ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Προηγμένο σύστημα υποβοήθησης οδηγού με αναγνώριση και προειδοποίηση ατυχήματος με χρήση Raspberry Pi Python και OpenCV

Κωνσταντίνος Κατσάνος
Α.Μ. 711161059

Εισηγητής: Βογιατζής Ιωάννης, Καθηγητής

(Κενό φύλλο)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Προηγμένο σύστημα υποβοήθησης οδηγού με αναγνώριση και προειδοποίηση ατυχήματος με χρήση Raspberry Pi Python και OpenCV

Κωνσταντίνος Κατσάνος
Α.Μ. 711161059

Εισηγητής:

Βογιατζής Ιωάννης, Καθηγητής

Ioannis Voyiatzis
Digitally signed by Ioannis Voyiatzis
Date: 2023.03.27 16:11:46 +03'00'

Εξεταστική Επιτροπή:

Σπυρίδων Ματιάτος

Spyridon Matiatos
Digitally signed by Spyridon Matiatos
Date: 2023.03.27 16:29:36 +03'00'

Αμοργίνος Ιωάννης

Ioannis Amorginos
Digitally signed by Ioannis Amorginos
Date: 2023.03.27 16:21:15 +03'00'

Ημερομηνία εξέτασης …./…./…….

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Κωνσταντίνος Κατσάνος του Χρήστου, με αριθμό μητρώου 711161059 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:
«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

(Κενό φύλλο)

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές μου, που μου εμπιστεύτηκαν τη εκπόνηση της συγκεκριμένης διπλωματικής εργασίας, για την εμπιστοσύνη και τη στήριξη που μου παρείχαν κατά τη διάρκεια της υλοποίησής της.

Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου, για την κατανόηση που επέδειξαν κατά το χρονικό διάστημα εκπόνησης της παρούσας εργασίας.

(Κενό φύλλο)

ΠΕΡΙΛΗΨΗ

Η πλειονότητα των τροχαίων ατυχημάτων οφείλεται σε περιστασιακή απόσπαση της προσοχής του οδηγού. Στα σύγχρονα αυτοκίνητα έχουν αρχίσει να εγκαθίστανται συστήματα υποβοήθησης του οδηγού που συμβάλλουν στην επαναφορά της προσοχής του οδηγού στην οδήγηση και συνεπώς και στη μείωση των ατυχημάτων. Ωστόσο για αρκετούς πολίτες η αγορά νέου αυτοκινήτου δεν είναι οικονομικά εφικτή, με αποτέλεσμα να απαιτείται μεγάλο χρονικό διάστημα μέχρις ότου το σύνολο των κυκλοφορούντων αυτοκινήτων να διαθέτει τέτοια συστήματα.

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ανοιχτού κώδικα λογισμικού ενσωματωμένου σε οικονομικά προσιτό Σύστημα Raspberry Pi, το οποίο μπορεί να εγκατασταθεί σε υφιστάμενα παλαιά οχήματα και με χρήση μηχανικής όρασης και άλλων αισθητήρων να έχει τη δυνατότητα να αναγνωρίσει και να προειδοποιήσει για επικείμενα τροχαία ατυχήματα ή άλλες επικίνδυνες καταστάσεις, με σκοπό την πρόληψή τους από τον οδηγό.

ABSTRACT

The majority of road accidents are due to occasional driver distraction. In modern cars, driver assistance systems have begun to be installed that help to restore the driver's attention to driving and therefore reduce accidents. However, for many citizens the purchase of a new car is not financially feasible, as a result of which a long period of time is required until all cars in circulation have such systems.

The purpose of this work is to develop open-source software embedded in an affordable Raspberry Pi System, which can be installed in existing old vehicles and with the use of computer vision and other sensors have the ability to recognize and warn of impending traffic accidents or other dangerous situations, with the goal of the driver taking action and preventing them from escalating.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Κεφάλαιο 1

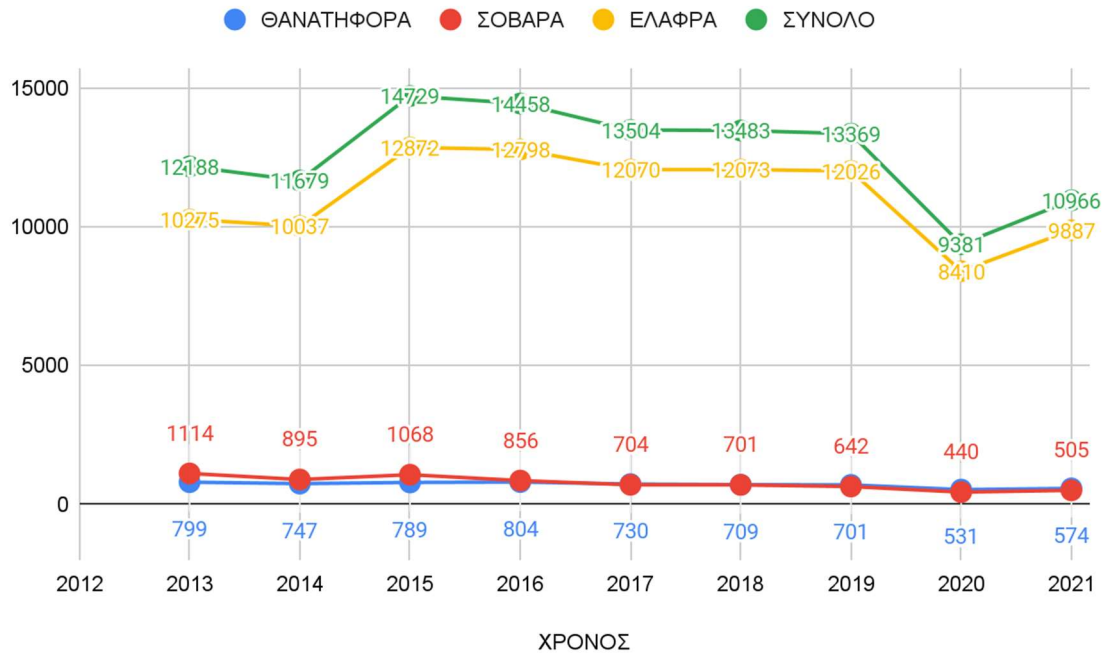
Οδική ασφάλεια και προηγμένα συστήματα υποβοήθησης του οδηγού

1.1 Τροχαία Ατυχήματα με εστίαση στα ατυχήματα εξαιτίας της απόσπασης της προσοχής του οδηγού

Το αυτοκίνητο έχει καταστεί προ πολλού απαραίτητο στον άνθρωπο διότι παρέχει τόσο προσωπικά (π.χ. κινητικότητα, ανεξαρτησία, ευκολία) όσο και κοινωνικά οφέλη (π.χ. δημιουργία θέσεων εργασίας, παροχή μεταφορικών υπηρεσιών) και πλέον το πλήθος των αυτοκινήτων που κυκλοφορούν στους δημόσιους δρόμους παγκοσμίως είναι εξαιρετικά μεγάλο και συνεχίζει να αυξάνεται ραγδαία, ιδίως στην Κίνα στην Ινδία και άλλες πρόσφατα εκβιομηχανισμένες χώρες.

Τα αυτοκίνητα αυτά σε συνδυασμό με την αστικοποίηση έχουν προκαλέσει μεγάλο συνωστισμό στις πόλεις, ιδιαίτερα τις μεγαλύτερες, με τη συμφόρηση της κυκλοφορίας («μποτιλιάρισμα»), τις ώρες αιχμής, να αποτελεί πλέον τον κανόνα και όχι την εξαίρεση.

Λόγω της μεγάλης αυτής συγκέντρωσης αυτοκινήτων τα κάθε είδους τροχαία ατυχήματα (ελαφρά, σοβαρά, θανατηφόρα) είναι δυστυχώς αναπόφευκτα και συνεπώς αναμενόμενα. Σύμφωνα με επίσημα στοιχεία της Ελληνικής Αστυνομίας (ΕΛ.ΑΣ.) **[1]** το ετήσιο πλήθος των τροχαίων ατυχημάτων (θανατηφόρων, σοβαρών, ελαφρών και συνολικό πλήθος) στην Ελλάδα από το έτος 2013 έως και το έτος 2021 είναι το εμφανιζόμενο στο Σχήμα 1.1:



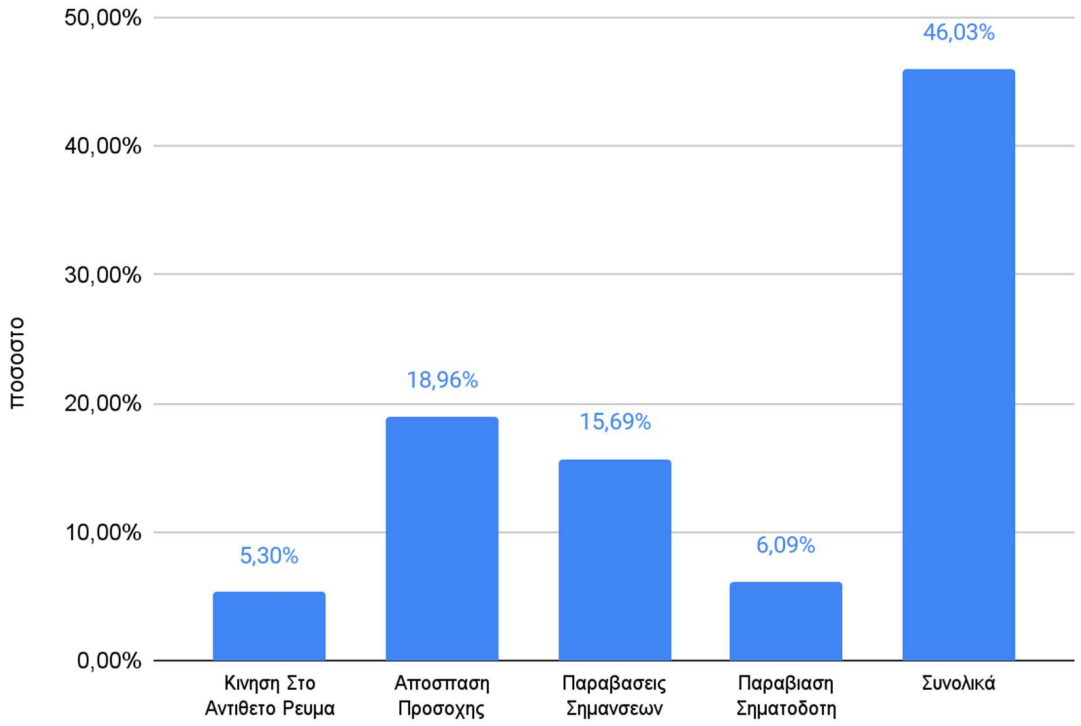
Σχήμα 1.1: Πλήθος τροχαίων ατυχημάτων περιόδου 2013 ÷ 2021

Βάσει των στοιχείων του Σχήματος 1.1 παρατηρείται ότι η συχνότητα εμφάνισης θανατηφόρων και σοβαρών ατυχημάτων βαίνει ελαφρώς μειούμενη στο εξεταζόμενο χρονικό διάστημα 2013 ÷ 2021. Η μείωση αυτή θα μπορούσε να αποδοθεί, κατά κύριο λόγο, στη συνεχή αντικατάσταση των παλαιών, μη επαρκώς ασφαλών, αυτοκινήτων από νέα σύγχρονα, σύμφωνα με στοιχεία της Διεύθυνσης Μεταφορών των Η.Π.Α. **[2]**.

Ωστόσο το συνολικό πλήθος των τροχαίων ατυχημάτων, παρά τη μείωση που εμφανίζεται κατά τα έτη 2020 και 2021 που μπορεί να αποδοθεί στη μειωμένη κινητικότητα των πολιτών εξαιτίας της πανδημίας Covid-19 **[3]**, διατηρείται σχεδόν σταθερό. Παρόμοια σταθεροποίηση έχει παρατηρηθεί και σε άλλες χώρες, σύμφωνα με στοιχεία των Ηνωμένων Εθνών (Statistics of Road Traffic Accidents in Europe and North America) **[4]**. Ποσοστό των ανωτέρω τροχαίων ατυχημάτων έχουν ως αιτία τον άνθρωπο – οδηγό, ο οποίος οδηγεί μη ασφαλώς είτε εκούσια (ηθελημένα) είτε ακούσια.

Τα ατυχήματα που συμβαίνουν εξαιτίας εκούσιας μη ασφαλούς οδήγησης θα μπορούσαν να μειωθούν μεσομακροπρόθεσμα μέσω της συμμετοχής οδηγών σε σεμινάρια ασφαλούς οδικής συμπεριφοράς, με σκοπό τη βελτίωση της οδικής κουλτούρας, της νοοτροπίας και του πολιτισμού (δηλαδή της συνείδησης των ανθρώπων κατά τη διάρκεια της κυκλοφορίας τους στο επικίνδυνο οδικό περιβάλλον) καθώς και, κυρίως, μέσω της εισαγωγής σχετικών σεμιναρίων στην πρωτοβάθμια και δευτεροβάθμια δημόσια εκπαίδευση.

Σύμφωνα με επίσημα στοιχεία της Ελληνικής Αστυνομίας (ΕΛ.ΑΣ.) **[1]**, τα κύρια αίτια πρόκλησης τροχαίων ατυχημάτων, στα οποία ως υπαίτιος καταγράφεται ο οδηγός, είναι τα εμφανιζόμενα στο ακόλουθο Σχήμα 1.2:



Σχήμα 1.2: Τροχαία ατυχήματα έτους 2021 εξαιτίας υπαιτιότητας οδηγού, ανά αιτία

Βάσει των στοιχείων του Σχήματος 1.2 φαίνεται ότι περίπου το 19% των αιτίων πρόκλησης ατυχήματος εξ΄ υπαιτιότητας του οδηγού οφείλονται στην απόσπαση της προσοχής του οδηγού από την οδήγηση.

1.2 Αιτίες απόσπασης προσοχής του οδηγού, συμβολή της απόσπασης προσοχής στην αύξηση των τροχαίων ατυχημάτων και μέτρα μείωσης του προβλήματος

Η απόσπαση της προσοχής του οδηγού οφείλεται στην εστίασή του σε ένα αντικείμενο, άτομο, δραστηριότητα ή συμβάν που δεν σχετίζεται με την οδήγηση και έχει ως συνέπεια τη μείωση της ετοιμότητάς του και της ικανότητας έγκαιρης λήψης αποφάσεων καθώς και την αύξηση της πιθανότητας πρόκλησης τροχαίου ατυχήματος.

Η απόσπαση της προσοχής του οδηγού μπορεί να οφείλεται τόσο σε εξωτερικούς (εκτός του οχήματος) όσο και, κυρίως, σε εσωτερικούς (εντός του οχήματος) παράγοντες και μπορεί να είναι οπτική (π.χ. ο οδηγός δεν κοιτά το δρόμο αλλά διαφημίσεις εκτός δρόμου), ηχητική (π.χ. η προσοχή του παρεμποδίζεται από διάφορους ήχους, όπως συνομιλίες, μουσική), φυσική (π.χ. χρησιμοποιεί το χέρι ή τα χέρια του για κάποια ενέργεια πέραν της οδήγησης) ή να οφείλεται στο ότι ο οδηγός σκέφτεται θέματα που τον απασχολούν ή συνομιλεί με τους συνεπιβαίνοντες ή είναι κουρασμένος και η συγκέντρωσή του στην οδήγηση μειώνεται.

Επισημαίνεται ότι καθώς το πλήθος και η πολυπλοκότητα των τεχνολογιών που χρησιμοποιούνται εντός των οχημάτων έχει αυξηθεί θεαματικά τα τελευταία χρόνια (εκτεταμένη χρήση κινητών τηλεφώνων, συστημάτων πλοήγησης, ηχητικών συστημάτων, άλλων συστημάτων υποβοήθησης του οδηγού κ.λπ. κατά τη διάρκεια της οδήγησης) είναι πιθανό ότι το ποσοστό των τροχαίων ατυχημάτων εξαιτίας απόσπασης προσοχής του οδηγού θα αυξηθεί τα επόμενα χρόνια.

Η αλληλεπίδραση του οδηγού με τους συνεπιβάτες και η χρήση κινητού τηλεφώνου αποτελούν τις κατά σειρά κυριότερες αιτίες απόσπασης προσοχής και πρόκλησης τροχαίου ατυχήματος, διότι οι οδηγοί καταναλώνουν αντίστοιχα το 12,2% και το 6,4% του χρόνου οδήγησης απασχολούμενοι με τις εν λόγω δραστηριότητες ενώ οδηγούν.

Η απόσπαση της προσοχής του οδηγού επηρεάζει τη συγκέντρωσή του (χέρια εκτός τιμονιού, μάτια εκτός δρόμου), τη συμπεριφορά του (ταχύτητα οχήματος, απόσταση από προπορευόμενο όχημα, απόσταση από τη δεξιά οριογραμμή, χρόνος αντίδρασης οδηγού) και αυξάνει την πιθανότητα εμπλοκής σε τροχαίο ατύχημα, όπως ιδίως:

- σύγκρουση με προπορευόμενο όχημα
- σύγκρουση με απρόβλεπτο αντικείμενο στο οδόστρωμα
- παρέκκλιση από τη λωρίδα κυκλοφορίας ή ακόμα και από αυτό το ρεύμα κυκλοφορίας του οχήματος
- παραβίαση σήματος «STOP» και άλλων σημάνσεων προτεραιότητας
- παραβίαση ερυθρού σηματοδότη

Για την αντιμετώπιση του προβλήματος της απόσπασης της προσοχής του οδηγού κατά την οδήγηση λαμβάνονται διεθνώς μια σειρά από μέτρα, όπως ιδίως νομοθετικά μέτρα για την απαγόρευση χρήσης κινητού τηλεφώνου κατά την οδήγηση, δημόσιες καμπάνιες ευαισθητοποίησης του κοινού για την επικινδυνότητα της απόσπασης προσοχής κατά την οδήγηση, σχετική εκπαίδευση των υποψήφιων οδηγών κατά τη διαδικασία της λήψης άδειας οδήγησης και τακτικές σχετικές εκπαιδεύσεις επαγγελματιών οδηγών, εγκατάσταση λωρίδων επαγρύπνησης (rumble strips or alert strips) με ήχο και δόνηση επί των οδών, απαγόρευση τοποθέτησης διαφημιστικών πινακίδων παραπλεύρως των δρόμων κ.λπ..

Όλα τα ανωτέρω αναφερόμενα στοιχεία, αναφορικά με την απόσπαση της προσοχής του οδηγού, την επικινδυνότητά της και τα μέτρα που λαμβάνονται για τη μείωση των δυσμενών επιπτώσεών της, περιλαμβάνονται σε πρόσφατη (Μάρτιος 2022) αναφορά της Ευρωπαϊκής Ένωσης (European Road Safety Observatory – Road Safety Thematic Report, Driver Distraction), που αφορά στην επίδραση της απόσπασης προσοχής των οδηγών στην οδική ασφάλεια **[5]**. Στην εν λόγω αναφορά αναφέρεται ότι, το ποσοστό πρόκλησης ατυχήματος εξαιτίας απόσπασης της προσοχής του οδηγού εκτιμάται περίπου στο 5 ÷ 25% του συνόλου των τροχαίων ατυχημάτων. Ωστόσο σημειώνεται ότι αφενός το ανωτέρω ποσοστό μπορεί να είναι ακόμη μεγαλύτερο, αλλά είναι δύσκολο να αποτιμηθεί επακριβώς λόγω των δυσκολιών κωδικοποίησης της απόσπασης της προσοχής του οδηγού ως παράγοντα που συμβάλλει σε ένα ατύχημα αφού αυτό συμβεί, και αφετέρου ότι σε νεότερες σχετικές έρευνες αναφέρεται ότι το ποσοστό συμμετοχής της απόσπασης προσοχής του οδηγού στο σύνολο των τροχαίων ατυχημάτων είναι υψηλότερο από την προαναφερόμενη εκτίμηση.

Σε αντίστοιχη αναφορά, του έτους 2016, της Διοίκησης Ασφάλειας Εθνικών Οδών της Διεύθυνσης Μεταφορών των Η.Π.Α. (Visual-Manual Driver Distraction Guidelines for Portable and Aftermarket Devices) **[6]** αναφέρεται ότι για το έτος 2015 το ποσοστό πρόκλησης ατυχήματος εξαιτίας απόσπασης της προσοχής του οδηγού ανήλθε περίπου στο 16% του συνόλου των τροχαίων ατυχημάτων και ότι ειδικά το αντίστοιχο ποσοστό των θανατηφόρων ατυχημάτων παρουσίασε αύξηση έναντι αυτού του προηγουμένου έτους (2014) κατά 8,8%.

Περαιτέρω με το άρθρο 6 του Ευρωπαϊκού Κανονισμού (EU) 2019/2144 **[7]**, που αφορά στην έγκριση τύπου όλων των μηχανοκίνητων οχημάτων, έχει καταστεί υποχρεωτική η εγκατάσταση προηγμένων συστημάτων σε όλες τις κατηγορίες οχημάτων. Μεταξύ των συστημάτων αυτών περιλαμβάνονται σύστημα προειδοποίησης υπνηλίας και διάσπασης προσοχής του οδηγού (driver drowsiness and attention warning) και προηγμένο σύστημα προειδοποίησης διάσπασης της προσοχής του οδηγού (advanced driver distraction warning), τα οποία έχουν καταστεί υποχρεωτικά από τις 6 Ιουλίου 2022 (άρθρα 6 και 19 του Κανονισμού).

Σύμφωνα με μελέτη, του έτους 2017, Ευρωπαϊκής Επιτροπής που αφορά σε εκτίμηση του κόστους των τροχαίων ατυχημάτων στις Ευρωπαϊκές χώρες (Crash cost estimation for European Countries) **[8]**, στην οποία συμμετέχει και η Ελλάδα (με εκπρόσωπο από το Ε.Μ.Π.), το συνολικό κόστος των τροχαίων ατυχημάτων εκτιμάται ότι κυμαίνεται μεταξύ του 0,4% και του 4,1% του Ακαθάριστου Εθνικού Προϊόντος (Α.Ε.Π.), αναλόγως της χώρας (το κόστος των θανατηφόρων ατυχημάτων εκτιμάται ότι ανέρχεται στο 7,4%÷55% του συνολικού κόστους, ενώ των ατυχημάτων με σοβαρούς και ελαφρούς τραυματισμούς στο 14%÷77% και 1,9%÷34% αντιστοίχως, αναλόγως της χώρας).

Ένα τυπικό (standard) κόστος θανατηφόρου ατυχήματος εκτιμάται σε 2,3m€, ενώ ενός ατυχήματος με σοβαρό και ελαφρό τραυματισμό στο 13% και 1% του κόστους του θανατηφόρου αντιστοίχως. Για τις 28 χώρες – μέλη της Ευρωπαϊκής Ένωσης το συνολικό κόστος εξαιτίας των τροχαίων ατυχημάτων εκτιμάται σε περίπου 270b€ ή 1,8% του Ευρωπαϊκού Α.Ε.Π. (στην Ελλάδα σε 2,746b€), αν και η εν λόγω τιμή θεωρείται υποεκτιμημένη (λόγω υποεκτίμησης στοιχείων κόστους από ορισμένες χώρες), με πλέον πιθανό το συνολικό κόστος να ανέρχεται στο 3% του Ευρωπαϊκού Α.Ε.Π.

1.3 Προηγμένα συστήματα υποβοήθησης του οδηγού (Συστήματα ADAS)

Η ασφάλεια των αυτοκινήτων ήταν εξαιρετικά σημαντική από τις πρώτες μέρες της χρήσης τους. Αρκετοί κατασκευαστές αυτοκινήτων προσπάθησαν να αντιμετωπίσουν αυτό το ζήτημα με την ανάπτυξη διαφόρων συστημάτων ασφαλείας τόσο για την προστασία των επιβατών μέσα στο όχημα όσο και για την πρόληψη τραυματισμών σε ανθρώπους έξω από αυτό.

Αυτά τα συστήματα ασφαλείας ταξινομούνται κυρίως σε δύο τύπους: 1) παθητικά (ή αντιδραστικά) και 2) ενεργητικά (ή προληπτικά).

Τα συστήματα παθητικής ασφάλειας προστατεύουν τους επιβάτες του οχήματος από τραυματισμούς μετά από σύγκρουση, π.χ. ζώνες ασφαλείας, αερόσακοι, ταμπλό με επένδυση.

Λόγω της συνεχούς ζήτησης των καταναλωτών για ασφαλέστερα οχήματα, συστήματα παθητικής ασφάλειας, που υπόκεινται σε συνεχή ανάπτυξη για πολλές δεκαετίες, έχουν πλέον ενισχυθεί από συστήματα ενεργητικής ασφάλειας, τα οποία επιδιώκουν να αποτρέψουν μια σύγκρουση.

Τα συστήματα ενεργητικής ασφαλείας είναι σήμερα ένας από τους κύριους τομείς ενδιαφέροντος και έχουν σημειώσει σημαντική ανάπτυξη στο σύγχρονο αυτοκίνητο. Αυτά τα συστήματα είναι κοινώς γνωστά ως Προηγμένα Συστήματα Υποβοήθησης Οδηγού (Advanced Driver Assistance Systems - ADAS) και γίνονται όλο και περισσότερο δημοφιλή ως τρόπος διαφοροποίησης των προσφορών των κατασκευαστών αυτοκινήτων ενώ προάγουν την ασφάλεια των επιβαινόντων στα οχήματα και των πεζών.

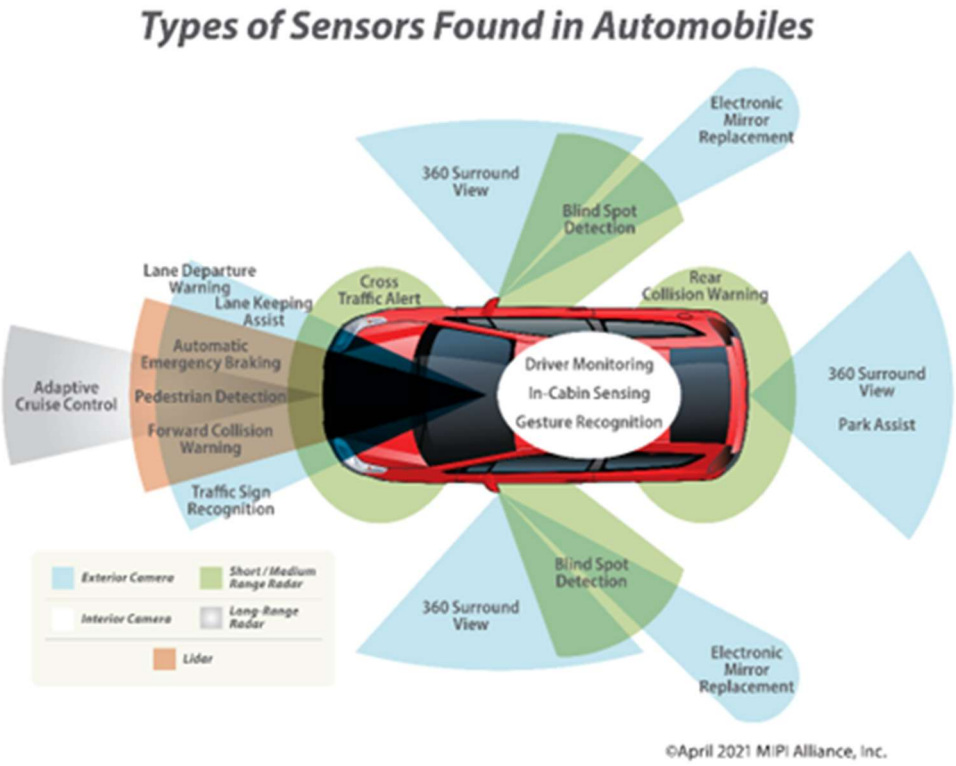
Δεδομένου ότι το νέο σύστημα αξιολόγησης του Euroncap **[9]** πιέζει για την υιοθέτηση των συστημάτων υποβοήθησης οδηγού, τα επόμενα χρόνια αναμένεται ραγδαία αύξηση της ενσωμάτωσής τους στα αυτοκίνητα.

Τα συστήματα ADAS βοηθούν τον οδηγό στην οδήγηση προειδοποιώντας τον σε περιπτώσεις μη ασφαλούς οδήγησης ή ακόμα και επιδρώντας στον έλεγχο του αυτοκινήτου σε επικίνδυνες καταστάσεις.

Τα μεγάλο πλήθος τροχαίων ατυχημάτων, το υψηλό κόστος τους και η αυξανόμενη ζήτηση των πελατών για έξυπνη ασφάλεια είναι μερικοί από τους βασικούς λόγους για την ανάπτυξη των ADAS.

Επιπλέον, με τον αυξανόμενο αριθμό των ηλεκτρονικών μονάδων ελέγχου και με την ενσωμάτωση διαφόρων τύπων αισθητήρων, υπάρχουν πλέον επαρκείς υπολογιστικές δυνατότητες στα οχήματα για την υποστήριξη εφαρμογών ADAS. Οι διαφορετικοί τύποι αισθητήρων, όπως π.χ. κάμερες, light detection and ranging (lidar), radio detection and ranging (radar), και αισθητήρες υπερήχων, επιτρέπουν μια ποικιλία από διαφορετικές λύσεις συστημάτων ADAS. Μεταξύ αυτών, σύστημα ADAS, που χρησιμοποιεί κυρίως κάμερες ως αισθητήρες όρασης, είναι δημοφιλές στα περισσότερα σύγχρονα οχήματα.

Στην Εικόνα 1.1 παρουσιάζονται μερικά από τα σύγχρονα χαρακτηριστικά των συστημάτων ADAS και οι αισθητήρες που χρησιμοποιούνται για την υλοποίησή τους **[10].**



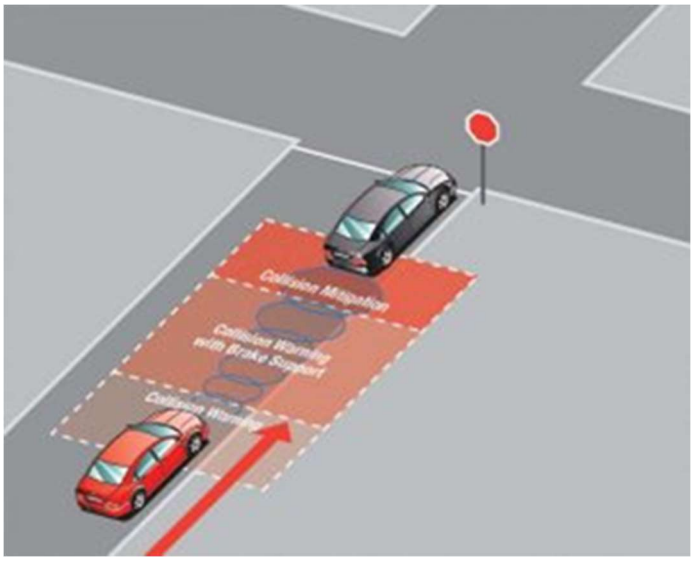
Εικόνα 1.1: Συστήματα ADAS και αισθητήρες τους **[10]**

Τα συστήματα αυτά είναι ιδανικά για τη μείωση των τροχαίων ατυχημάτων εξαιτίας απόσπασης προσοχής του οδηγού, διότι είτε επαναφέρουν την προσοχή του οδηγού στην οδήγηση του οχήματός του ώστε να δράσει αναλόγως στην κατάσταση που έχει προκύψει και να αποτρέψει ενδεχόμενο δυσμενές συμβάν είτε επιδρούν αυτά τα ίδια, αντί του οδηγού, στον έλεγχο του αυτοκινήτου, στην περίπτωση που ο τελευταίος δεν ανταποκρίνεται και δεν προβαίνει στις κατάλληλες κατά περίπτωση ενέργειες.

Οι δυνατότητες των διαφόρων συστημάτων ADAS **[11]** δεν είναι όλες κοινές, ωστόσο τα περισσότερα συστήματα έχουν τις ακόλουθες δύο βασικές δυνατότητες, οι οποίες είναι:

- Σύστημα προειδοποίησης μπροστινής σύγκρουσης (Forward Collision Warning -FCW):

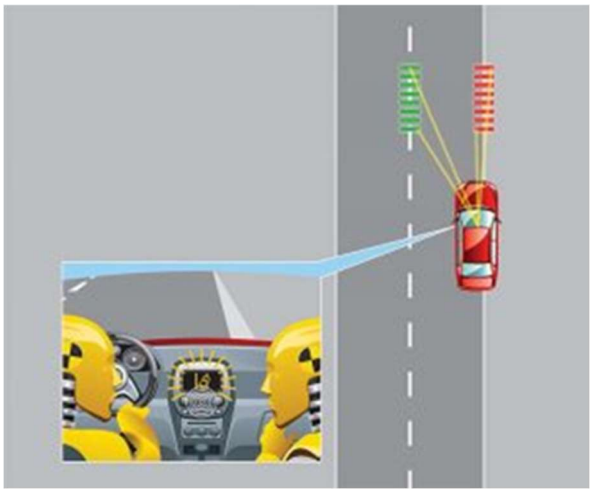
  Προειδοποίηση του οδηγού σε περίπτωση που η απόσταση του οχήματός του από το προπορευόμενο όχημα μειώνεται απότομα. Στο παρακάτω σχήμα φαίνεται η λειτουργία συστήματος προειδοποίησης μπροστινής σύγκρουσης και πέδησης, στο οποίο, αν ο οδηγός δεν αντιδράσει μετά την προειδοποίηση σύγκρουσης, η λειτουργία υποστήριξης πέδησης προετοιμάζει το σύστημα πέδησης για να αντιδράσει και εάν ο οδηγός δεν έχει εφαρμόσει τα φρένα, είναι δυνατόν να εφαρμοστεί αυτόματα ισχυρή δύναμη πέδησης.



Εικόνα 1.2 : Σύστημα προειδοποίησης μπροστινής σύγκρουσης **[11]**

- Σύστημα προειδοποίησης αποχώρησης από την λωρίδα κυκλοφορίας (Lane Departure Warning system - LDW):
  Προειδοποίηση του οδηγού σε περίπτωση που το όχημα ξεφεύγει από τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας που ακολουθεί. Πιο συγκεκριμένα στην περίπτωση αυτή μια κάμερα στερεωμένη στο εσωτερικό του παρμπρίζ κάτω από τον καθρέπτη ανιχνεύει τις λωρίδες κυκλοφορίας στο δρόμο. Εάν ο οδηγός αφήνει τη λωρίδα κυκλοφορίας που κινείται το όχημα χωρίς τη χρήση του φλας

προειδοποιεί τον οδηγό με ανάλογο οπτικό ή ηχητικό σήμα σε μια οθόνη. Σε μερικά συστήματα μπορεί ακόμη να δονείται το τιμόνι ή το κάθισμά του οδηγού και μάλιστα από την πλευρά που το όχημα παρασύρεται. Για να λειτουργήσουν τα συστήματα αυτά απαιτείται να υπάρχουν καθαρές οπτικά λωρίδες κυκλοφορίας (γραμμές) και καλές καιρικές συνθήκες.



Εικόνα 1.3 : Σύστημα προειδοποίησης αποχώρησης από την λωρίδα κυκλοφορίας **[11]**

Επιπρόσθετες λειτουργίες που διαθέτουν κάποια συστήματα είναι:

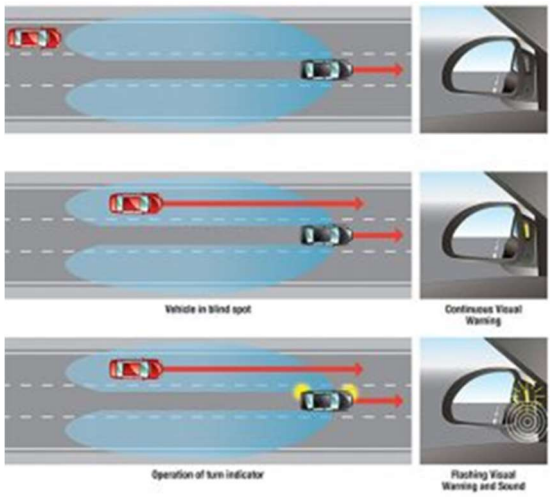
- Σύστημα αναγνώρισης σημάνσεων Κώδικα Οδικής Κυκλοφορίας - ΚΟΚ (Traffic Sign Recognition – TSR):

Τα συστήματα αυτά χρησιμοποιούν μια κάμερα η οποία μπορεί να διαβάζει τις πινακίδες σήμανσης, όπως π.χ. το όριο ταχύτητας και να τις εμφανίζει σε μια οθόνη στον πίνακα οργάνων.

Εικόνα 1.4 : Σύστημα αναγνώρισης σημάνσεων Κώδικα Οδικής Κυκλοφορίας **[11]**

- Σύστημα ανίχνευσης τυφλού σημείου (Blind spot monitor):

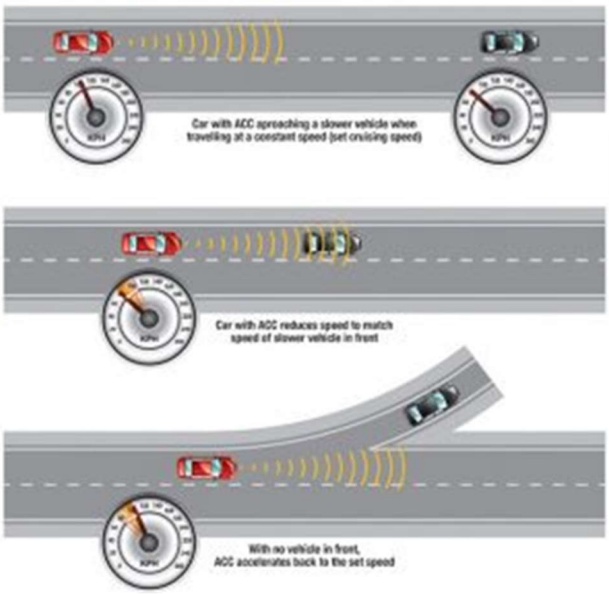
Χρήση καμερών ή ραντάρ για αναγνώριση αντικειμένων στην τυφλή γωνία οδήγησης (δηλαδή σε σημεία που δεν μπορεί να δει ο οδηγός μέσω των καθρεπτών) και ανάλογη προειδοποίηση του οδηγού όταν επιχειρεί να αλλάξει λωρίδα κυκλοφορίας



Εικόνα 1.5 : Σύστημα ανίχνευσης τυφλού σημείου **[11]**

- Προσαρμοζόμενο σύστημα ελέγχου ταχύτητας (Adaptive Cruise Control) :

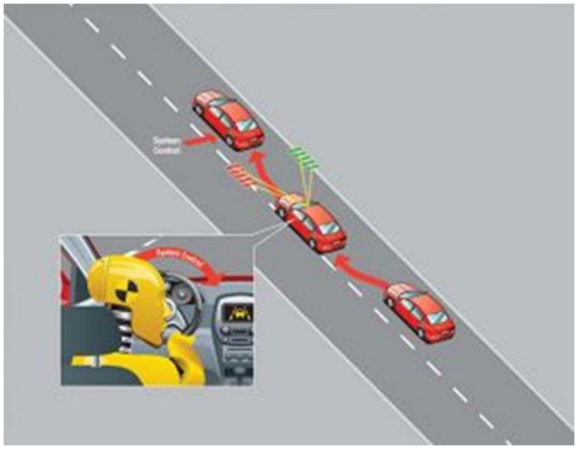
Αναγνώριση της απόστασης από το προπορευόμενο όχημα και διατήρηση της απόστασης αυτής με κατάλληλη επιτάχυνση ή επιβράδυνση του οχήματος. Ο οδηγός επίσης μπορεί να επιταχύνει και να επιβραδύνει ανά πάσα στιγμή όπως και με το standard Cruise Control. Επίσης υπάρχει και το πλέον εξελιγμένο ACC Stop και Go, το οποίο, αν το προπορευόμενο όχημα σταματήσει, έχει τη δυνατότητα να ακινητοποιήσει το όχημα και να το θέσει ξανά σε κίνηση (Stop και Go) για ταχύτητες κάτω από 30 Km/h.



Εικόνα 1.6 : Προσαρμοζόμενο σύστημα ελέγχου ταχύτητας **[11]**

- Σύστημα βοήθειας διατήρησης λωρίδας κυκλοφορίας (Lane Keeping Assistance)

Στην περίπτωση που το όχημα αρχίζει να παρασύρεται έξω από τη λωρίδα κυκλοφορίας χωρίς τη χρήση φλας, τότε εφαρμόζεται μια σύντομή διορθωτική δράση (διεύθυνσης ή πέδησης), μαζί με ένα προειδοποιητικό ήχο ταυτόχρονα.



Εικόνα 1.7 : Σύστημα βοήθειας διατήρησης λωρίδας κυκλοφορίας **[11]**

- Συστήματα ανίχνευσης υπνηλίας οδηγού (Driver drowsiness Monitoring Systems)

  Ένας κουρασμένος οδηγός ο οποίος παρασύρεται από το δρόμο, προειδοποιείται ηχητικά και οπτικά. Το σύστημα παρακολουθεί τις οδηγικές επιδόσεις του οδηγού και ειδοποιεί όταν εντοπίσει μια έλλειψη προσοχής ή υπνηλία.

- Ευφυής προσαρμογή ταχύτητας (Intelligent speed adaptation) :

  Αναγνώριση σημάνσεων ταχύτητας και προειδοποίηση του οδηγού όταν υπερβαίνει το εκάστοτε μέγιστο επιτρεπόμενο όριο ταχύτητας

- Αυτόματο παρκάρισμα (Automatic parking) :

  Αναγνώριση χώρου θέσεων στάθμευσης και αυτόματη πλοήγηση του αυτοκινήτου σε μια εξ' αυτών και στάθμευσή του

Τα περισσότερα συστήματα ADAS είναι καινούρια στην αγορά και συνεπώς είναι εγκατεστημένα σε μικρό ποσοστό των αυτοκινήτων που κυκλοφορούν στους δρόμους και για το λόγο αυτό δεν έχουν ακόμα διαδραματίσει σημαντικό ρόλο στην ασφάλεια, δηλαδή από στατιστική άποψη η συμβολή τους στη μείωση των τροχαίων ατυχημάτων εξαιτίας απόσπασης της προσοχής του οδηγού είναι επί του παρόντος μικρή.

Σύμφωνα με σχετική έρευνα **[13],** η ενσωμάτωση τέτοιων συστημάτων ξεκίνησε τα τελευταία πέντε (5) χρόνια και από το σύνολο των νέων επιβατηγών αυτοκινήτων που πουλήθηκαν στην Ευρώπη το 2ο τετράμηνο του έτους 2018 το 3% ήταν εφοδιασμένα με σύστημα ADAS επιπέδου αυτονομίας 2 (Level 2 autonomy – L2), δηλαδή με σύστημα υποβοήθησης του οδηγού, ενώ στο 2ο τετράμηνο του έτους 2019 το αντίστοιχο ποσοστό ανήλθε σε 8,8% (αύξηση 175%).

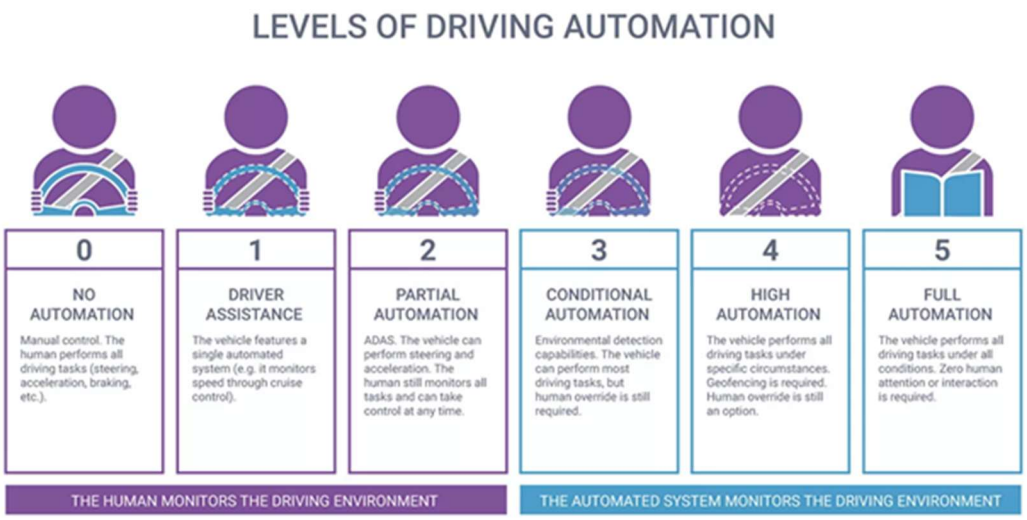
Επίσης σύμφωνα με αντίστοιχη έρευνα του έτους 2021 **[14]** το πλήθος των νέων επιβατηγών αυτοκινήτων με σύστημα ADAS L2 που πουλήθηκαν το έτος 2020 ήταν αυξημένο κατά 78% έναντι του αντίστοιχου πλήθους του προηγούμενου έτους.

Ειδικότερα στην Ευρώπη το έτος 2020 το 19% των νέων επιβατηγών αυτοκινήτων που πουλήθηκαν ήταν εφοδιασμένα με σύστημα ADAS L2.

Ωστόσο, ακόμα και αν όλες οι κατασκευάστριες εταιρείες οχημάτων ενσωματώσουν συστήματα ADAS σε όλα τα μοντέλα τους, μια ουσιαστική αντικατάσταση των παλαιών κυκλοφορούντων οχημάτων, που δεν διαθέτουν σύστημα ADAS, αναμένεται να διαρκέσει πολλά χρόνια.

Σύμφωνα με έρευνα **[12],** η μέση ηλικία των εν χρήσει Ι.Χ. οχημάτων στην Ευρωπαϊκή Ένωση ανέρχεται στα 12 έτη, με την Ελλάδα να βρίσκεται στην τελευταία θέση με μέση ηλικία τα 17 έτη. Από το σύνολο των Ι.Χ. οχημάτων, περίπου το 9% αφορούν σε οχήματα 5ετίας και νεότερα. Επομένως πρακτικά στους δρόμους συστήματα ADAS είναι εγκατεστημένα σε ποσοστό λιγότερο από το 9% των οχημάτων. Ευνόητο είναι ότι μια ουσιαστική αντικατάσταση των παλαιότερων οχημάτων με νέα, που θα διαθέτουν συστήματα ADAS, αναμένεται να διαρκέσει 5 ÷ 10 έτη.

Η Society of Automotive Engineers (SAE) ορίζει έξι (6) επίπεδα αυτοματισμού οδήγησης (Level of Driving Automation), που κυμαίνονται από 0 (πλήρως χειροκίνητο) έως 5 (πλήρως αυτόνομο), όπως φαίνεται στην Εικόνα 1.8 **[15]**.



Εικόνα 1.8 : Επίπεδα αυτοματισμού οδήγησης **[15]**

Ένα σύστημα ADAS, αναλόγως των δυνατοτήτων του, θα πρέπει να αναγνωρίζει αντικείμενα στο περιβάλλον του, όπως ιδίως:

- Λωρίδες κυκλοφορίας
- Οχήματα διαφόρων ειδών
- Ανθρώπους
- Σημάνσεις π.χ. τύπου STOP
- Φωτεινούς σηματοδότες

Οι λωρίδες κυκλοφορίας που η μορφή τους είναι απλή και δισδιάστατη και η θέση τους σταθερή μπορούν σχετικά εύκολα και γρήγορα να εκφραστούν μαθηματικά.

Τα υπόλοιπα όμως αντικείμενα έχουν πολύ περίπλοκη δομή και ως τρισδιάστατα αντικείμενα πολλές διαφορετικές οπτικές, οι δε άνθρωποι έχουν επιπλέον και μεταβαλλόμενη μορφή, οπότε η μαθηματική έκφρασή τους είναι αδύνατη.

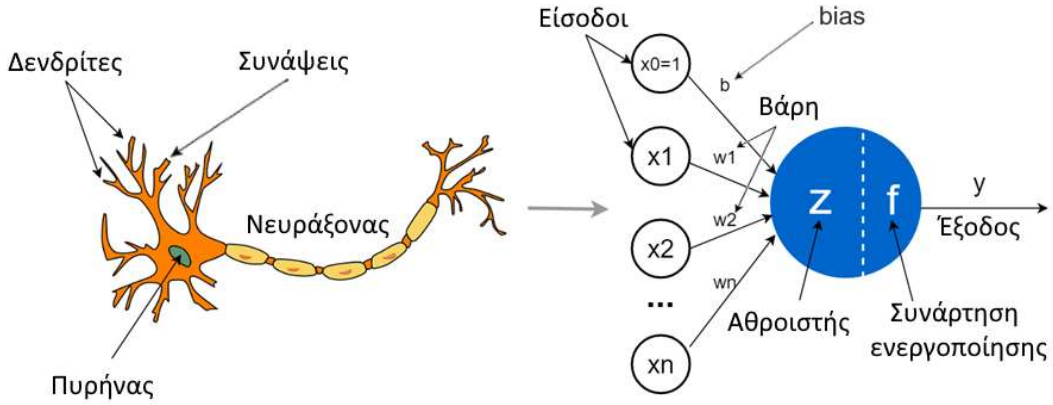
Για να εκπαιδευτεί ένας υπολογιστής στο τι είναι αυτά τα αντικείμενα και να τα αναγνωρίζει σε μια εικόνα χρησιμοποιείται Μηχανική Μάθηση (Machine Learning ή ML). Στο επόμενο κεφάλαιο θα δοθεί περιγραφή της μηχανικής μάθησης και των τεχνητών νευρωνικών δικτύων.

Κεφάλαιο 2

Τεχνητά Νευρωνικά Δίκτυα

2.1 Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα (Neural Networks) αποτελούνται από πολλούς τεχνητούς νευρώνες (artificial neuron), η δομή των οποίων είναι παρόμοια με αυτή των βιολογικών νευρώνων του ανθρώπινου εγκεφάλου. Κατ' αντιστοιχία με τον βιολογικό νευρώνα του ανθρώπινου εγκεφάλου, ένας τεχνητός νευρώνας είναι η θεμελιώδης δομική μονάδα του εκάστοτε τεχνητού νευρωνικού δικτύου.



Εικόνα 2.1: Αναλογία Βιολογικού (αριστερά) με Τεχνητό Νευρώνα (δεξιά) **[17]**

Ένας τεχνητός νευρώνας έχει πολλές **εισόδους** (inputs) **x0...xn**, (*στο βιολογικό νευρώνα λέγονται **δενδρίτες**)* που λαμβάνουν αριθμητικές τιμές. Οι είσοδοι μπορεί να είναι δεδομένα εισόδου ή έξοδοι άλλων νευρώνων Οι είσοδοι αυτές έχουν η κάθε μια ένα συγκεκριμένο **βάρος** (σημαντικότητα) **w0…wn**, (*στον βιολογικό νευρώνα αντιστοιχούν οι **συνάψεις**)* που λαμβάνουν αριθμητικές τιμές και ελέγχουν το επίπεδο σημαντικότητας της αντίστοιχης εισόδου (όσο υψηλότερη είναι η τιμή τόσο μεγαλύτερη η σημαντικότητα της εισόδου). Το άθροισμα των σταθμισμένων, αναλόγως του βάρους τους, εισόδων (**x0 * b + x1 * w1 + ... + xn * wn)** αφού περάσει από μια **συνάρτηση ενεργοποίησης** (activation function) **f**, (*στον βιολογικό νευρώνα αντιστοιχεί ο **πυρήνας**)* δίνει ένα αποτέλεσμα το οποίο αποτελεί την **έξοδο** (output) **y**, (*στον βιολογικό νευρώνα αντιστοιχεί ο **νευραξονας**),* που στέλνεται στον επόμενο νευρώνα ή νευρώνες.

Το **b (bias)** ενίοτε συμβολίζεται και ως βάρος **w0.** Ο λόγος που συμπεριλαμβάνεται το bias (πόλωση) στο άθροισμα των σταθμισμένων εισόδων είναι για να μετατοπιστεί η συνάρτηση ενεργοποίησης κάθε νευρώνα ώστε να μην ληφθεί μηδενική τιμή. Με άλλα

λόγια, εάν όλες οι είσοδοι **x1, x2, …, xn** είναι 0, το **z (αθροιστής)** είναι ίσο με την τιμή της πόλωσης. Η είσοδος **x0** παίρνει πάντα την τιμή 1.

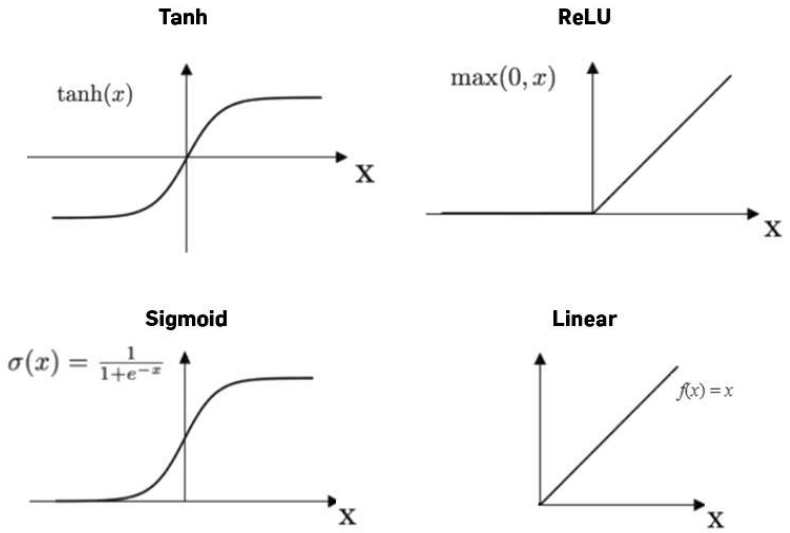
Γενικότερα η έξοδος του τεχνητού νευρώνα δίνεται από την συνάρτηση:

$$y = f[(x1 * w1 + ... + xn * wn) + b]$$

ή

$$y = f(x0 * w0 + x1 * w1 + ... + xn * wn)$$

Η συνάρτηση ενεργοποίησης **f** έχει πολλές μορφές, εκ των οποίων οι πλέον χρησιμοποιούμενες είναι οι εμφανιζόμενες στο Σχήμα 2.1:



Σχήμα 2.1: Μορφές Συνάρτησης Ενεργοποίησης

Ο άξονας Χ είναι η είσοδος (το άθροισμα των σταθμισμένων εισόδων) και ο άξονας Υ η έξοδος του τεχνητού νευρώνα.

Η ReLU, ειδικά από το 2018 και μετά, έχει καταστεί η πλέον χρησιμοποιούμενη συνάρτηση ενεργοποίησης.

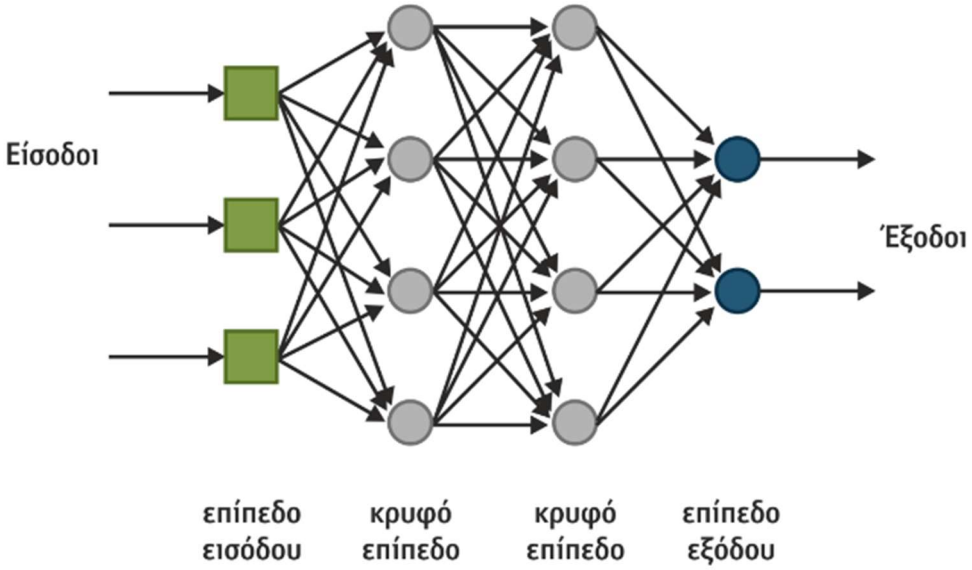
Τα Νευρωνικά δίκτυα είναι αποτέλεσμα συνένωσης πολλαπλών νευρώνων μεταξύ τους. Αναλόγως του πώς είναι συνδεδεμένοι οι νευρώνες μεταξύ τους, υπάρχουν δύο βασικές κατηγορίες νευρωνικών δικτύων: τα πρόσθιας τροφοδότησης (feed forward) και τα οπίσθιας τροφοδότησης (feed backward).

Στα Νευρωνικά δίκτυα πρόσθιας τροφοδότησης οι νευρώνες συνδέονται με τέτοιο τρόπο ώστε η έξοδος ενός προγενέστερου νευρώνα να είναι είσοδος για ένα μεταγενέστερο νευρώνα ή και νευρώνες σειριακά. Η δομή ενός νευρωνικού δικτύου διαφοροποιείται ανάλογα με τον σκοπό. Η πιο κλασική δομή έχει μια στήλη (επίπεδο ή layer) από νευρώνες εισόδου (Input Layer) που συνδέονται αναλόγως με μια άλλη στήλη κρυφών νευρώνων (Hidden Layer) που με την σειρά τους μπορεί να συνδέονται

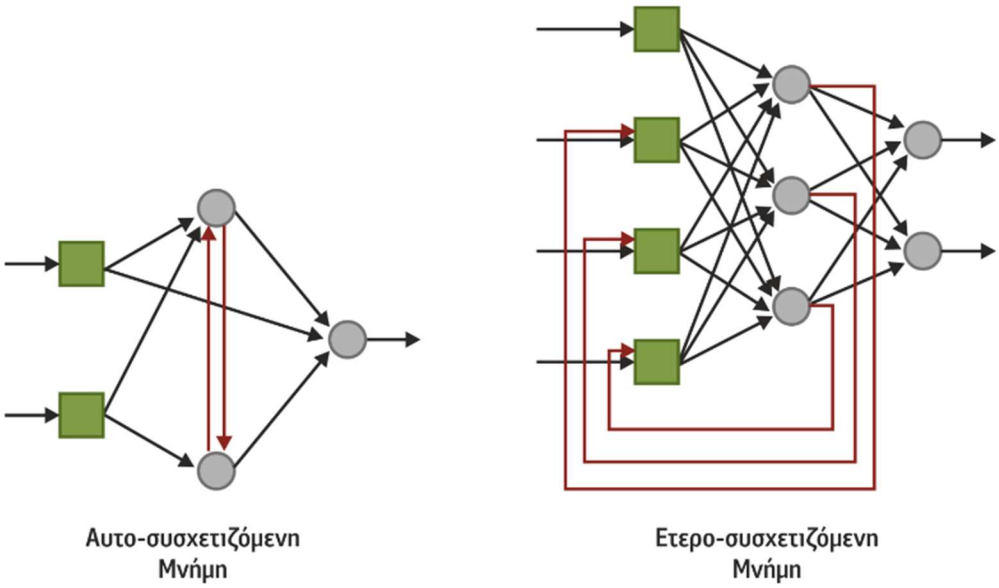
με περισσότερες κρυφές στήλες (Hidden Layers) μέχρις ότου το αποτέλεσμα φτάσει στην τελική στήλη εξόδου (Output Layer), όπου κάθε νευρώνας δηλώνει την κάθε πιθανή έξοδο και το ποσοστό ενεργοποίησης παρέχει την βεβαιότητα της επιλογής αυτής.

Στα Νευρωνικά δίκτυα οπίσθιας τροφοδότησης επιτρέπεται στους νευρώνες ενός επιπέδου να τροφοδοτούν και νευρώνες του ίδιου επιπέδου ή και προηγούμενων επιπέδων. Αν η τροφοδότηση αφορά κόμβους στο ίδιο επίπεδο, τότε τα δίκτυα καλούνται αυτοσυσχετιζόμενες μνήμες (autoassociated memories) διαφορετικά, καλούνται ετεροσυσχετιζόμενες μνήμες (heteroassociated memories).

Τα Νευρωνικά δίκτυα με πολλές στήλες (Layers) ονομάζονται βαθιά Νευρωνικά δίκτυα.



Εικόνα 2.2: Δομή απλού νευρωνικού δικτύου εμπρόσθιας τροφοδότησης **[16]**



Εικόνα 2.3: Δομή απλού νευρωνικού δικτύου οπίσθιας τροφοδότησης **[16]**

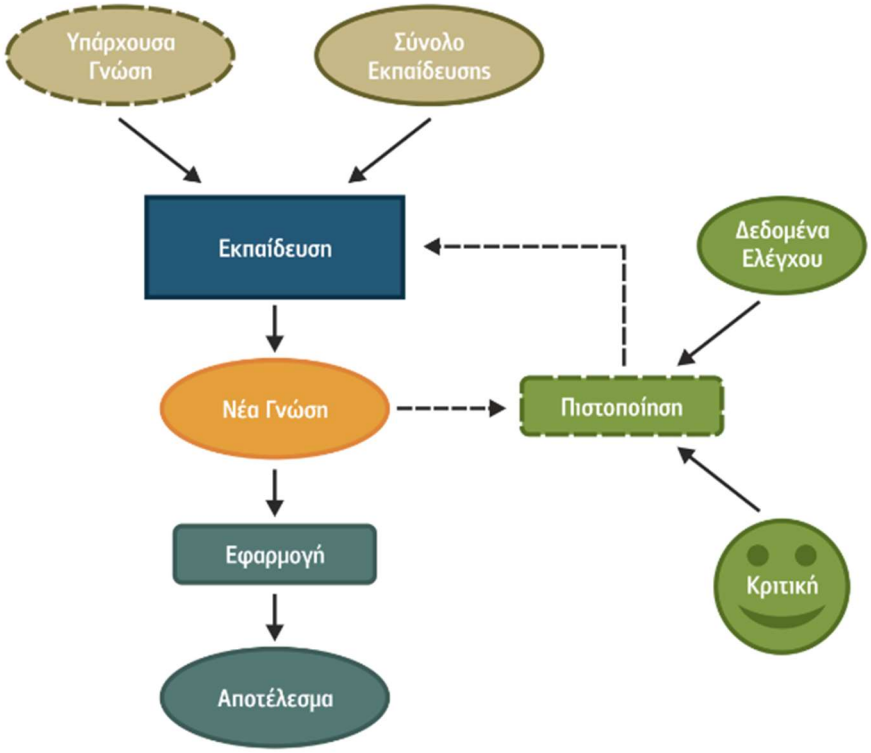
2.2 Μηχανική Μάθηση - Γενική περιγραφή

Η Μηχανική Μάθηση είναι ένα υποπεδίο της Τεχνητής Νοημοσύνης (ΤΝ). Έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν δηλαδή την απόδοσή τους σε κάποιους τομείς, μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας. Ασχολείται με τη μελέτη και δημιουργία αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους. Μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της Μηχανικής Μάθησης συγκαταλέγονται οι: Εξόρυξη Δεδομένων, Πιθανότητες και Στατιστική, Θεωρία της Πληροφορίας, Αριθμητική Βελτιστοποίηση, Θεωρία της Πολυπλοκότητας, Θεωρία Ελέγχου (προσαρμοστική), Ψυχολογία (εξελικτική, γνωστική), Νευροβιολογία και Γλωσσολογία **[16]**.

Οι εφαρμογές της Μηχανικής Μάθησης καλύπτουν πάρα πολλούς τομείς, όπως ιδίως, φυσική επεξεργασία γλώσσας (Natural Language Processing-NLP), αναγνώριση ομιλίας (speech recognition), ανάλυση συναισθήματος (sentiment analysis), αυτόματη μετάφραση (Machine translation), αναγνώριση αντικειμένων (object detection), ταξινόμηση εικόνων (image classification), ιατρική υπολογιστική όραση (διάγνωση σε ακτινογραφίες, αξονικές ή μαγνητικές τομογραφίες) και συνεχώς επεκτείνονται.

Η Μηχανική Μάθηση διακρίνεται σε τρεις κύριες κατηγορίες την επιτηρούμενη μάθηση (Supervised Learning), τη μη επιβλεπόμενη μάθηση (unsupervised learning) και την ενισχυτική μάθηση (reinforcement learning). Ειδικότερα στην επιβλεπόμενη μάθηση, που θα χρησιμοποιήσουμε στην παρούσα εργασία, ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους-δείγματα σε γνωστές-επιθυμητές εξόδους (σύνολο εκπαίδευσης), κάνοντας προβλέψεις και διορθώνοντας τις προβλέψεις σε περίπτωση λάθους με στόχο τη γενίκευση της συνάρτησης αυτής για εισόδους με άγνωστη έξοδο (σύνολο ελέγχου). Χρησιμοποιείται σε προβλήματα: Ταξινόμησης (Classification), Πρόγνωσης (Prediction) και Διερμηνείας (Interpretation) **[16]**.

Στο Σχήμα 2.4, αποτυπώνεται ο γενικός τρόπος λειτουργίας των αλγορίθμων Μηχανικής Μάθησης. Η βασικότερη φάση κάθε αλγόριθμου είναι η εκπαίδευση, όπου ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα σύνολο δεδομένων εκπαίδευσης (training set) προς επίτευξη του σκοπού του, τη δημιουργία νέας γνώσης. Επιπλέον, μπορεί είτε να χρησιμοποιήσει λιγότερο ή περισσότερο την υπάρχουσα γνώση είτε να μην τη χρησιμοποιήσει καθόλου. Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης της παραγόμενης νέας γνώσης. Συνήθως, η πιστοποίηση πραγματοποιείται καταρχάς από τον ίδιο τον αλγόριθμο μέσω διαδικασιών ανάκλησης (recall) με τη βοήθεια δεδομένων ελέγχου (test data) και, στη συνέχεια, μέσω κριτικής που κάνει ο χρήστης βάσει των γνώσεων που διαθέτει για το πρόβλημα που επιχειρεί να λύσει ο αλγόριθμος. Τέλος, η νέα γνώση δίνεται προς χρήση σε εφαρμογές στις οποίες είναι απαραίτητη, για να λυθούν πραγματικά προβλήματα **[16]**.



Σχήμα 2.4: Φάσεις Μηχανικής Μάθησης **[16]**

Η εκπαίδευση των νευρωνικών δικτύων έχει ως βασικό στόχο να βρεθεί ένας τρόπος αλλαγής των βαρών που θα έχει ως αποτέλεσμα την αλλαγή της γενικής συμπεριφοράς του δικτύου με την αύξηση της ικανότητάς του να παρέχει στο μέλλον μία επιθυμητή έξοδο μετά από μία δεδομένη είσοδο. Όταν η επιθυμητή έξοδος είναι εκ των προτέρων γνωστή λέμε ότι το δίκτυο μαθαίνει με επίβλεψη (supervised learning), αλλιώς μαθαίνει χωρίς επίβλεψη (unsupervised learning). Βασικό στοιχείο της αρχιτεκτονικής ενός νευρωνικού δικτύου είναι ο τρόπος ελέγχου της αλλαγής των βαρών κατά την εκπαίδευση, δηλαδή ο αλγόριθμος εκπαίδευσης (training algorithm) που υλοποιείται αποκλειστικά από το ίδιο το δίκτυο χωρίς εξωτερική επέμβαση. Γνωστότεροι αλγόριθμοι εκπαίδευσης είναι οι ακόλουθοι:

- Αλγόριθμος οπισθοδιάδοσης λάθους (backpropagation),
- Ανταγωνιστική μάθηση (competitive learning),
- Τυχαία μάθηση (random learning).

Κάθε αλγόριθμος εκπαίδευσης χρησιμοποιεί κάποιον κανόνα εκμάθησης, για να προσαρμόσει τα βάρη μεταξύ των νευρώνων του .

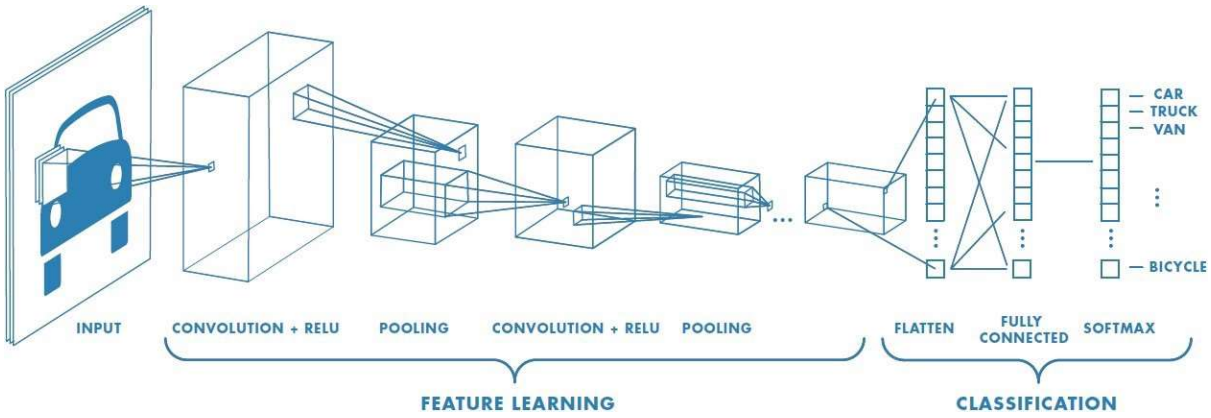
Οι αλγόριθμοι εκπαίδευσης χαρακτηρίζονται από τους κανόνες εκμάθησης (learning rules) που χρησιμοποιούνται, για να υπολογίσουν τα σφάλματα και να διορθώσουν τα βάρη των εσωτερικών νευρώνων του δικτύου. Σημαντική παράμετρο των κανόνων εκμάθησης αποτελεί ο ρυθμός εκμάθησης, που καθορίζει το πόσο γρήγορα συγκλίνει η μάθηση. Μεγάλος ρυθμός εκμάθησης μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση και σε ταλάντωση γύρω από τις βέλτιστες τιμών βαρών. Μικρός ρυθμός εκμάθησης

έχει ως αποτέλεσμα πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσει σε παγίδευση σε τοπικά ακρότατα **[16]**.

Ο υπολογισμός του σφάλματος μπορεί να γίνει με πολλές συναρτήσεις και η επιλογή της σωστής συνάρτησης σφάλματος (loss function) είναι αντικείμενο της μελέτης βελτιστοποίησης του αλγορίθμου. Η επανάληψη και ανανέωση των βαρών γίνεται αρκετές φορές μέχρι την μη περαιτέρω ελαχιστοποίηση του σφάλματος και τότε το μοντέλο θεωρείται ότι έχει εκπαιδευτεί και είναι έτοιμο για να προβεί σε προβλέψεις σε δεδομένα που δεν έχει ξαναδεί.

2.3 Συνελικτικά Νευρωνικά Δίκτυα

Τα δίκτυα αυτά είναι ένα είδος πολυεπίπεδων νευρωνικών δικτύων, ιδανικά για αναγνώριση αντικειμένων σε εικόνες (ταξινόμηση εικόνων, ανίχνευση αντικειμένων, ανίχνευση προσώπου, αναγνώριση οχήματος, αναγνώριση κειμένου κ.λπ.) γιατί εξάγουν τα χαρακτηριστικά τους ακόμα και όταν το αντικείμενο αυτό έχει υποστεί παραμόρφωση οποιουδήποτε είδους (Translation Invariance). Επιπρόσθετα πλεονεκτήματά τους είναι το μικρότερο πλήθος παραμέτρων (βάρη φίλτρων), η γρηγορότερη εκτέλεση του αλγορίθμου και οι μικρότερες απαιτήσεις μνήμης.



Εικόνα 2.5: Δομή συνελικτικού νευρωνικού δικτύου **[18]**

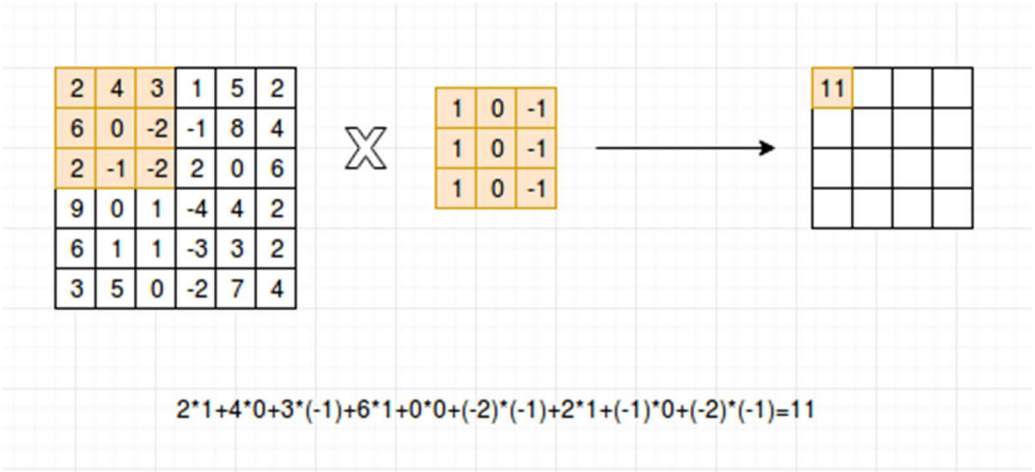
Ένα Συνελικτικό επίπεδο (convolutional layer) είναι ουσιαστικά ένα σύνολο από νευρώνες που εκτελούν συνέλιξη των φίλτρων, που έχουν προκαθοριστεί, με την εικόνα που δέχονται στην είσοδο. Στην αναγνώριση εικόνων η είσοδος είναι μία εικόνα τριών διαστάσεων (ύψος, πλάτος, βάθος), όπου το βάθος είναι τα τρία βασικά χρώματα ή με άλλα λόγια τα κανάλια (κόκκινο, πράσινο, μπλε).

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN) χρησιμοποιούν κυρίως τα εξής βήματα:

2.3.1 Επίπεδο συνέλιξης (Convolution Layer)

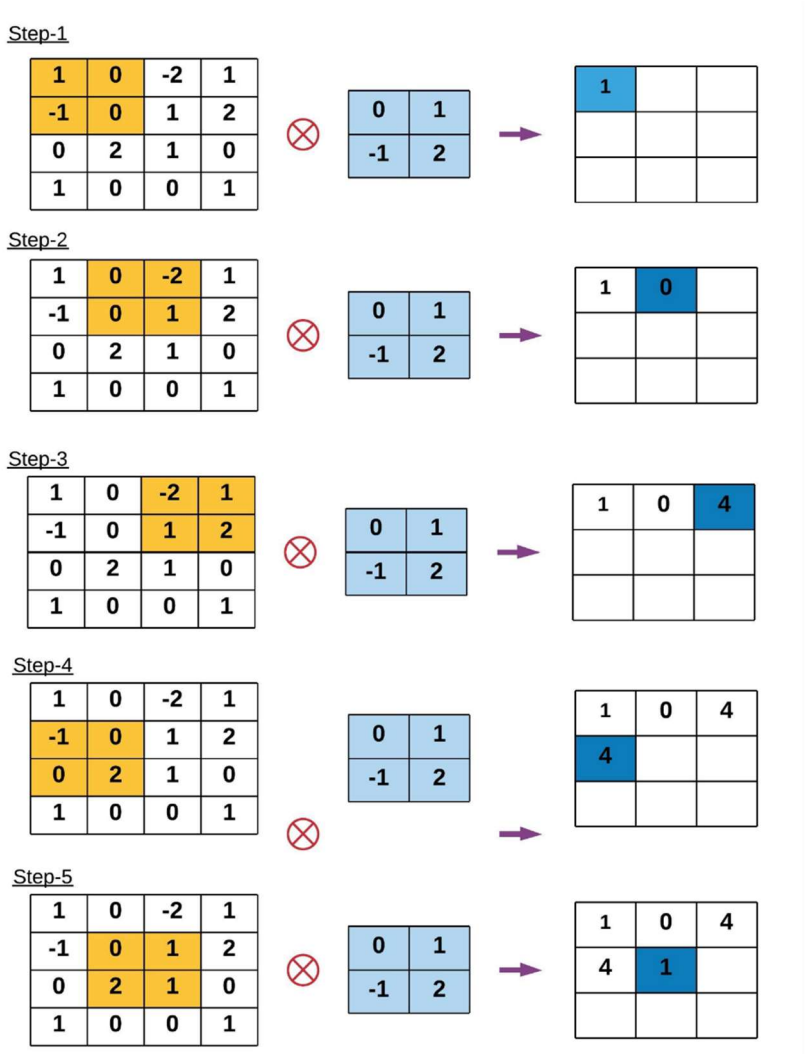
Ας αγνοήσουμε προς το παρόν τα κανάλια και ας δούμε πως λειτουργεί η συνέλιξη με δισδιάστατα δεδομένα. Στο Σχήμα 2.6 η είσοδος είναι μια δισδιάστατη εικόνα με ύψος 6 και πλάτος 6 ή 6x6. Το φίλτρο έχει διαστάσεις ύψος 3 και πλάτος 3 ή 3x3. Το σχήμα του παραθύρου της συνέλιξης δίνεται από τις διαστάσεις (ύψος και πλάτος) του φίλτρου, δηλαδή είναι 3x3. Τα σκιασμένα τμήματα στο Σχήμα 2.6 είναι το πρώτο στοιχείο εξόδου (11) καθώς και τα αντίστοιχα στοιχεία της εισόδου και του φίλτρου που χρησιμοποιούνται για τον υπολογισμό της συνέλιξης. Στην ουσία από τον πίνακα εισόδου 6x6 χρησιμοποιείται ένα τμήμα διαστάσεων 3x3, ίδιων δηλαδή διαστάσεων με αυτές του φίλτρου, και εκτελείται καταρχήν πολλαπλασιασμός των τιμών των κελιών του τμήματος της εικόνας εισόδου με τις αντίστοιχες τιμές των κελιών του φίλτρου και ακολούθως άθροιση των μερικών γινομένων. Αυτή η σειρά πράξεων ονομάζεται **συνέλιξη**. Το αποτέλεσμα θα είναι στην περίπτωσή μας μια τιμή (11) η οποία τίθεται στην πρώτη θέση του πίνακα εξόδου (πίνακα χαρακτηριστικών ή αποτελέσματος).

Στη δισδιάστατη λειτουργία της συνέλιξης, το παράθυρο συνέλιξης (ίδιων διαστάσεων με το φίλτρο) τοποθετείται αρχικά στην επάνω αριστερή γωνία του πίνακα εισόδου. Στη συνέχεια ολισθαίνει διαδοχικά κατά μήκος του πίνακα εισόδου, από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Πιο συγκεκριμένα στο παράδειγμα της εικόνας 2.7 στην επόμενη φάση θα μετακινήσουμε προς τα δεξιά τη θέση του τμήματος εισόδου 3x3 που παίρνουμε για την συνέλιξη κατά ένα πλήθος κελιών το οποίο το ονομάζουμε **βήμα (Stride).** Στην περίπτωση του Σχήματος 2.6 το βήμα είναι 1. Έτσι το δεύτερο στοιχείο του πίνακα εξόδου (δεξιά του 11) θα είναι ίσο με 4*1+3*0+1*(-1)+0*1+(-2)*0+(-1)*(-1)+(-1)*1+(-2)*0+2*(-1) = 1. Ο πίνακας εξόδου (αποτελέσματος) θα έχει μέγεθος $\frac{μεγεθους\ εισοδου - μεγεθους\ φιλτρου}{Βημα} + 1$ και συγκεκριμενα στην περιπτωση μας $\frac{6-3}{1} + 1 = 4$



Σχήμα 2.6: Παράδειγμα συνέλιξης εισόδου 6x6 με φίλτρο 3x3

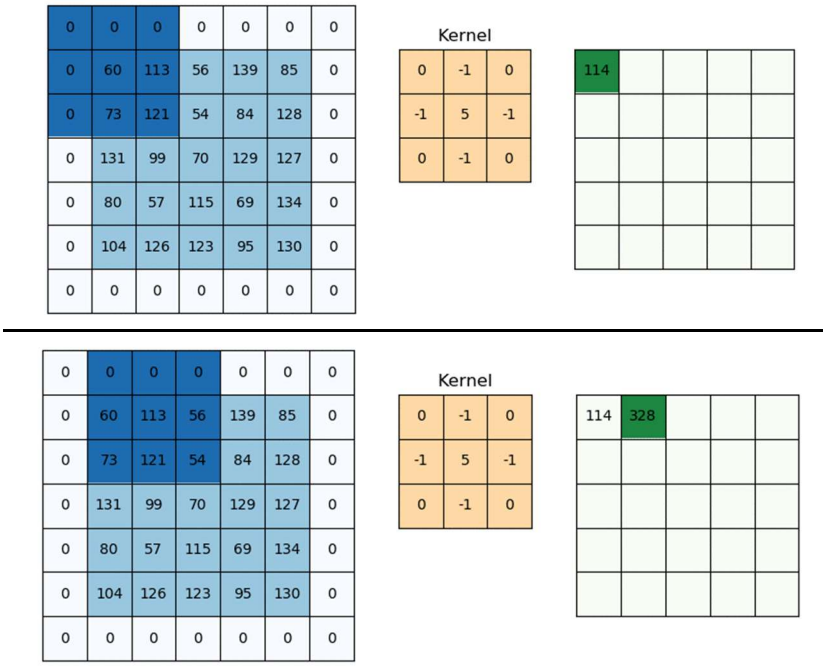
Στο Σχήμα 2.7 εμφανίζονται τα πρώτα πέντε (5) βήματα της συνέλιξης εισόδου 4x4 με φίλτρο 2x2 και βήμα (stride) ίσο με 1. Ο πίνακας εξόδου θα είναι 3x3.



Σχήμα 2.7: Τα πέντε πρώτα βήματα συνέλιξης εισόδου 4x4 με φίλτρο 2x2 **[19]**

Η συνέλιξη δίνει χρήσιμα αποτελέσματα όταν το κέντρο του φίλτρου περνάει από όλα τα κελιά της εισόδου. Στις περιπτώσεις που εξετάσαμε παραπάνω το κέντρο του φίλτρου δεν πέρασε από τα κελιά που ήταν στις άκρες του πίνακα εισόδου με αποτέλεσμα η έξοδος να έχει μειωμένες διαστάσεις, δηλαδή χάσαμε πληροφορία. Πράγματι στο Σχήμα 2.7 η αρχική εικόνα έχει διάσταση 4x4, το φίλτρο 2x2 και ο πίνακας της συνέλιξης 3x3 (με βήμα=1).

Ένας τρόπος να διατηρήσουμε την πληροφορία περνώντας από όλα τα κελιά της εισόδου είναι να επεκτείνουμε τον πίνακα εισόδου περιμετρικά με κελιά σταθερής τιμής, συνήθως 0 (Σχήμα 2.8). Αυτή η πράξη ονομάζεται **γέμισμα** (**padding).** Στην περίπτωση του Σχήματος 2.8 η αρχική εικόνα έχει διάσταση 5x5 και το φίλτρο έχει διάσταση 3x3 οπότε ο πίνακας της συνέλιξης, χωρίς γέμισμα, θα έχει διάσταση 3x3 με βήμα=1. Αν θέλουμε ο πίνακας της συνέλιξης να έχει ίδιες διαστάσεις με τον αρχική είσοδο, δηλαδή 5x5, θα πρέπει να ορίσουμε γέμισμα p=1 έτσι ώστε ο αρχικός πίνακας να γίνει 7x7.



Σχήμα 2.8: Παράδειγμα padding εισόδου σε 2 φάσεις με βήμα (stride) ίσο με 1

Αν n είναι η διάσταση του αρχικού πίνακα, f είναι η διάσταση του φίλτρου, s είναι το βήμα και p είναι το γέμισμα ο τύπος που δίνει τις διαστάσεις της συνέλιξης είναι ο παρακάτω:

$$\frac{n + 2p - f}{s} + 1$$

Εάν ο πίνακας εισόδου έχει διαστάσεις n_1, n_2 οι τύποι που δίνουν τις διαστάσεις της συνέλιξης είναι:

$$\frac{n1+2p-f}{s} + 1 \quad \text{και} \quad \frac{n2+2p-f}{s} + 1$$

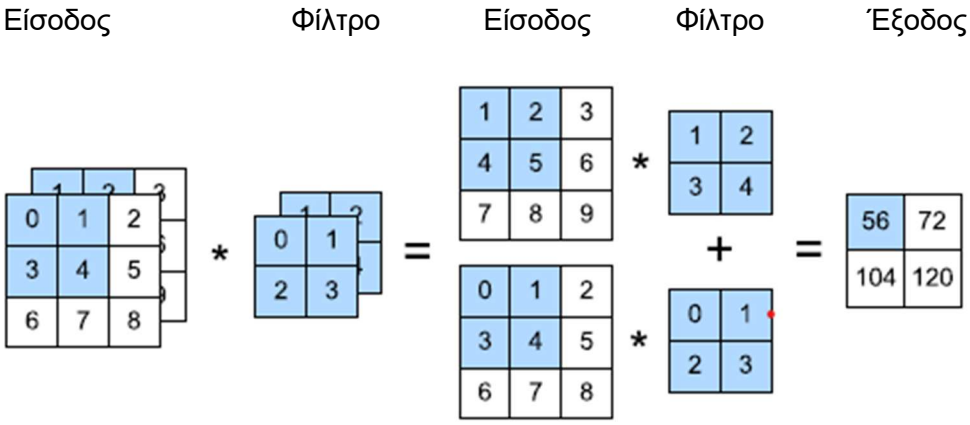
Η διαδικασία της συνέλιξης είναι παρόμοια και για περισσότερες από δύο διαστάσεις. Όταν τα δεδομένα εισόδου περιέχουν πολλά κανάλια (π.χ. K), πρέπει να κατασκευάσουμε ένα φίλτρο συνέλιξης με τον ίδιο αριθμό καναλιών με τα δεδομένα εισόδου (δηλαδή ένα φίλτρο διαστάσεων KxKxK), ώστε να μπορεί να πραγματοποιεί συνέλιξη με τα δεδομένα εισόδου.

Όπως αναφέραμε στην περίπτωση της συνέλιξης δύο διαστάσεων (σχήματα 2.6 και 2.7), το σχήμα του παραθύρου της συνέλιξης θα προσδιορίζεται από τις διαστάσεις του φίλτρου, δηλαδή, στην περίπτωση K καναλιών, θα πρέπει να είναι KxKxK.

Παίρνουμε λοιπόν από την εικόνα εισόδου ένα τμήμα διαστάσεων KxKxK (K ύψος, K πλάτος, K βάθος) πολλαπλασιάζονται τα στοιχεία του πρώτου καναλιού του φίλτρου με τα αντίστοιχα στοιχεία του παραθύρου της συνέλιξης του πρώτου καναλιού της εισόδου, τα στοιχεία του δεύτερου καναλιού του φίλτρου με τα αντίστοιχα στοιχεία του παραθύρου της συνέλιξης του δεύτερου καναλιού της εισόδου κ.ο.κ. και, μετά τον

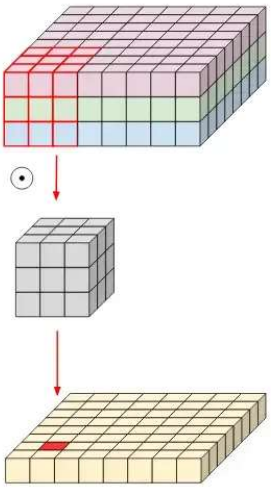
πολλαπλασιασμό των στοιχείων του K-στού καναλιού του φίλτρου με τα αντίστοιχα στοιχεία του παραθύρου της συνέλιξης του K-στού καναλιού της εισόδου, στη συνέχεια αθροίζονται όλα τα γινόμενα και προκύπτει το πρώτο στοιχείο του πίνακα εξόδου της συνέλιξης. Στη συνέχεια το φίλτρο μετακινείται, κατά τα γνωστά (από δεξιά προς τα αριστερά και από πάνω προς τα κάτω), στην επόμενη θέση και η διαδικασία συνεχίζεται έως ότου το φίλτρο «σαρώσει» όλη την εικόνα. Μετά τη συνέλιξη ο πίνακας που θα προκύψει στην έξοδο θα είναι δύο διαστάσεων.

Στο Σχήμα 2.9 παρέχεται ένα παράδειγμα συνέλιξης με 2 κανάλια εισόδου και φίλτρο διαστάσεων 2x2x2. Τα σκιασμένα τμήματα είναι το πρώτο στοιχείο εξόδου (56), τα κανάλια του φίλτρου και τα πρώτα παράθυρα της συνέλιξης των 2 καναλιών της εισόδου που χρησιμοποιούνται για τον υπολογισμό του πρώτου στοιχείου της εξόδου: (1x1+2x2+4x3+5x4)+(0x0+1x1 + 3x2 + 4x3) = 56.



Σχήμα 2.9: Συνέλιξη με 2 κανάλια εισόδου **[19]**

Στο παράδειγμα του Σχήματος 2.10 ως είσοδο έχουμε μια εικόνα τριών διαστάσεων, 8 ύψος, 8 πλάτος και 3 βάθος, όπου το βάθος στην περίπτωση μας είναι τα τρία βασικά χρώματα (κόκκινο, πράσινο, μπλε). Το αποτέλεσμα της συνέλιξής της με φίλτρο διαστάσεων 3x3x3 θα είναι ένας πίνακας διαστάσεων 8x8x1.

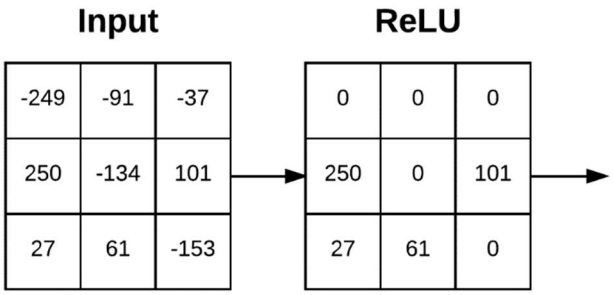


Σχήμα 2.10: Παράδειγμα συνέλιξης εισόδου 8x8x3 με φίλτρο 3x3x3 σε 8x8x1

Στην ουσία σε κάθε βήμα της διαδικασίας το φίλτρο KxKxK (3x3x3, στο παράδειγμα του σχήματος 2.10) μετακινείται πάνω στην είσοδο MxMxK (8x8x3, στο παράδειγμα του σχήματος 2.10) σε 2 διαστάσεις, και όλα τα στοιχεία του πολλαπλασιάζονται με τα αντίστοιχα στοιχεία της εισόδου. Συνεπώς KxKxK ή K^3 (3x3x3 ή 27, στο παράδειγμα του σχήματος 2.10) πολλαπλασιασμοί προστίθενται στο στοιχείο εξόδου.

Η παραπάνω διαδικασία γίνεται ξεχωριστά για ένα η περισσότερα φίλτρα. Μερικά μοντέλα έχουν 32÷1024 φίλτρα για κάθε κανάλι. Το αποτέλεσμα της συνέλιξης θα είναι ένας πίνακας βάθους όσα και τα φίλτρα που θα συνελίξουμε.

Στη συνέχεια σε όλα τα στοιχεία του πίνακα που προκύπτει από την συνέλιξη εφαρμόζεται μία συνάρτηση ενεργοποίησης. Συνήθως εφαρμόζεται η μη-γραμμική συνάρτηση ReLU (Σχήμα 2.11).

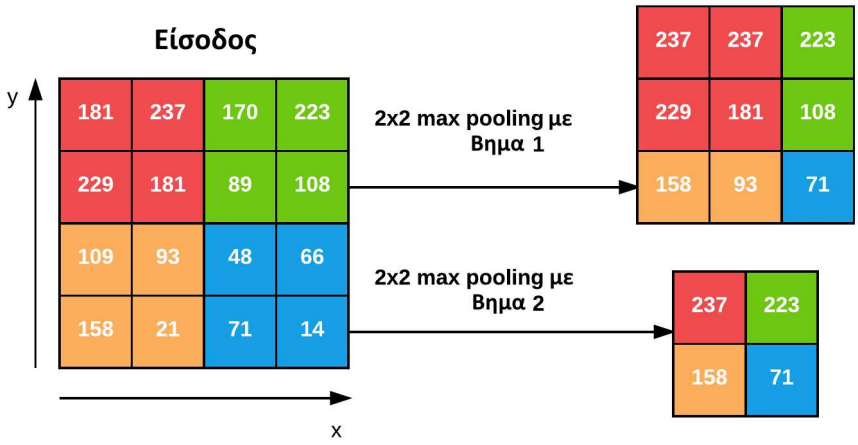


Σχήμα 2.11: Παράδειγμα εφαρμογής συνάρτησης ενεργοποίησης ReLU

2.3.2 Επίπεδο συγκέντρωσης (Pooling Layer)

Στο επίπεδο αυτό παίρνουμε την έξοδο της συνέλιξης και την περνάμε ως είσοδο στο επίπεδο συγκέντρωσης (Pooling layer). Ουσιαστικά εκτελούμε υποδειγματοληψία των χαρακτηριστικών της εισόδου, αφού η ανάλυση του παραγόμενου πίνακα χαρακτηριστικών μειώνεται. Σκοπός της συγκέντρωσης είναι η μείωση των διαστάσεων, ύψους και πλάτους, χωρίς να χάσουμε πολλή πληροφορία με αποτέλεσμα μείωση των παραμέτρων και πράξεων με όφελος τη μείωση του υπολογιστικού κόστους κατά την εκπαίδευση και εκτέλεση του δικτύου. Επιπλέον είναι και ένας αποδοτικός τρόπος για να μειώσουμε την υπερκάλυψη (overfit) στα δεδομένα μας, η οποία όταν έχει υψηλή τιμή σημαίνει ότι το μοντέλο έχει δώσει πολύ μεγάλη προσοχή στα δεδομένα εκπαίδευσης αλλά δεν μπορεί να γενικεύσει και να αποδώσει εξίσου καλά στα δεδομένα επαλήθευσης γιατί έχει απορροφήσει και το θόρυβο από τα δεδομένα εκπαίδευσης.

Όπως και στη συνέλιξη μπορούμε να χρησιμοποιήσουμε βήμα 1 ή 2 ανάλογα με τις διαστάσεις του πίνακα εξόδου που θέλουμε (Σχήμα 2.12)

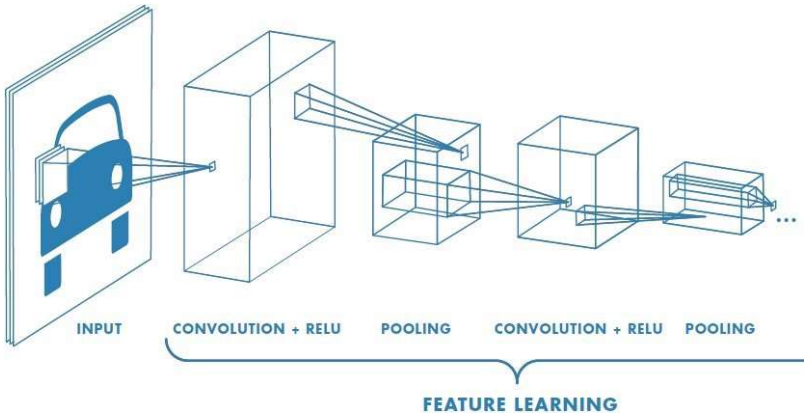


Σχήμα 2.12: Παράδειγμα συγκέντρωσης με βήμα 1 και βήμα 2

Η χρήση του επιπέδου συγκέντρωσης (Pooling) έχει αρχίσει να μειώνεται με τα περισσότερα νέα μοντέλα να επιλέγουν να αντικαταστήσουν το επίπεδο συγκέντρωσης - υποδειγματοληψίας με ένα επίπεδο συνέλιξης μεγαλύτερου βήματος (stride). Έχει παρατηρηθεί ότι η μέθοδος αυτή δίνει καλύτερα αποτελέσματα από το απλό Pooling.

2.3.3 Συνδυασμός Επίπεδων - Πλήρως Συνδεδεμένο Επίπεδο (FC)

Ο συνδυασμός των επίπεδων **συνέλιξης > συνάρτησης ενεργοποίησης > συγκέντρωσης** αποτελούν το **βασικό δομικό στοιχείο** των Συνελικτικών Νευρωνικών Δικτυών.

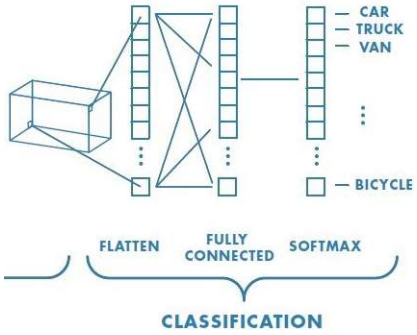


Σχήμα 2.13 : Βασικά δομικά στοιχεία Συνελικτικών Νευρωνικών Δικτυών (μάθηση χαρακτηριστικών) **[18]**

Τα περισσότερα δίκτυα έχουν πολλά τέτοια δομικά στοιχεία στη σειρά που αποτελούν την μάθηση χαρακτηριστικών (Σχήμα 2.13).

Το τελικό τους αποτέλεσμα είναι ένα επίπεδο μικρών διαστάσεων αλλά μεγάλου βάθους το οποίο το μετατρέπουμε σε μια μονοδιάστατη σειρά την οποία συνδέουμε πλήρως με το επόμενο επίπεδο **Πλήρως Συνδεδεμένο Επίπεδο** (Fully Connected Layer - FC) (Σχήμα 2.14). Το FC μετατρέπει τον πίνακα σε διάνυσμα (vector) με

πλήθος ψηφίων όσες είναι και οι πιθανές έξοδοι (αναγνωριζόμενα αντικείμενα), εφαρμόζοντας και μία συνάρτηση ενεργοποίησης (συνήθως την ReLU) και τέλος με την εφαρμογή μιας συνάρτησης softmax εξάγονται οι πιθανότητες της κάθε κατηγορίας αντικειμένου.



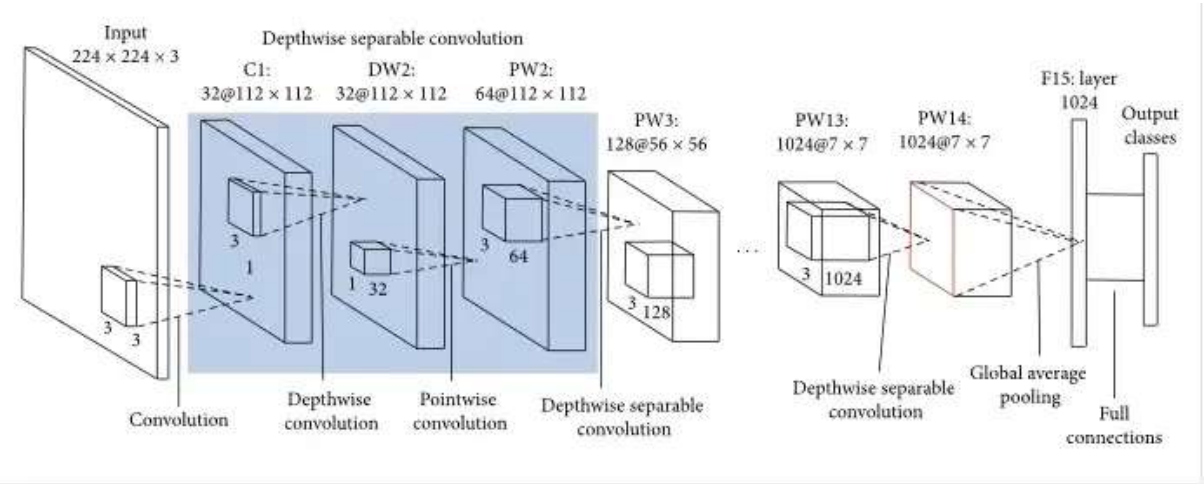
Σχήμα 2.14 : Ταξινόμηση αποτελεσμάτων δικτύου **[18]**

2.4 Συνελικτικό Νευρωνικό Δίκτυο MobileNet

Το μοντέλο **MobileNet** είναι ένα Συνελικτικό Νευρωνικό Δίκτυο ειδικά σχεδιασμένο για να έχει λίγες παραμέτρους και πράξεις ώστε να μπορεί να εκτελείται σε κινητά τηλέφωνα και χαμηλής ισχύος υπολογιστές, όπως το Raspberry Pi, αλλά να διατηρεί μια αρκετά καλή ακρίβεια. Για το μοντέλο αυτό έχουν αναπτυχθεί τρεις (3) εκδόσεις (versions).

2.4.1 Δομή του μοντέλου MobileNet v1 [21]

Η δομή του μοντέλου MobileNet v1 είναι η εξής:



Σχήμα 2.15: Δομή του μοντέλου MobileNet v1

Όπως φαίνεται στον Πίνακα 2.1, το μοντέλο δέχεται μια εικόνα διαστάσεων 224x224x3, εκτελεί μια απλή συνέλιξη με 32 φίλτρα και στη συνέχεια εκτελεί 13 επίπεδα από Διαχωρίσιμες ανά κανάλι Συνελίξεις (Depthwise Separable Convolutions) με πλήθος φίλτρων που διπλασιάζονται σε κάθε επίπεδο. Όταν το βάθος φτάσει 1024 με τη χρήση μέσης συγκέντρωσης (Avg Pooling) ενώνει ένα Πλήρως Συνδεδεμένο Επίπεδο (FC) που μας δίνει τις κλάσεις των αντικειμένων.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Πίνακας 2.1: Αρχιτεκτονική του μοντέλου MobileNet v1

Το πλέον σημαντικό χαρακτηριστικό του μοντέλου αυτού, με το οποίο μειώνεται το κόστος εκτέλεσης, είναι η χρήση των Διαχωρισμένων ανά κανάλι Συνελίξεων (Depthwise Separable Convolutions) που περιγράφονται στη συνέχεια.

Οι Διαχωρίσιμες ανά κανάλι Συνελίξεις αποτελούν εξέλιξη της κλασικής συνέλιξης της παραγράφου 2.3.1. Αποτελούνται από τα εξής δύο στάδια:

(α) Τη Συνέλιξη ανά κανάλι (Depthwise Convolution) και

(β) τη Συνέλιξη ανά σημείο (Pointwise Convolution).

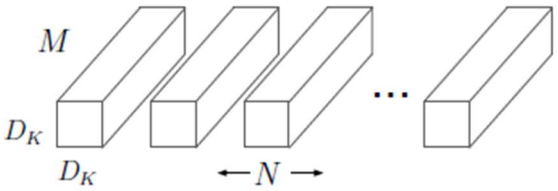
Η συνέλιξη ανά κανάλι είναι μια παραλλαγή της κλασικής συνέλιξης. Στην κλασική συνέλιξη που εκτελείται σε πολλαπλά κανάλια εισόδου, το φίλτρο είναι τόσο βαθύ όσο η είσοδος (Σχήματα 2.9 και 2.10). Στη συνέλιξη ανά κανάλι κάθε κανάλι διατηρείται ξεχωριστά - εξ ου και το όνομα ανά κανάλι. Η συνέλιξη αυτή εφαρμόζει ένα φίλτρο βάθους 1 σε κάθε κανάλι της εισόδου.

Στη συνέχεια η συνέλιξη ανά σημείο εφαρμόζει μια κλασσική συνέλιξη 1x1 για να συνδυάσει τις εξόδους της συνέλιξης ανά κανάλι.

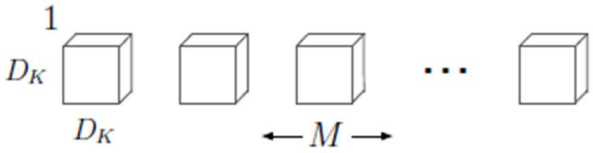
Η κλασική συνέλιξη, ταυτόχρονα σε ένα στάδιο, φιλτράρει και συνδυάζει τις εισόδους σε ένα νέο σετ εξόδων. Η διαχωρίσιμη ανά κανάλι συνέλιξη διαχωρίζει την κλασική συνέλιξη σε δύο επίπεδα, ένα πρώτο επίπεδο για το φιλτράρισμα των εισόδων και ένα δεύτερο για το συνδυασμό των εξόδων του πρώτου επιπέδου. Ο παραπάνω διαχωρισμός παρουσιάζει σημαντικά οφέλη διότι μειώνει δραστικά τόσο το πλήθος των παραμέτρων όσο και το υπολογιστικό κόστος σε σχέση με την κλασσική συνέλιξη, εκτός εάν η έξοδος έχει πολύ μικρό πλήθος καναλιών.

Στη συνέχεια δίνεται η θεωρητική ερμηνεία της διαχωρίσιμης ανά κανάλι συνέλιξης και αμέσως μετά ένα παράδειγμα για υποβοήθηση της κατανόησής της.

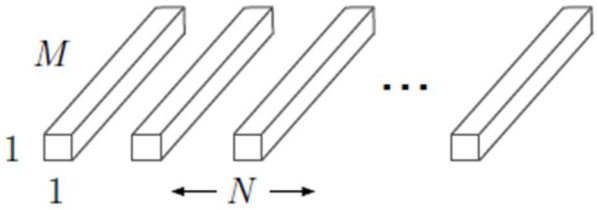
Στο Σχήμα 2.16 αναλύεται το πως η κλασική συνέλιξη [(σχήμα 2.16(α)] διαχωρίζεται στην συνέλιξη ανά κανάλι [σχήμα 2.16(β)] και στη συνέλιξη ανά σημείο [σχήμα 2.16(γ)].



(α) Κλασσική συνέλιξη



(β) Συνέλιξη ανά κανάλι



(γ) Συνέλιξη ανά σημείο

Σχήμα 2.16: Τα φίλτρα της κλασικής συνέλιξης στην εικόνα (α) αντικαθίστανται από δύο επίπεδα: Συνέλιξη ανά κανάλι στην εικόνα (β) και Συνέλιξη ανά σημείο στην εικόνα (γ) για να σχηματίσουν τη διαχωρίσιμη ανά κανάλι συνέλιξη

Για επεξήγηση του Σχήματος 2.16, ας υποθέσουμε ότι η κλασική συνέλιξη δέχεται είσοδο D_f x D_f x M, όπου D_f η διάσταση ύψους και πλάτους της εισόδου και M το πλήθος των καναλιών εισόδου και παράγει έξοδο D'_f x D'_f x N, όπου D'_f η διάσταση ύψους και πλάτους της εξόδου και N το πλήθος των καναλιών εξόδου (για απλούστευση των υπολογισμών θεωρούμε ίσες τις διαστάσεις ύψους και πλάτους εισόδου και εξόδου).

Στην κλασική συνέλιξη γίνεται χρήση φίλτρου διαστάσεων D_k x D_k x M x N, όπου D_k η διάσταση ύψους και πλάτους του φίλτρου, M το πλήθος των καναλιών του φίλτρου (ίσο με το πλήθος των καναλιών της εισόδου) και N το πλήθος των φίλτρων (όσο και το πλήθος των καναλιών εξόδου).

Το πλήθος των παραμέτρων της κλασικής συνέλιξης (standard convolution) P_{std} θα είναι :

$P_{std} = D_k \times D_k \times M \times N$

και το υπολογιστικό κόστος της C_{std} :

$C_{std} = D_k \times D_k \times M \times N \times D_f \times D_f$

Το πλήθος των παραμέτρων της διαχωρίσιμης ανά κανάλι συνέλιξης P_{dsc} θα είναι ίσο με το άθροισμα των παραμέτρων της συνέλιξης ανά κανάλι ($D_k \times D_k \times 1 \times M$) και της συνέλιξης ανά σημείο ($1 \times 1 \times M \times N$), δηλαδή :

$P_{dsc} = D_k \times D_k \times 1 \times M + 1 \times 1 \times M \times N$

και το υπολογιστικό κόστος της C_{dsc} θα είναι ίσο με το άθροισμα του κόστους της συνέλιξης ανά κανάλι ($D_k \times D_k \times M \times D_f \times D_f$) και της συνέλιξης ανά σημείο ($D'_f \times D'_f \times M \times N$), δηλαδή :

$C_{dsc} = D_k \times D_k \times M \times D_f \times D_f + D'_f \times D'_f \times M \times N$

Βάσει των προαναφερόμενων

$$\frac{P_{dsc}}{P_{std}} = \frac{D_k \times D_k \times 1 \times M \times 1 \times 1 \times M \times N}{D_k \times D_k \times M \times N \times D_f \times D_F} = \frac{1}{N} + \frac{1}{D_k^2}$$

και

$$\frac{C_{dsc}}{C_{std}} = \frac{D_k \times D_k \times M \times D_f \times D_F + D'_f \times D'_f \times M \times N}{D_k \times D_k \times M \times N \times D_f \times D_f} = \frac{1}{N} + \frac{D_f'^2}{D_k^2 \times D_f^2}$$

Εάν $D'_f = D_f$ θα είναι :

$$\frac{C_{dsc}}{C_{std}} = \frac{1}{N} + \frac{1}{D_k^2}$$

Για την πληρέστερη κατανόηση των προαναφερόμενων για τη Διαχωρίσιμη ανά κανάλι Συνέλιξη και του οφέλους της σε παραμέτρους και υπολογιστικό κόστος, δίνεται το ακόλουθο παράδειγμα, που αναλύεται στο Σχήμα 2.17:

Ας υποθέσουμε ότι $D_f = 8$, $M = 3$, $D_k = 3$, $N = 3$, όπου D_f η διάσταση ύψους και πλάτους της εισόδου, M το πλήθος των καναλιών εισόδου, D_k η διάσταση ύψους και πλάτους του φίλτρου και N το πλήθος των καναλιών εξόδου (ίσο με το πλήθος των φίλτρων). Στο παράδειγμα υποθέτουμε ότι το βήμα s=1 και ότι με χρήση γεμίσματος (padding) οι διαστάσεις (ύψους και πλάτους) εισόδου και εξόδου είναι ίσες.

Στο επίπεδο της Συνέλιξης ανά κανάλι χρησιμοποιούμε το ίδιο, βάθους 1, φίλτρο και για τα τρία κανάλια ξεχωριστά. Το πλήθος των παραμέτρων της Συνέλιξης ανά κανάλι θα είναι $D_k \times D_k \times 1 \times M = 3 \times 3 \times 1 \times 3 = 27$. Το υπολογιστικό κόστος θα είναι $D_k \times D_k \times M \times D_f \times D_f = 3 \times 3 \times 3 \times 8 \times 8 = 1728$. Στο επίπεδο της Συνέλιξης ανά σημείο το πλήθος των παραμέτρων θα είναι $1 \times 1 \times M \times N = 1 \times 1 \times 3 \times 3 = 9$ και το υπολογιστικό κόστος $D_f \times D_f \times M \times N = 8 \times 8 \times 3 \times 3 = 576$. Συνεπώς συνολικά η Διαχωρίσιμη ανά κανάλι

Συνέλιξη θα έχει 36 (=27+9) παραμέτρους και υπολογιστικό κόστος 2304 (=1728+576).

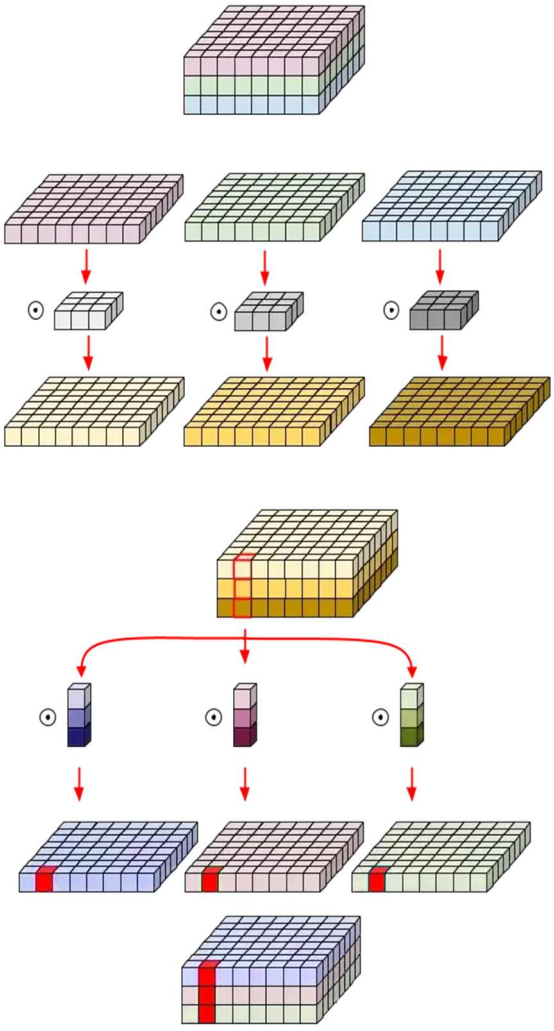
Η αντίστοιχη κλασική συνέλιξη θα είχε 81 (= 3 x 3 x 3 x 3) παραμέτρους και κόστος 5184 (= 3 x 3 x 3 x 8 x 8 x 3).

Παρατηρούμε ότι $\frac{36}{81} = \frac{2304}{5184} = 0{,}44 = \frac{1}{3} + \frac{1}{3^2} = \frac{1}{N} + \frac{1}{D_k^2}$, όπως αναμενόταν

Σημειώνεται ότι τα οφέλη της Διαχωρίσιμης ανά κανάλι Συνέλιξης σε μείωση παραμέτρων και υπολογιστικού κόστους είναι ακόμα μεγαλύτερα για μεγάλο πλήθος εξόδων.

Πράγματι εάν D_f = 128, M = 3, D_k = 3, N = 16, η Διαχωρίσιμη ανά κανάλι Συνέλιξη θα έχει 3x3x3+3x16 = 75 παραμέτρους και υπολογιστικό κόστος 3x3x3x128x128+128x128x3x16 = 1.228.800 ενώ η κλασική συνέλιξη 3x3x3x16 = 432 παραμέτρους και υπολογιστικό κόστος 3x3x3x128x128x16 = 7.077.888.

Παρατηρούμε και πάλι ότι $\frac{75}{432} = \frac{1.228.800}{7.077.888} = 0{,}1736 = \frac{1}{16} + \frac{1}{3^2} = \frac{1}{N} + \frac{1}{D_k^2}$



Σχήμα 2.17: Απεικόνιση διαχωρίσιμης ανά κανάλι συνέλιξης, με είσοδο 8x8x3, φίλτρο 3x3x3 και έξοδο 8x8x3

2.4.2 Διαφοροποιήσεις στο μοντέλο MobileNet v2 [22]

Η βασική ιδέα για την ανάπτυξη του MobileNet v2 ήταν η περαιτέρω, έναντι του MobileNet v1, μείωση των χρησιμοποιούμενων παραμέτρων διατηρώντας την ίδια ικανότητα αναγνώρισης. Ειδικότερα στο MobileNet v2 επιτυγχάνεται μείωση των παραμέτρων ανά επίπεδο μέσω της συμπίεσης των επίπεδων βάθους και της διατήρησης μη γραμμικότητας. Αναλυτικότερα αυτό επιτυγχάνεται με τις παρακάτω προσθήκες:

- **Επίπεδο συμφόρησης με επέκταση (bottleneck with expansion layer)**

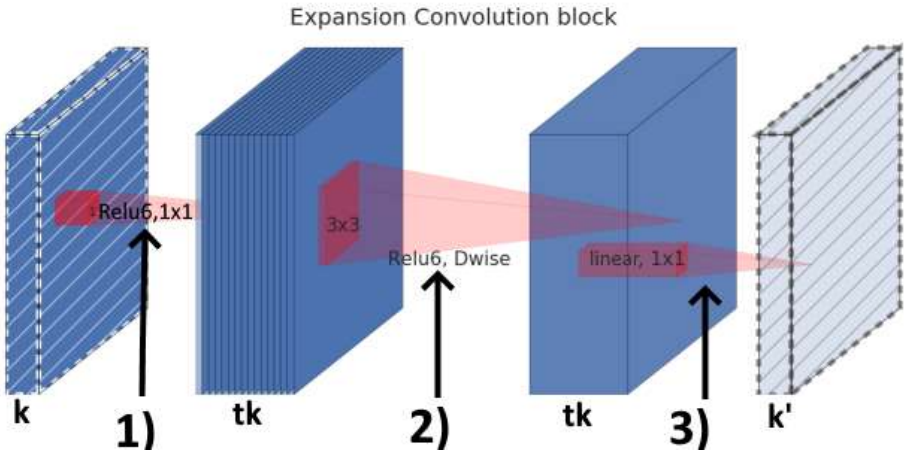
  Στο επίπεδο αυτό η βασική ιδέα είναι να γίνει συμπίεση χαρακτηριστικών πολλών επίπεδων σε λιγότερα. Αυτό επιτυγχάνεται με τη συμφόρηση με επέκταση, η οποία συνοπτικά περιγράφεται στη συνέχεια.

  Έστω ότι έχουμε μια είσοδο διαστάσεων **h × w × k,** όπου **h** είναι το ύψος, **w** είναι το πλάτος και **k** είναι το βάθος της. Η συμφόρηση με επέκταση αποτελείται από τα εξής βήματα (Πίνακας 2.2 και Σχήμα 2.18):

  1) Στο πρώτο βήμα εκτελούμε **Συνέλιξη ανά σημείο** με t×k φίλτρα διαστάσεων 1 × 1 × k (όπου **t** είναι ο παράγοντας εσωτερικής επέκτασης). Το αποτέλεσμα είναι μια ομάδα από t×k κανάλια βάθους 1, τα οποία και ενεργοποιούμε με τη συνάρτηση ReLU6.
  2) Στο δεύτερο βήμα παίρνουμε την ομάδα των t×k καναλιών βάθους 1 και εκτελούμε **Συνέλιξη ανά κανάλι** με ένα φίλτρο 3 × 3 × 1 και βήμα **s**. Το αποτέλεσμα είναι ένα επίπεδο διαστάσεων h/s × w/s × tk, το οποίο και ενεργοποιούμε με τη συνάρτηση ReLU6.
  3) Στο τρίτο και τελευταίο βήμα εκτελούμε πάλι **Συνέλιξη ανά σημείο** με k΄ φίλτρα (όπου k΄ είναι το τελικό συμπιεσμένο βάθος που θέλουμε) διαστάσεων 1 × 1 × tk. Το αποτέλεσμα είναι μια ομάδα από k΄ κανάλια βάθους 1 τα οποία δεν τα ενεργοποιούμε.

Input	Operator	Output
1) $h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
2) $h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
3) $\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Πίνακας 2.2 : Απεικόνιση βημάτων συμφόρησης με επέκταση

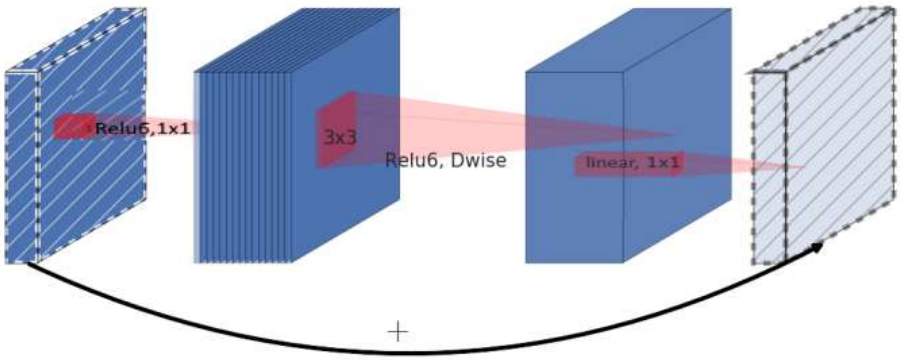


Σχήμα 2.18: Γραφική απεικόνιση βημάτων συμφόρησης με επέκταση

- **Ανεστραμμένο μπλοκ με κατάλοιπο (Inverted residuals)**

  Μια ακόμη ιδέα που ανακαλύφθηκε από άλλους ερευνητές είναι το μπλοκ με κατάλοιπο το οποίο προστέθηκε και στο μοντέλο αυτό.

  Όταν το μοντέλο γίνει πολύ βαθύ, δηλαδή έχει πολλά επίπεδα, παρατηρείται ότι η ακρίβεια του αρχίζει να μειώνεται. Είναι δηλαδή κάτι σαν «χαλασμένο τηλέφωνο» και όσο πιο πολλά είναι τα επίπεδα τόσο πιο πολύ αποκλίνει από το πραγματικό στόχο. Η λύση που επινοήθηκε είναι να αντικαταστήσουν τα περισσότερα μπλοκ με μπλοκ στα οποία η αρχική είσοδος προστίθεται στο αποτέλεσμά τους, ώστε αυτή να μην εξαφανιστεί πλήρως με το βάθος (Σχήμα 2.19).



Σχήμα 2.19 : Γραφική απεικόνιση συμφόρησης με επέκταση και κατάλοιπο

Τελική δομή του μοντέλου MobileNet v2

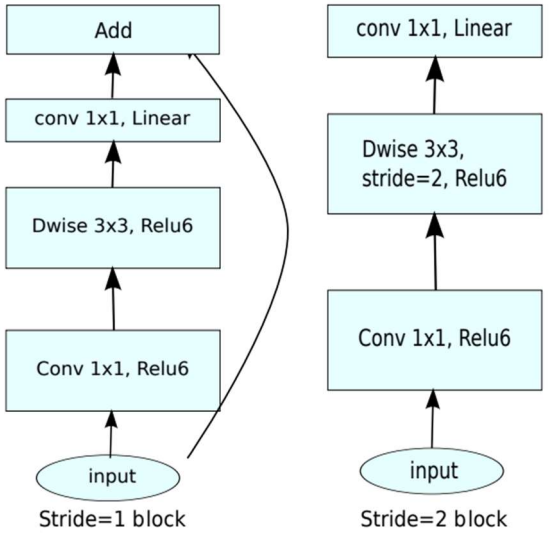
Η τελική δομή του MobileNet v2 είναι μια ανάμιξη των παραπάνω βελτιώσεων με την έκδοση MobileNet v1. Αναλυτικότερα τα βήματα είναι τα εξής (Πίνακας 2.3):

Input	Operator	t	k´	n	s	Output
$224^2 \times 3$	conv2d	-	32	1	2	$112^2 \times 32$
$112^2 \times 32$	bottleneck	1	16	1	1	$112^2 \times 16$
$112^2 \times 16$	bottleneck	6	24	2	2	$56^2 \times 24$
$56^2 \times 24$	bottleneck	6	32	3	2	$28^2 \times 32$
$28^2 \times 32$	bottleneck	6	64	4	2	$14^2 \times 64$
$14^2 \times 64$	bottleneck	6	96	3	1	$14^2 \times 96$
$14^2 \times 96$	bottleneck	6	160	3	2	$7^2 \times 160$
$7^2 \times 160$	bottleneck	6	320	1	1	$7^2 \times 320$
$7^2 \times 320$	conv2d 1x1	-	1280	1	1	$7^2 \times 1280$
$7^2 \times 1280$	avgpool 7x7	-	-	1	-	$1 \times 1 \times 1280$
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-		

Πίνακας 2.3 : Δομή του μοντέλου MobileNet v2

όπου:

- **t** είναι ο παράγοντας εσωτερικής επέκτασης
- **k´** είναι το τελικό συμπιεσμένο βάθος που θέλουμε
- **s** είναι το βήμα της συνέλιξης ανά κανάλι
- **n** είναι οι φορές που εκτελείται συμφόρηση σε κάθε γραμμή



Σχήμα 2.20 : Γραφική απεικόνιση των δυο ειδών συμφόρησης με επέκταση

Υπάρχουν 2 είδη συμφόρησης με επέκταση (Σχήμα 2.20), ένα με κατάλοιπο και ένα χωρίς, Τη συμφόρηση με επέκταση χωρίς κατάλοιπο (στα δεξιά του σχήματος 2.20) την χρησιμοποιούμε όταν θέλουμε να μειώσουμε τις X, Y διαστάσεις ενώ διαφορετικά χρησιμοποιούμε τη συμφόρηση με επέκταση και κατάλοιπο (στα αριστερά του σχήματος 2.20). Πιο συγκεκριμένα όταν το n (οι φορές που εκτελείται συμφόρηση με επέκταση σε κάθε γραμμή) είναι μεγαλύτερο του 1, τότε την πρώτη φορά θα χρησιμοποιήσουμε την χωρίς κατάλοιπο συμφόρηση με επέκταση μειώνοντας διαστάσεις, και τις άλλες φορές την με κατάλοιπο συμφόρηση με επέκταση διατηρώντας τις διαστάσεις ίδιες.

Για παράδειγμα ας πάρουμε την γραμμή 3 του Πίνακα 2.3:

Input	Operator	t	k´	n	s	Output
$112^2 \times 16$	bottleneck	6	24	2	2	$56^2 \times 24$

αφού n=2, η συμφόρηση με επέκταση θα εκτελεστεί 2 φορές, όπως εμφανίζεται παρακάτω:

Input	Operator	t	k´	s	Output
$112^2 \times 16$	bottleneck	6	24	2	$56^2 \times 24$
$56^2 \times 24$	bottleneck + res	6	24	1	$56^2 \times 24$

Στην πρώτη γραμμή εκτελείται συμφόρηση με επέκταση χωρίς κατάλοιπο στην είσοδο (112 × 112 × 16) με βήμα 2 (**s**), δηλαδή στο 2ο βήμα της συμφόρησης με επέκταση (Πίνακας 2.2, Σχήμα 2.18) εκτελείται συνέλιξη ανά κανάλι με βήμα 2, οπότε το αποτέλεσμα θα έχει διαστάσεις X, Y ίσες με το μισό της εισόδου (56 × 56). Το βάθος εξόδου θα πάρει την τιμή **k´** που θέλουμε δηλαδή 24.
Στην δεύτερη γραμμή (και σε όσες γραμμές και αν υπήρχαν, αναλόγως της τιμής του n) θα εκτελέσουμε συμφόρηση με επέκταση με κατάλοιπο στην είσοδο (56 × 56 × 24) με βήμα 1 (s), δηλαδή στο 2ο βήμα της συμφόρησης με επέκταση (Πίνακας 2.2, Σχήμα 2.18) εκτελείται συνέλιξη ανά κανάλι με βήμα 1, οπότε το αποτέλεσμα θα έχει διαστάσεις X, Y ίσες με αυτές της εισόδου (56 × 56). Το βάθος εξόδου **k´** θα παραμείνει ίδιο, δηλαδή 24.

Σύγκριση έκδοσης MobileNet v2 με την έκδοση MobileNet v1

Με τις παραπάνω αλλαγές η ακρίβεια της έκδοσης v2 αυξήθηκε κατά 1.4% (=72%-70.6%) και ταυτόχρονα το κόστος μειώθηκε κατά 33.6%{=(113-75) / 113}, έναντι της έκδοσης v1.

Network	Top 1	CPU
MobileNetV1	70.6	113ms
MobileNetV2	**72.0**	**75ms**

Πίνακας 2.4: Πίνακας σύγκρισης των εκδόσεων v1 και v2. Στη στήλη Top-1 δίνεται η ακρίβεια των μοντέλων ως ποσοστό και στη στήλη CPU ο χρόνος εκτέλεσης σε ms

2.4.3 Διαφοροποιήσεις στο μοντέλο MobileNet v3 [23]

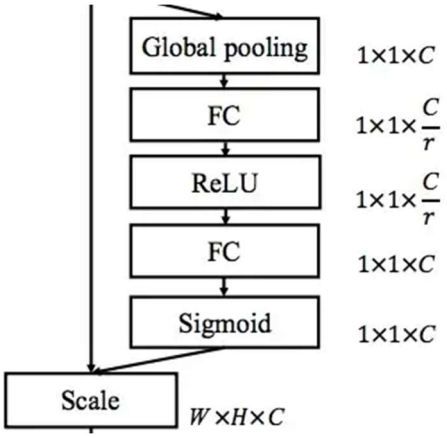
- **Συμπίεση-και-διέγερση (squeeze-and-excite)**

  Όταν κάνουμε συνέλιξη με πολλά φίλτρα κάποια από αυτά θα εξάγουν πιο χρήσιμο χάρτη χαρακτηριστικών από την είσοδο σε σχέση με άλλα. Η ιδέα είναι

λοιπόν να τονίσουμε αυτά τα φίλτρα. Αυτό γίνεται πρακτικά με την προσθήκη παραμέτρου σε κάθε φίλτρο ώστε κατά την εκπαίδευση του δικτύου να τονίσουμε τα πιο χρήσιμα φίλτρα.

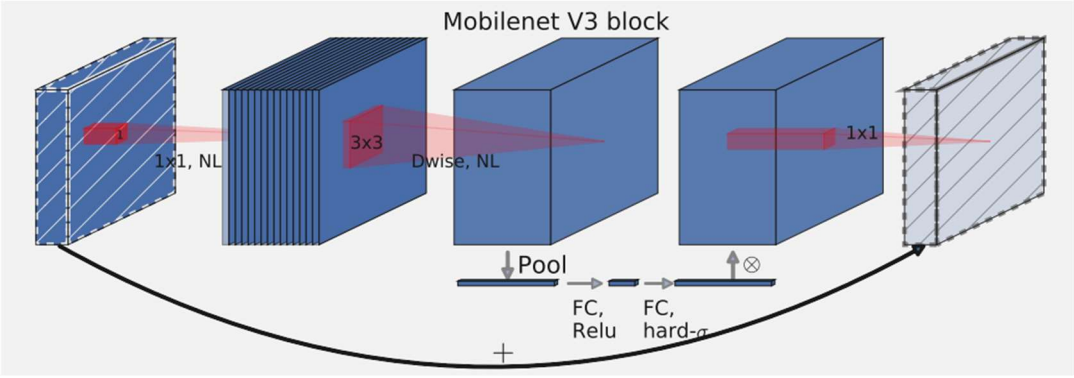
Η δομή της συμπίεσης-και-διέγερσης μετά από συνέλιξη είναι η εξής:

- **Ολική συγκέντρωση (global pooling)** των αποτελεσμάτων της συνέλιξης με κάθε φίλτρο σε ένα πίνακα διαστάσεων 1x1xC, όπου C το πλήθος των φίλτρων που χρησιμοποιήθηκαν
- Ένα **πλήρως συνδεδεμένο επίπεδο (FC),** το οποίο ενεργοποιείται μη γραμμικά με χρήση ReLU (μπορεί να γίνει μείωση του βάθους και πολυπλοκότητας με τη χρήση μιας μεταβλητής **r**).
- Ένα ακόμα **πλήρως συνδεδεμένο επίπεδο (FC)**
- Μια ενεργοποίηση με χρήση **Σιγμοειδούς καμπύλης,** για ποιο ομαλό διαχωρισμό μεταξύ των βαρών
- Και τέλος πολλαπλασιασμός του κάθε παραπάνω βάρους με κάθε χάρτη χαρακτηριστικών που είχε δώσει η συνέλιξη



Σχήμα 2.21 : Γραφική απεικόνιση συμπίεσης-και-διέγερσης

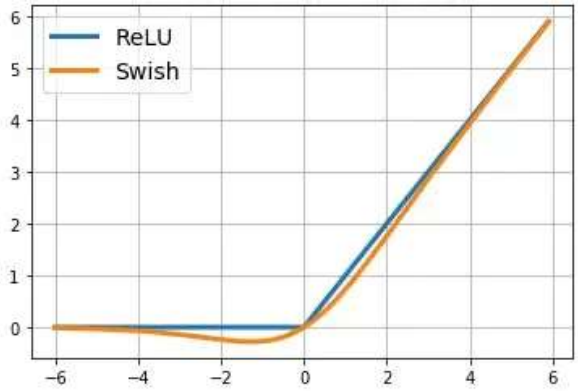
Στην περίπτωση του MobileNet v3 προστέθηκε ένα τέτοιο βήμα μέσα στην συμφόρηση με κατάλοιπο και πιο συγκεκριμένα μετά την συνέλιξη ανά κανάλι (Σχήμα 2.22).



Σχήμα 2.22 : Γραφική απεικόνιση συμφόρησης με κατάλοιπο και συμπίεση-και-διέγερση

- **Συνάρτηση μη γραμμικής ενεργοποίησης H-Swish**

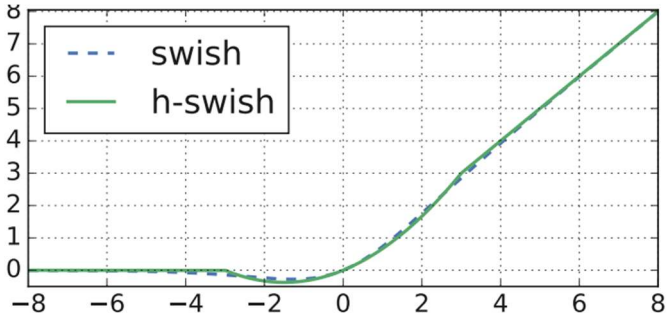
  Έχει αποδειχθεί από έρευνα [24] ότι η χρήση της μη γραμμικής συνάρτησης ενεργοποίησης Swish (Σχήμα 2.23), αντί της ReLU, προσφέρει καλύτερη ακρίβεια.



Σχήμα 2.23 : Γραφική απεικόνιση μη γραμμικής συνάρτησης ενεργοποίησης Swish και ReLU

Αλλά η χρήση της αντί της ReLU έχει μια μη μηδενική αύξηση στο κόστος εκτέλεσης του μοντέλου. Γι' αυτό δημιουργήθηκε μια νέα συνάρτηση η h-Swish (hard swish) (εξίσωση 1), η οποία διατηρεί το σχήμα σχεδόν ίδιο με την Swish (Σχήμα 2.24) και έχοντας την ReLU μέσα της ο υπολογισμός της είναι πολύ πιο γρήγορος.

$$hswish[x] = x\frac{ReLU6(x+3)}{6} \quad (1)$$



Σχήμα 2.24 : Γραφική απεικόνιση μη γραμμικής συνάρτησης ενεργοποίησης Swish και H-Swish

- **Επιπρόσθετες αλλαγές στην δομή και χρήση "Αναζήτηση Δομής Δικτύου"**

  Πέραν των προαναφερθέντων έγιναν και αρκετές επιπρόσθετες αλλαγές στη δομή του μοντέλου MobileNet v3, εκ των οποίων οι σημαντικότερες είναι:

  - Το μέγεθος των φίλτρων (σε μερικά επίπεδα χρησιμοποιήθηκε 5x5 μέγεθος)
  - Η επιλογή συνάρτησης ενεργοποίησης ανά επίπεδο (ReLU ή h-Swish)

- Η πρώτη συνέλιξη γίνεται με 16 μόνο φίλτρα
- Η χρήση συμπίεσης-και-διέγερσης ανά επίπεδο

Για περαιτέρω βελτιστοποίηση του μοντέλου MobileNet v3 χρησιμοποιήθηκε η τεχνική «**Αναζήτηση Δομής Δικτύου» ή Neural architecture search (NAS).** Η τεχνική αυτή ουσιαστικά αυτοματοποιεί τη σχεδίαση δομής δικτυών κάνοντας αλλαγές και ελέγχοντας την ακρίβειά του με βάση το αρχικό. Ειδικότερα στην v3 χρησιμοποιήθηκε το NAS για βελτιστοποίηση στο βάθος (ποσότητα φίλτρων) κάθε επιπέδου και στη μείωση ποσότητας επίπεδων συμφόρησης.

- **Τελική δομή του μοντέλου MobileNet v3**

Το αποτέλεσμα των παραπάνω διαφοροποιήσεων είναι η βελτίωση της ακρίβειας και της ταχύτητας του μοντέλου σε σχέση με την 2η έκδοσή του.
Επιπλέον αποφασίστηκε να σχεδιαστεί και μια μικρότερη και γρηγορότερη έκδοση που θα έχει λιγότερα επίπεδα και μειωμένο βάθος, διατηρώντας όμως τις παραπάνω διαφοροποιήσεις. Το αποτέλεσμα είναι η έκδοση **Small,** που είναι 10% λιγότερο ακριβής αλλά δύο (2) φορές πιο γρήγορη.
Η αρχική έκδοση (**Large)** έχει την παρακάτω δομή (Πίνακας 2.5), οπού **SE** είναι η χρήση συμπίεσης-και-διέγερσης, **NL** είναι ο τύπος συνάρτησης ενεργοποίησης **RE** (ReLU) ή **HS** (h-Swish), **exp_size** είναι το βάθος επέκτασης στα επίπεδα συμφόρησης και **s** είναι το βήμα (Stride).

Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Πίνακας 2.5 : Δομή του μοντέλου MobileNet v3-Large

Η έκδοση **Small** έχει παρόμοια δομή με την **Large** αλλά λιγότερα επίπεδα. Η δομή του μοντέλου MobileNet v3-Small εμφανίζεται στον Πίνακα 2.6.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Πίνακας 2.6: Δομή του μοντέλου MobileNet v3-Small

Σύγκριση έκδοσης MobileNet v3 με την έκδοση MobileNetV2

Με τις παραπάνω αλλαγές η ακρίβεια της έκδοσης v3-Large αυξήθηκε κατά 2.9% (=73.8%-70.9%) και ταυτόχρονα το κόστος μειώθηκε κατά 15.4%{=(44-52) / 52} σε σχέση με την έκδοση v2.
Με τη δημιουργία ειδικού μικρού μοντέλου v3-Small, αντί της μείωσης της ανάλυσης των επίπεδων κατά 65% (=1-0.35) όλου του μοντέλου που έγινε στην έκδοση v2 0.35, η ακρίβεια αυξήθηκε κατά 7.7% (=64.9-57.2%) και ταυτόχρονα το κόστος μειώθηκε κατά 7.2% {=(16.7-15.5) / 16.7} σε σχέση με την έκδοση v2.

Network	Top-1	P-1	P-2	P-3
V3-Large 1.0	**73.8**	44	42.5	31.7
V2 1.0	70.9	52	48.3	37.0
V3-Small	**64.9**	15.5	14.9	10.7
V2 0.35	57.2	16.7	15.6	11.9

Πίνακας 2.7 : Πίνακας σύγκρισης των εκδόσεων v2 και v3 (Small, Large). Στη στήλη Top-1 δίνεται η ακρίβεια των μοντέλων ως ποσοστό και στις στήλες P-n (n=1,2,3) δίνεται ο χρόνος εκτέλεσης σε ms της κάθε έκδοσης σε τρεις επεξεργαστικές μονάδες

Κεφάλαιο 3

Σχεδιασμός Συστήματος ADAS

3.1 Εξεταζόμενα υποσυστήματα Υποβοήθησης

Στην παρούσα εργασία σχεδιάζεται ένα αυτόνομο Σύστημα ADAS, το οποίο θα είναι δυνατόν να τοποθετηθεί σε οποιοδήποτε αυτοκίνητο, χρησιμοποιώντας ως υλικό ένα προσιτό σε κόστος μικροϋπολογιστή, τύπου Raspberry Pi, με μερικές προσθήκες (δομοστοιχεία).

Το υπόψη Σύστημα θα έχει τις εξής δυνατότητες υποβοήθησης του οδηγού:

- **Υποσύστημα προειδοποίησης αλλαγής λωρίδας κυκλοφορίας (Lane Departure Warning system - LDW):**

  Προειδοποίηση του οδηγού σε περίπτωση που το όχημα ξεφεύγει από τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας του

- **Υποσύστημα προειδοποίησης μπροστινής σύγκρουσης (Forward Collision Warning - FCW):**

  Προειδοποίηση του οδηγού σε περίπτωση που η απόσταση του οχήματός του από το προπορευόμενο όχημα ή πεζό μειώνεται απότομα

- **Υποσύστημα προειδοποίησης εκκίνησης προπορευόμενου αυτοκινήτου (Forward Vehicle Start Alarm - FVSA):**

  Προειδοποίηση του οδηγού όταν δεν αντιδρά σε περίπτωση που το προπορευόμενο όχημα απομακρύνεται (π.χ. μετά από φανάρι).

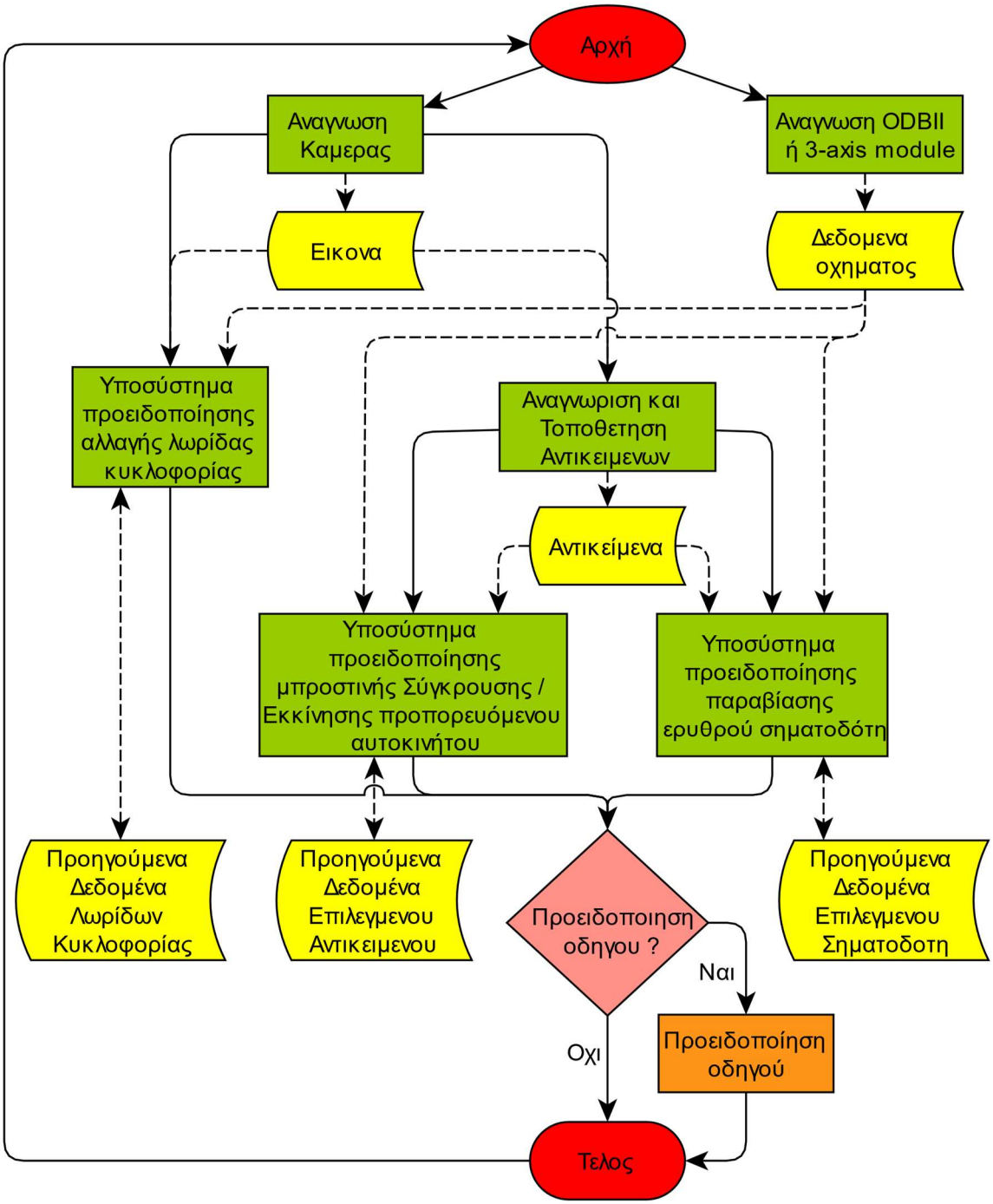
- **Υποσύστημα προειδοποίησης παραβίασης ερυθρού σηματοδότη (Traffic Light Warning - TLW):**

  Αναγνώριση ερυθρού σηματοδότη και προειδοποίηση του οδηγού σε περίπτωση έλλειψης πέδησης

Επισημαίνεται ότι ως «Σύστημα» στην παρούσα εργασία νοείται το σύνολο του λογισμικού που αναπτύχθηκε και των υλικών - εξοπλισμού που χρησιμοποιήθηκαν για τη δημιουργία του συστήματος ADAS, ενώ ως «υποσυστήματα» νοούνται τα ανωτέρω αναφερόμενα επιμέρους υποσυστήματα υποβοήθησης του οδηγού (προειδοποίησης αλλαγής λωρίδας, προειδοποίησης μπροστινής σύγκρουσης, προειδοποίησης εκκίνησης προπορευόμενου αυτοκινήτου, προειδοποίησης

παραβίασης ερυθρού σηματοδότη), που εξετάζονται και υλοποιούνται στην παρούσα εργασία μέσω του «Συστήματος».

Η δομή του Συστήματος ADAS που σχεδιάζεται στην παρούσα εργασία εμφανίζεται στο Σχήμα 3.1.



Σχήμα3.1 : Διάγραμμα δομής του Συστήματος ADAS που σχεδιάστηκε

Όσον αφορά στο λογισμικό που αναπτύχθηκε, το Σύστημα είναι γραμμένο στην ευέλικτη γλώσσα προγραμματισμού Python και χρησιμοποιεί βιβλιοθήκη της για να επικοινωνεί με τα δομοστοιχεία που έχουν προστεθεί στο μικροϋπολογιστή Raspberry Pi.

Για τον εντοπισμό των λωρίδων κυκλοφορίας έγινε χρήση του αλγορίθμου ανίχνευσης ακμών Canny και του αλγορίθμου Hough Transform αναγνώρισης ευθειών από σημεία ακμών.

Για την εκτέλεση του νευρωνικού δικτύου ssd_mobilenet_v3_small, το οποίο χρησιμοποιείται για τον εντοπισμό των διαφόρων ειδών αντικειμένων, έγινε χρήση της βιβλιοθήκης Tensorflow Lite, δηλαδή της υποέκδοσης της γνωστής βιβλιοθήκης λογισμικού ανοιχτού κώδικα Tensorflow που επιτρέπει εργασίες μηχανικής εκμάθησης, η οποία έχει σχεδιαστεί ειδικά για εκτέλεση σε χαμηλής ισχύος μονάδες επεξεργασίας.

Από το Νευρωνικό δίκτυο το Σύστημα λαμβάνει τις θέσεις των διαφόρων αντικειμένων και τις διαμοιράζει στις λειτουργίες του, οι οποίες συγκρίνοντας τη θέση, το μέγεθος και τη συχνότητα ταύτισής τους με αντικείμενα που προηγουμένως είχαν αναγνωριστεί και λαμβάνοντας επιπλέον υπόψη στοιχεία για την κατάσταση του οχήματος (π.χ. αλλαγή στην ταχύτητα) προειδοποιούν τον οδηγό αναλόγως.

Από πλευράς υλικού το Σύστημα χρησιμοποιεί ως εισόδους μια εικόνα και τη μεταβολή ταχύτητας του οχήματος. Την εικόνα την λαμβάνει από μια κάμερα τοποθετημένη στην καμπίνα του αυτοκίνητου κατά τρόπο τέτοιο ώστε να έχει παρόμοια οπτική με τον οδηγό. Τα δεδομένα μεταβολής ταχύτητας του οχήματος τα λαμβάνει από ενσωματωμένο κύκλωμα (3 axis gyro+accelerometer) ή από το όχημα (μέσω OBDII).

Η επεξεργασία γίνεται σε μικρού μεγέθους και χαμηλής ισχύος - κατανάλωσης υπολογιστή, τύπου Raspberry PI 4, ώστε και να μην καταλαμβάνεται πολύς χώρος και να είναι δυνατή η τροφοδοσία από το αυτοκίνητο. Επιπλέον τέτοιου είδους υπολογιστές είναι αρκετά προσιτοί στην τιμή και ευέλικτοι στις δυνατότητές τους, αφού διαθέτουν μια σειρά από pins («καρφίτσες») που μπορούν να ρυθμιστούν ως δέκτες ή πομποί, προσφέροντας τη δυνατότητα να αυξηθούν οι δυνατότητες του υπολογιστή μέσω ειδικών δομοστοιχείων (modules). Στην περίπτωση της παρούσας εργασίας ένα δομοστοιχείο που αναγνωρίζει τις αλλαγές ταχύτητας του οχήματος σε τρεις διαστάσεις και ένα δομοστοιχείο που διαθέτει φωτεινές ενδείξεις.

Το Σύστημα προειδοποιεί τον οδηγό σε περίπτωση κινδύνου μέσω της χρήσης ενός ηχείου ενώ για λιγότερο επικίνδυνες καταστάσεις ή για παροχή πληροφοριών σχετικά με την κατάσταση του Συστήματος χρησιμοποιεί τις φωτεινές ενδείξεις.

3.2 Θεωρητικό υπόβαθρο συστημάτων υποβοήθησης [25],[26],[27],[28],[29],[30]

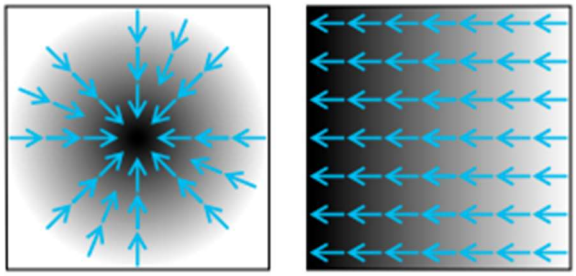
3.2.1 Αλγόριθμος Canny

Για την αναγνώριση των διαχωριστικών γραμμών των λωρίδων κυκλοφορίας χρησιμοποιείται καταρχήν ο ανιχνευτής ακμών Canny.

Η ανίχνευση ακμών έχει ως στόχο να μειώσει δραστικά τον όγκο των δεδομένων μιας εικόνας διατηρώντας ωστόσο τις βασικές της ιδιότητες, που αφορούν στα όρια των αντικειμένων τα οποία είναι απαραίτητα για την περαιτέρω επεξεργασία της εικόνας.

Οι ανιχνευτές ακμών αποτελούν εργαλεία επεξεργασίας εικόνας που είναι σχεδιασμένα για να ανιχνεύουν εικονοστοιχεία ακμών. Τα εικονοστοιχεία των ακμών είναι εικονοστοιχεία στα οποία η ένταση της συνάρτησης εικόνας μεταβάλλεται απότομα ενώ οι ακμές ορίζονται ως σύνολα συνδεδεμένων εικονοστοιχείων ακμών. Μια γραμμή μπορεί να θεωρηθεί ως ένα λεπτό τμήμα ακμής στο οποίο η ένταση του υποβάθρου στην κάθε πλευρά της γραμμής είτε είναι πολύ μεγαλύτερη είτε πολύ μικρότερη από την ένταση των εικονοστοιχείων της γραμμής.

Οι τοπικές μεταβολές στην ένταση μπορούν να ανιχνευθούν χρησιμοποιώντας παραγώγους πρώτης και δεύτερης τάξης μιας συνάρτησης της έντασης της εικόνας. Δεδομένου ότι μια εικόνα αποτελεί ένα δισδιάστατο πίνακα εικονοστοιχείων με διαφορετικές τιμές έντασης, η προσέγγιση που επιλέγεται για τον υπολογισμό της πρώτης ή δεύτερης παραγώγου σε κάθε θέση εικονοστοιχείου μιας εικόνας είναι η χρήση χωρικής συνέλιξης με κατάλληλα φίλτρα, μέσω των οποίων υπολογίζεται πόσο μεταβάλλεται η ένταση στους άξονες x, y, ώστε να βρεθεί το μέτρο και η κατεύθυνση της κλίσης της συνάρτησης της έντασης της εικόνας.



Σχήμα 3.2.: Η κλίση ως διάνυσμα αναπαρίσταται από τα μπλε βέλη και δηλώνει την κατεύθυνση της μεγαλύτερης μεταβολής μια βαθμωτής συνάρτησης. Το μέτρο (οι τιμές) της συνάρτησης αναπαρίσταται στην κλίμακα του γκρι και αυξάνει από το άσπρο (χαμηλές τιμές) προς το μαύρο χρώμα (υψηλές τιμές)

Ο ανιχνευτής Canny είναι ένας ανιχνευτής ακμών που αναπτύχθηκε από τον John F. Canny το 1986. Ο στόχος του Canny ήταν να αναπτύξει έναν βέλτιστο αλγόριθμο ως προς τα παρακάτω κριτήρια:

- Χαμηλό ρυθμό σφαλμάτων. Ο αλγόριθμος θα πρέπει να είναι σε θέση να εντοπίσει όλες τις ακμές χωρίς να υπάρχουν πλασματικές αποκρίσεις, δηλαδή

να ελαχιστοποιήσει την πιθανότητα ανίχνευσης σημείων που δεν αποτελούν ακμές. Συνεπώς ο αλγόριθμος θα πρέπει να ελαχιστοποιεί το θόρυβο, δηλαδή να μεγιστοποιεί το λόγο σήματος προς θόρυβο

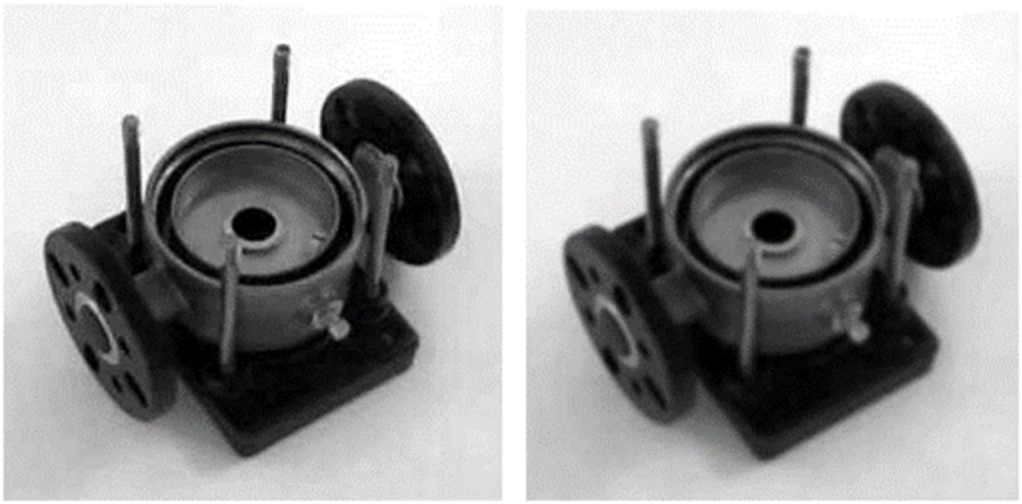
- Τα σημεία των ακμών πρέπει να είναι πολύ καλά εντοπισμένα Οι ακμές που εντοπίζονται πρέπει να είναι όσο το δυνατόν πλησιέστερα προς τις πραγματικές ακμές
- Απόκριση μοναδικού σημείου Ο ανιχνευτής θα πρέπει να επιστρέφει μόνο ένα σημείο για κάθε σημείο της πραγματικής ακμής.

Ο αλγόριθμος Canny αποτελείται από τα εξής βήματα:

- Εξομάλυνση της εικόνας εισόδου με ένα φίλτρο Gauss, ώστε να μειωθεί ο θόρυβος

  Ο θόρυβος που ενυπάρχει στις εικόνες που λήφθηκαν με χρήση κάμερας είναι αναπόφευκτος. Επιπρόσθετα η χρήση παραγώγων για την ανίχνευση ακμών οδηγεί σε αποτελέσματα ιδιαιτέρως ευαίσθητα στο θόρυβο. Για να μην υπάρχουν λανθασμένες ακμές αυτός ο θόρυβος πρέπει να αφαιρεθεί. Ένα τρόπος για την αφαίρεση του θορύβου είναι η χρήση ενός φίλτρου Gauss που χαρακτηρίζονται από ένα σύνολο ιδιοτήτων που τα καθιστούν ιδανικά για χρήση σε εφαρμογές επεξεργασίας εικόνας, διότι διατηρούν το σχήμα και την μορφολογία των ακμών. Στην εργασία έγινε χρήση του παρακάτω φίλτρου Gauss 3x3, δεδομένου ότι στην εικόνα που λαμβάνεται από την κάμερα του Συστήματος υπάρχει χαμηλός θόρυβος

0.07511361	0.1238414	0.07511361
0.1238414	0.20417996	0.1238414
0.07511361	0.1238414	0.07511361



Αρχική εικόνα(αριστερά) Εικόνα μετά την εξομάλυνση Gauss (δεξιά)

Εικόνα 3.1.: Παράδειγμα εξομάλυνσης εικόνας με χρήση φίλτρου Gauss

- Υπολογισμός του μέτρου (έντασης) και της γωνίας (κατεύθυνσης) της κλίσης της έντασης σε κάθε εικονοστοιχείο της εικόνας

  Ο αλγόριθμος Canny εντοπίζει τις ακμές στις περιοχές όπου η ένταση της εικόνας εμφανίζει τη μεγαλύτερη αλλαγή. Αυτές οι περιοχές βρίσκονται υπολογίζοντας τις παραγώγους της εικόνας I_x και I_y στους άξονες x και y αντίστοιχα. Για τον υπολογισμό των τιμών των παραγώγων του κάθε εικονοστοιχείου μπορούμε να χρησιμοποιήσουμε τα ζεύγη πυρήνων φίλτρων παραγώγων Roberts, Prewitt και Sobel. Στην παρούσα εργασία οι παράγωγοι I_x και I_y υπολογίζονται με συνέλιξη της συνάρτησης έντασης I με τα φίλτρα Sobel Kx και Ky αντίστοιχα:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

  Το μέτρο *G* και η κατεύθυνση *θ* της κλίσης υπολογίζονται από τους τύπους:

$$|G| = \sqrt{I_x^2 + I_y^2},$$
$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

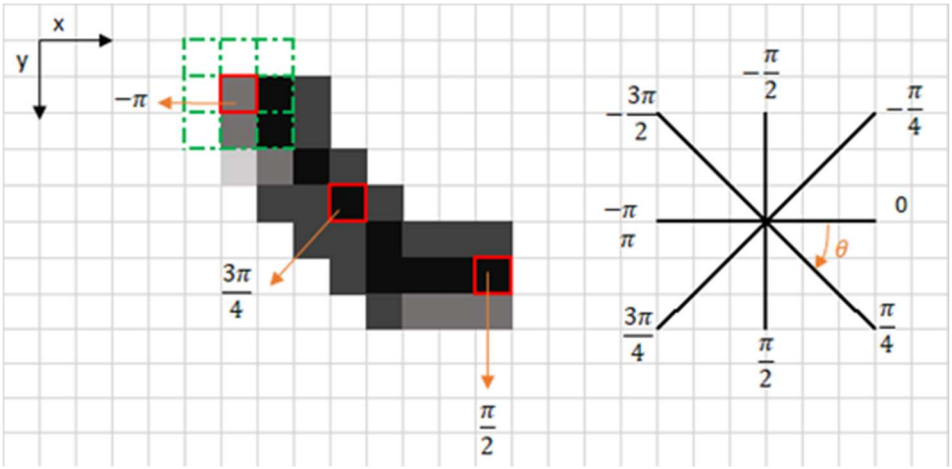
Εικόνα 3.2: Εικόνα μετά την εξομάλυνση Gauss (αριστερά), Εικόνα με εύρεση κλίσης (δεξιά)

- Εξάλειψη των μη μέγιστων τιμών του μέτρου της κλίσης στην εικόνα

  Στην ιδανική περίπτωση, η τελική εικόνα θα έπρεπε να έχει λεπτές ακμές. Ωστόσο η εικόνα-κλίση συνήθως περιέχει «ζάρες» μεγάλου εύρους γύρω από τα τοπικά μέγιστα. Συνεπώς πρέπει να προβούμε στην εξασθένηση αυτών των χαρακτηριστικών και μια μέθοδος είναι να προχωρήσουμε στην εξάλειψη των μη μεγίστων για να λεπτύνουμε τις ακμές.

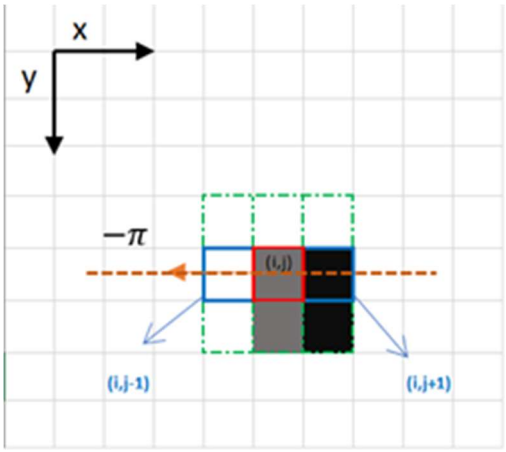
Η βασική αρχή είναι απλή: ο αλγόριθμος περνά από όλα τα σημεία στον πίνακα έντασης της κλίσης και βρίσκει τα εικονοστοιχεία με τη μέγιστη τιμή στις κατευθύνσεις των ακμών.

Παρακάτω δίνεται ένα απλό παράδειγμα:



Σχήμα 3.3: 1ο Παράδειγμα εξάλειψης των μη μέγιστων τιμών

Το κόκκινο πλαίσιο στην επάνω αριστερή γωνία που υπάρχει στην παραπάνω εικόνα, αντιπροσωπεύει ένα εικονοστοιχείο έντασης του πίνακα έντασης κλίσης που υποβάλλεται σε επεξεργασία. Η αντίστοιχη κατεύθυνση της ακμής αναπαρίσταται από το πορτοκαλί βέλος με γωνία - π ακτινίων (+/-180 μοίρες).



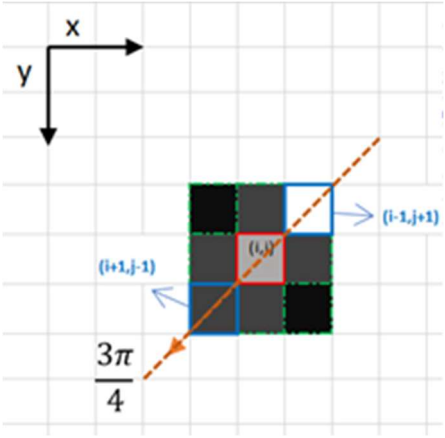
Σχήμα 3.4: 1ο Παράδειγμα εξάλειψης των μη μέγιστων τιμών (λεπτομέρεια)

Ας εστιάσουμε στην επάνω αριστερή γωνία με το εικονοστοιχείο με το κόκκινο πλαίσιο.

Η κατεύθυνση της ακμής είναι η πορτοκαλί διακεκομμένη γραμμή (οριζόντια από αριστερά προς τα δεξιά). Ο σκοπός του αλγόριθμου είναι να ελέγξει εάν τα εικονοστοιχεία στην ίδια κατεύθυνση είναι περισσότερο ή λιγότερο έντονα από αυτά που υποβάλλονται σε επεξεργασία. Στο παραπάνω παράδειγμα, το εικονοστοιχείο (i, j) υποβάλλεται σε επεξεργασία και τα εικονοστοιχεία στην ίδια κατεύθυνση επισημαίνονται με μπλε (i, j-1) και (i, j+1). Εάν ένα από αυτά τα δύο

εικονοστοιχεία είναι πιο έντονο από αυτό που υποβάλλεται σε επεξεργασία, τότε διατηρείται μόνο το πιο έντονο. Το εικονοστοιχείο (i, j-1) φαίνεται να είναι πιο έντονο, επειδή είναι λευκό (τιμή 255). Ως εκ τούτου, η τιμή της έντασης του τρέχοντος εικονοστοιχείου (i, j) ορίζεται στο 0. Εάν δεν υπάρχουν εικονοστοιχεία στην κατεύθυνση της ακμής με πιο έντονες τιμές, τότε η τιμή του τρέχοντος εικονοστοιχείου διατηρείται.

Ας εστιάσουμε τώρα σε ένα άλλο παράδειγμα:

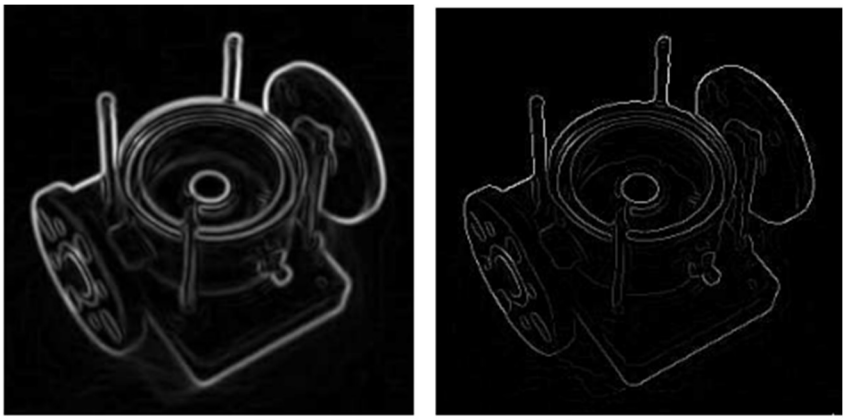


Σχήμα 3.5: 2ο Παράδειγμα εξάλειψης των μη μέγιστων τιμών

Σε αυτή την περίπτωση η κατεύθυνση είναι η πορτοκαλί διακεκομμένη διαγώνια γραμμή. Επομένως, το πιο έντονο εικονοστοιχείο προς αυτή την κατεύθυνση είναι το εικονοστοιχείο (i-1, j+1).

Συνοψίζοντας, κάθε εικονοστοιχείο έχει 2 κύρια κριτήρια (κατεύθυνση ακμής σε ακτίνια και ένταση εικονοστοιχείου (μεταξύ 0-255)).

Το αποτέλεσμα της διαδικασίας εξάλειψης των μη μεγίστων είναι η ίδια εικόνα με πιο λεπτές ακμές.



Εικόνα 3.3: Αποτέλεσμα εξάλειψης των μη μεγίστων

Ωστόσο υπάρχουν κάποιες διακυμάνσεις όσον αφορά την ένταση των ακμών. Ορισμένα εικονοστοιχεία φαίνεται να είναι πιο φωτεινά από άλλα και το μειονέκτημα αυτό εξαλείφεται με τα δύο επόμενα βήματα.

- Ανίχνευση των ακμών μέσω εφαρμογής κατωφλίωσης, δηλαδή εφαρμογής δύο διαφορετικών τιμών εντάσεων ως κατωφλίων, ενός χαμηλού και ενός υψηλού κατωφλίου

  Η εφαρμογής κατωφλίωσης υστέρησης στοχεύει στον εντοπισμό 3 ειδών εικονοστοιχείων: «ισχυρά», «ασθενή» και μη σχετικά:

  • Τα «ισχυρά» εικονοστοιχεία είναι τα εικονοστοιχεία που έχουν τόσο υψηλή ένταση που είμαστε σίγουροι ότι συμβάλλουν στην τελική ακμή.

  • Τα «ασθενή» εικονοστοιχεία είναι τα εικονοστοιχεία που έχουν μια τιμή έντασης που δεν είναι αρκετή για να θεωρηθούν «ισχυρά», αλλά δεν είναι αρκετά μικρά για να θεωρηθούν ως μη σχετικά με την ανίχνευση ακμών.

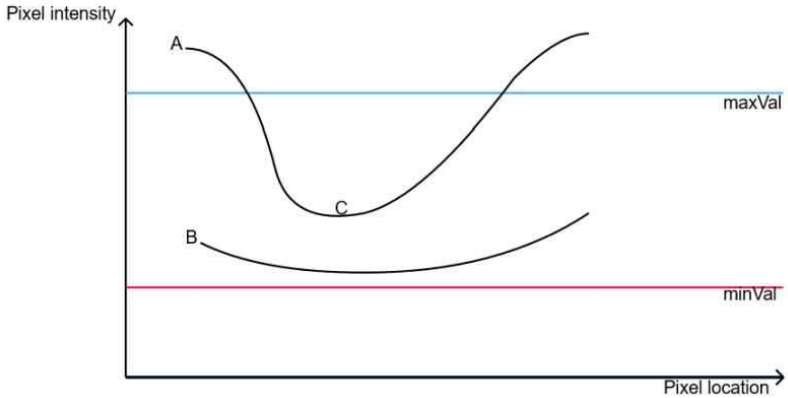
  • Άλλα εικονοστοιχεία θεωρούνται μη σχετικά με την ακμή.

  Τα ανωτέρω σχετιζόμενα με τα δύο κατώφλια έχουν ως εξής:

  • Το υψηλό κατώφλιο χρησιμοποιείται για τον προσδιορισμό των «ισχυρών» εικονοστοιχείων, δηλαδή των εικονοστοιχείων που έχουν ένταση υψηλότερη από το υψηλό κατώφλιο

  • Το χαμηλό κατώφλιο χρησιμοποιείται για τον προσδιορισμό των μη σχετικών εικονοστοιχείων, δηλαδή των εικονοστοιχείων που έχουν ένταση μικρότερη από το χαμηλό κατώφλιο

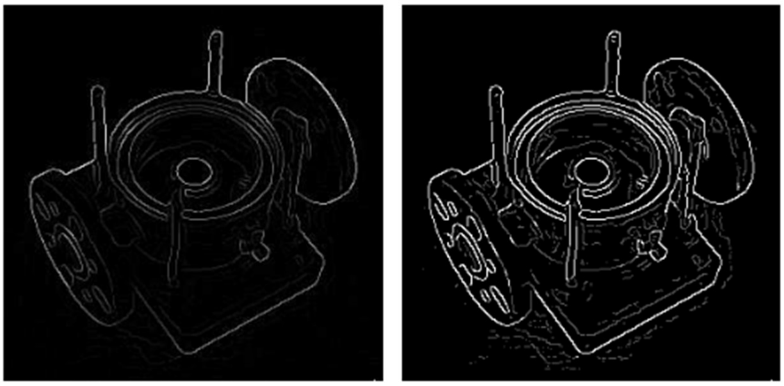
  • Όλα τα εικονοστοιχεία που έχουν ένταση μεταξύ των δύο κατωφλίων επισημαίνονται ως «ασθενή» και ο μηχανισμός ανάλυσης συνδεσιμότητας (επόμενο βήμα) θα μας βοηθήσει να εντοπίσουμε αυτά που θα μπορούσαν να θεωρηθούν ως» ισχυρά» και αυτά που θεωρούνται ως μη σχετικά.



Σχήμα 3.6: Διαγραμματική απεικόνιση ανίχνευσης ακμών με διπλή κατωφλίωση

Όσα εικονοστοιχεία έχουν μέτρο μεγαλύτερο του υψηλότερου κατωφλίου γίνονται λευκά και θεωρούνται ως «ισχυρά» εικονοστοιχεία της ακμής, όσων το μέτρο είναι ανάμεσα θεωρούνται ως «ασθενή» και όσων το μέτρο είναι μικρότερο του χαμηλότερου κατωφλίου μηδενίζονται και γίνονται μαύρα.

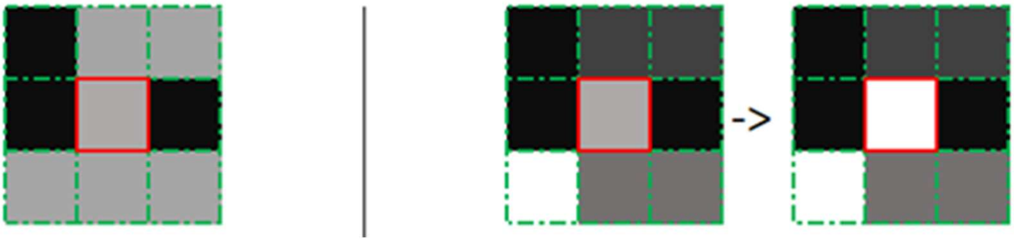
Το αποτέλεσμα αυτού του βήματος είναι μια εικόνα με τιμές έντασης μόνο 2 εικονοστοιχείων («ισχυρών» και «ασθενών»):



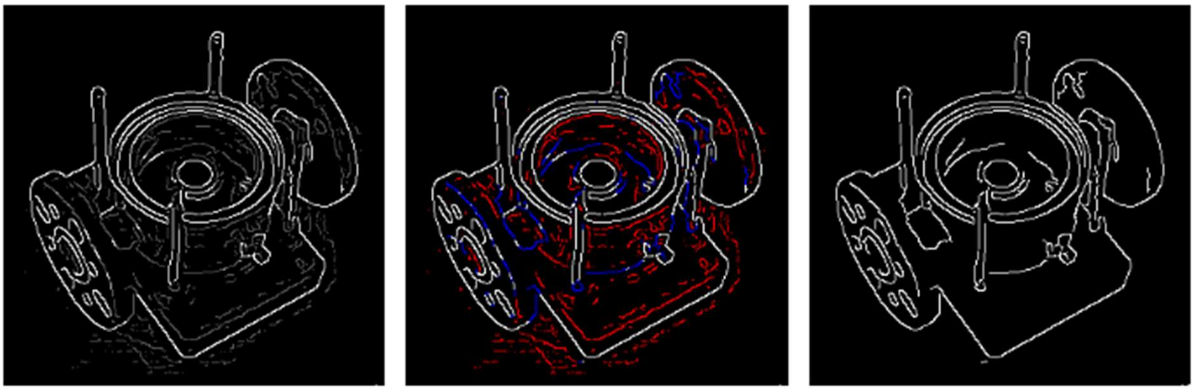
Εικόνα 3.4:Εικόνα εξάλειψης μη μεγίστων (αριστερά) – Εικόνα διπλής κατωφλίωσης (δεξιά), όπου τα «ισχυρά» εικονοστοιχεία είναι λευκά και τα «ασθενή» είναι γκρι

- Ανάλυση συνδεσιμότητας για την ανίχνευση και σύνδεση των ακμών

Με βάση τα αποτελέσματα του κατωφλίου, η ανάλυση συνδεσιμότητας συνίσταται στη μετατροπή «ασθενών» εικονοστοιχείων σε «ισχυρά», εάν και μόνο εάν τουλάχιστον ένα από τα εικονοστοιχεία γύρω από αυτό που υποβάλλεται σε επεξεργασία είναι «ισχυρό», όπως περιγράφεται παρακάτω:



Σχήμα 3.7: Όσα «ασθενή» εικονοστοιχεία συνορεύουν με «ισχυρά» εικονοστοιχεία γίνονται λευκά, αν όχι μηδενίζονται και γίνονται μαύρα.



Εικόνα 3.5: Αποτέλεσμα ανάλυσης συνδεσιμότητας για την ανίχνευση-σύνδεση ακμών
(α) Διπλή κατωφλίωση (β) Ανάλυση συνδεσιμότητας (γ) Τελική εικόνα
Η μεσαία εικόνα (β) δείχνει τις «ισχυρές» λευκές, τις «ασθενείς» που συνορεύουν με «ισχυρές» με μπλε και τις υπόλοιπες «ασθενείς» ακμές με κόκκινο χρώμα

3.2.2 Μετασχηματισμός Hough

Για την αναγνώριση των διαχωριστικών γραμμών των λωρίδων κυκλοφορίας, μετά τον ανιχνευτή ακμών Canny, χρησιμοποιείται ο αλγόριθμος Hough Transform.

Ο εντοπισμός της θέσης και ο προσδιορισμός της μορφής των γραμμών και των ακμών των ψηφιακών εικόνων είναι ένας από τους σημαντικότερους τομείς της ψηφιακής επεξεργασίας εικόνων. Μια από τις πλέον γνωστές τεχνικές είναι ο μετασχηματισμός Hough (Hough Transform), ο οποίος χρησιμοποιείται ευρύτατα για τον προσδιορισμό ευθύγραμμων τμημάτων και γενικότερα παραμετρικά εκφραζόμενων καμπυλών σε ψηφιακές εικόνες.

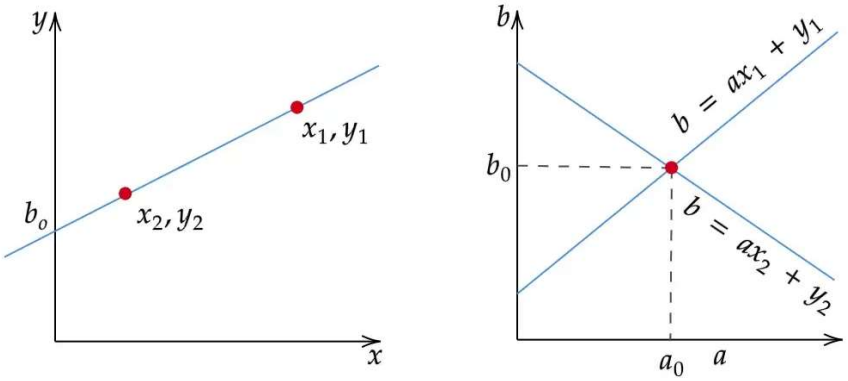
Με τη χρήση του μετασχηματισμού Hough καθίσταται δυνατόν από το σύνολο των σημείων μιας εικόνας να βρεθούν υποσύνολα αυτών των σημείων που κείνται πάνω σε ευθείες γραμμές. Ο αλγόριθμος Hough Transform αναγνωρίζει πολύπλοκες γραμμές στις φωτογραφίες με αντιστοίχιση σημείων ακμών στο χώρο Hough. Για το λόγο αυτό, πριν από τον αλγόριθμο Hough Transform είναι σημαντικό να εκτελείται ανίχνευση ακμών για να δημιουργηθεί μια εικόνα ακμών, η οποία στη συνέχεια θα χρησιμοποιηθεί ως είσοδος στον αλγόριθμο Hough (όπως προαναφέρθηκε στην παρούσα εργασία γίνεται χρήση του ανιχνευτή ακμών Canny).
Αυτές οι ακμές μπορεί να είναι ευθείες που αποτελούνται από πολλά σημεία, δηλαδή ψηφιά της εικόνας. Πρέπει συνεπώς να βρούμε την ευθεία που περνάει από όλα αυτά τα σημεία.

Έστω ότι έχουμε δυο σημεία (x_1,y_1), (x_2,y_2) της εικόνας, στο καρτεσιανό σύστημα συντεταγμένων (x, y). Τότε υπάρχει ευθεία με εξίσωση $\mathbf{y = ax + b}$ που περνάει από τα σημεία αυτά (Σχήμα 3.8, σύστημα συντεταγμένων x, y). Μπορούμε να αλλάξουμε την δομή της εξίσωσης παίρνοντας ως αγνώστους το a και το b, οπότε προκύπτει $\mathbf{b = -xa + y}$

Τώρα μπορούμε να φτιάξουμε ένα νέο σύστημα συντεταγμένων με άξονες a και b (Σχήμα 3.8, σύστημα συντεταγμένων a, b). Με αυτόν τον τρόπο κάθε σημείο της εικόνας (π.χ. το σημείο $\mathbf{x_1, y_1}$) μετασχηματίζεται σε μια ευθεία ($\mathbf{b=-ax_1+y_1}$) στο νέο σύστημα συντεταγμένων με άξονες a και b, που προκύπτει από όλα τα ζεύγη (a, b) που ικανοποιούν τη σχέση $\mathbf{y_1 = ax_1 + b}$. Για παράδειγμα για ένα σημείο $(x_i,y_i) = (1,1)$ της εικόνας προκύπτει η εξίσωση $1=a\cdot1+b$ δηλαδή η ευθεία $b=1-a$ στο σύστημα συντεταγμένων (a, b). Συνεπώς μέσω του μετασχηματισμού (a,b) η ευθεία $b=1-a$ αντιστοιχεί στο σημείο (1,1) του συστήματος συντεταγμένων (x,y).

Στόχος του μετασχηματισμού Hough είναι να μετατρέψουμε όλα τα σημεία (x_i,y_i) της εικόνας στο σύστημα (x,y) σε ευθείες στο σύστημα (a,b) διότι τα σημεία (a_j,b_j) που τέμνονται «πιο έντονα» οι ευθείες στο πεδίο (a,b) θα μας οδηγήσουν, μέσω των εξισώσεων $\mathbf{y = a_jx + b_j,}$ στις «εντονότερες» ευθείες της αρχικής εικόνας. Στο παράδειγμα του Σχήματος 3.8, όπου έχουμε μόνο 2 σημεία (x_1,y_1), (x_2,y_2), οι ευθείες $\mathbf{b=-ax_1+y_1}$ και $\mathbf{b=-ax_2+y_2}$ τέμνονται στο σημείο $\mathbf{(a_0, b_0),}$ συνεπώς η ευθεία που περνά

με επιτυχία από τα σημεία (x_1,y_1), (x_2,y_2) είναι η $\mathbf{y=a_0x+b_0}$. Προφανώς σε περίπτωση πολλών σημείων (x_i,y_i) θα υπάρχουν πολλά σημεία τομής των ευθειών $\mathbf{b=-ax_i+y_i}$, και από αυτά θα επιλεγούν εκείνα όπου τέμνονται «πιο έντονα» οι ευθείες $\mathbf{b=-ax_i+y_i}$, ώστε να προκύψουν ζεύγη τιμών **aj,bj** που θα αντιστοιχούν στις «εντονότερες» ευθείες $\mathbf{y=a_jx+b_j}$ της αρχικής εικόνας, δηλαδή σε αυτές που περνάνε με επιτυχία από τα σημεία (x_i,y_i) της αρχικής εικόνας.

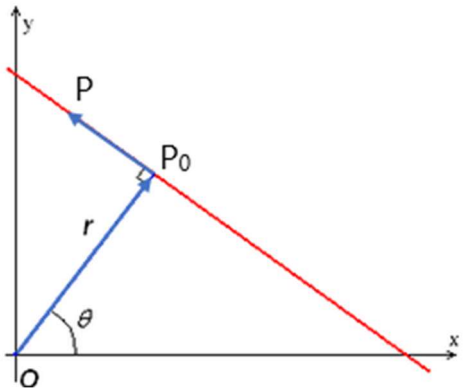


Σχήμα 3.8 : Αριστερά το σύστημα (x,y), δεξιά το σύστημα (a,b)

Στον παραπάνω μετασχηματισμό, που μετασχηματίζει τα σημεία της εικόνας σε ευθείες είναι αδύνατο να εντοπιστούν ευθείες κατακόρυφες, διότι αυτές θα είχαν άπειρη κλίση. Πράγματι στην κατακόρυφη ευθεία y=ax+b, το a που αντιπροσωπεύει την κλίση θα έπρεπε να είναι άπειρο. Το πρόβλημα αυτό λύθηκε με την προσέγγιση των Duda και Hart οι οποίοι αντικατέστησαν το σύστημα συντεταγμένων (a, b) με το σύστημα συντεταγμένων (θ, ρ) των πολικών παραμέτρων μιας ευθείας.
Στην περίπτωση αυτή η σχέση γίνεται:

$$r = x\cdot\cos\theta + y\cdot\sin\theta$$

όπου r είναι η απόσταση είναι απόσταση του σημείου τομής των αξόνων x,y από το κοντινότερο σημείο της ευθείας (P_0) και θ είναι η γωνία μεταξύ του άξονα x και της ευθείας που συνδέει το σημείο τομής των αξόνων με το ανωτέρω κοντινότερο σημείο



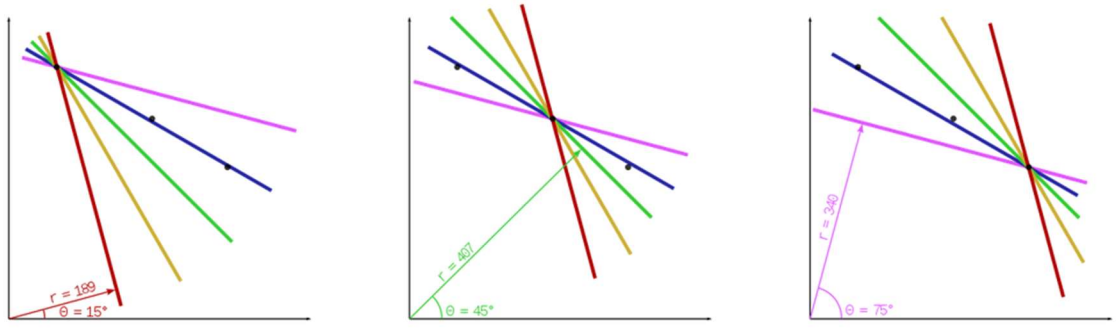
Σχήμα 3.9: Απεικόνιση μετασχηματισμού Hough

Πράγματι για κάθε σημείο P πάνω στη γραμμή το διάνυσμα $P\text{-}P_0$ είναι κάθετο στο διάνυσμα $P_0\text{-}0 = P_0$. Συνεπώς για κάθε σημείο P (x,y) της ευθείας θα πρέπει να ικανοποιείται η εξίσωση $(P\text{-}P_0)\circ P_0=0$ ή $P\circ P_0 = P_0\circ P_0$. Εφόσον P= (x,y) και $P_0=(r\cos\theta+r\sin\theta)$, με χρήση του τύπου του εσωτερικού γινομένου διανυσμάτων, θα έχουμε $x\cdot r\cdot\cos\theta + y\cdot r\cdot\sin\theta = r\cdot\cos\theta\cdot r\cdot\cos\theta + r\cdot\sin\theta\cdot r\cdot\sin\theta$ ή $r\cdot(x\cdot\cos\theta+y\cdot\sin\theta)=r^2\cdot(\cos^2\theta+\sin^2\theta)$ ή $r=x\cdot\cos\theta+y\cdot\sin\theta$.

Συνεπώς για κάθε σημείο (x,y) της εικόνας πρέπει να υπολογίσουμε τον μετασχηματισμό $r = x\cdot\cos\theta + y\cdot\sin\theta$. Για το λόγο αυτό θα πρέπει να δώσουμε διάφορες τιμές στη γωνία θ για να βρούμε τα r για κάθε σημείο της εικόνας. Καταρχήν καθορίζουμε τα όρια τιμών των παραμέτρων r και θ που θα λάβουμε υπόψη στους υπολογισμούς (συνήθως για τη γωνία θ είναι $-90\le\theta\le90$ ή $0\le\theta\le180$ και για την απόσταση $-D\le r\le D$ όπου D η μέγιστη απόσταση ανάμεσα στις απέναντι γωνίες μια εικόνας). Για κάθε γωνία θ_i που επιλέγουμε υπολογίζουμε για κάθε σημείο x,y την τιμή $r = x\cdot\cos\theta + y\cdot\sin\theta$. Κατόπιν δημιουργούμε ένα πίνακα r, θ, στον οποίο καταγράφουμε τις τιμές των θ και r του προηγούμενου βήματος.

Στο Σχήμα 3.10 καταγράφεται υπό μορφή παραδείγματος ο παραπάνω αλγόριθμος για τον υπολογισμό του μετασχηματισμού Hough για 3 σημεία (μαύρα σημεία στην εικόνα) και 5 επιλεγμένες γωνίες θ=15º,30º,45º,60º και 75º.

Για κάθε ένα από τα 3 σημεία φαίνονται οι ευθείες (με διαφορετικό χρώμα) που περνάνε από αυτό για κάθε τιμή της γωνίας θ. Κάθε μια από τις ευθείες αυτές είναι κάθετη στο ευθύγραμμο τμήμα μήκους r (με μορφή βέλους στην εικόνα), το οποίο σχηματίζει γωνία με τον άξονα των x, ίση με την τιμή της γωνίας θ βάσει της οποίας προσδιορίσθηκε το r από τον τύπο $r = x\cdot\cos\theta + y\cdot\sin\theta$. Οι διάφορες τιμές των θ και r καταγράφονται υπό μορφή πίνακα.



θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

Σχήμα 3.10: Παράδειγμα μετασχηματισμού Hough

Από την εικόνα παρατηρούμε ότι για τιμή της γωνίας θ ίση με 60º το μήκος των βελών είναι παραπλήσιο και για τα 3 σημεία της εικόνας και συνεπώς και οι αντίστοιχες ευθείες της εικόνας, εν προκειμένω οι μπλε χρώματος, είναι παρόμοιες. Επομένως τα 3 σημεία της εικόνας κείνται πλησιέστερα στην μπλε ευθεία, που σημαίνει ότι μέσω του μετασχηματισμού Hough προσδιορίσθηκε η ευθεία που συνδέει τα 3 σημεία της εικόνας

3.2.3 Επιλογή μοντέλου νευρωνικού δικτύου και εκπαίδευσή του

Όπως προαναφέρθηκε το υπό σχεδίαση Σύστημα, πέραν του υποσυστήματος προειδοποίησης αλλαγής λωρίδας, θα περιλαμβάνει τα υποσυστήματα προειδοποίησης μπροστινής σύγκρουσης, προειδοποίησης εκκίνησης προπορευόμενου αυτοκινήτου και προειδοποίησης παραβίασης ερυθρού σηματοδότη.

Ως εκ τούτου είναι ιδιαιτέρως σημαντική η επιλογή μοντέλου κατάλληλου για την αξιόπιστη και γρήγορη αναγνώριση των εν λόγω αντικειμένων με χρήση του χαμηλού κόστους και ισχύος εξοπλισμού που χρησιμοποιείται στην παρούσα εργασία.

Για τη αναγνώριση και τοποθέτηση των αντικειμένων στο χώρο χρησιμοποιήθηκε το μοντέλο νευρωνικού δικτύου MobileNet v3 Small που, όπως εξηγήθηκε στο Κεφάλαιο 2, είναι ιδανικό για χαμηλής ισχύος επεξεργαστές, όπως ο Raspberry PI 4, αφού έχει, συγκριτικά με άλλα ισοδύναμα μοντέλα, λίγες παραμέτρους και πράξεις και μπορεί να εκτελείται πολύ γρήγορα και με καλή ακρίβεια κάτι που είναι εξαιρετικά απαραίτητο σε ένα σύστημα υποβοήθησης του οδηγού, στο οποίο η μέγιστη ανταπόκριση είναι ζωτικής σημασίας.

Πιο συγκεκριμένα χρησιμοποιήθηκε το μοντέλο ssd_mobilenet_v3_small. Το πρόθεμα ssd δηλώνει τον αλγόριθμο επιλογής αντικειμένων σε μια εικόνα για να αναγνωριστούν από το MobileNet. Το μοντέλο ssd_mobilenet_v3_small εκπαιδεύτηκε πάνω στο σύνολο δεδομένων 200.000 εικόνων (COCO Dataset) και μπορεί να αναγνωρίσει 90 είδη αντικειμένων.

Από το σύνολο των παραπάνω ειδών αντικειμένων, αυτά που σχετίζονται με την παρούσα εργασία είναι το Traffic light, που λαμβάνεται υπόψη στο υποσύστημα προειδοποίησης παραβίασης ερυθρού σηματοδότη (Traffic Light Warning - TLW) καθώς και όλα τα παρακάτω αντικείμενα, που λαμβάνονται υπόψη στα υποσυστήματα προειδοποίησης μπροστινής σύγκρουσης (Forward Collision Warning - FCW) και προειδοποίησης εκκίνησης προπορευόμενου οχήματος (Forward Vehicle Start Alarm - FVSA)

- Αυτοκίνητο (Car)
- Ιππέας (rider)
- Μοτοσυκλέτα (motorcycle)
- Πεζός (person)
- Λεωφορείο (bus)
- Φορτηγό όχημα (truck)
- Ποδήλατο (bike)
- Τραίνο (train)

3.3 Υποσύστημα προειδοποίησης αλλαγής λωρίδας κυκλοφορίας

Το υποσύστημα προειδοποίησης αλλαγής λωρίδας κυκλοφορίας (Lane Departure Warning system - LDW) λαμβάνει μια εικόνα του δρόμου, απομονώνει τις διαχωριστικές γραμμές των λωρίδων κυκλοφορίας, τις ορίζει ως ευθείες στον χώρο και, με βάση τη θέση τους οπτικά, προειδοποιεί τον οδηγό σε περίπτωση που η θέση τους εισχωρεί σε μια ζώνη η οποία δηλώνει τη θέση των διαμηκών πλευρών του οχήματος.

Στην παρούσα παράγραφο αναφέρονται αρχικά τα βήματα που ακολουθούνται σε κάθε στιγμιότυπο (εικόνα) που λαμβάνεται από το Σύστημα (Βήματα ανά στιγμιότυπο) για την ανίχνευση των διαχωριστικών γραμμών των λωρίδων κυκλοφορίας και για τον ορισμό της περιοχής ενδιαφέροντος που καλύπτει τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας. Στη συνέχεια αναφέρονται τα βήματα που ακολουθούνται προκειμένου να εξασφαλίζεται η αξιόπιστη λειτουργία του υποσυστήματος προειδοποίησης αλλαγής λωρίδας σε βάθος χρόνου, κατά τη διαδοχική λήψη στιγμιότυπων (εικόνων).

3.3.1 Βήματα ανά στιγμιότυπο:

1. **Απομόνωση διαχωριστικών γραμμών λωρίδων κυκλοφορίας**

   Σε αυτό το βήμα γίνεται απομόνωση των διαχωριστικών γραμμών της λωρίδας κυκλοφορίας από το δρόμο και ταυτόχρονα αφαίρεση οποιουδήποτε αλλού χρώματος του περιβάλλοντος δεν παρουσιάζει ενδιαφέρον.

Εικόνα 3.6 : Παράδειγμα εικόνας (στιγμιότυπο) από προοπτική οδηγού

Για παράδειγμα (Εικόνα 3.6), έστω μια εικόνα που λαμβάνεται από το Σύστημα κάποια χρονική στιγμή (στιγμιότυπο) σε μια λεωφόρο. Θα πρέπει να απομονωθούν οι διαχωριστικές γραμμές της λωρίδας κυκλοφορίας από το

περιβάλλον και ειδικότερα από το οδόστρωμα. Οι διαχωριστικές γραμμές έχουν λευκό ή κίτρινο χρώμα και το οδόστρωμα γκρι. Για το λόγο αυτό τίθεται ένα δυναμικό (όχι στατικό) κατώτατο όριο φωτεινότητας το οποίο ταυτίζεται με την εκάστοτε (ανά στιγμιότυπο) φωτεινότητα του ιδίου του οδοστρώματος, γκρι χρώματος, ώστε να αφαιρείται το οδόστρωμα καθώς και σκιές, λακούβες, επιδιορθώσεις και άλλες ανωμαλίες του. Αυτό γίνεται μέσω της δειγματοληψίας ενός κομματιού του οδοστρώματος μπροστά από το όχημα από το οποίο εξάγεται η μέση φωτεινότητα.

Το αποτέλεσμα του βήματος αυτού είναι μια εικόνα, στην οποία στο επίπεδο του δρόμου φαίνονται μόνο οι διαχωριστικές γραμμές της λωρίδας κυκλοφορίας ακόμα και σε μεταβολές φωτεινότητας του οδοστρώματος σε περιπτώσεις αλλαγής του φωτισμού του περιβάλλοντος από σκιές, φώτα και θέση του ηλίου (Εικόνα 3.7) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.1 του Παραρτήματος Β).

Εικόνα 3.7 : Παράδειγμα απομόνωσης των διαχωριστικών γραμμών της λωρίδας κυκλοφορίας από τον δρόμο

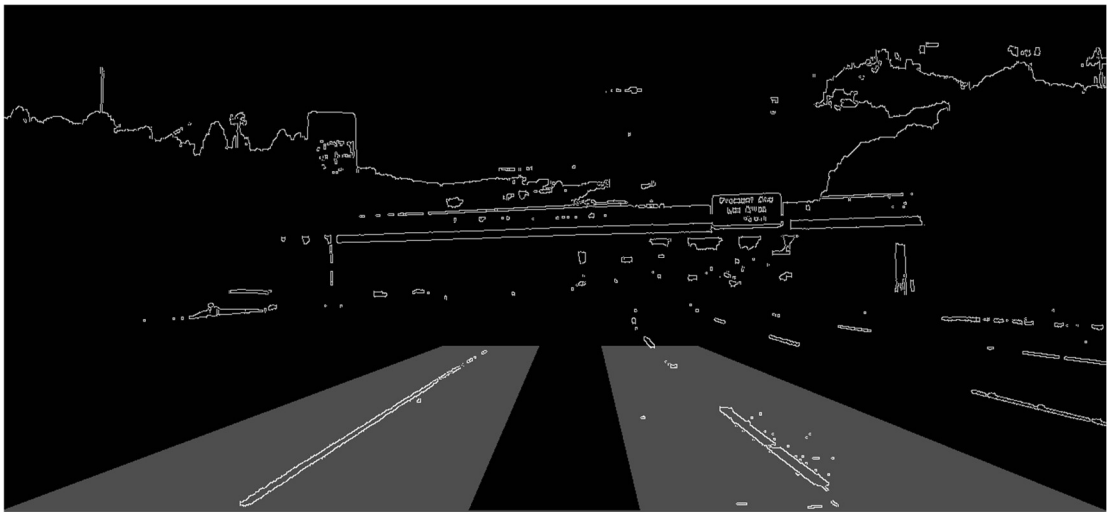
2. Εύρεση Ακμών

Στην εικόνα που έχει προκύψει από το προηγούμενο βήμα γίνεται εύρεση ακμών με τη χρήση του αλγορίθμου Canny (Εικόνα 3.8), ώστε να παραμείνουν μόνο οι ακμές των διαχωριστικών γραμμών της λωρίδας κυκλοφορίας. Με τον τρόπο αυτό αφαιρούνται ομοιόμορφες επιφάνειες της απομόνωσης του προηγούμενου βήματος, όπως (π.χ. ο ουρανός μπλε χρώματος) και διατηρούνται μόνο οι σημαντικές πληροφορίες εναλλαγών χρώματος (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.2 του Παραρτήματος Β).

Εικόνα 3.8 : Παράδειγμα εύρεσης ακμών στην απομονωμένη εικόνα

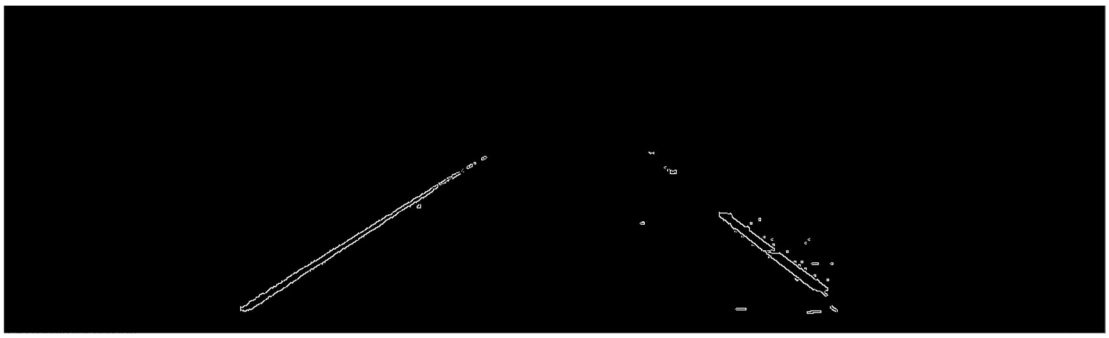
3. Ορισμός περιοχής ενδιαφέροντος

Σε αυτό το βήμα γίνεται αφαίρεση όλων των ακμών, εκτός από αυτές που βρίσκονται σε μια περιοχή ενδιαφέροντος που καλύπτει τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας. Το σχήμα της περιοχής ενδιαφέροντος που ορίστηκε αγνοεί τον κεντρικό άξονα του δρόμου και έχει ένα όριο ύψους ως προς τον ορίζοντα διότι σε μεγάλη απόσταση από το σύστημα η εικόνα που λαμβάνεται περιέχει θόρυβο και κυρίως αδιάφορη πληροφορία (Εικόνα 3.9).



Εικόνα 3.9 : Παράδειγμα σχήματος της περιοχής ενδιαφέροντος με γκρι

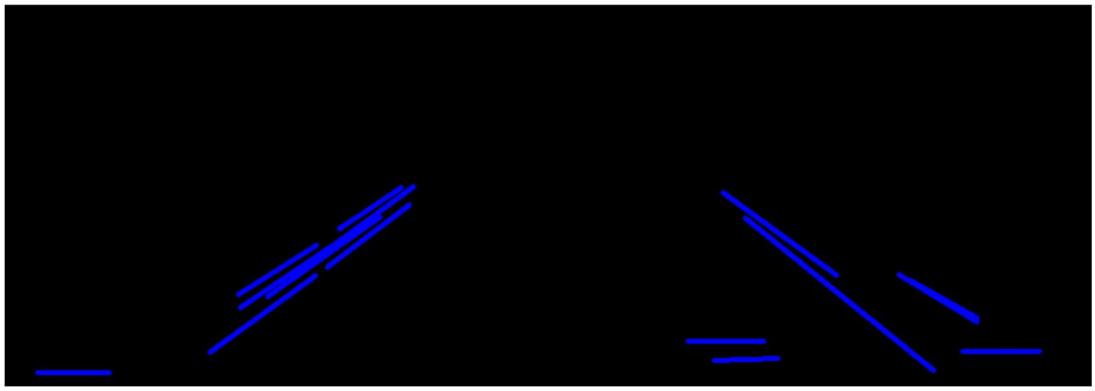
Το αποτέλεσμα του βήματος είναι μια εικόνα ψηφίων τύπου 0 και 1, που εν πολλοίς ορίζουν τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας (Εικόνα 3.10). (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.3 του Παραρτήματος Β).



Εικόνα 3.10 : Παράδειγμα αφαίρεσης ακμών εκτός περιοχής ενδιαφέροντος

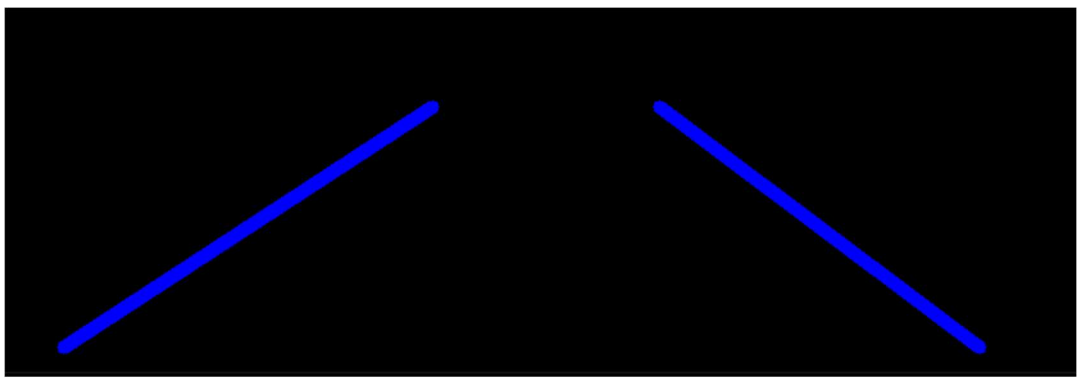
4. Εύρεση και ομαδοποίηση ευθείων με χρήση Houghlines

Στο παρόν βήμα γίνεται μετασχηματισμός των σημείων, δηλαδή του συνόλου των ψηφίων της εικόνας που προέκυψαν στο προηγούμενο βήμα, σε ευθείες (Houghlines), σύμφωνα με το μετασχηματισμό Hough που εξηγήθηκε προηγουμένως. Το αποτέλεσμα είναι ευθείες διάφορου μήκους και κατεύθυνσης (Εικόνα 3.11).



Εικόνα 3.11 : Παράδειγμα αναγνώρισης ευθείων από σημεία

Από τις ευθείες αυτές, διαγράφονται όλες εκείνες των οποίων η κατεύθυνση αποκλίνει κατά πολύ σε σχέση με τις κατευθύνσεις που αναμένεται να έχουν οι διαχωριστικές γραμμές της λωρίδας κυκλοφορίας και όσες απομένουν διαχωρίζονται σε δυο ομάδες, μια για κάθε πλευρά του δρόμου, και στη συνέχεια συνδυάζονται σε δυο μέσες ευθείες (Εικόνα 3.12).



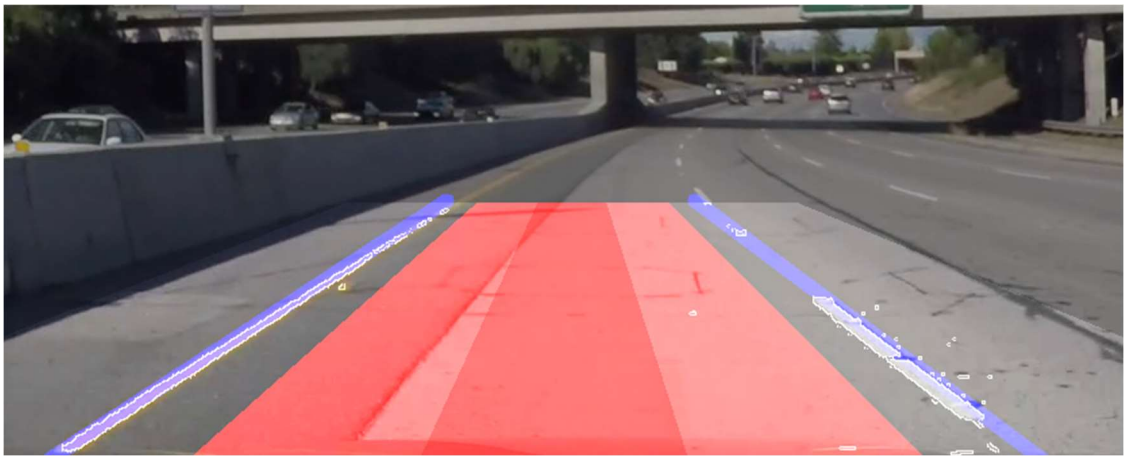
Εικόνα 3.12 : Παράδειγμα συνδυασμού πολλών ευθείων σε δυο μέσες ευθείες

Το αποτέλεσμα είναι δυο ευθείες που σχεδόν ταυτίζονται με τις πραγματικές διαχωριστικές γραμμές της λωρίδας κυκλοφορίας (Εικόνα 3.13). (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.4 του Παραρτήματος Β).

Εικόνα 3.13 : Παράδειγμα συνδυασμού όλων των προηγούμενων βημάτων

5. Καθορισμός ζώνης πορείας οχήματος

Η εικόνα εισόδου στο Σύστημα σχεδόν ταυτίζεται με την οπτική του οδηγού του οχήματος, συνεπώς υπάρχει μια ζώνη στην εικόνα που ορίζει την πορεία του αυτοκίνητου (Εικόνα 3.14). (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.5 του Παραρτήματος Β).



Εικόνα 3.14 : Παράδειγμα ζώνης πορείας ενός οχήματος ορισμένου πλάτους χρωματισμένη με κόκκινο χρώμα

6. Προειδοποίηση οδηγού

Όταν μια από τις ευθείες που ορίζουν τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας εισέρχεται στη ζώνη πορείας του οχήματος τότε αποστέλλεται σήμα προειδοποίησης στον οδηγό. (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.6 του Παραρτήματος Β).

3.3.2. Βήματα σε βάθος χρόνου:

1. **Ομαλοποίηση ευθείων στο χρόνο**

   Σε κάθε νέα εικόνα (στιγμιότυπο) που λαμβάνεται από το Σύστημα γίνεται και νέα αναζήτηση των διαχωριστικών γραμμών της λωρίδας κυκλοφορίας. Ωστόσο το αποτέλεσμα σε κάποιες λίγες περιπτώσεις μπορεί να είναι πολύ διαφορετικό σε σχέση με το αποτέλεσμα της προηγουμένης αναζήτησης ή ακόμα και εσφαλμένο (π.χ. το σύστημα ενδέχεται να αναγνωρίσει άλλες γραμμές ως διαχωριστικές γραμμές και όχι τις πραγματικές διαχωριστικές γραμμές της λωρίδας κυκλοφορίας εξαιτίας ύπαρξης διάβασης πεζών, έντονης σκίασης του οδοστρώματος, αντανακλάσεων λόγω βρεγμένου οδοστρώματος κ.λπ.).

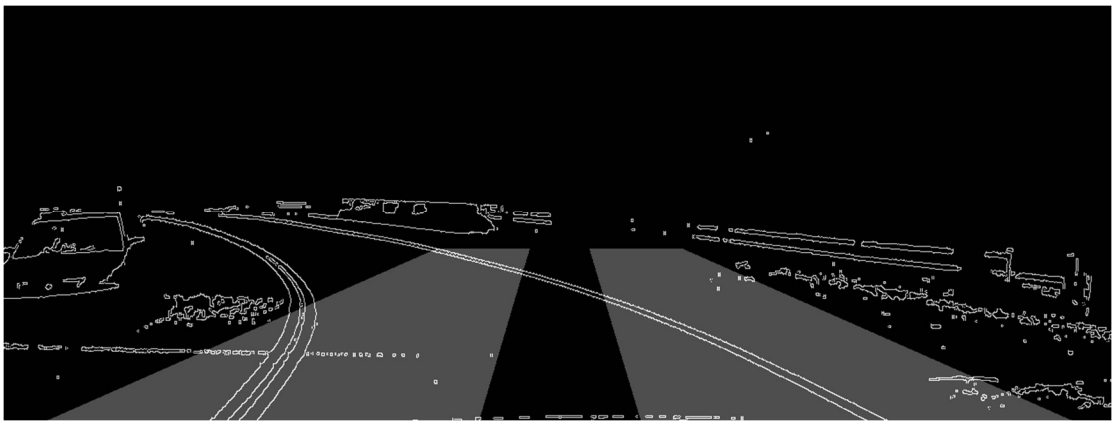
   Για το λόγο αυτό οι νέες ευθείες που ορίζουν τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας προσδιορίζονται λαμβάνοντας υπόψη και την προηγούμενη αναζήτηση, ώστε να εξασφαλίζεται μια ομαλοποίηση στο χρόνο και να μειώνονται οι περιπτώσεις λανθασμένης προειδοποίησης του οδηγού. (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.4 του Παραρτήματος Β).

2. **Καθυστέρηση ενεργοποίησης ειδοποίησης**

   Για την αποφυγή προειδοποίησης του οδηγού σε κάθε περίπτωση που το υποσύστημα αναγνωρίζει τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας μέσα στη ζώνη πορείας του οχήματος ακόμη και για πολύ μικρό χρονικό διάστημα, έχει εισαχθεί μια καθυστέρηση στην ενεργοποίηση της προειδοποίησης, ανά διαχωριστική γραμμή λωρίδας κυκλοφορίας. Πιο συγκεκριμένα έχει τεθεί ως προϋπόθεση ότι το υποσύστημα θα πρέπει να αναγνωρίσει διαχωριστική γραμμή λωρίδας κυκλοφορίας, ανά πλευρά του οχήματος, για κάποια συνεχόμενα στιγμιότυπα προκειμένου να ενεργοποιηθεί.

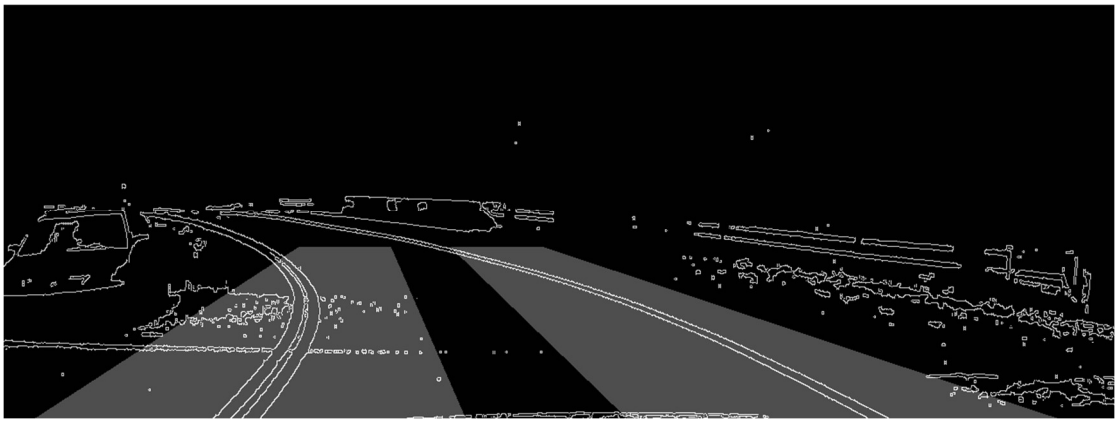
3. **Ορισμός δυναμικής περιοχής ενδιαφέροντος**

   Μια στατική περιοχή ενδιαφέροντος είναι ανεπαρκής. Όταν για παράδειγμα συναντάται μεταβολή στην κλίση του δρόμου ή απότομη στροφή πληροφορίες άσχετες με τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας εισέρχονται στην περιοχή ενδιαφέροντος και πληροφορίες χρήσιμες χάνονται (Εικόνα 3.15).



Εικόνα 3.15 : Παράδειγμα απότομης στροφής με στατική περιοχή ενδιαφέροντος

Μια λύση στο πρόβλημα αυτό είναι να εκμεταλλευτούμε τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας από προηγούμενες εικόνες ώστε να ορίσουμε ένα νέο κέντρο για την κατεύθυνση της δυναμικής τώρα πια περιοχής ενδιαφέροντος. Πράγματι οι διαχωριστικές γραμμές της λωρίδας κυκλοφορίας λόγω της προοπτικής τους συγκλίνουν σε σημείο που βρίσκεται περίπου στο κέντρο της λωρίδας κυκλοφορίας και στον ορίζοντα του δρόμου. Το σημείο αυτό τίθεται ως το σημείο σύγκλισης των δύο ζωνών της περιοχής ενδιαφέροντος. Καθώς το όχημα εισέρχεται σε μια στροφή το εκάστοτε (ανά στιγμιότυπο) νέο σημείο σύγκλισης των διαχωριστικών γραμμών θα μετακινείται σε σχέση με το προηγούμενο και συνεπώς θα μετακινείται αντίστοιχα το σημείο σύγκλισης των δύο ζωνών της περιοχής ενδιαφέροντος, με αποτέλεσμα η περιοχή ενδιαφέροντος να είναι πλέον δυναμική και να ακολουθεί την καμπύλη του δρόμου (Εικόνα 3.16). (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.3 του Παραρτήματος Β).



Εικόνα 3.16 : Παράδειγμα απότομης στροφής με δυναμική περιοχή ενδιαφέροντος

4. Απενεργοποίηση σε χαμηλές ταχύτητες

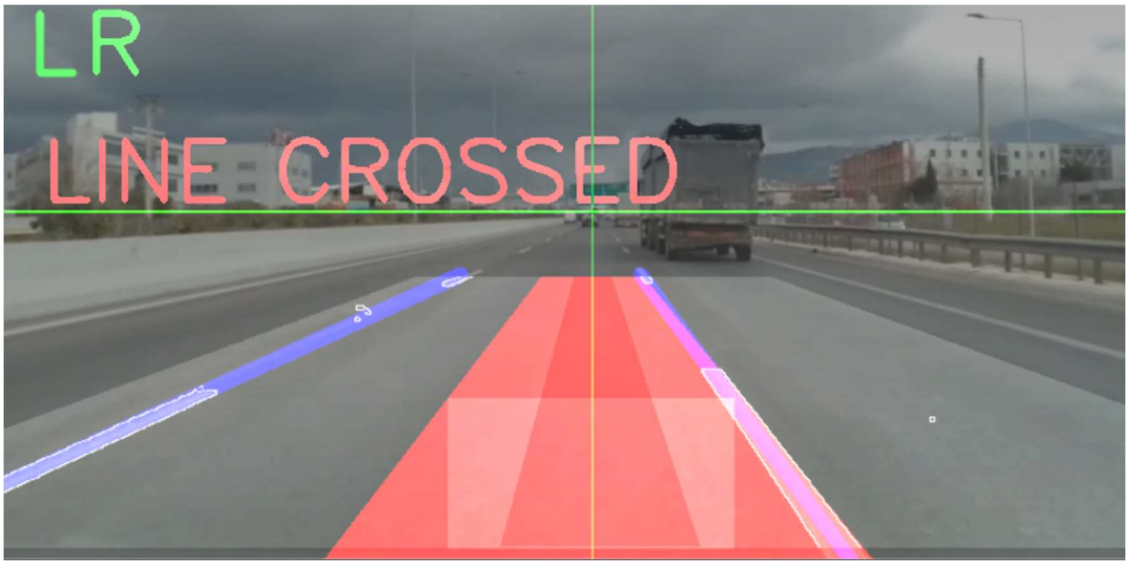
Σε περίπτωση που το όχημα κινείται με χαμηλές ταχύτητες, δηλαδή μικρότερες από 20 Km/h, που συνήθως συμβαίνει σε στροφές σε αστικούς δρόμους, τα οφέλη του υποσυστήματος γίνονται μηδαμινά και ειδικά σε κάποιες καταστάσεις μπορεί να προειδοποιήσουν λανθασμένα τον οδηγό (π.χ. λανθασμένη

ενεργοποίηση σε περίπτωση διαχωριστικών γραμμών θέσεων στάθμευσης). Για το λόγο αυτό το υποσύστημα απενεργοποιείται στις χαμηλές αυτές ταχύτητες (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.7 του Παραρτήματος Β).

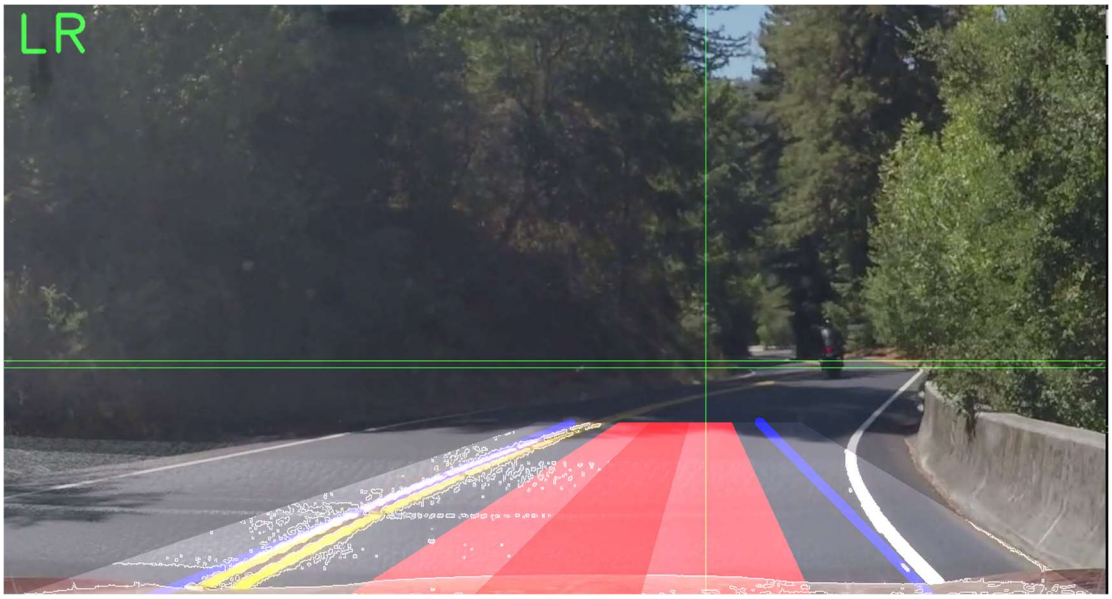
5. **Απενεργοποίηση σε περιπτώσεις ηθελημένης αλλαγής πορείας**

   Σε περίπτωση που ο οδηγός ενεργοποιήσει τα φώτα δείκτη κατεύθυνσης (φλας) τότε το υποσύστημα απενεργοποιείται (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §1.7 του Παραρτήματος Β)

Παραδείγματα λειτουργίας του υποσυστήματος προειδοποίησης αλλαγής λωρίδας κυκλοφορίας

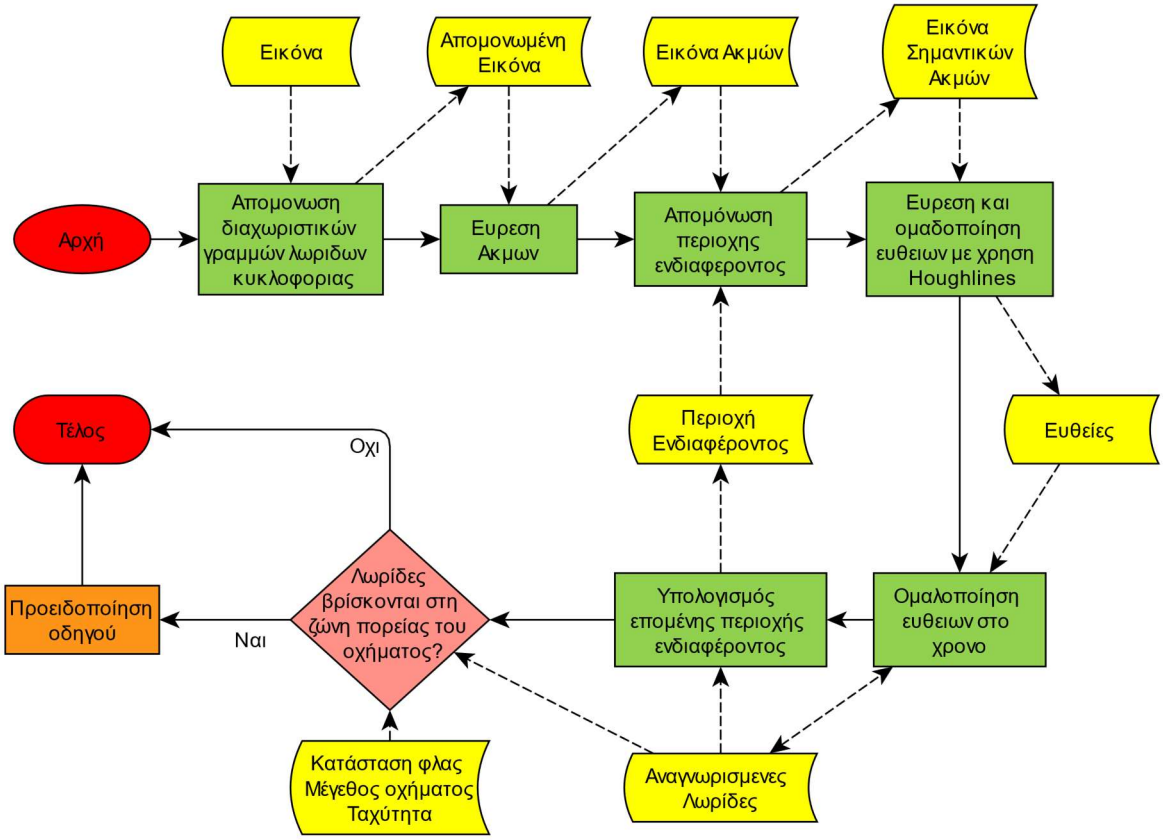


Εικόνα 3.17 : Παράδειγμα λειτουργίας του υποσυστήματος σε λεωφόρο ταχείας κυκλοφορίας



Εικόνα 3.18 : Παράδειγμα λειτουργίας του υποσυστήματος σε ορεινό δρόμο

Η **τελική δομή** του αλγορίθμου του υποσυστήματος είναι η εξής:



Σχήμα 3.11 : Διάγραμμα δομής του υποσυστήματος προειδοποίησης αποχώρησης από τη λωρίδα κυκλοφορίας

3.4 Υποσύστημα προειδοποίησης μπροστινής σύγκρουσης

Το υποσύστημα προειδοποίησης μπροστινής σύγκρουσης (Forward Collision Warning - FCW) δέχεται μια εικόνα (στιγμιότυπο) και τα αντικείμενα που περιέχει, τα οποία αναγνώρισε το Νευρωνικό δίκτυο ssd_mobilenet_v3_small. Στη συνέχεια επιλεγεί από τα αντικείμενα αυτά το πιο κοντινό στην ευθεία πορεία του οχήματος και ελέγχει την απόσταση από αυτό. Όταν αυτή η απόσταση μειωθεί απότομα προειδοποιεί τον οδηγό για επικείμενη σύγκρουση.

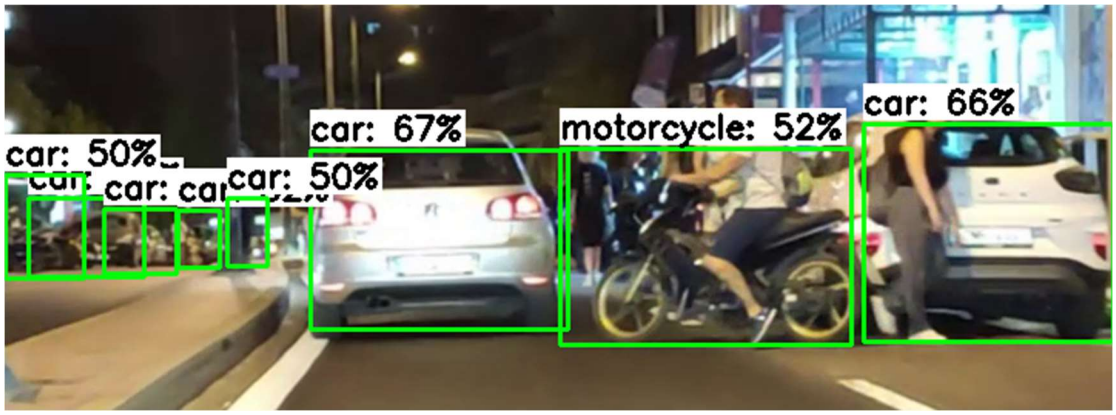
Στην παράγραφο αυτή αρχικά αναφέρονται τα βήματα που ακολουθούνται σε κάθε στιγμιότυπο (εικόνα) που λαμβάνεται από το Σύστημα (βήματα ανά στιγμιότυπο) για την επιλογή του σημαντικότερου αντικειμένου, δηλαδή του αντικειμένου με το οποίο υπάρχει μεγαλύτερος κίνδυνος μπροστινής σύγκρουσης. Στη συνέχεια αναφέρονται τα βήματα που ακολουθούνται προκειμένου να εξασφαλίζεται η αξιόπιστη λειτουργία του υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης σε βάθος χρόνου, κατά τη διαδοχική λήψη στιγμιότυπων (εικόνων).

Επισημαίνεται ότι, παρά το ότι στη συνέχεια της παρούσας παραγράφου γίνεται αναφορά σε περιπτώσεις μπροστινής σύγκρουσης με άλλο προπορευόμενο όχημα, τα αναφερόμενα ισχύουν και για τις περιπτώσεις μπροστινής σύγκρουσης με οποιοδήποτε άλλο αντικείμενο π.χ. πεζό, ποδήλατο ή ιππέα.

3.4.1 Βήματα ανά στιγμιότυπο (Εικόνα):

1. Οριοθετημένα «κουτιά»

Από την ανίχνευση των αντικειμένων στην εικόνα λαμβάνονται συντεταγμένες πολλών «κουτιών», που ορίζουν τις θέσεις και διαστάσεις των αντικειμένων καθώς και την πιθανότητα ένταξής τους σε συγκεκριμένη κατηγορία αντικειμένου, όπως φαίνεται στην εικόνα 3.19 (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §2.1 του Παραρτήματος Β).



Εικόνα 3.19 : Παράδειγμα αναγνωρισμένων αντικειμένων σε μια εικόνα

2. Εύρεση αντικειμένων στην πορεία του οχήματος

Με βάση τη θέση των «κουτιών» της προηγούμενης εικόνας και λόγω της προοπτικής (Σχήμα 3.12) μπορούμε να θεωρήσουμε ότι τα αντικείμενα στην πορεία μας είναι αυτά που τα κέντρα των «κουτιών» τους βρίσκονται προς το κέντρο της εικόνας ή τα «κουτιά» τους είναι αρκετά μεγάλα σε διαστάσεις ώστε τμήματά τους να βρίσκονται προς το κέντρο της εικόνας.

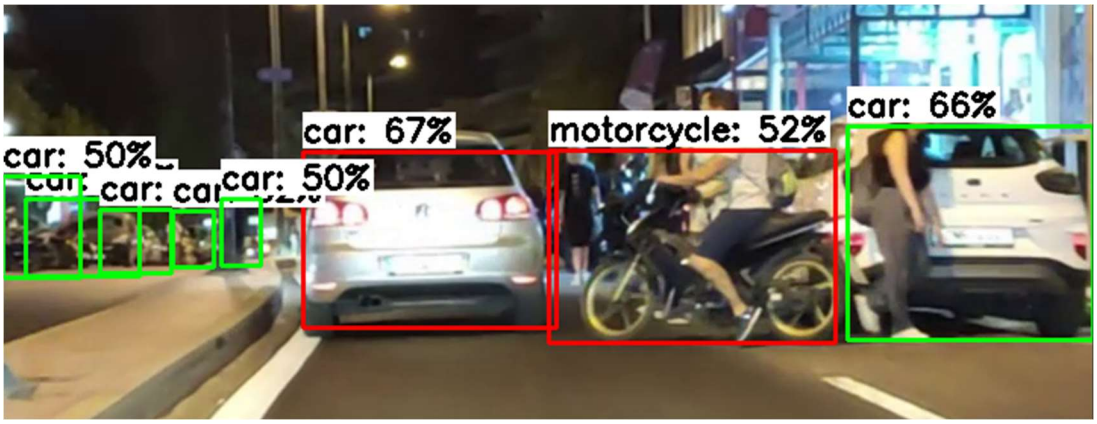
Σχήμα 3.12 : Παράδειγμα γραμμών προοπτικής

Με βάση τα ανωτέρω, τα αντικείμενα που θα είναι στην πορεία του οχήματος είναι αυτά που έχουν συνδυασμό μεγέθους «κουτιού» και απόστασης από το κέντρο της εικόνας, σύμφωνα με την εξίσωση:

$$W = \frac{H}{L}$$

Οπού **H** είναι το ύψος του «κουτιού», **L** είναι η απόσταση του κέντρου του «κουτιού» του αντικειμένου από το κέντρο της εικόνας και **W** είναι το τελικό βάρος, το οποίο και λαμβάνεται υπόψη στην αξιολόγηση των αντικειμένων σχετικά με το αν βρίσκονται στην πορεία του οχήματος.

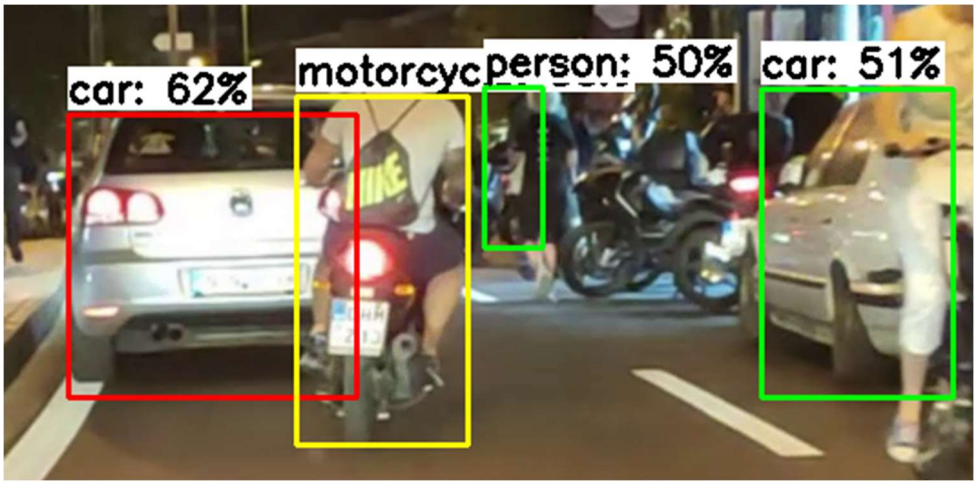
Θέτοντας ένα κατώτατο όριο για το βάρος μπορούμε να επιλέξουμε τα αντικείμενα που είναι στην πορεία μας. Στην παρακάτω Εικόνα 3. 20 φαίνονται τα εν λόγω αντικείμενα μέσα σε «κουτιά» κόκκινου χρώματος (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §2.2 του Παραρτήματος Β).



Εικόνα 3.20 : Παράδειγμα επιλογής αντικειμένων που βρίσκονται στην πορεία του οχήματος με κόκκινο

3. Επιλογή σημαντικότερου αντικειμένου

Η επιλογή του σημαντικότερου αντικειμένου γίνεται πάλι με τη χρήση της προοπτικής. Πιο συγκεκριμένα το σημαντικότερο αντικείμενο είναι το πιο κοντινό, δηλαδή με χρήση προοπτικής αυτό του οποίου οι συντεταγμένες της κάτω πλευράς του «κουτιού» του είναι χαμηλότερα από τις αντίστοιχες συντεταγμένες των κάτω πλευρών των «κουτιών» όλων των υπολοίπων αντικειμένων που βρίσκονται στην πορεία του οχήματος (Εικόνα 3.21) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §2.2 του Παραρτήματος Β).



Εικόνα 3.21 : Παράδειγμα επιλογής σημαντικότερου αντικειμένου με κίτρινο

3.4.2. Βήματα σε Βάθος χρόνου:

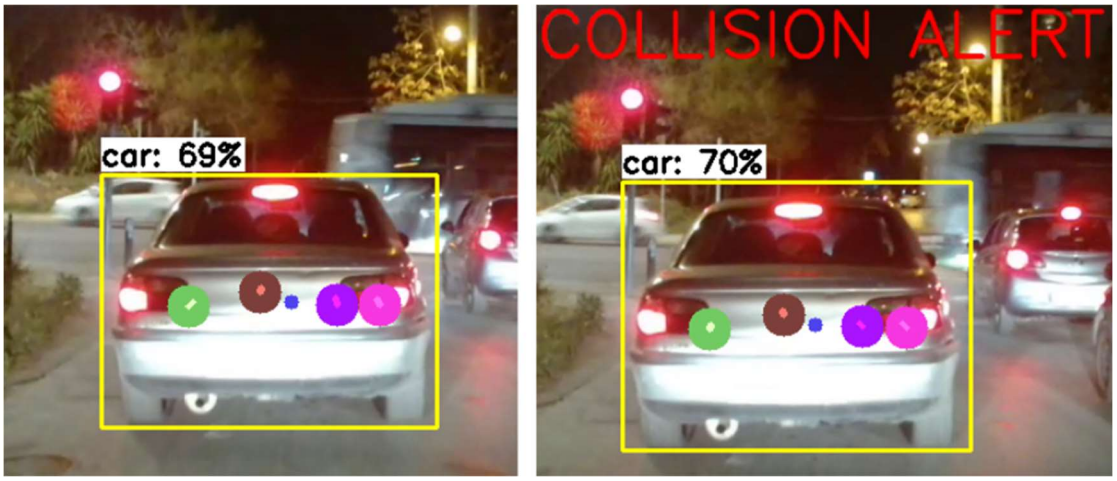
1. Παρακολούθηση μεγέθους και Προειδοποίηση

Αφού γίνει η επιλογή του σημαντικότερου αντικειμένου τότε σε κάθε επόμενο στιγμιότυπο (νέα εικόνα που λαμβάνεται από το Σύστημα) αποθηκεύεται το εμβαδόν του «κουτιού» του και συγκρινόμενο με το αντίστοιχο εμβαδόν που είχε ληφθεί στο προηγούμενο στιγμιότυπο δίνει μια γενική ιδέα μεταβολής της απόστασης από το αντικείμενο. Η χρήση όμως μόνο του νευρωνικού δικτύου για αναγνώριση απόστασης ενίοτε προκαλεί προβλήματα αφού σε μερικές περιπτώσεις, ειδικά τη νύχτα λόγω έλλειψης φωτισμού, τα όρια ενός αντικειμένου δεν μπορούν να καθοριστούν με την απαιτούμενη ακρίβεια. Αυτό οφείλεται κυρίως στη μειωμένη ακρίβεια που έχει ένα μικρό γρήγορο μοντέλο τύπου ssd MobileNet v3 Small. Αλλά ακόμα και αν γίνει χρήση ενός καλύτερου μοντέλου πάλι θα υπάρχουν περιπτώσεις που τα όρια του αντικειμένου δεν θα αναγνωριστούν σωστά λόγω μη επαρκούς φωτισμού.

Για το λόγο αυτό για τον υπολογισμό της μεταβολής της απόστασης γίνεται χρήση της απόστασης μεταξύ τεσσάρων (4) σημαντικών σημείων που βρίσκονται πάνω στο αντικείμενο.

Πιο συγκεκριμένα, αφού γίνει αναγνώριση του σημαντικότερου αντικειμένου από το Νευρωνικό δίκτυο, μέσα στο αναγνωρισμένο κουτί γίνεται αναζήτηση

των σημαντικότερων σημείων (γωνίες) με τη χρήση του αλγορίθμου Shi–Tomasi **[31]**. Αφού βρεθούν τα σημεία αυτά γίνεται υπολογισμός της μέσης απόστασης μεταξύ τους. Με τον τρόπο αυτό λαμβάνουμε ένα αντιπροσωπευτικό τμήμα του μεγέθους του αντικειμένου. Στη συνέχεια, σε κάθε στιγμιότυπο, γίνεται αναζήτηση της νέας θέσης αυτών των σημείων με τη χρήση της μεθόδου Lucas–Kanade **[32]**. Έτσι λαμβάνουμε ένα νέο αντιπροσωπευτικό τμήμα του μεγέθους του αντικειμένου το οποίο μπορεί να συγκριθεί πιο αποτελεσματικά με το αντίστοιχο του προηγούμενου στιγμιότυπου. Συνδυαστικά λοιπόν της απόστασης σημείων και του εμβαδού του αναγνωρισμένου αντικειμένου εξάγουμε τη μεταβολή μεγέθους του με μεγαλύτερη ακρίβεια σε δύσκολες συνθήκες περιβάλλοντος και φωτισμού (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §2.3 του Παραρτήματος Β).

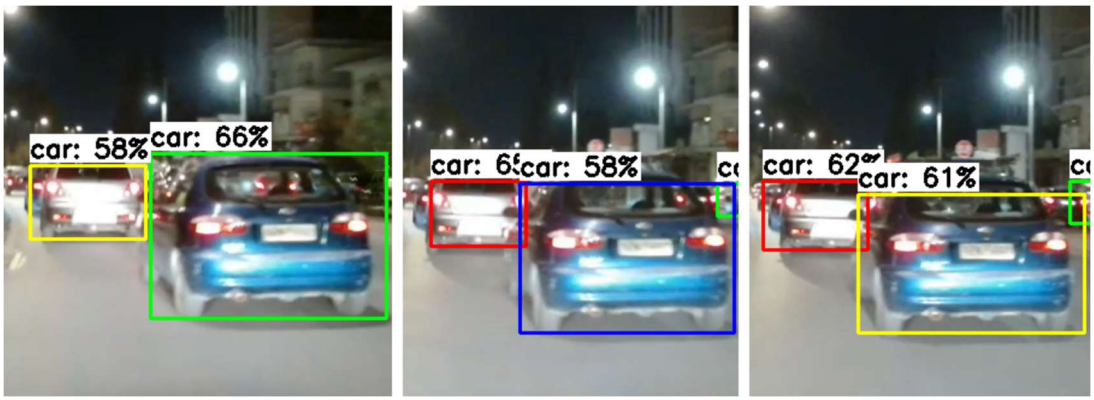


Εικόνα 3.22 : Παράδειγμα δυο διαδοχικών στιγμιότυπων προειδοποίησης του οδηγού για επικείμενη σύγκρουση με χρήση τεσσάρων σημαντικών σημείων

2. Αναγνώριση αλλαγής σημαντικότερου οχήματος

Το πρόβλημα που προκύπτει όταν γίνεται αλλαγή του σημαντικότερου οχήματος περιγράφεται με το εξής παράδειγμα. Έστω ότι το Σύστημα του οχήματός μας παρακολουθεί ένα προπορευόμενο όχημα που κινείται στην ιδιά λωρίδα με εμάς διατηρώντας την ίδια ταχύτητα και απόσταση από το όχημά μας. Όπως προαναφέρθηκε, το υποσύστημα προειδοποίησης μπροστινής σύγκρουσης καταγράφει το μέγεθος του προπορευόμενου οχήματος ώστε να αναγνωριστεί επικείμενη σύγκρουση με αυτό. Έστω τώρα ότι ένα νέο όχημα, που κινείται με την ίδια ταχύτητα με το όχημά μας, εισέρχεται στην λωρίδα μας ανάμεσα στο αρχικό προπορευόμενο όχημα και στο όχημά μας. Λόγω του ότι το νέο όχημα βρίσκεται σε μικρότερη απόσταση από το όχημά μας σε σχέση με το αρχικό προπορευόμενο όχημα, το υποσύστημα καταγράφει μια απότομη αύξηση εμβαδού του σημαντικού «κουτιού» και επειδή δεν γνωρίζει ότι έγινε αλλαγή του σημαντικότερου οχήματος υποθέτει ότι το αρχικό όχημα εκτέλεσε απότομη πέδηση οπότε προειδοποιεί λανθασμένα τον οδηγό για επικείμενη σύγκρουση.

Η μέθοδος που ακολουθείται για να αναγνωριστεί ότι έγινε αλλαγή του σημαντικότερου οχήματος είναι να καταγράφεται και η θέση του οχήματος στην προοπτική μας (εικόνα) και εάν η θέση αυτή αλλάζει αισθητά μεταξύ δυο συνεχομένων στιγμιότυπων τότε το «σημαντικότερο» αυτό όχημα θεωρείται νέο και δεν συγκρίνεται το μέγεθος του με το μέγεθος του προηγούμενου στιγμιότυπου. Με χρήση της μεθόδου αυτής, στο παραπάνω παράδειγμα το όχημα που εισέρχεται αναμεσά στο αρχικό όχημα και στο όχημά μας έχει πολύ διαφορετική θέση σε σχέση με το αρχικό οπότε θεωρείται νέο και το μέγεθος του για ένα στιγμιότυπο δεν συγκρίνεται με το μέγεθος του αρχικού οχήματος. Ωστόσο στη συνέχεια καθώς το νέο όχημα εισέρχεται πλήρως στην λωρίδα κυκλοφορίας μας η μεταβολή θέσης του είναι μικρή και ενεργοποιείται ξανά το υποσύστημα προειδοποίησης μπροστινής σύγκρουσης, συγκρίνοντας κατά τα γνωστά το μέγεθος του με το μέγεθος του εκάστοτε προηγούμενου στιγμιότυπου (Εικόνα 3.23) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §2.2 του Παραρτήματος Β).

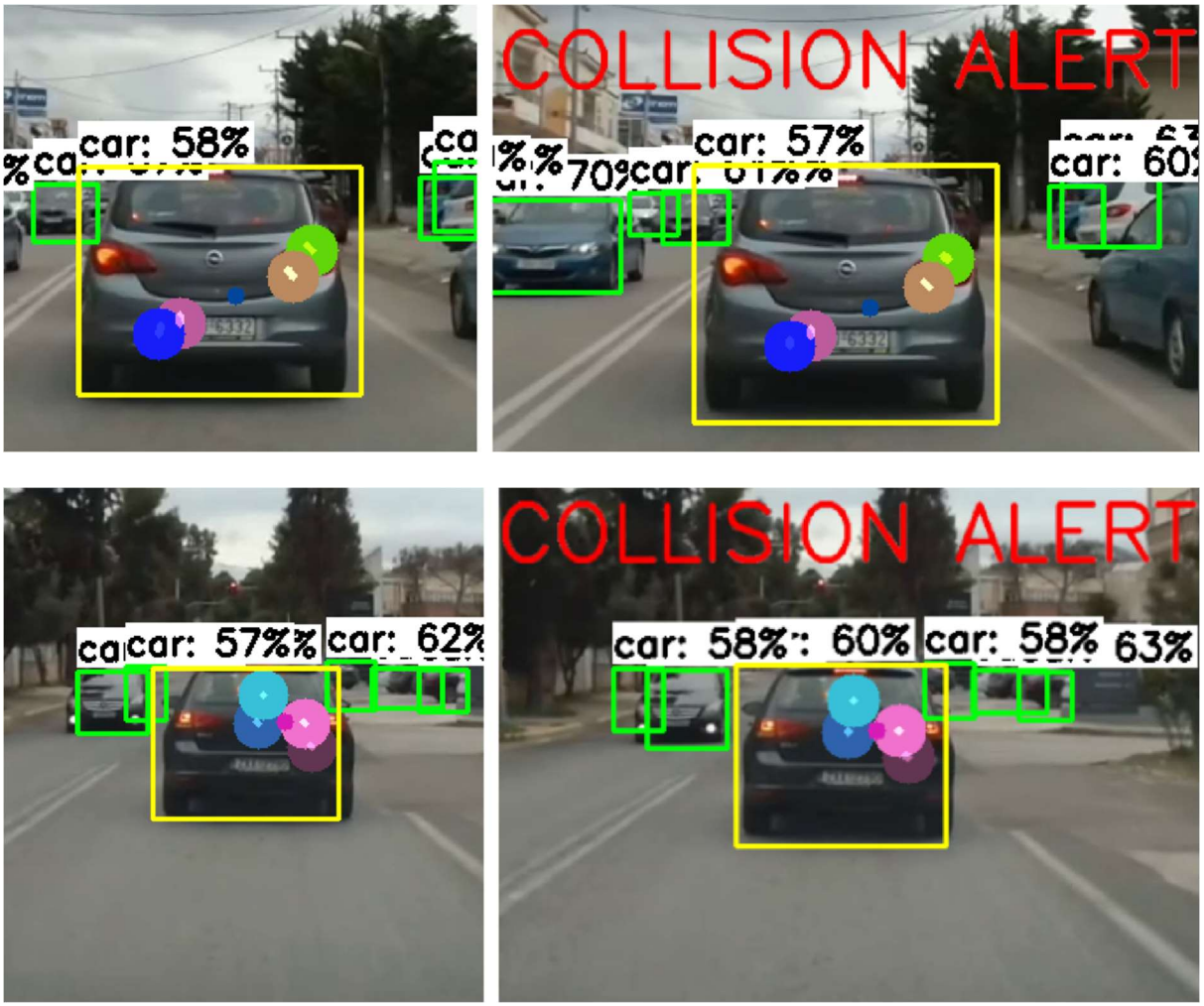


Εικόνα 3.23: Παράδειγμα τριών διαδοχικών στιγμιότυπων αλλαγής σημαντικότερου οχήματος. Αρχικά το γκρι αυτοκίνητο ήταν το σημαντικότερο, στην συνέχεια το μπλε μπήκε ανάμεσα, λόγω της μεγάλης μετατόπισης κέντρου έγινε αναγνώριση της αλλαγής αυτής και έγινε εκ νέου έναρξη σύγκρισης μεγέθους

3. Απόρριψη αντικειμένων που εφάπτονται στα άκρα της εικόνας

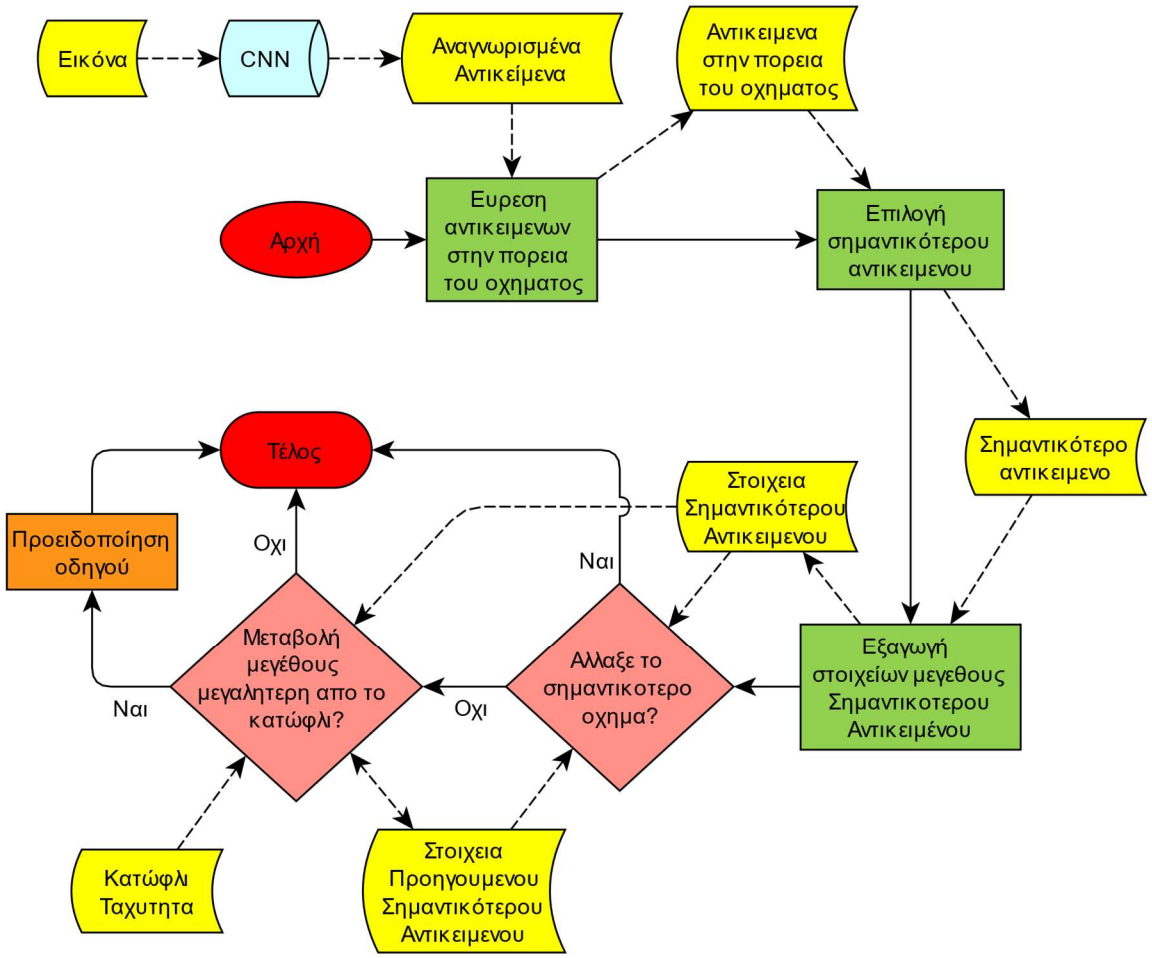
Όταν οι πλευρές του οριοθετημένου «κουτιού» ενός αντικειμένου εφάπτονται στις άκρες της εικόνας τότε τα αντικείμενα αγνοούνται διότι δεν μπορούμε να κρίνουμε αξιόπιστα το μέγεθος τους.

Παραδείγματα λειτουργίας του υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης



Εικόνα 3.24: Παραδείγματα δυο διαδοχικών στιγμιότυπων προειδοποίησης του οδηγού για επικείμενη μπροστινή σύγκρουση με χρήση τεσσάρων σημαντικών σημείων

Η **τελική δομή** του αλγορίθμου του υποσυστήματος είναι η εξής:



Σχήμα 3.13: Δομή του υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης

3.5 Υποσύστημα προειδοποίησης εκκίνησης προπορευόμενου οχήματος

Στην παράγραφο αυτή, όπως και στην προηγούμενη, αρχικά αναφέρονται τα βήματα που ακολουθούνται σε κάθε στιγμιότυπο (εικόνα) για την επιλογή του σημαντικότερου αντικειμένου που πρέπει να παρακολουθείται για να εντοπιστεί πιθανή εκκίνησή του. Στη συνέχεια αναφέρονται τα βήματα που ακολουθούνται προκειμένου να εξασφαλίζεται η αξιόπιστη λειτουργία του υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος σε βάθος χρόνου, κατά τη διαδοχική λήψη στιγμιότυπων (εικόνων). Τα αναφερόμενα, πέραν των οχημάτων, ισχύουν και για τις περιπτώσεις εκκίνησης οποιουδήποτε άλλου προπορευόμενου αντικειμένου π.χ. πεζού, ποδηλάτου ή ιππέα.

3.5.1 Βήματα ανά στιγμιότυπο (Εικόνα):

Τα βήματα είναι παρόμοια με τα αντίστοιχα βήματα του υποσυστήματος υποβοήθησης προειδοποίησης μπροστινής σύγκρουσης (παράγραφος 4 του παρόντος Κεφαλαίου).

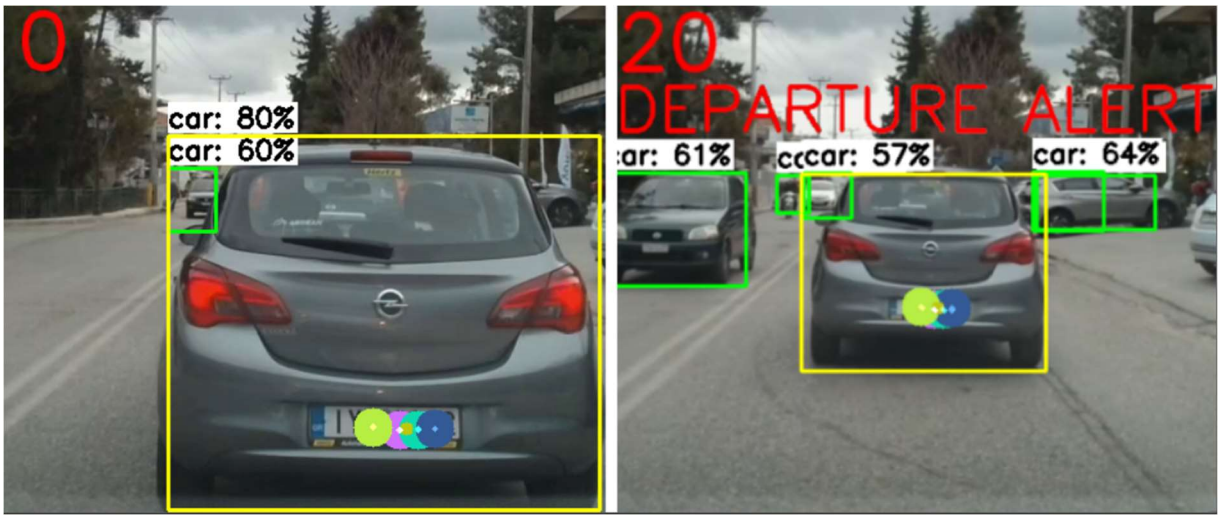
3.5.2 Βήματα σε Βάθος χρόνου:

Παρακολούθηση μεγέθους και προειδοποίηση

Εφόσον το όχημά μας είναι σταματημένο και το μέγεθος του προπορευόμενου σημαντικότερου οχήματος είναι αρκετά μεγάλο (δηλαδή είναι κοντά) τότε γίνεται ενεργοποίηση του υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος.

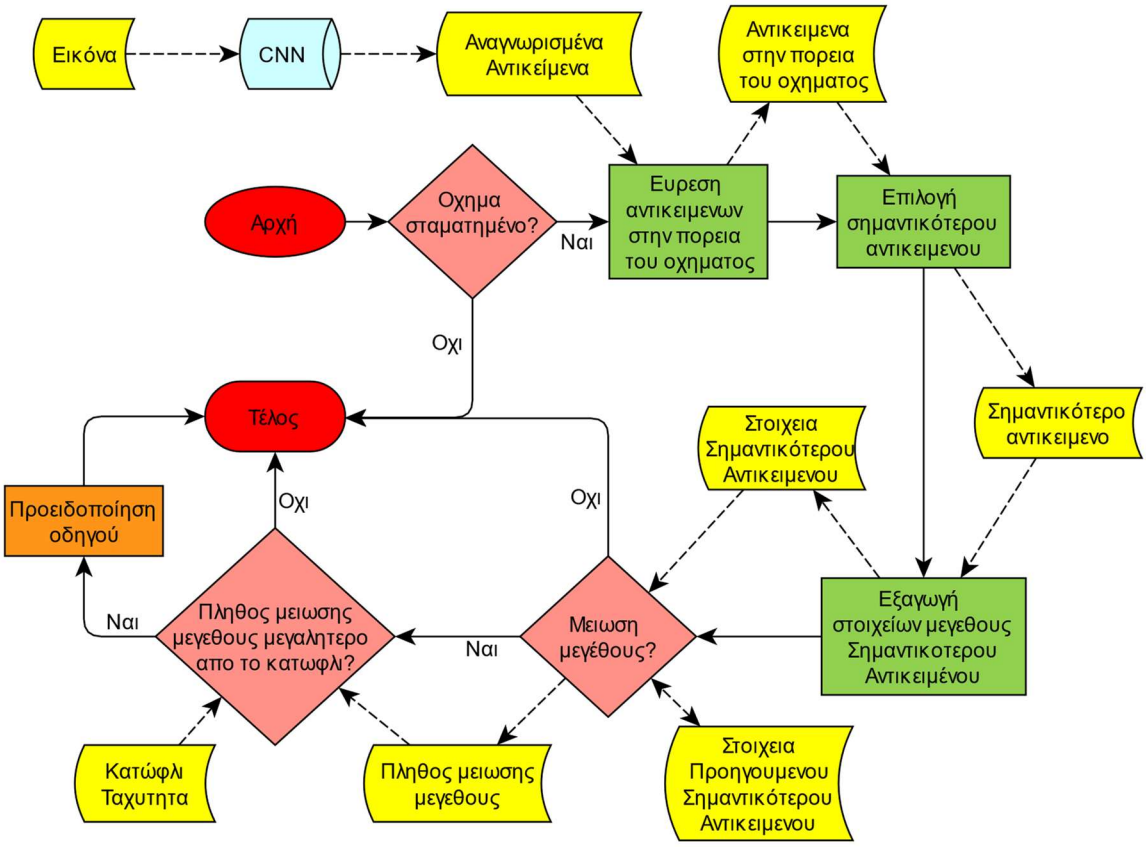
Σε κάθε στιγμιότυπο γίνεται εύρεση της μέσης απόστασης μεταξύ σημαντικών σημείων του (όπως εξηγήθηκε στην παράγραφο 3.4.2.1) και συγκρίνεται με την αντίστοιχη μέση απόσταση του προηγουμένου στιγμιότυπου. Όταν το μέγεθος του προπορευόμενου οχήματος (απόσταση μεταξύ σημαντικών σημείων του) μειωθεί, γίνεται έναρξη ελέγχου του μεγέθους (εφόσον το προπορευόμενο όχημα απομακρύνεται το μέγεθός του μειώνεται). Αν αυτό συνεχιστεί για δύο (2) δευτερόλεπτα και εφόσον το όχημα μας παραμένει ακίνητο γίνεται προειδοποίηση του οδηγού για την εκκίνηση του προπορευόμενου οχήματος (Εικόνα 3.25) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §3.1 του Παραρτήματος Β).

Παραδείγματα λειτουργίας του υποσυστήματος εκκίνησης προπορευόμενου οχήματος



Εικόνα 3.25: Παραδείγματα δυο διαδοχικών στιγμιότυπων προειδοποίησης του οδηγού για εκκίνηση προπορευόμενου οχήματος με χρήση τεσσάρων σημαντικών σημείων

Η **τελική δομή** του αλγορίθμου του υποσυστήματος είναι η εξής:



Σχήμα 3.14: Διάγραμμα δομής του υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος

3.6 Υποσύστημα προειδοποίησης παραβίασης ερυθρού σηματοδότη

Στην παράγραφο αυτή αρχικά αναφέρονται τα βήματα που ακολουθούνται σε κάθε στιγμιότυπο (εικόνα) που λαμβάνεται από το Σύστημα (βήματα ανά στιγμιότυπο) για την επιλογή του σημαντικότερου σηματοδότη (σε περίπτωση ύπαρξης περισσότερων του ενός), δηλαδή του σηματοδότη ο οποίος πρέπει να παρακολουθείται από το σύστημα. Στη συνέχεια αναφέρονται τα βήματα που ακολουθούνται προκειμένου να εξασφαλίζεται η αξιόπιστη λειτουργία του υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη σε βάθος χρόνου, κατά τη διαδοχική λήψη στιγμιότυπων (εικόνων).

3.6.1 Βήματα ανά στιγμιότυπο (Εικόνα):

Στην Ελλάδα και σε άλλες χώρες συναντάμε συνήθως δυο είδη σηματοδοτών, τους μεμονωμένους και τους διπλούς (Εικόνα 3.26). Οι διπλοί σηματοδότες αποτελούνται από δυο μεμονωμένους και συνήθως ο ένας αφορά τα οχήματα που συνεχίζουν στην ίδια πορεία και ο άλλος αφορά τα οχήματα που θέλουν να αλλάξουν πορεία.

(α) (β)

Εικόνα 3.26 : (α) Παράδειγμα μεμονωμένου (β) Παράδειγμα διπλού σηματοδότη (δυο σηματοδοτών που αφορούν διαφορετικές κατευθύνσεις)

3.6.1.1 Μεμονωμένοι σηματοδότες

1. **Οριοθετημένα «κουτιά» σηματοδότη**

   Από την ανίχνευση αντικειμένων στην εικόνα παίρνουμε πολλές συντεταγμένες «κουτιών» που ορίζουν τις θέσεις και διαστάσεις αντικειμένων που πιθανότατα εντάσσονται στην κατηγορία σηματοδότες (Εικόνα 3.27) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.1 του Παραρτήματος Β).

Εικόνα 3.27 : Παράδειγμα αναγνώρισης σηματοδότη σε μια εικόνα

2. Αποκοπή τμήματος εικόνας

Στη συνέχεια γίνεται αποκοπή από την εικόνα ενός τμήματός της, το οποίο περιέχει στο κέντρο του το «κουτί» του σηματοδότη και είναι διπλάσιο σε διαστάσεις από αυτό. Αυτό γίνεται διότι λόγω της χαμηλής ανάλυσης του νευρωνικού δικτύου και των μικρών διαστάσεων των σηματοδοτών εμφανίζεται συχνά αστοχία του συστήματος όσον αφορά στην τοποθέτηση «κουτιού» σε μικρών διαστάσεων αντικείμενα. (Εικόνα 3.28) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.1 του Παραρτήματος Β).

Εικόνα 3.28 : Παράδειγμα αποκομμένου τμήματος εικόνας που περιέχει σηματοδότη

3. Δημιουργία εικόνων φίλτρου ανά χρώμα

Το τμήμα αυτό περνά από ένα πράσινο και από ένα κόκκινο φίλτρο για να βρεθεί το χρώμα του ενεργού λαμπτήρα του σηματοδότη. Στην Εικόνα 3.29 εμφανίζονται τα αποτελέσματα του φιλτραρίσματος του σηματοδότη της εικόνας 3.42 που έχει πράσινο χρώμα με τα δύο φίλτρα. (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.1 του Παραρτήματος Β).

Εικόνα 3.29: Παράδειγμα πράσινου (αριστερά) και κόκκινου (δεξιά) φίλτρου

4. **Εύρεση χρώματος σηματοδότη**

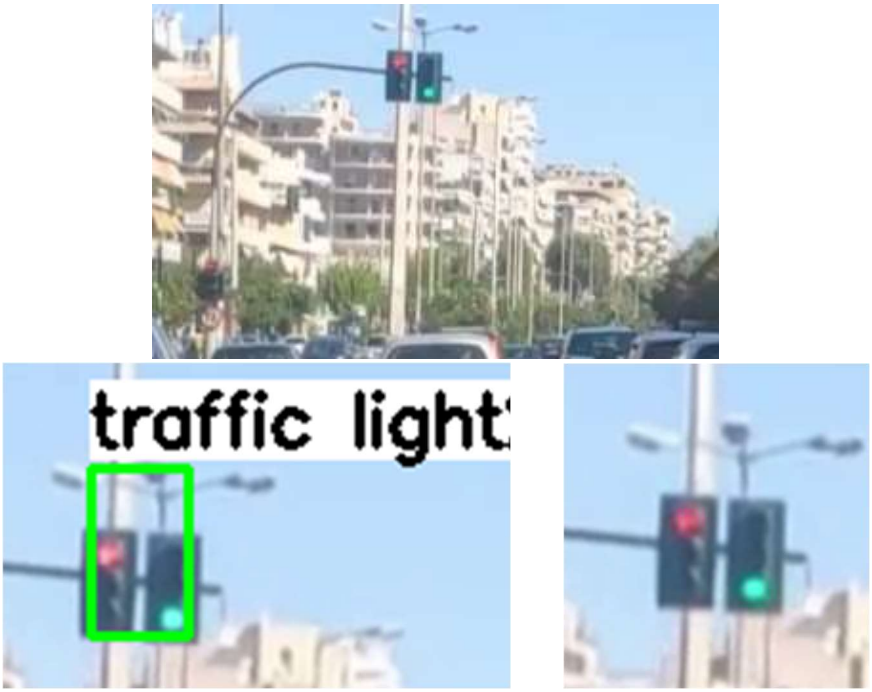
   Μετά από άθροιση των δυαδικών ψηφίων (0, 1) των αποτελεσμάτων του φιλτραρίσματος προκύπτουν δύο τιμές οι οποίες συγκρίνονται και η μεγαλύτερη δηλώνει το χρώμα του σηματοδότη, όπως φαίνεται στην εικόνα 3.30. (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.2 του Παραρτήματος Β).

Εικόνα 3.30 : Παράδειγμα αποτελέσματος ανίχνευσης χρώματος σηματοδότη

3.6.1.2 Διπλοί σηματοδότες

Στους δρόμους συναντώνται τρεις διαφορετικές καταστάσεις διπλών σηματοδοτών: Είτε και οι δύο είναι πράσινου χρώματος, είτε και οι δύο είναι κόκκινου χρώματος είτε ο ένας είναι κόκκινου χρώματος και ο άλλος πράσινου χρώματος. Στις πρώτες δύο καταστάσεις όπου τα χρώματα ταυτίζονται ο διπλός σηματοδότης λαμβάνεται υπόψη πρακτικά ως μεμονωμένος. Ωστόσο στην τρίτη κατάσταση, όπου ο ένας είναι κόκκινου χρώματος και ο άλλος πράσινου χρώματος, το σύστημα θα πρέπει να λαμβάνει διαφορετική απόφαση αναλόγως του εάν ο οδηγός έχει ενεργοποιήσει ή όχι τα φώτα δείκτη κατεύθυνσης (φλας).

Ας υποθέσουμε καταρχήν ότι δεν ξέρουμε το είδος του αναγνωρισμένου σηματοδότη της Εικόνας 3.31.



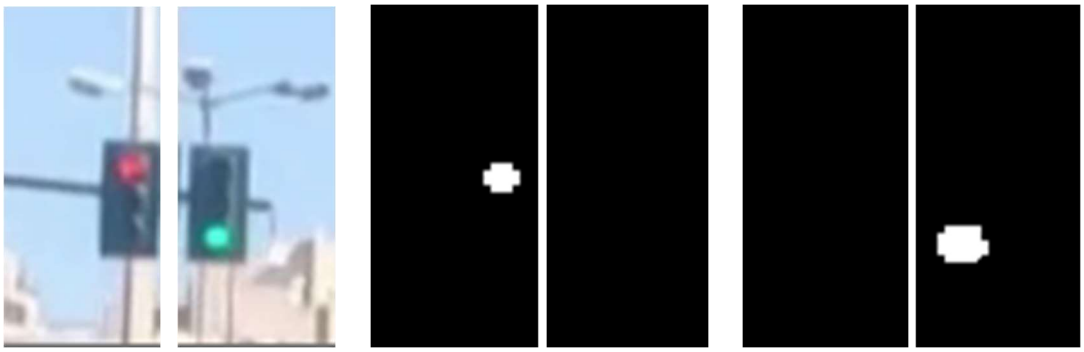
Εικόνα 3.31 : Παράδειγμα διπλού σηματοδότη

Στην αρχή από την ανίχνευση αντικειμένων στην εικόνα παίρνουμε συντεταγμένες «κουτιών» σηματοδοτών (βήμα 1 της παραγράφου 3.6.1.1), στη συνέχεια παίρνουμε το αποκομμένο τμήμα εικόνας (βήμα 2 της παραγράφου 3.6.1.1), το οποίο είναι αρκετά μεγαλύτερο για να περιέχει το διπλό σηματοδότη, και το περνάμε από τα δυο φίλτρα πράσινου και κόκκινου χρώματος (βήμα 3 της παραγράφου 3.6.1.1 - Εικόνα 3.32)

Εικόνα 3.32 : Παράδειγμα κόκκινου (αριστερά) και πράσινου (δεξιά) φίλτρου σε διπλό σηματοδότη

Όταν τα αποτελέσματα είναι και τα δυο θετικά, δηλαδή υπάρχουν και τα δυο χρώματα, τότε πρόκειται για διπλό σηματοδότη, όπου ο ένας είναι κόκκινου χρώματος και ο άλλος πράσινου χρώματος, δηλαδή πρόκειται για την περίπτωση που διαφέρει από την περίπτωση μεμονωμένου σηματοδότη.

Στη συνέχεια διαχωρίζουμε τις εικόνες των φίλτρων στη μέση, κάτι που αντιστοιχεί στο διαχωρισμό του διπλού σηματοδότη σε δύο μεμονωμένους, οπότε για κάθε μεμονωμένο σηματοδότη έχουμε δύο υποφίλτρα που αντιστοιχούν στο πράσινο και στο κόκκινο φίλτρο (Εικόνα 3.33).



Εικόνα 3.33: Παράδειγμα υποτμημάτων αρχικού τμήματος (αριστερά), υποτμημάτων κόκκινου φίλτρου (μέση) και υποτμημάτων πράσινου (δεξιά) φίλτρου σε διπλό σηματοδότη

Μετά από άθροιση των δυαδικών ψηφίων (0, 1) των αποτελεσμάτων του φιλτραρίσματος προκύπτουν δύο τιμές για κάθε μεμονωμένο σηματοδότη οι οποίες συγκρίνονται μεταξύ τους και η μεγαλύτερη δηλώνει το χρώμα του κάθε σηματοδότη, οπότε αναγνωρίζεται το χρώμα του σηματοδότη ανά κατεύθυνση (Εικόνα 3.34). Στη συνέχεια αν έχουν ενεργοποιηθεί τα φώτα δείκτη κατεύθυνσης (φλας) τότε το σύστημα θα λάβει το διπλό σηματοδότη ως κόκκινο ενώ αν δεν έχουν ενεργοποιηθεί τα φώτα δείκτη κατεύθυνσης (φλας) ως πράσινο. (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.3 του Παραρτήματος Β).

Εικόνα 3.34 : Παράδειγμα αποτελέσματος ανίχνευσης διπλού σηματοδότη

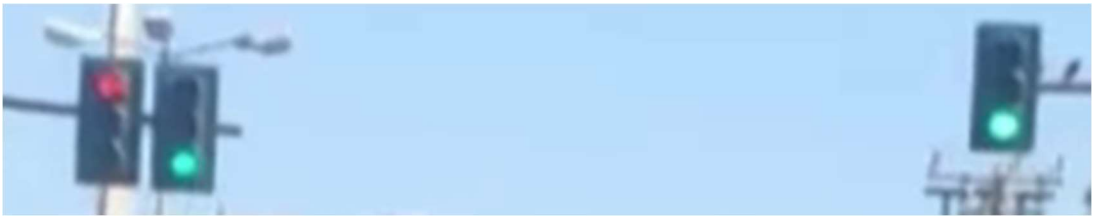
3.6.1.3 Επιλογή σημαντικότερου σηματοδότη

Στην ιδιά οπτική μπορεί να υπάρχουν πολλοί σηματοδότες που μπορεί να έχουν διαφορετικές σημάνσεις (κόκκινο, πράσινο). Για το λόγο αυτό γίνεται η επιλογή ότι ο σημαντικότερος σηματοδότης είναι αυτός που βρίσκεται πιο κοντά στο όχημά μας. (Εικόνα 3.35)

Εικόνα 3.35 : Παράδειγμα διαφορετικής σήμανσης σηματοδοτών στην ίδια οπτική

Πιο συγκεκριμένα ο σημαντικότερος σηματοδότης είναι ο πιο κοντινός, δηλαδή αυτός του οποίου, με χρήση προοπτικής, οι συντεταγμένες της πάνω πλευράς του «κουτιού» του είναι ψηλότερα από τις αντίστοιχες συντεταγμένες των πάνω πλευρών των «κουτιών» όλων των υπολοίπων σηματοδοτών.

Ωστόσο ένας διπλός σηματοδότης μας δίνει πληροφορία χρώματος και σε περίπτωση ευθείας πορείας και σε περίπτωση αλλαγής πορείας σε αντίθεση με έναν μεμονωμένο σηματοδότη.
Για το λόγο αυτό, στην περίπτωση που ένας διπλός σηματοδότης είναι σε παρόμοια απόσταση σε σχέση με έναν μεμονωμένο σηματοδότη τότε θα τίθεται ως σημαντικότερος ακόμα και αν ο μεμονωμένος λόγω προοπτικής φαίνεται λίγο πιο κοντά. (Εικόνα 3.36) (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.4 του Παραρτήματος Β).



Εικόνα 3.36 : Ο διπλός σηματοδότης αριστερά πρέπει να είναι σημαντικότερος του μεμονωμένου δεξιά πάρα το ότι ο μεμονωμένος φαίνεται πιο κοντά λόγω προοπτικής

3.6.2 Βήματα σε Βάθος χρόνου:

- **Προειδοποίηση οδηγού σε έλλειψη πέδησης**

  Όταν γίνει αναγνώριση κόκκινου σημαντικού σηματοδότη τότε το υποσύστημα ξεκινά τον έλεγχο της ταχύτητας του οχήματος. Αν το όχημα πλησιάσει αρκετά κοντά στον σημαντικό κόκκινο σηματοδότη, δηλαδή αν ο ερυθρός σηματοδότης είναι στα όρια εξόδου από το οπτικό πεδίο του Συστήματος και ταυτόχρονα υπάρχει έλλειψη πέδησης, τότε γίνεται συνεχώς προειδοποίηση του οδηγού έως ότου το όχημα περάσει τον σηματοδότη ή έως ότου ο οδηγός προβεί σε πέδηση του οχήματος (όποιο συμβεί πρώτο). (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.5 του Παραρτήματος Β).

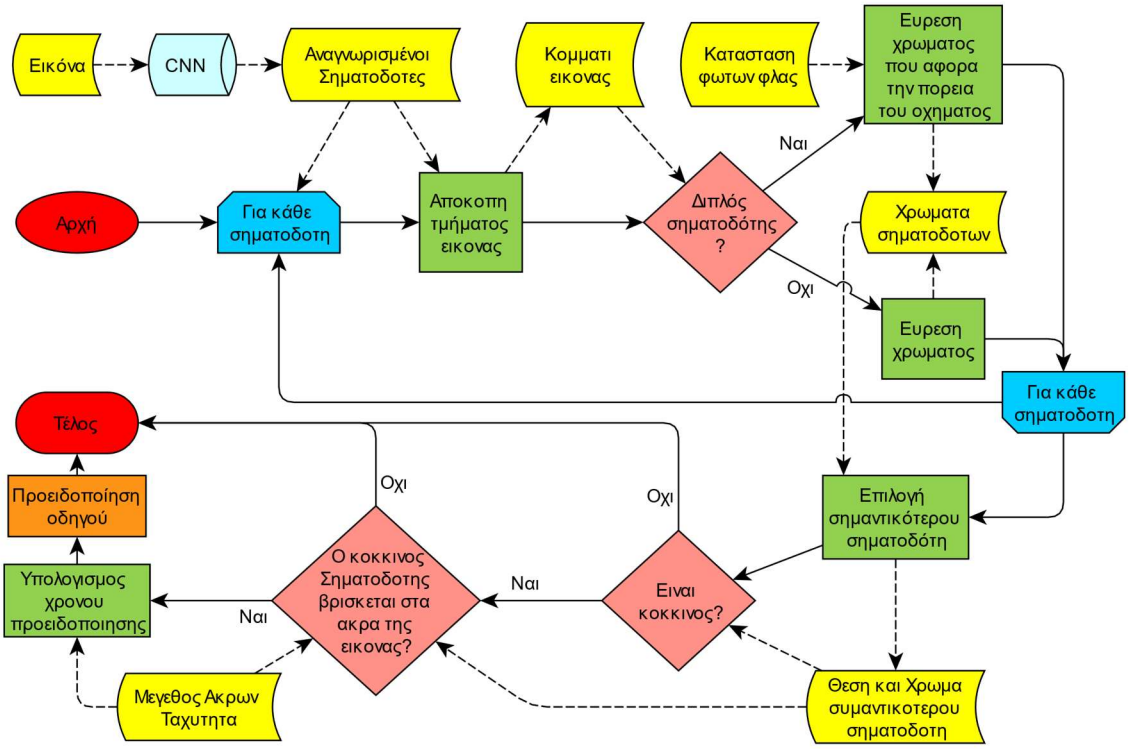
- **Καθυστέρηση ενεργοποίησης προειδοποίησης**

  Για την αποφυγή προειδοποίησης του οδηγού σε κάθε στιγμιότυπο που το υποσύστημα αναγνωρίζει κόκκινο σηματοδότη και έλλειψη πέδησης, έχει τεθεί μια καθυστέρηση στην ενεργοποίηση της προειδοποίησης. Πιο συγκεκριμένα το υποσύστημα θα πρέπει να αναγνωρίσει κόκκινο σηματοδότη για μερικά συνεχόμενα στιγμιότυπα (εικόνες) ώστε να ενεργοποιηθεί η προειδοποίηση και να μην αναγνωρίσει για αλλά τόσα για να απενεργοποιηθεί η προειδοποίηση (Σχετικό απόσπασμα του κώδικα αναφέρεται στην §4.6 του Παραρτήματος Β).

Παραδείγματα λειτουργίας του υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη

Εικόνα 3.37: Παραδείγματα δυο διαδοχικών στιγμιότυπων προειδοποίησης του οδηγού για επικείμενη παραβίαση ερυθρού σηματοδότη

Η **τελική δομή** του αλγορίθμου του υποσυστήματος είναι η εξής:



Σχήμα 3.15: Διάγραμμα δομής του υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη

Κεφάλαιο 4

Υλικό και Τοποθέτηση

Στο Κεφάλαιο 4 αναφέρονται τα επιμέρους υλικά που χρησιμοποιούνται για τη δόμηση του Συστήματος, το σχετικό κόστος προμήθειάς τους καθώς και ο τρόπος τοποθέτησης τους σε αυτοκίνητο.

4.1 Προαπαιτούμενα υλικά

4.1.1 Υπολογιστής Raspberry Pi

Το Raspberry Pi είναι ένας πλήρης υπολογιστής σε μέγεθος πιστωτικής κάρτας με χαμηλό κόστος που αναπτύχθηκε στο Ηνωμένο Βασίλειο. Μπορεί να συνδεθεί με οποιαδήποτε οθόνη και με τη χρήση πληκτρολογίου και ποντικιού να γίνει ένας τυπικός υπολογιστής.

Εικόνα 4.1: Υπολογιστής Raspberry Pi

Επιπλέον δίνει τη δυνατότητα σε άτομα που επιθυμούν να πειραματιστούν σε γλώσσες, όπως το Scratch και την Python, να τον μετατρέψουν σε μια επεξεργαστική μονάδα κατάλληλη για πολλές είδους κατασκευές και εργασίες.
Αυτή η ευελιξία οφείλεται στο λειτουργικό σύστημά του, που ονομάζεται Raspberry Pi OS και στη δυνατότητα του να συνδέεται με πολλούς αισθητήρες και συσκευές μέσω των υποδοχών GPIO που διαθέτει.

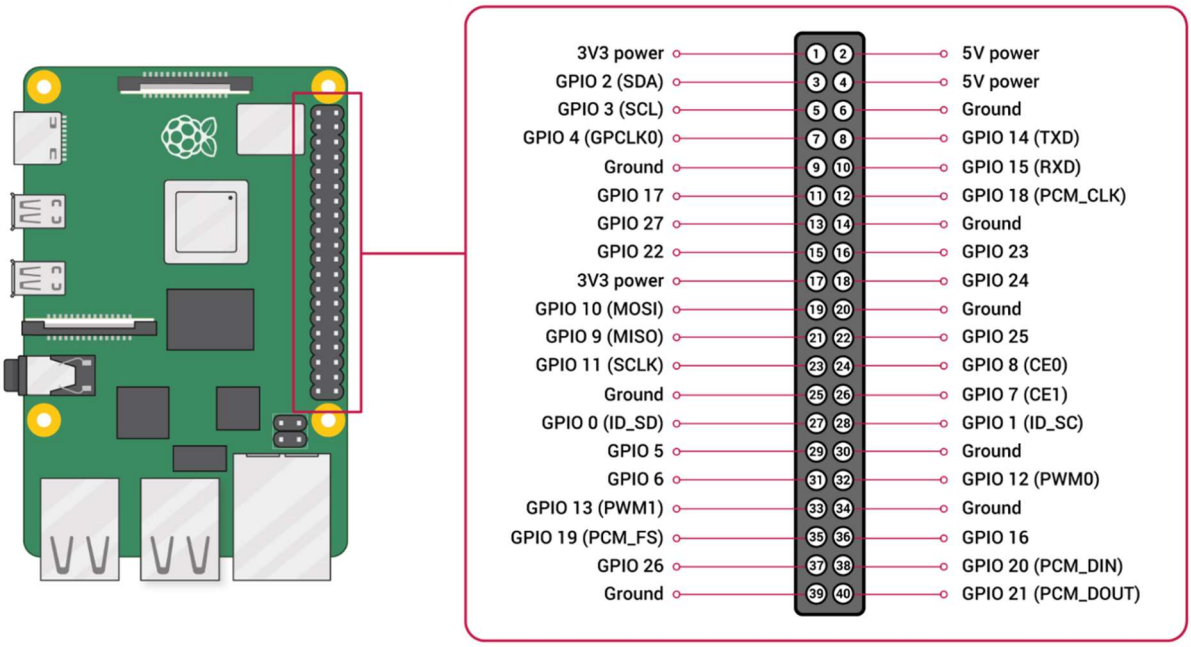
Λογισμικό
Το **Raspberry Pi OS** είναι ένα δωρεάν προσφερόμενο λειτουργικό σύστημα που βασίζεται στο ανοιχτού κώδικα λειτουργικό σύστημα Line και πιο συγκεκριμένα στο

Debian Linux. Είναι ειδικά σχεδιασμένο για βέλτιστη ταχύτητα σε επεξεργαστές αρχιτεκτονικής ARM.

Υλικό

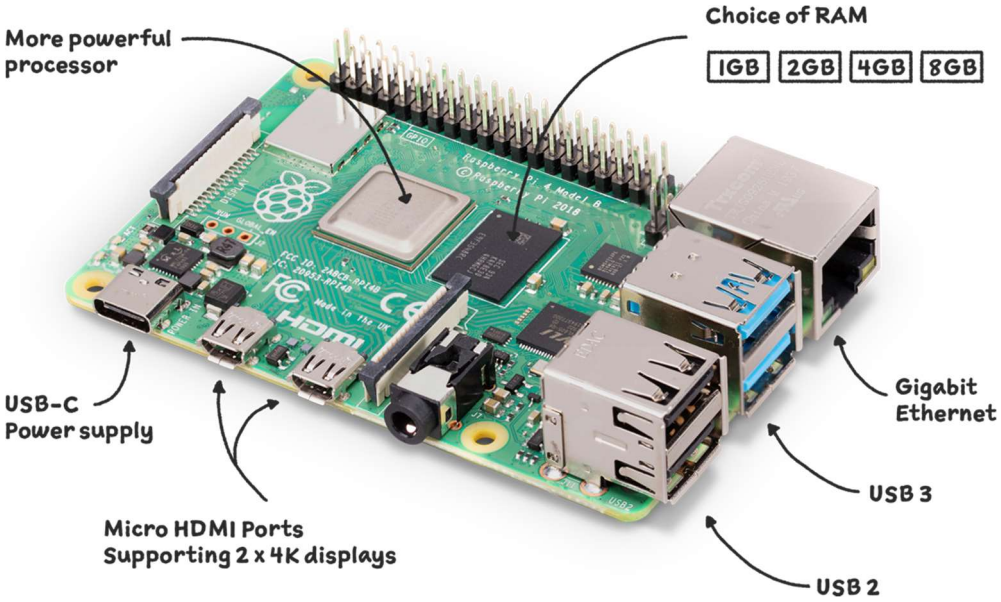
Το Raspberry Pi βασίζεται σε επεξεργαστή αρχιτεκτονικής **ARM** που είναι ενεργειακά αποδοτικότερη, σε σχέση με άλλες αρχιτεκτονικές, και διαθέτει όλες τις συνήθεις θύρες σύνδεσης ενός τυπικού υπολογιστή. Η τροφοδοσία γίνεται από ένα απλό USB type-c, προσφέροντας ευελιξία και ασφάλεια στην παροχή ρεύματος. Ωστόσο ο κυριότερος λόγος ευελιξίας του Raspberry Pi είναι οι υποδοχές GPIO.

Οι υποδοχές **General-purpose input/output** (GPIO) είναι υποδοχές γενικού τύπου, κάθε μια από τις οποίες μπορεί να προσφέρει τροφοδοσία, γείωση, σήματα εκτέλεσης, σειριακή επικοινωνία και μπορεί να ρυθμιστεί προγραμματιστικά ώστε να στέλνει οποιουδήποτε είδους σήματα. Έτσι δίνεται η δυνατότητα να συνδέεται και να επικοινωνεί με οποιουδήποτε είδους συσκευές και αισθητήρες μέσω κατάλληλων πρωτοκόλλων και σημάτων.



Σχήμα 4.1 : Δυνατότητες των υποδοχών GPIO του Raspberry Pi

Στην παρούσα εργασία έγινε χρήση του **Raspberry Pi 4 model B,** το οποίο ενσωματώνει ένα Broadcom BCM2711, 4-πύρηνο επεξεργαστή αρχιτεκτονικής ARM (ARM v8) 64-bit SoC Cortex-A72 χρονισμένο στα 1.5GHz. Διαθέτει 2GB μνήμης RAM, 2 θύρες USB 2.0, 2 θύρες USB 3.0, Audio jack 3.5mm, Ethernet 10/100/1000 Mbit/s, WiFi 2.4 GHz and 5.0 GHz 802.11ac και Bluetooth 5.0.



Εικόνα 4.2: Το Raspberry Pi που χρησιμοποιήθηκε και επεξήγηση των θυρών του

4.1.2 Κάμερα

Ως κάμερα του Συστήματος μπορεί να χρησιμοποιηθεί η ειδική, για το Raspberry Pi, κάμερα η οποία μπορεί να ρυθμιστεί πλήρως από το χρήστη. Επιπλέον μπορεί να χρησιμοποιηθεί και οποιαδήποτε άλλη κάμερα που συνδέεται μέσω USB καλωδίου, προσφέροντας ευελιξία στην τοποθέτησή της και μειώνοντας τον όγκο που καταλαμβάνεται στην καμπίνα του αυτοκίνητου, ωστόσο η ποιότητα της εικόνας εξαρτάται από το κόστος της κάμερας.

Pi Camera

Η ειδική, για το Raspberry Pi, Pi Camera αποτελείται από μια πλακέτα μικρών διαστάσεων με έναν δεκτή φωτός στο κέντρο της, που συνδέεται με το Raspberry Pi μέσω ειδικού καλωδίου και θύρας. Το Raspberry Pi αναλαμβάνει σχεδόν όλη την επεξεργασία του σήματος, προσφέροντας έτσι πλήρη ελευθέρια στην παραμετροποίηση και στον τρόπο λειτουργίας της κάμερας.

Εικόνα 4.3 : Η Pi Camera συνδεδεμένη με το Raspberry Pi μέσω του ειδικού καλωδίου - θύρας

Για τον ίδιο λόγο προσφέρει καλή ποιότητα εικόνας και μεγαλύτερο ρυθμό δειγματοληψίας σε σχέση με αυτόνομες κάμερες που κάνουν επεξεργασία του σήματος οι ίδιες πριν στείλουν την εικόνα ολοκληρωμένη στον Raspberry Pi.

Ωστόσο, λόγω του ότι η πληροφορία γίνεται εικόνα αφού φτάσει στο Raspberry Pi, το σήμα είναι ευαίσθητο σε ηλεκτρονικές παρεμβολές και αν γίνει χρήση μεγάλου μήκους καλωδίου (π.χ. 1 μέτρου), για να εγκατασταθεί η κάμερα ξεχωριστά από το Raspberry Pi, οι παρεμβολές αυξάνονται.

Αν αντιθέτως η κάμερα εγκατασταθεί μαζί με το Raspberry Pi, εξαιτίας του αυξημένου απαιτούμενου συνολικού χώρου (υπολογιστής και κάμερα) μειώνονται οι διαθέσιμες θέσεις τοποθέτησης του συστήματος στην καμπίνα του αυτοκινήτου.

USB Camera

Μια κάμερα που συνδέεται στον Raspberry Pi μέσω USB είναι λιγότερο ευέλικτη στη ρύθμιση της λειτουργίας της και μπορεί, αναλόγως του κόστους της, να παρέχει χειρότερη, εφάμιλλη ή και καλύτερη ποιότητα εικόνας και ρυθμό δειγματοληψίας από την Pi Camera. Ωστόσο, λόγω του ότι η επεξεργασία του σήματος σε εικόνα γίνεται από την ίδια την κάμερα και λόγω του ότι το USB εξασφαλίζει ψηφιακή μεταφορά πληροφορίας μπορεί να γίνει χρήση ενός, μεγάλου μήκους, τυπικού καλωδίου USB και δεν εμφανίζονται παρεμβολές στην εικόνα. Έτσι δίνεται η δυνατότητα να τοποθετηθεί η κάμερα ξεχωριστά από το Raspberry Pi καταλαμβάνοντας λιγότερο χώρο και αυξάνοντας τις πιθανές θέσεις εγκατάστασης στην καμπίνα του αυτοκινήτου.

Εικόνα 4.4 : Παράδειγμα USB κάμερας

Στην παρούσα εργασία έγινε χρήση της USB κάμερας Logitech Webcam C920.

Εικόνα 4.5 : Η USB κάμερα της παρούσας εργασίας

4.1.3 Ανάγνωση ταχύτητας - επιτάχυνσης του αυτοκινήτου

Το Σύστημα θα πρέπει να γνωρίζει την ταχύτητα και την επιτάχυνση του αυτοκίνητου, ώστε να μπορούν τα υποσυστήματα υποβοήθησης που εξετάζονται να προειδοποιούν τον οδηγό κατάλληλα όπως αναφέρθηκε στο Κεφάλαιο 3. Η ταχύτητα μπορεί να αναγνωσθεί με δυο τρόπους, με την χρήση της θύρας OBDII του αυτοκινήτου και με αυτόνομο σύστημα GPS.

OBDII

Η θύρα OBDII (On-Board Diagnostics II) είναι μια διαγνωστική θύρα που διαθέτουν όλα τα αυτοκίνητα μετά το έτος 2001. Χρησιμοποιείται κυρίως για έλεγχο προβλημάτων του κινητήρα του αυτοκινήτου, μέσω επικοινωνίας με τον «εγκέφαλο» του. Η θύρα είναι συνήθως τοποθετημένη κάτω αριστερά από το τιμόνι του αυτοκίνητου.

Εικόνα 4.6 : Παράδειγμα θέσης θύρας OBDII στην καμπίνα αυτοκίνητου

Τα περισσότερα οχήματα, εκτός από κωδικούς προβλημάτων του κινητήρα, παρέχουν πολλές ακόμα πληροφορίες για την κατάσταση του κινητήρα αλλά και του οχήματος (π.χ. στροφές κινητήρα, ταχύτητα οχήματος, κατανάλωση καυσίμου και διάφορες τιμές αισθητήρων του κινητήρα). Στην περίπτωση μας το Σύστημα χρειάζεται μόνο την ταχύτητα και από την ταχύτητα μπορεί να υπολογιστεί και η επιτάχυνση. Μερικά

σύγχρονα αυτοκίνητα παρέχουν και δεδομένα επιτάχυνσης - επιβράδυνσης αλλά τα περισσότερα από αυτά διαθέτουν δικά τους συστήματα ADAS.

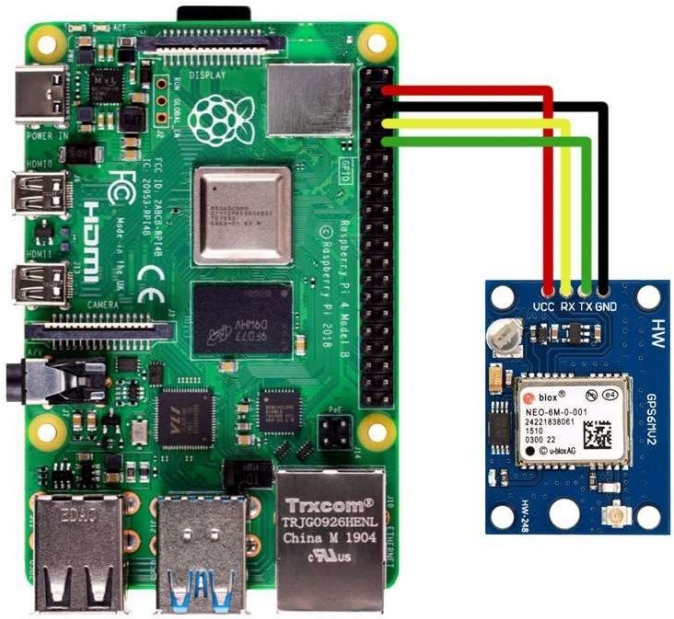
Η σύνδεση με το Raspberry Pi μπορεί να γίνει μέσω αντάπτορα ELM327, ο οποίος μετατρέπει την πληροφορία του «εγκεφάλου» του κινητήρα σε σειριακή ροή δεδομένων και στη συνέχεια μπορεί να τη στείλει στο Raspberry Pi μέσω καλωδίου USB ή ασύρματα μέσω Bluetooth.

Στην περίπτωση μας έγινε χρήση Bluetooth αντάπτορα ELM327, διότι λόγω της ασύρματης επικοινωνίας προσφέρει μεγαλύτερη ευελιξία στην επιλογή της θέσης εγκατάστασης του Raspberry Pi στην καμπίνα του αυτοκίνητου.

Εικόνα 4.7 : Ο Bluetooth αντάπτορα ELM327 που χρησιμοποιήθηκε

GPS Module

Στις περιπτώσεις που το όχημα δεν προσφέρει δεδομένα ταχύτητας μέσω OBDII μπορεί να γίνει χρήση ενός ξεχωριστού συστήματος GPS που θα συνδέεται και θα επικοινωνεί σειριακά με το Raspberry Pi μέσω των GPIO υποδοχών του.



Εικόνα 4.8 : Παράδειγμα σύνδεσης ενός GPS συστήματος με το Raspberry Pi

4.1.4 Αναγνώριση Φώτων Δείκτη Κατεύθυνσης (φλας)

Το Σύστημα θα πρέπει να γνωρίζει την κατάσταση των φώτων δείκτη κατεύθυνσης (φλας) του αυτοκίνητου, ώστε να μπορούν τα συστήματα υποβοήθησης να προειδοποιούν τον οδηγό κατάλληλα, όπως αναφέρθηκε στο Κεφάλαιο 3.

Υπάρχουν δύο τρόποι αναγνώρισης κατάστασης του Φώτων Δείκτη Κατεύθυνσης (φλας). Ο ένας είναι να γίνει απευθείας σύνδεση με το καλώδιο τροφοδοσίας των φωτών φλας μέσω level converter και ο άλλος είναι να διαβαστεί η θέση του μοχλού φλας μέσω Hall Effect αισθητήρα.

Level Converter

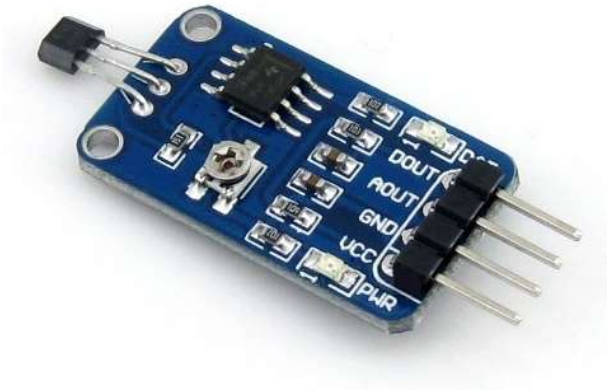
Το level converter είναι ένας μετατροπέας τάσης που στην περίπτωση μας θα συνδεθεί στην καλωδίωση των φωτών φλας και θα μετατρέψει τα 12V λειτουργίας των φώτων φλας σε 3.3V, ώστε να καταστεί δυνατό να διαβάσει το Raspberry Pi την κατάστασή τους μέσω των θυρών GPIO του Raspberry Pi. Με τη χρήση level converter επιτυγχάνεται σταθερή και αξιόπιστη απευθείας σύνδεση με το αυτοκίνητο αλλά απαιτείται μια αρκετά περίπλοκη διαδικασία εγκατάστασης και μπορεί να χρειαστεί μηχανικός για επιβεβαίωση της ορθής εγκατάστασης.

Εικόνα 4.9 : Παράδειγμα level converter 2 καναλιών

Αισθητήρας Hall Effect

Ο Hall Effect αισθητήρας είναι ένας αισθητήρας που δέχεται μια τάση και στη συνέχεια, βάσει του μαγνητικού πεδίου που βρίσκονται κοντά του μεταβάλλεται η τάση εξόδου του. Πιο συγκεκριμένα όσο πιο ισχυρό είναι το μαγνητικό πεδίο τόσο πιο μεγάλη η μεταβολή της τάσης εξόδου. Μπορούμε να εκμεταλλευτούμε την ιδιότητα αυτή τοποθετώντας τον αισθητήρα κάθετα στον άξονα κίνησης του μοχλού των φλας και με χρήση ενός μικρού μαγνήτη πάνω στο μοχλό μπορούμε να καταλάβουμε, μέσω της μεταβολής της έντασης του μαγνητικού πεδίου που δέχεται ο αισθητήρας, ότι γίνεται αντίστοιχα και μεταβολή της θέσης του μοχλού. Με ανάγνωση της τάσης εξόδου του αισθητήρα, μέσω των θυρών GPIO του Raspberry Pi μπορούμε να αναγνωρίσουμε τη θέση του μοχλού των φλας.

Ο τρόπος αναγνώρισης της κατάστασης του φώτων φλας με αισθητήρα Hall Effect είναι πιο ευαίσθητος σε παρεμβολές μαγνητικών πεδίων αλλά είναι πιο ασφαλής και δεν χρειάζεται επέμβαση στα ηλεκτρονικά του οχήματος. Για το λόγο αυτό είναι και ο τρόπος που χρησιμοποιήθηκε στην παρούσα εργασία.



Εικόνα 4.10 : Παράδειγμα αισθητήρα Hall Effect

4.1.5 Τροφοδοσία

Η τροφοδοσία του Raspberry Pi μπορεί να γίνει από ένα φορτιστή για κινητό τηλέφωνο, που συνδέεται στον αναπτήρα του αυτοκίνητου, ο οποίος υποστηρίζει πρωτόκολλο PD με 5V, 3A δηλαδή 15W και πάνω. Εναλλακτικά μπορεί να χρησιμοποιηθεί τροφοδοτικό που δέχεται 12V από την καλωδίωση του αυτοκίνητου και δίνει έξοδο τα 5V, 3A που χρειαζόμαστε.

Εικόνα 4.11 : Παράδειγμα τροφοδοτικού αναπτήρα (αριστερά) και τροφοδοτικού μέσω καλωδίωσης (δεξιά)

4.1.6 Ψύξη και Προστασία

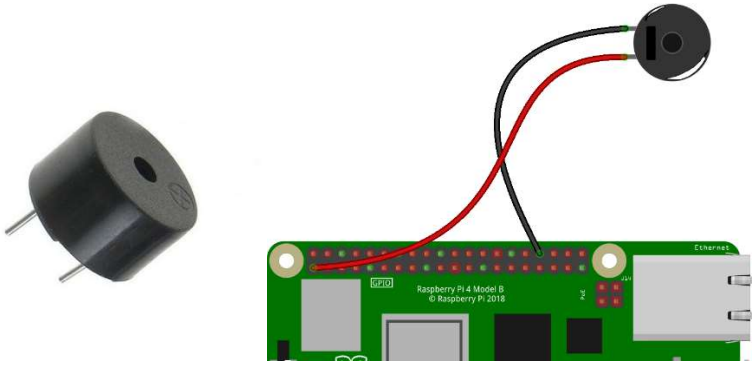
Οι θερμοκρασίες σε ένα αυτοκίνητο μπορούν να φτάσουν ακραίες τιμές, ειδικά το καλοκαίρι με ήλιο, γι’ αυτό το Raspberry Pi θα πρέπει να ψύχεται. Αυτό μπορεί να γίνει με τη χρήση μιας θήκης με ενσωματωμένο ανεμιστήρα ή θήκης από αλουμίνιο που λειτουργεί σαν ψήκτρα.

Εικόνα 4.12 : Παράδειγμα θήκης με ενσωματωμένο ανεμιστήρα (αριστερά) και θήκης από αλουμίνιο (δεξιά)

4.1.7 Προειδοποίηση του οδηγού

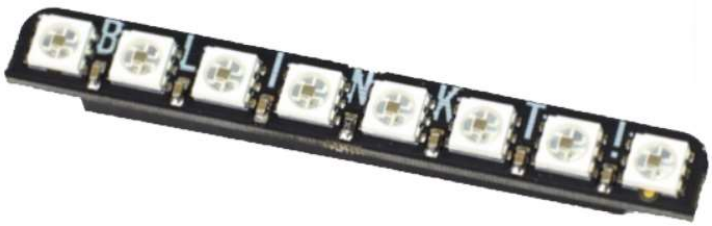
Η προειδοποίηση του οδηγού σε περίπτωση κίνδυνου γίνεται κυρίως μέσω της χρήσης ενός μικρού ηχείου, ενώ για λιγότερο επικίνδυνες καταστάσεις ή για παροχή πληροφοριών σχετικά με την κατάσταση του Συστήματος γίνεται χρήση φωτεινών ενδείξεων.

Για το ηχείο μπορεί να γίνει χρήση ενός μικρού buzzer που μπορεί τα τροφοδοτείται κατευθείαν από τις GPIO θύρες του Raspberry Pi χωρίς το σήμα να χρειάζεται επιπλέον ενίσχυση.



Εικόνα 4.13 : Παράδειγμα ηχείου τύπου buzzer και η σύνδεση του με τις GPIO θύρες του Raspberry Pi

Για τις φωτεινές ενδείξεις μπορεί να γίνει χρήση μιας λωρίδας από LED που τροφοδοτούνται και ελέγχονται κατευθείαν από τις GPIO θύρες του Raspberry Pi.



Εικόνα 4.14: Παράδειγμα φωτεινών ενδείξεων με χρήση λωρίδας με LED

4.2 Συνολικό Κόστος προμήθειας των εξαρτημάτων του Συστήματος

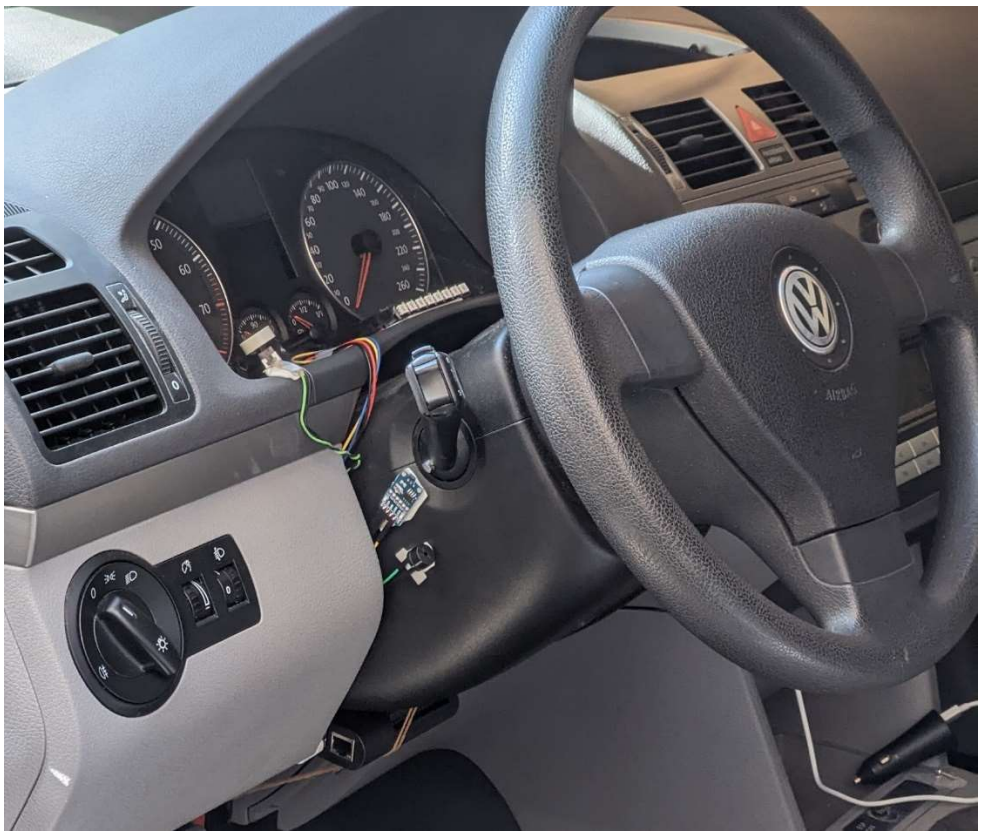
Τα κόστη των επιμέρους εξαρτημάτων του Συστήματος της παρούσας εργασίας (τιμές Δεκεμβρίου 2022) καθώς και το συνολικό κόστος του Συστήματος που υλοποιήθηκε έχουν ως εξής:

Raspberry Pi 4 (2GB) : 80€ **Κάμερα - USB :** 65€ (35€ σε περίπτωση χρήσης της **Pi Camera**) **Ανάγνωση ταχύτητας - OBDII :** 5€ (35€ σε περίπτωση χρήσης **GPS Module**) **Αναγνώριση Φώτων Φλας :** 5€ **Τροφοδοτικό :** 10€ **Θήκη ψύξης:** 7€ **Ηχείο :** 1€ (10€ σε περίπτωση χρήσης **ηχείου με ενισχυτή**) **Φωτεινές Ενδείξεις :** 7€
Συνολικό Κόστος Συστήματος: 180€

Σε περίπτωση που γίνει χρήση των ακριβότερων εξαρτημάτων το συνολικό κόστος του Συστήματος ανέρχεται περίπου στα 220€.

4.3 Τοποθέτηση του Συστήματος σε όχημα

Για τον έλεγχο της λειτουργίας του, το Σύστημα τοποθετήθηκε σε όχημα VW Touran 1.4 TSI, μοντέλο 2007, το οποίο λόγω ηλικίας δεν διαθέτει Σύστημα υποβοήθησης οδηγού.



Εικόνα 4.15 : Τοποθέτηση του Συστήματος σε όχημα VW Touran

4.3.1 Τοποθέτηση της κάμερας

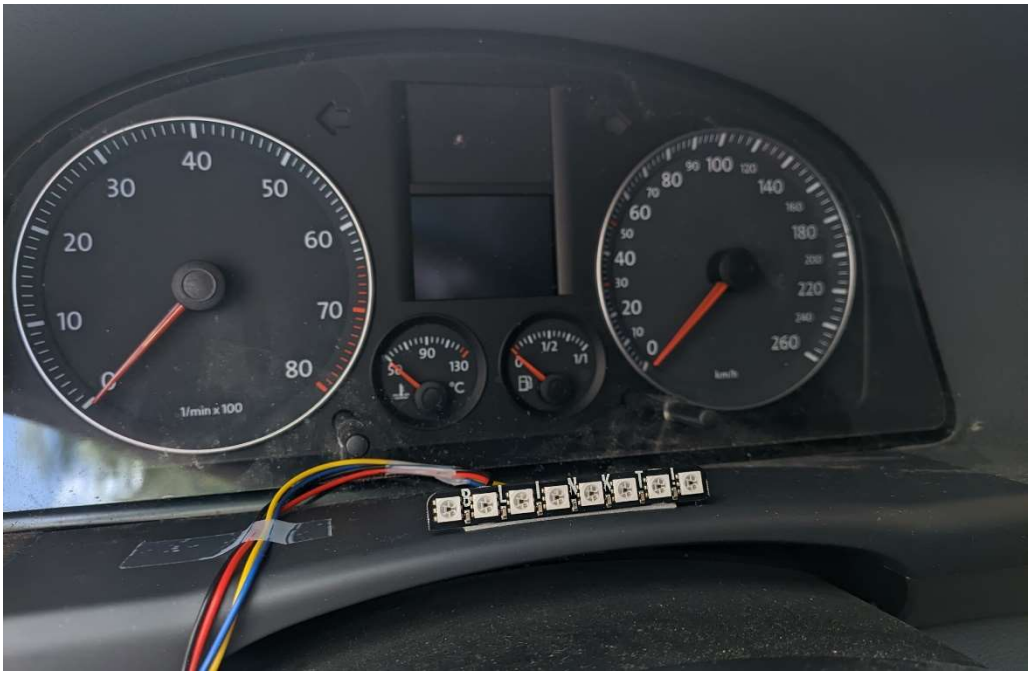
Η ιδανική θέση τοποθέτησης της κάμερας είναι πίσω από τον εσωτερικό μεσαίο καθρέφτη του οχήματος. Στην εν λόγω θέση η κάμερα έχει την καλύτερη δυνατή οπτική, ακόμα καλύτερη και από αυτή του οδηγού, διότι βρίσκεται στο κέντρο του οχήματος και αρκετά μπροστά ώστε οι, εκατέρωθεν του παρμπρίζ, κολώνες του οχήματος να μην εμποδίζουν την οπτική της. Επιπρόσθετα δεν επηρεάζει την ορατότητα του οδηγού αφού «κρύβεται» πίσω από τον καθρέφτη. Η σύνδεση με το Raspberry Pi γίνεται μέσω μεγάλου μήκους USB καλωδίου το οποίο διατρέχει περιμετρικά το παρμπρίζ και στη συνέχεια καταλήγει στην πλευρά του οδηγού ενδιάμεσα από την πόρτα και το τιμόνι. Η κλίση της κάμερας θα πρέπει να ρυθμιστεί ώστε να έχει τον ορίζοντα και τον άξονα κίνησης του οχήματος στο κέντρο της προοπτικής της.

Εικόνα 4.16 : Παράδειγμα τοποθέτησης της κάμερας πίσω από τον μεσαίο καθρέφτη

4.3.2 Τοποθέτηση περιφερειακών εξαρτημάτων

Τα περιφερειακά εξαρτήματα, όπως το ηχείο και οι φωτεινές ενδείξεις συνδέονται στις GPIO θύρες του Raspberry Pi απευθείας ή μέσω καλωδίου επέκτασης και μπορούν να τοποθετηθούν ως εξής:

Τα φώτα κατάστασης του Συστήματος θα τοποθετηθούν σε μια τέτοια θέση, ώστε ο οδηγός να μπορεί να τα βλέπει με την περιφερειακή του όραση. Στην περίπτωση μας τοποθετήθηκαν μπροστά στο κεντρικό ταμπλό ενδείξεων του αυτοκίνητου.



Εικόνα 4.17 : Τοποθέτηση φώτων κατάστασης (λωρίδα LED)

Ο Hall Effect αισθητήρας θα πρέπει να τοποθετηθεί κάθετα στον άξονα κίνησης του μοχλού ενεργοποίησης των φωτών αλλαγής πορείας του οχήματος.

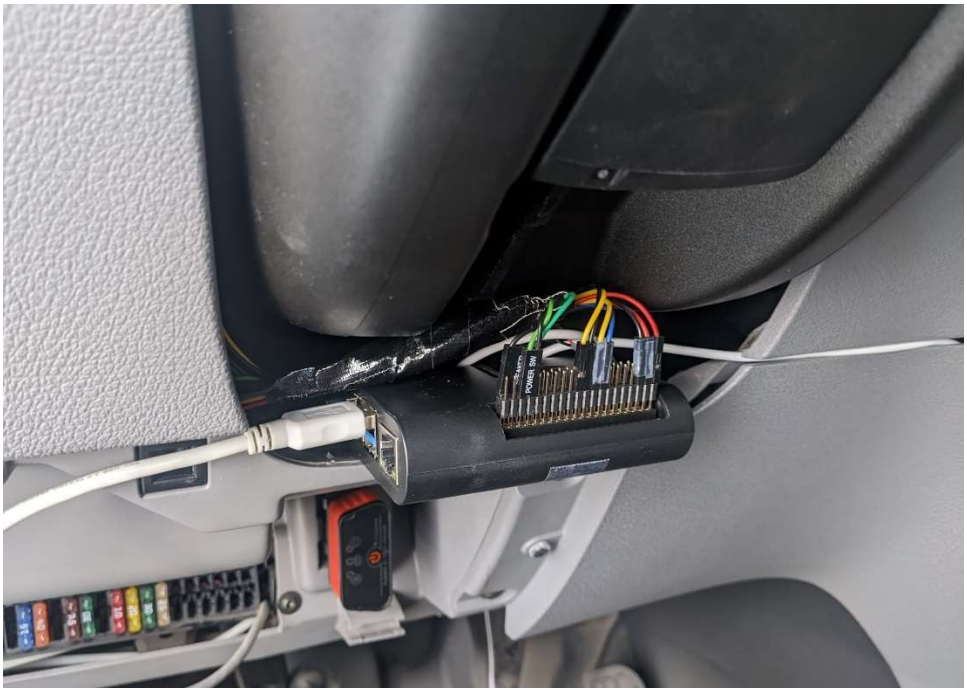
Εικόνα 4.18 : Τοποθέτηση Hall Effect αισθητήρα

Το ηχείο θα πρέπει να τοποθετηθεί κοντά στον οδηγό και να είναι στραμμένο προς αυτόν. Στην περίπτωση μας το ηχείο τοποθετήθηκε στα πλαγιά του τιμονιού του οχήματος.

Εικόνα 4.19 : Τοποθέτηση ηχείου

4.3.3 Τοποθέτηση Raspberry Pi

Το Raspberry Pi μπορεί να τοποθετηθεί οπουδήποτε αρκεί να είναι αρκετά κοντά στον οδηγό ώστε να μπορούν να συνδεθούν πάνω του τα περιφερειακά εξαρτήματα, η κάμερα και η τροφοδοσία. Ένα καλό τέτοιο σημείο είναι κάτω από το τιμόνι και πάνω από τα πόδια του οδηγού, όπου συνήθως υπάρχει κενός χώρος για την καλωδίωση του οχήματος.



Εικόνα 4.20: Τοποθέτηση Raspberry Pi

Στην συνέχεια μπορεί να γίνει ρύθμιση για αυτόματη εκκίνηση του Συστήματος και Ρύθμιση μεταβλητών του Συστήματος. (Αναλυτικότερη αναφορά στην §1 του Παραρτήματος).

Κεφάλαιο 5

Αποτίμηση της ακρίβειας του Συστήματος

Η ακρίβεια του Συστήματος αποτιμήθηκε ως ακολούθως. Αφού έγινε ενεργοποίηση του Συστήματος το όχημα ακολουθήσε πολλές δοκιμαστικές διαδρομές, συνολικής χρονικής διάρκειας άνω των 6 ωρών, οι οποίες περιλάμβαναν διαφορά είδη δρόμου (αυτοκινητόδρομοι, λεωφόροι, τοπικές οδοί κλπ.), επίπεδα κυκλοφοριακής συμφόρησης και συνθήκες φωτισμού (φυσικού ή τεχνητού) κατά τη διάρκεια της ημέρας όσο και κατά τη διάρκεια της νύχτας. Καθ΄ όλη την διάρκεια των διαδρομών έγινε καταγραφή της οπτικής του Συστήματος, των υπολογισμών του και των προειδοποιήσεών του σε μορφή βίντεο. Στη συνέχεια αποτιμήθηκε η ακρίβεια του Συστήματος, εξετάζοντας αν ήταν ορθές τόσο οι προειδοποιήσεις που έγιναν όσο και οι προειδοποιήσεις που δεν έγιναν. Εξετάστηκε δηλαδή η ακρίβεια του Συστήματος βάσει του παρακάτω τύπου:

$$\text{Ακρίβεια} = \frac{TP + TN}{TP+FP+TN+FN}$$

Οπού TP (True Positive) είναι οι ορθές προειδοποιήσεις, δηλαδή αυτές που ορθώς έγιναν, FP (False Positive) είναι οι λανθασμένες προειδοποιήσεις, δηλαδή αυτές που δεν θα έπρεπε να έχουν γίνει, TN (True Negative) είναι οι ορθές μη-προειδοποιήσεις, δηλαδή αυτές που ορθά δεν έγιναν και FN (False Negative) είναι οι λανθασμένες μη-προειδοποιήσεις, δηλαδή αυτές που λανθασμένα δεν έγιναν (ενώ θα έπρεπε να έχουν γίνει).

Το αποτέλεσμα είναι ένα ποσοστό που δηλώνει την ακρίβεια του Συστήματος.

5.1 Εξέταση υποσυστήματος προειδοποίησης αλλαγής λωρίδας κυκλοφορίας

Για την αποτίμηση της ακρίβειας του υποσυστήματος προειδοποίησης αλλαγής λωρίδας κυκλοφορίας καταμετρήθηκαν περιπτώσεις που το υποσύστημα προειδοποίησε σωστά για επικείμενη αλλαγή λωρίδας (ορθά θετικό), προειδοποίησε για επικείμενη αλλαγή λωρίδας που δεν έγινε (λανθασμένα θετικό) και δεν προειδοποίησε για επικείμενη αλλαγή λωρίδας ενώ αυτή έγινε (λανθασμένα αρνητικό).

Επισημαίνεται ότι δεν συμπεριλήφθηκαν στον τύπο της ακρίβειας, που αναφέρεται στην αρχή του παρόντος Κεφαλαίου, οι ορθές μη-προειδοποιήσεις, (TN), διότι η αναζήτηση λωρίδων κυκλοφορίας είναι μια συνεχής κατάσταση και όχι μεμονωμένα συμβάντα.

Η ακρίβεια του υποσυστήματος εξετάστηκε στα εξής είδη δρόμων:

Αυτοκινητόδρομος: Δρόμος υψηλής ταχύτητας, καλά διατηρημένος, με καθαρές

διαχωριστικές λωρίδες κυκλοφορίας

Αρτηριακός Δρόμος: Δρόμος μεσαίας ταχύτητας, με διαχωριστικές λωρίδες κυκλοφορίας μεταβλητής καθαρότητας

Συλλεκτήρια Οδός: Δρόμος μεσαίας ταχύτητας με περιστασιακή ύπαρξη διαχωριστικών λωρίδων κυκλοφορίας ελάχιστης καθαρότητας

Σημειώνεται ότι οι **Τοπικές Οδοί** δεν εξετάστηκαν στο παρόν υποσύστημα αφενός λόγω μη ύπαρξης διαχωριστικών λωρίδων κυκλοφορίας στο εν λόγω είδος δρόμων και αφετέρου λόγω χαμηλής ταχύτητας πορείας.

Τα αποτελέσματα που προέκυψαν από την εξέταση στο σύνολο των δοκιμαστικών διαδρομών φαίνονται στον επόμενο πίνακα, όπου το ποσοστό δηλώνει την ακρίβεια του υποσυστήματος και ο αριθμός εντός παρενθέσεως αποτελεί το σύνολο των δειγμάτων αλλαγής λωρίδας κυκλοφορίας.

Υποσύστημα Προειδοποίησης Αλλαγής Λωρίδας	**Μέρα**	**Νύχτα**
Ακρίβεια σε αυτοκινητόδρομο	95% (18)	73% (15)
Ακρίβεια σε αρτηριακό δρόμο	68.2% (22)	66% (15)
Ακρίβεια σε συλλεκτήρια οδό	66% (12)	60% (4)

5.2 Εξέταση υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης

Για την αποτίμηση της ακρίβειας του υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης χρησιμοποιήθηκε χαμηλό κατώφλι προειδοποίησης (μικρή αύξηση μεγέθους προπορευόμενου οχήματος μεταξύ ληφθέντων στιγμιότυπων), ώστε να είναι αυξημένη η ευαισθησία του υποσυστήματος για λογούς ασφάλειας. Καταμετρήθηκαν περιπτώσεις που το υποσύστημα προειδοποίησε σωστά για επικείμενη σύγκρουση με το προπορευόμενο όχημα (ορθά θετικό), προειδοποίησε λανθασμένα για μη επικείμενη σύγκρουση (λανθασμένα θετικό), δεν προειδοποίησε για επικείμενη σύγκρουση (λανθασμένα αρνητικό) και δεν προειδοποίησε για μη επικείμενη σύγκρουση (ορθά αρνητικό).

Τα αποτελέσματα που προέκυψαν από την εξέταση στο σύνολο των δοκιμαστικών διαδρομών φαίνονται στον επόμενο πίνακα όπου το ποσοστό δηλώνει την ακρίβεια και ο αριθμός εντός παρενθέσεως αποτελεί το σύνολο των δειγμάτων επικείμενης μπροστινής σύγκρουσης με προπορευόμενο όχημα.

Σημειώνεται ότι στο εν λόγω υποσύστημα αντιμετωπίστηκαν ενιαία οι επικείμενες συγκρούσεις για όλα τα είδη δρόμων, διότι η αύξηση μεγέθους προπορευόμενου οχήματος, που αποτελεί το κριτήριο προειδοποίησης, είναι κοινή σε όλα τα είδη δρόμων.

Υποσύστημα προειδοποίησης Μπροστινής Σύγκρουσης	Μέρα	Νύχτα
Ακρίβεια	82.8% (35)	72% (11)

5.3 Εξέταση υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος

Για την αποτίμηση της ακρίβειας του υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος καταμετρήθηκαν περιπτώσεις που το υποσύστημα προειδοποίησε σωστά για εκκίνηση προπορευόμενου οχήματος (ορθά θετικό), προειδοποίησε λανθασμένα για ακίνητο προπορευόμενο όχημα (λανθασμένα θετικό), δεν προειδοποίησε για εκκίνηση προπορευόμενου οχήματος (λανθασμένα αρνητικό) και δεν προειδοποίησε για ακίνητο προπορευόμενο όχημα (ορθά αρνητικό).

Τα αποτελέσματα που προέκυψαν από την εξέταση στο σύνολο των δοκιμαστικών διαδρομών φαίνονται στον επόμενο πίνακα όπου το ποσοστό δηλώνει την ακρίβεια και ο αριθμός εντός παρενθέσεως αποτελεί το σύνολο των δειγμάτων εκκίνησης προπορευόμενου οχήματος. Και στο εν λόγω υποσύστημα αντιμετωπίστηκαν ενιαία οι επικείμενες συγκρούσεις για όλα τα είδη δρόμων, για τους λόγους που αναφέρονται στην παράγραφο 5.2.

Υποσύστημα προειδοποίησης Εκκίνησης Προπορευόμενου Οχήματος	Μέρα	Νύχτα
Ακρίβεια	93% (14)	90.5% (21)

5.4 Εξέταση υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη

Για την αποτίμηση της ακρίβειας του υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη καταμετρήθηκαν για λογούς ασφάλειας όλες οι περιπτώσεις όπου το όχημα παραλίγο να παραβιάσει τον ερυθρό σηματοδότη, χωρίς να ληφθεί υπόψη ότι έγινε ασφαλής πέδηση του οχήματος (δηλαδή, πάρα το ότι στην πραγματικότητα έγινε ασφαλής πέδηση του οχήματος). Με βάση την ανωτέρω λογική δεν καταμετρήθηκαν περιπτώσεις που το υποσύστημα δεν προειδοποίησε για μη επικείμενη παραβίαση ερυθρού σηματοδότη (ορθά αρνητικό) αλλά καταμετρήθηκαν περιπτώσεις που το υποσύστημα προειδοποίησε σωστά για επικείμενη παραβίαση ερυθρού σηματοδότη (ορθά θετικό), προειδοποίησε λανθασμένα για μη επικείμενη παραβίαση ερυθρού σηματοδότη (λανθασμένα θετικό) και δεν προειδοποίησε για επικείμενη παραβίαση ερυθρού σηματοδότη (λανθασμένα αρνητικό).

Τα αποτελέσματα που προέκυψαν από την εξέταση στο σύνολο των δοκιμαστικών διαδρομών φαίνονται στον επόμενο πίνακα όπου το ποσοστό δηλώνει την ακρίβεια και ο αριθμός εντός παρενθέσεως αποτελεί το σύνολο των δειγμάτων προειδοποίησης παραβίασης ερυθρού σηματοδότη. Και στο εν λόγω υποσύστημα αντιμετωπίστηκαν ενιαία οι επικείμενες συγκρούσεις για όλα τα είδη δρόμων, για τους λόγους που αναφέρονται στην παράγραφο 5.2.

Υποσύστημα προειδοποίησης Παραβίασης Ερυθρού Σηματοδότη	**Μέρα**	**Νύχτα**
Ακρίβεια	64% (14)	44.4% (9)

Κεφάλαιο 6

Συμπεράσματα από την αξιολόγηση του Συστήματος, Πιθανές μελλοντικές βελτιώσεις

6.1 Συμπεράσματα από την αξιολόγηση του Συστήματος

Το Σύστημα σε γενικές γραμμές αποδείχτηκε αρκετά ικανό στο να προειδοποιεί τον οδηγό στις επικίνδυνες περιπτώσεις των πολλών δοκιμαστικών διαδρομών που έγιναν. Παρά το ότι δεν είναι αντίστοιχης ικανότητας με τα υπάρχοντα συστήματα της αυτοκινητοβιομηχανίας λόγω του σχετικά χαμηλού κόστους, της αυτόνομης φύσης του και της ευέλικτης τοποθέτησής του μπορεί να εγκατασταθεί σε οποιοδήποτε όχημα, προσφέροντας έτσι και σε αυτούς που δεν διαθέτουν όχημα της τελευταίας πενταετίας μια βασική επιπρόσθετη ασφάλεια σε περίπτωση απόσπασης της προσοχής τους κατά τη διάρκεια της οδήγησης.

6.1.1 Συμπεράσματα για το Λογισμικό του Συστήματος

Η Python αποδείχτηκε μια πολύ ευέλικτη γλώσσα που έδωσε στο Σύστημα τη δυνατότητα να μπορεί να εκτελεστεί σε οποιουδήποτε είδους υπολογιστή χωρίς αλλαγές στον κώδικα. Η απλή σύνταξη και δομή της διατήρησαν τον κώδικα αναγνώσιμο και εύκολα επεξεργάσιμο πάρα την πολυπλοκότητά του.

Ωστόσο το κυρίαρχο πλεονέκτημά της οφείλεται στις έτοιμες βιβλιοθήκες της οι οποίες δίνουν τη δυνατότητα το Σύστημα να επικοινωνεί με όλες τις συσκευές και αισθητήρες που απαιτήθηκαν. Η βασικότερη βιβλιοθήκη που χρησιμοποιήθηκε είναι το OpenCV, το οποίο μέσω των εξόδων εικόνας και σχεδιασμού σχημάτων που προσφέρει, παρείχε άμεση ανταπόκριση κατά την ανάπτυξη του Συστήματος βοηθώντας πολύ στην αποσφαλμάτωση.

Η εκτέλεση του μοντέλου νευρωνικού δικτύου ssd_mobilenet_v3_small, για την αναγνώριση αντικειμένων με τη χρήση της Tensorflow Lite βιβλιοθήκης, αποδείχτηκε ικανοποιητική κατά τη διάρκεια της ημέρας αναγνωρίζοντας τα περισσότερα αντικείμενα σε κάθε στιγμιότυπο με καλή σταθερότητα. Ωστόσο κατά τη διάρκεια της νύχτας διαπιστώθηκε μεγάλη πτώση της ακρίβειας, με τα περισσότερα αντικείμενα να αναγνωρίζονται περιστασιακά, μερικώς ή καθόλου. Αυτό είχε ως αποτέλεσμα την αντίστοιχα μεγάλη πτώση της ακρίβειας των υποσυστημάτων που χρησιμοποιούν τα αναγνωρισμένα αυτά αντικείμενα. Η εν λόγω μειωμένη ακρίβεια οφείλεται κυρίως στην εκπαίδευση του μοντέλου, στην οποία έγινε χρήση του συνόλου δεδομένων COCO Dataset. Το υπόψη μοντέλο είναι μεν μεγάλο και ποικίλο αλλά, εκ του αποτελέσματος, φαίνεται ότι δεν περιέχει πολλά δείγματα των επιλεγμένων αντικειμένων σε νυχτερινές καταστάσεις. Το πρόβλημα αυτό αντισταθμίστηκε περιστασιακά, μέσω της χρήσης

αλγορίθμων εξαγωγής και παρακολούθησης σημαντικών σημείων, οι οποίοι βελτιώνουν την ακρίβεια των υποσυστημάτων κατά τη διάρκεια της νύχτας αλλά και περαιτέρω της ημέρας.

6.1.1.1 Συμπεράσματα υποσυστήματος προειδοποίησης αλλαγής λωρίδας κυκλοφορίας

Το υποσύστημα αποχώρησης από την λωρίδα κυκλοφορίας (Lane Departure warning system) που υλοποιήθηκε λειτουργεί αρκετά ικανοποιητικά σε όλα τα είδη δρόμων συμπεριλαμβανομένων ακόμα και ορεινών δρόμων με στροφές (μέσω της δυναμικής περιοχής ενδιαφέροντος) και δρόμων με μεταβολές φωτεινότητας του οδοστρώματος, λόγω αλλαγών του φωτισμού του περιβάλλοντος εξαιτίας σκιών, φωτών και θέσης του ηλίου (μέσω της δειγματοληψίας της φωτεινότητας του δρόμου).

Ωστόσο, λόγω του καθορισμού ενός μοναδικού κάτω ορίου φωτεινότητας για το σύνολο της εικόνας (βήμα 1 της παραγράφου 3.3.1) δεν επιτυγχάνεται πάντα ορθή απομόνωση των λωρίδων κυκλοφορίας σε όλο το μήκος τους σε περιπτώσεις απότομης αλλαγής φωτεινότητας του δρόμου λόγω σκιών ή φώτων.

Κατά τη διάρκεια της εξέτασης του Συστήματος παρατηρήθηκε ότι το υποσύστημα ενώ αναγνωρίζει ορθά λωρίδες κυκλοφορίας μεγάλου φάσματος καθαρότητας παρουσιάζει μια αδυναμία να προειδοποιεί σε γρήγορες αλλαγές λωρίδας. Αυτό κυρίως οφείλεται στον χρόνο εκτέλεσης του Συστήματος από το Raspberry Pi, ο οποίος είναι της τάξης των 100 ms.

6.1.1.2 Συμπεράσματα υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης

Το υποσύστημα υποβοήθησης προειδοποίησης μπροστινής σύγκρουσης (Forward Collision Warning), που υλοποιήθηκε μπορεί να προειδοποιήσει αρκετά ικανοποιητικά σε επικείμενες συγκρούσεις στο οπίσθιο μέρος προπορευόμενου οχήματος και περιστασιακά στις περιπτώσεις σύγκρουσης με άνθρωπο ή με αντικείμενα στο δρόμο.

Κατά τη διάρκεια εξέτασης του Συστήματος παρατηρήθηκε ότι το υποσύστημα ενώ, τις περισσότερες φορές, αναγνωρίζει ορθά τη μεταβολή μεγέθους ενός σχετικά κοντά ευρισκόμενου προπορευόμενου οχήματος (ως τέτοιο νοείται όχημα που βρίσκεται σε απόσταση μικρότερη από περίπου 20 μέτρα) παρουσιάζει μια αδυναμία να αντιλαμβάνεται το μέγεθος σε μεγαλύτερες αποστάσεις. Αυτό σημαίνει ότι σε μεγάλες ταχύτητες κίνησης των οχημάτων, όπως στις εθνικές οδούς, που οι αποστάσεις πορείας μεταξύ των οχημάτων απαιτείται να είναι μεγαλύτερες, λόγω και του χρόνου εκτέλεσης του Συστήματος μια προειδοποίηση για επικείμενη σύγκρουση μπορεί να καθυστερήσει αρκετά και να μην είναι επαρκής, ιδιαίτερα σε περιπτώσεις που το προπορευόμενο όχημα επιβραδύνει σημαντικά, έως ακινησίας, για κάποιο λόγο.

6.1.1.3 Συμπεράσματα υποσυστήματος προειδοποίησης εκκίνησης προπορευόμενου οχήματος

Το υποσύστημα προειδοποίησης εκκίνησης προπορευόμενου οχήματο**ς** (Forward Vehicle Start Alarm) που υλοποιήθηκε μπορεί να προειδοποιήσει πολύ ικανοποιητικά σε εκκίνηση και σταθερή απομάκρυνση προπορευόμενου οχήματος, στην οποία δεν αντιδρά ο οδηγός για χρονικό διάστημα μεγαλύτερο των 2 δευτερολέπτων.

6.1.1.4 Συμπεράσματα υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη

Το υποσύστημα προειδοποίησης παραβίασης ερυθρού σηματοδότη (Traffic Light Warning) που υλοποιήθηκε μπορεί περιστασιακά να προειδοποιήσει σε επικείμενη παραβίαση ερυθρού σηματοδότη.

Κατά τη διάρκεια εξέτασης του Συστήματος παρατηρήθηκε ότι το υποσύστημα επειδή αναγνωρίζει περιστασιακά τους ερυθρούς σηματοδότες παρουσιάζει μια δυσκολία στη συνεχή τοποθέτησή τους στην οπτική του, με αποτέλεσμα αντίστοιχη δυσκολία στην ορθή προειδοποίηση του οδηγού. Αυτό οφείλεται στην ακρίβεια του μοντέλου και στο χρόνο εκτέλεσης του Συστήματος.

6.1.2 Συμπεράσματα για το Υλικό του Συστήματος

Το Raspberry Pi 4 αποδείχθηκε αρκετά ικανό στην εκτέλεση του Συστήματος, με μια ικανοποιητική ταχύτητα 100 ms (10 φορές το δευτερόλεπτο), πάρα το χαμηλό κόστος του και τον μη εξειδικευμένο, στην εκτέλεση νευρωνικών δικτυών, επεξεργαστή του. Ευνόητο είναι ότι μια υψηλότερη ταχύτητα θα ήταν επιθυμητή, ωστόσο σημειώνεται ότι ο χρόνος απόκρισης των 100 ms είναι ήδη αντίστοιχος ή καλύτερος από το χρόνο αντίδρασης του μέσου ανθρώπου.

Η ευελιξία και η ευκολία στη σύνδεσή του με πολλά είδη αισθητήρων και η εύκολη τροφοδότηση του συνέβαλε καθοριστικά στη γρήγορη δημιουργία ενός πρακτικού μικρού αυτονόμου συστήματος, για την εγκατάσταση και τη λειτουργία του οποίου απαιτούνται ελάχιστες ηλεκτρολογικές γνώσεις.

6.2 Πιθανές μελλοντικές βελτιώσεις του Συστήματος

6.2.1 Βελτιώσεις του Λογισμικού του Συστήματος

- Όπως αναφέρθηκε στα συμπεράσματα (παράγραφος 6.1.1) το κύριο μειονέκτημα του Συστήματος είναι η ικανότητα αναγνώρισης των επιλεγμένων αντικειμένων από το μοντέλο νευρωνικού δικτύου ssd_mobilenet_v3_small, το οποίο εκπαιδεύτηκε στο σύνολο δεδομένων Coco Dataset. Μια λύση για αυτό το μειονέκτημα είναι η εκπαίδευση του μοντέλου με τη χρήση ενός πιο εξειδικευμένου Dataset, που αποτελείται μόνο από τα αντικείμενα που μας ενδιαφέρουν σε μεγάλη ποικιλία. Ένα τέτοιο σύνολο δεδομένων είναι το Berkeley DeepDrive (BDD100K Dataset) που διαθέτει 100.000 εικόνες από την οπτική του οδηγού σε διάφορες συνθήκες, καιρικές και φωτισμού. Έτσι η ακρίβεια του Συστήματος θα μπορεί θα αυξηθεί χωρίς να γίνει μείωση της ταχύτητας.

- Η γλώσσα Python συνέβαλε πολύ στην ανάπτυξη του Συστήματος αλλά είναι κοινά αποδεκτό ότι συγκριτικά είναι πιο αργή από μια χαμηλότερου επιπέδου γλώσσα όπως η C++. Γι’ αυτό, αφού πλέον ολοκληρώθηκε η ανάπτυξη του Συστήματος, θα μπορούσε να γίνει η μετατροπή του κώδικα σε γλώσσα C++ αυξάνοντας την ταχύτητά του.

- Περαιτέρω βελτιστοποίηση του κώδικα με μεγαλύτερη χρήση παράλληλης εκτέλεσης ώστε να γίνει καλύτερη εκμετάλλευση των πολυπύρηνων επεξεργαστών, όπως του Raspberry Pi 4.

6.2.1.1 Βελτιώσεις του υποσυστήματος προειδοποίησης αλλαγής λωρίδων κυκλοφορίας

- Αξιολόγηση των αναγνωρισμένων ευθειών του βήματος 4 της παραγράφου 3.3.1 με βάση την απόστασή τους από την πορεία του οχήματος, κατά τρόπο ώστε η σημαντικότητά τους να είναι υψηλότερη όσο πιο κοντά βρίσκονται στην πορεία αυτή.

- Μείωση της μη ορθής απομόνωσης των λωρίδων κυκλοφορίας σε περιπτώσεις αλλαγής φωτεινότητας θα μπορούσε να επιτευχθεί μέσω κατάτμησης της εικόνας σε πολλές οριζόντιες ζώνες, δειγματοληψίας της φωτεινότητας της κάθε ζώνης και καθορισμού κάτω ορίου φωτεινότητας σε κάθε ζώνη. Με τον τρόπο αυτό θα γίνεται εξειδικευμένη απομόνωση των λωρίδων κυκλοφορίας σε κάθε ζώνη, διότι θα γίνεται σύγκριση της φωτεινότητας των λωρίδων με τη φωτεινότητα του τμήματος του δρόμου που υπάρχει στην ίδια ζώνη με αυτές.

- Αναγνώριση διαβάσεων πεζών, μέσω της αναγνώρισης πολλών παράλληλων ευθειών στο βήμα 4 της παραγράφου 3.3.1, και απενεργοποίηση του υποσυστήματος καθώς γίνεται διάσχισή τους.

- Αναγνώριση λωρίδων κυκλοφορίας βάσει και των δυο ακμών που διαθέτουν (μια σε κάθε πλευρά τους). Έτσι οι λωρίδες θα διακρίνονται από τις σκιές που συνήθως διαθέτουν μόνο μια ακμή (εναλλαγή φωτός με σκιά).

6.2.1.2 Βελτιώσεις του υποσυστήματος προειδοποίησης μπροστινής σύγκρουσης

- Αξιολόγηση της πιθανότητας επικείμενης σύγκρουσης με βάση την απόσταση από το προπορευόμενο αντικείμενο κρίνοντας το μέγεθος του από το πραγματικό μέγεθος του, τη γωνιά θέασης της κάμερας και το μέγεθος που καταλαμβάνει αυτό στην οπτική της. Στη συνέχεια με βάση την ταχύτητα του οχήματός μας, τη μεταβολή απόστασης από το προπορευόμενο αντικείμενο και το χρόνο πέδησης θα αποφασίζεται αν θα γίνει προειδοποίηση του οδηγού.

- Χρήση δεύτερης κάμερας με μικρότερη γωνιά θέασης που θα προσφέρει υψηλότερη λεπτομέρεια σε μεγαλύτερες αποστάσεις, αυξάνοντας την απόσταση που το υποσύστημα θα μπορεί να κρίνει ορθά για επικείμενη σύγκρουση.

6.2.1.3 Βελτιώσεις του υποσυστήματος προειδοποίησης παραβίασης ερυθρού σηματοδότη

- Ταυτόχρονη χρήση και ενός μικρότερου νευρωνικού δικτύου και εκπαίδευσή του με εξειδικευμένο Dataset συμβόλων (όπως π.χ. το MNIST) για τη διαφοροποίηση των ειδών σηματοδότη (βέλους, κανονικό) και την εξάλειψη της επιλογής, από το συνδυασμό των μοντέλων (του ssd_mobilenet_v3_small και του προαναφερόμενου μικρότερου νευρωνικού δικτύου), φώτων, των οποίων η δομή δεν ανήκει στα καθορισμένα είδη σηματοδότη.

- Με την παραπάνω γνώση συμβόλων θα μπορούμε να διακρίνουμε τα βέλη και την κατεύθυνση που αφορούν. Έτσι σε περίπτωση διπλού σηματοδότη που είναι κόκκινος για ευθεία πορεία και πράσινος για άλλη κατεύθυνση θα μπορεί να γίνει σωστή προειδοποίηση ή μη, με βάση την κατάσταση του μοχλού φώτων δείκτη κατεύθυνσης.

- Καθορισμός ιδιαίτερου, για κάθε κατεύθυνση πορείας, ορίου καθυστέρησης ενεργοποίησης προειδοποίησης (παράγραφος 3.6.2). Με τον τρόπο αυτό θα αποφεύγεται η καθυστέρηση της προειδοποίησης του οδηγού σε περίπτωση αλλαγής της κατάστασης του μοχλού φώτων δείκτη κατεύθυνσης σε περιπτώσεις αντίθετου χρώματος σε διπλούς σηματοδότες και σε περιπτώσεις μεμονωμένων σηματοδοτών, τύπου βέλους.

- Αναγνώριση σώματος σηματοδότη και αφαίρεση του φόντου. Έτσι θα καταστεί δυνατό να ληφθεί υπόψη και η θέση της φωτεινής ένδειξης πάνω στο σώμα (πάνω: κόκκινο, κάτω: πράσινο) για τον προσδιορισμό του χρώματος του σηματοδότη.

- Διαχωρισμός του κουτιού του διπλού σηματοδότη με βάση τα σώματα των επιμέρους σηματοδοτών αντί του διαχωρισμού του κουτιού στη μέση του (παράγραφος 3.6.1.2).

6.2.2 Βελτιώσεις Υλικού

- Με τη χρήση επιταχυντή εκτέλεσης νευρωνικών δικτυών, τύπου Coral USB Accelerator, θα ήταν δυνατός ο υποδιπλασιασμός του χρόνου του Συστήματος, από 100ms (10fps) σε 50ms (20fps), αυξάνοντας δραστικά την απόδοση του Συστήματος, ειδικά σε υψηλές ταχύτητες κίνησης του οχήματος, που κάθε κλάσμα δευτερολέπτου έχει μεγάλη σημασία.

- Χρήση μελλοντικής ταχύτερης έκδοσης του Raspberry Pi (5+) ή αλλού αυτονόμου υπολογιστή ιδίου τύπου και κόστους (όπως π.χ. RockPi 5A, Orange Pi 5), που διαθέτουν όμως επεξεργαστές νεότερης τεχνολογίας με ενσωματωμένο επιταχυντή νευρωνικών δικτυών για ακόμα μεγαλύτερη αύξηση της ταχύτητας του Συστήματος.

- Περαιτέρω τροποποίηση του λειτουργικού συστήματος του Raspberry Pi, ώστε να εκκινεί σε «headless mode», στο οποίο δεν θα φορτώνει τη διεπαφή με το χρήστη (UI) αλλά θα εκκινεί μόνο το Σύστημα, μειώνοντας έτσι το χρόνο εκκίνησης.

Παράρτημα Α’

Στο παράρτημα αυτό παρατίθεται ο οδηγός εγκατάστασης του Συστήματος στο Raspberry Pi.

1.1 Εγκατάσταση του Συστήματος στο Raspberry Pi

Αρχικά θα πρέπει να γίνει η εγκατάσταση της Python. Προτείνεται η έκδοση 3.10 ή παλαιότερη έκδοση. Η εγκατάσταση γίνεται μέσω της εντολής:

“`sudo apt install python3`”

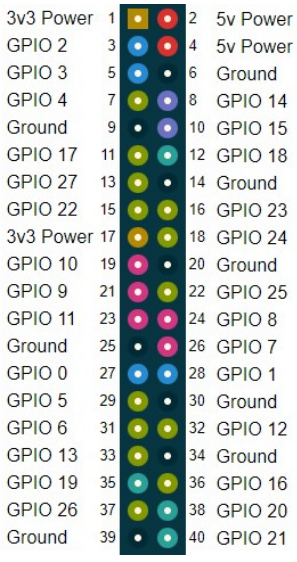
Στην συνέχεια θα πρέπει να γίνει εγκατάσταση των βιβλιοθηκών που χρησιμοποιεί το Σύστημα, οι οποίες είναι:
opencv_python v4.6, **numpy** v1.21, **tensorflow** v2.11, **pygame** v2.1.2, **obd** v0.7.1, **pyserial** v3.5.

Η εγκατάσταση γίνεται με την εκτέλεση της εντολής :

“`python -m pip install -r requirements.txt`”,

μέσα από τον φάκελο που έχει τοποθετηθεί το αρχείο “`requirements.txt`”.

1.2 Τοποθέτηση περιφεριακών στα GPIO pins



Αισθητηρας Hall effect:
Power 5V: Pin 2, **Ground**: Pin 20, **Data**: Pin 22 (GPIO 25)

Φωτεινές Ενδείξεις:
Power 5V: Pin 4, **Ground**: Pin 6, **Data**: Pins 16,18 (GPIO 23,24)

Ηχειο τυπου Buzzer:
Ground: Pin 39, **Data**: Pin 33 (GPIO 13)

Τα **Data** pins μπορούν να ρυθμιστούν και από το .py αρχείο αν οι παραπάνω θέσεις δεν είναι εφικτές

1.3 Σύνδεση συσκευής OBD2

Η σύνδεση της συσκευής OBD2 Bluetooth μπορεί να γίνει (καθώς το Raspberry Pi έχει τοποθετηθεί στο όχημα) μέσω SSH Shell από άλλο υπολογιστή ή κινητό τηλέφωνο. Ποιο συγκεκριμένα πρέπει να εκτελεστούν οι εξής εντολές:

```
Bluetoothctl
scan on
pair XX:XX:XX:XX:XX:XX (mac address του OBD2)
trust XX:XX:XX:XX:XX:XX
exit
```

1.4 Αυτόματη εκκίνηση του Συστήματος

Το Σύστημα ενεργοποιείται με την εκκίνηση του οχήματος, δηλαδή με την παροχή ρεύματος μέσω του τροφοδοτικού. Συνεπώς θα πρέπει το Raspberry Pi με το που ολοκληρώσει την διαδικασία εκκίνησης του (boot) να ξεκινάει αυτόματα το πρόγραμμα του Συστήματος. Αυτό γίνεται μέσω της εντολής :

```
"xterm -hold -e 'sudo rfcomm bind rfcomm9 XX:XX:XX:XX:XX:XX &&
                cd /home/pi/Desktop/Py-ADAS/ &&
        /usr/bin/python /home/pi/Desktop/Py-ADAS/PI-ADAS.py'"
```

(όπου XX:XX:XX:XX:XX:XX το mac address του OBD2 που βρέθηκε παραπάνω)

η οποία μετατρέπεται σε εκτελέσιμο αρχείο .desktop και τοποθετείται στο φάκελο autostart του λειτουργικού συστήματος. Έτσι το Σύστημα ενεργοποιείται αυτόματα με την εκκίνηση του οχήματος.

1.5 Ρύθμιση μεταβλητών του Συστήματος

Το Σύστημα είναι αρχικά ρυθμισμένο με μεταβλητές του μέσου αυτοκίνητου, οι οποίες μπορούν να τροποποιηθούν ώστε το Σύστημα να είναι ποιο αποδοτικό στο κάθε όχημα. Οι ρυθμίσεις είναι:

 # Pi Variables

- **buzzerPin = 13 # Ο αριθμός του GPIO pin για το buzzer ή ηχείο**
- **turnPin = 25 # Ο αριθμός του GPIO pin για την αναγνώριση της κατάστασης του φλας (Hall Effect/level converter)**

 # Lane Variables

- **carsize = 500 # (εικονοστοιχεία) Πλάτος ζώνης πορείας οχήματος**
- **fov = -250 # (εικονοστοιχεία) Σημείο έναρξης της περιοχής ενδιαφέροντος, που αλλάζει ανάλογα τη γωνιά θέασης της κάμερας**
- **laneLockThresh = 20 # (στιγμιότυπα) Ποσό καθυστέρησης ενεργοποίησης προειδοποίησης**
- **newLineW = 0.6 # (%) Ποσοστό ευαισθησίας σε κάθε νέα αναγνωριζομένη λωρίδα κυκλοφορίας**
- **brightsens = 10 # (0-255) Κατώτατο όριο διαφοράς αντίθεσης δρόμου και λωρίδας κυκλοφορίας**
- **starting_horizon_Ratio = 1/2 # (%) Αρχική τοποθέτηση ορίζοντα σε αναλογία της εικόνας.**
- **glassGradStreght = 65 #(0-255) Ισχύς αντιστάθμισης πτώσης φωτεινότητας, λόγω διάθλασης από την κλίση του παρμπρίζ του οχήματος**

Object Collision Variables

- **inPathSens = 1 # (%) Γενικό κατώτατο όριο ευαισθησίας επιλογής αντικειμένου που είναι στην πορεία του οχήματός μας**
- **weight_size = 0.9 # (%) Ειδικό κατώτατο όριο ευαισθησίας επιλογής αντικειμένου που είναι στην πορεία του οχήματός μας βάσει μεγέθους**
- **weight_center = 1 # (%) Ειδικό κατώτατο όριο ευαισθησίας επιλογής αντικειμένου που είναι στην πορεία του οχήματός μας βάσει απόστασης από το κέντρο**
- **newObjThresh = 40 # (εικονοστοιχεία) Κατώτατο όριο αλλαγής σημαντικού αντικειμένου**
- **CollisionSens = 0.06 # (%) Κατώτατο όριο ποσοστού αλλαγής εμβαδού κουτιού για προειδοποίηση**
- **trackingSens = 0.04 # (%) Κατώτατο όριο ποσοστού αλλαγής απόστασης σημείων αντικειμένου για προειδοποίηση**
- **trackingoffThresh = 7 # (%) Ανώτατο όριο ποσοστού αλλαγής Θέσης κέντρου σημείων αντικειμένου για προειδοποίηση**

Departure Variables

- **departureSens = 18 # (στιγμιότυπα) Κατώτατο όριο στιγμιότυπων αναγνωρισμένης απομάκρυνσης προπορευόμενου οχήματος για προειδοποίηση**
- **departureThresh = 30000 # (εικονοστοιχεία) Κατώτατο όριο μεγέθους αντικειμένου για την ενεργοποίηση της προειδοποίησης**

Light Variables

- **color_MINr = np.array([117, 60, 214], np.uint8) # (0-179,0-255,0-255) Κατώτατο όριο κόκκινου χρώματος για την αναγνώριση χρώματος σηματοδότη**
- **color_MAXr = np.array([179, 255, 255], np.uint8) # (0-179,0-255,0-255) Ανώτατο όριο κόκκινου χρώματος για την αναγνώριση χρώματος σηματοδότη**
- **color_MINg = np.array([30, 60, 190], np.uint8) # (0-179,0-255,0-255) Κατώτατο όριο πράσινου χρώματος για την αναγνώριση χρώματος σηματοδότη**
- **color_MAXg = np.array([40, 255, 255], np.uint8) # (0-179,0-255,0-255) Ανώτατο όριο πράσινου χρώματος για την αναγνώριση χρώματος σηματοδότη**
- **lightthresh = 5 # (εικονοστοιχεία) Κατώτατο όριο ποσού εικονοστοιχείων πράσινου ή κόκκινου χρώματος για την αναγνώριση χρώματος σηματοδότη**
- **lightChangeThresh = 4 # (στιγμιότυπα) Ποσό συνεχομένων αναγνωρισμένων σηματοδοτών για ενεργοποίηση προειδοποίησης**
- **egdeThresh = 50 # (εικονοστοιχεία) Μέγεθος ζώνης στα όρια της εικόνας για έναρξη προειδοποίησης**
- **accelthresh = 0 # (m/s) Κατώτατο όριο επιβράδυνσης για την απενεργοποίηση της προειδοποίησης**
- **minDistLight = 5 # (m) Απόσταση σηματοδότη όταν αυτός βρίσκεται στα όρια της οπτικής της κάμερας**

Παράρτημα Β’

Στο παράρτημα αυτό παρατίθενται σημαντικά κομμάτια κώδικα ανάπτυξης της παρούσας εφαρμογής.

1. Υποσύστημα προειδοποίησης αλλαγής λωρίδας κυκλοφορίας

1.1 Απομόνωση διαχωριστικών γραμμών λωρίδων κυκλοφορίας (§3.3.1, βήμα 1ο):

Συνάρτηση που παίρνει ένα τμήμα από μια εικόνα και βρίσκει τη μέση φωτεινότητά του.

```
def get_road_brightness(hsv_img):
   masked_image = hsv_img[y1:y2, x1:x2]
   average_color_per_row = np.max(masked_image, axis=1)
   average_color = np.average(average_color_per_row, axis=0)
   average_color = np.uint8(average_color)
   road_bright=average_color[2]
   return road_bright
```

Συνάρτηση που κατασκευάζει ένα φίλτρο φωτεινότητας με βάση το αποτέλεσμα της παραπάνω συνάρτησης και περνάει μια εικόνα από αυτό αφαιρώντας έτσι όλα τα αλλά μη επαρκώς φωτεινά χρώματα

```
color_MAX = np.array([179, 255, 255], np.uint8)
color_MIN = np.array([0, 0, get_road_brightness(hsv_img)],np.uint8)

frame_threshedW = cv2.inRange(hsv_img, color_MIN, color_MAX)
resultWfr = cv2.bitwise_and(frame, frame, mask=frame_threshedW)
```

1.2 Εύρεση Ακμών (§3.3.1 βήμα 2ο):

Εύρεση ακμών με τη χρήση του αλγορίθμου Canny αφού γίνει πρώτα μετατροπή της εικόνας σε ασπρόμαυρη και περαστεί από Gaussian φίλτρο μεγέθους 5x5.

```
def canny(self,img):
      gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
      blur1 = cv2.GaussianBlur(gray, (5, 5),0,0)
      canny = cv2.Canny(blur1, 80, 140)
      return canny
```

1.3 Ορισμός δυναμικής περιοχής ενδιαφέροντος (§3.3.1 βήμα 3ο και §3.3.2 βήμα 3ο):

Ο ορισμός της περιοχής ενδιαφέροντος γίνεται μέσω της δημιουργίας μιας μάσκας το σχήμα της οποίας είναι ένα τρίγωνο. Το τρίγωνο αυτό έχει βάση που καθορίζεται από το πλάτος του οχήματος και τη γωνιά θέασης της κάμερας (ρυθμίσιμες μεταβλητές) και η κορυφή του (που βρίσκεται απέναντι από τη βάση) ταυτίζεται με το σημείο φυγής του οχήματος. Στο τρίγωνο αυτό, αφού αγνοηθεί ο κεντρικός άξονας του δρόμου, καθορίζεται ένα όριο ύψους ως προς τον ορίζοντα, με αποτέλεσμα να απομείνει ένα τραπέζιο όπως φαίνεται στην εικόνα 3.19.

```
def region_of_interest(self,canny):
  self.width = canny.shape[1]
  mask = np.zeros_like(canny)

  roi = np.array([[
     (fov+camOffset, self.height-hood),
     (self.lanecenter, self.horizon),
     (self.width/2-carsize/7+camOffset, self.height-hood),
     (self.width/2+carsize/7+camOffset, self.height-hood),
     (self.lanecenter, self.horizon),
     (self.width-fov+camOffset, self.height-hood)]], np.int32)

  mask_limit = np.array([[
     (0, 0),(0, self.horizon*lane_depth),
     (self.width, self.horizon*lane_depth),
     (self.width, 0)]], np.int32)

  cv2.fillPoly(mask, roi, 255)
  cv2.fillPoly(mask, mask_limit, 0)
  masked_image = cv2.bitwise_and(canny, mask)
  return masked_image, mask
```

1.4 Εύρεση και ομαδοποίηση ευθείων με χρήση Houghlines και ομαλοποίηση τους στο χρόνο με βάση τις προηγούμενες (§3.3.1 βήμα 4ο και §3.3.2 βήμα 1ο):

Στο παρόν βήμα γίνεται μετασχηματισμός των σημείων, δηλαδή του συνόλου των ψηφίων της εικόνας που προέκυψαν στο προηγούμενο βήμα, σε ευθείες (Houghlines). Το αποτέλεσμα είναι ευθείες διάφορου μήκους και κατεύθυνσης οι οποίες περνάνε από την συνάρτηση average_slope_intercept η οποία φιλτράρει τις σημαντικές και τις ομαδοποιεί με βάση την κλίση τους και τη θέση τους. Στη συνέχεια, αν υπάρχουν ευθείες σε αυτό το στιγμιότυπο, συνδυάζονται σε δυο μέσες ευθείες. Στη

συνέχεια οι νέες ευθείες συνδυάζονται με τις ευθείες της προηγούμενης αναζήτησης (στιγμιότυπου) βάσει ενός συντελεστή βαρύτητας.

```
lines = cv2.HoughLinesP(cropped, 2, np.pi/180, 100,np.array([]),
                        minLineLength=60, maxLineGap=5)

def average_slope_intercept(self,image, lines):
    left_fit = []
    right_fit = []
    Rlane = image.shape[1]/2+50
    Llane = image.shape[1]/2-50
    if lines is not None:
        for line in lines:
            for x1, y1, x2, y2 in line:
                slope = (y1-y2)/(x1-x2)
                if 7 > slope > 0.3:
                    if x1 > Rlane:
                        yintercept = y2 - (slope*x2)
                        right_fit.append((slope, yintercept))
                elif -7 < slope < -0.3:
                    if x1 < Llane:
                        yintercept = y2 - (slope*x2)
                        left_fit.append((slope, yintercept))
    if left_fit:
        left_fit_average = np.average(left_fit, axis=0)
        left_line = self.make_points(image, left_fit_average)
    else:
        left_line = [[0, 0, 0, 0]]

    if right_fit:
        right_fit_average = np.average(right_fit, axis=0)
        right_line = self.make_points(image, right_fit_average)
    else:
        right_line = [[0, 0, 0, 0]]

    oldLineW = 1-newLineW
    if any(left_line[0]):
        if any(self.averaged_lines[0][0]):
            for i in range(0,3):
                self.averaged_lines[0][0][i] =
```

```
int(self.averaged_lines[0][0][i]*oldLineW+left_line[0][i]*newLineW)
      else:
         self.averaged_lines[0] = left_line
   else:
      self.averaged_lines[0] = left_line
   if any(right_line[0]):
      if any(self.averaged_lines[1][0]):
         for i in range(0,3):
            self.averaged_lines[0][0][i] =
int(self.averaged_lines[0][0][i]*oldLineW+left_line[0][i]*newLineW)
      else:
         self.averaged_lines[1] = right_line
   else:
      self.averaged_lines[1] = right_line

   return self.averaged_lines
```

1.5 Καθορισμός ζώνης πορείας οχήματος (§3.3.1 βήμα 3ο):

Από μια ρυθμιζόμενη τιμή του πλάτους του οχήματος καθορίζεται η ζώνη πορείας του οχήματος. Η ζώνη πορείας είναι ένα τρίγωνο που έχει βάση που καθορίζεται από το πλάτος του οχήματος (ρυθμίσιμη μεταβλητή) και η κορυφή του (που βρίσκεται απέναντι από τη βάση) ταυτίζεται με το σημείο φυγής του οχήματος. Στο τρίγωνο αυτό καθορίζεται ένα όριο ύψους ως προς τον ορίζοντα, με αποτέλεσμα να απομείνει ένα τραπέζιο όπως φαίνεται στην εικόνα 3.24.

```
mask = np.zeros_like(img)
carPathSpace = np.array([[
   (self.width / 2-carsize/2+camOffset, self.height+wheelOff),
   (self.lanecenter, self.horizon),
   (self.lanecenter, self.horizon),
   (self.width / 2 + carsize/2+camOffset, self.height+wheelOff)]],
                                          np.int32)
mask_limit = np.array([[
   (0, 0),
   (0, self.horizon*lane_depth),
   (self.width, self.horizon*lane_depth),
   (self.width, 0)]], np.int32)

cv2.fillPoly(mask, carPathSpace, color=(0, 0, 255))
cv2.fillPoly(mask, mask_limit, 0)
```

1.6 Προειδοποίηση οδηγού (§3.3.1 βήμα 6ο και §3.3.2 βήμα 2ο):

Προκειμένου να ενεργοποιηθεί το υποσύστημα ορίστηκε να έχει αναγνωρίσει διαχωριστική γραμμή λωρίδας κυκλοφορίας, ανά πλευρά του οχήματος, για κάποια συνεχόμενα στιγμιότυπα. Στη συνέχεια, όταν μια από τις ευθείες που ορίζουν τις διαχωριστικές γραμμές της λωρίδας κυκλοφορίας εισέρχεται στην ζώνη πορείας του οχήματος αποστέλλεται σήμα προειδοποίησης στον οδηγό.

```
a1 = (averaged_lines[0][0][0], averaged_lines[0][0][1])
resultLeftBot = cv2.pointPolygonTest(carPathSpace, a1, False)
b1 = (averaged_lines[1][0][0], averaged_lines[1][0][1])
resultRightBot = cv2.pointPolygonTest(carPathSpace, b1, False)

if averaged_lines[0] != [[0, 0, 0, 0]]:  # if in the frame a lane
                                         is found
  # if previous x(thresh) frames had the lanes
  if self.LLFrames > laneLockThresh:
    self.reachedleft = True
  else:  # else add it to the list
    self.LLFrames = self.LLFrames+1
else:  # if there is no lane in the frame
  if self.LLFrames > 0:  # if previous frames had lane
    self.LLFrames = self.LLFrames-1# remove a frame from history

if self.LLFrames == 0:
  self.reachedleft = False
if averaged_lines[1] != [[0, 0, 0, 0]]:
  if self.RLFrames > laneLockThresh:
    self.reachedright = True
  else:  # else add it to the list
    self.RLFrames = self.RLFrames+1
else:  # if there is no lane in the frame
  # print('no right')
  if self.RLFrames > 0:  # if previous frames had lane
    self.RLFrames = self.RLFrames-1  # remove a frame from
                                       history
if self.RLFrames == 0:
  self.reachedright = False

if self.reachedright:
  mask = warn(resultRightBot, mask)
```

```
  cv2.putText(mask, 'R', (60, 60), cv2.FONT_HERSHEY_SIMPLEX,
            2, (0, 255, 0), 3, cv2.LINE_AA)
if self.reachedleft:
  mask = warn(resultLeftBot, mask)
  cv2.putText(mask, 'L', (20, 60), cv2.FONT_HERSHEY_SIMPLEX,
            2, (0, 255, 0), 3, cv2.LINE_AA)
if averaged_lines is None:
  return np.zeros_like(img)
```

1.7 Απενεργοποίηση της προειδοποίησης σε χαμηλές ταχύτητες και σε περιπτώσεις ηθελημένης αλλαγής πορείας (§3.3.2 βήμα 4ο και 5ο):

Σε περίπτωση που το όχημα κινείται με χαμηλές ταχύτητες, δηλαδή μικρότερες από 20 Km/h και σε περίπτωση που ο οδηγός ενεργοποιήσει τα φώτα δείκτη κατεύθυνσης (φλας) τότε το υποσύστημα απενεργοποιείται.

```
if arg == 1 and ( not tsign_left and not tsign_right)
                                    and carSpeed > 20:
      print("LINE CROSSED")
      if not mixer.get_busy():
         sound.play()
      beep.play(1)
```

2. Υποσύστημα προειδοποίησης μπροστινής σύγκρουσης

2.1 Οριοθετημένα «κουτιά» (§3.4.1, βήμα 1ο):

Από την ανίχνευση των αντικειμένων στην εικόνα λαμβάνονται συντεταγμένες πολλών «κουτιών», που ορίζουν τις θέσεις και διαστάσεις των αντικειμένων.

```
ymin = int(max(1, (boxes[i][0] * Model.imH)))
xmin = int(max(1, (boxes[i][1] * Model.imW)))
ymax = int(min(Model.imH, (boxes[i][2] * Model.imH)))
xmax = int(min(Model.imW, (boxes[i][3] * Model.imW)))

# find area,center,lenghts
xlenght = xmax - xmin
ylenght = ymax - ymin
area = (xlenght) * (ylenght)
center = xmin + (xlenght) / 2
centery = ymin + (ylenght) / 2
```

2.2 Εύρεση αντικειμένων στην πορεία του οχήματος, επιλογή σημαντικότερου και αναγνώριση αλλαγής σημαντικότερου οχήματος (§3.4.1, βήμα 2ο, §3.4.1, βήμα 3ο, §3.4.1, βήμα 3ο):

Από τη συνάρτηση αυτή περνούν διαδοχικά όλα τα αντικείμενα και στο τέλος διαπιστώνεται ποιο είναι το σημαντικότερο. Αρχικά υπολογίζεται το βάρος W, το οποίο και λαμβάνεται υπόψη στην αξιολόγηση των αντικειμένων σχετικά με το αν βρίσκονται στην πορεία του οχήματος και στη συνέχεια επιλέγεται το κατώτερο στην οπτική της κάμερας. Στην συνέχεια ελέγχεται το κέντρο του σημαντικού οχήματος με το κέντρο του σημαντικού αντικειμένου από το προηγούμενο στιγμιότυπο αναγνωρίζοντας έτσι αν το όχημα είναι νέο.

```
def getObjOfIntrest(self):
  # To find the object that are in the straight path of the car
  # Calculate a weight for every car and vru based on the offset
              from the center but also the size of the object
    weight = ylenght * weight_size / (abs(vwidth / 2+ camOffset -
                          center) + 0.5) * weight_center
  # if the weight larger than paththresh and
  if weight >= inPathSens:
    # mark them red
    self.inPath=True
    # Select closest car as car of interest
    # (after every car has passed this function only the closest
                                  car will matter)
    # (a car that is closer is also lower in the 2d perspective
                                  projection plane)
    # based on the bottom of the object rectangle
    if ymax > self.COIymax:
      # check if the center of the car compared to the
                previous car of interest is above thresh
      # if a new car gets in front slowly then the center of
                                  the coi will change
      if abs(center - self.oldCOIcenter) > newObjThresh or
          abs(centery - self.oldCOIcentery) > newObjThresh*2:
        self.newCar = True  # then it is a new car
      else:
        self.newCar = False
      # add all values as coi values
      self.COIymax = ymax
      self.COIxmin = xmin
```

```
            self.COIymin = ymin
            self.COIxmax = xmax
            self.COIcenter = center
            self.COIcentery = centery
            self.COIylenght = ylenght
            self.COIxlenght = xlenght
            self.COIarea = area
        else:
            self.inPath=False
```

2.3 Παρακολούθηση μεγέθους και Προειδοποίηση (§3.4.2, βήμα 1ο και §3.4.2, βήμα 2ο):

Αρχικά στο σημαντικότερο αντικείμενο γίνεται αναζήτηση σημαντικών σημείων και υπολογίζεται η μέση απόστασή τους. Εφόσον δεν έχει γίνει αλλαγή σημαντικότερου αντικειμένου στο προηγούμενο στιγμιότυπο τότε υπολογίζεται η μεταβολή του μεγέθους του κουτιού του στους δύο άξονες και η μεταβολή της μέσης απόστασης των σημαντικών σημείων. Στην συνέχεια αν οι μεταβολές είναι μεγαλύτερες από συγκεκριμένα όρια (ρυθμίσιμες μεταβλητές) γίνεται προειδοποίηση του οδηγού.

```
def objColWarn(self):
    # Find if there is a collision or departure if stoped by
    # mesuring the change in size of the car in front
    #(coi) and comparing it to a threshhold but also taking
    # into account if threre is a new car
    self.objTracking() # Track coi using points
    if self.newCar:  # checked if new car in pt1
        # remove old coi area and mark new car as new coi
        self.oldCOIarea = 0
        self.oldCOIxlenght = 0
        self.oldCOIylenght = 0
        self.oldCOIareaDep = 0
    else:
        # car locked as car of intrest
        self.checkNew=True  # check if points have lost track

        if self.oldCOIylenght != 0 and self.oldCOIxlenght != 0 and
                                        carSpeed >= 5:
            yChangeP = (self.COIylenght - self.oldCOIylenght)/
                                        self.oldCOIylenght
```

```
        xChangeP = (self.COIxlenght - self.oldCOIxlenght)/
                                    self.oldCOIxlenght

        if self.COIxlenght>150 and (yChangeP > CollisionSens or
                            xChangeP > CollisionSens) and
                        self.trackdistChange <- trackingSens:
            print("COLLISION ALERT")
            frameout = warn(2,frameout)

    self.oldCOIarea = self.COIarea  # make new values old
    self.oldCOIxlenght = self.COIxlenght
    self.oldCOIylenght = self.COIylenght
# save the new center so we can check for new car in next frame
self.oldCOIcenter = self.COIcenter
self.oldCOIcentery = self.COIcentery
```

3. Υποσύστημα προειδοποίησης εκκίνησης προπορευόμενου οχήματος

3.1 Παρακολούθηση μεγέθους και προειδοποίηση (§3.5.2):

Όταν το μέγεθος του προπορευόμενου οχήματος είναι αρκετά μεγάλο (δηλαδή είναι κοντά μας) και το δικό μας όχημα είναι σταματημένο, τότε γίνεται έναρξη ελέγχου του μεγέθους (εφόσον το προπορευόμενο όχημα απομακρύνεται το μέγεθός του μειώνεται). Αν αυτό συνεχιστεί για δύο (2) δευτερόλεπτα (departureSens) και εφόσον το όχημά μας παραμένει ακίνητο γίνεται προειδοποίηση του οδηγού για την εκκίνηση του προπορευόμενου οχήματος.

```
if self.oldCOIarea!= 0 and self.COIarea != 0 and
            self.COIarea >  departureThresh and carSpeed <= 5:
    if not self.Dep:
        self.Dep=True
        self.frDep = 0

if self.Dep:
    if self.trackdistChange > 0.015 and carSpeed <= 5:
        if self.frDep >= departureSens:
            print("Departure Alert")
            frameout = warn(3,frameout)
            self.Dep=False
        self.oldCOIareaDep = self.COIarea
```

```
        self.oldCOlarea = self.COlarea
        self.frDep = self.frDep + 1
        self.cancel=0
    else:
        self.cancel=self.cancel+1
        if self.cancel > 5:
            self.Dep=False
            self.cancel=0
self.oldCOlareaDep = self.COlarea
self.oldCOlarea = self.COlarea
```

4. Υποσύστημα προειδοποίησης παραβίασης ερυθρού σηματοδότη

4.1 Αποκοπή τμήματος εικόνας από τα οριοθετημένα «κουτιά» σηματοδότη (§3.6.1.1 βήμα 1ο, 2ο και 3ο):

Από την ανίχνευση αντικειμένων στην εικόνα παίρνουμε πολλές συντεταγμένες «κουτιών» που ορίζουν τις θέσεις και διαστάσεις αντικειμένων που πιθανότατα εντάσσονται στην κατηγορία σηματοδότες. Στη συνέχεια γίνεται αποκοπή από την εικόνα ενός τμήματός της, το οποίο περιέχει στο κέντρο του το «κουτί» του σηματοδότη και είναι διπλάσιο σε διαστάσεις από αυτό. Το τμήμα αυτό περνά από ένα πράσινο και από ένα κόκκινο φίλτρο για να βρεθεί το χρώμα του ενεργού λαμπτήρα του σηματοδότη.

```
ymin = int(max(1, (boxes[i][0] * Model.imH)))
xmin = int(max(1, (boxes[i][1] * Model.imW)))
ymax = int(min(Model.imH, (boxes[i][2] * Model.imH)))
xmax = int(min(Model.imW, (boxes[i][3] * Model.imW)))
# find area,center,lenghts
xlenght = xmax - xmin
ylenght = ymax - ymin
area = (xlenght) * (ylenght)
center = xmin + (xlenght) / 2
centery = ymin + (ylenght) / 2
if object_name == "traffic light":
  getLightColorWindows(frame)

def getLightColorWindows(self,frame):
  crop_img = frame[abs(ymin - int(ylenght/2)):ymax +
          int(ylenght/2), abs(xmin - xlenght):xmax + xlenght]
```

```
  crop_imgInv = cv2.cvtColor(crop_img, cv2.COLOR_BGR2RGB)
  hsv_imgInv = cv2.cvtColor(crop_imgInv, cv2.COLOR_BGR2HSV)
  hsv_img_red = cv2.inRange(hsv_imgInv, color_MINr, color_MAXr)
  hsv_img_green = cv2.inRange(hsv_imgInv, color_MINg, color_MAXg)
  return hsv_img_red, hsv_img_green
```

4.2 Εύρεση χρώματος απλού σηματοδότη (§3.6.1.1 βήμα 4ο):

Μετά από άθροιση των δυαδικών ψηφίων (0, 1) των αποτελεσμάτων του φιλτραρίσματος προκύπτουν δύο τιμές οι οποίες συγκρίνονται και η μεγαλύτερη δηλώνει το χρώμα του σηματοδότη.

```
def checkNormalight(self,hsv_img_red, hsv_img_green):
  numRed = cv2.countNonZero(hsv_img_red)
  numGreen = cv2.countNonZero(hsv_img_green)

  if ( numRed > lightthresh) or ( numGreen > lightthresh):
     self.isActualLight = True
     if numGreen < numRed:
        self.lightcolor = False
     else:
        self.lightcolor = True
```

4.3 Εξαγωγή σημαντικού χρώματος Διπλού σηματοδότη (§3.6.1.2):

Όπως έχει αναφερθεί όταν τα αποτελέσματα του φιλτραρίσματος ταυτίζονται δηλαδή οι σηματοδότες έχουν το ίδιο χρώμα τότε ο διπλός σηματοδότης λαμβάνεται υπόψη πρακτικά ως μεμονωμένος. Όταν όμως τα αποτελέσματα του φιλτραρίσματος δεν ταυτίζονται (δηλαδή είναι και τα δυο θετικά), δηλαδή υπάρχουν και τα δυο χρώματα, τότε πρόκειται για διπλό σηματοδότη, όπου ο ένας είναι κόκκινου χρώματος και ο άλλος πράσινου χρώματος. Στη συνέχεια διαχωρίζουμε την εικόνα του κόκκινου φίλτρου στη μέση και παίρνουμε δύο υποφίλτρα που αντιστοιχούν στην κάθε πλευρά του κόκκινου φίλτρου. Μετά από άθροιση των δυαδικών ψηφίων (0, 1) των αποτελεσμάτων του φιλτραρίσματος προκύπτουν δύο τιμές για κάθε μεμονωμένο σηματοδότη οι οποίες συγκρίνονται μεταξύ τους και η μεγαλύτερη δηλώνει το χρώμα του κάθε σηματοδότη, οπότε αναγνωρίζεται το χρώμα του σηματοδότη ανά κατεύθυνση (Εικόνα 3.48). Στην συνέχεια αν έχουν ενεργοποιηθεί τα φώτα δείκτη κατεύθυνσης (φλας) τότε το σύστημα θα λάβει τον διπλό σηματοδότη ως κόκκινο ενώ αν δεν έχουν ενεργοποιηθεί τα φώτα δείκτη κατεύθυνσης (φλας) ως πράσινο. (Πιθανή βελτίωση περιγράφεται στην §6.2.1.3)

```
def checkDoubleLight(self,hsv_img_red, hsv_img_green):
  doubleleft, doubleright = False, False
```

```
  if cv2.countNonZero(hsv_img_red) > lightthresh and
                cv2.countNonZero(hsv_img_green) > lightthresh:
     self.doublelight = True
     self.isActualLight = True
     hsv_img_red_left = hsv_img_red[0:hsv_img_red.shape[0],
                           0:int(hsv_img_red.shape[1] / 2)]
     hsv_img_red_right = hsv_img_red[0:hsv_img_red.shape[0],
           int(hsv_img_red.shape[1] / 2):hsv_img_red.shape[1]]
     if cv2.countNonZero(hsv_img_red_left) >
                        cv2.countNonZero(hsv_img_red_right):
        self.lightxt = "<Red|Green^"
        doubleleft = True
        doubleright = False
     else:
        self.lightxt = "^Green|Red>"
        doubleleft = False
        doubleright = True
  else:
     self.doublelight = False
  if tsign_left and doubleleft:
     self.lightcolor = False
  elif tsign_right and doubleright:
     self.lightcolor = False
  else:
     self.lightcolor = True
```

4.4 Επιλογή σημαντικότερου σηματοδότη (§3.6.1.3):

Ο σημαντικότερος σηματοδότης είναι ο πιο κοντινός, δηλαδή αυτός του οποίου, με χρήση προοπτικής, οι συντεταγμένες της πάνω πλευράς του «κουτιού» του είναι ψηλότερα από τις αντίστοιχες συντεταγμένες των πάνω πλευρών των «κουτιών» όλων των υπολοίπων σηματοδοτών. Ο κοντινότερος διπλός σηματοδότης τίθεται ως σημαντικότερος ακόμα και αν ένας μεμονωμένος λόγω προοπτικής φαίνεται λίγο πιο κοντά.

```
def getLOI(self,ymin):
  if not self.prevdoublelight:
     if self.doublelight and not self.lightcolor:
        self.prevdoublelight = True
     if ymin < self.LOIymin or self.prevdoublelight:
        self.setLOI()
```

```
    else:
        if self.doublelight and ymin < self.LOIymin:
            self.setLOI()
def setLOI(self):
    if not self.lightcolor:
        self.LOIcolor = False
    else:
        self.LOIcolor = True
    self.LOIymin = ymin
    self.LOIymax = ymax
    self.LOIxmin = xmin
    self.LOIxmax = xmax
    self.LOIcenterx = center
    self.LOIcentery = centery
```

4.5 Προειδοποίηση οδηγού σε έλλειψη πέδησης (§3.6,2 βήμα 1ο):

Όταν γίνει αναγνώριση κόκκινου σημαντικού σηματοδότη τότε το υποσύστημα ξεκινά τον έλεγχο της ταχύτητας του οχήματος. Αν το όχημα πλησιάσει αρκετά κοντά στον σημαντικό κόκκινο σηματοδότη, δηλαδή αν ο ερυθρός σηματοδότης είναι στα όρια εξόδου από το οπτικό πεδίο του Συστήματος και ταυτόχρονα υπάρχει έλλειψη πέδησης, τότε γίνεται συνεχώς προειδοποίηση του οδηγού έως ότου το όχημα περάσει τον σηματοδότη ή έως ότου ο οδηγός προβεί σε πέδηση του οχήματος (όποιο συμβεί πρώτο).

```
def runRedWarn(self):
    # desc: Check if red has reached threshhold and if the position
    #   of the light of intrest is closing on the edge and
    #       deceleration is not detected Warn the driver

    #check if is red light and the position of the light of intrest
                                    is closing on the edge
    if self.reachedr and carSpeed > 5 and
            (vwidth-egdeThresh <= self.LOIcenterx or self.LOIcenterx
                    <= egdeThresh or egdeThresh >= self.LOIcentery):
        self.redWarn = True
        # store the last time a red light was detected
        self.warnStart = cv2.getTickCount()
            # calculate the time before the vehicle has passed
                                    the light based on speed
        timeDistLight = minDistLight / (carSpeed / 3.6)
```

```
    else:
        self.redWarn = False

    warncheck = cv2.getTickCount()
    # find the time since a red light was detected
    self.warntime = (warncheck - self.warnStart) / freq
    # if Deceleration is bellow a threshold and (red light is
            detected or time since a red light was detected is
    # smaller than the time it takes to pass the light) then warn
                                        the driver
    if (self.redWarn or self.warntime <= timeDistLight) and
                                    carAccel > accelthresh:
        print("RUN RED ALERT")
        frameout = warn(4,frameout)
```

4.6 Καθυστέρηση ενεργοποίησης προειδοποίησης (§3.6,2 βήμα 1ο):

Το υποσύστημα θα πρέπει να αναγνωρίσει συγκεκριμένο χρώμα σηματοδότη για μερικά συνεχόμενα στιγμιότυπα (εικόνες) ώστε να ενεργοποιηθεί η προειδοποίηση και να μην αναγνωρίσει συγκεκριμένο χρώμα σηματοδότη για μερικά συνεχόμενα στιγμιότυπα (εικόνες) ή να αναγνωρίσει το αντίθετο χρώμα για αλλά τόσα στιγμιότυπα (εικόνες) για να απενεργοποιηθεί η προειδοποίηση (αν είναι πράσινο δεν ενδιαφέρει ωστόσο υπάρχει πρακτικά ως δυνατότητα).

```
def threshColor(self):
    # desc:  Average many frames of light detection to remove random
            noise by using a single value (self.oldLOIcolor)
    #       with pos being green, neg red and zero no light
    if self.isActualLight:  # if in the frame a light is found
        if self.LOIcolor:  # if the most important light is green
            # if previous x(thresh) frames had the same light
            if self.oldLOIcolor > lightChangeThresh:
                self.reachedg = True
            else:  # else add it to the list
                self.oldLOIcolor = self.oldLOIcolor+1
        else:  # if that light is red
            # if previous x(thresh) frames had the same light
            if self.oldLOIcolor < -lightChangeThresh:
                self.reachedr=True
            else:  # else add it to the list
```

```
            self.oldLOIcolor = self.oldLOIcolor-1
    else:  # if there is no light in the frame
        if self.oldLOIcolor > 0:# if previous frames had green light
            self.oldLOIcolor = self.oldLOIcolor-1  # remove a frame
                                                   from history
        if self.oldLOIcolor < 0:  # if previous frames had red light
            self.oldLOIcolor = self.oldLOIcolor+1  # remove a frame
                                                   from history
    if self.oldLOIcolor == 0:
        self.reachedg = False
        self.reachedr = False
```

Βιβλιογραφία

[1]: Στατιστική Επετηρίδα, (τελευταία επίσκεψη 20.01.2023)

[2]: Learn the facts about new cars: Why newer cars are safer than ever before, (τελευταία επίσκεψη 20.01.2023)

[3]: Global impact of COVID-19 pandemic on road traffic collisions/World Journal of Emergency Surgery, (τελευταία επίσκεψη 20.01.2023)

[4]: Statistics of Road Traffic Accidents in Europe and North America, 2021, (τελευταία επίσκεψη 20.01.2023)

[5]: Road Safety Thematic Report – Driver distraction, (τελευταία επίσκεψη 20.01.2023)

[6]: Visual-Manual Driver Distraction Guidelines for Portable and Aftermarket Devices, by the National Highway Traffic Safety Administration on Dec 5, 2016, (τελευταία επίσκεψη 20.01.2023)

[7]: Regulation (EU) 2019/2144 of the European Parliament and of the Council of 27 November 2019 on type-approval requirements for motor vehicles and their trailers, and systems, components and separate technical units, (τελευταία επίσκεψη 20.01.2023)

[8]: Crash cost estimates for European countries, 2017, (τελευταία επίσκεψη 20.01.2023)

[9]: EURO NCAP Latest Safety Ratings,(τελευταία επίσκεψη 20.01.2023)

[10]: Automotive electronics revolution requires faster, smarter interfaces, (τελευταία επίσκεψη 20.01.2023)

[11]: Συστήματα υποβοήθησης οδηγού -ADAS, (τελευταία επίσκεψη 20.01.2023)

[12]: VEHICLES IN USE EUROPE 2023,(τελευταία επίσκεψη 16.02.2023)

[13]: Canalys: 8% of new cars in Europe sold with level 2 autonomy driving features, (τελευταία επίσκεψη 20.01.2023)

[14]: Canalys: Autonomous driving starts to hit mainstream as 3.5 million new cars had Level 2 features in Q4 2020,(τελευταία επίσκεψη 20.01.2023)

[15]: The 6 Levels of Vehicle Autonomy Explained,(τελευταία επίσκεψη 20.01.2023)

[16]: Τεχνητή νοημοσύνη, Γεωργούλη Αικατερίνη, 2015,
(τελευταία επίσκεψη 20.01.2023)

[17]: The Concept of Artificial Neurons (Perceptrons) in Neural Networks,
(τελευταία επίσκεψη 20.01.2023)

[18]: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way
(τελευταία επίσκεψη 20.01.2023)

[19]: Fundamental Concepts of Convolutional Neural Network,
(τελευταία επίσκεψη 20.01.2023)

[20]: Dive into Deep Learning,(τελευταία επίσκεψη 20.01.2023)

[21]: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, (τελευταία επίσκεψη 20.01.2023)

[22]: MobileNetV2: Inverted Residuals and Linear Bottlenecks,
(τελευταία επίσκεψη 20.01.2023)

[23]: Searching for MobileNetV3, (τελευταία επίσκεψη 20.01.2023)

[24]: Searching for Activation Functions,(τελευταία επίσκεψη 16.02.2023)

[25]: Gradient, (τελευταία επίσκεψη 20.01.2023)

[26]: Ψηφιακή επεξεργασία εικόνας, Rafael C. Gonzalez, Richard E. Woods, Εκδόσεις Τζιόλα, 4η έκδοση, 2018

[27]: Canny Edge Detection Step by Step in Python — Computer Vision,
(τελευταία επίσκεψη 20.01.2023)

[28]: Canny Edge Detection. (τελευταία επίσκεψη 20.01.2023)

[29]: Hough transform, (τελευταία επίσκεψη 20.01.2023)

[30]: Use Of The Hough Transformation To Detect Lines And Curves In Pictures, By: Richard O. Duda, Peter E. Hart,(τελευταία επίσκεψη 20.01.2023)

[31]: Shi-Tomasi Corner Detection, (τελευταία επίσκεψη 16.02.2023)

[32]: Lucas–Kanade method, (τελευταία επίσκεψη 16.02.2023)