



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΪΑΤΡΙΚΗ**

**Αναγνώριση προτύπων – αντικειμένων με εφαρμογή σε συστήματα  
οδικής κυκλοφορίας.**

**Καποθανάσης Ευάγγελος**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων**  
**Αναγνωστόπουλος Ιωάννης**

**Λαμία, 2023**



**UNIVERSITY OF THESSALY**

**SCHOOL OF SCIENCE**

**INFORMATICS AND COMPUTATIONAL BIOMEDICINE**

**Identification of standards – objects with application in road  
traffic systems.**

**Kapothanasis Evangelos**

**Master thesis**

**Anagnostopoulos Ioannis**

**Lamia**

**2023**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ  
ΚΑΤΕΥΘΥΝΣΗ**

**«ΥΠΟΛΟΓΙΣΤΙΚΗ ΙΑΤΡΙΚΗ ΚΑΙ ΒΙΟΛΟΓΙΑ»**

**Αναγνώριση προτύπων – αντικειμένων με εφαρμογή σε  
συστήματα οδικής κυκλοφορίας.**

**Ευάγγελος Καποθανάσης**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων**

**Ιωάννης Αναγνωστόπουλος**

**Λαμία, 2023**

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Αναγνώριση προτύπων – αντικειμένων με εφαρμογή σε συστήματα οδικής κυκλοφορίας» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Ημερομηνία

Υπογραφή

**Αναγνώριση προτύπων – αντικειμένων με εφαρμογή σε  
συστήματα οδικής κυκλοφορίας.**

**Καποθανάσης Ευάγγελος**

**Τριμελής Επιτροπή:**

Αναγνωστόπουλος Ιωάννης,

Δελήμπασης Κωνσταντίνος

Ραζής Γεράσιμος

**Επιστημονικός Σύμβουλος:**

Ιωάννης Αναγνωστόπουλος

## Περιεχόμενα

Περίληψη .....	8
Abstract .....	9
1 Εισαγωγή .....	10
2 Αναγνώριση Προτύπων & Dashcam. ....	12
2.1 Εισαγωγή.....	12
2.2 Αναγνώριση προτύπων και μέθοδοι ταξινόμησης.....	12
2.3 Αλγόριθμοι αναγνώρισης προτύπων.....	16
2.4 Δομή συστήματος ταξινόμηση προτύπων. ....	19
2.5 Σύστημα Dashcam .....	21
2.5.1 Τι είναι το σύστημα Dashcam; .....	21
2.5.2 Αλγόριθμοι λειτουργίας συστημάτων Dashcam.....	24
2.5.3 Σύστημα ADAS (Advanced Drive – Assistance Systems).....	25
3 Μηχανική μάθηση.....	29
3.1 Εισαγωγή.....	29
3.2 Τεχνητή νοημοσύνη .....	31
3.3 Νευρωνικά Δίκτυα .....	38
4 Προτεινόμενο σύστημα .....	53
4.1 Εισαγωγή.....	53
4.2 Υλικοτεχνικός εξοπλισμός – Ανάλυση.....	53
4.3 Τεχνολογίες λογισμικών .....	63
4.3.1 Λίγα λόγια για τη γλώσσα Python .....	63
4.3.2 OpenCV .....	64
4.3.3 YOLO V5.....	67
4.4 Object detection project .....	70
5 Συμπεράσματα – Μελλοντική έρευνα .....	77
5.1 Εισαγωγή.....	77
5.2 Συμπεράσματα .....	77
Βιβλιογραφία .....	79
Παράρτημα Ι .....	81

## Περίληψη

Η χρήση της τεχνολογίας στα συστήματα οδικής κυκλοφορίας έχει αλματώδη ανάπτυξη τα τελευταία χρόνια. Πλέον, όλα τα οχήματα παραγωγής επαγγελματικά και μη, φέρουν αρκετά έξυπνα συστήματα τα οποία μπορούν να επεκταθούν λειτουργικά κατ' επιλογή του ιδιοκτήτη. Κάποια από αυτά τα συστήματα είναι οι εγκατεστημένες κάμερες πορείας και οπισθοπορείας οι οποίες μπορούν να προβλέπουν την κίνηση του οχήματος και τα εμπόδια, συστήματα GPS και εύρεσης γρήγορης διαδρομής, συστήματα αναγνώρισης πινακίδων και εμποδίων, καθώς επίσης συστήματα έγκαιρης προειδοποίησης και καταγραφής. Όλα τα εξελιγμένα συστήματα, έχουν θετικό αποτέλεσμα στην κίνηση των οχημάτων, στην αποτροπή ατυχημάτων και στην κυκλοφοριακή συμφόρηση. Απώτερος σκοπός της εργασίας, είναι η ανάπτυξη ενός συστήματος αναγνώρισης προτύπων και εμποδίων με πλατφόρμες raspberry pi όπου θα μπορούσε να εφαρμοστεί στην κάμερα (Dashcam) ενός οχήματος. Επίσης, πραγματοποιείται εκτενής αναφορά σε συστήματα αναγνώρισης προτύπων, στο machine learning (μηχανική μάθηση), στα νευρωνικά δίκτυα και την τεχνητή νοημοσύνη (AI).

Λέξεις κλειδιά: Αναγνώριση προτύπων, μηχανική μάθηση, νευρωνικά δίκτυα, τεχνητή νοημοσύνη, Dashcam



## Abstract

The use of technology in road traffic systems has been developing rapidly in recent years. Now, all commercial and non-commercial production vehicles have several smart systems that can be expanded operationally at the choice of the owner. In some of these systems cameras and GPS systems are installed that can predict vehicle movement and obstacles and signs, as well as early warning and recording systems. All advanced systems have a positive effect on vehicle traffic, accident prevention and congestion. The dissertation aim at the development of a system for identifying patterns and barriers through raspberry pi platforms applied to a vehicle. Also this dissertation, provides an extensive analysis for pattern recognition systems, machine learning, neural networks and artificial intelligence (AI).

**Key words:** Pattern recognition, machine learning, neuronic networks, artificial intelligence, Dashcam

# 1 Εισαγωγή

## Σκοπός και στόχοι της διπλωματικής εργασίας.

Σκοπός της παρούσας διπλωματικής εργασίας, είναι η ανάπτυξη ενός συστήματος αναγνώρισης προτύπων με έμφαση στην οδική κυκλοφορία. Μέσα από την ανάπτυξη της διπλωματικής εργασίας και των επιμέρους κεφαλαίων, ο αναγνώστης θα λάβει πλήρη ενημέρωση και πληροφορίες, σχετικά με τον τρόπο αναγνώρισης προτύπων, τους αλγόριθμους που τρέχουν σε ένα τέτοιο σύστημα, τη μηχανική μάθηση και τα τεχνητά νευρωνικά δίκτυα.

Στη παρούσα διπλωματική εργασία, θα παρουσιάσουμε αναλυτικά πως γίνεται η αναγνώριση προτύπων με παραδείγματα, τις μεθόδους ταξινόμησης, τους αλγόριθμους αναγνώρισης προτύπων, τη δομή ενός συστήματος ταξινόμησης προτύπων καθώς επίσης και το σύστημα Dashcam. Επίσης, θα αναφερθούμε στη τεχνητή νοημοσύνη και τα νευρωνικά δίκτυα και θα δούμε με εικόνες και παραδείγματα, ένα σύστημα αναγνώρισης προτύπων υλοποιημένο στη πλατφόρμα Raspberry Pi 4B.

Στην παρούσα διπλωματική εργασία, θα αναλυθεί ο τρόπος με τον οποίο λειτουργεί ένα σύστημα αναγνώρισης προτύπων. Πιο συγκεκριμένα, θα αναλύσουμε τα επιμέρους στοιχεία του συστήματος, τους αλγόριθμους, το λογισμικό (software) και το υλικό (hardware) του συστήματος, καθώς επίσης τον τρόπο επεξεργασίας των πληροφοριών (εικόνες – βίντεο).

Στόχος της παρούσα διπλωματικής εργασίας, είναι να δώσει μια πλήρη εικόνα σχετικά με τη δομή και λειτουργία ενός συστήματος αναγνώρισης προτύπων με έμφαση στην οδική κυκλοφορία.

Η σημασία της διπλωματικής εργασίας προσδίδεται, από το γεγονός ότι μας δίνει μια ολοκληρωμένη εικόνα αναφορικά με τα συστήματα αναγνώρισης προτύπων και μας δείχνει τη δυναμική που έχουν αυτά τα συστήματα. Τέλος, απώτερος σκοπός της είναι να βοηθήσει ώστε να ξεπεραστούν οποιεσδήποτε δυσκολίες και περιορισμοί, ώστε η αναγνώριση προτύπων στα συστήματα εποπτείας οδικής κυκλοφορίας, να δώσουν ουσιώδεις λύσεις σε πολλά προβλήματα του ευρύ κοινωνικού συνόλου.

## Υλικοτεχνικός εξοπλισμός και λογισμικό που χρησιμοποιήθηκε.

Για τη δημιουργία του project χρησιμοποιήθηκε κατάλληλος τεχνικός εξοπλισμός και έγινε χρήση συγκεκριμένων λογισμικών. Παρακάτω, θα δούμε τη λίστα με το σύνολο του υλικοτεχνικού εξοπλισμού και των απαραίτητων λογισμικών.

- Τεχνικός εξοπλισμός (Hardware):
  - Raspberry Pi 4B
  - Οθόνη Touch 7’’
  - AC Adaptor
  - Micro SD 64GB
- Λογισμικό (Software):
  - Raspberry Pi OS
  - OpenCV
  - YOLOv5
- Γλώσσες προγραμματισμού:
  - Python

## Δομή της διπλωματικής εργασίας.

Το **πρώτο** κεφάλαιο αφορά την εισαγωγή της διπλωματικής εργασίας.

Στο **δεύτερο** κεφάλαιο, γίνεται εκτενής αναφορά σχετικά με την αναγνώριση προτύπων σε σχέση με συστήματα Dashcams. Ο αναγνώστης, θα ενημερωθεί σχετικά με το τι είναι η αναγνώριση ή ταξινόμηση προτύπων, ποια είναι η βαθμωτή συνάρτηση που αφορά τη διαδικασία ταξινόμησης και τους αλγόριθμους ταξινόμησης προτύπων. Επίσης, θα ενημερωθεί για τη δομή ενός συστήματος ταξινόμησης, για την αναγνώριση προτύπων με επίβλεψη και χωρίς επίβλεψη και τι είναι το σύστημα Dashcam που βρίσκει εφαρμογή σε οχήματα.

Στο **τρίτο** κεφάλαιο, γίνεται αναφορά στη τεχνητή νοημοσύνη και τα νευρωνικά δίκτυα. Πέραν της ανάλυσης και των δύο τμημάτων, ο αναγνώστης θα δει πως εφαρμόζονται στην αναγνώριση προτύπων και πιο συγκεκριμένα σε εφαρμογή συστημάτων οδικής κυκλοφορίας. Στο κεφάλαιο αυτό αναφέρονται πληροφορίες για τι είναι η τεχνητή νοημοσύνη και πως λειτουργεί, τι είναι τα νευρωνικά δίκτυα και πως αυτά λειτουργούν σε βιολογικό επίπεδο και σε επίπεδο συστημάτων υπολογιστών.

Το **τέταρτο** και τελευταίο κεφάλαιο, περιγράφει την υλοποιημένη εφαρμογή. Πραγματοποιείται μία εκτενής αναφορά για τον εξοπλισμό (υλικό) και το λογισμικό που χρησιμοποιήθηκε. Τέλος, αποτυπώνονται τα αποτελέσματά της, ενώ αναφέρεται ο ανοιχτός κώδικας που χρησιμοποιείται με επεξηγήσεις και ανάλυση, μέσα από πραγματικά πειράματα.

Τέλος, το **πέμπτο** και τελευταίο κεφάλαιο συνοψίζει τους στόχους και τα αποτελέσματα της διπλωματικής εργασίας, ενώ αναφέρεται και σε μελλοντικές επεκτάσεις της.

## 2 Αναγνώριση Προτύπων & Dashcam.

### 2.1 Εισαγωγή

Η **αναγνώριση προτύπων (Pattern Recognition)** είναι ένα επιστημονικό πεδίο με στόχο την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής ή διακριτικού στοιχείου σε εισαγόμενα δεδομένα, συνήθως κωδικοποιημένα ως αλληλουχίες αριθμών. Έτσι αυτά τα δεδομένα, κατηγοριοποιούνται βάση κάποιων κριτηρίων. Οι άνθρωποι σε σύγκριση με τις μηχανές, κατηγοριοποιούν τα δεδομένα τα οποία λαμβάνουν βάση της ευφυΐας τους. Μία υπολογιστική μηχανή ή ένα υπολογιστικό σύστημα κατηγοριοποιεί τα δεδομένα βάση κάποιων αλγορίθμων ταξινόμησης. Για να γίνει αυτό όμως, θα πρέπει πρώτα το σύστημα να εκπαιδευτεί, ώστε να αναγνωρίζει τα πρότυπα και να τα κατηγοριοποιεί.

Μία μηχανή ή ένα υπολογιστικό σύστημα μπορεί να αναγνωρίζει διάφορα πρότυπα όπως φωνή, δακτυλικά αποτυπώματα, εικόνες, ακόμα και στοιχειώδη σωματίδια από ηλεκτρονικά συστήματα ενός πειράματος υψηλών ενεργειών.

Η αναγνώριση προτύπων έχει αρκετά πεδία εφαρμογής όπως μηχανική όραση (Machine Vision), αναγνώριση χαρακτήρων (Character Recognition – OCR), αναγνώριση ομιλίας και προσώπου (Speech and Face Recognition), Βιομετρία (Biometrics), Βιοπληροφορική (Bioinformatics) κ.α..

Άρα αν αναφερθούμε γενικά στο τι είναι η αναγνώριση προτύπων θα μπορούσαμε να πούμε ότι είναι **μία διαδικασία ώστε να κάνουμε ένα υπολογιστικό σύστημα – έναν υπολογιστή να λειτουργεί έξυπνα στην επεξεργασία εισόδου ποικιλόμορφων δεδομένων**. Για παράδειγμα να μπορεί να αναγνωρίσει τα χαρακτηριστικά ενός προσώπου ή ενός αντικειμένου μέσα από μία φωτογραφία.

Στόχος λοιπόν της αναγνώρισης προτύπων είναι, όταν ένα υπολογιστικό σύστημα λάβει δεδομένα με υψηλές διακυμάνσεις – όπως στην επεξεργασία εικόνων εικόνες με διαφορετικά χαρακτηριστικά και διαφορετικές συνθήκες φωτισμού – να τα επεξεργάζεται και να εξαγάγει σημαντικά δεδομένα για τον άνθρωπο.

### 2.2 Αναγνώριση προτύπων και μέθοδοι ταξινόμησης

Αναγνώριση προτύπων ή ταξινόμηση προτύπων, ονομάζεται η διαδικασία κατά την οποία σήματα του περιβάλλοντος χώρου που αντιστοιχούν σε ένα αντικείμενο (πρότυπο), ταξινομούνται σε μία κατηγορία από ένα πεπερασμένο σύνολο κατηγοριών  $N$  (κατηγορίες αντικειμένων).

Οι κατηγορίες αντικειμένων αντιστοιχούν σε ομάδες αντικειμένων που έχουν κοινά χαρακτηριστικά και ιδιότητες. Η διαδικασία που ταξινομούνται τα πρότυπα παριστάνεται από μία βαθμωτή συνάρτηση, η οποία πραγματοποιεί αντιστοίχιση του χώρου μετρήσεων των προτύπων με τον χώρο  $N$  των διακριτών τιμών. Αυτό φαίνεται από την ακόλουθη συνάρτηση:

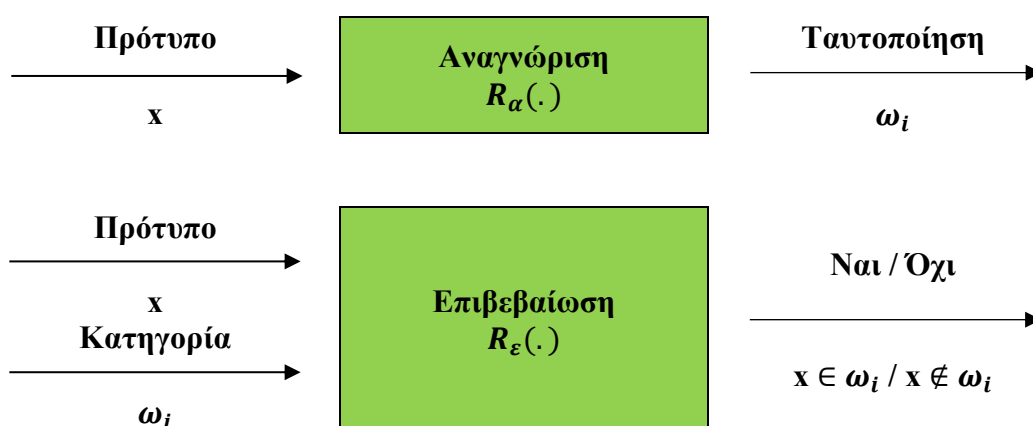
$$\omega_i = R_\alpha(\chi), R_\alpha: \mathcal{R}^P \rightarrow \{\omega_1, \omega_2, \dots, \omega_N\} \quad [2.1]$$

**Μηχανική ταξινόμηση χωρίς λήψη απόφασης** ονομάζεται η μηχανική ταξινόμηση κατά την οποία είτε θα έχει πολύ μικρή πιθανότητα λήψης λανθασμένης απόφασης είτε θα έχει πολύ μεγάλη αξιοπιστία.

Σε αυτή την περίπτωση η συνάρτηση [2.1] τροποποιείται ως εξής:

$$\omega_i = R_\alpha(\chi), R_\alpha: \mathcal{R}^P \rightarrow \{\omega_1, \omega_2, \dots, \omega_N, \omega_{N+1}\} \quad [2.2]$$

Η συνάρτηση [2.2] μας δίνει τη σχέση εισόδου – εξόδου της μηχανικής ταξινόμησης, όπου με  $\omega_{N+1}$  περιγράφεται η κατάσταση της μηχανής κατά την οποία το πρότυπο  $x$  δεν ταξινομείται σε καμία από τις  $N$  κατηγορίες αντικειμένων. Όταν θέλουμε να απορρίψουμε ή να επιβεβαιώσουμε τον ισχυρισμό πως το πρότυπο  $x$  ανήκει στην κατηγορία  $\omega_i$ , τότε αναφερόμαστε σε συστήματα επιβεβαίωσης προτύπων. Η διαφορά της διαδικασίας ταξινόμησης από την διαδικασία επιβεβαίωσης φαίνεται στα παρακάτω σχήματα:



### Ταξινόμηση προτύπων και βήματα κατασκευής ταξινόμησης.

Η ακόλουθη συνάρτηση [2.3] δύο μεταβλητών του προτύπου  $x$  και των σταθερών παραμέτρων  $w$  περιγράφει τη διαδικασία ταξινόμησης προτύπων.

$$y = R_p(x, w) \quad [2.3]$$

Ως  $W$  ορίζεται το είδος και το πλήθος των κατηγοριών που αναγνωρίζονται και ως  $X$  το πρότυπο που ενεργοποιεί το σύστημα ώστε να γίνει έναρξη της διαδικασίας ταξινόμησης σε ένα συγκεκριμένο πρότυπο. Τα βήματα ταξινόμησης προτύπων είναι τα εξής:

- Επιλογή μεθόδου για την αναγνώριση του άγνωστου προτύπου
- Εκπαίδευση συστήματος – υπολογισμός παραμέτρων  $w$
- Μέτρηση αξιοπιστίας συστήματος

Ως μεθόδους ταξινόμησης προτύπων ορίζουμε τα ακόλουθα:

- **Αιτιοκρατικές ή δομικές μέθοδοι (deterministic):** Στις αιτιοκρατικές μεθόδους γίνεται χρήση των συναρτήσεων απόστασης του αγνώστου προτύπου από τα πρότυπα κάθε μιας των κατηγοριών
- **Στοχαστικές μέθοδοι (stochastic, probabilistic):** Αναζήτηση της κατηγορίας εκείνης που μεγιστοποιεί την πιθανότητα εμφάνισής της, γνωρίζοντας την παραμετρική περιγραφή του προτύπου.

- **Μέθοδοι μη γραμμικών δικτύων ή νευρωνικά δίκτυα:** Προσομοίωση της διαδικασίας ταξινόμησης με μια συνάρτηση μεταφοράς, η οποία μπορεί να αναλυθεί σε ένα σύνολο μη – γραμμικών υπολογιστικών μονάδων.
- **Μέθοδοι ασαφών συνόλων (fuzzy recognizer):** Κατασκευή συστημάτων ταξινόμησης προτύπων, με τη χρήση στοιχείων από την θεωρία της ασαφούς λογικής και των ασαφών συνόλων.

Η εκπαίδευσης ενός συστήματος αναφορικά με την αναγνώριση και ταξινόμηση προτύπων είναι μία περίπλοκη διαδικασία. Ως εκπαίδευση ενός συστήματος ταξινόμησης ορίζουμε τη μέθοδο υπολογισμού της παραμέτρου  $w$  της συνάρτησης ταξινόμησης. Ο ρυθμός των επιτυχών ταξινομήσεων καθώς επίσης και το είδος των αντικειμένων που αναγνωρίζονται, επηρεάζονται καθοριστικά από την εκπαίδευση του συστήματος. Η επιλογή της καταλληλότερης μεθόδου εκπαίδευσης επηρεάζονται από τους ακόλουθους παράγοντες:

- ❖ Η διασπορά των προτύπων κάθε κατηγορίας.
- ❖ Η μέθοδος ταξινόμησης.
- ❖ Ο βαθμός κάλυψης του διανυσματικού χώρου των προτύπων, από περισσότερες από μία κατηγορίες.

### Διαδικασία εκπαίδευσης συστήματος

#### Κατευθυνόμενη εκπαίδευση συστήματος

Η κατευθυνόμενη εκπαίδευση αποτελεί τη μέθοδο προσδιορισμού των σταθερών παραμέτρων με την συχνότερη εμφάνιση. Στην κατευθυνόμενη εκπαίδευση η διαθέσιμη πληροφορία για την εκπαίδευση του συστήματος είναι ένας πεπερασμένος αριθμός σωστά ταξινομημένων παραδειγμάτων.

Έστω ότι έχουμε  $M$  διαθέσιμα παραδείγματα. Ακολουθώντας βλέπουμε τη συνάρτηση ταξινόμησης [2.4].

$$\Omega_{\omega} = \{(x_1, \omega_1), (x_2, \omega_2), \dots, (x_M, \omega_M)\} \quad [2.4]$$

Ανάγουμε τα παραδείγματα σε ζεύγη εισόδου – εξόδου στη συνάρτηση ταξινόμησης, επειδή ο σκοπός της εκπαίδευσης είναι ο υπολογισμός των σταθερών συντελεστών της συνάρτησης. Έτσι, η συνάρτηση τροποποιείται και γίνεται όπως φαίνεται παρακάτω – συνάρτηση [2.5].

$$\Omega_{\omega} = \{(x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)\} \quad [2.5]$$

Έστω τώρα, ότι το σύστημα διαθέτει  $N$  εξόδους και ότι το πεδίο τιμών κάθε εξόδου είναι πραγματικοί αριθμοί στο διάστημα  $[-a, a]$ ,  $a \in \mathbb{R}^+$ . Ορίζουμε ως μέθοδο ταξινόμησης την αναζήτηση της εξόδου με την μεγαλύτερη αριθμητική τιμή, όταν στην είσοδο τοποθετηθεί το παραμετρικό διάνυσμα του προτύπου.

$$y = (y_1, y_2, \dots, y_p) \quad \text{με} \quad \begin{aligned} y_j &= -a, x \notin \omega_j \\ y_j &= a, x \in \omega_j \end{aligned}$$

Σύμφωνα με τα παραπάνω, υπολογίζεται η μνήμη του συστήματος ταξινόμησης με κριτήριο την ελαχιστοποίηση του σφάλματος εκτίμησης της εξόδου για τα παραδείγματα εκπαίδευσης που είναι διαθέσιμα. Το μαθηματικό μοντέλο δίνεται από την εξίσωση [2.6], όπου  $d(R_a(x_i, w)y_i)$  είναι μια συνάρτηση σφάλματος που δίνει τον βαθμό απόκλισης της προσδοκώμενης απόκρισης του συστήματος ταξινόμησης από το πραγματικό διάνυσμα εξόδου

του συστήματος με 'μνήμη'  $w$ . Το σύμβολο  $\operatorname{argmin}_w f(w)$  δηλώνει την τιμή του διανύσματος για την οποία η συνάρτηση  $f(w)$  γίνεται ελάχιστη.

$$w' = \operatorname{argmin}_w \sum_{i=1}^M d(R_p(x_i, w), y_i) \quad [2.6]$$

### Αυτοεκπαίδευση συστήματος

Η δυσκολότερη μορφή εκπαίδευσης είναι η αυτοεκπαίδευση και αυτό γιατί η πληροφορία που είναι διαθέσιμη αφορά μόνο τη διανυσματική παράσταση  $\mathbf{M}$  προτύπων:

$$\Omega = \{x_1, x_2, \dots, x_M\}$$

Για να υπολογιστεί η μνήμη του συστήματος θα χρησιμοποιήσουμε τον ακόλουθο αλγόριθμο:

- ❖ Έστω ότι η μνήμη του συστήματος είναι γνωστή.
- ❖ Γίνεται εκτίμηση της κατηγορίας στην οποία ανήκει το κάθε πρότυπο εκπαίδευσης (υπολογισμός της συνάρτησης ταξινόμησης για κάθε ένα από τα παραδείγματα εκπαίδευσης).
- ❖ Δημιουργούνται τεχνητά παραδείγματα και επαναπροσδιορίζεται η μνήμη του συστήματος βάσει της κατευθυνόμενης εκπαίδευσης.

Η ελαχιστοποίηση της μεταβολής της μνήμης του συστήματος για δύο διαδοχικά βήματα, επιλέγεται ως κριτήριο σύγκλισης του επαναληπτικού τμήματος της μεθόδου. Οι ακόλουθες εξισώσεις εκφράζουν την επαναληπτική μέθοδο:

1. Υπολογισμός εξόδου της συνάρτησης ταξινόμησης για τα  $N$  παραδείγματα.

$$y_i^{(t)} = R_{\pi}(x_i, w^{(t)}), i = 1, N$$

2. Ταξινόμηση προτύπων.

$$\omega_i^{(t+1)} = \operatorname{argmax}_j (y_{ij}^{(t)}), i = 1, N$$

Όπου  $y_{i,j}$  είναι η συνιστώσα του διανύσματος  $y_i$ .

3. Κατασκευή των παραδειγμάτων

$$y_i^{(t+1)} = (y_{i1}, y_{i2}, \dots, y_{iN}) \quad \text{με} \quad \begin{aligned} y_i^{(t+1)} &= a, x_i \notin \omega_i^{(t+1)} \\ y_i^{(t+1)} &= -a, x_i \in \omega_i^{(t+1)} \end{aligned}$$

Για να υπολογίσουμε τη μνήμη του συστήματος αναφορικά με τη μέθοδο της κατευθυνόμενης εκπαίδευσης χρησιμοποιούμε την παρακάτω σχέση [2.7]. Για την περιγραφή της οποιασδήποτε επαναληπτικής μεθόδου, θα πρέπει να οριστούν ένα κριτήριο σύγκλισης το οποίο θα ελέγχει την συνθήκη τερματισμού του επαναληπτικού τμήματος της μεθόδου και τις αρχικές τιμές για τις μεταβλητές του.

$$w^{(t+1)} = \operatorname{argmin}_w \sum_{i=1}^M d(R_p(x_i, w^{(t)}), y_i^{(t+1)}) \quad [2.7]$$

## 2.3 Αλγόριθμοι αναγνώρισης προτύπων

Οι αλγόριθμοι για την αναγνώριση προτύπων εξαρτώνται από τον τύπο της εξόδου, σχετικά με το εάν η μάθηση είναι υπό επίβλεψη ή χωρίς και με το εάν ο αλγόριθμος είναι στατιστικός ή μη στη φύση. Οι στατιστικοί αλγόριθμοι μπορούν να κατηγοριοποιηθούν περαιτέρω ως παραγωγικοί ή διακριτικοί.

Παρακάτω θα αναφέρουμε επιγραμματικά κάποιους από τους αλγόριθμους που χρησιμοποιούνται στην αναγνώριση προτύπων.

- ❖ Αλγόριθμος ταξινόμησης
  - ❖ Αλγόριθμος ομαδοποίησης
  - ❖ Αλγόριθμος παλινδρόμησης
  - ❖ Αλγόριθμοι συλλογικής μάθησης
  - ❖ Γενικοί αλγόριθμοι
  - ❖ Αλγόριθμοι συντακτικής ανάλυσης
  - ❖ Κατηγορηματικοί αλγόριθμοι επισήμανσης ακολουθίας
- **Αλγόριθμοι ταξινόμησης**: Οι αλγόριθμοι ταξινόμησης αντιμετωπίζουν το πρόβλημα κατάταξης νέων αντικειμένων ή παρατηρήσεων (labeling) σε ήδη υπάρχουσες υποομάδες ενός δεδομένου συνόλου αντικειμένων (επιβλεπόμενη μάθηση) – Perceptron και Νευρωνικά Δίκτυα.
  - **Αλγόριθμοι ομαδοποίησης**: Οι αλγόριθμοι ομαδοποίησης αντιμετωπίζουν το πρόβλημα διαχωρισμού ενός υπάρχοντος συνόλου αντικειμένου σε κατηγορίες (clusters) με βάση κριτήρια ομοιότητας (ανεπίβλεπτη μάθηση) – Ιεραρχική ομαδοποίηση, K-μέσων ομαδοποίηση.
  - **Αλγόριθμοι παλινδρόμησης**: Οι αλγόριθμοι παλινδρόμησης προβλέπουν την τιμή μίας μεταβλητής συνεχών τιμών συναρτήσει άλλων μεταβλητών. Για παράδειγμα έχουμε τη Γραμμική Παλινδρόμηση, τα Νευρωνικά Δίκτυα, Ανάλυση Κύριων Συνιστωσών PCA, Ανάλυση Ανεξάρτητων Συνιστωσών ICA.

Στη μέθοδο σύγκρισης προτύπων (pattern matching) προσδιορίζονται αντιπροσωπευτικά πρότυπα από τα παραδείγματα εκπαίδευσης. Ένα ή περισσότερα πρότυπα ανά κατηγορία αποθηκεύονται στη μνήμη του συστήματος και η ταξινόμηση πραγματοποιείται με τη διαδικασία της σύγκρισης του αγνώστου προτύπου με τα πρότυπα όλων των κατηγοριών. Οι μέθοδοι σύγκρισης των προτύπων μπορούν να διακριθούν βάσει της μεθόδου υπολογισμού των πρωτότυπων των κατηγοριών. Δηλαδή **α.** όλα τα παραδείγματα εκπαίδευσης θεωρούνται πρωτότυπα, **β.** επιλέγοντας τα πλέον αντιπροσωπευτικά παραδείγματα και **γ.** δημιουργώντας νέα αντιπροσωπευτικά πρότυπα.

Στην **έυρεση κοινών χαρακτηριστικών** γίνεται προσπάθεια εντοπισμού κοινών γνωρισμάτων, που χαρακτηρίζουν τα περισσότερα ή όλα τα πρότυπα κάθε κατηγορίας. Κατά τη διαδικασία εκπαίδευσης υπολογίζεται το σύνολο των χαρακτηριστικών γνωρισμάτων που διαφοροποιεί τα πρότυπα στις διαφορετικές κατηγορίες. Αυτές οι μέθοδοι, χρησιμοποιούνται συνήθως στις περιπτώσεις εκείνες κατά τις οποίες το παραμετρικό διάνυσμα του προτύπου είναι πολύ μεγάλων διαστάσεων. Έτσι για παράδειγμα, θα γίνει χρήση αυτών των μεθόδων στην περίπτωση των ψηφιακών εικόνων, στα δεδομένα ιατρικών διαγνώσεων, στις λέξεις φυσικής γλώσσας και σε άλλες αρκετές περιπτώσεις.



Στις γλωσσικές ή συντακτικές μεθόδους μερικά αντικείμενα μπορούν να αναλυθούν σε εάν μικρό πλήθος απλούστερων αντικειμένων, τα οποία ονομάζονται *αρχέγονα πρότυπα* (primitive patterns). Πολλά αρχέγονα πρότυπα, συνδέονται μαζί τους με σχέσεις οι οποίες χαρακτηρίζουν τις κατηγορίες των προτύπων που θέλουμε να αναγνωρίσουμε. Χαρακτηριστικά παραδείγματα που μπορούν να περιγράψουν με τέτοιες είναι στη βιολογία τα χρωμοσώματα που αποτελούνται από ένα σύνολο απλών ομάδων οργανικών μορίων.

Στην **αναγνώριση των χρονικά μεταβαλλόμενων γεγονότων**, αναφερόμαστε σε μία ιδιαίτερη κατηγορία ταξινόμησης προτύπων που σχετίζεται με χρονικά μεταβαλλόμενα γεγονότα. Ένα παράδειγμα χρονικά μεταβαλλόμενων γεγονότων είναι η ανθρώπινη ομιλία, η οποία με την πάροδο του χρόνου μπορεί να μεταβάλλεται. Έτσι, το σύστημα θα πρέπει να αναγνωρίσει μία ομιλία, η οποία ενδεχομένως να έχει αλλοιωθεί με την πάροδο του χρόνου.

Μέτρηση της αξιοπιστίας ενός συστήματος ταξινόμησης προτύπων είναι μια πολύ σημαντική διαδικασία στην αναγνώριση προτύπων. Ο υπολογισμός του σφάλματος ταξινόμησης προτύπων, αποτελεί το σημαντικότερο κριτήριο αξιολόγησης του συστήματος, γιατί μας δίνει ένα ποσοτικό μέτρο της αξιοπιστίας των αποφάσεων που λαμβάνει. Η μέτρηση του σφάλματος επηρεάζεται από α) τη μέθοδο ταξινόμησης και εκπαίδευσης, β) το σύνολο των παραδειγμάτων εκπαίδευσης και γ) το σύνολο των μετρήσεων που χρησιμοποιούνται για να μετρήσουν το σφάλμα. Η αριθμητική τιμή του σφάλματος εξαρτάται από δύο παραμέτρους  $\theta_1$  και  $\theta_2$ ,  $\Sigma\varphi = \Sigma\varphi(\theta_1, \theta_2)$ . Ως  $\theta_1$ , ορίζουμε τις παραμέτρους εκείνες που περιγράφουν την επίδραση της διαδικασίας εκπαίδευσης στην μέτρηση του σφάλματος. Αντίστοιχα ως  $\theta_2$ , ορίζουμε τις παραμέτρους εκείνες που δείχνουν την επίδραση που έχουν στη μέτρηση του σφάλματος, τα παραδείγματα που χρησιμοποιούνται γι' αυτή την μέτρηση. Αν θεωρηθεί ότι η διαδικασία εκπαίδευσης σχεδιάζεται έτσι ώστε να ελαχιστοποιεί το σφάλμα της ταξινόμησης, για τα παραδείγματα που χρησιμοποιούνται στην εκπαίδευση τότε προκύπτει η ακόλουθη ανισότητα:

$$\Sigma\varphi(\theta_1, \theta_2) \leq \Sigma\varphi(\theta_1, \theta_2)$$

Στην πράξη, κάθε διαδικασία εκτίμησης των παραμέτρων του συστήματος ταξινόμησης πραγματοποιείται με τη χρήση ενός πεπερασμένου αριθμού παραδειγμάτων εκπαίδευσης. Όταν όμως έχουμε έναν απεριόριστο αριθμό παραδειγμάτων, έχουμε και ακριβή υπολογισμό σφάλματος. Ως σφάλμα ταξινόμησης προτύπων, ορίζουμε το μέγεθος  $\Sigma\varphi(\theta, \theta)$  για το οποίο η εκπαίδευση και το σφάλμα έχουν υπολογιστεί από έναν απεριόριστο αριθμό παραδειγμάτων. Επειδή ισχύει  $\lim_{N \rightarrow +\infty} \theta_N = \theta$  αποδεικνύεται ότι:

$$\lim_{N \rightarrow +\infty} \Sigma\varphi(\theta_N, \theta_N) = \Sigma\varphi(\theta, \theta)$$

Στην εκπαίδευση με πεπερασμένο αριθμό παραδειγμάτων, παρουσιάζεται το φαινόμενο να αλλοιώνεται η ακρίβεια της μέτρησης του σφάλματος. Αυτό το γεγονός, δημιουργεί την ανάγκη για διαχωρισμό των πραγματικών παραμέτρων του συστήματος ( $\theta$ ), από τις παραμέτρους που υπολογίζονται από  $N$  παραδείγματα ( $\theta_N$ ). Αυτό έχει ως αποτέλεσμα τις παρακάτω ανισότητες:

$$\begin{aligned} \Sigma\varphi(\theta, \theta) &\leq \Sigma\varphi(\theta_N, \theta) \\ \Sigma\varphi(\theta_N, \theta_N) &\leq \Sigma\varphi(\theta, \theta_N) \end{aligned}$$

Ως  $\Sigma\varphi(\theta_N, \theta_N)$  ορίζεται το σφάλμα ταξινόμησης όταν τα ίδια  $N$  παραδείγματα χρησιμοποιηθούν και για τη μέτρηση (**C-method**) και για την εκπαίδευση, ενώ ως  $\Sigma\varphi(\theta_N, \theta)$

ορίζεται το σφάλμα ταξινόμησης όταν τα  $N$  παραδείγματα χρησιμοποιούνται στην εκπαίδευση και απεριόριστος αριθμός παραδειγμάτων χρησιμοποιούνται για τη μέτρηση του σφάλματος (**U-method**). Έστω λοιπόν ότι έχουμε  $K$  διαφορετικά σύνολα παραδειγμάτων εκπαίδευσης, όπου για κάθε ένα από αυτά ισχύουν τα ακόλουθα:

$$\begin{aligned}\Sigma\varphi(\theta, \theta) &\leq \Sigma\varphi(\theta_N^{(k)}, \theta), k = 1, K \\ \Sigma\varphi(\theta_N^{(k)}, \theta_N^{(k)}) &\leq \Sigma\varphi(\theta, \theta_N^{(k)}), k = 1, K\end{aligned}$$

Οι αντίστοιχες μέσες τιμές των παραπάνω ανισοτήτων, θα εξακολουθούν να υπακούν στις ίδιες συνθήκες:

$$\begin{aligned}E(\Sigma\varphi(\theta, \theta)) &\leq E(\Sigma\varphi(\theta_N^{(k)}, \theta)), k = 1, K \\ E(\Sigma\varphi(\theta_N^{(k)}, \theta_N^{(k)})) &\leq E(\Sigma\varphi(\theta, \theta_N^{(k)})), k = 1, K\end{aligned}$$

Εάν υποθέσουμε ότι η μέση τιμή των σφαλμάτων, υπολογιζόμενη με τη μέθοδο με τη μέθοδο  $\Sigma\varphi(\theta, \theta_N^{(k)})$  μπορεί να προσεγγίσει το θεωρητικό σφάλμα, τότε η ισότητα  $E(\Sigma\varphi(\theta, \theta_N^{(k)})) \approx \Sigma\varphi(\theta, \theta)$  περιέχει στατιστική αλήθεια. Όσο μεγαλύτερος είναι ο αριθμός των παραδειγμάτων, τόσο περισσότερο ισχυροποιείται η σχέση. Βάση των προηγούμενων ανισοτήτων, το σφάλμα μπορεί να περιοριστεί σε δύο όρια όπως φαίνεται στην ακόλουθη ανισότητα:

$$E(\Sigma\varphi(\theta_N, \theta_N)) \leq \Sigma\varphi(\theta, \theta) \leq E(\Sigma\varphi(\theta_N, \theta))$$

Σύμφωνα με τις παραπάνω ανισότητες, σε περιπτώσεις που ο αριθμός των παραδειγμάτων είναι πεπερασμένος, το πραγματικό σφάλμα δεν μπορεί να υπολογιστεί με ακρίβεια σε πραγματικές εφαρμογές. Οι πιο διαδεδομένες μέθοδοι προσέγγισης των ορίων, μέσα στα οποία βρίσκεται το πραγματικό σφάλμα ενός συστήματος ταξινόμησης είναι η **Μέθοδος – C** και η **Μέθοδος – U**. Πιο αναλυτικά με τη **Μέθοδος – C**, ο υπολογισμός του σφάλματος γίνεται με τη χρήση των διαθέσιμων παραδειγμάτων, τόσο για τη μέτρηση του σφάλματος όσο και για την εκπαίδευση. Το υπολογιζόμενο σφάλμα είναι το μικρότερο δυνατό του συστήματος ταξινόμησης προτύπων. Αντίστοιχα με τη **Μέθοδος – U**, η μέτρηση του σφάλματος για κάθε σύστημα χωριστά γίνεται μέσω των παραδειγμάτων που δεν χρησιμοποιήθηκαν στην εκπαίδευση του συστήματος. Το άνω όριο του σφάλματος ταξινόμησης ισούται με τη μέση τιμή των σφαλμάτων

Για τον υπολογισμό του άνω ορίου του πραγματικού σφάλματος του συστήματος ταξινόμησης, χρησιμοποιούνται δύο μέθοδοι. Η πρώτη είναι η **μέθοδος διαχωρισμού των παραδειγμάτων σε σύνολο εκπαίδευσης και σύνολο ελέγχου** και η δεύτερη η **μέθοδος του αχρησιμοποίητου παραδείγματος**. Στην πρώτη μέθοδο τα παραδείγματα χωρίζονται σε δύο ομάδες. Σε εκείνα που χρησιμοποιούνται για εκπαίδευση και στα εναπομείναντα που χρησιμοποιούνται για την μέτρηση του σφάλματος. Η μέθοδος είναι ασφαλής μόνο όταν υπάρχει διαθέσιμος ένας μεγάλος αριθμός παραδειγμάτων, καθώς σε αντίθετη περίπτωση θα παρουσιάζονται λάθη στον υπολογισμό των παραμέτρων και στην αξιοπιστία του συστήματος. Η δεύτερη μέθοδος προτιμάται στις περιπτώσεις όπου ο διαθέσιμος αριθμός παραδειγμάτων είναι αρκετά μικρός. Το σύστημα αρχικά εκπαιδεύεται από  $N-1$  παραδείγματα, έπειτα το αχρησιμοποίητο παράδειγμα, χρησιμοποιείται για τον υπολογισμό του σφάλματος με αποτέλεσμα να έχουμε δύο τιμές. Η πρώτη τιμή είναι 0% σφάλμα στην περίπτωση της σωστής ταξινόμησης του παραδείγματος και η δεύτερη περίπτωση είναι 100% σφάλμα στην περίπτωση

που το παράδειγμα ταξινομήθηκε λάθος. Η διαδικασία επαναλαμβάνεται  $N$  φορές, αφαιρώντας κάθε φορά ένα διαφορετικό παράδειγμα από το σύνολο των  $N$  διαθέσιμων παραδειγμάτων.

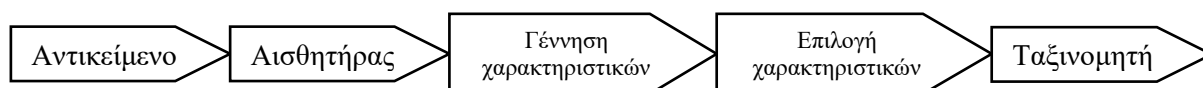
## 2.4 Δομή συστήματος ταξινόμηση προτύπων.

Στην αναγνώριση προτύπων (pattern recognition), το βασικό πρόβλημα είναι η καταχώρηση των άγνωστων αντικειμένων – προτύπων, στην σωστή κατηγορία – κλάση. Αυτό το πρόβλημα ονομάζεται επίσης και ταξινόμηση. Η αναγνώριση προτύπων βρίσκει εφαρμογή σε αρκετά επιστημονικά πεδία. Μερικά από αυτά είναι η μηχανική όραση, η αναγνώριση χαρακτήρων, η ιατρική διάγνωση υποβοηθούμενη από Η/Υ, η αναγνώριση ομιλίας και προσώπου, η βιομετρία, η ανάσυρση εικόνων από βάσεις δεδομένων, εξόρυξη δεδομένων, η Βιοπληροφορική κ.α.. Επίσης, η αναγνώριση προτύπων μπορεί να γίνει **με επίβλεψη** και **χωρίς επίβλεψη**. Όταν έχουμε αναγνώριση προτύπων με επίβλεψη, ο αριθμός των κλάσεων είναι γνωστός και έχουμε διαθέσιμα αντικείμενα για τα οποία είναι γνωστή η κλάση στην οποία ανήκουν. Στην περίπτωση της αναγνώρισης προτύπων χωρίς επίβλεψη ο αριθμός των κλάσεων είναι άγνωστος και για τα διαθέσιμα αντικείμενα που έχουμε δεν είναι γνωστή οποιαδήποτε πληροφορία σχετικά με την κλάση. Έτσι για παράδειγμα στην περίπτωση της κυτταρολογίας ορίζουμε ως αντικείμενα τους πυρήνες των κυττάρων και ως κατηγορίες καλοήθης και κακοήθης. Στην περίπτωση της παραγωγικής βιομηχανίας ως αντικείμενο τη βίδα και ως κατηγορίες ελαττωματική και μη ελαττωματική.

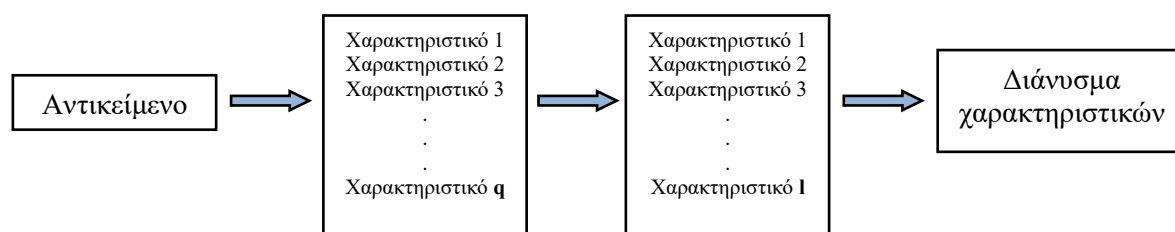
Ως χαρακτηριστικά (Features) ορίζουμε τις μετρήσιμες ποσότητες που λαμβάνονται από τα προς ταξινόμηση αντικείμενα. Η διαδικασία ταξινόμησης βασίζεται στις αντίστοιχες τιμές τους. Διανύσματα χαρακτηριστικών (Features vectors) ορίζεται ένας αριθμός τιμών  $x_1, \dots, x_l$  που συνιστούν το διάνυσμα  $\underline{x} = [x_1, \dots, x_l]^T \in \mathbf{R}^l$ . Τα διανύσματα χαρακτηριστικών θεωρούνται και ως τυχαίες μεταβλητές. Στην αναπαράσταση αντικειμένων σε σύστημα αναγνώρισης προτύπων ισχύει η ακόλουθη ισότητα:

$$\begin{aligned} \text{Αντικείμενο} &\Leftrightarrow \text{Διάνυσμα χαρακτηριστικών μετρήσεων (feature vectors)} \\ \text{Αντικείμενο} &\Leftrightarrow \text{Σημείο στο χώρο} \end{aligned}$$

Στην τυπική δομή ενός συστήματος ταξινόμησης προτύπων, ισχύει το ακόλουθο σχεδιάγραμμα:



Πως δουλεύει όμως το παραπάνω σχήμα;

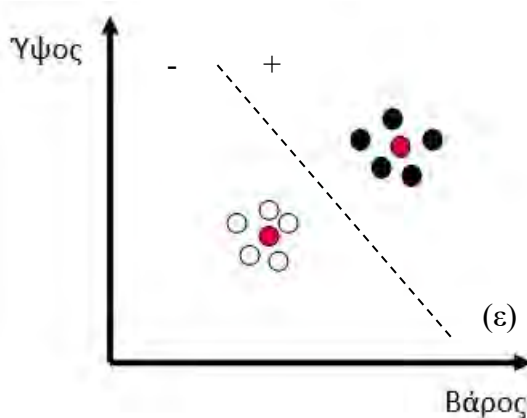


Ως ταξινομητής (classifier) ορίζεται ένα σύνολο συναρτήσεων των οποίων οι τιμές που είναι υπολογισμένες στο  $\chi$ , καθορίζουν την κλάση στην οποία θα καταχωρηθεί το αντίστοιχο πρότυπο. Η ύπαρξη ενός αντιπροσωπευτικού δείγματος προτύπων από κάθε κατηγορία είναι απαραίτητη. Έτσι όσο περισσότερα δείγματα είναι διαθέσιμα, τόσο πιο αντιπροσωπευτική εικόνα έχουμε για τις κλάσεις. Προβλήματα που αντιμετωπίζουμε στην ταξινόμηση είναι **a.** ποια χαρακτηριστικά θα επιλέξω και **b.** πως ο ταξινομητής θα διαχωρίσει τις κλάσεις. Ας δούμε το ακόλουθο παράδειγμα. Έστω, ότι θέλουμε να δημιουργήσουμε ένα σύστημα ταξινόμησης προτύπων που θα διαχωρίζει τους ποδοσφαιριστές **A** και τους χορευτές **B**. Ως αντικείμενα ορίζονται οι **αθλητές** (ποδοσφαιριστές και χορευτές) και ως **κατηγορίες** οι ποδοσφαιριστές (**A**) και οι χορευτές (**B**). Η ανάλυση θα ξεκινήσει από τα χαρακτηριστικά ύψος και βάρος.

	Χορευτές (B)	Καλαθοσφαιριστές (A)
Βάρος	Μικρό	Μεγάλο
Ύψος	Μικρό	Μεγάλο

Πίνακας 1

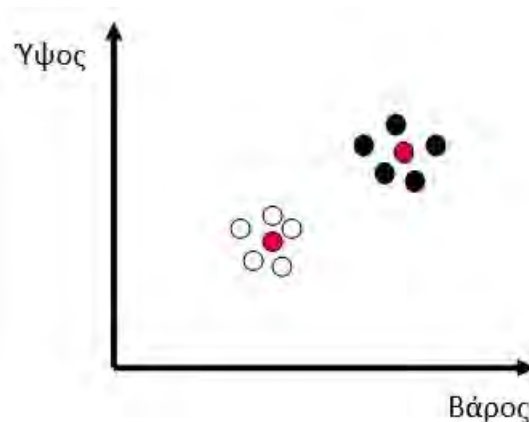
Γεννιέται το εξής ερώτημα. Πως καθορίζονται ποσοτικά οι έννοιες του «μεγάλου» και του «μικρού»; Ο καθορισμός πραγματοποιείται βάση ενός σωστού και αντιπροσωπευτικού δείγματος που λαμβάνεται και από τις κλάσεις. Όσο πιο πλούσιο είναι το δείγμα, τόσο πιο έγκυροι και σωστοί είναι οι παραπάνω χαρακτηρισμοί. Παρακάτω, έχουμε ένα σύνολο συναρτήσεων και κανόνων βάση των οποίων γίνεται η ταξινόμηση.



- Χορευτές (B)
- Καλαθοσφαιριστές (A)

Σύμφωνα με ένα χαρακτηριστικό διάνυσμα που δίνεται, ο ταξινομητής ξεχωρίζει εάν αυτό βρίσκεται στη θετική πλευρά ή στην αρνητική πλευρά της  $(\epsilon)$  και το κατατάσσει αντιστοίχως στις κατηγορίες (A) και (B). Σύμφωνα με το δοθέν χαρακτηριστικό διάνυσμα, ο ταξινομητής εντοπίζει το πλησιέστερο μέσο διάνυσμα και το κατατάσσει στη κατάλληλη κατηγορία. Υπάρχουν διάφοροι συνδυασμοί χαρακτηριστικών όπως απόχρωση ματιών – απόχρωση μαλλιών, απόχρωση μαλλιών – ύψος ή σκέτο ύψος. Ακόμα μπορεί να υπάρχουν και απλά χαρακτηριστικά όπως το ύψος. Η απόδοση του συστήματος εξαρτάται σε μεγάλο βαθμό από την ορθή επιλογή των χαρακτηριστικών. Άρα είναι τεράστιας σημασίας η χρήση των

χαρακτηριστικών που ξεχωρίζουν καλά τις κατηγορίες. Ως αποτέλεσμα των παραπάνω θα πρέπει να έχουμε **a**. Οι μέσες τιμές των χαρακτηριστικών θα πρέπει να διαφέρουν σημαντικά μεταξύ τους από κατηγορία σε κατηγορία και **b**. οι διασπορές γύρω από τις παραπάνω μέσες τιμές να είναι μικρές.



Στη συνέχεια, ο ταξινομητής, σύμφωνα με τα χαρακτηριστικά του δοθέντος διανύσματος, προσδιορίζει το πλησιέστερο μέσω διάστημα και το ταξινομεί στην αντιστοιχεί κατηγορία.

## 2.5 Σύστημα Dashcam

Η επικίνδυνη οδήγηση προκαλεί τραυματισμούς και θανάτους που μπορούν να προληφθούν. Για την προώθηση πρακτικών ασφαλούς οδήγησης, ένα σύστημα dashcam στο μπροστινό και πίσω παράθυρο του αυτοκινήτου, μπορεί να παρέχει οπτικό πλαίσιο περιβάλλοντος τέτοιο ώστε, όταν ένας τρίτος οδηγός παρουσιάζει επικίνδυνες οδηγικές συμπεριφορές να προειδοποιεί έγκαιρα το οδηγό του εν λόγω οχήματος. Ένα τέτοιο σύστημα θα μπορούσε να μας δώσει χρήσιμες πληροφορίες αναφορικά με τη θέση του οχήματος στη λωρίδα, την ακόλουθη απόσταση από το επόμενο όχημα καθώς επίσης και πληροφορίες αναφορικά με την θέση και την ταχύτητα άλλων οχημάτων σε πραγματικές συντεταγμένες. Το σύστημα εκτελεί πολλαπλές εργασίες υπολογιστικής όρασης, συμπεριλαμβανομένης της ανίχνευσης και παρακολούθησης 2D και 3D, της ανίχνευσης λωρίδας και της βαθμονόμησης κάμερας. Στη συνέχεια, αφού προηγηθεί η επεξεργασία των δεδομένων, οι πληροφορίες που εξάγονται χρησιμοποιούνται για την έγκαιρη προειδοποίηση και αποφυγή επικίνδυνων συμβάντων. Ως έγκαιρη προειδοποίηση επικίνδυνων συμβάντων μπορούν θεωρηθούν η απότομη αλλαγή λωρίδας ενός οχήματος, η απότομη πέδηση του προπορευόμενου οχήματος ή μοτοσυκλέτας ή ακόμη ένας πεζός που μπορεί να διασχίσει το δρόμο απρόσεκτα. Ένα σύστημα dashcam, μπορεί να λάβει πληροφορίες από τα οχήματα που προπορεύονται αναλύοντας την ταχύτητά τους και την κατεύθυνσή τους αλλά και άλλες πληροφορίες όπως η ροή της κυκλοφορίας.

### 2.5.1 Τι είναι το σύστημα Dashcam;

Το Dashcam ή αλλιώς κάμερες ταμπλό, είναι ειδικές συσκευές με ενσωματωμένη κάμερα που τοποθετούνται εσωτερικά του αυτοκινήτου και ως στόχο έχουν την καταγραφή των γεγονότων κατά την πορεία του οχήματος. Η χρήση των Dashcams είναι αρκετά διαδεδομένη τα τελευταία χρόνια. Η συμβολή της χρήσης του όμως είναι πολύ σημαντική, καθώς είναι το μόνο μη διαμφισβητούμενο μέσο απόδειξης που έχει ο οδηγός και όλοι οι εμπλεκόμενοι σε ένα ατύχημα. Το εν λόγω σύστημα είναι μία ηλεκτρονική συσκευή, η οποία αποτελείται από μία

κάμερα υψηλής ανάλυσης 720p HD ή 1080p FHD με υποδοχή κάρτας μνήμης MicroSD για την αποθήκευση του βίντεο σε μορφή αρχείου MP4 διαθέτοντας λογισμικό διαγραφής του παλαιού βίντεο. Επίσης, κάποιες συσκευές Dashcam διαθέτουν και άλλα επιπρόσθετα χαρακτηριστικά όπως αισθητήρες G, μικρόφωνο ηχογράφησης ή/και λειτουργία στάθμευσης. Το κυρίως σώμα της συσκευής διαθέτει ειδικά σχεδιασμένη βάση, ώστε να τοποθετείται είτε στο ταμπλό του αυτοκινήτου, είτε στο παρμπρίζ, είτε στον καθρέπτη.

Αρκετά Dashcam έχουν τη δυνατότητα συνδεσιμότητας με smartphone μέσω Bluetooth, wifi ή 4G για ζωντανή προβολή εικόνας και ήχου. Η τεχνολογία τους έχει προχωρήσει αρκετά, καθώς πολλά μοντέλα μπορούν να πραγματοποιήσουν και νυχτερινή λήψη. Τέλος, τα τελευταία και πιο εξελιγμένα μοντέλα διαθέτουν την «έξυπνη» τεχνολογία ADAS (Advanced Drive-Assistance System), παρακολουθώντας τις λωρίδες κυκλοφορίας και προπορευόμενα οχήματα σε περίπτωση που κάποιο αποκλίνει της πορείας του, ειδοποιώντας τον οδηγό με ηχητικά μηνύματα.



Εικόνα 1

Το Dashcam είναι μία ηλεκτρονική συσκευή καταγραφής εικόνας και ήχου τοποθετούμενη στο παρ-μπρίζ του αυτοκινήτου, το οποίο αποτελείται από ένα πολύπλοκο σύστημα αισθητήρων και καμερών. Ως κεντρικό στοιχείο θεωρείται η κεντρική πλακέτα ή αλλιώς ο «εγκέφαλος» της συσκευής. Έπειτα, αναλύοντας τα περιφερειακά μέρη της συσκευής, αναγνωρίζουμε ως το πιο βασικό την κάμερα ή τις κάμερες σε περίπτωση multiple cameras, συνεχίζουμε με τους επιμέρους αισθητήρες (π.χ. GPS, G-Sensors, WiFi, Bluetooth, Parking sensors κ.α.), την μνήμη αποθήκευσης των καταγεγραμμένων πληροφοριών εικόνας και ήχου (Micro SD), το σύστημα ενέργειας και φόρτισης αυτοκινήτου με τις διάφορες καλωδιώσεις και τέλος τα εξαρτήματα τοποθέτησης.

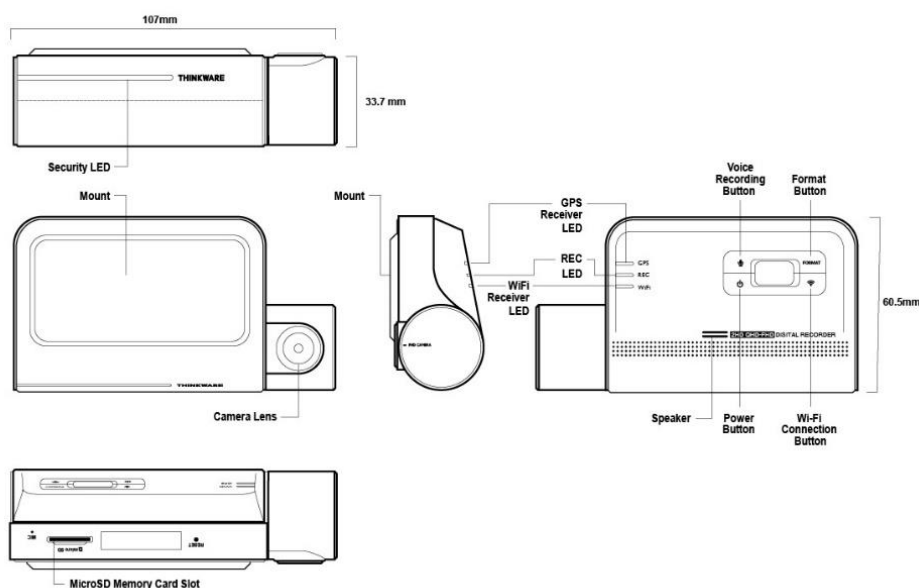
Ένα αρκετά σημαντικό μέρος στη δομή του Dashcam αποτελούν, οι έξτρα και ιδιαίτερες λειτουργίες που μπορεί να έχουν κάποιες συσκευές. Μία από αυτές είναι το σύστημα ADAS (Advanced Driver – Assistance Systems) όπου θα δούμε στο κεφάλαιο 2.5.3, το οποίο μέσω της κάμερας εντοπίζει και αντιλαμβάνεται εμπόδια, όπως τα προπορευόμενα αυτοκίνητα όταν σταματούν απότομα ή ξεκινούν από στάση, καθώς και την αλλαγή λωρίδας. Επίσης, έχει τη δυνατότητα να αναγνωρίζει τους ερυθρούς σηματοδότες και τις πινακίδες οδικής κυκλοφορίας ενημερώνοντας τον οδηγό έγκυρα και έγκαιρα.



Εικόνα 2



Εικόνα 3



Εικόνα 4

Στην εικόνα 2, βλέπουμε τι περιέχει το πακέτο αγοράς του συστήματος Dashcam Thinkware Q800 PRO και στην εικόνα 3 βλέπουμε τα επιπρόσθετα εξαρτήματα που μπορεί να προσθέσει ο κάτοχός του στο σύστημα αν το επιθυμεί. Στην εικόνα 4, βλέπουμε τη σχηματική αναπαράσταση του συστήματος Dashcam Thinkware Q800 PRO με διαστάσεις και όλα τα επιμέρους εξωτερικά χαρακτηριστικά.



## 2.5.2 Αλγόριθμοι λειτουργίας συστημάτων Dashcam

Οι αλγόριθμοι αποτελούν ένα μεγάλο και βασικό στοιχείο της επιστήμης της πληροφορικής και της τεχνολογίας. Βρίσκουν εφαρμογή σε πεδία όπως Hardware, Software, Γλώσσες προγραμματισμού, Ανάλυση δεδομένων κ.α.. Τι είναι όμως ο αλγόριθμος; Αλγόριθμος ονομάζεται μία πεπερασμένη σειρά ενεργειών με αρχή και τέλος, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, όπου σαν στόχο έχουν την επίλυση ενός συγκεκριμένου προβλήματος.

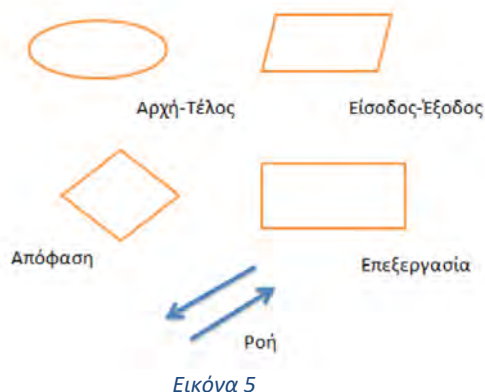
Τι χαρακτηριστικά πρέπει να έχει ένας αλγόριθμος:

- ❖ Να είναι σαφής (οι οδηγίες δεν είναι διφορούμενες ή ασαφείς).
- ❖ Να είναι πεπερασμένος (τερματίζεται σε πεπερασμένο αριθμό βημάτων).
- ❖ Να έχει ένα σαφώς ορισμένο σημείο εκκίνησης και ένα ή περισσότερα σημεία τερματισμού.
- ❖ Να είναι αποδοτικός (φθάνει στο αποτέλεσμα με τον καλύτερο δυνατό τρόπο).
- ❖ Να είναι γενικός (εφαρμόζεται σε όσο το δυνατόν περισσότερες παρόμοιες περιπτώσεις).

Πολλές φορές, όταν έχουμε απλά προβλήματα να λύσουμε και χρειαζόμαστε έναν αλγόριθμο είναι εύκολο να τον βρούμε ολόκληρο ή τμήμα του, μέσα από τη βιβλιογραφία. Άλλες φορές πάλι, όταν έχουμε σύνθετα προβλήματα να λύσουμε, θα πρέπει ο αλγόριθμος να αναπτυχθεί από τον προγραμματιστή. Όταν θέλουμε να αναπτύξουμε έναν αλγόριθμο, είναι πολύ σημαντικό να δώσουμε βάση στα παρακάτω ερωτήματα:

- ❖ Τι θα κάνει το πρόγραμμα όταν του δοθούν λανθασμένα στοιχεία για επεξεργασία;
- ❖ Πως και πότε θα σταματήσει η εκτέλεση του προγράμματος;
- ❖ Πως το πρόγραμμα θα επεξεργαστεί τα στοιχεία, ώστε να βρει τη λύση του προβλήματος;

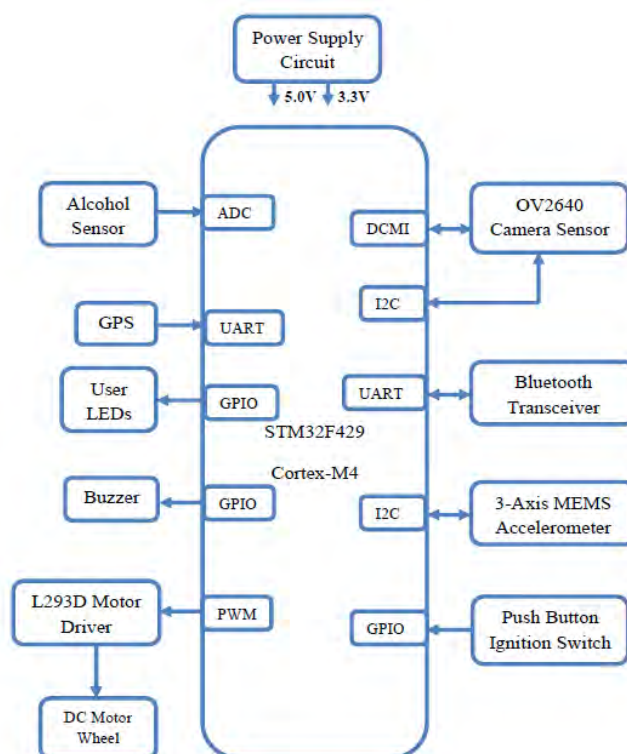
Η ανάπτυξη ενός αλγορίθμου βασίζεται στην ανάλυση του προβλήματος. Η λύση του αρχικού προβλήματος προκύπτει, εκτελώντας με τη σωστή σειρά τις εργασίες που αντιστοιχούν στις μονάδες του δομικού χάρτη του προβλήματος. Υπάρχουν διάφορες τεχνικές γραφικής ή διαγραμματικής αναπαράστασης των αλγορίθμων. Η πιο παλιά και πιο γνωστή είναι το διάγραμμα ροής (flow chart). Οι τεχνικές αυτές περιλαμβάνουν τη χρήση ειδικών γεωμετρικών σχημάτων ή συμβόλων, που το κάθε ένα δηλώνει μία συγκεκριμένη απλή ενέργεια ή λειτουργία του αλγορίθμου. Τα σχήματα, ενώνονται μεταξύ τους με βέλη που δηλώνουν τη σειρά εκτέλεσης των ενεργειών. Στην εικόνα 5, βλέπουμε τα στοιχεία ενός αλγορίθμου.



Εικόνα 5



Στην εικόνα 6, μπορούμε να δούμε την διαγραμματική αναπαράσταση του αλγόριθμου λειτουργίας ενός συστήματος Dashcam για δικυκλιστές. Το σύστημα περιλαμβάνει κάμερα, GPS, αισθητήρα μέτρησης επιτάχυνσης, αισθητήρα μέτρησης αλκοόλ, Bluetooth, εισόδους ενέργειας και διακόπτες. Ο αλγόριθμος αφορά την κατασκευή ενός συστήματος Dashcam με συγκεκριμένη λειτουργία. Το σύστημα μέσω μίας κάμερας, αναγνωρίζει εάν ο αναβάτης φοράει ή όχι το κράνος του και αντιστοίχως επιτρέπει ή όχι την έναυση του κινητήρα της μοτοσυκλέτας. Το σύστημα, συνεχίζει να ελέγχει τον αναβάτη εάν φοράει κράνος καθ' όλη τη διάρκεια της πορείας και σε περίπτωση που αφαιρέσει το κράνος, θα στείλει σήμα στις τοπικές αρχές ελέγχου κυκλοφορίας. Παράλληλα, ένας αισθητήρας μέτρησης αλκοόλ, ανιχνεύει εάν ο αναβάτης έχει πιεί και τον ενημερώνει. Ένα GPS και ένας αισθητήρας θέσης, βοηθούν το σύστημα να αναγνωρίσει εάν έχει συμβεί ατύχημα στον αναβάτη και στέλνουν μήνυμα sms στις τοπικές αρχές ελέγχου κυκλοφορίας και στα μέλη της οικογένειας. Τέλος, μια συσκευή Bluetooth η οποία συνδέεται με το κινητό μέσω εφαρμογής Android, παρέχει στο σύστημα τη δυνατότητα αποστολής sms.

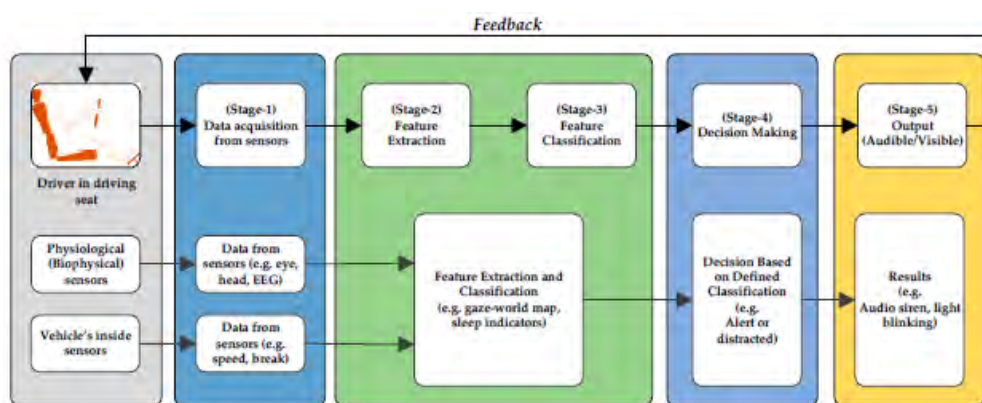


Εικόνα 6

### 2.5.3 Σύστημα ADAS (Advanced Drive – Assistance Systems).

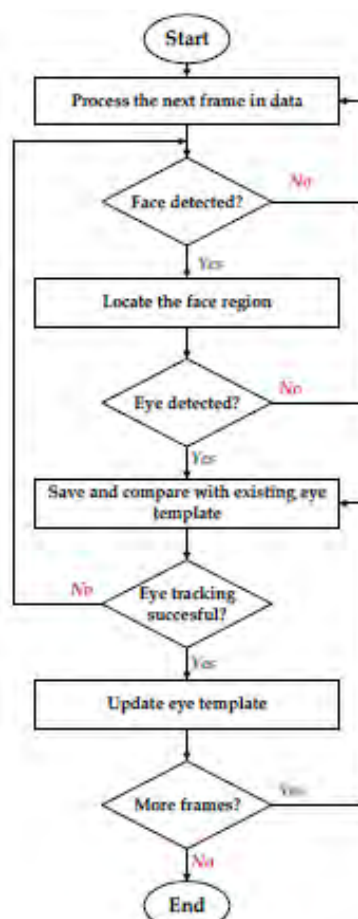
Στην προηγούμενη ενότητα είδαμε και αναλύσαμε τους αλγόριθμους και τον αλγόριθμο λειτουργίας ενός συστήματος Dashcam με σύνθετες λειτουργίες. Στην επεξεργασία και ανάλυση εικόνας, οι αλγόριθμοι διαδραματίζουν ίσως τον σημαντικότερο ρόλο. Αυτή η άποψη ενισχύεται, όταν καλούμαστε να σχεδιάσουμε έναν αλγόριθμο που θα αναγνωρίζει διάφορα στοιχεία σε μία εικόνα όπως εμπόδια, διαχωριστικές γραμμές, πινακίδες σήμανσης οδικής κυκλοφορίας κ.α.. Ένα τέτοιο σύστημα είναι το σύστημα ADAS (Advanced Drive – Assistance Systems), το οποίο αναγνωρίζει ανθρώπους, εμπόδια, φωτεινούς σηματοδότες και πινακίδες σήμανσης, ελέγχει την ταχύτητα και την πορεία των προπορευόμενων αυτοκινήτων, καθώς και την αλλαγή λωρίδας από άλλα αυτοκίνητα. Επίσης, ελέγχοντας την πορεία των προπορευόμενων αυτοκινήτων μέσω των

καμερών, μπορεί να ειδοποιήσει τον οδηγό για τυχόν σύγκρουση. Στην εικόνα 7 βλέπουμε τα στάδια ενός τυπικού αλγορίθμου οπτικών δεδομένων του συστήματος ADAS. Αυτό που διακρίνουμε είναι πως έχουμε έναν αρκετά σύνθετο αλγόριθμο.



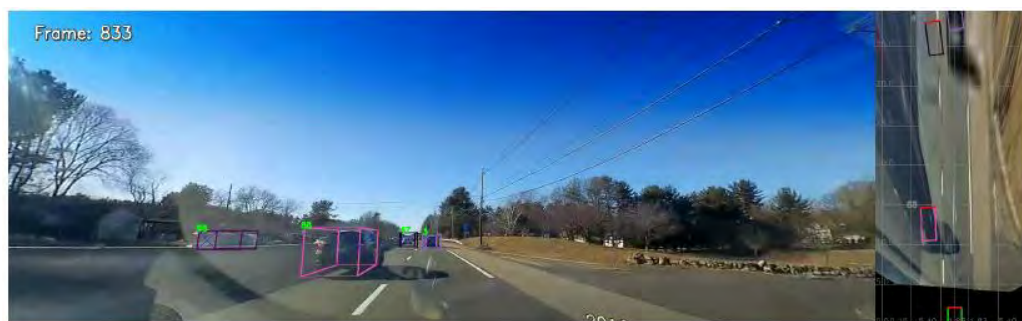
Εικόνα 7

Υπάρχουν ακόμη και πιο εξελιγμένα συστήματα ADAS, τα οποία μέσω ειδικών καμερών, μπορούν να παρακολουθούν και να ελέγχουν το βλέμμα και τα μάτια του οδηγού. Στην εικόνα 8, βλέπουμε το διάγραμμα ροής ενός αλγορίθμου παρακολούθησης ματιών του εν λόγω συστήματος.



Εικόνα 8

Στις εικόνες 9, 10 και 11, βλέπουμε πως το σύστημα ADAS, αναγνωρίζει αντικείμενα, οχήματα και φωτεινούς σηματοδότες.



Εικόνα 9

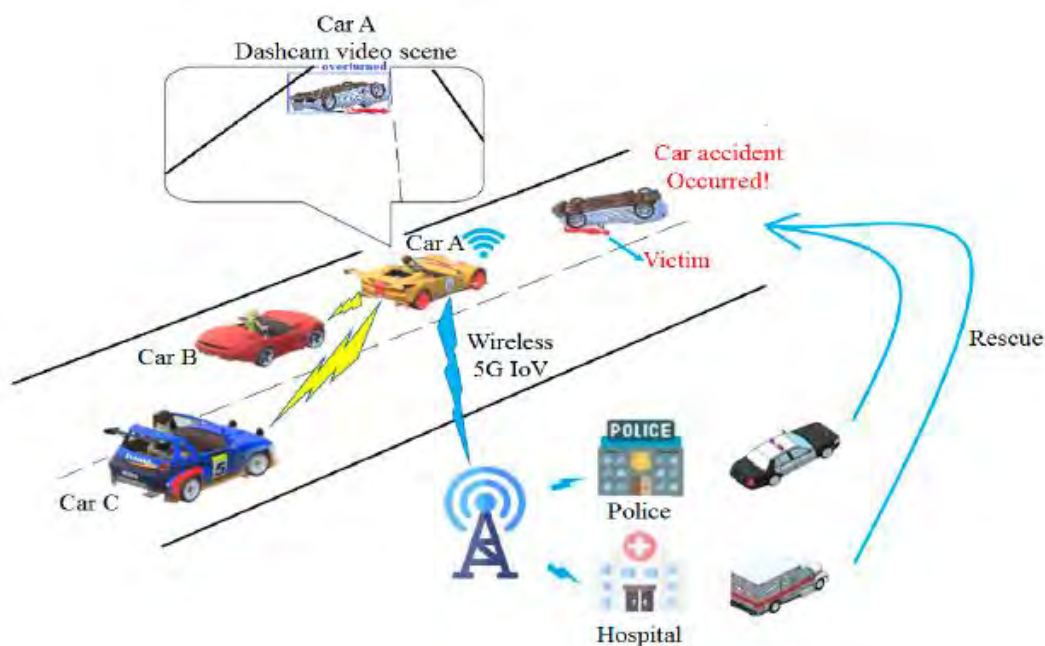


Εικόνα 10



Εικόνα 11

Αν αναλογιστούμε τον τρόπο λειτουργίας του συστήματος ADAS, συμπεραίνουμε πολύ εύκολα πως βασίζεται στην επεξεργασία και ανάλυση εικόνας. Το πιο βασικό όμως, είναι η αναγνώριση αντικειμένων σε μία εικόνα. Για να μπορέσει ένα σύστημα να αναγνωρίζει και να εντοπίζει αντικείμενα μέσα από μία εικόνα, θα πρέπει να υπάρχει ένας αλγόριθμος τεχνητής νοημοσύνης (AI – Artificial Intelligence algorithm). Η τεχνητή νοημοσύνη είναι ένας επιστημονικός όρος που επιτρέπει στους υπολογιστές να μιμούνται την ανθρώπινη συμπεριφορά. Το σύνολο των αλγορίθμων που έχουν εκπαιδευτεί σε δεδομένα που καθιστούν όλα αυτά δυνατά αναφέρονται στη μηχανική μάθηση. Η Μηχανική Μάθηση (Machine Learning) είναι υποσύνολο της Τεχνητής Νοημοσύνης και η βαθιά μάθηση (Deep Learning) με τη σειρά της, είναι ένα υποσύνολο της Μηχανικής μάθησης. Στην εικόνα 12 βλέπουμε το σχεδιάγραμμα ενός συστήματος Dashcam, όπου μπορεί να αναγνωρίζει ένα τροχαίο ατύχημα και παράλληλα να ενημερώσει μέσω ενός δικτύου 5G τα διερχόμενα αυτοκίνητα, τις τοπικές αρχές ελέγχου κυκλοφορίας καθώς και το πλησιέστερο νοσοκομείο.



Εικόνα 12

Φυσικά, για να υλοποιηθούν αυτές οι εργασίες στο σύστημα, γίνεται χρήση αλγορίθμων τεχνητής νοημοσύνης και μηχανικής μάθησης, ώστε να μπορεί το σύστημα να αναγνωρίζει έγκαιρα και έγκυρα το συμβάν και να ειδοποιεί άμεσα τα διερχόμενα αυτοκίνητα και τις τοπικές αρχές.

Τέλος, στο εμπόριο υπάρχουν ισχυρές πλατφόρμες – μίνι υπολογιστές που είναι κατάλληλοι για λειτουργίες AI (Artificial Intelligence). Ένα από αυτά είναι το Jetson nano της Nvidia – εικόνα 13. Έχει τη δυνατότητα να τρέξει παράλληλα πολλαπλά νευρωνικά δίκτυα για λειτουργίες όπως ταξινόμηση εικόνων, αναγνώριση αντικειμένων, τμηματοποίηση και επεξεργασία ομιλίας.



Εικόνα 13



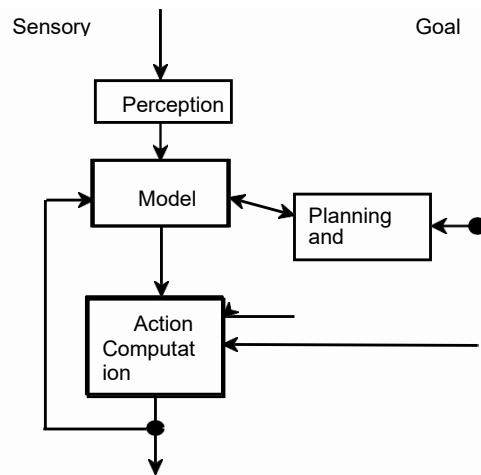
## 3 Μηχανική μάθηση

### 3.1 Εισαγωγή

Υπάρχει μεγάλη παρανόηση σχετικά με το τι είναι η μηχανική μάθηση, που τροφοδοτείται από την πρόσφατη επιτυχία και μερικές φορές την εντυπωσιακή κάλυψη από τα μέσα ενημέρωσης. Σε γενικές γραμμές, η μηχανική μάθηση είναι η εφαρμογή στατιστικών, μαθηματικών και αριθμητικών τεχνικών για την εξαγωγή κάποιας μορφής γνώσης από δεδομένα. Αυτή η «γνώση» μπορεί να μας προσφέρει κάποιο είδος σύνοψης, οπτικοποίησης, ομαδοποίησης ή ακόμα και προγνωστικής δύναμης πάνω σε σύνολα δεδομένων. Με όλα αυτά, είναι σημαντικό να τονίσουμε τους περιορισμούς της μηχανικής μάθησης. Δεν είναι και ούτε θα αντικαταστήσει ποτέ την κριτική σκέψη και τη μεθοδική, διαδικαστική εργασία στην επιστήμη των δεδομένων. Πράγματι, η μηχανική μάθηση μπορεί εύλογα να χαρακτηριστεί μια χαλαρή συλλογή επιστημονικών κλάδων και εργαλείων.

Η μάθηση, όπως και η νοημοσύνη, καλύπτει ένα τόσο ευρύ φάσμα διαδικασιών που είναι δύσκολο να οριστεί με ακρίβεια. Ένας ορισμός λεξικού περιλαμβάνει φράσεις όπως "απόκτηση γνώσης ή κατανόησης ή δεξιότητας μέσω μελέτης, διδασκαλίας ή εμπειρίας" και "τροποποίηση μιας τάσης συμπεριφοράς από την εμπειρία". Οι ζωολόγοι και οι ψυχολόγοι μελετούν τη μάθηση σε ζώα και ανθρώπους. Υπάρχουν αρκετοί παραλληλισμοί μεταξύ της μάθησης των ζώων και της μηχανικής μάθησης. Σίγουρα, πολλές τεχνικές στη μηχανική μάθηση προέρχονται από τις προσπάθειες των ψυχολόγων να κάνουν πιο ακριβείς τις θεωρίες τους για τη μάθηση των ζώων και των ανθρώπων μέσω υπολογιστικών μοντέλων. Φαίνεται επίσης πιθανό ότι οι έννοιες και οι τεχνικές που διερευνώνται από τους ερευνητές στη μηχανική μάθηση, μπορεί να φωτίσουν ορισμένες πτυχές της βιολογικής μάθησης. Όσον αφορά τις μηχανές, θα μπορούσαμε να πούμε σε πολύ γενικές γραμμές, ότι μια μηχανή μαθαίνει κάθε φορά που αλλάζει τη δομή, το πρόγραμμα ή τα δεδομένα της (με βάση τις εισροές της ή ως απάντηση σε εξωτερικές πληροφορίες) με τέτοιο τρόπο ώστε να βελτιώνεται η αναμενόμενη μελλοντική της απόδοση. Ορισμένες από αυτές τις αλλαγές, όπως η προσθήκη μιας εγγραφής σε μια βάση δεδομένων, εμπίπτουν άνετα στην αρμοδιότητα άλλων κλάδων και δεν είναι απαραίτητα καλύτερα κατανοητές για να ονομάζονται μάθηση. Για παράδειγμα όμως, όταν η απόδοση μιας μηχανής αναγνώρισης ομιλίας βελτιώνεται αφού ακούσουμε αρκετά δείγματα της ομιλίας ενός ατόμου, σε αυτή την περίπτωση να πούμε με άνεση ότι το μηχάνημα έχει μάθει. Η μηχανική μάθηση αναφέρεται συνήθως στις αλλαγές των συστημάτων, που εκτελούν εργασίες οι οποίες σχετίζονται με την τεχνητή νοημοσύνη (ΑΙ). Τέτοιες εργασίες περιλαμβάνουν αναγνώριση, διάγνωση, σχεδιασμό, έλεγχο ρομπότ, πρόβλεψη κ.α.. Οι "αλλαγές" μπορεί να είναι είτε βελτιώσεις σε ήδη εκτελούμενα συστήματα, είτε εξαρχής σύνθεση νέων συστημάτων. Για να είμαστε λίγο πιο συγκεκριμένοι, δείχνουμε την αρχιτεκτονική ενός τυπικού συστήματος Τεχνητής Νοημοσύνης (ΑΙ) «πράκτορα» στο σχήμα 1.

Αυτός ο παράγοντας αντιλαμβάνεται και μοντελοποιεί το περιβάλλον του, κάνοντας τις κατάλληλες ενέργειες προβλέποντας τα αποτελέσματά τους. Οι αλλαγές που έγιναν σε οποιοδήποτε από τα στοιχεία που εμφανίζονται στο σχήμα ενδέχεται να υπολογίζονται ως μάθηση. Μπορούν να χρησιμοποιηθούν διαφορετικοί μηχανισμοί μάθησης ανάλογα με το υποσύστημα που αλλάζει.



Σχήμα 1

Κάποιος θα μπορούσε να ρωτήσει: «Γιατί πρέπει να μάθουν οι μηχανές; Γιατί να μην σχεδιάσουμε μηχανές οι οποίες να αποδίδουν όπως θέλουμε εξ αρχής;» Υπάρχουν διάφοροι λόγοι για τους οποίους η μηχανική μάθηση είναι σημαντική. Φυσικά, έχουμε ήδη αναφέρει ότι η επίτευξη της μάθησης σε μηχανές μπορεί να μας βοηθήσει να κατανοήσουμε πώς μαθαίνουν τα ζώα και οι άνθρωποι. Υπάρχουν όμως και σημαντικοί μηχανικοί λόγοι όπως:

- Ορισμένες εργασίες δεν μπορούν να οριστούν καλά παρά μόνο με παραδείγματα. Δηλαδή, μπορεί να είμαστε σε θέση να καθορίσουμε ζεύγη εισόδου / εξόδου, αλλά όχι μια συνοπτική σχέση μεταξύ εισόδων και επιθυμητών εξόδων. Θα θέλαμε τα μηχανήματα να είναι σε θέση να προσαρμόσουν την εσωτερική τους δομή, για να παράγουν σωστές εξόδους για μεγάλο αριθμό δειγματοληπτικών εισόδων και έτσι να περιορίσουν κατάλληλα τη λειτουργία εισόδου / εξόδου τους, για να προσεγγίσουν τη σχέση που υπονοείται στα παραδείγματα.
- Είναι πιθανό ανάμεσα σε μεγάλους σωρούς δεδομένων να είναι κρυμμένες σημαντικές σχέσεις και συσχετίσεις. Οι μέθοδοι μηχανικής μάθησης μπορούν συχνά να χρησιμοποιηθούν για την εξαγωγή αυτών των σχέσεων (εξόρυξη δεδομένων).
- Οι ανθρώπινοι σχεδιασμοί συχνά παράγουν μηχανές που δεν λειτουργούν τόσο καλά όσο το επιθυμούμε στα περιβάλλοντα που χρησιμοποιούνται. Στην πραγματικότητα, ορισμένα χαρακτηριστικά του εργασιακού περιβάλλοντος, μπορεί να μην είναι πλήρως γνωστά κατά το χρόνο σχεδιασμού. Οι μέθοδοι μηχανικής μάθησης μπορούν να χρησιμοποιηθούν για τη βελτίωση των σχεδίων των ήδη υπαρχόντων μηχανών.
- Ο όγκος των διαθέσιμων γνώσεων σχετικά με ορισμένες εργασίες μπορεί να είναι πολύ μεγάλος για ρητή κωδικοποίηση από ανθρώπους. Οι μηχανές που μαθαίνουν αυτή τη γνώση σταδιακά, μπορεί να είναι σε θέση να συλλάβουν περισσότερα από αυτά που οι άνθρωποι θα ήθελαν να γράψουν.
- Τα περιβάλλοντα αλλάζουν με την πάροδο του χρόνου. Αυτό έχει ως αποτέλεσμα τα μηχανήματα που μπορούν να προσαρμοστούν σε ένα μεταβαλλόμενο περιβάλλον, να μειώσουν την ανάγκη για συνεχή επανασχεδιασμό.

- Νέες γνώσεις σχετικά με τα καθήκοντα ανακαλύπτονται συνεχώς από τους ανθρώπους. Το λεξιλόγιο αλλάζει. Υπάρχει μια συνεχής ροή νέων γεγονότων στον κόσμο όπου είναι αναγκαία για την εκμάθηση των μηχανών. Ο συνεχής επανασχεδιασμός των συστημάτων τεχνητής νοημοσύνης ώστε να συμμορφώνονται με τις νέες γνώσεις και τα νέα δεδομένα, δεν είναι πρακτικός και σε αυτό το πρόβλημα οι μέθοδοι μηχανικής μάθησης είναι η λύση.

## 3.2 Τεχνητή νοημοσύνη

Τεχνητή νοημοσύνη ονομάζεται ο κλάδος της πληροφορικής ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών προγραμμάτων που μιμούνται των άνθρωπο αποδίδοντας μία στοιχειώδη ευφυΐα όπως **μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, επίλυση προβλημάτων και κατανόηση**. Η τεχνητή νοημοσύνη ξεκίνησε τη δεκαετία του 1950 και είχε καταβολές από τα μαθηματικά, τη φιλοσοφία, τα οικονομικά ακόμα και τη γλωσσολογία. Τα παραπάνω, μας δίνουν τη δυνατότητα να δούμε τους στόχους των ερευνών. Έτσι, η Τεχνητή Νοημοσύνη αναφέρεται σε:

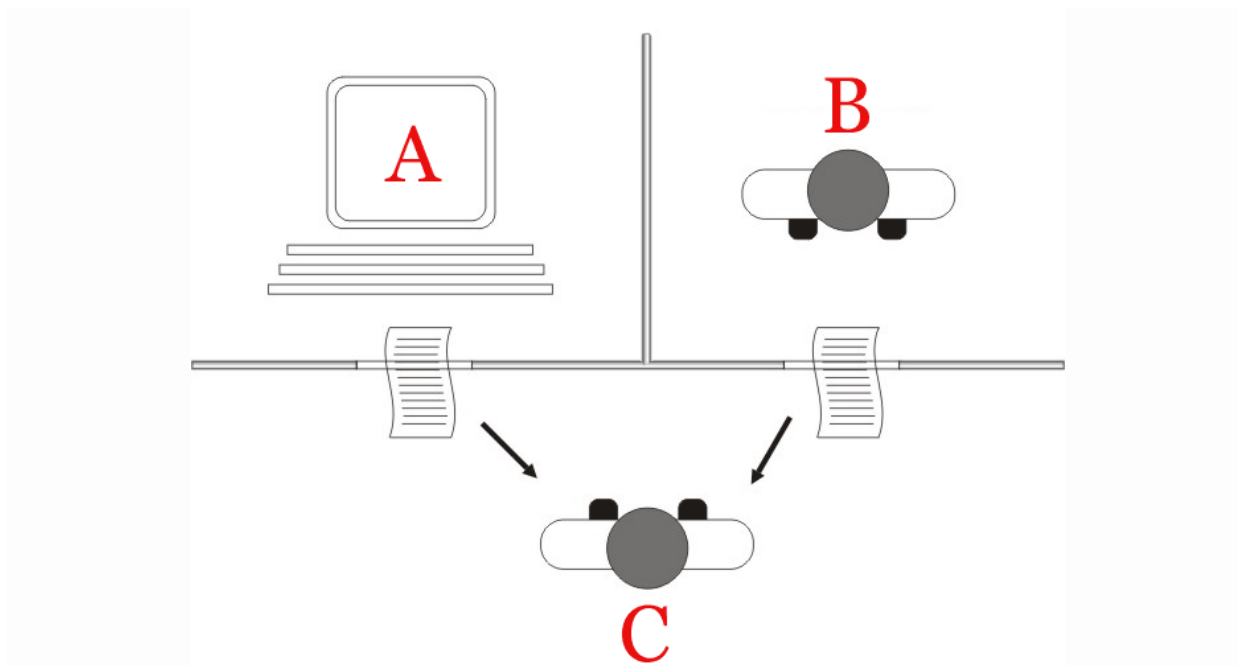
1. Συστήματα που συμπεριφέρονται όπως οι άνθρωποι.
2. Συστήματα που σκέφτονται όπως οι άνθρωποι.
3. Συστήματα που σκέφτονται ορθολογικά.
4. Συστήματα που συμπεριφέρονται ορθά.

Ας αναλύσουμε επιμέρους τα παραπάνω.

### 1. Να συμπεριφέρονται όπως οι άνθρωποι.

Η δοκιμή Turing εφευρέθηκε το 1950 από τον Alan Turing. Ο Alan Turing ήταν Άγγλος μαθηματικός. Θεωρείται πατέρας της επιστήμης των υπολογιστών, χάρη στην συμβολή του στη «θεωρία του υπολογισμού» το 1930. Το 1950 πρότεινε έναν τρόπο ώστε να διαπιστωθεί πειραματικά εάν μία μηχανή έχει γνωστικές δυνατότητες και μπορεί να σκεφτεί. Το παρουσίασε μέσω ενός επιστημονικού άρθρου με τίτλο «Computing Machinery and Intelligence» καθώς τον εργαζόταν στο Πανεπιστήμιο του Μάντσεστερ. Αυτός ο τρόπος είναι γνωστός ως δοκιμή Turing ή «**Turing Test**».

Το test Turing – εικόνα 14 είναι μία διαδικασία κατά την οποία, μία μηχανή δοκιμάζεται ώστε να αποδειχτεί εάν έχει ευφυή συμπεριφορά ισοδύναμη με αυτή του ανθρώπου. Σκεφτείτε, λέει ο Turing, έναν άντρα (Α) και μια γυναίκα (Β) σε ένα δωμάτιο που επικοινωνούν μέσω «τηλετύπου» με έναν άνθρωπο (C) που κάθεται σε ένα άλλο δωμάτιο. Στόχος του παιχνιδιού είναι ο άνθρωπος αυτός (Γ), αδιευκρίνιστου φύλου, με ερωτήματα που θέτει με τον τηλετύπο στους (Α) και (Β), να μαντέψει από τις απαντήσεις τους ποιος είναι ο άντρας και ποια είναι η γυναίκα. Η γυναίκα (Β) του απαντά προσπαθώντας να τον βοηθήσει και να τον πείσει ότι όντως είναι γυναίκα, ενώ ο άντρας (Α) προσπαθεί να τον παραπλανήσει ότι αυτός είναι πραγματικά η γυναίκα και όχι η (Β). Και οι δύο δηλαδή παίκτες (Α) και (Β) προσπαθούν να τον πείσουν ότι είναι γυναίκες και ο (C) πρέπει να καταλάβει ποιος λέει την αλήθεια και ποιος ψεύδεται. Τα ερωτήματα που γεννήθηκαν από τον Turing ήταν, τι θα γινόταν εάν σε αυτό το παιχνίδι στην θέση (Α), αντικαταστήσουμε τον άντρα με μια μηχανή; Ο (C) που προσπαθεί να μαντέψει ποιος είναι ποιος, θα έχει το ίδιο ποσοστό αποτυχίας σε αυτή την περίπτωση, όσο είχε και όταν το παιχνίδι παιζόταν με τον άντρα και την γυναίκα;



Εικόνα 14

Η ευφυΐα και οι ικανότητες ενός συστήματος δεν σταματάει όμως εκεί. Στη συνέχεια βλέπουμε μια σειρά από ικανότητες – υποτομείς ΤΝ που θα πρέπει να έχει ένα σύστημα τεχνητής νοημοσύνης.

- ❖ Παράσταση γνώσεων και εξαγωγή συμπερασμάτων.
- ❖ Μηχανική μάθηση.
- ❖ Επεξεργασία φυσικής γλώσσας.
- ❖ Υπολογιστική όραση.
- ❖ Σχεδιασμός ενεργειών.
- ❖ Ρομποτική.

Παράλληλα όμως παρουσιάζονται και κοινά προβλήματα από τους υποτομείς, όπως η αναζήτηση σε μεγάλους χώρους καταστάσεων, ο χειρισμός ασαφών πληροφοριών και η δυσκολία διατύπωσης αλγοριθμικής λύσης.

## 2. Να σκέφτονται ορθολογικά.

Σε αυτή την κατηγορία, η προσέγγιση των συστημάτων σε είναι βασισμένη στη λογική. Η ευφυής συμπεριφορά ενός συστήματος είναι αποτέλεσμα των ορθών συλλογισμών του. Έτσι ένα σύστημα τεχνητής νοημοσύνης θα πρέπει να κάνει χρήση κανόνων ορθής συλλογιστικής, τυπικών γλωσσών παράσταση γνώσης αλλά και ισχυρισμών. Στις μεθόδους συλλογισμού με κανόνες παρατηρούμε δύο κατηγορίες. Στην πρώτη κατηγορία έχουμε τον ορθό συλλογισμό (forward reasoning) - οδηγούμενο από τα δεδομένα (data driven reasoning) και τον ανάστροφο συλλογισμό (backward reasoning) – οδηγούμενο από τον στόχο (goal driven reasoning). Στην δεύτερη κατηγορία έχουμε τον συνεπαγωγικό συλλογισμό (deductive reasoning) και τον απαγωγικό συλλογισμό (abductive reasoning). Η πρώτη κατηγορία αφορά την οργάνωση της γνώσης στους κανόνες (επίπεδο σχεδίασης) και η δεύτερη κατηγορία αφορά τον βαθμό βεβαιότητας του συλλογισμού (επίπεδο σημασιολογίας). Οι δύο αυτές κατηγορίες έχουν άμεση σχέση μεταξύ τους. Αυτό φαίνεται στα ακόλουθα παραδείγματα.



- ❖ Συνεπαγωγικός – οδηγούμενος από τον στόχο (ανάστροφος συλλογισμός).  
Εάν έχω γρίπη τότε έχω πυρετό και πονοκέφαλο. {Βέβαιος συλλογισμός}
- ❖ Απαγωγικός – οδηγούμενος από τα δεδομένα (ορθός συλλογισμός)  
Εάν έχω πυρετό και πονοκέφαλο τότε έχω γρίπη. {Αβέβαιος συλλογισμός}

### 3. Να σκέφτονται σαν άνθρωποι.

Τα ευφυή συστήματα τεχνητής νοημοσύνης πρέπει να σκέφτονται σαν άνθρωποι. Για να επιτευχθεί αυτό θα πρέπει να γίνει εκτενής μελέτη της γνωσιακής επιστήμης με χρήση υπολογιστικών μοντέλων. Αυτό περιλαμβάνει τους μηχανισμούς ανθρώπινης νόησης, οι οποίοι έχουν άμεση σχέση με την ψυχολογία και τη νευροεπιστήμη. Τα συστήματα τεχνητής νοημοσύνης έχουν αρκετές συγγένειες με την ανθρώπινη σκέψη, αλλά στην ουσία πρόκειται για διαφορετικά επιστημονικά πεδία. Στις αρχές της τεχνητής νοημοσύνης η διαφορά δεν ήταν καθαρή, γι' αυτό και είχαν πραγματοποιηθεί αρκετά πειράματα με ανθρώπους. Το 1959 δημιουργήθηκε από τους Herbert A. Simon, J. C. Shaw, και Allen Newell το **General Problem Solver (GPS)**, ένα πρόγραμμα υπολογιστή το οποίο θα λειτουργούσε ως μία καθολική μηχανή επίλυσης προβλημάτων. Σε αντίθεση με το Logic Theory Project, το GPS θα λειτουργούσε με τη μέθοδο ανάλυσης μέσων – σκοπών.

Οποιοδήποτε πρόβλημα που μπορεί να εκφραστεί ως ένα σύνολο καλά διαμορφωμένων τύπων (Well-Form Formulas – WFFs) ή ρητρών Horn και που αποτελούν ένα κατευθυνόμενο γράφημα με μία ή περισσότερες πηγές (δηλαδή αξιώματα) και καταβόθρες (δηλαδή επιθυμητά συμπεράσματα), μπορεί να λυθεί με GPS. Οι αποδείξεις στην κατηγορηματική λογική και οι χώροι προβλημάτων ευκλείδειας γεωμετρίας, είναι πρωταρχικά παραδείγματα του τομέα της εφαρμοσιμότητας του GPS. Βασίστηκε στο θεωρητικό έργο των Simon και Newell για τις λογικές μηχανές. Το GPS ήταν το πρώτο πρόγραμμα υπολογιστή που διαχώριζε τις γνώσεις του για τα προβλήματα (κανόνες που παρουσιάζονται ως δεδομένα εισόδου) από τη στρατηγική του για τον τρόπο επίλυσης προβλημάτων (μια γενική μηχανή επίλυσης). Το GPS υλοποιήθηκε στη γλώσσα προγραμματισμού τρίτης τάξης IPL.

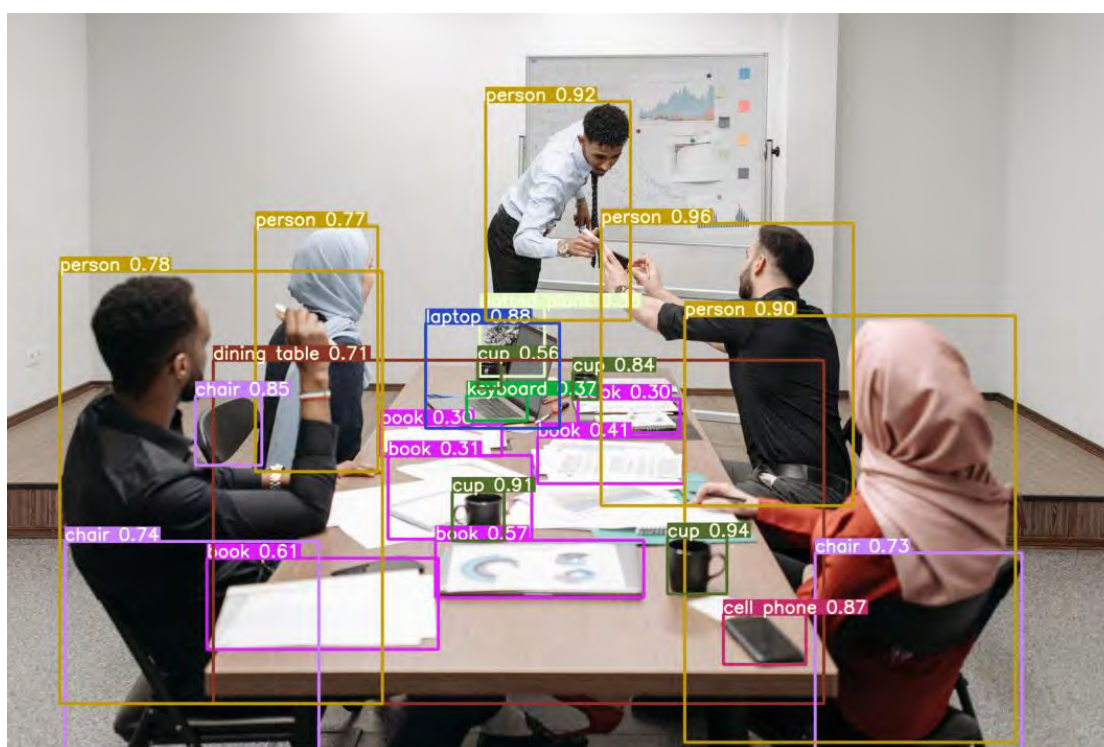
### 4. Να συμπεριφέρονται ορθά.

Σε ένα περίπλοκο περιβάλλον, η ορθή συμπεριφορά ενός συστήματος είναι αυτό που μας οδηγεί στο επιθυμητό αποτέλεσμα. Όταν υπάρχουν συνθήκες αβεβαιότητας, η ορθή συμπεριφορά μας οδηγεί στο καλύτερο αποτέλεσμα με τη μεγαλύτερη πιθανότητα. Αυτό όμως δεν συμβαίνει απαραίτητα μέσω των μηχανισμών ορθής συλλογιστικής. Μέγιστης σημασίας είναι η κατασκευή τεχνητών πρακτόρων που συμπεριφέρονται ορθά, δλδ να έχουν αυτόνομη δράση, κατανόηση του περιβάλλοντος, προσαρμογή στις αλλαγές και χρονική διάρκεια, κατανόηση των σκοπών άλλων πρακτόρων κ.α.. Τέλος, για να συμπεριφερθεί ένα σύστημα τεχνητής νοημοσύνης ορθά, θα πρέπει να συνδυάζονται και όλες οι ικανότητες που αναφέρθηκαν στο Turing test.

Η τεχνητή νοημοσύνη έχει τις καταβολές της στη φιλοσοφία, τη μάθηση, την αντίληψη και τον συλλογισμό, βασικές έννοιες που απασχολούσαν τους φιλοσόφους από την αρχαιότητα. Στην τεχνητή νοημοσύνη, υπάρχουν πρότυπα εκφράσεων που οδηγούν στα σωστά συμπεράσματα από τις σωστές εκφράσεις, γνωστά ως και οι «συλλογισμοί του Αριστοτέλη». Επί παραδείγματι, «Ο Σωκράτης είναι άνθρωπος. Όλοι οι άνθρωποι είναι θνητοί. Άρα, ο Σωκράτης είναι θνητός». Επίσης, έχει γίνει εμβάθυνση στη μελέτη της λογικής και της μαθηματικής λογικής. Δηλαδή μελετήθηκαν οι τρόποι του ορθού συλλογισμού και ευρύτερα ο τομέας της μαθηματικής λογικής, όπως η άλγεβρα Boole, η αναπαράσταση γνώσης με μαθηματικά

ορισμένες γλώσσες, όπως ο κατηγορηματικός λογισμός ή κατηγορηματική λογική που είναι συμβολική γλώσσα για την αναπαράσταση της γνώσης. Τέλος, η τεχνητή νοημοσύνη έχει επηρεαστεί και από άλλες επιστήμες όπως η ψυχολογία, η γλωσσολογία, οι οικονομικές επιστήμες, η ιατρική κ.α..

Η τεχνητή νοημοσύνη έχει δώσει τεράστια ώθηση στην εξέλιξη της αναγνώρισης προτύπων και εικόνων. Διεθνώς χρησιμοποιούνται διάφοροι όροι **image recognition**, **picture recognition** και **photo recognition** εναλλακτικά για την ίδια διεργασία. Όταν από ένα σύστημα ζητάμε να αναγνωρίσει ένα στοιχείο από μία εικόνα, στην ουσία ζητάμε να εντοπίσει και να κατηγοριοποιήσει το στοιχείο αυτό με επιτυχία. Όταν το σύστημα δει ένα αντικείμενο ή μία σκηνή, μπορεί αυτόματα να προσδιορίσει τα αντικείμενα ως διαφορετικές περιπτώσεις και να το συσχετίσει με τους επιμέρους ορισμούς. Ωστόσο, η οπτική αναγνώριση είναι ένα εξαιρετικά πολύπλοκο έργο για τα συστήματα και απαιτείται σημαντική επεξεργαστική ισχύς. Η αναγνώριση εικόνας με τεχνητή νοημοσύνη είναι μια μακροχρόνια έρευνα στον τομέα της υπολογιστικής όρασης. Με την πάροδο του χρόνου έχουν εξελιχθεί διάφορες μέθοδοι ώστε να μιμηθούν την ανθρώπινη όραση, με κοινό στόχο της αναγνώρισης εικόνας να παραμένει ο εντοπισμός των αντικειμένων και η κατάταξή τους σε διαφορετικές κατηγορίες. Στην εικόνα 15, βλέπουμε την αναγνώριση εικόνας με τον πιο ισχυρό ανιχνευτή αντικειμένων YOLO V.7 – Viso Suite.



Εικόνα 15

Παρακάτω θα δούμε ένα παράδειγμα ενός συστήματος dashcam και τον τρόπο αναγνώρισης οχημάτων μέσα από εικόνες. Στόχος του συστήματος είναι να εντοπίσει τα οχήματα που υπάρχουν στο οδόστρωμα. Θα δούμε τέσσερις διαφορετικές κατηγορίες αναφορικά με την αναγνώριση:

- ❖ Εξαγωγή χαρακτηριστικών.
- ❖ Εκπαίδευση του ταξινομητή.

- ❖ Αναζήτηση σε κυλιόμενο παράθυρο.
- ❖ Ανίχνευση και heatmap\* οχημάτων.

(\* Heatmap ή χάρτης θερμότητας, είναι μία τεχνική απεικόνισης δεδομένων, που δείχνει το μέγεθος ενός φαινομένου ως χρώμα σε δύο διαστάσεις. Η διακύμανση του χρώματος μπορεί να είναι λόγω της απόχρωσης ή της έντασης, δίνοντας εμφανείς οπτικές ενδείξεις στον αναγνώστη, σχετικά με το πώς το φαινόμενο είναι ομαδοποιημένο ή ποικίλλει στον χώρο.)

### Διαδικασία εξαγωγής χαρακτηριστικών.

Για να προσδιορίσουμε τα οχήματα σε μια εικόνα, χρειαζόμαστε μια ιδέα για το τι ακριβώς ψάχνουμε – τι χρειαζόμαστε, δηλαδή την «υπογραφή» του οχήματος. Αυτή η υπογραφή, πρέπει να προέρχεται από τις ίδιες τιμές των pixels, γιατί αυτές είναι όλες οι πληροφορίες που έχουμε. Για να δημιουργήσουμε την υπογραφή του οχήματος, δημιουργήσαμε τρία διαφορετικά σύνολα χαρακτηριστικών:

- **Χωρικά χαρακτηριστικά.** Χωρικά χαρακτηριστικά ονομάζονται οι ίδιες τιμές του εικονοστοιχείου (pixel). Γίνεται αλλαγή του μεγέθους της εικόνας και εξομάλυνση σε μια μονοδιάστατη συστοιχία και οι τιμές αυτές χρησιμοποιήθηκαν απευθείας σαν "υπογραφή". Εδώ χρησιμοποιήθηκαν εικόνες RGB.
- **Χαρακτηριστικά έγχρωμου ιστογράμματος.** Ένα ιστόγραμμα χρωμάτων αθροίζει τον αριθμό των τιμών pixel που πέφτουν σε ομοιόμορφα κατανεμημένους κάδους για το χρωματικό φάσμα της εικόνας. Αυτό ουσιαστικά μας λέει πως "υπάρχει πολύ σκούρο κόκκινο" ή "όχι πολύ ανοιχτό πράσινο" (όταν χρησιμοποιούνται εικόνες RGB). Αυτά τα σύνολα χρωμάτων, προστίθενται στην υπογραφή του οχήματος. Εδώ έχουν επιλεγεί 32 χρωματικοί «κάδοι» και εικόνες στον χρωματικό χώρο HSV για το συγκεκριμένο σετ χαρακτηριστικών. Στην εικόνα 16, βλέπουμε αριστερά μία εικόνα RGB και δεξιά μία απεικόνιση του καναλιού S για μία εικόνα HSV. Το S προέρχεται από τον κορεσμό (saturation).

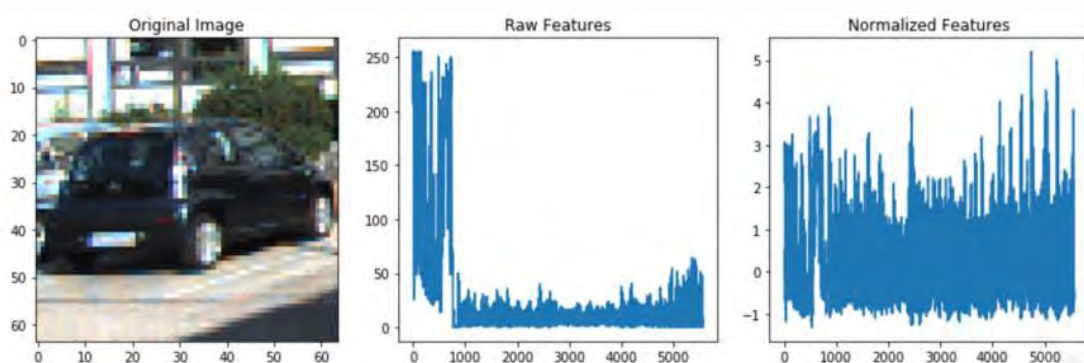


Εικόνα 16

- **Χαρακτηριστικά HOG (Histogram of Oriented Gradients).** Το ιστόγραμμα προσανατολισμένων διαβαθμίσεων, είναι ένα χαρακτηριστικό περιγραφής που χρησιμοποιείται σε μοντέλα υπολογιστικής όρασης και επεξεργασίας εικόνας, για το σκοπό της ανίχνευσης ενός αντικειμένου.

### Εκπαίδευση του ταξινομητή.

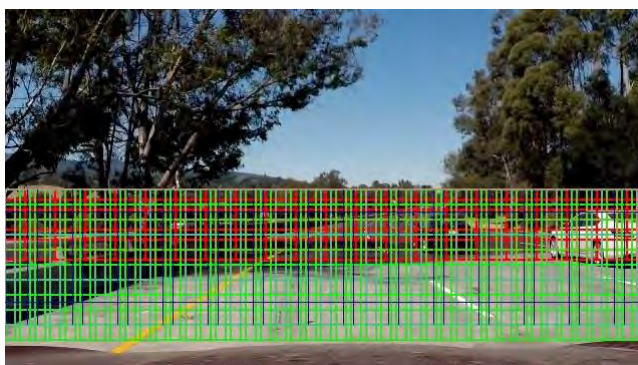
Για κάθε εικόνα που θέλουμε να εκπαιδύσουμε τον ταξινομητή μέσω της μηχανικής μάθησης, τα χαρακτηριστικά που αναφέραμε παραπάνω εξάγονται και συνενώνονται για να σχηματίσουν την εικόνα «υπογραφή». Εκπαιδεύτηκε μία γραμμική SVM (Support Vector Machine – Μηχανή Διανυσματικής Υποστήριξης) ταξινομητή σε πάνω από 17.000 εικόνες (το μισό εκ των οποίων περιείχε οχήματα). Τα χαρακτηριστικά ομαλοποιήθηκαν πριν περάσουν στον ταξινομητή. Τα αποτελέσματα των δοκιμών έδειξαν ακρίβεια πάνω από 99%. Στην εικόνα 17, φαίνεται ένα παράδειγμα οχήματος σε μία εικόνα του ταξινομητή που εκπαιδεύτηκε. Τα μη επεξεργασμένα χαρακτηριστικά, πρέπει να ομαλοποιηθούν ώστε να βοηθήσει ο ταξινομητής να μάθει καλύτερα.



Εικόνα 17

### Αναζήτηση σε κυλιόμενο παράθυρο.

Ένας εκπαιδευμένος ταξινομητής έχει επιτυχή λειτουργία, μόνο όταν του έχουν δοθούν παρόμοιες πληροφορίες για τις οποίες έχει εκπαιδευτεί. Στην παρούσα περίπτωση εκπαιδεύτηκε μία SVM σε εικόνες 64 x 64 pixels, ώστε να κάνει περικοπή των εικόνων που λαμβάνει από το dashcam. Το τμήμα της εικόνας που θα μπορούσε να ανιχνευθεί όχημα είναι το δεύτερο μισό, το τμήμα δηλ που είναι σκιαγραφημένο με κόκκινο και πράσινο – εικόνα 18. Έτσι, εκτελέστηκε μία αναζήτηση στο συγκεκριμένο τμήμα της εικόνας, ώστε να ελεγχθεί και να διαπιστωθεί εάν υπάρχει κάποιο όχημα.



Εικόνα 18

### Αναγνώριση και heatmap οχημάτων

Η αναζήτηση στο συγκεκριμένο τμήμα παράγει πολλά παράθυρα – ίσως εκατοντάδες. Κάθε ένα από αυτά τα παράθυρα αντιπροσωπεύει μια εικόνα που θα ελεγχθεί από τον ταξινομητή, πράγμα που σημαίνει ότι κάθε παράθυρο, πρέπει να έχει χαρακτηριστικά που έχουν



εξαχθεί για αυτό. Αυτό είναι το κύριο εμπόδιο της διαδικασίας. Ωστόσο, από τη στιγμή που τα χαρακτηριστικά εξορύσσονται και ταξινομούνται για κάθε «παράθυρο – εικόνα», τα παράθυρα που περιέχουν τα οχήματα παραμένουν. Δυστυχώς, αυτά τα παράθυρα συνήθως δεν παράγουν ωραία, ενιαία πλαίσια οριοθέτησης γύρω από τα οχήματα, όπως θέλουμε. Για να επιτευχθεί ένα ενιαίο πλαίσιο οριοθέτησης ανά όχημα, εφαρμόζεται ένας χάρτης θερμότητας «*heatmap*» και σχεδιάζεται ένα νέο πλαίσιο οριοθέτησης γύρω από τις «θερμές» περιοχές. Τα αποτελέσματα πριν και μετά τον χάρτη θερμότητας εμφανίζονται στις εικόνες 19 και 20.



Εικόνα 19



Εικόνα 20

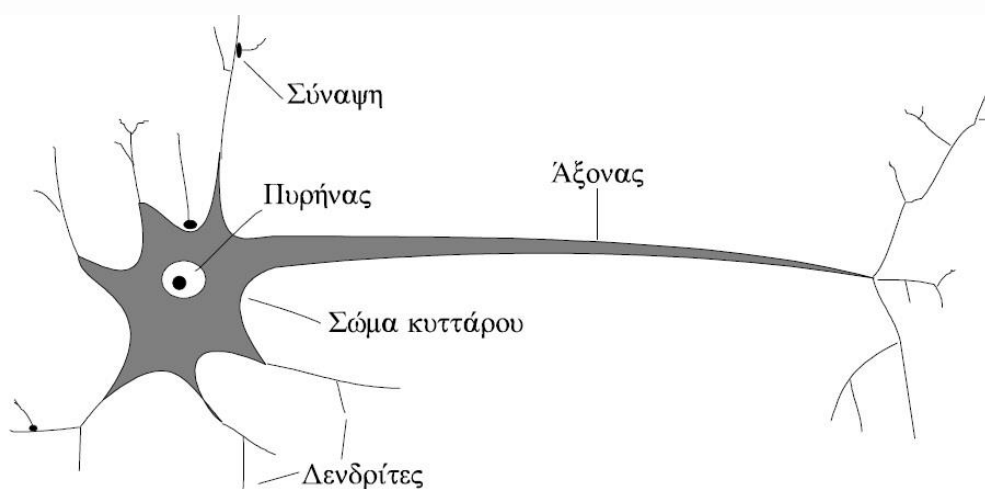
Στην εικόνα 6, φαίνονται τα παράθυρα που θεωρούνταν ότι περιέχουν οχήματα και στην εικόνα 7, βλέπουμε το τελικό αποτέλεσμα μετά την εφαρμογή ενός χάρτη θερμότητας «*heatmap*» όπου επιλέγεται ένα μόνο πλαίσιο οριοθέτησης.

### 3.3 Νευρωνικά Δίκτυα

Νευρωνικό δίκτυο ονομάζεται ένα κύκλωμα διασυνδεδεμένων νευρώνων. Στην περίπτωση βιολογικών νευρώνων, πρόκειται για ένα τμήμα νευρικού ιστού. Στην περίπτωση τεχνητών νευρώνων, πρόκειται για ένα αφηρημένο αλγοριθμικό κατασκεύασμα το οποίο εμπίπτει στον τομέα της υπολογιστικής νοημοσύνης, όταν στόχος του νευρωνικού δικτύου είναι η επίλυση κάποιου υπολογιστικού προβλήματος ή της υπολογιστικής νευροεπιστήμης, όταν στόχος είναι η υπολογιστική προσομοίωση της λειτουργίας των βιολογικών νευρωνικών δικτύων με βάση κάποιο μαθηματικό μοντέλο τους.

Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks) αρχικά προτάθηκαν ως ένα μαθηματικό μοντέλο προσομοίωσης της πολύπλοκης λειτουργίας του ανθρώπινου εγκεφάλου. Η δομή του εγκεφάλου είναι τέτοια, ώστε να επιτρέπει την παράλληλη επεξεργασία δεδομένων και τη δυνατότητα συνεχούς μάθησης μέσω της αλληλεπίδρασης με το περιβάλλον. Τα δύο αυτά βασικά χαρακτηριστικά συμβάλλουν στην ικανότητα, αφενός, να εκτελεί δύσκολα καθήκοντα, όπως ταχύτατη αναγνώριση μορφών, ταξινόμηση κ.ά., αφετέρου, να εξελίσσεται συνεχώς μαθαίνοντας από το περιβάλλον του κατά την αλληλεπίδρασή του με αυτό.

Η δομή του τεχνητού νευρωνικού δικτύου, μιμείται κατά το δυνατό εκείνη του βιολογικού νευρωνικού δικτύου όπως βλέπουμε στην εικόνα 21, ώστε να εμφανίζει παρόμοιες ιδιότητες. Κατ' αναλογία με ένα δίκτυο νευρώνων εγκεφάλου, ένα τεχνητό νευρωνικό δίκτυο αποτελείται από ένα σύνολο τεχνητών νευρώνων που αλληλοεπιδρούν, συνδεόμενοι μεταξύ τους με τις λεγόμενες συνάψεις (synapses). Ο βαθμός αλληλεπίδρασης είναι διαφορετικός για κάθε ζεύγος νευρώνων και καθορίζεται από τα λεγόμενα συναπτικά βάρη (synaptic weights).



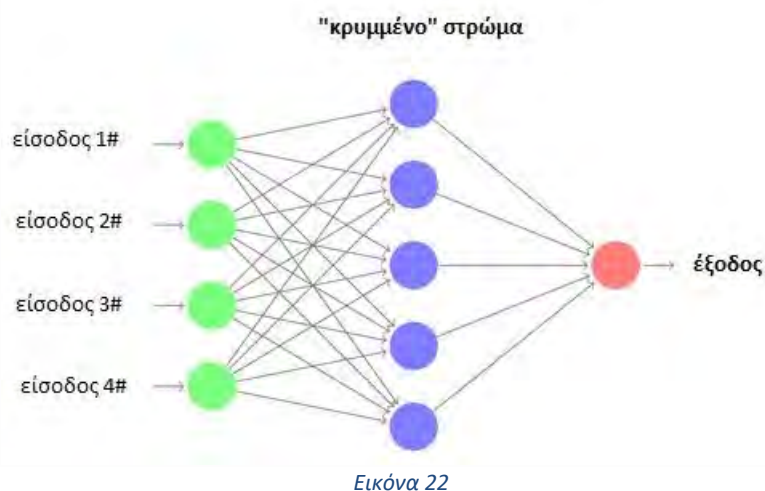
Εικόνα 21

Συγκεκριμένα, καθώς το νευρωνικό δίκτυο αλληλοεπιδρά με το περιβάλλον και μαθαίνει από αυτό, τα συναπτικά βάρη μεταβάλλονται συνεχώς, ενδυναμώνοντας ή αποδυναμώνοντας την ισχύ του κάθε δεσμού. Όλη η εμπειρική γνώση που αποκτά επομένως το νευρωνικό δίκτυο από το περιβάλλον, κωδικοποιείται στα συναπτικά βάρη. Αυτά αποτελούν το χαρακτηριστικό εκείνο που δίνει στο δίκτυο την ικανότητα για εξέλιξη και προσαρμογή στο περιβάλλον.

Υπάρχουν δύο τρόποι να εκπαιδύσουμε ένα δίκτυο. Κατά τον πρώτο τρόπο, η εκπαίδευση γίνεται με εποπτεία. Στην περίπτωση αυτή το δίκτυο τροφοδοτείται με ένα σύνολο γνωστών παραδειγμάτων, δηλαδή ένα σύνολο καταστάσεων στις οποίες μπορεί να περιέλθει το

δίκτυο, μαζί με τα αποτελέσματα που θέλουμε να δίνει το δίκτυο για τις καταστάσεις αυτές. Για να μάθει το δίκτυο τα παραδείγματα αυτά, χρησιμοποιούμε έναν αλγόριθμο εκπαίδευσης. Ο αλγόριθμος εκπαίδευσης που θα χρησιμοποιηθεί εξαρτάται από το εκάστοτε πρόβλημα και από τη δομή του δικτύου που επιλέγουμε για να το αντιμετωπίσουμε. Κατά το δεύτερο τρόπο, η εκπαίδευση γίνεται χωρίς εποπτεία. Στην περίπτωση αυτή το δίκτυο καλείται να αναγνωρίσει ομοιότητες και μοτίβα, σε δεδομένα που του έχουμε τροφοδοτήσει. Τα δεδομένα παρουσιάζονται στο δίκτυο και αυτό οφείλει να προσαρμοστεί, έτσι ώστε να τα χωρίσει σε ομάδες. Η διαδικασία αυτή επαναλαμβάνεται, ώσπου δεν παρατηρείται μεταβολή στην ταξινόμηση των δεδομένων.

Το βασικό πλεονέκτημα των τεχνητών νευρωνικών δικτύων είναι ότι μπορούν να αποθηκεύσουν γνώση και εμπειρία από το περιβάλλον, την οποία μπορεί στη συνέχεια να ανακαλέσει. Επιπλέον, έχει τη δυνατότητα να γενικεύει, δηλαδή να εξάγει τα βασικά χαρακτηριστικά ενός συστήματος, ακόμα και όταν αυτά είναι κρυμμένα σε θορυβώδη δεδομένα. Στην εικόνα 22 βλέπουμε τη δομή ενός τυπικού τεχνητού νευρωνικού δικτύου.

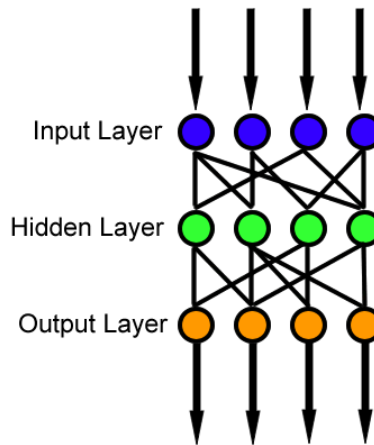


Για να αρχίσει το νευρωνικό δίκτυο να «μαθαίνει», θα χρειαστεί ένα μεγάλο αριθμό δεδομένων που θα είναι κατάλληλα επιλεγμένα έτσι ώστε μέσω της διαδικασίας της προπόνησης και οπισθοδρόμησης, να εξαχθούν τα κατάλληλα χαρακτηριστικά. Με αυτά το δίκτυο θα είναι σε θέση να εκτελέσει μία πιο πληροφορημένη διαδικασία αναγνώρισης. Για την βελτιστοποίηση αυτής της διαδικασίας υπάρχουν ορισμένες μεταβλητές οι οποίες αν τροποποιηθούν και θα λάβουν τις κατάλληλες τιμές επιτρέποντας στο δίκτυο να προπονηθεί πιο γρήγορα καθώς και να αναγνωρίζει με μεγαλύτερη ακρίβεια. Δύο από αυτές χρησιμοποιούνται και σε αυτήν την έρευνα και είναι ο ρυθμός μάθησης και το μέγεθος παρτίδας. Ο σκοπός τους είναι να καθορίζουν την ταχύτητα με την οποία αυξομειώνονται τα βάρη των φίλτρων και τον αριθμό των δειγμάτων που πρόκειται να διαδοθούν μέσω του δικτύου, αντίστοιχα. Υπάρχουν αρκετοί διαφορετικοί τύποι ΝΔ, και ο καθένας έχει αναπτυχθεί για να εξυπηρετεί και ένα διαφορετικό σκοπό, χρησιμοποιώντας διαφορετικές μαθηματικές λειτουργίες. Παρακάτω θα αναλύσουμε κάποιους τύπους νευρωνικών δικτύων, ανάλογα με τον τρόπο τον οποίο λειτουργούν.

#### ❖ **Feedforward Neural Network (FNN).**

Το νευρωνικό δίκτυο τροφοδότη είναι ο πρώτος και ο απλούστερος τύπος νευρωνικού δικτύου. Εδώ, οι πληροφορίες κινούνται μόνο προς μία κατεύθυνση προς

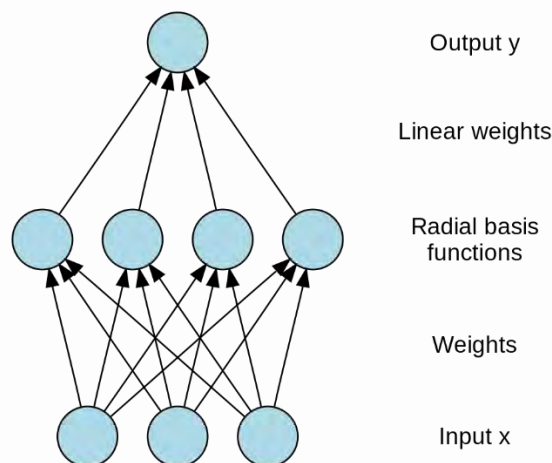
τα εμπρός από τους κόμβους εισόδου και μέσω των κρυφών κόμβων προς τους κόμβους εξόδου. Δεν υπάρχουν κύκλοι ή βρόγχοι στο συγκεκριμένο τύπο νευρωνικού δικτύου, οποίος βρίσκει εφαρμογή σε περιπτώσεις όπως η αναγνώριση ομιλίας και στο computer vision. Στην εικόνα 10 παρατηρούμε ένα νευρωνικό δίκτυο τύπου *feedforward*.



Εικόνα 23

#### ❖ Νευρωνικά δίκτυα με χρήση της συνάρτησης *Radial Basis*.

Τα νευρωνικά δίκτυα με τη χρήση της συνάρτησης *Radial Basis* διατυπώθηκαν για πρώτη φορά σε ένα έγγραφο του 1988 από τους Broomhead και Lowe, οι οποίοι ήταν και οι δύο ερευνητές στο Royal Signals και στο Radar Establishment. Στο πεδίο της μαθηματικής μοντελοποίησης, ένα δίκτυο συναρτήσεων ακτινικής βάσης είναι ένα τεχνητό νευρωνικό δίκτυο που χρησιμοποιεί συναρτήσεις ακτινικής βάσης ως λειτουργίες ενεργοποίησης. Η έξοδος του δικτύου είναι ένας γραμμικός συνδυασμός των ακτινικών συναρτήσεων βάσης, των εισόδων και των παραμέτρων των νευρώνων. Τα δίκτυα συναρτήσεων ακτινικής βάσης έχουν πολλές χρήσεις, συμπεριλαμβανομένης της προσέγγισης συναρτήσεων, της πρόβλεψης χρονοσειρών, της ταξινόμησης και του ελέγχου συστημάτων. Στην εικόνα 24, βλέπουμε ένα νευρωνικό δίκτυο με χρήση της συνάρτησης *Radial Basis*.

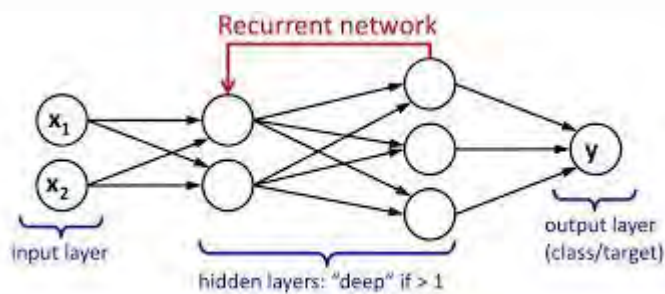


Εικόνα 24



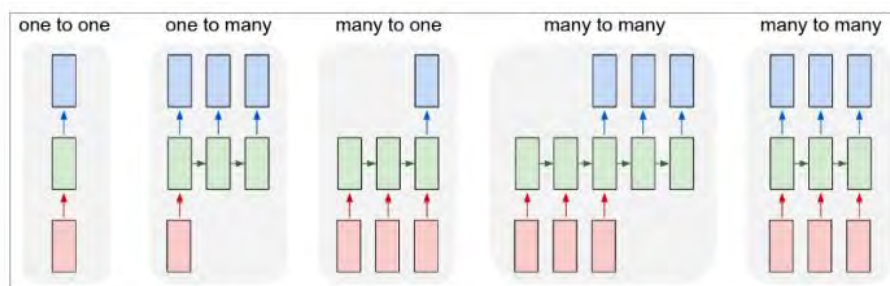
### ❖ Επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks).

Ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο είναι ένας τύπος νευρωνικού δικτύου που περιέχει βρόχους, επιτρέποντας την αποθήκευση πληροφοριών μέσα στο δίκτυο. Εν ολίγοις, τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα χρησιμοποιούν το συλλογισμό τους από προηγούμενες εμπειρίες, για να πληροφορήσουν τα επερχόμενα γεγονότα. Τα επαναλαμβανόμενα μοντέλα είναι πολύτιμα στην ικανότητά τους να αλληλουχούν διανύσματα, γεγονός που ανοίγει το API για την εκτέλεση πιο περίπλοκων εργασιών. Τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα μπορούν να θεωρηθούν ως μια σειρά από δίκτυα που συνδέονται μεταξύ τους. Συχνά έχουν μια αλυσιδωτή αρχιτεκτονική, που τα καθιστά εφαρμόσιμα σε εργασίες όπως η αναγνώριση ομιλίας, η μετάφραση γλώσσας κλπ. Στην εικόνα 25 παρατηρούμε ένα τυπικό διάγραμμα ενός Ε.Ν.Δ. .



Εικόνα 25

Ένα Ε.Ν.Δ. μπορεί να σχεδιαστεί για να λειτουργεί σε αλληλουχίες διανυσμάτων στην είσοδο, την έξοδο, ή και τα δύο. Για παράδειγμα, μια ακολουθία δεδομένων εισόδου, μπορεί να λάβει μια πρόταση ως είσοδο και να εξάγει μια θετική ή αρνητική τιμή συναισθήματος. Εναλλακτικά, μια έξοδος σε ακολουθία, μπορεί να πάρει μια εικόνα ως είσοδο και να παράγει μια πρόταση ως έξοδο. Ας φανταστούμε την εκπαίδευση ενός Ε.Ν.Δ. στη λέξη "happy", με τα γράμματα "h, a, p, y". Το Ε.Ν.Δ. θα εκπαιδευτεί σε τέσσερα ξεχωριστά παραδείγματα, καθένα από τα οποία αντιστοιχεί στην πιθανότητα ότι τα γράμματα θα πέσουν σε μια επιδιωκόμενη ακολουθία. Για παράδειγμα, το δίκτυο θα εκπαιδευτεί για να κατανοήσει την πιθανότητα ότι το γράμμα "a" θα πρέπει να ακολουθήσει στο πλαίσιο του "h". Ομοίως, το γράμμα "p" θα πρέπει να εμφανίζεται μετά από ακολουθίες του "ha". Και πάλι, μια πιθανότητα θα υπολογιστεί για το γράμμα "p" μετά την ακολουθία "hap." Η διαδικασία θα συνεχιστεί μέχρι να υπολογιστούν οι πιθανότητες να προσδιοριστεί η πιθανότητα των γραμμάτων να πέσουν στην επιθυμητή ακολουθία. Έτσι, καθώς το δίκτυο λαμβάνει κάθε είσοδο, θα καθορίσει την πιθανότητα του επόμενου γράμματος με βάση την πιθανότητα του προηγούμενου γράμματος ή της ακολουθίας. Με την πάροδο του χρόνου, το δίκτυο μπορεί να ενημερωθεί ώστε να παράγει αποτελέσματα με μεγαλύτερη ακρίβεια. Ένα κοινό παράδειγμα των Επαναλαμβανόμενων Νευρωνικών Δικτύων είναι η μηχανική μετάφραση. Για παράδειγμα, ένα νευρωνικό δίκτυο μπορεί να πάρει μια πρόταση εισόδου στα ισπανικά και να την μεταφράσει σε μια πρόταση στα αγγλικά. Το δίκτυο καθορίζει την πιθανότητα κάθε λέξης στην πρόταση εξόδου με βάση την ίδια τη λέξη και την προηγούμενη ακολουθία εξόδου. Στην εικόνα 26 φαίνονται οι τρόποι λειτουργίας ενός Ε.Ν.Δ. .

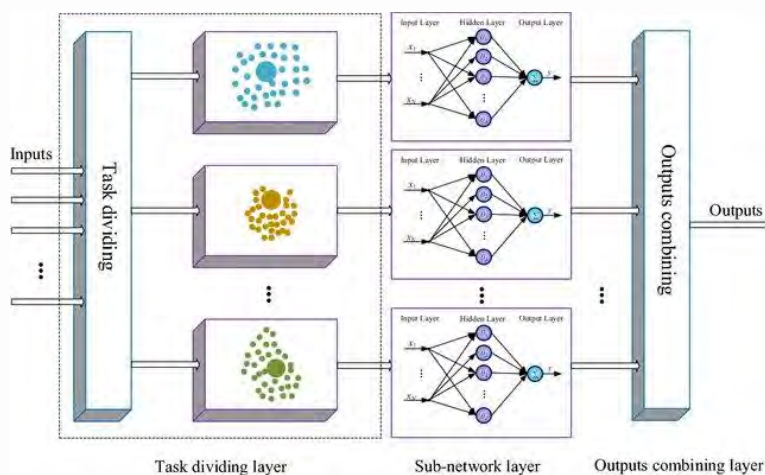


Εικόνα 26

### ❖ Αρθρωτά νευρωνικά δίκτυα (Modular Neural Networks).

Ένα αρθρωτό νευρωνικό δίκτυο είναι ένα τεχνητό νευρωνικό δίκτυο, που χαρακτηρίζεται από μια σειρά ανεξάρτητων νευρωνικών δικτύων που ελέγχονται από κάποιο ενδιάμεσο. Κάθε ανεξάρτητο νευρωνικό δίκτυο, χρησιμεύει ως μια μονάδα και λειτουργεί σε ξεχωριστές εισόδους για να επιτύχει κάποιο υπο-έργο του έργου που το δίκτυο ελπίζει να εκτελέσει. Ο ενδιάμεσος λαμβάνει τις εξόδους κάθε μονάδας και τις επεξεργάζεται, για την παραγωγή του δικτύου στο σύνολό του. Ο ενδιάμεσος αποδέχεται μόνο τις εξόδους των λειτουργικών μονάδων, δεν ανταποκρίνεται ούτε σηματοδοτεί με άλλο τρόπο τις λειτουργικές μονάδες. Επίσης, οι μονάδες δεν αλληλοεπιδρούν μεταξύ τους. Καθώς η έρευνα για τα τεχνητά νευρωνικά δίκτυα εξελίσσεται, είναι σκόπιμο τα τεχνητά νευρωνικά δίκτυα να συνεχίσουν να βασίζονται στη βιολογική τους έμπνευση και να μιμούνται την τμηματοποίηση και τη διαφοροποίηση που υπάρχει στον εγκέφαλο. Για παράδειγμα, ο εγκέφαλος χωρίζει το πολύπλοκο έργο της οπτικής αντίληψης σε πολλές υπο-εργασίες. Μέσα σε ένα μέρος του εγκεφάλου, που ονομάζεται θάλαμος, βρίσκεται ο πλάγιος γενετικός πυρήνας (Lateral Geniculate Nucleus), ο οποίος χωρίζεται σε στρώματα όπου επεξεργάζεται ξεχωριστά τα δύο βασικά συστατικά της όρασης, το χρώμα και την αντίθεση. Αφού ο πλάγιος γενετικός πυρήνας επεξεργάζεται κάθε συστατικό παράλληλα, περνάει το αποτέλεσμα σε άλλη περιοχή για να το μεταγλωττίσει.

Κάποιες εργασίες που χειρίζεται ο εγκέφαλος, όπως η όραση, χρησιμοποιούν μια ιεραρχία υπο-δικτύων. Ωστόσο, δεν είναι σαφές αν ορισμένοι ενδιάμεσοι συνδέουν αυτές τις ξεχωριστές διαδικασίες μεταξύ τους. Μάλλον, καθώς οι εργασίες γίνονται πιο αφηρημένες, οι μονάδες επικοινωνούν μεταξύ τους, σε αντίθεση με το μοντέλο αρθρωτού νευρωνικού δικτύου.

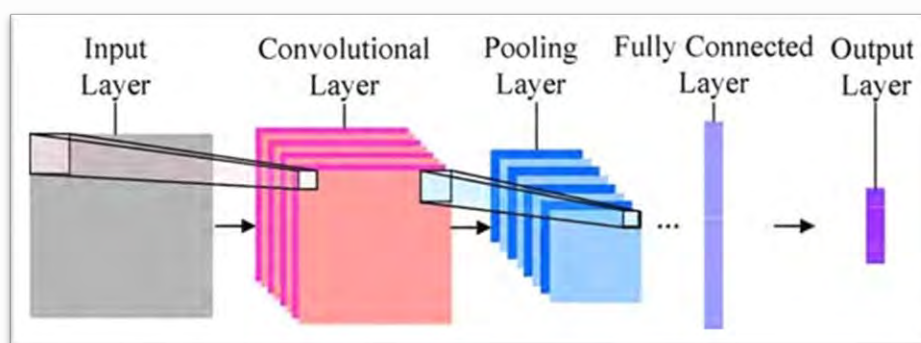


Εικόνα 27

## ❖ Συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks).

Κάθε τύπος νευρωνικού δικτύου ειδικεύεται στην επίλυση ξεχωριστής γκάμας προβλημάτων. Για την επεξεργασία εικόνας και βίντεο με σκοπό την αναγνώριση κοινών χαρακτηριστικών, χρησιμοποιούνται τα Συνελικτικά Νευρωνικά Δίκτυα. Στη βαθιά μάθηση, ένα Συνελικτικό Νευρωνικό Δίκτυο (Convolutional Neural Network ή CNN, ή ConvNet) είναι μια κατηγορία του τεχνητού νευρωνικού δικτύου, που εφαρμόζεται συνηθέστερα για την ανάλυση των οπτικών εικόνων. Τα CNN είναι επίσης γνωστά ως Shift Invariant ή Space Invariant Artificial Neural Networks (SIANN). Τα περισσότερα νευρωνικά δίκτυα δεν είναι αμετάβλητα στη μετάφραση, λόγω της λειτουργίας μείωσης δειγματοληψίας που εφαρμόζουν στην είσοδο. Έχουν εφαρμογές στην αναγνώριση εικόνας και βίντεο, συστήματα συστάσεων, ταξινόμηση εικόνας, κατακερματισμό εικόνας, ιατρική ανάλυση εικόνας, επεξεργασία φυσικής γλώσσας, διεπαφές εγκεφάλου-υπολογιστή, και οικονομικές χρονοσειρές.

Τα συνελικτικά νευρωνικά δίκτυα είναι όμοια με κανονικά Feedforward ΝΔ τα οποία δέχονται σαν είσοδο αρχεία εικόνας ή βίντεο. Ο κύριος σκοπός τέτοιων αρχιτεκτονικών είναι η αναγνώριση μοτίβων και αντικειμένων. Για τον άνθρωπο αυτή η διαδικασία είναι απλή, ωστόσο για έναν υπολογιστή όταν δέχεται μια εικόνα σαν είσοδο την αντιλαμβάνεται ως ένα πίνακα με αριθμούς. Αυτοί οι αριθμοί μπορούν να πάρουν αρκετές διαφορετικές τιμές που εξαρτώνται από την κωδικοποίηση της εικόνας. Οι πιο κοινές τεχνικές κωδικοποίησης είναι η RGB και η grayscale. Η πρώτη έχει τη δυνατότητα να αναγνωρίζει το χρώμα που αναπαριστά κάθε pixel σαν συνδυασμός των τριών κύριων χρωμάτων (κόκκινο, πράσινο και μπλε), αλλά αναγκαστικά είναι πιο ακριβή σε θέμα μνήμης. Η δεύτερη παρόλο που αδυνατεί να αναγνωρίσει την έννοια του χρώματος, δίνει στο κάθε pixel μια αριθμητική τιμή που αντιστοιχίζεται σε μια απόχρωση του χρώματος γκρι, επιτρέποντας ταχύτερους χρόνους αναγνώρισης.



Εικόνα 28

Τα ΣΝΔ διαθέτουν διαφορετικές αρχιτεκτονικές δικτύων από τα απλά ΝΔ. Το κάθε επίπεδο έχει τρεις διαστάσεις, ύψος, πλάτος, μήκος. Οι νευρώνες σε δύο διαφορετικά επίπεδα δεν είναι αναγκαστικό να είναι πλήρως συνδεδεμένοι μεταξύ τους. Το αποτέλεσμα του ΣΝΔ είναι ένα διάνυσμα αριθμών όπου η κάθε τιμή του, θα αναπαριστά μια πιθανότητα η εικόνα που λήφθηκε στην είσοδο να ανήκει σε μία κλάση (στην δική μας περίπτωση η κλάση είναι η ύπαρξη πεζού στην εικόνα). Επιπρόσθετα η δομή των ΣΝΔ αποτελείται από δύο κύρια κομμάτια. Το ένα είναι η εξαγωγή χαρακτηριστικών, το οποίο αφιερώνεται στην εξαγωγή χαρακτηριστικών

από την εικόνα, εφαρμόζοντας μια σειρά συνελίξεων (convolution) και συγκεντρώσεων (pooling). Το δεύτερο κομμάτι είναι η διαδικασία κατάταξης της εικόνας με βάση τα χαρακτηριστικά που έχουν ήδη βρεθεί.

Ο τρόπος εξαγωγής δεδομένων βασίζεται κυρίως στις συνελίξεις που εκτελούνται πάνω στην εικόνα που θα δοθεί σαν είσοδος στο ΣΝΔ. Η συνέλιξη εκτελείται με την βοήθεια φίλτρων παράγοντας έτσι έναν πολυδιάστατο πίνακα, ο οποίος ονομάζεται χάρτης χαρακτηριστικών. Έπειτα αυτός ο χάρτης χαρακτηριστικών θα δεχτεί αρκετή επεξεργασία μέσω άλλων μαθηματικών συναρτήσεων όπως είναι οι συνελίξεις, οι συγκεντρώσεις και Rectified Linear Units (ReLU), οι οποίες θα εφαρμόζονται σε διάφορα επίπεδα του ΣΝΔ. Αφότου λάβει τέλος αυτή η διαδικασία, ο τελικός χάρτης χαρακτηριστικών χρησιμοποιείται σαν είσοδος σε μια σειρά από πλήρως συνδεδεμένα στρώματα. Εντούτοις, τέτοιου είδους επίπεδα μπορούν να λάβουν σαν είσοδο αποκλειστικά μονοδιάστατους πίνακες, αναγκάζοντας έτσι τον πίνακα χαρακτηριστικών από τρισδιάστατο να μετατραπεί σε μονοδιάστατο για να παραχθεί το επιθυμητό αποτέλεσμα. Στην εικόνα 29, φαίνεται το παράδειγμα της εικόνας αριστερά με κωδικοποίηση RGB και της δεξιάς εικόνας με κωδικοποίηση grayscale.



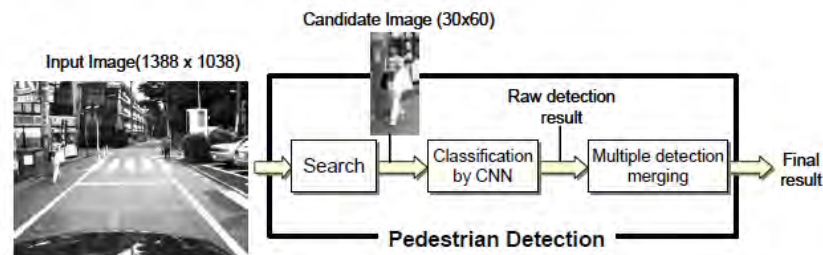
Εικόνα 29

Παρακάτω, θα δούμε ένα παράδειγμα αναφορικά με την αναγνώριση πεζών με τη χρήση των Συνελικτικών Νευρωνικών Δικτύων. Το επιστημονικό άρθρο, δημοσιεύθηκε στο **IEEE Proceedings – Intelligent Vehicles Symposium, 2005**, από μία ερευνητική ομάδα του τμήματος έρευνας και ανάπτυξης, της εταιρίας «DENSO IT LABORATORY INC.» που εδρεύει στην Ιαπωνία. Το σχηματικό διάγραμμα του συστήματος ανίχνευσης πεζών μας απεικονίζεται στην εικόνα 30. Η εικόνα εισόδου από τη φωτογραφική μηχανή αναζητείται με παράθυρο ανίχνευσης  $30 \times 60$  εικονοστοιχείων σε διάφορες αναλύσεις, παρόμοια με το σύστημα του. Το μέγεθος του βήματος είναι 3 pixel και προς τις δύο κατευθύνσεις και χρησιμοποιούμε το συντελεστή κλιμάκωσης 0,9 για την κατασκευή της πυραμίδας εικόνας. Οι εικόνες  $30 \times 60$  υποψήφιων εισάγονται στον ταξινομητή ΣΝΝ χωρίς καμία προ-επεξεργασία, όπως για παράδειγμα, η αντιστάθμιση του φωτισμού. Το ΣΝΝ δίνει μια βαθμολογία ταξινόμησης σε κάθε υποψήφιο, και οι υποψήφιοι με τη βαθμολογία τους υψηλότερη από ένα όριο αποθηκεύονται στον κατάλογο ανίχνευσης ακατέργαστων δεδομένων. Αυτή η λίστα ανίχνευσης ακατέργαστων δεδομένων, ταξινομείται σύμφωνα με τις βαθμολογίες ταξινόμησης. Δεδομένου ότι ο ταξινομητής του ΣΝΝ είναι ανθεκτικός σε μικρές μετατοπίσεις και αλλαγές μεγέθους, συνήθως κάθε πεζός θα παράγει διάφορα αποτελέσματα στον ακατέργαστο κατάλογο ανίχνευσης. Προκειμένου να αφαιρέσουμε πολλαπλά αποτελέσματα για τον ίδιο πεζό, εκτελούμε μια λειτουργία συγχώνευσης πολλαπλών ανιχνεύσεων πριν εξαγάγουμε το τελικό αποτέλεσμα ανίχνευσης. Η λειτουργία συγχώνευσης πολλαπλής ανίχνευσης χρησιμοποιεί τον ακόλουθο αλγόριθμο:



- a. Μετακίνηση του πρώτου στοιχείου της ακατέργαστης λίστας ανίχνευσης, στο τέλος της τελικής λίστας ανίχνευσης.
- b. Διαγραφή όλων των στοιχείων της ανεπεξέργαστης λίστας ανίχνευσης, που επικαλύπτονται περισσότερο από ένα προκαθορισμένο κατώφλι, με το τελευταίο στοιχείο της τελικής λίστας ανίχνευσης
- c. Μετάβαση στο a. αν η λίστα ανίχνευσης raw δεν είναι κενή

Μετά τον τερματισμό αυτού του αλγορίθμου, η τελική λίστα ανίχνευσης κρατά τα τελικά αποτελέσματα του συστήματος ανίχνευσης. Σημειώνουμε, ότι αυτός ο συγχωνεύμενος αλγόριθμος επιτρέπει την ανίχνευση επικαλυπτόμενων πεζών, εφόσον η επικάλυψη είναι μικρότερη από ένα όριο. Ένας πιο σύνθετος αλγόριθμος συγχώνευσης θα μπορούσε να κάνει χρήση του γεγονότος ότι οι σωστοί πεζοί συνήθως παράγουν περισσότερες πρωτογενείς ανιχνεύσεις από ότι τα ψευδή θετικά, και μια ανίχνευση πρωτογενών θα ήταν αποδεκτή μόνο αν υπάρχουν αρκετές άλλες στην ίδια περιοχή. Είναι επίσης επιθυμητό να χρησιμοποιηθούν πληροφορίες χρονολογικής ακολουθίας, με την παρακολούθηση ανιχνευμένων πεζών και να γίνει αποδεκτή η ανίχνευση μόνο αν επιμείνει. Η παρούσα εργασία ωστόσο, εστιάζει στο συστατικό κατηγοριοποίησης, καθώς είναι το κλειδί για υψηλή ακρίβεια ανίχνευσης.

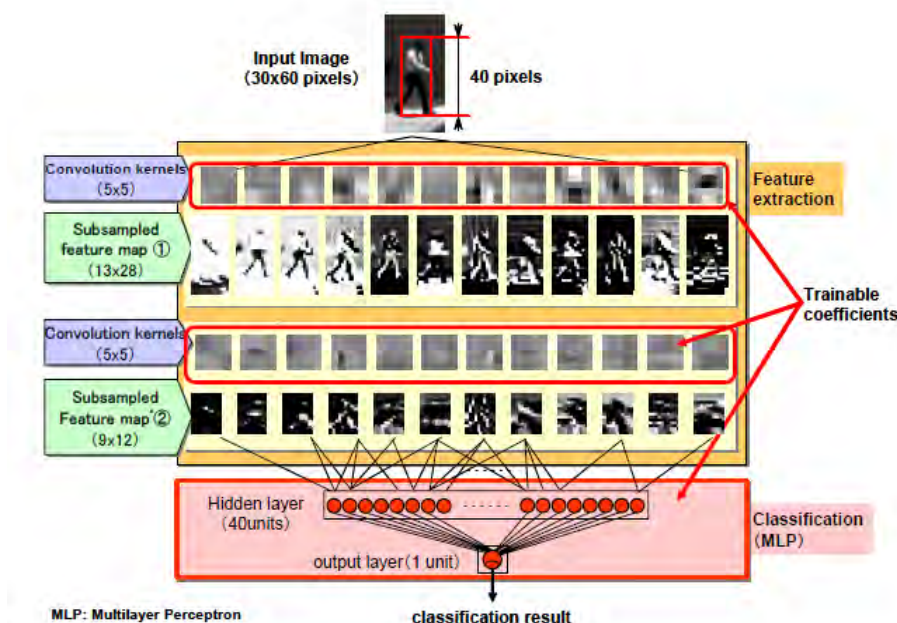


Εικόνα 30

Τα CNN είναι μια ειδική παραλλαγή των πολυστρωματικών perceptrons (Multilayer Perceptrons – MLP), στα οποία τα πρώτα επίπεδα είναι διαμορφωμένα έτσι, ώστε να λειτουργούν σαν ένας ιεραρχικός εξολκέας χαρακτηριστικών. Η διαφορά με το συνηθισμένο πλήρως συνδεδεμένο MLP, είναι ότι κάθε κόμβος επεξεργασίας στο χαρακτηριστικό που εξάγει τα επίπεδα – που ονομάζεται επίσης "χάρτες χαρακτηριστικών" – συνδέεται με ένα διαφορετικό υπο-ορθογώνιο του προηγούμενου επιπέδου και οι κόμβοι επεξεργασίας σε κάθε χάρτη χαρακτηριστικών, μοιράζονται το ίδιο διάνυσμα βάρους. Αυτή η δομή σύνδεσης διαμορφώνει ουσιαστικά την αντιστοίχιση των χαρακτηριστικών για την εκτέλεση ενός συμβατικού φιλτραρίσματος, με δυνατότητα μεταφοράς στην έξοδο του προηγούμενου επιπέδου. Οι χάρτες χαρακτηριστικών συνήθως, ακολουθούνται από ένα επίπεδο υπο-δειγματοληψίας που μειώνει τη διαστασιμότητα και επίσης βελτιώνει την ευρωστία σε μικρές παραμορφώσεις. Οι χάρτες χαρακτηριστικών υψηλότερου επιπέδου, μπορούν να βασίζονται σε διάφορους χάρτες χαμηλότερου επιπέδου, εφαρμόζοντας ένα ιεραρχικό μοντέλο χαρακτηριστικών που ομοιάζει με αυτό του οπτικού συστήματος θηλαστικών. Τα τελευταία στρώματα του CNN είναι πλήρως συνδεδεμένα, εφαρμόζοντας έναν γενικό ταξινομητή σκοπού πάνω στα χαρακτηριστικά που αποσπάστηκαν από τα προηγούμενα στρώματα.

Η δομή και τα αποτελέσματα της ενδιάμεσης επεξεργασίας του CNN που χρησιμοποιήθηκαν για τα πειράματα της εν λόγω εργασίας παρουσιάζονται στην εικόνα 31. Το μέγεθος του επιπέδου εισόδου είναι  $30 \times 60$  pixel. Το ύψος του πεζού στην εικόνα εισόδου είναι 40 pixel κατά τη διάρκεια της εκπαίδευσης. Το σχετικά μεγάλο περιθώριο των 10 pixel στην κορυφή και στο κάτω μέρος είναι απαραίτητο για να αντισταθμιστούν τα οριακά εφέ κατά τη

διάρκεια των δύο επιπέδων συνωστισμού. Υπάρχουν 12 χάρτες χαρακτηριστικών τόσο στο πρώτο όσο και στο δεύτερο επίπεδο. Η συνέλιξη και η υπο-δειγματοληψία, υλοποιούνται σε ένα μόνο βήμα για την υπολογιστική αποδοτικότητα. Κάθε χάρτης δυνατοτήτων δεύτερου επιπέδου συνδέεται με έναν χάρτη χαρακτηριστικών πρώτου επιπέδου ακριβώς (δεν εκτελείται συνδυασμός ιεραρχικών χαρακτηριστικών). Το μέγεθος του κρυφού επιπέδου, πλήρως συνδεδεμένο σε όλους τους χάρτες χαρακτηριστικών επιπέδου 2, είναι 40 μονάδες. Εκπαίδευσαν το CNN με τη λειτουργία σφάλματος διασταυρούμενης εντροπίας, χρησιμοποιώντας τον αλγόριθμο εκπαίδευσης στοχαστικής διαβάθμισης με σταθερό ρυθμό μάθησης.



Εικόνα 31

Τα νευρωνικά δίκτυα που έχουν εκπαιδευτεί με τη συνήθη λειτουργία ελαχιστοποίησης μέσου τετραγωνικού σφάλματος ή σταυρο-εντροπίας (cross-entropy) είναι επιρρεπή σε υπερ-προσαρμογή στα δεδομένα εκπαίδευσης και μπορεί να έχουν χαμηλές επιδόσεις σε ανεξάρτητα δεδομένα. Ο ταξινομητής μηχανής διανύσματος υποστήριξης (Support Vector Machine – SVM) βελτιστοποιεί μια λειτουργία σφάλματος, που ελαχιστοποιεί τις εσφαλμένες ταξινομήσεις στο σύνολο εκπαίδευσης και στον κανόνα L2  $\|w\|$  του διανύσματος βάρους  $w$  ταυτόχρονα. Γεωμετρικά, το  $\|w\|$  είναι αντιστρόφως ανάλογο προς το  $\Delta$  περιθώριο διαχωρισμού μεταξύ των σημείων εκπαίδευσης και του επιπέδου απόφασης:  $\Delta = \frac{1}{\|w\|}$ , συνεπώς, διατηρώντας  $\|w\|$  μικρά αποτελέσματα σε έναν ταξινομητή μεγάλων περιθωρίων με πιθανώς υψηλής δυνατότητας γενίκευσης. Στην πράξη, τα SVM-s χαρτογραφούν τα αρχικά διανύσματα εισόδου, σε έναν χώρο χαρακτηριστικών υψηλής διάστασης μέσω μιας μη γραμμικής χαρτογράφησης, προκειμένου να διασφαλιστεί η δυνατότητα διαχωρισμού. Στα πειράματα χρησιμοποιήθηκε ένας Γκαουσιανός πυρήνας με βελτιστοποιημένη διακύμανση. Οι προσομοιώσεις έγιναν με τη χρήση της βιβλιοθήκης Torch.

Η ιδέα της διατήρησης του  $\Delta$  διαχωριστικού περιθωρίου μεγάλου ή, ισοδύναμα, της διατήρησης  $\|w\|$  μικρά μπορούν επίσης να εφαρμοστούν σε νευρωνικά δίκτυα. Η συνήθης υλοποίηση περιλαμβάνει  $\|w\|$  στη συνάρτηση σφάλματος ως ρυθμιστής, που οδηγεί στον γνωστό αλγόριθμο απόσβεσης βάρους. Το βρήκαμε αριθμητικά πιο σταθερό να εφαρμόσουμε έναν κανόνα βάρους δεσμευμένος ρυθμιστής. Πιο συγκεκριμένα, αλλάζουμε την κλίμακα του

διανύσματος βάρους του κάθε νευρώνα κάθε φορά που η νόρμα του υπερβαίνει ένα όριο:  $w = w_{lim} \frac{w}{||w||}$  αν  $||w|| > w_{lim}$ . Στην εφαρμογή μας, η τιμή του ορίου του κανόνα βάρους,  $w_{lim}$ , εξαρτάται από το στρώμα. Η ρύθμιση  $w_{lim} = \infty$  έχει το ίδιο αποτέλεσμα με τη χρήση καθόλου ρυθμιστή.

Τα πειράματα αξιολόγησης πραγματοποιήθηκαν με τη χρήση μιας σειράς εικόνων  $30 \times 60$  εικονοστοιχείων που περιείχαν είτε πεζό «θετικά δείγματα» είτε εικόνα φόντου «αρνητικά δείγματα». Τόσο τα θετικά όσο και τα αρνητικά δείγματα συλλέχθηκαν χρησιμοποιώντας μια έγχρωμη κάμερα τοποθετημένη στην οροφή ενός αυτοκινήτου. Το αυτοκίνητο κινούνταν γύρω στο κέντρο του Τόκιο καταγράφοντας πάνω από 20 ώρες δεδομένων εικόνας. Τα δεδομένα κατανέμονται εξίσου μεταξύ των ζωνών ώρας πρωί, μεσημέρι και βράδυ (αργά το απόγευμα), ώστε να περιλαμβάνουν ποικιλία συνθηκών φωτισμού. Αν και οι παράμετροι της κάμερας είχαν ρυθμιστεί έτσι ώστε να ταιριάζουν με τις συνθήκες φωτισμού, το υψηλό δυναμικό εύρος και η γρήγορη αλλαγή των σκηνών της πραγματικής ζωής, είχαν ως αποτέλεσμα πολλές εικόνες να είναι υπερβολικές ή ελλιπείς. Αν και τα ανεπεξέργαστα δεδομένα καταγράφηκαν με τη χρήση πληροφοριών χρώματος, όλα τα πειράματα εκτελέστηκαν μετά τη μετατροπή των δεδομένων σε κλίμακα του γκρι.

Η εξαγωγή των θετικών δειγμάτων έγινε με τη σήμανση του πλαισίου οριοθέτησης των πεζών στις εικόνες. Εξετάστηκαν μόνο οι πεζοί που δεν αποκλείονταν από τα περιβαλλοντικά αντικείμενα, αλλά επιτρεπόταν η απόφραξη από μια τσάντα που μετέφερε ο πεζός. Δεν υπήρχε άλλος περιορισμός στην ορατότητα, δράση, προσανατολισμό και στάση του πεζού. Ως αποτέλεσμα, η μεταβλητότητα του συνόλου δεδομένων ήταν πολύ υψηλή και υπήρχαν πολλά παραδείγματα που κρίνονταν «δύσκολο να αναγνωριστούν» από τον ανθρώπινο παρατηρητή. Τα αρνητικά δείγματα παρήχθησαν αυτόματα με τη μέθοδο bootstrap των Sung και Poggio. Χρησιμοποιήθηκαν 200 εικόνες εισόδου για τη δημιουργία των αρνητικών δειγμάτων κατάρτισης και χρησιμοποιήθηκε διαφορετικό σύνολο 200 εικόνων για τη δημιουργία των αρνητικών δειγμάτων δοκιμής. Αυτή η μέθοδος παραγωγής αρνητικών δεδομένων, εξασφάλισε ότι ένα σχετικά μικρό σύνολο δειγμάτων περιείχε αρκετά δύσκολα παραδείγματα θα μπορούσαν να εκτελεστούν αποτελεσματικά. Δεν υπάρχει αλληλεπικάλυψη μεταξύ του χρονικού εύρους καταγραφής των δεδομένων εκπαίδευσης και δοκιμών, έτσι ώστε ο ίδιος πεζός να μην εμφανίζεται και στα δύο. Το μέγεθος των διαφόρων συνόλων δεδομένων εμφανίζεται στον πίνακα 2. Παραδείγματα θετικών και αρνητικών εικόνων παρουσιάζονται στην εικόνα 32.

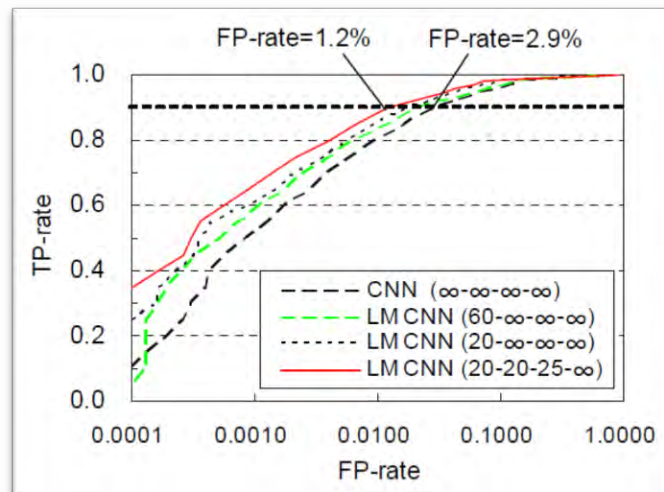
Data set	Positive samples	Negative samples
Training data	3,699	30,000
Test data	1,655	30,000

Πίνακας 2



Εικόνα 32

Αρχικά, εκπαιδεύτηκε ένα Σ.Ν.Δ. με τον τυπικό στοχαστικό αλγόριθμο οπισθοδιάδοσης χωρίς καμία νομιμοποίηση. Η καμπύλη ROC αυτού του ταξινομητή γραμμής – βάσης παρουσιάζεται ως η χαμηλότερη καμπύλη στην εικόνα 33 (υποδεικνύεται ως  $\infty - \infty - \infty - \infty$ ). Για αυτήν τη διαμόρφωση το ψευδώς θετικό ποσοστό (FP-rate) είναι 2,9% όταν το πραγματικό θετικό ποσοστό (TP-rate) είναι 90%.



Εικόνα 33

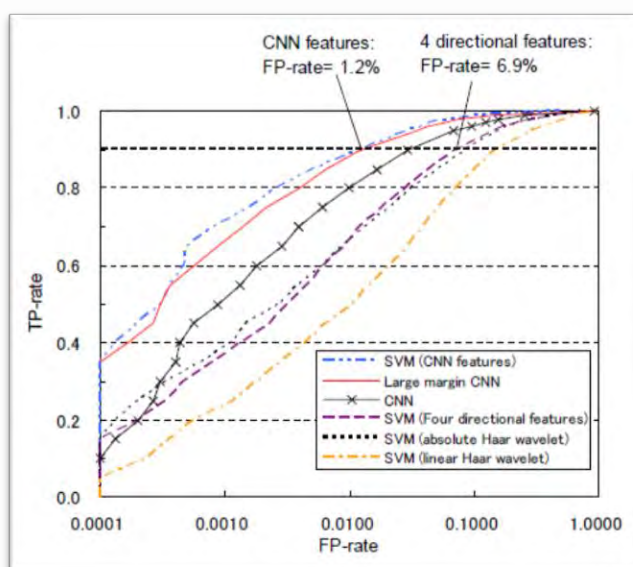
Η ακρίβεια του κατηγοριοποιητή του Σ.Ν.Δ. είναι με διαφορετικές ρυθμίσεις κανονικοποίησης. Οι ρυθμίσεις κανονικοποίησης για κάθε δίκτυο υποδεικνύονται στις γονικές θέσεις. Ο πρώτος αριθμός δηλώνει το όριο βάρους του επιπέδου εξόδου, ο δεύτερος για το κρυφό επίπεδο, ο τρίτος για τους χάρτες χαρακτηριστικών επιπέδου 2 και ο τελευταίος για τους χάρτες χαρακτηριστικών επιπέδου 1. Η ρύθμιση  $\infty$  σημαίνει ότι το δεδομένο επίπεδο δεν είναι ρυθμισμένο. Τέλος, συγκρίθηκε ο ταξινομητής Σ.Ν.Δ. με τον πιο δημοφιλή SVM (Support



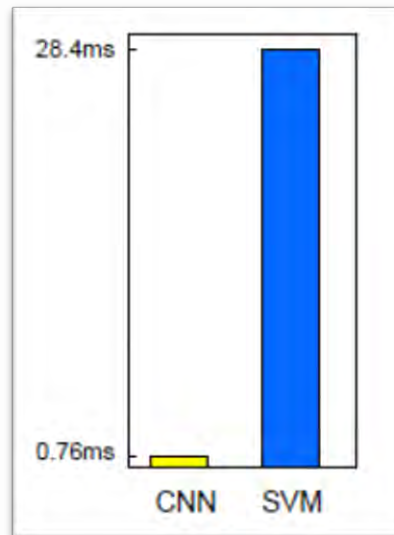
Vector Machine) σε ένα ελεγχόμενο περιβάλλον. Τα δεδομένα εκπαίδευσης και δοκιμών ήταν αυτά που περιγράφονται παραπάνω. Οι SVM-s του πυρήνα Gauss εκπαιδεύτηκαν με τη χρήση της βιβλιοθήκης Torch. Η διακύμανση του πυρήνα Gauss και η ανταλλαγή παραμέτρων μεταξύ του σφάλματος κατάρτισης και του περιθωρίου βελτιστοποιήθηκε ξεχωριστά για κάθε πείραμα. Ο αριθμός των διανυσμάτων υποστήριξης που προέκυψαν ήταν στην περιοχή των 5.000 και δεν διέφερε σημαντικά μεταξύ των διαφορετικών ρυθμίσεων των παραμέτρων.

**Ακρίβεια:** Αρχικά, αξιολογήθηκε η ακρίβεια του SVM χρησιμοποιώντας γραμμικά χαρακτηριστικά κυμάτων Χάαρ, απόλυτα χαρακτηριστικά κυμάτων Χάαρ και τέσσερα χαρακτηριστικά κατεύθυνσης. Όπως μπορούμε να δούμε από τις καμπύλες ROC που εμφανίζονται στην εικόνα 34, τόσο τα απόλυτα κυματικά κύματα Χάαρ όσο και τα τέσσερα χαρακτηριστικά κατεύθυνσης (που παίρνουν επίσης την απόλυτη τιμή των εξόδων φίλτρου) δίνουν πολύ καλύτερη ακρίβεια, από τους γραμμικούς συντελεστές κυματοειδών. Δεν υπάρχει μεγάλη διαφορά μεταξύ των τεσσάρων χαρακτηριστικών κατεύθυνσης και των απόλυτων κυμάτων Χάαρ. Ωστόσο, υπάρχει μια μεγάλη διαφορά μεταξύ της ακρίβειας αυτών των ταξινομητών SVM και αυτής του Σ.Ν.Δ.. Για παράδειγμα, το ψευδώς θετικό ποσοστό του SVM είναι 6,9%, ενώ του Σ.Ν.Δ. είναι μόνο 1,2% όταν το ποσοστό ανίχνευσης είναι 90%. Πιστεύουμε ότι αυτή η μεγάλη διαφορά στην ακρίβεια προκαλείται από τη διαφορά των χαρακτηριστικών. Προκειμένου να επιβεβαιωθεί αυτήν την υπόθεση, εκπαιδεύτηκε επίσης ένα SVM χρησιμοποιώντας τα χαρακτηριστικά που υπολογίστηκαν από το Σ.Ν.Δ.. Όπως μπορούμε να δούμε στην εικόνα 34 το SVM που χρησιμοποιεί τα χαρακτηριστικά Σ.Ν.Δ. έχει την υψηλότερη ακρίβεια, επιβεβαιώνοντας την υπόθεση ότι η υψηλή ακρίβεια του. Ο ταξινομητής CNN ήταν το αποτέλεσμα των δυνατοτήτων που έχουν βελτιστοποιηθεί για την εργασία.

**Χρόνος υπολογισμού:** Ο χρόνος που απαιτείται από το Σ.Ν.Δ. και από τον ταξινομητή SVM για την επεξεργασία μιας υποψήφιας εικόνας, συμπεριλαμβανομένου του υπολογισμού χαρακτηριστικών, εμφανίζεται στην εικόνα 35. Μπορούμε να δούμε ότι η SVM χρησιμοποιεί σχεδόν 40 φορές περισσότερο χρόνο από το Σ.Ν.Δ.. Αυτή η μεγάλη διαφορά προκαλείται από τη διαφορά στο μέγεθος των μοντέλων. Το Σ.Ν.Δ. έχει μόνο 40 πλήρως συνδεδεμένους κρυφούς κόμβους, αλλά το SVM έχει περισσότερα από 5000 διανύσματα υποστήριξης. Δεδομένου ότι ο υπολογισμός για έναν κρυφό κόμβο στο Σ.Ν.Δ. και ένα διάνυσμα υποστήριξης στο SVM είναι περίπου το ίδιο, ο λόγος των χρόνων υπολογισμού καθορίζεται από το λόγο των διανυσμάτων υποστήριξης και των κρυφών κόμβων.



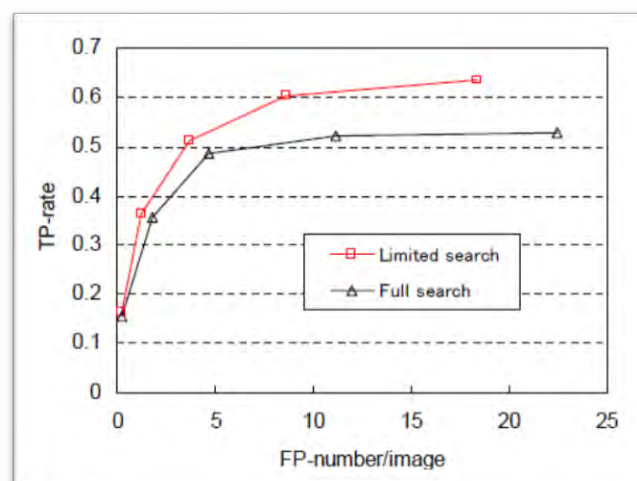
Εικόνα 34



Εικόνα 35

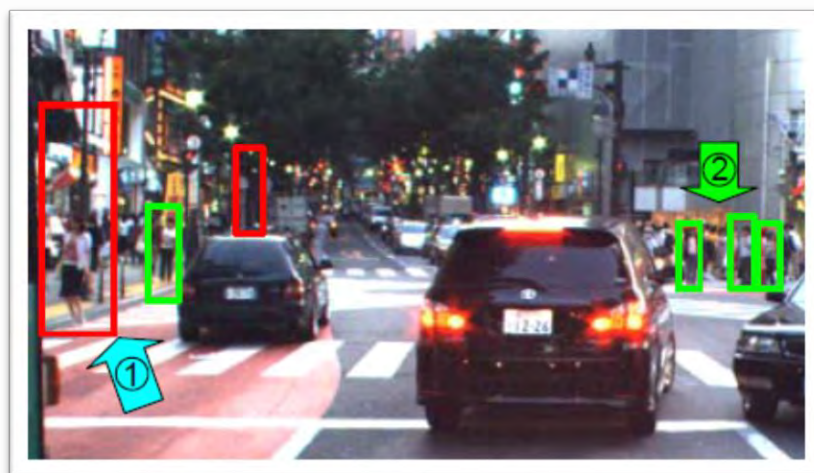
Η πλήρης αξιολόγηση του συστήματος πραγματοποιήθηκε με τη χρήση εικόνων  $50 \times 1388 \times 1038$  εικονοστοιχείων που επιλέχθηκαν ομοιόμορφα από το σύνολο δεδομένων που περιγράφεται παραπάνω. Οι εικόνες έχουν επεξεργαστεί με το σύστημα ανίχνευσης, με μεγάλο περιθώριο που βασίζεται στο Σ.Ν.Δ. και οι ανιχνευμένοι πεζοί αποθηκεύτηκαν σε ένα αρχείο. Έτσι, αυτές οι ανιχνεύσεις συγκρίνονται αυτόματα με τις μη αυτόματες αναφορές. Αυτή η αυτοματοποιημένη διαδικασία, εξασφαλίζει την επαναληψιμότητα των αποτελεσμάτων και επιτρέπει την αποτελεσματική διεξαγωγή της αξιολόγησης. Ένα αποτέλεσμα ανίχνευσης θεωρείται σωστό, αν το ύψος και η θέση ήταν εντός ενός περιθωρίου ανοχής για μία από τις αναφορές. Η ανοχή για το ύψος και την κατακόρυφη θέση του πλαισίου ανίχνευσης ήταν  $\pm 30\%$  και η ανοχή για την οριζόντια θέση του πλαισίου ανίχνευσης ήταν  $\pm 40\%$ .

**Αποτελέσματα αξιολόγησης:** Η αξιολόγηση διεξήχθη χρησιμοποιώντας δύο διαφορετικές ρυθμίσεις για τη διαδικασία αναζήτησης. Στη βασική ρύθμιση, πραγματοποιήσαμε την αναζήτηση για την «πλήρη εικόνα» (full search). Στην άλλη περίπτωση, η αναζήτηση στον κάθετο άξονα περιορίστηκε σε μια ζώνη πλάτους 200 pixel «περιορισμένη αναζήτηση» (limited search), με βάση εκτιμήσεις για τα πιθανά ύψη των πεζών και τη γεωμετρία της κάμερας. Οι καμπύλες ROC για τις δύο περιπτώσεις εμφανίζονται στην εικόνα 36. Οι δύο καμπύλες έχουν παραχθεί χρησιμοποιώντας τα ίδια μοντέλα και τα ίδια όρια ανίχνευσης.

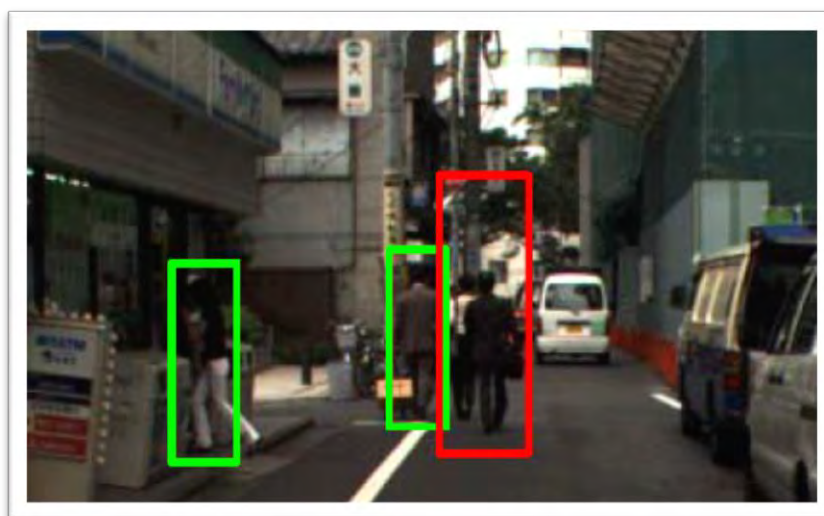


Εικόνα 36

Η εισαγωγή του περιορισμού της περιοχής αναζήτησης μείωσε τον αριθμό των ψευδών θετικών, δεδομένου ότι κάθε ένα από τα 5 αξιολογημένα σημεία μετατοπίστηκε προς τα αριστερά. Ωστόσο, η εισαγωγή του περιορισμού της περιοχής αναζήτησης όχι μόνο μείωσε τον αριθμό των ψευδών θετικών, αλλά αύξησε και τον αριθμό των ανιχνεύσεων, παρόλο που ακριβώς τα ίδια όρια έχουν χρησιμοποιηθεί και για τις δύο καμπύλες. Αναλύοντας την αιτία αυτού του φαινομένου, βρήκαμε ότι υπάρχουν πολλά ψευδή θετικά που επικαλύπτουν έναν πεζό, αλλά το μέγεθος του πλαισίου ανίχνευσης είναι εκτός του εύρους ανοχής. Λόγω του αλγόριθμου συγχώνευσης πολλαπλής ανίχνευσης, τέτοια ψευδή θετικά στοιχεία μπορεί να αποτρέψουν την ανίχνευση ενός πεζού. Αν η απόδοση του κατηγοριοποιητή είναι υψηλότερη για τα ψευδή θετικά απ' ό,τι για τον πεζό που επικαλύπτει, ο πεζός δεν μπορεί να ανιχνευθεί ακόμη και αν το όριο μειωθεί στο 0. Δύο τέτοια παραδείγματα παρουσιάζονται στις εικόνες 37 και 38. Η εξάλειψη αυτού του τύπου ψευδών θετικών αποτελεσμάτων από τον περιορισμό της περιοχής αναζήτησης, καθιστά δυνατή την ανίχνευση του πεζού που επικαλύπτονται. Ο λόγος για τον οποίο ο ταξινομητής δεν είναι ισχυρός έναντι αυτού του τύπου σφάλματος, είναι ότι τα αρνητικά δεδομένα εκπαίδευσης έχουν παραχθεί χρησιμοποιώντας εικόνες που δεν περιείχαν καθόλου πεζούς. Η καταστολή των ορθών ανιχνεύσεων με την επικάλυψη ψευδώς θετικών αποτελεσμάτων, εξηγεί επίσης γιατί οι καμπύλες ROC παρουσιάζουν κορεσμό, με ρυθμό ανίχνευσης μικρότερο από 1.



Εικόνα 37



Εικόνα 38

Συμπερασματικά, σε αυτήν την εργασία προτάθηκε και αξιολογήθηκε η χρήση του συμβατικού νευρωνικού δικτύου CNN (Convolutional Neural Network) ως ταξινομητή σε συστήματα ανίχνευσης πεζών. Το πιο σημαντικό συμπέρασμα από τα πειραματικά αποτελέσματα, είναι ότι τα καλά χαρακτηριστικά που βελτιστοποιούνται αυτόματα στο έργο ανίχνευσης είναι αναπόφευκτα για ανίχνευση υψηλής ακρίβειας. Ο ψευδώς θετικός ρυθμός (FP) του Σ.Ν.Δ. ήταν μικρότερος από το  $1/5^{ου}$  του ρυθμού FP του τυπικού συνδυασμού Haar wavelet + SVM. Ωστόσο, η ακρίβεια του ταξινομητή SVM ξεπέρασε την ακρίβεια του βασικού Σ.Ν.Δ. όταν μπορούσε να χρησιμοποιήσει τα ίδια χαρακτηριστικά, λόγω της χρήσης μεγάλης εκπαίδευσης περιθωρίου που είναι εγγενής στο SVM. Όταν το Σ.Ν.Δ. κανονικοποιήθηκε κατά τη διάρκεια της εκπαίδευσης, προκειμένου να αποκτήσει ένα μεγάλο όριο απόφασης, το ψευδώς θετικό (FP) ποσοστό μειώθηκε στο μισό του βασικού ποσοστού FP, σχεδόν να γίνει ισοδύναμο με το συνδυασμό CNN-features + SVM. Εκτός από την ικανότητά του να μαθαίνει τα βέλτιστα χαρακτηριστικά, ένα άλλο μεγάλο πλεονέκτημα του Σ.Ν.Δ. είναι η πολύ χαμηλή υπολογιστική πολυπλοκότητα του. Αν και η υλοποίησή του στο project απέχει πολύ από τη βέλτιστη, ο χρόνος για την επεξεργασία ενός υποψηφίου πεζού με το Σ.Ν.Δ. ήταν λιγότερο από το 3% του χρόνου που απαιτείται από το SVM. Επειδή το συνελκτικό νευρωνικό δίκτυο παρέχει την υψηλότερη ακρίβεια και χρειάζεται σημαντικά λιγότερο υπολογισμό, πιστεύεται ότι είναι ο καταλληλότερος ταξινομητής για την ανίχνευση πεζών, από το δημοφιλές συνδυασμό «Hand design feature» + SVM.

Ολοκληρώνοντας το παρόν κεφάλαιο, είναι σημαντικό να αναφέρουμε πως σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας, κάθε χρόνο περίπου 1.35 εκατομμύρια συνάνθρωποί μας χάνουν τη ζωή τους σε τροχαία δυστυχήματα. Επίσης 20 με 50 εκατομμύρια άνθρωποι υποφέρουν από σοβαρούς τραυματισμούς και πάρα πολύ εξ αυτών, μένουν ανάπηροι ως αποτέλεσμα του τραυματισμού τους. Επίσης, τα τροχαία δυστυχήματα θανατηφόρα ή μη, με σοβαρούς τραυματισμούς και αναπηρίες, έχουν σοβαρό αντίκτυπο στην οικονομία, όχι μόνο των θυμάτων και των οικογενειών αλλά και ολόκληρης της χώρας. Όταν ένας άνθρωπος πέσει θύμα τροχαίου δυστυχήματος, οι οικονομικές απώλειες για την οικογένειά του είναι τεράστιες, καθώς θα κληθούν να εγκαταλείψουν την εργασία τους για να τον φροντίσουν. Σε εθνικό επίπεδο, τα τροχαία δυστυχήματα επιφέρουν τεράστιες οικονομικές απώλειες για ένα κράτος από τα νοσήλια, τις αποζημιώσεις και την μείωση της παραγωγικότητας. Υπολογίζεται δε, πως το κόστος για κάθε κράτος ανέρχεται στο 3% του εγχώριου ακαθάριστου προϊόντος.

Είναι σημαντικό λοιπόν, να αναφέρουμε τη σπουδαιότητα της χρήσης ενός τέτοιου ευφυούς συστήματος. Ενός συστήματος, που δεν θα διαλευκάνει μόνο μία υπόθεση τροχαίου ατυχήματος, αλλά θα προειδοποιεί έγκαιρα τον οδηγό σε περίπτωση που υπάρχει κίνδυνος ατυχήματος. Βέβαια, ακόμη δεν έχουν διευκρινιστεί αρκετά θέματα σχετικά με τη νομοθεσία περί προσωπικών δεδομένων (GDPR). Η Αρχή Προστασίας Δεδομένων Προσωπικού Χαρακτήρα έχει δεχτεί πλήθος ερωτημάτων αναφορικά με το θέμα της χρήσης των καμερών στα αυτοκίνητα. Σε κάποιες χώρες όπως η Ιρλανδία, η Αρχή Προστασίας Δεδομένων Προσωπικού Χαρακτήρα παρέχει στους κατόχους συστημάτων Dashcam καθοδήγηση σχετικά με τη χρήση του. Ολοκληρώνοντας, η σωστή εκπαίδευση αναφορικά με την οδική παιδεία από τη σχολική ηλικία, η υπευθυνότητα στην οδική συμπεριφορά απέναντι στο κοινωνικό σύνολο και τα τελευταίας τεχνολογίας ευφυή συστήματα, είναι αρκετά για να βελτιώσουμε τα αρνητικά νούμερα σχετικά με τους θανάτους και τους βαρύς τραυματισμούς από τα τροχαία ατυχήματα.

## 4 Προτεινόμενο σύστημα

### 4.1 Εισαγωγή

Όπως είπαμε και στο προηγούμενο κεφάλαιο, η αναγνώριση προτύπων (Pattern Recognition) είναι ένα επιστημονικό πεδίο με στόχο, την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής ή διακριτικού στοιχείου σε εισαγόμενα δεδομένα, συνήθως κωδικοποιημένα ως αλληλουχίες αριθμών. Σ' αυτό το κεφάλαιο θα αναλύσουμε τον τρόπο κατασκευής και θα αναπτύξουμε ένα σύστημα τέτοιο, ώστε να μπορεί να αναγνωρίσει ανθρώπους, οχήματα, πινακίδες κυκλοφορίας και φωτεινούς σηματοδότες. Η βάση τους συστήματος θα είναι η πλατφόρμα Raspberry Pi 4 B, όπου με τον κατάλληλο προγραμματισμό θα μπορεί να εκτελέσει τη διαδικασία αναγνώρισης προτύπων σε φωτογραφίες.

Η αναγνώριση προτύπων βρίσκει εφαρμογή σε πολλές επιστήμες όπως η ιατρική και η βιοπληροφορική (ανάλυση δεδομένων DNA), πληροφορική (εξόρυξη δεδομένων), βιομετρικά συστήματα και συστήματα ασφαλείας (αναγνώριση προσώπων και βιομετρικών δεδομένων), στη βιομηχανία και στις παραγωγικές μονάδες κ.α. .

### 4.2 Υλικοτεχνικός εξοπλισμός – Ανάλυση

Η πλατφόρμα Raspberry Pi 4B θεωρήθηκε η πλέον κατάλληλη για την πραγματοποίηση του εν λόγω project. Στο σύνολο των υλικών που χρησιμοποιήσαμε ανήκουν η πλακέτα – μίνι υπολογιστής Raspberry Pi 4B (εικόνα 39), μία οθόνη LCD touch 7" (εικόνα 40), ένας AC Adaptor (εικόνα 41) με output 5.1V 3.0A DC μέσω του οποίου τροφοδοτείται το Raspberry Pi, ένα σετ πληκτρολογίου-mouse και μία πλαστική θήκη (εικόνα 42) για την προστασία της πλακέτας. Επίσης χρησιμοποιήθηκε μία κάρτα μνήμης micro SD χωρητικότητας 64 GB (εικόνα 43), που λειτουργεί ως σκληρό δίσκος για την αποθήκευση του λειτουργικού Raspberry Pi OS, των λοιπών λογισμικών που χρησιμοποιήθηκαν καθώς και των φωτογραφιών.

Τα Raspberry Pi είναι μια σειρά από μικρούς single-board υπολογιστές (SBCs), που αναπτύχθηκαν στο Ηνωμένο Βασίλειο από το Raspberry Pi Foundation σε συνεργασία με την Broadcom . Το project Raspberry Pi αρχικά, έγειρε προς την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών στα σχολεία και στις αναπτυσσόμενες χώρες. Το αρχικό μοντέλο έγινε πιο δημοφιλές από ό,τι αναμενόταν. Χρησιμοποιείται ευρέως σε πολλούς τομείς, όπως για την παρακολούθηση των καιρικών συνθηκών, λόγω του χαμηλού κόστους της, το διαμορφώσιμο και ανοικτό σχέδιο. Συνήθως, χρησιμοποιείται από χομπίστες που ασχολούνται με υπολογιστές και ηλεκτρονικά, λόγω του ότι περιέχει τα πρότυπα HDMI και USB. Μετά την κυκλοφορία του δεύτερου τύπου πλακέτας, το Ίδρυμα Raspberry Pi δημιουργήσει μια νέα οντότητα που ονομάζεται Raspberry Pi Trading. Τα περισσότερα Pis κατασκευάζονται σε ένα εργοστάσιο της Sony στο Pencoeed της Ουαλίας, ενώ άλλα κατασκευάζονται σε εργοστάσια της Κίνας και της Ιαπωνίας.





Εικόνα 39



Εικόνα 40

Υπάρχουν τρεις σειρές Raspberry Pi, και έχουν κυκλοφορήσει αρκετές γενιές από κάθε μία. Τα SBCs Του Raspberry Pi διαθέτουν σύστημα Broadcom σε τσιπ (SoC) με ενσωματωμένη κεντρική μονάδα επεξεργασίας (CPU) συμβατή με ARM και μονάδα επεξεργασίας γραφικών on-chip (GPU), ενώ το Raspberry Pi Pico διαθέτει σύστημα RP2040 σε τσιπ με ενσωματωμένη κεντρική μονάδα επεξεργασίας (CPU) συμβατή με ARM.

Το Raspberry Pi Model B πρώτης γενιάς κυκλοφόρησε τον Φεβρουάριο του 2012, ακολουθούμενο από το απλούστερο και φθηνότερο Model A. Το 2014, το Ίδρυμα κυκλοφόρησε έναν πίνακα με βελτιωμένο σχεδιασμό, το Raspberry Pi Model B+. Αυτές οι πλακέτες πρώτης γενιάς διαθέτουν επεξεργαστές ARM11, έχουν μέγεθος περίπου πιστωτικής κάρτας και αντιπροσωπεύουν τον τυπικό συντελεστή μορφής κύριας γραμμής. Τα βελτιωμένα μοντέλα A+ και B κυκλοφόρησαν μέσα σε ένα χρόνο. Μια "Υπολογιστική μονάδα" κυκλοφόρησε τον Απρίλιο του 2014 για ενσωματωμένες εφαρμογές. Το Raspberry Pi 2 κυκλοφόρησε τον Φεβρουάριο του 2015 και αρχικά διέθετε τετραπύρρηνο επεξεργαστή ARM Cortex-A7 900 MHz 32-bit με μνήμη RAM 1 GB. Η αναθεώρηση 1.2 διέθετε τετραπύρρηνο επεξεργαστή ARM Cortex-A53 900 MHz 64-bit (τον ίδιο με αυτόν του Raspberry Pi 3 Model B, αλλά υποχρονισμένο στα 900 MHz). Το Raspberry Pi 3 Model B κυκλοφόρησε τον Φεβρουάριο του 2016 με τετραπύρρηνο επεξεργαστή ARM Cortex-A53 1,2 GHz 64-bit, ενσωματωμένες δυνατότητες εκκίνησης Wi-Fi, Bluetooth και USB 802.11n. Στην «Ημέρα Pi» 2018, το Raspberry Pi 3 Model B+ κυκλοφόρησε με ταχύτερο επεξεργαστή 1,4 GHz, τρεις φορές ταχύτερο gigabit Ethernet (απόδοση που περιορίζεται σε περίπου 300 Mbit/s από την εσωτερική σύνδεση USB 2.0) και Wi-Fi διπλής ζώνης 2,4 / 5,0 GHz 802,11ac (100 Mbit/s). Άλλα χαρακτηριστικά είναι το Power over Ethernet (PoE) (με το πρόσθετο PoE HAT), η εκκίνηση USB και δικτύου όπου δεν απαιτείται πλέον κάρτα SD.



Εικόνα 41



Εικόνα 42



Εικόνα 43

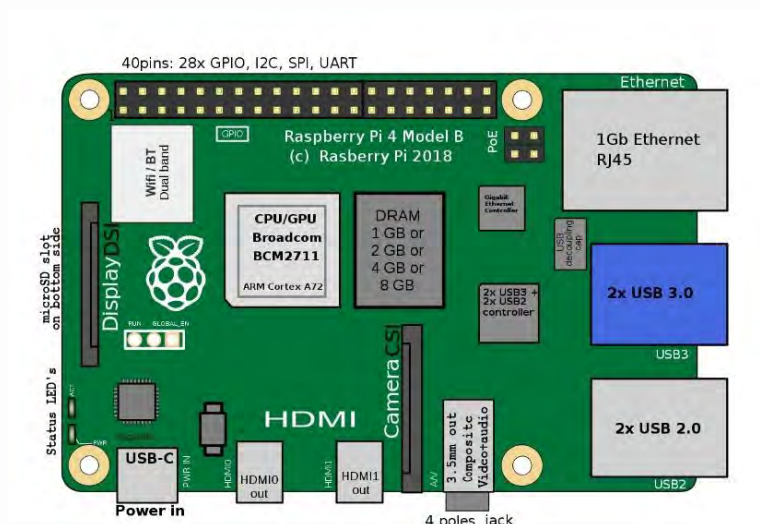
Το Raspberry Pi 4 Model B κυκλοφόρησε τον Ιούνιο του 2019 με τετραπύρρνο επεξεργαστή ARM Cortex-A72 1,5 GHz 64 bit, ενσωματωμένο Wi-Fi 802.11ac, Bluetooth 5, πλήρες gigabit Ethernet (η απόδοση δεν περιορίζεται), δύο θύρες USB 2.0, δύο θύρες USB 3.0, μνήμη RAM 1-8 GB και υποστήριξη διπλής οθόνης μέσω ενός ζεύγους θυρών micro HDMI (HDMI Type D) για ανάλυση έως και 4K. Η έκδοση με 1 GB RAM έχει εγκαταλειφθεί και οι τιμές της έκδοσης 2 GB έχουν μειωθεί. Η έκδοση 8 GB διαθέτει αναθεωρημένη πλακέτα κυκλώματος. Το Pi 4 τροφοδοτείται επίσης μέσω θύρας USB-C, επιτρέποντας την παροχή πρόσθετης ισχύος στα περιφερειακά, όταν χρησιμοποιείται με κατάλληλο τροφοδοτικό. Το Pi μπορεί να λειτουργήσει μόνο με τάση 5V και όχι 9V ή 12V όπως άλλοι μίνι υπολογιστές αυτής της κατηγορίας. Η αρχική πλακέτα Raspberry Pi 4 έχει ένα σχεδιαστικό ελάττωμα όπου τα καλώδια USB με ηλεκτρονική σήμανση τρίτου μέρους, όπως αυτά που χρησιμοποιούνται στα Apple MacBooks, το αναγνωρίζουν εσφαλμένα και αρνούνται να παρέχουν ισχύ. Το Tom's Hardware δοκίμασε 14 διαφορετικά καλώδια και διαπίστωσε ότι 11 από αυτά ενεργοποιήθηκαν και τροφοδότησαν το Pi χωρίς πρόβλημα. Το ελάττωμα σχεδιασμού διορθώθηκε στην αναθεώρηση 1.2 του πίνακα, που κυκλοφόρησε στα τέλη του 2019. Στα μέσα του 2021, εμφανίστηκαν μοντέλα Pi 4 B με το βελτιωμένο Broadcom BCM2711C0. Ο κατασκευαστής χρησιμοποιεί τώρα αυτό το τσιπ για τα Pi 4 B και Pi 400. Ωστόσο, η συχνότητα πρόσφυσης του Pi 4 B δεν αυξήθηκε στο εργοστάσιο.

Το Raspberry Pi 400 κυκλοφόρησε τον Νοέμβριο του 2020. Ένα σύγχρονο παράδειγμα υπολογιστή ηλεκτρολογίου, διαθέτει 4 GB μνήμης RAM LPDDR4 σε μια προσαρμοσμένη πλακέτα που προέρχεται από το υπάρχον Raspberry Pi 4 σε συνδυασμό με ένα ηλεκτρολόγιο σε μία μόνο περίπτωση. Η υπόθεση προήλθε από αυτή του ηλεκτρολογίου Raspberry Pi. Μια ισχυρή λύση ψύξης (δηλαδή μια ευρεία μεταλλική πλάκα) και ένα αναβαθμισμένο τροφοδοτικό μεταγωγής επιτρέπουν στον επεξεργαστή Broadcom BCM2711C0 του Raspberry Pi 400 να



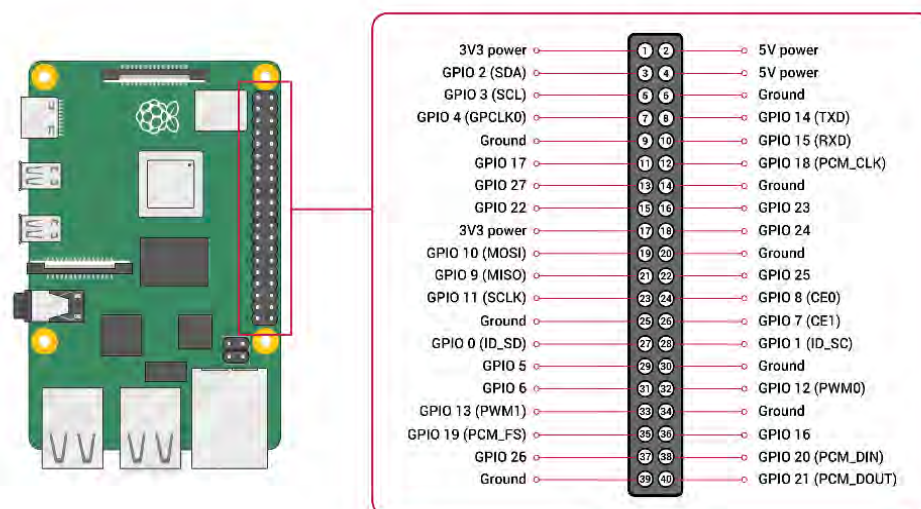
χρονομετρηθεί στα 1,8 GHz, ο οποίος είναι 20% ταχύτερος από τον επεξεργαστή Raspberry Pi 4 στον οποίο βασίζεται.

Επίσης, η Raspberry Pi έχει κυκλοφορήσει και τα μοντέλα Raspberry Pi Zero και Raspberry Pi Pico. Το Raspberry Pi Zero κυκλοφόρησε τον Νοέμβριο του 2015, με μικρότερο μέγεθος και μειωμένες δυνατότητες εισόδου/εξόδου (I/O) και γενικής χρήσης εισόδου/εξόδου (GPIO). Έχουν κυκλοφορήσει οι εκδόσεις Raspberry Pi v1.3 (2016), W (2017), WH (2018) και Zero 2 W (2021). Το Raspberry Pi Pico κυκλοφόρησε τον Ιανουάριο του 2021. Ήταν η πρώτη πλακέτα του Raspberry Pi που βασίστηκε σε ένα μόνο τσιπ μικροελεγκτή, το RP2040, το οποίο σχεδιάστηκε από την Raspberry Pi στο Ηνωμένο Βασίλειο. Το Pico έχει 264 KB μνήμης RAM και 2 MB μνήμης flash. Είναι προγραμματιζόμενο σε Micro Python, Circuit Python, C και Rust. Το Ίδρυμα Raspberry Pi συνεργάστηκε με τους Adafruit, Pimoroni, Arduino και SparkFun για την κατασκευή αξεσουάρ για το Raspberry Pi Pico και διάφορες άλλες σανίδες χρησιμοποιώντας την πλατφόρμα πυριτίου RP2040. Αντί να εκτελεί το ρόλο του υπολογιστή γενικής χρήσης, έχει σχεδιαστεί για φυσικούς υπολογιστές, με παρόμοια έννοια όπως ένα Arduino. Στις 30 Ιουνίου 2022, κυκλοφόρησε το Raspberry Pi Pico W, μια έκδοση του Pico με δυνατότητα Wi-Fi 802.11n. Το ασύρματο τσιπ CYW43439 στο Pico W υποστηρίζει επίσης Bluetooth, αλλά η δυνατότητα δεν ήταν ενεργοποιημένη κατά την εκκίνηση.



Εικόνα 44

Στην εικόνα 44 φαίνεται μία τυπική διάταξη της πλατφόρμας Raspberry Pi 4B, με ανάλυση των επιμέρους στοιχείων που έχει. Στο κέντρο βλέπουμε τη θέση της CPU/GPU και τη θέση της μνήμης RAM. Δεξιά, υπάρχουν οι θύρες USB 2.0 και 3.0 και μία θύρα Ethernet RJ45. Στο κάτω μέρος βλέπουμε τη θύρα USB-C στην οποία συνδέεται η παροχή ρεύματος, δύο θύρες micro HDMI, μία σειριακή θύρα κάμερας CSI (Camera Serial Interface) και μία έξοδο 4 πόλων Jack. Στην αριστερή πλευρά υπάρχει μία σειριακή θύρα οθόνης DSI (Display Serial Interface). Τέλος, στην πάνω πλευρά διακρίνονται το Dual Band 2.4/5.0 GHz WiFi και το τμήμα με 40 pins (ακίδες) εκ των οποίων 28 GPIO (general-purpose input/output), I2C, SPI και UART.



Εικόνα 45

Ένα ισχυρό χαρακτηριστικό του Raspberry Pi είναι η σειρά των ακίδων GPIO (είσοδος / έξοδος γενικής χρήσης) κατά μήκος της επάνω άκρης της πλακέτας. Μια κεφαλή GPIO 40 ακίδων βρίσκεται σε όλες τις τρέχουσες πλακέτες Raspberry Pi (όχι τόσο πυκνή στα Raspberry Pi Zero, Raspberry Pi Zero W και Raspberry Pi Zero 2 W). Πριν από το Raspberry Pi 1 Model B+ (2014), οι πίνακες περιλάμβαναν μια μικρότερη κεφαλή 26 ακίδων. Η κεφαλή GPIO σε όλους τους πίνακες (συμπεριλαμβανομένου του Raspberry Pi 400) έχει βήμα καρφίτσας 0,1 " (2,54 mm). Οποιοσδήποτε από τους ακροδέκτες GPIO μπορεί να οριστεί (σε λογισμικό) ως ακροδέκτης εισόδου ή εξόδου και να χρησιμοποιηθεί για ένα ευρύ φάσμα σκοπών. Στην εικόνα 45, φαίνονται σχηματικά τα χαρακτηριστικά των ακίδων GPIO του Raspberry Pi 4B.

Δύο ακίδες 5V και δύο ακίδες 3.3V υπάρχουν στον πίνακα, καθώς και ένας αριθμός ακίδων γείωσης (0V), οι οποίοι δεν μπορούν να ρυθμιστούν. Οι υπόλοιπες ακίδες είναι όλες ακίδες γενικής χρήσης 3.3V, που σημαίνει ότι οι εξόδοι και οι εισόδοι έχουν ρυθμιστεί σε 3.3V. Ένας ακροδέκτης GPIO που έχει οριστεί ως ακροδέκτης εξόδου, μπορεί να ρυθμιστεί σε υψηλό (3.3V) ή χαμηλό (0V). Αυτό γίνεται ευκολότερο, με τη χρήση εσωτερικών αντιστάσεων pull-up ή pull down. Οι ακίδες GPIO2 και GPIO3 έχουν σταθερές αντιστάσεις pull-up, αλλά για άλλες ακίδες αυτό μπορεί να ρυθμιστεί μέσω λογισμικού.

Εκτός από τις απλές συσκευές εισόδου και εξόδου, οι ακίδες GPIO μπορούν να χρησιμοποιηθούν με μια ποικιλία εναλλακτικών λειτουργιών, μερικές είναι διαθέσιμες σε όλες τις ακίδες, άλλες σε συγκεκριμένες ακίδες.

#### PWM (pulse-width modulation)

- Το λογισμικό PWM διαθέσιμο σε όλες τις ακίδες
- Υλικό PWM διαθέσιμο σε GPIO12, GPIO13, GPIO18, GPIO19

#### SPI

- SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
- SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

## I2C

- Data: (GPIO2); Clock (GPIO3)
- EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

## Serial

- ❖ TX (GPIO14); RX (GPIO15)

Εύκολη πρόσβαση μπορεί να αποκτηθεί στο Raspberry Pi GPIO, ανοίγοντας ένα παράθυρο τερματικού και εκτελώντας το **pinout** όπως φαίνεται στην εικόνα 46. Αυτό το εργαλείο παρέχεται από τη βιβλιοθήκη GPIO Zero Python, η οποία είναι εγκατεστημένη από προεπιλεγμένη εικόνα της επιφάνειας εργασίας του Raspberry Pi OS, αλλά όχι στο Raspberry Pi OS Lite.

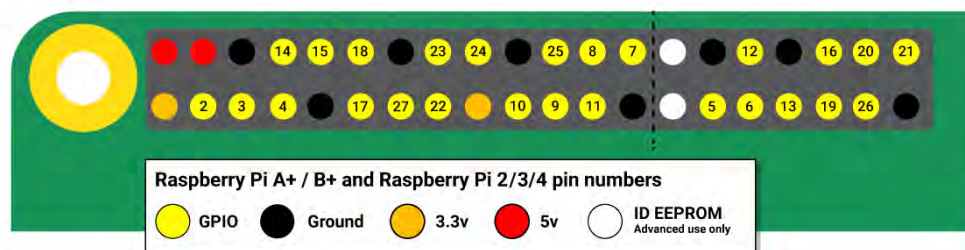


```
pi@raspberrypi:~$ pinout
Pi Model 3B V1.2
Revision : a02082
SoC      : BCM2837
RAM      : 1024Mb
Storage  : MicroSD
USB ports : 4 (excluding power)
Ethernet ports : 1
Wi-fi    : True
Bluetooth : True
Camera ports (CSI) : 1
Display ports (DSI) : 1

J8:
3V3 (1) (2) 5V
GPI02 (3) (4) 5V
GPI03 (5) (6) GND
GPI04 (7) (8) GPI014
GND (9) (10) GPI015
GPI017 (11) (12) GPI018
GPI027 (13) (14) GND
GPI022 (15) (16) GPI023
3V3 (17) (18) GPI024
GPI010 (19) (20) GND
GPI009 (21) (22) GPI025
GPI011 (23) (24) GPI08
GND (25) (26) GPI07
GPI00 (27) (28) GPI01
GPI05 (29) (30) GND
GPI06 (31) (32) GPI012
GPI013 (33) (34) GND
GPI019 (35) (36) GPI016
GPI026 (37) (38) GPI020
GND (39) (40) GPI021

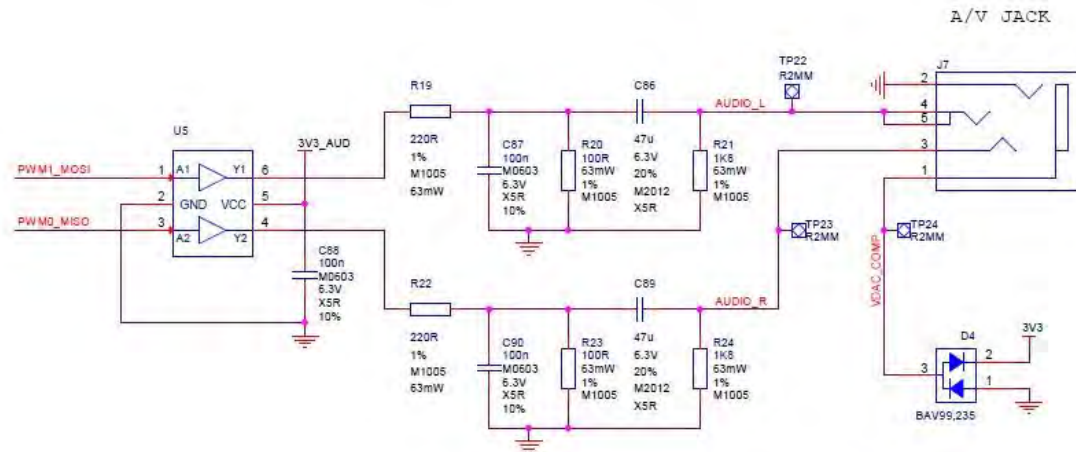
For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~$
```

Εικόνα 46

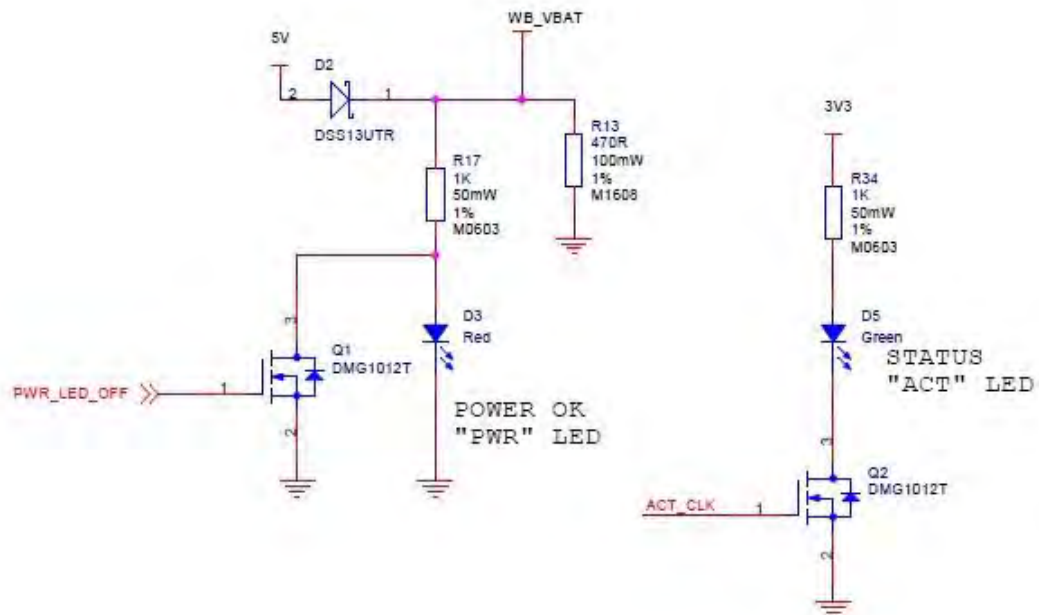


Εικόνα 47

Στην εικόνα 47, φαίνεται η χρωματική αναπαράσταση των pins όπου κατηγοριοποιούνται βάση των ιδιοτήτων τους. Τέλος, στις παρακάτω εικόνες παρατίθενται τα σχέδια των κυκλωμάτων. Τα σχέδια (εικόνες 48.a – 48.k), δημιουργήθηκαν από τον James Adams στις 18 Ιουνίου 2019 για λογαριασμό της Raspberry Pi.

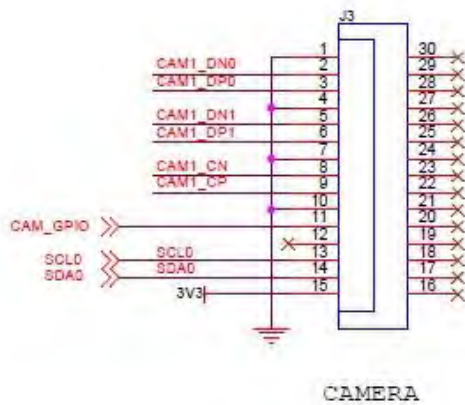


Εικόνα 48.a

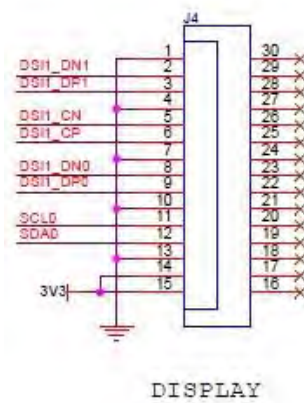


Εικόνα 48.b

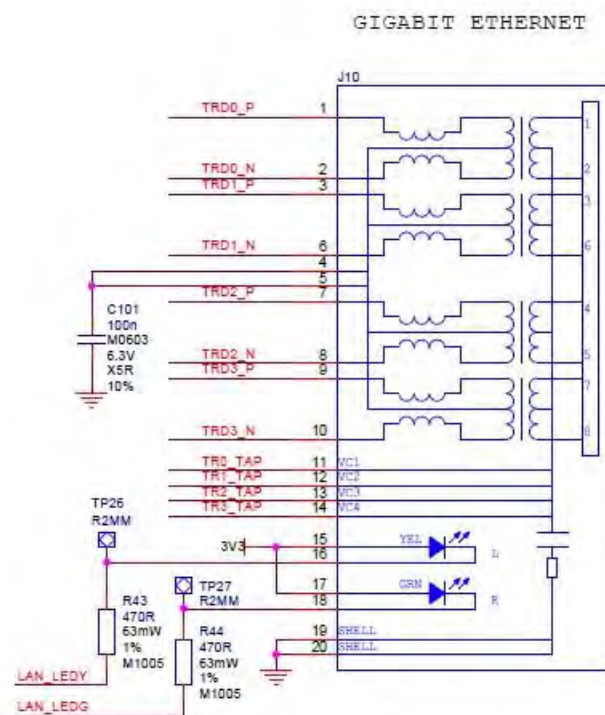




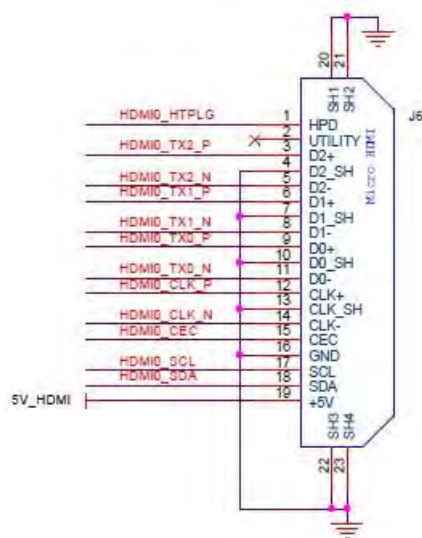
Εικόνα 48.c



Εικόνα 48.d

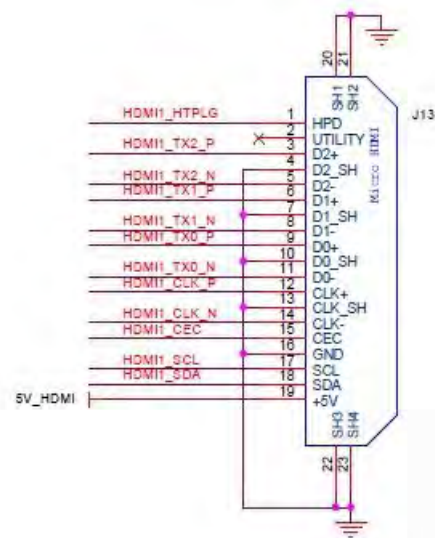


Εικόνα 48.e



MICRO HDMI 0

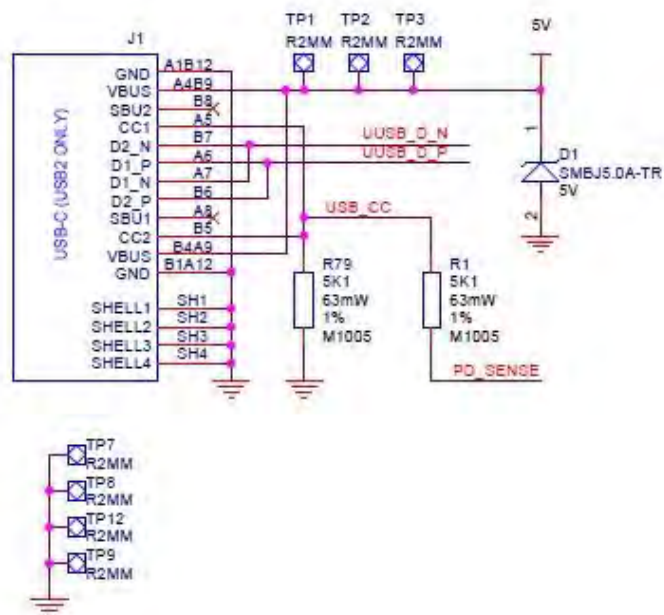
Εικόνα 48.f



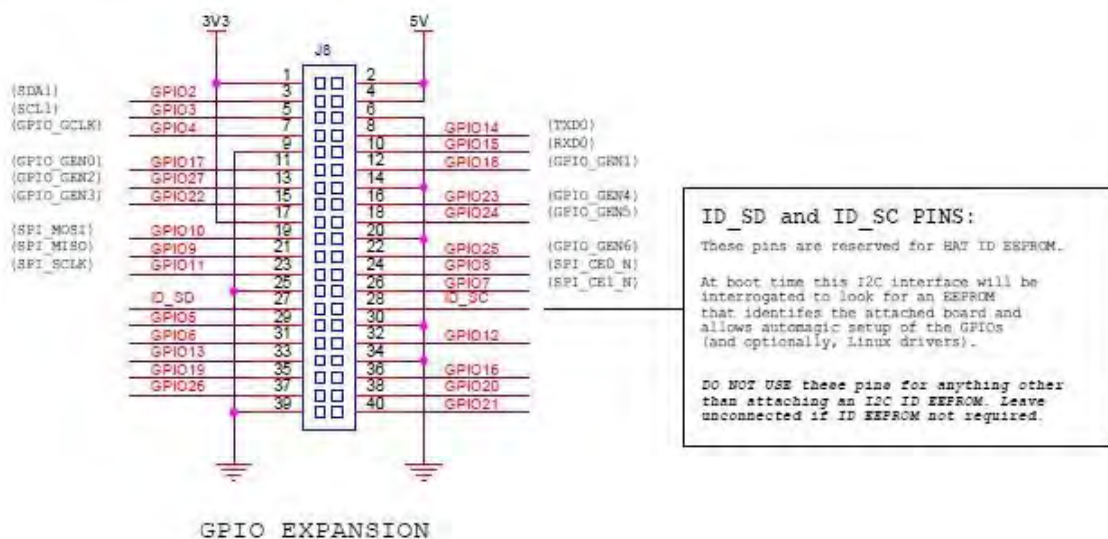
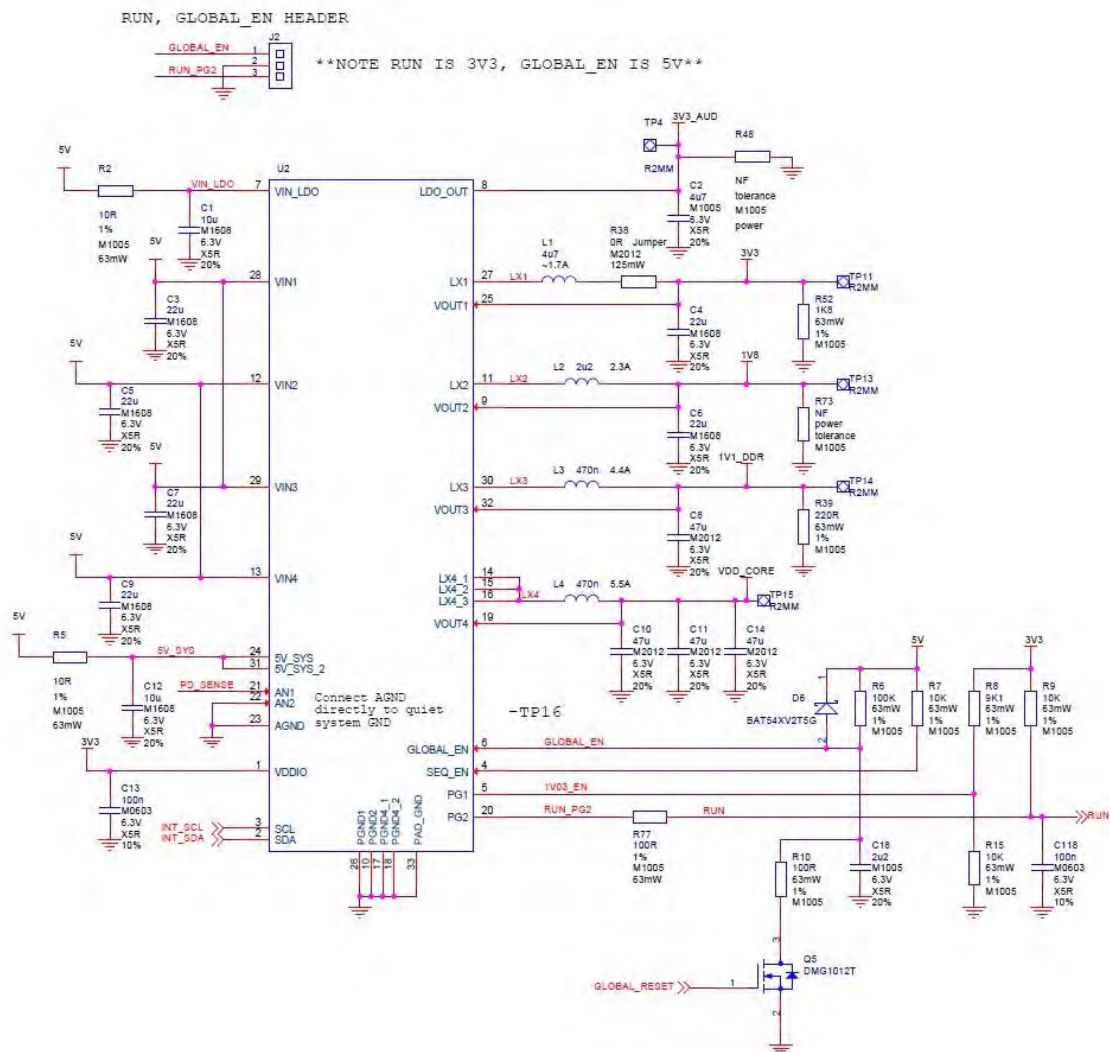
MICRO HDMI 1

Εικόνα 48.h

#### USB-C POWER IN



Εικόνα 48.i





## 4.3 Τεχνολογίες λογισμικών

Η χρήση της γλώσσας Python για τη δημιουργία και ολοκλήρωση του κώδικα, καθώς και η χρήση εξειδικευμένων λογισμικών ήταν καθοριστικές για την υλοποίηση του project. Πέραν του λογισμικού Raspberry Pi OS που χρησιμοποιήσαμε ως λειτουργικού της πλατφόρμας Raspberry Pi 4B, χρησιμοποιήθηκαν και δύο εξειδικευμένα λογισμικά το **Open CV** και το **YOLOV5** τα οποία και θα αναλύσουμε παρακάτω.

### 4.3.1 Λίγα λόγια για τη γλώσσα Python

Η Python είναι μια δυναμική γλώσσα προγραμματισμού διερμηνευόμενη, γενικού σκοπού, υψηλού επιπέδου. Ανήκει στις γλώσσες προστακτικού προγραμματισμού και υποστηρίζει διδακτικό και αντικειμενοστρεφές προγραμματισμό. Δημιουργήθηκε το 1989 από τον Ολλανδό Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica, το οποίο εδρεύει στο Άμστερνταμ της Ολλανδίας. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ό,τι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων. Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας, προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.

Αρχικά, η Python ήταν γλώσσα σεναρίων που χρησιμοποιήθηκε στο καταναμεημένο λειτουργικό σύστημα Amoeba, ικανή και για κλήσεις συστήματος. Θεωρείται διάδοχος της γλώσσας προγραμματισμού ABC, μια και αυτή υπήρξε η βασική πηγή έμπνευσης για τον Γκίντο βαν Ρόσσουμ. Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου του 2000. Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η έκδοση 3.0 (γνωστή και ως py3k ή python 3000). Πολλά από τα καινούργια χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στις εκδόσεις 2.6 και 2.7 που είναι προς τα πίσω συμβατές. Η Python 3.0 είναι ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις ώστε να διορθωθούν κάποια λάθη που υπήρχαν σε προγενέστερες εκδόσεις και να καταστεί ακόμα πιο σαφής ο απλός τρόπος με τον οποίο μπορούν να γίνουν κάποια πράγματα. Η Python 3.0 συνοδεύεται με το πρόγραμμα 2to3.py, που επιτρέπει την μετατροπή κώδικα της έκδοσης 2.0 σε 3.0. Η πιο πρόσφατη έκδοση είναι η 3.9.2 και δημοσιεύτηκε στις 26 Φεβρουαρίου 2021. Στην εικόνα 49 φαίνεται το λογότυπο της Python.



Εικόνα 49

Για τη συγγραφή προγραμμάτων είναι απαραίτητος ένας κειμενογράφος ή ακόμα καλύτερα ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE), το οποίο είναι ένα ειδικό λογισμικό για την ανάπτυξη εφαρμογών. Η Python έρχεται μαζί με ένα εύχρηστο και απλό περιβάλλον ανάπτυξης με την ονομασία IDLE. Τα αρχικά του προέρχονται από τις λέξεις Interactive DeveLopment Environment και είναι γραμμένο σε Python. Χρησιμοποιεί τη βιβλιοθήκη γραφικών Tkinter, οπότε μπορεί να εκτελεσθεί σε περιβάλλον Linux, Windows και Mac OS X. Το IDLE μας δίνει τη δυνατότητα να χρησιμοποιήσουμε διαδραστικά τον διερμηνευτή της γλώσσας, να γράψουμε και να επεξεργαστούμε προγράμματα, να τα αποθηκεύσουμε σε αρχεία, να τα εκτελέσουμε, να κάνουμε αποσφαλμάτωση (debugging) κ.α..

### 4.3.2 OpenCV

Το OpenCV (OpenCV Computer Vision Library) είναι μια βιβλιοθήκη λειτουργιών προγραμματισμού που στοχεύει κυρίως στην υπολογιστική όραση σε πραγματικό χρόνο. Αρχικά αναπτύχθηκε από την Intel, αργότερα υποστηρίχθηκε από την Willow Garage και στη συνέχεια από την Itseez (η οποία αργότερα εξαγοράστηκε από την Intel). Η βιβλιοθήκη είναι πολλαπλών πλατφορμών και δωρεάν για χρήση υπό την άδεια ανοιχτού κώδικα Apache 2. Ξεκινώντας από το 2011, το OpenCV διαθέτει επιτάχυνση GPU για λειτουργίες σε πραγματικό χρόνο.

Το project OpenCV ξεκίνησε επίσημα το 1999 και ήταν αρχικά μια ερευνητική πρωτοβουλία της Intel για την προώθηση εφαρμογών έντασης CPU, μέρος μιας σειράς έργων, συμπεριλαμβανομένης της ανίχνευσης ακτίνων σε πραγματικό χρόνο και των τοίχων οθόνης 3D. Οι κύριοι συντελεστές του έργου περιλάμβαναν έναν αριθμό ειδικών βελτιστοποίησης στην Intel Russia, καθώς και την ομάδα βιβλιοθήκης επιδόσεων της Intel. Στις πρώτες μέρες του OpenCV, οι στόχοι του έργου περιεγράφηκαν ως:

- ❖ Προχωρημένη έρευνα όρασης παρέχοντας όχι μόνο ανοιχτό αλλά και βελτιστοποιημένο κώδικα για βασικές υποδομές όρασης.
- ❖ Διάδοση της γνώσης της όρασης, παρέχοντας μια κοινή υποδομή στην οποία θα μπορούσαν να βασιστούν οι προγραμματιστές, έτσι ώστε ο κώδικας να είναι πιο ευανάγνωστος και μεταβιβάσιμος.
- ❖ Προώθηση εμπορικών εφαρμογών που βασίζονται στην όραση, καθιστώντας διαθέσιμο δωρεάν φορητό κώδικα βελτιστοποιημένο για απόδοση - με άδεια χρήσης που δεν απαιτούσε ο κώδικας να είναι ανοιχτός ή ελεύθερος.

Η πρώτη έκδοση «άλφα» του OpenCV κυκλοφόρησε στο κοινό στο Συνέδριο IEEE για την Υπολογιστική Όραση και την Αναγνώριση Προτύπων το 2000, και πέντε «βήτα» κυκλοφόρησαν μεταξύ 2001 και 2005. Η πρώτη έκδοση 1.0 κυκλοφόρησε το 2006. Μια έκδοση 1.1 κυκλοφόρησε τον Οκτώβριο του 2008. Η δεύτερη μεγάλη κυκλοφορία του OpenCV ήταν τον Οκτώβριο του 2009. Το OpenCV 2 περιλαμβάνει σημαντικές αλλαγές στη διεπαφή C++, με στόχο ευκολότερα, πιο ασφαλή για τον τύπο μοτίβα, νέες λειτουργίες και καλύτερες υλοποιήσεις για τις υπάρχουσες όσον αφορά την απόδοση (ειδικά σε συστήματα πολλαπλών πυρήνων). Οι επίσημες κυκλοφορίες πλέον, πραγματοποιούνται κάθε έξι μήνες και η ανάπτυξη γίνεται τώρα από μια ανεξάρτητη ρωσική ομάδα που υποστηρίζεται από εμπορικές εταιρείες.

Τον Αύγουστο του 2012, η υποστήριξη για το OpenCV αναλήφθηκε από ένα μη κερδοσκοπικό ίδρυμα OpenCV.org, το οποίο διατηρεί έναν ιστότοπο προγραμματιστή και χρήστη. Τον Μάιο του 2016, η Intel υπέγραψε συμφωνία για την εξαγορά της Itseez, κορυφαίου

προγραμματιστή του OpenCV. Τον Ιούλιο του 2020, το OpenCV ανακοίνωσε και ξεκίνησε μια καμπάνια Kickstarter για το OpenCV AI Kit, μια σειρά από ενότητες υλικού και προσθήκες στο OpenCV που υποστηρίζουν το Spatial AI. Στην εικόνα 50 φαίνεται το εμπορικό σήμα του OpenCV.



Εικόνα 50

Πως λειτουργεί όμως το OpenCV και τι πραγματικά κάνει; Το OpenCV είναι μια τεράστια open – source βιβλιοθήκη για την υπολογιστική όραση, μηχανική μάθηση και επεξεργασία εικόνας. Πλέον όμως στη σημερινή εποχή, παίζει σημαντικό ρόλο σε συστήματα που λειτουργούν και εξάγουν αποτελέσματα σε πραγματικό χρόνο. Με τη χρήση του, μπορεί κανείς να επεξεργαστεί εικόνες και βίντεο για να προσδιορίσει αντικείμενα, πρόσωπα, ή ακόμα και τον γραφικό χαρακτήρα ενός ανθρώπου. Όταν ενσωματωθεί με διάφορες βιβλιοθήκες, όπως NumPy και Python είναι ικανό να επεξεργαστεί σειρές δομών για την ανάλυση. Να προσδιορίσει το σχήδιο εικόνας και τα διάφορα χαρακτηριστικά γνωρίσματα, χρησιμοποιεί το διανυσματικό χώρο και εκτελεί μαθηματικές πράξεις σε αυτά τα χαρακτηριστικά. Το πρώτο OpenCV έκδοση **1.0. OpenCV** απελευθερώνεται κάτω από μια άδεια BSD και ως εκ τούτου, είναι δωρεάν, τόσο για την ακαδημαϊκή και όσο και για την εμπορική χρήση. Έχει διεπαφές C, C++, Python και Java και υποστηρίζει Windows, Linux, Mac OS, iOS και Android. Όταν σχεδιάστηκε το OpenCV, το επίκεντρο ήταν εφαρμογές υπολογιστικής απόδοσης σε πραγματικό χρόνο.

Υπάρχουν πολλές εφαρμογές που επιλύονται χρησιμοποιώντας το OpenCV, μερικές από αυτές παρατίθενται παρακάτω:

- ❖ Αναγνώριση προσώπου.
- ❖ Αυτοματοποιημένη επιθεώρηση και επιτήρηση.
- ❖ Αριθμός ατόμων – καταμέτρηση (κίνηση πεζών σε εμπορικό κέντρο κ.λπ.).
- ❖ Καταμέτρηση οχημάτων σε αυτοκινητόδρομους μαζί με τις ταχύτητές τους.
- ❖ Διαδραστικές εικαστικές εγκαταστάσεις.
- ❖ Ανίχνευση ανωμαλίας (ελαττώματος) στη διαδικασία κατασκευής (τα περιττά ελαττωματικά προϊόντα).
- ❖ Stitching εικόνων σε street view (διαδικασία κατά την οποία γίνεται ένωση πολλών εικόνων ίδιας βάσης με επικαλυπτόμενα πεδία, τα οποία παράγουν ένα τετμημένο πανόραμα ή μία εικόνα υψηλής ανάλυσης).
- ❖ Αναζήτηση και ανάκτηση βίντεο/εικόνων.
- ❖ Πλοήγηση και έλεγχος ρομπότ και αυτοκινήτου χωρίς οδηγό.
- ❖ Αναγνώριση αντικειμένων.
- ❖ Ανάλυση ιατρικής εικόνας.
- ❖ Ταινίες - 3D δομή από την κίνηση.

- ❖ Αναγνώριση διαφημίσεων τηλεοπτικών καναλιών.

Για την υποστήριξη ορισμένων από τους παραπάνω τομείς, το OpenCV περιλαμβάνει μια στατιστική βιβλιοθήκη μηχανικής μάθησης που περιέχει:

- ❖ Ενίσχυση
- ❖ Εκμάθηση δέντρου αποφάσεων
- ❖ Δέντρα που ενισχύουν την κλίση
- ❖ Αλγόριθμος μεγιστοποίησης προσδοκιών
- ❖  $k$ -πλησιέστερος αλγόριθμος γείτονα
- ❖ Αφελής ταξινομητής Bayes
- ❖ Τεχνητά νευρωνικά δίκτυα
- ❖ Τυχαίο δάσος
- ❖ Διανυσματική μηχανή υποστήριξης (SVM)
- ❖ Βαθιά νευρωνικά δίκτυα (DNN)

Λειτουργικότητα του OpenCV:

- ❖ Είσοδος / έξοδος εικόνας / βίντεο, επεξεργασία, οθόνη (πυρήνας, imgproc, highgui)
- ❖ Ανίχνευση αντικειμένων / χαρακτηριστικών (στόχος, features 2d)
- ❖ Μονοφωνική ή στερεοφωνική υπολογιστική όραση με βάση τη γεωμετρία (calib3d, stitching, βιντεοκασέτα)
- ❖ Υπολογιστική φωτογραφία (φωτογραφία, βίντεο)
- ❖ Μηχανική μάθηση και ομαδοποίηση
- ❖ Επιτάχυνση CUDA (gpu)

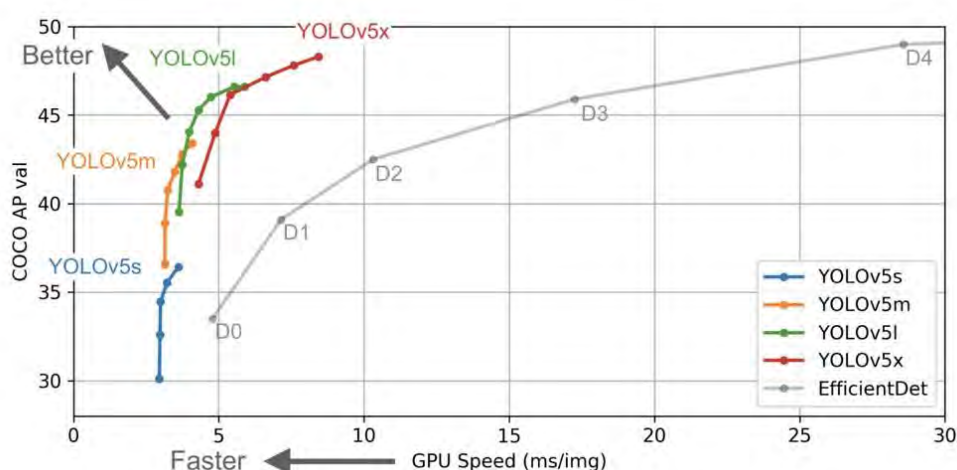
Η επεξεργασία εικόνας είναι μια μέθοδος για την εκτέλεση ορισμένων λειτουργιών σε μια εικόνα, προκειμένου να αποκτήσουμε μια βελτιωμένη εικόνα και / ή να εξάγουμε χρήσιμες πληροφορίες από αυτήν. Αν μιλάμε για τον βασικό ορισμό της επεξεργασίας εικόνας τότε **«Επεξεργασία εικόνας είναι η ανάλυση και ο χειρισμός μιας ψηφιοποιημένης εικόνας, ειδικά για τη βελτίωση της ποιότητάς της»**.

Μια ψηφιακή εικόνα μπορεί να οριστεί ως μια δισδιάστατη συνάρτηση  $f(x, y)$ , όπου  $x$  και  $y$  είναι χωρικές συντεταγμένες (επίπεδο) και το πλάτος οποιοδήποτε ζεύγος συντεταγμένων  $(x, y)$  ονομάζεται ένταση ή επίπεδο γκρι της εικόνας σε αυτό το σημείο. Με απλά λόγια, μια εικόνα δεν είναι τίποτα περισσότερο από μια δισδιάστατη μήτρα (3-D στην περίπτωση έγχρωμων εικόνων) που ορίζεται από τη μαθηματική συνάρτηση  $f(x, y)$  σε οποιοδήποτε σημείο δίνει την τιμή pixel, η τιμή αυτή περιγράφει πόσο φωτεινό είναι αυτό το εικονοστοιχείο και τι χρώμα πρέπει να είναι. Η επεξεργασία εικόνας είναι βασικά επεξεργασία σήματος, στην οποία η είσοδος είναι μια εικόνα και η έξοδος είναι εικόνα ή χαρακτηριστικά, σύμφωνα με τις απαιτήσεις που σχετίζονται με αυτήν την εικόνα. Η επεξεργασία εικόνας περιλαμβάνει βασικά τα ακόλουθα τρία βήματα:

- ❖ Εισαγωγή της εικόνας.
- ❖ Ανάλυση και χειρισμός της εικόνας.
- ❖ Έξοδος, στην οποία το αποτέλεσμα μπορεί να είναι τροποποίηση της εικόνας ή αναφορά που βασίζεται στην ανάλυση εικόνας.

### 4.3.3 YOLO V5

Το YOLOv5 είναι ένα από τα μοντέλα της οικογένειας μοντέλων υπολογιστικής όρασης **You Only Look Once** (YOLO). Το YOLOv5 χρησιμοποιείται συνήθως για την ανίχνευση αντικειμένων. Το YOLOv5 διατίθεται σε τέσσερις κύριες εκδόσεις: μικρές (s), μεσαίες (m), μεγάλες (l) και πολύ μεγάλες (x), καθεμία από τις οποίες προσφέρει προοδευτικά υψηλότερα ποσοστά ακρίβειας. Κάθε παραλλαγή χρειάζεται επίσης διαφορετικό χρόνο για να εκπαιδευτεί. Όπως φαίνεται στην εικόνα 51, η αρχική έκδοση του YOLOv5 δίνει υπόσχεση για ανίχνευση αντικειμένων τελευταίας τεχνολογίας.



Εικόνα 51

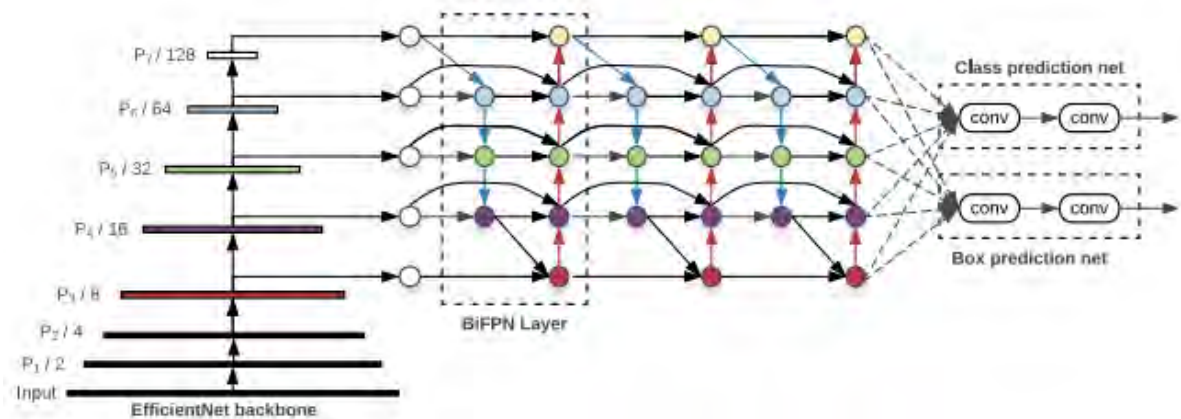
Στο γράφημα, ο στόχος είναι να παραχθεί ένα μοντέλο ανιχνευτή αντικειμένων, που είναι πολύ αποδοτικό (άξονας Y) σε σχέση με το χρόνο εξαγωγής συμπερασμάτων (άξονας X). Τα προκαταρκτικά αποτελέσματα δείχνουν ότι το YOLOv5 τα πάει εξαιρετικά καλά για το σκοπό αυτό, σε σχέση με άλλες τεχνικές αιχμής. Στο παραπάνω διάγραμμα, μπορείτε να δείτε ότι όλες οι παραλλαγές του YOLOv5 εκπαιδεύονται γρηγορότερα από το EfficientDet. Το πιο ακριβές μοντέλο του YOLOv5, το YOLOv5x, μπορεί να επεξεργαστεί εικόνες πολλές φορές πιο γρήγορα με παρόμοιο βαθμό ακρίβειας από το μοντέλο EfficientDet D4. Το YOLOv5 αντλεί το μεγαλύτερο μέρος της βελτίωσης της απόδοσής του από τις διαδικασίες εκπαίδευσης PyTorch, ενώ η αρχιτεκτονική μοντέλου παραμένει κοντά στο YOLOv4.

Μετά την πλήρη αναπαραγωγή της αρχιτεκτονικής μοντέλου και της εκπαιδευτικής διαδικασίας του YOLOv3, η Ultralytics άρχισε να κάνει ερευνητικές βελτιώσεις παράλληλα με τις αλλαγές στο σχεδιασμό του αποθετηρίου με στόχο να ενδυναμώσει χιλιάδες προγραμματιστές να εκπαιδεύσουν και να αναπτύξουν τους δικούς τους προσαρμοσμένους ανιχνευτές αντικειμένων για την ανίχνευση οποιουδήποτε αντικειμένου στον κόσμο. Αυτός είναι ένας στόχος που μοιραζόμαστε εδώ στο Roboflow. Αυτές οι εξελίξεις ονομάστηκαν αρχικά YOLOv4 λόγω της πρόσφατης κυκλοφορίας του YOLOv4 στο πλαίσιο Darknet, το μοντέλο μετονομάστηκε σε YOLOv5 για να αποφευχθούν προβλήματα στις εκδόσεις. Υπήρξε αρκετή συζήτηση γύρω από την ονομασία YOLOv5 στην αρχή και έτσι δημοσιεύθηκε ένα άρθρο, που συγκρίνει το YOLOv4 και το YOLOv5, όπου θα μπορούσε κάποιος και τα δύο μοντέλα δίπλα-δίπλα στα δικά σας δεδομένα. Αξίζει να σημειωθεί ότι, από τότε που δημοσιεύθηκε το



αποθετήριο, έχει σημειωθεί σημαντική ερευνητική πρόοδος στο YOLOv5, το οποίο αναμένεται να συνεχιστεί. Έτσι, το YOLOv5 δεν είναι σε καμία περίπτωση ένα τελικό μοντέλο, αλλά ένα μοντέλο το οποίο θα εξελιχθεί με την πάροδο του χρόνου.

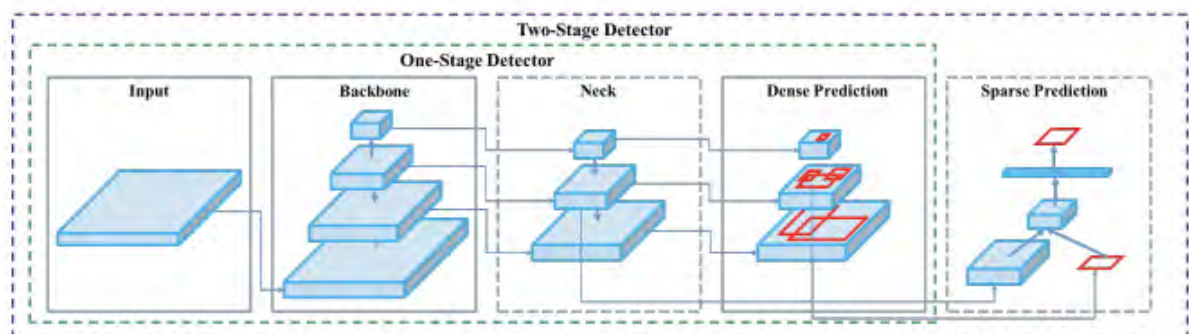
Η ανίχνευση αντικειμένων, μια περίπτωση χρήσης για την οποία έχει σχεδιαστεί το YOLOv5, περιλαμβάνει τη δημιουργία χαρακτηριστικών από εικόνες εισόδου. Αυτά τα χαρακτηριστικά τροφοδοτούνται στη συνέχεια μέσω ενός συστήματος πρόβλεψης, για να σχεδιάσουν κουτιά γύρω από αντικείμενα και να προβλέψουν τις κλάσεις τους. Στην εικόνα 52, φαίνεται η ανατομία ενός ανιχνευτή αντικειμένων.



Εικόνα 52

Το μοντέλο YOLO είναι ο πρώτος ανιχνευτής αντικειμένων, που συνέδεσε τη διαδικασία πρόβλεψης περιοριστικών πλαισίων με ετικέτες κλάσεων σε ένα δίκτυο διαφορισιμότητας από άκρο σε άκρο. Το δίκτυο YOLO αποτελείται από τρία βασικά κομμάτια τα οποία φαίνονται και στην εικόνα 53:

- ❖ Backbone (ραχοκοκαλιά): Ένα συνελκτικό νευρωνικό δίκτυο που συγκεντρώνει και σχηματίζει χαρακτηριστικά εικόνες με διαφορετικές λεπτομέρειες.
- ❖ Neck (λαιμός): Μια σειρά από επίπεδα όπου γίνεται ανάμιξη και συνδυασμός χαρακτηριστικών ώστε να περάσει στη πρόβλεψη.
- ❖ Head (κεφάλι): Καταναλώνει χαρακτηριστικά από το λαιμό και κάνει βήματα πρόβλεψης κουτιά και τάξεις.



Εικόνα 53

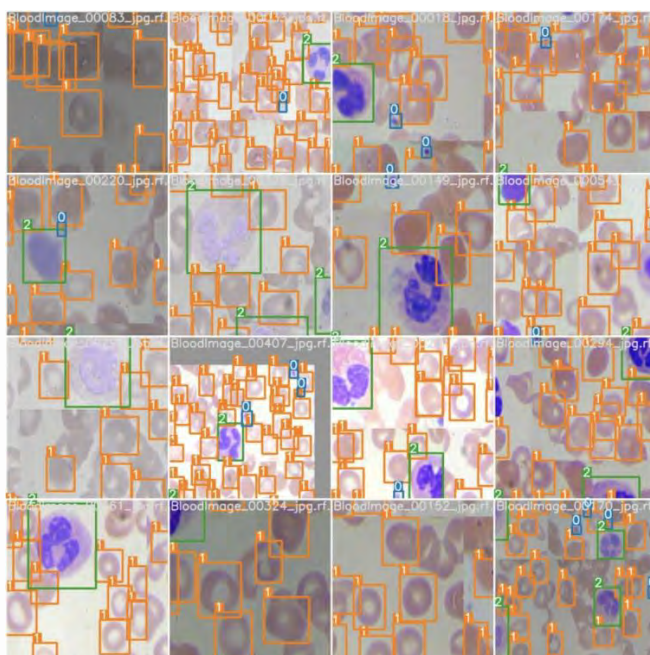
Σύμφωνα με τα παραπάνω, υπάρχουν πολλές προσεγγίσεις που μπορεί κανείς να ακολουθήσει, για να συνδυάσει διαφορετικές αρχιτεκτονικές σε κάθε σημαντικό στοιχείο. Οι συνεισφορές των YOLOv4 και YOLOv5 είναι πρωτίστως για την ενσωμάτωση ανακαλύψεων σε άλλους τομείς της υπολογιστικής όρασης και αποδεικνύουν ότι ως συλλογή, βελτιώνουν την ανίχνευση αντικειμένων YOLO. Οι διαδικασίες που λαμβάνονται για την εκπαίδευση ενός μοντέλου, είναι εξίσου σημαντικές με οποιονδήποτε παράγοντα για την τελική απόδοση ενός συστήματος ανίχνευσης αντικειμένων, αν και συχνά συζητούνται λιγότερο. Παρακάτω θα δούμε δύο κύριες διαδικασίες εκπαίδευσης στο YOLOv5:

- ❖ **Αύξηση δεδομένων:** Η αύξηση δεδομένων κάνει μετασχηματισμούς στα βασικά δεδομένα εκπαίδευσης, για να εκθέσει το μοντέλο σε ένα ευρύτερο φάσμα σημασιολογικών παραλλαγών, από το σύνολο εκπαίδευσης μεμονωμένα.
- ❖ **Υπολογισμοί απωλειών:** Το YOLO υπολογίζει μια συνάρτηση συνολικής απώλειας από τις συναρτήσεις GIoU, obj και απωλειών κλάσης. Αυτές οι συναρτήσεις, μπορούν να κατασκευαστούν προσεκτικά για να μεγιστοποιήσουν το στόχο της μέσης ακρίβειας.

Με κάθε παρτίδα εκπαίδευσης, το YOLOv5 περνά δεδομένα προπόνησης μέσω ενός loader δεδομένων, ο οποίος αυξάνει τα δεδομένα στο διαδίκτυο. Ο loader δεδομένων πραγματοποιεί τρία είδη αυξήσεων:

- ❖ Κλιμάκωση.
- ❖ Ρυθμίσεις χρωματικού χώρου.
- ❖ Μωσαϊκό επαύξησης.

Το πιο νέο από αυτά, είναι η αύξηση δεδομένων μωσαϊκού, η οποία συνδυάζει τέσσερις εικόνες σε τέσσερα πλακίδια τυχαίας αναλογίας. Ο loader ψηφιδωτών δεδομένων είναι εγγενής στο YOLOv3 PyTorch και τώρα στο repo YOLOv5. Η αύξηση μωσαϊκού είναι ιδιαίτερα χρήσιμη για το δημοφιλές σημείο αναφοράς ανίχνευσης αντικειμένων COCO, βοηθώντας το μοντέλο να μάθει να αντιμετωπίζει το γνωστό "πρόβλημα μικρών αντικειμένων" - όπου τα μικρά αντικείμενα δεν ανιχνεύονται με τόση ακρίβεια όσο τα μεγαλύτερα αντικείμενα. Παρακάτω, στην εικόνα 54, βλέπουμε μια φωτογραφία επαυξημένων εικόνων προπόνησης στο YOLOv5.



Εικόνα 54



## 4.4 Object detection project

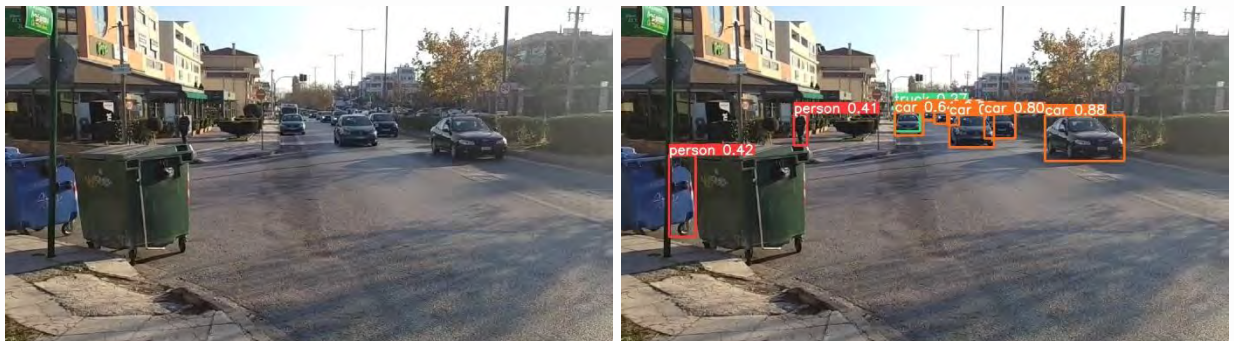
Στην παράγραφο 4.4, θα αναλύσουμε τα δεδομένα και τον κώδικα που χρησιμοποιήσαμε για την υλοποίηση του project. Είναι σημαντικό να αναφέρουμε, πως έγινε χρήση των παραπάνω λογισμικών και της γλώσσας προγραμματισμού Python, ώστε να ολοκληρωθεί το project με επιτυχία. Ο λήψη του κώδικα έγινε από τον ιστότοπο **GitHub**, μαζί με όλες πληροφορίες αναφορικά με τη διαδικασία εκτέλεσης των εργασιών. Πραγματοποιήθηκε μία διαδικασία από τη γραμμή εντολών του Raspberry Pi, ώστε να ληφθούν οι κατάλληλες βιβλιοθήκες και να γίνει δημιουργία των επιμέρους φακέλων με τα αντίστοιχα αρχεία που πρέπει για να τρέξει το πρόγραμμα και να εξάγει τα σωστά αποτελέσματα. Είναι σημαντικό εδώ, να αναφέρουμε πως πρώτα εγκαταστήσαμε τη γλώσσα Python και ακολούθως το OpenCV.

Ακολουθεί η λίστα εντολών για τη δημιουργία του Project:

```
>>sudo git clone https://github.com/freedomwebtech/yolov5raspberrypi4.git
>>cd yolov5raspberrypi4
>>ls
>>Sudo chmod 775 install.sh
>>ls
>>sudo ./install.sh
>>cd
>>sudo pip3 install yolov5
>>sudo chown -R pi:pi /usr/local/lib/py
>>sudo chown -R pi:pi /usr/local/lib/python 3.9/dist-packages/yolov5
```

Στην συνέχεια, ανοίγουμε το περιβάλλον ανάπτυξης Thonny και τρέχουμε τον κώδικα που παραθέτουμε στο Παράρτημα Ι.

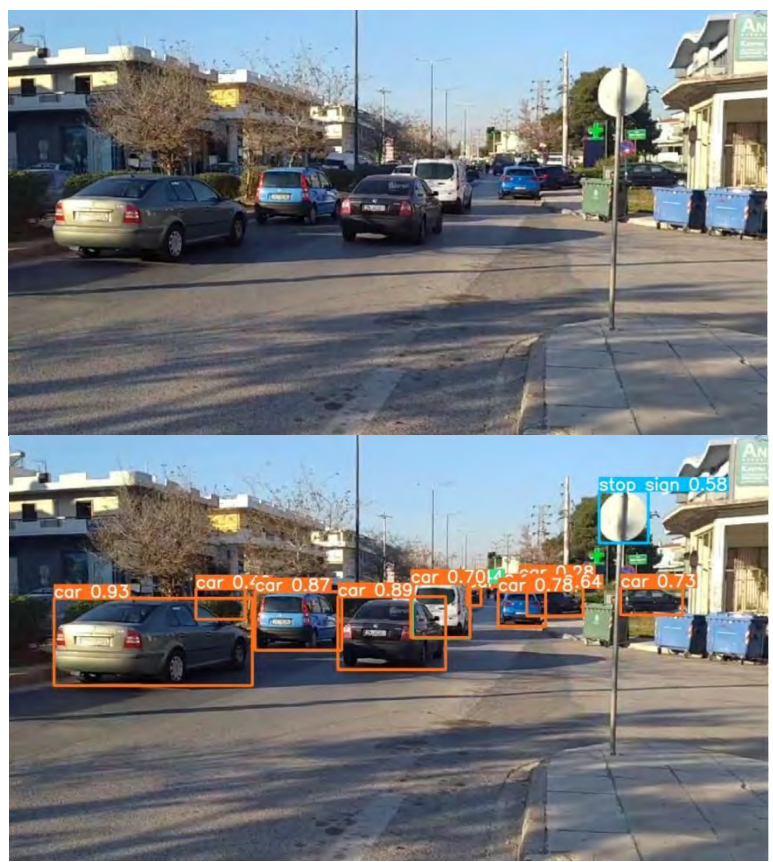
Οι εικόνες και τα βίντεο που θέλουμε να αναλύσουμε, τοποθετούνται στον φάκελο [/home/pi/freedomtech](#) και τα εξαγόμενα αποτελέσματα εμφανίζονται στον φάκελο [/home/pi/yoloresult/exp](#). Παρακάτω παρατίθενται φωτογραφίες πριν και μετά την επεξεργασία (εικόνες 55 – 65).



Εικόνα 55



Εικόνα 56



Εικόνα 57





Εικόνα 58

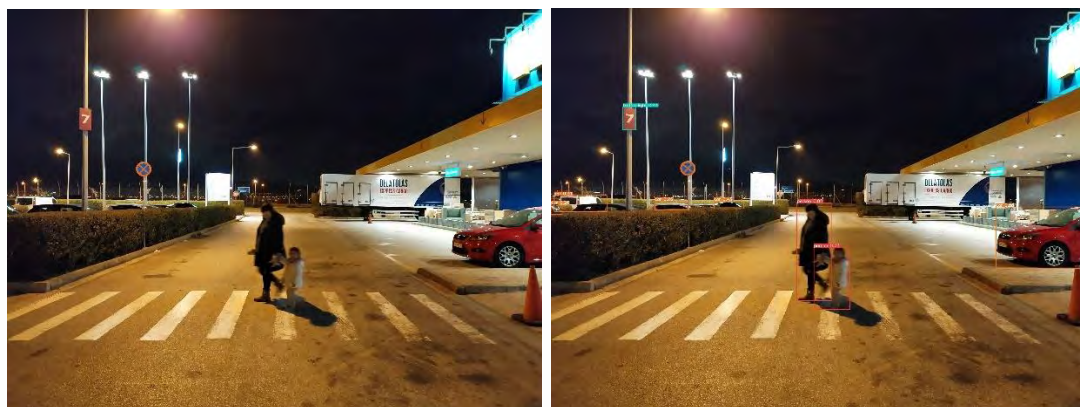


Εικόνα 59

Η παραπάνω φωτογραφίες (εικόνα 55 – εικόνα 59) είναι τμήματα βίντεο που λήφθηκε από στάση αστικών συγκοινωνιών. Όπως φαίνεται στα αποτελέσματα, η επεξεργασία των εικόνων και η εξαγωγή των αποτελεσμάτων είχαν σχεδόν απόλυτη επιτυχία. Έχει γίνει πλήρη αναγνώριση των αντικειμένων όπως αυτοκινήτων, μοτοσυκλετών, ανθρώπων και πινακίδων σήμανσης. Το πρόγραμμα αναγνώρισε και οχήματα τα οποία βρίσκονταν σε αρκετά δύσκολη θέση ώστε να εντοπιστούν. Στη συνέχεια, θα παρουσιάσουμε φωτογραφίες (εικόνα 60 – εικόνα 65) από τη θέση του οδηγού και του συνοδηγού, δηλαδή από το εσωτερικό του αυτοκινήτου. Είναι εικόνες που βλέπει ο οδηγός αλλά και ένα σύστημα Dashcam. Είναι λοιπόν, πολύ σημαντικό, να δούμε πως ανταποκρίθηκε το πρόγραμμα στις συγκεκριμένες φωτογραφίες.



Εικόνα 60

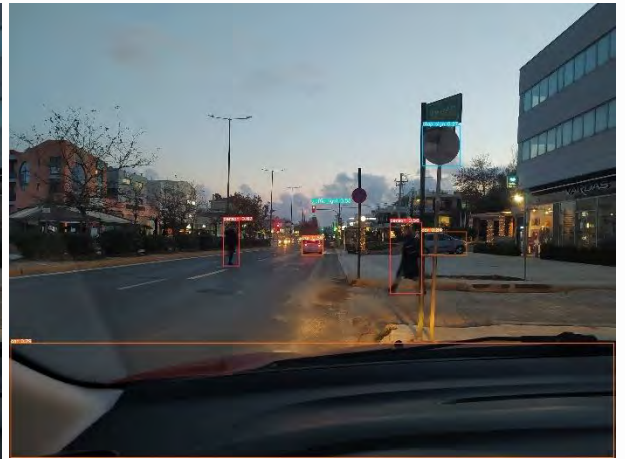
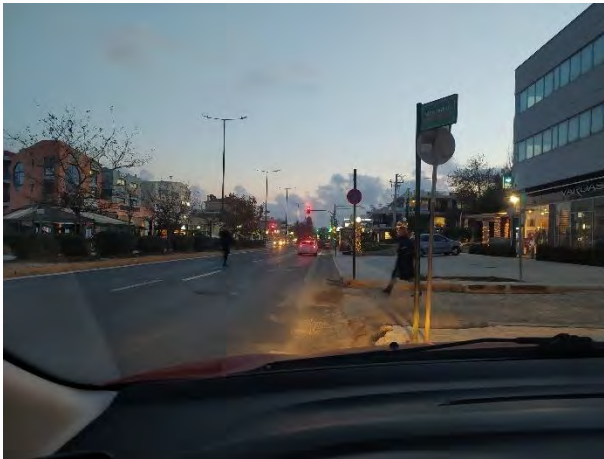


Εικόνα 61

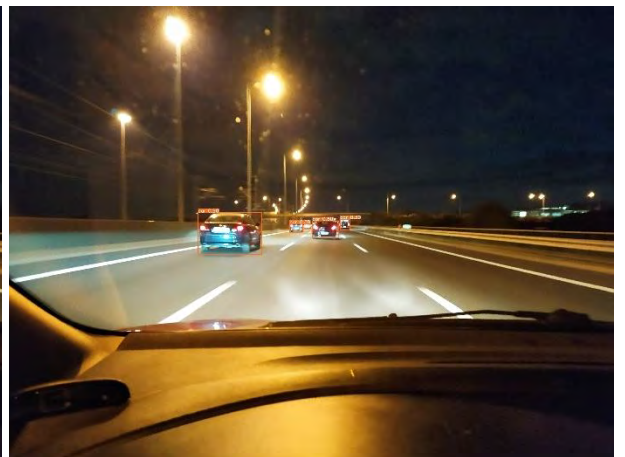




Εικόνα 62



Εικόνα 63



Εικόνα 64



Εικόνα 65

Βλέποντας τις φωτογραφίες, παρατηρούμε πως η αποδοτικότητα του συστήματος, ειδικά στις νυχτερινές και σε αυτές με μειωμένη φωτεινότητα είναι πολύ ικανοποιητική. Έγινε αναγνώριση οχημάτων, φαναριών και πινακίδων σήμανσης, όπως επίσης και πλήρης αναγνώρισης ατόμων. Η αναγνώριση ατόμων και ερυθρού σηματοδότη είναι ένα ιδιαίτερο σημείο στο οποίο πρέπει να σταθούμε, καθώς είναι τα βασικά στοιχεία αναγνώρισης στα οποία στηρίζεται η λειτουργία του συστήματος Dashcam. Πέραν τούτου, είναι δύο πολύ σημαντικά στοιχεία αναγνώρισης και προειδοποίησης του χρήστη (οδηγού) προς αποφυγή ατυχημάτων.

Τέλος, η εταιρία Ultralytics, δημιουργός των YOLOv5 και YOLOv8, μέσω της εφαρμογής Pytorch, δίνει τη δυνατότητα σε χρήστες να δημιουργήσουν το δικό τους project ξεκινώντας με λιγότερο από μόλις 6 γραμμές. Επίσης, έχει δημιουργήσει ειδική εφαρμογή για κινητά (mobile app), όπου μόλις ενεργοποιηθεί μπορεί ο χρήστης να πραγματοποιήσει αναγνώριση αντικειμένων. Κάνοντας download τα κατάλληλα datasets, μπορεί να γίνει ένα



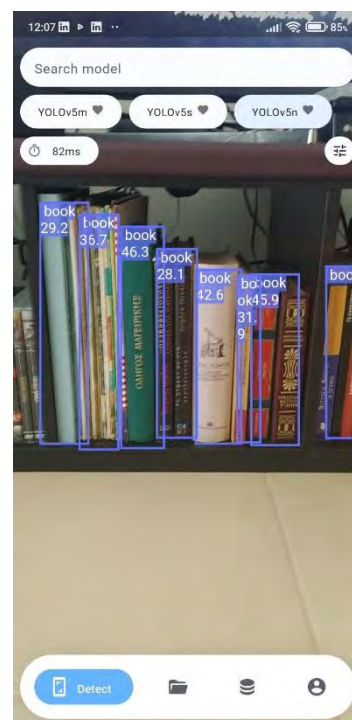
σημαντικό εργαλείο στα χέρια του χρήστη. Παρακάτω, βλέπουμε φωτογραφίες screen shot από την εφαρμογή του κινητού σε λειτουργία (εικόνα 66 – εικόνα 71).



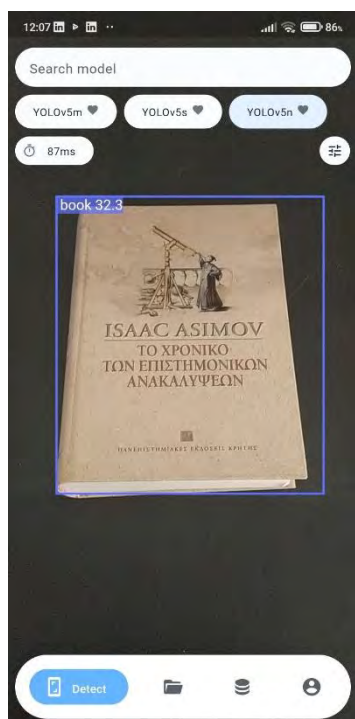
Εικόνα 66



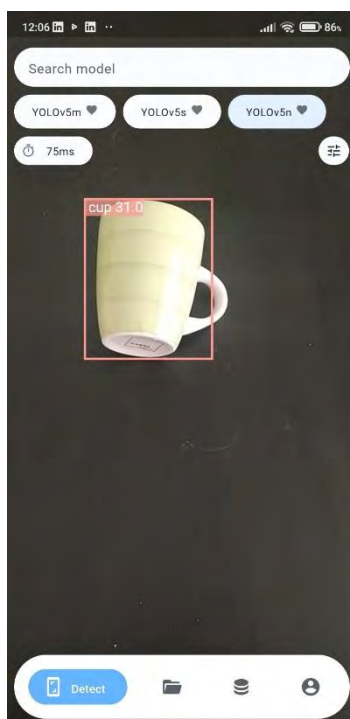
Εικόνα 67



Εικόνα 68



Εικόνα 69



Εικόνα 70



Εικόνα 71

## 5 Συμπεράσματα – Μελλοντική έρευνα

### 5.1 Εισαγωγή

Όπως αναφέραμε και στο κεφάλαιο 1 – εισαγωγή, σκοπός της παρούσας διπλωματικής εργασίας, είναι η δημιουργία ενός συστήματος αναγνώρισης προτύπων με έμφαση στην οδική κυκλοφορία. Μέσα από την ανάπτυξη της παρούσας διπλωματικής εργασίας ο αναγνώστης θα λάβει πλήρη ενημέρωση και πληροφορίες, σχετικά με τον τρόπο αναγνώρισης προτύπων, τους αλγόριθμους που τρέχουν σε ένα τέτοιο σύστημα, τη μηχανική μάθηση και τα τεχνητά νευρωνικά δίκτυα. Πραγματοποιήθηκε μία αναλυτική παρουσίαση της αναγνώρισης προτύπων και της ταξινόμησης τους με παραδείγματα, μέσω της πλήρους θεωρητικής και πρακτικής ανάλυσης. Στο θεωρητικό τμήμα της διπλωματικής εργασίας – κεφάλαια 2, 3 και 4 – ο αναγνώστης λαμβάνει πληροφορίες αναφορικά με τους αλγόριθμους αναγνώρισης προτύπων, τη δομή του συστήματος ταξινόμησής τους, την τεχνητή νοημοσύνη και τα νευρωνικά δίκτυα που «τρέχουν» σε τέτοια συστήματα. Τέλος, περιγράφεται μία πλήρης ανάπτυξη ενός συστήματος αναγνώρισης προτύπων μέσω της πλατφόρμας Raspberry Pi 4B.

### 5.2 Συμπεράσματα

Στόχος της παρούσα διπλωματικής εργασίας, είναι να δώσει μια πλήρη εικόνα στον αναγνώστη, αναφορικά με τη δομή και λειτουργία ενός συστήματος αναγνώρισης προτύπων με έμφαση στην οδική κυκλοφορία. Έρχεται να συμπληρώσει ένα κενό στην υπάρχουσα βιβλιογραφία, αναφορικά με τα συστήματα εποπτείας οδικής κυκλοφορίας. Θα μπορούσαμε να χαρακτηρίσουμε την παρούσα διπλωματική εργασία, ως έναν σύντομο και περιεκτικό οδηγό αναφορικά με την αναγνώριση προτύπων σε θεωρητικό κομμάτι αλλά και σε πρακτικό, αφού παρουσιάζει ένα μικρό πείραμα με ικανοποιητικά αποτελέσματα.

Η σημασία της διπλωματικής εργασίας προσδίδεται, από το γεγονός ότι μας δίνει μια ολοκληρωμένη εικόνα αναφορικά με τα συστήματα αναγνώρισης προτύπων και μας δείχνει τη δυναμική που έχουν αυτά τα συστήματα. Τέλος, απώτερος σκοπός της είναι να βοηθήσει ώστε να αποτυπωθούν αρχικές δυσκολίες και περιορισμοί, ώστε σε μελλοντική επέκτασή της σε συστήματα εποπτείας οδικής κυκλοφορίας, να δοθούν ουσιώδεις λύσεις.

Αξιόλογα σημεία στα οποία πρέπει να σταθούμε είναι στο τι είναι πραγματικά η αναγνώριση προτύπων (Pattern Recognition), ποιες είναι οι μέθοδοι αναγνώρισης και η διαδικασία ταξινόμησης καθώς επίσης τα τεχνητά νευρωνικά δίκτυα στα οποία βασίζεται η αναγνώριση προτύπων.

Έτσι λοιπόν, αναγνώριση προτύπων είναι ένα επιστημονικό πεδίο με στόχο την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής ή διακριτικού στοιχείου σε εισαγόμενα δεδομένα, συνήθως κωδικοποιημένα ως αλληλουχίες αριθμών. Τα δεδομένα αυτά, κατηγοριοποιούνται βάση κάποιων κριτηρίων. Οι άνθρωποι σε σύγκριση με τις μηχανές, κατηγοριοποιούν τα δεδομένα τα οποία λαμβάνουν βάση της ευφυΐας τους. Μία υπολογιστική μηχανή ή ένα υπολογιστικό σύστημα κατηγοριοποιεί τα δεδομένα βάση κάποιων αλγορίθμων ταξινόμησης. Για να γίνει αυτό όμως, θα πρέπει πρώτα το σύστημα να εκπαιδευτεί, ώστε να αναγνωρίζει τα πρότυπα και να τα κατηγοριοποιεί. Βάση των παραπάνω, θα μπορούσαμε να ορίσουμε την αναγνώριση προτύπων πιο ειδικά ως μία διαδικασία κατά την οποία ένα

υπολογιστικό σύστημα λειτουργεί έξυπνα στην επεξεργασία εισόδου ποικιλόμορφων δεδομένων. Η αναγνώριση προτύπων έχει αρκετά πεδία εφαρμογής όπως μηχανική όραση (Machine Vision), αναγνώριση χαρακτήρων (Character Recognition – OCR), αναγνώριση ομιλίας και προσώπου (Speech and Face Recognition), Βιομετρία (Biometrics), Βιοπληροφορική (Bioinformatics) κ.α..

Τα νευρωνικά δίκτυα, βρίσκουν σχεδόν απόλυτη εφαρμογή στην αναγνώριση προτύπων και την τεχνητή νοημοσύνη. Ένα σύστημα τεχνητής νοημοσύνης θα πρέπει **να σκέφτεται όπως ο άνθρωπος, να σκέφτεται ορθολογικά, να συμπεριφέρεται όπως ο άνθρωπος και να συμπεριφέρεται ορθά**. Αυτά τα χαρακτηριστικά είναι θεμελιώδη για την ανάπτυξη της τεχνητής νοημοσύνης γενικά και ειδικά στα συστήματα αναγνώρισης προτύπων. Τα νευρωνικά δίκτυα είναι ο δεύτερος πυλώνας και πολύ ουσιαστικός στα οποία στηρίζεται η αναγνώριση προτύπων. Τα νευρωνικά δίκτυα μιμούνται τον τρόπο λειτουργίας των βιολογικών νευρώνων. Υπάρχουν πολλοί τύποι νευρωνικών δικτύων όπως τα **Νευρωνικά Δίκτυα Τροφοδότη** – FNN (Feedforward Neural Network), τα νευρωνικά δίκτυα συνάρτησης Radial Basis, τα **Επαναλαμβανόμενα Νευρωνικά Δίκτυα** – RNN (Recurrent Neutral Networks), τα **Αρθρωτά Νευρωνικά Δίκτυα** – MNN (Modular Neutral Networks) και τα **Συνελικτικά Νευρωνικά Δίκτυα** CNN – (Convolutional Neural Networks).

Στόχος της παρούσας διπλωματικής εργασίας, είναι να αποτυπώσει όλες τις απαραίτητες αρχικές πληροφορίες για την έρευνα και τη δημιουργία ενός συστήματος αναγνώρισης προτύπων. Επιπλέον, προσθέτει πληροφορίες στη μικρής έκτασης κατασκευή ενός συστήματος ανοιχτών τεχνολογιών με μία οικονομική και συνάμα υπολογιστικά επαρκή πλατφόρμα όπως το Raspberry Pi 4B.

## Βιβλιογραφία

- Clark, D. (2017, Μάιος). Vehicle Detection Using Cameras for Self-Driving Cars. Ανάκτηση Οκτώβριος 2022, από <https://becominghuman.ai/vehicle-detection-using-cameras-for-self-driving-cars-db806dccc65>
- Deep Ai. (χ.χ.). Recurrent Neural Networks. Ανάκτηση Οκτώβριος 2022, από <https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network>
- Deuschle, W. J. (2018, Δεκέμβριος). *Undergraduate Fundamentals of Machine Learnings*. Cambridge , Massachusetts. Ανάκτηση Σεπτέμβριος 2022, από <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37364585>
- FreedomTech. (2022, Οκτώβριος). freedomwebtech / YOLOv5 Raspberry-pi4. Ανάκτηση Δεκέμβριος 2022, από <https://github.com/freedomwebtech/yolov5raspberry-pi4>
- Nilson, N. (1998, Νοέμβριος). *INTRODUCTION TO MACHINE LEARNING*. (D. ο. Robotics Laboratory, Επιμ.) Stanford, CA 94305, USA: Stanford University. Ανάκτηση Σεπτέμβριος 2022
- Pruegsanusak, K. (2020, May). <https://dspace.mit.edu>. Ανάκτηση 2022, από <https://dspace.mit.edu/Handle/1193017634-MIT>
- Qiao, J., Meng, X., Li, W., & Wilamowski, B. (2018, Οκτώβριος). A novel modular RBF neural network based on a brain-like partition method. *Neural Computing and Applications* . Ανάκτηση Οκτώβριος 2022
- Ramswarup, K. (2022, Οκτώβριος). OpenCV – Overview. Ανάκτηση Ιανουάριος 2023, από <https://www.geeksforgeeks.org/opencv-overview/>
- Raspberry\_Pi. (2023, Ιανουάριος). Raspberry Pi Hardware - Documentations. Ανάκτηση Ιανουάριος 2023, από <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#gpio-and-the-40-pin-header>
- Solawetz, J. (2022, Ιούνιος). What is YOLOv5? A Guide for Beginners. Ανάκτηση Ιανουάριος 2023, από <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>
- Szarvas, M., Yoshizawa, A., Yamamoto, M., & Ogata, J. (2005, Σεπτέμβριος 12). Pedestrian Detection with Convolutional Neural Networks. *IEEE Xplore*, σ. 6. doi:10.1109/IVS.2005.1505106
- Ultralytics. (2023, Ιανουάριος). Ανάκτηση από Ultralytics: <https://ultralytics.com/yolov5>
- Wikipedia. (2020). Turing Test. Ανάκτηση Σεπτέμβριος 2022, από [https://en.wikipedia.org/wiki/Turing\\_test](https://en.wikipedia.org/wiki/Turing_test)
- Wikipedia. (2021). General Problem Solver. Ανάκτηση Σεπτέμβριος 2022, από [https://en.wikipedia.org/wiki/General\\_Problem\\_Solver](https://en.wikipedia.org/wiki/General_Problem_Solver)
- Wikipedia. (2022). Alan Turing. Ανάκτηση Σεπτέμβριος 2022, από [https://en.wikipedia.org/wiki/Alan\\_Turing](https://en.wikipedia.org/wiki/Alan_Turing)
- Wikipedia. (2022). Artificial Intelligence. Ανάκτηση Σεπτέμβριος 2022, από [https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence)
- Wikipedia. (2022, Σεπτέμβριος). Feedforward neural network. Ανάκτηση Οκτώβριος 2022, από [https://en.wikipedia.org/wiki/Feedforward\\_neural\\_network](https://en.wikipedia.org/wiki/Feedforward_neural_network)

- Wikipedia. (2022, Σεπτέμβριος). Modular neural network. Ανάκτηση Οκτώβριος 2022, από [https://en.wikipedia.org/wiki/Modular\\_neural\\_network](https://en.wikipedia.org/wiki/Modular_neural_network)
- Wikipedia. (2022, Απρίλιος). Pattern recognitions. Ανάκτηση Μάιος 2022, από [https://en.wikipedia.org/wiki/Pattern\\_recognition](https://en.wikipedia.org/wiki/Pattern_recognition)
- Wikipedia. (2022, Δεκέμβριος). Python. Ανάκτηση Ιανουάριος 2023, από <https://el.wikipedia.org/wiki/Python>
- Wikipedia. (2022, Σεπτέμβριος). Radial basis function network. Ανάκτηση Οκτώβριος 2022, από [https://en.wikipedia.org/wiki/Radial\\_basis\\_function\\_network](https://en.wikipedia.org/wiki/Radial_basis_function_network)
- Wikipedia. (2023, Ιανουάριος). Image Stitching. Ανάκτηση Ιανουάριος 2023, από [https://en.wikipedia.org/wiki/Image\\_stitching](https://en.wikipedia.org/wiki/Image_stitching)
- Wikipedia. (2023, Ιανουάριος). OpenCV. Ανάκτηση Ιανουάριος 2023, από <https://en.wikipedia.org/wiki/OpenCV>
- Wikipedia. (2023, Ιανουάριος). Raspberry Pi. Ανάκτηση Ιανουάριος 2023, από [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- Ανδρουτσόπουλος, Ί. (2022). Τεχνητή Νοημοσύνη. Αθήνα. Ανάκτηση Σεπτέμβριος 2022, από <http://www.aueb.gr/user/ion/>
- Ευάγγελος, Δ. (2015). *Αναγνώριση Προτύπων Ι*. Ανάκτηση Ιούνιος 2022, από <https://eclass.upatras.gr/courses/EE652/>
- Κουτρομπάς, Σ., & Θεωδορίδης, Κ. (2015). <https://eclass.uoa.gr>. Ανάκτηση Αύγουστος 2022, από <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewivqOucqsj5AhV1hP0HHRYjDF4QFnoECA0QAQ&url=https%3A%2F%2Feclass.uoa.gr%2Fmodules%2Fdocument%2Ffile.php%2FD67%2F%25CE%25A5%25CE%25BB%25CE%25B9%25CE%25BA%25CF%258C%2520%25CE%25B3%25CE%2>
- Μαντσακώβ, Α. (2018, Οκτώβριος). Αναγνώριση πεζών σε συνθήκες σκότους με συνελεκτικά νευρωνικά δίκτυα. Βόλος. Ανάκτηση Οκτώβριος 2022
- Τσουχνικά, Μ. (χ.χ.). Νευρωνικά δίκτυα και εφαρμογές. Ανάκτηση Οκτώβριος 2022
- Χατζηλυγερούδης, Κ. (2004). Κανόνες, Αβέβαιος Συλλογισμός και Αναπαράσταση Γνώσης σε Πλαίσια. Πάτρα. Ανάκτηση Σεπτέμβριος 2022



## Παράρτημα Ι

```
1. # YOLOv5 🚀 by Ultralytics, GPL-3.0 license
2. """
3. Run YOLOv5 detection inference on images, videos, directories, globs, YouTube, webcam, streams, etc.
4.
5. Usage - sources:
6.   $ yolo detect --weights yolov5s.pt --source 0 # webcam
7.                                     img.jpg # image
8.                                     vid.mp4 # video
9.                                     screen # screenshot
10.                                    path/ # directory
11.                                    list.txt # list of images
12.                                    list.streams # list of streams
13.                                    'path/*.jpg' # glob
14.                                    'https://youtu.be/Zgi9g1ksQhC' # YouTube
15.                                    'rtsp://example.com/media.mp4' # RTSP, RTMP, HTTP stream
16.
17. Usage - formats:
18.   $ yolo detect --weights yolov5s.pt # PyTorch
19.                                     yolov5s.torchscript # TorchScript
20.                                     yolov5s.onnx # ONNX Runtime or OpenCV DNN with --dnn
21.                                     yolov5s_openvino_model # OpenVINO
22.                                     yolov5s.engine # TensorRT
23.                                     yolov5s.mlmodel # CoreML (macOS-only)
24.                                     yolov5s_saved_model # TensorFlow SavedModel
25.                                     yolov5s.pb # TensorFlow GraphDef
26.                                     yolov5s.tflite # TensorFlow Lite
27.                                     yolov5s_edgetpu.tflite # TensorFlow Edge TPU
28.                                     yolov5s_paddle_model # PaddlePaddle
29. """
30.
31. import argparse
32. import os
33. import platform
34. import sys
35. from pathlib import Path
36.
37. import torch
38.
39. FILE = Path(__file__).resolve()
40. ROOT = FILE.parents[0] # YOLOv5 root directory
41. if str(ROOT) not in sys.path:
42.     sys.path.append(str(ROOT)) # add ROOT to PATH
43. ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative
44.
45. from yolov5.models.common import DetectMultiBackend
46. from yolov5.utils.data_loaders import IMG_FORMATS, VID_FORMATS, LoadImages, LoadScreenshots, LoadStreams
47. from yolov5.utils.general import (LOGGER, Profile, check_file, check_img_size, check_imshow, check_requirements, colorstr, cv2,
48.                                   increment_path, non_max_suppression, print_args, scale_boxes, strip_optimizer, xyxy2xywh)
49. from yolov5.utils.plots import Annotator, colors, save_one_box
50. from yolov5.utils.torch_utils import select_device, smart_inference_mode
51.
52.
53. @smart_inference_mode()
54. def run(
55.     weights='yolov5s.pt', # model path or triton URL
56.     source=ROOT / 'home/pi/freedomtech', # file/dir/URL/glob/screen/0(webcam)
57.     data=ROOT / 'data/coco128.yaml', # dataset.yaml path
58.     imgsz=None, # inference size (height, width)
59.     img=None, # inference size (pixels)
60.     conf_thres=0.25, # confidence threshold
61.     iou_thres=0.45, # NMS IOU threshold
62.     max_det=1000, # maximum detections per image
63.     device='', # cuda device, i.e. 0 or 0,1,2,3 or cpu
64.     view_img=False, # show results
65.     save_txt=False, # save results to *.txt
66.     save_conf=False, # save confidences in --save-txt labels
67.     save_crop=False, # save cropped prediction boxes
68.     nosave=False, # do not save images/videos
69.     classes=None, # filter by class: --class 0, or --class 0 2 3
70.     agnostic_nms=False, # class-agnostic NMS
71.     augment=False, # augmented inference
72.     visualize=False, # visualize features
73.     update=False, # update all models
74.     project='home/pi/yoloresult', # save results to project/name
75.     name='exp', # save results to project/name
76.     exist_ok=False, # existing project/name ok, do not increment
77.     line_thickness=3, # bounding box thickness (pixels)
78.     hide_labels=False, # hide labels
79.     hide_conf=False, # hide confidences
80.     half=False, # use FP16 half-precision inference
81.     dnn=False, # use OpenCV DNN for ONNX inference
82.     vid_stride=1, # video frame-rate stride
83. ):
84.     source = str(source)
85.     save_img = not nosave and not source.endswith('.txt') # save inference images
86.     is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
87.     is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
88.     webcam = source.isnumeric() or source.endswith('.streams') or (is_url and not is_file)
89.     screenshot = source.lower().startswith('screen')
90.     if is_url and is_file:
91.         source = check_file(source) # download
92.
93.     if imgsz is None and img is None:
94.         imgsz = 640
```

```

95.     elif img is not None:
96.         imgsz = img
97.
98.     if isinstance(imgsz, int):
99.         imgsz = [imgsz, imgsz]
100.
101.     # Directories
102.     save_dir = increment_path(Path(project) / name, exist_ok=exist_ok) # increment run
103.     (save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True) # make dir
104.
105.     # Load model
106.     device = select_device(device)
107.     model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
108.     stride, names, pt = model.stride, model.names, model.pt
109.     imgsz = check_img_size(imgsz, s=stride) # check image size
110.
111.     # Dataloader
112.     bs = 1 # batch_size
113.     if webcam:
114.         view_img = check_imshow(warn=True)
115.         dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
116.         bs = len(dataset)
117.     elif screenshot:
118.         dataset = LoadScreenshots(source, img_size=imgsz, stride=stride, auto=pt)
119.     else:
120.         dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
121.     vid_path, vid_writer = [None] * bs, [None] * bs
122.
123.     # Run inference
124.     model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz)) # warmup
125.     seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
126.     for path, im, im0s, vid_cap, s in dataset:
127.         with dt[0]:
128.             im = torch.from_numpy(im).to(model.device)
129.             im = im.half() if model.fp16 else im.float() # uint8 to fp16/32
130.             im /= 255 # 0 - 255 to 0.0 - 1.0
131.             if len(im.shape) == 3:
132.                 im = im[None] # expand for batch dim
133.
134.         # Inference
135.         with dt[1]:
136.             visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else False
137.             pred = model(im, augment=augment, visualize=visualize)
138.
139.         # NMS
140.         with dt[2]:
141.             pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms, max_det=max_det)
142.
143.         # Second-stage classifier (optional)
144.         # pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)
145.
146.         # Process predictions
147.         for i, det in enumerate(pred): # per image
148.             seen += 1
149.             if webcam: # batch_size >= 1
150.                 p, im0, frame = path[i], im0s[i].copy(), dataset.count
151.                 s += f'{i}: '
152.             else:
153.                 p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)
154.
155.             p = Path(p) # to Path
156.             save_path = str(save_dir / p.name) # im.jpg
157.             txt_path = str(save_dir / 'labels' / p.name) + ('' if dataset.mode == 'image' else f'_{frame}') # im.txt
158.             s += '%gx%g ' % im.shape[2:] # print string
159.             gn = torch.tensor(im0.shape)[[1, 0, 1, 0]] # normalization gain whwh
160.             imc = im0.copy() if save_crop else im0 # for save_crop
161.             annotator = Annotator(im0, line_width=line_thickness, example=str(names))
162.             if len(det):
163.                 # Rescale boxes from img_size to im0 size
164.                 det[:, :4] = scale_boxes(im.shape[2:], det[:, :4], im0.shape).round()
165.
166.                 # Print results
167.                 for c in det[:, 5].unique():
168.                     n = (det[:, 5] == c).sum() # detections per class
169.                     s += f"{n} {names[int(c)]}'s' * (n > 1)}, " # add to string
170.
171.                 # Write results
172.                 for *xyxy, conf, cls in reversed(det):
173.                     if save_txt: # Write to file
174.                         xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist() # normalized xywh
175.                         line = (cls, *xywh, conf) if save_conf else (cls, *xywh) # label format
176.                         with open(f'{txt_path}.txt', 'a') as f:
177.                             f.write('%g ' * len(line).rstrip() % line + '\n')
178.
179.                     if save_img or save_crop or view_img: # Add bbox to image
180.                         c = int(cls) # integer class
181.                         label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {conf:.2f}')
182.                         annotator.box_label(xyxy, label, color=colors(c, True))
183.                     if save_crop:
184.                         save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] / f'{p.stem}.jpg', BGR=True)
185.
186.         # Stream results
187.         im0 = annotator.result()
188.         if view_img:
189.             if platform.system() == 'Linux' and p not in windows:
190.                 windows.append(p)
191.                 cv2.namedWindow(str(p), cv2.WINDOW_NORMAL | cv2.WINDOW_KEEPRATIO) # allow window resize (Linux)
192.                 cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
193.                 cv2.imshow(str(p), im0)

```

```

194.         cv2.waitKey(1) # 1 millisecond
195.
196.     # Save results (image with detections)
197.     if save_img:
198.         if dataset.mode == 'image':
199.             cv2.imwrite(save_path, im0)
200.         else: # 'video' or 'stream'
201.             if vid_path[i] != save_path: # new video
202.                 vid_path[i] = save_path
203.                 if isinstance(vid_writer[i], cv2.VideoWriter):
204.                     vid_writer[i].release() # release previous video writer
205.                 if vid_cap: # video
206.                     fps = vid_cap.get(cv2.CAP_PROP_FPS)
207.                     w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
208.                     h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
209.                 else: # stream
210.                     fps, w, h = 30, im0.shape[1], im0.shape[0]
211.                 save_path = str(Path(save_path).with_suffix('.mp4')) # force *.mp4 suffix on results videos
212.                 vid_writer[i] = cv2.VideoWriter(save_path, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
213.             vid_writer[i].write(im0)
214.
215.     # Print time (inference-only)
216.     LOGGER.info(f"{s}{t} if len(det) else '(no detections), '{dt[1].dt * 1E3:.1f}ms")
217.
218.     # Print results
219.     t = tuple(x.t / seen * 1E3 for x in dt) # speeds per image
220.     LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per image at shape {(1, 3, *imgsz)}' % t)
221.     if save_txt or save_img:
222.         s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}" if save_txt else ''
223.         LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")
224.     if update:
225.         strip_optimizer(weights[0]) # update model (to fix SourceChangeWarning)
226.
227.
228. def parse_opt():
229.     parser = argparse.ArgumentParser()
230.     parser.add_argument('--weights', nargs='+', type=str, default='yolov5s.pt', help='model path or triton URL')
231.     parser.add_argument('--source', type=str, default=ROOT / 'data/images', help='file/dir/URL/glob/screen/0(webcam)')
232.     parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
233.     parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
234.     parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
235.     parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
236.     parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
237.     parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
238.     parser.add_argument('--view-img', action='store_true', help='show results')
239.     parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
240.     parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
241.     parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
242.     parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
243.     parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
244.     parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
245.     parser.add_argument('--augment', action='store_true', help='augmented inference')
246.     parser.add_argument('--visualize', action='store_true', help='visualize features')
247.     parser.add_argument('--update', action='store_true', help='update all models')
248.     parser.add_argument('--project', default='runs/detect', help='save results to project/name')
249.     parser.add_argument('--name', default='exp', help='save results to project/name')
250.     parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
251.     parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
252.     parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
253.     parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
254.     parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
255.     parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
256.     parser.add_argument('--vid-stride', type=int, default=1, help='video frame-rate stride')
257.     opt = parser.parse_args()
258.     opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1 # expand
259.     print_args(vars(opt))
260.     return opt
261.
262.
263. def main():
264.     opt = parse_opt()
265.     #check_requirements(exclude=('tensorboard', 'thop'))
266.     run(**vars(opt))
267.
268.
269. if __name__ == "__main__":
270.     main()
271.
272.

```