



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ**  
**ΠΑΡΑΓΩΓΗΣ**  
**ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΑΙ**  
**ΤΗΛΕΜΑΤΙΚΗΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανίχνευση Αντικειμένων σε Πραγματικό Χρόνο  
με Βαθιά Μάθηση και Υπολογιστική Αιχμής**

**Γκιολένα Μάριο**

71444983

**Νικολάου Γρηγόριος – Λέκτορας Εφαρμογών**

Διπλωματική εργασία υποβληθείσα στο Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και  
Παραγωγής του Πανεπιστημίου Δυτικής Αττικής

ΑΘΗΝΑ, Σεπτέμβριος 2021

Πανεπιστήμιο Δυτικής Αττικής, Μηχανικών Βιομηχανικής Σχεδίασης  
και Παραγωγής  
Γκιολένα Μάριο  
© 2021 – Με την επιφύλαξη παντός δικαιώματος



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ**  
**ΠΑΡΑΓΩΓΗΣ**  
**ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΑΙ**  
**ΤΗΛΕΜΑΤΙΚΗΣ**

Η παρούσα διπλωματική εργασία παρουσιάστηκε

από τον

**Γκιολένα Μάριο**

71444983

την 14<sup>η</sup> Οκτωβρίου 2021

Επιτροπή Εξέτασης Διπλωματικής Εργασίας

Γ. Νικολάου

.....

Σ. Βασιλειάδου

.....

Χ. Δρόσος

.....

---

Η έγκριση της διπλωματικής εργασίας δεν υποδηλοί την αποδοχή των γνώμων του συγγραφέα.  
Κατά τη συγγραφή τηρήθηκαν οι αρχές της ακαδημαϊκής δεοντολογίας.

**ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Γκιολένα Μάριο του Λουάν, με αριθμό μητρώου 71444983 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και παραγωγής δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών,





**ΠΕΡΙΛΗΨΗ****Ανίχνευση Αντικειμένων σε Πραγματικό Χρόνο  
με Βαθιά Μάθηση και Υπολογιστική Αιχμής****Γκιολένα Μάριο**

Στην παρούσα εργασία με τίτλο ‘Ανίχνευση Αντικειμένων σε Πραγματικό Χρόνο με Βαθιά Μάθηση και Υπολογιστική Αιχμής’ γίνεται η υλοποίηση εφαρμογής ανίχνευσης μάσκας πάνω σε ανθρώπινα πρόσωπα σε πραγματικό χρόνο. Αυτό επιτυγχάνεται χρησιμοποιώντας ένα συνελκτικό νευρωνικό δίκτυο YOLO 5<sup>ns</sup> έκδοσης σε συνδυασμό με τον μικρού μεγέθους υπολογιστή της NVIDIA JetSon Nano. Στο έγγραφο αναπτύσσεται λεπτομερώς η διαδικασία από αναγνώριση αντικειμένου μέσω φωτογραφιών σε ανίχνευση του προσώπου σε πραγματικό χρόνο. Αυτό επιτυγχάνεται και γίνεται πιο κατανοητό χάρη στην δομή των κεφαλαίων, η οποία κάνει αναδρομή από την αναγνώριση του αντικειμένου, την χρήση βαθιάς μάθησης και της χρησιμότητάς της, της υπολογιστικής αιχμής και τέλος την παρουσίαση των εργαλείων της εφαρμογής και την υλοποίησής της. Η εργασία κλείνει με τα συμπεράσματα και τις παρατηρήσεις από αυτή την υλοποίηση, καθώς και με πιθανές αλλαγές/βελτιώσεις για το μέλλον. Η επιλογή του θέματος εμπνεύστηκε από την σπουδαιότητα της βαθιάς μάθησης στην καθημερινή ζωή των ανθρώπων και πως αυτή τελικά χρησιμοποιεί την τεχνολογία για σύμμαχο από τα πιο απλά πράγματα της καθημερινότητας μέχρι τα πιο προκλητικά. Η ανίχνευση μάσκας για ανθρώπινα πρόσωπα σε πραγματικό χρόνο είναι μια εργασία, η οποία έχει ευρεία χρήσεων στην σημερινή κοινωνία, καθώς και υπενθυμίζει στον άνθρωπο πως παρόλο που η φύση είναι απρόβλεπτη και ασυμβίβαστη με τα ανθρώπινα σχέδια, η τεχνολογία ως εργαλείο του έδινε πάντα την δυνατότητα να μην μένει ποτέ πίσω.

Λέξεις κλειδιά

Ανίχνευση Αντικειμένων, Βαθιά Μάθηση, Συνελκτικά Νευρωνικά Δίκτυα, Υπολογιστική Αιχμής, YOLO





**ABSTRACT****Real-Time Object Detection with Deep Learning and Edge Computing****Gjolena Mario**

In the present work entitled ‘Real Time Object Detection with Deep Learning and Edge Computing’, the implementation of a mask detection application on human faces in real time is implemented. This is achieved by using a 5th Edition YOLO convolutional neural network with the compact NVIDIA JetSon Nano PC. The document describes in detail the process from object recognition through photographs to real-time face detection. This is achieved and becomes more understandable thanks to the structure of the chapters, which by order they start from object recognition, the use of deep learning and its usefulness, the edge computing and finally the presentation of the tools of the application and its implementation. The work closes with the conclusions and observations from this implementation, as well as with possible changes/improvements for the future. The choice of topic was inspired by the importance of deep learning in people's daily lives and how it ultimately uses technology as an ally from the simplest things in everyday life to the most challenging. Real-time face mask detection is a widely used task in today's society, and it reminds man that although nature is unpredictable and incompatible with human designs, technology as a tool has always enabled him to never be left behind.

## Keywords

Object Detection, Deep Learning, Convolutional Neural Networks, Edge Computing, YOLO



**ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ**

Εικόνα 1. Διαφορετικές όψεις από τρισδιάστατο αντικείμενο .....	8
Εικόνα 2. Υπολογισμός και ταίριασμα περιβάλλοντος σχημάτων .....	12
Εικόνα 3. Χαρακτηριστικά εκπαίδευσης .....	25
Εικόνα 4. Χρήση των Συνελκτικών Νευρωνικών Δικτύων για ανίχνευση αντικειμένου .....	32
Εικόνα 5. Η βασική αρχιτεκτονική του Edge Computing .....	34
Εικόνα 6. Βιβλιοθήκες για επεξεργασία δεδομένων.....	57
Εικόνα 7. Κατέβασμα Δεδομένων .....	57
Εικόνα 8. Δημιουργία λιστών τοποθεσιών (directories).....	57
Εικόνα 9. Φόρτωση εικόνων σε λίστα .....	57
Εικόνα 10. Φόρτωση ετικετών σε λίστα.....	58
Εικόνα 11. Ανάγνωση απαραίτητων δεδομένων από xml.....	58
Εικόνα 12. Απόδοση id στις αντίστοιχες κλάσεις .....	58
Εικόνα 13. Μετατροπή μορφής των δεδομένων σε YOLOv5.....	59
Εικόνα 14. Χωρισμός δεδομένων σε 90% εκπαίδευσης και 10% επικύρωσης.....	59
Εικόνα 15. Δημιουργία φακέλου για την εκπαίδευση και μεταφορά των αντίστοιχων .....	59
Εικόνα 16. Δημιουργία φακέλου για την επικύρωση και μεταφορά των αντίστοιχων .....	59
Εικόνα 17. Δημιουργία φακέλου για την εκπαίδευση και δημιουργία των αντίστοιχων .....	59
Εικόνα 18. Δημιουργία φακέλου για την επικύρωση και δημιουργία των αντίστοιχων .....	60
Εικόνα 19. Δημιουργία αρχείου YAML για ρύθμιση των δεδομένων του YOLOv5 .....	60
Εικόνα 20. Συμπίεση δεδομένων και μετακίνηση αρχείου στον φάκελο YOLOv5.....	60
Εικόνα 21. Σύνδεση και ανέβασμα συμπιεσμένου αρχείου δεδομένων στο Google Drive ....	60
Εικόνα 22. Βιβλιοθήκες για εκπαίδευση μοντέλου .....	60

Εικόνα 23. Σύνδεση του Drive στο Google Colab .....	61
Εικόνα 24. Κατέβασμα του YOLOv5 και εγκατάσταση απαιτούμενων βιβλιοθηκών .....	61
Εικόνα 25. Έλεγχος έκδοσης PyTorch και συνδεδεμένης κάρτας γραφικών.....	61
Εικόνα 26. Αποσυμπίεση των δεδομένων .....	61
Εικόνα 27. Αλλαγή τοποθεσίας εκτέλεσης εντολών στο Jupyter.....	61
Εικόνα 28. Εκπαίδευση μοντέλου για 100 εποχές.....	62
Εικόνα 29. Τέλος εκπαίδευσης .....	62
Εικόνα 30. Αποτελέσματα εκπαίδευσης σε μορφή γραφήματος.....	63
Εικόνα 31. Δείγμα εικόνων επικύρωσης με τις αντίστοιχες ετικέτες τους.....	63
Εικόνα 32. Δείγμα εικόνων επικύρωσης με αναγνωρισμένες από το μοντέλο ετικέτες .....	64
Εικόνα 33. Σύνδεση με σειριακή στο JetSon Nano .....	65
Εικόνα 34. Ρύθμιση μεγέθους αποθηκευτικού χώρου .....	65
Εικόνα 35. Ρύθμιση τρόπου σύνδεσης με το διαδίκτυο .....	66
Εικόνα 36. Ρύθμιση ονόματος στο δίκτυο .....	66
Εικόνα 37. Ρύθμιση ισχύος επεξεργαστή και κάρτας γραφικών .....	67
Εικόνα 38. Απομακρυσμένη σύνδεση μέσω πρωτοκόλλου SSH .....	67
Εικόνα 39. Έλεγχος ενημερώσεων και ενημέρωση προγραμμάτων λογισμικού .....	68
Εικόνα 40. Εγκατάσταση προαπαιτούμενων προγραμμάτων λογισμικού.....	68
Εικόνα 41. Αναβάθμιση προγράμματος διαχείρισης βιβλιοθηκών Python.....	68
Εικόνα 42. Εγκατάσταση προαπαιτούμενων βιβλιοθηκών Python .....	68
Εικόνα 43. Εγκατάσταση βιβλιοθήκης Python OpenCV.....	68
Εικόνα 44. Κατέβασμα βιβλιοθήκης PyTorch έκδοσης 1.9.0 από την NVIDIA .....	68
Εικόνα 45. Εγκατάσταση PyTorch .....	69
Εικόνα 46. Κατέβασμα TorchVision έκδοσης 0.10.0 από το GitHub.....	69

---

Εικόνα 47. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού σε TorchVision .....	69
Εικόνα 48. Δημιουργία απαραίτητης μεταβλητής για το TorchVision .....	69
Εικόνα 49. Δημιουργία απαραίτητης μεταβλητής για το NumPy .....	69
Εικόνα 50. Εγκατάσταση του TorchVision .....	69
Εικόνα 51. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού στον αρχικό.....	69
Εικόνα 52. Κατέβασμα του YOLOv5 από το GitHub.....	69
Εικόνα 53. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού σε YOLOv5 .....	70
Εικόνα 54. Εγκατάσταση προαπαιτούμενων βιβλιοθηκών για το YOLOv5.....	70
Εικόνα 55. Κατέβασμα εκπαιδευμένου μοντέλου από το Google Drive .....	70
Εικόνα 56. Αλλαγή τρέχουσας τοποθεσίας εκτέλεσης τερματικού σε ‘yolov5’ .....	70
Εικόνα 57. Δημιουργία μεταβλητής απαραίτητη για το NumPy.....	70
Εικόνα 58. Εκτέλεση εκπαιδευμένου μοντέλου σε πραγματικό χρόνο .....	70
Εικόνα 59. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με μάσκα .....	71
Εικόνα 60. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με λανθασμένα .....	71
Εικόνα 61. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με λανθασμένα .....	71
Εικόνα 62. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο χωρίς μάσκα .....	71



**ΠΕΡΙΕΧΟΜΕΝΑ**

<b>ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ .....</b>	<b>V</b>
<b>ΠΕΡΙΛΗΨΗ.....</b>	<b>VII</b>
<b>ABSTRACT.....</b>	<b>IX</b>
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ.....</b>	<b>XI</b>
<b>ΠΕΡΙΕΧΟΜΕΝΑ.....</b>	<b>XV</b>
<b>ΠΡΟΛΟΓΟΣ.....</b>	<b>1</b>
<b>1.0 ΕΙΣΑΓΩΓΗ.....</b>	<b>3</b>
<b>2.0 ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ .....</b>	<b>5</b>
<b>2.1 ΑΝΑΓΝΩΡΙΣΗ ΒΑΣΗ ΦΩΤΕΙΝΟΤΗΤΑΣ.....</b>	<b>8</b>
<b>2.2 ΑΝΑΓΝΩΡΙΣΗ ΒΑΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ .....</b>	<b>9</b>
<b>2.2.1 Ταίριασμα με Παραμόρφωση.....</b>	<b>10</b>
<b>2.2.2 Περιβάλλον Σχήματος .....</b>	<b>11</b>
<b>2.3 ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ .....</b>	<b>13</b>
<b>2.3.1 Οπτική Προβολή.....</b>	<b>13</b>
<b>2.3.2 Ανίχνευση με Βαθιά Μάθηση .....</b>	<b>15</b>
<b>3.0 ΒΑΘΙΑ ΜΑΘΗΣΗ.....</b>	<b>19</b>
<b>3.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ .....</b>	<b>19</b>
<b>3.1.1 Επιβλεπόμενη Μάθηση .....</b>	<b>21</b>
<b>3.1.2 Μη Επιβλεπόμενη Μάθηση .....</b>	<b>26</b>
<b>3.1.3 Ημι-Επιβλεπόμενη Μάθηση .....</b>	<b>27</b>
<b>3.2 ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....</b>	<b>29</b>

---

3.2.1	Ανίχνευση Αντικειμένων με CNN .....	31
4.0	ΥΠΟΛΟΓΙΣΤΙΚΗ ΑΙΧΜΗΣ.....	33
4.1	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ .....	34
4.2	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ .....	35
4.2.1	Μετάδοση .....	35
4.2.2	Αποθήκευση .....	37
4.2.3	Υπολογισμός.....	39
4.3	ΥΠΟΛΟΓΙΣΤΙΚΗ ΑΙΧΜΗΣ ΚΑΙ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ 41	
4.4	ΕΙΚΟΝΟΠΟΙΗΣΗ .....	43
4.5	ΕΦΑΡΜΟΓΕΣ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ.....	44
5.0	ΕΦΑΡΜΟΓΗ.....	47
5.1	ΥΛΙΚΟ.....	47
5.2	ΛΟΓΙΣΜΙΚΟ .....	48
5.3	ΥΛΟΠΟΙΗΣΗ.....	56
5.3.1	Προετοιμασία Δεδομένων .....	57
5.3.2	Εκπαίδευση Μοντέλου .....	60
5.3.3	Αρχικοποίηση JetSon Nano και Εκτέλεση Μοντέλου.....	64
6.0	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	73
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	75



## ΠΡΟΛΟΓΟΣ

Με την δική μου σειρά να ευχαριστήσω τον Γ. Νικολάου, επιβλέπων καθηγητή και μέντορά μου για την πολύτιμη καθοδήγησή του και την διεκπεραίωση της εργασίας αυτής. Έπειτα να ευχαριστήσω την οικογένειά μου για την ψυχολογική της υποστήριξη, όπως και δεν θα παραλείψω τους συναδέλφους μου Ι. Βραχνάκη, Α. Σιώκου και Σ. Καρατερζίδη για την αναγκαία και πολύτιμη βοήθειά τους.



## 1.0 ΕΙΣΑΓΩΓΗ

Σε μια εποχή που η Τεχνητή Νοημοσύνη είναι ολοένα και πιο έντονα στην καθημερινότητα όλων, ανεξαρτήτως προκλήσεων, πάντα βρίσκει τρόπους να είναι επίκαιρη με καινοτόμες ιδέες και εφαρμογές. Μια από αυτές τις εφαρμογές που χρήζουν σημασίας λόγω συνθηκών είναι η ανίχνευση μάσκας για ένα πρόσωπο σε πραγματικό χρόνο.

Παρόλο που η ασθένεια του covid-19 έβαλε σε παύση ένα πολύ μεγάλο μέρος του πλανήτη το περασμένο διάστημα, αυτό δεν απέτρεψε την τεχνολογία να βρει τρόπους να παραμείνει εργαλείο στα χέρια των ανθρώπων, παρά την ενθάρρυνε να γίνει κομμάτι και εύχρηστο μάλιστα στις καθημερινές λειτουργίες. Η ανίχνευση μάσκας για ανθρώπινα πρόσωπα σε πραγματικό χρόνο είναι η εφαρμογή που υλοποιήθηκε στην παρούσα εργασία δεδομένης και της επικαιρότητάς του. Ένα ζήτημα, το οποίο πολλοί επιστήμονες και ερευνητές βάλθηκαν να υλοποιήσουν, λόγω της ευχρηστίας και της λύσης που προσφέρει στην πολιτεία.

Συνοπτικά, στα επόμενα 5 κεφάλαια που ακολουθούν γίνεται μια εκτενής και λεπτομερής αναδρομή για την διαδικασία της υλοποίησης της εφαρμογής ανίχνευσης της μάσκας σε πραγματικό χρόνο. Στο πρώτο κεφάλαιο γίνεται ανάλυση για την αναγνώριση και ανίχνευση αντικειμένων. Από το πως αναγνωρίζει αντικείμενα ο υπολογιστής και τα ταξινομεί σε μοτίβα, μέχρι την εξειδικευμένη ανίχνευση αντικειμένων με σύγχρονες τεχνικές βαθιάς μάθησης.

Στο δεύτερο κεφάλαιο γίνεται ανάπτυξη και μελέτη πάνω στην βαθιά μάθηση γενικότερα. Στην χρησιμότητα της μηχανικής μάθησης, των τεχνικών εκπαίδευσης που χρησιμοποιεί και τέλος, την εξειδικευμένη κατηγορία νευρωνικών δικτύων τα οποία είναι τα συνελκτικά δίκτυα. Ένα τέτοιο δίκτυο χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.

Στο τρίτο κεφάλαιο γίνεται αναφορά στην υπολογιστική αιχμής και στην χρησιμότητά της εν γένει στον κλάδο του Διαδικτύου των Αντικειμένων, σε σχέση με την υπολογιστική νέφους. Η υπολογιστή αιχμής είναι ένας νέος κλάδος που κατακλύζει την αγορά για την ευχρηστία στον υπολογισμό δεδομένων και γενικότερα για την ευκολία που έχει στην διαχείριση μεγάλου όγκου δεδομένων.

Στο τέταρτο κεφάλαιο περιγράφεται η εφαρμογή και ο τρόπος υλοποίησής της μαζί με όλα τα εργαλεία που χρησιμοποιήθηκαν, είτε υλικού είτε λογισμικού. Την πράξη της υπολογιστικής αιχμής που αναφέρθηκε προηγουμένως την πραγματοποιεί ένας μικρός υπερυπολογιστής της NVIDIA με ονομασία JetSon Nano, ο οποίος χρησιμοποιείται για να γίνει πράξη ο έλεγχος ανίχνευσης μάσκας προσώπων σε πραγματικό χρόνο. Το μοντέλο που

υλοποιεί αυτή την διαδικασία είναι το συνελκτικό νευρωνικό δίκτυο YOLO, το οποίο επεξεργάζεται όλα τα δεδομένα και μέσω κάποιων frameworks, που αναφέρονται στο κεφάλαιο, δίνουν υπόσταση στην εφαρμογή.

Τέλος, στο πέμπτο κεφάλαιο γίνεται αναφορά στα συμπεράσματα και τις παρατηρήσεις της εργασίας, καθώς και για το πως εξελίχθηκε σε γενικές γραμμές και τι θα μπορούσε να διορθωθεί/βελτιωθεί στην πορεία. Το JetSon Nano και το YOLO ήταν τα κύρια εργαλεία «πρωταγωνιστές» αυτής της εργασίας, γι' αυτό και γίνεται λεπτομερής αναφορά στην πορεία αυτού του εγγράφου και των συμπερασμάτων συγκεκριμένα.

Με αυτόν τον τρόπο, μέσω αυτής της εργασίας, είναι εμφανής η συμβολή της τεχνολογίας και ειδικότερα των νέων επιστημών της βαθιάς μάθησης για το πως δρουν άμεσα σε μείζων θέματα της καθημερινότητας. Αυτό είναι ένα δείγμα της εξέλιξης που επιφυλάσσει για το μέλλον και των προκλήσεων που θα έρθει αντιμέτωπη.

## 2.0 ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Η διαδικασία ταυτοποίησης των αντικειμένων στον χώρο για το ανθρώπινο επίπεδο ονομάζεται όραση. Στον κόσμο της υπολογιστικής μηχανής η συγκεκριμένη λειτουργία γίνεται περίπλοκη παρόλο που έχει ως πρότυπο την ανθρώπινη φύση. Στο ερώτημα «ποιο είναι το αντικείμενο στον χώρο;» απαντάει η ‘αναγνώριση αντικειμένων’.

Η αναγνώριση αντικειμένων (object recognition) περιγράφει την αξιόπιστη ταυτοποίηση ανθρώπων, ζώων, άψυχων αντικειμένων και γενικότερων αντικειμένων του περιβάλλοντα χώρου καθώς αυτά μπορούν να απεικονιστούν. Μερικές από τις εφαρμογές που έδωσαν ώθηση προς αυτή την κατεύθυνση είναι: η βιομετρική αναγνώριση, η ανάκτηση εικόνας με βάση το περιεχόμενο και η αναγνώριση χειρόγραφου κειμένου [1]. Συγκεκριμένα:

- Η βιομετρική αναγνώριση (biometric recognition) έχει ως περιεχόμενο τα δακτυλικά αποτυπώματα, τους σαρωτές ίριδος και φωτογραφίες ανθρώπων που πρέπει να ταυτιστούν με πραγματικά πρόσωπα. Αυτά συνήθως συναντώνται σε εγκληματολογικές έρευνες και σε συστήματα υψηλής ασφαλείας. Ακόμα και στην σημερινή τεχνολογία των κινητών συσκευών είναι συχνή η χρήση αναγνώρισης προσώπου ή δακτυλικού αποτυπώματος για την ταυτοποίηση του χρήστη.
- Η ανάκτηση εικόνας με βάση το περιεχόμενο παραμένει πρόκληση ακόμη και σήμερα. Το πρόβλημα που δοκιμάζεται να επιλύσει είναι η επιτυχημένη αναγνώριση ενός υποσυνόλου εικονοστοιχείων σε μια δεδομένη εικόνα με αποτέλεσμα την κατάταξη της εικόνας σε μια συγκεκριμένη κατηγορία. Ένα απλοϊκό παράδειγμα κατανόησης είναι η αναγνώριση μιας λέξης-κλειδί σε ένα μεγάλο κείμενο από έναν κειμενογράφο. Παραδείγματος χάριν στην ανάκτηση πληροφορίας (information retrieval) σε ένα έγγραφο εάν η συχνότητα εμφάνισης της λέξης ‘αστεροειδής’ είναι μεγάλη, τότε ξέρουμε ότι το έγγραφο σχετίζεται με το διάστημα.
- Στην αναγνώριση χειρόγραφου κειμένου (handwriting recognition) ανήκουν πρώτον και κύριο οι υπογραφές και οι σφραγίδες ηλεκτρονικής μορφής από συστήματα τύπου PDA ή σε pdf, το χρηματικό ποσό σε επιταγές κ.α.

Μεταβαίνοντας πιο συγκεκριμένα σε ένα ενδιαφέρον παράδειγμα που είναι η αναγνώριση προσώπου γίνεται αντιληπτό ότι για τον υπολογιστή αποτελεί ιδιαίτερη πρόκληση. Στόχος είναι να επιτευχθεί η ακριβής ταυτοποίηση του προσώπου ανεξαρτήτως της

στάσης του στον τρισδιάστατο χώρο, τις εκφάνσεις του προσώπου και την φωτεινότητα που θα αποτυπωθεί στην κάμερα. Στις τρεις αυτές πιθανότητες μπορεί να προκληθεί αποτυχημένη σύγκριση μεταξύ των εικονοστοιχείων κι αν η εκπαίδευση δεν είναι επαρκής, τότε το αποτέλεσμα κρίνεται απρόβλεπτο. Πάνω σε αυτό το ζήτημα στάθηκε η οπτική αναγνώρισης αντικειμένων, αφού φάνηκε ότι ανεξαρτήτως αντικειμένου μπορεί να προκληθεί αυτό το φαινόμενο.

Η μηχανική μάθηση σε τέτοιου είδους συστήματα αποτελεί μονόδρομο για την επίλυση αυτών των προβλημάτων. Για την επιτυχή διαδικασία, συνήθως χρησιμοποιείται κάποια συνάρτηση η οποία μαθαίνει μέσω ενός σετ από σκηνές με εικονιζόμενα πρόσωπα ως ‘πρόσωπο’ και από σκηνές χωρίς πρόσωπα ως ‘όχι πρόσωπο’. Αυτό θεωρείται επιβλεπόμενη μάθηση ή ταξινόμηση προτύπων και εδρεύει στην μελέτη και εφαρμογή αναγνώρισης αντικειμένων. Άλλες εφαρμογές που χρησιμοποιούνται έμμεσα είναι: οι πολυεπίπεδοι αισθητήρες, τα δένδρα αποφάσεων, οι ταξινομητές πλησιέστερου γείτονα και οι μηχανές πυρήνα [1].

Μια από τις πιο απαιτητικές δοκιμασίες που κλίνεται να αντιμετωπιστεί σε μια εικόνα είναι να ταυτοποιηθούν τα υποσύνολα των εικονοστοιχείων που την απαρτίζουν. Επομένως, η εικόνα κατατέμενεται και διαχωρίζεται σε υποσύνολα τα οποία προσδιορίζονται από κλάσεις αντικειμένων. Έτσι, πλέον, είναι εύκολο να τροφοδοτηθούν αυτά τα τμήματα ή και ομάδες αυτών, ως είσοδο σε έναν ταξινομητή, ο οποίος με την σειρά του θα είναι σε θέση να αναθέσει τις ανάλογες ετικέτες στο κάθε αντικείμενο. Συνηθίζεται η προσέγγιση αυτής της κατάτμησης να γίνεται από επάνω προς τα κάτω (top down), διότι η αντίστροφη προσέγγιση έχει μικρότερο ποσοστό αξιοπιστίας. Η συγκεκριμένη μέθοδος χρησιμοποιείται σήμερα κατά κόρον από τα συστήματα αναγνώρισης αντικειμένων παρόλο που οι νέες τεχνολογίες ποικίλουν από διαφορετικές μεθοδολογίες. Αρνητικό σε αυτό το γεγονός αποτελεί η μεγάλη υπολογιστή πολυπλοκότητα που φέρει η από πάνω προς τα κάτω (top down) προσέγγιση.

Μια δεύτερη και εξίσου σημαντική πρόκληση που καλείται να αντιμετωπιστεί είναι η αποτελεσματικότητα της αναγνώρισης του αντικειμένου παρά τις απρόβλεπτες συνθήκες φωτισμού ή μερικής παραμόρφωσης της αποτύπωσης στην στάση του αντικειμένου. Σε αντίθεση με την ανθρώπινη φύση η όραση έχει μεγάλη ανοχή στο λάθος και δεν τίθεται θέμα στην αναγνώριση αντικειμένων και προσώπων παρά την όποια διαφοροποίηση θα είχε υποστεί η απεικόνιση προς αναγνώριση. Ένα τέτοιο παράδειγμα θα μπορούσε να είναι ένα γράμμα της

αλφαβήτου ή ένα ψηφίο, που όσο κι αν παραμορφωθεί μπορεί να γίνει και πάλι η ακριβής ταυτοποίηση του σε ανθρώπινο επίπεδο.

Στον κόσμο της εικόνας και των γραφικών υπάρχουν μερικοί παράγοντες που επηρεάζουν σημαντικά την όψη ενός αντικειμένου στον χώρο, άρα και την αναγνώρισή του εν γένει. Αυτοί είναι οι γεωμετρικοί μετασχηματισμοί, όπως: η μετατόπιση (translation), η κλιμάκωση (scaling) και η περιστροφή (rotation), οι οποίοι είναι υπεύθυνοι για την φυσική αναπαράσταση και την όψη των αντικειμένων στον περιβάλλοντα χώρο μιας σκηνής. Το μάτι σε σχέση με τον φακό είναι σε θέση να αναγνωρίσει την διαφορετικότητα στα πρόσωπα και να ταξινομήσει πάραυτα το εικονιζόμενο είδωλο στην κατηγορία ‘πρόσωπο’. Για να φτάσει, λοιπόν, η κάμερα σε αυτό το επίπεδο γίνεται μόνο μέσω της μάθησης. Σε αυτό, όμως μεγάλη επιρροή έχουν οι γεωμετρικοί και φυσικοί μετασχηματισμοί για το εκπαιδευτικό σύνολο, εάν δεν θεωρηθούν διαχειρίσιμοι.

Προκειμένου να εξασφαλισθεί αυτή η σταθερότητα στο εικονιζόμενο τοπίο, μια αρκετά αξιόπιστη λύση είναι να παρθεί ως δεδομένο η θέση, η κλίμακα κι ο προσανατολισμός του αντικειμένου στον χώρο. Με αυτόν τον τρόπο οι σκιάσεις κι η όψη του αντικειμένου δεν απέχουν από το σύνολο του εκπαιδευτικού σετ κι ο ταξινομητής είναι σε θέση να αποδώσει με μεγαλύτερη ακρίβεια την ανάθεση της σωστής ετικέτας. Σε άλλη περίπτωση που δεν γίνεται προεπεξεργασία της περιοχής της εικόνας συνιστώνται περισσότερα παραδείγματα από στιγμιότυπα εκπαίδευσης, προκειμένου ο ταξινομητής να είναι σε θέση μέσω της μάθησης να αναγνωρίζει για την ίδια ετικέτα αντικείμενα με μεγαλύτερη γκάμα στις διαφοροποιήσεις των αναπαραστάσεών τους.

Προσεγγίζοντας τον δισδιάστατο χώρο για ένα παράδειγμα με βάση τα παραπάνω θα μπορούσε να είναι ένα έγγραφο κειμένου. Σε αυτή την περίπτωση, τα στιγμιότυπα εκπαίδευσης και ελέγχου αντίστοιχα είναι της μορφής δισδιάστατης εικόνας με φωτεινότητα. Παρόλα αυτά, ακόμα και στον τρισδιάστατο χώρο μπορεί να γίνει μια παρόμοια προσέγγιση. Για να αναπαρασταθεί ένα τρισδιάστατο αντικείμενο χρειάζονται περισσότερες εικόνες που να απεικονίζουν τις διαφορετικές οπτικές γωνίες του αντικειμένου. Έτσι μπορούν μετέπειτα να συγκριθούν με εικόνες αντίστοιχες από το σετ εκπαίδευσης και να ταξινομηθούν αναλόγως. Εκτός των εικόνων μια επίσης συχνή μέθοδος αναγνώρισης αντικειμένων είναι μέσω ιστογραμμάτων. Το χρώμα και η υφή των αντικειμένων είναι χαρακτηριστικά τα οποία μπορούν να αναπαρασταθούν με ιστογράμματα αλλά και με εμπειρικές κατανομές συχνότητας [1]. Ένα παράδειγμα παρμένο από την πραγματικότητα είναι η αναγνώριση μιας ζέβρας. Το

ανθρώπινο μάτι μπορεί να αναγνωρίσει μια ζέβρα κι από τις εναλλαγές στις χρωματιστές λωρίδες του ζώου. Παρόμοια προσέγγιση έχει κι η υπολογιστική μηχανή, όπου μέσω των εναλλαγών των χρωμάτων του λευκού και του μαύρου μπορεί να μετρηθεί σε ποσοστό εικονοστοιχείων. Σε επόμενο βήμα συγκρίνεται αυτό το ποσοστό με τα ποσοστά των εικόνων όπου ο αλγόριθμος δεν έχει συναντήσει ξανά. Μαζί με αυτό γίνεται και σύγκριση των ιστογραμμάτων των εικόνων για την ομοιότητα της υφής των αντικειμένων και του προσανατολισμού τους στον χώρο.



**Εικόνα 1. Διαφορετικές όψεις από τρισδιάστατο αντικείμενο**

Γενικότερα στον χώρο της αναγνώρισης αντικειμένων κυριαρχούν δύο προσεγγίσεις. Αυτές είναι οι: η αναγνώριση με βάση την φωτεινότητα (brightness-base recognition) όπου οι τιμές στην φωτεινότητα των εικονοστοιχείων είναι καθορισμένη και η αναγνώριση με βάση τα χαρακτηριστικά (featured-based recognition), όπου γίνονται αναγνωρίσιμα τα χαρακτηριστικά του αντικειμένου πάνω στο πεδίο της σκηνής. Από γεωγραφικής απόψεως τέτοια θα μπορούσε να είναι οι ακμές του αντικειμένου ή και κομβικά σημεία.

## 2.1 ΑΝΑΓΝΩΡΙΣΗ ΒΑΣΗ ΦΩΤΕΙΝΟΤΗΤΑΣ

Στην περίπτωση όπου θεωρείται βασικό χαρακτηριστικό η φωτεινότητα στο στιγμιότυπο του αντικειμένου ξετυλίγονται δύο ισοπίθανα σενάρια. Το πρώτο σενάριο θέτει για δεδομένο αντικείμενο να υπάρχει ένα σύνολο εικονοστοιχείων σε κάθε στιγμιότυπο το οποίο να το αποτυπώνει. Για κάθε τέτοιο σύνολο εικονοστοιχείων οι τιμές της φωτεινότητας είναι ίδιες. Το δεύτερο σενάριο θέτει να γίνει επεξεργασία της εικόνας μέσω γραμμικών φίλτρων και το αποτέλεσμα σύγκρισης για το σύνολο των εικονοστοιχείων να είναι οι διαφορετικές τιμές στην φωτεινότητα. Το συγκεκριμένο σενάριο είναι ευρέως χρησιμοποιούμενο κυρίως στην αναγνώριση χειρόγραφων ψηφίων.



Για την υλοποίηση αυτών των μεθόδων έχουν χρησιμοποιηθεί κατά καιρούς πολλά στατιστικά μοντέλα για αναγνώριση και ανίχνευση προσώπων από βάσεις δεδομένων, αλλά μεγαλύτερη απήχηση έχουν τα μοντέλα των νευρωνικών δικτύων. Ένα παράδειγμα χρήσης είναι ένα απλό νευρωνικό δίκτυο το οποίο δέχεται στην είσοδο του το στιγμιότυπο χωρίς να έχουν επεξεργασθεί τα δεδομένα των εικονοστοιχείων, δέντρα αποφάσεων με οριζόμενα χαρακτηριστικά και μοντέλα Bayes με χαρακτηριστικά που φέρουν κυματομορφές.

Ένα σημαντικό μειονέκτημα στην περίπτωση των ανεπεξέργαστων εικονοστοιχείων είναι πως όταν εισέρχονται στην είσοδο του νευρωνίου ως διάνυσμα χαρακτηριστικών, τότε δημιουργείται έντονος πλεονασμός. Αυτό σημαίνει ότι τα κοντινά μεταξύ τους εικονοστοιχεία δεν είναι εύκολα διαχωρίσιμα λόγω της ίδιας ποσότητας φωτισμού και της παρόμοιας γεωμετρίας. Συνηθίζεται να χρησιμοποιούνται τεχνικές ελάττωσης των δεδομένων για την αντιμετώπιση αυτού του προβλήματος. Τέτοιες τεχνικές που μετριάζουν τις διαστάσεις των διανυσμάτων χαρακτηριστικών έχουν δείξει ότι υπάρχει μεγαλύτερη ακρίβεια και ταχύτητα στην αναγνώριση προσώπων και αντικειμένων.

## 2.2 ΑΝΑΓΝΩΡΙΣΗ ΒΑΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

Πέρα από το χαρακτηριστικό της φωτεινότητας στα ανεπεξέργαστα εικονοστοιχεία χαρακτηριστικό ίδιας δυναμικής μπορεί να θεωρηθεί και η χωρική διάταξη του αντικειμένου πάνω στην εικόνα. Αυτό ερμηνεύεται ως συγκεκριμένες περιοχές του αντικειμένου και οι ακμές. Οι ακμές είναι μεγαλύτερης σημασίας επειδή υπερισχύουν για δύο λόγους. Ο πρώτος λόγος είναι ότι το πλήθος των ακμών μπροστά στις εικονοστοιχείων είναι πολύ πιο μικρό κι αυτό το κάνει εμφανώς πιο διαχειρίσιμο. Ο δεύτερος λόγος είναι ότι ανεξαρτήτως διαφοροποιήσεων στις τιμές της φωτεινότητας στα εικονοστοιχεία, οι ακμές είναι κατά μεγάλο ποσοστό πάντα διακριτές στις ίδιες θέσεις χωρίς μεγάλες διαφορές. Επομένως, έχουν την απαραίτητη σταθερότητα για να θεωρούνται βασικό χαρακτηριστικό. Οι ακμές έχουν την ιδιότητα να χρησιμοποιούνται με διάφορους τρόπους αναλόγως τον χώρο μελέτης. Στον δισδιάστατο χώρο αποτελούν απλά σημεία ενώ στον τρισδιάστατο χώρο απαρτίζουν περιοχές. Η διαφορά των ακμών στην συγκεκριμένη προσέγγιση βάση χαρακτηριστικών σε σχέση με την προηγούμενη βάση φωτεινότητας είναι ότι πλέον καθορίζουν μια θέση στον χώρο  $(x, y) -$

ως χαρακτηριστικό, ενώ πριν ήταν απλώς ένας δείκτης επιρροής για μια συνιστώσα του διανύσματος χαρακτηριστικών.

Σε αντίθεση με οποιοδήποτε άλλο στιγμιότυπο εικόνας οι γραμμικές απεικονίσεις είναι πολύ πιο εύκολες στην αναγνώριση για μια μηχανή κι αυτό εξαιτίας της διάταξης των ακμών που αποτελούν χαρακτηριστικό για ένα αντικείμενο. Έχοντας ένα δοσμένο σύνολο από πρότυπα αντικειμένων αποθηκευμένο και χρησιμοποιώντας τον ταξινομητή του πλησιέστερου γείτονα μπορεί να υπολογισθεί η ‘απόσταση’ ανάμεσα στις διαμορφώσεις των ακμών του αγνώστου αντικειμένου της εικόνας με τα δοσμένα αντικείμενα του εκπαιδευτικού συνόλου. Ο ταξινομητής κάνει έλεγχο με βάση τα πρότυπα που είναι αποθηκευμένα σε βιβλιοθήκες και επιλέγει την ομάδα του αντικειμένου με την μικρότερη ‘απόσταση’ από το εξεταζόμενο αντικείμενο

### 2.2.1 Ταίριασμα με Παραμόρφωση

Η ιδέα του ταιριάσματος με παραμόρφωση (deformable matching) είναι μια από τις κυρίαρχες ιδέες που επικράτησαν για τον ορισμό της απόστασης μεταξύ των εικόνων. Χάρη στα σύγχρονα γραφικά υπολογιστών η έννοια της μεταμόρφωσης (morphing) χρησιμοποιείται στην ευθυγράμμιση πανομοιότυπων σχημάτων μέσω μετασχηματισμών των συντεταγμένων. Η παραπάνω μέθοδος της ομοιότητας των σχημάτων είναι μια διαδικασία τριών σταδίων, η οποία αποτελείται από τα εξής βήματα:

1. Επίλυση του προβλήματος της αντιστοιχίας των δύο σχημάτων.
2. Χρήση αυτής της αντιστοιχίας για τον υπολογισμό ενός μετασχηματισμού ευθυγράμμισης.
3. Πρώτα υπολογισμός του σφάλματος ταύτισης μεταξύ των σημείων των δύο σχημάτων συνδυαστικά με έναν δείκτη καταμέτρησης της τάξης του μεγέθους των μετασχηματισμών ευθυγράμμισης. Έπειτα, υπολογισμός της απόστασης των δύο σχημάτων ως άθροισμα των προηγούμενων όρων.

Για να αναπαρασταθεί ένα σχήμα υπολογίζεται μέσω του σχήματος ή του περιγράμματός του ένα διακριτό σύνολο σημείων το οποίο γίνεται δειγματοληπτικά. Τα συγκεκριμένα σημεία μέσω ενός ανιχνευτή ακμών μπορούν να θεωρηθούν ως οι θέσεις των εικονοστοιχείων για αυτές τις ακμές. Έτσι μπορεί να αναπαρασταθεί ένα σύνολο  $N$  σημείων  $\{p_1, \dots, p_N\}$ . Στην Εικόνα 2.(a) και (b) εμφανίζονται τα δείγματα των δύο σχημάτων.

## 2.2.2 Περιβάλλον Σχήματος

Υποθέτοντας ένα συγκεκριμένο σημείο του παραπάνω συνόλου  $p_i$  σε συνάρτηση με το σύνολο των σημείων που ξεκινούν από αυτό το σημείο και καταλήγουν σε όλα τα δείγματα σημείων του σχήματος. Η διαμόρφωση ολόκληρου του σχήματος μπορεί, λοιπόν, να εκφραστεί ως προς το σημείο αναφοράς, μέσω αυτών των διανυσμάτων. Όλο αυτό υποστηρίζει την εξής ιδέα: το περιβάλλον σχήματος (shape context) είναι ένας περιγραφέας ο οποίος σχετίζεται με κάθε σημείο του δείγματος. Αυτός με την σειρά του δίνει την περιγραφή για τη χονδρική διάταξη του υπόλοιπου σχήματος ως προς το σημείο. Ειδικότερα, το περιβάλλον σχήματος για το  $p_i$  ορίζεται ως ένα χονδροειδές χωρικό ιστόγραμμα  $h_i$  με αντίστοιχες συντεταγμένες  $p_k - p_i$  για τα υπόλοιπα  $N - 1$  σημεία  $p_k$ . Για να εξασφαλισθεί ότι ο περιγραφέας θα είναι πιο επιρρεπής στην αναγνώριση των διαφορών μεταξύ των κοντινών εικονοστοιχείων χρησιμοποιείται ένα λογαριθμικό σύστημα πολικών συντεταγμένων για τον ορισμό των ομάδων του ιστογράμματος. Ένα τέτοιο παράδειγμα βρίσκεται στην Εικόνα 2.(c).

Δίνεται μεγαλύτερη έμφαση στην αμεταβλησία της μετατόπισης, διότι είναι εγγενής στον ορισμό με το περιβάλλον σχήματος από την στιγμή που όλες οι μετρήσεις λαμβάνονται σε σχέση με τα σημεία του αντικειμένου. Για να επιτευχθεί η συγκεκριμένη αμεταβλησία και στην κλιμάκωση (scaling) πρέπει όλες οι ακτινικές αποστάσεις να κανονικοποιηθούν βάση της μέσης απόστασης ανά ζευγάρια σημείων.

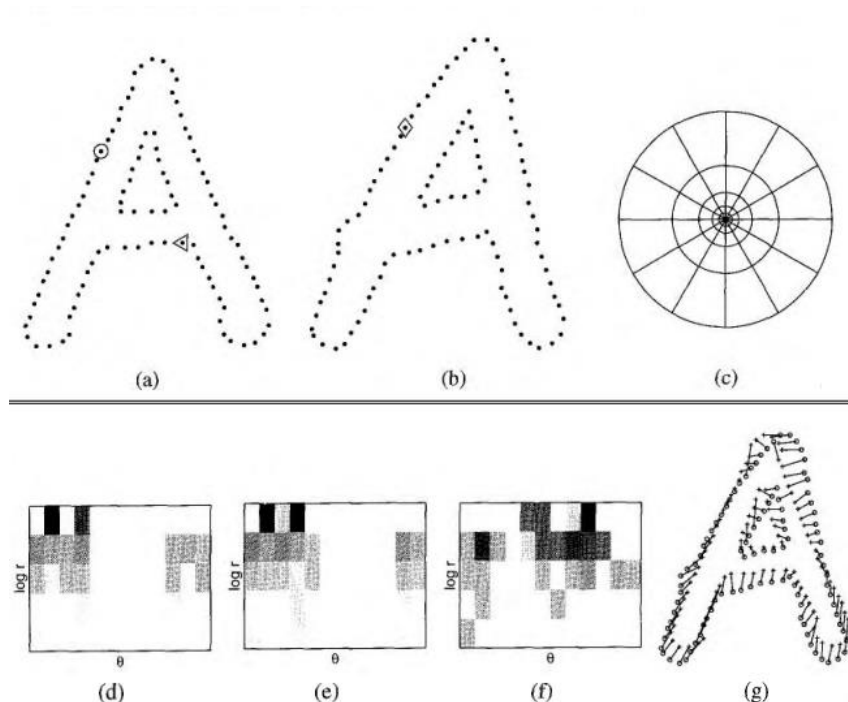
Στην Εικόνα 2.(a) και (b) εμφανίζεται το πρόβλημα της αντιστοιχίας μεταξύ δύο εικόνων πανομοιότυπων σχημάτων με ελάχιστες διαφορές στο οποίο τα περιβάλλοντα σχημάτων βοηθούν στην επίλυσή του. Για ένα δοσμένο σχήμα  $S$  τα περιβάλλοντα σχημάτων διαφέρουν για τα σημεία του. Όμως για τα αντίστοιχα (ομόλογα) σημεία σε δύο πανομοιότυπα σχήματα  $S$  και  $S'$  θα έχουν την τάση να έχουν παρόμοια περιβάλλοντα σχήματος. Έτσι για τα αντίστοιχα σημεία δύο σχημάτων μπορεί να οριστεί το πρόβλημα της εύρεσης ζευγαριών που έχουν παρόμοιο περιβάλλον σχήματος.

Για παράδειγμα, έστω για το πρώτο σχήμα ένα σημείο  $p_i$  και για το δεύτερο σχήμα ένα σημείο  $q_j$ . Το κόστος ταιριάσματος για τα δύο αυτά σημεία δίνεται από τον τύπο  $C_{ij} = p_i - q_j$ . Από την στιγμή που τα περιβάλλοντα σχήματος είναι κατανομές που απεικονίζονται ως ιστογράμματα λογικό επόμενο είναι να χρησιμοποιηθεί η απόσταση  $\chi^2$ :

$$C_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)},$$

όπου τα  $h_i(k)$  και  $h_j(k)$  υποδηλώνουν την ομάδα  $k$  των κανονικοποιημένων ιστογραμμάτων των  $p_i$  και  $q_j$  αντίστοιχα. Για να ελαχιστοποιηθεί το ολικό κόστος ταιριάσματος με βάση τον περιορισμό ότι το ταιρίασμα θα είναι αμφιμονοσήμαντο (ένα προς ένα) έστω ως δεδομένο το σύνολο τιμών κόστους  $C_{ij}$  μεταξύ όλων των σημείων  $i$  ανά ζεύγη στο πρώτο σχήμα και  $j$  στο δεύτερο. Η συγκεκριμένη προϋπόθεση είναι μία περίπτωση από το πρόβλημα του σταθμισμένου διμερούς ταιριάσματος (weighted bipartite matching).

Με γνωστές τις αντιστοιχίσεις των σημείων από τα δείγματα και μέσω της εκτίμησης ενός μετασχηματισμού ευθυγράμμισης που θα απεικονίζει το ένα σχήμα εφαπτόμενο στο άλλο αυτή η αντιστοιχία μπορεί να επεκταθεί και στο σύνολο του σχήματος. Σε αυτή την διαδικασία διακρίνονται οι κανονικοποιημένες καμπύλες thin plate splines. Από την στιγμή που τα σχήμα ευθυγραμμιστούν, ο υπολογισμός για τον βαθμό ομοιότητας είναι σχεδόν άμεσος. Για αντίστοιχα σημεία μαζί με την καμπυλική ενέργεια που σχετίζεται με την καμπύλη spline, η απόσταση μεταξύ δύο σχημάτων ορίζεται ως το σταθμισμένο άθροισμα μεταξύ των αποστάσεων των περιβάλλοντων σχημάτων. Βάση αυτού του μέτρου απόστασης χρησιμοποιώντας έναν απλό ταξινομητή πλησιέστερου γείτονα μπορεί να επιλυθεί το πρόβλημα της αναγνώρισης αντικειμένων.



Εικόνα 2. Υπολογισμός και ταιρίασμα περιβάλλοντος σχημάτων. (a, b) Δείγματα σημείων ακμών δύο σχημάτων. (c) Διάγραμμα του λογαριθμικού πολικού ιστογράμματος που χρησιμοποιήθηκε στον

υπολογισμό των περιβαλλόντων των σχημάτων. Χρησιμοποιούνται 5 ομάδες για το  $\log r$  και 12 ομάδες για το  $\theta$ . (d-f) Παραδείγματα περιβαλλόντων σχημάτων για τα δείγματα αναφοράς που ήταν σημειωμένα με τα σύμβολα  $\circ$ ,  $\diamond$  και  $\triangleleft$  στις Εικόνες (a, b). Κάθε περιβάλλον σχήματος είναι ένα λογαριθμικό πολικό ιστόγραμμα με τις συντεταγμένες για τα υπόλοιπα σημεία του συνόλου, με χρήση του σημείου αναφοράς ως αρχής των αξόνων. (Τα σκούρα πλαίσια σημαίνουν περισσότερα σημεία στην αντίστοιχη ομάδα). Προσοχή στην οπτική ομοιότητα των περιβαλλόντων σχημάτων για τα  $\circ$  και  $\diamond$ , τα οποία υπολογίστηκαν για σχετικά παρόμοια σημεία στα δύο σχήματα. Αντίθετα, το περιβάλλον σχήματος για το  $\triangleleft$  είναι πολύ διαφορετικό. (g) Αντιστοιχίες μεταξύ των (a) και (b) που βρέθηκαν με χρήση του διμερούς ταιριάσματος, με κόστος να καθορίζεται από την απόσταση  $\chi^2$  μεταξύ των ιστογραμμάτων [1].

## 2.3 ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Μετά από την επιτυχή αναγνώριση των αντικειμένων στα στιγμιότυπα των εικόνων την απάντηση στο ερώτημα «πού βρίσκεται το αντικείμενο στον χώρο;» έρχεται να δώσει η ‘ανίχνευση αντικειμένων’. Η ανίχνευση αντικειμένου είναι η διαδικασία προσδιορισμού του στιγμιότυπου της κλάσης στην οποία ανήκει το αντικείμενο και της εκτίμησης της θέσης του αντικειμένου εξάγοντας το πλαίσιο οριοθέτησης (ετικέτα) γύρω από το αντικείμενο. Η ανίχνευση ενός μεμονωμένου στιγμιότυπου της κλάσης από την εικόνα ονομάζεται ανίχνευση αντικειμένου μεμονωμένης κλάσης (single class object detection), ενώ η ανίχνευση των κλάσεων όλων των αντικειμένων που υπάρχουν στην εικόνα είναι γνωστή ως ανίχνευση αντικειμένου πολλαπλών κλάσεων (multi class object detection). Μερικές προκλήσεις που χρίζονται χειρισμού κατά την εκτέλεση ανίχνευσης ενός αντικειμένου είναι η μερική/πλήρης συγκάλυψη του αντικειμένου, διαφορετικές συνθήκες φωτισμού, στάσεις στην κάμερα, κλιμάκωση, κλπ. Γενικότερα, η ανίχνευση αντικειμένων είναι το πρώτο βήμα σε οποιαδήποτε δραστηριότητα οπτικής αναγνώρισης [2].

### 2.3.1 Οπτική Προβολή

Η οπτική προβολή περιγράφεται ως μια αντιληπτική ιδιότητα που κάνει μια περιοχή της εικόνας να ξεχωρίζει σε σχέση με το περιβάλλον της και να τραβά την προσοχή του παρατηρητή [3]. Προκειμένου ένα σύστημα μηχανικής όρασης να μπορεί να αναγνωρίζει ένα άγνωστο αντικείμενο χωρίς ετικέτα χρειάζεται η προσέγγιση της εκπαίδευσής του να γίνει όπως αυτή της ανθρώπινης φύσης.

Όπως και στο ανθρώπινο επίπεδο, έτσι κι η μηχανή πρέπει να είναι σε θέση να επιλέξει μέσα από την απορρέουσα πληροφορία των αισθητήρων την σχετική με την εικόνα και κατά συνέπεια με το αντικείμενο. Εάν η παρουσίαση ενός αντικειμένου γίνεται με σαφή τρόπο, δηλαδή ότι είναι οπτικά διακριτό σε σχέση με την υπόλοιπη σκηνή, μπορεί να επιτρέψει την μη επιβλεπόμενη εξαγωγή ενός τέτοιου αντικειμένου από την εικόνα που γίνεται αντιληπτή, έτσι ώστε να μπορεί να χρησιμοποιηθεί ως παράδειγμα μάθησης για μια μηχανή ανίχνευσης αντικειμένου.

Η ιδέα της οπτικής προβολής εμπνεύστηκε από την λειτουργία των πρώτων σταδίων επεξεργασίας του ανθρώπινου φακού. Στα πρωταρχικά στάδια της λειτουργίας της όρασης το μάτι εστιάζει ασυνείδητα, μεθόδου από κάτω προς τα πάνω (bottom-up), σε οπτικά ελκυστικές περιοχές της εικόνας προς αναγνώριση. Σαν όρος ‘οπτική ελκυστικότητα’ μπορούν να θεωρηθούν χαρακτηριστικά, όπως η ένταση, η αντίθεση και η κίνηση. Παρόλο που υπάρχουν αποκλειστικά βιολογικά βασισμένες προσεγγίσεις για τον υπολογισμό της οπτικής προβολής, τα περισσότερα από τα υπάρχοντα έργα δεν ισχυρίζονται ότι είναι βιολογικά εύλογα. Αντιθέτως, γίνεται χρήση καθαρά υπολογιστικών τεχνικών προκειμένου να επιτευχθεί αυτός ο στόχος.

Σε ένα από τα πρώτα έργα που πρωτοεμφανίστηκε η οπτική προβολή χρησιμοποιήθηκε μια προσέγγιση βασισμένη σε έναν υπολογισμό αντίθεσης κέντρου-περιβάλλοντος χώρου (center-surround contrast calculation) χρησιμοποιώντας την Διαφορά των Γκαουσιανών (Difference of Gaussians) [4]. Μερικές άλλες κοινότερες τεχνικές για τον υπολογισμό της οπτικής προβολής περιλαμβάνουν τις εξής:

- Τυχαίο βήδισμα βασισμένο σε γράφημα (graph based random walk) [5].
- Αποστάσεις χαρακτηριστικών κέντρου-περιβάλλοντος χώρου (center-surround feature distances) [6].
- Αντίθεση πολλαπλών κλιμακώσεων. Ιστόγραμμα κέντρου-περιβάλλοντος χώρου και χωρική κατανομή χρώματος [7].
- Χαρακτηριστικά χρώματος και φωτεινότητας [3].

Μια λιγότερο γνωστή προσέγγιση χρησιμοποιεί αναπαράσταση και τμηματοποίηση υπεργραφικής ευαισθησίας-περιεχομένου αντί να χρησιμοποιεί πιο κλασσικά έτοιμα χαρακτηριστικά και παραμέτρους για όλες τις εικόνες [8].

Στην επεξεργασία εικόνας, η ταυτοποίηση περιοχών οπτικής προβολής μιας εικόνας χρησιμοποιούνται για διάφορες λειτουργίες όπως η ευφυής αλλαγή μεγέθους εικόνας [9],

προσαρμοστική εμφάνιση εικόνας σε οθόνες μικρών συσκευών [10], βελτίωση της ανίχνευσης και αναγνώρισης αντικειμένων [11], ανάκτηση εικόνας βάσει περιεχομένου και προσαρμοστική συμπίεση εικόνας ή περιήγηση στη συλλογή εικόνων κλπ. Ανάλογα με τη συγκεκριμένη τεχνική, πολλές προσεγγίσεις έχουν ως έξοδο έναν χάρτη υπεροχής (saliency map), ο οποίος είναι μια εικόνα της οποίας οι εντάσεις των εικονοστοιχείων συσχετίζονται με την εμβέλεια των αντίστοιχων εικονοστοιχείων της αρχικής εικόνας [7]. Η επιλογή των περιοχών μεγάλης υπεροχής από τον χάρτη υπεροχής με εφαρμογή ενός κατωφλίου ή ενός αλγορίθμου τμηματοποίησης αναλύεται στη συνέχεια. Καταλήγει σε εξαγωγή ενός οπτικά σημαντικού αντικειμένου ή μιας ομάδας αντικειμένων αντί σε μόνο ένα σημασιολογικά μη-φασικό κομμάτι της εικόνας. Αυτήν την ιδιότητα εκμεταλλεύτηκαν διάφοροι συγγραφείς, όπως των [12]. Στο έργο τους ένας βιολογικά κινούμενος ανιχνευτής υπεροχής χρησιμοποιείται μαζί με έναν αλγόριθμο μη-επιβλεπόμενης ομαδοποίησης για να ομαδοποιήσουν μαζί εικόνες που περιέχουν οπτικά παρόμοια αντικείμενα. Αξιοσημείωτα στο έργο [13] παρουσιάζεται ένα καθαρά από κάτω προς τα πάνω (bottom-up) σύστημα που βασίζεται στην οπτική προσοχή, το οποίο διερευνά τη σκοπιμότητα της μη επιβλεπόμενης εκμάθησης αντικειμένων από εικόνες χωρίς ετικέτα. Τα πειράματα διεξάγονται με επιτυχία από τους συγγραφείς του σε πραγματικές φωτογραφίες υψηλής ανάλυσης και μέσω ενός ρομπότ-κινητού εξοπλισμένο με κάμερα, όπου εμφανίζεται η ικανότητα εκμάθησης αντικειμένων-ορόσημων κατά την πλοήγησή του σε ένα εσωτερικό περιβάλλον.

Η εξαγωγή ενός αντικειμένου ενδιαφέροντος οδηγείται από την υπεροχή (saliency) του, επομένως η ακριβής και γρήγορη ανίχνευση περιοχών υπεροχής είναι ζωτικής σημασίας. Παρόλο που υπάρχουν πολλές προσεγγίσεις σε αυτό το ζήτημα, δεν είναι όλες κατάλληλες για την υπολογιστική μηχανή του παραδείγματος. Συχνά δεν διαθέτουν ακρίβεια ή καλή ανάλυση στον τομέα συχνότητας, είναι σε θέση να εξαγάγουν μόνο ένα από τα αντικείμενα μεγάλης υπεροχής από την εικόνα ή είναι υπολογιστικά πολύ βαριά για να χρησιμοποιηθούν σε πραγματικό χρόνο [14]. Μια σύγκριση ορισμένων αλγορίθμων τελευταίας τεχνολογίας με αυτούς τους όρους μπορεί να βρεθεί στο [6].

### 2.3.2 Ανίχνευση με Βαθιά Μάθηση

Πρόσφατα, οι τεχνικές βαθιάς μάθησης έχουν αναδειχθεί ως ισχυρές μέθοδοι για την εκμάθηση αναπαράστασης χαρακτηριστικών αυτόματα μέσα από δεδομένα. Ειδικότερα, αυτές οι τεχνικές έχουν προσφέρει σημαντικές βελτιώσεις στην ανίχνευση αντικειμένων [15] [16].

Η ανίχνευση αντικειμένων μπορεί να ομαδοποιηθεί σε έναν από τους δύο τύπους: ανίχνευση συγκεκριμένων στιγμιότυπων έναντι της ανίχνευσης ευρέων κατηγοριών [17]. Ο πρώτος τύπος στοχεύει στον εντοπισμό στιγμιότυπων ενός συγκεκριμένου αντικειμένου (όπως το πρόσωπο του Ντόναλντ Τραμπ, ο Πύργος του Άιφελ ή ο σκύλος ενός γείτονα), ουσιαστικά ως πρόβλημα αντιστοίχισης.

Ο στόχος του δεύτερου τύπου είναι να ανιχνεύσει στιγμιότυπα (που συνήθως να μην έχει ξαναδεί προηγουμένως) από ορισμένες προκαθορισμένες κατηγορίες αντικειμένων (για παράδειγμα άνθρωπος, αυτοκίνητα, ποδήλατα και σκύλους). Ιστορικά, μεγάλο μέρος της προσπάθειας στον τομέα της ανίχνευσης αντικειμένων επικεντρώθηκε στην ανίχνευση μιας μεμονωμένης κατηγορίας (συνήθως προσώπων και πεζών) ή μερικών συγκεκριμένων κατηγοριών. Αντίθετα, τα τελευταία χρόνια, η ερευνητική κοινότητα έχει αρχίσει να κινείται προς τον πιο απαιτητικό στόχο της δημιουργίας συστημάτων ανίχνευσης αντικειμένων γενικής χρήσης, όπου το εύρος της ικανότητας ανίχνευσης αντικειμένων ανταγωνίζεται αυτό των ανθρώπων [18].

Η ανίχνευση αντικειμένων που βασίζεται στη βαθιά μάθηση είναι ένα πεδίο με γρήγορη ανάπτυξη. Τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNN) είναι ένα είδος νευρωνικών δικτύων πολλαπλών στρωμάτων που έχουν σχεδιαστεί για να αναγνωρίζουν οπτικά μοτίβα απευθείας από εικόνες εικονοστοιχείων και πολλές αρχιτεκτονικές του CNN όπως το AlexNet [19], το VGG [20], το GoogleNet [21], το ResNet [22], το MobileNets [23] έχουν αναπτυχθεί. Εκτός από την ανάπτυξη των CNN, πολλοί αλγόριθμοι ανίχνευσης αντικειμένων που βασίζονται στο CNN, όπως το R-CNN [24], το YOLO (You Look Only Once) [25] και το SSD (Single Shot MultiBox Detector) [26] έχουν προταθεί.

Σε αυτές τις μεθόδους, τα αντικείμενα ανιχνεύονται σε πραγματικό χρόνο, όχι ως φιγούρες αλλά ως ένα πλαίσιο οριοθέτησης (ετικέτα) που περιβάλλει το αντικείμενο. Όταν χρησιμοποιείται ανίχνευση αντικειμένου βάσει του CNN για τον προσδιορισμό της περιοχής ενδιαφέροντος (region of interest – ROI), η περιοχή ενδιαφέροντος (ROI) ορίζεται ως ένα ορθογώνιο πλαίσιο αντί για την φιγούρα του αντικειμένου. Ωστόσο, έχει αναφερθεί ότι η περιοχή ενδιαφέροντος (ROI) δεν χρειάζεται να είναι η φιγούρα του αντικειμένου στόχου και μπορεί να είναι ένα πλαίσιο οριοθέτησης που καλύπτει το μεγαλύτερο μέρος του και επαρκώς [27]. Έτσι, πιστεύεται ότι η ανίχνευση αντικειμένων που βασίζεται στη βαθιά μάθηση είναι αποτελεσματική για τον προσδιορισμό της περιοχής ενδιαφέροντος (ROI) [28].



Για την επεξεργασία σε πραγματικό χρόνο γενικά, η ταχύτητα ανίχνευσης αντικειμένων είναι εξίσου σημαντική με την ακρίβειά της. Ως εκ τούτου, ο συνδυασμός του MobileNet για την αρχιτεκτονική του CNN και το SSD-300 για το μοντέλο ανίχνευσης αντικειμένων (MobileNetSSD) χρησιμοποιείται ως τεχνική ανίχνευσης κινούμενων αντικειμένων, όπως ο [29] ανέφερε ότι το ταχύτερο μοντέλο εκείνη την εποχή ήταν το MobileNetSSD, λαμβάνοντας υπόψη την ισορροπία της ταχύτητας, της ακρίβειας και της χρήσης μνήμης με βάση τον στόχο.

Αναλόγως με το εάν χρησιμοποιείται η βαθιά μάθηση ή όχι, οι μέθοδοι ανίχνευσης αντικειμένων μπορούν να χωριστούν σε δύο κύριες κατηγορίες: τις χειρωνακτικές μεθόδους που βασίζονται σε χαρακτηριστικά και τις μεθόδους βαθιάς μάθησης [30]. Στην πρώτη δεκαετία του 21<sup>ου</sup> αιώνα, οι παραδοσιακές χειρωνακτικές μέθοδοι που βασίζονται σε χαρακτηριστικά είναι το κύριο ρεύμα επιρροής. Ωστόσο, για το 2010, η απόδοση της ανίχνευσης αντικειμένων τείνει να είναι σταθερή. Παρόλο που εξακολουθούν να προτείνονται πολλές μέθοδοι, η βελτίωση της απόδοσης είναι σχετικά περιορισμένη. Εν τω μεταξύ, η βαθιά μάθηση αρχίζει να δείχνει ανώτερη απόδοση για ορισμένες περιοχές όρασης του υπολογιστή (π.χ., ταξινόμηση εικόνας [22]).

Σύμφωνα με τον διάλογο βαθιάς ανίχνευσης αντικειμένων, οι υπάρχοντες μέθοδοι μπορούν να χωριστούν σε δύο κύριες κατηγορίες: τις μεθόδους δύο σταδίων [31] και τις μεθόδους ενός σταδίου [32]. Οι μέθοδοι δύο σταδίων δημιουργούν αρχικά κάποια προτεινόμενα υποψήφια αντικείμενα και στη συνέχεια κατατάσσουν αυτά τα αντικείμενα σε συγκεκριμένες κατηγορίες. Οι μέθοδοι ενός σταδίου εξάγουν και ταξινομούν ταυτόχρονα όλα τα υποψήφια αντικειμένων. Σε γενικές γραμμές, οι μέθοδοι δύο σταδίων έχουν σχετικά μικρότερη ταχύτητα ανίχνευσης και υψηλότερη ακρίβεια ανίχνευσης, ενώ οι μέθοδοι ενός σταδίου έχουν πολύ ταχύτερη ταχύτητα ανίχνευσης και παρόμοια ακρίβεια ανίχνευσης [33].

Σε αντίθεση από την πολλαπλών σταδίων διαδικασία των μεθόδων δύο σταδίων, οι μέθοδοι ενός σταδίου στοχεύουν στην ταυτόχρονη αναγνώριση της κατηγορίας αντικειμένου και της θέσης αντικειμένου. Μεταξύ των μεθόδων ενός σταδίου, το OverFeat [34], το YOLO, το SSD και το RetinaNet [32] είναι οι αντιπροσωπευτικές μέθοδοι. Στην παρούσα εργασία χρησιμοποιήθηκε από τις μεθόδους ενός σταδίου το YOLO που θα αναλυθεί και στην συνέχεια.



### 3.0 ΒΑΘΙΑ ΜΑΘΗΣΗ

Η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης, το οποίο είναι ουσιαστικά ένα νευρωνικό δίκτυο με τρία ή περισσότερα επίπεδα. Αυτά τα νευρωνικά δίκτυα προσπαθούν να προσομοιώσουν τη συμπεριφορά του ανθρώπινου εγκεφάλου - αν και απέχει πολύ από το να φτάσει την ικανότητά του - επιτρέποντάς του να «μαθαίνει» μέσα από μεγάλες ποσότητες δεδομένων. Ενώ ένα νευρωνικό δίκτυο με ένα μόνο στρώμα μπορεί να κάνει κατά προσέγγιση προβλέψεις, επιπλέον κρυμμένα επίπεδα σε αυτό μπορούν να βοηθήσουν στη βελτιστοποίηση και την ακρίβεια.

Η βαθιά μάθηση οδηγεί πολλές εφαρμογές και υπηρεσίες τεχνητής νοημοσύνης (AI) που βελτιώνουν την αυτοματοποίηση, εκτελώντας αναλυτικές και φυσικές εργασίες χωρίς την ανθρώπινη παρέμβαση. Η τεχνολογία βαθιάς εκμάθησης κρύβεται πίσω από τα καθημερινά προϊόντα και υπηρεσίες (όπως ψηφιακοί βοηθοί, τηλεχειριστήρια με φωνητική λειτουργία και εντοπισμός για απάτη σε πιστωτικές κάρτες), καθώς και αναδυόμενες τεχνολογίες (όπως αυτοκινούμενα οχήματα) [35].

### 3.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η προσπάθεια του ανθρώπου να κατανοήσει το περιβάλλον του μέσα από παρατηρήσεις και δημιουργώντας μια απλοϊκή (αφαιρετική) μορφή αυτού ονομάζεται μοντέλο (model). Επαγωγική μάθηση (inductive learning) ορίζεται ως ένα τέτοιο μοντέλο, ενώ σε γενικές γραμμές, η όλη διαδικασία ονομάζεται επαγωγή (induction). Ωστόσο, πρότυπα (patterns) ορίζονται οι νέες δομές που δημιουργεί ο άνθρωπος οργανώνοντας και συσχετίζοντας τις εμπειρίες και τις παραστάσεις του. Μηχανική μάθηση (machine learning) από ένα υπολογιστικό σύστημα, ονομάζεται η δημιουργία μοντέλων ή προτύπων από ένα σύνολο δεδομένων. Ένας γνωστός ορισμός για την μηχανική μάθηση από τον Mitchell είναι ο εξής: *"Ένα πρόγραμμα υπολογιστή θεωρείται ότι μαθαίνει από την εμπειρία  $E$  σε σχέση με μια κατηγορία εργασιών  $T$  και μια μετρική απόδοσης  $P$ , αν η απόδοση του σε εργασίες της  $T$ , όπως μετριοούνται από την  $P$ , βελτιώνονται με την εμπειρία  $E$ "* [36].

Για να παραχθεί ένα διάνυσμα εξόδου δοθέντος ενός συγκεκριμένου διανύσματος εισόδου χρειάζεται διαδικασία τροποποίησης των τιμών των βαρών του δικτύου. Αυτή η διαδικασία ονομάζεται μάθηση-εκπαίδευση (learning-training). Για ένα συγκεκριμένο

διάνυσμα εισόδου και στις αντίστοιχες τιμές βαρών, η διαδικασία υπολογισμού ενός διανύσματος εξόδου γι' αυτό το διάνυσμα ονομάζεται ανάκληση (recall). Έχουν δημιουργηθεί διάφορες τεχνικές μηχανικής μάθησης που έχουν χρησιμότητα αναλόγως την φύση του προβλήματος και συνηθίζεται να είναι οι τρεις παρακάτω:

- Επιβλεπόμενη μάθηση (supervised learning) ή μάθηση από παραδείγματα (learning from examples),
- Μη επιβλεπόμενη μάθηση (unsupervised learning) ή μάθηση μέσω παρατήρησης (learning from observation).
- Ημι-επιβλεπόμενη μάθηση (semi-supervised learning) ή ημι-κατευθυνόμενη μάθηση

Στη μάθηση με επίβλεψη, περιγραφή ενός μοντέλου θεωρείται μια έννοια ή μια συνάρτηση από ένα σύνολο δεδομένων, την οποία και το σύστημα καλείται να 'μάθει'. Στη μάθηση χωρίς επίβλεψη δεν υπάρχουν ομάδες από ένα σύνολο δεδομένων ή συσχετίσεις. Έτσι το σύστημα πρέπει μόνο του να τις ανακαλύψει δημιουργώντας τα κατάλληλα πρότυπα. Στη μάθηση με ημι-επίβλεψη γίνεται συνδυασμός των δύο παραπάνω μεθόδων.

Την πράξη στην θεωρία των τεχνικών μηχανικής μάθησης έρχονται να δώσουν τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ) τα οποία είναι προσομοιώματα αυτών του ανθρώπινου εγκεφάλου, όπως ειπώθηκε και νωρίτερα. Η υπολογιστική ισχύς των ΤΝΔ επιτυγχάνεται μέσω της παράλληλα κατανομημένης δομής τους και μέσω της ικανότητάς τους να μαθαίνουν και συνεπώς να γενικεύουν [37]. Αυτή η γενίκευση αντιπροσωπεύει την ικανότητά τους να εξαγουν λογικά συμπεράσματα, όταν στην είσοδο δοθούν δεδομένα άγνωστα, που δεν τα έχει ξαναδεί το νευρωνικό κατά την διαδικασία της εκπαίδευσης.

Η αναγνώριση προτύπων (προσώπων, αντικειμένων) είναι ένας από τους κύριους στόχους ενός ΤΝΔ. Οι νευρώνες είναι απλοί επεξεργαστές με μικρή υπολογιστική ισχύ, η οποία περιορίζεται σε έναν κανόνα που συνδυάζει σήματα εισόδου και σε μια συνάρτηση ενεργοποίησης που παράγει το σήμα εξόδου [38]. Τα σήματα εξόδου στέλνονται σε άλλες μονάδες των συνδέσεων που είναι γνωστές ως συναπτικά βάρη, τα οποία είτε ενισχύουν είτε αναστέλλουν το μεταφερόμενο σήμα. Η όλη διαδικασία επαναλαμβάνεται πολλές φορές και κάθε φορά τα συναπτικά βάρη τροποποιούνται, έτσι ώστε να παράγεται στο επίπεδο εξόδου το επιθυμητό αποτέλεσμα [39]. Χάρη σε αυτή την προσέγγιση μπορεί και προσομοιάζεται σήμερα η ανθρώπινη υπολογιστική καθώς συνεχίζει να αναπτύσσεται ολοένα και περισσότερο με ταχύτατους ρυθμούς.

Τα στάδια ανάπτυξης ενός ΤΝΔ είναι τα εξής τρία:

1. Εκπαίδευση (Training)
2. Αξιολόγηση (Validation)
3. Έλεγχος (Testing)

### 3.1.1 Επιβλεπόμενη Μάθηση

Σε αυτό το είδος εκμάθησης το ΤΝΔ εκπαιδεύεται έχοντας γνωστές εισόδους (διανύσματα ανεξαρτήτων μεταβλητών) οι οποίες αντιπροσωπεύουν κάποιες συγκεκριμένες, επιθυμητές εξόδους (διανύσματα εξαρτημένων μεταβλητών). Όλο αυτό το σύνολο δεδομένων αποτελεί το Σύνολο Εκπαίδευσης, το οποίο κατέχει ένα 70% από το συνολικό πλήθος των δεδομένων του προβλήματος. Σκοπός του ΤΝΔ είναι να είναι σε θέση να μαντεύει αποτελεσματικά εισόδους με άγνωστη έξοδο (Σύνολο Ελέγχου 15%). Αναλόγως την μορφή της εξόδου υπάρχουν τρεις κατηγορίες όπου διαχωρίζουν την επιβλεπόμενη μάθηση:

- Εάν η έξοδος αποτελείται από πραγματικές τιμές, τότε είναι ΤΝΔ Παλινδρόμησης (Regression ANN).
- Εάν η έξοδος είναι ονοματολογίες ετικετών των κλάσεων, τότε είναι ΤΝΔ Ταξινόμησης (Classification).
- Εάν η έξοδος είναι μόνο δύο ετικέτες κλάσεων, τότε θεωρείται δυαδική ταξινόμηση (Binary Classification).

Όσον αφορά την Ταξινόμηση αντιστοιχεί τα δεδομένα σε αντίστοιχες κλάσεις ή κατηγορίες. Συνήθως, αναφέρεται και ως επιβλεπόμενη μάθηση επειδή οι κλάσεις-κατηγορίες είναι προκαθορισμένες στο πρόβλημα πριν εισαχθούν τα δεδομένα στο νευρωνικό δίκτυο. Αλγόριθμοι αυτής της κατηγορίας λέγονται και ταξινομητές και απαιτούν οι κλάσεις να αντιστοιχίζονται με τις τιμές των γνωρισμάτων που χαρακτηρίζουν τα δεδομένα. Οι τεχνικές για Ταξινόμηση κάνουν χρήση από:

- Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines – SVMs)
- Νευρωνικά Δίκτυα

Με λιγότερη συχνή εμφάνιση αλλά εξίσου αξιοσημείωτα είναι οι εξής προσεγγίσεις:

- Πλέγματα (Lattices)
- Συμβολοσειρές (Strings)
- Χρονοσειρές (Time Series)

Αναλόγως τον τύπο της συνάρτησης εξόδου διακρίνονται και σε γραμμικές μεθόδους (linear methos) καθώς και σε μη γραμμικές μεθόδους (nonlinear methods).

### 3.1.1.1 Προσαρμογή Συναπτικών Βαρών

Κατά την διάρκεια της εκπαίδευσης λαμβάνει χώρα μια επαναληπτική διαδικασία η οποία έχει σκοπό στην αναπροσαρμογή των βαρών των συνάψεων μεταξύ των νευρώνων του ΤΝΔ, η οποία υλοποιείται μέσω κάποιου κανόνα μετατροπής (Hebbian Learning Rule). Το πλήθος των επαναλήψεων εκπαίδευσης ονομάζεται Εποχή (Epoch). Αναλυτικότερα, όταν εκπαιδεύεται το νευρωνικό δίκτυο εισάγονται ταυτόχρονα οι τιμές των διανυσμάτων για την είσοδο μαζί με τις τιμές των διανυσμάτων για την αντίστοιχη έξοδο. Το ΤΝΔ τότε μεταβάλλει τα συναπτικά βάρη προκειμένου οι τιμές που θα φέρει στην έξοδο να είναι όσο το δυνατόν πιο κοντά στις επιθυμητές πραγματικές τιμές. Πιο συγκεκριμένα, το ΤΝΔ θέτει τιμές στους συντελεστές βαρύτητας (συναπτικά βάρη) και ο κάθε νευρώνας αφού χρησιμοποιήσει την συνάρτηση άθροισης, που έχει οριστεί εξ' αρχής παράγει μια τιμή, η οποία με τη συνάρτηση μεταφοράς προωθείται στην τελική έξοδο του νευρωνικού δικτύου.

### 3.1.1.2 Επικύρωση

Στην συγκεκριμένη φάση χρησιμοποιείται το 15% του συνόλου των δεδομένων που δεν έχουν χρησιμοποιηθεί στο νευρωνικό δίκτυο για να πραγματοποιήσει την επικύρωση. Όσο γίνεται η διαδικασία της εκπαίδευσης το ΤΝΔ συνεχίζει να προσαρμόζει τα συναπτικά βάρη για όσο το σφάλμα παραμένει σταθερό. Την στιγμή που θα παρατηρηθεί ότι το σφάλμα αρχίζει να αποκτάει ανοδική πορεία, τότε η εκπαίδευση τερματίζεται.

Η εκπαιδευτική διαδικασία πρέπει να τερματίζει όταν έχει φτάσει τον προκαθορισμένο αριθμό επαναλήψεων (iterations) ή εάν το σφάλμα ξεπεράσει το προκαθορισμένο όριο ανοχής σφάλματος. Μέσω αυτής της διαδικασίας επικύρωσης (validation) είναι εύκολο να αποφευχθεί η υπερεκπαίδευση του δικτύου (overfitting).

### 3.1.1.3 Έλεγχος

Η φάση του ελέγχου είναι η τελευταία από τις τρεις φάσεις ανάπτυξης ενός ΤΝΔ και ξεκινάει μετά την ολοκλήρωση των άλλων δύο. Με δεδομένη την αρχιτεκτονική του το νευρωνικό δίκτυο πλέον χρησιμοποιεί το άλλο 15% του συνόλου των δεδομένων που δεν έχει

χρησιμοποιηθεί. Εννοείται εδώ πως αυτό το σύνολο δεν το έχει λάβει ποτέ πριν το ΤΝΔ (first time seen data). Εκτελείται σε ένα μόνο πέρασμα και το σύνολο των διανυσμάτων εισέρχεται στην είσοδο. Η διαφορά σε αυτή την φάση εκτέλεσης είναι ότι καθώς εισέρχονται τα διανύσματα στην είσοδο, το ΤΝΔ δεν τροφοδοτείται με διανύσματα εξόδου. Επομένως, είναι στην δική του ευχέρεια να αποφασίσει σωστά τις τιμές της εξόδου με βάση αυτά που έχει μάθει κατά την διάρκεια της εκπαίδευσης.

Τροχοπέδη στην ανάπτυξη αποτελεσματικών ΤΝΔ είναι η υποεκπαίδευση (underfitting) και η υπερεκπαίδευση (overfitting). Οι δύο πιο μεγάλες αιτίες για κακή απόδοση των αλγορίθμων μηχανικής μάθησης είναι χαρακτηριστικά αυτές οι δύο περιπτώσεις.

### **Υποεκπαίδευση**

Μια από τις πιο σημαντικές αρχές που διέπει τον κλάδο των νευρωνικών δικτύων αλλά και του υπόλοιπου συνόλου μοντέλων της μηχανικής μάθησης, είναι ότι ένα μοντέλο θα πρέπει να είναι όσο πιο απλοϊκό γίνεται. Τα απλά μοντέλα, εκτός του γεγονότος ότι κερδίζουν με το χαμηλό υπολογιστικό κόστος και την λιγιστή μνήμη που καταλαμβάνουν, έχουν και άλλα μη προφανή οφέλη όπως για παράδειγμα την ερμηνεία των αποτελεσμάτων ή αλλιώς η κατανόηση του κώδικα μεταξύ ερευνητών του χώρου είναι μακράν πιο βατή.

Παρόλα αυτά, τυχαίνει αρκετές φορές που ένα απλό μοντέλο δεν έχει την ικανότητα να λύσει κάποιο πρόβλημα εξαιτίας της πενιχρής αρχιτεκτονικής που διαθέτετε. Για παράδειγμα ένα απλό νευρωνικό δίκτυο με ένα και μοναδικό κρυφό στρώμα, δεν είναι σε θέση να εκτελέσει πολλούς μετασχηματισμούς στα δεδομένα, έχοντας ως αποτέλεσμα να μην μπορεί δώσει με επιτυχία τα ταξινομημένα δεδομένα. Από την άλλη, ένα νευρωνικό δίκτυο με μικρό αριθμό κρυφών στρωμάτων, δεν έχει πρόσβαση σε υψηλές διαστάσεις το οποίο και πάλι μπορεί να οδηγήσει σε λανθασμένη ταξινόμηση των στοιχείων. Επομένως, ένα μοντέλο που δεν μπορεί ούτε να μοντελοποιήσει τα δεδομένα εκπαίδευσης ούτε να γενικεύσει σε νέα δεδομένα ονομάζεται υποεκπαίδευση (underfitting) [40]. Γι' αυτό, λοιπόν, πρέπει πάντοτε να γίνεται επίβλεψη της πορείας του μοντέλου τόσο στο ζήτημα της ακρίβειας, όσο και στην τιμή του κόστους, έτσι ώστε να είναι βέβαιο ότι δεν έχει πέσει σε περίπτωση υποεκπαίδευσης.

Ένα ακόμη πρόβλημα το οποίο είναι πολύ πιθανό να οδηγήσει στην υποεκπαίδευση είναι η ελλιπής ποσότητα στοιχείων εκπαίδευσης. Μοντέλα, όπως τα νευρωνικά δίκτυα απαιτούν σύνολα εκπαίδευσης τα οποία διαθέτουν πληθώρα στοιχείων προκειμένου να συγκλίνουν. Τέλος, ένας ακόμα πολύ σημαντικός παράγοντας που μπορεί να οδηγήσει σε

υποεκπαίδευση είναι ο ρυθμός εκμάθησης του νευρωνικού δικτύου. Ένας ρυθμός μάθησης μικρής εμβέλειας μπορεί να οδηγήσει σε καθυστερημένη σύγκλιση του μοντέλου πράγμα το οποίο σχετίζεται άμεσα με περίπτωση υποεκπαίδευσης [41].

### **Υπερεκπαίδευση**

Στην αντίπερα όχθη, υπάρχει το πρόβλημα της υπερεκπαίδευσης (overfitting) των δικτύων το οποίο παρατηρείται όταν ένα μοντέλο λαμβάνει υψηλή ακρίβεια πάνω στο σύνολο εκπαίδευσης, αλλά χαμηλή στο σύνολο αξιολόγησης. Αυτό σημαίνει ότι το μοντέλο μαθαίνει τα δεδομένα εκπαίδευσης πολύ καλά, δηλαδή μαθαίνει τις λεπτομέρειες και το θόρυβο στα δεδομένα εκπαίδευσης, με αποτέλεσμα να επηρεάζει αρνητικά την απόδοση του μοντέλου σε νέα δεδομένα. Αυτό σημαίνει ότι το μοντέλο μαθαίνει τον θόρυβο που συλλέγεται από τα δεδομένα της εκπαίδευσης ως έννοια, όμως επειδή δεν εφαρμόζεται στα νέα δεδομένα κι έτσι δεν δύναται η ικανότητα γενίκευσης των μοντέλων [40]. Το πρόβλημα αυτό εντοπίζεται κυρίως σε ισχυρά μοντέλα τα οποία έχουν την ικανότητα να προσαρμόζονται πάρα πολύ καλά στο σύνολο εκπαίδευσης συμπεριλαμβανομένου ακόμα και των παρεκτρεπόμενων τιμών (outliers).

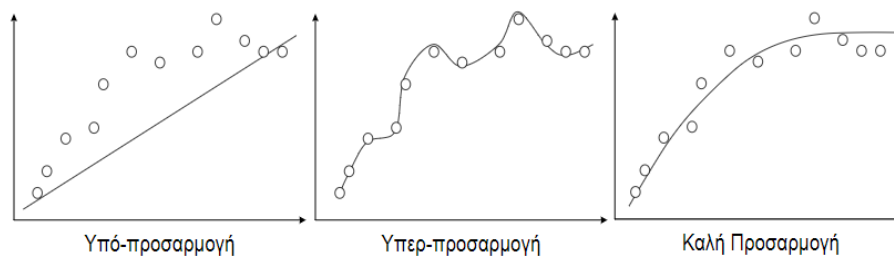
Οι λόγοι που μπορεί ένα νευρωνικό δίκτυο να οδηγηθεί σε υπερεκπαίδευση είναι τρεις. Ο πρώτος είναι η περίπτωση που τα διαθέσιμα δεδομένα δεν επαρκούν και δεν καλύπτουν όλο το εύρος των πραγματικών τιμών. Ο δεύτερος λόγος είναι στην περίπτωση που το νευρωνικό δίκτυο δεν τροφοδοτηθεί με καταλλήλως κατανεμημένα σύνολα εκπαίδευσης (training) και ελέγχου (testing). Ο τρίτος λόγος είναι όταν ο αριθμός των επαναλήψεων κατά την εκπαίδευση (εποχές) είναι αδικαιολόγητα μεγάλος.

Για να αποφευχθεί η υπερεκπαίδευση, συνιστάται να διατηρείται ένα ξεχωριστό σύνολο δεδομένων, το σύνολο επικύρωσης (validation set), πάνω στο οποίο να αξιολογείται το δίκτυο κατά την διάρκεια της εκπαίδευσης, όπως είχε αναφερθεί και νωρίτερα. Εάν παρατηρηθεί ότι από κάποια εποχή και μετέπειτα, η ακρίβεια του δικτύου πάνω στο σύνολο εκπαίδευσης αυξάνεται αλλά την ίδια στιγμή η ακρίβεια πάνω στο σύνολο επικύρωσης μειώνεται, τότε κατά πάσα πιθανότητα το δίκτυο έχει πέσει σε περίπτωση υπερεκπαίδευσης. Γι' αυτό είναι προτιμότερο να διακοπεί άμεσα η εκπαίδευση του δικτύου. Μια μέθοδος προς αυτή την κατεύθυνση είναι η μέθοδος Έγκαιρης Διακοπής ή αλλιώς Early Stopping. Σύμφωνα με αυτή την μέθοδο η εκπαίδευση του ΤΝΔ διακόπτεται περιοδικά (για παράδειγμα κάθε έναν



ορισμένο αριθμό από εποχές) και το ΤΝΔ δοκιμάζεται με τα δεδομένα του διανύσματος ελέγχου [37].

Ένας αρκετά χρησιμοποιούμενος τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα είναι μέσω της διασταυρωμένης επικύρωσης (cross validation). Στις περιπτώσεις που τα δεδομένα είναι σε μεγάλο βαθμό αραιά (sparse data), τότε μπορεί να μην είναι εφικτό να τεθεί αποτελεσματικά ένα μέρος από τα υπάρχοντα διανύσματα στην άκρη (leave one out) για να χρησιμοποιηθεί στην φάση του ελέγχου. Σε αυτή την περίπτωση, εάν τα δεδομένα χωριστούν ατυχώς το σφάλμα θα είναι παραπλανητικό. Γι' αυτόν τον λόγο χρησιμοποιείται η διασταυρούμενη επικύρωση (cross validation) ή το bootstrapping.



Εικόνα 3. Χαρακτηριστικά εκπαίδευσης

### ***Διασταυρωμένη Επικύρωση (K-Fold Cross Validation)***

Στην διασταυρωμένη επικύρωση (cross validation) τα δεδομένα απαρτίζονται από δύο σύνολα τα οποία είναι της εκπαίδευσης και της επικύρωσης. Το σύνολο εκπαίδευσης χρησιμοποιείται για την εκπαίδευση του μοντέλου, ενώ το σύνολο επικύρωσης χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου. Ουσιαστικά κάνει μια εκτίμηση για τις βέλτιστες παραμέτρους του μοντέλου και μόνο όταν αυτές έχουν σταθεροποιηθεί, τότε το μοντέλο εξετάζει το σύνολο ελέγχου.

Οι μετρήσεις (metrics) στο σύνολο εκπαίδευσης δίνουν την δυνατότητα να υπάρχει μια ξεκάθαρη εικόνα για το πώς προχωρά το μοντέλο καθώς εκπαιδεύεται. Βέβαια οι μετρήσεις στο σύνολο της επικύρωσης είναι αυτές που μετράνε την ποιότητα του μοντέλου, δηλαδή το πόσο καλά μπορεί να κάνει νέες προβλέψεις σε δεδομένα που δεν έχει ξαναδεί. Ως επί τω πλείστων χρησιμοποιείται το k-fold cross validation.

Σύμφωνα με τη μέθοδο k-fold cross validation, το σύνολο των δεδομένων χωρίζεται με τυχαίο τρόπο σε k ισάριθμα υποσύνολα (10-fold, 5-fold, 20-fold κ.λπ.) για k κύκλους από τα οποία σε κάθε επανάληψη τα k-1 χρησιμοποιούνται για εκπαίδευση, ενώ τα υπόλοιπα χρησιμοποιούνται ως σύνολο δοκιμής. Αυτή η διαδικασία εκτελείται επαναληπτικά k φορές αλλάζοντας κάθε φορά το ένα υποσύνολο που ξεχωρίστηκε για να είναι το νέο σύνολο ελέγχου. Οι δείκτες μέτρησης της επίδοσης σε αυτή την διαδικασία θεωρούνται οι αντίστοιχοι μέσοι όροι σφάλματος.

Στην περίπτωση που ο αριθμός των folds (k) είναι αρκετά μεγάλος, η εκτίμηση θα έχει μεγάλη ακρίβεια, η διακύμανση για το εύρος σφάλματος θα είναι μεγάλη καθώς και ο χρόνος εκτέλεσης θα είναι υψηλός. Στην άλλη περίπτωση όπου ο αριθμός των folds (k) είναι μικρός, τότε κατά συνέπεια η ακρίβεια σφάλματος κι ο χρόνος εκτέλεσης μειώνονται αντίστοιχα. Η επιλογή, λοιπόν, του αριθμού των folds εξαρτάται από το μέγεθος του συνόλου δεδομένων. Για μεγάλα σύνολα δεδομένων μια προσέγγιση των 3-folds έχει αποδειχθεί αρκετά ικανοποιητική. Μια πολύ συνηθισμένη προσέγγιση σε τέτοιες περιπτώσεις είναι αυτή των 10-folds. Ως κριτήριο αξιολόγησης χρησιμοποιείται ο τύπος για το Mean Absolute Percentage Error (MAPE) (Μέσο Απόλυτο Ποσοστό Σφάλματος) που δίνεται παρακάτω [39].

$$MAPE = \frac{1}{n} \sum_{i=1}^N \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| * 100$$

### 3.1.2 Μη Επιβλεπόμενη Μάθηση

Σε αυτό το είδος εκμάθησης ο εκάστοτε αλγόριθμος φτιάχνει ένα μοντέλο για ένα τυχαίο σύνολο εισόδων χωρίς να έχει επίγνωση για τις επιθυμητές εξόδους για το σύνολο εκπαίδευσης. Στη μη επιβλεπόμενη μάθηση είναι πολύ δύσκολο να γίνει ποσοτική αξιολόγηση για την απόδοση του συστήματος, διότι δεν υπάρχουν οι αντίστοιχες κλάσεις στα δεδομένα.

Για την εκτέλεση μη επιβλεπόμενης μάθησης θα μπορούσε να χρησιμοποιηθεί ένας κανόνας ανταγωνιστική μάθησης. Παραδείγματος χάριν, μπορεί να χρησιμοποιηθεί ένα νευρωνικό δίκτυο το οποίο αποτελείται από δύο επίπεδα. Το πρώτο επίπεδο είναι το επίπεδο εισόδου και το δεύτερο το ανταγωνιστικό επίπεδο. Δεδομένου κάποιου κανόνα εκμάθησης, όπως π.χ. του Hebb, στο επίπεδο εισόδου λαμβάνονται δεδομένα, έτσι ώστε στο ανταγωνιστικό επίπεδο που αποτελείται από νευρώνες να ανταγωνιστούν μεταξύ τους προκειμένου να αντιδράσουν στα χαρακτηριστικά των δεδομένων εισόδου. Με μια πιο απλή προσέγγιση, το

δίκτυο δουλεύει με την στρατηγική «ο νικητής παίρνει τα πάντα». Δεδομένης μιας τέτοιας στρατηγικής, όλοι οι νευρώνες με χαμηλή είσοδο απενεργοποιούνται και ενεργοποιείται μόνο αυτός με τη μεγαλύτερη συνολική είσοδο, όπου και κερδίζει στον ανταγωνισμό [42].

Κάποια γνωστά παραδείγματα μη επιβλεπόμενης μάθησης είναι η Συσταδοποίηση (Clustering), η Ελάττωση Διαστάσεων (Dimensionality Reduction), τα Συστήματα Πρότασης (Recommender Systems) και η Βαθιά Μάθηση (Deep Learning).

### 3.1.3 Ημι-Επιβλεπόμενη Μάθηση

Η ημι-επιβλεπόμενη μάθηση θεωρείται εκπαίδευση δύο σταδίων. Ο συγκεκριμένος τύπος μάθησης αποτελεί συνδυασμό των δύο παραπάνω μεθόδων που μόλις αναλύθηκαν. Στην φάση του πρώτου σταδίου πραγματοποιείται εκπαίδευση χωρίς επίβλεψη, η οποία κατανέμει τα δεδομένα σε ομάδες. Στη φάση του δεύτερου σταδίου πραγματοποιείται επιβλεπόμενη μάθηση όπου και η συνάρτηση απεικόνισης εισόδου-εξόδου δημιουργείται. Οι ομάδες που δημιουργούνται στην φάση των δύο πρώτων σταδίων συμβάλλουν στην αρχικοποίηση του ΤΝΔ στην διάρκεια του δεύτερου σταδίου.

#### 3.1.3.1 Ενισχυμένη Μάθηση

Στην ενισχυμένη μάθηση, στόχος είναι η ελαχιστοποίηση ενός βαθμωτού δείκτη απόδοσης κατά την διαδικασία της εκμάθησης, η οποία πραγματοποιείται μέσω συνεχούς αλληλεπίδρασης μιας αντιστοίχισης εισόδου-εξόδου με το περιβάλλον [42]. Με πιο απλά λόγια, βασίζεται σε διαδικασίες ανταμοιβής σε σχέση με το αποτέλεσμα. Έστω ένα παράδειγμα αυτόματης ταξινόμησης (classification) ανθρώπων σε πάσχοντες και υγιείς ως προς κάποια ασθένεια. Παρακάτω θα αναλυθούν ένα προς ένα τα χαρακτηριστικά της διαδικασίας που πρέπει να ακολουθηθεί, όπως στο βιβλίο των [39]:

- Δείγματα (Samples). Σημαντικό ρόλο παίζουν τα αντικείμενα ή στιγμιότυπα δεδομένων που χρησιμοποιούνται για την εκμάθηση ή την αξιολόγηση. Για το πρόβλημα της ταξινόμησης αυτού του παραδείγματος, τα δείγματα για κάποια ασθένεια αντιστοιχίζονται σε ένα σύνολο ανθρώπων από τους οποίους συλλέγονται τα απαραίτητα δεδομένα και χρησιμοποιούνται για την εκμάθηση και τον έλεγχο αποτελεσματικότητας της αυτόματης ταξινόμησης.

- **Χαρακτηριστικά (Features).** Τα χαρακτηριστικά ορίζονται ως το διάνυσμα που σχηματίζεται από ένα σύνολο γνωρισμάτων (attributes) το οποίο σχετίζεται με ένα δείγμα. Για το πρόβλημα της ταξινόμησης αυτού του παραδείγματος, μερικά χαρακτηριστικά δύναται να είναι διάφοροι ιατρικοί δείκτες, όπως το φύλο, η ηλικία, το ιατρικό ιστορικό για μια συγκεκριμένη ασθένεια ή γενικότερα, το αποτέλεσμα συγκεκριμένων εργαστηριακών ιατρικών εξετάσεων μαζί με πιθανούς δείκτες που προκύπτουν από κλινικές εξετάσεις.
- **Κατηγορίες ή Κλάσεις (Categories or Classes).** Κατηγορίες/Κλάσεις ορίζονται ως οι ομάδες στις οποίες ανατίθενται τα δείγματα. Για το συγκεκριμένο παράδειγμα όπου είναι οι κλάσεις «πάσχοντες» και «μη πάσχοντες», κάθε δείγμα κατηγοριοποιείται σε συγκεκριμένες κλάσεις, για τα προβλήματα της ταξινόμησης.
- **Σύνολο Εκπαίδευσης (Training Set).** Το σύνολο εκπαίδευσης αποτελείται από δείγματα τα οποία χρησιμοποιούνται κατά τη διάρκεια εκπαίδευσης για έναν αλγόριθμο εκμάθησης. Για ένα παράδειγμα ταξινόμησης της προκειμένης περίπτωσης, για ένα σύνολο ανθρώπων που έχουν ταξινομηθεί σε κατηγορίες «πάσχοντες ή μη πάσχοντες» αποτελεί το σύνολο εκπαίδευσης. Για κάθε σενάριο μάθησης αντιστοιχεί κι ένα διαφορετικό σύνολο εκπαίδευσης.
- **Σύνολο Επικύρωσης (Validation Set).** Το σύνολο επικύρωσης χρησιμοποιείται για τη ρύθμιση των παραμέτρων του αλγορίθμου εκμάθησης, το οποίο αποτελείται από δείγματα στα οποία είναι γνωστές οι κατηγορίες που ανήκουν. Στους αλγορίθμους εκμάθησης δίνεται η δυνατότητα να έχουν παραπάνω από μία ελεύθερες παραμέτρους, καθώς χρησιμοποιείται το σύνολο επικύρωσης για την επιλογή κατάλληλων τιμών γι' αυτές.

### 3.1.3.2 Βαθιά – Ρηχή Προσέγγιση

Είναι αποδεδειγμένο, ότι τα ΤΝΔ με ένα κρυφό επίπεδο το λιγότερο, δύναται να θεωρούνται και ως μέσο καθολικής προσέγγισης. Αυτός ήταν και ο λόγος που τα ΤΝΔ με ένα κρυφό επίπεδο κυριάρχησαν στον χώρο. Ένα τέτοιου είδους νευρωνικό δίκτυο μπορεί θεωρητικά να αναπαριστά οποιαδήποτε συνάρτηση. Αυτή η θεώρηση αντιτίθεται με τους περιορισμούς που διέπουν τα ΤΝΔ που δεν διαθέτουν κρυμμένα επίπεδα.

Πιο συγκεκριμένα, ένα νευρωνικό δίκτυο ενός επιπέδου, όπως το perceptron, είναι μόνο ένας γραμμικός ταξινομητής με αποτέλεσμα να μην είναι αποτελεσματικός στην

εκμάθηση μιας μεγάλης ποικιλίας εργασιών. Ένα τέτοιο νευρωνικό δίκτυο με ένα μόνο στρώμα, δεν μπορεί να μάθει να μοντελοποιεί συναρτήσεις τόσο απλές όσο η συνάρτηση XOR και άλλα μη γραμμικά διαχωρίσιμα προβλήματα ταξινόμησης.

Ήδη από την δεκαετία του '90 δοκιμάστηκαν επανειλημμένα τα ΤΝΔ με πολλά κρυφά υποεπίπεδα, αλλά η εκπαίδευσή τους δεν ήταν αποτελεσματική μέχρι που έκαναν την εμφάνισή τους ταχύτεροι επεξεργαστές. Τον τελευταίο καιρό έχει γίνει εισήγηση αρχιτεκτονικών πολλών κρυφών επιπέδων που ονομάζονται Βαθιές Αρχιτεκτονικές ΤΝΔ (Deep Networks) σε αντίθεση με τα ΤΝΔ ενός κρυφού επιπέδου που ονομάζονται Ρηχά ΤΝΔ (Shallow ANN). Τα Deep ANN αποδείχθηκαν πολύ αποτελεσματικά στη Μηχανική Όραση και στην Αναγνώριση Λόγου [43].

Η εκτίμηση που γίνεται για να βρεθεί ο ελάχιστος αριθμός των απαιτούμενων νευρώνων του κρυφού επιπέδου για την αναπαράσταση μιας δεδομένης συνάρτησης είναι ένα δύσκολο Μη Κυρτό πρόβλημα (Non Convex). Απαιτεί τη διερεύνηση της αραιότητας (sparsity) των ρηχών ΤΝΔ από την άποψη των προτύπων (norms) των συντελεστών βαρύτητας του επιπέδου εξόδου.

Πρόσφατα έχει έρθει στην επιφάνεια ένα εύρος διαφωνιών ανάμεσα στους μελετητές για την υπεροχή των Ρηχών ΤΝΔ (Shallow ANN) έναντι των Βαθέων ΤΝΔ (Deep ANN). Μια εμπειρική μελέτη των [44] έδειξε ότι τα ρηχά ΤΝΔ σε αρκετές περιπτώσεις μπορούν να μάθουν και να αποδώσουν καλύτερα από τα βαθιά χρησιμοποιώντας τον ίδιο αριθμό παραμέτρων. Η έρευνα παραμένει σε εξέλιξη και δεν έχει ακόμα διαμορφωθεί οριστικά το τελικό πλαίσιο αποδοχής ή μη της μιας προσέγγισης έναντι της άλλης. Η εμπειρία στην ανάπτυξη προτύπων ΤΝΔ έδειξε ότι πρέπει να αποφεύγονται τα ιδιαίτερα πολύπλοκα ΤΝΔ. Η απλότητα είναι υπεροχή [39]. Φυσικά αυτό δεν ισχύει όταν αντιμετωπίζουμε περιπτώσεις με μεγάλο αριθμό παραμέτρων [45].

### 3.2 ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Υπάρχουν πολλά είδη Νευρωνικών Δικτύων, τα οποία δημιουργήθηκαν για να επιλύσουν διάφορα είδη προβλημάτων κάθε φορά. Ένα από αυτά είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks) τα οποία γίνανε ευρέως γνωστά για προβλήματα τεχνητής όρασης. Αυτό που κάνει τα Συνελικτικά Δίκτυα να ξεχωρίζουν από τα άλλα, είναι το γεγονός

ότι υπάρχουν Συνελκτικά Επίπεδα. Τα επίπεδα αυτά έχουν στην διάθεσή τους Πυρήνες (Kernels) ή αλλιώς Φίλτρα (Filters), τα οποία αποδεδειγμένα έχουν πολύ μεγάλη απόδοση στην ανίχνευση χαρακτηριστικών και μοτίβων των οπτικών δεδομένων [41].

Τα συνελκτικά δίκτυα ή πιο σωστά συνελκτικά νευρωνικά δίκτυα CNN (Convolutional Neural Networks), είναι μια τοπολογία νευρωνικού δικτύου που μοιάζει με πλέγμα και εξειδικεύεται στην επεξεργασία δεδομένων [16]. Ο σχεδιασμός της αρχιτεκτονικής των νευρωνικών δικτύων CNN είχε βιολογική προσέγγιση και συγκεκριμένα από τον οπτικό φλοιό τη γάτας σε αντίθεση με του ανθρώπου, διότι διαθέτει μία περίπλοκη διάταξη κυττάρων. Η συγκεκριμένη διάταξη δημιουργεί συνδέσεις μεταξύ νευρωνικών κυττάρων όπου τα συγκεκριμένα έχουν ευαισθησίες σε υποπεριοχές του ορατού πεδίου, γεγονός το οποίο δίνει τη δυνατότητα σε αυτές τις υποπεριοχές να λειτουργούν ως τοπικά 'φίλτρα'. Δεδομένου αυτού του τρόπου, μια γάτα παραδείγματος χάριν θα μπορούσε να εκμεταλλευτεί τη μεγάλη χωρική συσχέτιση των διάφορων υποπεριοχών, προκειμένου να έχει καλύτερη αντίληψη του χώρου [46].

Τα συνελκτικά νευρωνικά δίκτυα CNN είναι υποσύνολο της κλάσης των Deep Neural Networks (DNNs) και είναι η κανονικοποιημένη μορφή των MLP (Multi Layer Perceptrons). Κυρίως χρησιμοποιούνται για εφαρμογές αναγνώρισης εικόνων και βίντεο, σε συστήματα που κάνουν συστάσεις (recommendation systems), σε ταξινόμηση εικόνων, σε αναλύσεις ιατρικών εικόνων και επεξεργασίας φυσικής γλώσσας. Μερικά παραδείγματα των συνελκτικών δικτύων περιλαμβάνουν δεδομένα από χρονοσειρές, σήματα και ακολουθίες, συμπεριλαμβανομένης και της γλώσσας, τα οποία μπορούν να θεωρηθούν ως μονοδιάστατο πλέγμα 1D. Επίσης τα δεδομένα εικόνας μπορούν και να θεωρηθούν ως ένα 2D πλέγμα εικονοστοιχείων (pixels) καθώς και 3D όταν πρόκειται για βίντεο. Τα CNN αποδείχτηκαν εξαιρετικά επιτυχημένα σε πρακτικές εφαρμογές. Τους δόθηκε η ονομασία «συνελκτικά», διότι γινόταν χρήση της μαθηματικής πράξης που ονομάζεται συνέλιξη. Τα συνελκτικά νευρωνικά δίκτυα, σε αντίθεση με τα απλά ΤΝΔ δεν χρησιμοποιούν πολλαπλασιασμό πινάκων αλλά την πράξη της συνέλιξης τουλάχιστον σε ένα από τα στρώματά τους. [47].

Τα δίκτυα CNN που συνήθως χρησιμοποιούνται και προσδιορίζονται από τα επίπεδα νευρώνων κατηγοριοποιούνται σε δύο μέρη βάση του [46]:

- Τα Συνελκτικά Επίπεδα (Convolutional Layers). Τα συνελκτικά επίπεδα αποτελούνται από ένα πλέγμα νευρώνων. Μια είσοδος (input) ενός νευρωνικού δικτύου, από ένα πλέγμα του προηγούμενου επιπέδου του οποίου τα βάρη

παραμένουν αμετάβλητα για όλους τους επόμενους νευρώνες αποτελεί το κάθε συνελκτικό επίπεδο. Έτσι γίνεται πιο κατανοητό το γεγονός, ότι κάθε προηγούμενο επίπεδο λειτουργεί συνελκτικά με το επόμενο του με το μέγεθος του εφαρμοζόμενου πλέγματος να είναι πιθανόν μεταβλητό σε καθένα από αυτά. Στις περιπτώσεις δισδιάστατων δεδομένων, τα πλέγματα πρέπει να είναι τετραγωνικά, για τρισδιάστατα δεδομένα κυβικά και ούτω καθεξής.

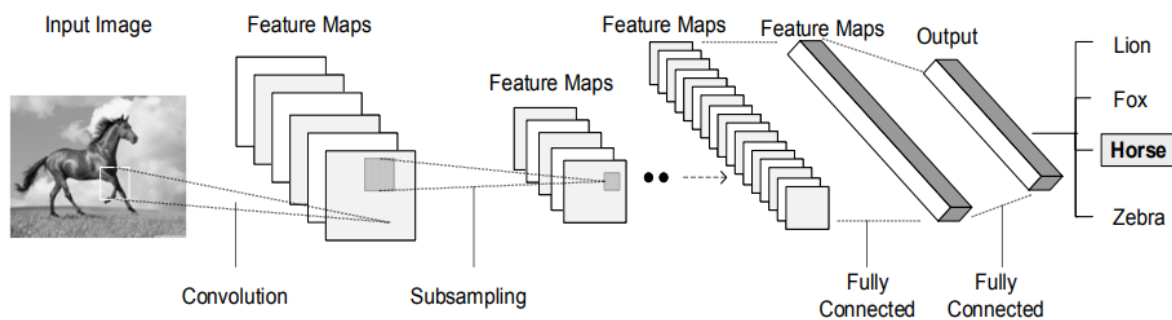
- Τα Επίπεδα Μέγιστης Συγκέντρωσης (Max-Pooling Layers). Μετά από τα συνελκτικά επίπεδα συνηθίζεται να βρίσκονται τα επίπεδα μέγιστης συγκέντρωσης. Γενικότερα, το προηγούμενο συνελκτικό επίπεδο αποστέλλει ως είσοδο στα επίπεδα συγκέντρωσης ένα πλέγμα από φέρει ως έξοδο μία μοναδική τιμή η οποία συνήθως είναι κάποια γραμμική παλινδρόμηση, η μέση τιμή του πλέγματος, η μέγιστη τιμή, κ.α. Επίπεδα μέγιστης συγκέντρωσης (Max-Pooling Layers) λέγονται τα επίπεδα που δίνουν πίσω την μέγιστη τιμή.

Δεδομένου του γεγονότος ότι αρκετά φυσικά σήματα δίνουν ιεραρχίες σύνθεσης, στις οποίες βρίσκονται χαρακτηριστικά υψηλών επιπέδων που προσκομίζονται συνθέτοντας χαρακτηριστικά χαμηλών επιπέδων, τα βαθιά νευρωνικά δίκτυα δεν το αφήνουν ανεκμετάλλευτο. Παραδείγματος χάρη, στην ανίχνευση εικόνων τα άκρα (edges) ως τοπικοί συνδυασμοί δημιουργούν μοτίβα, τα οποία με την σειρά τους ενώνονται σε κομμάτια και αυτά, τέλος, σχηματίζουν αντικείμενα. Παρόμοιες ιεραρχίες υπάρχουν στην ομιλία και το κείμενο. Η συγκέντρωση (pooling) επιτρέπει στις αναπαραστάσεις να διαφέρουν ελάχιστα όταν τα στοιχεία στο προηγούμενο στρώμα ποικίλλουν ως προς τη θέση και την εμφάνιση [48].

### 3.2.1 Ανίχνευση Αντικειμένων με CNN

Τα βαθιά CNN έχουν χρησιμοποιηθεί εκτενώς για τα προβλήματα ανίχνευσης αντικειμένων. Το CNN είναι ένας τύπος νευρωνικού δικτύου προώθησης και λειτουργεί με βάση την κατανομή βάρους. Η μετατροπή είναι μια ολοκλήρωση που δείχνει πώς μια συνάρτηση επικαλύπτεται με μια άλλη συνάρτηση και είναι ένα μείγμα δύο συναρτήσεων που πολλαπλασιάζονται. Στο παρακάτω σχήμα φαίνεται η πολυεπίπεδη αρχιτεκτονική του CNN για την ανίχνευση ενός αντικειμένου. Η εικόνα συγχέεται με τη λειτουργία ενεργοποίησης για λήψη των χαρτών χαρακτηριστικών (feature maps). Για να μειωθεί η χωρική πολυπλοκότητα του δικτύου, οι χάρτες χαρακτηριστικών αντιμετωπίζονται με επίπεδα συγκέντρωσης (pooling

layers) για να λάβουν αφηρημένους χάρτες χαρακτηριστικών (abstracted feature maps). Η παραπάνω διαδικασία επαναλαμβάνεται για τον επιθυμητό αριθμό φίλτρων και αναλόγως δημιουργούνται οι χάρτες χαρακτηριστικών. Σε τελικό στάδιο, αυτοί οι χάρτες χαρακτηριστικών υποβάλλονται σε επεξεργασία με πλήρως συνδεδεμένα επίπεδα για να αποκτήσουν έξοδο αναγνώρισης εικόνας που να δείχνει βαθμολογία εμπιστοσύνης (confidence score) για τις ετικέτες της κλάσης που ήταν να προβλεφθεί. Τα επίπεδα συγκέντρωσης είναι σταθερά στην μετάφραση (translation invariant). Οι χάρτες ενεργοποίησης τροφοδοτούνται ως είσοδος στα επίπεδα συγκέντρωσης. Λειτουργούν σε κάθε ενημερωμένη έκδοση κώδικα στον επιλεγμένο χάρτη [2].



Εικόνα 4. Χρήση των Συνελκτικών Νευρωνικών Δικτύων για ανίχνευση αντικειμένου [2]



#### 4.0 ΥΠΟΛΟΓΙΣΤΙΚΗ ΑΙΧΜΗΣ

Δεδομένου του γεγονότος πως πλέον οι σύγχρονες τεχνολογίες απαιτούν την αποθήκευση, επεξεργασία και χρήση δεδομένων μεγάλου όγκου, όχι μόνο σε ένα τοπικό δίκτυο αλλά και σε υπολογιστές αποκρυσμένους, προσβάσιμους μέσω διαδικτύου, είναι απαραίτητη η ύπαρξη τεχνολογιών που θα παρέχουν την διευκόλυνση αυτών των καταστάσεων. Αυτή προσφέρεται αρχικώς από το Cloud Computing που αποτελεί μία τεχνολογική υποδομή όπου κάθε χρήστης αποκτάει πρόσβαση σε λογισμικά και δεδομένα που βρίσκονται εκτός εμβέλειας από τον προσωπικό του υπολογιστή, δηλαδή συνδέεται σε ψηφιακές συσκευές άλλων χρηστών, μη γνωρίζοντας την φύση αλλά και την τοποθεσία των διακομιστών όπου βρίσκονται εγκατεστημένα το λογισμικό και τα δεδομένα, μέσω διαδικτυακής σύνδεσης.

Παρότι το Cloud Computing παρέχει τα οφέλη που προαναφέρθηκαν, απαιτείται ένα επιπλέον τεχνολογικό επίτευγμα που θα ενισχύσει τις επιδόσεις και τις αξιώσεις ενός συστήματος, όσον αφορά τους χρόνους απόκρισής του σε κρίσιμες και σύνθετες εφαρμογές. Αυτό είναι το Edge Computing που συμπληρώνει ως ένα υβριδικό περιβάλλον πληροφορικής το Cloud Computing, παρέχοντας την δυνατότητα να εκτελούνται πλήθη σύνθετων εργασιών υπολογισμού και επεξεργασίας από διασυνδεδεμένες συσκευές IoT.

Ως αποτέλεσμα, φέρει την μειωμένη καθυστέρηση στο χρόνο που λαμβάνουν λειτουργίες που αφορούν στην επικοινωνία και ανταλλαγή των πληροφοριών μεταξύ των κεντρικών διακομιστών, στην ορθή διαχείριση των πολυπληθών παραγόμενων πληροφοριών που λαμβάνονται από τις συσκευές αλλά και στον μετριασμό της εξάρτησής τους από το Cloud Computing. Η σημαντικότητα του είναι κύριο γνώρισμα για εφαρμογές μηχανικής μάθησης όπου ο υπολογισμός είναι άρρηκτα συνδεδεμένος με την επιτυχή αναγνώριση προσώπων, ανίχνευση αντικειμένων κλπ.

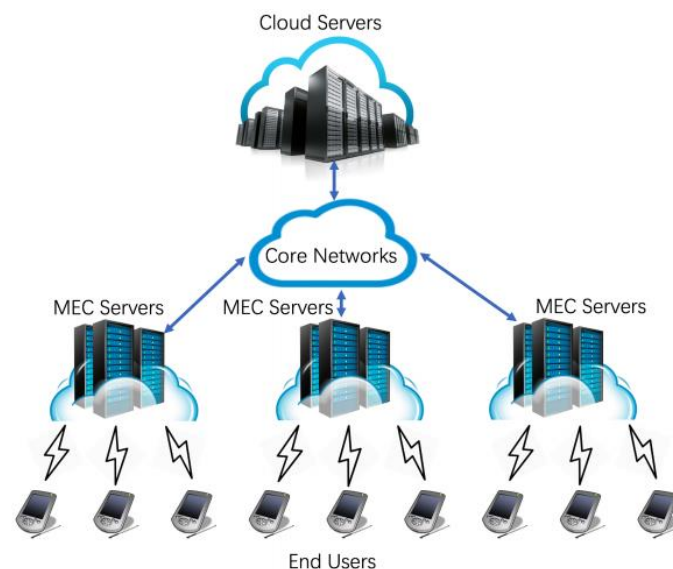
Το Edge Computing συμβάλλει σημαντικά στην μείωση του χρόνου απόκρισης επομένως, οδηγεί και στην μείωση του κόστους διασύνδεσης καθώς εξοικονομούνται πόροι του δικτύου υπολογιστών μέσω της επιλεγμένης αποστολής των απαραίτητων - για κάθε διαδικασία - πληροφοριών αλλά και στη δυνατότητα που υπάρχει για ανάλυση και φιλτράρισμα των δεδομένων πριν αποσταλούν, ισχυροποιώντας κατ' αυτόν τον τρόπο τις διασυνδεδεμένες εφαρμογές.

Το προκείμενο κεφάλαιο θα επικεντρωθεί στην τεχνολογία του Edge Computing, παρουσιάζοντας την αρχιτεκτονική του, στα πλεονεκτήματα που διαθέτει, τον ρόλο του στο

Διαδίκτυο των Αντικειμένων (IoT), την συμβολή του στην εικονοποίηση κι εν τέλει στις εφαρμογές στις οποίες βρίσκει χρήση.

#### 4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ

Το Edge Computing δομείται από τρία μέρη: το front-end, το near-end και το far-end [49]. Στην παρακάτω εικόνα αποτυπώνεται η αρχιτεκτονική του.



Εικόνα 5. Η βασική αρχιτεκτονική του Edge Computing [49]

Η περιγραφή των τριών μερών που απαρτίζουν το Edge Computing είναι η εξής:

- **Front-end:** Στο front-end της δομής του edge computing γίνεται η τοποθέτηση των λεγόμενων τελικών συσκευών, όπως για παράδειγμα οι αισθητήρες ή οι ενεργοποιητές. Σε αυτό το περιβάλλον, υπάρχει περισσότερη διαδραστικότητα και απόκριση για τον τελικό χρήστη. Μέσω της υπολογιστικής ισχύς από την πληθώρα των κοντινών συσκευών, το edge computing προσφέρεται σε κάποιες εφαρμογές/υπηρεσίες πραγματικού χρόνου. Με την πλειοψηφία των απαιτήσεων να μην μπορούν να καλυφθούν από το περιβάλλον του front-end, εξαιτίας της περιορισμένης ισχύς που έχουν οι τελικές συσκευές, οδηγούν την προώθηση των απαιτήσεων των πόρων στους διακομιστές.

- Near-end: Πλησίον του άκρου εγκαθίστανται πύλες που υποστηρίζουν τις ροές κυκλοφορίας εν μέσω δικτύων, με τους edge/cloudlet διακομιστές να έχουν πληθώρα απαιτήσεων για πόρους. Οι περισσότερες ενέργειες επεξεργασίας και αποθήκευσης δεδομένων, κατά την υπολογιστική του ακροδικτύου, μεταφέρονται σε αυτό το κοντινό-στο-άκρο περιβάλλον ούτως ώστε να επιτευχθεί πολύ μεγάλη αποδοτικότητα από τους τελικούς χρήστες σε ότι αφορά την επεξεργασία δεδομένων και την αποθήκευση τους, με μια σχεδόν αμελητέα αύξηση στην καθυστέρηση.
- Far-end: Η εγκατάσταση των διακομιστών cloud ακόμη πιο μακριά από τις τερματικές συσκευές έχει ως αποτέλεσμα την καθυστέρηση μετάδοσης, η οποία παίζει πολύ μεγάλο ρόλο στο δίκτυο. Ωστόσο, οι διακομιστές νέφους στον περιβάλλοντα χώρο του far-end δύναται να παρέχουν μεγαλύτερη υπολογιστική ισχύ και χώρο αποθήκευσης δεδομένων με μαζική και παράλληλη επεξεργασία των δεδομένων, εξόρυξη και διαχείριση μεγάλων δεδομένων, μηχανική μάθηση κα.

## 4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ

Το Edge Computing επιφέρει σημαντικά πλεονεκτήματα τα οποία αφορούν κυρίως στη μετάδοση, την αποθήκευση και τον υπολογισμό στις εξελισσόμενες εργασίες. Με γνώμονα αυτά τα κριτήρια, θα γίνει αναφορά στον τρόπο με τον οποίο προκύπτουν αυτά τα πλεονεκτήματα .

### 4.2.1 Μετάδοση

Η αξιολόγηση της αποδοτικότητας του δικτύου πραγματοποιείται με κριτήρια την καθυστέρηση, το εύρος ζώνης και την απώλεια πακέτων αλλά και την επιρροή που ασκείται στον χρόνο μετάδοσης. Ως βασικό προτέρημα του edge computing είναι η άμεση και γρήγορη μετάδοση, υποστηρίζοντας εν δυνάμει την ποιότητα υπηρεσιών των ευαίσθητων, ως προς τον χρόνο, εφαρμογών. Αυτό αποσκοπεί στην δημιουργία ενός χαμηλού κόστους και πραγματικού χρόνου συστήματος προκειμένου να επιτευχθεί η ανάλυση ζωντανών βιντεοσκοπήσεων, τα οποία εξάγονται από όλες τις διαθέσιμες κάμερες σε μια τοπική ανοικτή περιοχή. Αυτό το

σύστημα λειτουργεί σε μια ιεραρχία γεωγραφικά ταξινομημένων έξυπνων κόμβων ακροδικτύου και μεγάλων cloud υποδομών με εγγύηση τον μικρότερο χρόνο μετάδοσης από οποιοδήποτε άλλο δίκτυο. Το edge computing έχει επίσης δημιουργηθεί προκειμένου να φέρει λύση στο κώλυμα των πόρων δικτύου στο IoT. Μέσω της αποφόρτισης των τελικών χρηστών από την επεξεργασία και αποθήκευση δεδομένων, γίνεται αποσυμφόρηση δικτύου.

Σε μία εφαρμογή, η *καθυστέρηση* προκύπτει από το άθροισμα του χρόνου της καθυστέρησης επεξεργασίας και μετάδοσης. Η καθυστέρηση επεξεργασίας είναι εξαρτώμενη από τις δυνατότητες υπολογισμού που προσφέρονται από το σύστημα. Οι αισθητήρες είναι συχνά ενσωματωμένες συσκευές με περιορισμένη υπολογιστική ισχύ, ενώ οι υπηρεσίες των διακομιστών δικτύου έχουν μεγάλη ισχύ ώστε να παρέχουν γρήγορη επεξεργασία δεδομένων. Ωστόσο, η μετάδοση των δεδομένων μεταξύ των τερματικών συσκευών και των διακομιστών νέφους έχει ως συνέπεια την αύξηση στην καθυστέρηση μετάδοσης.

Οπότε, στόχος του edge computing είναι η εύρεση της ιδανικής ισορροπίας μεταξύ καθυστέρησης μετάδοσης και επεξεργασίας, ευελπιστώντας για την εύρεση της βέλτιστης λύσης βάση της οποίας αποφασίζεται αν ο υπολογισμός θα πραγματοποιηθεί τοπικά, αν θα μεταφερθεί στους κόμβους του άκρου ή αν θα μεταφερθεί στους διακομιστές νέφους.

Το γεγονός πως το IoT, ως επί των πλείστων, στηρίζεται σε πολλούς αισθητήρες, είναι μεγάλης σημασίας η παράμετρος της ορθής συλλογής δεδομένων καθώς, απαγορεύεται η μεταφορά αυτών των δεδομένων απευθείας στους διακομιστές νέφους εάν δεν είναι συμπιεσμένα ή επεξεργασμένα τα δεδομένα. Ο μαζικός όγκος δεδομένων οδηγεί στην κατανάλωση πολύ μεγάλου μέρους του εύρους ζώνης του δικτύου προκαλώντας επιδείνωση κάποιων καταστάσεων, όπως είναι η καθυστέρηση μετάδοσης και η απώλεια πακέτων. Επομένως, κρίνεται απαραίτητο να προηγηθεί μία επεξεργασία των δεδομένων ή ακόμη και η συγχώνευση πριν πραγματοποιηθεί η προώθηση τους στους απομακρυσμένους cloud διακομιστές. Επομένως, το ζήτημα είναι ο έλεγχος της ροής κυκλοφορίας μεταφέροντας την επεξεργασία δεδομένων και των εργασιών ομαδοποίησης με βέλτιστο τρόπο έτσι, ώστε να μειωθούν οι απαιτήσεις εύρους ζώνης των τελικών χρηστών, ενώ παράλληλα να μην υπάρχει υποβάθμιση της ποιότητας των δεδομένων.

Οι τελικές συσκευές στο IoT διαφέρουν εκτός από τους πόρους δικτύου και στους ενεργειακούς πόρους και την διάρκεια ζωής της μπαταρίας. Αυτοί είναι παράγοντες οι οποίοι πρέπει να λαμβάνονται σοβαρά υπόψη κάθε φορά που μια τερματική συσκευή εκτελεί την διαδικασία επεξεργασίας ή προώθησης δεδομένων. Η επέκταση της διάρκειας ζωής των

τερματικών συσκευών είναι κρίσιμη και ειδικότερα αυτών όπου η μπαταρία είναι περιορισμένη. Προκειμένου να επιτευχθεί αυτός ο στόχος, το edge computing έχει την δυνατότητα να ενσωματώσει μια ευέλικτη προσέγγιση για αποφόρτιση εργασιών, που λαμβάνει υπόψη τους ενεργειακούς πόρους της κάθε συσκευής.

Κατά την μετάδοση δικτύου υπάρχει ένα κόστος κεφαλίδας και ένα φορτίο που αντιστοιχεί σε κάθε πακέτο δεδομένων. Τα χαρακτηριστικά μοτίβα στα δεδομένα του IoT απαρτίζονται τα περισσότερα από πακέτα δεδομένων μικρού μεγέθους αλλά συνάμα από έναν μαζικό αριθμό IoT συσκευών, επιβαρύνοντας το δίκτυο, με την μείωση αυτής της επιβάρυνσης να αποτελεί πρόκληση για το edge computing. Με την βοήθεια των edge/cloudlet διακομιστών, τα περιττά πακέτα μπορούν να ενοποιηθούν και να προεπεξεργαστούν με σκοπό την μείωση της αχρείαστης επιβάρυνσης.

#### 4.2.2 Αποθήκευση

Πλήθος κόμβων ακροδικτύου είναι υπεύθυνοι για την εξυπηρέτηση των απαιτήσεων αποθήκευσης και εν αντιθέσει με το παραδοσιακό νέφος, η αποθήκευση που έχει βάση στο edge computing είναι ταξινομημένη στο άκρο της δικτυακής δομής, εξισορροπώντας την σε διάφορους κόμβους.

Προκειμένου να ικανοποιηθούν οι QoS (Quality of Service) απαιτήσεις, η αποθήκευση που βασίζεται στο edge computing μπορεί να εκμεταλλευτεί τεχνικές για την ικανοποίηση των απαιτήσεων απόδοσης και την διαθεσιμότητα, όπως είναι η εξισορρόπηση φόρτου και η αποκατάσταση απωλειών. Η εξισορρόπηση φόρτου έχει την ικανότητα να αποφορτίζει τις απαιτήσεις αποθήκευσης σε διαφορετικούς edge κόμβους, ελαχιστοποιώντας την κυκλοφορία στις συνδέσεις δικτύου, ενώ η αποκατάσταση απωλειών ενισχύει την ανίχνευση των αποτυχημένων δεδομένων κατά την μαζική ροή τους από πολυπληθείς πηγές.

1. Εξισορρόπηση αποθήκευσης: Στα IoT δίκτυα, είναι σύνηθες ο χώρος αποθήκευσης να είναι περιορισμένος. Ο όγκος παραγωγής ή συλλογής δεδομένων από τις συσκευές μεταδίδουν και αποθηκεύουν σε ένα διακομιστή αποθήκευσης. Επιπλέον, σε κάποιες περιπτώσεις, υπάρχουν IoT συσκευές που κάνουν μαζική παραγωγή δεδομένων ταυτόχρονα. Αν όλες οι συσκευές αποθηκεύουν ταυτόχρονα σε χώρο αποθήκευσης cloud, τότε το αποτέλεσμα θα είναι μια σημαντική επιβάρυνση στο δίκτυο. Όταν πραγματοποιείται παραγωγή δεδομένων τα οποία πρέπει να σταλούν στον αποθηκευτικό χώρο

μέσα σε ένα μικρό χρονικό διάστημα, πρέπει να γίνει έγκαιρη ενσωμάτωση στην διαδικασία ανάλυσης. Με το edge computing μεγιστοποιείται η αποσυμφόρηση των μεγάλων αποστάσεων του δικτύου όταν τα δεδομένα στέλνονται σε διαφορετικούς κόμβους.

2. Πολιτική αποκατάστασης: Είναι μια βασική απαίτηση στα συστήματα αποθήκευσης edge computing και η αξιοπιστία είναι εμφανώς σημαντική για την αποθήκευση και την ανάκτηση αναπαραστάσεων δεδομένων με ακρίβεια. Για να αυξηθεί η αξιοπιστία, πραγματοποιείται έλεγχος από το σύστημα για τη διαθεσιμότητα των κόμβων αποθήκευσης, τα διπλότυπα δεδομένα ή την χρήση άλλων κόμβων για περιττότητα.

Επίσης, υπάρχει η πιθανότητα μια υπηρεσία αποθήκευσης να σταματήσει να είναι διαθέσιμη για διάφορους λόγους και αυτό φέρει ως αποτέλεσμα την διεξαγωγή μίας περιοδικής ανίχνευσης για την παρακολούθηση των συστημάτων, ούτως ώστε να διασφαλιστεί η υγεία του χώρου αποθήκευσης αλλά και η διαθεσιμότητα των edge κόμβων. Επομένως, προκύπτει ότι οι υπηρεσίες αποθήκευσης σε κάποιο σημείο γίνονται μη διαθέσιμες. Σε ένα σύστημα cloud computing, οι περιττοί διακομιστές αποθήκευσης εγκαθίστανται προκειμένου να διαχειριστούν αυτό το ζήτημα και να δράσουν και να χρησιμοποιηθούν για την περιττή αποθήκευση. Στο IoT, μεγάλος αριθμός συσκευών έχουν την απαίτηση μιας συνεχούς αποθήκευσης δεδομένων, με αποτέλεσμα να προκύπτει κρισιμότητα της επιλογής παρόχου υπηρεσιών για διαθέσιμο χώρο αποθήκευσης.

Στο περιβάλλον IoT, η πολυάριθμη ομάδα συσκευών εισάγει την συνεχή απαίτηση για αποθήκευση δεδομένων. Αναντίρρητα, η ορθότητα των ευαίσθητων δεδομένων είναι αναγκαία, όπως τα προσωπικά δεδομένα που αφορούν την υγεία, οι μετρήσεις κατανάλωσης ενέργειας, κ.α. Έτσι, τα διανεμημένα συστήματα αποθήκευσης πρέπει να περιλαμβάνουν IoT περιβάλλοντα για την διαχείριση της μαζικής αυτής ζήτησης και διασφάλισης της ακρίβειας των δεδομένων.

Τα διανεμημένα συστήματα αποθήκευσης μπορούν να αυξήσουν την αξιοπιστία και να επεκτείνουν τον MTTF (Mean Time To Failure) με την χρήση αντιγράφων. Στα καταναμημένα συστήματα αποθήκευσης τα δεδομένα είναι χωρισμένα σε πολλά κομμάτια με το καθένα από αυτά να έχει προκαθορισμένο μέγεθος και μπλοκ κώδικα, ενώ έχουν και προκαθορισμένες επικαλύψεις μεταξύ τους. Αυτό έχει ως αποτέλεσμα την επανασυναρμολόγηση των δεδομένων που αποθηκεύονται σε κάθε κομμάτι από άλλα κομμάτια που συσχετίζονται μεταξύ τους. Η

αποθήκευση στο edge computing είναι ένα διανεμημένο σύστημα αποθήκευσης όχι μόνο λογικά αλλά και φυσικά. Έτσι, γίνεται η αντιγραφή των ευαίσθητων IoT δεδομένων και διαφορετικά κομμάτια από αυτά έχουν την δυνατότητα αποθήκευσης σε ξεχωριστές γεωγραφικές περιοχές, ελαχιστοποιώντας τον κίνδυνο να χαθούν δεδομένα.

### 4.2.3 Υπολογισμός

Οι υπολογισμοί έχουν μεγάλη και σημαντική συμβολή στην ικανοποίηση των απαιτήσεων, με κάθε εργασία τους να ανατίθεται σε διάφορους edge κόμβους, με τους συγκεκριμένους να καταναλώνουν μικρό ποσοστό της υπολογιστικής ισχύς που διατίθεται στους cloud διακομιστές.

Για να είναι επιτυχής ο προγραμματισμός αυτών των εργασιών βάσει edge computing, λαμβάνονται υπόψιν κάποιες προσεγγίσεις, όπως είναι η αποφόρτιση των υπολογισμών, η τιμολογιακή πολιτική και ο ορισμός προτεραιότητας, οι οποίες και παρατίθενται στη συνέχεια.

1. Αποφόρτιση Υπολογισμών: Η αποφόρτιση των υπολογισμών και η αύξηση της απόδοσής τους στηρίζεται στην ρύθμιση της τοποθεσίας των υπολογιστικών εργασιών, χωρίζοντας τις τοποθεσίες τοπικά, σε κόμβους edge/cloudlets και σε νέφος.
  - Τοπικά: Μέσω του σχεδιασμού ενός μηχανή-προς-μηχανή δικτύου (M2M) που δημιουργείται από τον πίνακα με τις IoT συσκευές και την απεικόνιση του μεγάλου αριθμού των γειτονικών συσκευών, επιτρέπεται στους τελικούς χρήστες η εκτέλεση κάποιων υπολογιστικών εργασιών και επιτυγχάνεται από τους ίδιους ο ελάχιστος χρόνος απόκρισης.
  - Σε κόμβους Edge/Cloudlets: Όσες απαιτήσεις πόρων των τελικών συσκευών δεν επιτυγχάνει να καλύψει το M2M, αναλαμβάνουν να τις ικανοποιήσουν οι διακομιστές edge/cloudlets πραγματοποιώντας προγραμματισμό των εργασιών τους. Αποσκοπούν στο να δεσμεύσουν το βέλτιστο σύνολο διακομιστών, υπό συγκεκριμένους περιορισμούς, ούτως ώστε να ελαχιστοποιηθεί όσο το δυνατόν περισσότερο η καθυστέρηση των υπολογισμών και της μετάδοσης, η κατανάλωση της

ενέργειας για τους υπολογισμούς και τις επικοινωνίες αλλά και η χρήση του απαραίτητου εύρους ζώνης από τις IoT συσκευές.

- Στο Νέφος: Όταν για κάποιες εργασίες, όπως είναι η αποθήκευση και η επεξεργασία δεδομένων, απαιτούνται περισσότεροι πόροι από αυτούς που προσφέρονται αλλά και δίχως να δεσμεύονται εξ' ολοκλήρου, ο υπολογισμός και η αποθήκευση πραγματοποιούνται στους διακομιστές νέφους. Αυτοί, λόγω του γεγονότος πως μέσα στο δίκτυο διαθέτουν την μεγαλύτερη υπολογιστική ισχύ, επιτυγχάνουν την μικρότερη υπολογιστική καθυστέρηση. Δεν ισχύει όμως το ίδιο και για την καθυστέρηση μετάδοσης, καθώς η απόσταση που χωρίζει τους διακομιστές νέφους με τις τερματικές συσκευές είναι μεγάλη, αποτελώντας έτσι μία πρόκληση για την εύρεση και επίτευξη μίας ισορροπίας μεταξύ των καθυστερήσεων υπολογισμού και μετάδοσης.
2. Τιμολογιακή Πολιτική: Η τιμολογιακή πολιτική αφορά την διαχείριση των πόρων, τους παρόχους που την αναλαμβάνουν και τα αποτελέσματα από την συνεργασία τους με τους τελικούς χρήστες. Έτσι, η πολιτική αυτή χωρίζεται σε ύπαρξη ενός μοναδικού παρόχου υπηρεσιών και στην ύπαρξη πολλαπλών παρόχων για την δέσμευση πόρων.
- Μοναδικός πάροχος υπηρεσιών: Η διαχείριση των υπολογιστικών και επικοινωνιακών πόρων από έναν μοναδικό πάροχο αφορά τις τιμές αυτών στους edge/cloudlet διακομιστές που εγκαταστάθηκαν σε διαφορετικές αποστάσεις από τις τερματικές συσκευές. Ο πάροχος ορίζει αυτές τις τιμές και ο τελικός χρήστης επιλέγει την χρήση των πόρων που του είναι απαραίτητοι και κατάλληλοι, ώστε να εξασφαλίσει το δυνατότερο χαμηλό οικονομικό κόστος και να μεταφέρει τον απαιτούμενο φόρτο των εργασιών.
  - Πολλαπλοί πάροχοι υπηρεσιών: Μέσω του IoT πραγματοποιείται σύνδεση ποικίλων διαφορετικών συσκευών που διαφέρουν και ως προς την πλευρά που ανήκουν. Αυτό μπορεί να σημαίνει, πως η κάθε εργασία για την επεξεργασία των δεδομένων του χρήστη αναλαμβάνεται από διαφορετικούς παρόχους, άρα, ενδεχομένως υπάρχει και διαφορετική χρέωση των υπηρεσιών που διαθέτει ο καθένας τους. Επομένως, μέσω της ορθής τιμολογιακής πολιτικής από τους αντίστοιχους παρόχους, επιτυγχάνεται η



επιβράβευση τους μέσω των υπηρεσιών και των πληρωμών από τους εκάστοτε τελικούς χρήστες., καθιστώντας την ως μία ανερχόμενη πολιτική για τα edge computing δίκτυα.

3. Προτεραιότητα: Η προτεραιότητα των εργασιών στοχεύει στην μεγιστοποίηση των οφελών από τις IoT εφαρμογές βάσει προγραμματισμού τους. Με αυτή την τακτική, κατατάσσονται οι εφαρμογές κατά βαθμό προτεραιότητας ώστε επί παραδείγματι οι εφαρμογές που είναι πραγματικού χρόνου να έχουν μεγαλύτερο βαθμό, λόγω της μικρότερης κατανάλωσης πόρων που έχουν. Αντίθετα, οι εφαρμογές που καταναλώνουν μεγάλο ποσοστό πόρων, να είναι χαμηλής προτεραιότητας, ελέγχοντας κατά αυτόν τον τρόπο την συνολική απόδοση.

#### 4.3 ΥΠΟΛΟΓΙΣΤΙΚΗ ΑΙΧΜΗΣ ΚΑΙ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

Ο ρόλος του Edge Computing αποδεικνύεται απαραίτητος για διασυνδεδεμένες εφαρμογές, οπότε ενισχύει το IoT μειώνοντας τον όγκο των δεδομένων που θα σταλθούν στο νέφος και τον χρόνο για να επιτευχθεί η πρόσβαση σε κάποια υπηρεσία. Παρακάτω, ακολουθούν τα βασικά γνωρίσματά του στο περιβάλλον των IoT, όπως είναι η συλλογή, η ανάλυση και η ασφάλεια των δεδομένων, οι έλεγχοι και η λήψη των αποφάσεων [50].

- Συλλογή δεδομένων: Οι Edge Computing συσκευές, συμπεριλαμβανομένων των αισθητήρων και των μηχανών, έχουν την δυνατότητα εξαγωγής δεδομένων για γρήγορη ανάλυση και εκτέλεσης άμεσων ενεργειών ή επεξεργασίας των δεδομένων με την μεταφορά να γίνεται από τον αλγόριθμο προς τα δεδομένα και όχι αντίστροφα. Αυτό έχει ως συνέπεια να αυξάνει η παραγωγικότητα και να αποφεύγονται τα ελαττώματα των προϊόντων αποδοτικά και ταχύτατα.
- Συμπερασματικοί Έλεγχοι: Οι συμπερασματικοί έλεγχοι αφορούν την ικανότητα μιας συσκευής να δίνει ερμηνεία στα πράγματα που βρίσκονται στο περιβάλλον της με υψηλή ακρίβεια. Αυτοί οι έλεγχοι συνεργάζονται επίσης με μια υποδομή η οποία ελέγχεται από άλλες οντότητες. Ωστόσο, θεωρείται δύσκολο να αποδοθεί συμπερασματικά η ικανότητα των συσκευών ακροδικτύου καθώς είναι εξαρτώμενη από τις πληροφορίες πλαισίου.

- **Ανάλυση Δεδομένων:** Μέσω της τεχνολογίας του Edge Computing γίνεται δυνατή η ανάλυση δεδομένων σε πραγματικό χρόνο. Όταν η ανάλυση δεδομένων συμβαίνει στο ίδιο σημείο στο οποίο εκείνα παράγονται, μειώνεται η καθυστέρηση μεταξύ της παραγωγής και της συλλογής τους. Έτσι, οι Edge Computing συσκευές έχουν την δυνατότητα συλλογής και ανάλυσης δεδομένων από τις περιβαλλόμενες συσκευές και κατά αυτόν τον τρόπο εξυπηρετούνται αυτοί που είναι υπεύθυνοι για την λήψη αποφάσεων καθώς τους παρέχονται πληροφορίες σπουδαίας βαρύτητας σε ελάχιστο χρόνο. Επιπλέον, οι συσκευές ακροδικτύου, χρησιμοποιούνται για τη μείωση του εύρους ζώνης και κατά συνέπεια του κόστους, εφόσον γίνεται τοπική επεξεργασία δεδομένων.
- **Λήψη Αποφάσεων:** Εφόσον πραγματοποιηθεί η ανάλυση των δεδομένων τοπικά, το επόμενο βήμα είναι η λήψη στρατηγικών αποφάσεων. Σε ένα σύστημα έξυπνων μεταφορών, κάθε όχημα παράγει μια μεγάλη συλλογή δεδομένων ανά δευτερόλεπτο και απαιτεί επεξεργασία πραγματικού χρόνου και σωστές αποφάσεις. Προκειμένου να είναι εφικτή η επεξεργασία σε πραγματικό χρόνο, πρέπει να γίνεται τοπικά σε οποιαδήποτε συσκευή ακροδικτύου, καθώς στην περίπτωση όπου τα δεδομένα στέλνονται στο νέφος ο χρόνος απόκρισης θα είναι μεγάλος. Ως αποτέλεσμα, το όχημα είναι ικανό να λάβει την σωστή απόφαση ακαριαία για να αποφύγει ανεπιθύμητες καταστάσεις
- **Επαυξημένη Ασφάλεια Δεδομένων:** Κατά την διαδικασία αποστολής των δεδομένων προκειμένου να επεξεργαστούν, ο φόβος για διαρροή δεδομένων αυξάνεται. Η συλλογή και η ανάλυση δεδομένων γίνονται τοπικά στο Edge Computing με κάθε δραστηριότητα να ανιχνεύεται εύκολα καθώς δεν απαιτείται εξεζητημένη δρομολόγηση.

Τόσο το Edge Computing όσο και το IoT, αναπτύσσονται ταχύτατα και ανεξάρτητα το ένα από το άλλο, όμως αυτό δεν απαγορεύει την συμβολή του πρώτου στην μετεξέλιξη του δεύτερου και στην επίλυση των ζητημάτων που εμφανίζονται προς αντιμετώπιση. Το Edge Computing συμβάλλει στην βελτίωση των αποδόσεων του IoT, οδηγώντας έτσι σε μία διατύπωση που ωθεί στην ενοποίηση και στην κοινή εφαρμογή τους, ούτως ώστε τα οφέλη που μπορούν να παρέχουν Cloud και Edge Computing στο IoT να αποδίδονται εν ευθέτω χρόνο, αξιοποιώντας τα πλεονεκτήματα και τις δυνατότητες του καθενός με μειωμένη

υπολογιστική ισχύ. Καθώς υπάρχει μεγάλη ανάπτυξη του αριθμού των IoT συσκευών, είθισται να εξαλείφονται σταδιακά οι γραμμές που τα διαχωρίζουν και να οδηγηθούν σε μία κοινή πορεία.

Όσα προηγήθηκαν, θα συμβάλλουν στο να ενισχυθούν οι απαιτήσεις και οι δυνατότητες των IoT, όπως είναι η μετάδοση, η αποθήκευση και ο υπολογισμός, αφού αντιμετωπίζουν αρκετά προβλήματα στην επιτυχή υλοποίησή τους και μέχρι πρότινος ήταν αδύνατη η εξάλειψη των αρνητικών αποτελεσμάτων και των ανεπιτυχών προσπαθειών.

Οι IoT συσκευές συλλέγουν και μεταδίδουν δεδομένα σε υπολογιστικούς κόμβους μεγαλύτερης ισχύς, στους οποίους γίνεται η επεξεργασία και η περαιτέρω ανάλυση των αρχικών δεδομένων. Καθότι οι συσκευές IoT δεν έχουν ως κύρια απαίτησή τους την μεγάλη υπολογιστική ικανότητα, μπορούν να ικανοποιηθούν κατάλληλα και σε πραγματικό χρόνο από τους edge κόμβους με ελαχιστοποίηση της ενέργειας κατανάλωσης αποφορτίζοντας τις υπολογιστικές εργασίες [50].

#### 4.4 ΕΙΚΟΝΟΠΟΙΗΣΗ

Οι εφαρμογές του IoT που έχουν ως βάση το edge computing, λόγω της δυναμικής που προσφέρεται από τους κατανεμημένους πόρους, είναι ικανές για υποστήριξη σεναρίων εξυπηρέτησης.

Η τεχνολογία του edge computing στο IoT εισάγει ένα ενδιάμεσο επίπεδο στο συμβατικό μοντέλο IoT-Cloud, ενισχύοντας την γεφύρωση και την αλληλεπίδραση του νέφους με το IoT. Επί της ουσίας, αυτό το στοιχείο ακροδικτύου λειτουργεί ως μια μικρή ή μέσου μεγέθους οντότητα έχοντας ως στόχο του την παροχή επιπλέον υπολογιστικής ισχύος, αποθηκευτικού χώρου και δικτυακών πόρων στις εφαρμογές που είναι εγκατεστημένες στις IoT συσκευές, το ακροδίκτυο και το νέφος.

Ανεξαρτήτως σεναρίου οι λειτουργίες μπορούν και εκτελούνται, είτε είναι σε κυψελωτούς σταθμούς βάσης, είτε σε IoT πύλες ή πιο γενικά σε κόμβους που έχουν μικρή ισχύ και μικρά κέντρα δεδομένων. Όλα αυτά, δύναται να ανήκουν σε έναν χρήστη, σε έναν πάροχο νέφους ή σε έναν φορέα τηλεπικοινωνιών. Ως προς την αρχιτεκτονική του, το επίπεδο edge πρέπει να βρίσκεται σε αρμονική συνεργασία με τις cloud υπηρεσίες και τις IoT συσκευές,

έχοντας τον ρόλο μιας γέφυρας για τα στοιχεία που προϋποθέτουν ένα διακριτό τρόπο αλληλεπίδρασης.

Γενικότερα, το edge computing είχε ως στόχο την ικανοποίηση βασικών απαιτήσεων όπως είναι η δυνατότητα κλιμάκωσης, η πολύ-μίσθωση, η ασφάλεια, το απόρρητο και η ευελιξία. Πολύ σημαντικός είναι επίσης ο εφοδιασμός του επιπέδου ακροδικτύου με εργαλεία που μπορούν να προσφέρουν ευελιξία, αποδοτικότητα και αυτοματοποίηση στις υπηρεσίες που παρέχονται. Κατά συνέπεια, στα στοιχεία που απαρτίζουν το ακροδίκτυο πρέπει να είναι εφικτή η ενσωμάτωση μεθόδων παροχής υπηρεσιών οι οποίες χαρακτηρίζονται για την ανεξαρτησία τους από της εφαρμογής που διαχειρίζονται και τα μοτίβα επικοινωνιών που χρησιμοποιούν.

Για να υποστηρίζονται πολλαπλά επίπεδα, τα στοιχεία του edge computing οφείλουν να προσαρμόζονται σε διαφορετικούς τύπους κυκλοφορίας αλλά και μεταβαλλόμενες ανάγκες εφαρμογών. Η λύση είναι η διασφάλιση της βέλτιστης αντιστάθμισης μεταξύ των απαιτήσεων σχεδιασμού, των συγκεκριμένων στόχων απόδοσης, και την διαχείριση των εφαρμογών σε όλο το φάσμα της προσέγγισης IoT, που βασίζεται στο edge computing. Συνεπώς, οι τεχνολογίες εικονικοποίησης επιδιώκουν στην ικανοποίηση τέτοιων απαιτήσεων. Ένα άμεσο πλεονέκτημα που αναδύεται από την εφαρμογή εικονικοποίησης στο edge computing πεδίο είναι η αποφυγή της αυστηρής εξάρτησης με οποιαδήποτε τεχνολογία ή περίπτωση χρήσης.

Προκειμένου να προωθηθεί η ενοποίηση με το cloud, η εικονικοποίηση δίνει την δυνατότητα ανάπτυξης διαφόρων πλατφορμών, έτσι ώστε να υπάρχει ένα κοινό περιβάλλον εκτέλεσης μεταξύ νέφους, ακροδικτύου και resource constrained συσκευών IoT. Η αποδοτική εκτέλεση των εφαρμογών οφείλεται στο γεγονός πως αναπτύσσοντας πολλαπλές πλατφόρμες, καθίσταται εφικτή η κοινή γλώσσα επικοινωνίας μεταξύ νέφους και άκρου, όπου αυτή είναι βέλτιστη, οδηγώντας στην ανταπόκριση στις αυστηρές απαιτήσεις απόδοσης πολύπλοκων σεναρίων IoT και την διασφάλιση περαιτέρω της κρίσιμης απαίτησης της πολυ-μίσθωσης [51].

#### **4.5 ΕΦΑΡΜΟΓΕΣ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΑΙΧΜΗΣ**

Με την πάροδο του χρόνου, το Edge Computing βρίσκει χρήση σε πολλαπλούς κλάδους εφαρμογών που αναπτύσσονται και επιζητούν την εκμετάλλευση των πλεονεκτημάτων που προσφέρει, όπως είναι παραδείγματος χάρι η βιομηχανία, η ιατρική, η

ναυτιλία κα. Προς το παρόν, η χρήση του πραγματοποιείται σε έξυπνες εφαρμογές και σε εφαρμογές συστημάτων, που καθιστούν αναγκαία την μικρή χρονική απόκριση για την έκδοση αποτελεσμάτων με κάποιες από αυτές να είναι τα έξυπνα σπίτια (smart homes), τα έξυπνα δίκτυα ενέργειας (smart grid), οι έξυπνες πόλεις (smart cities), οι έξυπνες εφοδιαστικές (smart logistics), τα συστήματα υγείας (healthcare), τα συστήματα παρακολούθησης βίντεο (video surveillance) και η παρακολούθηση περιβάλλοντος (environment monitoring). Ακολουθεί μία μερική παρουσίασή τους [50].

- Έξυπνο Σπίτι: Τα έξυπνα σπίτια όπου διαθέτουν έναν μεγάλο εξοπλισμό από IoT συσκευές βρίσκονται σε ένα πεδίο εφαρμογών του Edge Computing που είναι σε φάση ακμής. Οι IoT εφαρμογές για παρακολούθηση και μετρήσεις στα έξυπνα σπίτια θα δώσουν την δυνατότητα στους συνδρομητές να αποκτήσουν αυτοματοποιημένες και ακριβείς παρουσιάσεις από διαφορετικούς μετρητές και να έχουν πρόσβαση στα σχετικά τιμολόγια δίχως καθυστερήσεις. Ο σχεδιασμός αυτών των εφαρμογών είναι κατάλληλος για απομακρυσμένη παρακολούθηση και μέτρηση πολλών διαφορετικών πόρων όπως είναι το νερό, ο ηλεκτρισμός και το αέριο. Αυτή η συλλογή δεδομένων μπορεί να μεταφερθεί από τις συσκευές σε ένα διακομιστή ακροδικτύου προκειμένου να γίνει η επεξεργασία τους και η ανάλυση δεδομένων σε πραγματικό χρόνο.
- Έξυπνο Δίκτυο Ενέργειας: Λόγω των αυτοματοποιημένων μοτίβων στην κατανάλωση και την διανομή σε εφαρμογές τύπου Edge Computing και IoT, γίνεται χρήση τους για την έξυπνη διαχείριση της ενέργειας. Κόμβοι του Edge Computing κάνουν συνεισφορά στην συλλογή και επεξεργασία πραγματικού χρόνου αλλά και στην βελτίωση της ταχύτητας και της κατανομή φόρτου, ενώ το Cloud Computing μπορεί επίσης να λειτουργήσει ως ένα συνεργατικό εργαλείο, ώστε να είναι δυνατή η διαχείριση τεράστιου όγκου δεδομένων κατά την εγκατάσταση δικτύων ενέργειας σε μεγάλες περιοχές.
- Έξυπνες Πόλεις: Το Edge Computing στο IoT μπορεί επίσης να χρησιμοποιηθεί για το σχεδιασμό έξυπνων πόλεων. Πεδία εφαρμογής του περιλαμβάνουν τα συστήματα ελέγχου φωτισμού των δρόμων και οδών, τα συστήματα μέτρησης ποιότητας αέρα και νερού, η εξερεύνηση εξόδων ή διαδρομής διαφυγής σε

περίπτωση καταστροφής ή ατυχήματος και συστήματα ποτίσματος κήπων σε πόλεις.

- Έξυπνη Εφοδιαστική: Το Edge Computing στο IoT μπορεί επίσης να συμβάλει στην παραδοσιακή εφοδιαστική και να προσφέρει καινοτόμες και πολλά υποσχόμενες δυνατότητες που αυτοματοποιούν και διευκολύνουν την διαχείριση της ροής των πραγμάτων. Μέσω αυτού του συστήματος η ροή των συναλλαγών πραγματοποιείται ομαλά μεταξύ του παραγωγού του προϊόντος και του τελικού καταναλωτή σχετικά με τις παραμέτρους του κόστους και του χρόνου.
- Σύστημα Υγείας: Ένα ακόμα πεδίο εφαρμογής του Edge Computing είναι η ιατρική. Επιτρέπει στον τελικό χρήστη την αξιολόγηση των δεδομένων που εξάγονται, ώστε να γίνει πρόσληψη και αντίδραση ανάλογη των αποτελεσμάτων. Οι εφαρμογές υγείας χαρακτηρίζονται από την ευαισθησία που διαθέτουν ως προς την καθυστέρηση, με αποτέλεσμα να μην είναι ικανή μόνο η χρήση του Cloud Computing να την μειώσει αυτή την ευαισθησία, άρα να είναι απαραίτητη η συμβολή του Edge Computing.
- Σύστημα Παρακολούθησης Βίντεο: Μια από τις εφαρμογές του Edge Computing είναι η έξυπνη παρακολούθηση βίντεο για ποικίλους τομείς της καθημερινότητας, συμπεριλαμβανομένης της οικιακής ασφάλειας και της αντιτρομοκρατικής. Στην συνέχεια γίνεται αποθήκευση αυτών των βίντεο και η αποδοτική τους διαχείριση για εκτενέστερη επεξεργασία. Η εξαγωγή των απαραίτητων δεδομένων μπορεί να γίνει αυτόματα από τα αποθηκευμένα βίντεο μέσω διαφόρων εφαρμογών ασφάλειας.
- Παρακολούθηση Περιβάλλοντος: Το Edge Computing σε συνεργασία με το Cloud Computing στο IoT ενισχύουν την ποιότητα των υπάρχοντων συστημάτων παρακολούθησης. Ένα τέτοιο αυτοματοποιημένο σύστημα δουλεύει με την χρήση αισθητήρων και ενεργοποιητών. Ο σκοπός ανάπτυξης των εφαρμογών είναι η παρακολούθηση κρίσιμων οντοτήτων, οι οποίες έχουν σημαντική επίδραση στο περιβάλλον [52].

## 5.0 ΕΦΑΡΜΟΓΗ

Για την παρούσα εργασία υλοποιήθηκε εφαρμογή η οποία κάνει ανίχνευση μάσκας για ανθρώπινα πρόσωπα σε πραγματικό χρόνο. Σε αυτό το κεφάλαιο περιγράφονται αναλυτικά όλα τα εργαλεία, υλικού και λογισμικού, που χρησιμοποιήθηκαν, καθώς και η διαδικασία που ακολουθήθηκε για να λειτουργήσει το μοντέλο και να φέρει αποτέλεσμα.

### 5.1 ΥΛΙΚΟ

Στο υλικό (hardware) μέρος της εργασίας χρησιμοποιήθηκε το Jetson Nano Developer Kit της NVIDIA για την εκπαίδευση και τον έλεγχο του μοντέλου. Με την εισαγωγή του Jetson Nano™, η NVIDIA επιτρέπει την ανάπτυξη εκατομμυρίων νέων συσκευών τεχνητής νοημοσύνης (Artificial Intelligence – AI) που λειτουργούν αποτελεσματικά σε μια μικρών διαστάσεων υπολογιστική μηχανή αιχμής (edge computing machine).

Το Jetson Nano είναι ένας μικρός υπολογιστής τεχνητής νοημοσύνης, ο οποίος μπορεί να επεξεργάζεται δεδομένα από αισθητήρες υψηλής ανάλυσης, να περιεργάζεται πολλαπλούς αισθητήρες ταυτόχρονα και να τρέχει πολλαπλά νευρωνικά δίκτυα ανά αισθητήρα. Αυτό καθιστά εφικτό έναν ολόκληρο κόσμο νέων ενσωματωμένων εφαρμογών, συμπεριλαμβανομένων των Δικτύων Εγγραφής Βίντεο (Network Video Recorder - NVR) πρωταρχικού σταδίου, των οικιακών ρομπότ και των ευφυών πυλών (gateways) με δυνατότητες πλήρης ανάλυσης.

Η δύναμη του AI, που γεννήθηκε από υπερυπολογιστές, είναι πλέον διαθέσιμη στη μαζική αγορά ενσωματωμένων - μόνο με το Jetson Nano. Όλα ξεκινούν με ένα έτοιμο για παραγωγή μικρών διαστάσεων (69x45mm) System on Module (SOM) από τη NVIDIA. Έρχεται με GPU NVIDIA Maxwell 128 πυρήνων, τετραπύρηνο σύστημα επεξεργασίας ARM A57, κωδικοποιητή και αποκωδικοποιητή βίντεο και μνήμη 4 GB LPDDR4 και 16 GB eMMC. Διαθέτει επίσης πλήθος διεπαφών και IO (Inputs-Outputs), συμπεριλαμβανομένων IO υψηλής ταχύτητας για CSI, PCIe, Gigabit Ethernet και USB3, διεπαφές βίντεο, όπως HDMI και DisplayPort και ένα τυπικό IO για I2C, I2S, SPI και GPIO.

Το Jetson Nano έχει σχεδιαστεί για να μειώνει τον συνολικό χρόνο ανάπτυξης και να φέρνει προϊόντα στην αγορά γρηγορότερα, μειώνοντας τον χρόνο που αφιερώνεται για τον σχεδιασμό του υλικού, τον έλεγχο και την επικύρωση ενός πολύπλοκου, ισχυρού και

ενεργειακά αποδοτικού συστήματος AI. Ο σχεδιασμός συνοδεύεται από τη διαχείριση ενέργειας, τον χρονισμό (clocking), τη μνήμη και τα πλήρως προσβάσιμα IO. Οι προγραμματιστές μπορούν εύκολα να συνδέσουν τους αισθητήρες και το IO και να ξεκινήσουν την ανάπτυξη τμημάτων του συστήματος που αφορούν την εφαρμογή. Μαζί με την εκκίνηση (boot) και το πρόγραμμα flash, το Jetson Nano δίνει τη δυνατότητα στους προγραμματιστές απλά και τακτικά να ενημερώνουν τους αλγορίθμους τους κατά τη διάρκεια της ανάπτυξης, ακόμη και μετά την εφαρμογή του συστήματος.

Το Jetson Nano υποστηρίζει πολλά δίκτυα βαθιάς εκμάθησης, συμπεριλαμβανομένων των ResNet-50, SSD Mobilnet-V2, enet, Tiny YOLO V3, Posenet, VGG-19, Super Resolution, Unet και άλλων. Αυτά τα μοντέλα χρησιμοποιούνται για ταξινόμηση, ανίχνευση αντικειμένων, τμηματοποίηση (segmentation), εκτίμηση πόζας (pose estimation), πρόβλεψη συντήρησης (predictive maintenance), επεξεργασία εικόνας και άλλα. Επιπλέον, το Jetson Nano υποστηρίζει frameworks όπως το Caffe, το TensorFlow, το PyTorch, το Darknet, το MXNet και το Keras [53].

## 5.2 ΛΟΓΙΣΜΙΚΟ

Για την διαδικασία ανάπτυξης της εφαρμογής χρησιμοποιήθηκαν διάφορα εργαλεία λογισμικού στον υπολογιστή αλλά και στο Jetson Nano τα οποία θα αναλυθούν στην συνέχεια με την χρονική σειρά που έλαβε χώρα η εφαρμογή.

### *Google Colaboratory*

Έχοντας διαθέσιμα τα δεδομένα τα οποία είναι φωτογραφίες προσώπων με μάσκα το πρώτο και βασικό βήμα που πρέπει να γίνει είναι η επεξεργασία τους σε αναγνώσιμη μορφή για τα εργαλεία που θα εκπαιδευτούν πάνω σε αυτά, καθώς και ο διαχωρισμός τους σε δεδομένα εκπαίδευσης, επικύρωσης και ελέγχου. Αυτή η διαδικασία πραγματοποιείται στο Google Colab.

Το Colaboratory ή για συντομία «Colab», είναι προϊόν της Google Research. Το Colab επιτρέπει σε οποιονδήποτε να γράψει και να εκτελέσει αυθαίρετο κώδικα σε γλώσσα python μέσω του προγράμματος περιήγησης και είναι ιδιαίτερα κατάλληλο για μηχανική μάθηση, ανάλυση δεδομένων και εκπαίδευση. Πιο τεχνικά, το Colab είναι μια υπηρεσία που φιλοξενεί



το Jupyter που δεν απαιτεί εγκατάσταση και ρύθμιση παραμέτρων, ενώ παρέχει δωρεάν πρόσβαση σε υπολογιστικούς πόρους, συμπεριλαμβανομένων των GPU.

Το Colab είναι ελεύθερο για χρήση και για αυτόν το λόγο αυτοί οι πόροι του δεν είναι εγγυημένοι και δεν είναι απεριόριστοι, καθώς ακόμα και τα όρια χρήσης μερικές φορές έχουν διακυμάνσεις. Αυτό είναι απαραίτητο για να μπορεί το Colab να παρέχει δωρεάν πόρους. Από την άλλη πλευρά, το Jupyter είναι το έργο ανοιχτού κώδικα στο οποίο έχει βασιστεί το Colab. Το Colab δίνει τη δυνατότητα στους χρήστες να χρησιμοποιούν και να διαμοιράζονται Jupyter notebooks με άλλους χωρίς να χρειάζεται να κάνουν λήψη, εγκατάσταση ή εκτέλεση σε κάτι [54].

### ***Jupyter***

Το Project Jupyter είναι ένα μη κερδοσκοπικό έργο ανοιχτού κώδικα, που γεννήθηκε από το IPython Project το 2014 καθώς εξελίχθηκε για να υποστηρίξει την επιστήμη των διαδραστικών δεδομένων και τον επιστημονικό υπολογισμό σε όλες τις γλώσσες προγραμματισμού. Το Jupyter θα είναι πάντα 100% ανοιχτού κώδικα λογισμικό, δωρεάν για όλους και θα κυκλοφορεί υπό τους φιλελεύθερους όρους της τροποποιημένης άδειας BSD.

Το Jupyter αναπτύσσεται ανοιχτά στο GitHub, μέσω της συναίνεσης της κοινότητας του Jupyter. Όλες οι διαδικτυακές και προσωπικές αλληλεπιδράσεις και επικοινωνίες που σχετίζονται άμεσα με το έργο καλύπτονται από τον Κώδικα Δεοντολογίας Jupyter. Αυτός ο Κώδικας Δεοντολογίας θέτει προσδοκίες για να επιτρέψει σε μια κοινότητα ποικίλων χρηστών και συντελεστών να συμμετέχουν στο έργο με σεβασμό και ασφάλεια [55].

Τα Jupyter Notebook είναι ένα από τα πιο σημαντικά εργαλεία για τους επιστήμονες δεδομένων που χρησιμοποιούν Python. Αυτό οφείλεται στο γεγονός ότι αποτελούν ιδανικό περιβάλλον για την ανάπτυξη εύκολης αναπαραγωγής αγωγών (pipelines) ανάλυσης δεδομένων. Τα δεδομένα μπορούν να φορτωθούν, να μετατραπούν και να διαμορφωθούν όλα μέσα σε ένα Notebook, όπου είναι γρήγορο και εύκολο κάποιος χρήστης να δοκιμάσει τον κώδικα και να εξερευνήσει ιδέες στην πορεία. Επιπλέον, όλα αυτά μπορούν να θεωρηθούν "ενσωματωμένα" χρησιμοποιώντας μορφοποιημένο κείμενο, ώστε να μπορεί ο χρήστης να κάνει σημειώσεις για τον εαυτό του ή ακόμα και να δημιουργήσει μια δομημένη αναφορά.

Άλλες συγκρίσιμες πλατφόρμες, για παράδειγμα, το RStudio ή το Spyder παρουσιάζουν στον χρήστη πολλά παράθυρα, τα οποία προωθούν δύσκολες εργασίες, όπως αντιγραφή και επικόλληση κώδικα και επανεκτέλεση κώδικα που έχει ήδη εκτελεστεί. Αυτά

τα εργαλεία τείνουν επίσης να εμπλέκουν βρόχους προτροπής ανάγνωσης (Read Eval Prompt Loops - REPL) όπου ο κώδικας εκτελείται σε ένα τερματικό που έχει αποθηκεύσει η μνήμη. Αυτός ο τύπος περιβάλλοντος ανάπτυξης δεν είναι καλός για την εύκολη αναπαραγωγή και δεν είναι ιδανικός ούτε για ανάπτυξη κώδικα. Τα Jupyter Notebook επιλύουν όλα αυτά τα ζητήματα δίνοντας στον χρήστη ένα μόνο παράθυρο όπου εκτελούνται αποσπάσματα κώδικα και οι έξοδοι εμφανίζονται εσωτερικά. Αυτό επιτρέπει στους χρήστες να αναπτύσσουν τον κώδικα αποτελεσματικά και τους δίνει τη δυνατότητα να ανατρέχουν στην προηγούμενη εργασία για αναφορά ή ακόμα και να κάνουν αλλαγές [56].

## ***Python***

Η Python είναι μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου. Μπορεί να χειριστεί διάφορες εργασίες προγραμματισμού όπως τον αριθμητικό υπολογισμό, την ανάπτυξη ιστού (web), τον προγραμματισμό βάσεων δεδομένων, τον προγραμματισμό δικτύων, την παράλληλη επεξεργασία κ.λπ. Η Python είναι δημοφιλής για διάφορους λόγους, όπως:

1. Είναι δωρεάν.
2. Διατίθεται σε όλα τα δημοφιλή λειτουργικά συστήματα όπως Windows, Mac ή Linux.
3. Είναι μια γλώσσα που ερμηνεύεται. Ως εκ τούτου, οι προγραμματιστές μπορούν να δοκιμάσουν τμήματα κώδικα στη γραμμή εντολών πριν τον ενσωματώσουν στο πρόγραμμά τους. Δεν χρειάζεται μεταγλώττιση ή σύνδεση (linking).
4. Δίνει τη δυνατότητα γρηγορότερου προγραμματισμού.
5. Είναι συντακτικά απλούστερη από τις C/C++/Fortran. Γι' αυτό τον λόγο, είναι πολύ ευανάγνωστη και πιο εύκολη στο να διορθωθεί.
6. Έρχεται με διάφορες ενότητες προγράμματος (modules) που είναι τυπικές ή μπορούν να εγκατασταθούν σε υπάρχουσα εγκατάσταση Python. Αυτές οι ενότητες μπορούν να εκτελέσουν διάφορες εργασίες όπως η ανάγνωση και εγγραφή διαφόρων αρχείων, ο επιστημονικός υπολογισμός, η απεικόνιση δεδομένων κ.λπ.
7. Τα προγράμματα που είναι γραμμένα σε Python μπορούν να εκτελεστούν σε διάφορα λειτουργικά συστήματα ή πλατφόρμες έχοντας υποστεί μικρές αλλαγές ή και καθόλου.

8. Είναι μια γλώσσα δυναμικά γραμμένη. Επομένως, ο τύπος δεδομένων των μεταβλητών δεν χρειάζεται να δηλωθεί πριν από τη χρήση τους, διευκολύνοντας έτσι τα άτομα που έχουν λιγότερη εμπειρία με το να συντάσσουν κώδικα.
9. Διαθέτει μια κοινότητα προγραμματιστών και χρηστών αφοσιωμένη στο να κρατάει ενήμερη την γλώσσα.

Παρόλο που η Python έχει πολλά πλεονεκτήματα που την κατέστησαν μία από τις πιο δημοφιλείς γλώσσες, έχει μερικά μειονεκτήματα που συζητούνται παρακάτω:

- Δεδομένου ότι εστιάζει στη δυνατότητα του ταχύτερου προγραμματισμού, η ταχύτητα στην εκτέλεση πάσχει. Ένα πρόγραμμα Python μπορεί να είναι 10 ή παραπάνω φορές πιο αργό από ένα ισοδύναμο πρόγραμμα σε C για παράδειγμα, αλλά θα περιέχει 5 λιγότερες γραμμές κώδικα και μπορεί να προγραμματιστεί, ώστε να χειρίζεται εύκολα πολλούς τύπους δεδομένων. Αυτό το μειονέκτημα στον κώδικα της Python μπορεί να ξεπεραστεί κάνοντας μετατροπή των τμημάτων του κώδικα που απαιτεί υψηλή υπολογιστική ισχύ σε C/C++ ή με την κατάλληλη χρήση της δομής δεδομένων και των modules.
- Οι εσοχές κενών στον κώδικα δεν είναι προαιρετικές. Αυτό καθιστά τον κώδικα ευανάγνωστο. Ωστόσο, ένας κώδικας με πολλαπλούς βρόχους και άλλες διακλαδώσεις θα διαθέτει αλλεπάλληλες εσοχές στα δεξιά, καθιστώντας δύσκολη την ανάγνωση του.

Υπάρχουν πολλά περιβάλλοντα Python από τα οποία μπορεί να επιλέξει κανείς. Ορισμένα λειτουργικά συστήματα όπως το Mac, το Linux, το Unix κ.λπ. διαθέτουν ενσωματωμένο διερμηνέα. Ο διερμηνέας μπορεί να περιέχει όλα τα modules, αλλά δεν είναι έτοιμος για επιστημονικούς υπολογισμούς. Δημιουργήθηκαν και πωλήθηκαν εξειδικευμένες διανομές στην επιστημονική κοινότητα, προσχεδιασμένες με διάφορες επιστημονικά modules για Python. Κατά τη χρήση αυτών των διανομών, οι χρήστες δεν χρειάζεται να εγκαταστήσουν μεμονωμένα επιστημονικά modules. Εάν ένα συγκεκριμένο module που ενδιαφέρει κάποιον χρήστη δεν είναι διαθέσιμο στη διανομή, μπορεί να εγκατασταθεί μετέπειτα. Μία από τις πιο δημοφιλείς διανομές είναι η Anaconda [57].

Η Anaconda Python Distribution [58] παρέχει στους προγραμματιστές σχεδόν 100 από τα πιο δημοφιλή επιστημονικά modules σε Python, όπως αυτά του επιστημονικού υπολογισμού, της γραμμικής άλγεβρας, του συμβολικού υπολογισμού, της επεξεργασίας εικόνας, της επεξεργασίας σήματος, της οπτικοποίησης, την ενσωμάτωση προγραμμάτων

C/C++ σε Python κ.λπ. διανέμεται και συντηρείται από την Continuum Analytics. Είναι διαθέσιμο δωρεάν για ακαδημαϊκούς και διατίθεται με τιμή για όλους τους άλλους. Εκτός από τα διάφορα modules που είναι ενσωματωμένα στο Anaconda, οι προγραμματιστές μπορούν να εγκαταστήσουν κι άλλα modules χρησιμοποιώντας τον διαχειριστή πακέτων conda [57], χωρίς να επηρεάσουν την κύρια διανομή.

Δημιουργήθηκαν και υπάρχουν πολλά επιστημονικά modules Python διατίθεται στις διανομές Python. Μερικές από τις πιο δημοφιλή modules είναι:

1. `numpy`: Μια ισχυρή βιβλιοθήκη για τον χειρισμό συστοιχιών και πινάκων.
2. `scipy`: Παρέχει λειτουργίες για την εκτέλεση μαθηματικών πράξεων υψηλότερης τάξης, όπως φιλτράρισμα (filtering), στατιστική ανάλυση, επεξεργασία εικόνας κ.λπ.
3. `matplotlib`: Παρέχει λειτουργίες για σχεδιασμό και άλλες μορφές απεικόνισης.
4. Βιβλιοθήκη Απεικόνισης Python (Python Imaging Library): Παρέχει λειτουργίες για βασική ανάγνωση, γραφή και επεξεργασία εικόνας.
5. `scikits`: Ένα πρόσθετο πακέτο για το `scipy`. Τα modules στο `scikit` προορίζονται να προστεθούν στο `scipy` μετά την ανάπτυξη του προγράμματος [59].

## **YOLO**

Ο αλγόριθμος ανίχνευσης αντικειμένων με τη μέθοδο ενός σταδίου επιτυγχάνει πραγματικά εκπαίδευση από άκρο σε άκρο, όπως είχε αναφερθεί και στο τέλος του 1<sup>ου</sup> Κεφαλαίου. Αντιπροσωπευμένος από το YOLO (You Only Look Once), ο αλγόριθμος ανίχνευσης στόχων με βάση την παλινδρόμηση ολοκληρώνει τον προσδιορισμό της κατηγορίας στόχου και τη θέση του στόχου ταυτόχρονα. Ολόκληρη η δομή του δικτύου αποτελείται μόνο από περιστρεφόμενα στρώματα και την εικόνα εισόδου. Μετά τη λειτουργία της περιστροφής, η κατηγορία-στόχος και η θέση επιστρέφονται απευθείας. Επομένως, ο αλγόριθμος ανίχνευσης στόχων ενός σταδίου είναι ταχύτερος από έναν αλγόριθμο ανίχνευσης στόχων δύο σταδίων, ειδικά του YOLOv5, του οποίου έχει φτάσει σε προηγμένο επίπεδο ταχύτητας και ακρίβειας [60]. Στην παρούσα εργασία χρησιμοποιείται το YOLOv5.

Με την ανάπτυξη της τεχνολογίας, το YOLO επαναλαμβάνεται επίσης για να γίνει ταχύτερο, ισχυρότερο και καλύτερο. Τώρα η σειρά YOLO περιλαμβάνει τα YOLOv1, YOLOv2, YOLOv3, YOLOv4 και YOLOv5. Το YOLOv1 αναπτύσσεται με βάση το R-CNN. Είναι γνωστό ότι το R-CNN (Region Proposals + CNN) χρησιμοποιεί ένα συνελκτικό

νευρωνικό δίκτυο για την ανίχνευση στόχων και ταυτόχρονα, προσθέτει ένα SVM (Support Vector Machine) για πρόβλεψη ταξινόμησης. Η ανίχνευση θέσης και η ακρίβεια ταξινόμησης αντικειμένων είναι πολύ υψηλή. Αλλά ο υπολογισμός είναι βαρύς και η ταχύτητα ανίχνευσης είναι πολύ αργή.

Το YOLOv1 βασίζεται στο R-CNN. Οι εικόνες εισαγωγής υποβάλλονται σε επεξεργασία μόνο μία φορά (όπως και στην προέλευση του ονόματος του YOLO), εξάγονται διαφορετικά χαρακτηριστικά μέσω πολλαπλών συνελκτικών στρωμάτων και οι συνελκτικοί παράμετροι πυρήνα μοιράζονται κάθε φορά [61]. Η ταχύτητα ανίχνευσης εικόνας είναι πολύ γρήγορη. Είναι ταχύτερο από το προηγούμενο μοντέλο ανίχνευσης και μπορεί να καλύψει τις απαιτήσεις σε πραγματικό χρόνο. Μπορεί να φτάσει τα 45 καρέ ανά δευτερόλεπτο, χρησιμοποιώντας την πλήρη εικόνα ως περιβάλλουσα πληροφορία. Ωστόσο, το μειονέκτημα είναι ότι η ακρίβεια ανίχνευσης θέσης είναι χαμηλή και δεν μπορούν να ανιχνευθούν μικρά αντικείμενα.

Το YOLOv2 βελτιώνεται του YOLOv1. Το δίκτυο αναβαθμίζεται. Το YOLOv2 χρησιμοποιεί μέση συγκέντρωση (pooling), ταξινόμηση SoftMax και πλαίσιο πρόβλεψης Anchor. Επίσης, προτείνεται ένας συνδυασμός εκπαιδευτικής μεθόδου ταξινόμησης και ανίχνευσης στόχων μαζί. Λόγω αυτών των βελτιώσεων, η ακρίβεια βελτιώνεται προφανώς, ειδικά για την ανίχνευση μικρών αντικειμένων, το μειονέκτημα είναι ότι η ακρίβεια ανίχνευσης μικρών αντικειμένων δεν είναι και τόσο υψηλή.

Το YOLOv3 κάνει κάποιες βελτιώσεις. Το συνελκτικό στρώμα του YOLOv3 είναι περίπου 2,8 φορές μεγαλύτερο από αυτό του YOLOv2, αυξάνοντας το βάθος και το πάχος του δικτύου και αυξάνοντας έτσι την ακρίβεια του μοντέλου. Οι ταξινομητές SoftMax αντικαθίστανται από πολλαπλούς ταξινομητές Logistic [62].

Το YOLOv4 εμφανίστηκε το 2019. Ο κύριος σκοπός είναι ο σχεδιασμός ενός γρήγορου συστήματος ανίχνευσης στόχων που μπορεί να εφαρμοστεί σε πραγματικό εργασιακό περιβάλλον και να βελτιστοποιείται παράλληλα. Χρησιμοποιεί βελτιστοποίηση δεδομένων και μερικά από τα τελευταία tricks δικτύων βαθιάς εκμάθησης των τελευταίων χρόνων. Όπως βελτιστοποίηση δεδομένων CutMix, Swish και συναρτήσεις ενεργοποίησης Mish [63].

Το YOLOv5 είναι το τελευταίο προϊόν της σειράς YOLO. Το YOLOv5 είναι βελτιωμένο σε σχέση με το YOLOv4 και η ταχύτητα λειτουργίας του βελτιώνεται σημαντικά, με την μέγιστη ταχύτητα να αγγίζει τα 140 καρέ ανά δευτερόλεπτο. Ωστόσο, το μέγεθος του YOLOv5 είναι μικρό και το βάρος αρχείου είναι σχεδόν 90% μικρότερο από αυτό του

YOLOv4, το οποίο επιτρέπει στο YOLOv5 να αναπτυχθεί σε ενσωματωμένες συσκευές. Σε σύγκριση με το YOLOv4, το YOLOv5 έχει υψηλότερο ποσοστό ακρίβειας και καλύτερη ικανότητα αναγνώρισης μικρών αντικειμένων [64].

Το YOLO για να λειτουργήσει σωστά και να πραγματοποιηθεί η διαδικασία της εκπαίδευσης του μοντέλου χρειάζεται δύο frameworks. Αυτά είναι το OpenCV και το PyTorch, όπου και αναλύονται στη συνέχεια.

### *OpenCV*

Το OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη όρασης υπολογιστών ανοιχτού κώδικα, με άδεια BSD και περιλαμβάνει υλοποίηση εκατοντάδων αλγορίθμων όρασης υπολογιστή. Η βιβλιοθήκη, που προορίζεται κυρίως για όραση σε πραγματικό χρόνο, αναπτύχθηκε από την έρευνα της Intel Russia και τώρα υποστηρίζεται ενεργά από την Itseez. Το 2016, η Intel απέκτησε την Itseez.

Το OpenCV γράφεται κυρίως σε C και C++ και η κύρια διεπαφή του είναι σε C++. Έχει επίσης καλές διεπαφές σε Python, Java και MATLAB/Octave, και επίσης έχει wrappers σε άλλες γλώσσες (όπως C# και Ruby). Στην τελευταία έκδοση του OpenCV, υπάρχει υποστήριξη για CUDA και OpenCL για ενεργοποίηση της επιτάχυνσης GPU. Το OpenCV εκτελείται στις περισσότερες πλατφόρμες λειτουργικού συστήματος (όπως Windows, Linux, Mac OS X, Android, FreeBSD, OpenBSD, iOS και BlackBerry) [65].

Η όραση υπολογιστή (computer vision) είναι μια συλλογή αλγορίθμων που επιτρέπουν στον υπολογιστή να αναλύει μια εικόνα και να εξάγει χρήσιμες πληροφορίες. Χρησιμοποιείται σε πολλές εφαρμογές και γίνεται γρήγορα μέρος της καθημερινής ζωής. Οι περισσότερες νέες κάμερες μέτριας έως υψηλής τεχνολογίας έχουν ενσωματωμένη ανίχνευση προσώπου στα περισσότερα smartphones. Το Facebook χρησιμοποιεί την όραση του υπολογιστή για την ανίχνευση προσώπου. Η όραση υπολογιστών χρησιμοποιείται από τις ναυτιλιακές εταιρείες για την παρακολούθηση πακέτων στις αποθήκες τους. Και, φυσικά, χρησιμοποιείται στη ρομποτική για πλοήγηση, ανίχνευση αντικειμένων, αποφυγή αντικειμένων και πολλές άλλες συμπεριφορές.

Όλα ξεκινούν με μια εικόνα. Ο υπολογιστής αναλύει μια εικόνα για να εντοπίσει γραμμές, γωνίες και μια ευρεία περιοχή χρώματος. Αυτή η διαδικασία ονομάζεται εξαγωγή χαρακτηριστικών και είναι το πρώτο βήμα σχεδόν σε όλους τους αλγόριθμους όρασης

υπολογιστή. Μόλις εξαχθούν τα χαρακτηριστικά, ο υπολογιστής μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για πολλές διαφορετικές εργασίες. Η αναγνώριση προσώπου επιτυγχάνεται με τη σύγκριση των αυτών των χαρακτηριστικών σε σχέση δεδομένα χαρακτηριστικών για πρόσωπα με που περιέχουν αρχεία XML. Αυτά τα αρχεία XML ονομάζονται καταρράκτες (cascades). Διατίθενται για πολλούς διαφορετικούς τύπους αντικειμένων, όχι μόνο για πρόσωπα.

Αυτή η ίδια τεχνική μπορεί να χρησιμοποιηθεί για την αναγνώριση αντικειμένων. Απλώς παρέχονται ήδη στην εφαρμογή πληροφορίες με χαρακτηριστικά για τα αντικείμενα που ενδιαφέρουν. Η όραση υπολογιστή ενσωματώνει επίσης βίντεο. Η παρακολούθηση κίνησης (motion tracking) είναι μια κοινή εφαρμογή για την όραση υπολογιστή. Για την ανίχνευση κίνησης, ο υπολογιστής συγκρίνει μεμονωμένα καρέ από μια στατική κάμερα. Εάν δεν υπάρχει κίνηση, τα χαρακτηριστικά δεν θα αλλάξουν μεταξύ των πλαισίων. Έτσι, εάν ο υπολογιστής εντοπίσει διαφορές μεταξύ των καρέ, τότε πιθανότατα υπάρχει κίνηση.

Μια συναρπαστική, πρόσφατη εφαρμογή της όρασης υπολογιστή είναι η επαυξημένη πραγματικότητα. Τα εξαγόμενα χαρακτηριστικά από μια ροή βίντεο μπορούν να χρησιμοποιηθούν για τον προσδιορισμό ενός μοναδικού μοτίβου σε μια επιφάνεια. Επειδή ο υπολογιστής γνωρίζει το μοτίβο, μπορεί εύκολα να υπολογίσει τη γωνία της επιφάνειας. Στη συνέχεια, ένα τρισδιάστατο μοντέλο υπερτίθεται στο μοτίβο. Αυτό το τρισδιάστατο μοντέλο θα μπορούσε να είναι κάτι φυσικό, όπως ένα κτίριο, ή θα μπορούσε να είναι ένα επίπεδο αντικείμενο με δισδιάστατο κείμενο. Οι αρχιτέκτονες χρησιμοποιούν αυτήν την τεχνική για να δείξουν στους πελάτες πώς θα ήταν ένα κτίριο ως περίγραμμα. Τα μουσεία το χρησιμοποιούν για να παρέχουν περισσότερες πληροφορίες σχετικά με ένα έκθεμα ή έναν καλλιτέχνη [66].

## ***PyTorch***

Το PyTorch είναι μια προσφορά ανοιχτού κώδικα από το Facebook που διευκολύνει τη συγγραφή κώδικα βαθιάς εκμάθησης στην Python. Έχει δύο γενεές. Πρώτον, και ίσως όχι εντελώς τυχαία δεδομένου του ονόματός του, αντλεί πολλά χαρακτηριστικά και έννοιες από το Torch, το οποίο ήταν μια βιβλιοθήκη νευρωνικών δικτύων που βασίζεται στη Lua και χρονολογείται από το 2002. Ο άλλος κύριος γονέας του είναι το Chainer, που δημιουργήθηκε στην Ιαπωνία το 2015. Το Chainer ήταν μία από τις πρώτες βιβλιοθήκες νευρωνικών δικτύων που προσέφερε μια πρόθυμη προσέγγιση στη διαφοροποίηση, αντί να ορίσει στατικά γραφήματα, επιτρέποντας μεγαλύτερη ευελιξία στον τρόπο δημιουργίας, εκπαίδευσης και

λειτουργίας των δικτύων. Ο συνδυασμός της κληρονομιάς από το Torch μαζί με τις ιδέες του Chainer έχει κάνει το PyTorch δημοφιλή τα τελευταία δύο χρόνια<sup>1</sup>.

Η βιβλιοθήκη διαθέτει επίσης ενότητες που βοηθούν στον χειρισμό κειμένου, εικόνων και ήχου (torchtext, torchvision και torchaudio), μαζί με ενσωματωμένες παραλλαγές δημοφιλών αρχιτεκτονικών όπως το ResNet (με βάρη που μπορείτε να κατεβάσετε για να παρέχετε βοήθεια με τεχνικές). Εκτός από το Facebook, το PyTorch γνώρισε γρήγορα αποδοχή από τη βιομηχανία, με εταιρείες όπως το Twitter, το Salesforce, η Uber και η NVIDIA να το χρησιμοποιούν με διάφορους τρόπους για τη βαθιά εκμάθησή τους [67]. Τα χαρακτηριστικά για το PyTorch ταξινομούνται ανά κατάσταση έκδοσης ως εξής:

**Stable:** Αυτά τα χαρακτηριστικά θα διατηρηθούν μακροπρόθεσμα και γενικά δεν θα πρέπει να υπάρχουν σημαντικοί περιορισμοί απόδοσης ή κενά στην τεκμηρίωση. Αναμένεται επίσης να διατηρηθεί η συμβατότητα προς τα πίσω.

**Beta:** Αυτά τα χαρακτηριστικά επισημαίνονται ως Beta επειδή το API μπορεί να αλλάξει με βάση τα σχόλια των χρηστών, είτε επειδή η απόδοση πρέπει να βελτιωθεί ή επειδή η κάλυψη σε όλους τους χειριστές δεν έχει ολοκληρωθεί ακόμη. Για τα χαρακτηριστικά της έκδοσης Beta, θεωρείται δεδομένο ότι είναι τουλάχιστον λειτουργική η ταξινόμηση Stable.

**Prototype:** Αυτά τα χαρακτηριστικά δεν είναι συνήθως διαθέσιμα ως μέρος δυαδικών διανομών (binary distributions) όπως το PyPI ή η Conda, εκτός από μερικές περιπτώσεις που μπορούν να ενεργοποιηθούν παράμετροι, μέσω run-time flags και βρίσκονται σε πρώιμο στάδιο για ανατροφοδότηση και δοκιμή [68].

### 5.3 ΥΛΟΠΟΙΗΣΗ

Το εκπαιδευμένο πλέον μοντέλο, έχοντας κάνει χρήση των παραπάνω εργαλείων, ενσωματώνεται στο JetSon Nano όπου και γίνεται ο έλεγχος του νευρωνικού δικτύου με το σετ ελέγχου. Τα δεδομένα που χρησιμοποιήθηκαν πάρθηκαν από το Kaggle, τα οποία αποτελούνται από 853 αρχεία με τις κλάσεις ‘πρόσωπο με μάσκα’, ‘πρόσωπο με λανθασμένη τοποθέτηση μάσκας’ και ‘πρόσωπο χωρίς μάσκα’ [69]. Αναλυτικά παρουσιάζεται παρακάτω η διαδικασία που ακολουθήθηκε.

---

<sup>1</sup> Να σημειωθεί ότι το PyTorch δανείστηκε ιδέες από το Chainer κι όχι ουσιαστικό κώδικα.



### 5.3.1 Προετοιμασία Δεδομένων

Σε πρώτο στάδιο έγινε η προεπεξεργασία των δεδομένων στο Jupyter. Παρακάτω παρατίθεται σε μορφή φωτογραφιών ο κώδικας σε σειρά βημάτων προκειμένου να παρθούν τα δεδομένα, να επεξεργαστούν απαραίτητα και τέλος να φορτωθούν μέσω συμπιεσμένου αρχείου στο Google Drive για να τα κατέβουν στο Google Colab, να γίνει αποσυμπίεση και να εκπαιδευτεί το δίκτυο από το YOLOv5.

```
In [1]: from opendatasets import download
from tqdm import tqdm
from os import listdir, makedirs
from os.path import join, exists
from xml.etree.ElementTree import parse
from sklearn.model_selection import train_test_split
from shutil import copy, make_archive, move
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
```

**Εικόνα 6. Βιβλιοθήκες για επεξεργασία δεδομένων**

```
In [2]: download('https://www.kaggle.com/andrewmvd/face-mask-detection')
0%|          | 0.00/398M [00:00<?, ?B/s]
Downloading face-mask-detection.zip to .\face-mask-detection
100%|██████████| 398M/398M [00:39<00:00, 10.6MB/s]
```

**Εικόνα 7. Κατέβασμα Δεδομένων**

Αφού κατέβουν τα δεδομένα, έπειτα δημιουργούνται δύο λίστες με τις τοποθεσίες των φακέλων (directories) για τις ανίστοιχες εικόνες και ετικέτες τους, για το που βρίσκονται και που θα αποθηκευτούν στην συνέχεια.

```
In [3]: input_directory = {}
input_directory['images'] = 'face-mask-detection\images'
input_directory['annotations'] = 'face-mask-detection\annotations'

output_directory = {}
output_directory['images'] = 'yolov5\face-mask-detection\images'
output_directory['labels'] = 'yolov5\face-mask-detection\labels'
```

**Εικόνα 8. Δημιουργία λιστών τοποθεσιών (directories)**

```
In [4]: input_images = []
for image in tqdm(listdir(input_directory['images'])):
    if image[-4:] == '.png':
        input_images.append(join(input_directory['images'], image))
input_images.sort()
100%|██████████| 853/853 [00:00<00:00, 214325.84it/s]
```

**Εικόνα 9. Φόρτωση εικόνων σε λίστα**





```
In [13]: if not exists(join(join(output_directory['labels'], 'val'))):
         makedirs(join(join(output_directory['labels'], 'val')))
         for label in tqdm(validation_labels):
             label_text = []
             for subclass in label['data']:
                 label_text.append("{} {:.3f} {:.3f} {:.3f} {:.3f}".format(
                     subclass['class_id'],
                     subclass['x'],
                     subclass['y'],
                     subclass['width'],
                     subclass['height']
                 ))
             print('\n'.join(label_text), file=open(join(join(output_directory['labels'], 'val'), label['name']), "w"))

100%|████████████████████████████████████████████████████████████████████████████████| 86/86 [00:00<00:00, 532.07it/s]
```

Εικόνα 18. Δημιουργία φακέλου για την επικύρωση και δημιουργία των αντίστοιχων ετικετών

```
In [14]: text_buffer = []
         text_buffer.append('path: ../face-mask-detection')
         text_buffer.append('train: images/train')
         text_buffer.append('val: images/val')
         text_buffer.append('nc: 3')
         text_buffer.append('names: ["with_mask", "mask_worn_incorrect", "without_mask"]')
         print('\n'.join(text_buffer), file=open(join('yolov5', 'face-mask-detection.yaml'), 'w'))
```

Εικόνα 19. Δημιουργία αρχείου YAML για ρύθμιση των δεδομένων του YOLOv5

```
In [15]: make_archive('face-mask-detection', 'zip', 'yolov5')
         move('face-mask-detection.zip', 'yolov5')

Out[15]: 'yolov5\\face-mask-detection.zip'
```

Εικόνα 20. Συμπύεση δεδομένων και μετακίνηση αρχείου στον φάκελο YOLOv5

```
In [16]: gauth = GoogleAuth()
         drive = GoogleDrive(gauth)
         gfile = drive.CreateFile({'parents': [{'id': '1z7fjpl_VIU9nOZIcsSDWNqM1-RGji2w'}]})
         gfile.SetContentFile('yolov5\\face-mask-detection.zip')
         gfile['title'] = 'face-mask-detection.zip'
         gfile.Upload()

Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?client_id=717996484891-fb2jcuv9au9vfbkpluha3suuijilu3n.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&access_type=offline&response_type=code

Authentication successful.
```

Εικόνα 21. Σύνδεση και ανέβασμα συμπιεσμένου αρχείου δεδομένων στο Google Drive

### 5.3.2 Εκπαίδευση Μοντέλου

Ο παρακάτω κώδικας έλαβε χώρα και έτρεξε στο Google Colab.

```
✓ [1] from google.colab import drive
      import torch
```

Εικόνα 22. Βιβλιοθήκες για εκπαίδευση μοντέλου

```
✓ [2] drive.mount('/content/drive')
21s
Mounted at /content/drive
```

**Εικόνα 23. Σύνδεση του Drive στο Google Colab**

```
✓ [3] !python -m pip install wandb
11s
!git clone https://github.com/ultralytics/yolov5
!python -m pip install -r yolov5/requirements.txt

Collecting wandb
  Downloading wandb-0.12.4-py2.py3-none-any.whl (1.7 MB)
    | 1.7 MB 9.2 MB/s
Collecting shortuuid>=0.5.0
  Downloading shortuuid-1.0.1-py3-none-any.whl (7.5 kB)
Collecting sentry-sdk>=1.0.0
  Downloading sentry_sdk-1.4.3-py2.py3-none-any.whl (139 kB)
    | 139 kB 86.9 MB/s
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from wandb) (3.13)
Requirement already satisfied: six>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (1.15.0)
Requirement already satisfied: requests<3,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (2.23.0)
Collecting pathtools
  Downloading pathtools-0.1.2.tar.gz (11 kB)
Requirement already satisfied: Click!=8.0.0,>=7.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (7.1.2)
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.7/dist-packages (from wandb) (2.8.2)
Collecting yaspin>=1.0.0
  Downloading yaspin-2.1.0-py3-none-any.whl (18 kB)
Requirement already satisfied: psutil>=5.0.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (5.4.8)
Collecting subprocess32>=3.5.3
  Downloading subprocess32-3.5.4.tar.gz (97 kB)
    | 97 kB 7.9 MB/s
Collecting configparser>=3.8.1
  Downloading configparser-5.0.2-py3-none-any.whl (19 kB)
Collecting GitPython>=1.0.0
  Downloading GitPython-3.1.24-py3-none-any.whl (180 kB)
    | 180 kB 72.0 MB/s
Collecting docker-pycreds>=0.4.0
  Downloading docker_pycreds-0.4.0-py2.py3-none-any.whl (9.0 kB)
Requirement already satisfied: promise<3,>=2.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (2.3)
Requirement already satisfied: protobuf>=3.12.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (3.17.3)
Collecting gitdb<5,>=4.0.1
  Downloading gitdb-4.0.7-py3-none-any.whl (63 kB)
    | 63 kB 2.0 MB/s
```

**Εικόνα 24. Κατέβασμα του YOLOv5 και εγκατάσταση απαιτούμενων βιβλιοθηκών Python**

```
✓ [4] print(f"Torch {torch.__version__} ({torch.cuda.get_device_properties(0).name})")
0s
Torch 1.9.0+cu111 (Tesla P100-PCIE-16GB)
```

**Εικόνα 25. Έλεγχος έκδοσης PyTorch και συνδεδεμένης κάρτας γραφικών**

```
✓ [5] !unzip -q /content/drive/MyDrive/Thesis/face-mask-detection.zip
2s
```

**Εικόνα 26. Αποσυμπίεση των δεδομένων**

```
✓ [6] %cd yolov5
0s
/content/yolov5
```

**Εικόνα 27. Αλλαγή τοποθεσίας εκτέλεσης εντολών στο Jupyter**

```
[7] !python train.py \
--img 640 \
--batch 80 \
--epochs 100 \
--data ../face-mask-detection.yaml \
--weights yolov5s.pt \
--project '/content/drive/MyDrive/Thesis' \
--name 'face-mask-detection_(100-epochs)' \
--cache

Starting training for 100 epochs...

Epoch 0/99   gpu_mem 15.2G   box 0.1176   obj 0.05559   cls 0.03749   labels 416   img_size 640: 100% 10/10 [00:18<00:00, 1.81s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:01<00:00, 1.6
              all 86 408 0.000382 0.000931 7.47e-05 1.47e-05

Epoch 1/99   gpu_mem 15.2G   box 0.1089   obj 0.061   cls 0.03164   labels 365   img_size 640: 100% 10/10 [00:14<00:00, 1.46s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:01<00:00, 1.2
              all 86 408 0.000983 0.051 0.000565 0.000139

Epoch 2/99   gpu_mem 15.2G   box 0.1011   obj 0.06814   cls 0.02659   labels 424   img_size 640: 100% 10/10 [00:14<00:00, 1.44s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:01<00:00, 1.3
              all 86 408 0.012 0.0289 0.0035 0.000698

Epoch 3/99   gpu_mem 15.2G   box 0.09238   obj 0.06758   cls 0.02414   labels 338   img_size 640: 100% 10/10 [00:14<00:00, 1.44s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:01<00:00, 1.6
              all 86 408 0.718 0.0596 0.0248 0.00507

Epoch 4/99   gpu_mem 15.2G   box 0.08329   obj 0.06551   cls 0.02195   labels 388   img_size 640: 100% 10/10 [00:14<00:00, 1.44s/it]
              Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:01<00:00, 1.8
              all 86 408 0.765 0.105 0.0836 0.0186
```

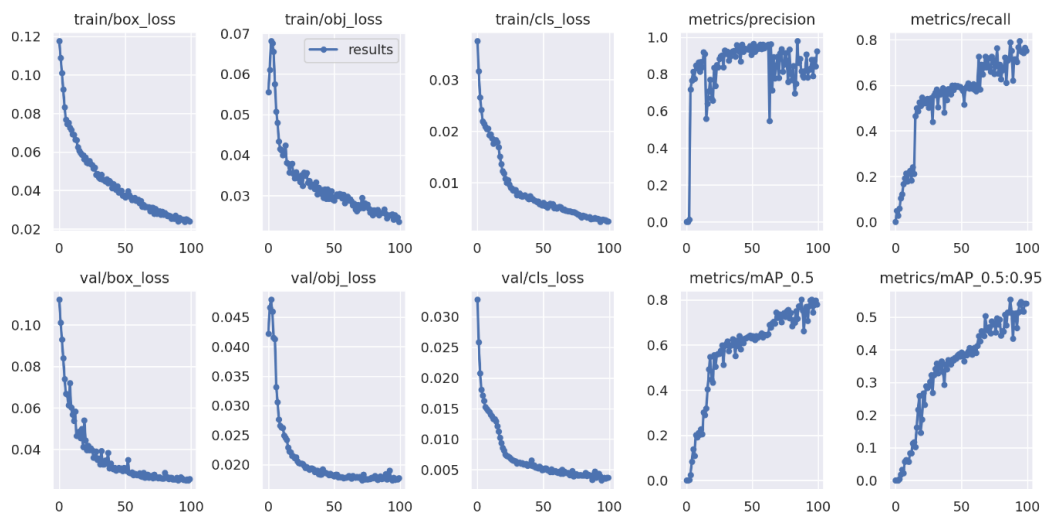
**Εικόνα 28. Εκπαίδευση μοντέλου για 100 εποχές, μέγεθος εικόνας 640x640 και παρτίδα 80 εικόνων και δείγμα των 6 πρώτων εποχών**

```
[7] Model Summary: 224 layers, 7059304 parameters, 0 gradients, 16.3 GFLOPs

Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 1/1 [00:02<00:00, 2.7
all 86 408 0.827 0.789 0.801 0.553
with_mask 86 358 0.932 0.958 0.973 0.669
mask_weared_incorrect 86 9 0.714 0.556 0.563 0.437
without_mask 86 41 0.834 0.854 0.867 0.553

wandb: Waiting for W&B process to finish, PID 399... (success).
wandb:
wandb: Run history:
wandb: metrics/mAP_0.5
wandb: metrics/mAP_0.5:0.95
wandb: metrics/precision
wandb: metrics/recall
wandb: train/box_loss
wandb: train/cls_loss
wandb: train/obj_loss
wandb: val/box_loss
wandb: val/cls_loss
wandb: val/obj_loss
wandb: x/lr0
wandb: x/lr1
wandb: x/lr2
wandb:
wandb: Run summary:
wandb: metrics/mAP_0.5 0.77924
wandb: metrics/mAP_0.5:0.95 0.5413
wandb: metrics/precision 0.92455
wandb: metrics/recall 0.75135
wandb: train/box_loss 0.02397
wandb: train/cls_loss 0.00257
wandb: train/obj_loss 0.02352
wandb: val/box_loss 0.02573
wandb: val/cls_loss 0.00372
wandb: val/obj_loss 0.0177
wandb: x/lr0 0.002
wandb: x/lr1 0.002
wandb: x/lr2 0.0021
```

**Εικόνα 29. Τέλος εκπαίδευσης**



Εικόνα 30. Αποτελέσματα εκπαίδευσης σε μορφή γραφήματος



Εικόνα 31. Δείγμα εικόνων επικύρωσης με τις αντίστοιχες ετικέτες τους



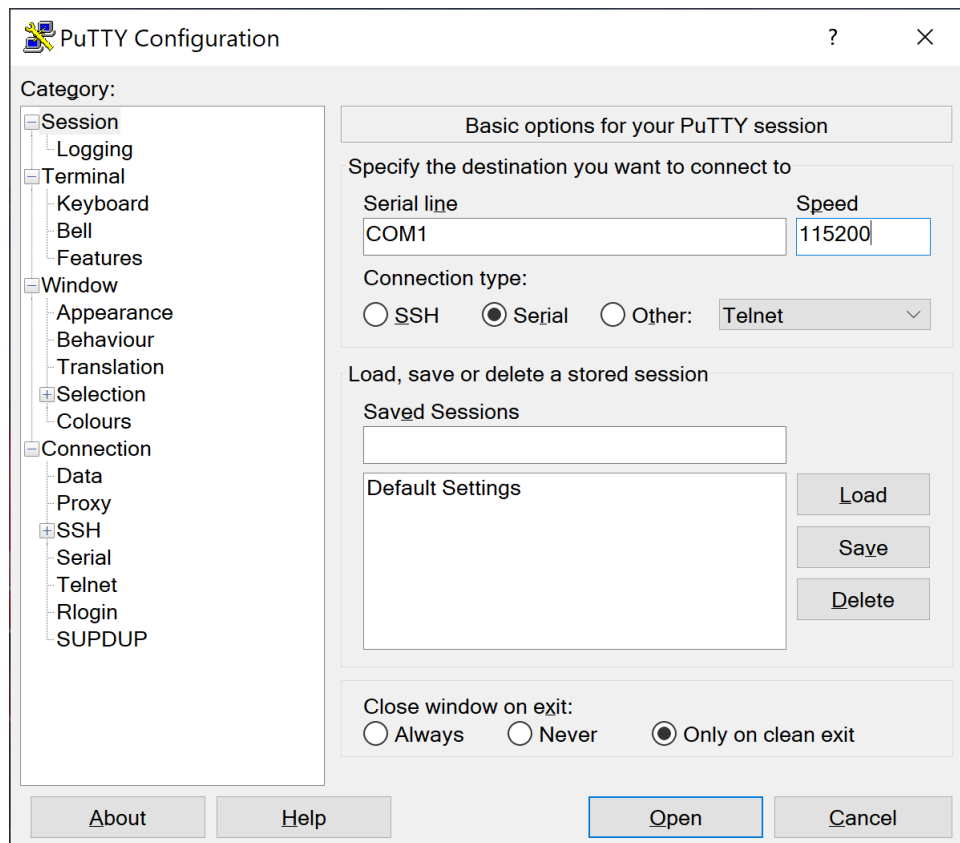


Εικόνα 32. Δείγμα εικόνων επικύρωσης με αναγνωρισμένες από το μοντέλο ετικέτες

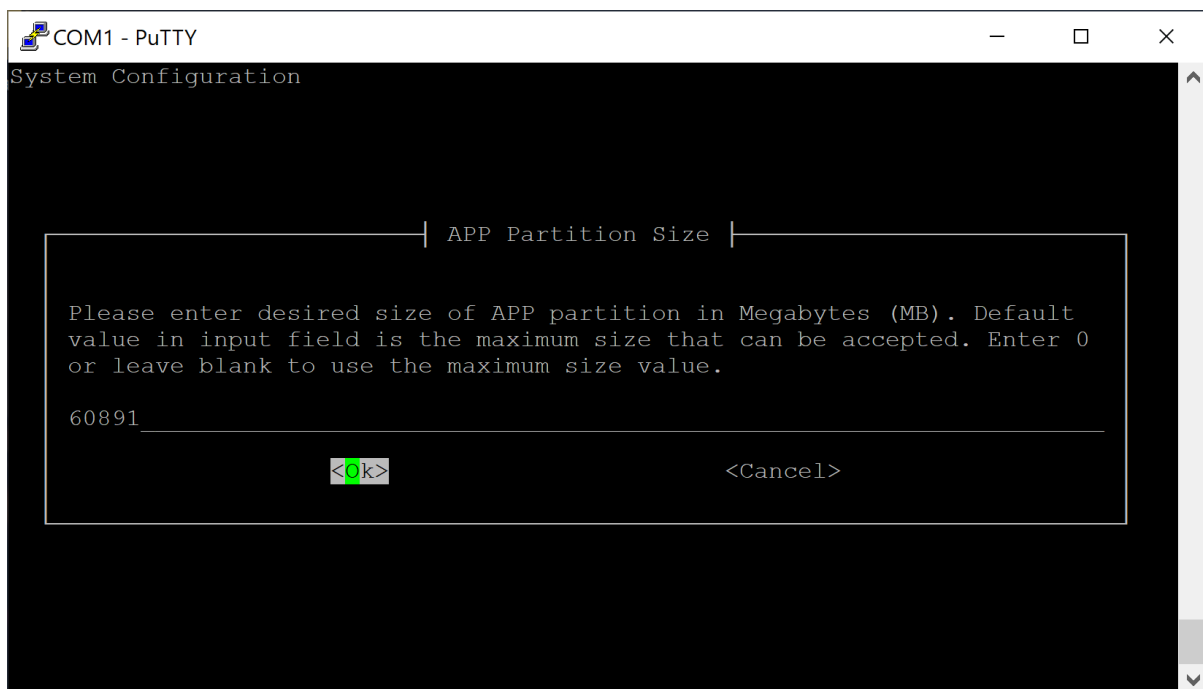
### 5.3.3 Αρχικοποίηση JetSon Nano και Εκτέλεση Μοντέλου

Για την αρχικοποίηση και λειτουργία του JetSon Nano χρειάστηκαν 3 εφαρμογές. Αρχικά κατέβηκε το ‘SD Memory Card Formatter’, το οποίο χρησιμοποιείται για να διαμορφωθεί η κάρτα μνήμης. Σε επόμενο βήμα κατέβηκε το image με το λειτουργικό. Έπειτα κατέβηκε το ‘BalenaEtcher’, το οποίο χρειάζεται για να μεταφερθεί το image με το λειτουργικό σύστημα στην κάρτα μνήμης. Τέλος, κατέβηκε η 3<sup>η</sup> εφαρμογή το ‘PuTTY’, το οποίο είναι απαραίτητο για να γίνει η σύνδεση αρχικά με σειριακή και έπειτα απομακρυσμένα με πρωτόκολλο SSH. Παρακάτω φαίνεται το configuration και τα βήματα που έγιναν για την αρχικοποίηση.

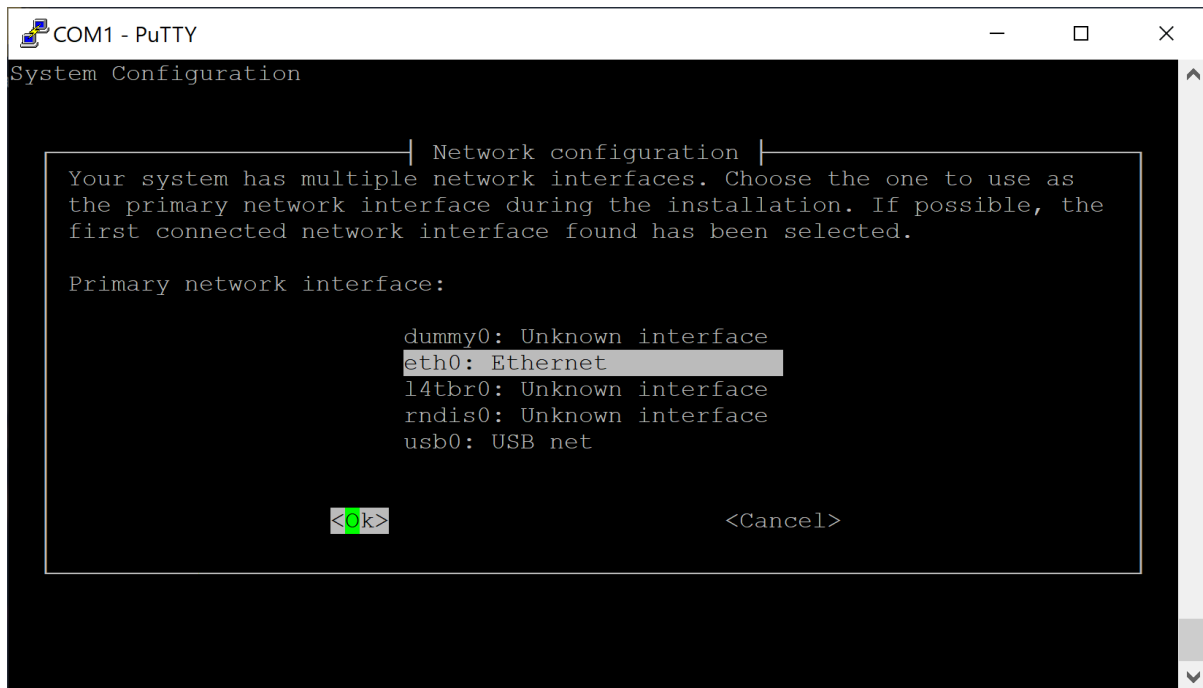




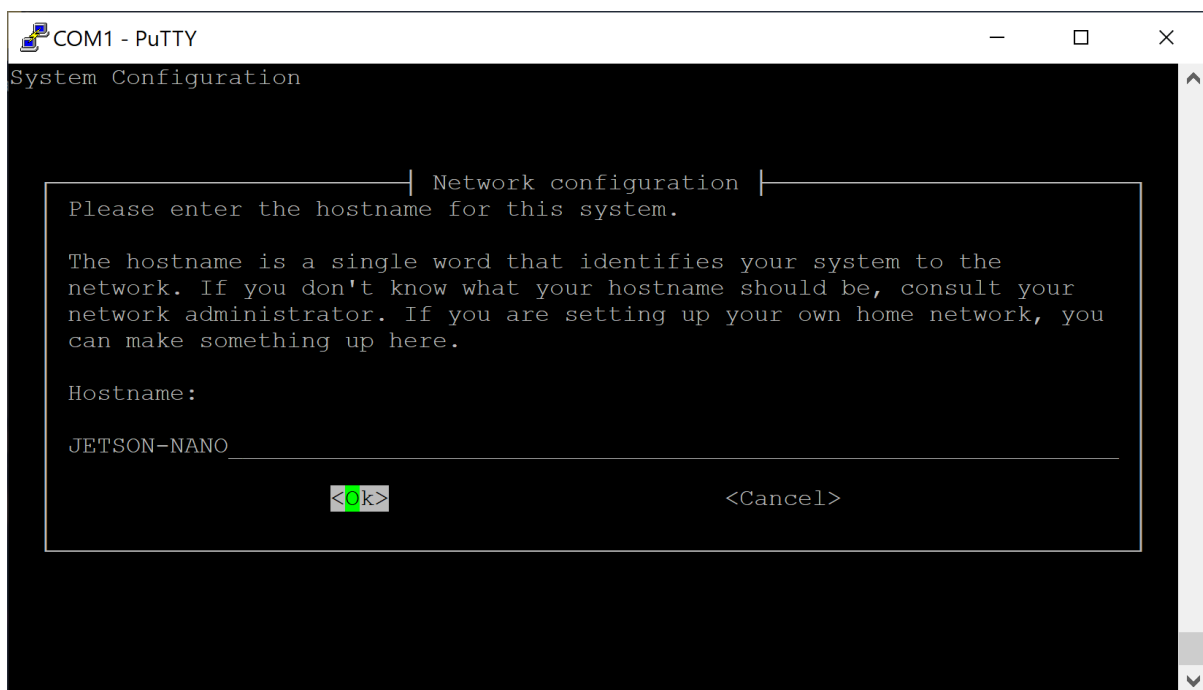
Εικόνα 33. Σύνδεση με σειριακή στο JetSon Nano



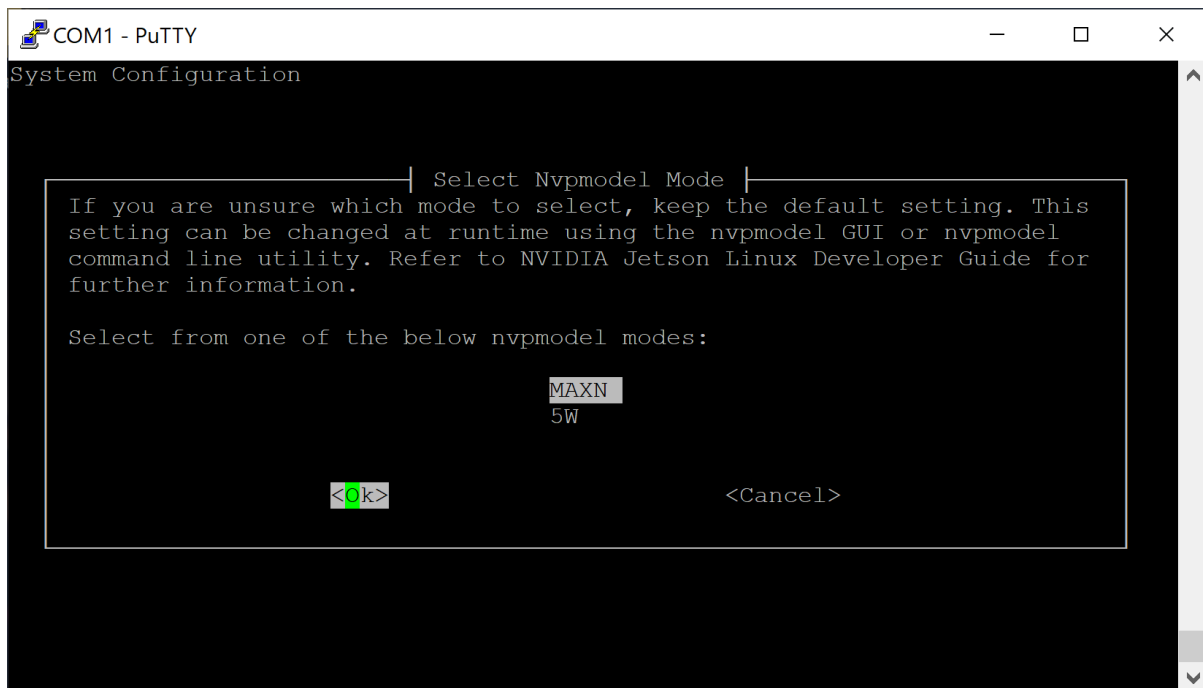
Εικόνα 34. Ρύθμιση μεγέθους αποθηκευτικού χώρου



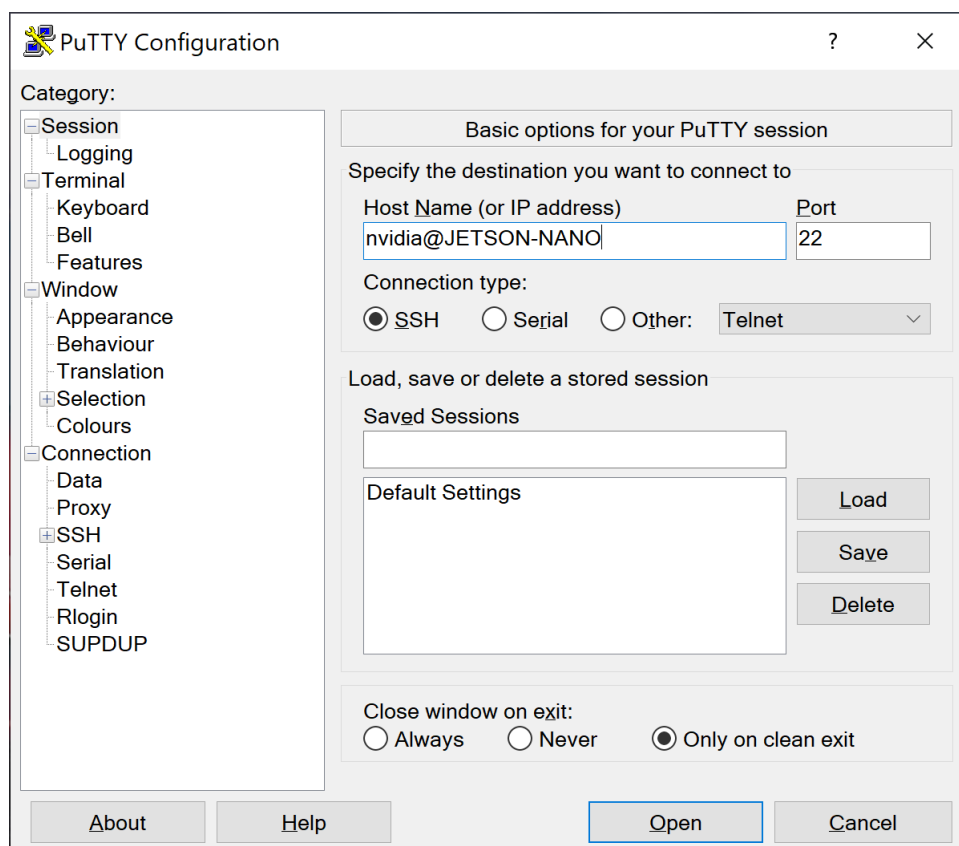
Εικόνα 35. Ρύθμιση τρόπου σύνδεσης με το διαδίκτυο



Εικόνα 36. Ρύθμιση ονόματος στο δίκτυο



Εικόνα 37. Ρύθμιση ισχύος επεξεργαστή και κάρτας γραφικών



Εικόνα 38. Απομακρυσμένη σύνδεση μέσω πρωτοκόλλου SSH

```
nvidia@JETSON-NANO: ~  
Using username "nvidia".  
nvidia@JETSON-NANO's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.9.253-tegra aarch64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
9 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
nvidia@JETSON-NANO:~$ sudo apt update && sudo apt upgrade
```

Εικόνα 39. Έλεγχος ενημερώσεων και ενημέρωση προγραμμάτων λογισμικού

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ sudo apt install python-pip python3-pip libopenmpi-dev lib  
openblas-base libpng-dev libjpeg-dev libavutil-dev libswresample-dev libswscale  
dev libavcodec-dev libavformat-dev
```

Εικόνα 40. Εγκατάσταση προαπαιτούμενων προγραμμάτων λογισμικού

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ python -m pip install --upgrade pip && python3 -m pip inst  
all pip
```

Εικόνα 41. Αναβάθμιση προγράμματος διαχείρισης βιβλιοθηκών Python

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ python3 -m pip install --upgrade setuptools wheel cython n  
umpy
```

Εικόνα 42. Εγκατάσταση προαπαιτούμενων βιβλιοθηκών Python

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ python3 -m pip install opencv-python
```

Εικόνα 43. Εγκατάσταση βιβλιοθήκης Python OpenCV

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ wget https://nvidia.box.com/shared/static/h1z9sw4bblybi0rm  
3tu8qjdj8hs051jbm.whl -O torch-1.9.0-cp36-cp36m-linux_aarch64.whl
```

Εικόνα 44. Κατέβασμα βιβλιοθήκης PyTorch έκδοσης 1.9.0 από την NVIDIA

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ python3 -m pip install torch-1.9.0-cp36-cp36m-linux_aarch64.whl
```

**Εικόνα 45. Εγκατάσταση PyTorch**

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ git clone --branch v0.10.0 https://github.com/pytorch/vision torchvision
```

**Εικόνα 46. Κατέβασμα TorchVision έκδοσης 0.10.0 από το GitHub**

```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ cd torchvision
```

**Εικόνα 47. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού σε TorchVision**

```
nvidia@JETSON-NANO: ~/torchvision  
nvidia@JETSON-NANO:~/torchvision$ export BUILD_VERSION=0.10.0
```

**Εικόνα 48. Δημιουργία απαραίτητης μεταβλητής για το TorchVision**

```
nvidia@JETSON-NANO: ~/torchvision  
nvidia@JETSON-NANO:~/torchvision$ export OPENBLAS_CORETYPE=ARMV8 python
```

**Εικόνα 49. Δημιουργία απαραίτητης μεταβλητής για το NumPy**

```
nvidia@JETSON-NANO: ~/torchvision  
nvidia@JETSON-NANO:~/torchvision$ python3 setup.py install --user
```

**Εικόνα 50. Εγκατάσταση του TorchVision**

```
nvidia@JETSON-NANO: ~/torchvision  
nvidia@JETSON-NANO:~/torchvision$ cd ../
```

**Εικόνα 51. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού στον αρχικό**

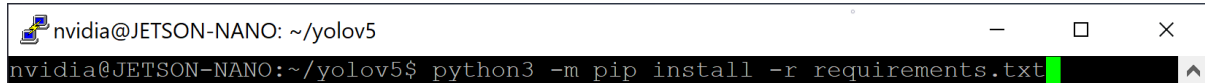
```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ git clone https://github.com/ultralytics/yolov5
```

**Εικόνα 52. Κατέβασμα του YOLOv5 από το GitHub**



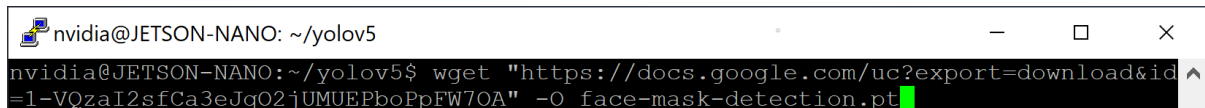
```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ cd yolov5
```

**Εικόνα 53. Αλλαγή τοποθεσίας εκτέλεσης εντολών τερματικού σε YOLOv5**




```
nvidia@JETSON-NANO: ~/yolov5  
nvidia@JETSON-NANO:~/yolov5$ python3 -m pip install -r requirements.txt
```

**Εικόνα 54. Εγκατάσταση προαπαιτούμενων βιβλιοθηκών για το YOLOv5**



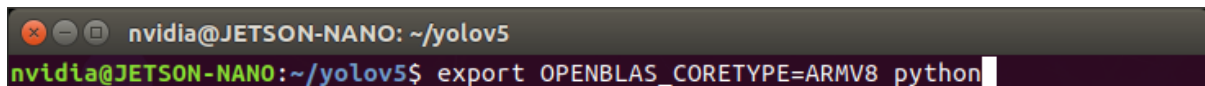
```
nvidia@JETSON-NANO: ~/yolov5  
nvidia@JETSON-NANO:~/yolov5$ wget "https://docs.google.com/uc?export=download&id=1-VQzaI2sfCa3eJqO2jUMUEPboPpFW7OA" -O face-mask-detection.pt
```

**Εικόνα 55. Κατέβασμα εκπαιδευμένου μοντέλου από το Google Drive**



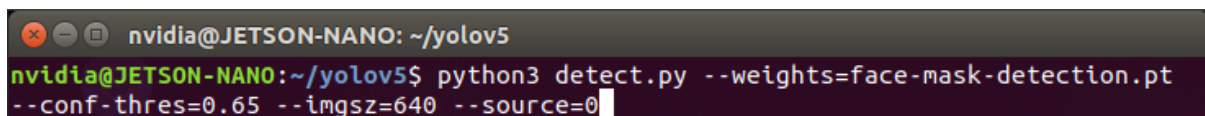
```
nvidia@JETSON-NANO: ~  
nvidia@JETSON-NANO:~$ cd yolov5
```

**Εικόνα 56. Αλλαγή τρέχουσας τοποθεσίας εκτέλεσης τερματικού σε 'yolov5'**



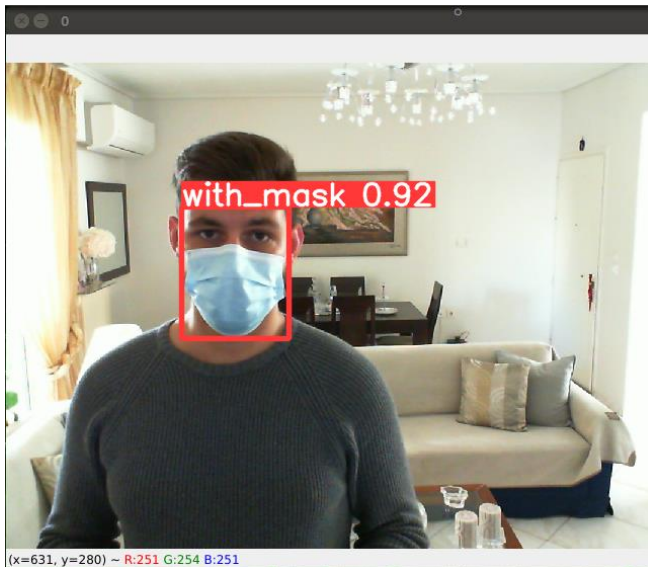
```
nvidia@JETSON-NANO: ~/yolov5  
nvidia@JETSON-NANO:~/yolov5$ export OPENBLAS_CORETYPE=ARMV8 python
```

**Εικόνα 57. Δημιουργία μεταβλητής απαραίτητη για το NumPy**

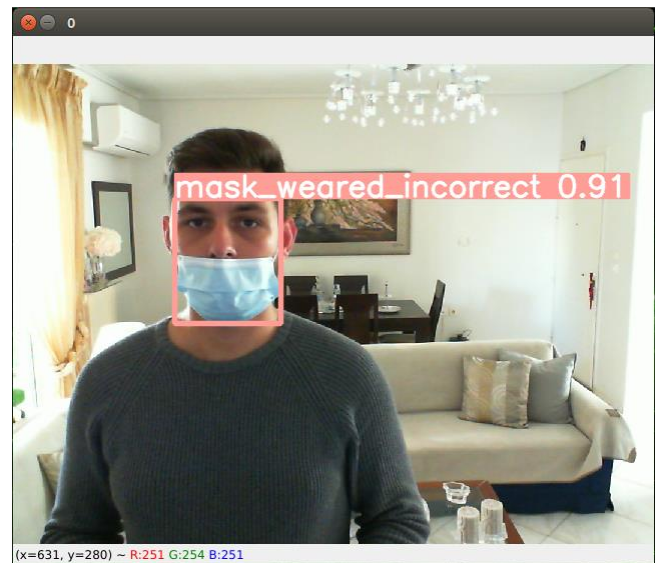


```
nvidia@JETSON-NANO: ~/yolov5  
nvidia@JETSON-NANO:~/yolov5$ python3 detect.py --weights=face-mask-detection.pt  
--conf-thres=0.65 --imgsz=640 --source=0
```

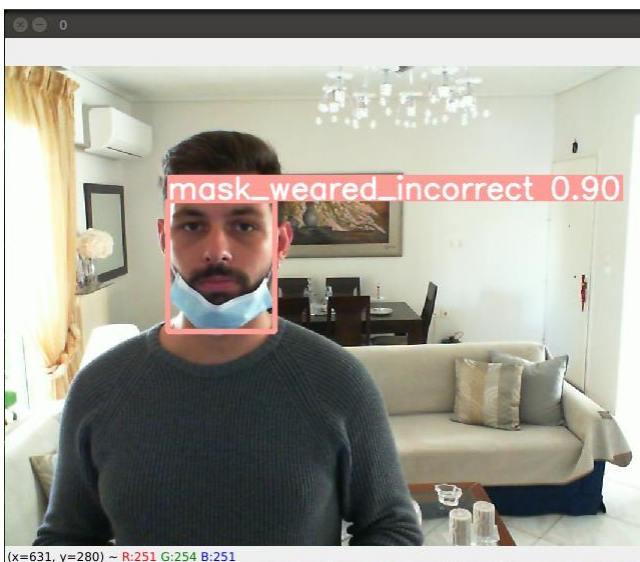
**Εικόνα 58. Εκτέλεση εκπαιδευμένου μοντέλου σε πραγματικό χρόνο με κατώφλι αυτοπεποίθησης 65% και μέγεθος εικόνας 640x640**



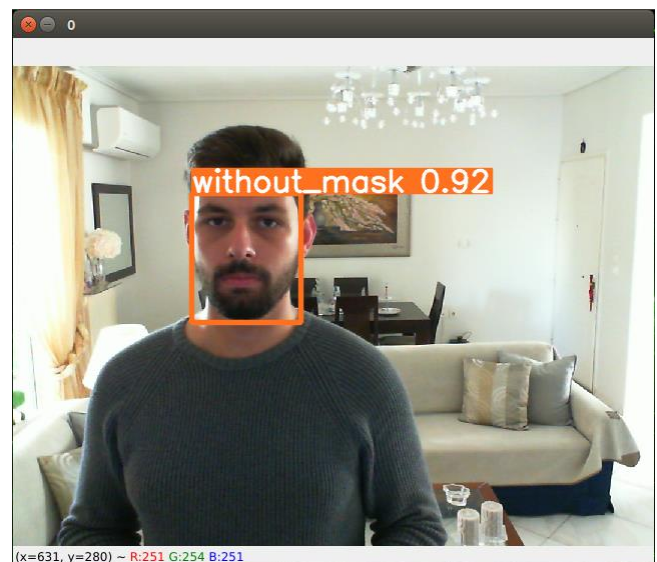
Εικόνα 59. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με μάσκα



Εικόνα 60. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με λανθασμένα τοποθετημένη μάσκα (κάτω από τη μύτη)



Εικόνα 61. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο με λανθασμένα τοποθετημένη μάσκα (στο πιγούνι)



Εικόνα 62. Στιγμιότυπο εκτέλεσης μοντέλου σε πραγματικό χρόνο χωρίς μάσκα





## 6.0 ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά την εκτέλεση της εφαρμογής και μέσω δοκιμών που γίνανε σε πραγματικό χρόνο με το εκπαιδευμένο μοντέλο, παρατηρήθηκε ότι μπορεί να αναγνωρίσει τις τρεις κλάσεις με σχετικά μεγάλη αυτοπεποίθηση (confidence), ανεξαρτήτως του αριθμού ατόμων που βρίσκονται σε ένα στιγμιότυπο εικόνας. Το JetSon Nano εκτελώντας το εκπαιδευμένο μοντέλο έχει τη δυνατότητα να κάνει 7 με 8 ανανεώσεις το δευτερόλεπτο (FPS – Frames Per Second).

Για να βελτιστοποιηθεί ακόμα περισσότερο θα μπορούσε να χρησιμοποιηθεί ένα μεγαλύτερο σετ δεδομένων, δηλαδή περισσότερες εικόνες για κάθε κλάση. Έτσι το μοντέλο που θα εκπαιδευτεί θα είχε πολύ μεγαλύτερη ακρίβεια, όπως και αντίστοιχα η εκπαίδευση να γίνει στο πιο σύνθετο μοντέλο του YOLOv5, το οποίο είναι το YOLOv5XL. Εκπαιδύοντας το μεγαλύτερο αυτό μοντέλο, για να μπορέσει να γίνει η αναγνώριση σε πραγματικό χρόνο με ανεκτές ανανεώσεις ανά δευτερόλεπτο (FPS) θα έπρεπε να χρησιμοποιηθεί ένας edge computer μεγαλύτερης ισχύος, όπως είναι το JetSon AGX Xavier της NVIDIA.

Τέλος, με αυτές τις αλλαγές, προτείνεται πως θα μπορούσε η εφαρμογή να χρησιμοποιηθεί σε εισόδους χώρων που απαιτείται η χρήση μάσκας και να γίνεται αναγνώριση και ενημέρωση με μεγάλη ακρίβεια ατόμων που είτε δεν φορούν μάσκα είτε την φορούν λανθασμένα, προκειμένου να παρθούν τα απαραίτητα μέτρα προστασίας.



**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] P. N. Stuart Russell, «24.5 Αναγνώριση Αντικειμένων,» σε *Τεχνητή Νοημοσύνη Μια Σύγχρονη Προσέγγιση Δεύτερη Αμερικάνικη Έκδοση*, Αθήνα, Κλειδάριθμος, 2005, pp. 982-991.
- [2] M. P. S. R. Ajeet Ram Pathaka, «Application of Deep Learning for Object Detection,» *Procedia Computer Science*, τόμ. 132, pp. 1707-1717, 2018.
- [3] S. H. F. E. a. S. S. R. Achanta, «Frequency-tuned salient region detection,» *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597-1604, 2009.
- [4] X. & Z. L. Hou, «Saliency Detection: A Spectral Residual Approach,» *IEEE Conference in Computer Vision and Pattern Recognition*, 2007.
- [5] K. C. P. P. Harel J, «Graph-based visual saliency,» *In: Advances in neural information processing systems*, τόμ. 19, p. 545–552, 2007.
- [6] E. F. W. P. S. S. Achanta R, «Salient region detection and segmentation,» *In: International conference on computer vision systems (ICVS'08). Lecture notes in computer science*, p. 66–75, 5008.
- [7] Y. Z. S. J. W. J. Z. N. T. X. S. H. Liu T, «Learning to detect a salient object,» *IEEE Trans Pattern Anal Mach Intell* 33, τόμ. 2, p. 353–367, 2011.
- [8] C. Z. F. H. F. D. Liang Z, «Salient object detection using content-sensitive hypergraph representation and partitioning,» *Pattern Recognit* 45, τόμ. 11, p. 3886–3901, 2012.
- [9] S. A. Avidan S, «Seam carving for content-aware image resizing,» *ACM Trans Graph* 26, 2007.
- [10] X. X. F. X. M. W. Z. H. Z. H. Chen LQ, «A visual attention model for adapting images on small displays,» *Multimed Syst*, τόμ. 4, p. 353–364, 2003.
- [11] I. L. Navalpakkam V, «An integrated model of top-down and bottom-up attention for optimizing detection speed,» *In: Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition, CVPR '06*, τόμ. 2, p. 2049–2056, 2006.

- [12] G. H. M. O. M. L. Borba GB, «An unsupervised method for clustering images based on their salient regions of interest,» *In: Proceedings of the 14th annual ACM international conference on multimedia*, p. 145–148, 2006.
- [13] W. D. K. C. P. P. Rutishauser U, «Is bottom-up attention useful for object recognition?,» *In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition*, τόμ. 2, p. 37–44, 2004.
- [14] . C. S. R. M. K. M. Dominik Maximilián Ramík, «A machine learning based intelligent vision system for autonomous object detection and recognition,» *Appl Intell*, τόμ. 40, p. 358–375, 2014.
- [15] G. & S. R. Hinton, «Reducing the dimensionality of data with neural networks,» *Science*, τόμ. 313, αρ. 5786, p. 504–507, 2006.
- [16] Y. B. L. B. Y. & H. P. LeCun, «Gradient based learning applied to document recognition,» *Proceedings of the IEEE*, τόμ. 86, αρ. 11, p. 2278–2324, 1998.
- [17] K. & L. B. Grauman, «Visual object recognition,» *Synthesis Lectures on Artificial Intelligence and Machine Learning*, τόμ. 5, αρ. 2, p. 1–181, 2011.
- [18] W. O. X. W. P. F. J. C. X. L. M. P. Li Liu, «Deep Learning for Generic Object Detection: A Survey,» *International Journal of Computer Vision (2020)*, τόμ. 128, p. 261–318, 2019.
- [19] A. S. I. H. G. Krizhevsky, «ImageNet classification with deep convolutional neural networks,» *In Proc. 25th. Int. Conf. Neural. Info. Process. Sys (NIPS 2012)*, τόμ. 1, p. 1097–1105, 2012.
- [20] K. Z. A. Simonyan, «Very Deep Convolutional Networks for Large-Scale Image Recognition,» *arXiv preprint arXiv*, pp. 1409-1556, 2014.
- [21] C. V. V. I. S. S. J. Szegedy, «Rethinking the inception architecture for computer vision,» *In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, p. 2818–2826, 2016.
- [22] K. Z. X. R. S. S. J. He, «Deep residual learning for image recognition,» *In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, p. 770–778, 2016 .

- [23] A. Z. M. C. B. K. D. W. W. W. T. A. M. A. H. Howard, «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,» *arXiv preprint arXiv*, pp. 1704-04861, 2017.
- [24] R. D. J. D. T. M. J. Girshick, «Rich feature hierarchies for accurate object detection and semantic segmentation,» *In: Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 580–587, 2014 .
- [25] S. D. R. G. A. F. Joseph Redmon, «You Only Look Once: Unified, Real-Time Object Detection,» *In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, p. 779–788, 2016.
- [26] W. A. D. E. D. S. C. R. S. F. C. B. A. Liu, «SSD: single Shot MultiBox detector,» *In: Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, τόμ. 1, p. 21–37, 2016.
- [27] S. I. S. S. H. Mori, «. A survey of diminished reality: techniques for visually concealing, eliminating, and seeing through real objects,» *IPSJ. Trans. Compute Vision. Appl.* 9, τόμ. 1, p. 1–17, 2017.
- [28] T. F. N. Y. Daiki Kido, «Diminished reality system with real-time object detection using deep learning for onsite landscape simulation during redevelopment,» *Environmental Modelling and Software*, τόμ. 131, pp. 1364-8152, 2020.
- [29] J. R. V. S. C. Z. M. K. A. F. A. F. I. W. Z. S. Y. G. S. M. K. Huang, «Speed/accuracy trade-offs for modern convolutional object detectors,» *In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, p. 7310–7319, 2016.
- [30] K. Z. X. R. S. a. S. J. He, «Spatial pyramid pooling in deep convolutional networks for visual recognition,» *IEEE Trans. Pattern Analysis and Machine Intelligence* , τόμ. 37, αρ. 9, p. 1904–1916, 2015.
- [31] J. L. Y. H. K. a. S. J. Dai, «R-FCN: Object detection via region-based fully convolutional networks,» *in Proc. Advances in Neural Information Processing Systems*, 2016.

- [32] T.-Y. G. P. G. R. H. K. a. D. P. Lin, «Focal loss for dense object detection,» in *Proc. Int. Conf. Comput. Vis*, 2017.
- [33] A. H. Y. P. E. G. X. F. Xiaoyue Jiang, «Deep Learning in Object Detection,» σε *Deep Learning in Object Detection and Recognition*, Singapore, Springer Nature Singapore Pte Ltd, 2019, pp. 19-53.
- [34] P. E. D. Z. X. M. M. F. R. a. L. Y. Sermanet, «Overfeat: Integrated recognition, localization and detection using convolutional networks,» In *ICLR*, 2014.
- [35] I. C. Education, «Deep Learning,» IBM , 1 May 2020. [Ηλεκτρονικό]. Available: <https://www.ibm.com/cloud/learn/deep-learning>.
- [36] T. M. Mitchel, «The Discipline of Machine Learning,» *CMU-ML-06-108*, pp. 1-7, 2006.
- [37] S. Haykin, «Neural Networks: A comprehensive foundation,» 1999.
- [38] R. Callan, «The Essence of Neural Networks,» *Prentice Hall*, 1999.
- [39] Α. Π. Λ. Ηλιάδης, «Τεχνητά Νευρωνικά Δίκτυα,» σε *Υπολογιστική Νοημοσύνη & Ευφρείς Πράκτορες*, Θεσσαλονίκη, Τζιόλα, 2017, pp. 141-206.
- [40] Α.-Ε. Μαρούντα, «Ταξινόμηση Εικόνων με Βαθιά Μάθηση,» Πάτρα, 2021.
- [41] Ε. Στούμπου, «Μεταφορά Γνώσης στη Βαθιά Μάθηση από Επιτροπές Νευρωνικών Δικτύων,» Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη, 2019.
- [42] S. Haykin, «Εισαγωγή,» σε *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, Αθήνα, Παπασωτηρίου, 2010, pp. 36-37.
- [43] S. O. Y.-W. T. Geoffrey E Hinton, «A fast learning algorithm for deep belief nets,» *Neural Comput*, τόμ. 18, αρ. 7, pp. 1527-54, 2006.
- [44] R. C. Lei Jimmy Ba, «Do Deep Nets Really Need to be Deep?,» *Draft for NIPS 2014*, pp. 1-10, 2014.
- [45] V. Kurkova, «Complexity estimates based on integral transforms induced by computational units,» *Neural Networks 33*, pp. 160-167, 2012.

- [46] K. Χρήστος, «Τεχνικές βαθιάς μηχανικής μάθησης και Συνελκτικά Νευρωνικά Δίκτυα για την ταξινόμηση υπερφασματικών δεδομένων,» ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ , Αθήνα, 2015.
- [47] Y. B. A. C. Ian Goodfellow, «CONVOLUTIONAL NETWORKS,» σε *Deep Learning*, MIT Press, 2016, pp. 330-372.
- [48] Y. B. G. H. Yann LeCun, «Deep learning,» *Nature*, τόμ. 521, p. 436–444, 27 May 2015.
- [49] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin και X. Yang, «A Survey on the Edge Computing for the Internet of Things,» *IEEE Access*, τόμ. 6, pp. 6900-6919, 2018.
- [50] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob και M. Imran, «The Role of Edge Computing in Internet of Things,» *IEEE Communications Magazine*, τόμ. 56, pp. 110-115, 2018.
- [51] R. Morabito, V. Cozzolino, A. Y. Ding, N. Bejar και J. Ott, «Consolidate IoT Edge Computing with Lightweight Virtualization,» *IEEE Network*, τόμ. 32, pp. 102-111, 2018.
- [52] I. Π. ΑΡΑΒΑΝΗΣ, «ΕΦΑΡΜΟΓΗ ΕΙΚΟΝΙΚΟΠΟΙΗΜΕΝΩΝ ΔΙΚΤΥΑΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΓΙΑ ΤΗΝ ΥΠΟΣΤΗΡΙΞΗ ΕΦΑΡΜΟΓΩΝ EDGE COMPUTING,» ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ, ΠΑΤΡΑ, 2020.
- [53] NVIDIA, «NVIDIA JETSON NANO BRINGING AI TO MILLIONS OF NEW DEVICES AT THE EDGE,» NVIDIA, 2019.
- [54] Google, «Google Colaboratory,» Google, [Ηλεκτρονικό]. Available: [https://colab.research.google.com/notebooks/intro.ipynb?utm\\_source=scs-index](https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index).
- [55] P. Jupyter, «Jupyter,» Project, 2021 September 2021. [Ηλεκτρονικό]. Available: <https://jupyter.org/about>.
- [56] A. Galea, «Introduction,» σε *APPLIED DATA SCIENCE WITH PYTHON AND JUPYTER*, Packt Publishing, 2018, pp. 20-21.
- [57] Anaconda, «CONDA,» Anaconda, 2017. [Ηλεκτρονικό]. Available: <https://docs.conda.io/en/latest/>.

- [58] A. Inc, «Data science technology for human sensemaking,» Anaconda Inc, 2021 . [Ηλεκτρονικό]. Available: <https://www.anaconda.com>.
- [59] S. P. Ravishankar Chityala, «Introduction to Python,» σε *Image Processing and Acquisition using Python, Second Edition*, CRC Press, pp. 3-6.
- [60] H. Z. Z. N. Fangbo Zhou, «Safety Helmet Detection Based on YOLOv5,» *IEEE International Conference on Power Electronics*, pp. 6-11, 22-24 January 2021.
- [61] Wanzhuandeepleaning, «Theory and Evolution of the YOLOv4 Model,» CSDN, 2020. [Ηλεκτρονικό]. Available: <https://blog.csdn.net/shajiyu1/article/details/105755280>.
- [62] A. F. Joseph Redmon, «YOLOv3: An Incremental Improvement,» 8 April 2018.
- [63] C.-Y. W. H.-Y. M. L. Alexey Bochkovskiy, «YOLOv4: Optimal Speed and Accuracy of Object Detection,» pp. 1-17, 2020.
- [64] B. L. J. P. Z. Z. Yifan Liu, «Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition,» *World Scientific Research Journal*, τόμ. 6, αρ. 11, pp. 276-284, 2020.
- [65] L. Joseph, «Introduction to OpenCV, OpenNI, and PCL,» σε *Learning Robotics Using Python*, BIRMINGHAM - MUMBAI, Packt Publishing, 2018, p. 173.
- [66] J. Cicolani, «Computer Vision,» σε *Beginning Robotics with Raspberry Pi and Arduino Using Python and OpenCV*, Pflugerville, Texas, USA, Apress, 2018, pp. 297-299.
- [67] I. Pointer, «PyTorch,» σε *Programming PyTorch for Deep Learning - Creating and Deploying Deep Learning Applications 2019*, Gravenstein Highway North, Sebastopol, CA 95472., O'Reilly Media, Inc., 2019, p. 7.
- [68] T. Contributors., «PYTORCH DOCUMENTATION,» PyTorch, 2019. [Ηλεκτρονικό]. Available: <https://pytorch.org/docs/stable/index.html#>.
- [69] Kaggle, «Face Mask Detection,» Kaggle. [Ηλεκτρονικό].



- [70] Z. S. M. I. Y. G. a. J. Y. S. M. I. Zhengxia Zou, «Object Detection in 20 Years: A Survey,» pp. 1-39, 2019.
- [71] W. O. X. W. P. F. J. C. X. L. M. P. Li Liu, «Deep Learning for Generic Object Detection: A Survey,» *International Journal of Computer Vision*, τόμ. 128, p. 261–318, 2020.
- [72] T. F. N. Y. Daiki Kido, «Diminished reality system with real-time object detection using deep learning for onsite landscape simulation during redevelopment,» *Environmental Modelling and Software* , τόμ. 131, pp. 1-10, 2020.
- [73] C. P. A. T. POGGIO, «A Trainable System for Object Detection,» *International Journal of Computer Vision*, τόμ. 38, αρ. 1, pp. 15-33, 2000.
- [74] C. S. A. T. D. Erhan, «Deep Neural Networks for Object Detection,» *Google, Inc*, pp. 1-9.
- [75] B. S. K. M. K. V. V. G. Padmapriya, «Deep convolutional neural network for real time object detection using tensor flow,» *Materials Today: Proceedings*, pp. 1-4, 2021.
- [76] S. D. R. G. A. F. Joseph Redmon, «You Only Look Once: Unified, Real-Time Object Detection,» *Allen Institute for AI, Facebook AI Research*, pp. 1-10.
- [77] J. Brownlee, «A Gentle Introduction to Object Recognition With Deep Learning,» *Machine Learning Mastery*, 27 January 2021. [Ηλεκτρονικό]. Available: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>.
- [78] J. C. Redmon, «YOLO: Real-Time Object Detection,» [Ηλεκτρονικό]. Available: <https://pjreddie.com/darknet/yolo/>.