ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«Πληροφοριακά Συστήματα και Υπηρεσίες»
ΚΑΤΕΥΘΥΝΣΗ
«Μεγάλα Δεδομένα και Αναλυτική»

Αναγνώριση Χειρονομιών σε Βίντεο Πραγματικού Χρόνου με Αλγορίθμους Μηχανικής Μάθησης και Βαθιάς Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΙΔΑΚΗΣ ΛΕΑΝΔΡΟΣ

Επιβλέπων : Ηλίας Μαγκλογιάννης
Καθηγητής Πανεπιστημίου Πειραιώς

Πειραιάς, Ιούλιος 2021

Ευαγγελιδάκης Λέανδρος

Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς

Copyright © **Ευαγγελιδάκης Λέανδρος**, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Περίληψη

Η Υπολογιστική Όραση και η Μηχανική Μάθηση είναι δύο μεγάλοι κλάδοι της Τεχνητής Νοημοσύνης που συνδυαστικά έχουν δώσει στα υπολογιστικά συστήματα την ικανότητα της τεχνητής όρασης και της αντίληψης. Πλέον, είναι εφικτή η αλληλεπίδραση ανθρώπου-υπολογιστή μέσω καμερών καθώς και η οπτική ανάλυση της ανθρώπινης συμπεριφοράς. Το αντικείμενο της παρούσας εργασίας είναι η υλοποίηση ενός συστήματος για την αυτόματη αναγνώριση στατικών χειρονομιών σε ακολουθίες βίντεο πραγματικού χρόνου, μέσω καμερών, χρησιμοποιώντας σύγχρονες τεχνικές και αλγορίθμους της κλασσικής Μηχανικής Μάθησης και της Βαθιάς Μάθησης.

Αρχικά γίνεται μια εισαγωγή στην Υπολογιστική Όραση και στις εφαρμογές της και περιγράφουμε το πρόβλημα της αναγνώρισης χειρονομιών. Έπειτα αναφερόμαστε στο πεδίο της Μηχανικής Μάθησης και στα είδη των προβλημάτων που πραγματεύεται. Περιγράφουμε τον τρόπο εκπαίδευσης ενός μοντέλου και τρόπους αξιολόγησης της απόδοσής του εστιάζοντας στην επίλυση προβλημάτων Κατηγοριοποίησης. Στη συνέχεια, κάνουμε μια εισαγωγή στη Βαθιά Μάθηση και στα Τεχνητά Νευρωνικά Δίκτυα. Αναφέρουμε τις βασικές αρχιτεκτονικές δικτύων που χρησιμοποιούνται σήμερα και τη διαδικασία εκπαίδευσης ενός Νευρωνικού Δικτύου καθώς και τεχνικές βελτιστοποίησής του. Επίσης, γίνεται αναφορά στη χρήση Συνελικτικών Νευρωνικών Δικτύων για την επίλυση προβλημάτων Υπολογιστικής Όρασης και παρουσιάζουμε κάποιους γνωστούς αλγορίθμους.

Τέλος, παρουσιάζεται η υλοποίηση ενός συστήματος αναγνώρισης στατικών χειρονομιών, χρησιμοποιώντας τη γλώσσα προγραμματισμού Python. Εφαρμόζουμε καινοτόμους αλγορίθμους με συνελικτικά νευρωνικά δίκτυα για την αναγνώριση σημείων του σκελετού του χεριού και στη συνέχεια εκπαιδεύουμε Μηχανές Διανυσμάτων Υποστήριξης και Νευρωνικά Δίκτυα Εμπρόσθιας Τροφοδότησης, για να κατηγοριοποιήσουμε τα σημεία αυτά ανάμεσα σε προκαθορισμένες στατικές χειρονομίες. Η υλοποίηση προορίζεται για εφαρμογές πραγματικού χρόνου, χρησιμοποιεί μόνο τον επεξεργαστή του υπολογιστή και λειτουργεί με απλές κάμερες RGB.

Λέξεις Κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Βαθιά Μάθηση, Υπολογιστική Όραση, Αναγνώριση Χειρονομιών, Κατηγοριοποίηση, Μηχανές Διανυσμάτων Υποστήριξης, Τεχνητά Νευρωνικά Δίκτυα

Abstract

Computer Vision and Machine Learning are two branches of Artificial Intelligence which in combination have granted computational systems the ability of artificial vision and perception. Nowadays, human-machine interaction and the analysis of human behavior can be achieved using cameras. The subject of the current thesis is the implementation of a system for automatic recognition of static hand gestures in real-time videos with cameras, using modern techniques and algorithms of Machine Learning and Deep Learning.

To begin with, we make an introduction into Computer Vision and its applications and we describe the problem of hand gesture recognition. Next, we refer to the field of Machine Learning and the main categories of the problems it includes. We describe the methodology of training a model and ways to measure its performance, focusing on Classification problems. Also, we introduce Deep Learning and Artificial Neural Networks. We present the most common network architectures used nowadays and the procedure of training a Neural Network and methods for optimization. Also, we mention the use of Convolutional Neural Networks for solving Computer Vision problems and we present some of the most known algorithms used for this task.

Finally, we present our implementation of the hand gesture recognition tool, using Python programming language. We apply innovative algorithms of Convolutional Neural Networks for the detection of several key points of the hand and we train a Support Vector Machines and a Feedforward Neural Network model for the classification of those points into several predetermined static gestures. The implementation is targeted for practical real-time applications, runs using the CPU of the computer and requires a common RGB camera.

Keywords

Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision, Hand Gesture Recognition, Classification, Support Vector Machines, Artificial Neural Networks

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ηλία Μαγκλογιάννη, πρόεδρο του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς, για την ευκαιρία που μου έδωσε να ασχοληθώ με αυτό το εξαιρετικά ενδιαφέρον αντικείμενο και για τη συνολική καθοδήγησή του. Οι διαλέξεις του κατά τη διάρκεια των μαθημάτων αποτέλεσαν βασικό κίνητρο για την ενασχόλησή μου με προβλήματα υπολογιστικής όρασης και μηχανικής μάθησης. Επίσης, θα ήθελα να ευχαριστήσω όλους του καθηγητές του μεταπτυχιακού προγράμματος σπουδών, από τους οποίους απέκτησα πολλές γνώσεις, οι οποίες θα αποτελέσουν σημαντικό εργαλείο για την επαγγελματική μου σταδιοδρομία. Τέλος, θέλω να ευχαριστήσω ολόψυχα την οικογένειά μου και τους φίλους για την υποστήριξή τους και την υπομονή τους και ιδιαίτερα τη μητέρα μου, η οποία είναι δίπλα μου όλα αυτά τα χρόνια.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη 3
Abstract 5
ΕΥΧΑΡΙΣΤΙΕΣ 7
ΚΕΦΑΛΑΙΟ 1 13
ΕΙΣΑΓΩΓΗ 13
1.1 Τεχνητή Νοημοσύνη 13
1.2 Υπολογιστική Όραση και Μηχανική Μάθηση 14
1.2.1 Εφαρμογές της Υπολογιστικής Όρασης 15
1.3 Το πρόβλημα της αναγνώρισης χειρονομιών 17
1.3 Σκοπός της διπλωματικής 19
1.4 Οργάνωση της διπλωματικής 20
1.5 Τεχνολογίες που χρησιμοποιήθηκαν 21
ΚΕΦΑΛΑΙΟ 2 23
ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ 23
2.1 Είδη Μάθησης 23
2.1.1 Επιτηρούμενη Μάθηση 23
2.1.2 Μη-Επιτηρούμενη Μάθηση 25
2.1.3 Ενισχυτική Μάθηση 26
2.2 Κατηγοριοποίηση Δεδομένων 27
2.3 Εκπαίδευση και αξιολόγηση ενός κατηγοριοποιητή 28
2.3.1 Σύνολα Εκπαίδευσης και Ελέγχου 28
2.3.2 Επιλογή Χαρακτηριστικών 29
2.3.3 Μετρικές αξιολόγησης 29
2.4 Μηχανές Διανυσμάτων Υποστήριξης 31
2.4.1 Γενική περιγραφή 31
2.4.2 Γραμμικά SVM 32
2.4.3 Μη-Γραμμικά SVM και το τέχνασμα του πυρήνα 33
ΚΕΦΑΛΑΙΟ 3 35
ΒΑΘΙΑ ΜΑΘΗΣΗ 35
3.1 Τεχνητά Νευρωνικά Δίκτυα 35
3.2 Αρχιτεκτονικές Νευρωνικών Δικτύων 38

3.2.1 Νευρωνικά Δίκτυα Εμπρόσθιας Τροφοδότησης 38
3.2.2 Συνελικτικά Νευρωνικά Δίκτυα 38
3.2.3 Αναδρομικά Νευρωνικά Δίκτυα 40
3.3 Εκπαίδευση ενός τεχνητού νευρωνικού δικτύου 40
3.3.1 Η συνάρτηση κόστους 41
3.3.2 Αλγόριθμοι Βελτιστοποίησης 42
3.3.3 Ο αλγόριθμος Οπισθοδιάδοσης 43
3.3.4 Το πρόβλημα της Υπερπροσαρμογής 44
3.3.5 Απόσυρση Νευρώνων (Dropout) 45
3.4 Συναρτήσεις Ενεργοποίησης 46
3.5 Εντοπισμός αντικειμένων σε εικόνες με Συνελικτικά Νευρωνικά Δίκτυα 48
3.5.1 Εισαγωγή 48
3.5.2 Region-Based Συνελικτικά Νευρωνικά Δίκτυα 49
3.5.3 Ο αλγόριθμος You Only Look Once (YOLO) 51
3.5.4 Single Shot Detector (SSD) 51
ΚΕΦΑΛΑΙΟ 4 53
ΥΠΟΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΗΣ ΑΝΑΓΝΩΡΙΣΗΣ ΣΤΑΤΙΚΩΝ ΧΕΙΡΟΝΟΜΙΩΝ 53
4.1 Γενική μεθοδολογία 53
4.2 Αναγνώριση χεριού μέσω του MediaPipe 54
4.3 Κατασκευή συνόλου δεδομένων 57
4.3.1 Περιγραφή 57
4.3.2 Εισαγωγή νέας χειρονομίας 58
4.4 Κατηγοριοποίηση χειρονομιών με Μηχανές Διανυσμάτων Υποστήριξης 58
4.4.1 Εκπαίδευση μοντέλου 59
4.4.2 Αξιολόγηση 60
4.4 Κατηγοριοποίηση χειρονομιών με Τεχνητά Νευρωνικά Δίκτυα 61
4.4.1 Επιλογή υπερπαραμέτρων 61
4.4.2 Εκπαίδευση μοντέλου 63
4.4.2 Αξιολόγηση 65
ΚΕΦΑΛΑΙΟ 5 66
ΣΥΜΠΕΡΑΣΜΑΤΑ 66

ΒΙΒΛΙΟΓΡΑΦΙΑ 68

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1-1: Η Υπολογιστική Όραση, η Μηχανική και η Βαθιά Μάθηση ως υποσύνολο της Τεχνητής Νοημοσύνης 14
Εικόνα 1-2: Το γάντι acceleGlove [1] 17
Εικόνα 1-3: Παράδειγμα προσέγγισης του σκελετού του χεριού [2] 18
Εικόνα 2-1: Επιτηρούμενη Μάθηση [3] 23
Εικόνα 2-2 Παράδειγμα Κατηγοριοποίησης (αριστερά) και Παλινδρόμησης (Δεξιά) [4] 24
Εικόνα 2-3: Μη-Επιτηρούμενη Μάθηση [5] 25
Εικόνα 2-4: Ενισχυτική Μάθηση [6] 26
Εικόνα 2-5: Spam Classification [7] 27
Εικόνα 2-6: Πίνακας Σύγχυσης [8] 29
Εικόνα 2-7: Ευθείες που διαχωρίζουν τα δεδομένα. Η ευθεία H_1 κάνει λάθος διαχωρισμό, η H_2 κάνει μεν σωστό διαχωρισμό αλλά με μικρό περιθώριο και η H_3 κάνει βέλτιστο διαχωρισμό των κλάσεων. [9] 31
Εικόνα 2-8: Εύρεση βέλτιστου υπερεπιπέδου μέσω των διανυσμάτων στήριξης [10] 32
Εικόνα 2-9: Αλλαγή πυρήνα σε ένα SVM για να μετατραπούν τα δεδομένα σε γραμμικώς διαχωρίσιμα [11] 33
Εικόνα 2-10: Κατηγοριοποίηση με SVM του συνόλου δεδομένων Iris χρησιμοποιώντας διαφορετικούς πυρήνες [25] 34
Εικόνα 3-1: Βιολογικό Νευρωνικό Δίκτυο [12] 36
Εικόνα 3-2: Η λειτουργία ενός τεχνητού νευρώνα [13] 37
Εικόνα 3-3: Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης (Feedforward) [14] 38
Εικόνα 3-4: Συνελικτικό Νευρωνικό Δίκτυο [15] 39
Εικόνα 3-5: Αναδρομικό Νευρωνικό Δίκτυο [16] 40
Εικόνα 3-6: Παραδείγματα Υποπροσαρμογής, Καλής Προσαρμογής και Υπερπροσαρμογής στα δεδομένα [17] 44
Εικόνα 3-7: Χωρισμός δεδομένων σε σύνολο εκπαίδευσης (training), ελέγχου (test) και επικύρωσης (validation) [18] 45
Εικόνα 3-8: Ένα Νευρωνικό Δίκτυο πριν και μετά την εφαρμογή Dropout [19] 45
Εικόνα 3-9: Η Λογιστική Συνάρτηση [21] 46
Εικόνα 3-10: Γράφημα υπερβολικής εφαπτομένης [22] 47
Εικόνα 3-11: Γραφικές απεικονίσεις των ReLU, Leaky ReLU και Randomized Leaky ReLU [23] 47
Εικόνα 3-12: Παράδειγμα λειτουργίας του αλγόριθμου R-CNN [27] 49
Εικόνα 3-13: Παράδειγμα λειτουργίας του αλγόριθμου Fast R-CNN [28] 50

Εικόνα 3-14: Δομή του αλγόριθμου Faster R-CNN [29] 50
Εικόνα 3-15: Ο αλγόριθμος Yolo [30] 51
Εικόνα 3-16: Η αρχιτεκτονική του δικτύου VGG-16 [37] 51
Εικόνα 3-17: Τα bounding boxes του αλγόριθμου SSD [31] 52
Εικόνα 3-18: Οι αρχιτεκτονικές των αλγορίθμων SSD και YOLO. [31] 52
Εικόνα 4-1: Γενική μεθοδολογία της υλοποίησης 54
Εικόνα 4-2 MediaPipe Hands – 21 Hand Landmarks [2] 55
Εικόνα 4-3: Παραδείγματα συνόλων εκπαίδευσης του αλγορίθμου του MediaPipe. Η επάνω σειρά αντιστοιχεί σε πραγματικές εικόνες και η κάτω σειρά σε συνθετικές. [2] .56
Εικόνα 4-4: Οι 11 χειρονομίες του συνόλου δεδομένων (G1 έως G11) [39] 57
Εικόνα 4-5: Χειρονομία thumbs-up (G12) 58
Εικόνα 4-6: Μοντέλο SVM – Πίνακες Σύγχυσης 59
Εικόνα 4-7: Μετρικές αξιολόγησης ανά χειρονομία, μοντέλο SVM 60
Εικόνα 4-8: Η γενική αρχιτεκτονική του Νευρωνικού Δικτύου 61
Εικόνα 4-9: Η ακρίβεια του μοντέλου για κάθε συνδυασμό υπερπαραμέτρων 62
Εικόνα 4-10: Το τελικό Νευρωνικό Δίκτυο 63
Εικόνα 4-11: Ακρίβεια του μοντέλου με Νευρωνικά Δίκτυα κατά τη διάρκεια εκπαίδευσης. Με σκούρο μπλε απεικονίζεται το σύνολο ελέγχου και με πορτοκαλί το σύνολο εκπαίδευσης. 63
Εικόνα 4-12: Σφάλμα του μοντέλου με Νευρωνικά Δίκτυα κατά τη διάρκεια εκπαίδευσης. Με σκούρο μπλε απεικονίζεται το σύνολο ελέγχου και με πορτοκαλί το σύνολο εκπαίδευσης. 64
Εικόνα 4-13: Πίνακας Σύγχυσης, Νευρωνικό Δίκτυο 64
Εικόνα 4-14: Κανονικοποιημένος Πίνακας Σύγχυσης, Νευρωνικό Δίκτυο 65

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η χρήση μεθόδων της Τεχνητής Νοημοσύνης όπως η Υπολογιστική Όραση, σε συνδυασμό με αλγορίθμους της Μηχανικής Μάθησης και της Βαθιάς Μάθησης, έχει καταφέρει να δώσει τη δυνατότητα σε υπολογιστικά συστήματα να αποκτήσουν την τεχνητή αίσθηση της όρασης. Δηλαδή προχωρημένες δυνατότητες αντίληψης και ανάλυσης των εικόνων που βλέπουν και του περιεχομένου που αναπαριστούν.

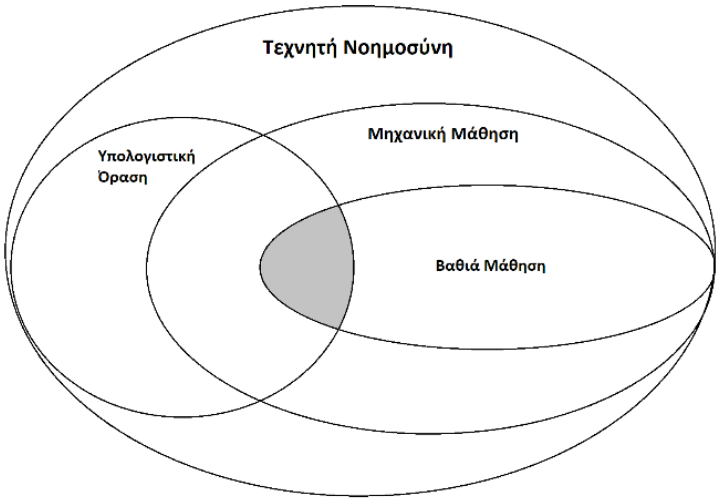
1.1 Τεχνητή Νοημοσύνη

Η *Τεχνητή Νοημοσύνη (Artificial Intelligence)* είναι ένας κλάδος της πληροφορικής που στοχεύει στην υλοποίηση υπολογιστικών συστημάτων τα οποία προσομοιώνουν λειτουργίες της ανθρώπινης νοημοσύνης όπως αυτή της μάθησης, της επίλυσης προβλημάτων, της προσαρμοστικότητας και της εξαγωγής συμπερασμάτων. Ακριβής ορισμός της Τεχνητής Νοημοσύνης δεν υπάρχει και γενικά θεωρείται ένα μεγάλο επιστημονικό πεδίο που αποτελείται από επιμέρους τομείς όπως τη Μηχανική Μάθηση (Machine Learning), τη Βαθιά Μάθηση (Deep Learning) και την Υπολογιστική Όραση (Computer Vision) με τους οποίους ασχολούμαστε και στην παρούσα εργασία.

Το 1950 ο Alan Turing διατύπωσε την ερώτηση «Can machines think?», δηλαδή αν οι υπολογιστές μπορούν να σκέφτονται. Από μια σκοπιά, η Τεχνητή Νοημοσύνη προσπαθεί να απαντήσει σε αυτό το ερώτημα καταφατικά, είναι δηλαδή η προσπάθεια απόδοσης της βιολογικής σκέψης, σε ένα υπολογιστικό μοντέλο. Στη σύγχρονη εποχή, με τη ραγδαία αύξηση της τεχνολογίας και της ολοένα και αυξανόμενης διαθέσιμης υπολογιστικής ισχύος, η τεχνητή νοημοσύνη αποτελεί βασικό εργαλείο ανάπτυξης και καινοτομίας σε όλους τους κλάδους της κοινωνίας. Σε πολλούς τομείς, έχει ήδη ξεπεράσει σε δυνατότητες την ανθρώπινη ευφυία και αναμφίβολα στο προσεχές μέλλον, πληθώρα ανθρωπίνων διεργασιών θα αντικατασταθεί με τέτοιου είδους ευφυή συστήματα.

Ενδεικτικά παραδείγματα της τεχνητής νοημοσύνης είναι τα αυτό-οδηγούμενα οχήματα (self-driving vehicles), όπως drones και αυτοκίνητα, οι αυτόματες ιατρικές διαγνώσεις, η αναγνώριση προτύπων σε εικόνες και βίντεο (pattern recognition), τα συστήματα συστάσεων (recommendation systems), οι ψηφιακοί προσωπικοί βοηθοί (digital personal assistants) κ.λπ.

1.2 Υπολογιστική Όραση και Μηχανική Μάθηση



Εικόνα 1-1: Η Υπολογιστική Όραση, η Μηχανική και η Βαθιά Μάθηση ως υποσύνολο της Τεχνητής Νοημοσύνης

Ένας δημοφιλής κλάδος της Τεχνητής Νοημοσύνης, είναι αυτός της Υπολογιστικής Όρασης ή της Όρασης των Υπολογιστών (Computer Vision) ο οποίος στοχεύει στην αποδοτική περιγραφή του περιεχομένου μιας εικόνας ή μιας αλληλουχίας από εικόνες (βίντεο), από ένα υπολογιστικό σύστημα. Γίνεται δηλαδή προσπάθεια να αποκτήσει ο υπολογιστικής τεχνητά την αίσθηση της όρασης και των τεχνικών οπτικής ανάλυσης που διαθέτουν οι βιολογικοί οργανισμοί. Τα τελευταία χρόνια με την αύξηση της τεχνολογίας ο κλάδος αυτός παρουσιάζει σημαντική άνθηση και χρησιμοποιείται σε συνδυασμό με αλγορίθμους της Μηχανικής Μάθησης. Η Υπολογιστική Όραση έχει εφαρμογές σε πολλά διαφορετικά επιστημονικά πεδία όπως αυτά της Ιατρικής, της Ρομποτικής, της Νευροβιολογίας και της Γεωργίας.

Οι αλγόριθμοι της Μηχανικής Μάθησης έχουν επιτρέψει την επίλυση πολλών προβλημάτων με μεγάλη πολυπλοκότητα. Νέα μοντέλα και τεχνικές δημοσιεύονται πλέον με εντυπωσιακό ρυθμό και χρόνο με το χρόνο η ακρίβεια των αλγορίθμων ολοένα και αυξάνεται. Ένα υποσύνολο της Μηχανικής Μάθησης είναι η Βαθιά Μάθηση που χρησιμοποιεί βαθιά Νευρωνικά Δίκτυα για την επίλυση προβλημάτων. Μια γνωστή αρχιτεκτονική βαθιών νευρωνικών δικτύων είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks ή CNN), τα οποία έχουν εξαιρετικά αποτελέσματα σε προβλήματα Υπολογιστικής Όρασης και πλέον θεωρούνται ο καλύτερος τρόπος για την επίλυση τέτοιων προβλημάτων ανάλυσης εικόνων και βίντεο.

Με τη χρήση τέτοιου είδους αρχιτεκτονικών μπορούμε να κατασκευάσουμε μοντέλα με εξαιρετικές δυνατότητες γενίκευσης της γνώσης που μαθαίνουν και να τα επαναχρησιμοποιήσουμε σε άλλα προβλήματα με μικρές αλλαγές. Για παράδειγμα, στην αναγνώριση αντικειμένων σε εικόνες (object recognition), χρησιμοποιούμε αρχιτεκτονικές οι οποίες έχουν ήδη ένα γενικό τρόπο εκτέλεσης των υπολογισμών και εμείς απλώς τροφοδοτούμε το δίκτυο με εικόνες από το εκάστοτε αντικείμενο που ζητούμε από το μοντέλο να αναγνωρίσει. Δηλαδή το νευρωνικό δίκτυο μαθαίνει γενικά τη διαδικασία που πρέπει να ακολουθήσει για να εντοπίζει ένα οποιοδήποτε αντικείμενο, και αναμένει μια τελική εκπαίδευση από το χρήστη για προσαρμοστεί στο εκάστοτε πρόβλημα.

1.2.1 Εφαρμογές της Υπολογιστικής Όρασης

Ιατρική

Χαρακτηριστικό παράδειγμα εφαρμογής είναι η ανάλυση ιατρικών εικόνων, όπως είναι οι ακτινογραφίες και τα αποτελέσματα μιας μαγνητικής τομογραφίας. Μέθοδοι υπολογιστικής όρασης χρησιμοποιούνται για ιατρικές διαγνώσεις και σε πολλές παθήσεις που απαιτείται οπτική ανάλυση εξετάσεων έχουν αποδειχθεί ίδιας αποτελεσματικότητας με τις διαγνώσεις των γιατρών. Ιατροί διαφόρων ειδικοτήτων χρησιμοποιούν τέτοιου είδους εργαλεία για την ανάλυση πολυδιάστατων εικόνων ώστε να εντοπίσουν όγκους και άλλες πιθανές ανωμαλίες.

Αυτό-οδηγούμενα οχήματα

Οι βιομηχανίες οχημάτων τα τελευταία χρόνια έχουν δώσει αρκετή σημασία στην κατασκευή οχημάτων με την ικανότητα να κινούνται χωρίς ανθρώπινη επίβλεψη. Ένα όχημα μπορεί πλέον να εξοπλιστεί με κάμερες και αισθητήρες και να αποκτήσει την ικανότητα να αναγνωρίζει τις οδικές λωρίδες, τους φωτεινούς σηματοδότες, πλανόδιους πεζούς ακόμη και την πρόβλεψη ενός ατυχήματος, υπολογίζοντας τις πιθανές μελλοντικές τροχιές των αντικειμένων του περίγυρού του. Η μαζική χρήση τέτοιου είδους έξυπνων οχημάτων αναμφίβολα θα έχει σημαντική επίδραση σε πολλούς τομείς της κοινωνίας.

Στρατιωτικές εφαρμογές

Η Υπολογιστική Όραση χρησιμοποιείται εκτενώς σε πολλές στρατιωτικές εφαρμογές και μπορεί να ενσωματωθεί σε διάφορα οπλικά συστήματα. Παραδείγματα αυτών είναι η οπτική αναγνώριση εχθρικών στρατευμάτων ή οχημάτων, η καθοδήγηση πυραύλων και η αυτόματη ανάλυση περιοχών ενδιαφέροντος. Τα drones που χρησιμοποιούν πολλοί στρατοί, διαθέτουν συστήματα αναγνώρισης προσώπου και αντικειμένων και μπορούν να συλλέγουν αυτόματα χρήσιμα δεδομένα χωρίς ανθρώπινη επίβλεψη.

Αυτοματισμοί σε γραμμές παραγωγής

Πολλές βιομηχανίες παραγωγής προϊόντων χρησιμοποιούν τεχνικές Υπολογιστικής Όρασης για πληθώρα εργασιών. Οι γραμμές παραγωγής εφοδιάζονται με ρομποτικά συστήματα τα οποία επιτελούν αυτόματες διεργασίες έχοντας ενσωματωμένες κάμερες. Ένα εργοστάσιο μπορεί να ελέγχει αν οι εργαζόμενοι φορούν τον απαραίτητο εξοπλισμό όπως μάσκες προστασίας, κράνη, γάντια κ.λπ. Επίσης υπάρχουν πολλές εφαρμογές στον έλεγχο ποιότητας προϊόντων καθώς τέτοιου είδους διεργασίες συχνά απαιτούν μεγάλο πλήθος υπαλλήλων.

Αλληλεπίδραση ανθρώπου-υπολογιστή

Σημαντική εφαρμογή της Υπολογιστικής Όρασης είναι η αλληλεπίδραση του ανθρώπου με υπολογιστικά συστήματα με τη χρήση οπτικών μέσων. Ο υπολογιστής πλέον μπορεί να αντιληφθεί τις φυσικές κινήσεις του ανθρώπου και με βάση αυτές να εκτελέσει μια προκαθορισμένη εντολή. Ρομποτικά συστήματα μπορούν να μιμούνται τις ανθρώπινες κινήσεις και έτσι να είναι εφικτός ο έλεγχός τους απομακρυσμένα και χωρίς επιπλέον εξοπλισμό χρησιμοποιώντας μόνο μια κάμερα. Είναι ήδη διαθέσιμα πολλά μοντέλα που εστιάζουν στην ανάλυση κίνηση και αναγνωρίζουν τα ανθρώπινα μέλη (πρόσωπο, χέρια, πόδια κ.λπ.) καθώς και τις κινήσεις που κάνει καθένα από αυτά, με μεγάλη ακρίβεια.

Συστήματα παρακολούθησης

Σε συστήματα παρακολούθησης είναι εφικτή η αυτόματη αναγνώριση συγκεκριμένων συμβάντων μέσω καμερών. Παραδείγματα τέτοιου είδους συμβάντων είναι η αναγνώριση της ανθρώπινης παρουσίας, η ταυτοποίηση ενός ανθρώπου με βάση τα χαρακτηριστικά του προσώπου του (face recognition) και η αναγνώριση κακόβουλων ενεργειών όπως η προσέγγιση απαγορευμένων σημείων, ο εντοπισμός οχημάτων (vehicle detection) και η αναγνώριση του αριθμού της πινακίδας τους (license plate recognition).

Γεωργία

Στη Γεωργία η Υπολογιστική Όραση μπορεί να προβλέψει την αναμενόμενη ωφέλιμη σοδιά (yield estimation), να αναγνωρίσει την παρουσία ασθενειών σε φυτά (plant disease recognition), να χρησιμοποιηθεί στην παραγωγή αυτόματων οχημάτων για γεωργικές εργασίες, να ελέγξει την ποιότητα τροφίμων (food quality control) κ.λπ.

1.3 Το πρόβλημα της αναγνώρισης χειρονομιών

Ένα είδος τεχνολογίας που έχει αναδυθεί τα τελευταία χρόνια και έχει πολλές σημαντικές εφαρμογές είναι αυτό της χρήσης διεπαφών (interfaces) χωρίς να απαιτείται η αφή (touchless), αλλά μέσω κινήσεων του σώματος. Ο άνθρωπος αλληλοεπιδρά με το περιβάλλον συνήθως χρησιμοποιώντας τα χέρια του και έτσι η αναγνώριση των χεριών και των κινήσεών τους έχει αποτελέσει ένα πολύ ενδιαφέρον ζήτημα εξαιρετικής σημασίας.

Για να χρησιμοποιήσουμε έναν ηλεκτρονικό υπολογιστή διαθέτουμε φυσικές συσκευές ελέγχου όπως είναι το ποντίκι και το πληκτρολόγιο. Πλέον είναι εφικτός ο χειρισμός ηλεκτρονικών συσκευών χωρίς να πρέπει να αγγίξουμε τίποτα, δηλαδή χωρίς την απαίτηση συσκευών ελέγχου. Αντί να χρησιμοποιούμε το ποντίκι μπορούμε να εκπαιδεύσουμε τον υπολογιστή να αναγνωρίζει, μέσω μιας κάμερας, ένα δάχτυλο του χεριού ώστε να μετακινείται ο κέρσορας ανάλογα με την κίνηση του δακτύλου, και να ορίσουμε μια συγκεκριμένη κίνηση να αντιστοιχεί στην επιλογή (κλικ) που θα κάναμε παραδοσιακά με ένα ποντίκι. Ένα άλλο παράδειγμα είναι σε ζητήματα ασφάλειας. Μπορούμε να ορίσουμε κάποιες χειρονομίες ως τον προσωπικό κωδικό για να εκτελεστεί μια ενέργεια όπως το ξεκλείδωμα του κινητού τηλεφώνου ή μιας ηλεκτρικής πόρτας. Εκτός όμως από τον έλεγχο συσκευών μια σπουδαία εφαρμογή είναι η αναγνώριση νοημάτων που κάνουμε με τα χέρια όπως για παράδειγμα στη περίπτωση της νοηματικής γλώσσας. Πλέον είναι εφικτή η αυτόματη αναγνώριση της νοηματικής γλώσσας χρησιμοποιώντας απλές κάμερες. Επίσης, στην περίοδο της πανδημίας του COVID-19 έγινε ιδιαίτερα αντιληπτή η αξία του χειρισμού συστημάτων χωρίς να απαιτείται το άγγιγμα των συσκευών, καθώς αποτρέπεται η μετάδοση μικροβίων μέσω της αφής.

Αρχικά, το πρόβλημα της αναγνώρισης χειρονομιών (hand gesture recognition) αντιμετωπίστηκε με τη χρήση αισθητήρων ενσωματωμένων σε γάντια μέσω των οποίων γίνεται οι αναγνώριση των κινήσεων των δαχτύλων του χεριού. Ένα παράδειγμα είναι το γάντι acceleGlove που κατασκευάστηκε στο M.I.T. και χρησιμοποιήθηκε στη ρομποτική.

Εικόνα 1-2: Το γάντι acceleGlove [1]

Στη συνέχεια, αντί για γάντια χρησιμοποιήθηκαν οπτικά μέσα και πλέον το πρόβλημα της αναγνώρισης χειρονομιών δεν απαιτεί απαραίτητα τη χρήση αισθητήρων και γίνεται αποδοτικά μέσω καμερών, χρησιμοποιώντας μεθόδους της Υπολογιστικής Όρασης και προχωρημένα μοντέλα Μηχανικής Μάθησης και Βαθιάς Μάθησης, όπως γίνεται και στην παρούσα εργασία. Γενικά, υπάρχουν τεχνικές που απαιτούν συγκεκριμένο εξοπλισμό, όπως για παράδειγμα τη χρήση καμερών βάθους (depth-aware cameras) οι οποίες μπορούν να κάνουν εκτίμηση του βάθους των αντικειμένων της εικόνας. Μια άλλη τεχνική απαιτεί την ταυτόχρονη χρήση πολλών καμερών, τοποθετώντας τες σε μια συγκεκριμένη τοπολογία. Τα πρόσφατα χρόνια γίνεται μεγάλη προσπάθεια στο να μπορούμε να αναγνωρίζουμε τις κινήσεις των χεριών χρησιμοποιώντας μία μόνο κάμερα απλού τύπου.

Οι αλγόριθμοι για την αναγνώριση χειρονομιών διαφέρουν ως προς τον τρόπο που αντιμετωπίζουν το πρόβλημα και γενικά χωρίζονται σε δύο κατηγορίες. Αυτές που θεωρούν το πρόβλημα ως κλασσικό πρόβλημα κατηγοριοποίησης εικόνων (Appearance-based), και αυτές που βασίζονται στη χρήση των συντεταγμένων σημείων κλειδιά (key points) του χεριού (3D-model based). Στην πρώτη περίπτωση θεωρούμε μια χειρονομία ως ένα αντικείμενο ή μια εικόνα, το οποίο αναζητούμε σε νέες εικόνες. Στη δεύτερη, θεωρούμε μια χειρονομία ως ένα σύνολο από σημεία στο χώρο ή στο επίπεδο, δηλαδή ως απλά αριθμητικά δεδομένα. Η τεχνική που χρησιμοποιούμε στην παρούσα εργασία ανήκει στη δεύτερη κατηγορία και βασίζεται στην ανάλυση του σκελετού του χεριού (Skeletal-based). Συγκεκριμένα, εντοπίζουμε ένα μικρό σχετικά πλήθος από σημεία-κλειδιά (key-points) του χεριού και τα ενώνουμε μεταξύ τους με ευθύγραμμα τμήματα, για να πάρουμε μια καλή προσέγγιση του πραγματικού σχήματος του σκελετού του χεριού. Αυτό προσφέρει σημαντικά πλεονεκτήματα καθώς το μοντέλο εστιάζει μόνο σε ένα μικρό πλήθος σημείων, αντί να επεξεργάζεται όλα τα σημεία του χεριού.

Εικόνα 1-3: Παράδειγμα προσέγγισης του σκελετού του χεριού [2]

1.3 Σκοπός της διπλωματικής

Η Μηχανική Μάθηση προσφέρει πληθώρα μεθόδων και τεχνικών για την ανάλυση δεδομένων και πλέον αποτελεί το βασικό εργαλείο για την επίλυση προβλημάτων της Υπολογιστικής Όρασης. Ο βασικός σκοπός της παρούσας διπλωματικής εργασίας είναι να αξιοποιήσουμε σύγχρονες μεθόδους και καινοτόμους αλγορίθμους της Μηχανικής Μάθησης, για την αναγνώριση προκαθορισμένων στατικών χειρονομιών σε ζωντανή ροή βίντεο.

Το πρόβλημα αυτό γενικά απαιτεί αρκετή υπολογιστική ισχύ και γι'αυτό χρησιμοποιούνται συχνά σύγχρονες κάρτες γραφικών, καθώς διαθέτουν τάξης μεγέθους περισσότερους πυρήνες απ'ότι οι επεξεργαστές των υπολογιστών καθώς και δυνατότητες παράλληλης επεξεργασίας. Με τις πρόσφατες καινοτομίες στο χώρο της Μηχανικής Μάθησης και της Βαθιάς Μάθησης, έχουν δημοσιευτεί εργαλεία και αρχιτεκτονικές αλγορίθμων μέσω των οποίων μπορούμε να πετύχουμε εξίσου καλά αποτελέσματα, κάνοντας χρήση μόνο του επεξεργαστή ενός υπολογιστή. Έτσι το κόστος για πρακτικές εφαρμογές μειώνεται σημαντικά και δεν απαιτείται η αγορά νέου εξοπλισμού. Αυτός είναι και ένας ακόμη στόχος της παρούσας εργασίας, να πετύχουμε δηλαδή αναγνώριση χειρονομιών πραγματικού χρόνου, χωρίς να απαιτείται μεγάλη υπολογιστική ισχύ.

Για την υλοποίηση έγινε προσπάθεια χρήσης τον πιο σύγχρονων τεχνικών και αλγορίθμων με γνώμονα την ταχύτητα και την ελαχιστοποίηση των υπολογιστικών απαιτήσεων. Χρησιμοποιούμε έναν πρόσφατα δημοσιευμένο αλγόριθμο συνελικτικών νευρωνικών δικτύων για τον εντοπισμό της παλάμης και των δακτύλων του χεριού σε μια εικόνα. Στη συνέχεια, εκπαιδεύουμε μια μηχανή διανυσμάτων υποστήριξης και ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης για την αναγνώριση στατικών χειρονομιών σε βίντεο μέσω μιας απλής κάμερας RGB. Ο βασικός στόχος της υλοποίησης είναι να μπορεί να χρησιμοποιηθεί σε πραγματικό χρόνο. Επίσης ο κώδικας είναι δομημένος με τέτοιο τρόπο ώστε να μπορεί να χρησιμοποιηθεί εύκολα χρησιμοποιώντας διαφορετικό σύνολο δεδομένων (χειρονομιών), από αυτό που χρησιμοποιούμε στην παρούσα εργασία.

1.4 Οργάνωση της διπλωματικής

Στο Κεφάλαιο 2 κάνουμε μια εισαγωγή στο πεδίο της Μηχανικής Μάθησης, των προβλημάτων που πραγματεύεται και των αλγορίθμων που χρησιμοποιούνται για την επίλυσή τους. Εστιάζουμε στο πρόβλημα της Κατηγοριοποίησης και παρουσιάζουμε το μοντέλο των Μηχανών Διανυσμάτων Υποστήριξης.

Στο Κεφάλαιο 3 αναφερόμαστε στη Βαθιά Μάθηση. Παρουσιάζουμε συνοπτικά τη λειτουργία των νευρωνικών δικτύων, τις διάφορες αρχιτεκτονικές που χρησιμοποιούνται καθώς και τον τρόπο εκπαίδευσής τους. Επίσης περιγράφουμε μερικούς από τους πιο γνωστούς αλγόριθμους που χρησιμοποιούνται στην Υπολογιστική Όραση και χρησιμοποιούν Συνελικτικά Νευρωνικά Δίκτυα.

Στο Κεφάλαιο 4 γίνεται η υλοποίηση του εργαλείου για την αναγνώριση των χειρονομιών. Παρουσιάζουμε τη μεθοδολογία που ακολουθούμε καθώς και τα εργαλεία που χρησιμοποιήθηκαν. Στη συνέχεια περιγράφουμε τα μοντέλα Μηχανικής Μάθησης και Βαθιάς Μάθησης που αναπτύξαμε για την επίλυση του προβλήματος και σχολιάζουμε τα αποτελέσματά τους.

Στο Κεφάλαιο 5 διατυπώνονται τα συμπεράσματα που προκύπτουν καθώς και αναφορές για μελλοντικές βελτιώσεις.

1.5 Τεχνολογίες που χρησιμοποιήθηκαν

Παρακάτω γίνεται μια συνοπτική παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνηση της παρούσας διπλωματικής εργασίας.

Το υπολογιστικό περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση αποτελείται από επεξεργαστή AMD Ryzen 5 – 1600 6c 3.2 GHz, μνήμη RAM DDR5 16GB, σκληρό δίσκο SSD και κάμερα η Logitech C170. Ως λειτουργικό σύστημα χρησιμοποιήθηκε Linux – Ubuntu.

Python

Η γλώσσα προγραμματισμού Python είναι μια από τις πιο δημοφιλείς γλώσσες γενικού σκοπού, η οποία έχει αναγνωριστεί ως το βασικό εργαλείο για τον προγραμματισμό σε θέματα ανάλυσης δεδομένων και σε εφαρμογές Τεχνητής Νοημοσύνης όπως είναι αυτή του θέματος της παρούσας διπλωματικής εργασίας. Είναι ιδιαίτερα φιλική προς το χρήστη και έχει τη δυνατότητα να εκτελεί πολύπλοκες διεργασίες με σημαντικά λιγότερες γραμμές κώδικα απ'ότι άλλες δημοφιλείς γλώσσες προγραμματισμού. Η χρήση της δεν περιορίζεται σε επιστημονικούς τομείς καθώς χρησιμοποιείται εκτενώς και με μεγάλη επιτυχία, για την πλειοψηφία των τεχνολογικών αναγκών όπως οι διαδικτυακές εφαρμογές και η ανάπτυξη ιστοσελίδων, η διαχείριση συστημάτων βάσεων δεδομένων και η κατασκευή διεπιφανειών χρήστη.

OpenCV

Η βιβλιοθήκη ανοιχτού κώδικα OpenCV (Open Computer Vision) περιέχει πληθώρα λειτουργιών προγραμματισμού με βασικό αντικείμενο τις εφαρμογές στην Υπολογιστική Όραση. Περιέχει παραπάνω από 2500 αλγορίθμους Μηχανικής Μάθησης που σχετίζονται με την όραση των υπολογιστών και χρησιμοποιείται ιδιαίτερα στις περισσότερες εμπορικές εφαρμογές που σχετίζονται με τέτοιους τομείς. Ενδεικτικές δυνατότητες που προσφέρει είναι η αναγνώριση προσώπων και αντικειμένων σε εικόνες και βίντεο, ο εντοπισμός ανθρωπίνων κινήσεων και κινήσεων αντικειμένων, η εξαγωγή τρισδιάστατων μοντέλων αντικειμένων και η επεξεργασία εικόνων. Υποστηρίζει τις γλώσσες προγραμματισμού Python, C++ , Java και MATLAB και είναι διαθέσιμη για όλα τα γνωστά λειτουργικά συστήματα καθώς και για συσκευές Android.

Scikit-Learn

Η βιβλιοθήκη Scikit-Learn ή sklearn είναι η πιο διαδεδομένη βιβλιοθήκη της γλώσσας προγραμματισμού Python που αποτελείται από πολλές μεθόδους της Μηχανικής Μάθησης. Χρησιμοποιείται για προβλήματα κατηγοριοποίησης, παλινδρόμησης και συσταδοποίησης και είναι εξοπλισμένη με τους πιο σύγχρονους αλγορίθμους. Παρέχει επίσης πολλές μεθόδους μείωσης διαστάσεων και προεπεξεργασίας των δεδομένων.

Tensorflow & Keras

Η βιβλιοθήκη Tensorflow αποτελεί την πιο διαδεδομένη πλατφόρμα για την κατασκευή και διαχείριση μοντέλων Μηχανικής Μάθησης. Χρησιμοποιείται από αρχάριους αλλά και από προχωρημένους χρήστες και προσφέρει πολλές λειτουργίες για την εύκολη εκπαίδευση μοντέλων που μέχρι πρότινος απαιτούσαν πολλές γραμμές κώδικα. Παρέχει όλα τα εργαλεία που κάποιος χρειάζεται για να χρησιμοποιήσει κάποιος ιδιαίτερα πολύπλοκες και σύνθετες μεθόδους και αλγορίθμους της Μηχανικής Μάθησης. Η βιβλιοθήκη Keras είναι ένα δημοφιλές εργαλείο που παρέχει τη δυνατότητα σχεδίασης σε υψηλό επίπεδο των δομών που χρησιμοποιούνται στους τομείς της Βαθιάς Μηχανικής Μάθησης. Χρησιμοποιείται ως διεπαφή για τη βιβλιοθήκη Tensorflow και υποστηρίζει τη γλώσσα προγραμματισμού Python. Είναι φιλική προς τον χρήστη και έχει σχεδιαστεί με τέτοιο τρόπο ώστε ο πειραματισμός με τα Νευρωνικά Δίκτυα να είναι όσο το δυνατόν λιγότερο χρονοβόρος, αρκετά παραμετροποιήσιμος αλλά και ανεκτικός σε δομικές αλλαγές.

MediaPipe

Πρόκειται για ένα ανοιχτής πρόσβασης πλαίσιο λογισμικού της Google που χρησιμοποιείται για το σχεδιασμό και την υλοποίηση διαδικασιών Μηχανικής Μάθησης σε δεδομένα χρονοσειρών όπως είναι οι εικόνες, τα βίντεο κ.λπ. Το MediaPipe είναι εφοδιασμένο με καινοτόμους αλγορίθμους και υποστηρίζει όλα τα λειτουργικά συστήματα καθώς και φορητές συσκευές, και είναι πλέον μία από τις πιο διαδεδομένες και εύχρηστες λύσεις για την ενασχόληση με ζητήματα Υπολογιστικής Όρασης και την εφαρμογή των πιο πρόσφατων αλγορίθμων της Google. Χαρακτηριστικές εφαρμογές που προσφέρει το MediaPipe είναι η αναγνώριση αντικειμένων, προσώπου, χεριών και ίριδας του ματιού.

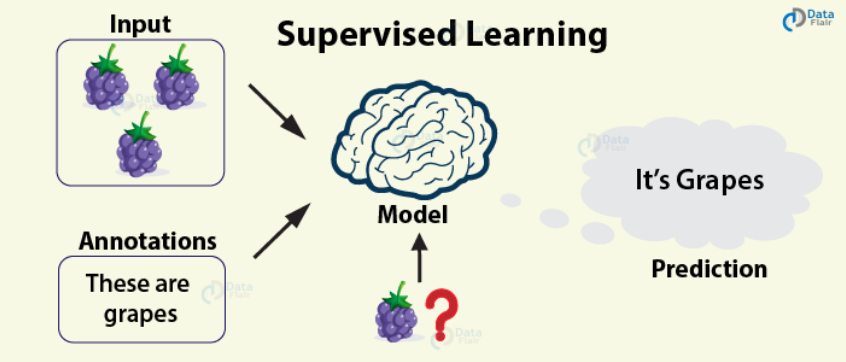
ΚΕΦΑΛΑΙΟ 2

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η Μηχανική Μάθηση (Machine Learning) αποτελεί ένα πεδίο της Τεχνητής Νοημοσύνης που δίνει την ικανότητα της εκμάθησης στα υπολογιστικά συστήματα. Στόχος είναι ο σχεδιασμός κατάλληλων αλγορίθμων, ώστε ο υπολογιστής να μάθει να αναλύει πειραματικά δεδομένα με τέτοιο τρόπο ώστε να εξάγει από αυτά χρήσιμες πληροφορίες, να παίρνει αποφάσεις και να κάνει προβλέψεις. Η Μηχανική Μάθηση έχει άμεση σχέση με την Επιστήμη των Υπολογιστών την Υπολογιστική Στατιστική, τη Μαθηματική Βελτιστοποίηση και την Εξόρυξη γνώσης (Data mining).

2.1 Είδη Μάθησης

2.1.1 Επιτηρούμενη Μάθηση



Εικόνα 2-1: Επιτηρούμενη Μάθηση [3]

Στην *Επιτηρούμενη Μάθηση (Supervised Learning)* ο υπολογιστής εκπαιδεύεται σε δεδομένα της μορφής (είσοδος, έξοδος), στο να μαθαίνει να αντιστοιχεί τα δεδομένα εισόδου στα δεδομένα εξόδου. Εκπαιδεύεται δηλαδή να αναγνωρίζει τον τρόπο με τον οποίο σχετίζονται μεταξύ τους τα δεδομένα, ώστε να μπορεί να μιμηθεί τον τρόπο αυτό και να μπορεί να προβλέψει την έξοδο νέων δεδομένων εισόδου. Τα κύρια προβλήματα της επιτηρούμενης μάθησης είναι η *Κατηγοριοποίηση ή Ταξινόμηση (Classification)* και η *Παλινδρόμηση (Regression)*.

Κατηγοριοποίηση (Classification)

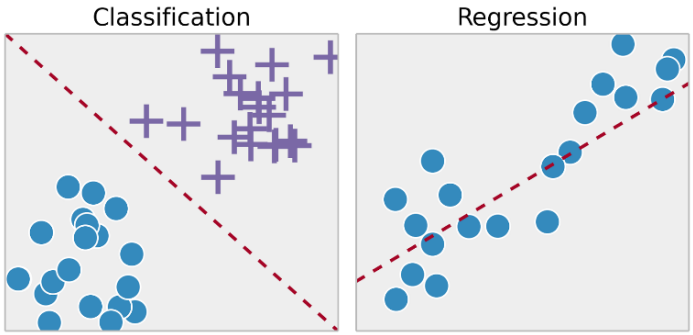
Στα προβλήματα Κατηγοριοποίησης, τα δεδομένα περιέχουν σύνολα παρατηρήσεων τα οποία αντιστοιχούν σε κάποιες διακριτές και προκαθορισμένες κατηγορίες ή κλάσεις. Ενδεικτικά παραδείγματα κατηγοριοποίησης είναι ο εντοπισμός spam ηλεκτρονικών μηνυμάτων (email spam detection), η αναγνώριση ενός χειρόγραφου χαρακτήρα μεταξύ των γνωστών χαρακτήρων της φυσικής γλώσσας (optical character recognition) και η κατηγοριοποίηση μιας κριτικής για ένα προϊόν ως θετική ή όχι (sentiment analysis). Για την επίλυση προβλημάτων κατηγοριοποίησης έχουν αναπτυχθεί πολλοί αλγόριθμοι και γενικά η επιλογή μεταξύ αυτών γίνεται ανάλογα με τη φύση του εκάστοτε προβλήματος. Γνωστοί αλγόριθμοι κατηγοριοποίησης είναι:

- Λογιστική Παλινδρόμηση (Logistic Regression)
- Naïve Bayes Classifier
- Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines ή SVM)
- k-Πλησιέστεροι Γείτονες (k-Nearest Neighbors ή k-NN)
- Τυχαία Δάση (Random Forests)
- Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks ή ANN)

Παλινδρόμηση (Regression)

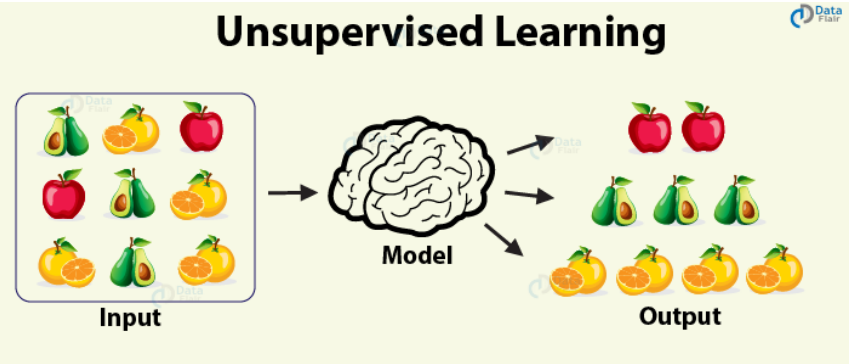
Στα προβλήματα Παλινδρόμησης, οι παρατηρήσεις δεν αναφέρονται σε διακριτές κλάσεις και το αποτέλεσμα είναι μια ποσοτική μεταβλητή, δηλαδή αριθμητικές ποσότητες. Η ανάλυση παλινδρόμησης είναι ένα κλασσικό πρόβλημα της Στατιστικής, παραδείγματα εφαρμογών της είναι η πρόβλεψη οικονομικών χρονοσειρών, η πρόβλεψη πωλήσεων για ένα προϊόν κ.λπ. Ενδεικτικοί αλγόριθμοι παλινδρόμησης είναι:

- Γραμμική Παλινδρόμηση (Linear Regression)
- Πολυωνυμική Παλινδρόμηση (Polynomial Regression)
- Penalized Παλινδρόμηση (Lasso, Ridge, Elastic Net)
- Παλινδρόμηση με Διανύσματα Υποστήριξης (Support Vector Regression ή SVR)
- Παλινδρόμηση με Κατάβαση Δυναμικού (Gradient Decent Regression)



Εικόνα 2-2 Παράδειγμα Κατηγοριοποίησης (αριστερά) και Παλινδρόμησης (Δεξιά) [4]

2.1.2 Μη-Επιτηρούμενη Μάθηση



Εικόνα 2-3: Μη-Επιτηρούμενη Μάθηση [5]

Στη *Μη-Επιτηρούμενη Μάθηση (Unsupervised Learning)* τα δεδομένα δεν διαθέτουν κάποιο γνωστό αποτέλεσμα (unlabeled data) και ο σκοπός του αλγορίθμου είναι να εντοπίσει επιμέρους δομές και κρυμμένα μοτίβα που εμπεριέχονται στα δεδομένα. Βασικοί τομείς πάνω στους οποίους εδράζεται η μη-επιτηρούμενη μάθηση είναι η ανάλυση συστάδων (cluster analysis) και η ανάλυση πρώτων παραγόντων (principal component analysis). Οι κύριες εφαρμογές της είναι ο σχεδιασμός αλγορίθμων για συσταδοποίηση (clustering), εύρεση κανόνων σύνδεσης (association rules) και η μείωση διαστάσεων των δεδομένων (dimensionality reduction).

Συσταδοποίηση (Clustering)

Η *Συσταδοποίηση* είναι μια τεχνική εξόρυξης γνώσης η οποία ομαδοποιεί τα δεδομένα με βάση ομοιότητες ή διαφορές που παρουσιάζουν μεταξύ τους. Υπάρχουν αρκετές τεχνικές διαφορετικής φύσεως για συσταδοποίηση, άλλες χρησιμοποιούν την απόσταση των δεδομένων μεταξύ τους ως προς διάφορες μετρικές, για να εντοπίσουν πυκνές περιοχές και να τις ομαδοποιήσουν σε μια συστάδα, ενώ άλλες χρησιμοποιούν πιθανοτικά μοντέλα όπου τα δεδομένα συσταδοποιούνται ανάλογα με την πιθανότητα να ανήκουν σε μια ορισμένη στατιστική κατανομή. Ενδεικτικές εφαρμογές της Συσταδοποίησης είναι τα Συστήματα Προτάσεων (Recommender Systems), ο Εντοπισμός Ανωμαλιών (Anomaly Detection) , η εύρεση κοινοτήτων σε κοινωνικά δίκτυα (social network analysis) και η Τμηματοποίηση Εικόνας (Image Segmentation).

Γνωστοί αλγόριθμοι Συσταδοποίησης είναι οι εξής:

- Hierarchical Clustering
- K-Means Clustering
- DBSCAN (Density-based spatial clustering of applications with noise)
- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
- OPTICS (Ordering points to identify the clustering structure)

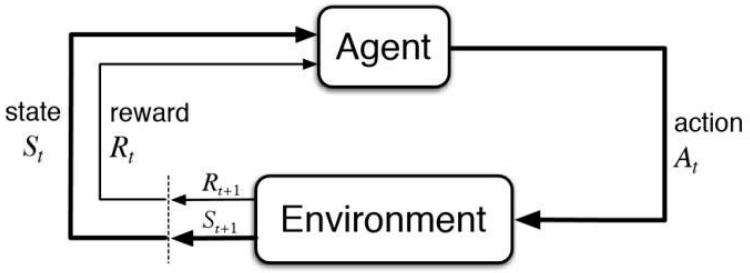
Κανόνες Σύνδεσης (Association Rules)

Η εύρεση *κανόνων σύνδεσης* αναφέρεται σε μεθόδους της μηχανικής μάθησης που στοχεύουν στον εντοπισμό χρήσιμων σχέσεων (κανόνων) μεταξύ μεταβλητών σε σύνολα δεδομένων. Χρησιμοποιούνται για να προσδιορίσουν τη συσχέτιση που παρουσιάζουν δεδομένα σε μεγάλες βάσεις δεδομένων και αρχικά χρησιμοποιήθηκαν για την εύρεση συσχετίσεων μεταξύ των προϊόντων που αγοράζουν οι καταναλωτές. Μέσω αυτών, μπορούμε να εντοπίσουμε δεσμούς μεταξύ διαφορετικών προϊόντων και να προβλέψουμε τα προϊόντα που πιθανώς να προτιμήσει ένας καταναλωτής, δεδομένου ότι έχει ήδη αγοράσει κάποια άλλα.

Μείωση Διαστάσεων (Dimensionality Reduction)

Με τον όρο μείωση των διαστάσεων, αναφερόμαστε σε τεχνικές που χρησιμοποιούμε όταν έχουμε να αντιμετωπίσουμε προβλήματα με μεγάλο πλήθος μεταβλητών. Μέσω αυτών των τεχνικών μπορούμε να περιορίσουμε σημαντικά το πλήθος των μεταβλητών που έχουμε να επεξεργαστούμε, αυξάνοντας έτσι την απόδοση των μοντέλων μηχανικής μάθησης και επιτρέποντας πολλές φορές την οπτικοποίηση των δεδομένων που αρχικώς ήταν αδύνατη. Το πιο γνωστό παράδειγμα μεθόδου μείωσης διαστάσεων προκύπτει από τη Στατιστική και τη Γραμμική Άλγεβρα και είναι η Ανάλυση σε Κύριους Παράγοντες (Principal Component Analysis ή PCA).

2.1.3 Ενισχυτική Μάθηση



Εικόνα 2-4: Ενισχυτική Μάθηση [6]

Η *Ενισχυτική Μάθηση (Reinforcement Learning)* εξετάζει προβλήματα στα οποία το σύστημα, μέσω κάποιων υπολογιστικών πρακτόρων (agents), μαθαίνει από το περιβάλλον που εξετάζει (environment) διαδραστικά και χωρίς να έχει απαραίτητα πρότερη γνώση. Οι πράκτορες αναλαμβάνουν δράσεις (actions) και προσπαθούν να πετύχουν συλλογικά το μέγιστο όφελος, μέσω ενός συστήματος επιβράβευσης όταν εκτελούν μια ωφέλιμη ενέργεια.

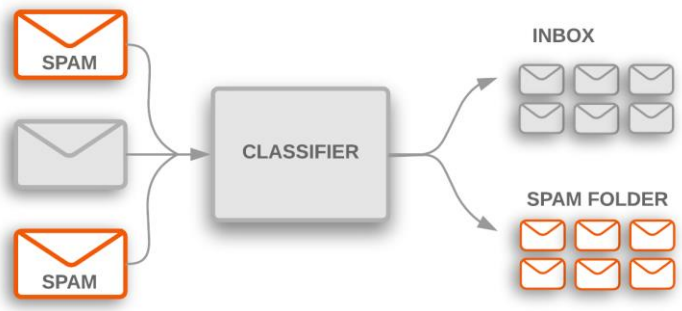
Ο τομέας αυτός της μηχανικής μάθησης έχει μεγάλη σύνδεση με το γραμμικό προγραμματισμό καθώς χρησιμοποιεί πολλές τέτοιες τεχνικές και συνήθως χρησιμοποιεί Μαρκοβιανές διαδικασίες απόφασης (Markov decision processes) που είναι μια επέκταση των Μαρκοβιανών αλυσίδων που συναντούμε στις Στοχαστικές Ανελίξεις.

2.2 Κατηγοριοποίηση Δεδομένων

Όπως αναφέρθηκε και στην προηγούμενη παράγραφο της εποπτευόμενης μάθησης, η *Κατηγοριοποίηση* ή *Ταξινόμηση (Classification)* είναι το πρόβλημα της αντιστοίχισης μιας παρατήρησης σε μια κλάση (class) ή κατηγορία. Αυτό γίνεται υπολογίζοντας μια συνάρτηση η οποία μαθαίνει να αντιστοιχεί τα δεδομένα εισόδου σε μία από τις κατηγορίες και ονομάζεται *κατηγοριοποιητής* ή *ταξινομητής (classifier).*

Για την εκπαίδευση ενός κατηγοριοποιητή χρησιμοποιούμε ένα σύνολο δεδομένων που περιλαμβάνει ζεύγη παρατηρήσεων και των αντίστοιχων κατηγοριών που ανήκουν. Ο σκοπός του κατηγοριοποιητή είναι να μπορέσει να κατηγοριοποιήσει επιτυχώς μια νέα παρατήρηση με βάση την εμπειρία που έμαθε κατά τη διάρκεια της εκπαίδευσής του.

Χαρακτηριστικό παράδειγμα είναι η κατηγοριοποίηση ενός ηλεκτρονικού μηνύματος σε spam ή όχι και φαίνεται σχηματικά στην παρακάτω εικόνα. Ο κατηγοριοποιητής μαθαίνει να επεξεργάζεται την επικεφαλίδα και το περιεχόμενο του μηνύματος αναλύοντας τη φυσική γλώσσα και συγκρίνει τα χαρακτηριστικά του με άλλα μηνύματα παρόμοιων χαρκτηριστικών για να λάβει την τελική του απόφαση.



Εικόνα 2-5: Spam Classification [7]

Ανάλογα με το πλήθος των κλάσεων η κατηγοριοποίηση μπορεί να αναφέρεται ως *Binary* (Δυαδική) για προβλήματα με δύο κλάσεις, ή ως *Multiclass* classification για προβλήματα με πολλές κλάσεις όπως στην περίπτωση της παρούσας εργασίας. Οι περισσότεροι ταξινομητές έχουν ως στόχο τη δυαδική κατηγοριοποίηση και γι'αυτό συνήθως τα προβλήματα με πολλές κλάσεις αντιμετωπίζονται με την παράλληλη χρήση πολλών δυαδικών κατηγοριοποιητών.

Γενικά, οι αλγόριθμοι ταξινόμησης χωρίζονται σε δύο μεγάλες κατηγορίες. Σε αυτούς που επεξεργάζονται κάθε φορά όλο το σύνολο δεδομένων (Lazy Learners) όπου η κατηγοριοποίηση γίνεται με βάση παρόμοιες παρατηρήσεις στο εκάστοτε σύνολο δεδομένων και σε αυτούς που υπολογίζουν μία φορά το μοντέλο κατηγοριοποίησης πριν εισάγουμε νέα δεδομένα (Eager Learners).

Παράδειγμα της πρώτης κατηγορίας είναι ο αλγόριθμος των k-Πλησιέστερων Γειτόνων (k-Nearest Neighbors) και της δεύτερης οι αλγόριθμοι Naïve Bayes, τα Δένδρα απόφασης και τα Τεχνητά Νευρωνικά Δίκτυα.

Μερικά παραδείγματα προβλημάτων κατηγοριοποίησης αναφέρονται παρακάτω:

- Αναγνώριση παθήσεων όπως ο Διαβήτης, το Πάρκινσον και το Αλτζχάιμερ μέσω αριθμητικών δεδομένων και ο εντοπισμός παθήσεων σε ιατρικές εικόνες όπως η Πνευμονία, η νόσος του COVID-19 κ.λπ.
- Κατηγοριοποίηση ειδήσεων (news classification) ανάλογα με το περιεχόμενό τους όπως για παράδειγμα αθλητικές ειδήσεις, πολιτικές, υγείας, ιστορικά άρθρα, επιστημονικά άρθρα κ.λπ.
- Ανάλυση συναισθήματος (Sentiment analysis) κειμένου όπου αυτό μπορεί να είναι ένα σχόλιο στο Twitter ή μια κριτική για μια ταινία ή ένα προϊόν.
- Αναγνώριση χειρόγραφων γραμμάτων σε εικόνες και βίντεο (Handwriting detection).
- Ταυτοποίηση χρηστών με βάση εικόνες του προσώπου (Face Recognition).
- Αναγνώριση της μιας ανθρώπινης φυσικής δραστηριότητας με βάση δεδομένα από αισθητήρες όπως η αναγνώριση του περπατήματος και του τρεξίματος μέσω ενός smartwatch.

2.3 Εκπαίδευση και αξιολόγηση ενός κατηγοριοποιητή

Σε αυτή την ενότητα θα περιγράψουμε συνοπτικά τη διαδικασία της εκπαίδευσης ενός μοντέλου Μηχανικής Μάθησης για προβλήματα Κατηγοριοποίησης καθώς και τρόπους αξιολόγησης της απόδοσής του.

2.3.1 Σύνολα Εκπαίδευσης και Ελέγχου

Το βασικό στάδιο για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης είναι ο διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης (training set) και ελέγχου (testing set). Αυτό συνήθως γίνεται επιλέγοντας ένα ποσοστό του αρχικού συνόλου των δεδομένων ως σύνολο εκπαίδευσης και χρησιμοποιώντας το υπόλοιπο ως σύνολο ελέγχου. Όπως φαίνεται και από τις ονομασίες τους, το μοντέλο εκπαιδεύεται χρησιμοποιώντας το σύνολο εκπαίδευσης και εφαρμόζει τη γνώση που έμαθε στο σύνολο ελέγχου για να υπολογίσει την ακρίβειά του.

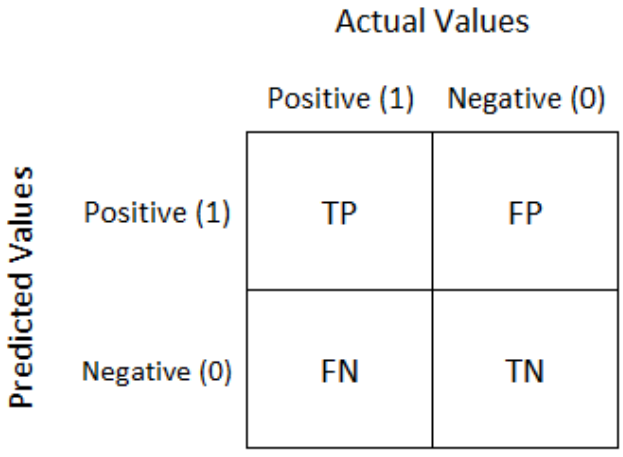
2.3.2 Επιλογή Χαρακτηριστικών

Ένας παράγοντας που επηρεάζει πολύ την ακρίβεια ενός μοντέλου μηχανικής μάθησης είναι η επιλογή των χαρακτηριστικών (feature selection), δηλαδή των μεταβλητών που θα χρησιμοποιήσουμε για να το εκπαιδεύσουμε. Άσχετα δεδομένα ή δεδομένα με μεγάλη συσχέτιση (correlation) μεταξύ τους, οδηγούν το μοντέλο σε λάθος συμπεράσματα και χαμηλά ποσοστά ακρίβειας. Η επιλογή των κατάλληλων χαρακτηριστικών μπορεί να επιτευχθεί με αυτόματες διαδικασίες ή και χειροκίνητα.

Υπάρχουν πολλά στατιστικά τεστ που μας βοηθούν να εντοπίσουμε τα χαρακτηριστικά με τη μεγαλύτερη κρισιμότητα και να επιλέξουμε μόνο αυτά και αντίστοιχα να απορρίψουμε χαρακτηριστικά μικρής σημασίας. Ένα γνωστό παράδειγμα αυτού του ζητήματος είναι γνωστό από τη Γραμμική Παλινδρόμηση. Αν χρησιμοποιήσουμε στο ίδιο μοντέλο μεταβλητές με μεγάλη συσχέτιση τότε προκύπτει το πρόβλημα της πολυσυγραμμικότητας, όπου το μοντέλο παρουσιάζει πολύ χαμηλή απόδοση παρόλο που όλες οι μεταβλητές, μεμονωμένα, θεωρούνται σημαντικές για το πρόβλημα.

2.3.3 Μετρικές αξιολόγησης

Για την αξιολόγηση ενός κατηγοριοποιητή εφαρμόζουμε το μοντέλο που υπολογίσαμε στο σύνολο εκπαίδευσης και καταγράφουμε το ποσοστό των παρατηρήσεων που κατηγοριοποιήθηκαν επιτυχώς και αυτών που κατηγοριοποιήθηκαν λανθασμένα. Έτσι κατασκευάζουμε τον Πίνακα Σύγχυσης (Confusion Matrix) που παρουσιάζει αυτά τα αποτελέσματα σε μορφή πίνακα και μέσω αυτού υπολογίζουμε τις μετρικές που θα μας βοηθήσουν στην αξιολόγηση του κατηγοριοποιητή. Παρακάτω απεικονίζεται ένας πίνακας σύγχυσης για την περίπτωση δυαδικής κατηγοριοποίησης μεταξύ των κλάσεων 1 και 0.



Εικόνα 2-6: Πίνακας Σύγχυσης [8]

TP (True Positive): Οι παρατηρήσεις της κλάσης 1 που κατηγοριοποιήθηκαν επιτυχώς.

FP (False Positive): Παρατηρήσεις που ενώ ανήκουν στην κλάση 0 κατηγοριοποιήθηκαν στην κλάση 1.

FN (False Negative): Αυτές που ενώ ανήκουν στην κλάση 0 κατηγοριοποιήθηκαν στην κλάση 1.

TN (True Negative): Οι παρατηρήσεις της κλάσης 0 που κατηγοριοποιήθηκαν επιτυχώς.

Με βάση τα παραπάνω χαρακτηριστικά υπολογίζουμε τα μέτρα αξιολόγησης τα οποία παρουσιάζουμε ενδεικτικά:

Ορθότητα (Accuracy):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Ευαισθησία (Sensitivity ή Recall):

$$Sensitivity = \frac{TP}{TP + FN}$$

Ακρίβεια (Precision):

$$Precision = \frac{TP}{TP + FP}$$

Εξειδίκευση (Specificity):

$$Specificity = \frac{TN}{TN + FP}$$

F-Measure:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Για την επιλογή του καλύτερου ταξινομητή βασιζόμαστε στις παραπάνω μετρήσεις και του μέσου όρου τους. Συνήθως παίρνουμε και τον σταθμισμένο μέσο όρο (weighted average) χρησιμοποιώντας ως βάρη το λόγο της πληθικότητας κάθε κλάσης με το συνολικό πλήθος παρατηρήσεων .

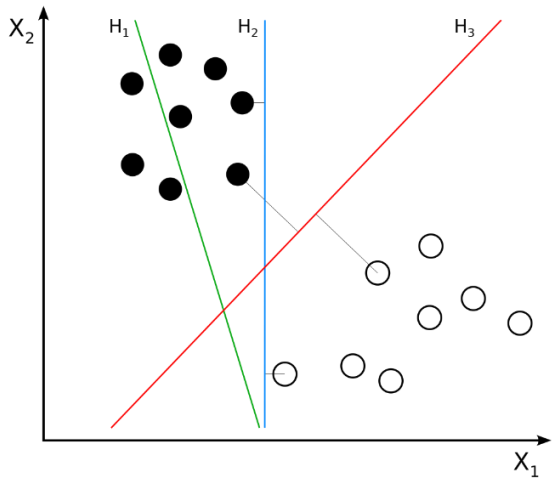
2.4 Μηχανές Διανυσμάτων Υποστήριξης

Οι *Μηχανές Διανυσμάτων Υποστήριξης (ή SVM)* είναι αλγόριθμοι που χρησιμοποιούνται για προβλήματα κατηγοριοποίησης, παλινδρόμησης αλλά και συσταδοποίησης. Στην παρούσα εργασία τις χρησιμοποιούμε για την κατηγοριοποίηση των στατικών χειρονομιών που έχουμε προκαθορίσει.

2.4.1 Γενική περιγραφή

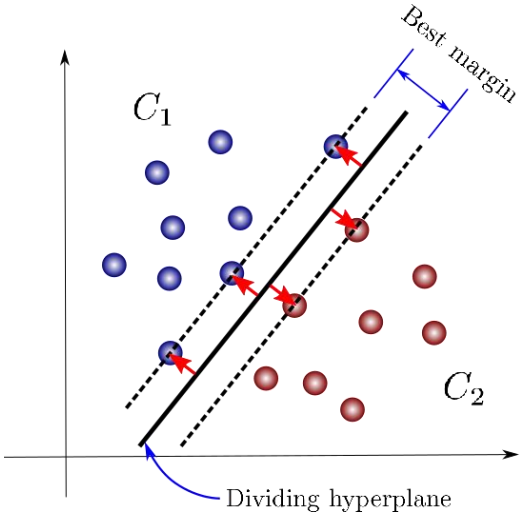
Ο στόχος ενός SVM είναι να υπολογίσει ένα *υπερεπίπεδο (hyperplane)* που διαχωρίζει τα δεδομένα. Έστω ένας χώρος Α με n διαστάσεις. Ένα υπερεπίπεδο είναι ένας υπόχωρος του Α με n-1 διαστάσεις.

Στην περίπτωση δισδιάστατων δεδομένων το υπερεπίπεδο αυτό είναι μια επίπεδη γραμμή ενώ στον τρισδιάστατο χώρο είναι ένα επίπεδο. Η εύρεση ενός τέτοιου υπερεπιπέδου έχει ένα σύνολο λύσεων και θα πρέπει να υπολογίσουμε τη βέλτιστη μεταξύ αυτών. Για παράδειγμα στα δισδιάστατα δεδομένα υπάρχουν πολλές ευθείες που να τα διαχωρίζουν αλλά δεν το κάνουν όλες με το βέλτιστο τρόπο, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2-7: Ευθείες που διαχωρίζουν τα δεδομένα. Η ευθεία H_1 κάνει λάθος διαχωρισμό, η H_2 κάνει μεν σωστό διαχωρισμό αλλά με μικρό περιθώριο και η H_3 κάνει βέλτιστο διαχωρισμό των κλάσεων. [9]

Για τον υπολογισμό του βέλτιστου υπερεπιπέδου πρώτα υπολογίζονται τα *διανύσματα υποστήριξης (support vectors)* τα οποία είναι τα σημεία που βρίσκονται πλησιέστερα στο υπερεπίπεδο και επηρεάζουν περισσότερο τον προσανατολισμό του. Στη συνέχεια αναπροσαρμόζουμε την ευθεία του υπερεπιπέδου με στόχο τη μεγιστοποίηση της απόστασης από τα διανύσματα υποστήριξης και έτσι προκύπτει το *υπερεπίπεδο μεγίστου περιθωρίου (maximum-margin hyperplane)*. Στην παρακάτω εικόνα απεικονίζεται το βέλτιστο υπερεπίπεδο (ευθεία) που διαχωρίζει δεδομένα σε δύο υποσύνολα καθώς και η επιλογή των διανυσμάτων υποστήριξης.



Εικόνα 2-8: Εύρεση βέλτιστου υπερεπιπέδου μέσω των διανυσμάτων στήριξης [10]

2.4.2 Γραμμικά SVM

Τα παραδείγματα των προηγούμενων εικόνων αναφέρονται σε δεδομένα τα οποία μπορούν να διαχωριστούν γραμμικά (linear separation) και απλά αναζητείται η ευθεία που τα διαχωρίζει με το βέλτιστο τρόπο και σε τέτοιες περιπτώσεις λέμε ότι χρησιμοποιούμε ένα Hard-margin διαχωριστή. Αυτού του είδους η κατηγοριοποίηση ονομάζεται Γραμμική (Linear Classification) καθώς τα δεδομένα είναι γραμμικώς διαχωρίσιμα. Ο μαθηματικός φορμαλισμός για τέτοιου είδους προβλήματα αποτυπώνεται στους παρακάτω τύπους:

Έστω $(x_1, y_1), \dots, (x_n, y_n)$ το σύνολο των n παρατηρήσεων που πρέπει να διαχωρίσουμε σε δύο κλάσεις, έστω 1 και -1. Η μεταβλητή y_i μας δείχνει σε ποια κλάση ανήκει η i-οστή παρατήρηση x_i που είναι ένα πολυδιάστατο διάνυσμα. Γενικά ένα υπερεπίπεδο ικανοποιεί την εξίσωση:

$$\boldsymbol{w^T x - b = 0}$$

Όπου με w συμβολίζουμε το κανονικό διάνυσμα ως προς την ευθεία και η παράμετρος $\frac{b}{\|b\|}$ καθορίζει τη μετατόπιση της ευθείας κατά τη διεύθυνση του w.

Για να επιλέξουμε το βέλτιστο υπερεπίπεδο μεταξύ γραμμικώς διαχωρίσιμων δεδομένων πρέπει να επιλύσουμε το ακόλουθο πρόβλημα βελτιστοποίησης:

Ελαχιστοποίηση του $\|w\|$ ως προς την ανίσωση:

$$\boldsymbol{y_i(w^T - b) \geq 1, \forall 1 \leq i \leq n}$$

Έτσι υπολογίζουμε τα w και τα b που καθορίζουν επακριβώς το υπερεπίπεδο που ψάχνουμε.

Στην περίπτωση που τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα έχουμε το λεγόμενο Soft-Margin πρόβλημα. Σε αυτή την περίπτωση χρησιμοποιείται η συνάρτηση κόστους Hinge η οποία περιγράφεται με τον παρακάτω τύπο:

$$\boldsymbol{max}\Big(\mathbf{0}, \mathbf{1} - \boldsymbol{y_i}(\boldsymbol{w^T} - \boldsymbol{b})\Big)$$

Εδώ το πρόβλημα βελτιστοποίησης είναι η ελαχιστοποίηση της ακόλουθης ποσότητας:

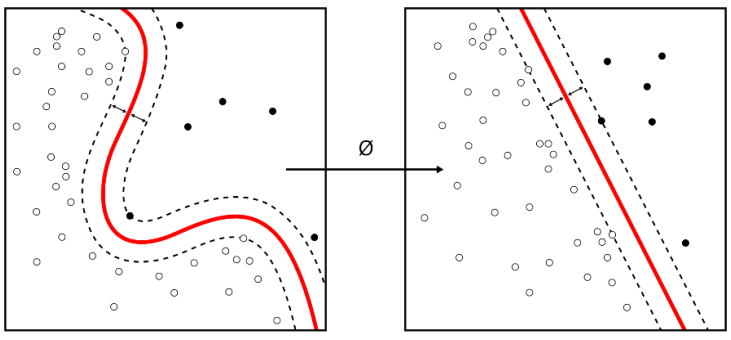
$$\left[\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{max}\left(\mathbf{0}, \mathbf{1} - \boldsymbol{y_i}(\boldsymbol{w^T} - \boldsymbol{b})\right)\right] + \boldsymbol{\lambda} \cdot \|\boldsymbol{w}\|^{\mathbf{2}}$$

Όπου η σταθερά λ καθορίζεται σύμφωνα με τις ανάγκες του εκάστοτε προβλήματος.

2.4.3 Μη-Γραμμικά SVM και το τέχνασμα του πυρήνα

Στις μαθηματικές σχέσεις που είδαμε έως τώρα οι πολλαπλασιασμοί μεταξύ των διανυσμάτων αναφέρονται στο εσωτερικό τους γινόμενο το οποίο είναι μια γραμμική συνάρτηση. Αντί για το εσωτερικό γινόμενο συχνά χρησιμοποιούμε άλλες, μη-γραμμικές συναρτήσεις, οι οποίες μετασχηματίζουν το χώρο που ορίζονται τα δεδομένα. Η ιδέα είναι ότι τα δεδομένα μπορεί να μην είναι γραμμικώς διαχωρίσιμα στην αρχική τους μορφή, αλλά μπορούν να μετατραπούν σε διαχωρίσιμα, μετασχηματίζοντας το χώρο που ορίζονται.

Η συνάρτηση που χρησιμοποιούμε στα SVM αντί του εσωτερικού γινομένου ονομάζεται *συνάρτηση πυρήνα (kernel function)* και η διαδικασία επιλογής μιας μη γραμμικής συνάρτησης ως πυρήνα ονομάστηκε *τέχνασμα του πυρήνα (kernel trick)* [11].



Εικόνα 2-9: Αλλαγή πυρήνα σε ένα SVM για να μετατραπούν τα δεδομένα σε γραμμικώς διαχωρίσιμα [11]

Κάποιες γνωστές συναρτήσεις πυρήνα είναι οι εξής:

Γραμμικός Πυρήνας (Linear) : Χρησιμοποιείται για δεδομένα που είναι ήδη γραμμικώς διαχωρίσιμα.

$$K(x_i, x_j) = x_i^T x_j$$

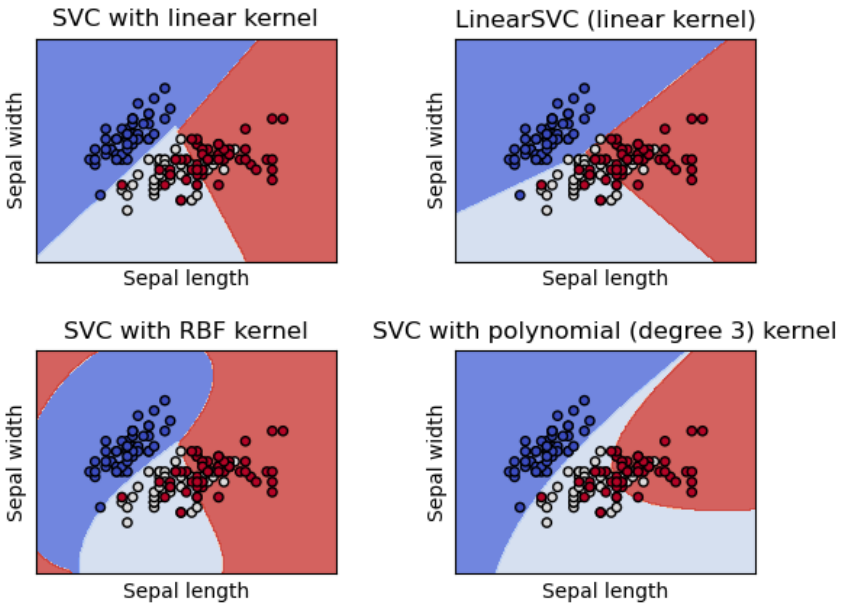
Πολυωνυμικός (Polynomial) : Ένα πολυώνυμο βαθμού d που επιλέγει ο χρήστης.

$$K(x_i, x_j) = (x_i^T x_j + 1)^d$$

Πυρήνας Ακτινικής Βάσης (Radial Basis Function): Η θετική παράμετρος γ επιλέγεται από το χρήστη.

$$K(x_i, x_j) = \exp\left\{-\gamma \|x_i - x_j\|^2\right\}, \gamma > 0$$

Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα κατηγοριοποίησης σε τρείς κλάσεις, χρησιμοποιώντας διαφορετικούς πυρήνες.



Εικόνα 2-10: Κατηγοριοποίηση με SVM του συνόλου δεδομένων Iris χρησιμοποιώντας διαφορετικούς πυρήνες [25]

ΚΕΦΑΛΑΙΟ 3

ΒΑΘΙΑ ΜΑΘΗΣΗ

Με τον όρο *Βαθιά Μάθηση (Deep Learning)* αναφερόμαστε στον τομέα της Μηχανικής Μάθησης που χρησιμοποιεί τεχνητά νευρωνικά δίκτυα για να εκπαιδεύσει ένα υπολογιστικό σύστημα να επιλύει προβλήματα.

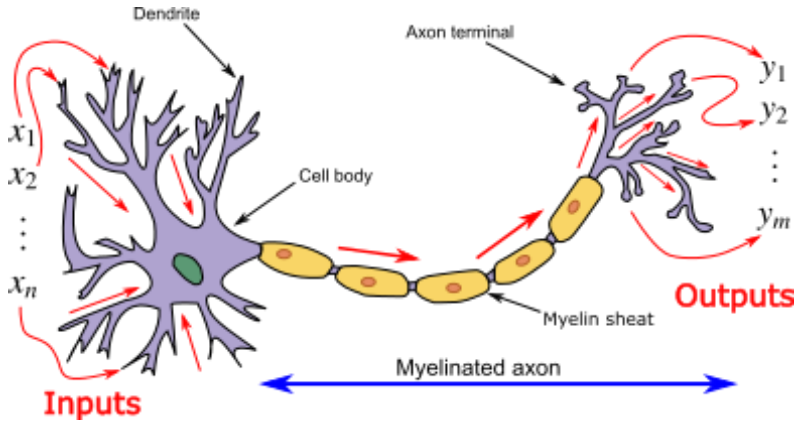
Όπως θα δούμε παρακάτω, ένα νευρωνικό δίκτυο μπορεί να αποτελείται από πολλά στρώματα (layers) καθένα από τα οποία αποτελείται από ένα πλήθος νευρώνων. Ο όρος 'Βαθιά' στη Βαθιά Μάθηση προκύπτει απ' τη χρήση βαθιών νευρωνικών δικτύων, δηλαδή δικτύων με πολλά ενδιάμεσα στρώματα. Προφανώς, όσο αυξάνουμε το πλήθος των ενδιάμεσων στρωμάτων και το πλήθος των νευρώνων, η πολυπλοκότητα του μοντέλου αυξάνεται σημαντικά και έχει μεγαλύτερο υπολογιστικό κόστος. Γι'αυτό και αρχικά, ήταν δύσκολη η πειραματική αξιολόγηση σύνθετων αρχιτεκτονικών. Με την εξέλιξη της τεχνολογίας, η αύξηση της διαθέσιμης υπολογιστικής ισχύος σε συνδυασμό με την αύξηση της διαθεσιμότητας μεγάλων συνόλων δεδομένων, έχει καταστήσει διαχειρίσιμη την εκπαίδευση και αξιολόγηση εξαιρετικά περίπλοκων μοντέλων και σύνθετων αρχιτεκτονικών, με τεράστιο πλήθος παραμέτρων. Ο κλάδος της Βαθιάς Μάθησης έχουν σημειώσει εντυπωσιακές δυνατότητες επίλυσης όλων των προβλημάτων της Μηχανικής Μάθησης και αποτελεί το ισχυρότερο εργαλείο της Τεχνητής Νοημοσύνης για την εκπαίδευση των μηχανών να επιλύουν προβλήματα.

3.1 Τεχνητά Νευρωνικά Δίκτυα

Τα *Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)* ή απλώς *Νευρωνικά Δίκτυα* είναι υπολογιστικά μοντέλα που χρησιμοποιούνται για την επίλυση όλων των ειδών των προβλημάτων και σαν ιδέα είναι επηρεασμένα από τους βιολογικούς νευρώνες του εγκεφάλου. Στους ζώντες οργανισμούς, το νευρικό σύστημα είναι υπεύθυνο για πληθώρα διεργασιών όπως η μάθηση, η μνήμη κ.λπ. και αποτελείται από ένα μεγάλο πλήθος νευρώνων οι οποίοι επεξεργάζονται τα ηλεκτρικά σήματα που δέχονται και συνολικά αποτελούν ένα μεγάλο δίκτυο με σύνθετες δομές το οποίο συνεχώς αναδιαρθρώνεται και επεκτείνεται.

Αντίστοιχα με τα βιολογικά νευρωνικά δίκτυα, τα ΤΝΔ είναι συνδεδεμένα δίκτυα από νευρώνες (neurons), καθένας από τους οποίους αντιστοιχεί σε έναν υπολογιστικό κόμβο ο οποίος δέχεται σαν είσοδο αριθμητικά δεδομένα, εκτελεί μαθηματικούς υπολογισμούς και παράγει μια έξοδο σαν αποτέλεσμα, η οποία είτε τροφοδοτεί άλλους κόμβους ή είναι η τελική έξοδος του δικτύου.

Η βασική διαφορά με τα βιολογικά δίκτυα είναι ότι στα ΤΝΔ η δομή του δικτύου είναι προκαθορισμένη και παραμένει ίδια κατά τη διάρκεια της εκπαίδευσης, αντίθετα, στους βιολογικούς οργανισμούς τα δίκτυα μπορούν να αλλάζουν δυναμικά.



Εικόνα 3-1: Βιολογικό Νευρωνικό Δίκτυο [12]

Για την καλύτερη κατανόηση της λειτουργίας ενός τεχνητού νευρώνα ας αναλογιστούμε το εξής παράδειγμα. Έστω ότι ένας γιατρός θέλει να αποφασίσει για το εάν ένας ασθενής πάσχει από μια συγκεκριμένη ασθένεια, έχοντας στη διάθεσή του τα αποτελέσματα κάποιων εξετάσεων και τα γενικά χαρακτηριστικά του ασθενή όπως η ηλικία, το βάρος, το ύψος του κ.λπ. Για τη διάγνωση, θα πρέπει να δώσει μια βαρύτητα στις παραπάνω μεταβλητές και μετά από κάποιες μαθηματικές διαδικασίες θα καταλήξει σε ένα αποτέλεσμα. Ανάλογα αν αυτό ξεπερνάει ή όχι κάποιες τιμές, ο γιατρός θα αποφασίσει αν ο ασθενής πάσχει ή όχι από τη συγκεκριμένη ασθένεια. Ένας τεχνητός νευρώνας θα ακολουθήσει παρόμοια διαδικασία σταθμίζοντας τις μεταβλητές εισόδου και μαθαίνοντας τα κριτήρια που πρέπει να χρησιμοποιήσει ώστε να αποφανθεί για το αν οι μεταβλητές αυτές αντιστοιχούν σε μια πάθηση η όχι. Η στάθμιση των μεταβλητών γίνεται πολλαπλασιάζοντας τις μεταβλητές του προβλήματος με σταθερές και ο βασικός στόχος ενός νευρωνικού δικτύου είναι η κατάλληλη επιλογή αυτών.

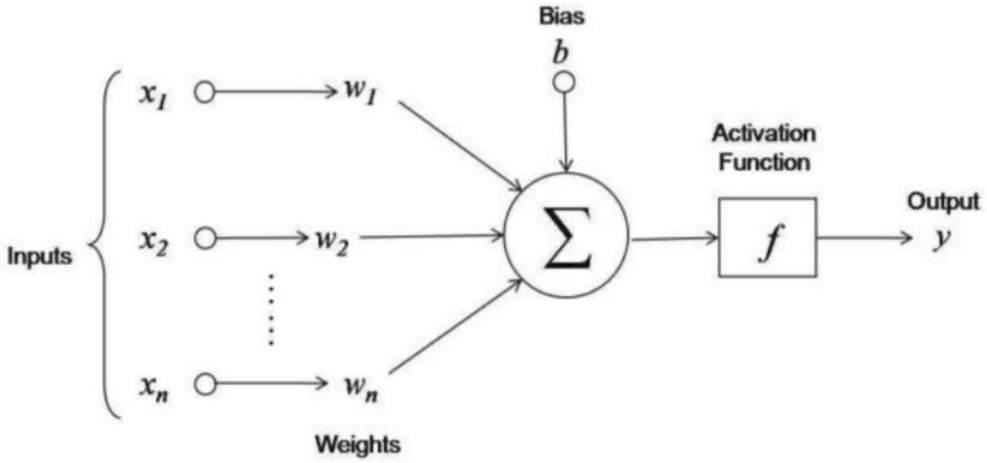
Τα ΤΝΔ πρωτοεμφανίστηκαν το 1943 όταν οι McCulloch και Pitts δημιούργησαν το πρώτο υπολογιστικό μοντέλο για νευρωνικά δίκτυα. Τις επόμενες δεκαετίες πολλοί επιστήμονες επέκτειναν τις δυνατότητες των ΤΝΔ χρησιμοποιώντας μαθηματικές μεθόδους βελτιστοποίησης και εργαλεία από τον δυναμικό προγραμματισμό. Στις αρχές του 21^{ου} αιώνα με τη ραγδαία αύξηση της διαθέσιμης υπολογιστικής ισχύος των σύγχρονων επεξεργαστικών μονάδων και των τεχνικών παράλληλης επεξεργασίας δεδομένων τα ΤΝΔ βρήκαν σημαντική άνθηση και σημείωσαν πρωτοφανείς επιδόσεις σε δύσκολα προβλήματα όπως αυτά της οπτικής αναγνώρισης.

Οι νευρώνες στα ΤΝΔ μπορεί να είναι νευρώνες εισόδου οι οποίοι τροφοδοτούν τους υπολογιστικούς νευρώνες με τα δεδομένα του προβλήματος (inputs), νευρώνες εξόδου οι οποίοι εξάγουν το τελικό αποτέλεσμα της όλης διαδικασίας (output) και υπολογιστικοί (ή κρυφοί) νευρώνες (hidden) οι οποίοι εκτελούν κάποιες μαθηματικές πράξεις στο ενδιάμεσο στάδιο. Συγκεκριμένα, πολλαπλασιάζουν τα δεδομένα εισόδου με έναν συντελεστή που ονομάζεται βάρος (weight), προσθέτουν μια σταθερά που ονομάζεται πόλωση ή προκατάληψη (bias) και υπολογίζουν το συνολικό άθροισμά τους, στο οποίο εφαρμόζεται μια συνάρτηση ενεργοποίησης (activation function) και η τιμή της οποίας είναι το αποτέλεσμα εξόδου του νευρώνα.

Η μαθηματική περιγραφή της λειτουργίας ενός τεχνητού νευρώνα με n συνδέσεις εισόδου περιγράφεται με την παρακάτω σχέση:

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

Όπου με x_i συμβολίζουμε τα n σήματα εισόδου, w_i τα βάρη των συνάψεων, b την πόλωση και με $f(\cdot)$ τη συνάρτηση ενεργοποίησης του νευρώνα.



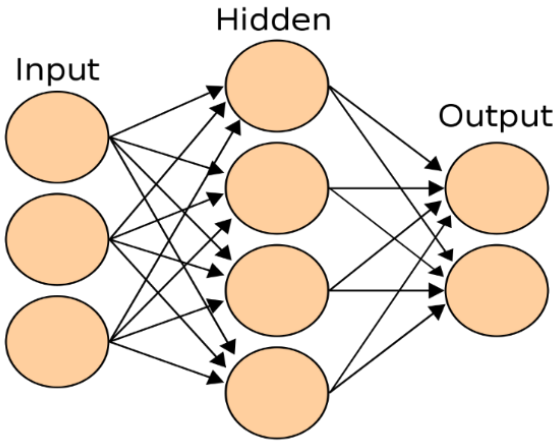
Εικόνα 3-2: Η λειτουργία ενός τεχνητού νευρώνα [13]

3.2 Αρχιτεκτονικές Νευρωνικών Δικτύων

Παρακάτω θα περιγράψουμε συνοπτικά κάποιες βασικές αρχιτεκτονικές νευρωνικών δικτύων.

3.2.1 Νευρωνικά Δίκτυα Εμπρόσθιας Τροφοδότησης

Στα *Νευρωνικά Δίκτυα Εμπρόσθιας Τροφοδότησης (Feedforward Neural Networks)* οι συνδέσεις μεταξύ των νευρώνων ακολουθούν μια συγκεκριμένη ροή και δεν σχηματίζουν κύκλους. Δηλαδή κάθε νευρώνας θα στείλει πληροφορία στο επόμενο στη σειρά στρώμα νευρώνων αλλά ποτέ στο προηγούμενο. Το πιο απλό παράδειγμα feedforward μοντέλου είναι αυτό του single-layer perceptron, που αποτελείται από ένα μόνο στρώμα με νευρώνες εξόδου, οι οποίοι τροφοδοτούνται με τα δεδομένα ως έχουν και υπολογίζουν τις μαθηματικές πράξεις που είδαμε στην περιγραφή του νευρώνα. Στην περίπτωση του *Πολυστρωματικού Perceptron (Multi-layer Perceptron ή MLP),* έχουμε πλήθος από στρώματα τα οποία είναι πλήρως συνδεδεμένα μεταξύ τους (fully connected nodes)



Εικόνα 3-3: Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης (Feedforward) [14]

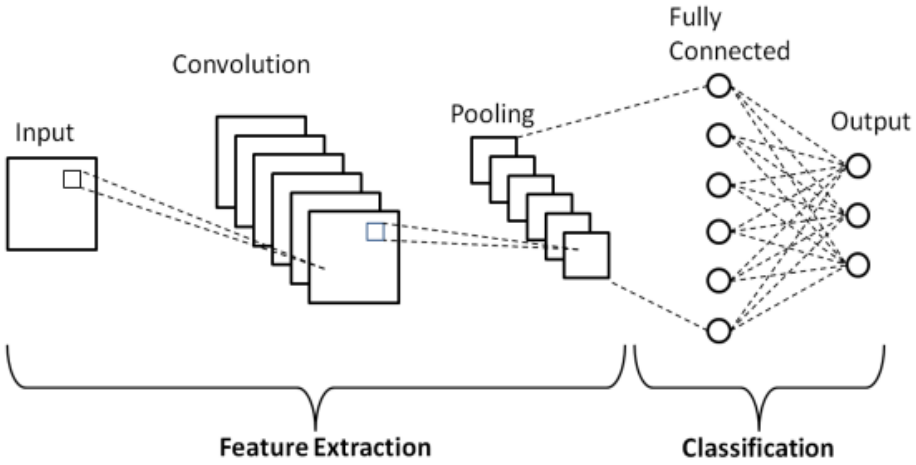
3.2.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα *Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks ή CNN)* γενικά έχουν τοπολογία παρόμοια με τα MLP με την έννοια ότι υπάρχει ιεραρχία στη μετάδοση της πληροφορίας μεταξύ των επιπέδων αλλά χρησιμοποιούν και επίπεδα που υπολογίζουν μαθηματικές συνελίξεις (convolutions). Τα Συνελικτικά στρώματα αποτελούνται από φίλτρα με τα οποία συνελίσσεται η τιμή εισόδου και χρησιμοποιούν και αυτά τα δικά τους βάρη όπως και οι νευρώνες του δικτύου. Τα φίλτρα παράγουν έναν δισδιάστατο χάρτη ενεργοποίησης (activation map) που πρακτικά σημαίνει ότι το μοντέλο μαθαίνει να αναγνωρίζει συγκεκριμένα μοτίβα τα οποία αποτελούν σημαντικά χαρακτηριστικά του προβλήματος.

Επίσης, χρησιμοποιούνται σε συνδυασμό με στρώματα συσσώρευσης (pooling layers) με στόχο τη βελτίωση της απόδοσης του μοντέλου. Ένα στρώμα συσσώρευσης δέχεται την είσοδο του προηγούμενου στρώματος συνέλιξης και εφαρμόζει έναν τελεστή συσσώρευσης (pooling operator). Συνήθως αυτός ο τελεστής είναι η συνάρτηση του μεγίστου (max-pooling), δηλαδή δέχεται ένα σύνολο από τιμές και επιστρέφει τη μεγαλύτερη. Το τελικό στρώμα ενός CNN, αποτελείται από ένα πλήρως συνδεδεμένο δίκτυο.

Τα CNN εφαρμόζονται κυρίως σε προβλήματα υπολογιστικής όρασης, όπως αυτά της ανάλυσης εικόνων και βίντεο, της αναγνώρισης ανθρώπινης φωνής και γενικά σε προβλήματα όπου οι τιμές εισόδου είναι πολυδιάστατες. Για παράδειγμα, μια ψηφιακή εικόνα αναπαρίσταται από ένα δισδιάστατο πίνακα όπου κάθε στοιχείο αντιστοιχεί σε ένα pixel, καθένα από τα οποία μπορεί να αποτελείται από επιμέρους διαστάσεις όπως στο μοντέλο RGB (Red Green Blue) όπου κάθε pixel αποτελεί ένα τρισδιάστατο διάνυσμα που περιέχει τις τιμές της έντασης του κόκκινου, πράσινου και μπλε χρώματος. Αν χρησιμοποιούσαμε MLP δίκτυα για την ανάλυση εικόνων θα έπρεπε να μετατρέψουμε σε μονοδιάστατο διάνυσμα τα δεδομένα. Για μια εικόνα μεγέθους 1920 x 1080 (Full High Definition), το διάνυσμα αυτό θα είχε 2.073.600 στοιχεία και θα απαιτούσε υψηλό υπολογιστικό κόστος. Τα CNN μειώνουν σημαντικά τις διαστάσεις των δεδομένων με τη χρήση των φίλτρων και μαθαίνουν να αναγνωρίζουν πολύ καλύτερα τις επιμέρους δομές που υπάρχουν σε μια εικόνα, αποκτούν δηλαδή καλύτερη «αντίληψη». Τέτοιου είδους δομές μπορεί να είναι οι ακμές, το χρώμα και άλλα χαρακτηριστικά που περιγράφουν το αντικείμενο που απεικονίζεται.

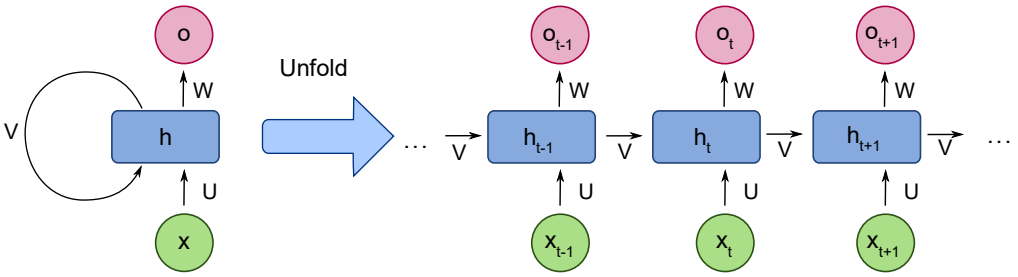
Στην παρακάτω εικόνα περιγράφεται ένα Συνελικτικό Νευρωνικό Δίκτυο που αποτελείται από συνδυασμό συνελικτικών στρωμάτων και στρωμάτων συσσώρευσης.



Εικόνα 3-4: Συνελικτικό Νευρωνικό Δίκτυο [15]

3.2.3 Αναδρομικά Νευρωνικά Δίκτυα

Στα *Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks ή RNN),* ο μετασχηματισμός των δεδομένων εισόδου εξαρτάται και από τις προηγούμενες εισόδους που έχουν ήδη επεξεργαστεί. Διαισθητικά αυτό σημαίνει ότι το δίκτυο αποκτά την ικανότητα να παίρνει μια απόφαση για το παρών λαμβάνοντας υπόψιν και το παρελθόν. Είναι ιδιαίτερα αποτελεσματικά σε προβλήματα με ακολουθιακά δεδομένα (sequential data) όπως είναι η γραφή, η ανθρώπινη φωνή, μια ακολουθία εικόνων (βίντεο) κ.λπ.



Εικόνα 3-5: Αναδρομικό Νευρωνικό Δίκτυο [16]

Γενικά, τα RNN δεν αναγνωρίζουν αποτελεσματικά εξαρτήσεις μακράς διάρκειας σε ακολουθιακά δεδομένα εξαιτίας του προβλήματος της εξαφάνισης της κλίσης (vanishing gradient problem). Ο αλγόριθμος Back Propagation υπολογίζει τις παραγώγους της συνάρτησης του κόστους η οποία όμως εξαρτάται και από παρελθοντικά σημεία και αυτό γίνεται επαναληπτικά. Έτσι καταλήγουμε σε ένα γινόμενο πολλών παραγόντων οι οποίοι μπορούν σχεδόν να μηδενίζουν την τιμή του αποτελέσματος και να οδηγούν στην αποτυχία του μοντέλου. Το ζήτημα αυτό αντιμετωπίστηκε με τα αναδρομικά δίκτυα *Μακράς Βραχείας Μνήμης (Long Term Short Term Memory ή LSTM)* και τα *Gated Recurrent Units (GRUs)* που είναι μια απλοποιημένη παραλλαγή των LSTM.

3.3 Εκπαίδευση ενός τεχνητού νευρωνικού δικτύου

Η εκπαίδευση ενός ΤΝΔ μοιάζει με τον τρόπο που μαθαίνει ο άνθρωπος να επιτελεί μια διεργασία. Εκτελεί μια ενέργεια η οποία έχει κάποιο αποτέλεσμα και ανάλογα με το αν το αποτέλεσμα είναι ωφέλιμο ή όχι, προσπαθεί να βελτιώσει τον τρόπο με τον οποίο την εκτέλεσε. Έτσι και σε ένα νευρωνικό δίκτυο, η εκπαίδευση γίνεται επαναληπτικά μέσω δοκιμών όπου σε κάθε επανάληψη κάνουμε διορθώσεις στο μοντέλο.

3.3.1 Η συνάρτηση κόστους

Η *συνάρτηση κόστους* ή λάθους (loss function) κατά τη διάρκεια της εκπαίδευσης θα είναι η ένδειξη για το μέγεθος της απόκλισης των αποτελεσμάτων από τις πραγματικές τιμές. Δηλαδή ένα μέτρο εκτίμησης του λάθους ώστε να μπορέσει να γίνει σωστή αναπροσαρμογή των παραμέτρων του μοντέλου με στόχο την ελαχιστοποίησή του. Συνεπώς το πρόβλημα πλέον είναι η ελαχιστοποίηση της συνάρτησης κόστους, ένα πρόβλημα μαθηματικής βελτιστοποίησης.

Στη Στατιστική, οι συναρτήσεις κόστους χρησιμοποιούνται για την εκτίμηση των παραμέτρων μοντέλων όπως στη γραμμική παλινδρόμηση που χρησιμοποιείται το μέσο τετραγωνικό σφάλμα. Με τη βελτιστοποίηση της συνάρτησης, υπολογίζουμε τις τιμές των βαρών που θα πρέπει να χρησιμοποιήσουμε στο μοντέλο για να πετύχουμε τη βέλτιστη ακρίβεια.

Ενδεικτικά παραδείγματα συναρτήσεων κόστους που χρησιμοποιούμε στην πράξη είναι:

Το *Μέσο Τετραγωνικό Σφάλμα* (Mean Squared Error ή MSE). Χρησιμοποιείται στα προβλήματα της Παλινδρόμησης όπου γίνεται πρόβλεψη πραγματικών αριθμών. Η μεταβλητή y αντιστοιχεί στις τιμές που παράγει το μοντέλο και με $\hat{y}$ συμβολίζουμε τις πραγματικές τιμές των δεδομένων.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Η Δ*ιωνυμική Διασταυρωμένη Εντροπία* (Binary Cross-Entropy ή BCE). Την εφαρμόζουμε στα προβλήματα διωνυμικής κατηγοριοποίησης.

$$BCE = -\sum_{i=1}^{2} y_i \cdot log\big(P(y_i)\big)$$

Για προβλήματα κατηγοριοποίησης πολλαπλών κλάσεων χρησιμοποιούμε την *Κατηγορική Διασταυρωμένη Εντροπία* (Categorical Cross-Entropy ή CCE)

$$CCE = -\sum_{i=1}^{n} y_i \cdot log\big(P(y_i)\big)$$

3.3.2 Αλγόριθμοι Βελτιστοποίησης

Ο πιο γνωστός αλγόριθμος βελτιστοποίησης είναι η *Κατάβαση Δυναμικού (Gradient Descent)* που χρησιμοποιείται για τον εντοπισμό τοπικών ελαχίστων σε μια διαφορίσιμη συνάρτηση. Η ιδέα είναι ότι εφόσον η συνάρτηση είναι διαφορίσιμη σε μια περιοχή γύρω από ένα σημείο, μπορούμε σε κάθε επανάληψη να υπολογίζουμε την παράγωγό της, δηλαδή την κλίση, και να μετακινούμαστε προς την αντίθετη κατεύθυνση. Πρακτικά αυτό σημαίνει ότι σε κάθε επανάληψη, η ανανέωση της τιμής ενός βάρους που υπολογίζει ένας νευρώνας θα προκύπτει αφαιρώντας από αυτήν, την παράγωγό της πολλαπλασιασμένη με μια σταθερά που ονομάζεται *ρυθμός εκμάθησης (learning rate).*

Η μαθηματική σχέση για την ανανέωση των βαρών του νευρωνικού δικτύου με μέθοδο gradient descent είναι η ακόλουθη:

$$\boldsymbol{w_i = w_i - \eta \cdot \frac{\partial L}{\partial w_i}}$$

Όπου με το γράμμα η αναφερόμαστε στο ρυθμό εκμάθησης και $L(\cdot)$ είναι η συνάρτηση κόστους που θέλουμε να ελαχιστοποιήσουμε.

Όταν οι συναρτήσεις κόστους έχουν ορισμένες ιδιότητες, όπως το να είναι κυρτές ή η παράγωγός τους να είναι Lipschitz, τότε με προσεκτική επιλογή του ρυθμού εκμάθησης, μπορούμε να εγγυηθούμε τη σύγκλιση της μεθόδου σε τοπικό ελάχιστο ενώ για κυρτές συναρτήσεις, το τοπικό ελάχιστο ταυτίζεται με το ολικό οπότε προκύπτει αυτόματα η βέλτιστη λύση.

Η μέθοδος Gradient Descent μπορεί να χρειαστεί αρκετό χρόνο για να συγκλίνει όπως όταν βρεθεί στην περιοχή ενός σαγματικού σημείου, δηλαδή σε σημείο που μηδενίζεται μεν η παράγωγος αλλά δεν αποτελεί ούτε ελάχιστο ούτε μέγιστο. Μια παραλλαγή της μεθόδου που συγκλίνει ταχύτερα είναι η *Στοχαστική Κατάβαση Δυναμικού* (*Stochastic Gradient Descent ή sgd),* η οποία χρησιμοποιεί ένα υποσύνολο των δεδομένων για τον υπολογισμό των παραγώγων. Η πληθικότητα του υποσυνόλου αυτού ονομάζεται batch size και συνήθως είναι ένα ποσοστό επί της εκατό του συνόλου εκπαίδευσης.

Στη μέθοδο gradient descent ο ρυθμός εκμάθησης είναι ο ίδιος για όλες τις παραμέτρους και αποτελεί μια ακόμη υπερπαράμετρο του δικτύου που θα πρέπει να βελτιστοποιήσουμε. Ένας επίσης πολύ γνωστός αλγόριθμος τύπου gradient descent που όμως χρησιμοποιεί διαφορετικούς ρυθμούς εκμάθησης είναι ο αλγόριθμος AdaGrad [22]. Χρησιμοποιείται ως μια πιο ισχυρή έκδοση του αλγορίθμου sgd. Ο AdaGrad υπολογίζει ένα ρυθμό εκμάθησης για κάθε διάσταση των δεδομένων και δίνει μεγαλύτερη τιμή στις μεταβλητές που εμφανίζονται αραιά και μικρή τιμή στις συχνές μεταβλητές. Γι’αυτό και είναι πολύ χρήσιμος στα προβλήματα που σχετίζονται με αραιά διανύσματα όπως στην ανάλυση φυσικής γλώσσας.

Άλλοι γνωστοί αλγόριθμοι είναι οι Adam, RMSprop, AdaDelta, AdaMax, Nadam και AMSGrad.

Η επιλογή του αλγόριθμου που θα χρησιμοποιήσουμε είναι κι αυτή ένας παράγοντας που πρέπει να επιλέξουμε μέσω δοκιμών. Ανάλογα με τη φύση του προβλήματος και του είδους των δεδομένων που αναλύουμε, ίσως μπορούμε να έχουμε μια ένδειξη για το ποιος αλγόριθμος θα ήταν μια καλή επιλογή, όπως για παράδειγμα όταν έχουμε αραιά δεδομένα (sparse data).

3.3.3 Ο αλγόριθμος Οπισθοδιάδοσης

Όπως είδαμε για να λύσουμε προβλήματα βελτιστοποίησης απαιτείται ο υπολογισμός παραγώγων. Ο αλγόριθμος *Οπισθοδιάδοσης (Backpropagation)* είναι η μέθοδος που χρησιμοποιούμε για το σκοπό αυτό. Υπενθυμίζουμε ότι η παράγωγος της συνάρτησης κόστους μετράει τη μεταβολή της συνάρτησης, για μικρές μεταβολές στην τιμή της εισόδου. Καθώς τα δεδομένα είναι γενικά διανύσματα, η παράγωγος αναφέρεται στο σύνολο των μερικών παραγώγων για κάθε διάνυσμα εισόδου.

Έστω η συνάρτηση κόστους L και $w_{jk}^{(l)}$ το βάρος του j-οστού σημείου στο l -στρώμα. Κάθε νευρώνας υπολογίζει την ποσότητα

$$z_j^{(l)} = \sum_{k=1}^{m} w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}$$

όπου m το πλήθος των νευρώνων στο $l-1$ στρώμα. Έπειτα εφαρμόζουμε σε αυτή την ποσότητα τη συνάρτηση ενεργοποίησης.

Για τον υπολογισμό της, ως προς τα βάρη, χρησιμοποιούμε τον κανόνα της αλυσίδας (chain rule):

$$\frac{\partial L}{\partial w_{jk}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial w_{jk}^{(l)}}$$

Και έτσι προκύπτουν οι εξισώσεις:

$$\frac{\partial z_j^{(l)}}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \quad \text{και} \quad \frac{\partial L}{\partial w_{jk}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} a_k^{(l-1)}$$

Ο όρος $\delta_j^{(l)} = \frac{\partial L}{\partial z_j^{(l)}}$ είναι η τοπική κλίση (local gradient) και αναφέρεται στο διάνυσμα των σφαλμάτων που σχετίζονται με το l – στρώμα.

Αντίστοιχα για την πόλωση (bias) $b_j^{(l)}$ ο κανόνας της αλυσίδας δίνει:

$$\frac{\partial L}{\partial b_{jk}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial b_{jk}^{(l)}}$$

και μέσω παραγώγισης προκύπτουν οι σχέσεις:

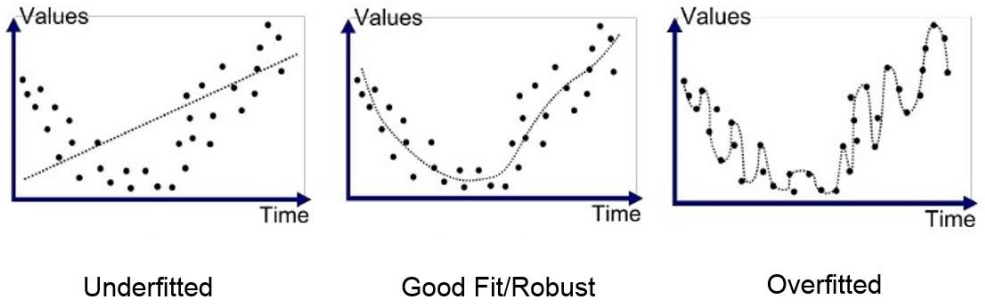
$$\frac{\partial z_j^{(l)}}{\partial b_{jk}^{(l)}} = 1 \text{ και } \frac{\partial L}{\partial b_{jk}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}}$$

Η τελική κλίση της συνάρτησης κόστους που υπολογίζει ο αλγόριθμος back propagation υπολογίζεται από τους τύπους:

$$\frac{\partial L}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \delta_j^{(l)} \text{ και } \frac{\partial L}{\partial b_{jk}^{(l)}} = \delta_j^{(l)}$$

3.3.4 Το πρόβλημα της Υπερπροσαρμογής

Το πρόβλημα της *Υπερπροσαρμογής* ή *Overfitting* προκύπτει όταν το μοντέλο αντί να γενικεύει τη γνώση που μαθαίνει, την υπερ-προσαρμόζει στα εκάστοτε δεδομένα με τα οποία εκπαιδεύεται. Αυτό έχει ως αποτέλεσμα υψηλό ποσοστό ακρίβειας στο σύνολο εκπαίδευσης και χαμηλό στο σύνολο ελέγχου. Υπάρχει και το αντίθετο πρόβλημα αυτό της *Υποπροσαρμογής (Underfitting)* όπου το μοντέλο δεν μπορεί να αναγνωρίσει επιτυχώς τη δομή των δεδομένων και έχει χαμηλή ακρίβεια, όπως στις περιπτώσεις που χρησιμοποιούμε γραμμικά μοντέλα σε μη-γραμμικά δεδομένα.

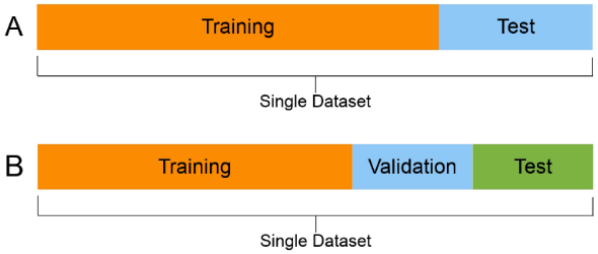


Εικόνα 3-6: Παραδείγματα Υποπροσαρμογής, Καλής Προσαρμογής και Υπερπροσαρμογής στα δεδομένα [17]

Για να αποφύγουμε τέτοιου είδους προβλήματα χρησιμοποιούμε διάφορες τεχνικές όπως τη χρήση συνόλου επικύρωσης (validation set) και την απόσυρση ενός ποσοστού των νευρώνων μεταξύ των στρωμάτων του δικτύου, όπως θα δούμε στη συνέχεια.

Το σύνολο Επικύρωσης

Συχνά για να αποφύγουμε το πρόβλημα του Overfit χρησιμοποιούμε ένα υποσύνολο των δεδομένων ως σύνολο επικύρωσης (validation set), πριν χωρίσουμε αρχικώς τα δεδομένα σε σύνολο εκπαίδευσης (training set) και σύνολο ελέγχου (test set). Κατά τη διάρκεια της εκπαίδευσης χρησιμοποιούμε το σύνολο επικύρωσης και παρατηρούμε αν η ακρίβεια του μοντέλου αυξάνεται όσο αυξάνεται και η ακρίβεια στο σύνολο εκπαίδευσης. Σε αντίθετη περίπτωση έχουμε ένδειξη για υπερπροσαρμογή του μοντέλου στα δεδομένα.

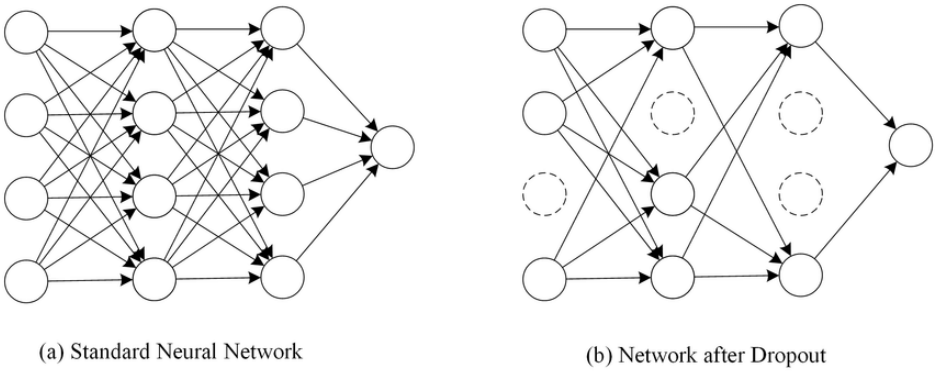


Εικόνα 3-7: Χωρισμός δεδομένων σε σύνολο εκπαίδευσης (training), ελέγχου (test) και επικύρωσης (validation) [18]

3.3.5 Απόσυρση Νευρώνων (Dropout)

Με τον όρο *Απόσυρση Νευρώνων (Dropout)* αναφερόμαστε σε μια τεχνική για βελτίωση της μάθησης των βαθιών νευρωνικών δικτύων. Καθώς το πλήθος των νευρώνων και των στρωμάτων αυξάνεται υπάρχει μεγάλη πιθανότητα να υπάρξει το πρόβλημα του Overfitting που περιγράψαμε προηγουμένως.

Το Dropout στοχεύει στο να αποσύρει από τη μοντέλο άσχετη γνώση, δηλαδή πληροφορίες που απλά θα αύξαναν την πολυπλοκότητά του χωρίς να βελτιώνουν την ακρίβεια. Αυτό επιτυγχάνεται μέσω της απόσυρσης ενός ποσοστού των συνολικών νευρώνων. Το ποσοστό των νευρώνων που επιλέγουμε να αποσύρουμε θα αποκτήσει μηδενικά βάρη και συνεπώς δεν θα έχει καμία συνεισφορά στο μοντέλο.



Εικόνα 3-8: Ένα Νευρωνικό Δίκτυο πριν και μετά την εφαρμογή Dropout [19]

3.4 Συναρτήσεις Ενεργοποίησης

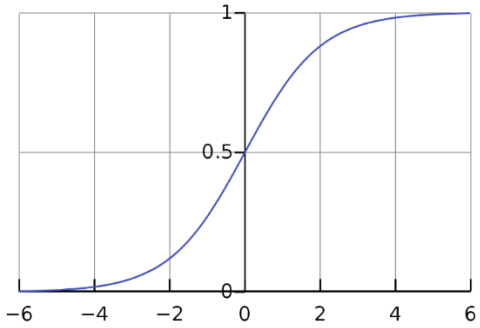
Όπως αναφέρθηκε, μέσω των συναρτήσεων ενεργοποίησης (activation functions) οι νευρώνες υπολογίζουν την τιμή εξόδου (output), συγκρίνοντας την τιμή αυτή με ένα κατώφλι (threshold). Ο νευρώνας είτε θα ενεργοποιηθεί και θα αποστείλει την έξοδό του στον επόμενο (ή ως τελική έξοδο) είτε θα μείνει αδρανής. Στα επόμενα θα παρουσιάσουμε συνοπτικά τις βασικές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται. Αυτές μπορεί να είναι γραμμικές ή μη-γραμμικές, αν και στην πράξη προτιμώνται κυρίως οι μη-γραμμικές, στις οποίες θα αναφερθούμε και εμείς.

Λογιστική Συνάρτηση

Η *λογιστική συνάρτηση (logistic function)* δίνεται απ' τον τύπο:

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

Για κ=1 έχουμε την τυπική λογιστική συνάρτηση η οποία απεικονίζεται και στην παραπάνω εικόνα. Είναι μια από τις πιο γνωστές συναρτήσεις ενεργοποίησης καθώς έχει εφαρμογές σε προβλήματα από πολλά επιστημονικά πεδία.



Εικόνα 3-9: Η Λογιστική Συνάρτηση [21]

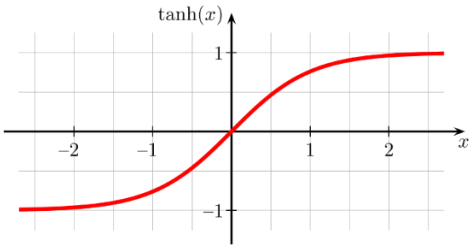
Αρχικά χρησιμοποιήθηκε για τη μοντελοποίηση της πληθυσμιακής αύξησης όπου ο ρυθμός αναπαραγωγής είναι ανάλογος με το τρέχον πλήθος του πληθυσμού και με το πλήθος των διαθέσιμων πόρων. Επίσης, χρησιμοποιείται ιδιαίτερα στη Στατιστική όπου έχουμε τη γνωστή Λογιστική Παλινδρόμηση η οποία μοντελοποιεί την πιθανότητα να συμβεί ένα γεγονός ανάλογα με τις τιμές των μεταβλητών που σχετίζεται.

Συνάρτηση υπερβολικής εφαπτομένης

Η *συνάρτηση της υπερβολικής εφαπτομένης (tanh)* είναι γνωστή από την τριγωνομετρία και χρησιμοποιείται συνήθως για προβλήματα διωνυμικής κατηγοριοποίησης.

Ο μαθηματικός τύπος που την περιγράφει είναι:

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



Εικόνα 3-10: Γράφημα υπερβολικής εφαπτομένης [22]

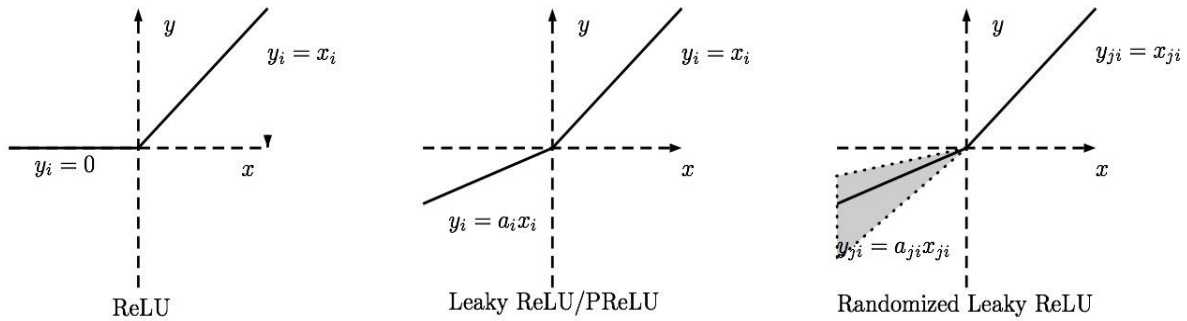
Όπως φαίνεται και από το γράφημά της, η υπερβολική εφαπτομένη σε αντίθεση με τη λογιστική συνάρτηση, διατηρεί το πρόσημο των τιμών.

Γραμμικές μονάδες ανόρθωσης

Η *γραμμική μονάδα ανόρθωσης (rectified linear unit ή ReLU)* είναι η πιο διαδεδομένη συνάρτηση ενεργοποίησης και έχει πεδίο τιμών ίδιο με το πεδίο ορισμού όταν η είσοδος είναι θετικός αριθμός και μηδέν όταν είναι αρνητικός. Χρησιμοποιούνται αρκετά στα πεδία της υπολογιστικής όρασης και της αναγνώρισης φωνής. Ορίζεται ως :

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

Ένα ζήτημα που μπορεί να προκύψει με τη ReLU είναι όταν καθώς όλες οι αρνητικές τιμές απεικονίζονται στο μηδέν, μπορεί να προκληθεί αδράνεια στους νευρώνες καθώς δεν θα ενεργοποιούνται σχεδόν καθόλου. Για να αντιμετωπιστεί αυτό το πρόβλημα υπάρχουν και επεκτάσεις της όπου αντί για μηδέν βάζουμε μια πολύ μικρή τιμή (Leaky ReLU) ή μια σταθερά (Randomized Leaky ReLU).



Εικόνα 3-11: Γραφικές απεικονίσεις των ReLU, Leaky ReLU και Randomized Leaky ReLU [23]

Η συνάρτηση Softmax

Η συνάρτηση *Softmax* ή αλλιώς *κανονικοποιημένη εκθετική συνάρτηση* είναι μια γενίκευση της λογιστικής συνάρτησης σε περισσότερες διαστάσεις και χρησιμοποιείται ως συνάρτηση ενεργοποίησης για προβλήματα κατηγοριοποίησης πολλών κλάσεων. Η συνάρτηση δέχεται ως είσοδο πολυδιάστατα διανύσματα και τα κανονικοποιεί, επιστρέφοντας ένα ίδιας διάστασης διάνυσμα που αποτελείται όμως από πιθανότητες. Ο μαθηματικός τύπος φαίνεται παρακάτω:

$$f(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_i}}$$

Όπου k είναι το πλήθος των διαστάσεων των δεδομένων, και $z \in \mathbb{R}^k$

3.5 Εντοπισμός αντικειμένων σε εικόνες με Συνελικτικά Νευρωνικά Δίκτυα

3.5.1 Εισαγωγή

Ο κλάδος της Βαθιάς Μάθησης έχει κάνει εξαιρετική πρόοδο στην επίλυση πολλών προβλημάτων της Υπολογιστικής Όρασης, χρησιμοποιώντας κυρίως Συνελικτικά Νευρωνικά Δίκτυα (CNN). Το πρόβλημα του Εντοπισμού Αντικειμένων (Object Detection) είναι να εκπαιδεύσουμε ένα μοντέλο να εντοπίζει ένα ή περισσότερα αντικείμενα σε μια εικόνα.

Το βασικό ζητούμενο στον εντοπισμό αντικειμένων είναι να υπολογίσουμε ένα ορθογώνιο περίγραμμα (bounding box), συνήθως ένα παραλληλόγραμμο, που να περικλείει το αντικείμενο όσο το δυνατόν καλύτερα. Στην περίπτωση που έχουμε πολλά αντικείμενα, πρέπει να υπολογίσουμε πολλά bounding boxes. Αυτή είναι και η βασική διαφορά του προβλήματος αυτού από τα κλασσικά προβλήματα κατηγοριοποίησης που αναφερθήκαμε. Το τελικό αποτέλεσμα εδώ δεν είναι εξ αρχής γνωστό καθώς δεν γνωρίζουμε πόσες φορές ένα αντικείμενο μπορεί να απεικονίζεται σε μια εικόνα.

Μια απλοϊκή αντιμετώπιση του προβλήματος αυτού θα ήταν να εντοπίσουμε πολλές περιοχές ενδιαφέροντος από μια εικόνα και να χρησιμοποιήσουμε CNN για να εντοπίσουμε την παρουσία του αντικειμένου σε κάθε μία από αυτές. Αυτό όμως είναι μια χρονοβόρα διαδικασία και έχει μεγάλο υπολογιστικό κόστος. Για την αντιμετώπιση αυτού του ζητήματος έχουν αναπτυχθεί διάφορες αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων που επιλύουν προβλήματα εντοπισμού αντικειμένων αποδοτικά, κάποιες απ’ τις οποίες θα περιγράψουμε συνοπτικά παρακάτω.

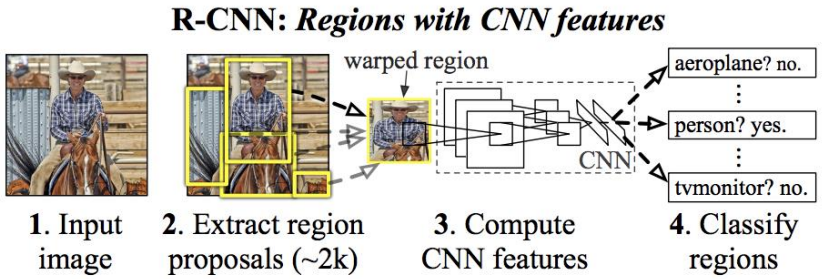
3.5.2 Region-Based Συνελικτικά Νευρωνικά Δίκτυα

Τα *Region-Based CNN* [26] δέχονται εικόνες ως είσοδο και υπολογίζουν *περιοχές ενδιαφέροντος (regions of interest ή ROI)* τα οποία είναι τα πιθανά bounding boxes για το αντικείμενο που θέλουμε να εντοπίσουμε σε μια εικόνα.

R-CNN

Ο πρώτος αλγόριθμος τέτοιου είδους, ο R-CNN, δημοσιεύτηκε από τους Girshick et al. [27] και χρησιμοποιήθηκε για να αντιμετωπίσει το πρόβλημα της εύρεσης υπερβολικά πολλών ROI's που υπήρχε έως τότε χρησιμοποιώντας τα απλά CNN's. Λειτουργεί χρησιμοποιώντας έναν αλγόριθμο Επιλεκτικής Αναζήτησης (Selective Search) για να εντοπίσει 2000 προτεινόμενες περιοχές (region proposals) και επικεντρώνεται στο πρόβλημα της αναγνώρισης αντικειμένων σε αυτές τις περιοχές. Έπειτα, χρησιμοποιεί ένα CNN για να αναλύσει τις περιοχές αυτές και επιστρέφει ως έξοδο ένα διάνυσμα με 4096 διαστάσεις, το οποίο χρησιμοποιείται ως είσοδος για ένα μοντέλο SVM για την τελική αναγνώριση της ύπαρξης ή όχι του αντικειμένου στην εκάστοτε περιοχή.

Συγκριτικά με τις μεθόδους πριν τη δημοσίευσή του, ο αλγόριθμος R-CNN έχει πολύ καλά αποτελέσματα αλλά χρειάζεται σχεδόν ένα λεπτό για τον έλεγχο μιας εικόνας και δεν προτείνεται για εφαρμογές πραγματικού χρόνου. Στην παρακάτω εικόνα αποτυπώνεται συνοπτικά η λειτουργία του R-CNN:

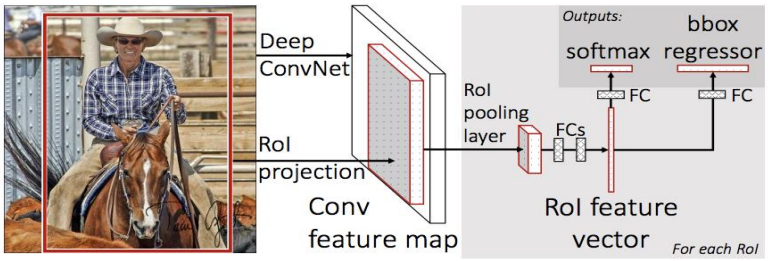


Εικόνα 3-12: Παράδειγμα λειτουργίας του αλγόριθμου R-CNN [27]

Fast R-CNN

Μια βελτίωση του R-CNN είναι ο αλγόριθμος Fast R-CNN [28] που δημοσιεύτηκε από τους ίδιους συγγραφείς δύο χρόνια αργότερα. Λειτουργεί με παρόμοιο τρόπο αλλά αντί να υπολογίζει εξ αρχής τις προτεινόμενες περιοχές, και μετά να τις τροφοδοτεί στο συνελικτικό δίκτυο, πρώτα τροφοδοτεί την εικόνα εισόδου στο CNN, για να υπολογίσει ένα χάρτη χαρακτηριστικών (feature map). Μέσω αυτού, εντοπίζει τις προτεινόμενες περιοχές που θα εξετάσει, και τις αποστέλλει σε ένα στρώμα pooling (ROI pooling layer) στο οποίο εφαρμόζεται η συνάρτηση ενεργοποίησης Softmax για την τελική κατηγοριοποίηση. Αυτό

έχει ως αποτέλεσμα να μην χρειάζεται κάθε φορά να υπολογίζει 2000 περιοχές ανά εικόνα και να εξοικονομεί αρκετό χρόνο.

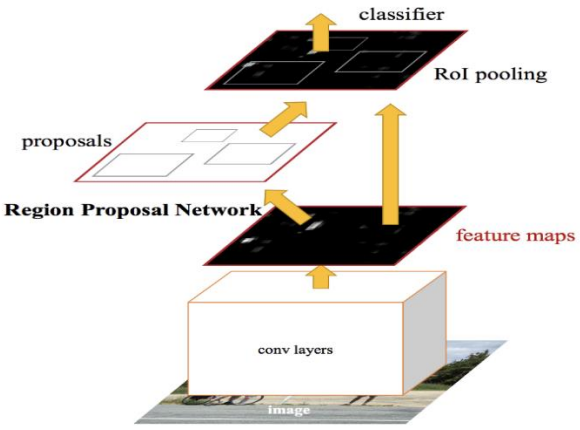


Εικόνα 3-13: Παράδειγμα λειτουργίας του αλγόριθμου Fast R-CNN [28]

Faster R-CNN

Και στις δύο περιπτώσεις, χρησιμοποιείται ένας αλγόριθμος Επιλεκτικής Αναζήτησης για την εύρεση των προτεινόμενων περιοχών, ο οποίος έχει αρκετό υπολογιστικό κόστος. Οι Saoqing et al. [29] δημοσίευσαν τον αλγόριθμο Faster R-CNN ο οποίος δεν χρησιμοποιεί τον αλγόριθμο της Επιλεκτικής Αναζήτησης και ενσωματώνει στο ίδιο το συνελικτικό δίκτυο την ικανότητα εντοπισμού των περιοχών ενδιαφέροντος.

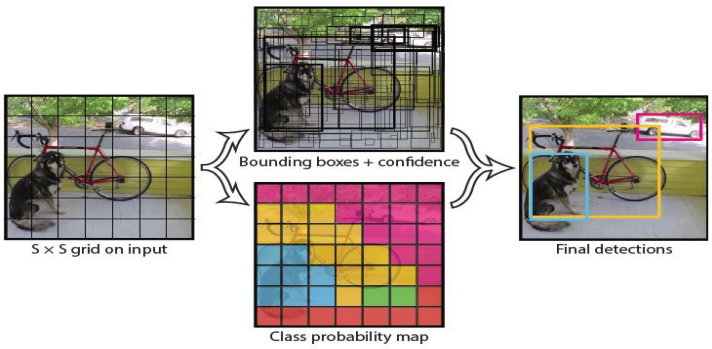
Αυτό γίνεται προσθέτοντας ένα πλήρως συνδεδεμένο συνελικτικό δίκτυο το οποίο εξετάζει το χάρτη χαρακτηριστικών που παράγεται στο πρώτο στάδιο, και επιστρέφει τις πιθανές περιοχές ενδιαφέροντος μαζί με ένα συντελεστή εμπιστοσύνης (confidence score) που τις χαρακτηρίζει. Ο συντελεστής αντιστοιχεί στην πιθανότητα να βρίσκεται κάποιο αντικείμενο στην περιοχή αυτή. Όπως και στον αλγόριθμο Fast R-CNN, οι περιοχές ενδιαφέροντος εισάγονται σε ένα στρώμα pooling το οποίο και χρησιμοποιείται για την τελική κατηγοριοποίηση. Ο αλγόριθμος Faster R-CNN έχει πολύ καλές επιδόσεις συγκριτικά με τους προγενέστερους αλγόριθμους και μπορεί να χρησιμοποιηθεί και για εφαρμογές πραγματικού χρόνου. Η γενική δομή του αλγορίθμου φαίνεται στην παρακάτω εικόνα:



Εικόνα 3-14: Δομή του αλγόριθμου Faster R-CNN [29]

3.5.3 Ο αλγόριθμος You Only Look Once (YOLO)

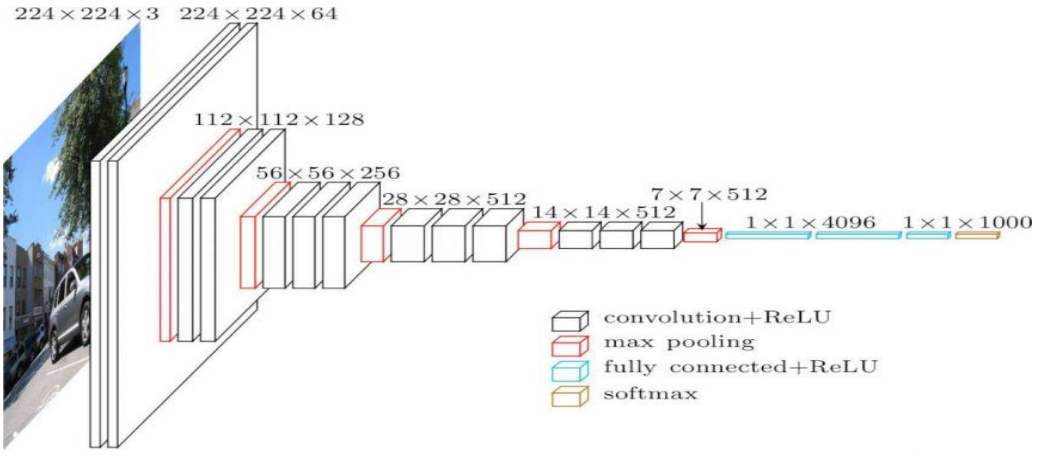
Όπως είδαμε, οι Region-Based αλγόριθμοι χρησιμοποιούν δύο βασικές διεργασίες. Μία για να εντοπίσουν τις περιοχές ενδιαφέροντος και τα bounding boxes και μία για να αναγνωρίσουν την ύπαρξη του αντικειμένου στις περιοχές αυτές. Ο αλγόριθμος YOLO [30] διαφέρει από τις Region-Based προσεγγίσεις και χρησιμοποιεί μόνο ένα συνελικτικό νευρωνικό δίκτυο τόσο για τον υπολογισμό των bounding boxes όσο και για τον εντοπισμό των αντικειμένων σε αυτές. Είναι τάξης μεγέθους ταχύτερος απ' τους προηγούμενους με εξαιρετική απόδοση σε προβλήματα πραγματικού χρόνου. Η εικόνα χωρίζεται σε ένα πλέγμα διαστάσεων SxS και για κάθε στοιχείο του πλέγματος (κελί) υπολογίζεται ένα πλήθος bounding boxes συνοδευόμενα από ένα συντελεστή εμπιστοσύνης (confidence score). Σχηματικά αυτό φαίνεται στην ακόλουθη εικόνα:



Εικόνα 3-15: Ο αλγόριθμος Yolo [30]

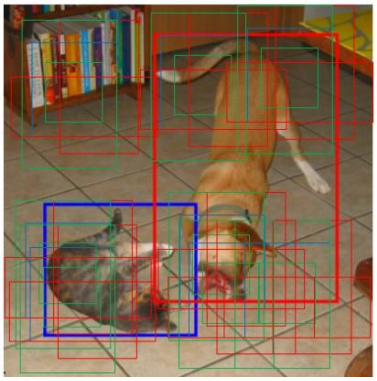
3.5.4 Single Shot Detector (SSD)

Ο αλγόριθμος Single Shot Detector ή SSD [31], όπως και ο αλγόριθμος YOLO που είδαμε προηγουμένως, χρησιμοποιεί ένα μόνο πέρασμα (single shot) της εικόνας για τον εντοπισμό αντικειμένων. Χρησιμοποιεί ως βάση το δίκτυο VGG-16 των Simonyan et al. [32] το οποίο είναι εξαιρετικά δημοφιλές και χρησιμοποιείται σε προβλήματα αναγνώρισης εικόνας (image recognition) για την εξαγωγή χαρακτηριστικών. Στην ακόλουθη εικόνα φαίνεται η αρχιτεκτονική του δικτύου VGG-16:



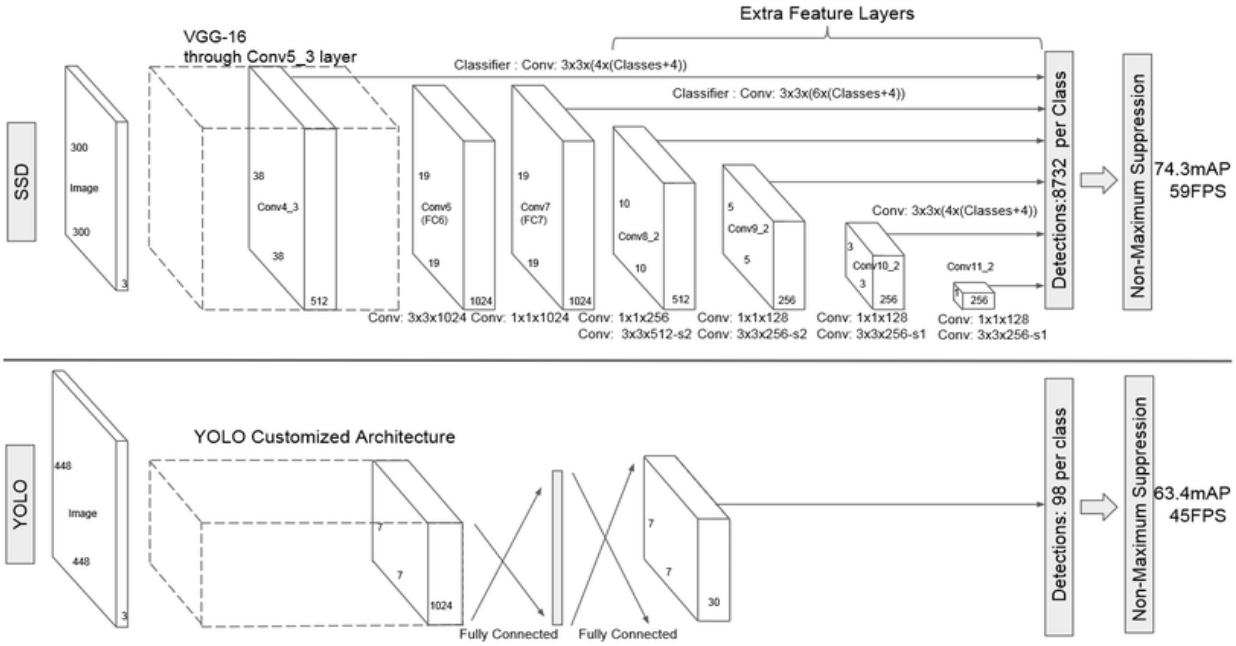
Εικόνα 3-16: Η αρχιτεκτονική του δικτύου VGG-16 [37]

Στον αλγόριθμο SSD, το δίκτυο VGG-16 συνοδεύεται από επιμέρους συνελικτικά στρώματα τα οποία προοδευτικά έχουν όλο και μικρότερες διαστάσεις. Τα αρχικά, μεγάλου μεγέθους στρώματα, αναγνωρίζουν καλύτερα μικρά αντικείμενα ενώ τα βαθιά στρώματα γενικεύουν τη γνώση αναγνωρίζοντας καλύτερα τα μεγάλα αντικείμενα. Τα ενδιάμεσα στρώματα εξετάζουν bounding boxes διαφορετικών διαστάσεων και κλίμακας. Η ιδέα είναι ότι γενικά, κάποια αντικείμενα λόγω της φυσικής τους γεωμετρίας αναγνωρίζονται καλύτερα με πλατιά ορθογώνια ενώ άλλα με φαρδιά, όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 3-17: Τα bounding boxes του αλγόριθμου SSD [31]

Στο ακόλουθο σχήμα απεικονίζονται η αρχιτεκτονικές των αλγορίθμων YOLO και SSD. Οι αλγόριθμοι εκπαιδεύονται χρησιμοποιώντας το σύνολο δεδομένων VOC2007 [38].



Εικόνα 3-18: Οι αρχιτεκτονικές των αλγορίθμων SSD και YOLO. [31]

ΚΕΦΑΛΑΙΟ 4

ΥΠΟΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΗΣ ΑΝΑΓΝΩΡΙΣΗΣ ΣΤΑΤΙΚΩΝ ΧΕΙΡΟΝΟΜΙΩΝ

Στο κεφάλαιο αυτό θα παρουσιάσουμε ένα σύστημα για την αυτόματη αναγνώριση στατικών χειρονομιών σε real-time βίντεο, χρησιμοποιώντας μια κάμερα RGB και έναν υπολογιστή. Για την υλοποίηση χρησιμοποιήθηκε λειτουργικό σύστημα Ubuntu Linux και η γλώσσα προγραμματισμού Python 3.7, χρησιμοποιώντας ως διεπαφή ένα Jupyter Notebook. Ο κώδικας είναι σε τέτοια μορφή ώστε να μπορεί να επαναχρησιμοποιηθεί με μικρές αλλαγές.

4.1 Γενική μεθοδολογία

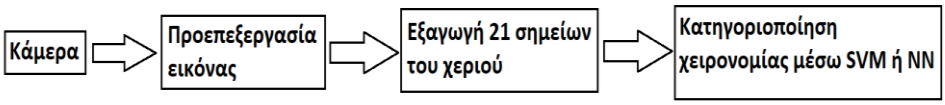
Τα κύρια βήματα που ακολουθούμε για την υλοποίηση είναι τα εξής:

Εκπαίδευση Μοντέλων:

Αρχικά, κατασκευάζουμε ένα σύνολο δεδομένων που αποτελείται από 1565 εικόνες, όπως θα δούμε στην επόμενη ενότητα. Οι εικόνες περιλαμβάνουν 12 στατικές χειρονομίες που εκτελούνται με το δεξί χέρι. Μέσω του αλγορίθμου του MediaPipe, που θα περιγράψουμε στη συνέχεια, εξάγουμε για κάθε μία, τις κανονικοποιημένες συντεταγμένες (x,y,z) 21 σημείων του χεριού, τις οποίες αποθηκεύουμε σε ένα αρχείο csv, μαζί με το όνομα της χειρονομίας που αντιστοιχούν. Χρησιμοποιώντας αυτό το σύνολο, εκπαιδεύουμε μια Μηχανή Διανυσμάτων Υποστήριξης (SVM) και ένα Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης Feedforward Neural Network) για την κατηγοριοποίηση των χειρονομιών.

Αναγνώριση χειρονομίας:

Χρησιμοποιώντας τη βιβλιοθήκη OpenCV της Python, αντλούμε το εισερχόμενο βίντεο από την κάμερα, κάνοντας κάποιες απαραίτητες τροποποιήσεις σε κάθε στιγμιότυπο, ώστε να μετατραπεί στη μορφή που θα απαιτηθεί για τα επόμενα βήματα. Συγκεκριμένα, αντιστρέφουμε οριζοντίως την εικόνα (horizontal flip) και τη μετατρέπουμε από τη μορφή BGR, σε μορφή RBG. Σε κάθε στιγμιότυπο του βίντεο (frame), εφαρμόζουμε τον αλγόριθμο του MediaPipe, ο οποίος επιστρέφει τις συντεταγμένες των 21 σημείων του χεριού. Εφαρμόζοντας ένα από τα μοντέλα κατηγοριοποίησης, επιστρέφεται η χειρονομία που προβλέπει το μοντέλο καθώς και το ποσοστό ακρίβειάς του.



Εικόνα 4-1: Γενική μεθοδολογία της υλοποίησης

4.2 Αναγνώριση χεριού μέσω του MediaPipe

Το MediaPipe [35] είναι ένα πλαίσιο λογισμικού ανοικτού κώδικα (open source framework) που προσφέρει καινοτόμες λύσεις σε προβλήματα Υπολογιστικής Όρασης και Μηχανικής Μάθησης. Ο σκοπός του MediaPipe είναι να καταστήσει σχετικά απλή τη χρήση σύγχρονων αλγορίθμων Βαθιάς Μάθησης με σύνθετες αρχιτεκτονικές. Εμείς χρησιμοποιούμε τον αλγόριθμο που χρησιμοποιεί το MediaPipe Hands [2],[36] , ο οποίος χρησιμοποιείται για την αναγνώριση χεριών σε εικόνες και βίντεο πραγματικού χρόνου (Real-Time Hand Tracking), χρησιμοποιώντας μία απλή RGB κάμερα. Παρακάτω θα αναφέρουμε τα βασικά χαρακτηριστικά του. Η αναλυτική παρουσίαση μπορεί να βρεθεί στη βιβλιογραφία.

Ο αλγόριθμος χρησιμοποιεί δύο επιμέρους μοντέλα τα οποία λειτουργούν συνδυαστικά. Το μοντέλο BlazePalm Detector, που βασίζεται στο μοντέλο SSD που αναφέραμε στο προηγούμενο κεφάλαιο, και ένα μοντέλο παλινδρόμησης, το Hand Landmark Model. Χρησιμοποιώντας το πρώτο μοντέλο γίνεται η εξαγωγή της περιοχής της εικόνας που βρίσκεται το χέρι και μέσω του μοντέλου παλινδρόμησης υπολογίζονται οι θέσεις 21 σημείων του χεριού (hand landmarks).

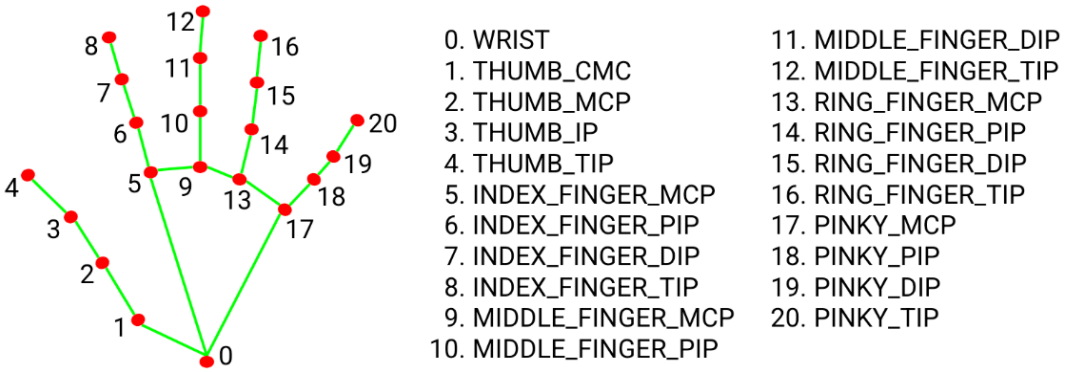
BlazePalm Detector

Το μοντέλο του BlazePalm Detector είναι υπεύθυνο για την αναγνώριση της παλάμης του χεριού σε εικόνες. Γενικά, ο εντοπισμός της παλάμης αντί ολόκληρου του χεριού προσφέρει αρκετά πλεονεκτήματα καθώς η παλάμη από μόνη της αποτελεί ένα αρκετά πιο συμπαγές σώμα απ’ότι ολόκληρο το χέρι μαζί με τα δάκτυλα. Η παλάμη μοντελοποιείται από τον αλγόριθμο μέσω τετράγωνων bounding boxes, αντί ορθογώνιων, που είδαμε στους αλγορίθμους YOLO και SSD, μειώνοντας έτσι τις υπολογιστικές απαιτήσεις του αλγορίθμου. Για την εξαγωγή των χαρακτηριστικών χρησιμοποιείται ένα σύστημα encoder-decoder παρόμοιο με αυτό του FPN (Feature Pyramid Networks) [33]. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας ως συνάρτηση κόστους την focal loss [34], που είναι μια παραλλαγή της διασταυρωμένης εντροπίας (cross-entropy loss).

Hand Landmark Model

Αφότου εντοπιστεί η παλάμη του χεριού στην εικόνα εισόδου, το μοντέλο Hand Landmark χρησιμοποιεί τεχνικές παλινδρόμησης για να υπολογίσει τις συντεταγμένες 21 σημείων του χεριού. Το μοντέλο μαθαίνει να αναγνωρίζει ακόμη και μερικώς εμφανή χέρια, καθώς και περιπτώσεις που ένα μέρος του χεριού αυτο-επικαλύπτεται. Κάθε ένα από τα 21 σημεία του χεριού (hand landmarks) αποτελείται από τις συντεταγμένες x, y και από μια ποσότητα z που είναι μια προσέγγιση του σχετικού βάθους τους, ως προς τον καρπό του χεριού. Οι συντεταγμένες κανονικοποιούνται ως προς τις διαστάσεις της συνολικής εικόνας. Επίσης, το μοντέλο υπολογίζει την πιθανότητα της παρουσίας χεριού στην εικόνα εισόδου (hand presence) και μέσω διωνυμικής κατηγοριοποίησης υπολογίζεται αν το χέρι είναι το δεξί ή το αριστερό (handedness).

Τα 21 σημεία του χεριού απεικονίζονται στην παρακάτω εικόνα.

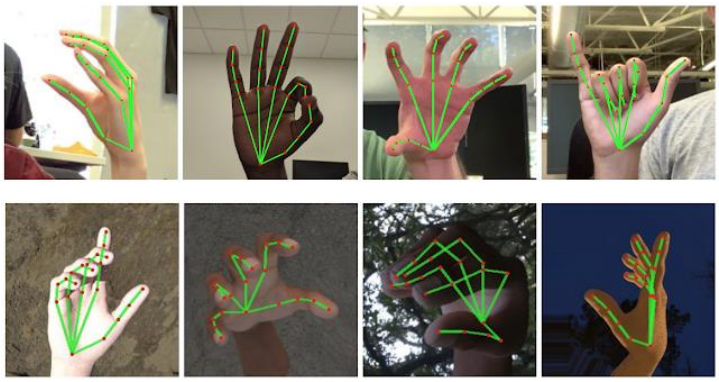


Εικόνα 4-2 MediaPipe Hands – 21 Hand Landmarks [2]

Για την εκπαίδευση των μοντέλων χρησιμοποιήθηκαν τρία σύνολα δεδομένων:

- In-the-wild dataset: Περιέχει 6000 εικόνες σε διάφορα περιβάλλοντα και με διαφορετικές συνθήκες φωτισμού.
- In-house dataset: Περιέχει 10.000 εικόνες που απεικονίζουν πολλές διαφορετικές χειρονομίες που εκτέλεσαν 30 άνθρωποι.
- Synthetic dataset: Αποτελείται από 100.000 συνθετικές εικόνες.

Το μοντέλο BlazePalm εκπαιδεύτηκε χρησιμοποιώντας το In-the-wild σύνολο δεδομένων και τα υπόλοιπα σύνολα χρησιμοποιήθηκαν για την εκπαίδευση του Hand Landmark μοντέλου. Στην επόμενη εικόνα φαίνονται παραδείγματα από τα επιμέρους σύνολα δεδομένων καθώς και η εφαρμογή του αλγορίθμου σε αυτά.



Εικόνα 4-3: Παραδείγματα συνόλων εκπαίδευσης του αλγορίθμου του MediaPipe. Η επάνω σειρά αντιστοιχεί σε πραγματικές εικόνες και η κάτω σειρά σε συνθετικές. [2]

Παράμετροι αλγορίθμου

Ο αλγόριθμος έχει τέσσερις βασικές παραμέτρους που καθορίζει ο χρήστης.

- **STATIC_IMAGE_MODE**: Αυτή η παράμετρος χρησιμοποιείται για τις περιπτώσεις που θέλουμε να εφαρμόσουμε τον αλγόριθμο σε στατικές εικόνες και παίρνει τιμές true/false. Ενεργοποιήσαμε αυτή την παράμετρο κατά τη διαδικασία της κατασκευής του συνόλου δεδομένων, όταν δηλαδή εξετάσαμε τις εικόνες που περιέχουν τις χειρονομίες για τον εντοπισμό των συντεταγμένων των σημείων του χεριού. Για την εφαρμογή του αλγορίθμου σε βίντεο, θέτουμε την παράμετρο αυτή σε false. Σε αυτή την περίπτωση ο αλγόριθμος χρησιμοποιεί την αναγνώριση που κάνει στα πρώτα στιγμιότυπα του βίντεο που πρωτοεμφανίζεται το χέρι, ώστε να έχει καλύτερη απόδοση για τη συνέχεια.

- **MAX_NUM_HANDS**: Αναφέρεται στο μέγιστο πλήθος των χεριών που σκοπεύουμε να αναγνωρίσουμε. Στη συγκεκριμένη υλοποίηση θέτουμε την τιμή 1. Αυτό είναι αρκετά χρήσιμο γιατί εφόσον εντοπιστεί ένα χέρι, η παρουσία άλλων δεν θα επηρεάσει την αποτελεσματικότητα του αλγορίθμου και θα συνεχιστεί η αναγνώριση με το χέρι αυτό.

- **MIN_DETECTION_CONFIDENCE**: Το ελάχιστο ποσοστό ώστε να θεωρηθεί επιτυχημένος ο εντοπισμός χεριών στην εικόνα εισόδου. Παίρνει τιμές στο διάστημα [0.0, 1.0]. Στη δική μας υλοποίηση χρησιμοποιούμε ως τιμή το 0.5.

- **MIN_TRACKING_CONFIDENCE**: Ο αλγόριθμος μόλις εντοπίσει τα σημεία του χεριού συνεχίζει να ακολουθεί την κίνηση αυτών και δεν εφαρμόζει εκ νέου την αναγνώριση της παλάμης σε κάθε νέο στιγμιότυπο. Η παράμετρος αυτή παίρνει τιμές από στο διάστημα [0.0, 1.0] και αναφέρεται στο ποσοστό που θέλουμε να θεωρείται επιτυχής η παρακολούθηση των σημείων. Εμείς χρησιμοποιούμε την τιμή 0.5.

4.3 Κατασκευή συνόλου δεδομένων

4.3.1 Περιγραφή

Το σύνολο δεδομένων που χρησιμοποιούμε βασίζεται στο [41], το οποίο αποτελείται από στατικές χειρονομίες που εκτέλεσαν 4 διαφορετικοί άνθρωποι, χρησιμοποιώντας την κάμερα Creative Senz3D. Εκτελέστηκαν 11 διαφορετικές χειρονομίες, 30 φορές η κάθε μία, δηλαδή συνολικά 1320 φορές. Υπάρχουν διαθέσιμες και άλλες πληροφορίες όπως το χρώμα και το βάθος των δειγμάτων αλλά δεν τα χρησιμοποιούμε στην υλοποίησή μας καθώς προορίζεται για χρήση με απλές κάμερες, χωρίς αισθητήρες βάθους. Το σύνολο αυτό έχει χρησιμοποιηθεί στις δημοσιεύσεις [39] και [40] για προβλήματα αναγνώρισης χειρονομιών.

Επιπρόσθετα με το παραπάνω σύνολο, έγιναν επιπλέον λήψεις από εμάς για κάθε χειρονομία, υπό διάφορες γωνίες λήψης και αποστάσεις από την κάμερα, οι οποίες προστέθηκαν στο αρχικό σύνολο.

Οι χειρονομίες φαίνονται στην παρακάτω εικόνα :

Εικόνα 4-4: Οι 11 χειρονομίες του συνόλου δεδομένων (G1 έως G11) [39]

4.3.2 Εισαγωγή νέας χειρονομίας

Επίσης προσθέσαμε μία νέα χειρονομία, για την οποία λήφθηκαν 30 στιγμιότυπα. Η νέα αυτή χειρονομία χρησιμοποιείται για να αξιολογήσουμε την απόδοση σε πιο πραγματικές συνθήκες, χρησιμοποιώντας δηλαδή μικρό σύνολο παρατηρήσεων. Η επιπλέον χειρονομία που εισάγουμε είναι η “thumbs-up” (G12) :

Εικόνα 4-5: Χειρονομία thumbs-up (G12)

Οι εικόνες του συνόλου δεδομένων είναι ομαδοποιημένες ανά χρήστη και ανά χειρονομία, σε διαφορετικούς φακέλους. Δηλαδή, το σύνολο αποτελείται από φακέλους που αντιστοιχούν σε κάθε χρήστη και κάθε ένας φάκελος από αυτούς, περιέχει έναν υποφάκελο για κάθε μία από τις χειρονομίες, όπου βρίσκονται οι ληφθείσες εικόνες.

Χρησιμοποιώντας τον αλγόριθμο του MediaPipe, γίνεται η επεξεργασία των εικόνων και η εξαγωγή των 21 σημείων του χεριού που αναγνωρίζει. Σε κάποιες εικόνες, ο αλγόριθμος δεν αναγνώριζε επιτυχώς τα σημεία και έτσι τις απορρίψαμε. Συγκεκριμένα, απορρίψαμε 33 εικόνες από τις 1628 που είχαμε συνολικά. Δηλαδή το 2% του συνόλου.

Το τελικό σύνολο αποτελείται από 1595 εικόνες. Χρησιμοποιούμε το ένα τρίτο (33%) αυτών ως σύνολο ελέγχου (527) και το υπόλοιπο ως σύνολο εκπαίδευσης (1068). Σε κάθε εικόνα, αντιστοιχεί ένα σύνολο από 21 σημεία τριών διαστάσεων, δηλαδή συνολικά 63 παρατηρήσεις.

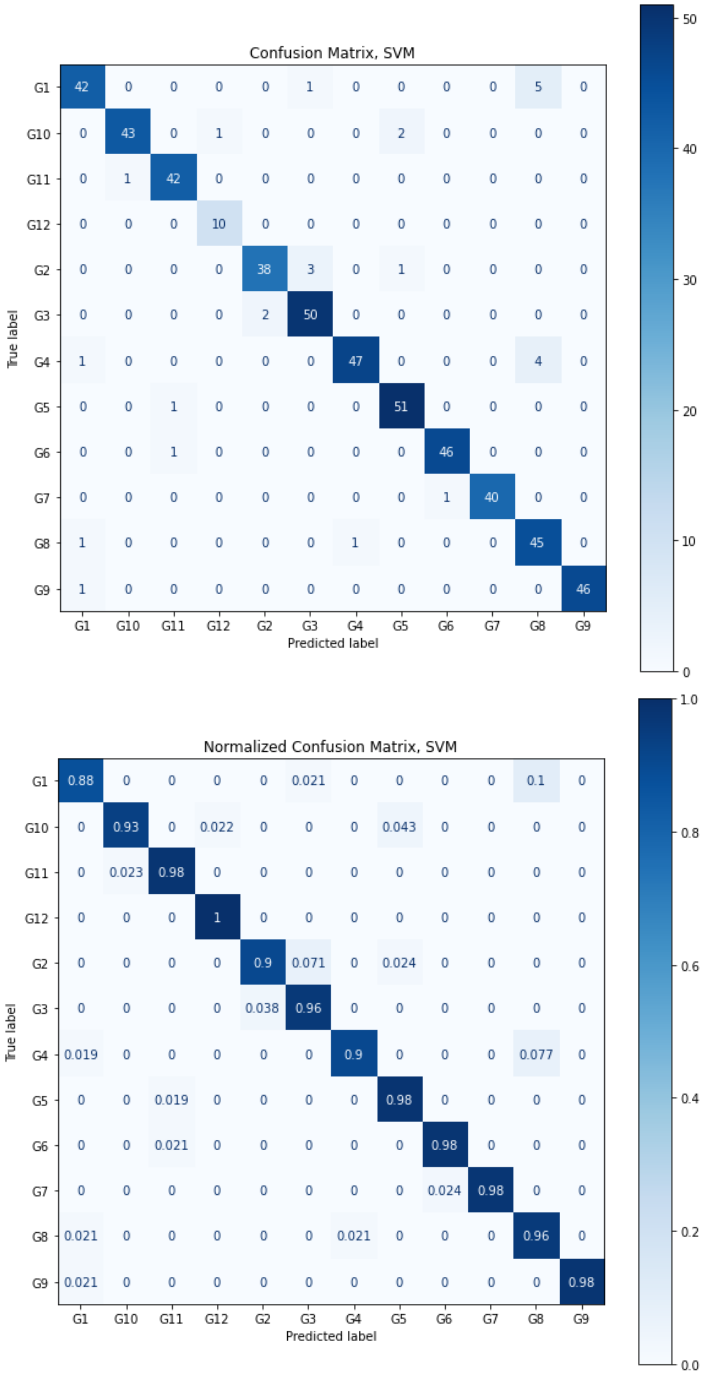
4.4 Κατηγοριοποίηση χειρονομιών με Μηχανές Διανυσμάτων Υποστήριξης

Όσον αφορά τους αλγορίθμους Μηχανικής Μάθησης, η συνιστώμενη και πιο δημοφιλής πρακτική για τέτοιου είδους εφαρμογές είναι η εφαρμογή Μηχανών Διανυσμάτων Υποστήριξης (SVM).

Για την εκπαίδευση του μοντέλου, χρησιμοποιούμε τη βιβλιοθήκη της Python Scikit-Learn [42].

4.4.1 Εκπαίδευση μοντέλου

Για την κατάλληλη επιλογή των παραμέτρων του μοντέλου SVM έγιναν διάφορες δοκιμές χρησιμοποιώντας διαφορετικές συναρτήσεις πυρήνα (kernel functions). Το μοντέλο που επιλέξαμε χρησιμοποιεί τη συνάρτηση rbf (radial basis function), με σταθερά κανονικοποίησης C=70. Η μέση ακρίβεια του μοντέλου υπολογίστηκε σε **94.8%**. Παρακάτω φαίνεται ο πίνακας σύγχυσης (και ο κανονικοποιημένος) που περιγράφει τα αποτελέσματα του μοντέλου καθώς και οι μετρικές αξιολόγησης της απόδοσής του:



Εικόνα 4-6: Μοντέλο SVM – Πίνακες Σύγχυσης

	precision	recall	f1-score	support
G1	0.93	0.88	0.90	48
G10	0.98	0.93	0.96	46
G11	0.95	0.98	0.97	43
G12	0.91	1.00	0.95	10
G2	0.95	0.90	0.93	42
G3	0.93	0.96	0.94	52
G4	0.98	0.90	0.94	52
G5	0.94	0.98	0.96	52
G6	0.98	0.98	0.98	47
G7	1.00	0.98	0.99	41
G8	0.83	0.96	0.89	47
G9	1.00	0.98	0.99	47
accuracy			0.95	527
macro avg	0.95	0.95	0.95	527
weighted avg	0.95	0.95	0.95	527

Εικόνα 4-7: Μετρικές αξιολόγησης ανά χειρονομία, μοντέλο SVM

4.4.2 Αξιολόγηση

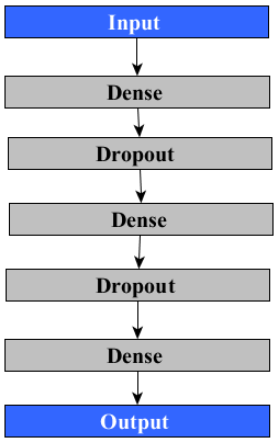
Γενικά, η αξιολόγηση ενός μοντέλου κατηγοριοποίησης, γίνεται μέσω του πίνακα σύγχυσης και διαφόρων μετρικών που μπορούν να εξαχθούν από αυτόν, όπως είδαμε σε προηγούμενο κεφάλαιο. Στη συγκεκριμένη περίπτωση, το βασικό ζητούμενο είναι η εφαρμογή του μοντέλου σε ροή στιγμιοτύπων (βίντεο) πραγματικού χρόνου. Έτσι η ουσιαστική αξιολόγηση του μοντέλου πρέπει να γίνει με πρακτικές δοκιμές, δοκιμάζοντάς το σε διάφορες συνθήκες, όπως σε διαφορετικό φωτισμό, διαφορετική γωνία λήψης κ.λπ.

Απ'ότι παρατηρήσαμε στις δοκιμές, το μοντέλο SVM έχει καλή ακρίβεια και αναγνωρίζει ικανοποιητικά τις χειρονομίες. Γι'αυτό άλλωστε και σε παρόμοια προβλήματα αναγνώρισης χειρονομιών χρησιμοποιούνται κατά κύριο λόγο Μηχανές Διανυσμάτων Υποστήριξης. Επίσης ο χρόνος εκπαίδευσης είναι πολύ μικρός καθώς και ο χρόνος που απαιτεί το μοντέλο για την αναγνώριση σε πραγματικό χρόνο.

Οι χειρονομίες του συνόλου δεδομένων εκτελούνται γενικά από μια σταθερή θέση. Ο χρήστης είναι καθιστός σε ένα γραφείο και η κάμερα είναι στην οθόνη που έχει μπροστά του, δηλαδή περίπου στο ύψος του λαιμού. Επίσης, οι χειρονομίες εκτελούνται με το δεξί χέρι, απεικονίζονται κατά κύριο λόγο δεξιότερα από το μέσο της εικόνας, ως προς την οπτική του χρήστη που τις εκτελεί. Όμως, αλγόριθμος του MediaPipe εξάγει τις συντεταγμένες των σημείων του χεριού κανονικοποιημένες ως προς τις συνολικές διαστάσεις τις εικόνας. Αυτό έχει ως αποτέλεσμα το μοντέλο να έχει υψηλή ακρίβεια, όταν οι χειρονομίες εκτελούνται από παρόμοια θέση με αυτή των δεδομένων εισόδου, αλλά χαμηλότερη ακρίβεια σε αντίθετη περίπτωση.

4.4 Κατηγοριοποίηση χειρονομιών με Τεχνητά Νευρωνικά Δίκτυα

Για τη χρήση Τεχνητών Νευρωνικών Δικτύων χρησιμοποιήθηκε η βιβλιοθήκη tensorflow-keras μέσω της οποίας εκπαιδεύσαμε ένα Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης (Feedforward Neural Network). Το δίκτυό μας αποτελείται από δύο πλήρως συνδεδεμένα στρώματα (Dense Layers), δύο στρώματα απόρριψης νευρώνων (Dropout Layers) ανάμεσά τους, και ένα τελικό dense στρώμα που αποτελεί και την έξοδο του δικτύου, όπως φαίνεται στο παρακάτω σχήμα:



Εικόνα 4-8: Η γενική αρχιτεκτονική του Νευρωνικού Δικτύου

4.4.1 Επιλογή υπερπαραμέτρων

Για την επιλογή του βέλτιστου δικτύου απαιτούνται τεχνικές εύρεσης του συνδυασμού των υπερπαραμέτρων που επιτυγχάνει την καλύτερη ακρίβεια.

Οι υπερπαράμετροι (hyperparameters) που μας ενδιαφέρουν για το συγκεκριμένο δίκτυο είναι οι εξής:

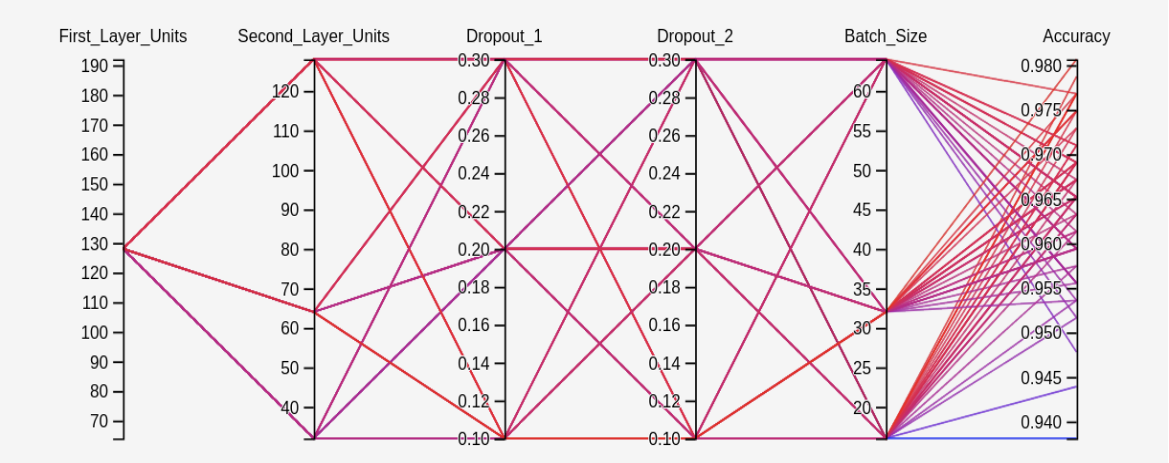
- Πλήθος νευρώνων για τα ενδιάμεσα στρώματα του δικτύου
- Ποσοστό νευρώνων προς απόρριψη από τα στρώματα Dropout
- Η συνάρτηση βελτιστοποίησης της μεθόδου
- Η συνάρτηση ενεργοποίησης
- Η συνάρτηση κόστους
- Batch-size
- Πλήθος επαναλήψεων εκπαίδευσης (Epochs)

Συνολικά δηλαδή, έχουμε 7 υπερπαραμέτρους που πρέπει να εξετάσουμε.

Η εξαντλητική δοκιμή όλων των δυνατών συνδυασμών είναι υπολογιστικά μια αρκετά απαιτητική διαδικασία. Αν επιλέξουμε 3 διαφορετικές τιμές για κάθε παράμετρο, πρέπει συνολικά να εκπαιδεύσουμε το μοντέλο 3^7 ή 19.683 φορές. Γι'αυτό κάποιες τις επιλέγουμε εμπειρικά ή σύμφωνα με τα γενικά πρότυπα που ακολουθούνται, και άλλες τις υπολογίζουμε με εργαλεία που αυτοματοποιούν τη διαδικασία.

Εξαντλητική αναζήτηση βέλτιστου συνδυασμού

Για το πλήθος του πρώτου ενδιάμεσου στρώματος και τις συναρτήσεις βελτιστοποίησης και ενεργοποίησης, έγιναν κάποιες δοκιμές και επιλέχθηκαν χειροκίνητα. Για τις υπόλοιπες, έγινε εξαντλητική αναζήτηση αυτόματα χρησιμοποιώντας το εργαλείο HParams [43] που διαθέτει η βιβλιοθήκη tensorflow. Τα αποτελέσματα από τη διαδικασία υπολογισμού των βέλτιστων υπερπαραμέτρων φαίνονται στην παρακάτω εικόνα:

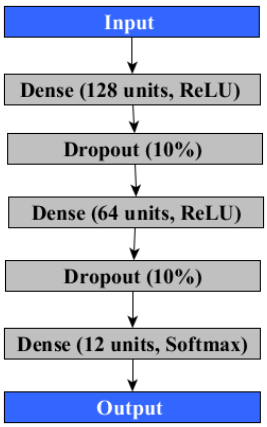


Εικόνα 4-9: Η ακρίβεια του μοντέλου για κάθε συνδυασμό υπερπαραμέτρων

Οι τελικές υπερπαράμετροι που επιλέχθηκαν είναι οι εξής:

- Πλήθος νευρώνων 1ου στρώματος: **128** , Συνάρτηση Ενεργοποίησης : **ReLU**
- Πλήθος νευρώνων 2ου στρώματος: **64** , Συνάρτηση Ενεργοποίησης : **ReLU**
- Dropout 1ου στρώματος: **10%**
- Dropout 2ου στρώματος: **10%**
- Batch Size: **32**
- Αλγόριθμος βελτιστοποίησης: **Adam**
- Συνάρτηση Ενεργοποίησης τελικού στρώματος: **Softmax**
- Epochs: **150**
- Συνάρτηση κόστους : **Categorical Crossentropy**

Και η αρχιτεκτονική του νευρωνικού δικτύου σχηματικά:



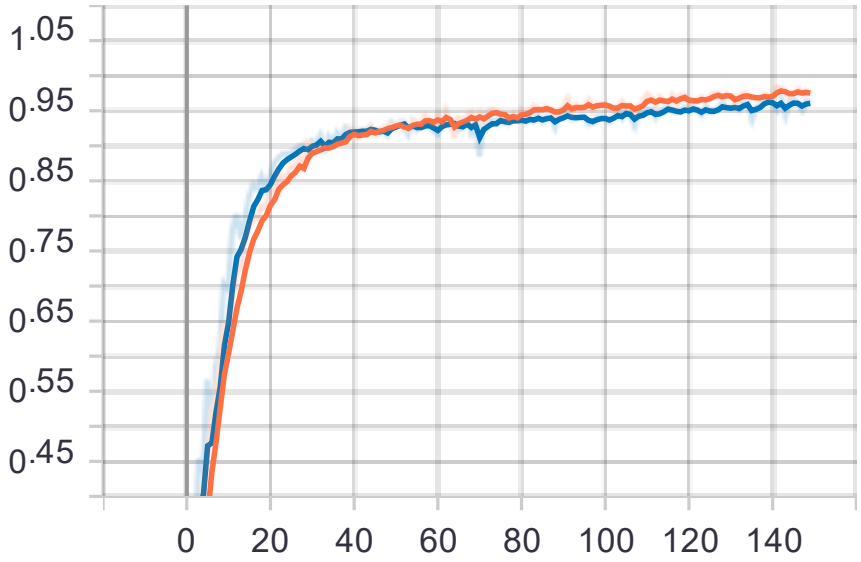
Εικόνα 4-10: Το τελικό Νευρωνικό Δίκτυο

4.4.2 Εκπαίδευση μοντέλου

Εκπαιδεύοντας το νευρωνικό δίκτυο προέκυψαν τα εξής αποτελέσματα για την ακρίβεια:

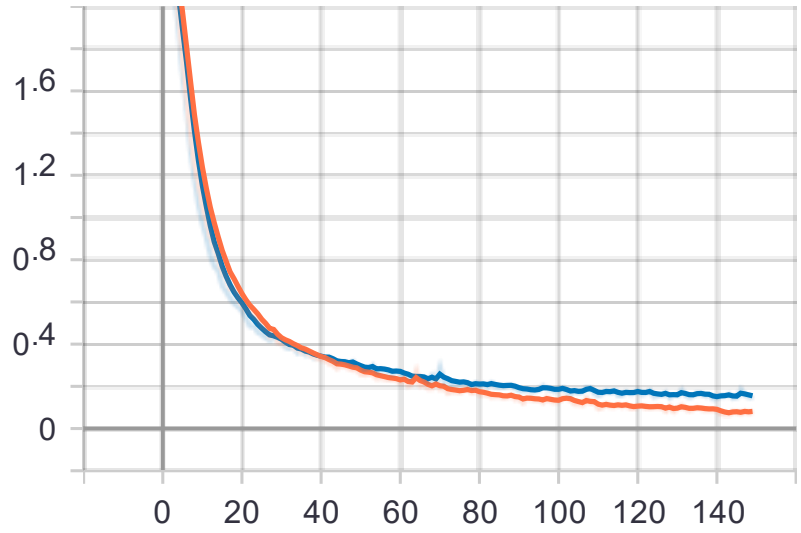
- Σύνολο εκπαίδευσης : **98.8%**
- Σύνολο ελέγχου: **96.2%**

Η ακρίβεια ανά επανάληψη απεικονίζεται στο παρακάτω γράφημα:



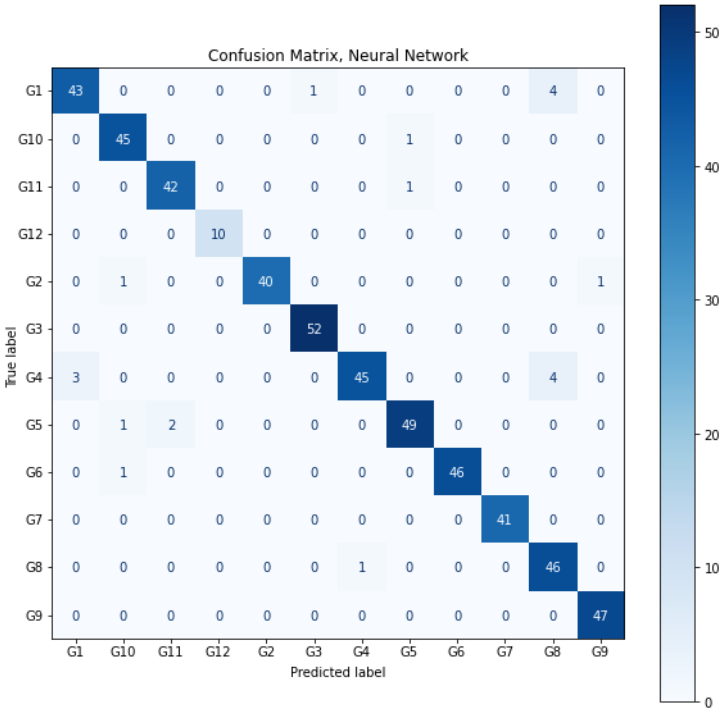
Εικόνα 4-11: Ακρίβεια του μοντέλου με Νευρωνικά Δίκτυα κατά τη διάρκεια εκπαίδευσης. Με σκούρο μπλε απεικονίζεται το σύνολο ελέγχου και με πορτοκαλί το σύνολο εκπαίδευσης.

Και το σφάλμα ανά επανάληψη :

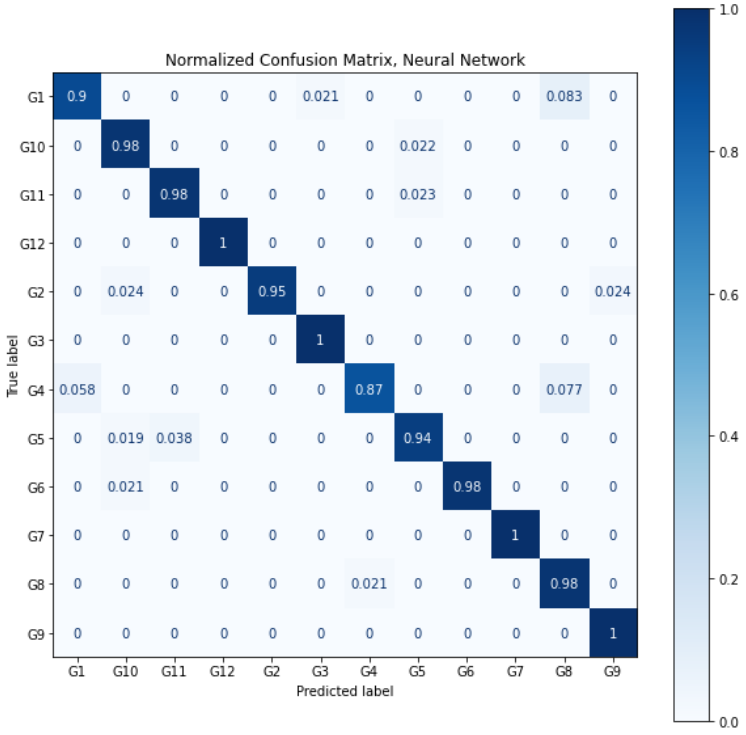


Εικόνα 4-12: Σφάλμα του μοντέλου με Νευρωνικά Δίκτυα κατά τη διάρκεια εκπαίδευσης. Με σκούρο μπλε απεικονίζεται το σύνολο ελέγχου και με πορτοκαλί το σύνολο εκπαίδευσης.

Ο πίνακας σύγχυσης και ο κανονικοποιημένος πίνακας σύγχυσης φαίνεται παρακάτω:



Εικόνα 4-13: Πίνακας Σύγχυσης, Νευρωνικό Δίκτυο



Εικόνα 4-14: Κανονικοποιημένος Πίνακας Σύγχυσης, Νευρωνικό Δίκτυο

4.4.2 Αξιολόγηση

Εδώ πρέπει να επισημάνουμε το εξής. Η βιβλιοθήκη tensorflow προσφέρει τη δυνατότητα να αξιοποιήσουμε όλους τους διαθέσιμους πυρήνες του υπολογιστή (multiprocessing). Στην περίπτωση αυτή το μοντέλο θα αναγνωρίζει μεν τις χειρονομίες ταχύτερα, αλλά θα ήταν δύσκολη η χρήση του παράλληλα με άλλα προγράμματα του υπολογιστή, καθώς θα αντλούσε όλη τη διαθέσιμη υπολογιστική ισχύ. Εμείς επιλέγουμε να χρησιμοποιήσουμε έναν μόνο πυρήνα.

Το μοντέλο, όπως φαίνεται και στα παραπάνω διαγράμματα, επιτυγχάνει υψηλή ακρίβεια κατηγοριοποίησης και στις δοκιμές που κάναμε είχε πολύ ικανοποιητική απόδοση. Φαίνεται να γενικεύει αρκετά καλύτερα τη γνώση που μαθαίνει συγκριτικά με το μοντέλο SVM και μπορεί να αναγνωρίσει χειρονομίες υπό αρκετά διαφορετική γωνία εκτέλεσης. Επίσης, αναγνωρίζει και χειρονομίες που εκτελούνται σε διαφορετικές περιοχές της εικόνας, συγκριτικά με αυτές του αρχικού συνόλου δεδομένων.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία εξετάσαμε το πρόβλημα της αναγνώρισης στατικών χειρονομιών σε βίντεο πραγματικού χρόνου. Σκοπός ήταν να υλοποιήσουμε ένα σύστημα που να αναγνωρίζει προκαθορισμένες στατικές χειρονομίες, χρησιμοποιώντας έναν υπολογιστή και μια απλή κάμερα. Όπως διαπιστώσαμε, εφαρμόζοντας σύγχρονες τεχνικές και αλγορίθμους της Μηχανικής Μάθησης και της Βαθιάς Μάθησης, μπορούμε να αναγνωρίσουμε χειρονομίες με μεγάλη ακρίβεια και αποδοτικά, χωρίς να απαιτείται ισχυρή υπολογιστική ισχύ ή ακριβός εξοπλισμός.

Για την αναγνώριση μιας χειρονομίας στην υλοποίησή μας, εκτελούνται δύο κεντρικές διεργασίες. Ο εντοπισμός του χεριού στην εικόνα και η αναγνώριση της χειρονομίας που εκτελείται. Για τον εντοπισμό του χεριού, βασιστήκαμε σε έναν καινοτόμο αλγόριθμο (MediaPipe Hands) που χρησιμοποιεί συνελικτικά νευρωνικά δίκτυα και παλινδρόμηση, για τον εντοπισμό 21 σημείων του σκελετού του χεριού. Για την αναγνώριση των χειρονομιών, εκπαιδεύσαμε δύο μοντέλα κατηγοριοποίησης, μια μηχανή διανυσμάτων υποστήριξης (SVM) και ένα τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης.

Ο αλγόριθμος του MediaPipe, όπως παρατηρήσαμε, έχει εξαιρετική ακρίβεια αναγνώρισης του χεριού και λειτουργεί πολύ καλά ακόμα και σε ακραίες περιπτώσεις, όπως όταν υπάρχει υπερβολική φωτεινότητα ή όταν το χέρι δεν φαίνεται καλά στην εικόνα. Ο αλγόριθμος μπορεί να εφαρμοστεί σε όλων των ειδών τα συστήματα εκτός του υπολογιστή, όπως οι φορητές συσκευές.

Για το μοντέλο SVM η διαδικασία εκπαίδευσης και χρήσης του ήταν αρκετά εύκολη και σύντομη. Παρατηρήσαμε ότι η χρήση της συνάρτησης rbf ως πυρήνα είναι μια πολύ καλή επιλογή και επιτρέπει στον αλγόριθμο να διαχωρίζει τα δεδομένα επιτυχώς. Στην πράξη, μπορεί να υπάρξει σύγχυση του μοντέλου κατά την αναγνώριση χειρονομιών που μοιάζουν πολύ μεταξύ τους και γι’αυτό προτείνεται η χρήση του με μικρό πλήθος χειρονομιών, κατά προτίμηση αισθητά διαφορετικών. Επίσης, οι μεγάλες αλλαγές στη γωνία λήψης καθώς και σε άλλες παραμέτρους του περιβάλλοντος, επηρεάζουν σημαντικά την ακρίβεια αναγνώρισης. Τέλος, ο αλγόριθμος του SVM έχει χαμηλές απαιτήσεις μνήμης και αυτό τον καθιστά πολύ καλή επιλογή για εφαρμογές σε απλά υπολογιστικά συστήματα. Για το νευρωνικό δίκτυο, ο σχεδιασμός του και η εύρεση των κατάλληλων τιμών για τις υπερπαραμέτρους του, όπως είδαμε είναι μια χρονοβόρα διαδικασία και μπορεί να απαιτήσει μεγάλο πλήθος δοκιμών. Οι τελικές επιλογές που κάναμε φαίνεται να αποδίδουν καλά και να επιτυγχάνουν πολύ υψηλά ποσοστά ακρίβειας. Το μοντέλο έχει μεγάλη ανοχή σε κακές συνθήκες εκτέλεσης των χειρονομιών και μπορεί να διακρίνει ικανοποιητικά όλες τις χειρονομίες, ακόμα κι αν μοιάζουν πολύ μεταξύ τους.

Σημαντικό παράγοντα του προβλήματος αποτελεί το σύνολο δεδομένων που χρησιμοποιούμε. Στη δική μας εργασία κατασκευάσαμε ένα μικρό σχετικά σύνολο, το οποίο όμως αποδείχθηκε αρκετό για να επιτύχουμε ικανοποιητικά αποτελέσματα. Στο σύνολο προσθέσαμε και μια νέα χειρονομία χρησιμοποιώντας λίγα μόνο δείγματα και παρατηρήσαμε ότι τα μοντέλα είχαν αρκετά καλή ακρίβεια αναγνώρισης. Αυτό είναι σημαντικό αποτέλεσμα καθώς δείχνει ότι δεν απαιτείται μεγάλο πλήθος εικόνων για την εκπαίδευση των μοντέλων σε νέες χειρονομίες.

Επίσης, ο αλγόριθμος εντοπισμού των σημείων του χεριού υπολογίζει τις συντεταγμένες ως προς τις συνολικές διαστάσεις τις εικόνας και όχι ως προς την περιοχή που βρίσκεται το χέρι (bounding box). Αυτό έχει ως αποτέλεσμα το μοντέλο SVM να μαθαίνει να αναγνωρίζει κυρίως χειρονομίες που εκτελούνται από παρόμοια θέση με αυτές που το εκπαιδεύσαμε και να έχει χαμηλότερη ακρίβεια αλλού. Αντίθετα, το μοντέλο των νευρωνικών δικτύων ξεπερνάει αυτό το ζήτημα και γενικεύει πολύ καλύτερα τη γνώση που μαθαίνει, αναγνωρίζοντας επιτυχώς χειρονομίες σε όλες τις περιοχές τις εικόνας.

Συνοψίζοντας, κάποια βασικά συμπεράσματα που προκύπτουν για τα δύο μοντέλα είναι:

Μηχανή Διανυσμάτων Υποστήριξης:

- Απαιτείται ελάχιστος χρόνος εκπαίδευσης και εφαρμογής του μοντέλου
- Προτείνεται για χρήση σε συστήματα με χαμηλή υπολογιστική ισχύ
- Πολύ καλή ακρίβεια όταν οι χειρονομίες εκτελούνται από παρόμοιες θέσεις με αυτές του συνόλου δεδομένων, χαμηλότερη σε αντίθετη περίπτωση
- Η αλλαγή της κλίσης του χεριού κατά την εκτέλεση των χειρονομιών μειώνει την αποτελεσματικότητά του

Νευρωνικό Δίκτυο:

- Απαιτείται σχετικά μικρός χρόνος εκπαίδευσης και εφαρμογής
- Εξαιρετική ακρίβεια αναγνώρισης των χειρονομιών, σε όλα τα σημεία της εικόνας
- Μεγάλη ανοχή σε αλλαγές της κλίσης του χεριού κατά την εκτέλεση των χειρονομιών
- Συνιστάται για χρήση με υπολογιστή ή με συσκευές που διαθέτουν σχετικά σύγχρονους επεξεργαστές

Κλείνοντας, αναφέρουμε ενδεικτικά κάποια παραδείγματα εφαρμογών που μπορούν να επιλυθούν χρησιμοποιώντας τη μεθοδολογία που περιγράψαμε:

- Αυτόματη αναγνώριση της νοηματικής γλώσσας
- Έλεγχος ηλεκτρονικών συσκευών και μηχανημάτων μέσω κινήσεων του χεριού
- Χρήση χειρονομιών ως προσωπικός κωδικός για ξεκλείδωμα συσκευών
- Ανάλυση ανθρώπινων δραστηριοτήτων που χρησιμοποιούνται οι κινήσεις των χεριών

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Το γάντι acceleGlove
https://netdna.coolthings.com/wp-content/uploads/2009/07/3acceleglove.jpg

[2] MediaPipe Hands
https://google.github.io/mediapipe/solutions/hands.html

[3], [5] Supervised and Unsupervised Learning
https://data-flair.training/blogs/types-of-machine-learning-algorithms

[4] Classification vs Regression
https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

[6] Reinforcement Learning
https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

[7] Spam Classification
https://medium.com/@naveeen.kumar.k/naive-bayes-spam-detection-7d087cc96d9d

[8] Confusion Matrix
https://en.wikipedia.org/wiki/Confusion_matrix

[9] SVM Classification
https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3

[10] Maximum Margin Hyperplane
https://miro.medium.com/max/600/1*FGzasG-84SHk_XXSClcoqA.png

[11] Support Vector Machine
https://en.wikipedia.org/wiki/Support-vector_machine

[12], [14] Biological Neural Network
https://en.wikipedia.org/wiki/Artificial_neural_network

[13] Artificial Neuron
https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/

[15] Convolutional Neural Network
https://www.researchgate.net/publication/336805909/figure/fig1/AS:817888827023360@1572011300751/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26.ppm

[16] Recurrent Neural Network
https://en.wikipedia.org/wiki/Recurrent_neural_network

[17] Machine Learning: How to Prevent Overfitting
https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9

[18] Training, validation, and test sets
https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

[17] Sigmoid Function
https://en.wikipedia.org/wiki/Sigmoid_function

[18] Logistic Function
https://en.wikipedia.org/wiki/Logistic_function

[19] Hyperbolic tangent
https://mathworld.wolfram.com/HyperbolicTangent.html

[20]
Dropout: A Simple Way to Prevent Neural Networks from Overfitting
Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov;
15(56):1929−1958, 2014.
https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

[21] Rectified Linear Unit
https://ychai.uk/notes/2019/05/10/NN/Activation-functions-introduction/

[22] Adagrad algorithm
Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12, 2121−2159. Retrieved from http://jmlr.org/papers/v12/duchi11a.html

[23] Which Optimizer to use
https://ruder.io/optimizing-gradient-descent/index.html#fnref9

[24] scikit-learn Classification
https://scikit-learn.org/stable/modules/svm.html

[25] Softmax Function
https://en.wikipedia.org/wiki/Softmax_function

[26] Region Based Convolutional Neural Networks
https://en.wikipedia.org/wiki/Region_Based_Convolutional_Neural_Networks

[27] R-CNN: Rich feature hierarchies for accurate object detection and semantic segmentation
https://arxiv.org/pdf/1311.2524.pdf

[28] Fast R-CNN
https://arxiv.org/pdf/1504.08083.pdf

[29] Faster R-CNN: Towards Real-Time ObjectDetection with Region Proposal Networks
https://arxiv.org/pdf/1506.01497.pdf

[30] You Only Look Once:Unified, Real-Time Object Detection
https://arxiv.org/pdf/1506.02640v5.pdf

[31] SSD: Single Shot MultiBox Detector
https://arxiv.org/pdf/1512.02325.pdf

[32] Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG-16)
https://arxiv.org/pdf/1409.1556.pdf

[33] Feature Pyramid Networks for Object Detection
https://arxiv.org/pdf/1612.03144.pdf

[34] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He,and Piotr Doll´ar.Focal loss for dense object detection.CoRR, abs/1708.02002, 2017.

[35] MediaPipe Framework
https://google.github.io/mediapipe/

[36] MediaPipe Hands: On-device Real-time Hand Tracking
https://arxiv.org/pdf/2006.10214.pdf

[37] VGG16 – Convolutional Network for Classification and Detection
https://neurohive.io/en/popular-networks/vgg16/

[38] PASCAL Visual Object Classes Challenge 2007
http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html

[39] A. Memo, L. Minto, P. Zanuttigh, "Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition", STAG: Smart Tools & Apps for Graphics, 2015

[40] A. Memo, P. Zanuttigh, "Head-mounted gesture controlled interface for human-computer interaction", Multimedia Tools and Applications, 2017

[41] Creative Senz3D Dataset
https://lttm.dei.unipd.it/downloads/gesture/#senz3d

[42] Scikit-Learn SVM Classifier
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[43] Hyperparameter tuning using HParams
https://www.tensorflow.org/tensorboard/hyperparameter_tuning_with_hparams