



ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ /
**TECHNICAL
UNIVERSITY
OF CRETE**

Βέλτιστη Διαστασιολόγηση και Τοποθέτηση
Σταθμών Φόρτισης Ηλεκτρικών Οχημάτων με
τη βοήθεια Εναέριας Επιτήρησης της
Κυκλοφορίας.

Διπλωματική εργασία

Αρβανίτης Χρυσόστομος

Πολυτεχνείο Κρήτης

Τμήμα ΗΜΜΥ

Εξεταστική Επιτροπή:

Αναπληρωτής Καθηγητής Κανέλλος Φώτιος (Επιβλέπων)

Καθηγητής Σταυρακάκης Γεώργιος

Καθηγητής Παρτσινέβελος Παναγιώτης

Χανιά 2023

Ευχαριστίες

Ευχαριστώ τους καθηγητές Κανέλλο Φώτιο και Παρτσινέβελο Παναγιώτη για την καθοδήγηση κατά τη διάρκεια εκπόνησης της πτυχιακής εργασίας. Επιπλέον, ευχαριστώ το εργαστηριακό προσωπικό του κυρίου Παρτσινέβελου για τη βοήθεια στην καταγραφή των βίντεο με το drone καθώς και τον συμφοιτητή μου Κωσταντίνο Λεωντσίνη που μου επέτρεψε να επεκτείνω τη δική του εργασία.

Abstract

Global warming has brought about significant climate change, triggering a chain reaction that threatens human existence. Vehicles with internal combustion engines emit carbon dioxide (CO_2), a gaseous pollutant that enhances the phenomenon, into the environment. With the rise in the world's population linked to the increase in vehicles with conventional engines, it is becoming increasingly imperative to produce vehicles that do not emit harmful pollutants. This realization coupled with the fact that oil reserves are running out has led automakers to focus on producing electric cars that come with advantages and disadvantages over those with internal combustion engines.

In this thesis, a method is followed to find the optimal locations for the placement of charging stations along the Northern Road Axis of Crete (NRAC). First, to calculate the speeds of moving vehicles on the NRAC, video recordings were made with a drone and from these videos the dataset was extracted to train the pre-trained weights of YOLOv5s. The new weights were then used to detect the vehicles in the videos, combined with OpenCV to detect their motion, in Python programming language, and finally the velocities of the moving vehicles were extracted.

In the next stage, an energy evaluation is performed to estimate the power consumption of the vehicle battery according to their dynamic equations. The number of moving vehicles for each hour of the day was calculated from a survey by the Ministry of Infrastructure and Transport. The scenarios are Chania-Rethymnon, Rethymnon-Heraklion and Chania-Heraklion. The speed assigned to each vehicle is linked to the actual speeds calculated from the videos while the charging and stress rate are linked to the corresponding scenario. Finally, an inference is drawn about whether a driver is looking for a charging station along the motorway based on the battery charge rate and driver anxiety rate by a fuzzy logic system.

Then, by adjusting the parameters of PSO appropriately for each scenario, the optimal locations for the placement of charging stations are calculated based on the cost function. The number of chargers is calculated from the charging time and the number of vehicles at each station.

To summarize, the thesis focuses on the most efficient placement of charging stations for the above scenarios, using the fuzzy logic system, mentioned above, to estimate the mileage a driver searches for a charging station and the Particle Swarm Optimazion algorithm for optimal placement of charging stations and the number of chargers at them.

Περίληψη

Το φαινόμενο του θερμοκηπίου έχει επιφέρει σημαντικές κλιματικές αλλαγές πυροδοτώντας μια αλυσιδωτή αντίδραση που απειλή την ανθρώπινη ύπαρξη. Τα οχήματα με μηχανές εσωτερικής καύσης, εκπέμπουν στο περιβάλλον διοξείδιο του άνθρακα (CO_2) που είναι αέριος ρύπος που ενισχύει το φαινόμενο. Με την άνοδο του πληθυσμού του πλανήτη που συνδέεται από την αύξηση των οχημάτων με συμβατικούς κινητήρες γίνεται όλο και πιο επιτακτική η ανάγκη για την παραγωγή οχημάτων που δεν εκπέμπουν επιβλαβείς ρύπους. Αυτή η συνειδητοποίηση σε συνδυασμό με το γεγονός ότι τα αποθέματα πετρελαίου εξαντλούνται, οδήγησε τις αυτοκινητοβιομηχανίες στο να επικεντρωθούν στην παραγωγή ηλεκτρικών αυτοκινήτων που έρχονται με πλεονεκτήματα αλλά και μειονεκτήματα έναντι αυτών με μηχανές εσωτερικής καύσης.

Στην διπλωματική εργασία ακολουθείτε μέθοδος για την εύρεση των βέλτιστων σημείων τοποθέτησης σταθμών φόρτισης κατά μήκος του Βόρειου Οδικού Άξονα της Κρήτης (BOAK). Αρχικά, για τον υπολογισμό των ταχυτήτων των κινούμενων οχημάτων στον BOAK, έγιναν καταγραφές βίντεο με μη επανδρωμένο αεροσκάφος και από τα βίντεο αυτά εξάχθηκε το σύνολο δεδομένων που δόθηκε για εκπαίδευση των προεκπαιδευμένων βαρών του YOLOv5s. Στην συνέχεια, χρησιμοποιήθηκαν τα νέα βάρη για την ανίχνευση των οχημάτων στα βίντεο, σε συνδυασμό με την OpenCV για την ανίχνευση κίνησης αυτών, σε γλώσσα προγραμματισμού Python, και τελικά εξάχθηκαν οι ταχύτητες των κινούμενων οχημάτων.

Σε επόμενο στάδιο, γίνεται η ενεργειακή αξιολόγηση για να εκτιμηθεί η κατανάλωση ενέργειας της μπαταρίας των οχημάτων σύμφωνα με τις δυναμικές τους εξισώσεις. Ο αριθμός των κινούμενων οχημάτων για κάθε ώρα της ημέρας υπολογίστηκε από έρευνα του υπουργείου υποδομών και συγκοινωνιών. Τα σενάρια είναι Χανιά-Ρέθυμνο, Ρέθυμνο-Ηράκλειο και Χανιά-Ηράκλειο. Οι ταχύτητα που αποδίδεται σε κάθε όχημα συνδέεται με τις πραγματικές ταχύτητες που υπολογίστηκαν από τα βίντεο ενώ το

ποσοστό φόρτισης και άγχους συνδέονται με το αντίστοιχο σενάριο. Τελικά, εξάγεται το συμπέρασμα για το αν ένας οδηγός αναζητάει για σταθμό φόρτισης κατά μήκος του BOAK, με βάση το ποσοστό φόρτισης μπαταρίας και το ποσοστό άγχους του οδηγού από σύστημα ασαφούς λογικής (fuzzy).

Στην συνέχεια, ρυθμίζοντας κατάλληλα για κάθε σενάριο τις παραμέτρους του PSO, υπολογίζονται τα βέλτιστα σημεία για την τοποθέτηση σταθμών φόρτισης, με βάση την συνάρτηση κόστους. Το πλήθος των φορτιστών υπολογίζεται από τον χρόνο φόρτισης και το πλήθος των οχημάτων σε κάθε σταθμό.

Ανακεφαλαιώνοντας, η διπλωματική εργασία εστιάζει στην αποδοτικότερη τοποθέτηση σταθμών φόρτισης για τα παραπάνω σενάρια, χρησιμοποιώντας το σύστημα ασαφούς λογικής, που αναφέρθηκε παραπάνω, για την εκτίμηση του χιλιομέτρου που κάποιος οδηγός αναζητά σταθμό φόρτισης και τον αλγόριθμο Particle Swarm Optimization για την βέλτιστη τοποθέτηση σταθμών φόρτισης και του πλήθους των φορτιστών σε αυτούς.

Περιεχόμενα

Ευχαριστίες	1
Abstract	1
Περίληψη	4
1 Εισαγωγικό Μέρος	17
1.1 Εισαγωγή	17
1.2 Ιστορική αναδρομή ηλεκτρικών αυτοκινήτων	18
1.3 Προηγούμενη δουλειά	19
1.4 Στόχος Διπλωματικής Εργασίας	20
2 Μέρος Α1	21
2.1 Εισαγωγή	21
2.2 Εισαγωγή στη Μηχανική Μάθηση	22
2.3 Είδη και Αλγόριθμοι Μηχανικής Μάθησης	23
2.3.1 Επιβλεπόμενη Μάθηση	23
2.3.2 Μη Επιβλεπόμενη Μάθηση	27
2.3.3 Ενισχυτική Μάθηση	28
2.4 Εισαγωγή στη Μηχανική Όραση	29
2.5 Ψηφιακή εικόνα	30
2.6 Συνελικτικά Νευρωνικά Δίκτυα(Convolutional Neural Networks)	31
2.6.1 Συνελικτικό στρώμα	32
2.6.2 Συγκεντρωτικό στρώμα	33
2.6.3 Πλήρες συνδεδεμένο στρώμα	34
2.7 Αλγόριθμοι ανίχνευσης αντικειμένων	35

2.7.1	R-CNN	35
2.7.2	Fast R-CNN	36
2.7.3	Faster R-CNN	37
2.7.4	YOLO	39
2.7.4.1	Εισαγωγή στον αλγόριθμο YOLO	39
2.7.4.2	YOLOv1	40
2.7.4.3	YOLOv2	42
2.7.4.4	YOLOv3	42
2.7.4.5	YOLOv4	44
2.7.4.6	YOLOv5	46
2.8	Αξιολόγηση απόδοσης αλγορίθμων ανίχνευσης αντικειμένων	47
2.9	Non-Maximum Suppresion	50
2.9.1	Soft-NMS	51
2.10	Παρακολούθηση κίνησης αντικειμένων	53
2.10.1	Αλγόριθμος SORT	54
2.10.2	Αλγόριθμος DeepSORT	54
2.11	Αλγόριθμος αφαίρεσεις παρασκήνιου (Background subtraction algorithms)	56
2.11.1	GMG	57
2.11.2	Mixture of Gaussians (MOG)	57
2.11.3	Mixture of Gaussians 2 (MOG2)	57
3	Μέρος A2	58
3.1	Εισαγωγή	58
3.2	Χρήσιμες πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν	59
3.2.1	Μη επανδρωμένα αεροσκάφη(UAVs)	59
3.2.2	Βιβλιοθήκη FFMPEG - Roboflow	61
3.2.3	Anaconda - Weight and Biased	62
3.2.4	Python and PyTorch	63
3.2.5	Open Source Computer Vision Library(OpenCV)	63
3.3	Διαδικασία εκπαίδευσης βαρέων του αλγόριθμου ανίχνευσης αντικειμένων	64
3.3.1	Σημεία καταγραφών	64
3.3.2	Εξαγωγή καρέ και ανάθεση ετικετών	65

3.3.3	Διαδικασία εκπαίδευσης	67
3.3.4	Αξιολόγηση μοντέλου	71
3.3.5	Αλληλεπίδραση με τα εκπαιδευμένα βάρη	73
3.3.6	Διαδικασία εξαγωγής ταχυτήτων	74
3.3.6.1	Εξαγωγή σε τύπο ONNX	74
3.3.6.2	Ανίχνευση αντικειμένων	75
3.3.6.3	Ανίχνευση κίνησης αντικειμένων	75
3.3.6.4	Εξαγωγή ταχυτήτων	76
3.3.6.5	Βέλτιστη εξαγωγή ταχυτήτων	77
4	Μέρος Β	84
4.1	Εισαγωγή	84
4.2	Εργαλείο Matlab	84
4.3	Δυναμικές εξισώσεις οχημάτων	86
4.4	Διαδικασία ενεργειακής αξιολόγησης	88
4.4.1	Ηλεκτρικά αυτοκίνητα	88
4.4.2	Ωριαίος κυκλοφοριακός φόρτος	91
4.4.3	Χρήση συνάρτησης Gendist	101
4.4.4	Χρήση συστήματος ασαφούς λογικής (Fuzzy)	105
4.4.5	Σύνοψη	109
4.5	Particle swarm optimazation Algoritm	110
4.6	Διαδικασία βέλτιστης τοποθέτησης σταθμών φόρτισης	113
4.6.1	Υπολογισμός χρόνου φόρτισης οχημάτων, πλήθους οχημάτων και φορτιστών στους σταθμούς	113
4.6.2	Αξιοποίηση αλγόριθμου	115
4.6.2.1	Παράμετροι PSO	116
4.6.2.2	Συνάρτηση κόστους PSO	117
5	Παρουσίαση και τεκμηρίωση αποτελεσμάτων	122
5.1	Εισαγωγή	122
5.2	Χανιά-Ρέθυμνο	122
5.3	Ρέθυμνο-Ηράκλειο	132
5.4	Χανιά-Ηράκλειο	142

5.5 Συμπεράσματα - Παρατηρήσεις	151
---	-----

Κατάλογος Σχημάτων

1.1	Model T (Ford official website)	18
1.2	Toyota Prius - Tesla Logo	19
2.1	Σχήμα Μηχανικής Μάθησης (Πηγή: eurixgroup)	22
2.2	Είδη Μηχανικής Μάθησης (Πηγή: https://sciencemystic.com/machine-learning/)	23
2.3	Παράδειγμα Επιβλεπόμενης Μάθησης (Πηγή: Javapoint)	24
2.4	Παράδειγμα Γραμμικής Παλινδρόμησης (Πηγή: Javapoint)	25
2.5	Παράδειγμα Δυαδικής Ταξινόμησης (Πηγή: Javapoint)	26
2.6	Παράδειγμα Μη Επιβλεπόμενης Μάθησης (Πηγή: Javapoint)	27
2.7	Ενισχυτική Μάθηση (Πηγή: Javapoint)	28
2.8	Εικόνες ως πίνακες	31
2.9	Συνελικτικά Νευρωνικά Δίκτυα (Πηγή: medium)	32
2.10	Συνέλιξη μεταξύ εικόνας εισόδου και φίλτρου (Πηγή: IBM)	33
2.11	Παράδειγμα zero padding (Πηγή: Medium)	33
2.12	Max/Average Pooling - (Πηγή: Medium)	34
2.13	R-CNN (Πηγή: https://arxiv.org/pdf/1311.2524.pdf).	36
2.14	Fast R-CNN (Πηγή: https://arxiv.org/pdf/1504.08083.pdf)	37
2.15	Faster R-CNN (Πηγή: https://arxiv.org/pdf/1506.01497.pdf)	38
2.16	YOLOv1	41
2.17	Darknet-53 architecture with 5 residual blocks (square box) containing bottle neck structure (1x1 followed by 3x3 convolutional layers). (Yanjia LiYanjia Li, 2019)	43
2.18	YOLOv3	44
2.19	Στάδια Ανίχνευσης - Bag of Freebies YOLOv4	46

2.20 IoU[1]	48
2.21 Παράδειγμα καμπύλης Preciso-Recall με παρεμβολή[1]	49
2.22 Παράδειγμα NMS[2]	51
2.23 ΠροβλήματαNMS (Πηγή: https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c)	51
2.24 Soft-NMS[2]	52
2.25 Παράδειγμα Soft NMS (Πηγή: https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c)	52
2.26 Παράδειγμα ΑλγόριθμουSORT [3]	54
2.27 Παράδειγμα DeepSORT (Πηγή: https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/)	55
2.28 Παράδειγμα Background Subtraction - Πηγή: (https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html)	56
3.1 Τύποι μη επανδρωμένων αεροσκαφών	60
3.2 DJI Phantom4 Pro v2.0 (official website)	60
3.3 FFMPEG - Roboflow	61
3.4 Conda - WB	62
3.5 Python - PyTorch	63
3.6 OpenCV-logo (official website)	63
3.7 Τοποθεσίες καταγραφής βίντεο - BOAK	65
3.8 Τύποι οχημάτων (Πηγή: https://mechanicbase.com/cars/different-car-models-types/)	66
3.9 Παράδειγμα ανάθεσης ετικετών σε εικόνες	67
3.10 Παράδειγμα αρχείου ετικετών	67
3.11 Εγκατάσταση YOLOv5	68
3.12 Επεξεργασία αρχείου data.yaml	68
3.13 Διαδικασία εκπαίδευσης	68
3.14 Σύγκριση προεκαπιδευμένων βάρων YOLOv5 (Πηγή: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data)	69
3.15 last epoch	70
3.16 epoch 1 to 5	70
3.17 Γραφική απεικόνιση μετρικών του μοντέλου	71

3.18 Καμπύλη Precision-Recall	71
3.19 losses	72
3.20 Detect.py	74
3.21 Αποτελέσματα Detect.py	74
3.22 Εξαγωγή ως ONNX	75
3.23 Δείγμα παρακολούθησης κίνησης	76
3.24 Δείγμα της μεθόδου	77
3.25 Ανίχνευση πριν την κόκκινη γραμμή	79
3.26 Παράδειγμα εφαρμογής της μάσκας	79
3.27 Ακριβώς πριν περάσει την πράσινη γραμμή	79
3.28 Η απόσταση υπολογισμένη στο google earth	80
3.29 καρέ YOLO	80
4.1 Λογότυπο Matlab[4]	84
4.2 Διαδρομή Χανιά - Ρέθυμνο (A)	90
4.3 Διαδρομή Χανιά - Ηράκλειο ($RS2 - A$)	90
4.4 Διαδρομή Ρέθυμνο - Ηράκλειο ($R = \frac{RS2-A}{2}$)	91
4.5 Τοποθεσίες αυτογραφικών μηχανισμάτων που χρησιμοποιούνται για την καταγραφή των κινούμενων οχημάτων	91
4.6 Αριθμός κινούμενων οχημάτων στον BOAK κατά την θερινή περίοδο (μέση τιμή μήνα)	92
4.7 Αριθμός κινούμενων οχημάτων στον BOAK κατά την χειμερινή περίοδο (μέση τιμή μήνα)	93
4.8 Μέση ημερήσια κυκλοφορία και στις 2 κατευθύνσεις ανά σταθμό και μήνα	93
4.9 Ποσοστό φόρτισης μπαταρίας διαδρομών	103
4.10 Κατανομή ποσοστού άγχους του οδηγού σε ότι αφορά το ποσοστό φόρτισης της μπαταρίας.	104
4.11 Ποσοστό αρχικής φόρτισης μπαταρίας ως είσοδο στο σύστημα fuzzy . .	106
4.12 Ποσοστό άγχους ως είσοδο στο σύστημα fuzzy	107
4.13 Έξοδος συστήματος fuzzy	108
4.14 Έξοδος συστήματος fuzzy συναρτήσει των δύο εισόδων	109
4.15 Μεταβολή κόστους σε σχέση με το είδος, την τοποθεσία και τις εφαρ- μογές, φορτιστών επιπέδου 2 - Πηγή [5]	119

5.1	Ιστογράμμα διαδρομής Χανιά - Ρέθυμνο	123
5.2	Audi Q4 e-tron 35:Ταχύτητα $96 \frac{km}{h}$, Άγχος 49% , Μπαταρία 48% ,Χιλιόμετρο 48 °	123
5.3	Fiat 500e: Ταχύτητα $76 \frac{km}{h}$, Άγχος 52% ,Μπαταρία 53% , Χιλιόμετρο 63°	124
5.4	Nissan Leaf:Ταχύτητα $56 \frac{km}{h}$, Άγχος 48% , Μπαταρία 43% , Χιλιόμετρο 37°	124
5.5	Mazda MX-30:Ταχύτητα $96 \frac{km}{h}$, Άγχος 42% , Μπαταρία 67% , Χιλιόμετρο 56°	125
5.6	Πρόβλημα 4 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)	126
5.7	Πρόβλημα 5 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)	126
5.8	Πρόβλημα 6 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)	127
5.9	Πρόβλημα 7 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)	127
5.10	Πρόβλημα 8 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)	128
5.11	Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Χανιά - Ρέθυμνο	128
5.12	Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Χανιά - Ρέθυμνο	129
5.13	Πρόβλημα 4 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο	129
5.14	Πρόβλημα 5 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο	130
5.15	Πρόβλημα 6 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο	130
5.16	Πρόβλημα 7 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο	131

5.17 Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο	131
5.18 Ιστογράμμο διαδρομής Ρέθυμνο - Ηράκλειο	133
5.19 Audi Q4 e-tron 35: Ταχύτητα $125 \frac{km}{h}$, Άγχος 39%, Μπαταρία 68%, Χιλιόμετρο 73 °	133
5.20 Fiat 500e: Ταχύτητα $96 \frac{km}{h}$, Άγχος 21%, Μπαταρία 59%, Χιλιόμετρο 64°	134
5.21 Nissan Leaf: Ταχύτητα $66 \frac{km}{h}$, Άγχος 59%, Μπαταρία 58%, Χιλιόμετρο 77°	134
5.22 Mazda MX-30: Ταχύτητα $106 \frac{km}{h}$, Άγχος 48%, Μπαταρία 60%, Χιλιόμε- τρο 53°	135
5.23 Πρόβλημα 4 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)	136
5.24 Πρόβλημα 5 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)	136
5.25 Πρόβλημα 6 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)	137
5.26 Πρόβλημα 7 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)	137
5.27 Πρόβλημα 8 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)	138
5.28 Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοπο- θέτησης 5 σταθμών Ρέθυμνο - Ηράκλειο	138
5.29 Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοπο- θέτησης 5 σταθμών Ρέθυμνο - Ηράκλειο	139
5.30 Πρόβλημα 4 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο	139
5.31 Πρόβλημα 5 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο	140
5.32 Πρόβλημα 6 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο	140
5.33 Πρόβλημα 7 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο	141

5.34 Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο	141
5.35 Ιστογράμμο διαδρομής Χανιά - Ηράκλειο	143
5.36 Audi Q4 e-tron 35: Ταχύτητα πριν το Ρέθυμνο $96 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $125 \frac{km}{h}$, Άγχος 47 %, Μπαταρία 78 %, Χιλιόμετρο 117 °	143
5.37 Fiat 500e: Ταχύτητα πριν το Ρέθυμνο $86 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυ- μνο $96 \frac{km}{h}$, Άγχος 26 %, Μπαταρία 82 %, Χιλιόμετρο 128.7 °	144
5.38 Nissan Leaf: Ταχύτητα πριν το Ρέθυμνο $66 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυ- μνο $76 \frac{km}{h}$, Άγχος 36 %, Μπαταρία 71 %, Χιλιόμετρο 110 °	144
5.39 Mazda MX-30: Ταχύτητα πριν το Ρέθυμνο $66 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $116 \frac{km}{h}$, Άγχος 53 %, Μπαταρία 79 %, Χιλιόμετρο 94 °	145
5.40 Πρόβλημα 8 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)	146
5.41 Πρόβλημα 10 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)	146
5.42 Πρόβλημα 12 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)	147
5.43 Πρόβλημα 14 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)	147
5.44 Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοπο- θέτησης 10 σταθμών Χανιά - Ηράκλειο	148
5.45 Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοπο- θέτησης 10 σταθμών Χανιά - Ηράκλειο	148
5.46 Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο	149
5.47 Πρόβλημα 10 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο	149
5.48 Πρόβλημα 12 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο	150
5.49 Πρόβλημα 14 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο	150

Κατάλογος Πινάκων

3.1 Ταχύτητες παρούσας εργασίας - ομαδοποιημένες	81
3.2 Ταχύτητες προηγούμενης εργασίας(Λεωντσίνη) - ομαδοποιημένες	82
3.3 Σύνολο ταχυτήτων - ομαδοποιημένο	82
4.1 Σχέση πυκνότητας αέρα με θερμοκρασία - Πηγή (https://www.engineersedge.com/calculators/air-density.htm) . .	87
4.2 συντελεστή οπισθέλκουσας του αέρα σε σχέση με τον τύπο οχήματος - Πηγή([6])	87
4.3 Τιμή συντελεστή αντίστασης κύλισης υπό συνθήκες - Πηγή([6])	88
4.4 Χαρακτηριστικά Audi Q4 e-tron 35 - Πηγή([7])	89
4.5 Χαρακτηριστικά Fiat 500e Hatchback - Πηγή([8])	89
4.6 Χαρακτηριστικά Nissan Leaf - Πηγή([9])	89
4.7 Χαρακτηριστικά Mazda MX-30 - Πηγή([10])	90
4.8 Ωριαίος κυκλοφοριακός φόρτος για τον σταθμό RS-2	95
4.9 Ωριαίος κυκλοφοριακός φόρτος για τον σταθμό RS-4	96
4.10 Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Χανιά-Ρέθυμνο (A) .	98
4.11 Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Χανιά-Ηράκλειο(RS2– A)	99
4.12 Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Ρέθυμνο-Ηράκλειο(R)	100
4.13 Πίνακας πιθανοτήτων για κάθε ομάδα ταχυτήτων	102
4.14 Χαρακτηρισμός επιπέδων ποσοστού φόρτισης της μπαταρίας	106
4.15 Χαρακτηρισμός επιπέδων ποσοστού άγχους οδηγού	106
4.16 Έξοδος συστήματος fuzzy	107

Κεφάλαιο 1

Εισαγωγικό Μέρος

1.1 Εισαγωγή

Είναι γεγονός ότι τις τελευταίες δεκαετίες ο πληθυσμός έχει αυξηθεί ραγδαία, λόγω της οικονομικής και τεχνολογικής ανάπτυξης. Σε έρευνα του οργανισμού United Nations αναφέρεται ότι ο πληθυσμός έχει φτάσει στα 7.97 δισεκατομμύρια το 2022. Αυτή η αύξηση του πληθυσμού αυξάνει και τις ανάγκες κατανάλωσης ορυκτών καυσίμων και συνεπώς επιδεινώνεται το φαινόμενο του θερμοκηπίου, καθώς η καύση συμβατικών καυσίμων απελευθερώνει διοξείδιο του άνθρακα στην ατμόσφαιρα, φέρνοντας στο προσκήνιο ένα πολύ επίκαιρο πρόβλημα, αυτό της υπερθέρμανσης του πλανήτη (global warming). Επιπλέον, σε έρευνα του [11] αναφέρεται ότι τα αποθέματα βενζίνη αναμένεται να τελειώσουν το 2068. Συνεπώς, το γεγονός ότι τα αυτοκίνητα με μηχανές εσωτερικής καύσης εκπέμπουν διοξείδιο του άνθρακα στο περιβάλλον σε συνδυασμό με το ότι τα ορυκτά καύσιμα σταδιακά εξαντλούνται οδήγησε τις σύγχρονες αυτοκινητοβιομηχανίες να στραφούν στην παραγωγή οχημάτων φιλικών προς το περιβάλλον. Τέτοια οχήματα είναι ηλεκτροκίνητα τα οποία δεν εκπέμπουν CO_2 στο περιβάλλον και καταστέλλουν το παραπάνω φαινόμενο. Η αλλαγή του στόλου των οχημάτων των αυτοκίνητα με συμβατικούς κινητήρες σε ηλεκτροκίνητα είναι κάτι που σταδιακά αρχίζει να συμβαίνει, με τις αυτοκινητοβιομηχανίες να ανταγωνίζονται, παράγοντας αξιόπιστα ηλεκτρικά αυτοκίνητα σε όλο και πιο προνομιακές τιμές. Η αλλαγή αυτή ίσως χρειαστεί χρόνο, γεγονός που μας επιτρέπει να προετοιμαστούμε κατάλληλα, μέσα από μελέτες, για την αποδοτικότερη αξιοποίηση τους με τα τεχνολογικά μέσα που διαθέτουμε.

1.2 Ιστορική αναδρομή ηλεκτρικών αυτοκινήτων

Αρχικά η ιδέα των ηλεκτρικών κινητήρων δεν είναι κάποιο νέο εγχείρημα αλλά εμφανίστηκε τον 18ο αιώνα. Τον 19ο αιώνα ξεκίνησε η παραγωγή των πρώτων ηλεκτρικών αυτοκινήτων που ήταν πιο δημοφιλή από τα βενζινοκίνητα και τα ατμοκίνητα καθώς ήταν ανθόρυβα, ευκολότερα στην οδήγηση και δεν εκπέμπαν επιβλαβής ρύπους. Αξιοσημείωτο είναι το γεγονός ότι ένας από τους διάσημους εφευρέτες εκείνης της εποχής, ο Thomas Edison, πίστευε ότι η τεχνολογία των ηλεκτρικών αυτοκινήτων ήταν ανώτερη και προσπάθησε να δημιουργήσει καλύτερες ηλεκτρικές μπαταρίες. Ωστόσο, ο Henry Ford με το Model T ,



Σχήμα 1.1: Model T (Ford official website)

που ήταν βενζινοκίνητο, κατέκλυσε την αγορά, καθώς ήταν οικονομικότερο και η βενζίνη δεν κόστιζε ακριβά λόγω του ότι εκείνη την περίοδο υπήρχε μεγάλο απόθεμα. Έτσι τα ηλεκτρικά αυτοκίνητα παραμερίστηκαν, τα αυτοκίνητα με μηχανές εσωτερικής καύσης συνέχισαν να εξελίσσονται και έτσι δεν υπήρχε η ανάγκη παραγωγής άλλου είδους αυτοκινήτων.

Έπειτα από χρόνια, το γεγονός ότι τα αποθέματα πετρελαίου άρχισαν να στερεύουν, οδήγησε αρκετούς κατασκευαστές αυτοκινήτων στο να αρχίσουν να εξερευνούν αυτοκίνητα με διαφορετικά καύσιμα, συμπεριλαμβανομένου των ηλεκτρικών αυτοκινήτων. Ωστόσο, το μεγάλο μειονέκτημα των ηλεκτρικών αυτοκινήτων ήταν η χαμηλή ταχύτητα και η μικρή τους αυτονομία. Συγκεκριμένα εκείνη την περίοδο, το 1970, τα ηλεκτρικά αυτοκίνητα μπορούσαν να αναπτύξουν ταχύτητα μόλις $60 - 80 \text{ km/h}$ και η αυτονομία τους κυμαινόταν στο εύρος $60 - 70 \text{ km}$. Η πραγματική ανάπτυξη των ηλεκτρικών αυτοκινήτων ξεκίνησε τον 21ο αιώνα, με τη μαζική παραγωγή του αμιγώς ηλεκτρικού αυτοκινήτου Toyota Prius hybrid το οποίο έγινε ευρέως γνωστό το 2000 και με την ίδρυση της start up Tesla Motors στο Silicon Valley [12] [13].

Τα επόμενα χρόνια ξεκίνησαν περισσότεροι δημιουργοί αυτοκινήτων να κατασκευάζουν ηλεκτρικά αυτοκίνητα, ξεκίνησε η εγκατάσταση όλο και περισσότερων σταθμών φόρτισης ενώ η ανάπτυξη των μπαταριών λιθίου βοήθησαν στην κατασκευή οικονομι-



(a) Prius 1997 (Toyota official website)



(b) Tesla Logo (Tesla official website)

Figure 1.2: Toyota Prius - Tesla Logo

κότερων και αποδοτικότερων ηλεκτρικών αυτοκινήτων, δίνοντας περισσότερες επιλογές στους καταναλωτές. Είναι σημαντικό να αναφερθούν μερικά από τα πλεονεκτήματα των ηλεκτρικών αυτοκινήτων. Σε αυτά υπάγονται το ότι είναι φιλικά προς το περιβάλλον και η ηλεκτρική ενέργεια που απαιτείται για τη φόρτιση του μπορεί να προέρχεται από πυρηνικά εργοστάσια ή από ανανεώσιμες πηγές ενέργειας. Ωστόσο, παρά την ανάπτυξη της τεχνολογίας, ακόμα και σήμερα τα ηλεκτρικά αυτοκίνητα διατηρούν κάποια μειονεκτήματα. Σε αυτά συμπεριλαμβάνεται το μέγεθος της μπαταρίας με αυτονομία από 95 – 695km ανάλογα το μοντέλο, όπως αναφέρεται στο [14]. Επιπλέον, η μπαταρίες έχουν περιορισμένη διάρκεια ζωής (από 3 μέχρι 5 χρόνια) και το κόστος σε περίπτωση αντικατάστασης τους είναι μεγάλο. Σημαντικό πρόβλημα επίσης είναι το γεγονός ότι στην Ελλάδα δεν υπάρχουν αρκετοί σταθμοί φόρτισης σε εθνικές οδούς και σε μικρότερες πόλεις, γεγονός που αποθαρρύνει του καταναλωτές μικρότερων πόλεων, που δεν διαθέτουν τα απαραίτητα μέσα, στο να αγοράσουν ένα ηλεκτρικό αυτοκίνητο.

1.3 Προηγούμενη δουλειά

Χρησιμές πληροφορίες για το θεωρητικό υπόβαθρο, για την μέθοδο ανίχνευσης κίνησης καθώς και για την ενεργειακή αξιολόγηση παραχωρήθηκαν από την διπλωματική εργασία με τίτλο 'Εκτίμηση κατανάλωσης ενέργειας ηλεκτροκινήτων οχημάτων μέσω εναέριας παρατήρησης της κυκλοφορίας' του συμφοιτητή Κωσταντίνου Λεωνσίνη. Συγκεκριμένα συμπεριλήφθηκαν οι ταχύτητες που εξάχθηκαν από βίντεο τριών τοποθεσιών που καταγράφηκαν με εναέριο μέσω παρακολούθησης, η τεχνική παρακολούθησης κίνησης των οχημάτων, σχετική έρευνα για τον κυκλοφοριακό φόρτο του Βόρειου Οδικού Άξονα της Κρήτης (BOAK), θεωρητικό υπόβαθρο για της δυναμικές εξισώσεις

κίνησης των τεσσάρων οχημάτων καθώς και η μέθοδος της ενεργειακής αξιολόγησης.

1.4 Στόχος Διπλωματικής Εργασίας

Τα ηλεκτρικά αυτοκίνητα στο κοντινό μέλλον θα αντικαταστήσουν αυτά με τις μηχανές εσωτερικής καύσης που χρησιμοποιούν συμβατικά καύσιμα, καθώς δεν εκπέμπουν επιβλαβής για το περιβάλλον ρύπους και καταστέλλουν το φαινόμενο υπερθέρμανσης του πλανήτη. Ένας ακόμη σημαντικός παράγοντας που οδήγησε τις αυτοκινητοβιομηχανίες να στραφούν στην εντατική παραγωγή ηλεκτρικών αυτοκινήτων είναι το γεγονός ότι τα αποθέματα πετρελαίου λιγοστεύουν και αναμένετε κάποια στιγμή στο μέλλον να εξαντληθούν. Είναι ωφέλιμο να αναγνωρίσουμε ότι το μεγαλύτερο πρόβλημα που αντιμετωπίζουν τα ηλεκτρικά αυτοκίνητα στην Ελλάδα είναι η φόρτιση της μπαταρίας τους. Ήδη στα μεγάλα αστικά κέντρα υπάρχουν οι απαραίτητες δομές για την υποστήριξη αυτής της τεχνολογίας. Ωστόσο, οι μικρότερες πόλεις, οι εθνικοί οδοί και οι αυτοκινητόδρομοι δε διαθέτουν ακόμα τα κατάλληλα μέσα για τη φόρτιση της μπαταρίας των ηλεκτρικών αυτοκινήτων. Στην παρούσα διπλωματική εργασία γίνεται μελέτη για τη βέλτιστη τοποθέτησή ηλεκτρικών σταθμών φόρτισης αυτοκινήτων στον Βόρειο Οδικό Άξονας Κρήτης (BOAK) εκτιμώντας την κατανάλωση ενέργειας των ηλεκτρικών αυτοκινήτων που κινούνται σε αυτόν με τη βοήθεια εναέριου μέσου παρακολούθησης.

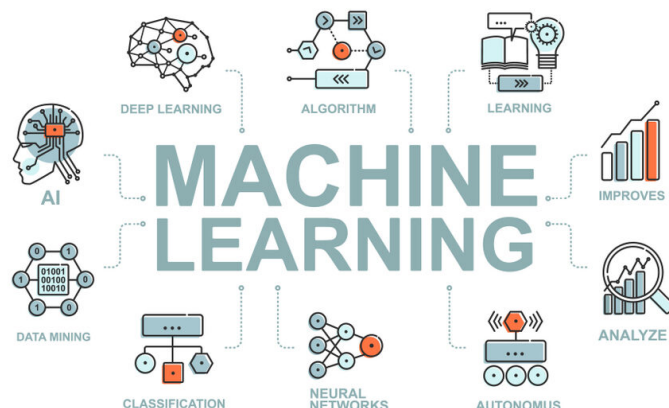
Κεφάλαιο 2

Μέρος A1

2.1 Εισαγωγή

Στο Μέρος A1, αναπτύσσεται σταδιακά το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση της πειραματικής διαδικασίας που ακολουθεί στο Μέρος A2. Πιο συγκεκριμένα επεξηγούνται βασικές αρχές της μηχανικής μάθησης και της μηχανικής όρασης, αναλύονται βασικές αρχές των συνελικτικών νευρωνικών δικτύων, αναφέρεται ο τρόπος λειτουργίας και οι αρχιτεκτονικές των βασικότερων αλγόριθμων ανίχνευσης αντικειμένων, δίνοντας ιδιαίτερη έμφαση στην οικογένεια αλγορίθμων YOLO. Επίσης, εξετάζονται μετρικές αξιολόγησης της απόδοσης αλγορίθμων ανίχνευσης αντικειμένων. Επιπλέον, εξετάζεται η διαδικασία παρακολούθησης κίνησης αντικειμένων και οι δυο σημαντικότεροι αλγόριθμοι παρακολούθησης κίνησης, ο SORT και ο DeepSort. Τέλος, εξηγείται η μέθοδος αφαίρεσης φόντου background subtraction καθώς και τρεις αλγόριθμοι αφαίρεσης φόντου.

2.2 Εισαγωγή στη Μηχανική Μάθηση

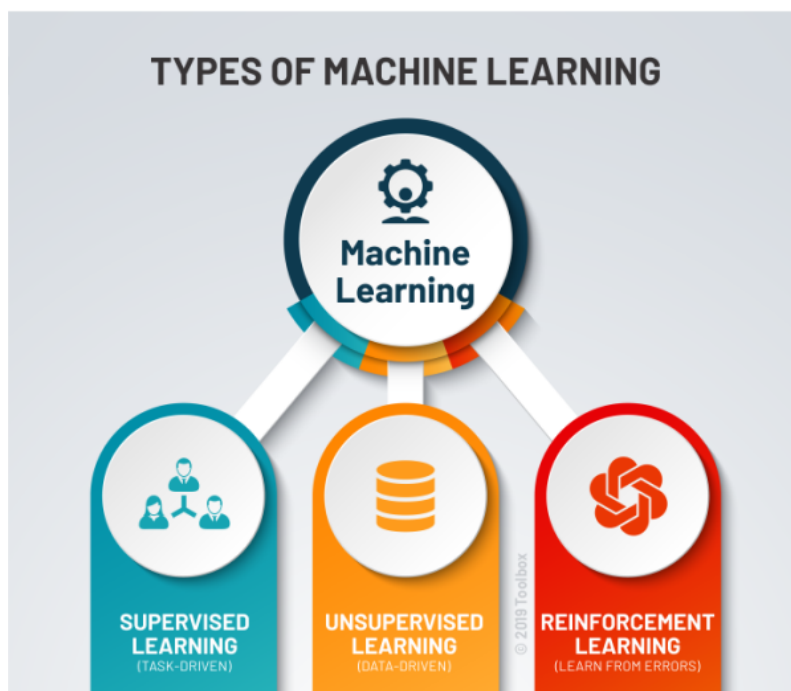


Σχήμα 2.1: Σχήμα Μηχανικής Μάθησης (Πηγή: eurixgroup)

Η Μηχανική Μάθηση (Machine Learning) είναι τομέας της Τεχνητής Νοημοσύνης (Artificial Intelligence) που παρέχει σε λογισμικά προγράμματα την ικανότητα να μαθαίνουν αυτόματα και να βελτιώνονται χωρίς να είναι ρητά προγραμματισμένα. Για αυτή τη διαδικασία εκμαθήσεις απαιτούνται δεδομένα για την αναγνώριση μοτίβων που βρίσκονται σε αυτά. Ο Tom Mitchell, ένας από τους προτοπόρους της επιστήμης της μηχανικής μάθησης, αναφέρει τη μηχανική μάθηση ως τη μελέτη αλγορίθμων υπολογιστή που επιτρέπει σε υπολογιστήκα προγράμματα να βελτιώνονται αυτόματα μέσω των εμπειριών. Μπορούμε να σκεφτούμε τον αλγόριθμο ως ένα σύνολο οδηγιών που καθορίζονται από κάποιον προγραμματιστή και μπορεί να επεξεργαστεί ο υπολογιστής. Η μηχανική μάθηση λειτουργεί παρόμοια με την ανάπτυξη ενός παιδιού. Όσο το παιδί μεγαλώνει, η εμπειρία E (Experience) στην εκτέλεση μιας εργασίας T (Task) αυξάνεται με αποτέλεσμα τη βελτίωση της επίδοσης P (Performance). Στη μηχανική μάθηση μια μηχανή αναλαμβάνει μια εργασία T , την εκτελεί και μετράει την απόδοσή της P . Μια μηχανή διαθέτει μεγάλο πλήθος δεδομένων και έτσι λοιπόν καθώς επεξεργάζεται αυτά τα δεδομένα η εμπειρία E αυξάνεται με την πάροδο του χρόνου γεγονός που οδηγεί σε μεγάλη απόδοση P . Με αυτό τον τρόπο, έπειτα από την επεξεργασία των δεδομένων η ακρίβεια του μοντέλου μηχανικής μάθησης αυξάνεται και οδηγεί σε ακριβείς προβλέψεις. Μερικά χαρακτηριστικά παραδείγματα αλγορίθμων μηχανικής μάθησης είναι αλγόριθμοι που πραγματοποιούν προβλέψεις μετοχών, αλγόριθμοι που προτείνουν προϊόντα όπως ρούχα, ταινίες, μουσική με βάση προϊόντα που έχουμε αγοράσει/αναζη-

τήσει στο παρελθόν, αλγόριθμοι που διαχωρίζουν τα ανεπιθύμητα email από τα χρήσιμα καθώς και αλγόριθμοι εκμάθησης αυτόματης οδήγησης σε συστήματα πλοήγησης.[15]

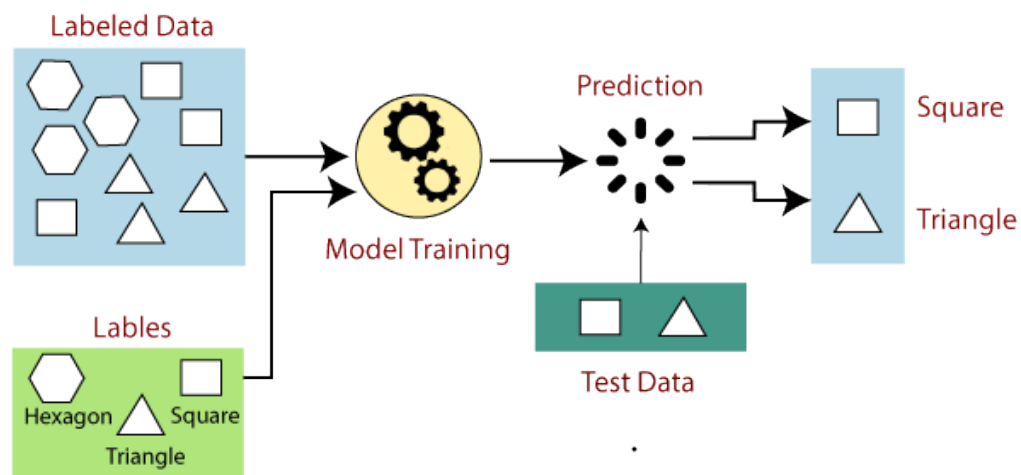
2.3 Είδη και Αλγόριθμοι Μηχανικής Μάθησης



Σχήμα 2.2: Είδη Μηχανικής Μάθησης (Πηγή: <https://sciencemystic.com/machine-learning/>)

2.3.1 Επιβλεπόμενη Μάθηση

Η επιβλεπόμενη μάθηση (Supervised learning) είναι το είδος της μηχανικής μάθησης κατά το οποίο η εκπαίδευση του αλγορίθμου γίνεται με δεδομένα που έχουν ετικέτες και με βάση αυτά τα δεδομένα γίνονται προβλέψεις για την έξοδο. Δεδομένα με ετικέτες σημαίνει ότι για κάποιες εισόδους έχουν επισημανθεί οι επιθυμητές έξοδοι από κάποιον ερευνητή. Τα δεδομένα που δίνονται στη μηχανή λειτουργούν ως επιβλέπων που εκπαιδεύει τη μηχανή να προβλέπει την επιθυμητή έξοδο. Η λογική είναι παρόμοια με αυτή που ένας καθηγητής εκπαιδεύει έναν μαθητή. Σκοπός είναι ο αλγόριθμος επιβλεπόμενης μάθησης να βρει σχέσεις μεταξύ εισόδου και εξόδου. Μετά από την εκπαίδευση του αλγορίθμου με δεδομένα με ετικέτες δοκιμάζονται νέα δεδομένα στον εκπαιδευμένο αλγόριθμο για να προβλέψει τις εξόδους.



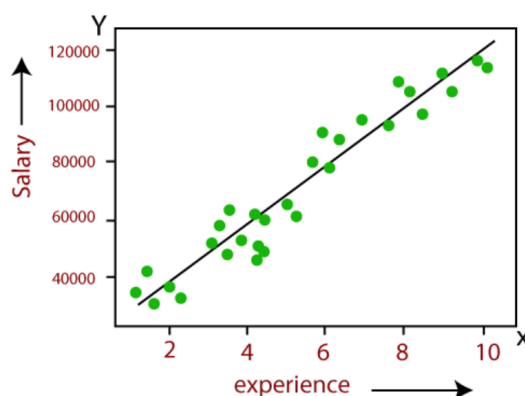
Σχήμα 2.3: Παράδειγμα Επιβλεπόμενης Μάθησης (Πηγή: Javapoint)

Δίνεται ένα παράδειγμα για καλύτερη κατανόηση. Στο παράδειγμα 2.3 δίνεται ένα σύνολο δεδομένων για την αναγνώριση σχήματος με ετικέτες τετράγωνο, τρίγωνο και εξάγωνο, γίνεται εκπαίδευση του αλγόριθμου και έπειτα γίνονται δοκιμές στον εκπαιδευμένο αλγόριθμο για να διαπιστωθεί αν προβλέπει το σωστό σχήμα.[16] Η επιβλεπόμενη μάθηση χωρίζεται περαιτέρω σε αλγόριθμους **Παλινδρόμησης (Regression)** και **Ταξινόμησης (Classification)**.

- Παλινδρόμηση (Regression):** Η παλινδρόμηση είναι μια στατιστική μέθοδος που μοντελοποιεί τη σχέση μεταξύ μιας εξαρτημένης και μιας ανεξάρτητης μεταβλητής με μία ή περισσότερες ανεξάρτητες μεταβλητές. Πιο συγκεκριμένα μας βοηθάει να καταλάβουμε πως η τιμή μιας εξαρτημένης μεταβλητής αλλάζει σε σχέση με μια ανεξάρτητη όταν οι υπόλοιπες ανεξάρτητες μεταβλητές παραμένουν σταθερές. Προβλέπει συνεχείς μεταβλητές όπως θερμοκρασία, ηλικία και μισθός. Η παλινδρόμηση μας βοηθάει να βρούμε τη σχέση μεταξύ μεταβλητών και μας επιτρέπει να προβλέψουμε συνεχείς εξόδους βασισμένοι σε προηγούμενες προβλέψεις. Στην παλινδρόμηση σχεδιάζουμε μια γραφική με σημεία των μεταβλητών που ταιριάζουν καλύτερα στα δεδομένα και χρησιμοποιώντας αυτά τα σημεία το μοντέλο μηχανικής μάθησης πραγματοποιεί προβλέψεις. Η παλινδρόμηση δείχνει μια γραμμή ή καμπύλη που διέρχεται από όλα τα σημεία δεδομένων στο γράφημα που εξηγήθηκε με τρόπο τέτοιο ώστε η κάθετη απόσταση μεταξύ τη γραμμής και των σημείων να είναι η ελάχιστη και η απόσταση αυτή δείχνει αν το μοντέλο έχει ισχυρή σχέση ή όχι.

Τα σημαντικότερα είδη παλινδρόμησης είναι:

- Linear Regression(Γραμμική Παλινδρόμηση)
- Logistic Regression(Λογιστική παλινδρόμηση)
- Polynomial Regression(Πολυωνυμική παλινδρόμηση)
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression [17]



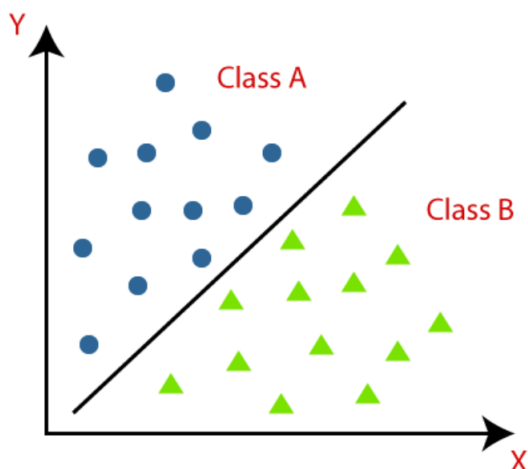
Σχήμα 2.4: Παράδειγμα Γραμμικής Παλινδρόμησης (Πηγή: Javapoint)

- **Ταξινόμηση(Classification):**Ο αλγόριθμος της ταξινόμησης χρησιμοποιείται για την αναγνώριση κατηγοριών σε νέες παρατηρήσεις βασισμένες στα δεδομένα εκπαίδευσης. Στην ταξινόμηση το πρόγραμμα μαθαίνει από τα δεδομένα εκπαίδευσης που του δίνονται και στη συνέχεια κατηγοριοποιεί τις νέες παρατηρήσεις σε έναν αριθμό κλάσεων. Οι κλάσεις αυτές συνήθως καλούνται ετικέτες(labels) ή κατηγορίες.Ο αλγόριθμος ταξινόμησης δέχεται δεδομένα με ετικέτα, δηλαδή δέχεται εισόδους με την αντίστοιχη έξοδο.Οι έξοδοι στην ταξινόμηση είναι διακριτές, καθώς είναι κάποια κλάση(όπως αυτοκίνητο τύπου A ή αυτοκίνητο τύπου B).Η μαθηματική σχέση μεταξύ εισόδου και εξόδου είναι: $y = f(x)$ όπου $y = \text{categorical output}$.Ο κύριος σκοπός του αλγόριθμου ταξινόμησης είναι η αναγνώριση κατηγοριών σε ένα σύνολο δεδομένων (dataset).

Υπάρχουν δύο είδη ταξινόμησης:

- Η Δυαδική ταξινόμηση, όπου οι κατηγορίες είναι μόνο δύο όπως για παράδειγμα αυτοκίνητο ή μηχανή.

- Η πολλαπλών κλάσεων ταξινόμηση όπου οι κατηγορίες μπορεί να είναι αυτοκίνητο τύπου A, αυτοκίνητο τύπου B, αυτοκίνητο τύπου Γ.



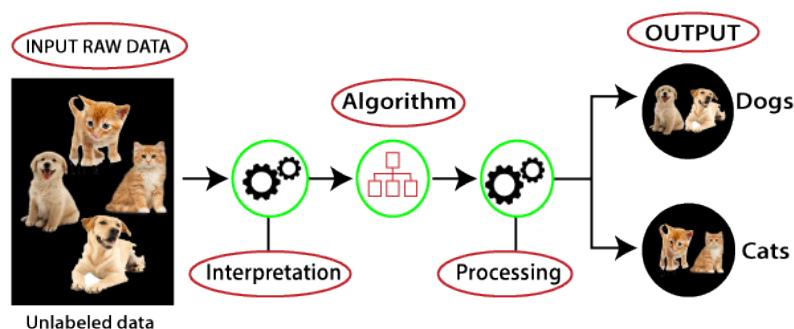
Σχήμα 2.5: Παράδειγμα Δυναδικής Ταξινόμησης (Πηγή: Javapoint)

Επίσης υπάρχουν και δύο τρόποι εκμάθησης:

- Ο Lazy Learner, όπου πρώτα αποθηκεύονται τα δεδομένα εκπαίδευσης του dataset και στη συνέχεια λαμβάνονται τα δεδομένα δοκιμών του. Σε αυτή την περίπτωση η ταξινόμηση πραγματοποιείται βασιζόμενη στα πιο σχετικά δεδομένα που είναι αποθηκευμένα στο dataset εκπαίδευσης. Χρειάζεται λιγότερο χρόνο στην εκπαίδευση αλλά απαιτείται περισσότερος χρόνος για τις προβλέψεις. Τέτοιοι αλγόριθμοι είναι οι K-NN algorithm, Case-based reasoning.
- Ο Eager Learner, όπου αναπτύσσεται ένα μοντέλο ταξινόμησης βασισμένο στα δεδομένα εκπαίδευσης του dataset πριν ληφθούν τα δεδομένα δοκιμών του. Σε αυτή την περίπτωση απαιτείται περισσότερος χρόνος κατά τη διαδικασία εκπαίδευσης και λιγότερος χρόνος για τις προβλέψεις. Τέτοιοι αλγόριθμοι είναι οι Decision Trees, Naïve Bayes, ANN.[18]

2.3.2 Μη Επιβλεπόμενη Μάθηση

Η μη επιβλεπόμενη μάθηση (Unsupervised Learning) είναι μια μέθοδος μηχανικής μάθησης στην οποία τα δεδομένα δεν έχουν ετικέτες και συνεπώς δεν υπάρχει επίβλεψη στα δεδομένα που δίνονται για εκπαίδευση στο μοντέλο. Το μοντέλο πρέπει να αναγνωρίσει κρυφά μοτίβα και πληροφορίες από τα δεδομένα που του δίνονται. Σκοπός του είναι να βρει τη δομή του dataset και να ομαδοποιήσει τα δεδομένα σύμφωνα με τις ομοιότητές και διαφορές τους με τη χρήση αλγορίθμων όπως οι k-means clustering, Decision tree.



Σχήμα 2.6: Παράδειγμα Μη Επιβλεπόμενης Μάθησης (Πηγή: Javapoint)

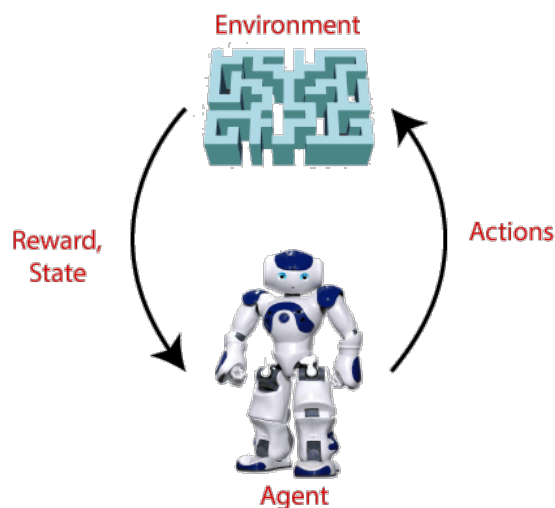
Τα δεδομένα εισόδου δεν έχουν ετικέτα. Αρχικά βρίσκει τα κρυφά μοτίβα στα δεδομένα, έπειτα εφαρμόζει έναν αλγόριθμο όπως οι k-means clustering, Decision tree και ο αλγόριθμος κατηγοριοποιεί τα δεδομένα σύμφωνα με τις ομοιότητες και τις διαφορές τους.

Οι αλγόριθμοι μη επιβλεπόμενης μάθησης χωρίζονται στους:

- Ομαδοποίησης(Clustering):Στους αλγόριθμους ομαδοποίησης τα αντικείμενα με τις περισσότερες ομοιότητες ομαδοποιούνται στο ίδιο σύνολο ενώ αυτά με λιγότερες ή καθόλου ομοιότητες ομαδοποιούνται σε άλλο σύνολο.Με αυτό τον τρόπο δημιουργούνται ομάδες με κοινά χαρακτηριστικά.Χαρακτηριστικά παραδείγματα είναι οι προτάσεις προϊόντων από εταιρίες όπως η Amazon σύμφωνα με αγορές που έγιναν στο παρελθόν καθώς και οι προτάσεις ταινιών από εταιρίες όπως το Netflix σύμφωνα με ταινίες που παρακολούθησαμε στο παρελθόν.
- Συσχετισμού(Association):Οι αλγόριθμοι συσχετισμού χρησιμοποιούνται για την εύρεση σχέσεων μεταξύ μεταβλητών σε ένα μεγάλο σύνολο δεδομένων. Καθο-

ρίζει το σύνολο των δεδομένων που συμβαίνουν με τον ίδιο τρόπο σε ένα σύνολο δεδομένων. Περίπτωση χρήσης του είναι στο Marketing όταν γίνεται ο συσχετισμός ότι ο χρήστης που αγοράζει το x προϊόν συνήθως αγοράζει και το y προϊόν.[19]

2.3.3 Ενισχυτική Μάθηση



Σχήμα 2.7: Ενισχυτική Μάθηση (Πηγή:Javapoint)

Η ενισχυτική μάθηση(Reinforcement Learning) είναι μια τεχνική μηχανικής μάθησης η οποία βασίζεται στην ανατροφοδότηση καθώς ο πράκτορας ενεργεί σε ένα περιβάλλον και παρατηρεί τα αποτελέσματα των ενεργειών αυτών.Για κάθε επιθυμητή ενέργεια ο πράκτορας λαμβάνει μια επιβράβευση ως ανατροφοδότηση ενώ για κάθε ανεπιθύμητη ενέργεια λαμβάνει ως ανατροφοδότηση μια ποινή.Δεν υπάρχουν ετικέτες στα δεδομένα συνεπώς ο πράκτορας περιορίζεται στο να εκπαιδευτεί από την εμπειρίας του.Ο πράκτορας εξερευνάει το περιβάλλον με κύριο σκοπό να βελτιώσει την απόδοση του λαμβάνοντας το μέγιστο δυνατό αποτέλεσμα επιβραβεύσεων.Μπορούμε να πούμε λοιπόν πως η ενισχυτική μάθηση είναι ένας τύπος μηχανικής μάθησης όπου ένας έξυπνος πράκτορας, δηλαδή ένα πρόγραμμα υπολογιστή, αλληλεπιδρά με ένα περιβάλλον και μαθαίνει να ενεργεί σε αυτό. Παράδειγμα χρήσης της ενισχυτικής μάθησης είναι η εκπαίδευση ενός αυτόματου συστήματος οδήγησης.Χρησιμοποιείται και στη ρομποτική και στα ηλεκτρονικά παιχνίδια για την εκπαίδευση πρακτόρων.Οι αλγόριθμοι ενισχυτικής μάθησης χρησιμοποιούνται κυρίως σε εφαρμογές τεχνητής νοημοσύνης(AI). Τέτοιοι αλγόριθμοι είναι:

- Q-Learning
- State Action Reward State Action (SARSA)
- Deep Q Neural Network [20]

2.4 Εισαγωγή στη Μηχανική Όραση

Η μηχανική όραση (Computer Vision) είναι τομέας της τεχνητής νοημοσύνης που επιτρέπει στα υπολογιστικά συστήματα να αντλούν χρήσιμες πληροφορίες από ψηφιακές εικόνες, βίντεο και άλλα οπτικά μέσα και να ενεργεί ή να κάνει προτάσεις βασισμένος σε αυτές τις πληροφορίες. Η μηχανική όραση επιτρέπει σε έναν υπολογιστή να βλέπει, να παρατηρεί και να καταλαβαίνει. Λειτουργεί όπως η ανθρώπινη όραση, με τη διαφορά ότι η ανθρώπινη όραση έχει ένα προβάδισμα καθώς η εκπαιδεύεται καθ' όλη τη διάρκεια της ζωής ενός ανθρώπου και μπορεί να διακρίνει αντικείμενα, να ελέγξει αποστάσεις, κινήσεις και να διακρίνει λάθη σε μια φωτογραφία. Η μηχανική όραση εκπαιδεύει μηχανές για να εκτελούν αυτές τις λειτουργίες με τη διαφορά ότι πρέπει να το κάνει σε πολύ λιγότερο χρόνο με κάμερες, δεδομένα και αλγόριθμους. Συστήματα εκπαιδευμένα με αλγόριθμους μηχανικής όρασης μπορούν να ξεπεράσουν τις ανθρώπινες δυνατότητες όρασης. Η μηχανική όραση χρειάζεται πολλά δεδομένα. Αναλύει τα δεδομένα επαναληπτικά μέχρι να διακρίνει τις διαφορές και τελικά να αναγνωρίζει τις εικόνες. Για παράδειγμα, για την εκπαίδευση ενός υπολογιστή για την αναγνώριση του είδους ενός αυτοκινήτου χρειάζεται ένας πολύ μεγάλος αριθμός εικόνων με αυτοκίνητα πολλών ειδών καθώς και παρόμοια αντικείμενα έτσι ώστε να μπορεί να διακρίνει τις διαφορές και να αναγνωρίζει το είδος.

Η Μηχανική Μάθηση χρησιμοποιεί μοντέλα αλγόριθμων που επιτρέπουν στον υπολογιστή να εκπαιδευτεί αυτόματα για το περιεχόμενο των οπτικών δεδομένων. Με την τροφοδότηση των απαιτούμενων δεδομένων στο μοντέλο, ο υπολογιστής εκπαιδεύεται αυτόματα με τη βοήθεια αλγόριθμων και μπορεί να διακρίνει τις εικόνες, χωρίς να προγραμματιστεί από κάποιον. Ένα Συνελικτικό Νευρωνικό Δίκτυο (Convolutional Neural Networks (CNN)) εκπαιδεύει ένα μοντέλο μηχανικής ή βαθιάς μάθησης τμηματοποιώντας τις εικόνες σε εικονοστοιχεία (pixels) στις οποίες έχουν δοθεί ετικέτες. Χρησιμοποιεί τις ετικέτες αυτές για να εκτελέσει την πράξη της συνέλιξης και να πραγματοποιήσει προβλέψεις στα οπτικά δεδομένα. Το CNN εκτελεί συνέλιξεις και ελέγχει την ακρίβεια

των προβλέψεων μέσα από μια σειρά αναδρομών μέχρι να αρχίσουν να πραγματοποιούνται σωστές προβλέψεις. Πρώτα διακρίνει τις γωνίες και τα απλά σχήματα και στη συνέχεια συμπληρώνει τις πληροφορίες εκτελώντας αναδρομές στις προβλέψεις. Το CNN χρησιμοποιείται για την κατανόηση απλών εικόνων. Με παρόμοιο τρόπο χρησιμοποιείται ένα recurrent neural network (RNN) για εφαρμογές βίντεο για την κατανόηση του πως οι εικόνες σε μια σειρά από καρέ σχετίζονται μεταξύ τους. [21]

2.5 Ψηφιακή εικόνα

Αρχικά είναι χρήσιμο να αναφέρουμε κάποιες εισαγωγικές γνώσεις για τις ψηφιακές εικόνες. Μπορούμε να θεωρήσουμε μια ασπρόμαυρη εικόνα ως έναν πίνακα 2-D (x, y) που το γινόμενο της κάθε διάστασης μας δίνει το πλήθος των εικονοστοιχείων ενώ η τιμή του κάθε εικονοστοιχείου καθορίζεται από το πλήθος των επιπέδων χβάντισης. Για παράδειγμα, σε μια εικόνα με 8 επίπεδα χβάντισης ($G = 2^8 = 255$), η βάση είναι 2 καθώς η εικόνα είναι ψηφιακή (0 ή 1), η τιμή που παίρνει κάθε εικονοστοιχείο (x, y) είναι $0 \leq (x, y) \leq G - 1 = 255$ με το 0 να είναι μαύρο, το 255 να είναι άσπρο ενώ όλες οι ενδιάμεσες τιμές να είναι αποχρώσεις αυτών. Η ανάλυση της εικόνας είναι $x \times y \times m$ με $m = 1$ καθώς η εικόνα είναι ασπρόμαυρη. Για μια έγχρωμη εικόνα RGB ισχύουν όμοια πράγματα με τη διαφορά του ότι υπάρχουν 3 πίνακες 2-D, ένας για το Red κανάλι, ένας για το Green και ένας για το Blue και η ανάλυση είναι $x \times y \times m$ με $m = 3$ καθώς έχουμε 3 κανάλια. Εύκολα μετατρέπουμε μια RGB εικόνα σε HSV ή YCbCr.

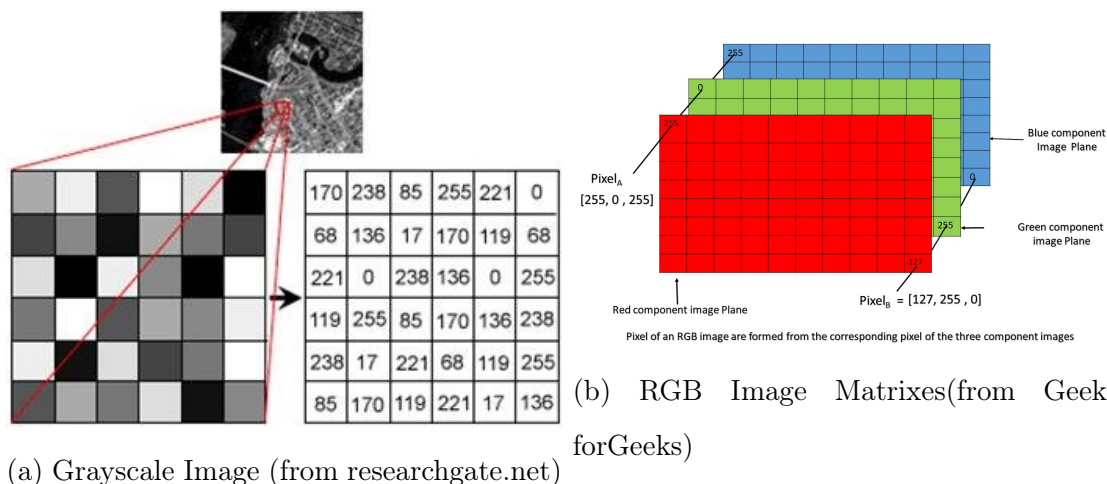
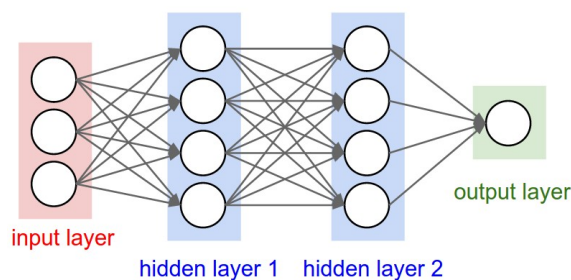


Figure 2.8: Εικόνες ως πίνακες

2.6 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)

Τα Νευρωνικά Δίκτυα είναι ένα υποσύνολο της μηχανικής μάθησης (Machine Learning) και αποτελούν τον πυρήνα των αλγορίθμων βαθιάς μάθησης (Deep Learning). Αποτελούνται από στρώματα κόμβων, εμπεριέχοντας ένα στρώμα εισόδου, ένα ή περισσότερα κρυφά στρώματα και ένα στρώμα εξόδου. Κάθε κόμβος συνδέεται με κάποιον άλλο και έχει ένα σχετικό βάρος και κατώφλι. Αν η έξοδος ενός κόμβου είναι μεγαλύτερη από την καθορισμένη τιμή κατωφλίου ο κόμβος ενεργοποιείται και στέλνει τα δεδομένα του στο επόμενο στρώμα του δικτύου. Διαφορετικά δεν αποστέλλονται δεδομένα στο επόμενο στρώμα του δικτύου. Υπάρχουν πολλοί τύποι νευρωνικών δικτύων που χρησιμοποιούνται σε διαφορετικές περιπτώσεις. Για παράδειγμα, τα recurrent neural networks (RNN) χρησιμοποιούνται συνήθως για στην επεξεργασία φυσικών γλωσσών και στην αναγνώριση φωνής ενώ τα Convolutional Neural Networks (CNN) χρησιμοποιούνται για ταξινόμηση και εφαρμογές μηχανικής μάθησης. Τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) παρέχουν μια αναβαθμισμένη προσέγγιση στην ταξινόμηση εικόνων και στην αναγνώριση αντικειμένων εκμεταλλευόμενα αρχές της γραμμικής άλγεβρας και συγκεκριμένα τον πολλαπλασιασμό πινάκων για την αναγνώριση μοτίβων σε μια εικόνα. Απαιτούν μεγάλη υπολογιστική ισχύ και για αυτό απαιτείται κάρτα γραφικών για την εκπαίδευση μοντέλων. [22]

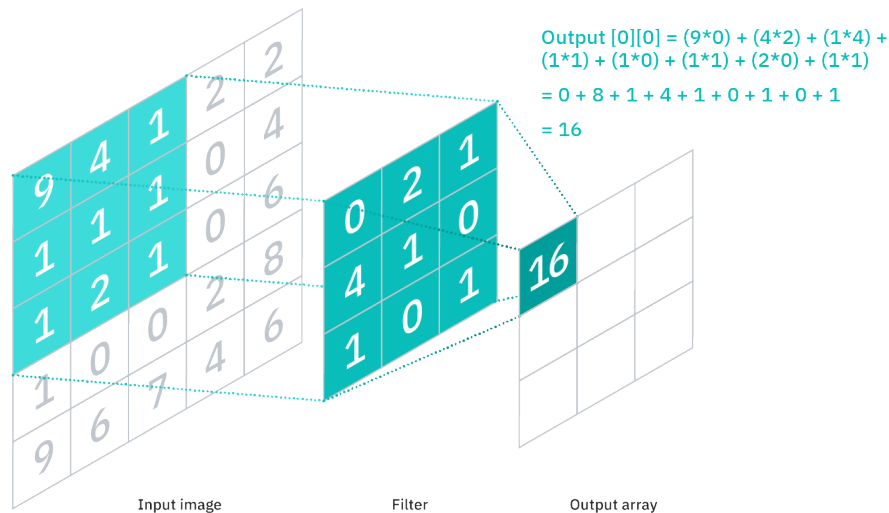


Σχήμα 2.9: Συνελικτικά Νευρωνικά Δίκτυα (Πηγή: medium)

Τα Συνελικτικά Νευρωνικά Δίκτυα(CNN) διαχωρίζονται από τα Νευρωνικά Δίκτυα(Neural Networks) καθώς έχουν καλύτερη απόδοση σε δεδομένα εικόνας, ομιλίας, ήχου και σημάτων.Υπάρχουν τρεις κύριες κατηγορίες στρωμάτων που αναλύονται παρακάτω.

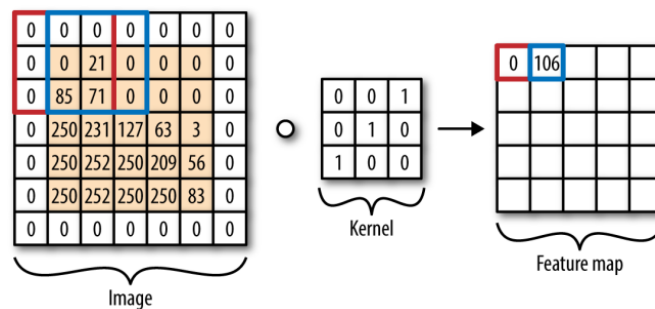
2.6.1 Συνελικτικό στρώμα

Το Συνελικτικό στρώμα(Convolutional layer) είναι ο πυρήνας του (CNN) και σε αυτό πραγματοποιείται το μεγαλύτερο μέρος των υπολογισμών.Δέχεται τα δεδομένα εισόδου και το φίλτρο και εξάγει τον πίνακα χαρακτηριστικών(feature map).Αν υποθέσουμε ότι η είσοδος είναι μια RGB έγχρωμη εικόνα, που αποτελείται από έναν πίνακα εικονοστοιχείων τριών διαστάσεων (ύψος,πλάτος,βάθος) τότε μετακινώντας ένα φίλτρο, γνωστό ως kernel, σε κάθε RGB κανάλι πάνω στην εικόνα εφαρμόζουμε την πράξη της συνέλιξης.Το φίλτρο είναι ένας πίνακας 2-D που περιέχει βάρη.Το μέγεθος του διαφέρει αλλά συνήθως είναι ένας 3×3 πίνακας γενικά ένας πίνακας $m \times m$ με m περιττός αριθμός.Το φίλτρο εφαρμόζεται σε μια περιοχή της εικόνας και εξάγεται ένα patch από την εικόνα εισόδου και το φίλτρο.Αυτό το patch δίνεται στον πίνακα εξόδου.Το φίλτρο μετακινείται κατά έναν αριθμό εικονοστοιχείων(stride) και η διαδικασία επαναλαμβάνεται μέχρι να σαρώσει όλη την εικόνα.Το τελικό αποτέλεσμα της διαδικασίας αυτής παράγει των πίνακα χαρακτηριστικών(feature map) για κάθε κανάλι της RGB εικόνας.



Σχήμα 2.10: Συνέλιξη μεταξύ εικόνας εισόδου και φίλτρου (Πηγή: IBM)

Σε περίπτωση που το φίλτρο δεν εφαρμόζει ακριβώς στην εικόνα εισόδου χρησιμοποιείται η τεχνική του zero padding η οποία θέτει όλα τα στοιχεία που βρίσκονται εκτός της εικόνας σε 0 και παράγει μια έξοδο ίδιου μεγέθους με την εικόνα εισόδου πριν το padding.[2.11](#)



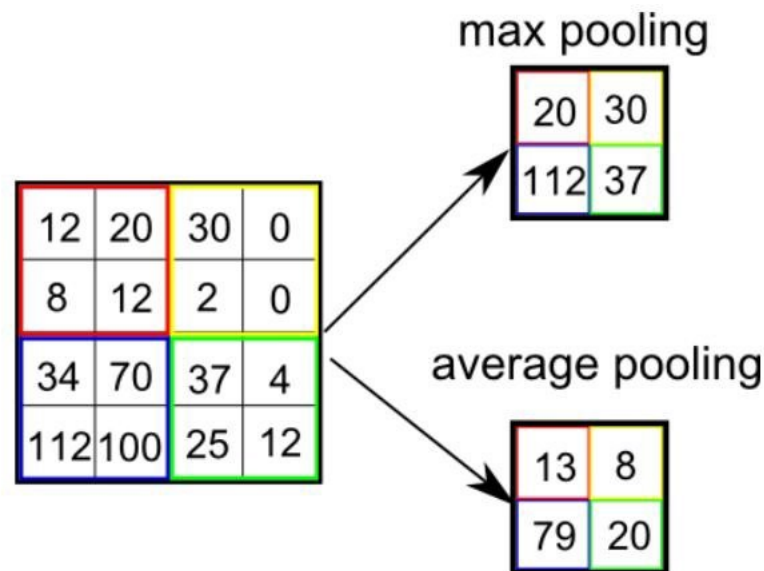
Σχήμα 2.11: Παράδειγμα zero padding(Πηγή: Medium)

Οι τιμές των βαρών του φίλτρου παραμένουν σταθερές καθώς αυτό σαρώνει την εικόνα. Ωστόσο, προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης μέσω της διαδικασίας της ανάστροφης διάδοσης.

2.6.2 Συγκεντρωτικό στρώμα

Το Συγκεντρωτικό στρώμα (Pooling layer) εφαρμόζει υποδειγματοληψία μειώνοντας τη χωρική του πίνακα χαρακτηριστικών. Λειτουργεί με τρόπο παρόμοιο με του συνελικτικού στρώματος, δηλαδή ένα φίλτρο σαρώνει τα εικονοστοιχεία της εισόδου με

τη διαφορά ότι το φίλτρο δεν έχει βάρη αλλά εφαρμόζει κάποια συνάρτηση στις τιμές του patch που επεξεργάζεται και δημιουργεί έναν πίνακα εξόδου. Υπάρχουν δύο είδη pooling. Το Max pooling, όπου για κάθε patch επιλέγεται το στοιχείο με τη μεγαλύτερη τιμή και το Average pooling, όπου υπολογίζει τη μέση τιμή των στοιχείων του εκάστοτε patch.



Σχήμα 2.12: Max/Average Pooling - (Πηγή: Medium)

2.6.3 Πλήρες συνδεδεμένο στρώμα

Στο Πλήρες συνδεδεμένο στρώμα (Fully-connected layer) γίνεται η ταξινόμηση με βάση τον πίνακα χαρακτηριστικών που εξάχθηκαν στο προηγούμενο στρώμα (pooling layer). Ο πίνακας χαρακτηριστικών που εξάχθηκε μετατρέπεται σε έναν μονοδιάστατο πίνακα. Το συνελικτικό και συγκεντρωτικό στρώμα χρησιμοποιούν τον Rectified Linear Unit (ReLU) μετασχηματισμό στον πίνακα χαρακτηριστικών, προσδίδοντας μη γραμμικότητα στο μοντέλο, ενώ το πλήρες συνδεδεμένο στρώμα εκμεταλλεύεται μια softmax activation function για να ταξινομήσει τις εισόδους κατάλληλα, προσδίδοντας μια πιθανότητα από 0 μέχρι 1 σε κάθε κλάση που ανιχνεύεται στην εικόνα με το άθροισμα των πιθανοτήτων αυτών να αθροίζει στο 1. [23]

2.7 Αλγόριθμοι ανίχνευσης αντικειμένων

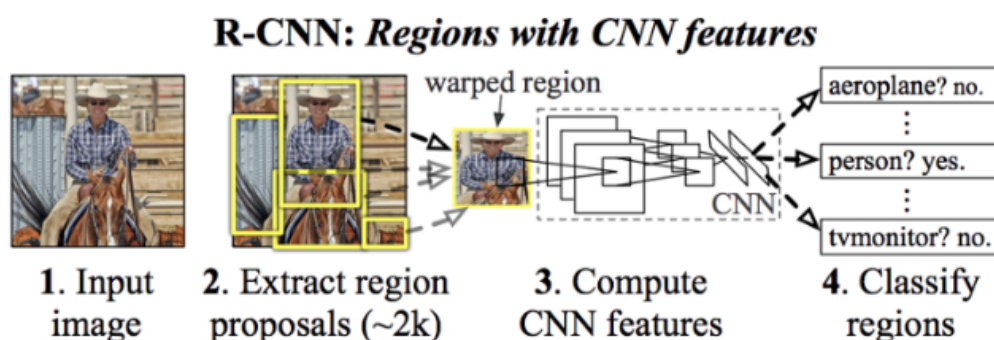
Η Μηχανική Όραση είναι αδιαμφισβήτητα ένας τομέας μεγάλου ενδιαφέροντος τα τελευταία χρόνια. Η ανίχνευση αντικειμένων (Object Detection) είναι σημαντικό κεφάλαιο της Μηχανικής Όρασης. Η εξέλιξης των μονάδων των ηλεκτρονικών υπολογιστών όπως είναι οι επεξεργαστές και οι κάρτες γραφικών επέτρεψαν στους επιστήμονες να αναπτύξουν και να εξελίσουν αλγόριθμους ανίχνευσης αντικειμένων. Αυτοί οι αλγόριθμοι χρησιμοποιούνται σε πλήθος εφαρμογών, όπως σε κλειστά συστήματα παρακολούθησης, στην ανίχνευση οχημάτων, καθώς και στην παρακολούθηση δημόσιων χώρων. Η διαφορά μεταξύ των αλγορίθμων ανίχνευσης αντικειμένων με τους αλγόριθμους κατηγοριοποίησης έγκειται στο γεγονός ότι στους αλγόριθμους ανίχνευσης αντικειμένων σχεδιάζουμε ένα πλαίσιο οριοθέτησης στα αντικείμενα που μας ενδιαφέρουν. Παρακάτω αναλύονται σημαντικοί αλγόριθμοι ανίχνευσης αντικειμένων.

2.7.1 R-CNN

Ο R-CNN ήταν ένας από τα πρώτα μοντέλα που χρησιμοποιεί συνελικτικά νευρωνικά δίκτυα(CNN) για την ανίχνευση αντικειμένων. Σκοπός του είναι να πάρει μια εικόνα και να αναγνωρίσει σωστά τα κύρια αντικείμενα σχεδιάζοντας πλαίσια οριοθέτησης γύρω από αυτά. Ο R-CNN προτείνει ένα πλήθος περιοχών(≈ 2000) στην εικόνα και ελέγχει αν αντιστοιχεί σε κάποιο αντικείμενο. Οι προτάσεις περιοχών γίνονται μέσω της διαδικασίας της επιλεκτικής αναζήτησης(Selective Search). Ο αλγόριθμος επιλεκτικής αναζήτησης ελέγχει την εικόνα μέσω παραθύρων διαφορετικού μεγέθους και για κάθε μέγεθος προσπαθεί να ομαδοποιήσει τα γειτονικά εικονοστοιχεία με βάση την υφή, το χρώμα και την ένταση για την αναγνώριση αντικειμένων. Μετά την εξαγωγή των περιοχών απόφασης ο R-CNN στοιβάζει κάθε περιοχή απόφασης σε ένα διάνυσμα προκαθορισμένου μεγέθους και τις δίνει σε ένα (CNN). Στο στρώμα εξόδου του CNN ο R-CNN προσθέτει ένα Support Vector Machine (SVM) το οποίο ταξινομεί αν είναι αντικείμενο και αν ναι ποιο αντικείμενο είναι.

Ο R-CNN λειτουργεί αρκετά καλά με βέλτιστη ακρίβεια που φτάνει mAP(mean average precision) 66% στο dataset PASCAL VOC 2007 ενώ στο dataset PASCAL VOC 2012 το mAP είναι 62%, αλλά είναι αρκετά αργός. Αυτό γιατί απαιτεί την προώθηση

κάθε μιας απο τις προτεινόμενες περιοχές για κάθε εικόνα στο CNN.Επίσης, πρέπει να εκπαιδεύσει τρία διαφορετικά μοντέλα.Το CNN για την εξαγωγή των χαρακτηριστικών της εικόνας, τον ταξινομέα που προβλέπει την κλάση και το μοντέλο παλινδρόμησης για την ακριβή εφαρμογή των πλαισίων οριοθέτησης στο αντικείμενα.Η ανίχνευση απαιτεί 47 δευτερόλεπτα για κάθε εικόνα. Γεγονός που υποδηλώνει ότι δεν μπορεί να πραγματοποιήσει προβλέψεις σε πραγματικό χρόνο.[24]

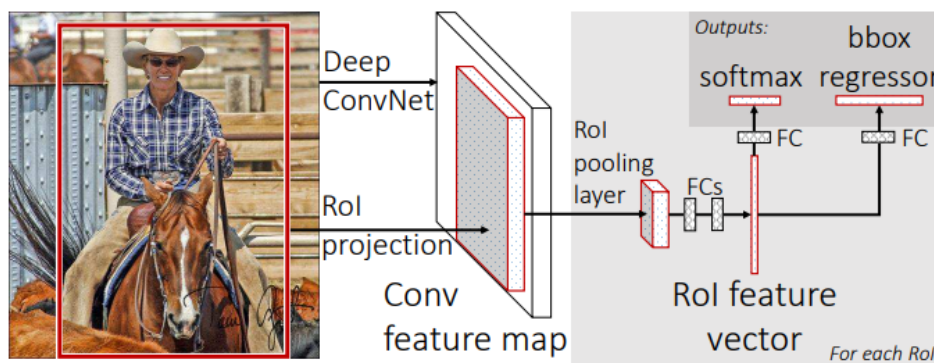


Σχήμα 2.13: R-CNN (Πηγή: <https://arxiv.org/pdf/1311.2524.pdf>).

2.7.2 Fast R-CNN

Ο Fast R-CNN δέχεται ως είσοδο μια εικόνα και ένα σύνολο προτεινόμενων περιοχών.Το δίκτυο αρχικά επεξεργάζεται ολόκληρη την εικόνα με αρκετά συνελικτικά και max pooling στρώματα για την εξαγωγή του πίνακα χαρακτηριστικών (feature map).Έπειτα για κάθε περιοχή που προτάθηκε από τον αλγόριθμο Selective Search, ένα συγκεντρωτικό στρώμα περιοχής ενδιαφέροντος(region of interest (RoI) pooling layer) εξάγει ένα διάνυσμα χαρακτηριστικών σταθερού μεγέθους από τον πίνακα χαρακτηριστικών(feature map).Κάθε διάνυσμα χαρακτηριστικών τροφοδοτείται σε μια ακολουθία από πλήρες συνδεδεμένα στρώματα(Fully-Connected Layers) όπου τελικά διακλαδίζονται σε δυο στρώματα εξόδου, ένα που παράγει τη softmax πιθανότητα για τις κλάσεις των αντικειμένων και ένα στρώμα που έχει ως έξοδο τέσσερις πραγματικές τιμές για κάθε κλάση αντικειμένου.Κάθε τετράδα εξόδου αντιστοιχεί στην θέση πλαισίου οριοθέτησης(bounding box), συντεταγμένες πάνω αριστεράς περιοχής, ύψος και πλάτος για μία από τις κλάσεις.Ο Fast R-CNN εκπαιδεύει 9 φορές γρηγορότερα το

CNN VGG16 από τον R-CNN ενώ είναι 213 φορές γρηγορότερος στον χρόνο δοκιμών. Επιπλέον, η ανίχνευση απαιτεί 2 δευτερόλεπτα για κάθε εικόνα με το CNN VGG16 στην κάρτα γραφικών(GPU). Η ακρίβεια του Fast R-CNN στο dataset PASCAL VOC 2007 έχει mAP 66.9% ενώ dataset PASCAL VOC 2012 έχει mAP 66%. [25]

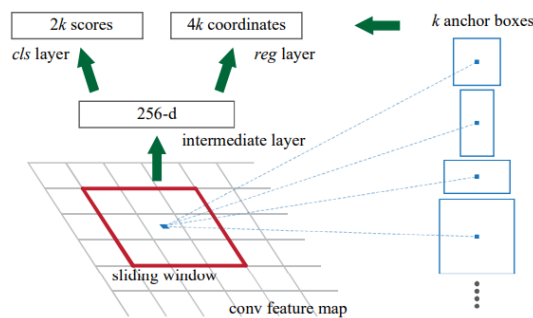


Σχήμα 2.14: Fast R-CNN (Πηγή: <https://arxiv.org/pdf/1504.08083.pdf>)

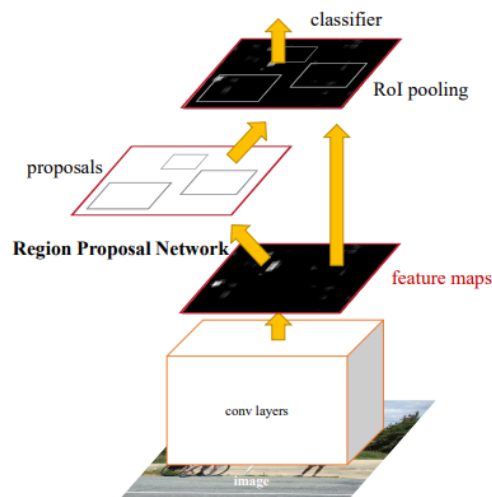
2.7.3 Faster R-CNN

Στον Fast R-CNN οι προτάσεις περιοχών γινόντουσαν από τον αλγόριθμο επιλεκτικής αναζήτησης(Selectice Search), μια αρκετά αργή διεργασία που αποδείχτηκε πως ήταν ο λόγος που προκαλούσε συμφόρηση στην όλη διαδικασία. Ο Faster R-CNN βρήκε τρόπο να πραγματοποιεί τις προτάσεις περιοχών με πολύ μικρό κόστος χρόνου. Η παρατήρηση ήταν ότι η περιοχή προτάσεων εξαρτάται από τον πίνακα χαρακτηριστικών της εικόνας που είχε ήδη υπολογιστεί από το CNN. Συνεπώς, γιατί να μη χρησιμοποιηθούν ξανά τα αποτελέσματα του CNN για τις προτάσεις περιοχών αντί να εκτελεστεί ξεχωριστά Selectice Search; Πράγματι αυτό ακριβώς κατάφερε η ομάδα του Faster R-CNN. Σε αυτό το μοντέλο χρησιμοποιείται ένα CNN για την πρόταση περιοχών και την ταξινόμηση. Με αυτό τον τρόπο εκπαιδεύεται μόνο ένα CNN. Οι προτάσεις περιοχών πραγματοποιούνται με την προσθήκη ενός πλήρους συνελικτικού δικτύου πάνω από τα χαρακτηριστικά που δημιουργεί το CNN δημιουργώντας το Region Proposal Network(RPN). Το δίκτυο RPN δέχεται ως είσοδο τον πίνακα χαρακτηριστικών του CNN και εξάγει πλαίσια αγκύρωσης (anchor boxes), ως προτάσεις περιοχών, μέσω ενός συρόμενου παραθύρου, που εφαρμόζεται στον πίνακα χαρακτηριστικών. Τέλος, με τη βοήθεια του RoI συγκεντρωτικού στρώματος αναδιαμορφώνονται οι προβλεπόμε-

νες προτάσεις περιοχών και τροφοδοτούνται σε μια ακολουθία πλήρως συνδεδεμένων στρωμάτων, με σκοπό την αναγνώριση και πλαισίωση του αντικειμένου στην εικόνα. Επιπλέον, η ανίχνευση απαιτεί 0.2 δευτερόλεπτα, σε αντίθεση με τον Fast R-CNN που απαιτεί 2 δευτερόλεπτα που χρησιμοποιεί Selective Search. Η ακρίβεια του Fast R-CNN στο dataset PASCAL VOC 2007 έχει mAP 73.2% ενώ dataset PASCAL VOC 2012 έχει mAP 70.4%. [26] [27]



(a) RPN



(b) Faster R-CNN

Figure 2.15: Faster R-CNN (Πηγή: <https://arxiv.org/pdf/1506.01497.pdf>)

2.7.4 YOLO

2.7.4.1 Εισαγωγή στον αλγόριθμο YOLO

Στους παραπάνω αλγόριθμους η ανίχνευση αντικειμένων αποτελούταν από τουλάχιστον δυο στάδια με το πρώτο να αποτελεί την εξαγωγή των περιοχών, χρησιμοποιώντας ένα συρόμενο παράθυρο διαφορετικού μεγέθους, και το δεύτερο να πραγματοποιεί την ταξινόμηση για τον καθορισμό της κλάσης του ανιχνευόμενου αντικειμένου. Το 2015 ο ερευνητής Joseph Redmon και οι συνεργάτες του παρουσίασαν ένα σύστημα ανίχνευσης αντικειμένων που εκτελεί όλα τα απαραίτητα στάδια για την ανίχνευση του αντικειμένου χρησιμοποιώντας μόνο ένα CNN, τον YOLO algorithm (You Only Look Once). Ανάγει την ανίχνευση δεδομένων ως ένα απλό πρόβλημα παλινδρόμησης, απευθείας από τα εικονοστοιχεία της εικόνας σε συντεταγμένες πλαισίων οριοθέτησης και πιθανότητες για τις κλάσεις. Το ενοποιημένο μοντέλο προβλέπει ταυτόχρονα πολλαπλά πλαίσια οριοθέτησης και κλάσεων πιθανοτήτων για τα αντικείμενα που καλύπτονται από τα πλαίσια.

Ο αλγόριθμος YOLO επέφερε εντυπωσιακά αποτελέσματα σε σχέση με τους προηγούμενους αλγόριθμους ανίχνευσης αντικειμένων τόσο στην ταχύτητα όσο και στην ακρίβεια για την ανίχνευση και τον καθορισμό των συντεταγμένων του αντικειμένου, καθώς εκτελεί προβλέψεις σε πραγματικό χρόνο. Τα επόμενα 5 χρόνια ο αλγόριθμος YOLO αναβαθμίστηκε σε πέντε εκδόσεις (συμπεριλαμβανόμενης της πρώτης) με πολλές από τις πιο καινοτόμες ιδέες να έρχονται από ερευνητές της κοινότητας της μηχανικής όρασης. Οι πρώτες τρεις εκδόσεις αναπτύχθηκαν από τον δημιουργό του αλγόριθμου YOLO τον Joseph Redmon. Στις αρχές του 2020, το επίσημο Github του YOLO αναβαθμίστηκε στην έκδοση 4, YOLOv4, που δημοσιεύτηκε από τον Alexey Bochkovskiy, τον Ρώσο προγραμματιστή που δημιούργησε τις πρώτες τρεις εκδόσεις του Yolo βασιζόμενος στο Darknet framework του Joseph Redmon. Έναν μήνα μετά από την κυκλοφορία του YOLOv4, ο ερευνητής Glenn Jocher και το ινστιτούτο ερευνών Ultralytics LLC δημιούργησαν τον αλγόριθμο YOLO στο PyTorch framework και δημοσίευσαν τον YOLOv5 με μικρές διαφορές και βελτιώσεις. Αν και δεν αναπτύχθηκε από τα μέλη της ομάδας, ο YOLOv5 απέδωσε εκπληκτικά αποτελέσματα σε ό,τι αφορά την απόδοση σε σχέση με τις προηγούμενες εκδόσεις.

2.7.4.2 YOLOv1

Η κύρια ιδέα του YOLOv1 είναι να εφαρμόσει ένα πλέγμα κελιών μεγέθους $S \times S$ (7×7 Default) πάνω σε μια εικόνα. Αν το κέντρο του αντικειμένου βρίσκεται εντός ενός κελιού του πλέγματος τότε το κελί αυτό είναι υπεύθυνο για την ανίχνευση του αντικειμένου. [2.16α](#) Επομένως, όλα τα άλλα κελιά αγνοούν την εμφάνιση του αντικειμένου που εμφανίζεται σε πολλαπλά κελιά.

Για να σχεδιάσουμε την ανίχνευση αντικειμένου, κάθε κελί του πλέγματος προβλέπει B πλαίσια οριοθέτησης με τις παραμέτρους τους και τα σκορ σιγουριάς για αυτά τα πλαίσια. Αυτά τα σκορ σιγουριάς υποδεικνύουν την παρουσία ή την απουσία ενός αντικειμένου σε ένα κελί του πλέγματος. Το σκορ σιγουριάς ορίζεται ως:

$$confidenc\ score = p(Object) * IoU_{pred}^{truth}$$

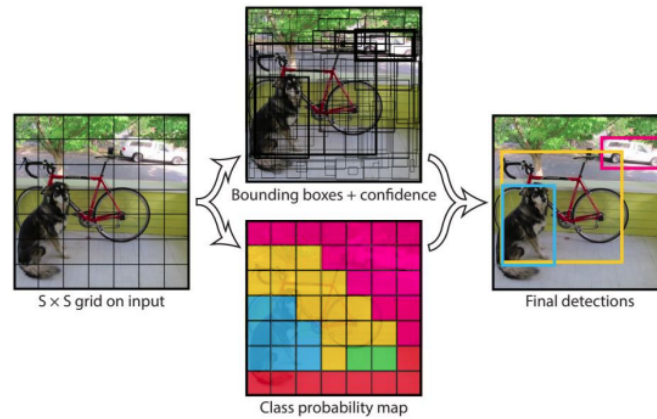
οπού $p(Object)$ η πιθανότητα να υπάρχει ένα αντικείμενο εντός του κελιού και παίρνει τιμές στο εύρος $[0, 1]$, με την τιμή να είναι κοντά στο 0 όταν δεν υπάρχει αντικείμενο στην εικόνα, και IoU_{truth}^{pred} να είναι η τομή στην ένωση του προβλεπόμενου πλαισίου και του πραγματικού πλαισίου. Κάθε πλαίσιο οριοθέτησης αποτελείται από 4 παραμέτρους (x, y, w, h) όπου (x, y) το κέντρο του πλαισίου οριοθέτησης w το πλάτος και h το ύψος. Το μοντέλο έχει πλέγμα κελιών $S \times S$ για μια εικόνα. Κάθε κελί προβλέπει B πλαίσια οριοθέτησης που αποτελούνται από 5 παραμέτρους $(x, y, w, h, confidence)$ και τις πιθανότητες προβλέψεων C για τις κλάσεις. Η έξοδος τους YOLO θα είναι:

$$S \times S(5 \times B + C)$$

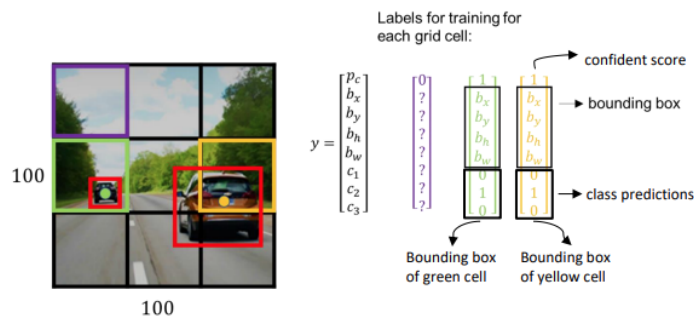
Ο σκοπός του αλγορίθμου YOLO είναι να ανιχνεύσει ένα αντικείμενο προβλέποντας με ακρίβεια το περιεχόμενο του πλαισίου οριοθέτησης του αντικείμενου και εντοπίζοντας το αντικείμενο με βάση τις συντεταγμένες του πλαισίου οριοθέτησης. Συνεπώς, το διάνυσμα των προβλεπόμενων πλαισίων οριοθέτησης αναφέρεται στο διάνυσμα εξόδου \vec{y} ενώ τα πραγματικά πλαίσια οριοθέτησης αναφέρονται στο αντίστοιχο διάνυσμα με ετικέτα y . [2.16β](#) Ο YOLO εφαρμόζει Non-Maximum Suppression (NMS) [2.9](#) για να απορρίψει τα πλαίσια οριοθέτησης που δεν περιέχουν αντικείμενα ή περιέχουν το ίδιο αντικείμενο. Επιλέγοντας μια τιμή κατωφλίου ο NMS αφαιρεί όλα τα πλαίσια οριοθέτησης που έχουν IoU μεγαλύτερο από την τιμή κατωφλίου.

Η αρχιτεκτονική του αλγορίθμου ονομάζεται Darknet, επεξεργάζεται τα χαρακτηριστικά της εικόνας και ακολουθείται από 2 πλήρες συνελικτικά στρώματα για την

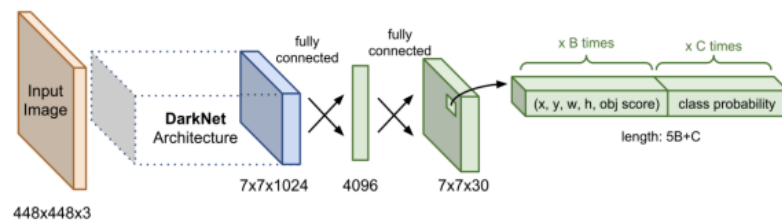
εκτέλεση προβλέψεων για τα πλαίσια οριοθέτησης των αντικείμενα. Το μοντέλο αυτό αξιολογήθηκε στο Pascal VOC dataset με $S = 7, B = 2, C = 20$. Το fast-YOLO model έχει 9 CNN στρώματα στο Darknet και χρησιμοποιείται για απλά σύνολα δεδομένων, ενώ το normal-YOLO model έχει 24 CNN και χρησιμοποιείται για πιο σύνθετα σύνολα. 2.16c.



(a) YOLOv1 (Redmon, et al., 2016)



(b) Εξόδοι YOLOv1(datahacker.rs, 2018)



(c) Αρχιτεκτονική YOLOv1[28]

Figure 2.16: YOLOv1

2.7.4.3 YOLOv2

Στην αρχική έκδοση του YOLOv1, τα πρώτα 20 συνελικτικά στρώματα χρησιμοποιούνται για την εκπαίδευση του εξαγωγέα χαρακτηριστικών με εικόνα εισόδου 224×224 . Τα εναπομείναντα 4 συνελικτικά στρώματα και τα 2 πλήρες συνδεδεμένα προστέθηκαν και η ανάλυση της εικόνας εισόδου αυξήθηκε στα 448×448 για να χρησιμοποιηθούν από τον ανιχνευτή αντικειμένων. Αντίθετα στην έκδοση YOLOv2 αφού ολοκληρωθεί η φάση της εκπαίδευσης του εξαγωγέα χαρακτηριστικών με την εικόνα εισόδου 224×224 , το μοντέλο συνεχίζει την εκπαίδευση του εξαγωγέα χαρακτηριστικών για 10 ακόμα epochs με εικόνα εισόδου 448×448 πρώτου χρησιμοποιηθεί η αρχιτεκτονική για την εκπαίδευση του ανιχνευτή αντικειμένων. Αυτό βοηθάει το μοντέλο να προσαρμοστεί στη μεγάλη ανάλυση 448×448 αντί να αυξήσει ξαφνικά το μέγεθος της εικόνας όταν η φάση εκπαίδευσης του εξαγωγέα χαρακτηριστικών προχωρήσει στη φάση εκπαίδευσης του ανιχνευτή αντικειμένων. Αυτό το δίκτυο υψηλής ποιότητας ταξινόμησης αύξησε το mAP κατά 4%.

Στον YOLOv2, ο δημιουργός εισήγαγε μια αρχιτεκτονική πλαισίων αγκίστρωσης για την πρόβλεψη των πλαισίων οριοθέτησης αντί να χρησιμοποιεί πλήρες συνδεδεμένα συνελικτικά στρώματα όπως έκανε ο YOLOv1. Τα πλαίσια αγκίστρωσης είναι μια λίστα από προκαθορισμένα πλαίσια που ταιριάζουν καλύτερα με το επιθυμητό αντικείμενο. Με τα πλαίσια αγκίστρωσης ο YOLOv2 αύξησε το mAP κατά περίπου 5%.

2.7.4.4 YOLOv3

Για τα deep neural networks περισσότερα στρώματα σημαίνει μεγαλύτερη ακρίβεια. Ο YOLOv2 χρησιμοποιεί περισσότερα στρώματα από τον YOLOv1 αλλά η εικόνα εισόδου υποδειγματοληπτείται όταν προωθείται στα πιο βαθιά στρώματα οδηγώντας σε απώλεια των λεπτομερειών των χαρακτηριστικών. Αυτή η δυσκολία του YOLOv2 τον εμποδίζει στο να ανιχνεύσει μικρά αντικείμενα. Με το ResNet παραλείπονται οι συνδέσεις για να προωθηθούν οι ενεργοποιήσεις στα πιο βαθιά στρώματα χωρίς να εξαφανίζονται σταδιακά τα χαρακτηριστικά. Ο YOLOv3 ήρθε με καλύτερη αρχιτεκτονική με τη χρήση του εξαγωγέα χαρακτηριστικών να είναι ο συνδυασμός των YOLOv2, Darknet-53 (53 convolutional layers), και Residual Networks (ResNet). Η δομή του δικτύου είναι 1×1 ακολουθούμενο από 3×3 συνελικτικά στρώματα μέσα σε κάθε residual block 2.17 συν μια παράλειψη σύνδεσης. Χάρη στο residual block του ResNet τα επικαλυπτόμενα

στρώματα δε θα μειώσουν την απόδοση του δικτύου. Επιπρόσθετα, τα βαθύτερα στρώματα λαμβάνουν περισσότερες λεπτομέρειες άμεσα από ότι τα αρχικά στρώματα και έτσι δε χάνεται το πλήθος των λεπτομερειών των χαρακτηριστικών. Ο YOLOv3 έχει συνολικά 106 πλήρες συνδεδεμένα συνελικτικά στρώματα.

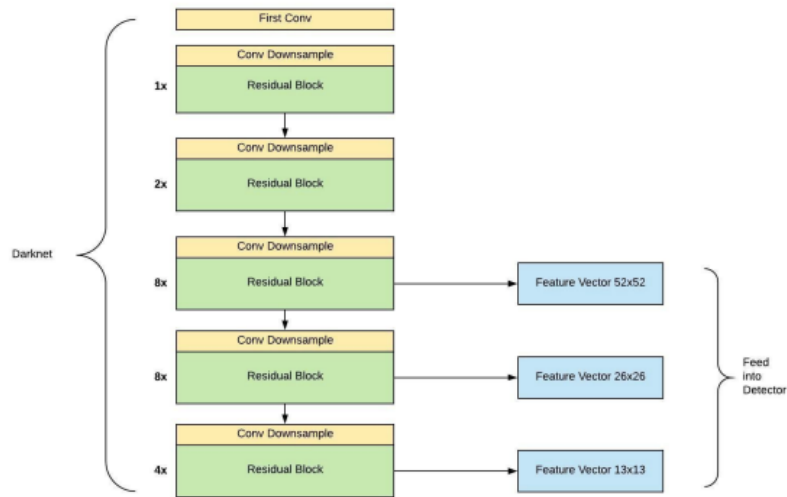
Το πιο αξιοσημείωτο χαρακτηριστικό του YOLOv3 είναι ότι πραγματοποιεί ανιχνεύσεις σε τρεις διαφορετικές κλίμακες. Τα τρία residual blocks χρησιμοποιούνται για ανιχνευτές τριών διαφορετικών κλιμάκων. Πιο συγκεκριμένα ο YOLOv3 κάνει προβλέψεις στο στρώμα 82,94 και 106 με βήμα εικονοστοιχείων (stride) 32,16 και 8 αντίστοιχα 2.18β. Ο YOLOv3 όπως και ο YOLOv2 σε κάθε πλέγμα κελιών ανιχνεύει πολλά αντικείμενα ταυτόχρονα. Τα πλαίσια οριοθέτησης θα ανιχνεύσουν διαφορετικά αντικείμενα ακόμα και αν βρίσκονται στο ίδιο πλέγμα κελιών για αυτό και τα πλαίσια οριοθέτησης έχουν το δικό τους σύνολο από πιθανότητες προβλέψεων κλάσεων αντί να τις μοιράζονται. Η έξοδος του YOLOv3 είναι:

$$S \times S \times (B \times (5 + C)).$$

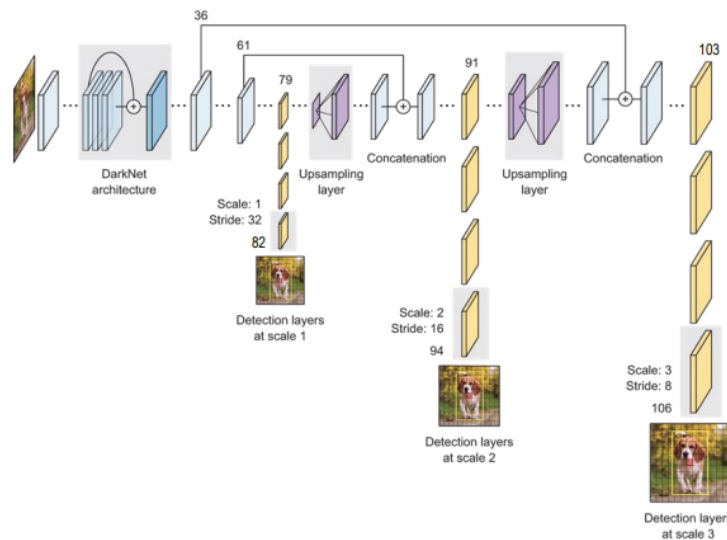
Οι ανιχνεύσεις με στρώματα διαφορετικής κλίμακας επιλύει το πρόβλημα του YOLOv2 στην ανίχνευση μικρών αντικειμένων. Με το στρώμα 52×52 να είναι υπεύθυνο για την ανίχνευση μικρών αντικειμένων ενώ το στρώμα 13×13 για την ανίχνευση μεγάλων αντικειμένων.

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 2.17: Darknet-53 architecture with 5 residual blocks (square box) containing bottle neck structure (1x1 followed by 3x3 convolutional layers). (Yanjia LiYanjia Li, 2019)



(a) Multi-scale detector was appended aside network to make detection 3 times in 3 different scales. (Yanjia LiYanjia Li, 2019)



(b) YOLOv3 network architecture(Kathuria, 2018)

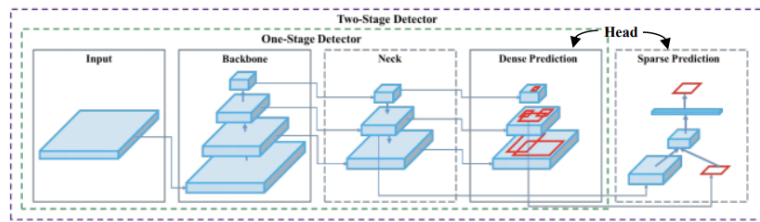
Figure 2.18: YOLOv3

2.7.4.5 YOLOv4

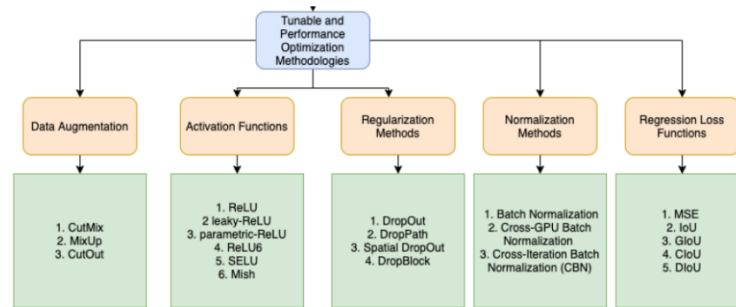
Υπάρχουν αλγόριθμοι ανίχνευσης αντικειμένων ενός βήματος καθώς και δύο βημάτων. Τα κοινά τους σημεία βρίσκονται στο ότι τα χαρακτηριστικά της εικόνας εισόδου θα συμπιεστούν στον εξαγωγέα χαρακτηριστικών (Backbone) και στη συνέχεια θα προωθηθούν στον ανιχνευτή αντικειμένων (συμπεριλαμβανομένου του Detection Neck και

Detection Head). Το Detection Neck λειτουργεί ως αθροιστής των χαρακτηριστικών ο οποίος συνδυάζει τα χαρακτηρίστηκα που σχηματίστηκαν στο Backbone για να προετοιμαστεί για το βήμα της ανίχνευσης στο Detection Head. Η διαφορά εμφανίζεται στο Head που είναι υπεύθυνο για την ανίχνευση συμπεριλαμβανομένου της τοποθέτησης και ταξινόμησης του κάθε πλαισίου οριοθέτησης. Οι δυο βημάτων ανιχνευτές υλοποιούν τις δυο αυτές εργασίες ξεχωριστά και συνδυάζουν το αποτέλεσμα τους αργότερα (Sparse Detection), ενώ οι ενός βήματος ανιχνευτές τις υλοποιούν ταυτόχρονα (Dense Detection). Ο YOLO είναι ενός σταδίου αλγόριθμος. Οι δημιουργοί του YOLOv4 εφάρμοσαν πολλά πειράματα με πολλές καινοτόμες ιδέες της μηχανικής όρασης. [2.19α](#)

Ο YOLOv4 χρησιμοποιεί ως Backbone το CSPDarknet53 που προέρχεται από την αρχιτεκτονική DenseNet, που σχεδιάστηκε για να συνδέει τα στρώματα σε ένα πολύ βαθύ νευρωνικό δίκτυο με σκοπό την εξάλειψη της απώλειας χαρακτηριστικών όπως έκανε και το ResNet. Ο YOLOv4 χρησιμοποιεί και CSP (Cross Stage Partial) που λειτουργεί όμοια με το DenseNet με τη διαφορά του ότι αντί να χρησιμοποιήσει ολόκληρο τον πίνακα χαρακτηριστικών (feature map) τον χωρίζει σε 2 ποσότητες. Μια θα προωθηθεί στο dense block και μια θα σταλεί στο επόμενο στάδιο χωρίς επεξεργασία. Συνδυάζοντάς τις ιδέες αυτών των αρχιτεκτονικών στον YOLOv4 αντικαταστάθηκαν τα residual blocks που υπήρχαν στον YOLOv3 από dense blocks. Το CSP διατηρεί χαρακτηριστικά κατά τη μετάδοση, προτρέποντας στο δίκτυο να επαναχρησιμοποιήσει χαρακτηριστικά και μειώνει τον αριθμό των παραμέτρων του δικτύου, βοηθώντας στην διατήρηση των λεπτομερειών των χαρακτηριστικών για προώθηση σε βαθύτερα στρώματα πιο αποδοτικά. Στο Neck ο YOLOv4 χρησιμοποιεί ένα τροποποιημένο Path Aggregation Network (PAN) ενώ το Head είναι παρόμοιο με αυτό του YOLOv3. Πραγματοποιεί δηλαδή προβλέψεις με βάση ένα διάνυσμα που περιέχει τις συντεταγμένες του πλαισίου οριοθέτησης (center, height, width), τα σκορ σιγουριάς και τις πιθανότητες των κλάσεων. Ο YOLOv4 για τη βελτίωση της απόδοσης του χρησιμοποιεί το Bag of Freebies όπου το Bag αναφέρεται στο σύνολο των μεθόδων βελτίωσης και το Freebis στις βελτιώσεις στο Bag που βοηθούν στην αύξηση της απόδοσης και της ακρίβειας του μοντέλου χωρίς να κοστίζουν στο hardware. [2.19β](#).



(a) Two concepts of architectural object detection. (Solawetz, 2020)



(b) Bag of Freebies(Kanjee, 2020)

Figure 2.19: Στάδια Ανίχνευσης - Bag of Freebies YOLOv4

2.7.4.6 YOLOv5

Ο YOLOv5 κυκλοφόρησε ένα μήνα μετά τον YOLOv4. Η αρχιτεκτονική τους είναι όμοια και το όνομα YOLOv5 (5η έκδοση του YOLO) προκάλεσε δυσάρεσκα σε πολλούς καθώς δεν εμπεριέχει πλήθος καθηλωτικών βελτιώσεων σε σχέση με τον YOLOv4. Ωστόσο, ο YOLOv5 είναι γραμμένος σε γλώσσα προγραμματισμού Python αντί για C, που ήταν γραμμένες οι προηγούμενες εκδόσεις. Αυτό κάνει την εγκατάσταση και την ενσωμάτωση του στις συσκευές δικτύου προγραμμάτων ευκολότερη. Επιπρόσθετα η κοινότητά PyTorch είναι μεγαλύτερη από αυτή του Darknet που σημαίνει ότι το PyTorch θα αποκτήσει περισσότερους συνεισφορείς και μεγαλύτερη ανάπτυξη στο μέλλον. Το γεγονός ότι ο YOLOv5 και ο YOLOv4 είναι γραμμένα σε διαφορετικές γλώσσες προγραμματισμού σε δυο διαφορετικά framework καθιστά τη σύγκρισή τους δύσκολη. Αργότερα βέβαια ο YOLOv5 αποδείχτηκε ότι είχε υψηλότερη επίδοση από τον YOLOv4 κάτω από συγκεκριμένες συνθήκες και κέρδισε την εμπιστοσύνη της κοινότητας της μηχανικής όρασης. Επίσης, το .cfg αρχείο του YOLOv4 αντικαταστήθηκε από το .yaml στον YOLOv5 για το file configuration. [28]

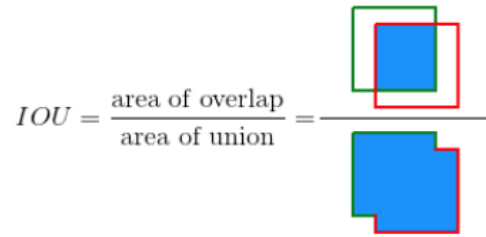
2.8 Αξιολόγηση απόδοσης αλγορίθμων ανίχνευσης αντικειμένων

Ανάμεσα στα διαφορετικά σύνολα δεδομένων που χρησιμοποιούνται για την ανίχνευση δεδομένων και την επιστημονική κοινότητα, οι πιο διαδεδομένες μετρικές που χρησιμοποιούνται για να μετρήσουν την ακρίβεια μιας ανίχνευσης είναι η AP (Average Precision). Προτού εξετάσουμε τις παραλλαγές της AP πρέπει πρώτα να εξετάσουμε κάποιες έννοιες αυτών. Οι πιο σημαντικές είναι:

- True positive (TP): Μια σωστή ανίχνευση ενός πραγματικού πλαισίου οριοθέτησης.
- False positive (FP): Μια εσφαλμένη ανίχνευση ενός αντικειμένου που δεν υπάρχει ή μια άστοχη ανίχνευση ενός αντικειμένου που υπάρχει.
- False negative (FN): Μη ανίχνευση του πραγματικού πλαισίου οριοθέτησης.

Είναι σημαντικό να σημειώσουμε ότι στην ανίχνευση αντικειμένων δεν υπάρχει η έννοια του True Negative (TN) καθώς υπάρχουν άπειρα πλαίσια οριοθέτησης που δε θα έπρεπε να ανιχνευθούν σε μια δοθήσα εικόνα. Οι παραπάνω ορισμοί βοηθούν στη διαπίστωση του τι σημαίνει σωστή ανίχνευση και τι λάθος. Ένας απλός τρόπος για να το κάνουμε αυτό είναι να χρησιμοποιήσουμε την τομή της ένωσης (Intersection over Union). Η IoU είναι μια μετρική βασισμένη στον δείκτη Jaccard, ένας συντελεστής ομοιότητας δυο συνόλων δεδομένων. Από τη σκοπιά της ανίχνευσης δεδομένων, η IoU μετράει την τομή μεταξύ του προβλεπόμενου πλαισίου οριοθέτησης B_p και του πραγματικού B_{gt} διαιρούμενο από την περιοχή της ένωσης αυτών.

$$J(B_p, B_{gt}) = IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$



Σχήμα 2.20: IoU[1]

Συγκρίνοντας το IoU με ένα κατώφλι (threshold) t , μπορούμε να ταξινομήσουμε μια ανίχνευση ως σωστή ή εσφαλμένη. Αν $IoU \geq t$ τότε η ανίχνευση θεωρείται σωστή. Αν $IoU < t$ τότε η ανίχνευση είναι εσφαλμένη. Η εκτίμηση των μεθόδων ανίχνευσης αντικειμένων βασίζεται κυρίως στις έννοιες του Precision(P) και στο Recall(R).

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}}$$

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}}$$

Precision είναι η ικανότητα ενός μοντέλου να αναγνωρίζει μόνο σχετικά αντικείμενα. Είναι το ποσοστό των σωστών θετικών προβλέψεων. Recall είναι η ικανότητα ενός μοντέλου να βρίσκει όλα τα πραγματικά πλαίσια οριοθέτησης. Είναι το ποσοστό των σωστών θετικών προβλέψεων ανάμεσα σε όλα τα πραγματικά πλαίσια οριοθέτησης.

Μπορούμε να δούμε την καμπύλη precision-recall σαν μια ανταλλαγή μεταξύ precision και recall για τις διάφορες τιμές βεβαιότητας που σχετίζονται με το πλαίσιο οριοθέτησης που παράγει ο ανιχνευτής. Αν η βεβαιότητα του ανιχνευτή είναι με τέτοιο τρόπο ώστε το false positive(FP) να είναι χαμηλό τότε το precision θα είναι υψηλό. Ωστόσο, σε αυτή την περίπτωση πολλές θετικές ανιχνεύσεις μπορεί να μη βρεθούν οδηγώντας σε υψηλό false negative(FN) και συνεπώς σε χαμηλό recall. Από την άλλη αν δέχεται περισσότερες θετικές, τότε το recall θα αυξηθεί αλλά το FP θα αυξηθεί και αυτό μειώνοντας το precision. Ωστόσο, ένας καλός ανιχνευτής αντικειμένων θα βρει όλα τα πραγματικά πλαίσια οριοθέτησης ($FN = 0 = \text{high recall}$) όσο αναγνωρίζει μόνο σχετικά αντικείμενα ($FP = 0 = \text{high precision}$). Επομένως ένας ανιχνευτής αντικειμένων μπορεί να θεωρηθεί καλός αν το precision παραμένει υψηλό όσο αυξάνεται το recall που σημαίνει ότι αν η τιμή του κατωφλίου σιγουριάς ποικίλλει τότε το precision και το recall θα παραμείνουν υψηλά. Άρα μια μεγάλη περιοχή κάτω από την καμπύλη ((under the curve area)AUC,) υποδηλώνουν υψηλό precision και recall. Δυστυχώς

στην πράξη η καμπύλη precision-recall μοιάζει συνήθως με μια ζικ-ζακ γραμμή προκαλώντας προκλήσεις σε μια ακριβείς μέτρηση του AUC. Αυτό το πρόβλημα εξάλειφεται με την επεξεργασία της precision-recall καμπύλης για την αφαίρεση του ζικ-ζακ πριν την εκτίμησή της AUC.

Ένας πιθανός τρόπος επεξεργασία για την εξάλειψη του προβλήματος είναι η παρεμβολή σε όλα τα σημεία(all-point interpolation):

$$AP_{all} = \sum_n (R_{n+1} - R_n) P_{interp}(R_{n+1}),$$

με

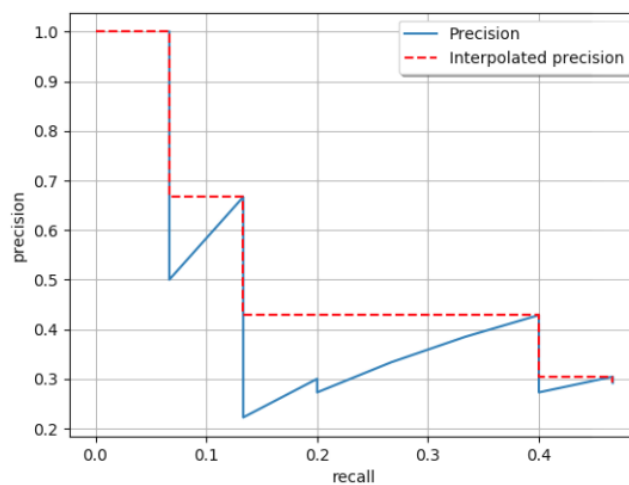
$$P_{interp}(R_{n+1}) = \max_{R \geq R_{n+1}} P(R).$$

Το AP λαμβάνεται παρεμβάλλοντας το precision σε κάθε επίπεδο, παίρνοντας το μέγιστο precision του οποίου η τιμή recall είναι μεγαλύτερη ή ίση απο το R_{n+1} .

Το mAP (mean Average Precision) χρησιμοποιείται για να μετρήσει την ακρίβεια του ανιχνευτή αντικειμένων σε όλες τις κλάσεις ενός συγκεκριμένου συνόλου δεδομένων(dataset). Το mAP είναι απλά το AP πάνω σε όλες τις κλάσεις:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i,$$

με το AP_i να είναι το AP της i -οστής κλάσης και N να είναι ο συνολικός αριθμός των κλάσεων που αξιολογούνται.[1]

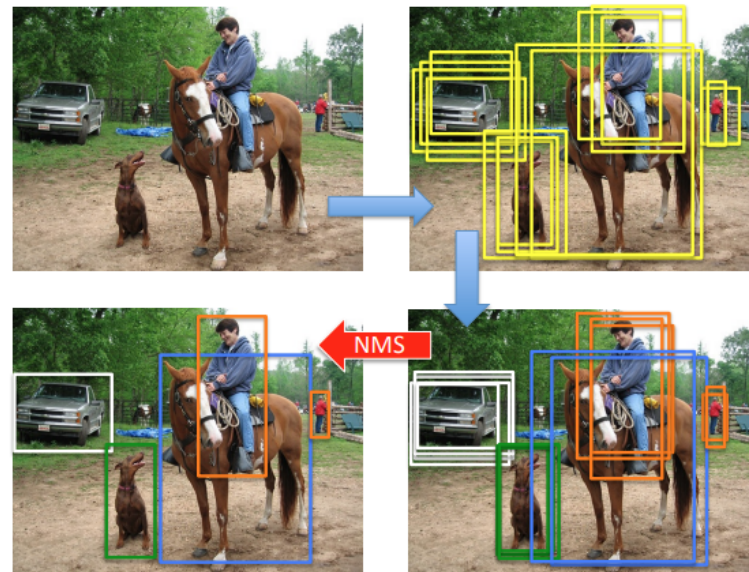


Σχήμα 2.21: Παράδειγμα καμπύλης Preciso-Recall με παρεμβολή[1]

2.9 Non-Maximum Suppression

Οι αλγόριθμοι ανίχνευσης αντικειμένων εκτελούν μια σειρά συνελιζέων σε μια εικόνα χρησιμοποιώντας ένα Συνελικτικό Νευρωνικό Δίκτυο. Το δίκτυο διακλαδίζεται στα δύο με τον ένα κλάδο να παράγει τις περιοχές προτάσεων, ενώ ο άλλος εκτελεί ταξινόμηση και οπισθοδρόμηση με τα χαρακτηριστικά που εξάγονται από τα συγκεντρωτικά στρώματα των προτάσεων περιοχής που παράγονται από το δίκτυο. Το δίκτυο προτάσεων παράγει σκορ ταξινόμησης και συντελεστές οπισθοδρόμησης για τα πλαίσια αγκίστρωσης για πολλές κλίμακες και αναλογίες διαστάσεων που είναι τοποθετημένα σε κάθε εικονοστοιχείο του πίνακα χαρακτηριστικών (feature map). Έπειτα κατατάσσει τα πλαίσια αγκίστρωσης και επιλέγει τα καλύτερα K πλαίσια αγκίστρωσης για τα οποία ο συντελεστής οπισθοδρόμησης του πλαισίου οριοθέτησης προστίθεται για να αποκτήσει τις συντεταγμένες σε επίπεδο εικόνας για κάθε αγκίστρωση. Εφαρμόζεται non-maximum suppression (NMS) στα καλύτερα K πλαίσια αγκίστρωσης και τελικά παράγεται η πρόταση περιοχής.

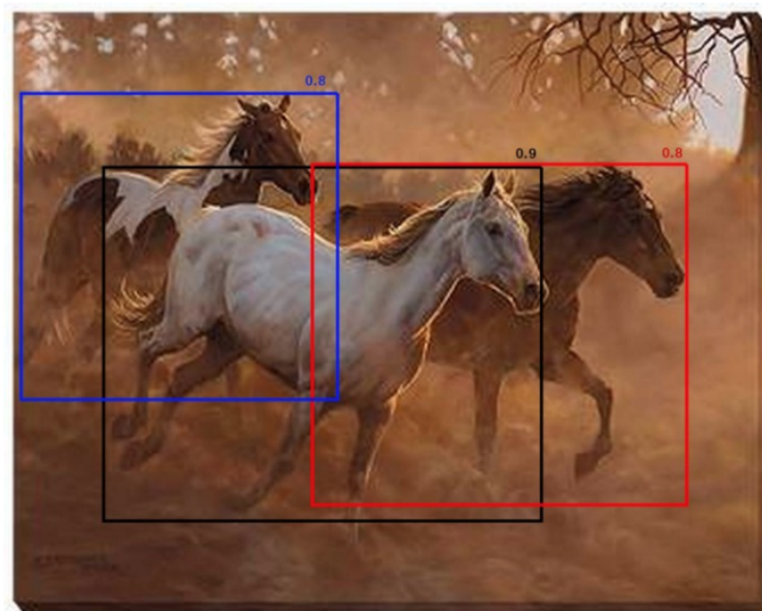
Το δίκτυο ταξινόμησης παράγει σκορ ταξινόμησης για κάθε πρόταση που παράγεται από το δίκτυο. Μιας και δεν υπάρχουν περιορισμοί στο δίκτυο που να το αναγκάζουν να παράγει μια μοναδική περιοχή ενδιαφέροντος (ROI) για ένα αντικείμενο, υπάρχει περίπτωση να γίνουν πολλαπλές προτάσεις που αναφέρονται στο ίδιο αντικείμενο. Για τον λόγο αυτό, πέρα από το πρώτο σωστό πλαίσιο οριοθέτησης, όλα τα άλλα πλαίσια που αναφέρονται στο ίδιο αντικείμενο θα παράγουν false positive. Για την εξάλειψη του προβλήματος εφαρμόζεται non-maximum suppression στα ανιχνευόμενα πλαίσια κάθε κλάσης, δηλαδή διατηρείται το πλαίσιο με το μεγαλύτερο σκορ σιγουριάς για την ύπαρξη αντικειμένου, συγκρίνεται με όλα τα πλαίσια με μικρότερη πιθανότητα και απορρίπτεται αυτά που έχουν μεγαλύτερο IoU [2.20](#) από το οριζόμενο κατώφλι, αφού αν η επικάλυψη είναι μεγάλη τότε το πιο πιθανό είναι να αναφέρονται στο ίδιο αντικείμενο. Η τεχνική του NMS δεν κοστίζει ακριβώς σε υπολογιστική δύναμη. [\[2\]](#)



Σχήμα 2.22: Παράδειγμα NMS[2]

2.9.1 Soft-NMS

Ο NMS είχε το εξής πρόβλημα, αν δυο αντικείμενα βρισκόντουσαν το ένα δίπλα στο άλλο τότε ένα από αυτά θα αφαιρευθεί.



Σχήμα 2.23: Προβλήματα NMS (Πηγή: <https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>)

Όπως φαίνεται στην εικόνα 2.23 ενώ υπάρχουν τρία αντικείμενα στην εικόνα ο αλγόριθ-

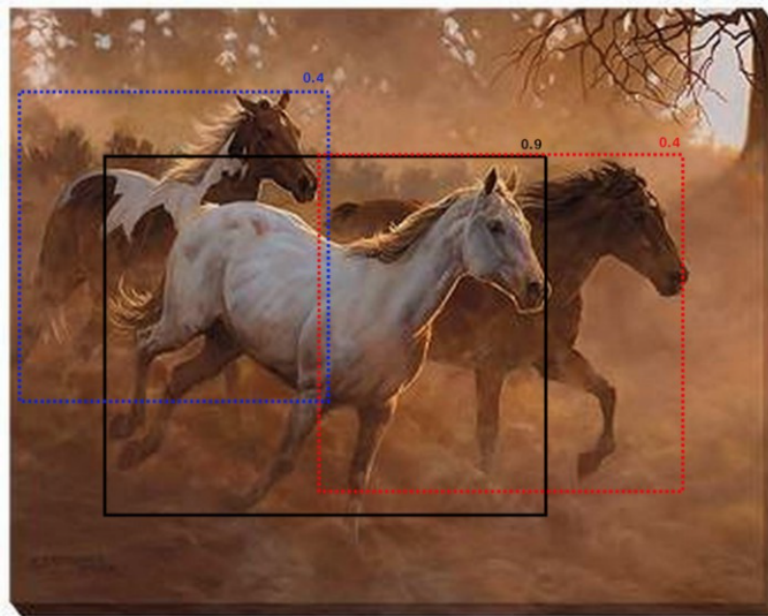
μος NMS θα κρατήσει την ανίχνευση μόνο του ενός, που φαίνεται με μαύρο χρώμα και θα απορρίψει τα άλλα δυο πλαίσια παρόλο που έχουν μεγάλο σκορ.

Αυτό το πρόβλημα έλυσε ο Soft NMS. Στον Soft NMS αντί να αφαιρούνται εντελώς οι προτάσεις με υψηλή IoU και υψηλή σιγουριά, μειώνεται η σιγουριά των προτεινόμενων προτάσεων στην τιμή του IoU ως εξής:

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ 0, & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

Σχήμα 2.24: Soft-NMS[2]

όπου s_i το σκορ πρότασης i , b_i το πλαίσιο οριοθέτησης της πρότασης i , \mathcal{M} το πλαίσιο που αντιστοιχεί στη μέγιστη σιγουριά και N_t το IOU κατώφλι(threshold). Το αποτέλεσμα του Soft NMS στο 2.23 παράδειγμα είναι:



Σχήμα 2.25: Παράδειγμα Soft NMS (Πηγή: <https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>)

2.10 Παρακολούθηση κίνησης αντικειμένων

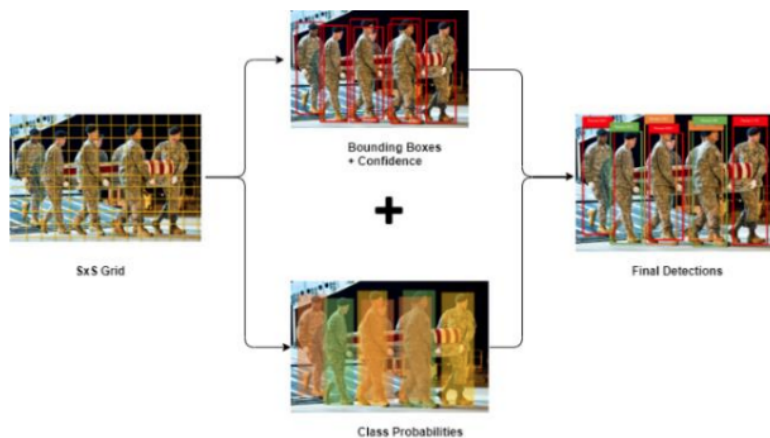
Η οπτική παρακολούθηση κίνησης αντικειμένων(object tracking) αποτελεί έναν από τους πιο σημαντικούς τομείς της μηχανικής όρασης. Η παρακολούθηση κίνησης αντικειμένων είναι η διαδικασία κατά την οποία παρακολουθείται ένα(ή περισσότερα) κινούμενο αντικείμενο σε βάθος χρόνου. Ο σκοπός της οπτικής παρακολούθησης κίνησης αντικειμένων σε ένα συνεχόμενο βίντεο εικόνων είναι να ανιχνεύσει ή να συνδέσει τα υπο παρακολούθηση αντικείμενα. Οι προκλήσεις που αντιμετωπίζονται σε αυτού του είδους τα προβλήματα είναι αρκετές. Μια από αυτές είναι όταν το αντικείμενο κινείται γρηγορότερα από τον ρυθμό των καρέ του βίντεο και όταν το υπο παρακολούθηση αντικείμενο αλλάζει προσανατολισμό στη διάρκεια του χρόνου. Επίσης, το αντικείμενο μπορεί να είναι παραμορφωμένο, θολό ή να κρύβεται από κάποιο άλλο αντικείμενο. Η παρακολούθηση αντικειμένων μπορεί να είναι ενός αντικειμένου(single object tracking(SOT)), όπου παρακολουθείται μόνο ένα αντικείμενο ακόμα και όταν υπάρχουν πολλά αντικείμενα στην εικόνα καθώς και multiple object tracking(MOT), όπου παρακολουθούνται πολλαπλά αντικείμενα. Ο YOLO είναι ο γρηγορότερος αλγόριθμος στην ανίχνευση αντικειμένων και αν και δεν είναι πλέον ο πιο ακριβής αλγόριθμος για την ανίχνευση αντικειμένων, όταν χρειάζεται οι ανιχνεύσεις να συμβαίνουν σε πραγματικό χρόνο χωρίς να χάνεται πολύ ακρίβεια, είναι μια πολύ καλή επιλογή. Όπως αναφέρθηκε [2.7.4](#) ο YOLO χρησιμοποιεί ένα συνελικτικό δίκτυο για να προβλέψει πλαίσια οριοθέτησης και να δώσει πιθανότητες στις ανιχνευόμενες κλάσεις.

Οι ιχνηλάτες αντικειμένων μπορούν να κατηγοριοποιηθούν στους TBD (Tracking by Detection) και DFT (Detection-Free Tracking). Ένας ιχνηλάτης αντικειμένων μπορεί να παρακολουθήσει ένα αντικείμενο σε μια σειρά από καρέ. Τέτοια αντικείμενα θα μπορούσαν να είναι ένα όχημα, ένας άνθρωπος ή μια μπάλα. Η διαδικασία ξεκινάει με την αναγνώριση όλων των πιθανών ανιχνεύσεων σε ένα καρέ παρακολούθησης αντικειμένων και αναθέτοντας ένα αναγνωριστικό(ID) σε αυτά. Στα επόμενα καρέ θα επιχειρηθεί να προωθηθεί το ισχύον ID, αν το αντικείμενο απομακρυνθεί από την εικόνα τότε το ID θα αφαιρεθεί. Αν ένα νέο αντικείμενο εμφανιστεί στην εικόνα τότε ένα νέο ID πρέπει να ανατεθεί σε αυτό. Αυτό είναι μια προκλητική διαδικασία καθώς τα αντικείμενα μπορεί να μοιάζουν συνεπώς ο εξαναγκασμός του να αλλάξουμε ένα ID μπορεί να οδηγήσει στην απώλεια κάποιας παρακολούθησης. Απώλεια παρακολούθησης μπορεί να συμβεί και όταν ένα αντικείμενο βρεθεί πίσω από κάποιο άλλο και επανεμφανιστεί σε επόμενο

καρέ.[3]

2.10.1 Αλγόριθμος SORT

Ο SORT είναι ένας online object tracker που λειτουργεί με της αρχή της παρακολούθησης με βάση την ανίχνευση. Αυτή η μέθοδος χρησιμοποιεί έναν αξιόπιστο ανιχνευτή για την ανίχνευση των αντικειμένων και τον αλγόριθμο Hungarian καθώς και το φίλτρο Kalman για την παρακολούθηση των αντικειμένων. Ο SORT παρακολουθεί κάθε ανίχνευση αναθέτοντας ένα μοναδικό αναγνωριστικό (ID) σε κάθε πλαίσιο οριοθέτησης. Όταν το αντικείμενο χανθεί για τους λόγους που αναφέρθηκαν παραπάνω, ο ανιχνευτής αναθέτει ένα νέο ID και ξεκινάει την παρακολούθηση του νέου αντικείμενου.[3]

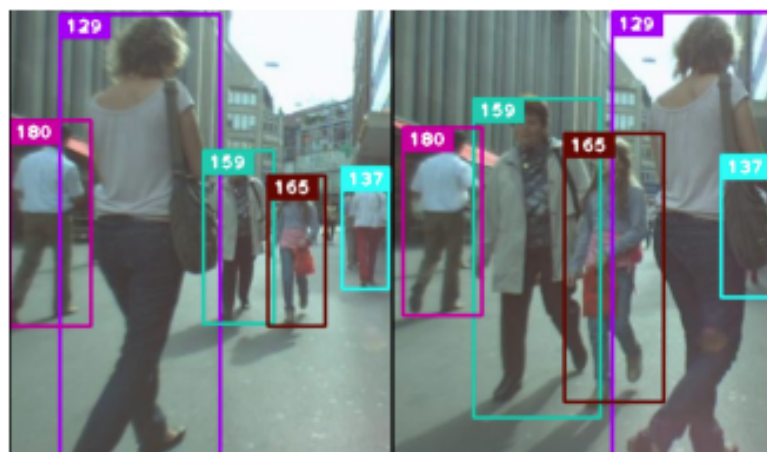


Σχήμα 2.26: Παράδειγμα Αλγόριθμου SORT [3]

2.10.2 Αλγόριθμος DeepSORT

Όπως αναφέρθηκε παραπάνω ο απλός online SORT tracker πραγματικού χρόνου είναι ένας απλός και αποτελεσματικός tracker που χρησιμοποιεί το φίλτρο Kalman και τον Hungarian αλγόριθμο για την παρακολούθηση αντικειμένων. Ωστόσο, η αποτελεσματικότητα του αλγόριθμου SORT μειώνεται όταν το αντικείμενο φράσσεται και αλλάζει η οπτική της κάμερας. Ο δημιουργός του DeepSORT παρουσίασε μια άλλη μετρική απόστασης βασισμένη στην εμφάνιση ενός αντικείμενου για τη βελτίωση του SORT. Ο DeepSORT αλγόριθμος έχει τα εξής χαρακτηριστικά:

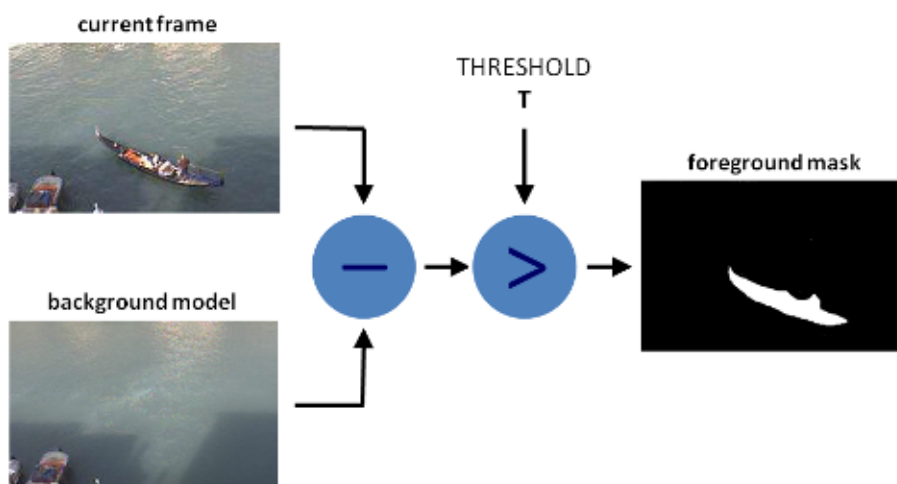
- track estimator που είναι η μετρική τοποθεσίας που χρησιμοποιεί intersection over union (IoU) στην απόσταση μεταξύ του ανιχνευόμενου και του προβλεπόμενου πλαισίου οριοθέτησης.
- appearance descriptor που είναι το νέο στοιχείο του αλγόριθμου DeepSORT. Οι πληροφορίες εμφάνισης εξάγονται από το CNN έτσι ώστε τα χαρακτηριστικά μιας κλάσης αντικειμένων να είναι όμοια, και τα χαρακτηριστικά από τις διαφορετικές κλάσεις αντικειμένων να είναι διαφορετικά στον χώρο των χαρακτηριστικών.
- Συσχέτιση δεδομένων που αναθέτει το ανιχνευόμενο πλαίσιο οριοθέτησης σε ένα αντικείμενο που η κίνηση του ήδη παρακολουθείτε χρησιμοποιώντας τη μετρική τοποθεσίας και εμφάνισης. Κάθε υπάρχον ιχνηλάτιση είναι ένα ID αντικειμένου.
- Track handling για το ιχνηλατούμενο αντικείμενο. Αν η αναγνώριση ενός νέου πλαισίου οριοθέτησης (bounding box) δεν μπορεί να συσχετιστεί με κανένα καρέ παρακολούθηση τότε θα τοποθετηθεί σε δοκιμαστική παρακολούθηση. Ο DeepSORT θα επιχειρήσει να συσχετίσει τη δοκιμαστική παρακολούθηση με άλλες παρακολουθήσεις σε μεταγενέστερα καρέ. Αν η συσχέτιση είναι επιτυχής τότε η παρακολούθηση ενημερώνεται. Διαφορετικά η δοκιμαστική παρακολούθηση θα αφαιρεθεί. [29]



Σχήμα 2.27: Παράδειγμα DeepSORT (Πηγή: <https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/>)

2.11 Αλγόριθμος αφαιρέσεις παρασκήνιου (Background subtraction algorithms)

Κάθε καρέ ενός βίντεο μπορεί να χωριστεί σε δύο διαφορετικές περιοχές: Το προσκήνιο (foreground), που ομαδοποιεί τα εικονοστοιχεία που είναι μέρος του αντικειμένου ενδιαφέροντος και το παρασκήνιο (background), το οποίο ομαδοποιεί όλα τα εικονοστοιχεία που δεν είναι μέρος του αντικειμένου ενδιαφέροντος. Με βάση τον παραπάνω διαχωρισμό, κάθε σύστημα που έχει στόχο την αυτόματη ανίχνευση αντικειμένων, με κατάτμηση και παρακολούθηση, πρέπει να είναι ικανό να διακρίνει αυτές τις δύο περιοχές. Η αφαίρεση παρασκήνιου (background subtraction) είναι μια διαδεδομένη και ευρέως χρησιμοποιούμενη τεχνική για την παραγωγή μιας μάσκας προσκήνιου δηλαδή μιας δυαδικής εικόνας που εμπεριέχει τα εικονοστοιχεία του κινούμενου αντικειμένου της σκηνής. Αφαίρεση φόντου γίνεται όταν τα οπτικά δεδομένα προέρχονται από στατικές κάμερες. Επίσης, με τον όρο δυαδική εικόνα θεωρούμε μια εικόνα που τα εικονοστοιχεία παίρνουν τιμές 0 για τα μαύρα ή 1 για τα άσπρα, με βάση κάποιο κατώφλι αναφοράς. Κατά την αφαίρεση φόντου υπολογίζεται η μάσκα προσκήνιου πραγματοποιώντας μια αφαίρεση μεταξύ του προσκήνιου (foreground) και του παρασκήνιου (background).



Σχήμα 2.28: Παράδειγμα Background Subtraction - Πηγή:

(https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html)

Στη βιβλιοθήκη OpenCV είναι διαθέσιμοι οι αλγόριθμοι MOG, MOG2 και GMG.

2.11.1 GMG

Ο αλγόριθμος GMG, μοντελοποιεί το background συνδυάζοντάς μια Bayesian Inference και το φίλτρο Kalman. Στο πρώτο στάδιο της μεθόδου συγκεντρώνονται, για όλα τα εικονοστοιχεία, οι τιμές βάρων που εξαρτώνται από τη διάρκεια που ένα χρώμα παραμένει σε μια θέση. Για κάθε καρέ, προστίθενται νέες παρατηρήσεις στο μοντέλο, ανανεώνοντας αυτές τις τιμές. Τα χρώματα που παραμένουν σταθερά για ένα καθορισμένο ποσό χρόνου θεωρούνται ως background. Στο δεύτερο στάδιο φιλτράρονται τα εικονοστοιχεία του foreground για τη μείωση του θορύβου που προκλήθηκε στο πρώτο στάδιο.

2.11.2 Mixture of Gaussians (MOG)

Σε αυτή τη μέθοδο, μια μίξη από k Gaussian κατανομές μοντελοποιεί κάθε εικονοστοιχείο του παρασκήνιου(background), με το k να παίρνει τιμές μεταξύ 3 και 5. Ο δημιουργός υποθέτει ότι οι διαφορετικές κατανομές αναπαριστούν κάθε διαφορετικό χρώμα παρασκήνιου και προσκήνιου. Το βάρος της κάθε κατανομής που χρησιμοποιεί το μοντέλο είναι ανάλογο του χρόνου παραμονής του κάθε χρώματος στο εικονοστοιχείο. Για τον λόγο αυτό όταν το βάρος ενός εικονοστοιχείου κατανομής είναι χαμηλό τότε αυτό το εικονοστοιχείο ταξινομείται ως προσκήνιο.

2.11.3 Mixture of Gaussians 2 (MOG2)

Η μέθοδος MOG είχε τον περιορισμό του να χρησιμοποιεί έναν προκαθορισμένο αριθμό κατανομών. Χρησιμοποιώντας μεταβλητό ποσό από Gaussian κατανομές σε κάθε εικονοστοιχείο η μέθοδος MOG2 επιτυγχάνει μια καλύτερη αναπαράσταση της πολυπλοκότητας των χρωμάτων σε κάθε καρέ. Η μέθοδος αυτή χρησιμοποιήθηκε σε επόμενο κεφάλαιο για την ανίχνευση κίνησης των οχημάτων και για αυτό θα αναλυθούν και οι παράμετροι εισόδου της. Στο OpenCV η MOG2 παίρνει τρεις παραμέτρους που μπορεί να αλλάζουν σε κάθε διαφορετικό βίντεο. Οι παράμετροι αυτοί είναι το history που είναι υπεύθυνο για τον αριθμό των καρέ που θα χρησιμοποιήσει η μέθοδος για να μοντελοποιήσει το background, το varThreshold που συσχετίζει την τιμή του βάρους του εικονοστοιχείου στο παρόν καρέ με τις τιμές του μοντέλου και το detectShadows που επιτρέπει ή όχι την ανίχνευση σκιών.[30] [31]

Κεφάλαιο 3

Μέρος Α2

3.1 Εισαγωγή

Στο μέρος αυτό αναφέρονται χρήσιμες πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της μεθόδου εξαγωγής των ταχυτήτων των κινούμενων οχημάτων μέσα από τα βίντεο. Επιπλέον, αναλύεται η διαδικασία εξαγωγής του συνόλου δεδομένων που δίνονται για εκπαίδευση στον YOLOv5s, επεξηγείται η ανάθεση ετικετών και η διαδικασία εκπαίδευσης των προεκπαιδευμένων βαρών του YOLOv5s. Επιπρόσθετα, αναλύονται τα αποτελέσματα της εκπαίδευσης και αξιολογείται η απόδοση αυτών. Στη συνέχεια εξηγούνται οι διαδικασίες ανίχνευσης και παρακολούθησης κίνησης των αντικειμένων στα βίντεο και τέλος η μέθοδος εξαγωγής των ταχυτήτων.

3.2 Χρήσιμες πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν

3.2.1 Μη επανδρωμένα αεροσκάφη(UAVs)

Τα μη επανδρωμένα αεροσκάφη (unmanned aerial vehicles) γνωστά και ως drones ελέγχονται από κάποιον απομακρυσμένο χρήστη ή βασίζονται σε κάποιο λογισμικό για να πραγματοποιούν αυτόνομες πτήσεις. Πολλά UAVs διαθέτουν κάμερες για τη συλλογή οπτικών δεδομένων καθώς και προπέλες για να πραγματοποιούν σταθερές πτήσεις. Είναι χρήσιμα σε πληθώρα εφαρμογών και έχουν χρησιμοποιηθεί από τομείς όπως η παρακολούθηση της κίνησης κυκλοφορίας, βιντεογραφία, έρευνα και διάσωση, συγκινησιμότητες, γεωργία, ιατρική, διανομή προμηθειών καθώς και στη συντήρηση απομακρυσμένων εγκαταστάσεων ανανεώσιμων πηγών ενέργειας. Ανάλογα με τον τύπο τους ταξιδεύουν σε διαφορετικά ύψη και αποστάσεις. Τα κοντινού εύρους (close range UAVs) έχουν εύρος μέχρι περίπου τα 50km, τα μικρού εύρους (short range UAVs) έχουν εύρος μέχρι περίπου τα 150km, τα μεσαίου εύρους (mid range) έχουν εύρος μέχρι περίπου τα 650km ενώ τα αντοχής (endurance UAVs) έχουν εύρος μεγαλύτερο από 650km και πετάνε σε ύψος έως και περίπου 1km.

Τα κύρια εξαρτήματα ενός UAV είναι:

- Χειριστήριο πτήσης
- Μπαταρία
- Κεραία
- GPS
- Κάμερα
- Αισθητήρες
- Επιταχυνσιόμετρο, για τον υπολογισμό της ταχύτητας
- Αλτίμετρο, για τον υπολογισμό του ύψους

Τα UAVs χωρίζονται σε 3 κύριες κατηγορίες:

- Μονού έλικα (Single Rotor)

- Σταθερού πτερυγίου(Fixed Wing)
- Πολλαπλού έλικα(Multi Rotor)



(α') Μονού έλικα - Πηγή: unsplash.com



(β') Σταθερού πτερυγίου - Πηγή: (AgEagle website)



(γ') Πολλαπλού έλικα - Πηγή: (DJI official website)

Σχήμα 3.1: Τύποι μη επανδρωμένων αεροσκαφών

[32] [33]

Το drone που χρησιμοποιήθηκε σε συνεργασία με το εργαστηριακό προσωπικό για τις λήψεις των βίντεο είναι το Phantom 4 Pro V2.0 τα χαρακτηριστήρια του οποίου αναγράφονται λεπτομερώς στο επίσημο website: <https://www.dji.com/gr/phantom-4-pro-v2/specs>.



Σχήμα 3.2: DJI Phantom4 Pro v2.0 (official website)

Ο τύπος αυτού του μη επανδρωμένου αεροσκάφους είναι πολλαπλού έλικα. Η καταγραφή των βίντεο πραγματοποιήθηκε από σταθερή θέση συνεπώς τα χαρακτηριστικά που μας ενδιαφέρουν είναι η δυνατότητα καταγραφής βίντεο 4K : 3840×2160 και η χωρητικότητα μπαταρίας του 5870mAh τύπου LiPo 4S με δυνατότητα διάρκεια πτήσης μέχρι περίπου 30λεπτά καθώς και η αντίσταση σε αέρα ταχύτητας 10m/s. Διαθέτει επίσης και GPS και μπορεί να αποθηκεύσει πληροφορίες σχετικά με το σημείο που πραγματοποιήθηκε κάθε καταγραφή.

3.2.2 Βιβλιοθήκη FFMPEG - Roboflow



(a) FFMPEG logo(official website)



(b) Roboflow logo(official website)

Figure 3.3: FFMPEG - Roboflow

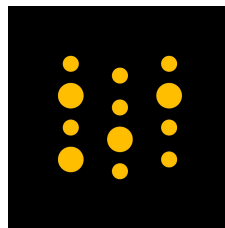
Η FFMPEG είναι λογισμικό ανοικτού κώδικα. Είναι μια βιβλιοθήκη πολυμέσων που μπορεί να κάνει decode, encode, transcode, mux, demux, stream, filter συμβατή με Linux, Windows, MacOS. [34] Χρησιμοποιήθηκε για την εξαγωγή εικόνων από τα βίντεο των καταγραφών που έγιναν με το μη επανδρωμένο αεροσκάφος, για τη δημιουργία του συνόλου δεδομένων(dataset) με βάση το οποίο θα γίνει εκπαίδευση των προ εκπαιδευμένων βαρών(weights) του yolov5s.

Το Roboflow δίνει τη δυνατότητα σε προγραμματιστές να χτίσουν τη δική τους εφαρμογή μηχανικής όρασης. Παρέχει τα απαραίτητα εργαλεία για τη μετατροπή εικόνων σε προσαρμοσμένα εκπαιδευμένα μοντέλα μηχανικής όρασης που μπορέσουν να χρησιμοποιηθούν σε εφαρμογές. Υποστηρίζει μοντέλα ανίχνευσης και ταξινόμησης. Μπορεί να χρησιμοποιηθεί για την ανάθεση ετικετών σε εικόνες, για επεξεργασία εικόνας, για την εκπαίδευση μοντέλων, για την εύρεση εικόνων από δημόσια datasets κ.α. [35] Ωστόσο, την παρούσα διπλωματική χρησιμοποιήθηκε για την ανάθεση ετικετών του dataset που δημιουργήθηκε από τα δεδομένα που συλλέχθηκαν από το drone.

3.2.3 Anaconda - Weight and Biased



(a) Anaconda logo(official website)



(b) Weight and Biased logo(official website)

Figure 3.4: Conda - WB

Το Conda είναι πακέτο ανοικτού κώδικα και διαχείρισης συστημάτων που είναι συμβατό με Windows, macOS, και Linux. Το Conda εγκαταστή, εκτελεί και αναβαθμίζει πακέτα και τις εξαρτήσεις αυτών ευκολά και γρήγορα. Επίσης, δημιουργεί αποθηκεύει, φορτώνει και αλλάζει μεταξύ εικονικών περιβάλλοντων τοπικά στον υπολογιστή. Δημιουργήθηκε για προγράμματα Python αλλά μπορεί να υποστηρίξει και πακέτα και προγράμματα οποιασδήποτε γλώσσας. Στην πτυχιακή χρησιμοποιήθηκε ως εικονικό περιβάλλον που διαθέτει όλα τα απαραίτητα πακέτα για την εκπαίδευση των βάσεων του yolon5s με βάση το σύνολο δεδομένων που εξάχθηκε για τον έλεγχο της εγκυρότητας των ανιχνεύσεων και την εκτέλεση των σεναρίων. [36]

Το Weights and Biased είναι μια πλατφόρμα μηχανικής μάθησης για προγραμματιστές για να δημιουργήσουν καλύτερα μοντέλα γρηγορότερα. Δίνει τη δυνατότητα στον προγραμματιστή να παρακολουθεί την εξέλιξη των πειραμάτων, των εκδόσεων και των αναδρομών στο σύνολο δεδομένων, να αξιολογεί την απόδοση του μοντέλου, να επαναδημιουργήσει μοντέλα, να οπτικοποιήσει τα αποτελέσματα και να μπορεί να εντοπίσει παλινδρομήσεις και να διαμοιραστεί αυτές τις πληροφορίες με συναδέλφους. [37] Στη διπλωματική χρησιμοποιήθηκε για την οπτικοποίηση των δεδομένων που εξάγονται κατά την εκπαίδευση των βάσεων.

3.2.4 Python and PyTorch



(a) Python-logo(official website)



(b) PyTorch logo(official website)

Figure 3.5: Python - PyTorch

Η Python είναι διεργμηνευόμενη, αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου με δυναμική σημασιολογία. Οι υψηλού επιπέδου ενσωματωμένες δομές της σε συνδυασμό με τη δυναμική πληκτρολόγηση και τη δυναμική δέσμευση την καθιστούν πολύ ελκυστική για την ταχεία ανάπτυξη εφαρμογών. Χρησιμοποιείται επίσης ως γλώσσα επέκτασης ή συνδετική γλώσσα για τη σύνδεση διαθέσιμων μερών. Είναι απλή, με εύκολη σύνταξη και γίνεται εύκολα κατανοητή μειώνοντας έτσι το κόστος της συντήρησης του προγράμματος. Η Python υποστηρίζει ενότητες και πακέτα που βοηθούν τον αφηρωτό προγραμματισμό και την επαναχρησιμοποίηση κώδικα. Ο διεργμηνέας της και η εκτεταμένη προκαθορισμένη βιβλιοθήκη της είναι διαθέσιμη δωρεάν για όλες τις μεγάλες πλατφόρμες (πχ Windows, Linux, MacOS) και μπορεί να διανεμηθεί ελεύθερα. [38]

Το Pytorch είναι ένα framework βασισμένο στη βιβλιοθήκη Torch. Χρησιμοποιείται για εφαρμογές μηχανικής όρασης και είναι λογισμικό ανοιχτού κώδικα. Οι διεπαφές της είναι κυρίως σε python, ωστόσο διαθέτει και διεπαφές C++.

3.2.5 Open Source Computer Vision Library (OpenCV)



Σχήμα 3.6: OpenCV-logo (official website)

Η OpenCV είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για εφαρμογές μηχανικής όρασης και μάθησης που εγγενώς είναι γραμμένη στη γλώσσα προγραμματισμού

C++. Δημιουργήθηκε για να παρέχει κοινή υποδομή για εφαρμογές μηχανικής όρασης και να επιταχύνει τη χρήση της μηχανικής αντίληψης σε εμπορικά προϊόντα. Η βιβλιοθήκη παρέχει περισσότερους από 2500 βελτιστοποιημένους αλγόριθμους, που εμπεριέχουν περιεκτικό σύνολο από συμβατικούς και νέας τεχνολογίας αλγόριθμους μηχανικής μάθησης και μηχανικής όρασης. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την ανίχνευση και αναγνώριση προσώπων, την αναγνώριση αντικειμένων, την ταξινόμηση ανθρώπινων ενεργειών σε βίντεο, την παρακολούθηση κινούμενης κάμερας, την παρακολούθηση κινούμενων αντικειμένων, την εξαγωγή τρισδιάστατων μοντέλων κ.α. Η OpenCV αποτελείται από μια κοινότητα που ξεπερνάει τις 47 χιλιάδες και έχει χρησιμοποιηθεί από περίπου 18 εκατομμύρια χρήστες. Χρησιμοποιείται εκτενώς από εταιρίες, ερευνητικές ομάδες και κυβερνητικούς οργανισμούς. Διαθέτει διεπαφές C++, Python, Java και Matlab και είναι συμβατή με Windows, Linux, Android, iOS και Mac OS. Η OpenCV επικεντρώνεται κυρίως σε εφαρμογές όρασης σε πραγματικό χρόνο. Μπορεί να χτιστεί με CUDA ή OpenCL έτσι ώστε η επεξεργασία να γίνεται στην κάρτα γραφικών αντί του επεξεργαστή και να επιτευχθούν πολλαπλάσιες ταχύτητες. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν συναρτήσεις της OpenCV τόσο κατά τη διαδικασία ανίχνευσης όσο και κατά τη διαδικασία παρακολούθησης κίνησης και εξαγωγής των ταχυτήτων των κινούμενων οχημάτων. Δε χρειάστηκε να χτιστεί με CUDA ή OpenCL καθώς ο custom yolov5s αλγόριθμος δεν απαιτεί μεγάλη υπολογιστική δύναμη. [39]

3.3 Διαδικασία εκπαίδευσης βαρέων του αλγόριθμου ανίχνευσης αντικειμένων

3.3.1 Σημεία καταγραφών

Αρχικά καταγραφθήκαν βίντεο από το μη επανδρωμένο αεροσκάφος Phantom 4 Pro V2.0, με τη βοήθεια του εργαστηριακού προσωπικού, κατά μήκος του Βόρειο Οδικού Άξονα Κρήτης (BOAK) σε 8 σημεία τα οποία απέχουν μεταξύ τους περίπου 5 χιλιόμετρα, για να μπορούν να αποδώσουν ένα ρεαλιστικό μοντέλο. Στα βίντεο αυτά στη συνέχεια γίνεται ανίχνευση και παρακολούθηση κίνησης των αυτοκινήτων και εξάγονται οι ταχύτητες αυτών.



Σχήμα 3.7: Τοποθεσίες καταγραφής βίντεο - BOAK

Στο σχήμα 3.7 φαίνονται οι 8 τοποθεσίες λήψεις βίντεο καθώς και 3 τοποθεσίες (με την κατάληξη v1) που αναπαριστούν τα σημεία καταγραφών της εργασίας 'Εκτίμηση κατανάλωσης ενέργειας ηλεκτροκίνητων οχημάτων μέσω εναέριας παρατήρησης της κυκλοφορίας' του Κωσταντίνου Λεωντσίνη. Όπως αναφέρθηκε παραπάνω το drone διαθέτει GPS συνεπώς τα βίντεο διέθεταν υπότιτλους σχετικά με την πτήση που δίνουν πληροφορίες για το ύψος που βρισκόταν το drone κατά τη διάρκεια της πτήσης καθώς και την τοποθεσία.

3.3.2 Εξαγωγή καρέ και ανάθεση ετικετών

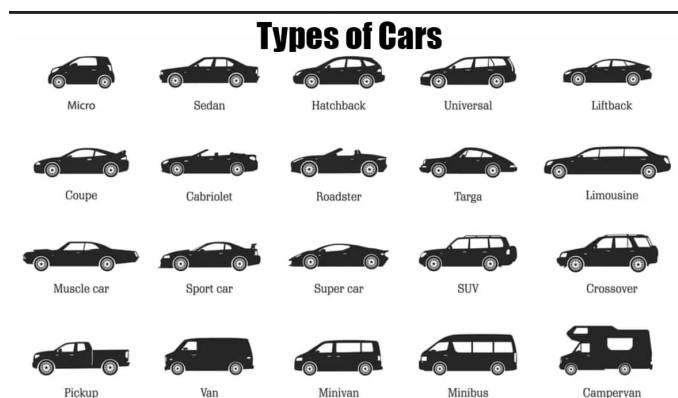
Για τη δημιουργία του συνόλου δεδομένων για την εκπαίδευση των βάσεων του YOLOv5s χρησιμοποιήθηκε η FFMPEG η οποία εξάγει καρέ για κάθε βίντεο με ρυθμιζόμενο ρυθμό ανα δευτερόλεπτο. Με την εντολή `ffmpeg -i input.mp4 -r 1 image-%4d.jpg` εξάγονται 1 καρέ το δευτερόλεπτο και αποθηκεύονται με το όνομα `image0001, image0002` κλπ. Στη συνέχεια διατηρήθηκαν μόνο τα καρέ στα οποία βρισκόταν κάποιο αντικείμενο. Ακολουθήθηκε αυτή η διαδικασία για όλα τα βίντεο και τελικά δημιουργήθηκε ένα σύνολο δεδομένων 1376 εικόνων. Στη συνέχεια έγινε η ανάθεση ετικετών για κάθε όχημα που υπήρχε σε κάθε εικόνα στο εργαλείο roboflow. Η αρχική προσέγγιση ήταν να ταξινομείται κάθε όχημα χωριστά στις κλάσεις `micro`, `hatchback`, `sedan`, `wagon`, `jeep`, `SUV`, `minivan`, `van`, `minibus` δυστυχώς όμως το αποτέλεσμα του μοντέλου αυτού δεν ήταν το επιθυμητό καθώς για να έχουμε σωστή ταξινόμηση για τόσες πολλές κλάσεις που έχουν τόσα κοινά χαρακτηριστικά απαιτείται ένα πολύ με-

γάλο σύνολο δεδομένων. Συνεπώς, έγινε σύμπτυξη των κλάσεων με βάση τα κοινά τους χαρακτηριστικά. Πιο συγκεκριμένα οι κλάσεις είναι τρεις:

- Η κλάση TYPE-A που περιλαμβάνει micro, hatchback, sedan, wagon
- Η κλάση TYPE-B που περιλαμβάνει Jeep, SUV, minivan, pickup
- Η κλάση TYPE-C που περιλαμβάνει van, minibus

Είναι επίσης χρήσιμο να αναφερθεί ότι το σύνολο δεδομένων χωρίστηκε ως εξής:

- Δεδομένα εκπαίδευσης(training set) 963 εικόνες αποτελώντας το 70% του συνόλου δεδομένων, που προορίζονται για τη διαδικασία της εκπαίδευσης
- Δεδομένα αξιολόγησης(Validation set) 265 εικόνες αποτελώντας το 19% του συνόλου δεδομένων, που προορίζονται για τη διαδικασία της αξιολόγησης
- Δεδομένα δοκιμών(testing set) 148 εικόνες αποτελώντας το 11% του συνόλου δεδομένων, που προορίζονται για τη διαδικασία των δοκιμών



Σχήμα 3.8: Τύποι οχημάτων

(Πηγή: <https://mechanicbase.com/cars/different-car-models-types/>)

Τέλος, μετά την ολοκλήρωση ανάθεσης ετικετών σε όλες τις εικόνες έγινε εξαγωγή των δεδομένων σε YOLO v5 Pytorch format. Το αρχείο περιέχει τις εικόνες εκπαίδευσης, επαλήθευσης και δοκιμών μαζί αρχεία ετικετών για κάθε εικόνα καθώς και το data.yaml αρχείο. Τα αρχεία ετικετών σε κανονικοποιημένη μορφή, δηλαδή διαιρώντας



Σχήμα 3.9: Παράδειγμα ανάθεσης ετικετών σε εικόνες

τις συντεταγμένες εικονοστοιχείου του bounding box με τις διαστάσεις της εικόνας, έχουν ως εξής:

class	x_center	y_center	width	height
0	0.5390625	0.5390625	0.10390625	0.10078125
0	0.81328125	0.11796875	0.04921875	0.06875
1	0.86015625	0.11796875	0.05	0.08671875
0	0.3640625	0.22421875	0.06484375	0.07421875

Σχήμα 3.10: Παράδειγμα αρχείου ετικετών

Θυμίζουμε ότι στον YOLOv5 τα πλαίσια οριοθέτησης αντιπροσωπεύονται από τις παραμέτρους (x_center,y_center,width,height), με (x_center,y_center) να είναι οι συντεταγμένες του κέντρου του πλαισίου ενώ width και height να είναι το μήκος και το ύψος του πλαισίου.

3.3.3 Διαδικασία εκπαίδευσης

Ξεκινώντας, γίνεται η εγκατάσταση του YOLOv5 και όλων των απαραίτητων βιβλιοθηκών από το επίσημο git των ultralytics τοπικά στο εικονικό conda περιβάλλον που δημιουργήθηκε για αυτόν τον σκοπό.


```
activate conda yolov5
git clone https://github.com/ultralytics/yolov5
cd yolov5
pip install -r requirements.txt
```

Σχήμα 3.11: Εγκατάσταση YOLOv5

Έπειτα χρειάζεται η κατάλληλη επεξεργασία του αρχείου data.yaml για να ξεκινήσει η διαδικασία της εκπαίδευσης με τις κατάλληλες παραμέτρους.

```
path: /home/chrisar/anaconda3/envs/yolov5/yolov5/data/dataset
train: /home/chrisar/anaconda3/envs/yolov5/yolov5/data/dataset/train/images
val: /home/chrisar/anaconda3/envs/yolov5/yolov5/data/dataset/valid/images

nc: 3
names: ['TYPE-A', 'TYPE-B', 'TYPE-C']
```

Σχήμα 3.12: Επεξεργασία αρχείου data.yaml

Στη συνέχεια ξεκινάει η εκπαίδευση με την παρακάτω εντολή:

```
python3 train.py
--img 640 --batch -1
--epochs 600
--data /home/chrisar/anaconda3/envs/yolov5/yolov5/data/vehicle_dataset640/data.yaml
--weights yolov5s.pt
```

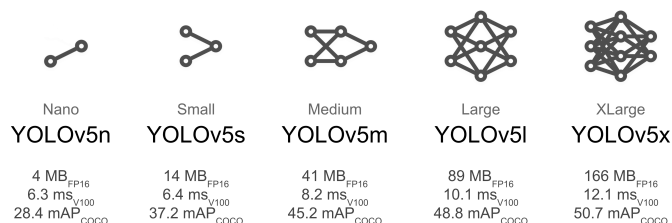
Σχήμα 3.13: Διαδικασία εκπαίδευσης

Παρακάτω εξηγούνται οι παράμετροι της εικόνα 3.13:

- `--img 640`: καθορίζει το μέγεθος της εικόνας. Ο δημιουργός του YOLOv5 αναφέρει ότι όταν τα αντικείμενα της εικόνας είναι μικρά επιλέγεται μεγάλη ανάλυση ενώ σε περίπτωση που τα αντικείμενα είναι μεγάλα δίνεται μικρή ανάλυση. Δόθηκε μέγεθος εικόνας 640×640 καθώς υπάρχουν λήψεις όπου τα οχήματα δεν είναι αρκετά κοντά και φαίνονται μικρά. Η ανάλυση αυτή έδωσε ικανοποιητικά αποτελέσματα. Επίσης αναφέρει ότι για καλύτερα αποτελέσματα η ανάλυση της εικόνας κατά την ανίχνευση πρέπει να έχει ίδιο μέγεθος με αυτό που δόθηκε στην εκπαίδευση.
- `--batch -1`: αθροίζει το μέγεθος του batch, με την τιμή -1 γίνεται προσέγγιση του καλύτερου μέγεθους. Η προώθηση χιλιάδων εικόνων στο νευρωνικό δίκτυο ταυτόχρονα βοηθάει τον αριθμό των βάρων που εκπαιδεύει το μοντέλο σε ένα epoch να αυξηθεί πολύ. Για τον λόγο αυτό, το σύνολο δεδομένων συνήθως χωρίζεται

σε πολλά batches που αποτελούνται από n εικόνες και εκπαιδεύονται το ένα batch μετά το άλλο. Το αποτέλεσμα του κάθε batch αποθηκεύεται στη RAM και αθροίζεται όταν τελειώσει η εκπαίδευση για όλα τα batches.

- `-epochs 600`: καθορίζει το μέγεθος του εκπαιδευόμενου epoch. Το epoch είναι υπεύθυνο για την εκμάθηση όλων των εικόνων εισόδου, δηλαδή εκπαιδεύει όλες τις εισόδους. Το σύνολο δεδομένων είναι χωρισμένο σε πολλά batches, ένα epoch θα είναι υπεύθυνο για την εκπαίδευση όλων των batches. Ο αριθμός των epoch αναπαριστά τον αριθμό των επαναλήψεων που το μοντέλο εκπαιδεύει όλες τις εισόδους του και ανανεώνει τα βάρη του έτσι ώστε να πλησιάσει τις τιμές των πραγματικών ετικετών που ανατέθηκαν.
- `-data`: Εμπεριέχει τη σύνοψη του συνόλου δεδομένων. Η διαδικασία αξιολόγηση του μοντέλου πραγματοποιείται αμέσως μετά από κάθε epoch, έτσι το μοντέλο θα έχει πρόσβαση στην τοποθεσία που βρίσκονται τα δεδομένα που προορίζονται για αξιολόγηση και δοκιμές από το μονοπάτι που βρίσκεται στο αρχείο `data.yaml`.
- `-weights`: Επιλέγονται τα προ εκπαιδευμένα βάρη του YOLOv5s ως σημείο αναφοράς κατά την εκκίνηση της διαδικασίας της εκπαίδευσης. [40]



Σχήμα 3.14: Σύγκριση προεκαπιδευμένων βάρων YOLOv5

(Πηγή: <https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data>)

Η εκπαίδευση έγινε τοπικά σε φορητό υπολογιστή με κάρτα γραφικών NVIDIA GeForce GTX 1050 και διήρκεσε 8h 34m 23s. Έγιναν αρκετές δοκιμές με διάφορες αναλύσεις εικόνας εισόδου και διαφορετικό αριθμό epoches, των οποίων η εκπαίδευση απαιτούσε αντίστοιχους χρόνους, με τα καλύτερα αποτελέσματα να δίνονται για ανάλυση εικόνας 640×640 και 409 epoch. Επιπλέον, είναι χρήσιμο να αναφερθεί ότι τα αποτελέσματα του τελευταίου epoch δεν είναι πάντα αυτά που έχουν τα καλύτερα

αποτελέσματα. Μπορεί να έχουν επιτευχθεί καλύτερα αποτελέσματα σε προηγούμενο epoch.

Epoch	gpu_mem	box	obj	cls	labels	img_size	
409/599	3.07G	0.01314	0.00639	0.0008522	59	640: 4%	2/57 [00:02<01:09, 1.26s/it]
409/599	3.07G	0.01371	0.007191	0.001786	64	640: 5%	3/57 [00:03<01:08, 1.26s/it]
409/599	3.07G	0.0147	0.006998	0.001713	90	640: 9%	5/57 [00:06<01:05, 1.27s/it]
409/599	3.07G	0.01435	0.006839	0.001868	52	640: 11%	6/57 [00:07<01:04, 1.26s/it]
409/599	3.07G	0.01493	0.007583	0.001917	81	640: 14%	8/57 [00:10<01:02, 1.27s/it]
409/599	3.07G	0.01502	0.00787	0.001857	83	640: 16%	9/57 [00:11<01:01, 1.27s/it]
409/599	3.07G	0.01519	0.007895	0.001923	76	640: 19%	11/57 [00:13<00:58, 1.27s/it]
409/599	3.07G	0.01486	0.008241	0.002485	98	640: 23%	13/57 [00:16<00:55, 1.26s/it]
409/599	3.07G	0.01508	0.008328	0.002396	98	640: 25%	14/57 [00:17<00:54, 1.26s/it]
409/599	3.07G	0.01533	0.008662	0.002307	118	640: 28%	16/57 [00:20<00:51, 1.27s/it]
409/599	3.07G	0.01542	0.008538	0.002326	58	640: 30%	17/57 [00:21<00:50, 1.26s/it]
409/599	3.07G	0.01504	0.008424	0.002257	61	640: 33%	19/57 [00:24<00:47, 1.26s/it]
409/599	3.07G	0.01514	0.008591	0.002294	93	640: 37%	21/57 [00:26<00:45, 1.27s/it]
409/599	3.07G	0.01499	0.008481	0.00225	52	640: 39%	22/57 [00:27<00:44, 1.26s/it]
409/599	3.07G	0.01528	0.008578	0.00246	89	640: 42%	24/57 [00:30<00:41, 1.27s/it]
409/599	3.07G	0.0155	0.008566	0.002411	95	640: 44%	25/57 [00:31<00:40, 1.27s/it]
409/599	3.07G	0.01536	0.008488	0.002309	40	640: 47%	27/57 [00:34<00:37, 1.26s/it]
409/599	3.07G	0.01526	0.008461	0.002265	73	640: 49%	28/57 [00:35<00:36, 1.27s/it]
409/599	3.07G	0.01515	0.008337	0.002189	43	640: 53%	30/57 [00:37<00:34, 1.26s/it]
409/599	3.07G	0.01507	0.008281	0.00215	67	640: 56%	32/57 [00:40<00:31, 1.27s/it]
409/599	3.07G	0.01504	0.008301	0.002139	76	640: 58%	33/57 [00:41<00:30, 1.27s/it]
409/599	3.07G	0.01499	0.008267	0.002072	54	640: 61%	35/57 [00:44<00:27, 1.26s/it]
409/599	3.07G	0.01493	0.008284	0.00207	68	640: 63%	36/57 [00:45<00:26, 1.27s/it]
409/599	3.07G	0.01489	0.008305	0.002087	67	640: 67%	38/57 [00:48<00:24, 1.26s/it]
409/599	3.07G	0.01475	0.00821	0.00205	67	640: 70%	40/57 [00:50<00:21, 1.27s/it]
409/599	3.07G	0.01473	0.0082	0.002059	59	640: 72%	41/57 [00:51<00:20, 1.26s/it]
409/599	3.07G	0.01473	0.008176	0.00202	69	640: 75%	43/57 [00:54<00:17, 1.26s/it]
409/599	3.07G	0.01485	0.008211	0.002023	88	640: 77%	44/57 [00:55<00:16, 1.27s/it]
409/599	3.07G	0.01475	0.008152	0.001978	52	640: 81%	46/57 [00:58<00:13, 1.27s/it]
409/599	3.07G	0.01466	0.008141	0.001954	62	640: 82%	47/57 [00:59<00:12, 1.27s/it]
409/599	3.07G	0.01475	0.008155	0.001984	77	640: 86%	49/57 [01:02<00:10, 1.27s/it]
409/599	3.07G	0.01486	0.008167	0.001956	59	640: 89%	51/57 [01:04<00:07, 1.26s/it]
409/599	3.07G	0.01487	0.008131	0.001971	41	640: 91%	52/57 [01:05<00:06, 1.27s/it]
409/599	3.07G	0.0149	0.008136	0.001947	55	640: 95%	54/57 [01:08<00:03, 1.26s/it]
409/599	3.07G	0.01496	0.008165	0.001935	64	640: 96%	55/57 [01:09<00:02, 1.26s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 12%	1/8 [00:00<00:06, 1.02it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 38%	3/8 [00:02<00:04, 1.02it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 62%	5/8 [00:04<00:02, 1.02it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 88%	7/8 [00:06<00:00, 1.02it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 0%	0/8 [00:00<?, 2it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 25%	2/8 [00:02<00:08, 1.45s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 50%	4/8 [00:06<00:06, 1.56s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 62%	5/8 [00:07<00:04, 1.39s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:10<00:00, 1.28s/it]
all	265	762	0.429	0.221	0.125	0.0338	8/8 [00:10<00:00, 1.28s/it]

Σχήμα 3.15: last epoch

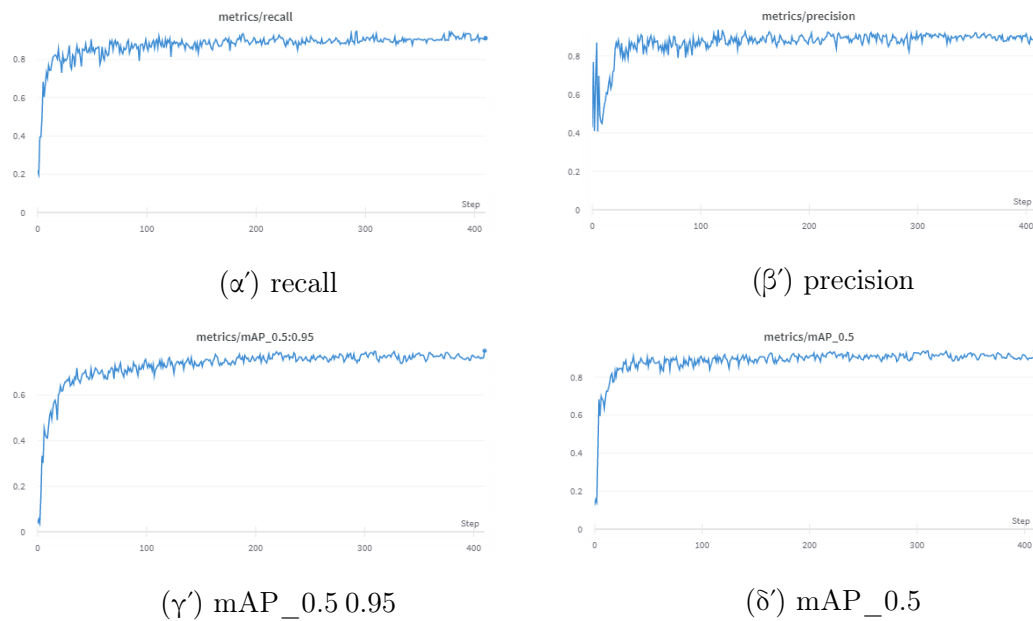
Starting training for 600 epochs...

Epoch	gpu_mem	box	obj	cls	labels	img_size	
0/599	3.07G	0.09615	0.03805	0.03182	37	640: 100%	57/57 [01:08<00:00, 1.20s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:07<00:00, 1.03it/s]
all	265	762	0.429	0.221	0.125	0.0338	
Epoch	gpu_mem	box	obj	cls	labels	img_size	
1/599	3.06G	0.07434	0.02423	0.02461	33	640: 100%	57/57 [01:07<00:00, 1.18s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:07<00:00, 1.13it/s]
all	265	762	0.768	0.199	0.156	0.0549	
Epoch	gpu_mem	box	obj	cls	labels	img_size	
2/599	3.07G	0.06795	0.02051	0.02445	38	640: 100%	57/57 [01:05<00:00, 1.16s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:06<00:00, 1.17it/s]
all	265	762	0.409	0.394	0.141	0.0369	
Epoch	gpu_mem	box	obj	cls	labels	img_size	
3/599	3.07G	0.05972	0.01933	0.02263	50	640: 100%	57/57 [01:05<00:00, 1.15s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:06<00:00, 1.16it/s]
all	265	762	0.653	0.396	0.397	0.155	
Epoch	gpu_mem	box	obj	cls	labels	img_size	
4/599	3.07G	0.05208	0.01564	0.02156	27	640: 100%	57/57 [01:05<00:00, 1.14s/it]
Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	8/8 [00:06<00:00, 1.17it/s]
all	265	762	0.869	0.5	0.683	0.334	

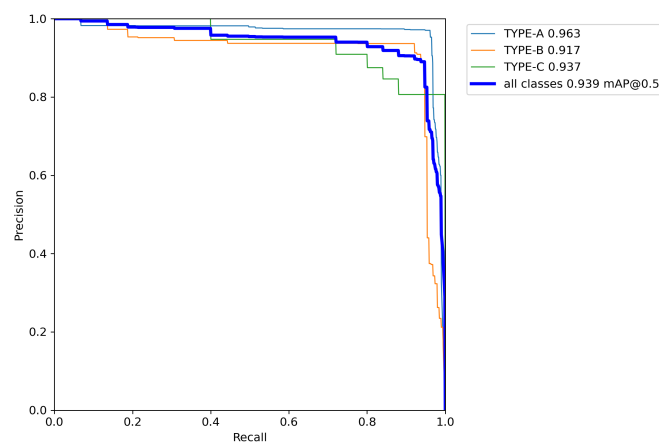
Σχήμα 3.16: epoch 1 to 5

3.3.4 Αξιολόγηση μοντέλου

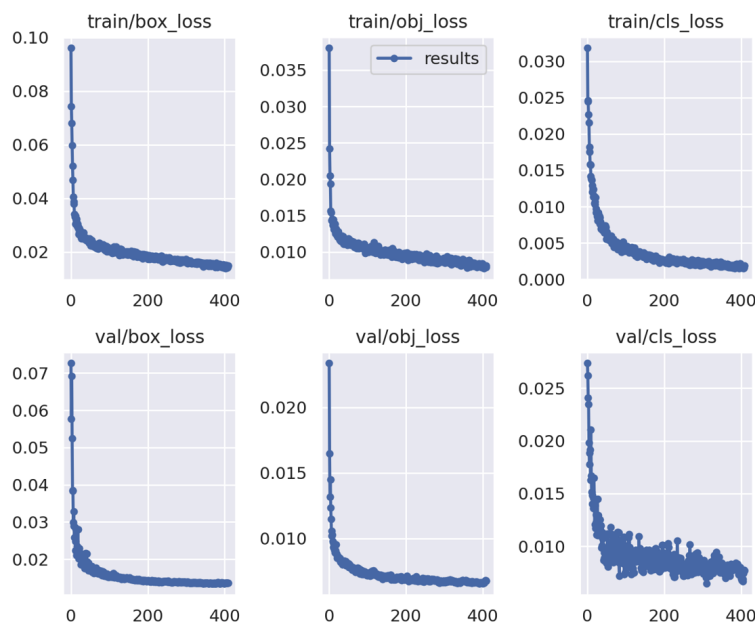
Το αρχείο output.log καθώς και η γραφική απεικόνιση των μετρικών αξιολόγησης του μοντέλου αποθηκεύτηκαν στο Weights and Biases για όλα τα στάδια κατά τη διάρκεια της εκπαίδευσης. Τα αποτελέσματα αυτά φαίνονται παρακάτω:



Σχήμα 3.17: Γραφική απεικόνιση μετρικών του μοντέλου



Σχήμα 3.18: Καμπύλη Precision-Recall



Σχήμα 3.19: losses

Το precision, το recall και η καμπύλη Precision-Recall εξηγούνται στο κεφάλαιο 2.8, το mAP_0.5 υπολογίζεται με τον μέσο όρο των τιμών Μέσης Ακρίβειας (AP) σε διάφορες κατηγορίες αντικειμένων, με σταθερό κατώφλι IoU 0.5. Αυτή η μετρική χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου στον εντοπισμό αντικειμένων με ελάχιστη επικάλυψη 50% μεταξύ του προβλεπόμενου οριοθετημένου πλαισίου και του πραγματικού. Το mAP-0.5:0.95 υπολογίζεται με τον μέσο όρο των τιμών AP σε διάφορες κατηγορίες αντικειμένων, αλλά με μεταβαλλόμενα κατώφλια IoU που κυμαίνονται από 0.5 έως 0.95 με μέγεθος βήματος 0.05. Αυτή η μετρική χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου στον εντοπισμό αντικειμένων με διαφορετικά επίπεδα επικάλυψης μεταξύ του προβλεπόμενου πλαισίου οριοθέτησης και του πραγματικού.

Με άλλα λόγια, το mAP_0,5 αξιολογεί την απόδοση του μοντέλου σε ένα μόνο, σταθερό επίπεδο επικάλυψης, ενώ το mAP-0,5:0,95 αξιολογεί την απόδοση του μοντέλου σε ένα εύρος επιπέδων επικάλυψης. Η τελευταία θεωρείται γενικά πιο ολοκληρωμένη μετρική αξιολόγησης, καθώς παρέχει μια καλύτερη συνολική εικόνα της ικανότητας του μοντέλου να ανιχνεύει αντικείμενα με διαφορετικά επίπεδα ακρίβειας.

Το box_loss (ή αλλιώς regression loss) μετρά τη διαφορά μεταξύ των προβλέψεων και της πραγματικής θέσης του αντικειμένου σε κάθε πλαίσιο (ή bounding box). Η συνάρτηση απώλειας που χρησιμοποιείται συνήθως είναι η Mean Squared Error (MSE)

ή η Smooth L1 loss.

Το `obj_loss` (ή αλλιώς `objectness loss`) μετρά τον βαθμό εμπιστοσύνης του μοντέλου ότι σε ένα πλαίσιο περιέχεται αντικείμενο. Η συνάρτηση απώλειας που χρησιμοποιείται συνήθως είναι η Binary Cross Entropy (BCE) loss.

Το `cls_loss` (ή αλλιώς `classification loss`) μετρά τη διαφορά μεταξύ των προβλέψεων και της πραγματικής κατηγορίας του αντικειμένου σε κάθε πλαίσιο. Η συνάρτηση απώλειας που χρησιμοποιείται συνήθως είναι η Cross Entropy loss.

Τα αποτελέσματα αυτά είναι για ένα σύνολο δεδομένων που αποτελείται από 1376 εικόνες. Το μοντέλο εκτελεί 409 epoches. Φαίνεται λοιπόν ότι η ακρίβεια του μοντέλου είναι πολύ καλή. Μετά από κάθε epoch οι μετρητές recall, precision, mAP φαίνεται ότι αυξάνονται αρχικά ραγδαία και στη συνέχεια υπάρχουν μικρές ανοδικές και καθοδικές κλιμακώσεις. Επιπλέον, από τις γραφικές των απωλειών φαίνεται ότι οι απώλειες μειώνονται ραγδαία μετά από κάθε epoch και από ένα σημείο και μετά μειώνονται με πολύ αργό ρυθμό κάτι το οποίο είναι επιθυμητό. Αποδεικνύεται λοιπόν πως μόνο με την αρχική αρχιτεκτονική, χωρίς τη χρήση μεθόδων βελτιστοποίησης να είναι ενσωματωμένες στο μοντέλο, δημιουργείται ένα μοντέλο που όχι μόνο είναι γρήγορο αλλά είναι και ακριβές.

Στο τέλος της εκπαίδευσης δημιουργούνται δύο αρχεία όπου είναι αποθηκευμένα τα αποτελέσματα των βαρών. Το ένα αρχείο είναι το `best.pt` και το άλλο είναι το `last.pt`. Στο `last.pt` αποθηκεύονται οι τιμές των βάρων στο τελευταίο epoch ενώ στο `best.pt` αποθηκεύονται οι τιμές των βάρων για το epoch με τα καλύτερα αποτελέσματα. Το μέγεθος του κάθε αρχείου δεν ξεπερνάει τα 14.5MB καθιστώντας το μοντέλο πολύ ελαφρύ και εύχρηστο.

3.3.5 Αλληλεπίδραση με τα εκπαιδευμένα βάρη

Τα εκπαιδευμένα βάρη μπορούν να χρησιμοποιηθούν για να ανιχνεύσουν τις τρεις κατηγορίες οχημάτων σε κάθε εικόνα/βίντεο. Αν κάποια κατηγορία οχήματος ανιχνεύεται, τότε ένα πλαίσιο οριοθέτησης σχεδιάζεται γύρω από το αντικείμενο μαζί με την πιθανότητα αναγνώρισης για το αντικείμενο αυτό. Για τη διαπίστωση του αν το μοντέλο ανιχνεύει τα αντικείμενα ταξινομώντας τα σωστά, αλληλεπιδρούμε με το μοντέλο και τα βίντεο.

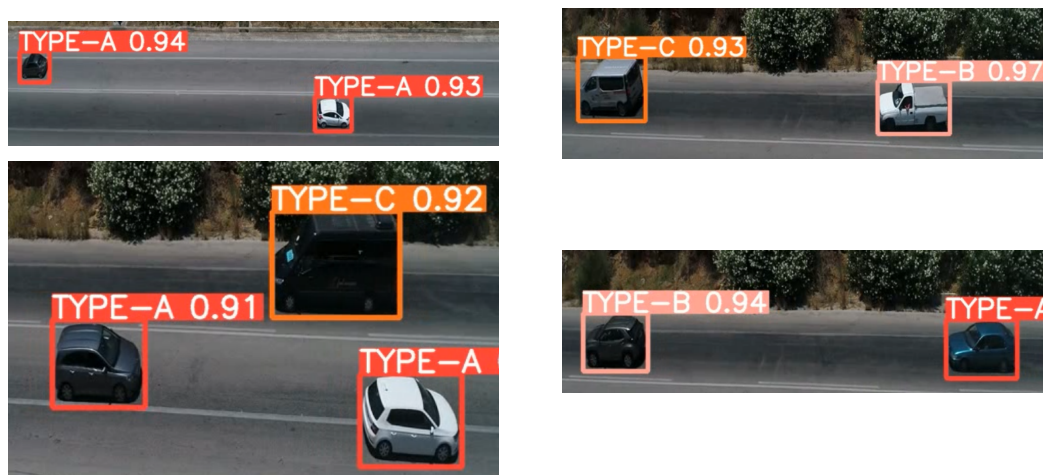
Με την εντολή [3.20](#) γίνεται η ανίχνευση στο βίντεο, όπου `-weights` τα εκπαιδευμένα

βάρη, `-img 640` η ανάλυση της εικόνας που γίνεται η ανίχνευση, `-conf 0.7` η πιθανότητα κατωφλίου για τα αντικείμενα που αναγνωρίζονται και `-source` η πηγή εισόδου του οπτικού μέσου που δίνεται για ανίχνευση.

```
python3 /home/chrisar/anaconda3/envs/yolov5/yolov5/detect.py
--weights /home/chrisar/Thesis/yolov5_vehicle_detection/runs/train/vehicle_detection_TYPEv2_s_420ep_x640/weights/best.pt
--img 640
--conf 0.7
--source /home/chrisar/Thesis/Voak_videos/a1_640.mp4
```

Σχήμα 3.20: Detect.py

Μετά την ανίχνευση τα βίντεο στα οποία φαίνονται οι ανιχνεύσεις αποθηκεύονται. Ενδεικτικά αποτελέσματα φαίνονται στο σχήμα 3.21.



Σχήμα 3.21: Αποτελέσματα Detect.py

Παρατηρούμε ότι στα παραπάνω ενδεικτικά εικονιζόμενα καρέ τα hatchback αναγνωρίζονται σωστά ως TYPE-A τα pickup trucks και SUVs ως TYPE-B ενώ τα Vans και τα minibuses ως TYPE-C. Φαίνεται λοιπόν ότι το μοντέλο λειτουργεί σωστά ανιχνεύοντας και ταξινομώντας τα οχήματα στα εικονιζόμενα καρέ. Οι ανιχνεύσεις έγιναν και αποθηκεύτηκαν για όλα τα βίντεο δίνοντας αξιόλογα αποτελέσματα.

3.3.6 Διαδικασία εξαγωγής ταχυτήτων

3.3.6.1 Εξαγωγή σε τύπο ONNX

Αρχικά τα εκπαιδευμένα βάρη `best.pt` είναι σε μορφή κατάλληλη για PyTorch. Για την επεξεργασία των ανιχνεύσεων σε συνδυασμό με τη βιβλιοθήκη OpenCV αλλά και άλλων βιβλιοθηκών είναι απαραίτητη η μετατροπή των βάρων σε μορφή `.onnx`. Αυτό γίνεται

με την εντολή:

```
python export.py
--weights best.pt
--include torchscript onnx
```

Σχήμα 3.22: Εξαγωγή ως ONNX

3.3.6.2 Ανίχνευση αντικειμένων

Τώρα που πλέον το μοντέλο είναι στον κατάλληλο τύπο μπορεί να χρησιμοποιηθεί σε συνδυασμό με τη OpenCV. Εισάγουμε τα βάρη με την εντολή:

`net = cv2.dnn.readNet('yolov5s.onnx')`. Στη συνέχεια το καρέ που δίνεται ως είσοδο στον YOLOv5 πρέπει να είναι στη μορφή RGB με κανονικοποιημένες τιμές εικονοστοιχείων και μέγεθος 640×640 . Η διαδικασία αυτή γίνεται στη συνάρτηση `format_yolov5()`. Στη συνέχεια στη συνάρτηση `detect` πραγματοποιούνται οι ανιχνεύσεις. Το περιεχόμενο των ανιχνεύσεων αυτών ξετυλίγεται στη `wrap_detection()`. Οι ανιχνεύσεις αυτές είναι της μορφής ενός πίνακα δύο διαστάσεων που αποτελείται από 25200 στήλες με κάθε στήλη να αποτελεί μια ανίχνευση και να έχει στις πρώτες θέσεις πληροφορίες για το πλαίσιο οριοθέτησης (`x_center, y_center, width, height`) στην πέμπτη θέση το σκορ σιγουριάς ενώ στην έκτη, έβδομη και όγδοη το σκορ της κλάσης TYPE-A, TYPE-B και TYPE-C αντίστοιχα. Βέβαια, δεν διατηρούνται και οι 25200 στήλες καθώς χρησιμοποιείται ένα κατώφλι για να απορρίψει τις τιμές χαμηλής σιγουριάς (π.χ. *confidence* < 0.7). Τέλος, χρησιμοποιούμε τον αλγόριθμο Non-Maximum Suppression (NMS) για να αφαιρέσουμε τις διπλές και τις επικαλυπτόμενες ανιχνεύσεις.

3.3.6.3 Ανίχνευση κίνησης αντικειμένων

Για την παρακολούθηση κίνησης των οχημάτων διατηρούνται οι συντεταγμένες του κέντρου του πλαισίου οριοθέτησης των ανιχνευόμενων αντικειμένων του παρόντος και του προηγούμενου καρέ. Στη συνέχεια συγκρίνεται για κάθε ζεύγος συντεταγμένων στα δύο αυτά καρέ η απόσταση μεταξύ αυτών και αν είναι κάτω από μια συγκεκριμένη απόσταση τότε θεωρείται ότι είναι το ίδιο αντικείμενο. Σε διαφορετική περίπτωση

ανατίθεται νέο αναγνωριστικό στο ανιχνευόμενο αντικείμενο. Στον αυτοσχέδιο αυτόν ανιχνευτή αφαιρούνται επίσης τα αναγνωριστικά για τα οχήματα που αποχωρούν από το προσκήνιο. Αυτή η διαδικασία θα μπορούσε να γίνει και με τους έτοιμους αλγόριθμους SORT και DeepSORT ωστόσο απαιτούν αρκετή υπολογιστική ισχύ, κάτι που δυστυχώς δεν ήταν διαθέσιμο καθώς τα πειράματα εκτελούνται σε φορητό υπολογιστή.



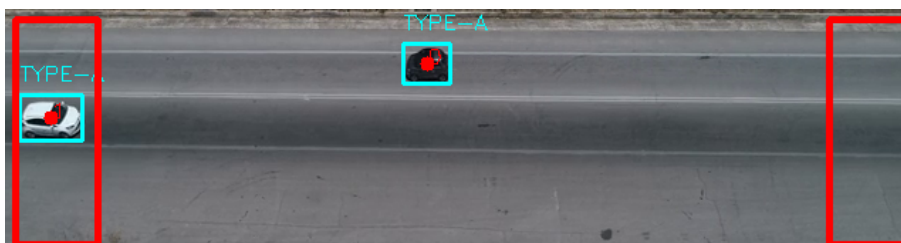
Σχήμα 3.23: Δείγμα παρακολούθησης κίνησης

3.3.6.4 Εξαγωγή ταχυτήτων

Για την εξαγωγή των ταχυτήτων συνδυάστηκε η τεχνική ανίχνευσης και παρακολούθησης κίνησης των οχημάτων. Συγκεκριμένα σχεδιάστηκαν στα βίντεο δύο περιοχές. Η δεξιά περιοχή και η αριστερή περιοχή. Με τη συνάρτηση `cv2.pointPolygonTest()` ελέγχεται αν το κέντρο του πλαισίου οριοθέτησης των ανιχνευόμενων οχημάτων έχει διέλθει από τη δεξιά περιοχή και αν ναι δηλαδή η συνάρτηση `cv2.pointPolygonTest()` επιστρέφει την τιμή 1 τότε διατηρείται σε λεξικό το αναγνωριστικό του οχήματος μαζί με το καρέ στο οποίο βρέθηκε στην περιοχή αυτή. Έπειτα ελέγχουμε αν στο λεξικό υπάρχει το αναγνωριστικό και τότε αν το αναγνωριστικό αυτό βρεθεί στην αριστερή περιοχή τότε διατηρείται το αναγνωριστικό και το καρέ που βρέθηκε στην περιοχή αυτή και μετράτε η ταχύτητα του. Συγκεκριμένα η απόσταση της δεξιάς με την αριστερή περιοχή μετράτε στο Google earth, γνωρίζουμε τον ρυθμό καταγραφής του βίντεο (fps) και αφαιρώντας το καρέ που βρέθηκε το αυτοκίνητο στην αριστερή περιοχή από το καρέ που το όχημα βρέθηκε στη δεξιά περιοχή, μπορούμε να βρούμε πόσα καρέ πέρασαν μέχρι το όχημα να διανύσει τη συγκεκριμένη απόσταση. Συνεπώς, η ταχύτητα προκύπτει ως εξής:

$$speed = \left(\frac{distance}{frames \times spf} \right) \times 3.6 km/h,$$

όπου $spf = \frac{1}{fps}$, frames είναι ο αριθμός των καρέ που το όχημα περνάει από τη δεξιά στην αριστερή περιοχή ενώ με τον πολλαπλασιασμό με 3.6 γίνεται η μετατροπή από m/s σε km/h. Η ίδια διαδικασία ακολουθήθηκε για τα οχήματα που διέρχονται από τα αριστερά προς τα δεξιά.



Σχήμα 3.24: Δείγμα της μεθόδου

Η μέθοδος αυτή ωστόσο δεν ήταν εφικτή. Παρόλο που έγινε αποκοπή των βίντεο από ανάλυση 4K σε 640×640 , κάτι που καθιστά την επεξεργασία βίντεο δύσκολη, και πάλι η ταχύτητα εκτέλεσης του βίντεο δεν ξεπερνούσε τα 6 FPS. Τα παραπάνω ήταν αποθαρρυντικοί παράγοντες για την εφαρμογή αυτής της μεθόδου. Η λύση δόθηκε από την διπλωματική εργασία με τίτλο **‘‘Εκτίμηση κατανάλωσης ενέργειας ηλεκτροκίνητων οχημάτων μέσω εναέριας παρατήρησης της κυκλοφορίας’’** του συμφοιτητή **Κωσταντίνου Λεωνστίτη**. Υιοθετήθηκαν τεχνικές με την διαφορά του ότι για την ανίχνευση χρησιμοποιείται ο custom YOLOv5s που δημιουργήθηκε όπως εξηγήθηκε παραπάνω. Παρακάτω εξηγείται αναλυτικά η μέθοδος.

3.3.6.5 Βέλτιστη εξαγωγή ταχυτήτων

Για την εξαγωγή των ταχυτήτων των βίντεο που καταγραφθήκαν στις 8 περιοχές όπως εξηγείται στο 3.3.1 συνδιδάστηκε ο custom YOLOv5s, για την ανίχνευση των οχημάτων που ανήκουν σε μια από τις κατηγορίες TYPE-A, TYPE-B, TYPE-C, μαζί με τη βιβλιοθήκη OpenCV, για την παρακολούθηση κίνησης των οχημάτων. Πιο συγκεκριμένα, χρησιμοποιείται αρχικά η MOG2 στο καρέ και το αποτέλεσμα αυτής υπόκειται μορφολογική επεξεργασία με δομικό στοιχείο (structural element) ένα φίλτρο (kernel) περιττού μεγέθους. Στην ουσία εκτελείται ένα opening δηλαδή αρχικά γίνεται ένα erosion στο καρέ με κάποιο kernel και στο αποτέλεσμα αυτού εκτελείται dilation με κάποιο kernel. Σκοπός της παραπάνω διαδικασίας είναι η αφαίρεση του θορύβου. Μετά τη διαδικασία αυτή το καρέ είναι μια δυαδική εικόνα που δεν έχει θόρυβο. Στο καρέ

αυτό εντοπίζονται τα contours, δηλαδή τα συνεχείς σημεία που έχουν το ίδιο χρώμα ή την ίδια ένταση. Αν το contour είναι μεγαλύτερο από έναν αριθμό εικονοστοιχείων που ορίζεται σε κάθε βίντεο και βρίσκεται πριν από μια κόκκινη γραμμή που ορίζεται σε κάθε βίντεο τότε το καρέ δίνεται στον custom YOLOv5s για ανίχνευση.

Στη συνάρτηση YOLOv5_frame συνδυάζοντάς τις συναρτήσεις που χρησιμοποιούνται για την ανίχνευση αντικειμένων που εξηγούνται στο 3.3.6.2 εξάγεται το σύμπερασμα του αν στο καρέ που δόθηκε για ανάγνωση εμπεριέχεται κάποιο όχημα που είναι TYPE-A, TYPE-B, TYPE-C. Πρέπει να σημειωθεί επίσης ότι οι συντεταγμένες (x,y,w,h) των contours που είναι μεγαλύτερα από τον ορισμένο αριθμό εικονοστοιχείων διατηρούνται σε ένα λεξικό μαζί με το ID, τις φορές που αναγνωρίστηκε ως όχημα όταν δόθηκε στον YOLOv5_frame, τον αριθμό των φορών που δόθηκε το contour για ανάγνωση στον custom YOLOv5s, και τέλος τον αριθμό των καρέ που χρειάστηκε το όχημα να διανύσει την απόσταση μεταξύ των δυο πράσινων γραμμών. Αν η ευκλείδεια απόσταση των συντεταγμένων (x,y) του contour του προηγούμενου και του παρόντος καρέ είναι μικρότερη από κάποιον αριθμό εικονοστοιχείων που ορίζονται ανάλογα το βίντεο, τότε θεωρείται πως είναι το ίδιο όχημα, ανανεώνονται οι συντεταγμένες του και διατηρείται το id. Διαφορετικά θεωρείται ότι βρέθηκε νέο όχημα και του δίνεται id. Αν ο αριθμός των φορών που αναγνωρίστηκε στο καρέ κάποιο όχημα προς τις φορές που δόθηκαν καρέ για ανάγνωση πριν την κόκκινη γραμμή είναι μεγαλύτερος από κάποιο ποσοστό που ορίζεται σε κάθε βίντεο τότε τοποθετείται ένα πλαίσιο γύρω από το όχημα και παρακολουθείται η κίνηση του όσο βρίσκεται εντός των δύο πράσινων γραμμών.

Πρέπει να σημειωθεί πως για κάθε βίντεο γράφονται δυο παρόμοια scripts, ένα για τη δεξιά λωρίδα κυκλοφορίας και ένα για την αριστερή. Σε κάθε σενάριο τοποθετούνται η κόκκινη και οι δυο πράσινες γραμμές σε τοποθεσία κατάλληλη για το κάθε σενάριο. Επιπρόσθετα σε κάθε σενάριο δε δίνεται ολόκληρο το καρέ παρά μόνο η περιοχή που έχει τη λωρίδα κυκλοφορίας ενδιαφέροντος του σεναρίου. Επίσης, πριν την κόκκινη γραμμή τα καρέ που περιέχουν contour που ικανοποιούν τη συνθήκη για ανίχνευση, δεν δίνονται ολόκληρα στη YOLOv5_frame παρά μόνο η ελάχιστη περιοχή που απαιτείται για επιτυχής ανίχνευση. Μεταξύ των πράσινων γραμμών μετριάται το πλήθος των καρέ που χρειάστηκαν το όχημα για να διανύσει την απόσταση. Η απόσταση μεταξύ των πράσινων γραμμών εξάγεται με καλή προσέγγιση από το Google earth καθώς στα

βίντεο υπήρχαν, όπως έχει ήδη αναφερθεί, υπότιτλοι που διέθεταν τις συντεταγμένες που έγινε η πτήση του drone. Η εξίσωση για την επιτυχής αναγνώριση είναι:

$$\frac{times\ itsYolo}{times\ it\ sent} = percent\%$$

ενώ για την ταχύτητα είναι:

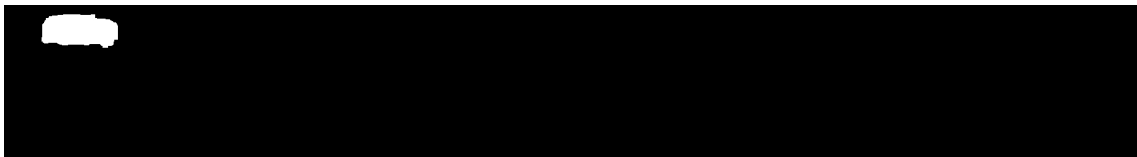
$$speed = \left(\frac{Google\ distance}{Frames\ Needed \times spf} \right) \times 3.6km/h,$$

όπου $spf = \frac{1}{fps}$, Frames Needed είναι ο αριθμός των καρέ που χρειάζεται το όχημα να διανύσει την απόσταση μεταξύ των πράσινων γραμμών, ενώ με τον πολλαπλασιασμό με 3.6 γίνεται η μετατροπή από m/s σε km/h. Η εξίσωση $time = Frames\ Needed \times spf$ (seconds) δίνει τον χρόνο που χρειάζεται το όχημα για να διανύσει την απόσταση. Με εντολές της OpenCV λαμβάνονται ο συνολικός αριθμός των καρέ καθώς και τα fps που καταγράφηκε το βίντεο.

Παρακάτω δίνονται παραδείγματα για καλύτερη κατανόηση.



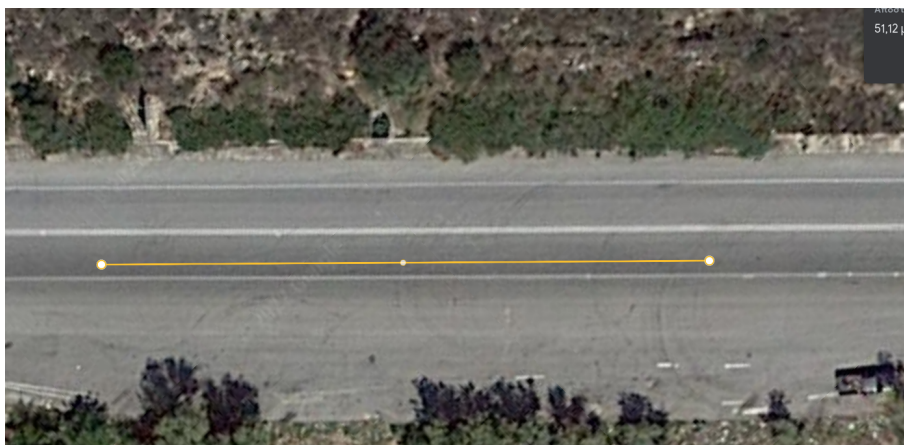
Σχήμα 3.25: Ανίχνευση πριν την κόκκινη γραμμή



Σχήμα 3.26: Παράδειγμα εφαρμογής της μάσκας



Σχήμα 3.27: Ακριβώς πριν περάσει την πράσινη γραμμή



Σχήμα 3.28: Η απόστατη υπολογισμένη στο google earth



Σχήμα 3.29: καρέ YOLO

Στο παραπάνω παράδειγμα, παρουσιάζεται το σενάριο της αριστερής λωρίδας. Πριν την κόκκινη γραμμή το καρέ που βρίσκεται το όχημα με το contour που ικανοποιεί τις συνθήκες μεγέθους, δίνεται για ανίχνευση στον yolo. Στη δυαδική εικόνα mask φαίνεται το contour. Ενώ στην εικόνα yolo frame φαίνεται το καρέ που δόθηκε στον yolo για ανίχνευση. Το όχημα δίνεται στον yolo για ανίχνευση 27 φορές και αναγνωρίζεται ως κάποιο TYPE 27 φορές. Τα καρέ που διανύει μεταξύ των πράσινων γραμμών είναι 148, η απόσταση στο Google earth είναι 51m και η καταγραφή του βίντεο 60 fps συνεπώς η ταχύτητα υπολογίζεται 74km/h. Όμοια εργαζόμαστε και για το σενάριο δεξιάς λωρίδας αλλάζοντας τις γραμμές. Είναι σημαντικό να αναφερθεί ότι ο αλγόριθμος εκτελείται στον επεξεργαστή και όχι στην κάρτα γραφικών καθώς και ότι η εικόνα είσοδο στον YOLOv5s είναι 640×640 και σημειώνει ταχύτητα 6-7fps όταν ανιχνεύει όχημα ενώ για την παρακολούθηση κίνησης οχημάτων στην περιοχή των πράσινων γραμμών σημειώνει ταχύτητα από 50 έως 70 fps. Το γεγονός ότι ο αλγόριθμος τρέχει στον επεξεργαστή σε αυτή την ταχύτητα τον καθιστά πολύ ευέλικτο καθώς μπορεί να δοκιμαστεί σε υπολογιστές με χαμηλή υπολογιστική ισχύ. Η ταχύτητα αυτή θα μπορούσε να αυξηθεί αν είχε χτιστεί η OpenCV με CUDA κάτι που δεν ήταν απαραίτητο καθώς η ταχύτητα επεξεργασίας είναι πολύ ικανοποιητική και τα βίντεο είναι μικρής διάρκειας με το με-

γαλύτερο να είναι 5 λεπτά.Επιπλέον, δημιουργήθηκε μια συνάρτηση περιστροφής της εικόνας σε περίπτωση που το βίντεο είχε καταγραφή υπό γωνία και δεν ήταν εύκολο να αποκοπεί το άλλο ρεύμα κυκλοφορίας.Υπάρχει και αρχείο Readme.txt με περισσότερες πληροφορίες για τον κώδικα.

Οι ταχύτητες που εξάχθηκαν από τις 8 περιοχές στην παρούσα διπλωματική είναι 495.Σε συνδυασμό με τις 405 ταχύτητες από τις 3 περιοχές από προηγούμενη εργασία του Κωνσταντίνου Λεωντσίνη συνολικά εξάχθηκαν στον BOAK 900 ταχύτητες.Τα αποτελέσματα αυτών φαίνονται στον πίνακα [3.3](#)

Αριθμός Οχημάτων	Εύρος Ταχυτήτων(km/h)
1	≤ 40
0	40-50
3	50-60
21	60-70
46	70-80
119	80-90
126	90-100
64	100-110
58	110-120
32	120-130
13	130-140
9	140-150
3	150-160

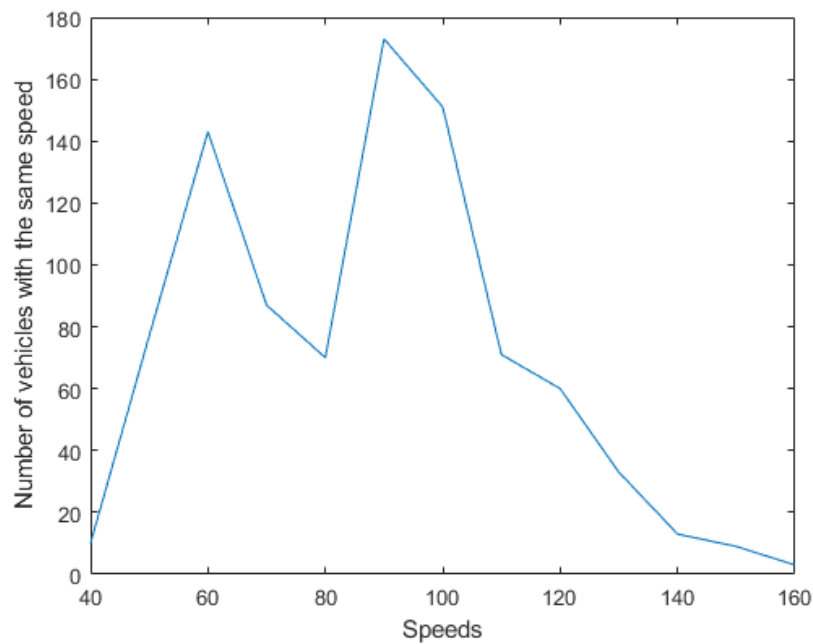
Πίνακας 3.1: Ταχύτητες παρούσας εργασίας - ομαδοποιημένες

Αριθμός Οχημάτων	Εύρος Ταχυτήτων(km/h)
9	≤ 40
77	40-50
140	50-60
66	60-70
24	70-80
54	80-90
25	90-100
7	100-110
2	110-120
1	120-130

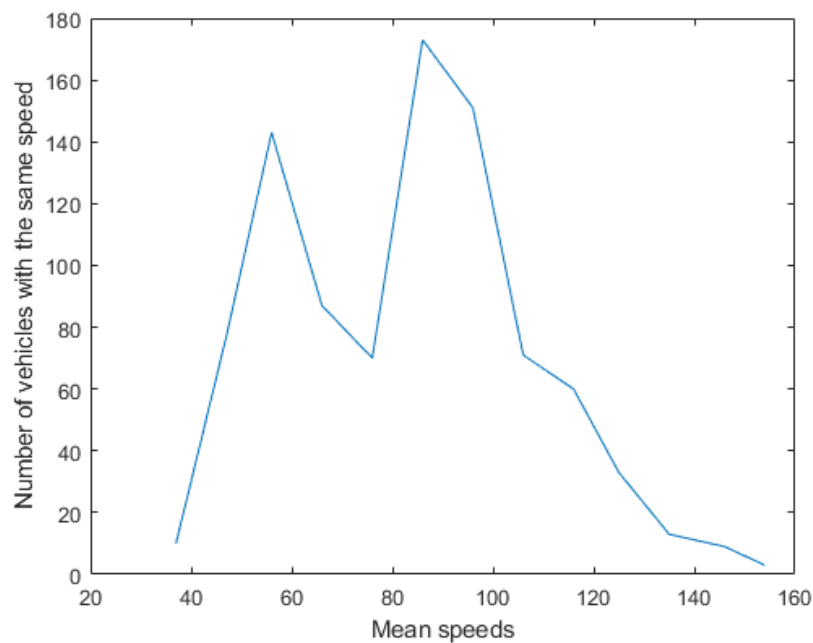
Πίνακας 3.2: Ταχύτητες προηγούμενης εργασίας(Λεωντσίνη) - ομαδοποιημένες

Αριθμός Οχημάτων	Εύρος Ταχυτήτων(km/h)
10	≤ 40
77	40-50
143	50-60
87	60-70
70	70-80
173	80-90
151	90-100
71	100-110
60	110-120
33	120-130
13	130-140
9	140-150
3	150-160

Πίνακας 3.3: Σύνολο ταχυτήτων - ομαδοποιημένο



(α') Για τις 13 ομάδες εύρους ταχυτήτων σχεδιάζεται το διάγραμμα σχέσης ταχύτητας με αριθμό οχημάτων. Οι ταχύτητες για κάθε ομάδα εύρους είναι με βήμα 10



(β') Για τις 13 ομάδες εύρους ταχυτήτων σχεδιάζεται το διάγραμμα σχέσης ταχύτητας με αριθμό οχημάτων. Οι ταχύτητες για κάθε ομάδα εύρους είναι η μέση τιμή της κάθε ομάδας

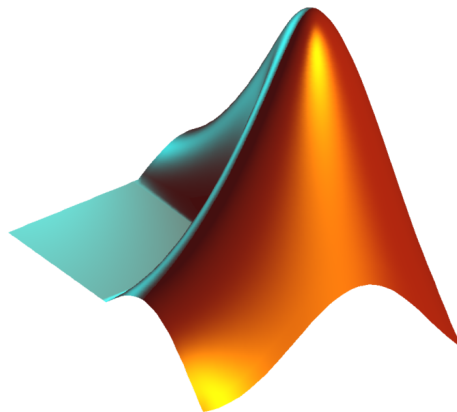
Κεφάλαιο 4

Μέρος Β

4.1 Εισαγωγή

Στο κεφάλαιο αυτό αναλύονται οι δυναμικές εξισώσεις κίνησης οχημάτων, το σύστημα ασαφούς λογικής fuzzy, η διαδικασία ενεργειακής αξιολόγησης τεσσάρων ηλεκτρικών οχημάτων, ο αλγόριθμος particle swarm optimization (PSO), η διαδικασία βέλτιστης τοποθέτησης σταθμών φόρτισης και το πλήθος των φορτιστών στους σταθμούς.

4.2 Εργαλείο Matlab



Σχήμα 4.1: Λογότυπο Matlab[4]

Το εργαλείο Matlab χρησιμοποιείται για την ενεργειακή αξιολόγηση των τεσσάρων αμιγώς ηλεκτροκίνητων αυτοκινήτων και για την εύρεση των βέλτιστων σημείων τοποθέτησης των σταθμών φόρτισης και του πλήθους των φορτιστών σε αυτά.

Το Matlab είναι μια προγραμματιστική πλατφόρμα που σχεδιάστηκε για μηχανικούς και επιστήμονες, για την ανάλυση και τον σχεδιασμό συστημάτων και προϊόντων. Η γλώσσα προγραμματισμού Matlab είναι βασισμένη σε πίνακες και μπορεί και πραγματοποιεί τις περισσότερες φυσικές εκφράσεις μαθηματικών υπολογισμών. Χρησιμοποιείται σε εφαρμογές όπως:

- Ανάλυση δεδομένων
- Ανάπτυξη αλγορίθμων
- Δημιουργία μοντέλων και εφαρμογών

Πρόκειται για ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο που χρησιμοποιείται σε εφαρμογές επιστημονικών υπολογισμών. Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων MATtrix LABoratory δηλαδή εργαστήριο πινάκων. Το MATLAB είναι ένα διαδραστικό πρόγραμμα για αριθμητικούς υπολογισμούς και εξεικόνιση δεδομένων με δυνατότητες προγραμματισμού που το καθιστούν ένα ισχυρό και χρήσιμο εργαλείο στις μαθηματικές και φυσικές επιστήμες. Στις αρχικές του εκδοχές δεν έκανε συμβολικούς υπολογισμούς. Στις νεότερες εκδοχές του, το πακέτο περιλαμβάνει εργαλεία που επιτρέπουν συμβολικούς υπολογισμούς. Όπως υποδηλώνεται και από το όνομα του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικού πίνακα κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κ.α.

Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Το MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας. Με διαφορετικά λόγια, δεν βρίσκει την ακριβή λύση αλλά μια προσεγγιστική λύση ενός προβλήματος. (βιβλίο [41])

4.3 Δυναμικές εξισώσεις οχημάτων

Από το βιβλίο [6] των Mehrdad Ehsani, Yimin Gao, Stefano Longo, Kambiz Ebrahimi εξάγονται οι δυναμικές εξισώσεις κίνησης με βάση τις οποίες υπολογίζεται η κατανάλωση ενέργειας σε ένα ηλεκτρικό όχημα. Οι εξισώσεις αυτές είναι οι παρακάτω:

$$F_{tr} = F_{ad} + F_{roll} + F_g + F_i$$

$$F_{ad} = \frac{1}{2} \times \rho \times A_f \times C_d \times (V - V_m)^2$$

$$F_{roll} = C_{rr} \times m_v \times g \times \cos(\theta)$$

$$F_g = m_v \times g \times \sin(\theta)$$

$$F_i = m_v \times a$$

όπου F_{tr} η ελκτική δύναμη (Tractive Force), F_{ad} η αεροδυναμική δύναμη (Aerodynamic Drag Force), F_{roll} η δύναμη της αντίστασης κύλισης (Rolling Resistance Force), F_g η δύναμη του βάρους του οχήματος σε σχέση με την κλίση του οδοστρώματος (Grading Resistance Force) και F_i η αδρανειακή δύναμη (Inertial Force).

Ένα όχημα κινούμενο υπό κάποια ταχύτητα δέχεται από τον αέρα μια δύναμη που επιρρεάζει την κίνησή του. Αυτή η δύναμη ονομάζεται αεροδυναμική δύναμη. Η αεροδυναμική δύναμη είναι συνάρτηση της ταχύτητας V σε m/s , της εμπρόσθιας επιφάνειας του οχήματος (A_f) σε m^2 , το σχήμα του οχήματος, την πυκνότητα του αέρα (ρ) σε $\frac{kg}{m^3}$, η οποία εξαρτάται από την θερμοκρασία του περιβάλλοντος 4.1, τον συντελεστή οπισθέλκουσας του αέρα (C_d) 4.2, και το V_m , που είναι η ταχύτητα του αέρα εκφρασμένη και αυτή σε m/s , που είτε ωθεί το όχημα προς την κατεύθυνση που κινείται, οπότε έχει αρνητικό πρόσημο, είτε του αντιστέκεται, οπότε έχει θετικό πρόσημο.

θερμοκρασία ($^{\circ}C$)	Πυκνότητα αέρα($\frac{kg}{m^3}$) (ad)
0	1.292
5	1.269
10	1.247
15	1.225
20	1.204
25	1.184
30	1.165
40	1.127

Πίνακας 4.1: Σχέση πυκνότητας αέρα με θερμοκρασία -

Πηγή (<https://www.engineersedge.com/calculators/air-density.htm>)

Τύπος οχήματος	συντελεστή οπισθέλκουσας του αέρα (Cd)
Ανοιχτό κάμπριο	0.5-0.7
Μικρό φορτηγό	0.5-0.7
Τύπου ponton	0.4-0.55
Σφηνοειδές σχήμα	0.3-0.4
Σχήμα K	0.23

Πίνακας 4.2: συντελεστή οπισθέλκουσας του αέρα σε σχέση με τον τύπο οχήματος -

Πηγή([6])

Η αντίστασης κύλισης των λάστιχων πάνω σε σκληρές επιφάνειες προκαλείται κυρίως λόγω υστέρισης του υλικού του λάστιχου. Αυτό οφείλεται στην εκτροπή του σκελετού ενώ το ελαστικό κυλάει. Η υστέρηση προκαλεί ασύμμετρη κατανομή στις δυνάμεις που αλληλεπιδρούν με το έδαφος. Ο συντελεστής αντίστασης κύλισης (C_{rr}) λοιπόν είναι συνάρτηση του υλικού του λάστιχου, της δομής του, της θερμοκρασίας του, της πίεσης αέρα σε αυτό, της σκληρότητας του δρόμου, το υλικό του δρόμου καθώς και την παρουσία ή απουσία υγρού στον δρόμο. Οι τυπικές τιμές του συντελεστής αντίστασης κύλισης φαίνονται στον πίνακα 4.3. Σήμερα, για οικονομία καυσίμου, έχουν αναπτυχθεί στα επιβατικά αυτοκίνητα λάστιχα χαμηλής πίεσης.

Στην εξίσωση δύναμης του βάρους του οχήματος το m_v εκφράζει το βάρος του

οχήματος σε kg, το g την επιτάχυνση της βαρύτητας και το ϑ του οδοστρώματος πάνω στο οποίο κινείται το όχημα και τέλος στην εξίσωση της αδρανειακής δύναμης το α αποτελεί την επιτάχυνση του οχήματος εκφρασμένη σε $\frac{m}{s^2}$.

Συνθήκες	Συντελεστής αντίστασης κύλισης (Crr)
δρόμος με τσιμέντο ή ασφαλτο	0.013
δρόμος με χαλίκι ή παλιά ασφαλτο	0.02
ασφαλτικό σκυρόδεμα	0.025
μη ασφαλτοστρωμένος δρόμος	0.05

Πίνακας 4.3: Τιμή συντελεστή αντίστασης κύλισης υπό συνθήκες - Πηγή([6])

Συνεπώς, από το άθροισμα των παραπάνω εξισώσεων υπολογίζεται η ελκτική δύναμη με βάση την οποία υπολογίζεται την ενέργεια που καταναλώνεται από ένα όχημα με μία δεδομένη ταχύτητα. Πιο συγκεκριμένα, το γινόμενο της ελκτικής δύναμης με την ταχύτητα του οχήματος εκφρασμένη σε m/s ισούται με την ενέργειας που καταναλώνεται.

$$P = F_{tr} \times V$$

Έχοντας λοιπόν την πληροφορία για την ενέργειας που καταναλώνεται για μια δεδομένη ταχύτητα και τον χρόνο που απαιτείται έτσι ώστε ένα όχημα να διανύσει μια απόσταση με την δεδομένη αυτή ταχύτητα, μπορεί να υπολογιστεί το ποσοστό φόρτισης της μπαταρίας που απομένει στο όχημα.

4.4 Διαδικασία ενεργειακής αξιολόγησης

4.4.1 Ηλεκτρικά αυτοκίνητα

Τα τέσσερα αμιγώς ηλεκτρικά αυτοκίνητα που επιλέχθηκαν είναι το Audi Q4 e-tron 35 (4.4), Fiat 500e Hatchback(4.5), Nissan Leaf(4.6) και το Mazda MX-30(4.7). Οι διαδρομές που μελετούνται είναι η διαδρομή Χανιά - Ρέθυμνο(4.2) μήκους 65 χιλιομέτρων, η διαδρομή Χανιά - Ηράκλειο(4.3) μήκους 142 χιλιομέτρων και η διαδρομή Ρέθυμνο - Ηράκλειο(4.4) μήκους 77 χιλιομέτρων. Παρακάτω παρουσιάζονται τα χαρακτηριστικά των επιλεγμένων αυτοκινήτων καθώς και οι διαδρομές.

Audi Q4 e-tron 35	
Χωριτηκότητα μπαταρίας	55kWh
Χωριτηκότητα μπαταρίας για οδήγηση	52kWh
Αυτονομία	285km
Βάρος	1965kg
Μέγιστο μεταφερόμενο βάρος	585kg
Συντελεστής οπισθέλκουσας	0.26
Εμπρόσθια επιφάνεια	3,0437m ²

Πίνακας 4.4: Χαρακτηριστικά Audi Q4 e-tron 35 - Πηγή([7])

Fiat 500e Hatchback	
Χωριτηκότητα μπαταρίας	42kWh
Χωριτηκότητα μπαταρίας για οδήγηση	37,3kWh
Αυτονομία	235km
Βάρος	1365kg
Μέγιστο μεταφερόμενο βάρος	400kg
Συντελεστής οπισθέλκουσας	0.311
Εμπρόσθια επιφάνεια	2,5699m ²

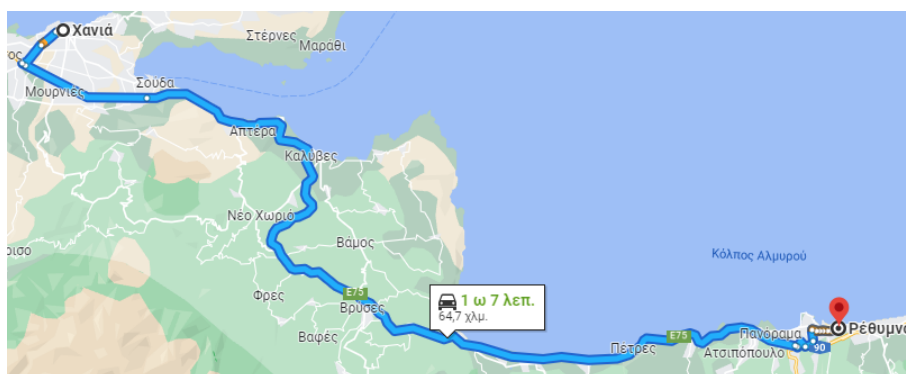
Πίνακας 4.5: Χαρακτηριστικά Fiat 500e Hatchback - Πηγή([8])

Nissan Leaf	
Χωριτηκότητα μπαταρίας	40kWh
Χωριτηκότητα μπαταρίας για οδήγηση	37kWh
Αυτονομία	225km
Βάρος	1580kg
Μέγιστο μεταφερόμενο βάρος	490kg
Συντελεστής οπισθέλκουσας	0.32
Εμπρόσθια επιφάνεια	2,7356m ²

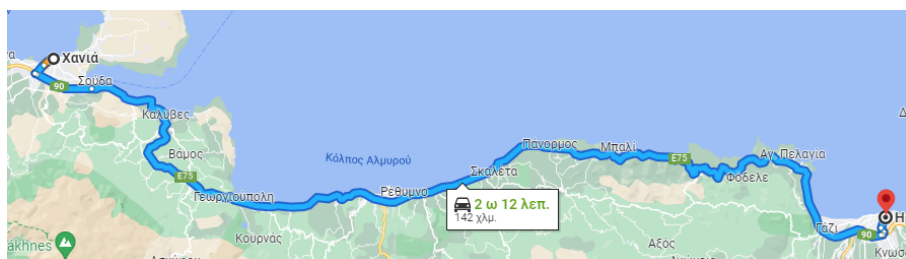
Πίνακας 4.6: Χαρακτηριστικά Nissan Leaf - Πηγή([9])

Mazda MX-30	
Χωριτηρότητα μπαταρίας	35, 5kWh
Χωριτηρότητα μπαταρίας για οδήγηση	30kWh
Αυτονομία	170km
Βάρος	1720kg
Μέγιστο μεταφερόμενο βάρος	474kg
Συντελεστής οπισθέλκουσας	0.29
Εμπρόσθιας επιφάνεια	2, 8182 ^m 2

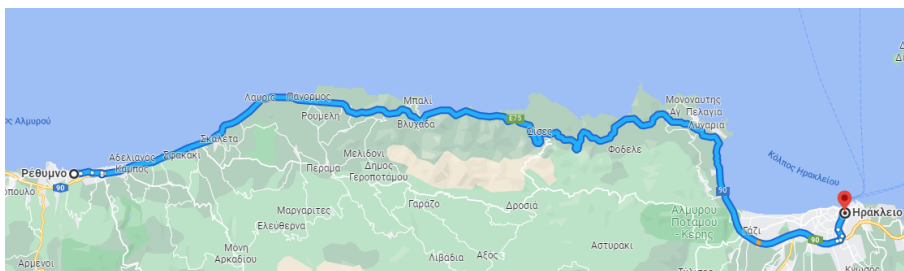
Πίνακας 4.7: Χαρακτηριστικά Mazda MX-30 - Πηγή([10])



Σχήμα 4.2: Διαδρομή Χανιά - Ρέθυμνο (Α)



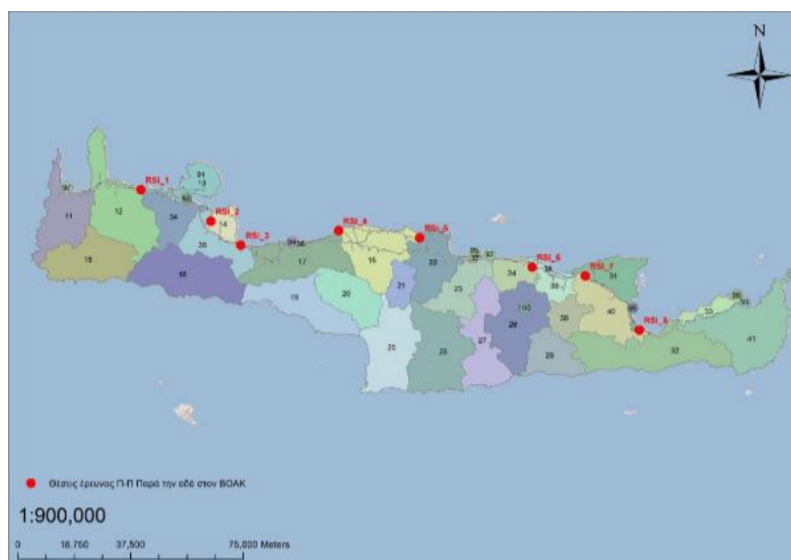
Σχήμα 4.3: Διαδρομή Χανιά - Ηράκλειο (RS2 – Α)



Σχήμα 4.4: Διαδρομή Ρέθυμνο - Ηράκλειο ($R = \frac{RS2-A}{2}$)

4.4.2 Ωριαίος κυκλοφοριακός φόρτος

Στοιχεία σχετικά με το ωριαίο κυκλοφοριακό φόρτο, χρησιμοποιήθηκαν από την εργασία Έκτιμηση κατανάλωσης ενέργειας ηλεκτροκίνητων οχημάτων μέσω εναέριας παρατήρησης της κυκλοφορίας, όπου αναφέρεται πως τα στοιχεία αυτά προέρχονται από έρευνα του υπουργείου υποδομών και συγκοινωνιών για τον BOAK για το έτος 2018 και χρησιμοποιούνται μετά από επικοινωνία με την διεύθυνση Δ16 του συγκεκριμένου υπουργείου. Συγκεκριμένα, τοποθετούνται 8 αυτογραφικά μηχανήματα κατά μήκος του BOAK για την καταμέτρηση των οχημάτων που κινούνται σε κάθε περιοχή όπως φαίνεται στο σχήμα 4.5. Οι σταθμοί αυτοί από τα αριστερά προς τα δεξιά είναι ο RS-1, RS-2, RS-3, RS-4, RS-5, RS-6, RS-7, RS-8.



Σχήμα 4.5: Τοποθεσίες αυτογραφικών μηχανιμάτων που χρησιμοποιούνται για την καταγραφή των κινούμενων οχημάτων

Στους πίνακες 4.6, 4.7 παρουσιάζεται το άθροισμα των κυκλοφοριακών φόρτων για τους

συγκεκριμένους σταθμούς έρευνας ανά ώρα και ανά κατεύθυνση για την θερινή και χειμερινή περίοδο αντίστοιχα. Επίσης, στον πίνακα 4.8 απεικονίζεται η μέση ημερήσια κυκλοφορία και για τις δύο κατευθύνσεις ανά σταθμό και μήνα μετρήσεων.

ΩΡΑ	Προς Ανατολή	Προς Δύση	Σύνολο
00:00-01:00	1.171	1.099	2.270
01:00-02:00	718	697	1.416
02:00-03:00	398	387	785
03:00-04:00	266	241	507
04:00-05:00	280	271	551
05:00-06:00	485	476	961
06:00-07:00	1.327	1.169	2.496
07:00-08:00	2.443	2.220	4.662
08:00-09:00	3.092	3.061	6.153
09:00-10:00	3.597	3.954	7.550
10:00-11:00	4.063	4.523	8.586
11:00-12:00	4.217	4.608	8.825
12:00-13:00	3.851	4.046	7.897
13:00-14:00	3.488	3.559	7.047
14:00-15:00	3.520	3.407	6.926
15:00-16:00	3.688	3.705	7.393
16:00-17:00	3.699	3.640	7.339
17:00-18:00	3.863	3.696	7.559
18:00-19:00	4.080	3.714	7.794
19:00-20:00	4.057	3.410	7.466
20:00-21:00	3.311	2.829	6.140
21:00-22:00	2.655	2.259	4.914
22:00-23:00	2.014	1.870	3.884
23:00-24:00	1.649	1.503	3.152
ΣΥΝΟΛΟ	61.931	60.341	122.272

Σχήμα 4.6: Αριθμός κινούμενων οχημάτων στον BOAK κατά την θερινή περίοδο (μέση τιμή μήνα)

ΩΡΑ	Προς Ανατολή	Προς Δύση	Σύνολο
00:00-01:00	240	200	441
01:00-02:00	111	122	233
02:00-03:00	71	67	138
03:00-04:00	60	67	128
04:00-05:00	85	106	191
05:00-06:00	284	240	523
06:00-07:00	765	621	1.386
07:00-08:00	1.700	1.711	3.411
08:00-09:00	2.235	2.024	4.259
09:00-10:00	2.109	1.971	4.080
10:00-11:00	1.874	1.838	3.712
11:00-12:00	1.779	1.647	3.426
12:00-13:00	1.782	1.659	3.441
13:00-14:00	1.811	1.763	3.574
14:00-15:00	1.966	1.881	3.847
15:00-16:00	2.117	2.057	4.175
16:00-17:00	2.057	2.012	4.069
17:00-18:00	1.875	1.830	3.704
18:00-19:00	1.535	1.514	3.048
19:00-20:00	1.172	1.207	2.380
20:00-21:00	974	924	1.898
21:00-22:00	829	776	1.605
22:00-23:00	585	522	1.107
23:00-24:00	395	338	733
ΣΥΝΟΛΟ	28.411	27.099	55.509

Σχήμα 4.7: Αριθμός κινούμενων οχημάτων στον ΒΟΑΚ κατά την χειμερινή περίοδο (μέση τιμή μήνα)

Θέση Έρευνας	Ιούλιος 2018	Αύγουστος 2018	Σεπτέμβριος 2018	Νοέμβριος 2018
ATC-1 (RSI-1)	22.244	22.741	18.892	11.500
ATC-2A (RSI-2)	17.298	18.001	14.980	7.990
ATC-3 (RSI-3)	17.492	18.107	15.164	7.136
ATC-4 (RSI-4)	18.335	18.218	16.438	8.069
ATC-5 (RSI-5)	14.432	14.492	12.681	5.804
ATC-6 (RSI-6)	22.102	22.652	19.694	9.772
ATC-7 (RSI-7)	13.890	14.235	12.317	5.596
ATC-8 (RSI-8)	8.030	8.489	6.896	3.479

Σχήμα 4.8: Μέση ημερήσια κυκλοφορία και στις 2 κατευθύνσεις ανά σταθμό και μήνα

Εντούτοις, τα δεδομένα που αξιοποιήθηκαν στην διπλωματική εργασία ήταν από 2 από τους 8 σταθμούς. Ειδικότερα οι σταθμοί των οποίων τα δεδομένα αξιοποιήθηκαν ήταν ο RS-2 και ο RS-4 καθώς εξετάζονται οι διαδρομές Χανιά-Ρέθυμνο και Χανιά-Ηράκλειο. Για να υπολογιστεί ο αριθμός των οχημάτων που κινούνται σε κάθε σταθμό απαιτείται το ποσοστό του κάθε σταθμού σε σχέση με το σύνολο των οχημάτων που κινούνται σε όλους τους σταθμούς. Για την θερινή περίοδο λαμβάνεται ως μήνας αναφοράς ο Αύγουστος ενώ η χειμερινή περίοδος δεν λαμβάνεται υπόψιν εξαιτίας της τεράστιας διαφοράς του αριθμού των οχημάτων με την θερινή περίοδο. Για παράδειγμα, από τον σταθμό RS-2 περνάνε 18001 οχήματα, ενώ μέσα στην μέρα περνάνε συνολικά 136935 από όλους τους σταθμούς. Συνεπώς από τον RS-2 διέρχεται συνολικά $a = \frac{18001}{136935} = 0.131$ δηλαδή 13.1%. Επομένως, για τον υπολογισμό των οχημάτων που διέρχονται από τον σταθμό κάθε ώρα απαιτείται ο πολλαπλασιασμός του 13,1% με το σύνολο των οχημάτων που κινούνται εκείνη την ώρα προς μία κατεύθυνση. Σύμφωνα με τα παραπάνω δημιουργούνται οι πίνακες 4.8, 4.9 όπου αναφέρεται ο αριθμός των οχημάτων που κινούνται στους σταθμούς RS-2 και RS-4.

Ώρα	Προς Ανατολή
00:00 - 01:00	153
01:00 - 02:00	94
02:00 - 03:00	52
03:00 - 04:00	35
04:00 - 05:00	37
05:00 - 06:00	63
06:00 - 07:00	173
07:00 - 08:00	320
08:00 - 09:00	405
09:00 - 10:00	471
10:00 - 11:00	532
11:00 - 12:00	552
12:00 - 13:00	504
13:00 - 14:00	457
14:00 - 15:00	532
15:00 - 16:00	483
16:00 - 17:00	484
17:00 - 18:00	506
18:00 - 19:00	534
19:00 - 20:00	531
20:00 - 21:00	433
21:00 - 22:00	347
22:00 - 23:00	263
23:00 - 24:00	216

Πίνακας 4.8: Ωριαίος κυκλοφοριακός φόρτος για τον σταθμό RS-2

Ώρα	Προς Ανατολή
00:00 - 01:00	155
01:00 - 02:00	95
02:00 - 03:00	52
03:00 - 04:00	35
04:00 - 05:00	37
05:00 - 06:00	64
06:00 - 07:00	176
07:00 - 08:00	324
08:00 - 09:00	411
09:00 - 10:00	478
10:00 - 11:00	540
11:00 - 12:00	560
12:00 - 13:00	512
13:00 - 14:00	463
14:00 - 15:00	468
15:00 - 16:00	490
16:00 - 17:00	491
17:00 - 18:00	513
18:00 - 19:00	542
19:00 - 20:00	539
20:00 - 21:00	440
21:00 - 22:00	353
22:00 - 23:00	267
23:00 - 24:00	219

Πίνακας 4.9: Ωριαίος κυκλοφοριακός φόρτος για τον σταθμό RS-4

Για τον υπολογισμό των συνολικών οχημάτων, που πραγματοποιούν την διαδρομή Χανιά-Ρέθυμνο κάθε ώρα, απαιτείται η λύση της εξίσωσης

$$RS2 - A + R = RS4 \Rightarrow RS2 - A + \frac{RS2 - A}{2} = RS4 \Rightarrow A = \frac{-2RS4 + 3RS2}{3}$$

, όπου RS-2 αποτελεί τον αριθμό των οχημάτων που κινούνται στον σταθμό RS-2, δηλαδή τον αριθμό των οχημάτων που φεύγουν από τα Χανιά, A αποτελεί τον αριθμό των οχημάτων που πραγματοποιούν την διαδρομή Χανιά-Ρέθυμνο, R αποτελεί τον αριθμό των οχημάτων που φεύγουν από το Ρέθυμνο για την πόλη του Ηρακλείου και τέλος το RS4 αποτελεί τον αριθμό των οχημάτων που διέρχονται από τον σταθμό RS-4.

Γίνεται η θεώριση ότι το R έχει την μισή τιμή του συνόλου των αυτοκινήτων που πραγματοποιούν την διαδρομή Χανιά-Ηράκλειο($\frac{RS2-A}{2}$). Έτσι λοιπόν προκύπτει πίνακας 4.10 που απεικονίζει τον αριθμό των οχημάτων που πραγματοποιούν την διαδρομή Χανιά-Ρέθυμνο(A).

Ώρα	Προς Ανατολή
00:00 - 01:00	50
01:00 - 02:00	31
02:00 - 03:00	18
03:00 - 04:00	12
04:00 - 05:00	13
05:00 - 06:00	21
06:00 - 07:00	56
07:00 - 08:00	104
08:00 - 09:00	131
09:00 - 10:00	153
10:00 - 11:00	172
11:00 - 12:00	179
12:00 - 13:00	163
13:00 - 14:00	149
14:00 - 15:00	149
15:00 - 16:00	157
16:00 - 17:00	157
17:00 - 18:00	164
18:00 - 19:00	173
19:00 - 20:00	172
20:00 - 21:00	140
21:00 - 22:00	112
22:00 - 23:00	70
23:00 - 24:00	85

Πίνακας 4.10: Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Χανιά-Ρέθυμνο (Α)

Ο συνολικός αριθμός οχημάτων, που πραγματοποιούν την διαδρομή Χανιά-Ηράκλειο κάθε ώρα προκύπτει από τον υπολογισμό της αφαίρεσης του αριθμού των οχημάτων που κινούνται στον σταθμό RS-2 και των οχημάτων που μένουν στο Ρέθυμνο. $(RS2 - A)$

4.11

Ώρα	Προς Ανατολή
00:00 - 01:00	103
01:00 - 02:00	63
02:00 - 03:00	34
03:00 - 04:00	23
04:00 - 05:00	24
05:00 - 06:00	42
06:00 - 07:00	117
07:00 - 08:00	216
08:00 - 09:00	274
09:00 - 10:00	318
10:00 - 11:00	360
11:00 - 12:00	373
12:00 - 13:00	341
13:00 - 14:00	308
14:00 - 15:00	312
15:00 - 16:00	326
16:00 - 17:00	327
17:00 - 18:00	342
18:00 - 19:00	361
19:00 - 20:00	359
20:00 - 21:00	293
21:00 - 22:00	235
22:00 - 23:00	178
23:00 - 24:00	146

Πίνακας 4.11: Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Χανιά-Ηράκλειο($RS2-A$)

Τέλος, ο αριθμός των οχημάτων για τους οποίους πραγματοποιείται ενεργειακή αξιολόγηση για την διαδρομή Ρέθυμνο-Ηράκλειο, προκύπτει από τον υπολογισμό της εξίσωσης $R = \frac{RS2-A}{2}$, δεδομένου ότι υπάρχει η παραδοχή ότι τα οχήματα που φε-

ύγουν από το Ρέθυμνο είναι τα μισά σε σχέση με αυτά που φεύγουν από τα Χανιά με προορισμό το Ηράκλειο. Σύμφωνα με τα παραπάνω προκύπτει ο πίνακας 4.12.

Ώρα	Προς Ανατολή
00:00 - 01:00	52
01:00 - 02:00	32
02:00 - 03:00	18
03:00 - 04:00	12
04:00 - 05:00	13
05:00 - 06:00	22
06:00 - 07:00	59
07:00 - 08:00	108
08:00 - 09:00	137
09:00 - 10:00	160
10:00 - 11:00	180
11:00 - 12:00	187
12:00 - 13:00	171
13:00 - 14:00	155
14:00 - 15:00	156
15:00 - 16:00	164
16:00 - 17:00	164
17:00 - 18:00	171
18:00 - 19:00	181
19:00 - 20:00	180
20:00 - 21:00	147
21:00 - 22:00	118
22:00 - 23:00	89
23:00 - 24:00	73

Πίνακας 4.12: Ωριαίος κυκλοφοριακός φόρτος για την διαδρομή Ρέθυμνο-Ηράκλειο(R)

4.4.3 Χρήση συνάρτησης Gendist

Η gendist είναι μια συνάρτηση της Matlab που παράγει έναν $N \times M$ πίνακα που περιέχει τυχαίους αριθμούς, χρησιμοποιώντας μια διακριτή κατανομή πιθανότητας. Στην παρούσα διπλωματική αξιοποιήθηκε για την δημιουργία του πίνακα που περιέχει τις ταχύτητες των οχημάτων που κινούνται κάθε ώρα της ημέρας, το ποσοστό αρχικής φόρτισης της μπαταρίας αυτών καθώς και το άγχος των οδηγών, με τρόπο αντίστοιχο με αυτόν της διπλωματικής 'Εκτίμηση κατανάλωσης ενέργειας ηλεκτροκίνητων οχημάτων μέσω εναέριας παρατήρησης της κυκλοφορίας'. Αναλυτικότερα, η gendist(P,N,M) δέχεται ως είσοδο τον πίνακα P που περιέχει τιμές που σχηματίζουν μία διακριτή κατανομή πιθανότητας για κάθε θέση του P. Η συνάρτηση δίνει ως έξοδο έναν $N \times M$ πίνακα που περιέχει ακεραίους αριθμούς που αντιστοιχούν στις θέσεις του P και οι οποίοι επιλέγονται τυχαία με βάση την κατανομή. Οι τιμές του πίνακα P κανονικοποιούνται και τα N,M του πίνακα εξόδου παίρνουν τιμές μεγαλύτερες ή ίσες του 1. [42]

Για καλύτερη κατανόηση δίνεται παρακάτω ένα παράδειγμα. Αν $P = [0.2 \ 0.4 \ 0.4]$, το $\text{sum}(P) = 1$, τότε το αποτέλεσμα της gendist μπορεί να πάρει τις τιμές 1,2 και 3, που αναφέρονται στους αντίστοιχους δείκτες του P. Αν κληθεί η gendist ως $T = \text{gendist}(P, 1, 10)$ τότε η έξοδος T κατά μέσο όρο θα περιέχει δύο φορές το 1, τέσσερις φορές το 2 και τέσσερις φορές το 3, σύμφωνα με τις τιμές του P. Δηλαδή μια πιθανή έξοδος θα μπορούσε να είναι $T = 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 1 \ 3 \ 1 \ 2$. Με την εντολή $Y = X(T)$ δημιουργείται ένας πίνακας μεγέθους ίδιο με αυτό το T, στο παρόν παράδειγμα Y πίνακας 1×10 , με τον Y να παίρνει τιμές από τον πίνακα X με βάση τον δείκτη που παίρνει από το T.

Στον πίνακα 3.3 φαίνεται ο αριθμός των ταχυτήτων σε σχέση με την ομάδα εύρους ταχύτητας που ανοίκουν. Για τα οχήματα κάθε ομάδας εξάχθηκε η μέση τιμή της ταχύτητας. Το σύνολο των ταχυτήτων που εξάχθηκε στον BOAK είναι 900 ταχύτητες, συνεπώς η πιθανότητα εμφάνισης της μέσης τιμής ταχύτητας για κάθε ομάδα υπολογίζεται ως:

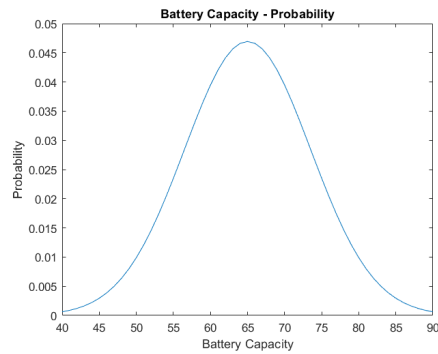
$$\text{probability} = \frac{\text{number of vehicles of the group}}{\text{total number of vehicles}}$$

Πιθανότητα εμφάνισης	Μέση τιμή ταχύτητας ομάδων
0.0111	37
0.0856	47
0.1589	56
0.0967	66
0.0778	76
0.1922	86
0.1678	96
0.0789	106
0.0667	116
0.0367	125
0.0144	135
0.01	146
0.0033	154

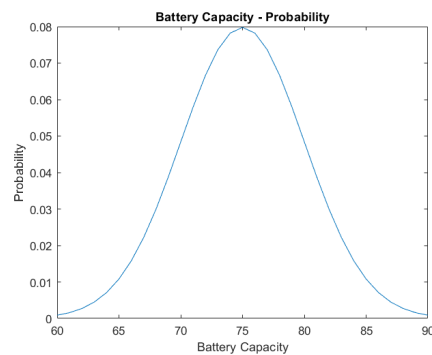
Πίνακας 4.13: Πίνακας πιθανοτήτων για κάθε ομάδα ταχυτήτων

Χρησιμοποιείται λοιπόν η gendist και δέχεται ως όρισμα τον πίνακα πιθανοτήτων 4.13 καθώς και τον αριθμό των ηλεκτρικών οχημάτων που κινούνται για κάποια ώρα της ημέρας. Ως έξοδος δίνεται ένας πίνακας που περιέχει ταχύτητες με την συχνότητα εμφάνισης κάθε ταχύτητας να εξαρτάται από τον πίνακα πιθανοτήτων με τρόπο αντίστοιχο όπως περιγράφηκε στο παράδειγμα 4.4.3. Αυτό έγινε για όλα τα οχήματα για κάθε ώρα της ημέρας.

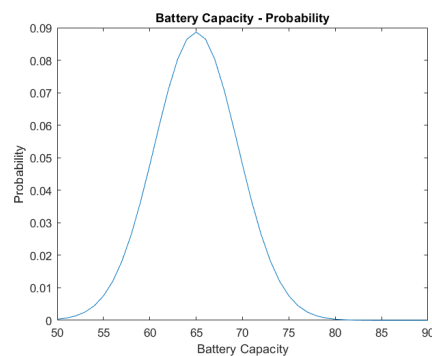
Έπειτα υποθέτουμε ότι όλα τα οχήματα ξεκινούν από τα Χανιά και με το ποσοστό φόρτισης της μπαταρίας του κάθε οχήματος να ακολουθεί γκαουσιανή κατανομή. Στην περίπτωση που πρόκειται για την διαδρομή Χανιά-Ρέθυμνο το ποσοστό αρχικής φόρτισης παίρνει τιμές μεταξύ 40-90% με μέση τιμή, $m_x = 0.65(65\%)$ και διασπορά $\sigma_x = 8.5$. Για την διαδρομή Χανιά-Ηράκλειο το ποσοστό αρχικής φόρτισης της μπαταρίας κυμαίνεται 60-90% με μέση τιμή $m_x = 0.75(75\%)$ και διασπορά $\sigma_x = 5$. Τέλος για την διαδρομή Ρέθυμνο-Ηράκλειο το ποσοστό αρχικής φόρτισης τις μπαταρίας παίρνει τιμές μεταξύ 50-90% με μέση τιμή $m_x = 0.65(65\%)$ και διασπορά $\sigma_x = 4.5$. Στο σχήμα 4.9 απεικονίζεται το ποσοστό αρχικής φόρτισης της μπαταρίας για κάθε διαδρομή.



(α') Ποσοστό φόρτισης της μπαταρίας στην διαδρομή Χανιά-Ρέθυμνο



(β') Ποσοστό φόρτισης της μπαταρίας στην διαδρομή Χανιά-Ηράκλειο



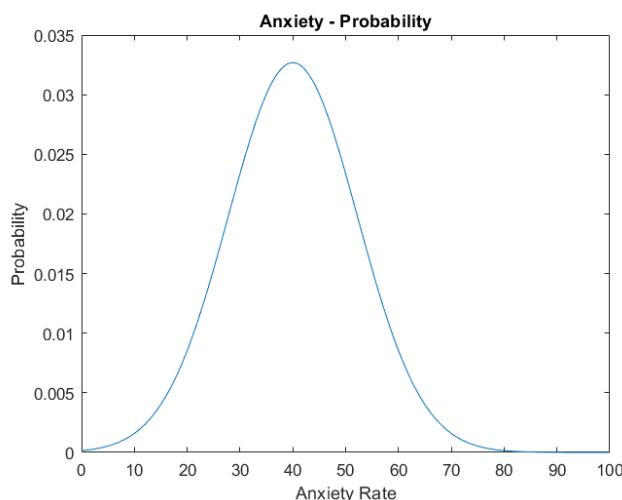
(γ') Ποσοστό φόρτισης της μπαταρίας στην διαδρομή Ρέθυμνο-Ηράκλειο

Σχήμα 4.9: Ποσοστό φόρτισης μπαταρίας διαδρομών

Όπως αναφέρεται στο 4.4.1 το μήκος κάθε διαδρομής σε χιλιόμετρα είναι διαφορετικό. Συνεπώς οι μεγαλύτερου μήκος διαδρομές απαιτούν μεγαλύτερο αρχικό ποσοστό φόρτισης. Πιο συγκεκριμένα τα οχήματα που πραγματοποιούν την διαδρομή Χανιά-Ηράκλειο θα έχουν το μεγαλύτερο ποσοστό φόρτισης, επειτά αυτά που ακολουθούν

την διαδρομή Ρέθυμνο-Ηράκλειο και τέλος αυτά που ακολουθούν την διαδρομή Χανιά-Ρέθυμνο. Για τις 24 ώρες της ημέρας για κάθε ώρα γνωρίζουμε τον αριθμό των οχημάτων που κινούνται κατά μήκος του BOAK. Αξιοποιώντας αυτή την πληροφορία με την βοήθεια της gendist δημιουργούνται πίνακες που περιέχουν τυχαίες τιμές για το ποσοστό αρχικής φόρτισης της μπαταρίας. Οι τιμές αυτές δημιουργούνται με βάση την γκαουσιανή κατανομή όπως αναφέρθηκε παραπάνω, ανάλογα με την διαδρομή. Τέλος το μέγεθος των πινάκων αυτών εξαρτάται από το πλήθος των οχημάτων που κινούνται κάθε ώρα.

Με την ίδια λογική δημιουργούνται και οι πίνακες που προσδιορίζουν το ποσοστό άγχους σε ότι αφορά την φόρτιση της μπαταρίας για κάθε οδηγό. Συγκεκριμένα για κάθε ώρα και για κάθε αυτοκίνητο ανατίθεται για τον οδηγό αυτό το ποσοστό άγχους. Οι τιμές του άγχους προσδιορίζονται επίσης από μια γκαουσιανή κατανομή με τιμές που κυμαίνονται από 0-100%. Με το 0% να αντιστοιχεί σε μηδενικό άγχος του οδηγού σε ότι αφορά το ποσοστό φόρτισης της μπαταρίας και το 100% να αντιστοιχεί σε οδηγό με πάρα πολύ μεγάλο άγχος. Για την γκαουσιανή κατανομή που χρησιμοποιήθηκε για ανάθεση ποσοστού άγχους σε οδηγούς, η μέση τιμή είναι $m_x = 0.4(40\%)$ και η διασπορά $\sigma_x = 12.2$. Στο σχήμα 4.10 φαίνεται η κατανομή ποσοστού άγχους του οδηγού σε ότι αφορά το ποσοστό φόρτισης της μπαταρίας.



Σχήμα 4.10: Κατανομή ποσοστού άγχους του οδηγού σε ότι αφορά το ποσοστό φόρτισης της μπαταρίας.

Για τα οχήματα που κινούνται τις 24 ώρες της ημέρας, με την βοήθεια της gendist δημιουργούνται πίνακες που περιέχουν το ποσοστό άγχους του κάθε οδηγού οχήματος

με το μέγεθος αυτών να εξαρτάται από τον αριθμό των οχημάτων που κινούνται κάθε ώρα.

4.4.4 Χρήση συστήματος ασαφούς λογικής (Fuzzy)

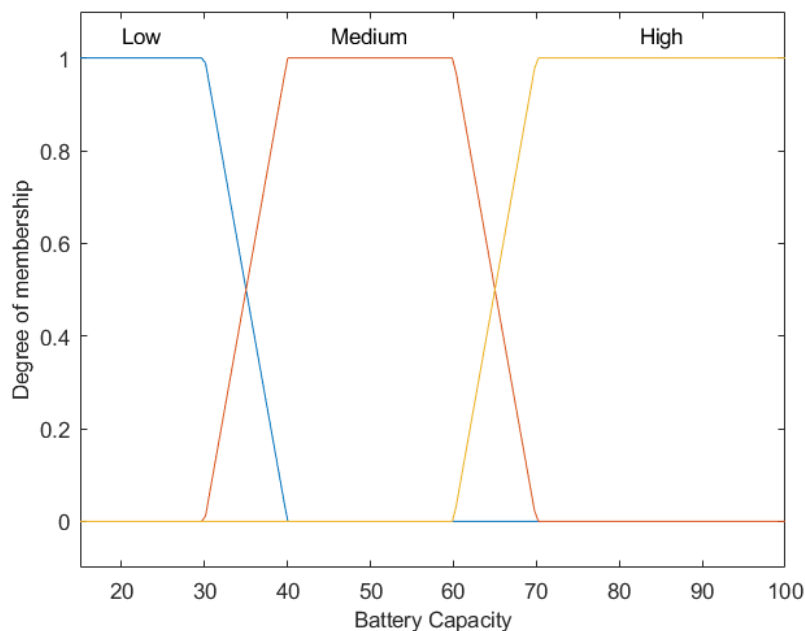
Για το χιλιόμετρο που ένας οδηγός αναζητά σταθμό φόρτισης, εφαρμόζεται ένα σύστημα ασαφούς λογικής (Fuzzy Inference System). Τα χαρακτηριστικά του συστήματος αυτού είναι τα προκαθορισμένα (default). Αναλυτικότερα ο τύπος του συστήματος είναι mamdani. Η μορφή του συστήματος μοιάζει με $\text{if } x \text{ is } A \text{ then } y \text{ is } B$ με τις εξόδους να αποτελούν ασαφές σύνολα. Για την σύνταξη των κανόνων χρησιμοποιούνται οι μέθοδοι `and_method` και `or_method`. Στην περίπτωση που ένας κανόνας έχει πολλές εισόδους τότε οι τελεστές OR και AND καθορίζουν το αποτέλεσμα του κανόνα. Όταν χρησιμοποιείται ο τελεστής AND και η `and_method` είναι η \min τότε η τιμή που αποτιμάται είναι η μικρότερη από τις τιμές εισόδου του κανόνα. Αντίθετα, στην περίπτωση του τελεστή OR με την μέθοδο `or_method` να είναι η \max , τότε η τιμή που αποτιμάται είναι η μεγαλύτερη από τις τιμές εισόδου του κανόνα. Για το συγκεκριμένο σύστημα ασαφούς λογικής (mamdani) αξιοποιείται η μέθοδος `imp_method` (Implication Method) για τον υπολογισμό ασαφούς συνόλου και η προκαθορισμένη επιλογή είναι η \min . Η μέθοδος `agg_method` (Aggregation Method) αποτελεί την διαδικασία κατά την οποία τα ασαφή σύνολα που αντιπροσωπεύουν τις εξόδους κάθε κανόνα συνδυάζονται σε ένα ενιαίο ασαφές σύνολο. Η προκαθορισμένη επιλογή, που προτείνεται για το σύστημα mamdani, είναι η \max . Τέλος η μέθοδος `defuzz_method` (Defuzzification Method) αποτιμά την τιμή εξόδου του συστήματος δεδομένου ενός συνόλου που δέχεται από την `agg_method`, με προκαθορισμένη τιμή για το σύστημα mamdani να είναι centroid. [43]

Στο εν λόγω σύστημα δίνονται ως είσοδοι το ποσοστό φόρτισης της μπαταρίας το οποίο μειώνεται κατά την διάρκεια της διαδρομής και το ποσοστό άγχους του οδηγού το οποίο παραμένει σταθερό. Ως έξοδος δίνεται η πιθανότητα κάποιος οδηγός να αναζητήσει σταθμό φόρτισης ή όχι. Το ποσοστό αρχικής φόρτισης της μπαταρίας χωρίζεται σε τρία επίπεδα 4.14

Ποσοστό φόρτισης της μπαταρίας	Χαρακτηρισμός επιπέδου στάθμης μπαταρίας
15-40%	Low
30-70%	Medium
60-100%	High

Πίνακας 4.14: Χαρακτηρισμός επιπέδων ποσοστού φόρτισης της μπαταρίας

Όσον αφορά την συνάρτηση μέλους(membership function) για την παραπάνω είσοδο είναι η trapmf, που μετατρέπει τις τιμές του κάθε επιπέδου σε μορφή τραπεζίου



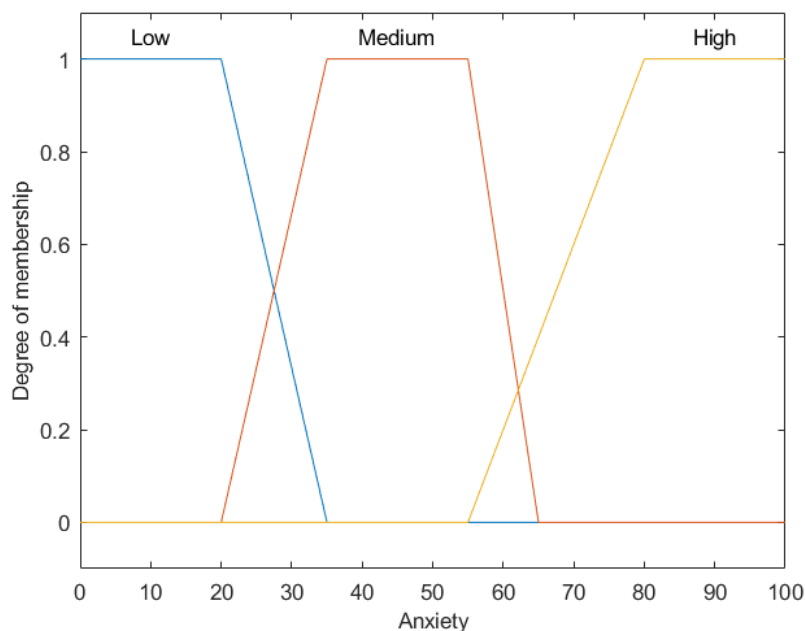
Σχήμα 4.11: Ποσοστό αρχικής φόρτισης μπαταρίας ως είσοδο στο σύστημα fuzzy

Το ποσοστό άγχους του οδηγού χωρίζεται σε τρία επίπεδα όπως φαίνεται στον πίνακα 4.15.

Ποσοστό άγχους οδηγού	Χαρακτηρισμός επιπέδου άγχους οδηγού
0-35%	Low
20-65%	Medium
55-100%	High

Πίνακας 4.15: Χαρακτηρισμός επιπέδων ποσοστού άγχους οδηγού

Όσον αφορά την συνάρτηση μέλους(membership function) για την παραπάνω είσοδο είναι η trapmf,που μετατρέπει τις τιμές του κάθε επιπέδου σε μορφή τραπεζίου.4.12



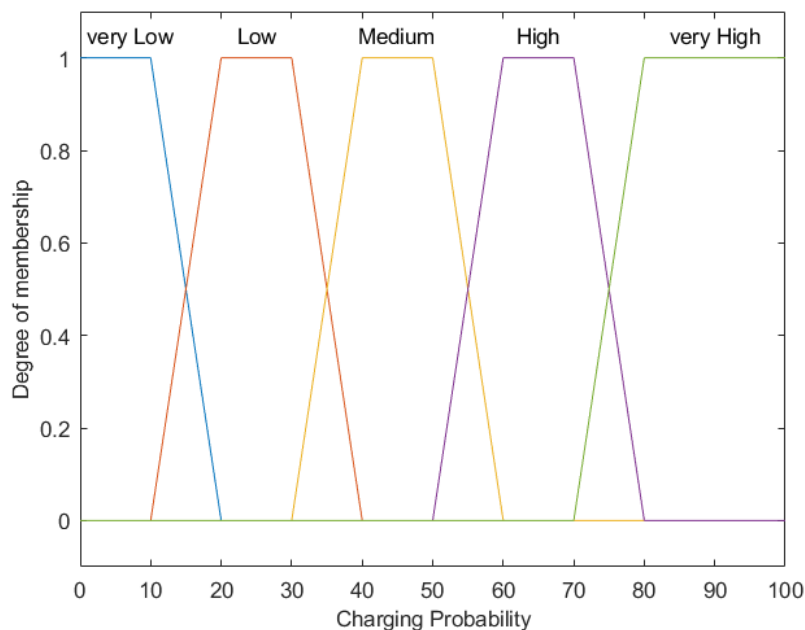
Σχήμα 4.12: Ποσοστό άγχους ως είσοδο στο σύστημα fuzzy

Τέλος η έξοδος του συστήματος, δηλαδή η πιθανότητα αναζήτησης σταθμού φόρτισης από τον οδηγό προκειμένου να φορτίσει την μπαταρία,χωρίζεται σε πέντε επίπεδα που απεικονίζονται στον παρακάτω πίνακα.4.16

Πιθανότητα αναζήτησης σταθμού φόρτισης	Χαρακτηρισμός επιπέδου
0-20%	Very low
10-40%	Low
30-60%	Medium
50-80%	High
70-100%	Very high

Πίνακας 4.16: Έξοδος συστήματος fuzzy

Η συνάρτηση μέλους(membership function) για την έξοδο του συστήματος είναι η trapmf,που μετατρέπει τις τιμές του κάθε επιπέδου σε μορφή τραπεζίου και απεικονίζεται 4.13



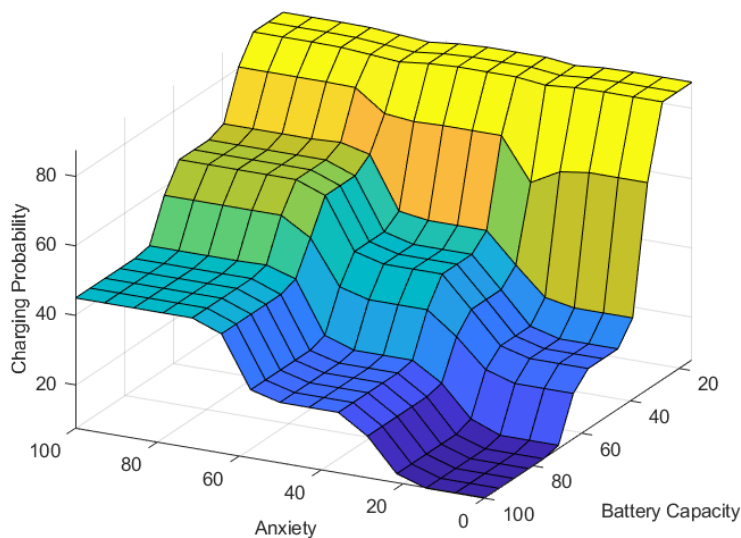
Σχήμα 4.13: Έξοδος συστήματος fuzzy

Η σύνταξη κανόνων είναι απαραίτητη για την λειτουργία του συστήματος. Κάθε μία από τις δύο εισόδους έχει τρία επίπεδα συνεπώς το ελάχιστο πλήθος κανόνων είναι εννέα και συνδέονται με τα πέντε επίπεδα της εξόδου. Είναι χρήσιμο να αναφερθεί ότι κάθε κανόνας έχει βάρος μεταξύ 0 και 1 και η τιμή αυτού μπορεί να αλλάξει σε περίπτωση που κάποιος κανόνας εμφανίζεται περισσότερο στο σύστημα. Ωστόσο, στο υπο μελέτη σύστημα επιλέχθηκε η ίδια τιμή για τα βάρη του κάθε κανόνα, με την τιμή αυτή να είναι 1. Οι κανόνες που συντάχθηκαν ήταν 9 και είναι οι παρακάτω:

1. If (Anxiety is Low) and (Battery Capacity is Low), then (Charging Probability is very High) (1)
2. If (Anxiety is Low) and (Battery Capacity is Medium), then (Charging Probability is Low) (1)
3. If (Anxiety is Low) and (Battery Capacity is High), then (Charging Probability is very Low) (1)
4. If (Anxiety is Medium) and (Battery Capacity is Low), then (Charging Probability is very High) (1)
5. If (Anxiety is Medium) and (Battery Capacity is Medium), then (Charging Probability is Medium) (1)

6. If (Anxiety is Medium) and (Battery Capacity is High), then (Charging Probability is Low) (1)
7. If (Anxiety is High) and (Battery Capacity is Low), then (Charging Probability is very High) (1)
8. If (Anxiety is High) and (Battery Capacity is Medium), then (Charging Probability is High) (1)
9. If (Anxiety is High) and (Battery Capacity is High), then (Charging Probability is Medium) (1)

Το σύστημα ασαφούς λογικής fuzzy δημιουργεί μια σχέση μεταξύ των τιμών εισόδου και εξόδου με βάση τα δεδομένα εισόδου, εξόδου και τους κανόνες που αναφέρθηκαν παραπάνω. Στο σχήμα 4.14 απεικονίζεται η έξοδος του συστήματος σε συνάρτηση με το ποσοστό φόρτισης της μπαταρίας και το ποσοστό άγχους του οδηγού.



Σχήμα 4.14: Έξοδος συστήματος fuzzy συναρτήσει των δύο εισόδων

4.4.5 Σύνοψη

Η ενεργειακή κατανάλωση υπολογίζεται από τις δυναμικές εξισώσεις οχημάτων που αναφέρθηκαν στην ενότητα 4.3 για τα οχήματα που κινούνται κάθε ώρα της ημέρας. Η τιμή του συντελεστή αντίστασης κύλισης είναι 0,02, καθώς τα οχήματα κινούνται σε

παλιά ασφάλτο, και η πυκνότητα του αέρα δέχεται την τιμή 1,184 που αντιστοιχεί σε 25 °C. Οι ταχύτητες των οχημάτων κατά την διάρκεια μιας διαδρομής θεωρούνται σταθερές με την ταχύτητα των οχημάτων να παίρνει την μέση τιμή από τις ομάδες του πίνακα 3.3. Για την διαδρομή Χανιά-Ρέθυμνο οι ταχύτητες παίρνουν τιμές από την μέση τιμή της ομάδας 3 μέχρι 7, δηλαδή $56 - 96 \frac{km}{h}$, για την διαδρομή Ρέθυμνο-Ηράκλειο οι ταχύτητες παίρνουν τιμές από την μέση τιμή της ομάδας 3 μέχρι 10, δηλαδή $56 - 125 \frac{km}{h}$ ενώ για την διαδρομή Χανιά-Ηράκλειο αποτελούν ένα συνδυασμό των παραπάνω, δηλαδή η μέση ταχύτητα για την διαδρομή Χανιά-Ρέθυμνο κυμαίνεται μεταξύ των $56 - 96 \frac{km}{h}$ και για την διαδρομή Ρέθυμνο-Ηράκλειο η μέση ταχύτητα κυμαίνεται μεταξύ $56 - 125 \frac{km}{h}$.

Στην συνέχεια, εκτιμάται η πτώση του ποσοστού φόρτισης της μπαταρίας του οχήματος ανά 5 λεπτά. Η πτώση αυτή για κάθε όχημα είναι σταθερή, καθώς γίνεται η θεώρηση ότι η ταχύτητα του κάθε οχήματος δεν αλλάζει. Με την εισαγωγή του ποσοστού φόρτισης της μπαταρίας ανά 5 λεπτά και το ποσοστό άγχους του οδηγού στο σύστημα fuzzy δίνεται ως έξοδος η πιθανότητα ο οδηγός να αναζητήσει σταθμό φόρτισης. Στην περίπτωση που η τιμή αυτή είναι μεγαλύτερη ή ίση του 75% τότε ο οδηγός αναζητά σταθμό φόρτισης. Με βάση την ταχύτητα του αυτοκινήτου και το λεπτό που το σύστημα fuzzy δίνει ως έξοδο πιθανότητα μεγαλύτερη ή ίση του 75% υπολογίζεται το χιλιόμετρο που ένας οδηγός αναζητάει σταθμό φόρτισης. Επιπρόσθετα, για κάθε οδηγό που αναζητάει σταθμό φόρτισης διατηρείται η πληροφορία σχετικά με το χιλιόμετρο αναζήτησης, την ώρα, τον αύξον αριθμό, την μάρκα του οχήματος, το ποσό φόρτισης της μπαταρίας όταν ο οδηγός αναζητάει σταθμό φόρτισης και την κατανάλωση ενέργειας ανα χιλιόμετρο. Η κατανάλωση ενέργειας ανα χιλιόμετρο σχετίζεται με τις δυναμικές εξισώσεις του κάθε οχήματος και την ταχύτητα.

4.5 Particle swarm optimazation Algoritm

Ο Particle swarm optimazation Algoritm (PSO) είναι μια μέθοδος βελτιστοποίησης καθολικής εύρεσης που διατυπώθηκε αρχικά από τους Doctor Kennedy, Eberhart το 1995. Αναπτύχθηκε από τη νοημοσύνη των σμήνων και βασίζεται στην έρευνα της συμπεριφοράς κίνησης του σμήνου πτηνών και του κοπαδιού ψαριών. Κατά τη διάρκεια αναζήτησης φαγητού, τα πτηνά είτε διασκορπίζονται είτε πηγαίνουν μαζί προτού εντοπίσουν το μέρος όπου μπορούν να βρουν φαγητό. Καθώς τα πτηνά αναζητούν για

φαγητό από μέρος σε μέρος, υπάρχει πάντα ένα πτηνό που μπορεί να εντοπίσει το φαγητό πολύ καλά στο μέρος που βρίσκεται. Το πτηνό αυτό έχει αυξημένη αντίληψη του μέρους όπου μπορεί να βρεθεί φαγητό, έχοντας τις καλύτερες πληροφορίες για τους πόρους των τροφίμων. Λόγω του γεγονότος ότι μεταδίδουν χρήσιμες πληροφορίες όσο αναζητούν για φαγητό από μέρος σε μέρος ανά πάσα στιγμή, τα πτηνά τελικά θα συσσωρευτούν στο μέρος όπου μπορεί να βρεθεί φαγητό. Όσον αφορά τον αλγόριθμο Particle swarm optimization, η λύση σμήνου (swarm) του αλγόριθμου συγκρίνεται με αυτή του σμήνου πτηνών καθώς η κίνηση των πτηνών από μέρος σε μέρος ισοδυναμεί με την ανάπτυξη της λύσης του αλγόριθμου, χρήσιμες πληροφορίες ισοδυναμούν με την καλύτερη λύση, και οι προμήθειες φαγητού ισοδυναμούν με την καλύτερη λύση κατά τη διάρκεια ολόκληρης της πορείας. Η καλύτερη λύση μπορεί να βρεθεί στον αλγόριθμο PSO από τη συνεργασία μεταξύ κάθε ατόμου. Το σωματίδιο που δε διαθέτει ποιότητα και όγκο πληροφοριών χρησιμοποιείται ως μονάδα, και το απλό μοτίβο συμπεριφοράς ρυθμίζεται για κάθε σωματίδιο για να δείξει την πολυπλοκότητα ολόκληρου του σμήνου σωματιδίων (particle swarm). Ο αλγόριθμος αυτός μπορεί να χρησιμοποιηθεί για τον υπολογισμό πολύπλοκων προβλημάτων βελτιστοποίησης.

Εξαιτίας των πολλών πλεονεκτημάτων συμπεριλαμβανομένου της απλότητας και της εύκολης υλοποίησης, ο αλγόριθμος μπορεί να χρησιμοποιηθεί ευρέως σε τομείς όπως είναι η βελτιστοποίηση συναρτήσεων, μοντέλα ταξινόμησης, μελέτη μηχανών, εκπαίδευση νευρωνικών δικτύων, επεξεργασία σήματος, ασαφών συστημάτων ελέγχου και αυτοματισμού.

Στην απλή του περίπτωση ο PSO αποτελείται από n σωματίδια (particles), και οι συνθήκες του κάθε σωματιδίου σχετίζονται με την πιθανή λύση στον n -διάστατο χώρο. Τα σωματίδια μεταβάλλουν τη θέση του σύμφωνα με τρεις βασικές αρχές:

- Διατήρησή της αδράνειας
- μεταβολή των συνθηκών τους, σύμφωνα με τη βέλτιστη θέση
- μεταβολή των συνθηκών τους, σύμφωνα με τη βέλτιστη θέση του σμήνου

Η θέση κάθε σωματιδίου του σμήνου επηρεάζεται από τη βέλτιστη θέση κατά τη διάρκεια της κίνησης (ατομική εμπειρία) αλλά και της θέσης του καλύτερου σωματιδίου στο περιβάλλον του (εμπειρία εγγύτητας). Όταν ολόκληρο το σμήνος των σωματιδίων περιβάλλει ένα σωματίδιο, η βέλτιστη θέση του περιβάλλοντος είναι ίση με αυτήν του

βέλτιστου σωματιδίου του σμήνους· αυτός ο αλγόριθμος ονομάζεται συνολικός PSO. Εάν χρησιμοποιείται ένα στενό περιβάλλον στον αλγόριθμο, αυτός ο αλγόριθμος ονομάζεται μερικός PSO.

Κάθε σωματίδιο μπορεί να απεικονιστεί από την τρέχουσα ταχύτητα και θέση του, την βέλτιστη ατομική θέση του και τη βέλτιστη θέση του περιγυρού του. Στον μερικό PSO, η ταχύτητα και η θέση του κάθε σωματιδίου μεταβάλλεται σύμφωνα με τις παρακάτω εξισώσεις (Shi Y, Eberhart R C, 1998):

$$v_{id}^{k+1} = v_{id}^k + c1r_1^k(pbest_{id}^k - x_{id}^k) + c2r_2^k(gbest_d^k - x_{id}^k)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}$$

Στην παραπάνω ισότητα, τα v_{id}^k, x_{id}^k διαχωρίζουν την ταχύτητα του σωματιδίου i από τις k φορές και τη n -διάστατη ποσότητα της θέσης του. Το $pbest_{id}^k$ αντιπροσωπεύει την ατομική i n -διάστατη ποσότητα στη βέλτιστη του θέση στις k φορές. Το $gbest_{id}^k$ είναι η n -διάστατη ποσότητα του σμήνου στη βέλτιστη του θέση. Για να αποφευχθεί η περίπτωση που τα σωματίδια του σμήνου είναι πολύ μακριά από τον χώρο έρευνας, η ταχύτητα του σωματιδίου που δημιουργείται σε κάθε κατεύθυνση παίρνει τιμές μεταξύ $[-V_{dmax}, V_{dmax}]$. Τα $c1, c2$ ρυθμίζουν την ταχύτητα του σωματιδίου. Συνήθως, τα $c1, c2$ είναι ίσα και αθροίζουν στο 2. Τα $r1, r2$ είναι τυχαίοι αριθμοί και παίρνουν τιμές από 0 έως 1.

Στον τοπικό PSO, αντί να αναζητούμε το βέλτιστο σωματίδιο του σμήνου, κάθε σωματίδιο θα αναζητήσει το βέλτιστο σωματίδιο στο περιβάλλον του για να ρυθμίσει την ταχύτητα και τη θέση του. Επίσης, η εξίσωση για την ταχύτητα και τη θέση του σωματιδίου είναι ίδια με αυτή του συνολικού PSO.

Τα εσωτερικά βάρη ενσωματώθηκαν στις εξισώσεις από τους (Eberhart R C, Shi Y, 1998; Eberhart R C, Shi Y, 2000). Ένα εσωτερικό βάρος w είναι ένας αναλογικός παράγοντας που σχετίζεται με την ταχύτητα της προηγούμενης φοράς και η εξίσωση για τη μεταβολή της ταχύτητας είναι η ακόλουθη:

$$v_{id}^{k+1} = wv_{id}^k + c1r_1^k(pbest_{id}^k - x_{id}^k) + c2r_2^k(gbest_d^k - x_{id}^k)$$

Η επιρροή της τελευταίας ταχύτητας στην τρέχουσα ταχύτητα μπορεί να ρυθμιστεί από τα εσωτερικά βάρη. Όσο μεγαλύτερο είναι το w τόσο μεγαλύτερη είναι η ικανότητα αναζήτησης του συνολικού PSO, και όσο μικρότερο είναι το w , τόσο μεγαλύτερη

είναι η ικανότητα αναζήτησης για τον μερικό PSO. Γενικά, το w είναι ίσο με 1, έτσι στην πιο πρόσφατη περίοδο των μερικών παραγωγών, υπάρχει έλλειψη της ικανότητας αναζήτησης για το μέρος. Πειραματικά αποτελέσματα υποδεικνύουν ότι ο PSO έχει τη μεγαλύτερη ταχύτητα σύγκλισης όταν το w παίρνει τιμές μεταξύ 0.8 και 1.2. Κατά τη διάρκεια δοκιμών, το w περιορίζεται από 0.9 έως 0.4 σύμφωνα με τη γραμμική μείωση, όπου ο PSO αναζητάει στον μεγάλο χώρο στην αρχή και εντοπίζει γρήγορα τη θέση όπου βρίσκεται η βέλτιστη λύση. Όσο το w μειώνεται, η ταχύτητα του σωματιδίου θα μειωθεί και αυτή για να αναζητήσει για τη λεπτή μερική. Η μέθοδος επιταχύνει την ταχύτητα σύγκλισης και η συνάρτηση της PSO βελτιώνεται. Όταν το πρόβλημα που τίθεται είναι περίπλοκο, αυτή η μέθοδος συνιστάτε για τη λύση του.

Στο άρθρο του (Clerc M, 1999) παρουσιάστηκε ένας PSO για την αύξηση του συντελεστή σύγκλισης με τις ακόλουθες εξισώσεις θέσης και ταχύτητας:

$$v_{id} = Xv_{id} + c1rand()(p_{id} - x_{id}) + c2rand()(p_{gd} - x_{id})$$

$$X = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}$$

$\phi = c1 + c2 > 4$. Γενικά, το ϕ είναι ίσο με 4.1 άρα το X είναι ίσο με 0.729. [44]

4.6 Διαδικασία βέλτιστης τοποθέτησης σταθμών φόρτισης

4.6.1 Υπολογισμός χρόνου φόρτισης οχημάτων, πλήθους οχημάτων και φορτιστών στους σταθμούς

Στην ενότητα 4.4 περιγράφεται η διαδικασία ενεργειακής αξιολόγησης κατά την οποία εξάγεται η πληροφορία για το χιλιόμετρο που αναζητάει για σταθμό φόρτισης κάθε οδηγός. Με βάση αυτή την πληροφορία, γνωρίζοντας το πλήθος των οχημάτων που αναζητούν για σταθμό σε κάποιο χιλιόμετρο και το χιλιόμετρο που τοποθετούνται οι σταθμοί φόρτισης εξάγεται η πληροφορία που αφορά τον αριθμό των αυτοκινήτων που σταματούν σε κάθε σταθμό. Τα οχήματα που αναζητούν για σταθμό φόρτισης αποθηκεύονται σε έναν πίνακα μαζί με ετικέτες που διατηρούν πληροφορία σχετικά με το χιλιόμετρο που αναζητούν για σταθμό, την ώρα που κινούνται, τον αύξοντα αριθμό του

οχήματος καθώς και ετικέτα σχετικά με τη μάρκα του οχήματος. Στη συνέχεια τα αυτοκίνητα που σταματούν στον ίδιο σταθμό, την ίδια ώρα αθροίζονται και δημιουργείται ο πίνακας ωριαίας κυκλοφορίας στους σταθμούς. Επιπλέον, για τα οχήματα που αναζητούν για σταθμούς φόρτισης υπολογίζεται ο χρόνος πλήρους και μερικής φόρτισης μαζί με τις ετικέτες που αφορούν τον χρόνο φόρτισης σε λεπτά, την ώρα που κινείται το όχημα, ο αύξοντας αριθμός καθώς και η μάρκα του οχήματος. Η διαδικασία υπολογισμού του χρόνου πλήρους και μερικής φόρτισης είναι η εξής: όταν ο οδηγός αρχίσει να αναζητάει για σταθμό φόρτισης, διατηρείται πληροφορία σχετικά με τη στάθμη της μπαταρίας στο χιλιόμετρο που ξεκίνησε η αναζήτηση καθώς και ο υπολογισμός που αφορά την κατανάλωση ενέργειας ανα χιλιόμετρο. Η κατανάλωση ενέργειας ανα χιλιόμετρο υπολογίζεται με βάση την ταχύτητα και τις δυναμικές εξισώσεις που σχετίζονται με τη μάρκα του οχήματος. Επιπρόσθετα υπάρχει πληροφορία σχετικά με το χιλιόμετρο που βρίσκεται ο επόμενος σταθμός, συνεπώς :

$$\textit{kilometre until CS} = \textit{kilometre of CS} - \textit{kilometre on search}$$

$$\textit{Extra power} = \textit{power per kilometre} * \textit{kilometre until CS}$$

$$\textit{Battery in station} = \textit{Battery on search} - \textit{Extra power}$$

Θεωρούμε ότι για τον χρόνο μερικής φόρτισης προστίθεται στο όχημα μόνο η ενέργεια που απαιτείται για να φτάσει στον προορισμό του διατηρώντας την εφεδρεία του, καθώς σε πολλές περιπτώσεις η ενέργεια της μπαταρίας είναι αρκετή για να φτάσει ο οδηγός στον προορισμό του αλλά η εφεδρεία που θέλει να έχει ο κάθε οδηγός στην μπαταρία εξαρτάται από το άγχος του. Η ενέργεια που χρειάζεται για να φτάσει ένας οδηγός στον προορισμό του υπολογίζεται:

$$\textit{power} = (\textit{destination length} - \textit{kilometre of CS}) * \textit{power per km}$$

Για τον χρόνο πλήρους φόρτισης προστίθεται στο όχημα τόση ενέργεια όση χρειάζεται για να φορτίσει πλήρως δηλαδή:

$$\textit{power} = \textit{Battery fully charged} - \textit{Battery in Station}$$

Ο κανόνας για τον υπολογισμό του χρόνου πλήρους και μερικής φόρτισης είναι:

$$\textit{time} = \frac{\textit{power}}{\textit{charge power} * PF}$$

, όπου charge power η ενέργεια του φορτιστή σε kW και PF ο power factor του. Σε περίπτωση που ο σταθμός φόρτισης τοποθετείται στο τελευταίο χιλιόμετρο η αρκετά κοντά στην διαδρομή, για την μερική φόρτιση, απαιτείται από πολύ λίγο έως και καθόλου ενέργεια. Συνεπώς ο χρόνος μερικής φόρτισης είναι πολύ κοντά στο 0 ή και 0. Για τον λόγο αυτό, στην περίπτωση που ο σταθμός τοποθετείται σε εύρος έως και 5 χιλιόμετρα πριν τον προορισμό γίνεται η θεώριση πως ο οδηγός θα φορτίσει 3000W και υπολογίζεται ο χρόνος μερικής φόρτισης.

Για το πόσοι φορτιστές τοποθετούνται σε κάθε σταθμό ακολουθείται η παρακάτω μέθοδος. Γνωρίζοντας το χιλιόμετρο που αναζητάει για σταθμό φόρτισης κάθε οδηγός, υπολογίζεται για το κάθε όχημα ο σταθμός που θα σταματήσει. Γίνεται η θεώριση ότι 70% των οδηγών φορτίζουν μερικός το όχημα τους ενώ το 30% πλήρως με τυχαίο τρόπο, και με βάση τον χρόνο φόρτισης εξάγεται η πληροφορία για το πόσα πεντάλεπτα φορτίζει το κάθε όχημα. Με βάση το πλήθος πενταλέπτων φόρτισης των οχημάτων που σταματούν στον υπο μελέτη σταθμό, γεμίζει ένας πίνακας [288,120] για τις 24 ώρες της ημέρας καθώς μια ώρα έχει $\frac{60}{5} = 12$ πεντάλεπτα και άρα μια μέρα έχει $12 * 24 = 288$ και 120 ένα πάνω όριο που αναπαριστά το πιθανό πλήθος των φορτιστών στο σταθμό. Με βάση την ώρα και τα πακέτα πενταλέπτων γεμίζει ο πίνακας αυτός από την 1 έως την 24 ώρα (1 έως 288) με 1 ανάλογα με το πλήθος των πενταλέπτων (slot) που καταλαμβάνει κάθε όχημα που σταματάει στον υπο μελέτη σταθμό. Στο τέλος υπολογίζεται για τον εκάστοτε σταθμό φόρτισης το πλήθος των φορτιστών που πρέπει να τοποθετηθούν βρίσκοντας την μεγαλύτερη στήλη που έχει 1. Συνεπώς, το πλήθος των φορτιστών κάθε σταθμού συνδέεται με την ώρα που στον σταθμό ήταν κατειλημμένοι οι περισσότεροι φορτιστές.

Ο PSO αναζητάει την βέλτιστη λύση δοκιμάζοντας διαφορετικούς συνδυασμούς τοποθέτησης των σταθμών φόρτισης κατά μήκος της διαδρομής και υπολογίζει το κόστος με βάση το πλήθος των φορτιστών. Τελικά για κάθε πρόβλημα διατηρεί το καλύτερο κόστος.

4.6.2 Αξιοποίηση αλγόριθμου

Αρχικά, τα χιλιόμετρα που τοποθετηθήκαν οι σταθμοί φόρτισης ήταν ανά 5 χιλιόμετρα. Ωστόσο, το αυθαίρετο αυτό μοντέλο δεν ήταν το βέλτιστο καθώς πρέπει να γίνουν πολλοί συνδυασμοί που αφορούν την τοποθέτηση των σταθμών σε κάθε χιλιόμετρο

και από αυτούς τελικά να επιλεχθεί αυτός που δίνει τη βέλτιστη λύση. Μια συστηματική λύση μπορεί να δοθεί από τον αλγόριθμο **particle swarm optimization**, που αναφέρθηκε στο 4.5, ο οποίος μπορεί και διατρέχει πολλούς συνδυασμούς σχετικά με το πόσοι θα είναι οι σταθμοί που θα τοποθετηθούν και σε πιο χιλιόμετρο και τελικά να επιλέξει τον συνδυασμό με το μικρότερο κόστος, που θεωρείται και η βέλτιστη λύση. Η συνάρτηση κόστους του PSO περιγράφεται στο 4.6.2.2.

4.6.2.1 Παράμετροι PSO

Όπως αναφέρθηκε παραπάνω ο PSO επαναλαμβάνει το πείραμα κάνοντας δοκιμές για διαφορετικές θέσεις σταθμών φόρτισης κατά μήκος της διαδρομής. Αρχικά ορίζεται το μέγεθος του vector x . Το vector αυτό περιέχει το χιλιόμετρο που τοποθετείται κάθε σταθμός. Στη συνέχεια ορίζεται ένα κάτω όριο, που στο παρόν πρόβλημα σχετίζεται με το χιλιόμετρο που ξεκινούν να αναζητούν τα οχήματα για σταθμό φόρτισης σε κάθε διαδρομή και ένα πάνω όριο που σχετίζεται με το τελευταίο χιλιόμετρο της κάθε διαδρομής. Το vector x , το μέγεθος του οποίου ορίζεται, παίρνει τιμές μεταξύ του πάνω και του κάτω ορίου, οι τιμές αυτές ταξινομούνται κατά αύξουσα σειρά για να τοποθετείται το χιλιόμετρο του κάθε σταθμού με τη σωστή σειρά, και διερευνάται το κόστος της λύσης για αυτό το x . Επιπλέον, παράμετροι είναι ο αριθμός των αναδρομών (iterations) και ο πληθυσμός (population) του σμήνου. Σε κάθε αναδρομή, για όλα τα μέλη του πληθυσμού, ανανεώνεται η ταχύτητα και η θέση τους με βάση τις εξισώσεις 4.5, 4.5 και διερευνείται το κόστος για την νέα θέση. Για τις λύσεις που δεν είναι επιθυμητές προστίθενται ποινές στο κόστος, οι ποινές αναλύονται στην ενότητα 4.6.2.2.

Για παράδειγμα, αν θέσουμε το vector x να έχει μέγεθος 4 τότε το πρόβλημα θα διερευνηθεί λύση για τοποθέτηση 4 σταθμών φόρτισης. Αν το κάτω όριο είναι 21 και το πάνω 65 τότε πιθανός συνδυασμός του x για κάποιο σωματίδιο (particle) του σμήνου, για το σενάριο Χανιά - Ρέθυμνο, θα μπορούσε να είναι $\vec{x} = [21 \ 49.3 \ 58.6 \ 65]$. Αυτό σημαίνει ότι ο πρώτος σταθμός θα τοποθετηθεί στο 21 χιλιόμετρο, ο δεύτερος στο 49.3 χιλιόμετρο κοκ. Διερευνάται το κόστος για αυτό το x , ανανεώνονται η θέση και η ταχύτητα και διερευνάται το κόστος για το νέο x . Η διαδικασία αυτή επαναλαμβάνεται για όλες τις αναδρομές και σε κάθε αναδρομή για κάθε μέλος του πληθυσμού του σμήνου. Τελικό αποτέλεσμα είναι η θέση του σωματιδίου που έδωσε το μικρότερο κόστος.

Για τη διαδρομή Χανιά - Ρέθυμνο ερευνάται η τοποθέτηση για το πρόβλημα 4,5,6,7 και 8 σταθμών, με κάτω όριο το 21,πάνω όριο το 65 και με πληθυσμό 1000 για 100 αναδρομές για κάθε πρόβλημα.Για τη διαδρομή Ρέθυμνο - Ηράκλειο διερευνάται η τοποθέτηση για το πρόβλημα 4,5,6,7 και 8 σταθμών, με κάτω όριο 32,πάνω όριο 77 και με πληθυσμό 1000 για 100 αναδρομές για κάθε πρόβλημα. Τέλος, για τη διαδρομή Χανιά - Ηράκλειο ερευνάται η τοποθέτηση για το πρόβλημα 8,10,12 και 14 σταθμών με κάτω όριο 48,πάνω όριο 142 και με πληθυσμό 1000 για 100 αναδρομές για κάθε πρόβλημα.[45]

4.6.2.2 Συνάρτηση κόστους PSO

Η συνάρτηση κόστους αφορά το κόστος αγοράς και εγκατάστασης των φορτιστών ηλεκτρικών αυτοκινήτων των σταθμών φόρτισης.Το κόστος αυτό εξαρτάται από τη χώρα εγκατάστασής, από τον πληθωρισμό καθώς και από την τοποθεσία εγκατάστασης.Μερικοί από τους παράγοντες κόστους αγοράς και εγκατάστασης φορτιστών είναι οι παρακάτω:

- Οι φορτιστές ανάλογα με τον τύπο τους διακρίνονται σε 3 τύπους.
 - Επιπέδου 1 (AC) που παρέχουν ενέργεια 1.4/1.9 kW.
 - Επιπέδου 2 (AC) που παρέχουν ενέργεια 3.4/6.6/19.2/22 kW.
 - Επιπέδου 3 (DCFC) που παρέχουν ενέργειας 40/90 kW.

Με το κόστος των φορτιστών επιπέδου 3 να είναι πολύ μεγαλύτερο από αυτό των άλλων δύο κατηγοριών (έως και 10 φορές ακριβότεροι από τους φορτιστές επιπέδου 2).

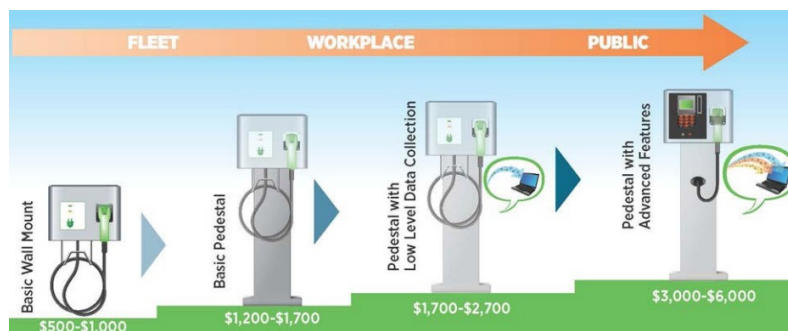
- Το κόστος είναι αναλογικό με το πλήθος των θυρών.Οι θύρες μπορεί να είναι:
 - Μονή θύρα (single port)
 - Διπλή θύρα (dual port)
 - Τρείς θύρες
 - Τέσσερις θύρες

Ωστόσο, χρειάζεται προσοχή καθώς σε κάθε περίπτωση μόνο ένα όχημα φορτίζει κάθε φορά.Δηλαδή αν ο φορτιστής παρέχει 2 θύρες αυτό δε σημαίνει ότι τα οχήματα φορτίζουν παράλληλα, αλλά ακολουθιακά.

- Επιπλέον παράγοντας που καθορίζει το κόστος είναι και το είδος του φορτιστή που προορίζεται για αγορά και εγκατάσταση. Τα είδη είναι:
 - Φορτιστές τοίχου(wall chargers)
 - φορτιστές βάνθρου(pedestal chargers)

Οι φορτιστές τοίχου είναι πιο οικονομικοί και στην αγορά αλλά και στην εγκατάσταση και προτιμούνται στις περιπτώσεις που είναι εφικτή η εγκατάστασή τους. Η εγκατάστασή τους είναι εφικτή όταν υπάρχουν κοντά εγκαταστάσεις όπως εμπορικά κέντρα για την τοποθέτησή τους σε τοίχο.
- Παράγοντας που καθορίζει το κόστος είναι επίσης και η δυνατότητα σύνδεσης σε δίκτυο. Οι φορτιστές που συνδέονται σε δίκτυο μοιράζονται πληροφορίες μεταξύ τους και μπορούν να παρέχουν υπηρεσίες όπως πληρωμές με POS, δυνατότητα κράτησης φορτιστή μέσω εφαρμογής, αναγνώριση χρήστη σε περίπτωση που είναι εγγεγραμμένος στον πάροχο και οθόνη με κάποιο interface για την αλληλεπίδραση με τους χρήστες. Οι φορτιστές ηλεκτρικών οχημάτων από αυτή την άποψη μπορεί να έχουν:
 - σύνδεση σε δίκτυο
 - χωρίς σύνδεση σε δίκτυο αλλά με επιπλέον ενσωματωμένες εφαρμογές
 - χωρίς σύνδεση σε δίκτυο χωρίς επιπλέον εφαρμογές.
- Το κόστος εγκατάστασης εξαρτάται από:
 - Το εργατικό κόστος της εκάστοτε χώρας για τις απαραίτητες εργασίες
 - Την ανάγκη τροποποίησης ή αναβάθμισης του ηλεκτρικού πίνακα
 - Τον τύπο και το είδος του φορτιστή που τίθεται για εγκατάσταση

Οι επιπέδου 3 φορτιστές έχουν πολύ μεγαλύτερο κόστος εγκατάστασης καθώς απαιτείται ξεχωριστό δίκτυο διανομής ενέργειας για την ηλεκτροδότηση τους και ακριβότερα υλικά. Όσον αφορά το είδος οι φορτιστές βάνθρου είναι ακριβότεροι από τους φορτιστές τοίχου.[5]



Σχήμα 4.15: Μεταβολή κόστους σε σχέση με το είδος, την τοποθεσία και τις εφαρμογές, φορτιστών επιπέδου 2 - Πηγή [5]

Οι μη αποδεκτές λύσεις που θα δώσει ο PSO θα πρέπει να απορριφθούν. Θεωρούμε ως μη αποδεκτή λύση όταν τοποθετείται σταθμός στον οποίο δεν σταματάει κάποιο όχημα και όταν οι σταθμοί δεν είναι ταξινομημένοι κατά αύξουσα σειρά. Όπως αναφέρθηκε παραπάνω γίνεται ταξινόμηση του vector x , που περιέχει το χιλιόμετρο που τοποθετείται ο κάθε σταθμός, συνεπώς οι σταθμοί τοποθετούνται χιλιομετρικά με τη σωστή σειρά. Στους περιορισμούς προστίθενται και οι περιπτώσεις που τοποθετείται σταθμός στο ίδιο χιλιόμετρο, ο επόμενος σταθμός που τοποθετείται δεν απέχει 5 χιλιόμετρα από τον προηγούμενο, όταν τοποθετούνται οι σταθμοί με τέτοιο τρόπο που τα οχήματα που αναζητούν σταθμό να μην εξυπηρετούνται και όταν σε έναν σταθμό οι φορτιστές είναι περισσότεροι από 20. Συνεπώς, για τις παραπάνω περιπτώσεις πρέπει να προστίθεται στο κόστος ποινές, με τέτοιο τρόπο ώστε να είναι ευδιάκριτο αν η λύση που δίνεται δεν ανταποκρίνεται στους περιορισμούς αλλά και σε τι βαθμό.

Για την περίπτωση που οδηγός αναζήτησε για σταθμό και δεν εξυπηρετήθηκε, καθώς μετά το χιλιόμετρο αναζήτησης δεν υπήρχε διαθέσιμος σταθμός, στο κόστος προστίθεται ένα ποσό που υπολογίζεται από την εξίσωση:

$$b = w1 * n^2$$

, όπου n ο αριθμός των οχημάτων που δεν εξυπηρετήθηκαν και $w1$ ένα ορισμένο βάρος. Επιπλέον ποινή υπάγεται στο κόστος στην περίπτωση που οι σταθμοί δεν απέχουν μεταξύ τους 5 χιλιόμετρα. Το ποσό που προστίθεται στο κόστος υπολογίζεται από την εξίσωση:

$$c = w2 * m^2$$

, όπου m ο αριθμός των σταθμών που δεν απέχουν μεταξύ τους 5 χιλιόμετρα και $w2$ ένα ορισμένο βάρος.

Επιπρόσθετα, όταν δύο σταθμοί τοποθετούνται στο ίδιο χιλιόμετρο προστίθεται στο κόστος ποινή που υπολογίζεται από την εξίσωση:

$$d = w3 * z^2$$

,όπου z το πλήθος των σταθμών που τοποθετούνται στο ίδιο χιλιόμετρο και $w3$ ένα ορισμένο βάρος. Για να δίνονται ως καλύτερες λύσεις από τον PSO οι περιπτώσεις που οι φορτιστές τοποθετούνται ομοιόμορφα στους σταθμούς φόρτισης,όταν αυτό είναι εφικτό, επιβάλλεται ποινή διπλάσιου κόστους αναλογη με το πλήθος των επιπλέον φορτιστών:

$$ext = 6300 * (chargers - 20)$$

, στην περίπτωση που τοποθετηθεί κάποιος σταθμός και δεν σταματήσει κάποιο όχημα σε αυτόν η λύση απορρίπτεται με κόστος:

$$E = 6e6$$

, αν οι σταθμοί δεν τοποθετηθούν κατα αύξουσα σειρά η λύση επίσης απορρίπτεται με κόστος:

$$f = 6e6$$

Με βάση τον κυκλοφοριακό φόρτο κατά μήκος του BOAK, οι φορτιστές επιπέδου 2 (AC) των $22kW$ επαρκούν για την εξυπηρέτηση του φορτίου με προνομιακό κόστος έναντι των φορτιστών επιπέδου 3, όπου το κόστος αγοράς και εγκατάστασης είναι πολλαπλάσιο.Συνεπώς, για τη διπλωματική αυτή εργασία θεωρείται ότι οι φορτιστές που πρόκειται να αγορασθούν και να εγκατασταθούν στους σταθμούς φόρτισης που τοποθετούνται κατά μήκος του BOAK είναι επιπέδου 2 (AC) των $22kW$, τύπου βάρθρου(pedestal) καθώς δεν υπάρχει δυνατότητα εγκατάστασης φορτιστών τοίχου, μονής θύρας, με έξτρα λειτουργίες με κόστος αγοράς φορτιστή 3100 ευρώ και κόστος εγκατάστασης 3200 ευρώ. Το κόστος επιλέχθηκε με βάση το μέσο κόστος από την έρευνα τη [5].Το κόστος αγοράς και εγκατάστασης υπολογίζεται από την εξίσωση:

$$a = \#parkinglots * (cost\ of\ charger + installation\ cost)$$

Τελικά το κόστος υπολογίζεται:

$$cost = a + b + c + d + ext + E + f$$

Ωστόσο το πραγματικό κόστος εξάγεται από το κόστος αγοράς και εγκατάστασης των φορτιστών. Οι ποινές ορίζονται για να βρίσκει ο PSO τις καλύτερες λύσεις με βάση τις προδιαγραφές του προβλήματος.

Κεφάλαιο 5

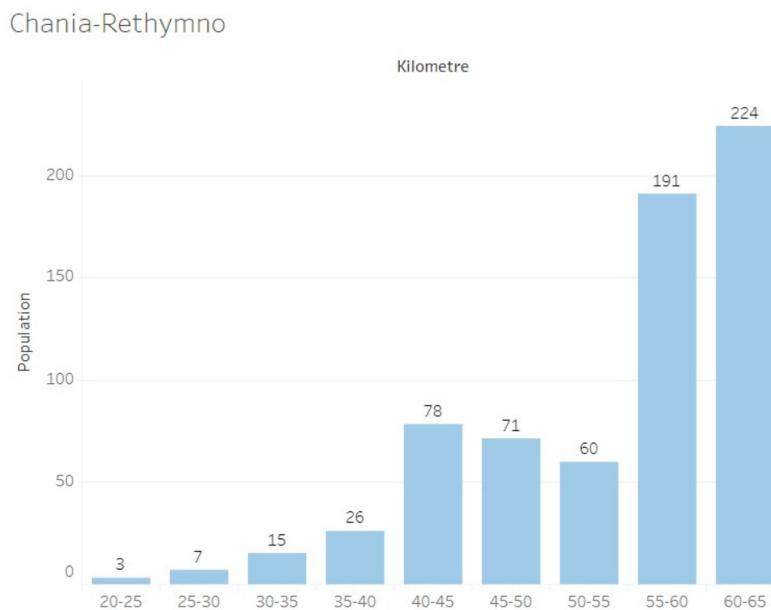
Παρουσίαση και τεκμηρίωση αποτελεσμάτων

5.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα της ενεργειακής αξιολόγησης των τεσσάρων ηλεκτρικών αυτοκινήτων και της βέλτιστης τοποθέτησης σταθμών φόρτισης καθώς και το πλήθος φορτιστών σε αυτούς για κάθε σενάριο.

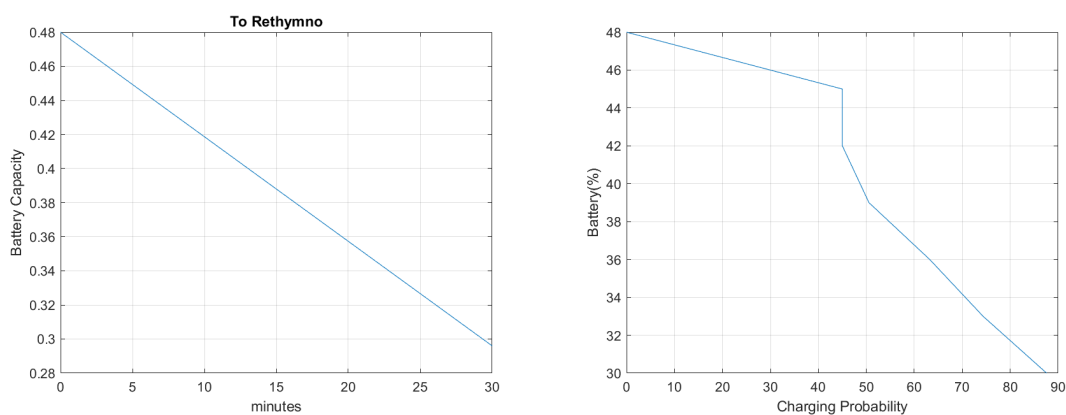
5.2 Χανιά-Ρέθυμνο

Αρχικά, από την διαδικασία της ενεργειακής αξιολόγησης συμπεραίνεται ποιο είναι το χιλιόμετρο που κάθε οδηγός αναζητάει για σταθμό φόρτισης. Συγκεκριμένα με βάση την ταχύτητα του κάθε οχήματος, η οποία παραμένει σταθερή για όλη την διαδρομή, υπολογίζεται η κατανάλωση ενέργειας από τις εξισώσεις κίνησης. Σε κάθε όχημα ανατίθεται ποσοστό αρχικής φόρτισης και ποσοστό άγχους του οδηγού, το οποίο θεωρείται σταθερό, και τελικά από το σύστημα ασαφούς λογικής εξάγεται το χιλιόμετρο που ο κάθε οδηγός αναζητάει για σταθμό φόρτισης ελέγχοντας κάθε 5 λεπτά το ποσοστό φόρτισης της μπαταρίας και το ποσοστό άγχους του οδηγού.



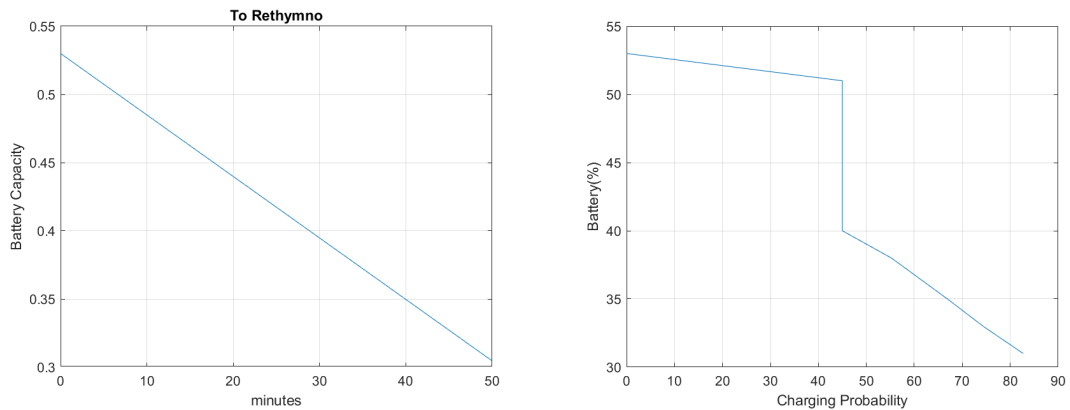
Σχήμα 5.1: Ιστόγραμμα διαδρομής Χανιά - Ρέθυμνο

Παρακάτω παρουσιάζονται ενδεικτικά παραδείγματα για οδηγούς αυτοκινήτων που αναζητούν σταθμό φόρτισης κατά την διάρκεια της διαδρομής Χανιά-Ρέθυμνο. Απεικονίζεται παράδειγμα για κάθε ένα από τα τέσσερα αμιγώς ηλεκτρικά αυτοκίνητα και συγκεκριμένα παρουσιάζεται διάγραμμα με την πτώση της μπαταρίας κάθε πέντε λεπτά καθώς και το αποτέλεσμα του συστήματος ασαφούς λογικής.



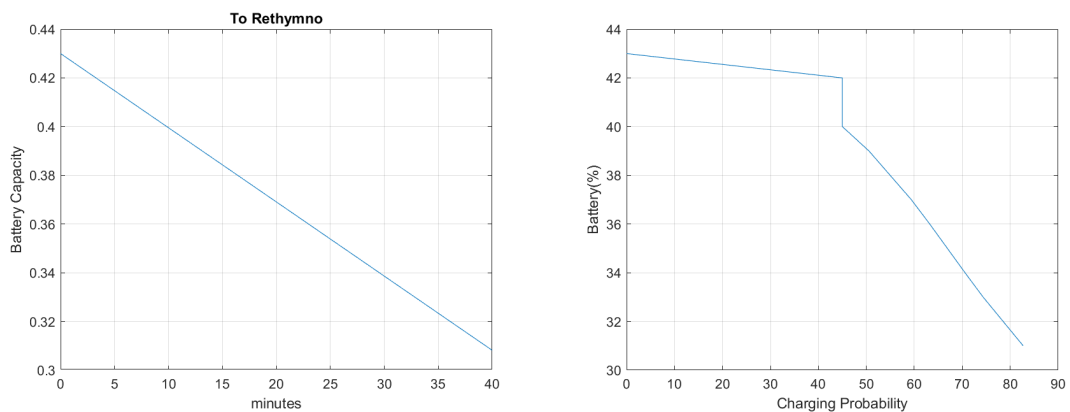
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.2: Audi Q4 e-tron 35: Ταχύτητα $96 \frac{km}{h}$, Άγχος 49% , Μπαταρία 48% , Χιλιόμετρο 48 °



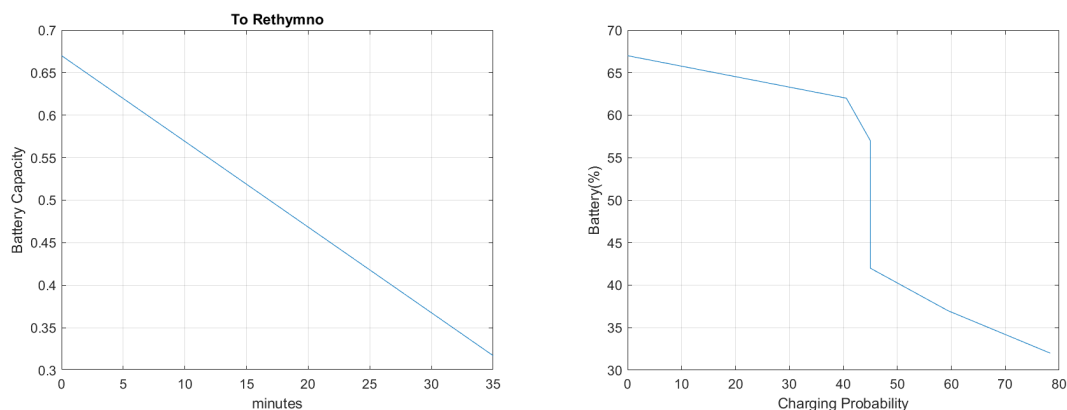
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασafούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.3: Fiat 500e: Ταχύτητα $76 \frac{km}{h}$, Άγχος 52% ,Μπαταρία 53% , Χιλιόμετρο 63°



(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασafούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.4: Nissan Leaf:Ταχύτητα $56 \frac{km}{h}$, Άγχος 48% , Μπαταρία 43% , Χιλιόμετρο 37°










(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.5: Mazda MX-30: Ταχύτητα $96 \frac{km}{h}$, Άγχος 42%, Μπαταρία 67%, Χιλιόμετρο 56°

Έπειτα, με βάση το χιλιόμετρο αναζήτησης σταθμού φόρτισης κάθε οδηγού, υπολογίζεται το ποιο όχημα και το πλήθος των οχημάτων που σταματάει σε κάθε σταθμό. Επιπλέον υπολογίζεται ο χρόνος μερικής και πλήρους φόρτισης και εξετάζοντας για κάθε σταθμό τα οχήματα που σταματούν σε αυτόν, υπολογίζεται το πλήθος των φορτιστών με βάση τον χρόνο φόρτισης του κάθε οχήματος. Ο αλγόριθμος PSO επαναλαμβάνει τον υπολογισμό του προβλήματος για διαφορετικούς συνδυασμούς τοποθέτησης των σταθμών φόρτισης και υπολογίζει το κόστος, όπου τελικά επιλέγει το καλύτερο. Είναι χρήσιμο να σημειωθεί ότι ο πίνακας με το χιλιόμετρο που αναζητάει ο οδηγός για σταθμό φόρτισης είναι ο ίδιος για κάθε επίλυση προβλήματος του PSO. Παρακάτω παρουσιάζονται τα αποτελέσματα του PSO για την επίλυση προβλήματος τοποθέτησης 4, 5, 6, 7 και 8 σταθμών.









Το πλήθος φορτιστών, το χιλιόμετρο τοποθέτησης και το κόστος για τις βέλτιστες λύσεις κάθε προβλήματος είναι:

**Problem of 4 Charging stations
Chania - Rethymno**

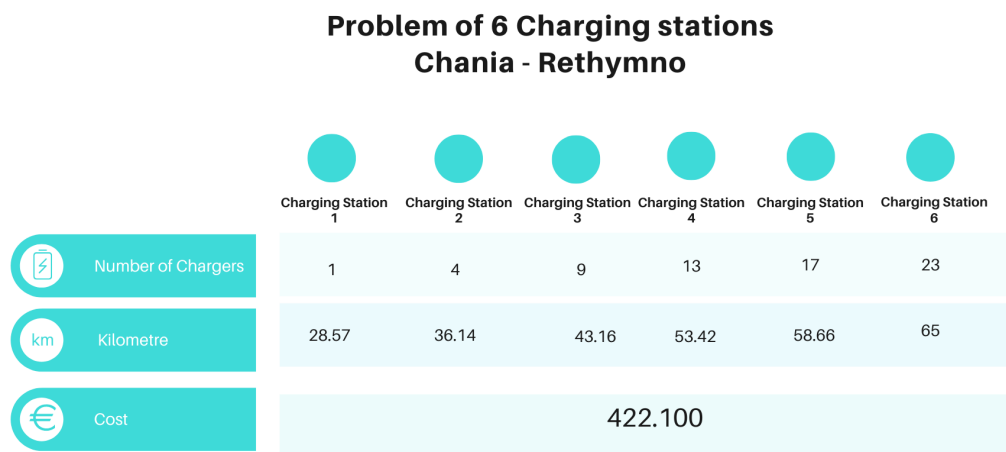
	 Charging Station 1	 Charging Station 2	 Charging Station 3	 Charging Station 4
 Number of Chargers	11	15	17	23
 Kilometre	40.86	51.83	58.15	65
 Cost	415.800			

Σχήμα 5.6: Πρόβλημα 4 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)

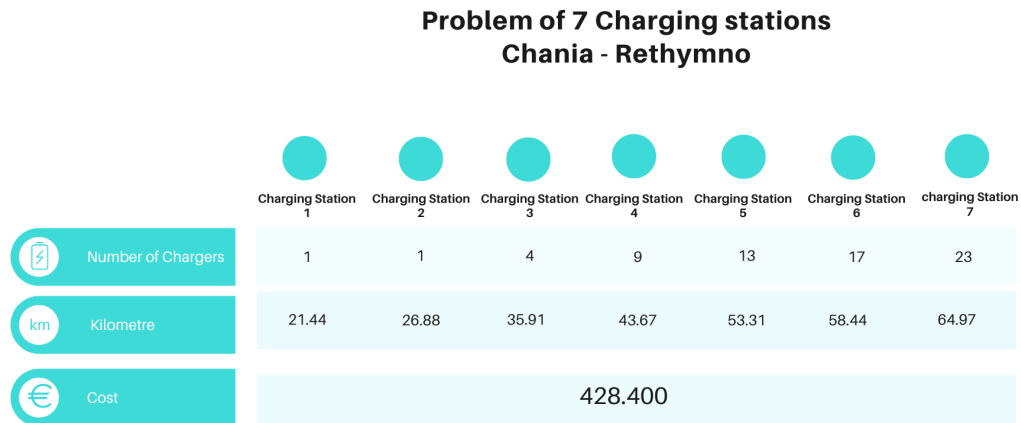
**Problem of 5 Charging stations
Chania - Rethymno**

	 Charging Station 1	 Charging Station 2	 Charging Station 3	 Charging Station 4	 Charging Station 5
 Number of Chargers	4	10	13	17	23
 Kilometre	35.77	45.89	52.67	58.36	64.57
 Cost	422.100				

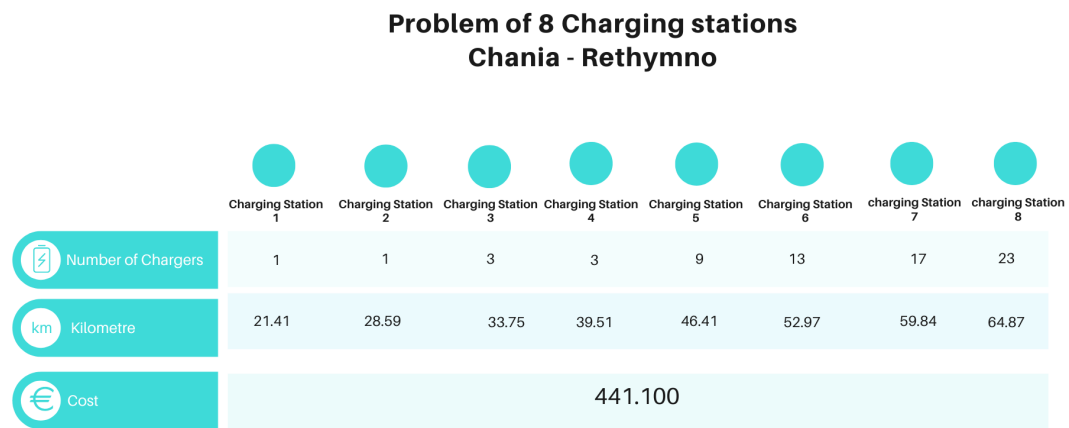
Σχήμα 5.7: Πρόβλημα 5 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)



Σχήμα 5.8: Πρόβλημα 6 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)

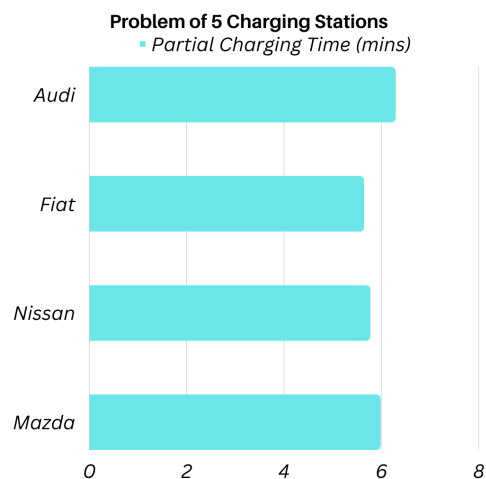


Σχήμα 5.9: Πρόβλημα 7 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)



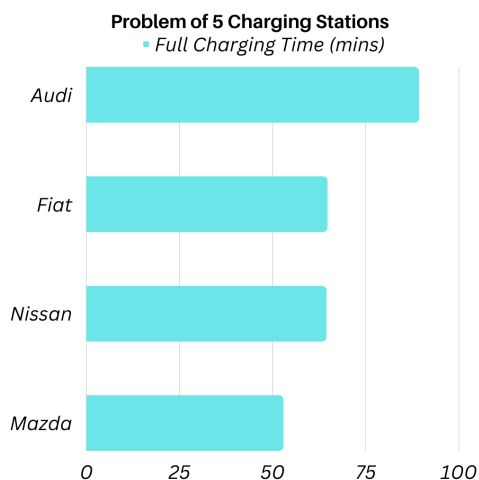
Σχήμα 5.10: Πρόβλημα 8 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ρέθυμνο)

Ενδεικτικές τιμή του μέσου χρόνου μερικής φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 5 σταθμών:



Σχήμα 5.11: Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Χανιά - Ρέθυμνο

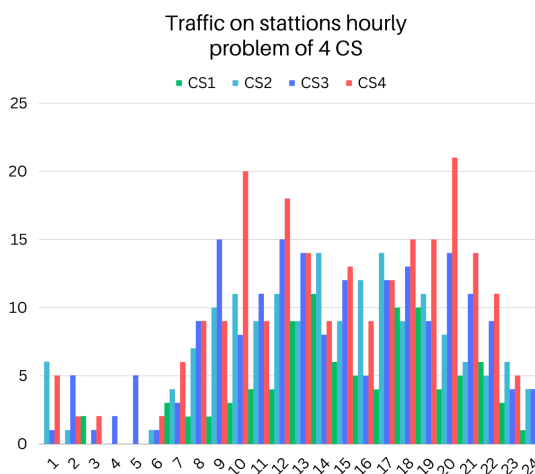
Ενδεικτικές τιμές του μέσου χρόνου πλήρους φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 5 σταθμών:



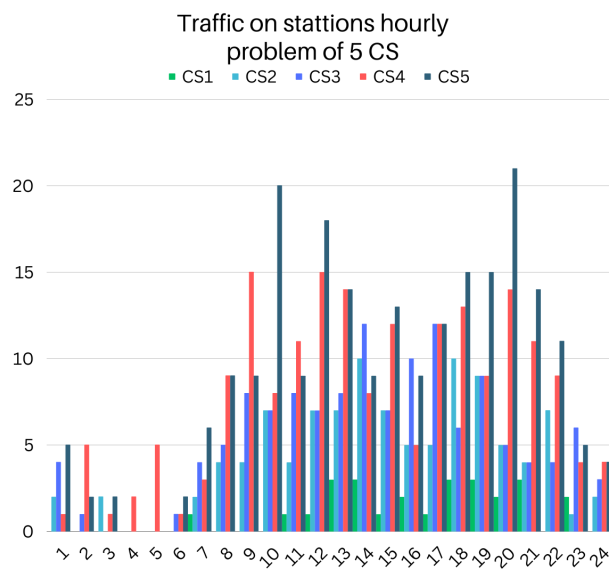
Σχήμα 5.12: Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Χανιά - Ρέθυμνο

Οι χρόνοι φόρτισης και για τα υπόλοιπα προβλήματα τοποθέτησης σταθμών της διαδρομής Χανιά - Ρέθυμνο είναι παρόμοιοι.

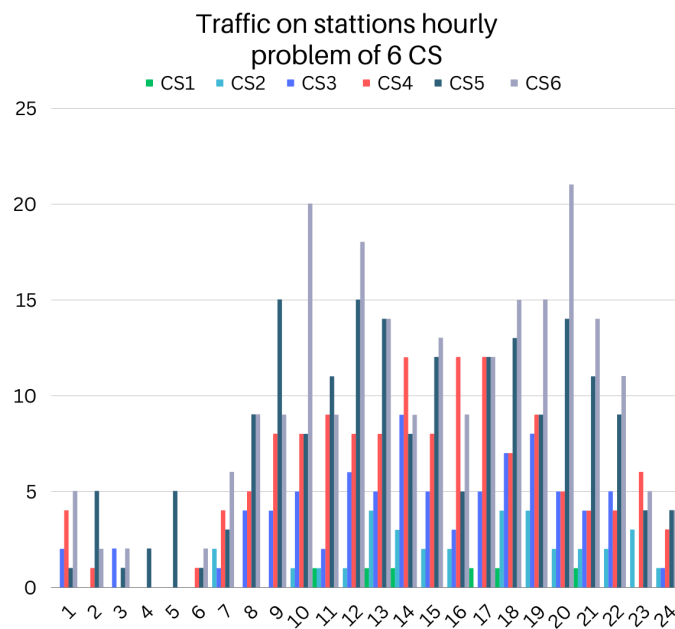
Το πλήθος των οχημάτων που σταματούν σε κάθε σταθμό για κάθε ώρα της ημέρας είναι:



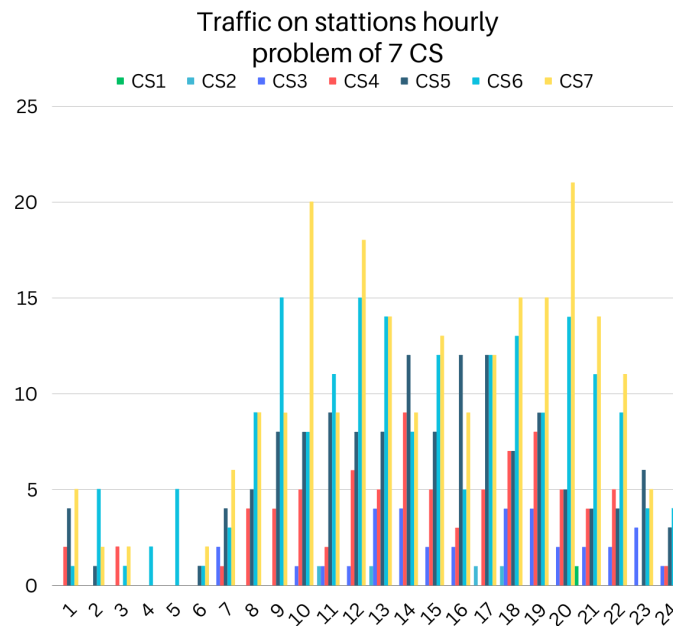
Σχήμα 5.13: Πρόβλημα 4 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο



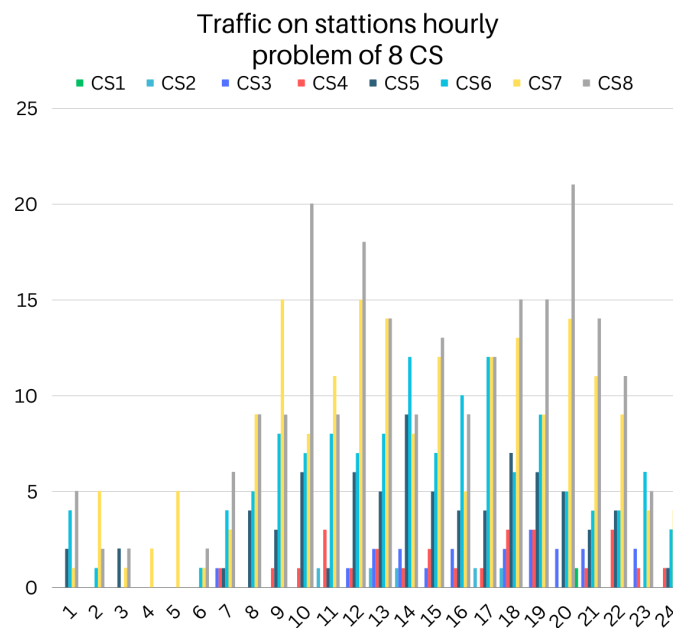
Σχήμα 5.14: Πρόβλημα 5 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο



Σχήμα 5.15: Πρόβλημα 6 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο



Σχήμα 5.16: Πρόβλημα 7 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο



Σχήμα 5.17: Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ρέθυμνο

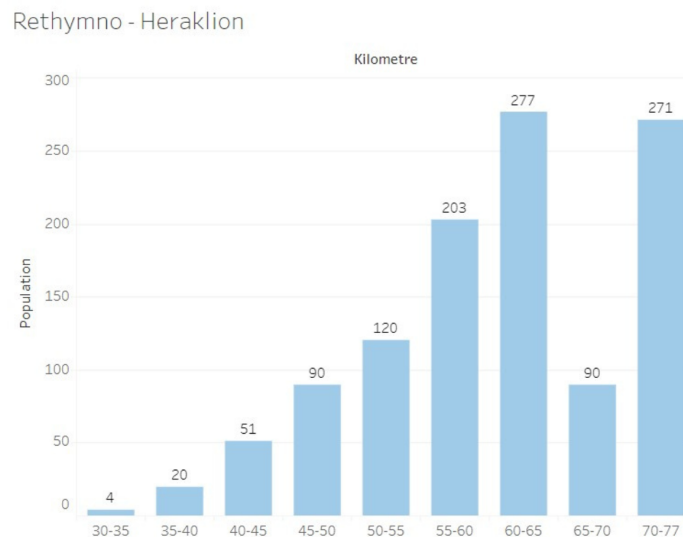
Από τα πειράματα φαίνεται ότι το μεγαλύτερο πλήθος των οδηγών αναζητούν για σταθμούς φόρτισης μετά το 55 χιλιόμετρο. Συνεπώς η τοποθέτηση των περισσότε-

ρων φορτιστών περιμένουμε να γίνει στους σταθμούς που τοποθετούνται μετά το 55 χιλιόμετρο, κάτι το οποίο πράγματι συμβαίνει. Ωστόσο, χρειάζεται προσοχή αφού το πλήθος των φορτιστών σε κάθε σταθμό συνδέεται με την ώρα που σταμάτησε κάθε όχημα σε αυτόν αλλά και με τον χρόνο φόρτισης του κάθε οχήματος. Συνεπώς, κάποιο όχημα που παραμένει στον σταθμό για χρόνο περισσότερο από 1 ώρα συμπεριλαμβάνεται και στην επόμενη, για να γίνει σωστή εκτίμηση του πλήθους των φορτιστών. Για αυτόν τον λόγο οι φορτιστές σε κάποιους σταθμούς φαίνονται να είναι λίγο περισσότεροι από το πλήθος των οχημάτων που σταματούν σε αυτόν την ώρα που είχε την περισσότερη κίνηση. Επιπλέον φαίνεται ότι οι ώρες που παρατηρείται μεγαλύτερη κίνηση στους σταθμούς είναι από τις 9 το πρωί έως τις 9 το βράδυ.

5.3 Ρέθυμνο-Ηράκλειο

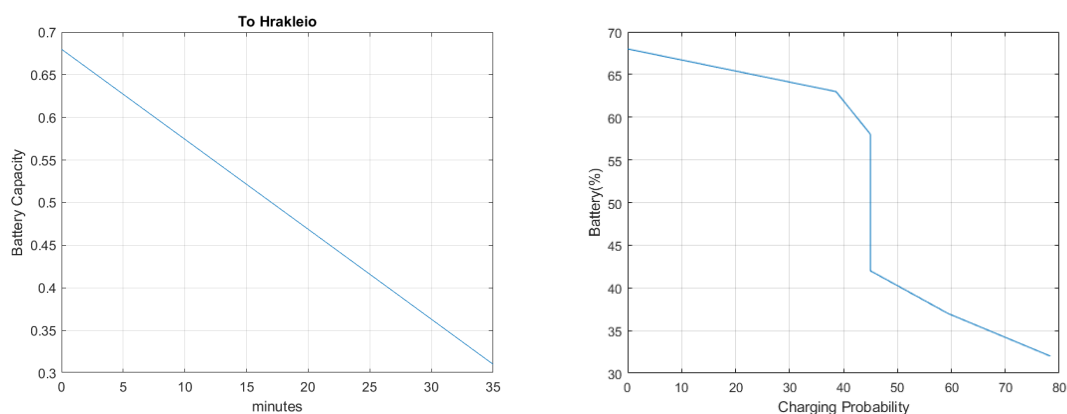
Αρχικά, από την διαδικασία της ενεργειακής αξιολόγησης συμπεραίνεται ποιο είναι το χιλιόμετρο που κάθε οδηγός αναζητάει για σταθμό φόρτισης. Συγκεκριμένα με βάση την ταχύτητα του κάθε οχήματος, η οποία παραμένει σταθερή για όλη την διαδρομή, υπολογίζεται η κατανάλωση ενέργειας από τις εξισώσεις κίνησης. Σε κάθε όχημα ανατίθεται ποσοστό αρχικής φόρτισης και ποσοστό άγχους του οδηγού, το οποίο θεωρείται σταθερό, και τελικά από το σύστημα ασαφούς λογικής εξάγεται το χιλιόμετρο που ο κάθε οδηγός αναζητάει για σταθμό φόρτισης ελέγχοντας κάθε 5 λεπτά το ποσοστό φόρτισης της μπαταρίας και το ποσοστό άγχους του οδηγού.

Στο ιστόγραμμα [5.18](#) , φαίνεται το χιλιόμετρο και το πλήθος των οδηγών που αναζητούν σταθμό φόρτισης σε αυτό.



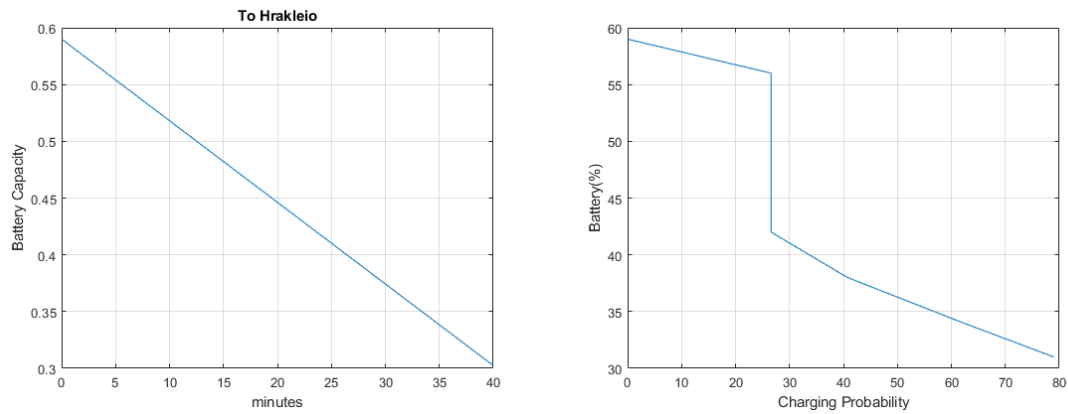
Σχήμα 5.18: Ιστόγραμμα διαδρομής Ρέθυμνο - Ηράκλειο

Παρακάτω παρουσιάζονται ενδεικτικά παραδείγματα για οδηγούς αυτοκινήτων που αναζητούν σταθμό φόρτισης κατά την διάρκεια της διαδρομής Χανιά-Ρέθυμνο. Απεικονίζεται παράδειγμα για κάθε ένα από τα τέσσερα αμιγώς ηλεκτρικά αυτοκίνητα και συγκεκριμένα παρουσιάζεται διάγραμμα με την πτώση της μπαταρίας κάθε πέντε λεπτά καθώς και το αποτέλεσμα του συστήματος ασαφούς λογικής.



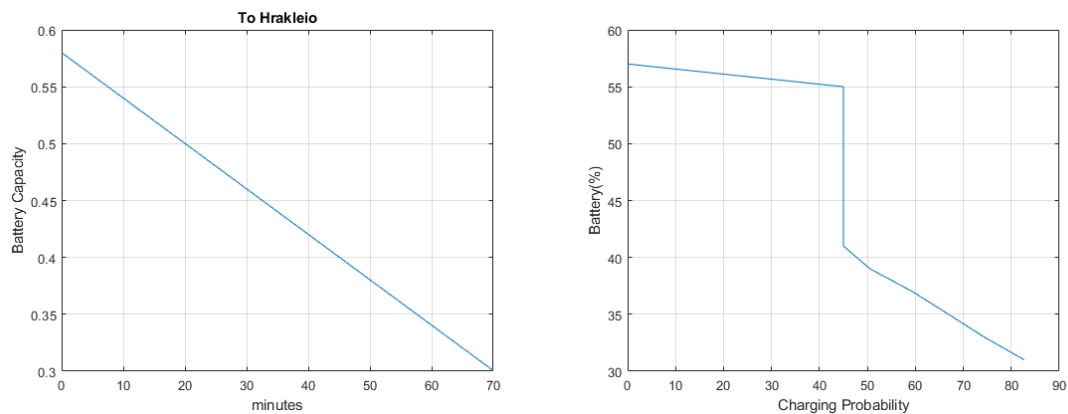
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.19: Audi Q4 e-tron 35: Ταχύτητα $125 \frac{km}{h}$, Άγχος 39%, Μπαταρία 68%, Χιλιόμετρο 73 °



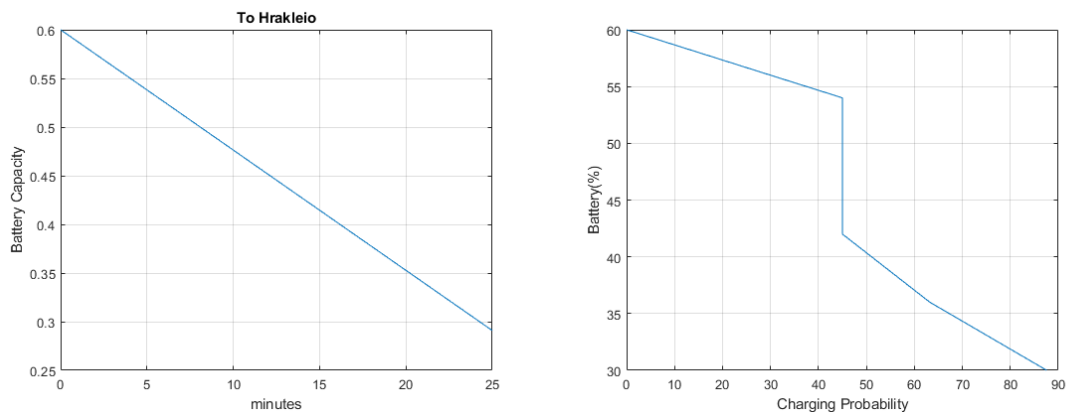
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.20: Fiat 500e: Ταχύτητα $96 \frac{km}{h}$, Άγχος 21%, Μπαταρία 59%, Χιλιόμετρο 64°



(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.21: Nissan Leaf: Ταχύτητα $66 \frac{km}{h}$, Άγχος 59%, Μπαταρία 58%, Χιλιόμετρο 77°










(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.22: Mazda MX-30: Ταχύτητα $106 \frac{km}{h}$, Άγχος 48%, Μπαταρία 60%, Χιλιόμετρο 53°

Έπειτα, με βάση το χιλιόμετρο αναζήτησης σταθμού φόρτισης κάθε οδηγού, υπολογίζεται το ποιο όχημα και το πλήθος των οχημάτων που σταματάει σε κάθε σταθμό. Επιπλέον υπολογίζεται ο χρόνος μερικής και πλήρους φόρτισης και εξετάζοντας για κάθε σταθμό τα οχήματα που σταματούν σε αυτόν, υπολογίζεται το πλήθος των φορτιστών με βάση τον χρόνο φόρτισης του κάθε οχήματος. Ο αλγόριθμος PSO επαναλαμβάνει τον υπολογισμό του προβλήματος για διαφορετικούς συνδιασμούς τοποθέτησης των σταθμών φόρτισης και υπολογίζει το κόστος, όπου τελικά επιλέγει το καλύτερο. Είναι χρήσιμο να σημειωθεί ότι ο πίνακας με το χιλιόμετρο που αναζητάει ο οδηγός για σταθμό φόρτισης είναι ο ίδιος για κάθε επίλυση προβλήματος του PSO. Παρακάτω παρουσιάζονται τα αποτελέσματα του PSO για την επίλυση προβλήματος τοποθέτησης 4, 5, 6, 7 και 8 σταθμών.









Το πλήθος φορτιστών, το χιλιόμετρο τοποθέτησης και το κόστος για τις βέλτιστες λύσεις κάθε προβλήματος είναι:

Problem of 4 Charging stations Rethymno-Heraklion

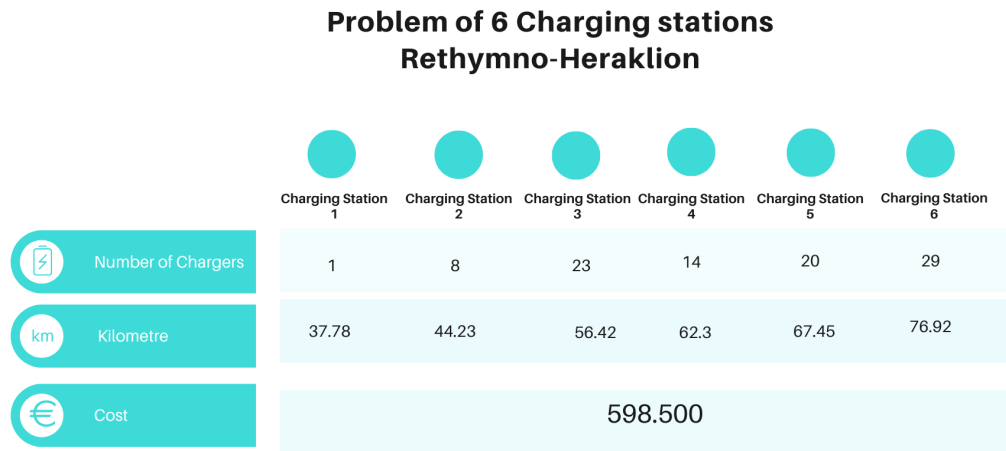
	 Charging Station 1	 Charging Station 2	 Charging Station 3	 Charging Station 4
 Number of Chargers	23	22	20	29
 Kilometre	54.19	62	67.08	76.9
 Cost	592.200			

Σχήμα 5.23: Πρόβλημα 4 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)

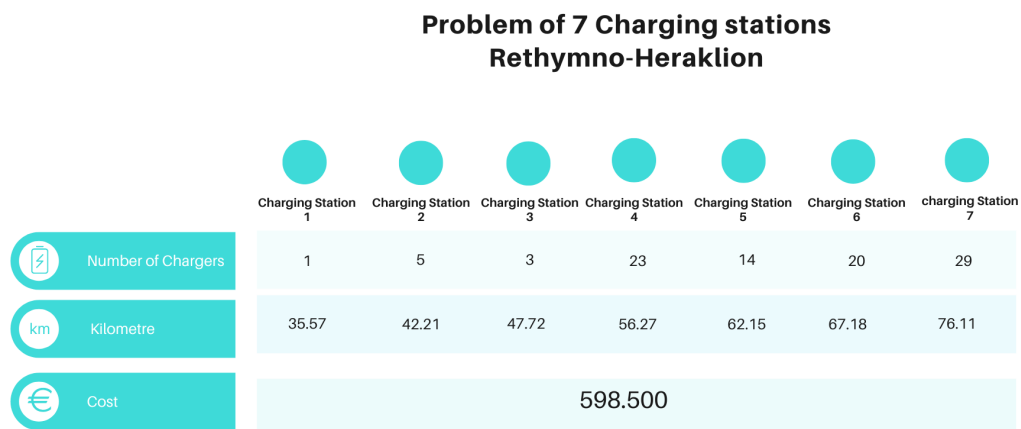
Problem of 5 Charging stations Rethymno-Heraklion

	 Charging Station 1	 Charging Station 2	 Charging Station 3	 Charging Station 4	 Charging Station 5
 Number of Chargers	3	21	22	20	29
 Kilometre	40.05	54.91	62.17	67.52	76.12
 Cost	598.500				

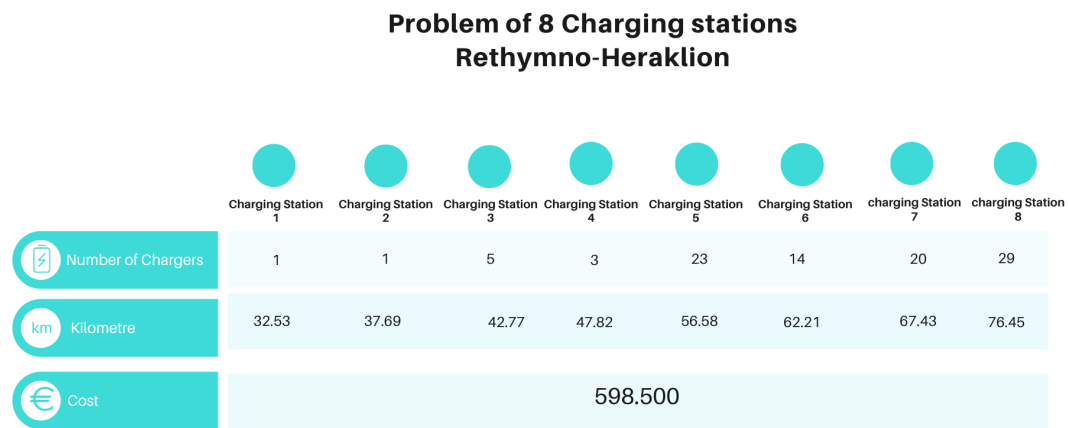
Σχήμα 5.24: Πρόβλημα 5 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)



Σχήμα 5.25: Πρόβλημα 6 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)

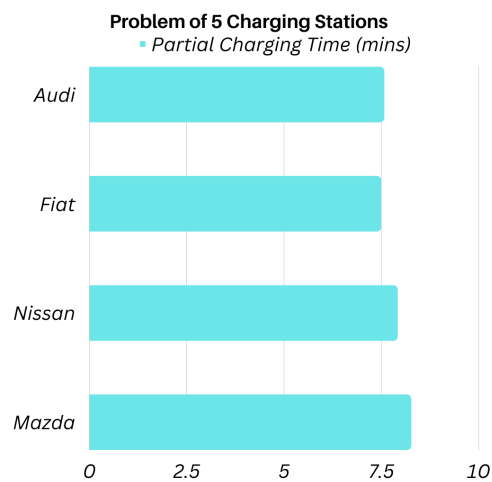


Σχήμα 5.26: Πρόβλημα 7 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)



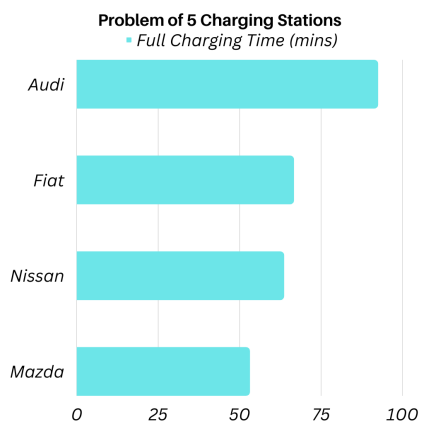
Σχήμα 5.27: Πρόβλημα 8 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Ρέθυμνο-Ηράκλειο)

Ενδεικτικές τιμές του μέσου χρόνου μερικής φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 5 σταθμών:



Σχήμα 5.28: Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Ρέθυμνο - Ηράκλειο

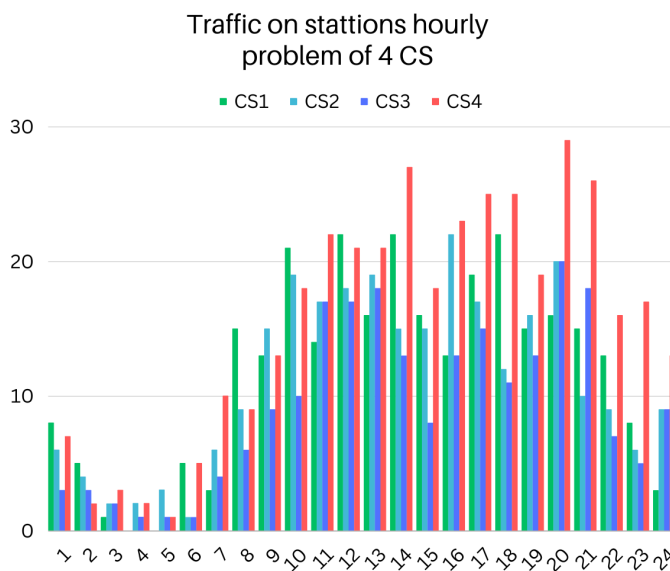
Ενδεικτικές τιμές του μέσου χρόνου πλήρους φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 5 σταθμών:



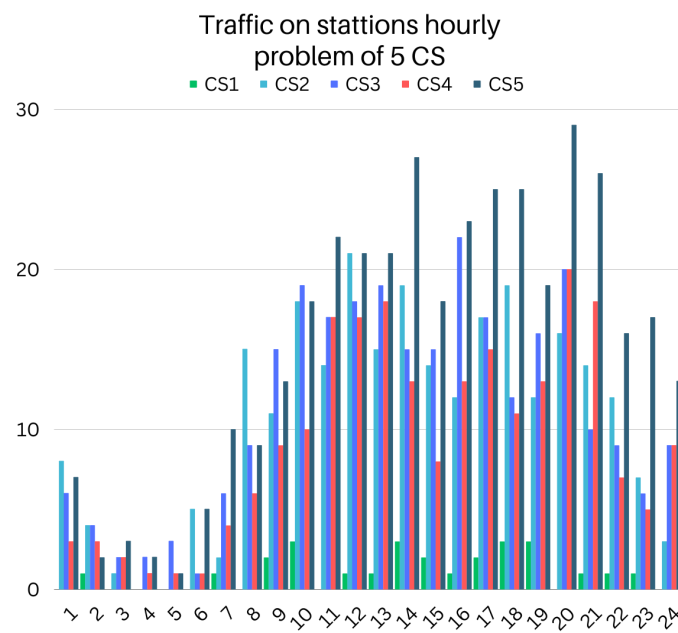
Σχήμα 5.29: Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 5 σταθμών Ρέθυμνο - Ηράκλειο

Οι χρόνοι φόρτισης και για τα υπόλοιπα προβλήματα τοποθέτησης σταθμών της διαδρομής Ρέθυμνο - Ηράκλειο είναι παρόμοιοι.

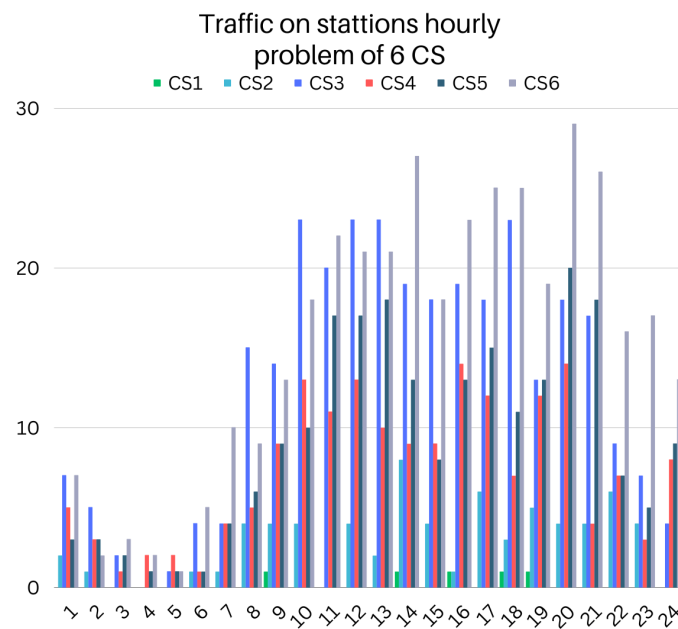
Το πλήθος των οχημάτων που σταματούν σε κάθε σταθμό για κάθε ώρα της ημέρας είναι:



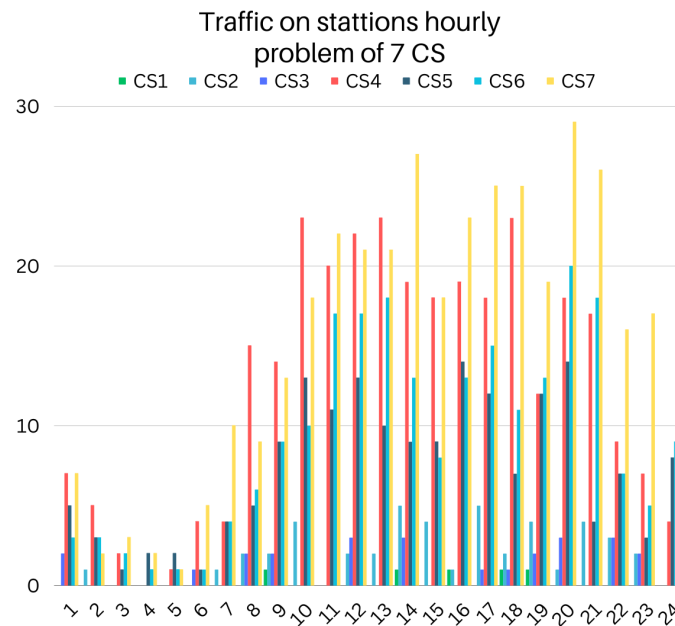
Σχήμα 5.30: Πρόβλημα 4 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο



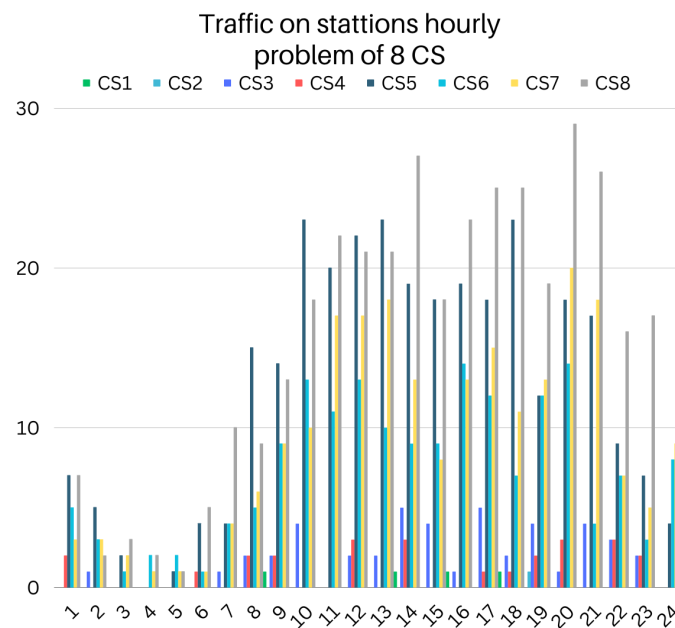
Σχήμα 5.31: Πρόβλημα 5 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο



Σχήμα 5.32: Πρόβλημα 6 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο



Σχήμα 5.33: Πρόβλημα 7 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο



Σχήμα 5.34: Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Ρέθυμνο - Ηράκλειο

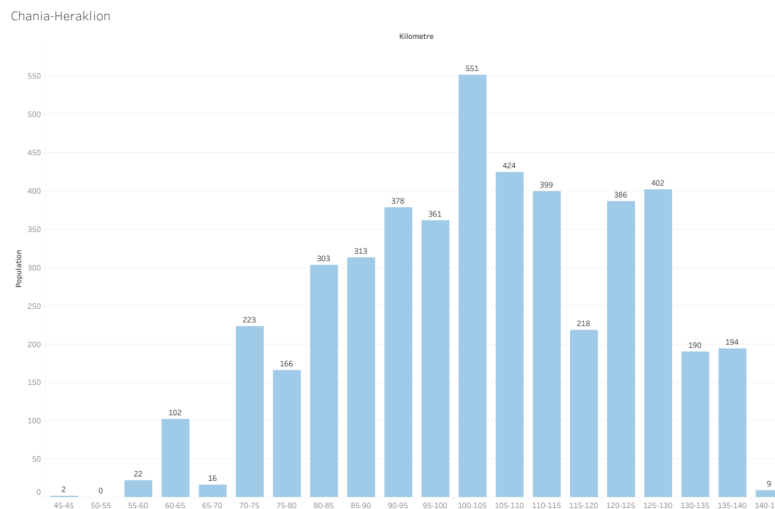
Από τα πειράματα φαίνεται ότι το μεγαλύτερο πλήθος των οδηγών αναζητούν για σταθμούς φόρτισης στο 55-65 και στο 70-77 χιλιόμετρο. Αυτό υποδηλώνει ότι το με-

γαλύτερο πλήθος των φορτιστών θα τοποθετηθεί στους σταθμούς που τοποθετούνται μεταξύ αυτού του εύρους χιλιομέτρων, κάτι το οποίο πράγματι συμβαίνει. Ωστόσο, χρειάζεται προσοχή αφού το πλήθος των φορτιστών σε κάθε σταθμό συνδέεται με την ώρα που σταμάτησε κάθε όχημα σε αυτόν αλλά και με τον χρόνο φόρτισης του κάθε οχήματος. Συνεπώς, κάποιο όχημα που παραμένει στον σταθμό για χρόνο περισσότερο από 1 ώρα συμπεριλαμβάνεται και στην επόμενη, για να γίνει σωστή εκτίμηση για το πλήθος των φορτιστών. Για αυτόν τον λόγο οι φορτιστές σε κάποιους σταθμούς φαίνονται να είναι λίγο περισσότεροι από το πλήθος των οχημάτων που σταματούν σε αυτόν την ώρα που σημειώνεται η περισσότερη κίνηση. Τέλος, μεγαλύτερη κίνηση στους σταθμούς παρατηρείται από τις 10 το πρωί έως και τις 9 το βράδυ.

5.4 Χανιά-Ηράκλειο

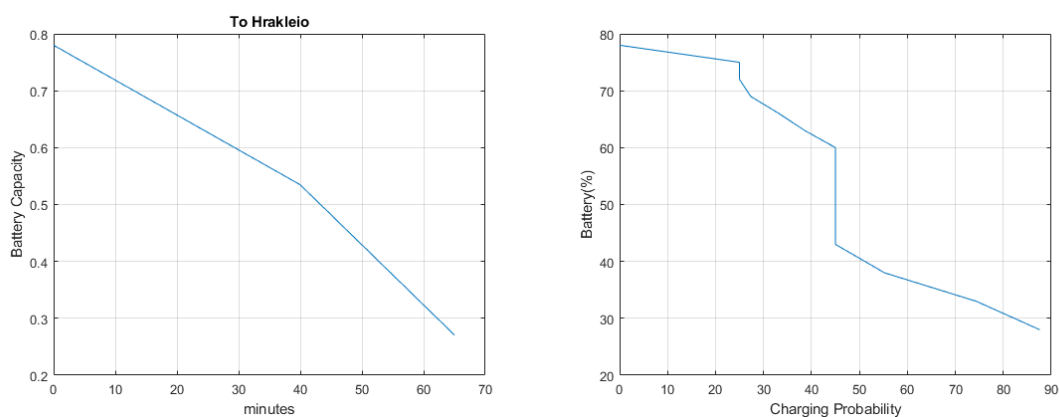
Όμοια με τα προηγούμενα 2 σενάρια, από την διαδικασία της ενεργειακής αξιολόγησης συμπεραίνεται ποιο είναι το χιλιόμετρο που κάθε οδηγός αναζητάει για σταθμό φόρτισης. Συγκεκριμένα με βάση την ταχύτητα του κάθε οχήματος, η οποία παραμένει σταθερή για όλη την διαδρομή, υπολογίζεται η κατανάλωση ενέργειας από τις εξισώσεις κίνησης. Σε κάθε όχημα ανατίθεται ποσοστό αρχικής φόρτισης και ποσοστό άγχους του οδηγού, το οποίο θεωρείται σταθερό, και τελικά από το σύστημα ασαφούς λογικής εξάγεται το χιλιόμετρο που ο κάθε οδηγός αναζητάει για σταθμό φόρτισης ελέγχοντας κάθε 5 λεπτά το ποσοστό φόρτισης της μπαταρίας και το ποσοστό άγχους του οδηγού. Αποτελέσματα ενεργειακής αξιολόγησης:

Στο ιστόγραμμα [5.35](#), φαίνεται το χιλιόμετρο και το πλήθος των οδηγών που αναζητούν σταθμό φόρτισης σε αυτό.



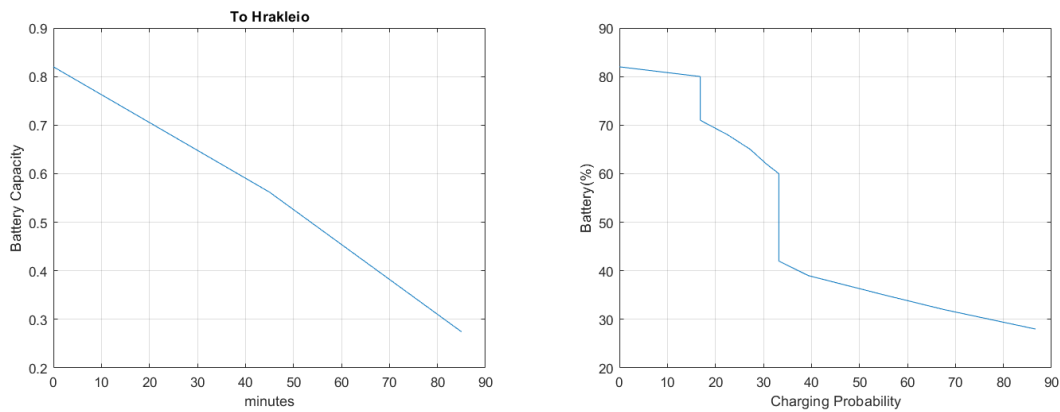
Σχήμα 5.35: Ιστόγραμμα διαδρομής Χανιά - Ηράκλειο

Παρακάτω παρουσιάζονται ενδεικτικά παραδείγματα για οδηγούς αυτοκινήτων που αναζητούν σταθμό φόρτισης κατά την διάρκεια της διαδρομής Χανιά-Ηράκλειο. Απεικονίζεται παράδειγμα για κάθε ένα από τα τέσσερα αμιγώς ηλεκτρικά αυτοκίνητα και συγκεκριμένα παρουσιάζεται διάγραμμα με την πτώση της μπαταρίας κάθε πέντε λεπτά καθώς και το αποτέλεσμα του συστήματος ασαφούς λογικής.



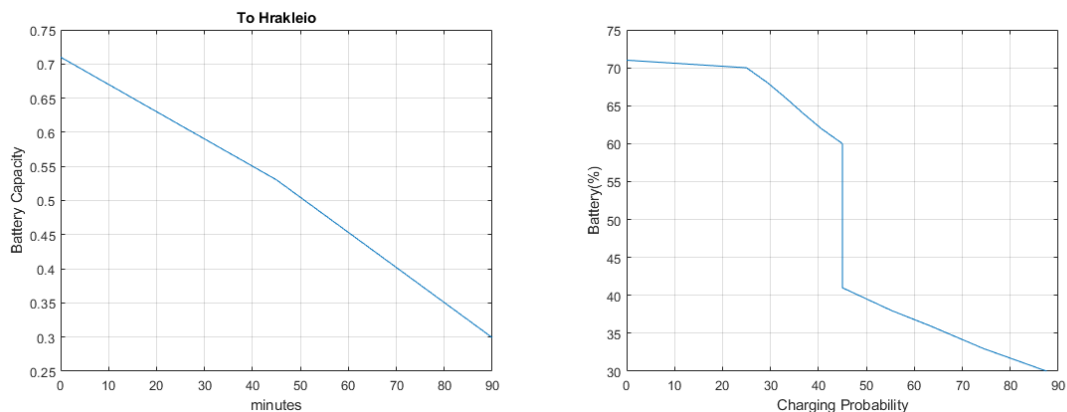
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.36: Audi Q4 e-tron 35: Ταχύτητα πριν το Ρέθυμνο $96 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $125 \frac{km}{h}$, Άγχος 47 %, Μπαταρία 78 %, Χιλιόμετρο 117 °



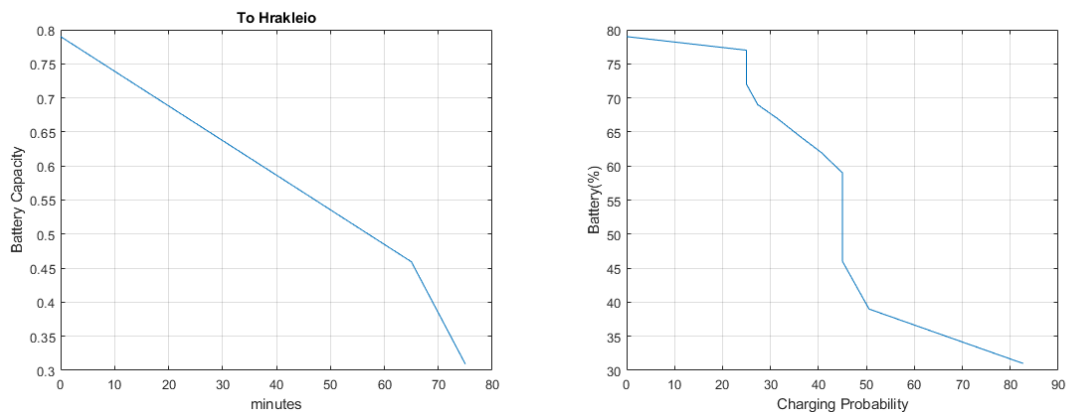
(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.37: Fiat 500e: Ταχύτητα πριν το Ρέθυμνο $86 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $96 \frac{km}{h}$, Άγχος 26 %, Μπαταρία 82 %, Χιλιόμετρο 128.7 °



(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.38: Nissan Leaf: Ταχύτητα πριν το Ρέθυμνο $66 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $76 \frac{km}{h}$, Άγχος 36 %, Μπαταρία 71 %, Χιλιόμετρο 110 °














(α') Διάγραμμα ποσοστού φόρτισης μπαταρίας (β') Διάγραμμα εξόδου συστήματος ασαφούς μέχρι την στιγμή αναζήτησης σταθμού φόρτι- λογικής σε σχέση με το ποσοστό φόρτισης της μπαταρίας

Σχήμα 5.39: Mazda MX-30: Ταχύτητα πριν το Ρέθυμνο $66 \frac{km}{h}$, Ταχύτητα μετά το Ρέθυμνο $116 \frac{km}{h}$, Άγχος 53 %, Μπαταρία 79 %, Χιλιόμετρο 94 °

Έπειτα, με βάση το χιλιόμετρο αναζήτησης σταθμού φόρτισης κάθε οδηγού, υπολογίζεται το ποιο όχημα και το πλήθος των οχημάτων που σταματάει σε κάθε σταθμό. Επιπλέον υπολογίζεται ο χρόνος μερικής και πλήρους φόρτισης και εξετάζοντας για κάθε σταθμό τα οχήματα που σταματούν σε αυτόν, υπολογίζεται το πλήθος των φορτιστών με βάση τον χρόνο φόρτισης του κάθε οχήματος. Ο αλγόριθμος PSO επαναλαμβάνει τον υπολογισμό του προβλήματος για διαφορετικούς συνδιασμούς τοποθέτησης των σταθμών φόρτισης και υπολογίζει το κόστος, όπου τελικά επιλέγει το καλύτερο. Είναι χρήσιμο να σημειωθεί ότι ο πίνακας με το χιλιόμετρο που αναζητάει ο οδηγός για σταθμό φόρτισης είναι ο ίδιος για κάθε επίλυση προβλήματος του PSO. Παρακάτω παρουσιάζονται τα αποτελέσματα του PSO για την επίλυση προβλήματος τοποθέτησης 8, 10, 12 και 14 σταθμών.














Το πλήθος φορτιστών, το χιλιόμετρο τοποθέτησης και το κόστος για τις βέλτιστες λύσεις κάθε προβλήματος είναι:

**Problem of 8 Charging stations
Chania-Heraklion**

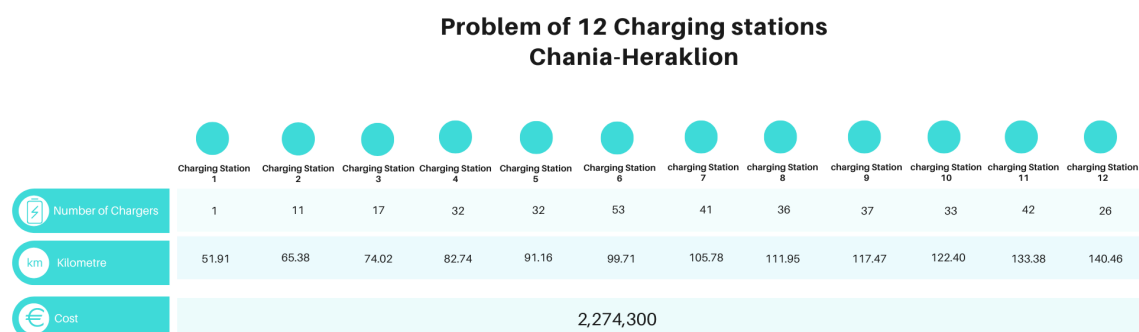
								
	Charging Station 1	Charging Station 2	Charging Station 3	Charging Station 4	Charging Station 5	Charging Station 6	charging Station 7	charging Station 8
 Number of Chargers	33	25	31	54	40	36	73	57
 Kilometre	75.25	82.78	89.51	98.06	105.41	112.67	125.05	139.85
 Cost	2,198,700							

Σχήμα 5.40: Πρόβλημα 8 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)

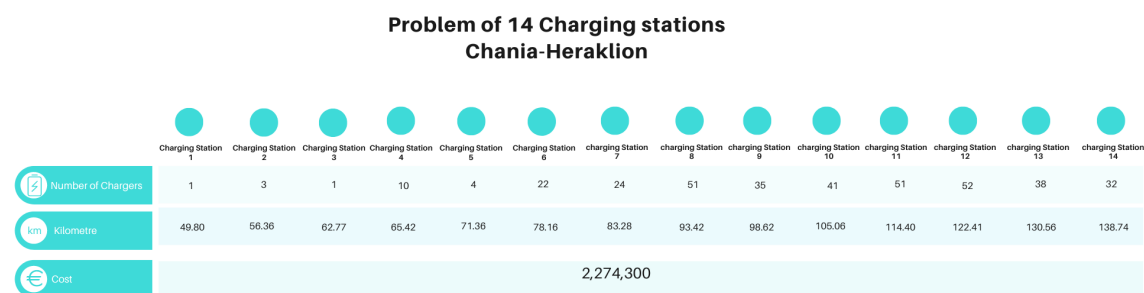
**Problem of 10 Charging stations
Chania-Heraklion**

										
	Charging Station 1	Charging Station 2	Charging Station 3	Charging Station 4	Charging Station 5	Charging Station 6	charging Station 7	charging Station 8	charging Station 9	charging Station 10
 Number of Chargers	11	24	25	26	61	40	46	47	28	47
 Kilometre	66.94	75.85	82.68	88.19	98.50	105.43	113.11	121.91	127.90	140.75
 Cost	2,236,500									

Σχήμα 5.41: Πρόβλημα 10 σταθμών - πλήθος φορτιστών, χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)

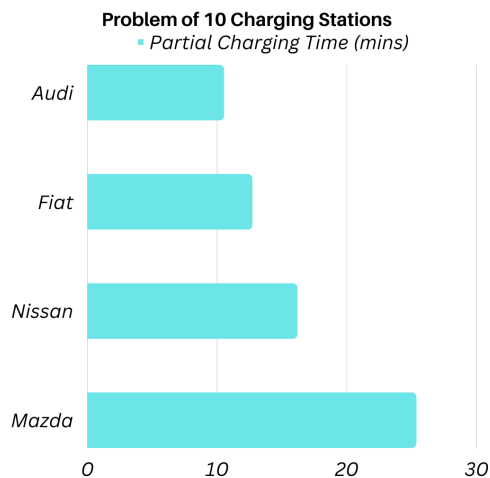


Σχήμα 5.42: Πρόβλημα 12 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)



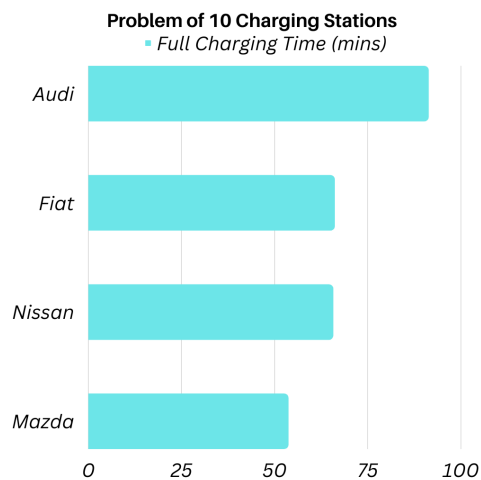
Σχήμα 5.43: Πρόβλημα 14 σταθμών - πλήθος φορτιστών,χιλιόμετρο τοποθέτησης και κόστος (Χανιά-Ηράκλειο)

Ενδεικτικές τιμές του μέσου χρόνου μερικής φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 10 σταθμών:



Σχήμα 5.44: Μέση τιμή χρόνου μερικής φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 10 σταθμών Χανιά - Ηράκλειο

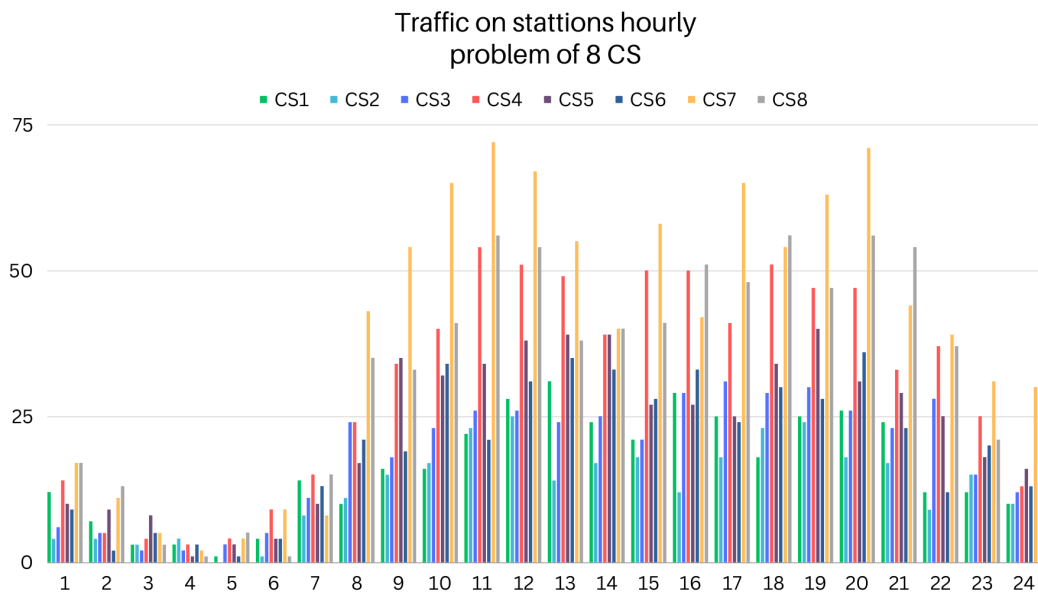
Ενδεικτικές τιμές του μέσου χρόνου πλήρους φόρτισης για κάθε μάρκα για την βέλτιστη λύση τοποθέτησης 10 σταθμών:



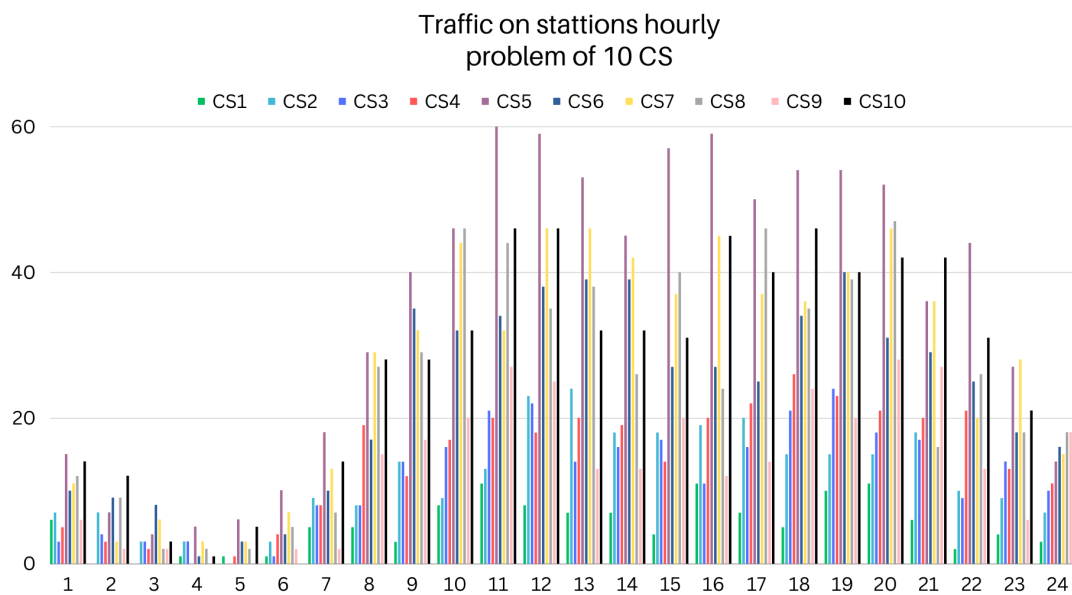
Σχήμα 5.45: Μέση τιμή χρόνου πλήρους φόρτισης ανα μάρκα για το πρόβλημα τοποθέτησης 10 σταθμών Χανιά - Ηράκλειο

Οι χρόνοι φόρτισης και για τα υπόλοιπα προβλήματα τοποθέτησης σταθμών της διαδρομής Χανιά - Ηράκλειο είναι παρόμοιοι.

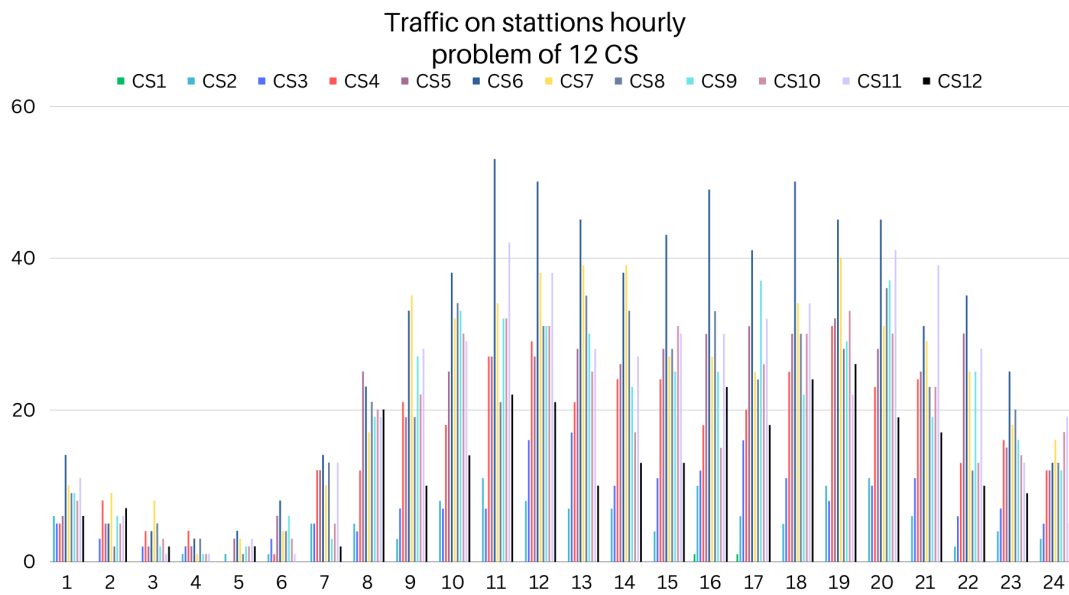
Το πλήθος των οχημάτων που σταματούν σε κάθε σταθμό για κάθε ώρα της ημέρας είναι:



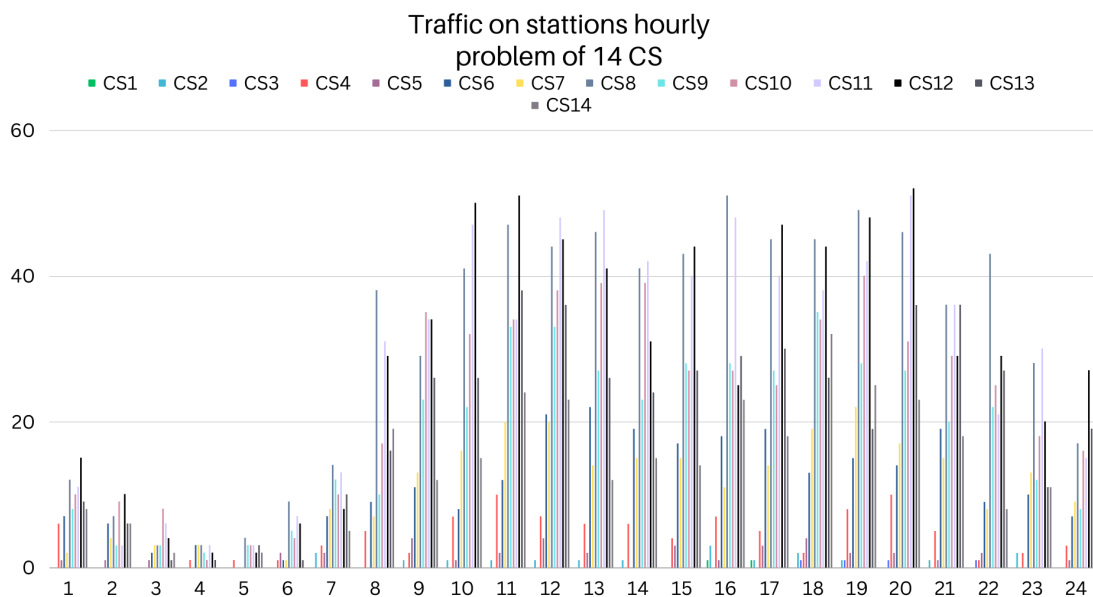
Σχήμα 5.46: Πρόβλημα 8 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο



Σχήμα 5.47: Πρόβλημα 10 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο



Σχήμα 5.48: Πρόβλημα 12 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο



Σχήμα 5.49: Πρόβλημα 14 σταθμών - πλήθος οχημάτων ανα σταθμό και ώρα Χανιά - Ηράκλειο

Από τα πειράματα φαίνεται ότι οι περισσότεροι οδηγοί αναζητούν για σταθμούς φόρτισης μεταξύ του 70-140 χιλιομέτρου με το μεγαλύτερο πλήθος να σημειώνεται στο

100-105 χιλιόμετρο. Περιμένουμε λοιπόν ότι οι περισσότεροι φορτιστές θα τοποθετηθούν στους σταθμούς που θα βρίσκονται σε αυτό το εύρος, κάτι το οποίο πράγματι συμβαίνει. Ωστόσο, για το πλήθος των φορτιστών χρειάζεται προσοχή αφού το πλήθος τους σε κάθε σταθμό συνδέεται με την ώρα που σταμάτησε κάθε όχημα σε αυτόν αλλά και με τον χρόνο φόρτισης του κάθε οχήματος. Συνεπώς, κάποιο όχημα που παραμένει στον σταθμό για χρόνο περισσότερο από 1 ώρα συμπεριλαμβάνεται και στην επόμενη, για να γίνει σωστή εκτίμηση για το πλήθος των φορτιστών. Για αυτόν τον λόγο οι φορτιστές σε κάποιους σταθμούς φαίνονται να είναι λίγο περισσότεροι από το πλήθος των οχημάτων που σταματούν σε αυτόν την ώρα που σημειώνεται η περισσότερη κίνηση. Επιπλέον, μεγαλύτερη κίνηση στους σταθμούς παρατηρείται από τις 8 το πρωί έως τις 11 το βράδυ.

5.5 Συμπεράσματα - Παρατηρήσεις

- Με βάση τα αποτελέσματα της ενεργειακής αξιολόγησης και το πλήθος των σταθμών, ανάλογα με το πρόβλημα που ερευνάται, υπολογίζεται η βέλτιστη τοποθέτηση των σταθμών καθώς και το πλήθος των φορτιστών. Ωστόσο, χρειάζεται προσοχή καθώς:
 - Το πλήθος των οχημάτων που κινούνται κάθε ώρα για κάθε σενάριο είναι σταθερό και δεν εξαρτάται από κάποια κατανομή
 - Το ποσοστό αρχικής φόρτισης της μπαταρίας και το ποσοστό άγχους που δίνεται σε κάθε οδηγό δίνεται βάση γκαουσιανής κατανομής
 - Η ταχύτητα που δίνεται σε κάθε σενάριο σε κάθε όχημα δίνεται με βάση το ποσοστό εμφάνισης της ταχύτητας από τις πραγματικές ταχύτητες που υπολογίστηκαν
 - Συνεπώς για την διερεύνηση της βέλτιστης τοποθέτησης και του πλήθους των φορτιστών για κάθε σενάριο και κάθε πρόβλημα χρησιμοποιούνται τα αποτελέσματα της ίδιας ενεργειακής αξιολόγησης
- Η τοποθέτηση των σταθμών εξαρτάται από την συνάρτηση κόστους, με στόχο την ομοιόμορφη κατανομή των φορτιστών στους σταθμούς, την εξυπηρέτηση όλων των οχημάτων και την αποφυγή τοποθέτησης σταθμών σε κοντινή απόσταση. Το

- πλήθος των φορτιστών που τοποθετείται σε κάθε σταθμό εξαρτάται από τον αριθμό των οχημάτων που σταματούν στον σταθμό, την ώρα της ημέρας που σταματούν σε αυτόν καθώς και τον χρόνο φόρτισης που απαιτείται για κάθε όχημα.
- Από τις προσομοιώσεις, με βάση την ενεργειακή αξιολόγηση φαίνεται ότι το μεγαλύτερο μέρος των οχημάτων για τα σενάρια Χανιά – Ρέθυμνο και Ρέθυμνο - Ηράκλειο σταματούν στους τελευταίους σταθμούς, δηλαδή στους σταθμούς που συνήθως είναι κοντά στον προορισμό, ενώ στο σενάριο Χανιά – Ηράκλειο φαίνεται να μοιράζονται στους σταθμούς που τοποθετούνται ενδιάμεσα της διαδρομής. Επιπλέον, οι ώρες που σημειώνεται περισσότερη κίνηση στους σταθμούς, διαφέρουν σε κάθε σενάριο.
 - Στο Χανιά – Ρέθυμνο παρατηρείται μεγαλύτερη κίνηση στους σταθμούς από τις 9 το πρωί έως τις 9 το βράδυ.
 - Στο Ρέθυμνο - Ηράκλειο παρατηρείται μεγαλύτερη κίνηση στους σταθμούς από τις 10 το πρωί έως και τις 9 το βράδυ.
 - Στο Χανιά – Ηράκλειο παρατηρείται μεγαλύτερη κίνηση στους σταθμούς από τις 8 το πρωί έως τις 11 το βράδυ.

Bibliography

- [1] Rafael Padilla, Sergio Netto, and Eduardo da Silva. A survey on performance metrics for object-detection algorithms. 07 2020.
- [2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.
- [3] Akansha Bathija and Grishma Sharma. Visual object detection and tracking using yolo and sort. *International Journal of Engineering Research Technology*, 8(11), 2019.
- [4] matlab. <https://www.mathworks.com/discovery/what-is-matlab.html>.
- [5] Margaret Smith and Johnathan Castellano. Costs associated with non-residential electric vehicle supply equipment: Factors to consider in the implementation of electric vehicle charging stations. Technical report, 2015.
- [6] Mehrdad Ehsani, Yimin Gao, Stefano Longo, and Kambiz M Ebrahimi. *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018.
- [7] Audi q4e-tron35. <https://ev-database.org/car/1179/Audi-Q4-e-tron-35>.
- [8] Fiat-500e-hatchback. <https://ev-database.org/car/1285/Fiat-500e-Hatchback-42-kWh>.
- [9] Nissan-leaf. <https://ev-database.org/car/1106/Nissan-Leaf>.
- [10] Mazda-mx-30. <https://ev-database.org/car/1680/Mazda-MX-30>.
- [11] Naeem Abas, A Kalair, and Nasrullah Khan. Review of fossil fuels and future energy technologies. *Futures*, 69:31–49, 2015.

- [12] REBECCA MATULKA. The history of the electric car. <https://www.energy.gov/articles/history-electric-car>, SEPTEMBER 15, 2014.
- [13] Massimo Guarnieri. Looking back to electric cars. In *2012 Third IEEE HISTory of ELection-technology CONference (HISTELCON)*, pages 1–6. IEEE, 2012.
- [14] Range of full electric vehicles. <https://ev-database.org/cheatsheet/range-electric-car>, 2022 EV Database - v4.3.
- [15] What is machine learning (ml)? <https://towardsai.net/p/machine-learning/what-is-machine-learning-ml-b58162f97ec7>, April 30, 2019.
- [16] Supervised machine learning. <https://www.javatpoint.com/supervised-machine-learning>.
- [17] Regression analysis in machine learning. <https://www.javatpoint.com/regression-analysis-in-machine-learning>.
- [18] Classification algorithm in machine learning. <https://www.javatpoint.com/classification-algorithm-in-machine-learning>.
- [19] Unsupervised machine learning. <https://www.javatpoint.com/unsupervised-machine-learning>.
- [20] Reinforcement learning tutorial. <https://www.javatpoint.com/reinforcement-learning#Algorithm>.
- [21] IBM. What is computer vision? <https://www.ibm.com/topics/computer-vision>.
- [22] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [23] IBM. Convolutional neural networks. <https://www.ibm.com/cloud/learn/convolutional-neural-networks>.
- [24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

- [25] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [27] Nikhil Yadav and Utkarsh Binay. Comparative study of object detection algorithms. *International Research Journal of Engineering and Technology (IR-JET)*, 4(11):586–591, 2017.
- [28] Do Thuan. Evolution of yolo algorithm and yolov5: The state-of-the-art object detention algorithm. 2021.
- [29] Tuan Linh Dang, Gia Tuyen Nguyen, and Thang Cao. Object tracking using improved deep sort yolov3 architecture. *ICIC Express Letters*, 14(10):961–969, 2020.
- [30] Leandro Arab Marcomini and André Luiz Cunha. A comparison between background modelling methods for vehicle segmentation in highway traffic videos. *arXiv preprint arXiv:1810.02835*, 2018.
- [31] Background subtraction. https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html.
- [32] Matthew Urwin. Drone technology: What is a drone? <https://builtin.com/drones>, August 18, 2022.
- [33] Ben Lutkevich. drone (uav). <https://www.techtarget.com/iotagenda/definition/drone>.
- [34] Ffmpeg. <https://ffmpeg.org/>.
- [35] Roboflow. <https://docs.roboflow.com/>.
- [36] Anaconda. <https://www.anaconda.com/>.
- [37] Weight and biased. <https://docs.wandb.ai/>.
- [38] Python. <https://www.python.org/doc/essays/blurb/>.

- [39] OpenCV. <https://opencv.org/>.
- [40] Glenn Jocher. yolov5. <https://github.com/ultralytics/yolov5>.
- [41] Γεώργιος Γεωργίου, Χρίστος Ξενοφώντος. *Εισαγωγή στη MATLAB*. 2007.
- [42] Random Numbers from a Discrete Distribution. <https://www.mathworks.com/matlabcentral/fileexchange/34101-random-numbers-from-a-discrete-distribution>.
- [43] fuzzy. <https://www.mathworks.com/help/fuzzy/newfis.html>.
- [44] Bai, Qinghai. Analysis of particle swarm optimization algorithm. *Computer and information science*, 3(1):180, 2010.
- [45] Particle swarm optimization algorithm. <https://yarpiz.com/50/ypea102-particle-swarm-optimization>.