$

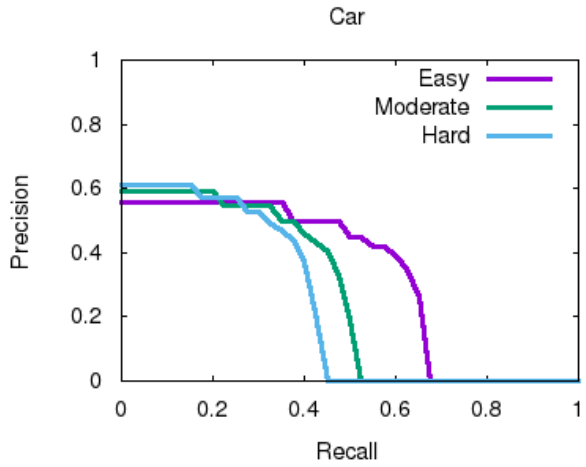
Επίσης γίνεται εναλλαγή augmentation (τροποποίησης) και μη των δεδομένων για την δημιουργία νέων δεδομένων για εκπαίδευση. Στις πρώτες 5 εποχές δεν γίνονται μετασχηματισμοί και χρησιμοποιούνται τα αρχικά δεδομένα ενώ στην εποχή 6 ενεργοποιείται το augmentation των δεδομένων για να εκτιμηθεί η δυνατότητα του μοντέλου να μαθαίνει από clouds που έχουν περιστραφεί ή έχουν αλλάξει κλίμακα. Στις επόμενες εποχές απενεργοποιείται πάλι το augmentation. Το early stopping ενεργοποιήθηκε στην εποχή 7.

M7	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP	3d AP
	30	2	-	1.5	1	-	2000	2000	def	def	57.51 48.19 43.72	32.48 26.99 24.43

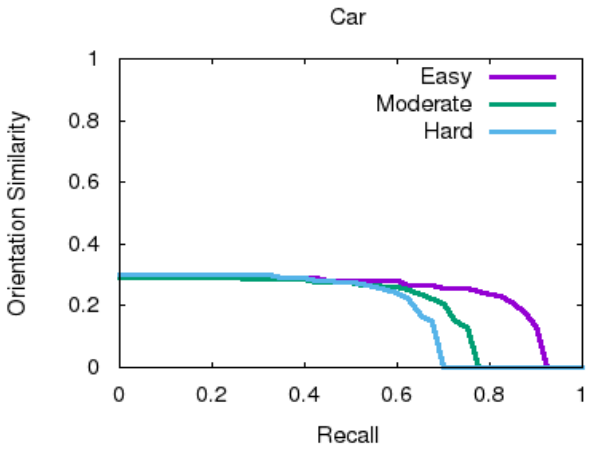
Πίνακας 5.7.



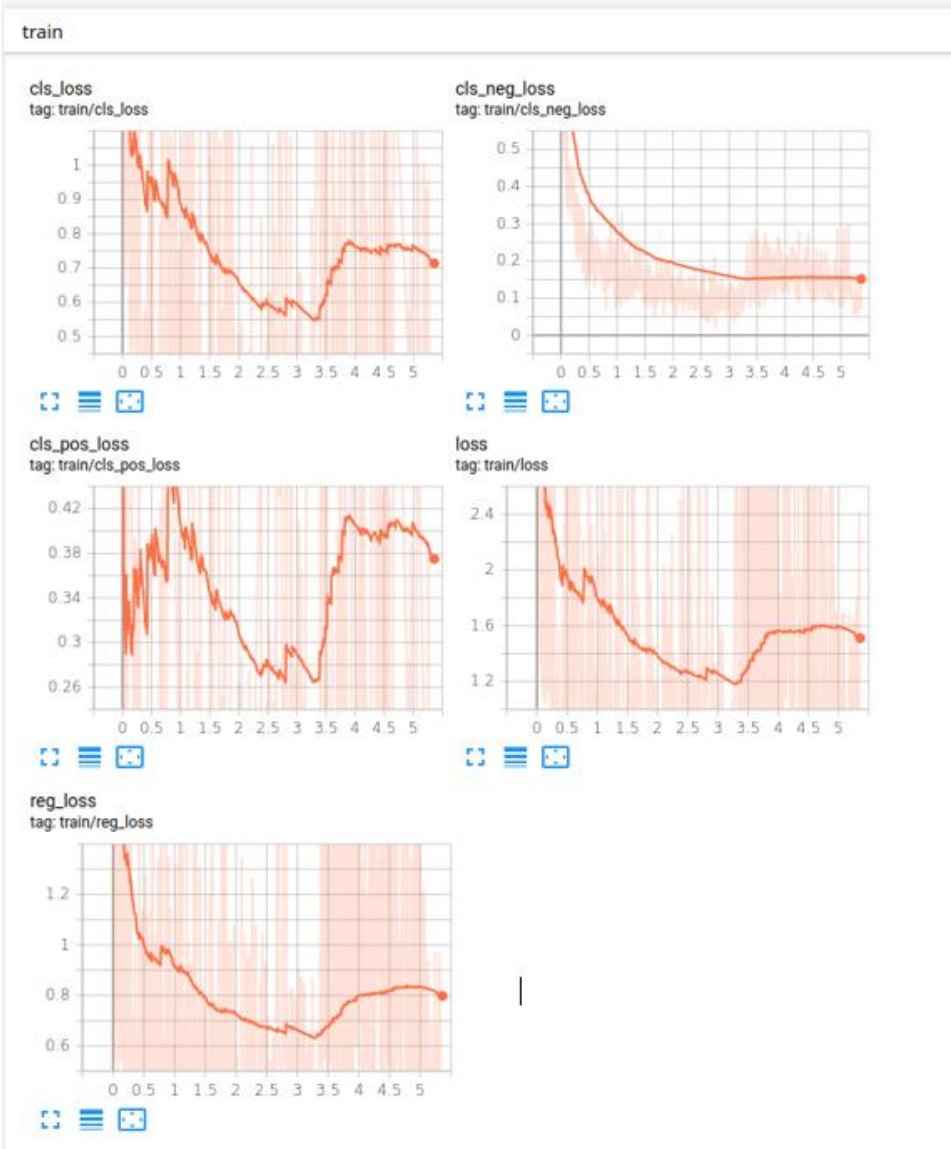
Σχήμα 5.16. *Precision Recall curve για Bird's eye view detection.*



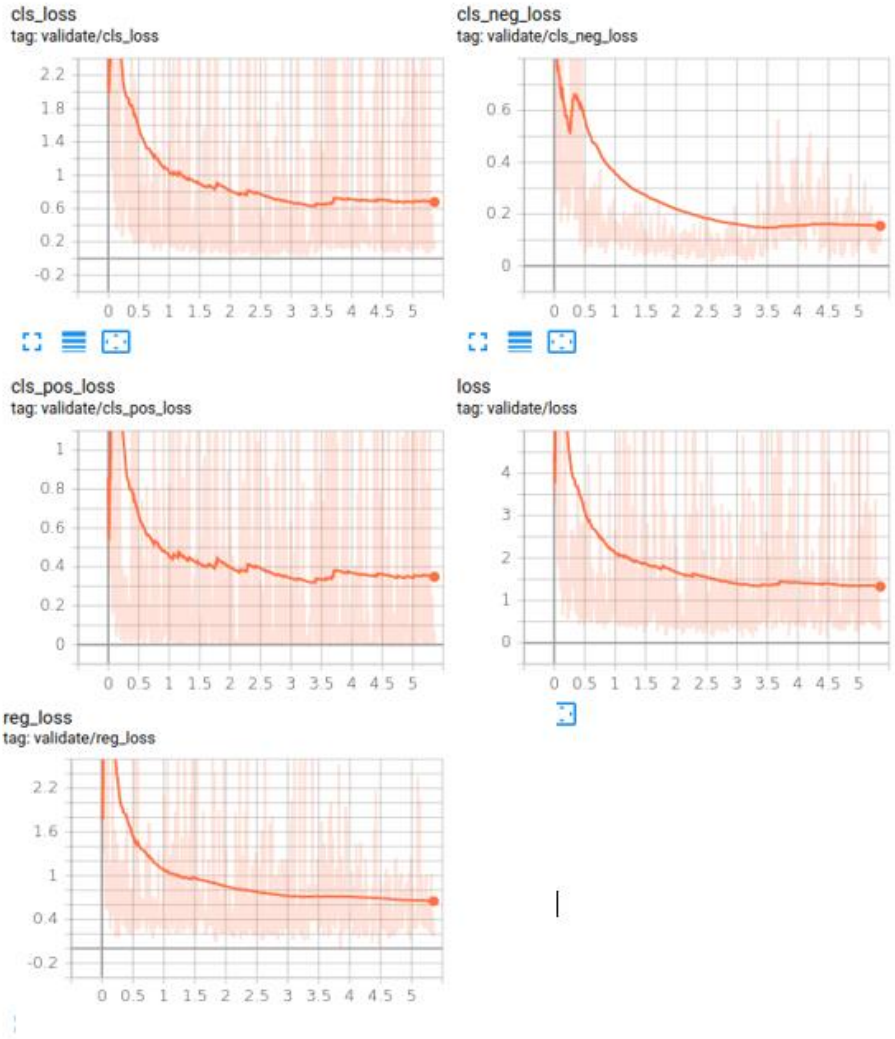
Σχήμα 5.17. *Precision Recall curve για 3d detection.*



Σχήμα 5.18. *Precision Recall Curve για orientation similarity.*



Σχήμα 5.19. *Γραφικές απωλειών στο σύνολο εκπαίδευσης.*



Σχήμα 5.20. *Γραφικές απωλειών στο σύνολο επαλήθευσης.*

Τα predictions και η οπτικοποίηση τους είναι παρόμοια με το μοντέλο 7 και δεν παρατίθενται εδώ.

8. Μοντέλο 8

Ακολουθως υιοθετήθηκε η τροποποποίηση των μετασχηματισμών ώστε να περιορίζονται σε περιστροφή και αλλαγή κλίμακας των clouds. Η περιστροφή είναι στο διάστημα $[-\pi/4, \pi/4]$. Ορίστηκε μέχρι την εποχή 15 το μοντέλο να εκπαιδεύεται στα υπάρχοντα δεδομένα και στην συνέχεια να ενεργοποιούνται οι μετασχηματισμοί για την δημιουργία ποικιλίας στο σύνολο εκπαίδευσης. Ταυτόχρονα τα δεδομένα περιορίστηκαν σε 1000 για εκπαίδευση και 1000 για επαλήθευση εφ'όσον κρίνεται πως η εκπαίδευση καθυστερούσε με περισσότερα δεδομένα. Ο ρυθμός μάθησης τέθηκε στο 0.0002 και οι εποχές ορίστηκαν 40, χωρίς early stopping. Επιπλέον αλλαγές έγιναν στο RPN μοντέλο ώστε να παράγει συνολικά το πολύ 12 προβλέψεις και όχι 20 που ήταν οι ορισμένες στο μοντέλο του κεφαλαίου 4. Επίσης σαν όριο IoU μεταξύ παραγόμενων κυτίων (predicted bounding boxes) που έχουν ήδη επιλεγεί αποφασίζεται το κατώφλι 0.1 σύμφωνα με το μοντέλο του κεφαλαίου 4. Αυτό σημαίνει ότι τα κυτία που έχουν $IoU \geq 0.1$ με κυτία ήδη επιλεγμένα από το μοντέλο απορρίπτονται από το non max suppression. Η μέθοδος υλοποιείται με το API του Tensorflow και παρατίθεται το σχετικό απόσπασμα από το Tensorflow API καθώς και η τροποποίησή του.

```
tf.image.non_max_suppression
(
        boxes,
        scores,
        max_output_size,
        iou_threshold=0.5,
        score_threshold=float('inf'),
        name=None
)
```

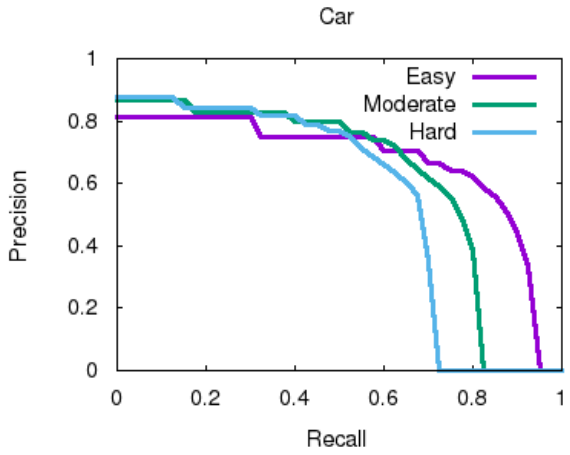
```
__C.RPN_NMS_POST_TOPK=12
__C.RPN_NMS_THRESH=0.1

tf.image.non_max_suppression
(
        boxes,
        scores,
        max_output_size=__C.RPN_NMS_POST_TOPK,
        iou_threshold=__C.RPN_NMS_THRESH,
        score_threshold=float('inf'),
        name=None
)
```

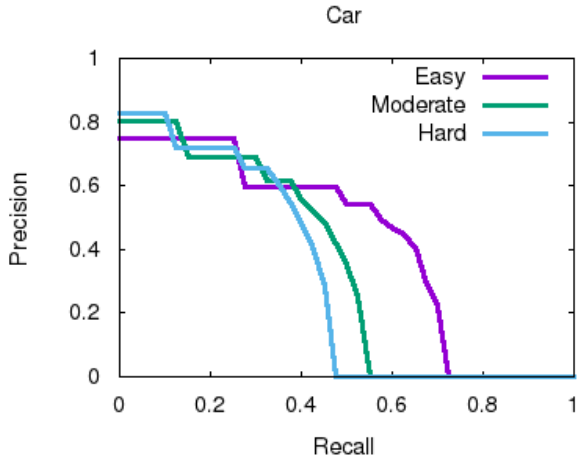
Εικόνα 5.9. *Η συνάρτηση non max suppression από το API του Tensorflow (αριστερά) και η τροποποιημένη συνάρτηση αλλάζοντας το IoU threshold και το max_output_size (δεξιά).*

	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP	3d AP
M8	40	2	0.0002	1.5	1	-	1000	1000	def	def	65.31 61.07 54.82	42.47 35.40 31.82

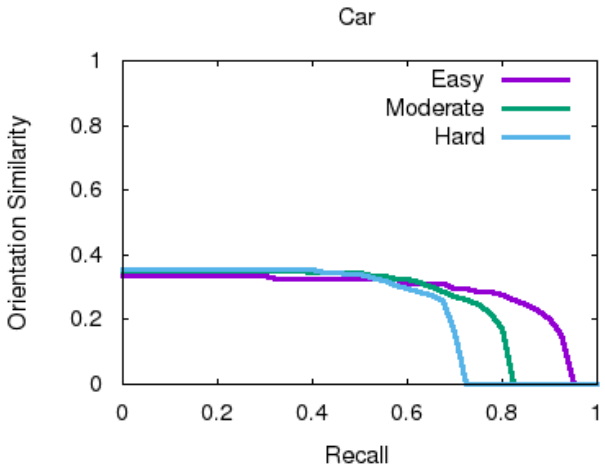
Πίνακας 5.8



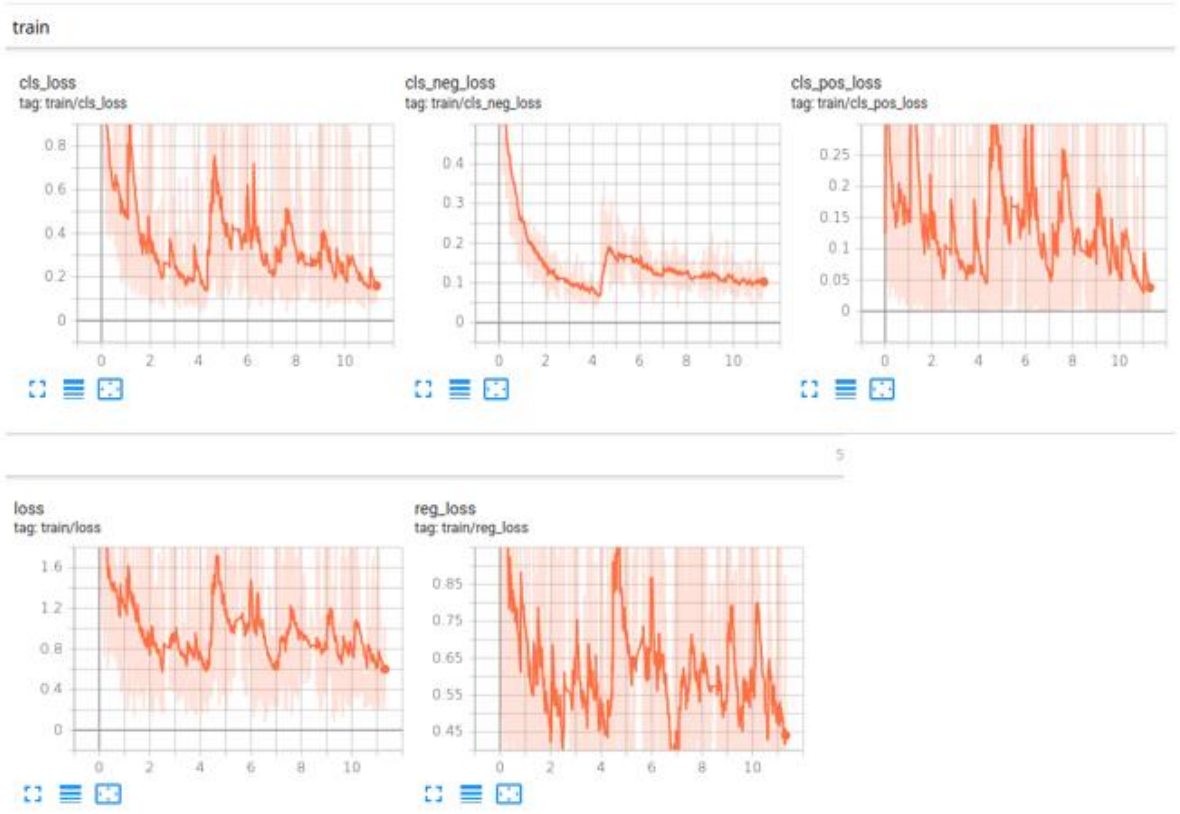
Σχήμα 5.21. *Precision Recall curve για Bird's eye view detection.*



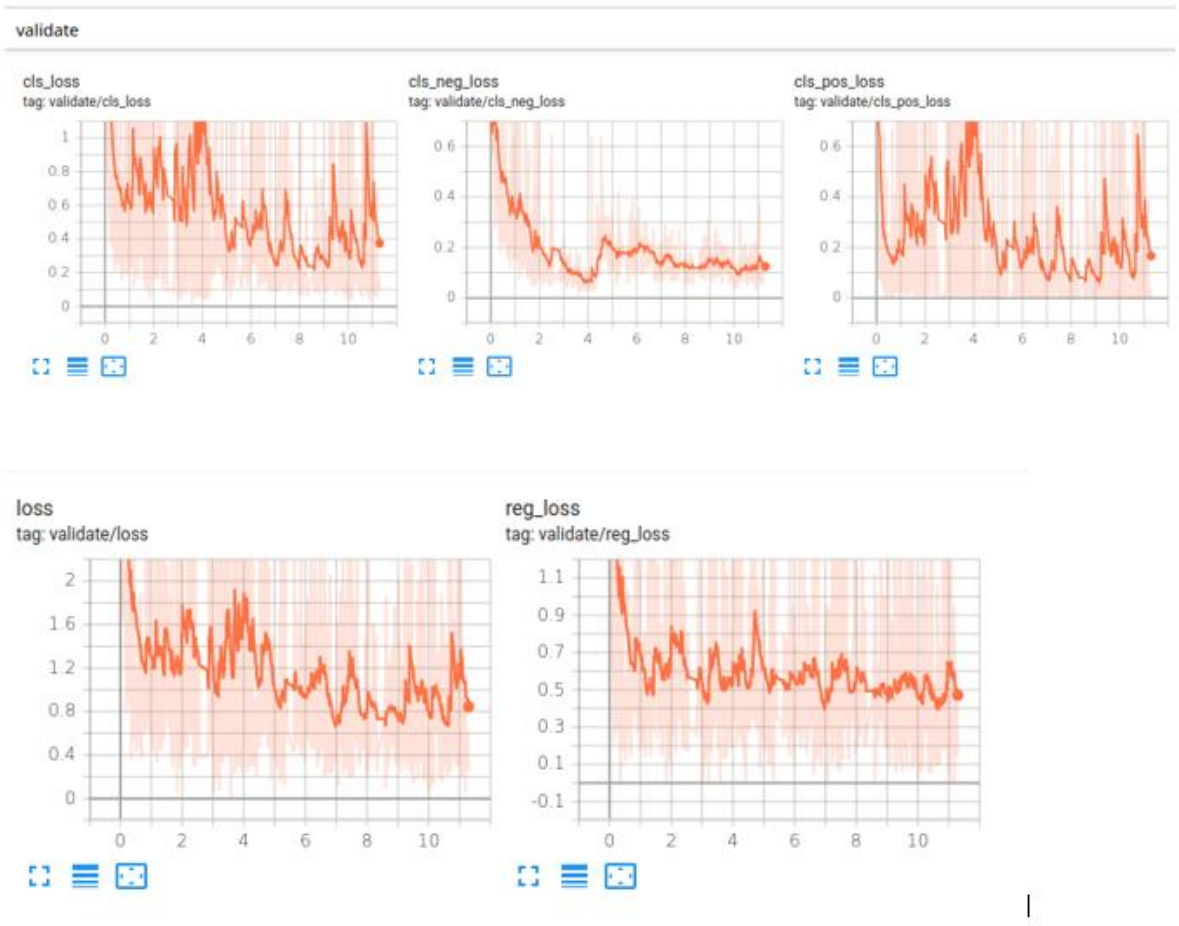
Σχήμα 5.22. *Precision Recall curve για 3D detection.*



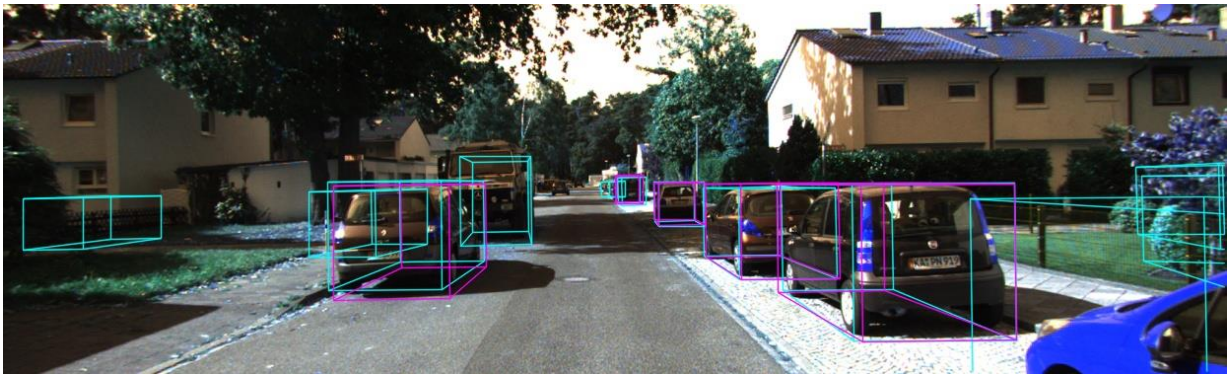
Σχήμα 5.23. *Precision Recall Curve για orientation similarity.*



Σχήμα 5.24 Γραφικές απωλειών στο σύνολο εκπαίδευσης.



Σχήμα 5.25. *Γραφικές απωλειών στο σύνολο επαλήθευσης.*



Εικόνα 5.10. Προβλέψεις σε πρόσθια όψη

Εικόνα 5.11. *Προβλέψεις σε κάτοψη*

9. Μοντέλο 9

Σαν τελευταίο μοντέλο για σύγκριση παρατίθεται ήδη εκπαιδευμένο μοντέλο για τρισδιάστατη αναγνώριση αντικειμένων όπου η εκπαίδευση έχει γίνει με διαχωρισμό των clouds σε 3712 clouds στο σύνολο εκπαίδευσης και 3769 clouds στο σύνολο επαλήθευσης. Χρησιμοποιείται ρυθμός μάθησης με μείωση σύμφωνα με το σχήμα $lr = lr \times 0.01, \text{εποχές} \in (0,80]$, $lr = lr \times 0.1, \text{εποχές} \in (80, 120]$, $lr = lr \times 0.01, \text{εποχές} > 120$. Το μοντέλο RPN δίνει σαν μέγιστο αριθμό predictions (κυτία αναγνώρισης) 20 και το κατώφλι IoU για απόρριψη κυτίων είναι 0.1.

M9	E	BS	lr	a	b	aug	Tr	Val	Conv	VFE	BEV AP			3d AP		
	120	2	0.001	1.0	10.0	def	3712	3769	def	def	70.32	66.82	64.63	43.05	38.5	36.31

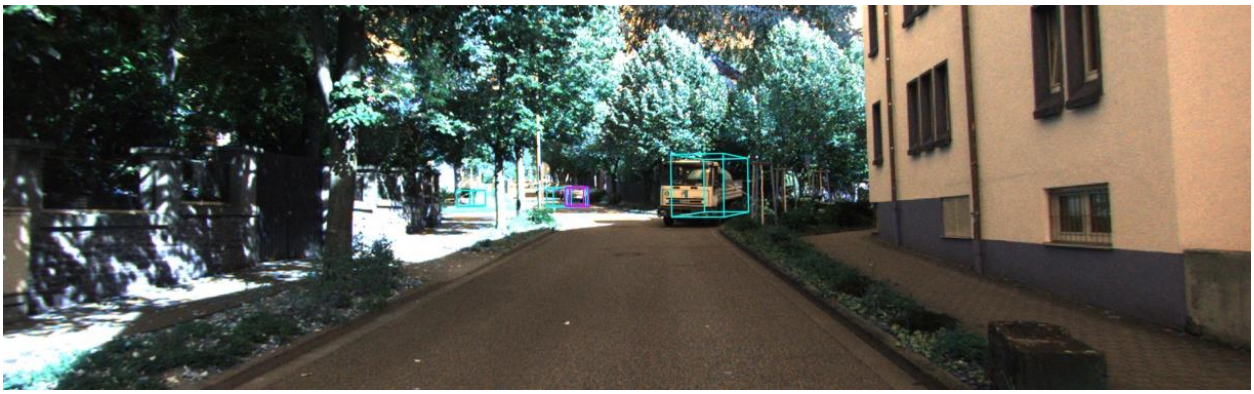
Πίνακας 5.9.

5.1.5 Σύνθετες τεχνικές Διαχωρισμού των Δεδομένων

Όπως αναφέρθηκε στο κεφάλαιο 4 υπάρχει ανισορροπία κλάσεων και τα αυτοκίνητα ξεπερνούν τις υπόλοιπες κλάσεις με αποτέλεσμα το εκάστοτε μοντέλο να δυσκολεύεται στην αναγνώριση και απόρριψη ειδικά κλάσεων παρόμοιων με τα «Car», όπως «Truck», «Van» αλλά ακόμη και των κατηγοριών «Misc», ειδικά όταν έχουν παρόμοιο μέγεθος και σχήμα με αυτοκίνητα. Η κλάση «Pedestrian» από την άλλη δεν αποτελεί σημαντικό πρόβλημα για το μοντέλο εφόσον ο χωρισμός των voxels σε ειδικό μέγεθος και το μέγεθος του anchor, που είναι ειδικά διαμορφωμένα για την αναγνώριση των αυτοκινήτων, σε συνδυασμό με την εκπαίδευση του RPN φαίνεται πως δίνουν στο μοντέλο την δυνατότητα να διακρίνει τις κλάσεις αυτές.

Εικόνα 5.12. *Το μοντέλο μπορεί να διακρίνει ότι στο στιγμιότυπο δεν υπάρχει αυτοκίνητο.*

Εικόνα 5.13. *Το μοντέλο αναγνωρίζει την στάση στο τραμ και το σχήμα στα δεξιά σαν αυτοκίνητα.*



Εικόνα 5.14. *Το μοντέλο αναγνωρίζει το φορτηγό ως αυτοκίνητο.*

Για τους παραπάνω λόγους έγινε προσπάθεια στην παρούσα μελέτη να γίνει πιο έξυπνος διαχωρισμός των δεδομένων σε δεδομένα εκπαίδευσης σύμφωνα και με το [81]. Δοκιμάστηκε λοιπόν ο χωρισμός των clouds σε:

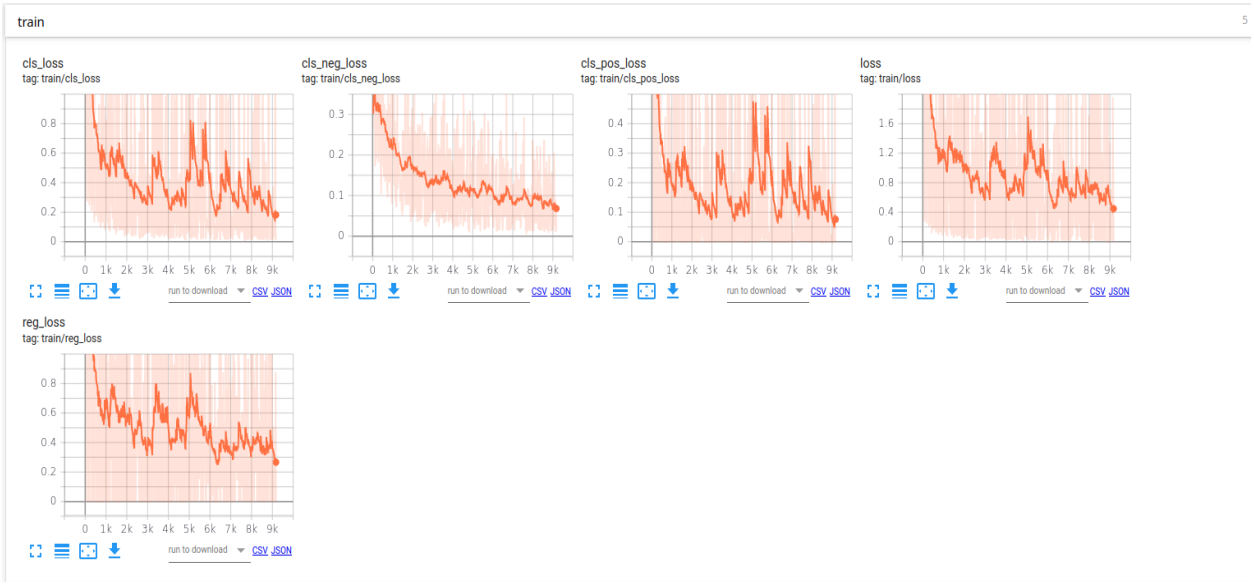
- 1. clouds με «Car»στιγμιότυπα και οποιαδήποτε άλλη κλάση: 5991 samples
- 2. clouds με «Car» μόνο: 3488 samples
- 3. clouds με «Car», «Truck» , «Van»: 1982 samples
- 4. clouds με «Car» , «Cyclists»: 1127 samples
- 5. clouds με «Pedestrian» μόνο: 418 samples
- 6. clouds με «Car», «Pedestrian», «Cyclist»: 1305 samples
- 7. clouds με «Car», «Pedestrian» χωρίς «Cyclist»: 780 samples

Στις παραπάνω διαιρέσεις τα στιγμιότυπα με επιπλέον κλάσεις «Misc», «Van», «Truck», «Tram», «Don’t Care» δεν απομονώθηκαν παραπάνω λόγω της πολύ μικρής συχνότητας εμφάνισης τους.

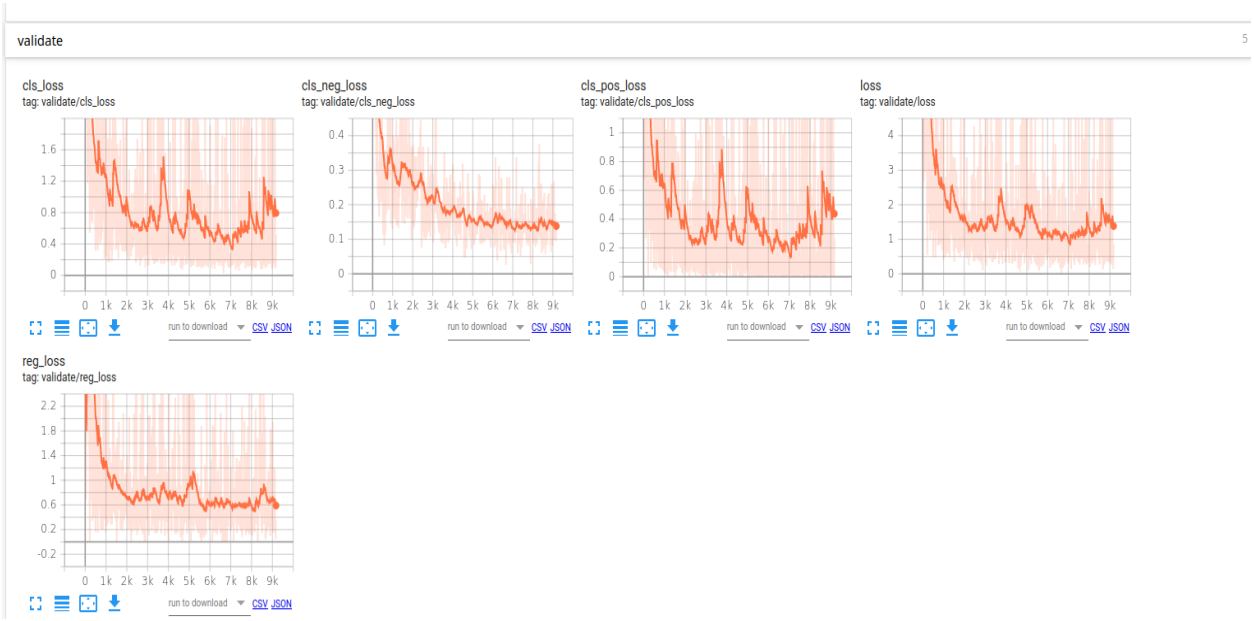
Στην συνέχεια δοκιμάστηκε το μοντέλο (FeatureNet, RPN) να εκπαιδευτεί αρχικά σε δεδομένα από το dataset 2) ή και το 3) και στην συνέχεια μόλις συγκλίνει να προστεθούν δεδομένα από dataset όπως το 5) και το 7) ώστε να γίνει προσπάθεια το μοντέλο να μάθει να διακρίνει τις διαφορετικές κλάσεις. Συνολικά αρχικά δεδομένα και augmented δεδομένα έφτασαν τα 3718 δεδομένα εκπαίδευσης και 2000 δεδομένα επαλήθευσης ενώ οι εποχές ορίστηκαν ίσες με 20. Η εκπαίδευση σταμάτησε στην 5η εποχή καθώς το μοντέλο συνέκλινε ήδη σε ελάχιστο και σταμάτησε η μάθηση. Ο ρυθμός μάθησης τέθηκε στο 0.0002 για τις 5 εποχές ενώ ακολουθήθηκε

το σχήμα $lr = 0.0002$, εποχές $\in (0,5]$, $lr = lr \times 0.1$, εποχές $\in (5,10]$, $lr = lr \times 0.01$, εποχές $\in (10,15]$, $l\,r = lr \times 0.1$, εποχές $\in (15,20]$.

Οι παραπάνω απόπειρες δεν οδήγησαν σε βελτίωση της ακρίβειας καθώς φαίνεται πως η προσθήκη clouds με απομονωμένες κλάσεις δυσκολεύει την εκμάθηση του μοντέλου ενώ η εκπαίδευση είναι αργή και οι απώλειες στην ανίχνευση της ύπαρξης ή όχι αυτοκινήτου δεν φτάνουν στα ολικά ελάχιστα.



Σχήμα 5.19. *Γραφικές απωλειών στο σύνολο εκπαίδευσης.*



Σχήμα 5.20. *Γραφικές απωλειών στο σύνολο επαλήθευσης.*

5.2 Συμπεράσματα

5.2.1 Μοντέλα ενότητας 5.1.3.

Μοντέλο 1.

Παρατηρούμε στις γραφικές precision recall curve του μοντέλου 1 ότι και το precision και το recall είναι πολύ χαμηλά και στις 2 διαστάσεις και στις 3 διαστάσεις. Τα αποτελέσματα για το precision είναι επίσης υπερβολικά χαμηλά. Ο ρυθμός μάθησης 0.01 κρίνεται πολύ υψηλός για την ικανοποιητική μάθηση του μοντέλου και δεν επιτρέπει στο μοντέλο να εντοπίσει τα ελάχιστα της συνάρτησης απωλειών με αποτέλεσμα πολλά ψευδώς θετικά αυτοκίνητα αλλά και μη ικανοποιητική διόρθωση των κυτίων αναγνώρισης στην περίπτωση σωστής αναγνώρισης όπως φαίνεται και από τις 2 εικόνες με τις προβλέψεις.

Μοντέλο 2.

Παρατηρείται βελτίωση της ακρίβειας με την αλλαγή του ρυθμού εκμάθησης σε χαμηλότερες τιμές. Ενώ τα ψευδώς θετικά αυτοκίνητα πάλι είναι αρκετά, στα σωστά αναγνωρισμένα αυτοκίνητα η διόρθωση των κυτίων αναγνώρισης είναι τώρα πολύ υψηλότερης ποιότητας. Τα precison recall curve δείχνουν ότι το μεγαλύτερο πρόβλημα είναι τα FP καθώς το precision παραμένει αρκετά χαμηλό σε όλες τις κλάσεις δυσκολίας. Μπορεί να μειωθεί το IoU threshold για να αυξηθεί κι άλλο το recall στις κλάσεις moderate και hard αλλά το χαμηλό precision θα συνεχίζει να αποτελεί πρόβλημα και η μείωση του threshold θα την μειώσει παραπάνω.

Μοντέλο 3.

Παρατηρείται σαφής βελτίωση της ακρίβειας συνεπώς η αύξηση των εποχών και ο χαμηλότερος ρυθμός εκμάθησης είναι καθοριστικής σημασίας για την εκμάθηση των χαρακτηριστικών των clouds.Τα precision και recall είναι αρκετά κοντά και έχουν υψηλότερες τιμές με κατάλληλη επιλογή IoU threshold. Παρ'όλα αυτά στις πιο δύσκολες κλάσης (moderate, hard) και το precision και το recall είναι κοντά στο 0.5-0.6 στην δισδιάστατη αναγνώριση. Στην τρισδιάστατη αναγνώριση το precision είναι πάλι αρκετά χαμηλό σε όλες τις κλάσεις δυσκολίας.

Μοντέλο 4.

Όπως φαίνεται περαιτέρω αύξηση των εποχών και μείωση του ρυθμού μάθησης δεν οδηγεί σε βελτίωση των αποτελεσμάτων τουλάχιστον για τον συγκεκριμένο αριθμό συνόλου εκπαίδευσης και επαλήθευσης. Μετά τις 30 εποχές πιθανότατα στα 400 δεδομένα το μοντέλο οδηγείται σε overfitting και η μάθηση διακόπτεται.

Μοντέλο 5.

Όπως παρατηρείται στον πίνακα, με την προσθήκη του Early stopping βελτιώθηκε η ακρίβεια στην τρισδιάστατη αναγνώριση. Το recall στην αναγνώριση 2 διαστάσεων μπορεί να φτάσει σε υψηλότερες τιμές (0.82) στις πιο εύκολες κλάσεις(easy) ενώ στην moderate δυσκολία προσεγγίζει το 0.67.Σ την πιο δύσκολη κατηγορία hard φτάνει μέχρι το 0.6. Το max precision είναι κοντά στο 0.6. Στην τρισδιάστατη αναγνώριση το precision είναι πιο χαμηλό πράγμα που εξηγεί και την εμφάνιση FP.

Μοντέλο 6.

Δεν παρατηρείται βελτίωση της ακρίβειας και συνεπώς η αύξηση του patience οδηγεί το μοντέλο στο να υπερκεράσει το ελάχιστο της συνάρτησης απωλειών καταλήγοντας σε ελάχιστο που δεν είναι βέλτιστο.

Σε όλα τα παραπάνω μοντέλα παρατηρείται πολύ χαμηλό orientation precision (0.2-0.4) ενώ το orientation recall μπορεί να φτάσει σε υψηλότερες τιμές (0.8). Αυτό σημαίνει ότι υπάρχουν πολλά FP ενώ με μείωση του threshold είναι δυνατό να επιτυγχάνονται περισσότεροι σωστοί προσανατολισμοί με κόστος όμως στο precision.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Από τα παραπάνω μοντέλα παρατηρείται ότι με 400 δεδομένα εκπαίδευσης και 400 δεδομένα επαλήθευσης η εκπαίδευση πάνω από ένα κατώφλι εποχών δεν βελτιώνει την ακρίβεια του μοντέλου. Επίσης, ο ρυθμός μάθησης κοντά στο 0.0001 φαίνεται πως αποτελεί έναν ικανοποιητικό ρυθμό μάθησης καθώς μεγαλύτεροι ρυθμοί μάθησης οδηγούν το μοντέλο στο να υπερκεράσει μικρότερα και επιθυμητά ελάχιστα (overshoot). Το early stopping με μικρό es_patience φαίνεται να αποτελεί αξιόλογο εργαλείο για την αποφυγή του overshoot. Οι μεγαλύτερες προκλήσεις του μοντέλου φαίνεται πως είναι η απόρριψη των ψευδώς θετικών και η αναγνώριση σε 3 διαστάσεις.

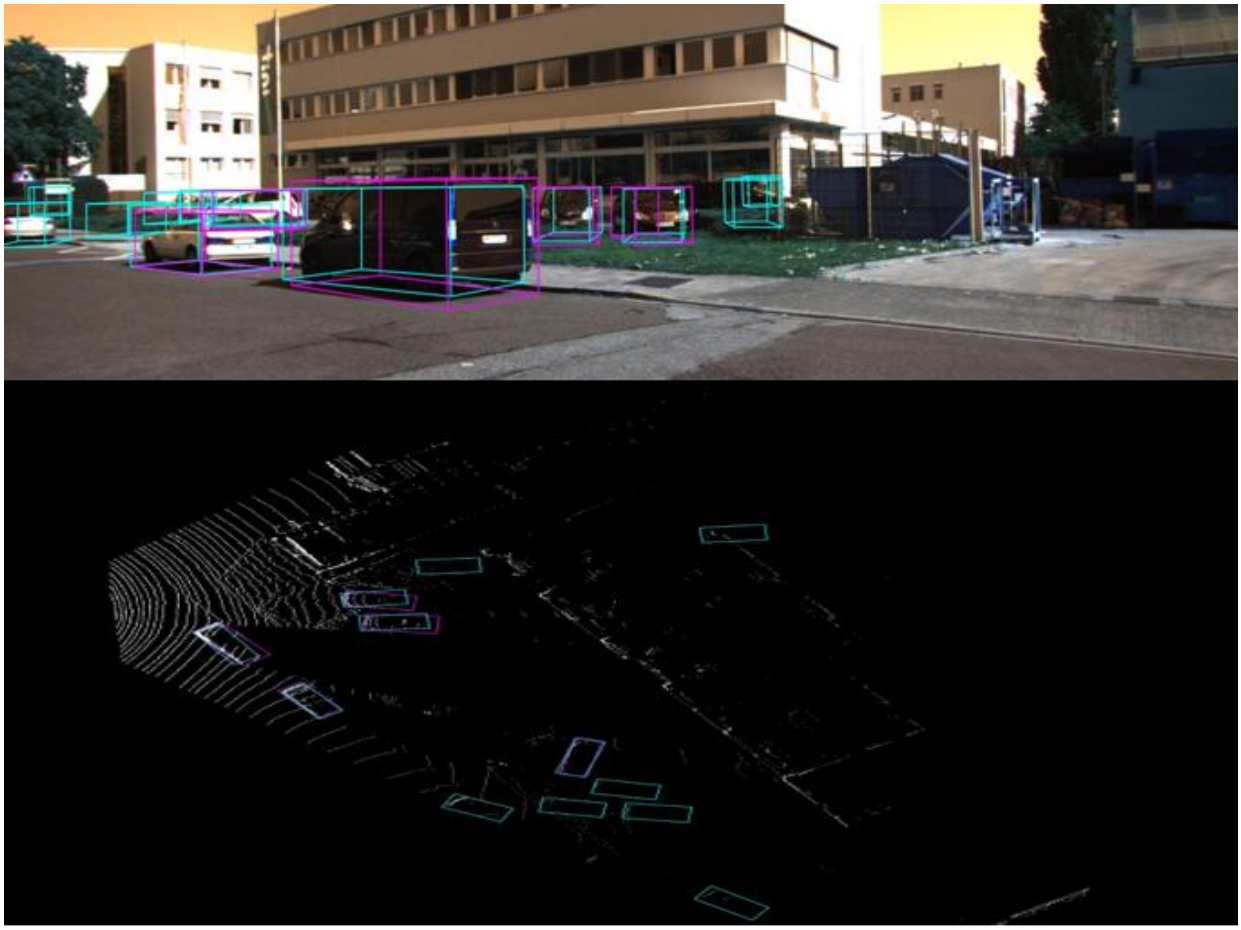
5.2.2 Μοντέλα ενότητας 5.1.4.

Μοντέλο 7.

Ο συνδυασμός λιγότερων μετασχηματισμών επιταχύνει την σύγκλιση ενώ βελτιώνεται και η ακρίβεια της αναγνώρισης στις 3 διαστάσεις. Παρατηρείται αύξηση του precision και του recall στις 2 διαστάσεις. Το precision φαίνεται να φτάνει στο 0.78-0.79 και για τις 3 κλάσεις δυσκολίας ενώ το recall έχει εκτείνεται από το 0.7-0.95 για τις 3 κλάσεις δυσκολίας. Το πρόβλημα συνεχίζει να είναι το precision εφόσον η αύξηση του recall είναι δυνατή αν μειωθεί το IoU threshold αλλά αυτό μπορεί αυξήσει τα FP λόγω του μεγάλου class imbalance. Παρατηρώντας τις γραφικές απωλειών είναι σαφές ότι το early stopping είναι απαραίτητο για να μην γίνεται overshoot των βέλτιστων ελαχίστων. Συγκεκριμένα από το training dataset προκύπτει ότι το μοντέλο δεν μπορεί να μάθει πέρα από τα 3.5K βήματα, όπου και εμφανίζεται ένα ελάχιστο. Η απώλεια loss είναι βέβαια πάλι υψηλή 1.2. Στο σύνολο επαλήθευσης το regression loss κρατάει το συνολικό loss ψηλά και προκύπτει ότι τα προβλεπόμενα κυτία αναγνώρισης δεν έχουν το βέλτιστο overlap με τα πραγματικά κυτία γεγονός που δημιουργεί και υψηλό Orientation Loss (AOS).

Μοντέλο 8.

Παρατηρείται βελτίωση της ακρίβειας και στους 2 τύπους αναγνώρισης με το μοντέλο να σημειώνει πρόοδο στην αναγνώριση των αυτοκινήτων με διαφορετικές γωνίες και προσανατολισμούς όπως φαίνεται από το στιγμιότυπο 2 του αντίστοιχου μοντέλου. Παρατηρείται ακόμη μεγαλύτερη αύξηση του precision και του recall και στις 2 και στις 3 διαστάσεις ενώ φαίνεται να βελτιώνεται και το average orientation similarity. Στις γραφικές απωλειών παρατηρείται ότι και πάλι εμφανίζεται ένα ελάχιστο και μετά από ένα σημείο το μοντέλο συγκλίνει.



Εικόνα 5.15. *Το μοντέλο παρουσιάζει καλύτερες επιδόσεις όσον αφορά τον εντοπισμό αυτοκινήτων με στραμμένο προσανατολισμό.*

Το μοντέλο 8 προσεγγίζει το προεκπαιδευμένο μοντέλο το οποίο παρουσιάζει αύξηση 3-5% στο precision. Σημαντική παρατήρηση είναι ότι ακόμα η τρισδιάστατη αναγνώριση παρουσιάζει χαμηλότερη ακρίβεια (43% easy). Τα μοντέλα 7-9 παρουσιάζουν και αυτά πρόβλημα στο precision όσον αφορά το orientation similarity ενώ το recall μπορεί να φτάνει σε αρκετά υψηλές τιμές (0.7-0.9+).

5.2.3 Σχολιασμός των τεχνικών Διαχωρισμού δεδομένων

Οι τεχνικές που χρησιμοποιήθηκαν για την αντιμετώπιση της ανισορροπίας κλάσεων είναι subsampling των στιγμιότυπων με αυτοκίνητα και duplication και augmentation των δεδομενων με άλλες κλάσεις οι οποίες παρουσιάζουν ομοιότητες με το σχήμα των αυτοκινήτων όπως «Trucks», «Vans» [81]. Επίσης γίνεται εμπλουτισμός με κλάσεις όπως «Cyclist» και «Pedestrian». Οι παραπάνω απόπειρες δεν οδήγησαν σε βελτίωση της ακρίβειας καθώς φαίνεται πως η προσθήκη clouds με απομονωμένες κλάσεις δυσκολεύει την εκμάθηση του μοντέλου ενώ η εκπαίδευση είναι αργή και οι απώλειες στην ανίχνευση της ύπαρξης ή όχι αυτοκινήτου δεν φτάνουν σε ελάχιστα τα οποία δεν είναι αρκούντως χαμηλά όπως φαίνεται και από τις γραφικές απωλειών.

5.3 Περαιτέρω δοκιμές και Μελλοντική Εργασία

5.3.1 Τροποποίηση VFE layers

Με στόχο την εκτίμηση του ρόλου που διαδραματίζει η αλυσίδα των VFE επιπέδων για την εκμάθηση των χαρακτηριστικών του κάθε cloud επιχειρήθηκε να τροποποιηθούν τα υπάρχοντα VFE layers του VFE δικτύου. Η προτεινόμενη αρχιτεκτονική [82], Voxelnet ορίζει 2 επίπεδα VFE:

```
self.vfe1 = VFE Layer (32, 'VFE-1')
self.vfe2 = VFE Layer (128, 'VFE-2')
```

Στην συνέχεια προστέθηκαν άλλα 2 ενδιάμεσα επίπεδα VFE με στόχο την πιο αργή αύξηση των διαστάσεων του Feature πριν την είσοδο στα ενδιάμεσα συνελικτικά επίπεδα (Convolutional Middle layers):

```
self.vfe1 = VFELayer(16, 'VFE-1')
self.vfe2 = VFELayer(32, 'VFE-2')
self.vfe3 = VFELayer(64, 'VFE-3')
self.vfe4 = VFE Layer (128, 'VFE-4')
```

όπου τα παραπάνω επίπεδα είναι διαδοχικά. Η παραπάνω αλλαγή δεν επηρέασε την απόδοση του μοντέλου με αποτέλεσμα να διατηρηθούν τα 2 προτεινόμενα επίπεδα VFE για την αναγνώριση αυτοκινήτων. Αυτό σημαίνει ότι ο αριθμός των linear layers στο feature encoding δεν αποτελεί σημαντικό παράγοντα για την απόδοση του μοντέλου σε αντίθεση με την επιλογή των τελικών διαστάσεων που αποτελεί σημαντικότερο παράγοντα.

5.3.2 Αλλαγή του αριθμού των sampled points

Για την εκτίμηση του ρόλου που παίζει ο αριθμός των points που λαμβάνονται από κάθε Voxel έγινε δοκιμή αύξησης του αριθμού των points που λαμβάνονται από κάθε Voxel. Η προτεινόμενη αρχιτεκτονική ορίζει δειγματοληψία 35 points από τα Voxels που έχουν πάνω από 35. Έτσι έγινε δοκιμή να αυξηθούν στην αρχή σε 40 και στην συνέχεια σε 45. Δεν παρατηρήθηκε σημαντική βελτίωση της ακρίβειας και μείωση των απωλειών για τις αυτές τιμές και έτσι υιοθετήθηκε ο προτεινόμενος αριθμός 35 για τα παραπάνω μοντέλα.

5.3.3 Συμπεράσματα και Μελλοντική Εργασία

Ο εντοπισμός των αυτοκινήτων σε 2 διαστάσεις (σε κάτοψη) αποτελεί ευκολότερη διαδικασία για το εκάστοτε μοντέλο το οποίο σημειώνει αξιόλογες επιδόσεις με μεγαλύτερη πρόκληση να αποτελούν τα αυτοκίνητα τα οποία είναι παρκαρισμένα ή κινούνται σε γωνίες σε σχέση με το σύστημα αξόνων. Στα απλούστερα μοντέλα (5.1.3) μεγάλο μέρος των απωλειών είναι η απώλεια από λάθος ή όχι απολύτως σωστά στραμμένο κυτίο αναγνώρισης το οποίο καταλαμβάνει ένα μέρος του εκάστοτε αυτοκινήτου αλλά όχι ολόκληρο το αυτοκίνητο. Στο μοντέλο 8 η εναλλαγή αρχικών clouds και περιστραμμένων clouds φαίνεται πως διευκολύνει την εκμάθηση διαφορετικών γωνιών των αυτοκινήτων. Η τρισδιάστατη αναγνώριση αποτελεί την μεγαλύτερη πρόκληση για το RPN μοντέλο, ταυτόχρονα με την αναγνώριση στραμμένων αυτοκινήτων που φαίνεται πως οδηγούν σε χαμηλό orientation similarity, το οποίο το regression δεν αντιμετωπίζει πάντα επιτυχώς. Τα μοντέλα που δοκιμάστηκαν και σχολιάστηκαν είναι απλώς μερικά από τα πολλά πιθανά μοντέλα που μπορούν να δοκιμαστούν για τον εντοπισμό τρισδιάστατων αυτοκινήτων με στόχο την ενσωμάτωση τους σε συστήματα αυτόνομης οδήγησης αλλά και σε συστήματα υποβοήθησης του οδηγού.

- Εφόσον παρατηρήθηκε ότι η αρχική προεπεξεργασία των Point Clouds και η διατήρηση των σημείων με reflectance = 0 είναι πολύ σημαντική για την συλλογή χρήσιμων χαρακτηριστικών των clouds προτείνεται **η δοκιμή διατήρησης και των points με reflectance < 0** με στόχο πλουσιότερη πληροφορία σαν είσοδο στο μοντέλο αναγνώρισης.

- Ένα άλλο σημαντικό πρόβλημα στον εντοπισμό αυτοκινήτων αλλά και γενικα αντικειμένων στο οδικό δίκτυο είναι όπως παρατηρήθηκε ο προσανατολισμός των αντικειμένων στον δρόμο. Για την αντιμετώπιση του προβλήματος αυτού προτείνεται **η χρήση διαφορετικών μεθόδων IoU όπως περιστραμμένου IoU** [83] με στόχο να υπολογίζονται οι τομές σε αντικείμενα διαφορετικών προσανατολισμών και όχι απλώς πάνω στους άξονες.

- Επιπλέον για την κωδικοποίηση των points στο Voxel Feature μπορεί να οριστεί **να απορρίπτονται voxels με έναν πολύ μικρό αριθμό από points του cloud** εφόσον πιθανώς τέτοια Voxels δεν περικλείουν χρήσιμη πληροφορία και ίσως δυσκολεύουν το δίκτυο Encoding να διαμορφώσει μια εικόνα για το σχήμα και την μορφή του κάθε Voxel.

- Για την δημιουργία περισσότερων προτάσεων-προβλέψεων στο Regional Proposal Network μπορεί επίσης να δοκιμαστεί σε κάθε sliding window **μεγαλύτερος αριθμός από προκαθορισμένα κυτία αναγνώρισης** (anchors) και κυρίως με περισσότερες πιθανές γωνίες (όχι μόνο 0 και 90), ώστε να συλλαμβάνονται περισσότερα αυτοκίνητα με στραμμένο προσανατολισμό και να αυξάνεται το Orientation Precision.

- Όσον αφορά τα Convolutional Layers είναι επίσης μια καλή ιδέα **να δοκιμαστούν διαφορετικοί συνδυασμοί** μεγέθους πυρήνα και βήματος ή και αριθμού φίλτρων και γεμίσματος.

Βιβλιογραφία

[1] B. Marr, "A short history of machine learning every manager should know," *https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=68f4d8a515e7.,* 2016.

[2] SAGAR SHARMA, "What The Hell is a Perceptron," *https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53,* 2017.

[3] Haykin, Simon, in *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 2009, pp. 10-13,47-55.

[4] Manu Shaurya, "Mcculloch-Pitts-neuron-vs-perceptron-model," *https://medium.com/@manushaurya/mcculloch-pitts-neuron-vs-perceptron-model-8668ed82c36*, 2019.

[5] Haykin, Simon, in *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 2009, pp. 122-134.

[6] David E. Rumelhart and Geoffrey E. Hinton and Ronald J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533-536, 1986.

[7] He, Sijun, "The 1986 Backpropagation Paper," *https://sijunhe.github.io/blog/2017/03/27/reading-notes-the-1986-backpropagation-paper/.*

[8] Thomas M. Cover and Peter E. Hart, "Nearest neighbor pattern classification," *IEEE TRANSACTIONS OF INFORMATION THEORY,* vol. 13, pp. 21-27, 1967.

[9] Leif E. Peterson, "K-nearest neighbor," *Scholarpedia,* vol. 4, p. 1883, 2009.

[10] "Long short-term memory," *https://en.wikipedia.org/wiki/Long_short-term_memory.*

[11] "a brief history of machine learning," *https://www.dataversity.net/a-brief-history-of-machine-learning/#.*

[12] IBM Cloud Education, "Supervised Learning," *https://www.ibm.com/cloud/learn/supervised-learning,* 2020.

[13] "Supervised Learning," *https://en.wikipedia.org/wiki/Supervised_learning.*

[14] "Unsupervised Learning," *https://en.wikipedia.org/wiki/Unsupervised_learning.*

[15] "Semi-supervised Learning," *https://en.wikipedia.org/wiki/Semi-supervised_learning.*

[16] Jason Brownlee, "What Is Semi-Supervised Learning," *https://machinelearningmastery.com/what-is-semi-supervised-learning/,* 2021.

[17] "Reinforcement Learning," *https://en.wikipedia.org/wiki/Reinforcement_learning.*

[18] "Computer Vision," *https://paperswithcode.com/area/computer-vision.*

[19] JORDAN, JEREMY, "An overview of object detection: one-stage methods.," *https://www.jeremyjordan.me/object-detection-one-stage/,* 2018.

[20] Park, Sieun, "A guide to Two-stage Object Detection: R-CNN, FPN, Mask R-CNN," *https://medium.com/codex/a-guide-to-two-stage-object-detection-r-cnn-fpn-mask-r-cnn-and-more-54c2e168438c,* 2021.

[21] "Self-driving car," *https://en.wikipedia.org/wiki/Self-driving_car.*

[22] Haykin, Simon, in *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 2009, pp. 1-12.

[23] LeDoux, Joseph, in *Synaptic Self: How Our Brains Become Who We Are*, 2003, pp. 51-56.

[24] Haykin, Simon, in *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 2009, pp. 3-6.

[25] Haykin, Simon, in *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 2009, pp. 47-50.

[26] "Multilayer Perceptron," *https://en.wikipedia.org/wiki/Multilayer_perceptron.*

[27] Marc Peter Deisenroth, A. Aldo Faisal , Cheng Soon Ong, in *MATHEMATICS FOR MACHINE LEARNING*, Cambridge University Press, 2020, pp. 145-148.

[28] "Gradient Descent," *https://en.wikipedia.org/wiki/Gradient_descent.*

[29] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *ArXiv,* vol. abs/1609.04747, 2016.

[30] Ning Qian, "On the momentum term in gradient descent learning algorithms," vol. 12, no. 1, pp. 145-151, 1999.

[31] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/k^2)," *Central Economics and Mathematics Institute,USSR Academy of Sciences, Moscow,* vol. 269, no. 3, pp. 543-547, 1983.

[32] John C. Duchi and Elad Hazan and Yoram Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," in *J. Mach. Learn. Res.*, vol. 12, 2010.

[33] Matthew D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *ArXiv,* vol. abs/1212.5701, 2012.

[34] Hinton, Geoff, "Neural Networks for Machine Learning ,Lecture 6a," *https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.*

[35] "RMSProp," *https://paperswithcode.com/method/rmsprop.*

[36] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *CoRR,* vol. abs/1412.6980, 2015.

[37] "Training ,validation, and test data sets," *https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets.*

[38] IBM Cloud Education, "Overfitting," *https://www.ibm.com/cloud/learn/overfitting,* 2021.

[39] "Early Stopping," *https://en.wikipedia.org/wiki/Early_stopping.*

[40] IBM Cloud Education, "Underfitting," *https://www.ibm.com/cloud/learn/underfitting.*

[41] "Activation function," *https://en.wikipedia.org/wiki/Activation_function.*

[42] "Activation Functions," *https://paperswithcode.com/methods/category/activation-functions.*

[43] Baheti, Pragati, "Activation Functions in Neural Networks [12 Types & Use Cases]," *https://www.v7labs.com/blog/neural-networks-activation-functions,* 2022.

[44] IBM Cloud Education, "Deep Learning," *https://www.ibm.com/cloud/learn/deep-learning.*

[45] Brajesh Kumar, "Convolutional Neural Networks: A Brief History of their Evolution," *https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597,* 2021.

[46] Yann LeCun and Léon Bottou and Yoshua Bengio and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE,* vol. 86, pp. 2278-2324, 1998.

[47] Fei-Fei Li & Justin Johnson & Serena Yeung, "Stanford Lecture 5,Convolutional Neural Networks," *http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf,* Spring 2017.

[48] "Tensorflow 1 Version Layers," *https://www.tensorflow.org/api_docs/python/tf/compat/v1/layers.*

[49] Sergey Ioffe and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *ICML*, 2015.

[50] Kuan Wei, "Understand Transposed Convolutions," *https://towardsdatascience.com/understand-transposed-convolutions-and-build-your-own-transposed-convolution-layer-from-scratch-4f5d97b2967,* 2020.

[51] MYO NeuralNet , "Calculating the Output Size of Convolutions and Transpose Convolutions," *https://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html,* 2020.

[52] Aqeel Anwar, "What is Transposed Convolutional Layer?," *https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11,* 2020.

[53] "Precision and recall," *https://en.wikipedia.org/wiki/Precision_and_recall.*

[54] "Jaccard index," *https://en.wikipedia.org/wiki/Jaccard_index.*

[55] Kukil, "Intersection over Union (IoU) in Object Detection and Segmentation," *https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/,* 2022.

[56] Mark Everingham and Luc Van Gool and Christopher K. I. Williams and John M. Winn and Andrew Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision,* vol. 88, p. 11, June 2009.

[57] Jonathan Hui, "mAP (mean Average Precision) for Object Detection," *https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173,* 2018.

[58] Andreas Geiger and Philip Lenz and Raquel Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3354-3361, 2012.

[59] "Cross Entropy," *https://en.wikipedia.org/wiki/Cross_entropy.*

[60] "Cross-Entropy," *https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.*

[61] Ross B. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV),* p. 1442, 2015.

[62] Ross B. Girshick and Jeff Donahue and Trevor Darrell and Jitendra Malik*, "R*ich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*,"* 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp.* 580-587, 2014*.*

[63] "Support-vector machine", https://en.wikipedia.org/wiki/Support-vector_machine.

[64] Uijlings, J. R. R. and van de Sande, K. E. A. and Gevers, T. and Smeulders, A. W. M., "Selective Search for Object Recognition," *International Journal of Computer Vision,* vol. 104, no. 2, pp. 154-171, 2013.

[65] Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, pp. 1137-1149, 2015.

[66] Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, p. 1141, 2015.

[67] Ross B. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV),* pp. 1440-1448, 2015.

[68] "Voxel," *https://en.wikipedia.org/wiki/Voxel.*

[69] Martin Engelcke and Dushyant Rao and Dominic Zeng Wang and Chi Hay Tong and Ingmar Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," *2017 IEEE International Conference on Robotics and Automation (ICRA),* pp. 1355-1361, 2017.

[70] Dominic Zeng Wang and Ingmar Posner, "Voting for Voting in Online Point Cloud Object Detection.," *Robotics: Science and Systems,* 2015.

[71] C. Premebida, J. Carreira, J. Batista and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data,," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 4112-4117, 2014.

[72] M. Enzweiler and D. M. Gavrila, "A Multilevel Mixture-of-Experts Framework for Pedestrian Classification," *IEEE Transactions on Image Processing,* vol. 20, no. 10, pp. 2967-2979, 2011.

[73] Yin Zhou and Oncel Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 4490-4499, 2018.

[74] Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, p. 1139, 2015.

[75] Andreas Geiger and Philip Lenz and Christoph Stiller and Raquel Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research,* vol. 32, pp. 1231-1237, 2013.

[76] Andreas Geiger and Frank Moosmann and Omer Car and Bernhard Schuster, "Automatic camera and range sensor calibration using a single shot," *2012 IEEE International Conference on Robotics and Automation,* pp. 3936-3943, 2012.

[77] Hamid Serry, "KITTI Analysis," *https://anyverse.ai/synthetic-data/synthetic-data-develop-rustworthy-autonomous-driving-system-chapter2/* .

[78] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR,* vol. abs/1409.1556, 2015.

[79] A. G. Howard, "Some Improvements on Deep Convolutional Neural Network Based Image Classification," *CoRR,* vol. abs/1312.5402, 2014.

[80] Kaiming He and X. Zhang and Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2016.

[81] Benjin Zhu and Zhengkai Jiang and Xiangxin Zhou and Zeming Li and Gang Yu, "Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection," *ArXiv,* vol. abs/1908.09492, 2019.

[82] Yin Zhou and Oncel Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object

Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* p. 4494-4495, 2018.

[83] Yu Zheng and Danyang Zhang and Sinan Xie and Jiwen Lu and Jie Zhou, "Rotation-Robust Intersection over Union for 3D Object Detection," *ECCV,* 2020.

[84] 3D Car Detection Project , https://github.com/staks1/3dObjectDetection