



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη συστήματος αυτόματου διαχωρισμού
ανακυκλώσιμων φιαλών και ποτηριών με χρήση βαθιάς
μάθησης στη μηχανική όραση και ακοή»

Του φοιτητή
Κατάκη Χριστόφορου
Αρ. Μητρώου:52007m

Επιβλέπων
Αργύρης Χατζόπουλος
Επίκουρος καθηγητής

Φεβρουάριος 2024

Τίτλος Δ.Ε. Ανάπτυξη συστήματος αυτόματου διαχωρισμού ανακυκλώσιμων φιαλών και ποτηριών με χρήση βαθιάς μάθησης στη μηχανική όραση και ακοή

Κωδικός Δ.Ε. 23313

Όνοματεπώνυμο φοιτητή Χριστόφορος Καλτάκης

Όνοματεπώνυμο εισηγητή Αργύρης Χατζόπουλος

Ημερομηνία ανάληψης Δ.Ε. 04-11-2023

Ημερομηνία περάτωσης Δ.Ε. 04-03-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα» στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Καλτάκη Χριστόφορου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένεια μου, τους καθηγητές μου και τους φίλους μου»

Πρόλογος

Έπειτα από συζήτηση, με τον Αργύρη Χατζόπουλο Επίκουρο Καθηγητή στο τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος (πρώην τμήμα Μηχανικών Ηλεκτρονικής ΤΕ– Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ινστιτούτο Θεσσαλονίκης) και μέλος του Εργαστηρίου Προηγμένων Ηλεκτρονικών Συστημάτων, για ανάληψη διπλωματική εργασίας στα πλαίσια του Μεταπτυχιακού Προγράμματος Σπουδών που διοργανώνει το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου με τίτλο “Εφαρμοσμένα Ηλεκτρονικά Συστήματα” και λόγω της επαγγελματικής ενασχόλησης του μεταπτυχιακού φοιτητή στη δημιουργία ΙΟΤ συσκευών για τη διαχείριση των απορριμμάτων, αλλά και της θέλησης του για γνωριμία με νέους τομείς της επιστήμης, αποφασιστικέ η δημιουργία ενός συστήματος διαλογής απορριμμάτων το οποίο που θα χρησιμοποιεί Τεχνητή Νοημοσύνη. Η εμπλοκή του φοιτητή με τον τομέα της Τεχνητής Νοημοσύνης είχε ως αποτέλεσμα, να αποκτήσει γνώση και εμπειρία πάνω στον υποτομέα της Τ.Ν, τη μηχανική μάθηση και συγκεκριμένα στα Συνελικτικά Νευρωνικά Δίκτυα, στη μηχανική όραση και τη μηχανική ακοή.

Περίληψη

Η παρούσα διπλωματική εργασία επικεντρώνεται στη δημιουργία ενός συστήματος αυτόματου διαχωρισμού ανακυκλώσιμων φιαλών και ποτηριών με τη χρήση Τεχνητής Νοημοσύνης με δύο αισθήσεις, τη μηχανική όραση και ακοή, το σύστημα αυτό αναπτύχθηκε για να συμβάλει στη χρονική αλλά και ποιοτική βελτίωση της διαλογής των προαναφερθέντων ανακυκλώσιμων υλικών στη πηγή. Η εφαρμογή αξιοποιεί τις τεχνικές μηχανικής οράσεως και ακοής για την αναγνώριση και κατηγοριοποίηση των αντικειμένων μέσω βαθιάς μάθησης. Αποτελείται από ένα σωλήνα οπού απορρίπτονται τα υλικά προς διαχωρισμό, από ένα Η/Υ όπου έχει στηθεί η εφαρμογή της μηχανικής μάθησης και μία αναπτυσσόμενη πλακέτα arduino η οποία είναι τοποθετημένη πάνω σε ένα ξύλινο φορείο και χρησιμοποιείται για τις κινήσεις που χρειάζονται να γίνουν για να γίνει η ταξινόμηση των υλικών. Στο φορείο βρίσκεται επίσης ένα ηλεκτρομαγνητικό έμβολο ελεγχόμενο από το arduino το οποίο χρησιμοποιείται για να χτυπήσει το αντικείμενο ώστε να παραχθεί ο ήχος και ένα μικρόφωνο για τη συλλογή του ήχου. Στην άλλη άκρη βρίσκεται μια κάμερα η οποία μαζί με το μικρόφωνο συνδέονται στον Η/Υ και συλλέγουν την εικόνα και τον ήχο, από τα αντικείμενα που έχει τοποθετηθεί στον σωλήνα, τα οποία αποτελούν τα δεδομένα τα οποία χρησιμοποιεί η εφαρμογή για να εξάγει το συμπέρασμα της κλάσης που ανήκει το αντικείμενο. Τα αποτελέσματα της έρευνας για το συγκεκριμένο σύστημα έδειξαν ότι συνδυάζοντας και τον ήχο και την εικόνα αυξάνεται το ποσοστό της επιτυχημένης πρόβλεψης.

«Development of an automatic sorting system for recyclable bottles and glasses using deep learning in machine vision and hearing»

«Christoforos & Kaltakis»

Abstract

This thesis is focused on the development of an automatic separation system that separates bottles and glasses made from different materials, using Artificial intelligence with two senses, machine vision and machine hearing. This system is developed to contribute to the time and quality improvement of the shorting of recyclable materials in source. The application leverages machine vision and hearing techniques to recognize and categorize objects through deep learning. It consists of a pipe where the materials to be shorting are discarded, it has a PC where the machine learning application is set up and an arduino development board which is placed on a wooden moving platform and is used for the movements that need to be done, so that the system can make the classification of materials. Also on the moving platform is an electromagnetic piston controlled by the arduino which it used to hit the object to produce the sound and there is too a microphone to collect the sound. At the other end it is placed a camera which, together with the microphone, connects to the PC and collects the image and sound from the objects placed in the tube, which are the data that the application uses to make the decision in which class of material the object belongs. The results of the research on the specific system showed that combining both sound and image increases the percentage of a successful prediction.

Ευχαριστίες

Ευχαριστώ πολύ την οικογένεια μου για τη στήριξη που μου έχουν παρέχει όλα αυτά τα χρόνια και ιδιαίτερα τη σύζυγο μου Ιωάννα και την κόρη μου Θεοδώρα για την υπομονή που έχουν κάνει. Ευχαριστώ επίσης τον Αργύρη Χατζόπουλο για την εμπιστοσύνη και την έμπνευση, ευχαριστώ και όλους τους καθηγητές του πρώην τμήματος Μηχανικών Ηλεκτρονικής ΤΕ και του Μεταπτυχιακού Προγράμματος Εφαρμοσμένα Ηλεκτρονικά Συστήματα του ΔΠΙΑΕ για της γνώσης που μας προσέφεραν.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων.....	xii
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	14
Κεφάλαιο 2ο: Θεωρητικό Μέρος.....	15
2.1 Εισαγωγή.....	15
2.2 Το πρόβλημα της διαλογής των ανακυκλώσιμων υλικών.....	15
2.3 Η Νοημοσύνη.....	15
2.3.1 Η Τεχνητή Νοημοσύνη	17
2.3.2 Η Ιστορική Εξέλιξη της Τεχνητής Νοημοσύνης.....	19
2.3.3 Εφαρμογές της Τεχνητής Νοημοσύνης.....	23
2.4 Μάθηση.....	23
2.4.1 Η μηχανική μάθηση.....	23
2.5 Τα Τεχνητά Νευρωνικά Δίκτυα.....	27
2.5.1 Ο βιολογικός νευρώνας	27
2.5.2 Λειτουργία ενός βιολογικού νευρώνα.....	28
2.5.3 Η δομή του τεχνητού νευρώνα.....	28
2.5.4 Συναρτήσεις ενεργοποίησης.....	30
2.5.5 Πόλωση	33
2.6 Νευρωνικά Δίκτυα	34
2.6.1 Βαθιά νευρωνικά δίκτυα	35
2.6.2 Συνελκτικά νευρωνικά δίκτυα.....	36
2.6.3 Στάδια επεξεργασίας των δεδομένων στα CNNs	37
2.7 Αξιολόγηση των CNNs	43
2.7.1 Πίνακας σύγχυσης (Confusion Matrix)	45
2.7.2 Προβλήματα στη μάθηση των CNNs.....	47
2.7.3 Αντιμετώπιση προβλημάτων	47

2.8	Επίλογος.....	48
Κεφάλαιο 3ο: Πειραματική διαδικασία.....		49
3.1	Εισαγωγή.....	49
3.2	Η εικόνα	49
3.3	Επεξεργασία στις εικόνες.....	50
3.4	Ο ήχος.....	51
3.5	Επεξεργασία ήχου	52
3.6	Η γλώσσα προγραμματισμού Python.....	56
3.7	Βιβλιοθήκες της Python που χρησιμοποιηθήκανε	56
3.8	Μετρήσεις και αποτελέσματα	57
3.8.1	Υλικό που χρησιμοποιήθηκε	57
3.8.2	Μοντέλα και αποτελέσματα για εικόνες	58
3.8.3	Ταξινομητές ήχου.....	65
3.8.4	Συνδυασμός των δύο αισθήσεων.....	68
3.9	Κατασκευή για φυσική αναγνώριση	69
3.9.1	Arduino & Arduino Uno	70
3.9.2	Οδηγός βηματικού κινητήρα.....	71
3.9.3	Ο βηματικός κινητήρας	72
3.9.4	Ηλεκτρομαγνητικό έμβολο.....	75
3.9.5	Mosfet σε λειτουργία διακόπτη	76
3.9.6	Κάμερα.....	77
3.9.7	Μικρόφωνο.....	78
3.9.8	Λειτουργία του συστήματος	78
Κεφάλαιο 4ο: Συμπεράσματα.....		83
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		84
ΠΑΡΑΡΤΗΜΑ Α : Χρήσιμα προγράμματα σε python.....		85

Κατάλογος Σχημάτων

Σχήμα 2.1 Οι τομείς τις Τεχνητής Νοημοσύνης.....	25
Σχήμα 2.2 Βιολογικοί νευρώνες με σύναψη	27
Σχήμα 2.3 Δομή Τεχνητού Νευρώνα	29
Σχήμα 2.4 Σιγμοειδής συνάρτηση.....	30
Σχήμα 2.5 Υπερβολική εφαπτομένη	31
Σχήμα 2.6 Γραμμική.....	31
Σχήμα 2.7 Ανορθωμένη γραμμική μονάδα	32
Σχήμα 2.8 Softmax	33
Σχήμα 2.9 Μετατόπιση της συνάρτησης ενεργοποίησης λόγω πόλωσης.....	34
Σχήμα 2.10 Αρχιτεκτονική ρηχών Νευρωνικών Δικτύων.....	35
Σχήμα 2.11 Βαθύ Νευρωνικό Δίκτυο με 5 κρυφά επίπεδα	36
Σχήμα 2.12 Είσοδος φίλτρο και αποτέλεσμα της συνέλιξης.....	38
Σχήμα 2.13 Είσοδος, φίλτρο και το αποτέλεσμα της διασταυρούμενης συσχέτισης.....	38
Σχήμα 2.14 Εφαρμογή συνέλιξης.....	39
Σχήμα 2.15 Συνέλιξη σε έγχρωμη εικόνα	40
Σχήμα 2.16 Πίνακα διάστασης 4x4.....	41
Σχήμα 2.17 Πίνακας 2x2 αποτέλεσμα του maxpooling	41
Σχήμα 2.18 Πίνακας 2x2 αποτέλεσμα average pooling	42
Σχήμα 2.19 Μία μόνο τιμή αποτέλεσμα του Global average pooling.....	42
Σχήμα 2.20 Παράδειγμα πίνακα σύγχυσης 2 κλάσεων	46
Σχήμα 2.21 Παράδειγμα πίνακα σύγχυσης 3 κλάσεων	47
Σχήμα 3.1 Εμφάνιση των τριών καναλιών.....	50
Σχήμα 3.2 Πίνακας (εικόνας) 4x4x3	50
Σχήμα 3.3 Στάδια επεξεργασίας.....	51
Σχήμα 3.4 Αναλογικό σε ψηφιακό	52
Σχήμα 3.5 Διαγράμματα ήχων στο πεδίο του χρόνου	53
Σχήμα 3.6 Διάγραμμα ήχου μετά από μετασχηματισμό Fourier.....	54
Σχήμα 3.7 Σημα ήχου πριν και μετά το φιλτράρισμα	54
Σχήμα 3.8 Μετασχηματισμός STFT	55
Σχήμα 3.9 Απεικόνιση του Mel-filterbank.....	55
Σχήμα 3.10 MFCC από αλουμίνιο και γυαλί	56
Σχήμα 3.11 Confusion Matrix από το πρώτο μοντέλο	59
Σχήμα 3.12 Vgg19 Confusion Matrix	61
Σχήμα 3.13 Προβλέψεις Vgg19	61
Σχήμα 3.14 Πίνακας σύγχυσης InceptionResNet	63
Σχήμα 3.15 Δείγμα εικόνων από την πρόβλεψη	63
Σχήμα 3.16 Πίνακας σύγχυσης ConvNeXtXLarge	65
Σχήμα 3.17 Προβλέψεις από το μοντέλο ConvNeXtXLarge	65
Σχήμα 3.18 Πίνακας σύγχυσης πρώτου ταξινομητή ήχου	67
Σχήμα 3.19 Πίνακας σύγχυσης δεύτερου ταξινομητή ήχου.....	68
Σχήμα 3.20 Το σύστημα διάλογης	69
Σχήμα 3.21 Arduino uno	70
Σχήμα 3.22 EasyDriver	72
Σχήμα 3.23 Nema 17	73

Σχήμα 3.24 Εσωτερικό βηματικού κινητήρα	73
Σχήμα 3.25 Full step.....	74
Σχήμα 3.26 Half step.....	74
Σχήμα 3.27 Ηλεκτρομαγνητικό έμβολο	76
Σχήμα 3.28 Mosfet ως διακόπτης.....	77
Σχήμα 3.29 Κάμερα Razer Kiyo	77
Σχήμα 3.30 Το μικρόφωνο Trust GX212	78
Σχήμα 3.31 Το σύστημα διαλογής	79
Σχήμα 3.32 Κάμερα απέναντι από το υλικό.....	80
Σχήμα 3.33 Πρόβλεψη στην εικόνα από τον ταξινομητή	80
Σχήμα 3.34 Συλλογή του ήχου	81
Σχήμα 3.35 Πρόβλεψη ήχου, εικόνας και συνδυαστική	82

Κατάλογος Πινάκων

Πίνακας 3-1 Πρώτο μοντέλο CNN1 που σχεδιαστικέ για ταξινόμηση των εικόνων.....	58
Πίνακας 3-2 Δομή του μοντέλου CNN Vgg19	60
Πίνακας 3-3 Δομή του μοντέλου InceptionResNet.....	62
Πίνακας 3-4 Αρχιτεκτονική του Convnexxt xlarge	64
Πίνακας 3-5 Δομή CNN για ήχο	66
Πίνακας 3-6 Δεύτερο CNN για ήχο	67
Πίνακας 3-7 Παράδειγμα συνδυασμού πιθανοτήτων πρόβλεψης μοντέλων	69
Πίνακας 3-8 Χαρακτηριστικά του Arduino uno	71
Πίνακας 3-9 Χαρακτηριστικά Nema 17.....	75
Πίνακας 3-10 Χαρακτηριστικά ηλεκτρομαγνητικού εμβόλου.....	76
Πίνακας 3-11 Τα χαρακτηριστικά του IRLR3636U	76
Πίνακας 3-12 Χαρακτηριστικά της κάμερας	77

Συντομογραφίες

T.N	Τεχνητή Νοημοσύνη
M.M	Μηχανική Μάθηση
CNN.	Convolution Neural Networks
E.E	Ευρωπαϊκή Ένωση
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
FClass	False Class
TClass	True Class
NASNet	Neural Architecture Search Network
Vgg19	Visual Geometry Group
ResNet	Residual Network
H/Y	Ηλεκτρονικός Υπολογιστής

Κεφάλαιο 1ο: Εισαγωγή

Η εκτινασόμενη ανάπτυξη της τεχνολογίας έφερε και την κατακόρυφη άνοδο του βιοτικού επιπέδου στις περισσότερες χώρες του πλανήτη. Η αύξηση του βιοτικού επιπέδου δημιούργησε ένα νέο τρόπο ζωής, στον οποίο ο άνθρωπος “ζει για να καταναλώνει αγαθά” με λίγα λόγια έχει γίνει υπερκαταναλωτής. Λόγω της υπερκατανάλωσης, ο όγκος των απορριμμάτων που δημιουργείται καθημερινά είναι τεράστιος. Έτσι οι προηγμένες χώρες βρίσκονται αντιμέτωπες με την απαιτητική πρόκληση της αποτελεσματικής διαχείρισης αυτών των αποβλήτων. Η προσπάθεια μείωσης του κινδύνου μετάδοσης μεταδοτικών ασθενειών στον άνθρωπο και η προστασία του περιβάλλοντος αποτελούν προτεραιότητες που απαιτούν σοφές και ολοκληρωμένες πρακτικές διαχείρισης απορριμμάτων. Μέσα σε αυτό το πλαίσιο η Ευρωπαϊκή ένωση έχει θεσπίσει αποφασιστικά μέτρα για την διαχείριση των απορριμμάτων, υποχρεώνοντας τα κράτη μέλη να προβούν σε μεταρρυθμίσεις οι οποίες να προωθούν την ανακύκλωση και γενικά την κυκλική οικονομία.

Η Ελλάδα αν και κράτος μέλος της Ε.Ε, δεν τηρεί τα μέτρα που έχουν θεσπιστεί από την Ε.Ε, με αποτέλεσμα να καλείτε κάθε εξάμηνο να πληρώνει πρόστιμα για την μη τήρηση των μέτρων. Λόγω της μη τήρησης των μέτρων της ανακύκλωσης, η χώρα μας το 2022 καλέστηκε να πληρώσει 127εκατομηνια ευρώ.

Η μη τήρηση των μέτρων στη χώρα μας έχει τρεις κύριους παράγοντες, συμφέροντα εταιριών, κρατικές υποδομές και την παιδεία στην ανακύκλωση. Στις κρατικές υποδομές αλλά και τη παιδεία ανήκει ένα από τα προβλήματα που θα αναπτύξουμε σε επόμενο κεφάλαιο και είναι η διαλογή των ανακυκλώσιμων υλικών.

Ο συνδυασμός τεχνολογικών και καινοτόμων λύσεων με βιώσιμες πρακτικές κυκλικής οικονομίας αποτελεί τον δρόμο προς την εξασφάλιση υγιεινών συνθηκών ζωής και την προστασία του περιβάλλοντος για τις μελλοντικές γενιές.

Τα τελευταία χρόνια με την ραγδαία ανάπτυξη της τεχνολογίας εμφανίστηκε και ο τομέας της Τεχνητής Νοημοσύνης, διάφοροι ερευνητές από το τομέα αυτό έχουν εστιάσει την έρευνα τους στη δημιουργία νοήμονών συστημάτων τα οποία συμβάλουν στη διαχείριση των απορριμμάτων μειώνοντας το χρόνο της διαλογής αλλά και αυξάνοντας τη ποιότητα της διαλογής.

Σκοπός της παρούσας εργασίας, είναι να παρουσιάσουμε ένα σύστημα αυτόματου διαχωρισμού ανακυκλώσιμων φιαλών και ποτηριών χρησιμοποιώντας τη Τεχνητή Νοημοσύνη με δύο αισθήσεις, τη τεχνητή όραση και τη τεχνητή ακοή, το σύστημα αυτό θα συμβάλει στη διαλογή των προαναφερθέντων ανακυκλώσιμων υλικών στη πηγή.

Στο πρώτο μέρος της εργασίας αυτής αφιερώθηκε αρκετός χρόνος ώστε να γίνει εκμάθηση των απαραίτητων ορισμών αλλά και των τεχνικών που χρησιμοποιούνται και που θα χρησιμοποιηθούν στη συνέχεια και από εμάς, για τη δημιουργία του συστήματος. Αφιερώθηκε επίσης χρόνος στην εκμάθηση λογισμικού και της γλώσσας προγραμματισμού Python με τις απαραίτητες βιβλιοθήκες της.

Στο δεύτερο μέρος αυτή της εργασίας έγινε αναζήτηση για άλλες έρευνες που έχουν πραγματοποιηθεί στο τομέα της Τεχνητής Νοημοσύνης και κυρίως στον υποτομέα της Μηχανικής Μάθησης, της τεχνητής όρασης και της τεχνητής ακοής, αλλά και στη χρήση αυτών σε συστήματα διαλογής

ανακυκλώσιμων υλικών. Ωστε να λάβουμε τη γνώση για τα είδη υπάρχοντα συστήματα και τους τρόπους που έχουν χρησιμοποιηθεί.

Στο τελευταίο μέρος της εργασίας δημιουργήθηκε η βάση δεδομένων η οποία είναι αναγκαία για τη διεξαγωγή των πειραμάτων. Στη συνέχεια πραγματοποιήθηκαν πειράματα σε κάθε μία από τις δύο αισθήσεις (όραση, ακοή) χωριστά για την εξαγωγή των αποτελεσμάτων τις κάθε μιας στο διαχωρισμό των 4 υλικών. και στο τέλος έγιναν πειράματα στη χρήση και των δύο αισθήσεων για την εξαγωγή των αποτελεσμάτων και τη δημιουργία του συστήματος.

Κεφάλαιο 2ο: Θεωρητικό Μέρος

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται αναφορά στο απαραίτητο θεωρητικό υπόβαθρο αλλά και τις έννοιες που χρειάζεται κάποιος για να μπορέσει να εισαχθεί στο χώρο της Τεχνίτης νοημοσύνης και της βαθιάς μάθησης. Αρχικά γίνεται αναφορά στο πρόβλημα της διαλογής των ανακυκλώσιμων υλικών που οι εργασία προσπαθεί να συμβάλει στην ελάττωση του. Στη συνέχεια παρουσιάζονται η Νοημοσύνη και η μάθηση καθώς και τα τεχνητά νευρωνικά δίκτυα.

2.2 Το πρόβλημα της διαλογής των ανακυκλώσιμων υλικών

Με τον ορό ανακύκλωση εννοούμε τη διαδικασία με την οποία υλικά τα οποία έχουν χρησιμοποιηθεί προηγουμένως περνούν από κάποιες κατεργασίες με σκοπό να μετατραπούν σε νέα προϊόντα προς χρήση. Αυτό συμβάλλει στην μείωση της κατανάλωσης των πόρων και στην μείωση της περιβαλλοντικής μόλυνσης.

Πως όμως γίνεται η ανακύκλωση των υλικών σε πολλά κράτη, όπως και στη χώρα μας; Αναφορικά στην Ελλάδα έχουν μοιραστεί κάδοι χρώματος μπλέ οπού εκεί απορρίπτονται από τους δημότες τα υλικά τα οποία είναι προς ανακύκλωση. Γυαλιά, πλαστικά, χαρτιά, μεταλλικά (αλουμινένια), όλα τα ανακυκλώσιμα υλικά σε ένα και μόνο κάδο. Είναι ο τρόπος που τα πετάμε που τα αναμειγνύει και το καθιστά δύσκολο να τα διαχωρίσουμε στη συνέχεια. Έτσι για ευκολία κυρίως των καταναλωτών (πολίτες) ξοδεύουμε μεγάλα ποσά σε υποδομές στις οποίες γίνεται ο διαχωρισμός των ανακυκλώσιμων υλικών. Χρησιμοποιούμε αυτό τον παράδοξο τρόπο ανακύκλωσης που είναι μη παραγωγικός και μη ποιοτικός (λόγω δυσκολίας στη διαλογή), ενώ υπάρχει και ο προφανής τρόπος και πλέον υποχρεωτικός από την Ε.Ε. να γίνει η διαλογή κατευθείαν από τον καταναλωτή με λίγα λόγια να γίνει διαλογή στη πηγή των απορριμμάτων. Αυτό προϋποθέτει, παιδεία ανακύκλωσης, κάδους και οχήματα συλλογής ξεχωριστά για κάθε υλικό.

2.3 Η Νοημοσύνη

Τι είναι η νοημοσύνη, πως ορίζεται και ποια στοιχεία την συνθέτουν; Λόγω της δυσκολίας του προσδιορισμού του ορισμού τις νοημοσύνης, κατά καιρούς έχουν δοθεί διάφορες ερμηνείες στον ορισμό της νοημοσύνης κάποιες από αυτές:

“Ως η κατευθυνόμενη προς στόχο προσαρμοστική συμπεριφορά ” [1]

“Ως η ικανότητα χειρισμού αφηρημένων εννοιών, επίλυσης προβλημάτων και μάθησης” [2]

Η λέξη νοημοσύνη στον άνθρωπο, δείχνει την ικανότητα του ανθρώπου στο να αποκτά γνώση, να επεξεργάζεται και να εφαρμόζει πράγματα τα οποία του έχει προσφέρει η γνώση, να κατανοεί, να σκέφτεται λογικά και να λύνει προβλήματα με χρήση των παραπάνω.

Σύμφωνα με τον (Howard Earl Gardner,) ,αναπτυξιακού ψυχολόγου, ο οποίος είναι ειδικευμένος στη μελέτη της ανάπτυξης του ανθρώπινου εγκεφάλου και συμπεριφοράς από τη γέννηση έως την ενήλικη ζωή, στο βιβλίο του "Frames of Mind: The Theory of Multiple Intelligences" . Υπάρχουν 8 είδη νοημοσύνης:

1. Λεκτική η αλλιώς γλωσσική Νοημοσύνη:

Αυτό το είδος νοημοσύνης έχει να κάνει με τη δυνατότητα που έχουμε ως άνθρωποι να σκεφτόμαστε και να εκφραζόμαστε μέσω λέξεων είτε γραπτά είτε προφορικά. Αυτή η ικανότητα μας επιτρέπει να κατανοούμε τη σημασία και τη σειρά των λέξεων ώστε να χρησιμοποιούμε τη γλώσσα με αποτελεσματικό τρόπο. Το είδος αυτής της νοημοσύνης είναι πιο εμφανή σε ανθρώπους που ασχολούνται με τη λογοτεχνία και τη επικοινωνία, όπως π.χ ποιητές, δημοσιογράφους.

2. Λογικομαθηματική Νοημοσύνη:

Η λογικομαθηματική νοημοσύνη αναδεικνύει την ικανότητα του ανθρώπου να σκέφτεται και να χρησιμοποιεί αριθμούς, να αναγνωρίζει μοτίβα, να διατυπώνει υποθέσεις και να εφαρμόζει μαθηματικές εκφράσεις. Με αυτή την ικανότητα μπορούμε να κατανοούμε σχέσεις και επαναλαμβανόμενες δομές σε δεδομένα, ώστε να χρησιμοποιούμε αφηρημένη σκέψη και συμβολισμούς για να λύσουμε προβλήματα. Η λογικομαθηματική νοημοσύνη συχνά εκδηλώνεται σε ανθρώπους που ασχολούνται με τα μαθηματικά, τις επιστήμες και την έρευνα.

3. Κινησθητική νοημοσύνη:

Η κινησθητική νοημοσύνη είναι η δεξιότητα να χειρίζεσαι αντικείμενα με επιδεξιότητα και να εκτελείς διάφορες σωματικές ενέργειες με ακρίβεια και συντονισμό. Αυτή η ικανότητα περιλαμβάνει την αίσθηση του σωστού χρόνου για να εκτελεστεί μια ενέργεια και η δυνατότητα για βελτίωση της εκτέλεσης αυτής της ενέργειας σε συνεργασία μεταξύ του μυαλού (σκέψη) και του σώματος. Η δεξιότητα αυτή είναι πιο έντονη σε αθλητές, χορευτές, χειρουργούς, χειροτέχνες.

4. Μουσική Νοημοσύνη:

Η μουσική νοημοσύνη αφορά την ικανότητα να αντιλαμβάνεται και να αναγνωρίζει τόνους, ρυθμούς, χροιές και ήχους. Αυτή η ικανότητα επιτρέπει στον άνθρωπο να απολαμβάνει, να δημιουργεί και να αναπαράγει μουσική. Εκδηλώνεται μέσα από την εργασία συνθετών, μαέστρων, μουσικών, τραγουδιστών και ευαίσθητων ακροατών. Συχνά, υπάρχει μια στενή σύνδεση μεταξύ της μουσικής και των συναισθημάτων, και η μουσική νοημοσύνη μπορεί να μοιράζεται κοινές διαδικασίες σκέψης με τη μαθηματική νοημοσύνη.

5. Διαπροσωπική νοημοσύνη:

Η διαπροσωπική νοημοσύνη αναφέρεται στην ικανότητα ενός ατόμου να κατανοεί και να αλληλοεπιδρά σωστά με τους άλλους. Αυτή η δεξιότητα περιλαμβάνει την αποτελεσματική επικοινωνία, την ικανότητα αναγνώρισης των συναισθημάτων και των αναγκών των άλλων, καθώς και τη δυνατότητα να διαχειρίζεται αποτελεσματικά τις σχέσεις μεταξύ ατόμων. Άτομα

με υψηλή διαπροσωπική νοημοσύνη είναι συνήθως εξαιρετικοί στη διαχείριση των κοινωνικών δυναμικών και των συναισθημάτων των άλλων, και είναι ικανά να δημιουργούν καλές σχέσεις και να επιλύουν διαφορές με επιτυχία. Σε οποιοδήποτε περιβάλλον, από την προσωπική ζωή έως την επαγγελματική, η διαπροσωπική νοημοσύνη είναι κρίσιμη για την επιτυχή αλληλεπίδραση και την επίτευξη των κοινών στόχων.

6. Ενδοπροσωπική νοημοσύνη:

Η ενδοπροσωπική νοημοσύνη αναφέρεται στην ικανότητα ενός ατόμου να γνωρίζει και να κατανοεί τον εαυτό του, τις σκέψεις του και τα συναισθήματά του, και να χρησιμοποιεί αυτή την κατανόηση για να οδηγήσει τη ζωή του. Είναι η ικανότητα να αναγνωρίζεις τις δικές σου ανάγκες, επιθυμίες και φιλοδοξίες, και να λαμβάνεις αποφάσεις που αντανακλούν τις προσωπικές σου αξίες και στόχους. Η ενδοπροσωπική νοημοσύνη δεν περιορίζεται μόνο στην αυτογνωσία, αλλά περιλαμβάνει επίσης την ευαισθησία στις ανθρώπινες συνθήκες και την ικανότητα να αναγνωρίζεις και να συμπορεύεσαι με τα συναισθήματα και τις ανάγκες των άλλων. Αυτή η δεξιότητα είναι εμφανής σε ανθρώπους που έχουν εξειδικευμένες γνώσεις στη ψυχολογία, την πνευματική ανάπτυξη και τη φιλοσοφία.

7. Χωροταξική νοημοσύνη:

Η χωροταξική νοημοσύνη αναφέρεται στην ικανότητα του ανθρώπου να σκέφτεται και να αντιλαμβάνεται τον κόσμο γύρω του μέσω των τριών διαστάσεων. Συγκαταλέγονται σε αυτή την νοημοσύνη η δυνατότητα να απεικονίζει ο άνθρωπος στο μυαλό του εικόνες και αντικείμενα σε διαφορετικές προοπτικές, να κατανοεί τις σχέσεις μεταξύ των αντικειμένων σε έναν χώρο και να εκτιμά τις αποστάσεις και τις διαστάσεις. Επίσης, συμπεριλαμβάνεται η ικανότητα να διαχειρίζεται γραφικά, εικόνες και έργα τέχνης και να εκφράζει τη φαντασία του μέσω αυτών. Άνθρωποι που εργάζονται σε επαγγέλματα όπως ναυτικοί, πιλότοι, γλύπτες, ζωγράφοι και αρχιτέκτονες είναι γνωστοί για την ανάπτυξη και χρήση της χωροταξικής νοημοσύνης.

8. Φυσιοκρατική νοημοσύνη:

Η φυσιοκρατική νοημοσύνη αναφέρεται στην ικανότητα του ανθρώπου να αναγνωρίζει τη ζωή γύρω του, συμπεριλαμβανομένων των ζώων και των φυτών, καθώς και την ευαισθησία του σε άλλα φυσικά φαινόμενα όπως τα σύννεφα και τα σχηματισμένα πετρώματα. Αυτή η ικανότητα είναι σημαντική στον αρχαίο ρόλο του ανθρώπου ως κυνηγός, τροφосуλλέκτης και αγρότης, ενώ σήμερα παραμένει απαραίτητη σε επαγγέλματα όπως η βοτανολογία και η μαγειρική. Εκτιμάται επίσης ότι η σύγχρονη κοινωνία εκμεταλλεύεται αυτήν τη φυσική νοημοσύνη, καθώς μπορεί να χρησιμοποιηθεί για την αναγνώριση και τη διάκριση αντικειμένων στην καθημερινή ζωή.

2.3.1 Η Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη (TN) αποτελεί έναν από τους πιο επαναστατικούς και επιδραστικούς τομείς της σύγχρονης επιστήμης και τεχνολογίας. Πρόκειται για έναν διακλαδισμό της επιστήμης των υπολογιστών που ασχολείται με τη δημιουργία υπολογιστικών συστημάτων που έχουν τη δυνατότητα να αντιμετωπίζουν ένα ευρύ φάσμα εργασιών όπου απαιτείται ανθρώπινη νοημοσύνη, κατανοώντας την κατάσταση, εκτελώντας εργασίες, μαθαίνοντας από εμπειρία και προσαρμόζοντας τη

συμπεριφορά τους στο περιβάλλον. Αυτό συμπεριλαμβάνει την ικανότητα αναγνώρισης προτύπων, τη λήψη αποφάσεων, την επίλυση προβλημάτων και την αυτόματη εκμάθηση από τα δεδομένα. Πως όμως ορίζετε ότι ένα σύστημα ανήκει στην Τ.Ν; κατά καιρούς έχουν αποδοθεί πολύ ορισμοί. Μερικοί από τους ορισμούς που έχουν αναφέρει [3]είναι:

1. *"Η ικανότητα ενός συστήματος να ενεργεί κατάλληλα σε ένα αβέβαιο περιβάλλον, όπου η κατάλληλη ενέργεια είναι αυτή που αυξάνει την πιθανότητα επιτυχίας, και η επιτυχία είναι η επίτευξη συμπεριφορικών υποστόχων που υποστηρίζουν τον τελικό στόχο του συστήματος."*
2. *"Οποιοδήποτε σύστημα που δημιουργεί προσαρμοστική συμπεριφορά για να επιτύχει στόχους σε μια σειρά περιβαλλόντων μπορεί να χαρακτηριστεί ως έξυπνο."*
3. *"Επίτευξη σύνθετων στόχων σε σύνθετα περιβάλλοντα"*
4. *"Τα έξυπνα συστήματα αναμένεται να λειτουργούν, και να λειτουργούν καλά, σε πολλαπλά διαφορετικά περιβάλλοντα. Η ιδιότητά της εξυπνάδας τους επιτρέπει να μεγιστοποιήσουν την πιθανότητα επιτυχίας ακόμα και αν δεν είναι διαθέσιμη πλήρης γνώση της κατάστασης. Η λειτουργία των έξυπνων συστημάτων δεν μπορεί να θεωρηθεί ξεχωριστά από το περιβάλλον και την πραγματική κατάσταση, συμπεριλαμβανομένου του στόχου."*
5. *"Η απόδοση νοημοσύνης είναι η επιτυχημένη (δηλαδή, επιτυχής επίτευξη στόχων) απόδοση του συστήματος σε ένα πολύπλοκο περιβάλλον."*
6. *"Η νοημοσύνη είναι η ικανότητα να χρησιμοποιούνται βέλτιστα περιορισμένοι πόροι - συμπεριλαμβανομένου του χρόνου - για την επίτευξη στόχων."*
7. *"Η νοημοσύνη είναι η δύναμη να βρίσκει κάποιος γρήγορα μια επαρκή λύση σε αυτό που φαίνεται από πριν (για τους παρατηρητές) να είναι ένας τεράστιος χώρος αναζήτησης."*
8. *"Η νοημοσύνη μετρά την ικανότητα ενός πράκτορα να επιτύχει στόχους σε μια ευρεία γκάμα περιβαλλόντων."*
9. *"Να πηγαίνει καλά σε μια ευρεία γκάμα εργασιών είναι μια εμπειρική οριοθέτηση της 'νοημοσύνης'."*
10. *"Η νοημοσύνη είναι η υπολογιστική πλευρά της ικανότητας ενός συστήματος για την επίτευξη στόχων στον κόσμο. Διαφορετικοί τύποι και βαθμοί νοημοσύνης συμβαίνουν σε ανθρώπους, πολλά ζώα και μερικές μηχανές."*
11. *"Η νοημοσύνη είναι η ικανότητα να λύνει δύσκολα προβλήματα."*
12. *"Η νοημοσύνη είναι η ικανότητα να επεξεργάζεται πληροφορίες κατάλληλα σε ένα πολύπλοκο περιβάλλον. Τα κριτήρια της καταλληλότητας δεν είναι προκαθορισμένα και επομένως δεν είναι διαθέσιμα από πριν. Αποκτώνται ως αποτέλεσμα της επεξεργασίας πληροφοριών."*

13. *"Σε οποιαδήποτε πραγματική κατάσταση, η συμπεριφορά που είναι κατάλληλη για τους σκοπούς του συστήματος και προσαρμοστική στις απαιτήσεις του περιβάλλοντος μπορεί να συμβεί, εντός ορισμένων ορίων ταχύτητας και πολυπλοκότητας."*
14. *"Ένας έξυπνος πράκτορας κάνει αυτό που είναι κατάλληλο για τις συνθήκες και τον στόχο του, είναι ευέλικτος σε μεταβαλλόμενα περιβάλλοντα και μεταβαλλόμενους στόχους, μαθαίνει από την εμπειρία, και κάνει κατάλληλες επιλογές δεδομένων των αντιλήψεων και της πεπερασμένης υπολογιστικής δυνατότητας."*
15. *"Η νοημοσύνη σημαίνει να βελτιώνεται με το χρόνο."*
16. *"Η νοημοσύνη είναι η ικανότητα ενός συστήματος επεξεργασίας πληροφοριών να προσαρμόζεται στο περιβάλλον του με ανεπαρκείς γνώσεις και πόρους."*
17. *"Η νοημοσύνη είναι η δυνατότητα για μια νοητική επιτυχημένη ζωή."*
18. *"Τα βασικά, πεδία-ανεξάρτητες δεξιότητες που απαιτούνται για την απόκτηση ευρείας γκάμας ειδικής γνώσης - η ικανότητα να μαθαίνει οτιδήποτε. Η επίτευξη αυτού με την 'τεχνητή γενική νοημοσύνη' (AGI) απαιτεί ένα υψηλά προσαρμοστικό, γενικό σύστημα που μπορεί να αποκτήσει αυτόνομα μια εξαιρετικά ευρεία γκάμα συγκεκριμένης γνώσης και δεξιοτήτων και μπορεί να βελτιώνει την ίδια την γνωστική της ικανότητα μέσω αυτοδιανομής μάθησης."*

Ένας γενικός ορισμός, ο οποίος αναφέρεται [4] είναι: "Η τεχνητή νοημοσύνη είναι ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά τα οποία αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως για παράδειγμα η επίλυση προβλημάτων, η αντίληψη και κατανόηση εικόνων, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας κτλ".

2.3.2 Η Ιστορική Εξέλιξη της Τεχνητής Νοημοσύνης

Η εξέλιξη της τεχνητής νοημοσύνης ξεκινά από την αρχαία Ελλάδα, οι πρώτες αναφορές βρίσκονται στη μυθολογία όπου ο Δαίδαλος προσπαθούσε να κάνει τεχνητούς ανθρώπους, τον Αριστοτέλη που είναι θεμελιωτής της λογικής, καθώς και τον Αλέξανδρο Ήρωνα που είχε επινοήσει μηχανικούς αυτοματισμούς για πόρτες, αυλαίες παραστάσεων κ.α.

Μετέπειτα από τον 13ο ως τον 17ο αιώνα μ.Χ είχαμε μηχανικούς που δημιουργούσαν αυτοματισμούς και έννοιες, τα οποία αποτέλεσαν την αρχή για τους υπολογιστές αλλά και άλλα τεχνολογικά κατορθώματα των μετέπειτα αιώνων έως και σήμερα.

Όμως ουσιαστική πρόοδο είχαμε από τον 19ο μ.Χ αιώνα και μετά, με καθοριστικό το συνέδριο Τεχνητής Νοημοσύνης, το οποίο διοργανώθηκε στο Dartmouth της Μασαχουσέτης το 1956 στο οποίο παραβρέθηκαν σπουδαίοι επιστήμονες και ερευνητές. Στο συνέδριο αυτό αναπτύχθηκαν τα θέματα, της θεωρίας αυτομάτων, τα νευρωνικά δίκτυα και η μελέτη της αφύιας. Στο συνέδριο αυτό επίσης παρουσιάστηκε από δυο ερευνητές τους Allan Newell και Herbert Simon, ένα πρόγραμμα με το όνομα Logic Theorist το οποίο αποδείκνυε τα περισσότερα από τα μαθηματικά θεωρήματα των Russell και Whitehead που αναφερόταν στο Principia Mathematica. Όμως το πιο σημαντικό γεγονός στο συνέδριο

Κεφάλαιο 2

αυτό ήταν η αποδοχή του ονόματος που πρότεινε ο McCarthy για τη νέα ερευνητική περιοχή το οποίο είναι Τεχνητή Νοημοσύνη. [4]

Ιστορικά γεγονότα που συνέβαλαν στην εξέλιξη της τεχνίτης νοημοσύνης είναι με χρονολογική σειρά:

1. *Αρχαιότητα (ίσως 10 με 15 π.Χ. έως 1 με 2 μ.Χ): Ο Αλέξανδρος Ήρωνας δημιούργησε μηχανικούς αυτοματισμούς με τη χρήση της δύναμης του νερού και του ατμού.* [5]
2. 1206: Ο Ebru İz Bin Rezzaz Al Jazari, ήταν πρωτοπόρος μηχανικός που κατασκεύαζε μηχανήματα αυτομάτου ελέγχου που λειτουργούσαν με ενέργεια το νερό. [5]
3. 1623: Ο Wilhelm Schickard εφηύρε το μηχανικό υπολογιστή που μπορούσε να κάνει 4 βασικές μαθηματικές πράξεις. [5]
4. 1672: Ο Gottfried Leibniz, δημιούργησε το σύστημα της δυαδικής λογικής, το οποίο είναι η βάση των σημερινών υπολογιστών. [5]
5. 1822-1859: Ο Charle Babbage σχεδίασε δυο μηχανικούς υπολογιστές. [5]
6. 1854: George Boole πρότεινε τη λογική δυαδική άλγεβρα που πήρε και το όνομα του (Άλγεβρα Boole). [4]
7. 1879: Gottlieb Frege έθεσε τις βάσεις του κατηγορηματικού λογισμού [4]
8. 1923: Karel Čapek εισήγαγε για πρώτη φορά την έννοια του ρομπότ στο θεατρικό έργο "Rossum's Universal Robots" [5]
9. 1931: Kurt Gödel απέδειξε το θεώρημα της μη πληρότητας που έχει πάρει και το όνομα του. [5]
10. 1936: Konrad Zuse ανέπτυξε ένα προγραμματιζόμενο υπολογιστικό μηχάνημα που ονόμασε Z1, το οποίο θεωρείται ένας από τους πρώτους ψηφιακούς υπολογιστές στον κόσμο. Αυτός ο υπολογιστής είχε μνήμη 64K. [5]
11. 1943: Warren McCulloch και ο Walter Pitts παρουσιάζεται μια θεωρητική τεχνητή μοντελοποίηση νευρώνων της λειτουργία του εγκεφάλου, προτείνοντας ένα σύστημα τεχνητών νευρώνων που λειτουργούν με βάση λογικούς κανόνες. [4]
12. 1946: Τέθηκε σε λειτουργία ο πρώτος ηλεκτρονικός υπολογιστής ο ENIAC ο οποίος είχε μέγεθος ενός δωματίου και ζύγιζε 30 τόνους. [5]
13. 1948: Ο John von Neumann, παρουσίασε την ιδέα ενός αυτό -αντιγραφόμενου προγράμματος στο άρθρο του "The General and Logical Theory of Automata". [5]

14. 1949: Ο Donald Hebb, παρουσίασε τη θεωρία της "Hebbian learning". η θεωρία αναφέρει ότι η δύναμη μιας συνάψεως μεταξύ νευρώνων αυξάνεται όταν οι νευρώνες ενεργοποιούνται ταυτόχρονα και είναι υπεύθυνοι για μια συγκεκριμένη λειτουργία. [5]
15. 1950: Ο Alan Turing, θεμελιωτής της επιστήμης της Τεχνητής Νοημοσύνης, εισήγαγε την έννοια του Τεστ Turing. Αυτό το τεστ προτάθηκε ως τρόπος να αξιολογηθεί η νοημοσύνη ενός μηχανήματος ή ενός υπολογιστή. [5]
16. 1951: Marvin Minsky, Dean Edmonds υλοποίησαν το πρώτο νευρωνικό δίκτυο με λυχνίες το λεγόμενο SNARC. [4]
17. 1956: Έγινε μεγάλο επιστημονικό συνέδριο στο Dartmouth της Μασαχουσέτης, προτάθηκε από τον J.McCarthy και έγινε αποδεκτό το όνομα της ερευνητικής περιοχής ως Τεχνητή Νοημοσύνη. [4]
18. 1956: Allen Newell, Herben Simon, στο ίδιο συνέδριο στο Dartmouth παρουσίασαν το Logic Theorist, πρόγραμμα το οποίο αποδείκνυε τα περισσότερα από τα θεωρήματα των Russell και Whitehead. [4]
19. 1957: Οι Allen Newell, Herben Simon, παρουσίασαν το πρόγραμμα General Problem Solver, GPS. Ο Γενικός Επίλυτης Προβλημάτων ήταν ένα πρόγραμμα που είχε σχεδιαστεί για να επιλύει προβλήματα αλλά είχε περιορισμούς στη πολυπλοκότητα των προβλημάτων. [4]
20. 1958: Ο John McCarthy ορίζει τη γλώσσα προγραμματισμού LISP ως γλώσσα προγραμματισμού για την Τεχνητή Νοημοσύνη. [5]
21. 1958: Ο Friedberg πρότεινε τη τεχνική του machine evolution, που σήμερα ονομάζεται Genetic Programming. [4]
22. 1960: Ο J.C.R Licklider έγραψε το διάσημο άρθρο "Man-Computer Symbiosis" το 1960, όπου προέβλεπε την εμφάνιση ενός συστήματος όπου οι άνθρωποι θα συνεργάζονταν με τους υπολογιστές για την επίλυση προβλημάτων. [5]
23. 1962: Η εταιρεία Unimation έγινε η πρώτη εταιρεία που παρήγαγε robot για τη βιομηχανία. [5]
24. 1962: Ο Rosenblatt παρουσίασε το θεώρημα της σύγκλισης του perseptron. [4]
25. 1963: Ο McCarthy ίδρυσε το εργαστήριο Τεχνητής Νοημοσύνης στο Staandford. [4]
26. 1965: Ο Weizenbraum ήταν δημιούργησε το πρόγραμμα Eliza το οποίο ήταν και το πρώτο που πέρασε το Turning Test. [5]
27. 1966: Αναπτύχθηκε στο Ινστιτούτο Έρευνας του Standford το ρομπότ με το όνομα Shakey, το οποίο είχε εξοπλιστεί με μια τηλεοπτική κάμερα, αισθητήρες εμβέλειας και αισθητήρες σύγκρουσης, επιτρέποντάς του να πλοηγείται στο περιβάλλον του. [5]

28. 1973: Η DARPA (Defense Advanced Research Projects Agency) ξεκίνησε την ανάπτυξη των πρωτοκόλλων TCP/IP. Τα πρωτόκολλα TCP/IP αποτέλεσαν τη βάση για το διαδίκτυο που γνωρίζουμε σήμερα. [5]
29. 1974: Το internet ξεκίνησε να χρησιμοποιείται από το ευρύ κοινό. [5]
30. 1978: Ο Herbert Simon, βραβεύτηκε με βραβείο Νόμπελ για το θεώρημα της Περιορισμένης ορθολογικότητας, το θεώρημα επηρέασε την ανάπτυξη μοντέλων λήψης αποφάσεων και συστημάτων ευφυούς λήψης αποφάσεων στον τομέα της τεχνητής νοημοσύνης. [5]
31. 1981: Η IBM παρουσίασε τον πρώτο προσωπικό υπολογιστή. [5]
32. 1993: Ξεκίνησε η παραγωγή του Cog, ενός ρομπότ που μοιάζει με ανθρώπινο στο MIT. [5]
33. 1997: Το Deep Blue, ένας υπερυπολογιστής που αναπτύχθηκε από την IBM, νίκησε τον παγκόσμιο πρωταθλητή σκάκι Gary Kasparov. [5]
34. 1998: Το Furby, το πρώτο ρομπότ τεχνητής νοημοσύνης, εισήχθη στην αγορά. [5]
35. 2000: Το Kismet, ένα ρομπότ που μπορεί να χρησιμοποιήσει κινήσεις και μίμηση κινήσεων στην επικοινωνία, παρουσιάστηκε. [5]
36. 2005: Εισήχθη ο Asimo, ένα ανθρωποειδές ρομπότ που αναπτύχθηκε από την Honda, επιδεικνύοντας προηγμένη ρομποτική και ικανότητες τεχνητής νοημοσύνης. [5]
37. 2010: Ο Asimo έγινε ικανός να ενεργοποιείται με τη δύναμη του μυαλού. [5]
38. 2011: Το IBM Watson κέρδισε στο γνωστό αμερικανικό τηλεοπτικό παιχνίδι Jeopardy, επιδεικνύοντας προηγμένες ικανότητες στη γνώση και την κατανόηση της φυσικής γλώσσας.
39. 2014: Το Facebook ξεκίνησε το ερευνητικό πρόγραμμα DeepFace, το οποίο επιδιώκει την αναγνώριση προσώπων με χρήση βαθιών νευρωνικών δικτύων.
40. 2016: Η Google DeepMind ανέπτυξε τον AlphaGo, ένα σύστημα τεχνητής νοημοσύνης που κατάφερε να νικήσει τον παγκόσμιο πρωταθλητή στο Go, ένα από τα πιο πολύπλοκα επιτραπέζια παιχνίδια.
41. 2017: Η OpenAI ανακοίνωσε την δημιουργία του AlphaZero, ενός συστήματος που μπορεί να μάθει να παίζει παιχνίδια χωρίς καμία προηγούμενη γνώση ή εκπαίδευση, κερδίζοντας σε στρατηγικά παιχνίδια όπως το σκάκι, το Go και το shogi.
42. 2018: Η Waymo, η θυγατρική εταιρεία της Alphabet (μητρική εταιρεία της Google), ξεκίνησε τις δοκιμές αυτόνομων οχημάτων χωρίς οδηγό σε πραγματικές κυκλοφοριακές συνθήκες.

43. 2020: Η OpenAI κυκλοφόρησε την GPT-3, την τρίτη γενιά του Generative Pre-trained Transformer, ένα πολύ μεγάλο και πολύπλοκο μοντέλο γλωσσικής επεξεργασίας που μπορεί να παράγει παρόμοιο κείμενο με τον άνθρωπο με εντυπωσιακή ακρίβεια.

2.3.3 Εφαρμογές της Τεχνητής Νοημοσύνης

Η ΤΝ έχει εφαρμογές σε πολλούς τομείς της ζωής μας. Στον τομέα της υγείας, χρησιμοποιείται για τη διάγνωση ασθενειών και τον σχεδιασμό θεραπειών. Στη βιομηχανία, βελτιώνει τις διαδικασίες παραγωγής και τον έλεγχο ποιότητας. Στην ψυχαγωγία, δημιουργεί εξελιγμένες εμπειρίες για τους χρήστες, όπως συστάσεις για ταινίες και μουσική. Στην οικονομία, χρησιμοποιείται για τον αναλυτικό υπολογισμό κινδύνου και την πρόβλεψη των χρηματοοικονομικών αγορών. Στην εκπαίδευση, διευκολύνει την προσαρμογή της διδασκαλίας στις ανάγκες του κάθε μαθητή.

Η ΤΝ έχει επίσης σημαντική επίδραση στην αυτονομία των μηχανών και των ρομπότ. Η ανάπτυξη αυτόνομων οχημάτων που μπορούν να λειτουργούν χωρίς ανθρώπινη παρέμβαση είναι ένα από τα κορυφαία επιτεύγματα της ΤΝ. Επίσης, η ΤΝ έχει εφαρμογές στην ασφάλεια, καθώς μπορεί να ανιχνεύσει ανωμαλίες στα δεδομένα και να προβλέψει επιθέσεις.

Με την συνεχή εξέλιξη της τεχνολογίας, η ΤΝ αναμένεται να συνεχίσει να επεκτείνεται και να εμπλέκεται σε νέους τομείς. Στον τομέα της ρομποτικής, η δημιουργία ανθρωποειδών ρομπότ που μπορούν να αλληλοεπιδρούν με τους ανθρώπους αναμένεται να γίνει όλο και πιο κοινή. Στην υγεία, η ΤΝ μπορεί να συμβάλει στην προσωποποίηση της θεραπείας μέσω της ανάλυσης του γενετικού προφίλ του κάθε ασθενούς.

2.4 Μάθηση

Η μάθηση είναι η διαδικασία κατά την οποία ο άνθρωπος αποκτά γνώσεις, δεξιότητες, αντιλήψεις, συμπεριφορές και προσαρμόζεται στο περιβάλλον του. Είναι μία σύνθετη και δυναμική διαδικασία η οποία συμβαίνει καθ' όλη τη διάρκεια της ζωής του ανθρώπου.

Για παράδειγμα το παιδί μαθαίνει να μιλά (αποκτά τη δεξιότητα της ομιλίας) από τους γονείς του μέσω της ακρόασης και της επανάληψης φράσεων και λέξεων.

Μαθαίνει να περπατά βλέποντας τους γύρω του να περπατούν και με την βοήθεια των γονέων κάνει τα πρώτα του βήματα. Στο σχολείο αποκτά γνώσεις σε διάφορα θέματα όπως τα μαθηματικά, η γλώσσα, κ.α μέσω της μελέτης. Στη συνέχεια αποκτά γνώσεις και δεξιότητες για την εργασία μέσω της εκπαίδευσης και της πρακτικής εμπειρίας. Μαθαίνει να προσαρμόζεται γενικά σύμφωνα με τις εμπειρίες του. [6]

2.4.1 Η μηχανική μάθηση

Η μηχανική μάθηση πραγματοποιείται τη διαδικασία βελτίωσης της επίδοσης ενός συστήματος σε μία συγκεκριμένη εργασία, ενώ πρώτα έχει παρατήρηση πολλά παραδείγματα. Για να υπάρξει μηχανική μάθηση τρία πράγματα χρειάζονται.

- Να υπάρχει μια συγκεκριμένη εργασία που πρέπει το σύστημα να εκτελέσει.
- Να υπάρχουν δεδομένα τα οποία το σύστημα θα εκπαιδευτεί.
- Να υπάρχει κριτήριο αξιολόγησης της επίδοσης του συστήματος.

Η μηχανική μάθηση είναι ένας επιστημονικός υποτομέας από τους πολλούς υποτομείς της Τεχνητής Νοημοσύνης «Σχήμα 2.1». Η μηχανική μάθηση δέχεται στην είσοδο της δεδομένα, γνωστά και ως παραδείγματα ή πρότυπα, τα οποία μπορεί να προέρχονται από ποικίλες πηγές. Αυτές οι πηγές μπορεί

να περιλαμβάνουν αισθητήρες όπως αισθητήρες θερμοκρασίας, υγρασίας, οπτικούς αισθητήρες (όπως κάμερες), αλλά και άλλους τύπους αισθητήρων που μετρούν διάφορες φυσικές ή ψηφιακές παραμέτρους. Επιπλέον, τα δεδομένα μπορεί να προέρχονται από καταγραφές αποτελεσμάτων πειραμάτων ή μετρήσεων, καταγραφές από έρευνες και μελέτες, ή ακόμη και από καταγραφές συμβάντων και επεισοδίων στο παρελθόν. Όλα αυτά τα δεδομένα συχνά αποθηκεύονται σε βάσεις δεδομένων, οι οποίες λειτουργούν ως αποθετήρια πληροφοριών για την εκπαίδευση και τη λειτουργία των αλγορίθμων μηχανικής μάθησης που συχνά αναφέρονται και ως μοντέλα μηχανικής μάθησης. [7]

Τα δεδομένα αυτά διαμορφώνονται κατάλληλα προτού χρησιμοποιηθούν για την εκπαίδευση των μοντέλων της μηχανικής μάθησης. Αυτή η διαδικασία περιλαμβάνει τον καθαρισμό των δεδομένων, την επεξεργασία και την ανάλυση τους, καθώς και τη δημιουργία χαρακτηριστικών που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Κατά τη διαδικασία του καθαρισμού, αφαιρούνται ή διορθώνονται ανωμαλίες, απουσίες ή ακόμη και εκτροπές στα δεδομένα. Στη συνέχεια, γίνεται επεξεργασία των δεδομένων, συμπεριλαμβανομένης της κανονικοποίησης ή της κλιμάκωσης των χαρακτηριστικών, προκειμένου να εξασφαλιστεί η συνοχή και η αποτελεσματικότητα των δεδομένων. Έπειτα, δημιουργούνται τα χαρακτηριστικά ή γνωρίσματα που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Αυτά τα χαρακτηριστικά είναι ουσιαστικά οι μεταβλητές που περιγράφουν τα δεδομένα και βοηθούν το μοντέλο να εκμεταλλευτεί τα πρότυπα και τις σχέσεις μεταξύ τους για την εκπαίδευση. Συνολικά, αυτή η διαδικασία διασφαλίζει ότι τα δεδομένα είναι έτοιμα και κατάλληλα για την εκπαίδευση του μοντέλου μηχανικής μάθησης, επιτρέποντας του να αποκτήσει την καλύτερη δυνατή κατανόηση των προτύπων και των σχέσεων στα δεδομένα. Το μοντέλο της μηχανικής μάθησης, αφού εκπαιδευτεί με τα δεδομένα, μπορεί να κάνει προβλέψεις για νέα - άγνωστα δεδομένα με βάση τα προηγούμενα παραδείγματα που έχει εκπαιδευτεί. Αυτό σημαίνει ότι το μοντέλο μπορεί να αναγνωρίσει πρότυπα και τάσεις στα δεδομένα εκπαίδευσης και να τα εφαρμόσει σε νέα δεδομένα για την παραγωγή προβλέψεων ή αποφάσεων.

Έτσι μπορούμε να πούμε ότι έχουμε τρία στάδια στη διαδικασία μηχανικής μάθησης ενός μοντέλου:

1. Εκπαίδευση(Training)

Κατά τη διάρκεια του σταδίου της εκπαίδευσης σε ένα μοντέλο μηχανικής μάθησης, το μοντέλο εκπαιδεύεται χρησιμοποιώντας ένα σύνολο δεδομένων εκπαίδευσης. Κατά τη διάρκεια της εκπαίδευσης, επιπλέον, είναι συνηθισμένο να παρακολουθούμε την απόδοση του μοντέλου στο σύνολο δεδομένων εκπαίδευσης, καθώς και σε ένα ανεξάρτητο σύνολο δεδομένων επικύρωσης ή ελέγχου.

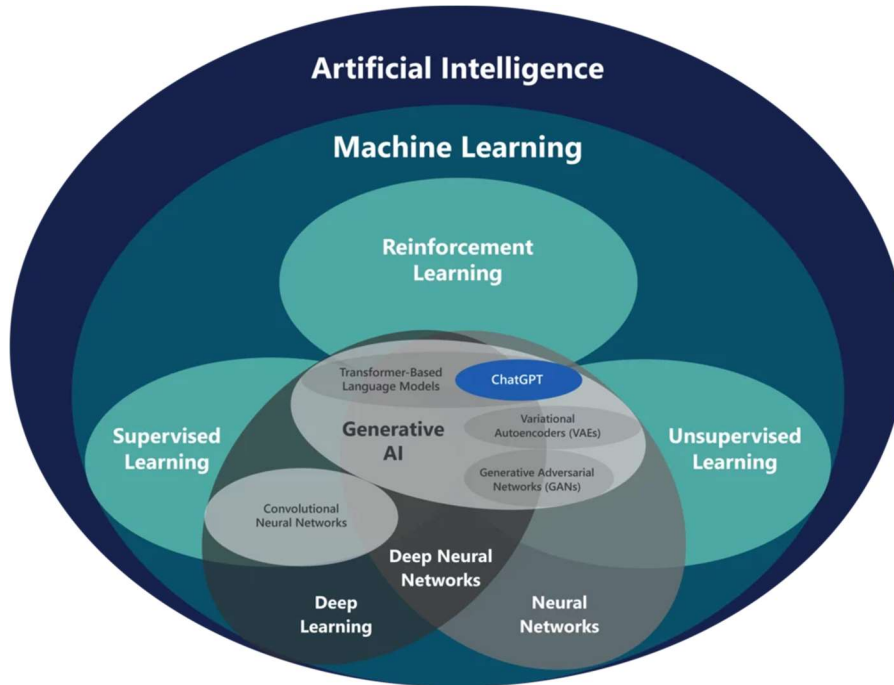
2. Επικύρωση(Validation)

Στο στάδιο της επικύρωσης (validation), το μοντέλο αξιολογείται χρησιμοποιώντας ένα ανεξάρτητο σύνολο δεδομένων που δεν έχει χρησιμοποιηθεί κατά τη διάρκεια της εκπαίδευσης. Το σύνολο αυτό δεδομένων χρησιμοποιείται για να εκτιμηθεί η απόδοση του μοντέλου και να γίνει η αξιολόγησή του σε ένα περιβάλλον που δεν έχει δει κατά τη διάρκεια της εκπαίδευσης. Συγκεκριμένα, κατά τη διάρκεια του σταδίου της επικύρωσης, γίνεται πρόβλεψη, αξιολόγηση της πρόβλεψης, προσαρμογή του μοντέλου βάση των αποτελεσμάτων (επίβλεψη). Το στάδιο της επικύρωσης είναι σημαντικό για τον έλεγχο της γενίκευσης του μοντέλου και την αποφυγή της υπερπροσαρμογής (overfitting).

3. Πρόβλεψη(Prediction)

Στο στάδιο της πρόβλεψης, το εκπαιδευμένο μοντέλο χρησιμοποιείται για να προβλέψει τις ετικέτες ή τις εξόδους για νέα, μη επεξεργασμένα δεδομένα που δεν

έχει δει κατά τη διάρκεια της εκπαίδευσης ή της επικύρωσης. Συγκεκριμένα, κατά τη διάρκεια του σταδίου της πρόβλεψης, γίνεται πρόβλεψη και αξιολόγηση για να δούμε την ποιότητα των προβλέψεων αλλά και πιθανές βελτιώσεις που μπορούμε να κάνουμε.



Σχήμα 2.1 Οι τομείς της Τεχνητής Νοημοσύνης

[Πηγή: <https://global.hitachi-solutions.com/blog/generative-ai-manufacturing/>]

Η μηχανική μάθηση χωρίζεται σε τρεις μεγάλες κατηγορίες ανάλογα με τον τρόπο που εκπαιδεύεται ο κάθε αλγόριθμος και είναι:

1. Επιβλεπόμενη μάθηση:

Επιβλεπόμενη μάθηση η οποία αποτελεί έναν ισχυρό και ευέλικτο τρόπο για την εκπαίδευση μοντέλων πρόβλεψης ή ταξινόμησης. Στην επιβλεπόμενη μάθηση, το σύστημα λαμβάνει ένα σύνολο ετικετοποιημένων δεδομένων εκπαίδευσης, όπου κάθε παράδειγμα συνοδεύεται από μια ετικέτα ή μια επιθυμητή έξοδο.

Ο στόχος είναι να εκπαιδευτεί ένα μοντέλο που θα μπορεί να προβλέπει ή αλλιώς να ταξινομεί τις ετικέτες των νέων δεδομένων που δεν έχει δει πριν. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο προσαρμόζει τις παραμέτρους του μέσω της ανάλυσης των δεδομένων εκπαίδευσης και της προσπάθειας να ελαχιστοποιήσει το σφάλμα μεταξύ των πραγματικών και των προβλεπόμενων ετικετών. Αυτή η διαδικασία συνεχίζεται μέχρις ότου το μοντέλο επιτύχει μια ικανοποιητική απόδοση στα δεδομένα εκπαίδευσης.

Όταν το μοντέλο εκπαιδευτεί, μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις για νέα, μη ετικετοποιημένα δεδομένα. Τα δεδομένα αυτά μπορεί να προέρχονται από τον πραγματικό κόσμο και να περιλαμβάνουν παραδείγματα που το μοντέλο δεν έχει δει προηγουμένως. Η απόδοση του μοντέλου σε αυτά τα δεδομένα είναι ένας κρίσιμος παράγοντας που καθορίζει την αποτελεσματικότητα και την αξιοπιστία του.

Τα πλεονεκτήματα της επιβλεπόμενης μάθησης περιλαμβάνουν την ικανότητά της να αντιμετωπίζει πολύπλοκα προβλήματα και τη δυνατότητά της να προβλέπει τις ετικέτες για νέα δεδομένα με υψηλή ακρίβεια. Ωστόσο, απαιτεί μεγάλο όγκο ετικετοποιημένων δεδομένων

για την εκπαίδευση και τη βελτιστοποίηση του μοντέλου, καθώς και επαρκή υπολογιστικούς πόρους για τη διεξαγωγή των υπολογισμών.

2. Μη επιβλεπόμενη μάθηση:

Μη επιβλεπόμενη μάθηση, το μοντέλο πρέπει να ανακαλύπτει μοτίβα ή δομές στα δεδομένα χωρίς την προϋπόθεση της ύπαρξης ετικετών. Αντίθετα με την επιβλεπόμενη μάθηση, όπου το μοντέλο λαμβάνει σαφείς ετικέτες για κάθε παράδειγμα εκπαίδευσης, στην μη επιβλεπόμενη μάθηση το μοντέλο πρέπει να ανακαλύψει τη δομή των δεδομένων με βάση τις αυθαίρετες σχέσεις ή πρότυπα που ενδέχεται να υπάρχουν μεταξύ τους.

Κατά την εκπαίδευση σε ένα σύστημα μη επιβλεπόμενης μάθησης, το μοντέλο αναλύει τα δεδομένα και προσπαθεί να εντοπίσει κρυμμένες δομικές πληροφορίες ή πιθανά κατηγορίες. Αυτό μπορεί να γίνει μέσω μεθόδων όπως οι αλγόριθμοι συστάδας, οι οποίοι ομαδοποιούν τα δεδομένα σε ομάδες με βάση την ομοιότητα, ή μέσω της ανάλυσης συστάδων για την εντοπισμό τυχόν δομικών μοτίβων.

Μια από τις μεγάλες προκλήσεις στην μη επιβλεπόμενη μάθηση είναι η ερμηνεία των αποτελεσμάτων. Αφού το μοντέλο ανακαλύψει τα πρότυπα ή τις δομές, η αξιολόγηση και η ερμηνεία τους απαιτούν συνήθως ανθρώπινη παρέμβαση και κρίση. Επίσης, η μη επιβλεπόμενη μάθηση μπορεί να απαιτεί περισσότερους υπολογιστικούς πόρους σε σύγκριση με την επιβλεπόμενη μάθηση, καθώς οι αλγόριθμοι πρέπει να εξερευνήσουν τον χώρο των δεδομένων χωρίς καθοδήγηση.

Παρά τις προκλήσεις αυτές, η μη επιβλεπόμενη μάθηση είναι ιδιαίτερα χρήσιμη όταν τα δεδομένα είναι απλά πολλά ή όταν οι ετικέτες δεν είναι εύκολα διαθέσιμες. Επιπλέον, μπορεί να αποκαλύψει νέες πτυχές ή κρυμμένες σχέσεις στα δεδομένα, που μπορούν να οδηγήσουν σε πολύτιμες ειδοποιήσεις ή ευκαιρίες. Συνεπώς, η μη επιβλεπόμενη μάθηση παρέχει έναν ισχυρό μηχανισμό για την ανάλυση και την ανακάλυψη πληροφοριών σε μεγάλες και πολύπλοκες συλλογές δεδομένων.

3. Ενισχυτική μάθηση:

Ενισχυτική μάθηση, σε αυτή τη κατηγορία μηχανικής μάθησης, οι αλγόριθμοι (αναφέρονται ως πράκτορες) εκπαιδεύονται να λαμβάνουν αποφάσεις βασισμένες σε ανταμοιβή, η ανταμοιβή μπορεί να είναι θετική ή αρνητική. Κατά τη διαδικασία αυτή, το σύστημα επικοινωνεί με το περιβάλλον του και λαμβάνει την ανταμοιβή για κάθε ενέργεια που πραγματοποιεί. Στη συνέχεια, αναπροσαρμόζει τη συμπεριφορά του βάση των ανταμοιβών που έχει λάβει, με στόχο τη μεγιστοποίηση της συνολικής ανταμοιβής που λαμβάνει μακροπρόθεσμα.

Το κύριο πλεονέκτημα της ενισχυτικής μάθησης είναι η ικανότητά της να μάθει από την εμπειρία, χωρίς την ανάγκη για ετικετοποιημένα δεδομένα εκπαίδευσης ή γνωστούς κανόνες. Αυτό το κάνει ιδιαίτερα χρήσιμη σε περιβάλλοντα όπου οι συνθήκες είναι αβέβαιες ή μπορούν να μεταβάλλονται με τον χρόνο. Επιπλέον, η ενισχυτική μάθηση μπορεί να επιτρέψει στα συστήματα να προσαρμοστούν σε πολύπλοκα περιβάλλοντα και να προσφέρουν βελτιώσεις στη συμπεριφορά τους με τον χρόνο.

Ωστόσο, η ενισχυτική μάθηση μπορεί να είναι προκλητική στην υλοποίηση, καθώς απαιτεί συνεχή αλληλεπίδραση με το περιβάλλον και συχνές ενημερώσεις του μοντέλου βάσει των ανταμοιβών που λαμβάνει. Επιπλέον, η εξαγωγή αξιόπιστων αποτελεσμάτων μπορεί να απαιτεί μεγάλο αριθμό δοκιμών ή επαναλαμβανόμενες δράσεις, ιδιαίτερα σε περιβάλλοντα με υψηλή δυναμική.

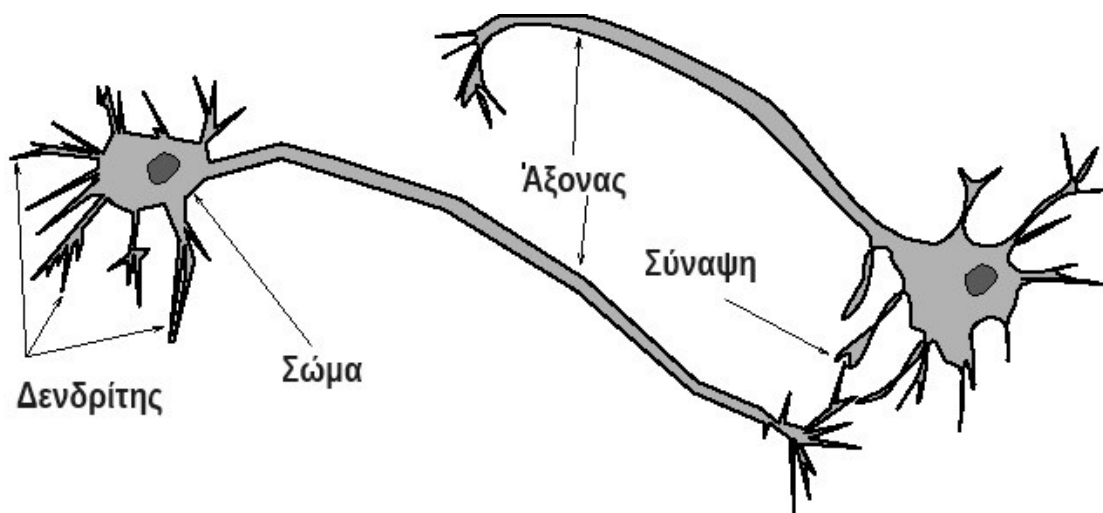
Τέλος, η ενισχυτική μάθηση έχει ευρεία εφαρμογή σε πολλούς τομείς, όπως τα αυτόνομα ρομποτικά συστήματα, η αυτόματη συμπεριφορά λογισμικού και η αναγνώριση προτύπων σε μεγάλα σύνολα δεδομένων. Με τη συνεχή εξέλιξη της τεχνολογίας και των αλγορίθμων, η ενισχυτική μάθηση αναμένεται να συνεχίσει να αποτελεί έναν σημαντικό πυλώνα στον χώρο της μηχανικής μάθησης και γενικά της Τεχνητής Νοημοσύνης.

2.5 Τα Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα αποτελούν ένα σημαντικό πεδίο της επιστημονικής έρευνας και της μηχανικής μάθησης. Η έρευνα γύρω από τα νευρωνικά δίκτυα ξεκίνησε με την επιθυμία να μοντελοποιηθεί ο εγκέφαλος και η λειτουργία του ανθρώπινου νευρικού συστήματος. Οι πρώτες προσπάθειες είχαν ως στόχο την κατανόηση της δομής και της λειτουργίας των βιολογικών νευρώνων του εγκεφάλου, καθώς και της προσομοίωσης τους με χρήση υπολογιστικών μοντέλων. Με την πάροδο του χρόνου, η έρευνα εξελίχθηκε προς την ανάπτυξη αλγορίθμων μηχανικής μάθησης και τεχνικών εκπαίδευσης των νευρωνικών δικτύων, με στόχο την επίλυση πρακτικών προβλημάτων σε πολλούς τομείς, όπως η εικονική αναγνώριση, η φωνητική αναγνώριση, η επεξεργασία φυσικής γλώσσας και η αυτοματοποιημένη λήψη αποφάσεων. Η έρευνα στον τομέα των τεχνητών νευρωνικών δικτύων συνεχίζεται σήμερα με την ανάπτυξη πιο προηγμένων αρχιτεκτονικών δικτύων, την εξερεύνηση νέων μεθόδων εκπαίδευσης, την εφαρμογή τους σε νέους τομείς και την ανάπτυξη βελτιστοποιημένων αλγορίθμων μάθησης.

2.5.1 Ο βιολογικός νευρώνας

Σε αυτό το σημείο θα κάνουμε μια μικρή αναφορά στο βιολογικό νευρώνα του εγκεφάλου για να αναφέρουμε και να εξηγήσουμε έννοιες τις οποίες θα χρησιμοποιήσουμε και σε επόμενα κεφάλαια της εργασίας. Τι είναι λοιπόν ο βιολογικός νευρώνας; Ο βιολογικός νευρώνας είναι η βασική μονάδα λειτουργίας του νευρικού συστήματος στους ζωντανούς οργανισμούς. Αποτελεί το βασικό δομικό και λειτουργικό στοιχείο του εγκεφάλου, του νωτιαίου μυελού και των νευρικών παραμέτρων σε όλο το σώμα.



Σχήμα 2.2 Βιολογικοί νευρώνες με σύναψη

[Πηγή: <https://blog.knoldus.com/first-interaction-artificial-neural-network/>]

Όπως βλέπουμε από την «Σχήμα 2.2» ο νευρώνας αποτελείται από 4 κύρια μέρη:

1. Δενδρίτες.
Οι δενδρίτες είναι ουσιαστικά, οι είσοδοι του νευρώνα εδώ δέχεται τα σήματα από άλλους νευρώνες ή του αισθητήρες του σώματος.
2. Σώμα.
Εδώ βρίσκονται ο πυρήνας και άλλα βασικά κυτταρικά στοιχεία, όπως τα μιτοχόνδρια, που παρέχουν ενέργεια στο νευρώνα.
3. Άξονας.
Είναι η έξοδος του νευρώνα που μεταφέρει τα ηλεκτρικά σήματα από το σώμα του νευρώνα προς άλλους νευρώνες, τους μυς ή τους αδένες. Το μήκος του άξονα στους νευρώνες ποικίλει και ξεκινά από μερικά χιλιοστά και φτάνει ως πάνω από το ένα μέτρο.
4. Συνάψεις.
Είναι τα σημεία όπου ενώνονται οι διακλαδώσεις του άξονα ενός νευρώνα, με τους δενδρίτες από άλλους νευρώνες. Οι συνάψεις ουσιαστικά είναι σαν μία μεμβράνη η οποία περιέχει ηλεκτροχημικό υλικό το οποίο μεταδίδει την ηλεκτρική ενέργεια. Το ποσοστό της ηλεκτρικής ενέργειας η οποία μεταδίδεται από τον άξονα στους δενδρίτες ονομάζεται συναπτικό βάρος. Οι συνάψεις ανάλογα με το αν ενεργοποιούν η αδρανοποιούν τον νευρώνα ονομάζονται ενισχυτικές ή ανασταλτικές.

2.5.2 Λειτουργία ενός βιολογικού νευρώνα

Ας εξετάσουμε τα βήματα λειτουργίας ενός νευρώνα.

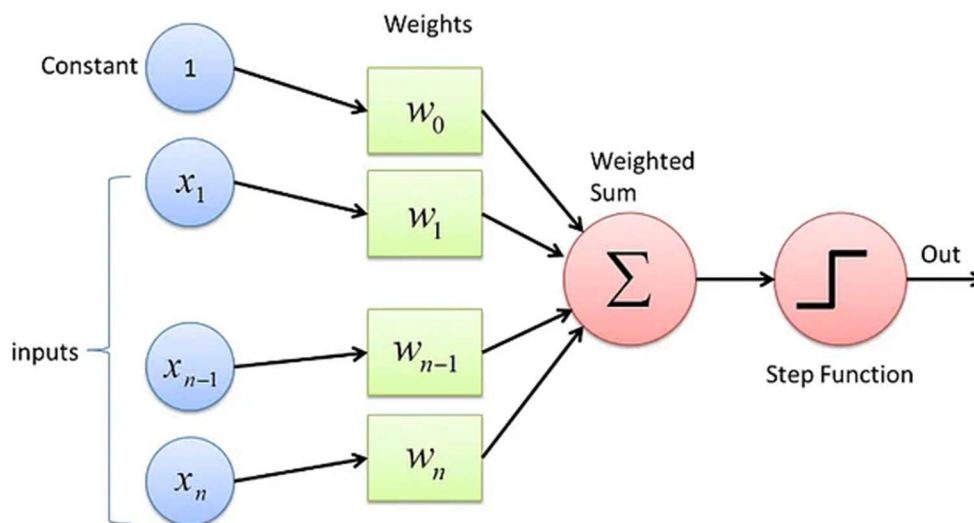
1. Λήψη Σήματος:
Ο νευρώνας λαμβάνει ηλεκτρικά σήματα από τις συνάψεις με άλλους νευρώνες μέσω των δενδριτών του. Αυτά τα σήματα είναι ηλεκτρικά φορτισμένοι παλμοί από άλλους νευρώνες.
2. Ενσωμάτωση Σήματος:
Τα σήματα που λαμβάνονται από τους δενδρίτες ενσωματώνονται στο σώμα του νευρώνα όπου σταθμίζεται το κάθε ηλεκτρικό φορτίο με το συνοπτικό βάρος. Εκεί, η συνολικοί εισερχόμενη διέγερση των σημάτων εξετάζονται για να ληφθεί μια απόφαση.
3. Εκπομπή Σήματος:
Εάν η συνολική εισερχόμενη διέγερση ξεπεράσει ένα καθορισμένο όριο, τότε ο νευρώνας αποφασίζει να εκπέμψει έναν ηλεκτρικούς παλμού. Αυτοί οι παλμοί, οι οποίοι είναι ηλεκτρικά φορτισμένοι, διαδίδονται κατά μήκος του άξονα του νευρώνα.

Συνοψίζοντας, οι δενδρίτες λοιπόν λειτουργούν ως η είσοδος του νευρώνα, λαμβάνοντας τα σήματα από άλλους νευρώνες και εισάγοντάς τα στο σώμα του νευρώνα. Η ηλεκτρική τάση του νευρώνα, επηρεάζεται από τα εισερχόμενα σήματα και αν ξεπεράσει ένα καθορισμένο κατώφλι, τότε ο νευρώνας εκπέμπει ηλεκτρικούς παλμούς αλλιώς παραμένει αδρανής.

2.5.3 Η δομή του τεχνητού νευρώνα

Η δομή του Τεχνητού Νευρώνα μοιάζει με τη δομή του βιολογικού Νευρώνα, διότι όπως βλέπουμε το «Σχήμα 2.3» μπορούμε να το χωρίσουμε επίσης σε 4 κυρίως μέρη.

1. Είσοδοι (οι αντίστοιχοι βιολογικοί δενδρίτες):
ο Τεχνητός νευρώνας δέχεται, σήματα - συνεχείς μεταβλητές ,και όχι ηλεκτρικά όπως στον βιολογικό νευρώνα ,στις εισόδους του (εικόνες, σήματα από αισθητήρες, κτλ).
2. Βάρη (οι αντίστοιχες βιολογικές συνάψεις):
Αυτά τα σήματα επηρεάζονται άμεσα από τα βάρη με τα οποία πολλαπλασιάζονται τα σήματα εισόδου, τα βάρη ουσιαστικά είναι οι συνάψεις που έχουν οι βιολογικοί νευρώνες. Έχουμε όμως και μια επιπλέον είσοδο η οποία ονομάζεται πόλωση του νευρώνα.
3. Σώμα (κρυφό επίπεδο) (το αντίστοιχο βιολογικό σώμα):
Στο σώμα του τεχνητού νευρώνα έχει δύο στάδια το πρώτο στάδιο κάνει άθροιση όλων των εισερχόμενων, επηρεασμένων από βάρη, σημάτων και παράγει την ποσότητα S ουσιαστικά έχει έναν αθροιστή. Το δεύτερο έχει τη συνάρτηση ενεργοποίησης, η οποία είναι ένα είδος φίλτρου που δέχεται την ποσότητα S και διαμορφώνει την τελική τιμή του σήματος εξόδου.
4. Έξοδος. (ο αντίστοιχος βιολογικός Άξονας)
Η έξοδος είναι μόνο μία π.χ 0 ή 1, αλλά όπως και στην απόληξη του βιολογικού άξονα μπορεί να συνδεθεί με πολλούς εισόδους από άλλους νευρώνες.



Source

Σχήμα 2.3 Δομή Τεχνητού Νευρώνα

[πηγή: <https://medium.com/@cprajenjit32/perceptron-a-simple-yet-mighty-machine-learning-algorithm-9ff6b7d86a71>]

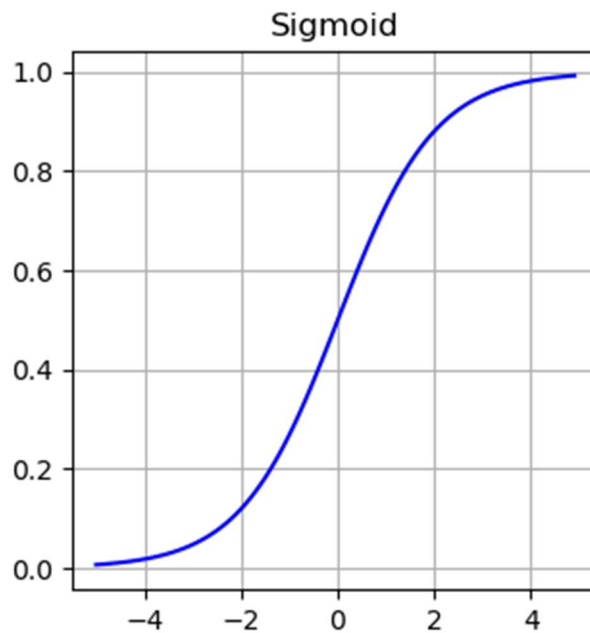
2.5.4 Συναρτήσεις ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης χρησιμοποιούνται στα νευρωνικά δίκτυα για να εισάγουν μη γραμμικότητα στις ενεργοποιήσεις των νευρώνων. Αυτό επιτρέπει στα δίκτυα να μαθαίνουν πολύπλοκες σχέσεις μεταξύ των χαρακτηριστικών των δεδομένων εισόδου.

Ορισμένες από τις πιο κοινές συναρτήσεις ενεργοποίησης διατυπώνονται παρακάτω με τις εκφράσεις και τις γραφικές παραστάσεις των συναρτήσεων:

Σιγμοειδής «Σχήμα 2.4» (Sigmoid): Μετατρέπει τις εισόδους σε μια καμπύλη σιγμοειδούς μορφής, μεταξύ 0 και 1. Χρησιμοποιείται συνήθως σε δίκτυα που εκτελούν δυαδική ταξινόμηση και εκφράζεται με τη σχέση (2.1)

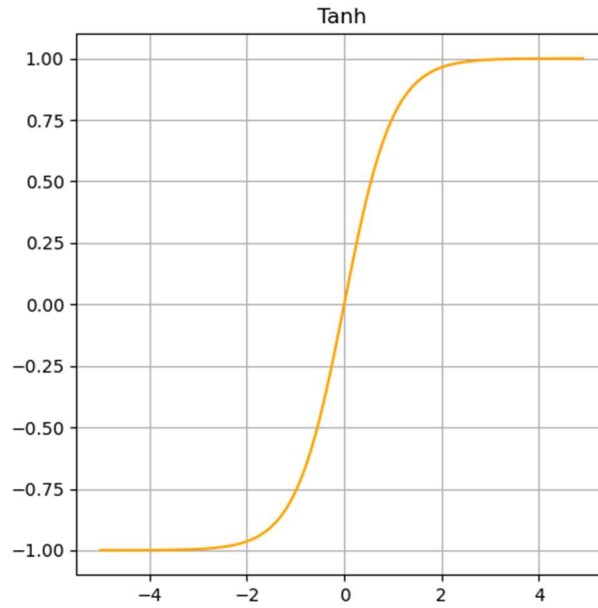
$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$



Σχήμα 2.4 Σιγμοειδής συνάρτηση

Υπερβολική Εφαπτομένη «Σχήμα 2.5» (Hyperbolic Tangent - Tanh): Παρόμοια με τη σιγμοειδή συνάρτηση, αλλά μετατρέπει τις εισόδους σε μια καμπύλη μεταξύ -1 και 1 και εκφράζεται με τη σχέση (2.2)

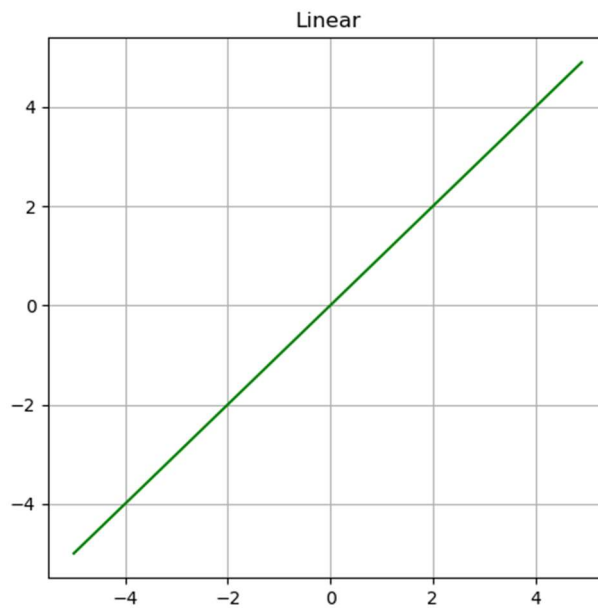
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$



Σχήμα 2.5 Υπερβολική εφαπτομένη

Γραμμική «Σχήμα 2.6» (Linear): Δεν εφαρμόζει καμία μετατροπή στην είσοδο. Χρησιμοποιείται συνήθως σε επίπεδα εξόδου. Εκφράζεται με τη σχέση (2.3)

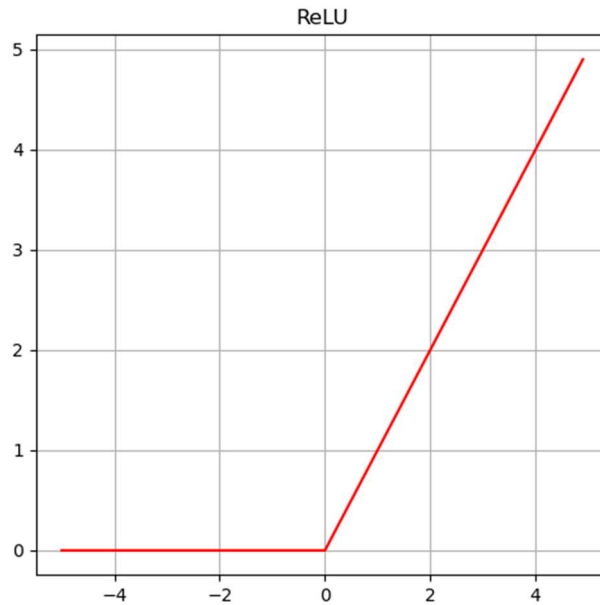
$$f(x) = x \quad (2.3)$$



Σχήμα 2.6 Γραμμική

Ανορθωμένη γραμμική μονάδα «Σχήμα 2.7» (Rectified Linear Unit): Αποτελείται από μια γραμμική λειτουργία που είναι μηδενική για αρνητικές τιμές και ταυτίζεται με την είσοδο για θετικές τιμές. Είναι μια από τις πιο δημοφιλείς συναρτήσεις ενεργοποίησης και εκφράζεται με τη σχέση (2.4)

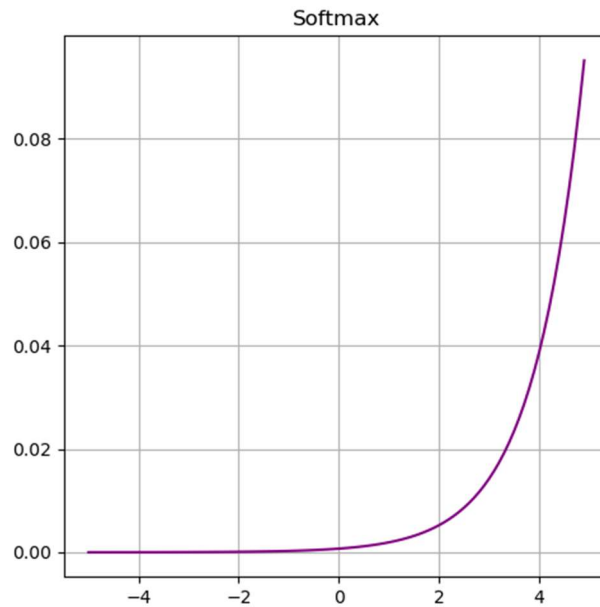
$$\mathbf{ReLU}(x) = \mathbf{max}(0, x) \quad (2.4)$$



Σχήμα 2.7 Ανορθωμένη γραμμική μονάδα

Συνάρτηση «Σχήμα 2.8» (Softmax): Χρησιμοποιείται συνήθως στο τελευταίο επίπεδο ενός νευρωνικού δικτύου για ταξινόμηση περισσότερων από δύο κατηγοριών, καθώς μετατρέπει τις εισόδους σε πιθανότητες που αθροίζονται σε 1. Εκφράζεται με τη σχέση (2.5)

$$\mathbf{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j^N e^{x_j}} \quad (2.5)$$



Σχήμα 2.8 Softmax

2.5.5 Πόλωση

Η σταθερά πόλωσης b (bias) σε ένα νευρωνικό δίκτυο αναπαριστά έναν πρόσθετο όρο που προστίθεται στον γινόμενο της εισόδου και των βαρών για κάθε νευρώνα σε ένα επίπεδο. Η προσθήκη αυτής της σταθεράς επιτρέπει στο νευρώνα να μάθει πιο σύνθετες συναρτήσεις από τα δεδομένα εισόδου.

Η είσοδος αυτή έχει ουσιαστικά τιμή πάντα 1, άρα η τιμή της εισόδου επηρεάζεται αποκλειστικά από το βάρος b_0 . Αν υποθέσουμε ότι έχουμε ένα νευρώνα με συνάρτηση ενεργοποίησης Ανορθωμένη γραμμική μονάδα (Relu). Που εκφράζεται με τη σχέση (2.6)

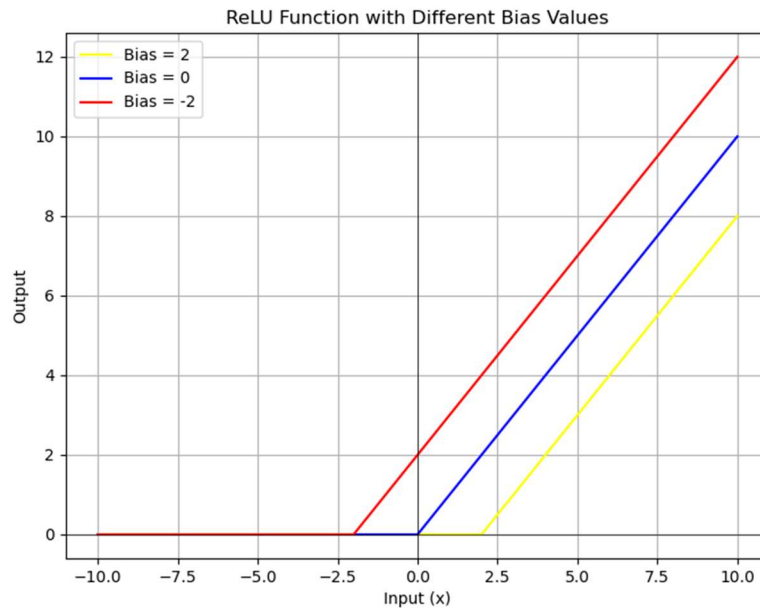
$$\mathbf{RELU}(X) = \mathbf{max}(0, x) \quad (2.6)$$

Έχοντας τη πόλωση η παραπάνω έκφραση θα γίνει:

$$\mathbf{RELU}(x) = \mathbf{max}(0, x + x0 * b) \text{ το } x0 \text{ είναι } 1 \text{ άρα} \quad (2.7)$$

$$\mathbf{RELU}(x) = \mathbf{max}(0, x + b)$$

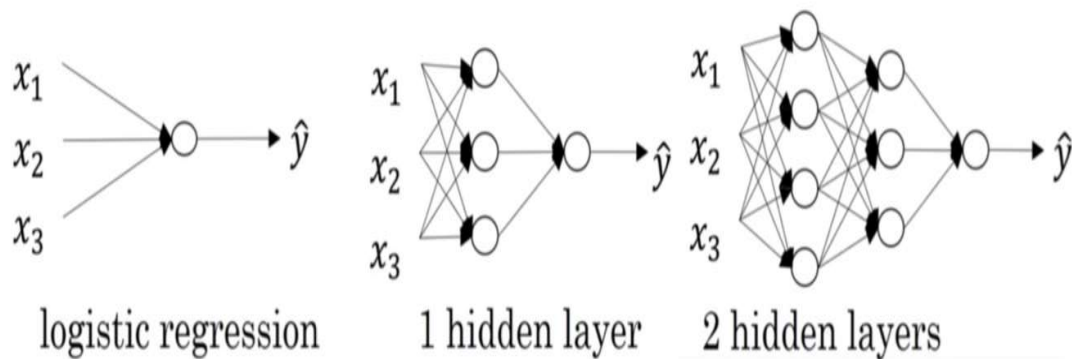
Προσθέτοντας την είσοδο πόλωσης μετατοπίζουμε την συνάρτηση ενεργοποίησης στον άξονα x αριστερά για αρνητικές τιμές του b και δεξιά για θετικές τιμές του b «».



Σχήμα 2.9 Μετατόπιση της συνάρτησης ενεργοποίησης λόγω πόλωσης.

2.6 Νευρωνικά Δίκτυα

Οι τεχνητή νευρώνες στις χρήση τους συνήθως αποτελούνται από συστοιχίες τεχνητών νευρώνων ή καλύτερα από επίπεδα νευρώνων. Έτσι με βάσει τα πόσα επίπεδα νευρώνων υπάρχουν χωρίζονται σε ρηγά νευρωνικά δίκτυα (Shallow Neural networks) και βαθιά νευρωνικά δίκτυα (Deep Neural Networks) θα αναφερθούμε με λεπτομέρεια στο επόμενο κεφάλαιο. Τα ρηγά νευρωνικά δίκτυα αποτελούνται συνήθως από το επίπεδο εισόδου, ένα με δυο κρυφά επίπεδα και το επίπεδο εξόδου «Σχήμα 2.10». Τα κρυφά επίπεδα είναι αυτά τα οποία περιέχουν τους νευρώνες που γίνεται η εκπαίδευση του μοντέλου. Σε αυτά τα μοντέλα λέμε ότι επιτυγχάνεται κατά κάποιο τρόπο ρηγή τεχνητή μάθηση η απλά Τεχνητή Μάθηση.



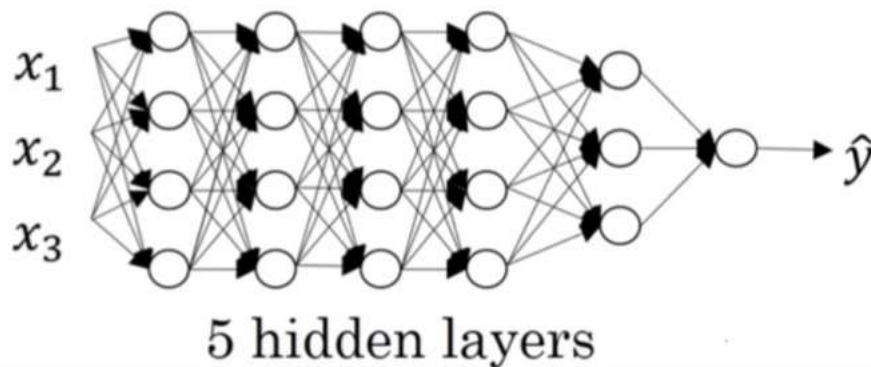
Σχήμα 2.10 Αρχιτεκτονική ρηχών Νευρωνικών Δικτύων

[πηγή: <https://velog.io/@hseop/Deep-Neural-Networks-Module-4-Deep-Neural-Network>]

Το πιο δημοφιλές μοντέλο ρηχής μάθησης είναι το Perceptron, το οποίο αποτελείται από έναν επίπεδο νευρώνων που συνδέονται άμεσα με την είσοδο. Το perceptron δέχεται είσοδο από τις χαρακτηριστικές μεταβλητές, πολλαπλασιάζει κάθε είσοδο με το αντίστοιχο βάρος και εφαρμόζει μια συνάρτηση ενεργοποίησης (συνήθως σιγμοειδή ή ανορθωμένη γραμμική μονάδα). Οι έξοδοι αυτών των υπολογισμών συνδυάζονται για να δοθεί η τελική έξοδος.

2.6.1 Βαθιά νευρωνικά δίκτυα

Τα βαθιά νευρωνικά δίκτυα συνήθως δομούνται με πάνω από δύο κρυφά επίπεδα και συνήθως πολύ περισσότερα από δύο επίπεδα «Σχήμα 2.11». Δημιουργήθηκαν με στόχο να λύνουν πολύπλοκα προβλήματα άλλα απέδειξαν ότι και στα απλά προβλήματα υπερτερούν σε σχέση με τα ρηχά. Η βαθιά δομή τους επιτρέπει την ανίχνευση χαρακτηριστικών σε διαφορετικά επίπεδα αφαιρώντας σταδιακά πιο αφηρημένα χαρακτηριστικά καθώς προχωρούν στα υψηλότερα επίπεδα. Οι έρευνα για τα Βαθιά νευρωνικά δίκτυα αποτελούν το αντικείμενο του τομέα της Βαθιάς μάθησης. Μάλιστα ο όρος Βαθιά μάθηση ουσιαστικά αφορά τα επίπεδα τα οποία απαρτίζουν το συγκεκριμένο δίκτυο που όπως προαναφέρθηκε συνήθως είναι πολλά.



Σχήμα 2.11 Βαθύ Νευρωνικό Δίκτυο με 5 κρυφά επίπεδα

[πηγή: <https://velog.io/@hseop/Deep-Neural-Networks-Module-4-Deep-Neural-Network>]

2.6.2 Συνελκτικά νευρωνικά δίκτυα

Σε αυτό το σημείο κρίνεται σκόπιμο να αναφερθούμε στα Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks) εν συντομία CNN στα οποία είναι στηριγμένη και η εργασία αυτή. Τα CNN είναι τα ποιο διαδεδομένα βαθιά νευρωνικά δίκτυα, τα οποία έχουν χρήση κυρίως σε ταξινόμηση εικόνων, αναγνώριση αντικειμένων, αναγνώριση προσώπων, αναγνώριση κειμένου και σε πολλές άλλες χρήσεις.

Τα Συνελκτικά Νευρωνικά Δίκτυα είναι η προσπάθεια προσομοίωσης του τμήματος του εγκεφάλου που είναι υπεύθυνο για την όραση μας. Η έρευνα η οποία ενέπνευσε τη δημιουργία Συνελκτικών Δικτύων είναι των David H. Hubel και Torsten Wiesel η οποία αφορούσε την λειτουργία των νευρώνων στο λοβό του αμφιβληστροειδούς. Οι νευρώνες αυτοί έχουν τη δυνατότητα να αντιδρούν σε συγκεκριμένα χαρακτηριστικά των εικόνων, όπως άκρες και γωνίες. Αυτές οι ανακαλύψεις συνέβαλαν στην κατανόηση του τρόπου λειτουργίας του ανθρώπινου αμφιβληστροειδούς.

Την αρχή για το υπολογιστικό κομμάτι των CNN αποτέλεσε το Μοντέλο του Kunihiko Fukushima το 1980 με το μοντέλο ενός νευρωνικού δικτύου που ονομάστηκε “Neocognitron”, το οποίο είναι ένα πρόγραμμα που επιδιώκει να αντιγράψει τη λειτουργία του ανθρώπινου εγκεφάλου σε σχέση με την αναγνώριση μοτίβων σε εικόνες. Το Neocognitron εμπνεύστηκε από την ανατροφοδότηση που συμβαίνει στον οπτικό φλοιό του εγκεφάλου και περιλάμβανε μια ιεραρχία από στρώματα νευρώνων που επεξεργάζονταν την είσοδο σταδιακά. Το μοντέλο αυτό χρησιμοποιήθηκε για την αναγνώριση προτύπων σε εικόνες και ήταν ένα από τα πρώτα που ενέπνευσαν τη δημιουργία των σύγχρονων Convolutional Neural Networks (CNNs). Στη συνέχεια τη σκυτάλη ανέλαβε ο Yann LeCun ο οποίος βασισμένος στην εργασία του Fukushima (ουσιαστικά είναι ο συνεχιστής εκείνης της έρευνας) ανέπτυξε ένα άλλο πιο λεπτομερές μοντέλο ΝΔ το 1989, που ονομάστηκε LeNet-5. Το LeNet-5 ήταν ένα από τα πρώτα πλήρως λειτουργικά CNNs και χρησιμοποιήθηκε ευρέως για την αναγνώριση χειρόγραφων ψηφίων σε εικόνες. Αυτό το μοντέλο αποτέλεσε ένα σημαντικό βήμα στην ανάπτυξη

των CNNs και έθεσε τις βάσεις για πολλές εφαρμογές στον τομέα της υπολογιστικής όρασης και της μηχανικής μάθησης.

Τα CNNs δεν αντιμετωπίζουν τα δεδομένα στην είσοδο σε διανυσματική μορφή, δηλαδή μια σειρά από τιμές των pixels αλλά ως πίνακες δύο και τριών διαστάσεων. Αυτό έχει ως αποτέλεσμα να διατηρείται η χωρική πληροφορία που με την μετατροπή σε μία σειρά από τιμές των pixels χανόταν.

2.6.3 Στάδια επεξεργασίας των δεδομένων στα CNNs

Τα CNNs περιλαμβάνουν τρία στάδια επεξεργασίας των δεδομένων (εικόνων). Τα οποία αναλυτικά και με σειρά λειτουργίας είναι:

- Πρώτο στάδιο, “Συνέλιξη”.

Στη Συνέλιξη έχουμε ένα σύνολο από φίλτρα εντοπισμού χαρακτηριστικών (αναφέρονται ως kernels ή feature detectors), τα χαρακτηριστικά αυτά μπορεί να είναι ακμές, σχήματα, ή χρώματα, τα οποία βρίσκονται σε διάφορα σημεία της εικόνας. Σε αυτά τα χαρακτηριστικά που θα εξαχθούν από το στάδιο της συνέλιξης θα εκπαιδευτεί και το δίκτυο(μοντέλο).

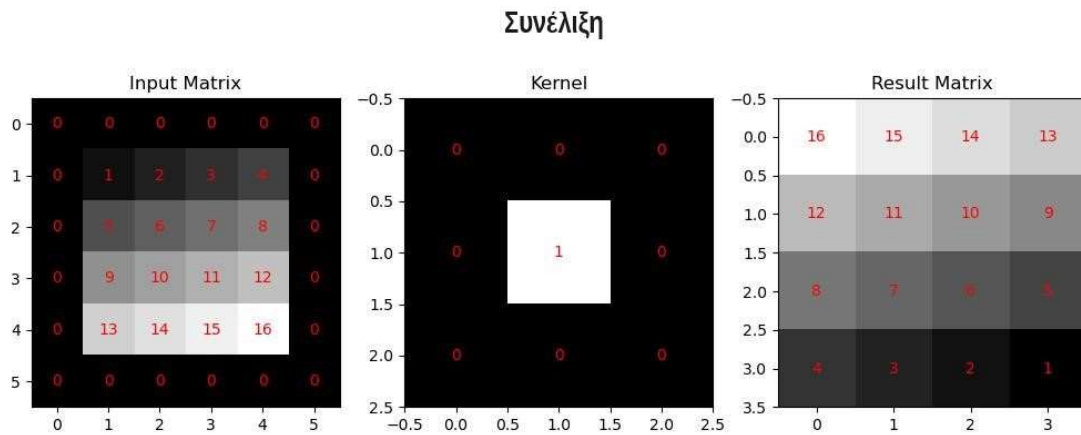
Αυτό που ουσιαστικά γίνεται στη συνέλιξη σε ένα συνελκτικό νευρωνικό δίκτυο, ένα φίλτρο ή kernel περνά πάνω από την είσοδο, χωρίζοντάς την σε μικρότερα τμήματα ή παραθύρα. Για κάθε παράθυρο, η συνέλιξη συνίσταται στο να πολλαπλασιάζονται οι τιμές του πίνακα του φίλτρου με τις αντίστοιχες τιμές του παραθύρου εισόδου. Αυτές οι πολλαπλασιάσεις πραγματοποιούνται για κάθε παράθυρο εισόδου καθώς το φίλτρο κινείται πάνω από την είσοδο.

Στη συνέχεια, τα αποτελέσματα των πολλαπλασιασμών συνοψίζονται σε έναν μεμονωμένο αριθμό με την εκτέλεση μιας συνάρτησης σύνοψης, όπως η πρόσθεση, και το αποτέλεσμα αυτό αποτελεί την έξοδο της συνέλιξης. Αυτή η διαδικασία επαναλαμβάνεται για κάθε θέση του φίλτρου πάνω στην είσοδο, παράγοντας έτσι έναν χάρτη χαρακτηριστικών ή feature map ως έξοδο της συνέλιξης.

Στη πραγματικότητα η συνέλιξη έχει επικρατήσει σαν όρος αλλά δεν εφαρμόζεται συνέλιξη (convolution), καθώς η μαθηματική διαδικασία που εφαρμόζεται είναι η Cross-Correlation (διασταυρούμενη συσχέτιση). Στη πραγματικότητα αυτές οι δύο μαθηματικές διαδικασίες είναι πανομοιότυπες και όπως θα δούμε η μόνη διαφορά της Convolution είναι ότι αναστρέφει το πίνακα και ως προς x αλλά και ως προς y. Η Συνέλιξη συμβολίζεται με $G = h * f$ και η μαθηματική της έκφραση είναι η σχέση (2.8)

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k h[u, v] F[i - u, i - v] \quad (2.8)$$

και έχει ως αποτέλεσμα το «Σχήμα 2.12»:

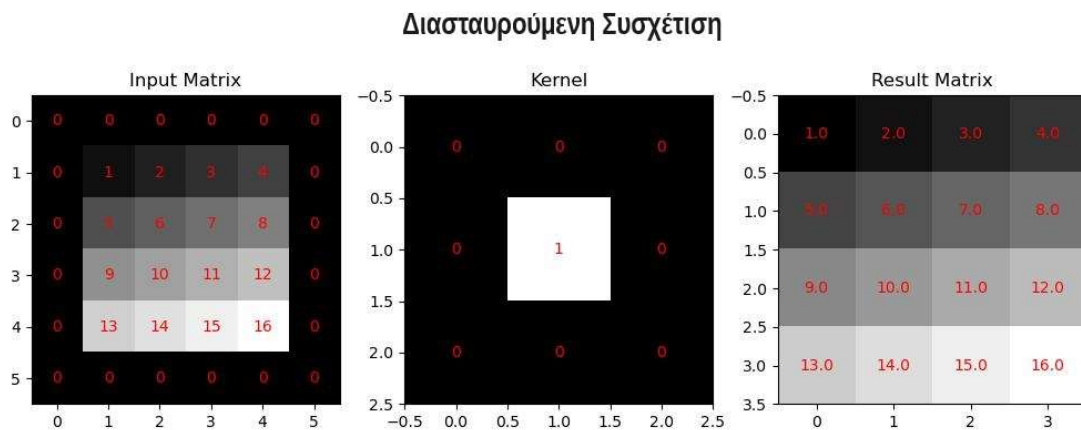


Σχήμα 2.12 Είσοδος φίλτρο και αποτέλεσμα της συνέλιξης

Ενώ η διασταυρούμενη συσχέτιση συμβολίζεται με $G = h \otimes f$ και η μαθηματική της έκφραση είναι η σχέση (2.9):

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k h[u, v] F[i + u, i + v] \quad (2.9)$$

και έχει ως αποτέλεσμα την «Σχήμα 2.13»



Σχήμα 2.13 Είσοδος, φίλτρο και το αποτέλεσμα της διασταυρούμενης συσχέτισης

Όπως παρατηρούμε η διαφορά στις δύο εκφράσεις είναι στη τελευταία αγκύλη στη συνέλιξη κάνουμε αφαίρεση ενώ αντίθετα στη διασταυρούμενη συσχέτιση κάνουμε πρόσθεση.

Για να δώσουμε 1 παράδειγμα του υπολογισμού που γίνεται παρακάτω «Σχήμα 2.14» βλέπουμε ένα πίνακα εισόδου, ένα φίλτρο και το αποτέλεσμα εφαρμόζουμε το φίλτρο αρχικά πάνω αριστερά (κόκκινο περίγραμμα) και έχουμε:

$$G = 1 * 1 + 0 * 2 + (-1) * 3 + 1 * 5 + 0 * 6 + (-1) * 7 + 1 * 9 + 0 * 10 + (-1) * 11 = -6$$

Πάνω δεξιά με το πράσινο περίγραμμα έχουμε:

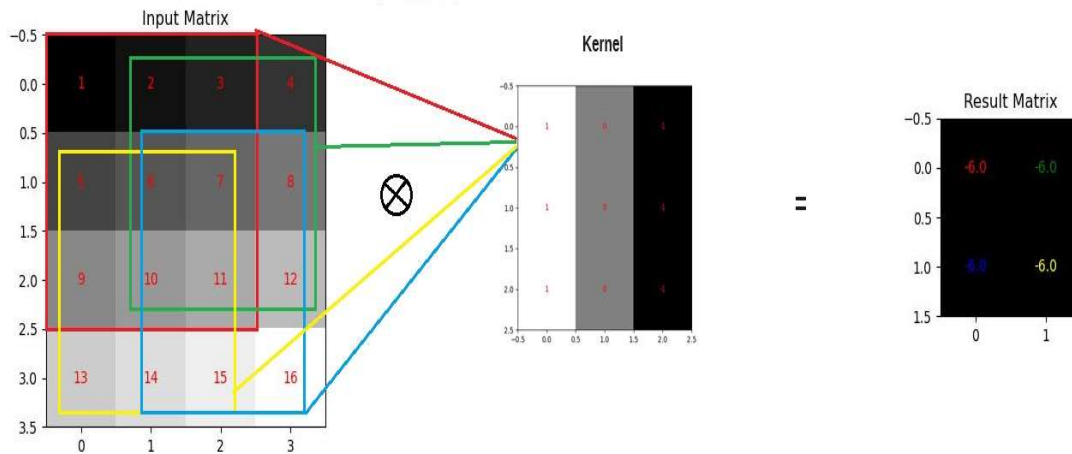
$$G = 1 * 2 + 0 * 3 + (-1) * 4 + 1 * 6 + 0 * 7 + (-1) * 8 + 1 * 10 + 0 * 11 + (-1) * 12 = -6$$

Κάτω αριστερά με το κίτρινο περίγραμμα έχουμε:

$$G = 1 * 5 + 0 * 6 + (-1) * 7 + 1 * 9 + 0 * 10 + (-1) * 11 + 1 * 13 + 0 * 14 + (-1) * 15 = -6$$

Κάτω δεξιά με το μπλε περίγραμμα έχουμε:

$$G = 1 * 6 + 0 * 7 + (-1) * 8 + 1 * 10 + 0 * 11 + (-1) * 12 + 1 * 14 + 0 * 15 + (-1) * 16 = -6$$



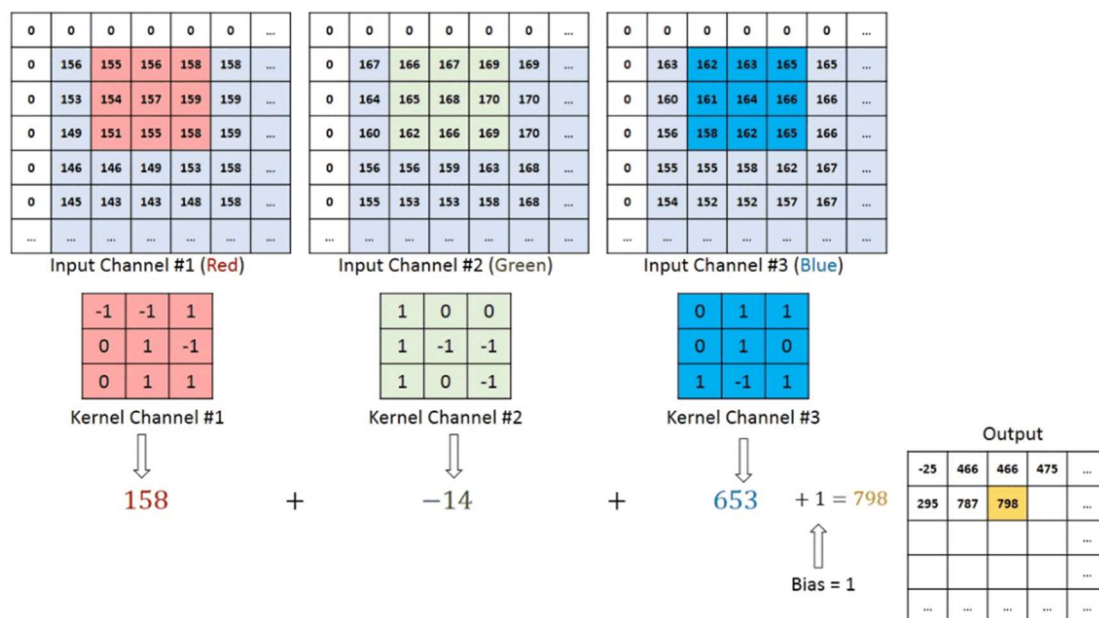
Σχήμα 2.14 Εφαρμογή συνέλιξης

Όπως παρατηρούμε αυτό έχει ως αποτέλεσμα τρία προβλήματα:

- Απώλεια πληροφοριών στα άκρα: Κατά τη συνέλιξη, τα άκρα της εισόδου έχουν λιγότερες ευκαιρίες να συμμετάσχουν σε πολλαπλά στάδια συνέλιξης. Αυτό μπορεί να οδηγήσει σε απώλεια πληροφοριών σε αυτές τις περιοχές της εικόνας.
- Εξόρυξη συνόρων: Καθιστά δυσκολότερο τον εντοπισμό συνόρων και άλλων σημαντικών χαρακτηριστικών στις περιοχές των άκρων της εικόνας.
- Διαστάσεις εξόδου: Μειώνει τις διαστάσεις του τελικού πίνακα εξόδου μετά τη συνέλιξη. Αυτό μπορεί να επηρεάσει τη δυνατότητα του δικτύου να μάθει και να αναπαραστήσει σωστά τα χαρακτηριστικά της εικόνας.

Γι' αυτό το λόγο προσθέτουμε κελιά περιμετρικά του πίνακα, όπου τα γεμίζουμε με μηδενικά για να μην έχουν επιρροή «Σχήμα 2.13» στο input matrix. Αυτό ονομάζεται padding και συγκεκριμένα zero-padding

Στο παράδειγμα που είδαμε κάνουμε ‘‘συνέλιξη’’ σε πίνακα 2 διαστάσεων δηλαδή σε gray-scale εικόνα σε μια έγχρωμη εικόνα ως είσοδο έχουμε τρεις τέτοιους πίνακες που ο καθένας αντιπροσωπεύει 1 από τα κύρια χρώματα δηλαδή το κόκκινο το πράσινο και το μπλε(RGB). Η συνέλιξη γίνεται με τον ίδιο τρόπο και στους τρεις νέους πίνακες και το αποτέλεσμα από κάθε πίνακα προστίθεται και προκύπτει ένας δυσδιάστατος πίνακας. «Σχήμα 2.15»



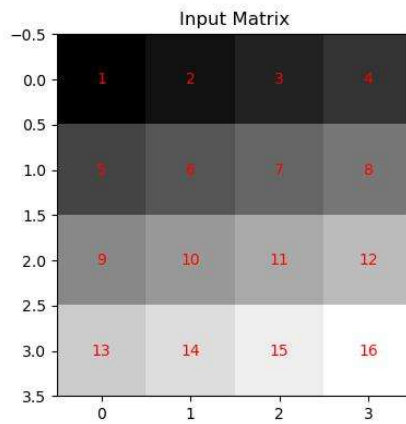
Σχήμα 2.15 Συνέλιξη σε έγχρωμη εικόνα

[πηγή: <https://stats.stackexchange.com/questions/534080/how-convolutional-layer-work-exactly-in-rgb-image-processing>]

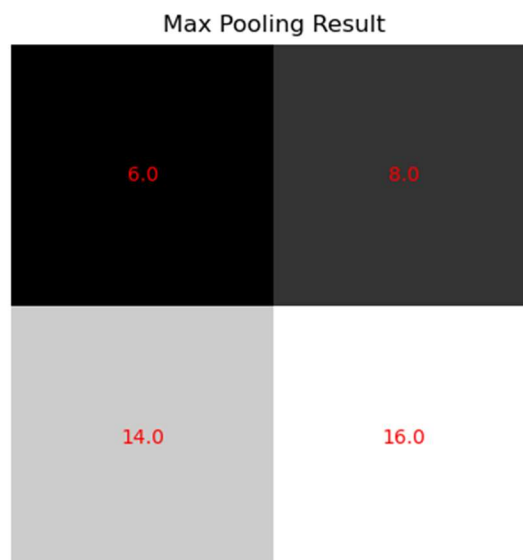
- Δεύτερο στάδιο, σμίκρυνσης (pooling).

Στα συνελκτικά νευρωνικά δίκτυα, μετά από κάθε κρυφό επίπεδο που περιλαμβάνει πολλά φίλτρα συνέλιξης, συνήθως ακολουθεί ένα επίπεδο σμίκρυνσης (pooling layer). Το επίπεδο αυτό έχει ως στόχο τη μείωση της χωρικής διάστασης των χαρακτηριστικών χωρίς να αλλάξει την αριθμότητα τους. Αυτό γίνεται με τη συνάθροιση περιοχών των χαρακτηριστικών και τη λήψη ενός εκπροσώπου για κάθε περιοχή. Τα είδη των φίλτρων σμίκρυνσης περιλαμβάνουν:

Max Pooling: Επιλέγει το μέγιστο τιμές από κάθε περιοχή των χαρακτηριστικών. Για παράδειγμα ας πούμε ότι έχουμε ένα πίνακα με διάσταση 4x4 «Σχήμα 2.16» και εφαρμόζουμε ένα επίπεδο σμίκρυνσης 2x2 με βήμα 2(stride) τότε ο νέος πίνακας θα είναι 2x2 και θα περιέχει το μέγιστο αριθμό από κάθε 2x2 επίπεδο σμίκρυνσης που εφαρμόστηκε και στο συγκεκριμένο είναι 4 επίπεδα και συγκρίνανε το 1,2,5,6 το 3,4,7,8 το 9,10,13,14 και το 11,12,15,16 ο νέος πίνακας 2x2 που προέκυψε είναι «Σχήμα 2.17».

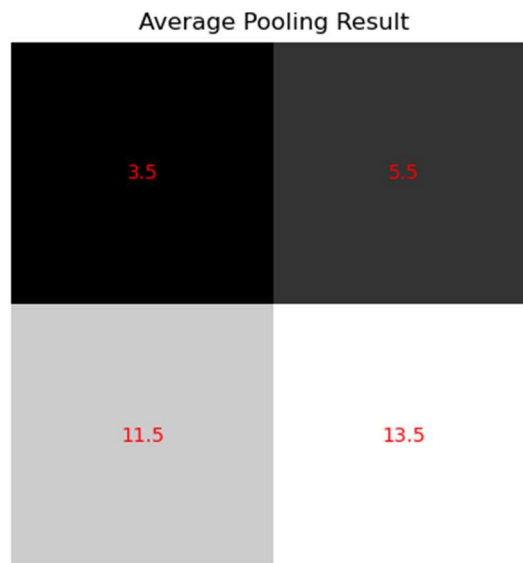


Σχήμα 2.16 Πίνακα διάστασης 4x4



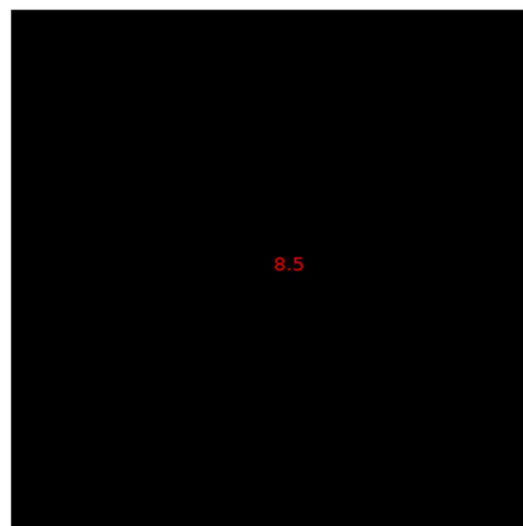
Σχήμα 2.17 Πίνακας 2x2 αποτέλεσμα του maxpooling

Average Pooling: Υπολογίζει τον μέσο όρο των τιμών από κάθε περιοχή των χαρακτηριστικών. Για παράδειγμα ας πούμε ότι έχουμε ένα πίνακα με διάσταση 4x4 «Σχήμα 2.16» και εφαρμόζουμε ένα επίπεδο σμίκρυνσης 2x2 με βήμα 2(stride) τότε ο νέος πίνακας θα είναι 2x2 και θα περιέχει το μέσο όρο από κάθε 2x2 επίπεδο σμίκρυνσης που εφαρμόστηκε και στο συγκεκριμένο είναι 4 επίπεδα και συγκρίνανε το 1,2,5,6 το 3,4,7,8 το 9,10,13,14 και το 11,12,15,16 ο νέος πίνακας 2x2 που προέκυψε είναι «Σχήμα 2.18».



Σχήμα 2.18 Πίνακας 2x2 αποτέλεσμα average pooling

Global Average Pooling: Υπολογίζει τον μέσο όρο των τιμών όλων των χαρακτηριστικών. Οπότε ο χάρτης χαρακτηριστικών μετά την εφαρμογή του Global Average Pooling θα έχει μόνο μία τιμή «Σχήμα 2.19».



Σχήμα 2.19 Μία μόνο τιμή αποτέλεσμα του Global average pooling

Το φίλτρο σμίκρυνσης μειώνει την αριθμητική πολυπλοκότητα του μοντέλου και τον αριθμό των παραμέτρων, αυξάνει την ανεκτικότητα στη μετατόπιση (translation invariance) και να μειώνει τον κίνδυνο υπερπροσαρμογής.

- Τρίτο στάδιο, πλήρως συνδεδεμένα επίπεδα (fully connected layers).

Τα πλήρως συνδεδεμένα επίπεδα είναι το τελικό στάδιο του CNN εδώ οι δυσδιάστατοι πίνακες που προέρχονται από τα προηγούμενα επίπεδα μετατρέπονται σε μονοδιάστατους πίνακες. Συνήθως έχει δύο επίπεδα με συνάρτηση ενεργοποίησης RELU και ένα ακόμη τελευταίο επίπεδο softmax. Τα στάδια αυτά και τρία μαζί μας κάνουν τον ταξινομητή ο οποίος δίνει το αποτέλεσμα της κλάσης της εικόνας σύμφωνα με τα χαρακτηριστικά που έχουν εξαχθεί από τα προηγούμενα επίπεδα.

2.7 Αξιολόγηση των CNNs

Πολλά από τα προβλήματα ταξινόμησης που καλούνται να λύσουν τα CNNs είναι με 2 κλάσεις, όπου τα δεδομένα χωρίζονται σε πρότυπα με A κατάσταση και πρότυπα με B κατάσταση. Άσχετα ως προς ποια κατάσταση (A προς B ή B προς A) θέλουμε να εξετάσουμε το πρόβλημα στις δύο κλάσεις, η έξοδος που παίρνουμε πάντα είναι 0 ή 1.

Για παράδειγμα ένα από τα διαδεδομένα πρόβλημα ταξινόμησης που κυκλοφορούν στο διαδίκτυο είναι η μόλυνση από ελονοσία. Σε αυτό το παράδειγμα το CNN εκπαιδεύεται σε 2 κλάσεις, η μια κλάση έχει δεδομένα (εικόνες από μικροσκόπιο) με ερυθρά αιμοσφαίρια προσβεβλημένα από ελονοσία και η άλλη κλάση έχει δεδομένα (εικόνες από μικροσκόπιο) με υγιή ερυθρά αιμοσφαίρια. Άρα το CNN αφού εκπαιδευτεί καλείται να προβλέψει αν είναι μολυσμένα η υγιή.

Αυτό όμως δεν συμβαίνει στη πράξη καθώς το σύστημα υπόκειται σε λάθη δηλαδή ενώ περιμένουμε να αντιστοιχεί κάθε μολυσμένο δεδομένο στην κλάση των μολυσμένων και κάθε υγιές στην κλάση των υγιή, κάποια από τα μολυσμένα τα αντιστοιχεί στην κλάση των υγιή και κάποια υγιή τα αντιστοιχεί στη κλάση των μολυσμένων. Έτσι έχουμε 4 περιπτώσεις αντί για δυο, ας υποθέσουμε ότι για τα υγιή η έξοδος του CNN μας δίνει 0 και για τα μολυσμένα μας δίνει 1 Βλέπουμε τις εξής περιπτώσεις:

- Πρώτη περίπτωση να προβλεφθεί ως υγιές και όντως να ανήκει στη κλάση των υγιή. (Αυτή η περίπτωση ονομάζεται Αληθώς αρνητική πρόβλεψη "True Negative" εν συντομία TN).
- Δεύτερη περίπτωση να προβλεφθεί ως υγιές και να ανήκει στην κλάση των μολυσμένων. (Αυτή η περίπτωση ονομάζεται Ψευδός αρνητική πρόβλεψη "False Negative" εν συντομία FN).
- Τρίτη περίπτωση να προβλεφθεί ως μολυσμένο και να ανήκει στη κλάση των μολυσμένων. (Αυτή η περίπτωση ονομάζεται Αληθώς Θετική πρόβλεψη "True Positive" εν συντομία TP).
- Τέταρτη περίπτωση να προβλεφθεί ως μολυσμένο και να ανήκει στην κλάση των υγιή. (Αυτή η περίπτωση ονομάζεται Ψευδός Θετική πρόβλεψη "False Positive" εν συντομία FP).

Ένα πρώτο μέτρο αξιολόγησης της επίδοσης του μοντέλου είναι η ακρίβεια. Το μέτρο της ακρίβειας (accuracy) υπολογίζεται ως ο λόγος των σωστών προβλέψεων (True Positives - TP και True Negatives - TN) προς τον συνολικό αριθμό των προβλέψεων. Για δύο κλάσεις (δυναμική ταξινόμηση), η ακρίβεια μπορεί να υπολογιστεί με τη σχέση (2.10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

Όταν έχουμε περισσότερες από δύο κλάσεις, το μέτρο της ακρίβειας ισχύει εξίσου, αλλά αξιολογεί τη συνολική ικανότητα του μοντέλου να κάνει σωστές προβλέψεις σε όλες τις κλάσεις.

Για παράδειγμα, αν έχουμε τρεις κλάσεις (A, B, C), τότε η ακρίβεια μπορεί να υπολογιστεί ως η μέση ακρίβεια ανά κλάση. Είναι δηλαδή το ποσοστό των σωστών προβλέψεων για κάθε κλάση. Αν έχουμε πίνακα σύγχυσης (confusion matrix) τον οποίο θα αναφέρουμε παρακάτω, μπορούμε να υπολογίσουμε την ακρίβεια για κάθε κλάση ξεχωριστά και να πάρουμε το μέσο όρο αυτών των τιμών. Αυτή η προσέγγιση λειτουργεί για οποιοδήποτε αριθμό κλάσεων.

Το μέτρο της ακρίβειας είναι ένα σημαντικό μέτρο για την αξιολόγηση της απόδοσης ενός μοντέλου, αλλά σε ορισμένες περιπτώσεις μπορεί να μην είναι επαρκές για να κατανοήσουμε πλήρως τη συμπεριφορά του μοντέλου, ειδικά όταν έχουμε μη ισορροπημένα δεδομένα ή διαφορετικά κόστη για τα λάθη ταξινόμησης. Για παράδειγμα αν έχουμε 10 εικόνες με μολυσμένα από ελονοσία ερυθρά αιμοσφαίρια και 500 εικόνες από υγιή ερυθρά αιμοσφαίρια. Ας υποθέσουμε σε αυτό το παράδειγμα ότι το μοντέλο μας προβλέπει με μεγάλη επιτυχία τα υγιή ερυθρά αιμοσφαίρια TN 499 στα 500, ενώ τα μολυσμένα από ελονοσία ερυθρά αιμοσφαίρια δεν τα προβλέπει με μεγάλη επιτυχία TP1 στα 10, λόγω της ανισορροπίας των δεδομένων (10 μολυσμένα 500 υγιή) η ακρίβεια θα έχει λανθασμένα υψηλό ποσοστό το οποίο θα είναι παραπλανητικό.

Γι' αυτό το λόγο προστέθηκαν δυο ακόμα μέτρα αξιολόγησης. Η ευστοχία(Precision) και η ανάκληση(Recall).

Το μέτρο της ευστοχίας μετρά το ποσοστό των προβλέψεων που είναι σωστές αναφορικά με το σύνολο των προβλέψεων που ανήκουν σε μια συγκεκριμένη κατηγορία. Στην ουσία, δείχνει πόσο απόλυτα σωστές είναι οι προβλέψεις μας για μια συγκεκριμένη κατηγορία. Υπολογίζεται ως ο λόγος του αριθμού των σωστών προβλέψεων για μια κατηγορία προς τον συνολικό αριθμό των προβλέψεων που ανήκουν σε αυτήν την κατηγορία Έτσι, μας δίνει μια ιδέα για το πόσο "αξιόπιστο" είναι το μοντέλο όταν προβλέπει μια θετική κατηγορία. Εκφράζεται για συγκεκριμένη κλάση c με τη σχέση (2.11):

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \quad (2.11)$$

Το μέτρο της ανάκλησης είναι ένα μέτρο μέτρησης της ικανότητας ενός ταξινομητή να ανιχνεύει όλα τα δείγματα της θετικής κλάσης σε ένα σύνολο δεδομένων. Στην ουσία, μας δείχνει πόσο αποτελεσματικά ο ταξινομητής ανιχνεύει τα πραγματικά θετικά δείγματα. Ορίζεται ως η αναλογία των πραγματικών θετικών δειγμάτων που ταξινομήθηκαν σωστά ως προς τον συνολικό αριθμό των πραγματικών θετικών δειγμάτων. Εκφράζεται με την ακόλουθη σχέση (2.12):

$$Recall = \frac{TP}{TP + FN} \quad (2.12)$$

Αυτά τα δύο μέτρα από μόνα τους δεν περιγράφουν τη συνολική απόδοση του ταξινομητή των δύο κλάσεων. Για τη συνολική απόδοση του ταξινομητή έχουμε το μέτρο F (F-measure ή F1-score) που ορίζεται ως ο αρμονικός μέσος της ανάκλησης και της ακρίβειας. Εκφράζεται με την ακόλουθη σχέση (2.13):

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.13)$$

Τα δύο μέτρα Precision και Recall όπως βλέπουμε εστιάζουν το αποτέλεσμα στη μία κλάση π.χ για τα μολυσμένα ερυθρά αιμοσφαίρια . Αυτό μπορεί να είναι εντάξει όταν καταρχήν έχουμε την κλάση αυτή ορίσει ως 1 θετική και δεύτερο όταν η κλάση που έχει οριστεί 0 μας είναι αδιάφορη. Όμως στις περισσότερες περιπτώσεις μας ενδιαφέρει η απόδοση του ταξινομητή και στις δύο κλάσεις γι' αυτό έχουν ειπωθεί ακόμα δύο μέτρα, το μέτρο της εξειδίκευσης (specificity) και της ευαισθησίας (sensitivity).

Το μέτρο της εξειδίκευσης είναι ένα μέτρο που μετρά την ικανότητα ενός ταξινομητή να αναγνωρίζει τα αρνητικά παραδείγματα. Συχνά χρησιμοποιείται στο πλαίσιο της ιατρικής διάγνωσης και των διαγνωστικών δοκιμών. Ορίζεται ως η αναλογία των σωστά ταξινομημένων αρνητικών παραδειγμάτων (True Negatives) προς τον συνολικό αριθμό των πραγματικών αρνητικών παραδειγμάτων (True Negatives + False Positives). Η σχέση που την εκφράζει είναι (2.14):

$$Specificity = \frac{TN}{TN + FP} \quad (2.14)$$

Το μέτρο της ευαισθησίας, γνωστό επίσης ως True Positive Rate (TPR) ή Recall, μετρά την ικανότητα ενός ταξινομητή να αναγνωρίζει τα θετικά παραδείγματα. Είναι ένα σημαντικό μέτρο στην αξιολόγηση της απόδοσης ενός μοντέλου, ιδιαίτερα όταν έχουμε μη ισορροπημένα δεδομένα. Το Sensitivity ορίζεται ως η αναλογία των σωστά ταξινομημένων θετικών παραδειγμάτων (True Positives) προς τον συνολικό αριθμό των πραγματικών θετικών παραδειγμάτων (True Positives + False Negatives). Η σχέση που την εκφράζει είναι (2.15):

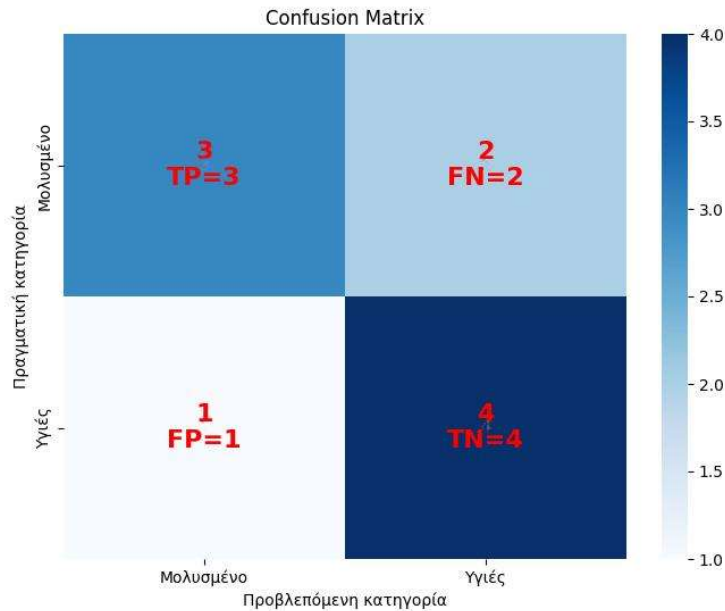
$$Sensitivity = \frac{TP}{TP + FN} \quad (2.15)$$

2.7.1 Πίνακας σύγχυσης (Confusion Matrix)

Ο πίνακας σύγχυσης είναι ένα εργαλείο που χρησιμοποιείται στην αξιολόγηση αλγορίθμων μηχανικής μάθησης για προβλήματα κατηγοριοποίησης. Απεικονίζει τον βαθμό σύγχυσης (confusion) μεταξύ των πραγματικών κλάσεων και των κλάσεων που προέβλεψε ο αλγόριθμος. Στο παραπάνω παράδειγμα με τα ερυθρά αιμοσφαίρια τα οποία είναι προσβεβλημένα από την ελονοσία και τα υγιή

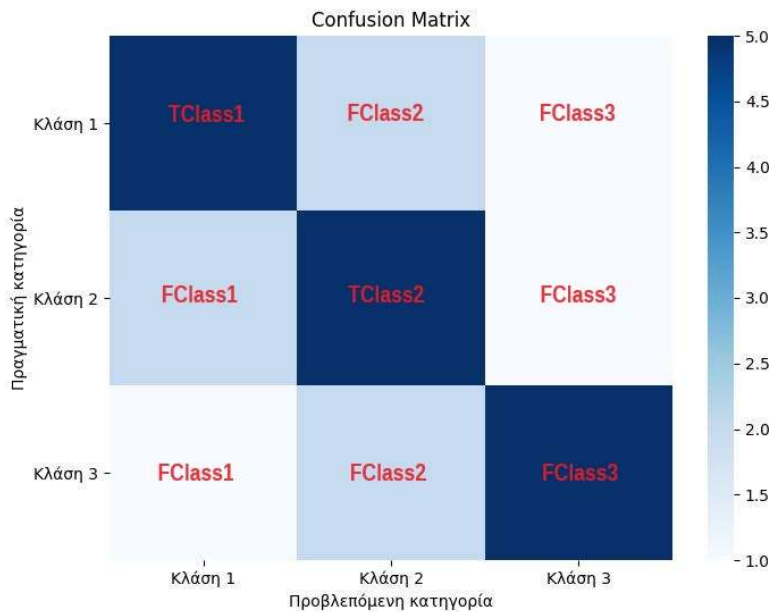
Κεφάλαιο 2

ερυθρά αιμοσφαίρια, όπως είπαμε έχουμε τέσσερις περιπτώσεις TP, TN, FP, FN. Αν υποθέσουμε ότι ταξινομητής που χρησιμοποιήθηκε έκανε 10 προβλέψεις και βρήκε 3 μολυσμένα που ήταν όντως μολυσμένα, 2 υγιή τα οποία ήταν μολυσμένα, 4 υγιή που ήταν όντως υγιή και 1 που ήταν υγιή αλλά το πρόβλεψε ως μολυσμένο τότε ο πίνακας σύγχυσης γίνεται «».



Σχήμα 2.20 Παράδειγμα πίνακα σύγχυσης 2 κλάσεων

Ο πίνακας αυτός για κάθε μία νέα κλάση προσθέτει μία γραμμή και μία στήλη. Αν δηλαδή είχαμε μια ακόμη κλάση ο πίνακας θα γινόταν 3x3 έτσι θα είχαμε 3 πραγματικές κατηγορίες στον άξονα y και 3 κατηγορίες προβλέψεων στον άξονα x. Αρά για την κάθε κλάση έχουμε 1 σημείο στον άξονα x,y το οποίο θα είναι TClass σωστή κλάση και πρόβλεψη και 2 σημεία για την κάθε κλάση που θα είναι FClass λανθασμένη πρόβλεψη λάθος κλάση όπως παρατηρούμε και στο «».



Σχήμα 2.21 Παράδειγμα πίνακα σύγχυσης 3 κλάσεων

2.7.2 Προβλήματα στη μάθηση των CNNs

Αν ο ταξινομητής μας έχει υψηλές επιδόσεις και στην εκπαίδευση αλλά και στη πρόβλεψη τότε μπορούμε να πούμε ότι δεν το μοντέλο δεν παρουσιάζει προβλήματα και γενικεύει καλά. Με τον όρο γενίκευση λέμε την “ικανότητα να προβλέπουμε σωστά την έξοδο t_i για οποιαδήποτε δεδομένα εισόδου x_i ακόμα και αν δεν τα έχουμε ξαναδεί κατά την εκπαίδευση” [7]. Από την άλλη όμως οι ταξινομητές μπορεί να αντιμετωπίσουν διάφορα προβλήματα, τα οποία μπορούν να επηρεάσουν την απόδοσή τους.

Κάποια από τα κύρια προβλήματα περιλαμβάνουν:

- **Υπερπροσαρμογή (Overfitting):** Αυτό συμβαίνει όταν ένα μοντέλο μαθαίνει να "μνημονεύει" τα δεδομένα εκπαίδευσης αντί να γενικεύει τις πραγματικές τάσεις και τα μοτίβα των δεδομένων. Αυτό μπορεί να οδηγήσει σε κακή απόδοση σε νέα, απρόβλεπτα δεδομένα.
- **Υποπροσαρμογή (Underfitting):** Αυτό συμβαίνει όταν το μοντέλο δεν έχει μάθει αρκετά από τα δεδομένα εκπαίδευσης και δεν μπορεί να καταστήσει ακριβείς προβλέψεις για τα νέα δεδομένα.
- **Μη ισορροπημένα δεδομένα (Imbalanced Data):** Όταν οι κλάσεις του προβλήματος δεν έχουν ισορροπημένο αριθμό παραδειγμάτων, το μοντέλο μπορεί να επηρεαστεί και να έχει προκαταλήψεις προς τις πιο κοινές κλάσεις.
- **Ανεπαρκής ποσότητα δεδομένων (Insufficient Data):** Όταν τα δεδομένα είναι ανεπαρκή για την εκπαίδευση ενός αξιόπιστου μοντέλου, η απόδοση του μπορεί να είναι χαμηλή ή να μην είναι εφικτή η εκπαίδευσή του.
- **Εξωτερικές μεταβολές (Outliers):** Η παρουσία εκτός των προβλεπόμενων τιμών μπορεί να επηρεάσει την απόδοση του μοντέλου, ειδικά αν δεν αντιμετωπίζονται σωστά.

2.7.3 Αντιμετώπιση προβλημάτων

Για να αντιμετωπίζουμε ή ελαττώσουμε τα παραπάνω προβλήματα μπορούμε να χρησιμοποιήσουμε τις εξής μεθόδους:

- Συλλογή περισσότερων δεδομένων, η χρήση περισσότερων δεδομένων εκπαίδευσης μπορεί να βοηθήσει το μοντέλο να μάθει γενικότερα χαρακτηριστικά και να αποφύγει το overfitting
- Επεξεργασία των δεδομένων (Data Preprocessing): Αυτό μπορεί να περιλαμβάνει την αφαίρεση ή την αντικατάσταση τιμών που δεν έχουν καταγραφεί, την κωδικοποίηση των κατηγοριών των μεταβλητών, την κανονικοποίηση των δεδομένων ή τη μείωση της διάστασης.
- Επιλογή χαρακτηριστικών (Feature Selection): Αυτό μπορεί να περιλαμβάνει την επιλογή των σημαντικότερων χαρακτηριστικών για την εκπαίδευση του μοντέλου και την αποφυγή υπερπροσαρμογής.
- Απεικόνιση των δεδομένων (Data Visualization): Η απεικόνιση των δεδομένων μπορεί να βοηθήσει στην κατανόηση των σχέσεων μεταξύ των χαρακτηριστικών και την ανίχνευση ενδεχόμενων προβλημάτων ή ανωμαλιών.
- Εξερεύνηση διαφορετικών αλγορίθμων (Algorithm Exploration): Αυτό συμπεριλαμβάνει τη δοκιμή διαφορετικών αλγορίθμων μηχανικής μάθησης για το ίδιο πρόβλημα και την αξιολόγηση των αποτελεσμάτων τους.
- Κανονικοποίηση (Normalization): Η κανονικοποίηση των δεδομένων, όπως η κανονικοποίηση min-max ή η κανονικοποίηση Z-score, μπορεί να βοηθήσει στην αποφυγή της υπερεκπαίδευσης στο μοντέλο.
- Dropout: Η τεχνική dropout αποτρέπει το overfitting με το να απενεργοποιεί τυχαία νευρώνες κατά την εκπαίδευση. Αυτό επιτρέπει στο μοντέλο να μην εξαρτάται υπερβολικά από συγκεκριμένες πληροφορίες και να μάθει πιο γενικευμένα χαρακτηριστικά.
- Κανονικοποίηση παρτίδας (Batch Normalization): Η κανονικοποίηση παρτίδας είναι μια τεχνική που χρησιμοποιείται για να βοηθήσει στην εξομάλυνση των κατανομών εξόδου των επιπέδων ενός νευρωνικού δικτύου, βοηθώντας έτσι στη μείωση του overfitting.
- Περιορισμός της πολυπλοκότητας του μοντέλου: Μερικές φορές, το overfitting μπορεί να προκύψει λόγω της υπερβολικής πολυπλοκότητας του μοντέλου. Μειώνοντας τον αριθμό των επιπέδων ή των νευρώνων, μπορεί να βελτιωθεί η γενίκευση του μοντέλου.

2.8 Επίλογος

Καθ' όλη τη διάρκεια αυτού του κεφαλαίου, εξερευνήσαμε τον συναρπαστικό κόσμο της τεχνητής νοημοσύνης και είδαμε ότι η τεχνητή νοημοσύνη είναι ουσιαστικά η προσπάθεια για τη δημιουργία μηχανών που μπορούν να εκτελούν εργασίες που απαιτούν ανθρώπινη νοημοσύνη. Αναδείξαμε

επίσης τον υποτομέα της TN, την μηχανική μάθηση που αναφέρεται στην ικανότητα των υπολογιστών να μαθαίνουν από δεδομένα χωρίς να προγραμματίζονται εκ των προτέρων εκτενώς. Η μηχανική μάθηση περιλαμβάνει αλγόριθμους που επιτρέπουν στους υπολογιστές να βελτιώνουν την απόδοσή τους καθώς αποκτούν εμπειρία από τα δεδομένα. Τέλος εστίασαμε ιδιαίτερα στα νευρωνικά δίκτυα τα οποία αποτελούν μέρος των αλγορίθμων που χρησιμοποιούμε για την εκπαίδευση και τη λειτουργία των συστημάτων τεχνητής νοημοσύνης και μηχανικής μάθησης. Μέσω της ανάλυσης δεδομένων και της αναγνώρισης προτύπων, τα νευρωνικά δίκτυα έχουν επιτρέψει την ανάπτυξη προηγμένων εφαρμογών σε πολλούς τομείς, όπως η αναγνώριση εικόνων, η αυτόματη μετάφραση, η αυτόνομη οδήγηση κ.α

Κεφάλαιο 3ο: Πειραματική διαδικασία

3.1 Εισαγωγή

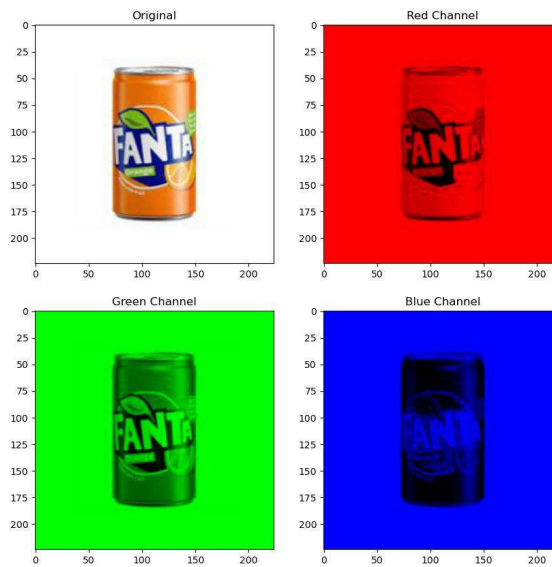
Σε αυτό το κεφάλαιο θα γίνει μια λεπτομερές επεξήγηση της πειραματικής διαδικασίας, αρχικά θα αναφερθούμε στη δομή της εικόνας, δηλαδή τι είναι αυτό που λαμβάνουν τα συστήματα και πως μπορούμε να επεξεργαστούμε κατάλληλα ώστε να κρατήσουμε τη πληροφορία που χρειαζόμαστε. Στη συνέχεια θα αναφερθούμε στον ήχο, τι είναι αυτό που λαμβάνουν τα συστήματα και την κατάλληλη επεξεργασία που χρειάζεται. Μετά θα αναφερθούμε στη γλώσσα προγραμματισμού την οποία χρησιμοποιήσαμε και τέλος θα δείξουμε μετρήσεις και αποτελέσματα από την πειραματική διαδικασία.

3.2 Η εικόνα

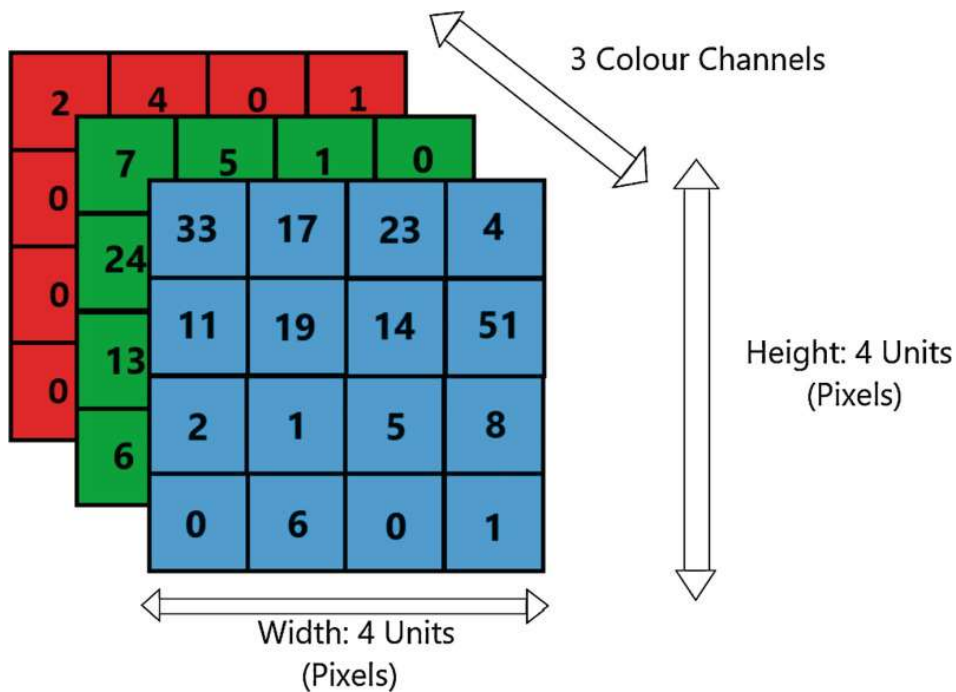
Η εικόνα που αντιλαμβανόμαστε είναι αποτέλεσμα της αντανάκλασης του φωτός από αντικείμενα που βρίσκονται στο πεδίο της όρασής μας. Ένα υπολογιστικό σύστημα δεν αντιλαμβάνεται μία εικόνα όπως ένας άνθρωπος. Οι εικόνες που βλέπει ο άνθρωπος μέσω του οφθαλμού, δηλαδή η πληροφορία που συλλέγει από τις ανακλάσεις του φωτός, είναι αναλογικής φύσεως, ενώ το υπολογιστικό σύστημα αντιλαμβάνεται ψηφιακή πληροφορία, καθώς όπως γνωρίζουμε το σύμπαν γύρο μας περικλείεται από αναλογικά σήματα.

Το υπολογιστικό σύστημα για να λάβει την εικόνα, ως αισθητήρα –μετατροπέα χρησιμοποιεί την κάμερα, η οποία μετατρέπει το αναλογικό σήμα από της ανακλάσεις του φωτός σε ψηφιακό σήμα, στην ουσία αυτό που αντιλαμβάνεται είναι μια αλληλουχία από bits ως μια “σειρά” από αριθμούς 0 και 1 τα λεγόμενα εικονοστοιχεία τα οποία δίνουν την πληροφορία της φωτεινότητας, το πλάτος και το μήκος της εικόνας καθώς και τα κανάλια από τα οποία αποτελείται. Τα κανάλια σε μία έγχρωμη εικόνα είναι 3, το κόκκινο κανάλι, το πράσινο κανάλι και το μπλε κανάλι «Σχήμα 3.1». Για παράδειγμα σε μία έγχρωμη φωτογραφία με 224x220 έχουμε 224 εικονοστοιχεία στο πλάτος και 220 εικονοστοιχεία στο ύψος για το κάθε κανάλι, οπότε στο κάθε κανάλι έχουμε $224 \times 220 = 49280$ εικονοστοιχεία.

Αρα αυτό που δέχεται το υπολογιστικό σύστημα είναι ένας πίνακας τριών διαστάσεων, ο πίνακας αυτός περιέχει τις τιμές των εικονοστοιχείων για κάθε σημείο της εικόνας (στη συγκεκριμένη 49280 σημεία) και στα τρία κανάλια. Με άλλα λόγια, κάθε στοιχείο του πίνακα περιέχει μια τριάδα τιμών που αντιστοιχούν στις τιμές των χρωματικών καναλιών (R, G, B) για το συγκεκριμένο εικονοστοιχείο. Κάθε τιμή είναι ένας αριθμός από 0 έως 255, που αντιπροσωπεύει την φωτεινότητα του σχετικού χρώματος σε αυτό το συγκεκριμένο εικονοστοιχείο «Σχήμα 3.2».



Σχήμα 3.1 Εμφάνιση των τριών καναλιών.



Σχήμα 3.2 Πίνακας (εικόνας) 4x4x3

[πηγή: https://www.researchgate.net/figure/A-3D-tensor-of-a-Red-Green-Blue-RGB-image-of-a-dimension-of-4-A-4-A-3_fig2_357820593]

3.3 Επεξεργασία στις εικόνες

Για να βοηθήσουμε τον ταξινομητή να διακρίνει τα απαιτούμενα χαρακτηριστικά από τα δεδομένα (εικόνες) τα οποία του εισάγουμε πρέπει να επεξεργαστούμε κατάλληλα τα δεδομένα αυτά. Στην

εργασία αυτή στις εικόνες (με τη χρήση τη γλώσσας python) έγιναν οι παρακάτω επεξεργασίες «Σχήμα 3.3»:

1. Αλλαγή μεγέθους των εικόνων. Οι εικόνες μετατράπηκαν σε 224x224 χωρίς να αλλοιωθούν. Η επεξεργασία αυτή έγινε διότι οι ταξινομητές που θα χρησιμοποιήθηκαν, θέλουν τα δεδομένα τους να έχουν τις ίδιες διαστάσεις. Στις εικόνες που είχαν μεγαλύτερο μέγεθος έγινε ουσιαστικά σμίκρυνση δηλαδή μια εικόνα με μέγεθος 500 x 400 έγινε 200 x 100 και στη συνέχεια τα σημεία που υπολείπονταν για να γίνει 224x224 γέμισαν με λευκό (τιμή 255 και στα τρία κανάλια), αντίστοιχα για τις εικόνες που είχαν μικρότερες διαστάσεις τα σημεία που υπολείπονταν για να γίνει 224x224 γέμισαν με επίσης με λευκό.
2. Αφαίρεση φόντου(Background). Έγινε αντικατάσταση του φόντου σε λευκό στις εικόνες για να μείνει στην εικόνα μόνο η χρήσιμη πληροφορία, δηλαδή η φιάλη το ποτήρι κ.τ.λ.
3. Εύρεση ακμών με τη μέθοδο Canny(Canny edge detector) στην εικόνα. Είναι η πιο διαδεδομένη μέθοδος εύρεσης ακμών σε εικόνα λόγω τις ακρίβειας που προσφέρει. Με αυτό το τρόπο δίνονται οι ακμές και μόνο σαν είσοδο στον ταξινομητή για να μπορέσει πιο εύκολα να διακρίνει τα χαρακτηριστικά τις κάθε εικόνας.



Σχήμα 3.3 Στάδια επεξεργασίας

[Πηγή: <https://3dwealthadvisors.com/blog/why-are-soda-cans-different-in-hawaii/>]

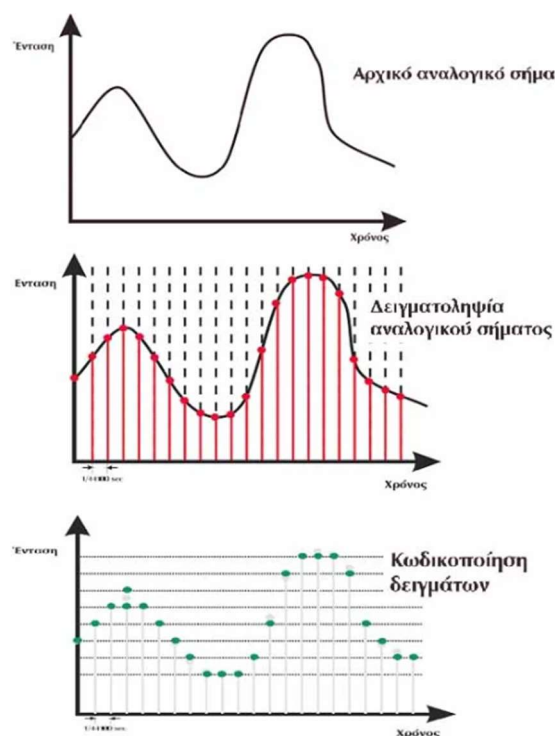
3.4 Ο ήχος

Ο ήχος είναι το ερέθισμα που διεγείρει το αισθητήριο τις ακοής το οποίο βρίσκεται στα αυτιά μας. Οι ήχοι που ακούμε είναι προϊόν παλλόμενων σωμάτων τα οποία με τη σειρά τους θέτουν σε κίνηση τα μόρια του αέρα με τα οποία έρχονται σε επαφή. Έτσι όπως αντιλαμβανόμαστε οι ήχοι στο περιβάλλον είναι αναλογικά σήματα.

Για να λάβει το υπολογιστικό σύστημα τον ήχο χρησιμοποιεί ένα μικρόφωνο το οποίο κάνει τη μετατροπή του ήχου από τις διαφορές στην ατμοσφαιρική πίεση σε ηλεκτρικό αναλογικό σήμα και στη συνέχεια μέσω της κάρτας ήχου το μετατρέπει σε ψηφιακό σήμα που αντιπροσωπεύεται από μια ακολουθία από δυαδικά bits. Για να γίνει η μετατροπή του σήματος από αναλογικό σε ψηφιακό ακολουθείτε η εξής διαδικασία «Σχήμα 3.4»:

- Δειγματοληψία (Sampling): Κατά τη δειγματοληψία, το αναλογικό σήμα ήχου που λαμβάνεται από το μικρόφωνο, που αντιπροσωπεύει τις διαφορές στην ατμοσφαιρική πίεση, κατά διαστήματα χρόνου και μετατρέπεται σε μια ακολουθία από δείγματα.
- Κβαντισμός (Quantization): Κάθε δείγμα αναλογικού ήχου κβαντίζεται, δηλαδή αντιστοιχίζεται σε έναν αριθμό που αντιπροσωπεύει την τιμή του ήχου σε αυτήν τη συγκεκριμένη χρονική στιγμή. Αυτός ο αριθμός αποθηκεύεται ψηφιακά.
- Κωδικοποίηση (Encoding): Τα κβαντισμένα δείγματα κωδικοποιούνται σε δυαδική μορφή, δηλαδή σε μια σειρά από bits, με τη χρήση της κάρτας ήχου.

Με αυτόν τον τρόπο, ο αναλογικός ήχος μετατρέπεται σε μια ακολουθία από δυαδικά bits που αντιπροσωπεύουν τον ψηφιακό ήχο, ο οποίος μπορεί να αναγνωριστεί και να επεξεργαστεί από τον υπολογιστή.

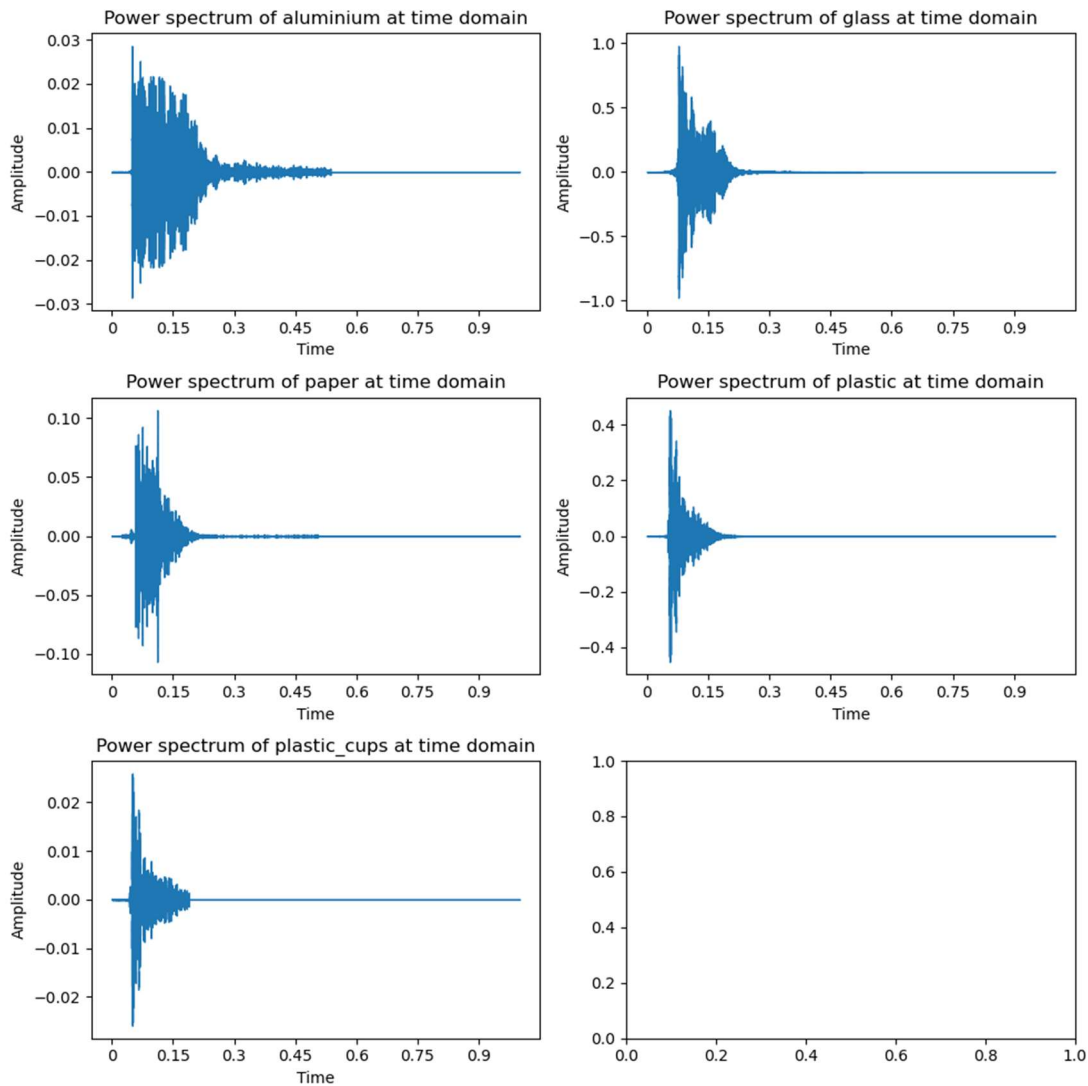


Σχήμα 3.4 Αναλογικό σε ψηφιακό

[πηγή: <https://rethnet.wordpress.com/2009/01/02/sound/>]

3.5 Επεξεργασία ήχου

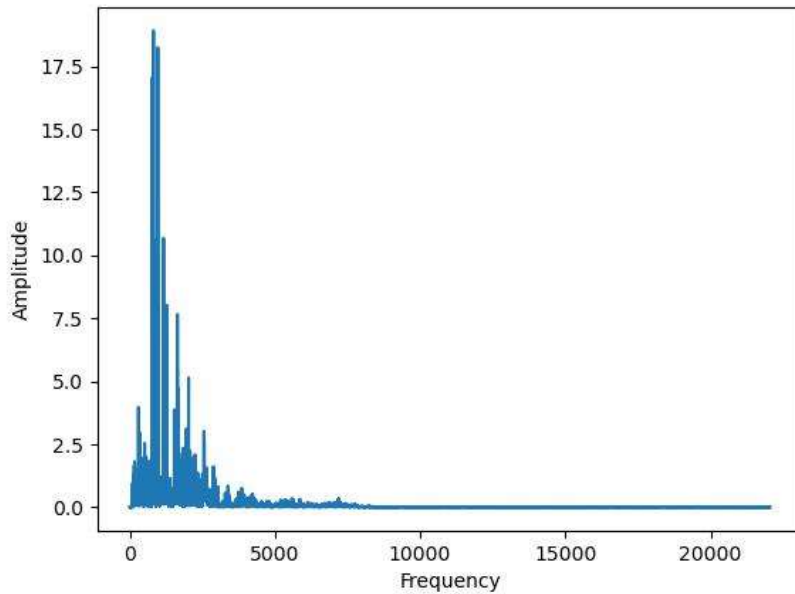
Το ψηφιακό σήμα που καταγράφεται από το υπολογιστικό σύστημα μέσω του μικροφώνου και της κάρτας ήχου αντιπροσωπεύει την αναλογική κυματομορφή του ήχου σε ψηφιακή μορφή. Τα bits σε αυτό το σήμα δείχνουν το πλάτος του ήχου, δηλαδή την ένταση του ήχου ανά μονάδα χρόνου«Σχήμα 3.5».



Σχήμα 3.5 Διαγράμματα ήχων στο πεδίο του χρόνου

Αυτή η πληροφορία αναπαριστά τη ένταση του ήχου για κάθε χρονικής στιγμής, επιτρέποντας έτσι στον υπολογιστή να καταγράψει και να αναπαράγει τον ήχο με ψηφιακό τρόπο. Όμως όπως καταλαβαίνουμε αυτή η πληροφορία δεν μπορεί να μας χαρακτηρίσει από τι προέρχεται ο ήχος αυτός, δηλαδή το τι είναι αυτό που το παράγει. Ο χαρακτηρισμός του κάθε ήχου προέρχεται από τη συχνότητα του.

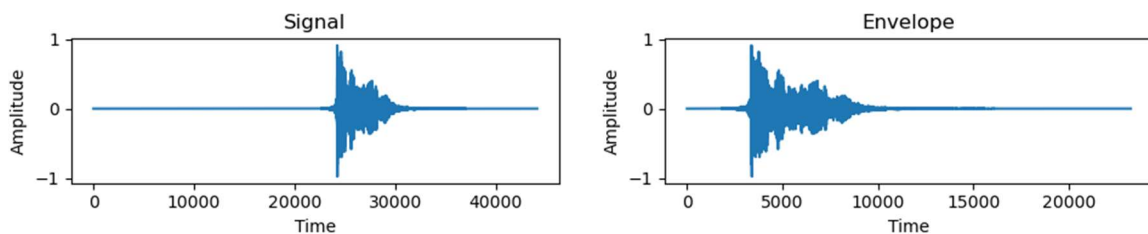
Για να βρεθούν οι συχνότητες χρειάζεται να γίνει μετατροπή του ήχου από το πεδίο του χρόνου στο πεδίο της συχνότητας. Η μετατροπή ενός σήματος από το πεδίο του χρόνου στο πεδίο της συχνότητας γίνεται με τη χρήση ενός μαθηματικού μετασχηματισμού, γνωστό ως μετασχηματισμός Fourier. Αυτός ο μετασχηματισμός λαμβάνει ένα σήμα στο πεδίο του χρόνου και το μετατρέπει σε ένα σήμα στο πεδίο της συχνότητας, αποκαλύπτοντας τις συχνότητες που απαρτίζουν το αρχικό σήμα. Ο μετασχηματισμός Fourier επιτρέπει να δούμε ποιες συχνότητες είναι παρούσες στο αρχικό σήμα και με ποια ένταση, δηλαδή μας φανερώνει το φάσμα του σήματος. Αυτό είναι χρήσιμο για να κατανοήσουμε τη σύνθεση του σήματος και να ανιχνεύσουμε πιθανές μη χρήσιμες συχνότητες ή θορύβους «Σχήμα 3.6».



Σχήμα 3.6 Διάγραμμα ήχου μετά από μετασχηματισμό Fourier

Στα πλαίσια της εργασίας και έπειτα από έρευνα για την εξαγωγή των χαρακτηριστικών του ήχου χρησιμοποιήθηκε η διαδικασία του Mel - Frequency Cepstral Coefficients η οποία είναι μια ευρέως χρησιμοποιημένη διαδικασία για την εμφάνιση των χαρακτηριστικών του ήχου τα βήματα που ακολουθούνται για αυτή τη διαδικασία είναι τα εξής:

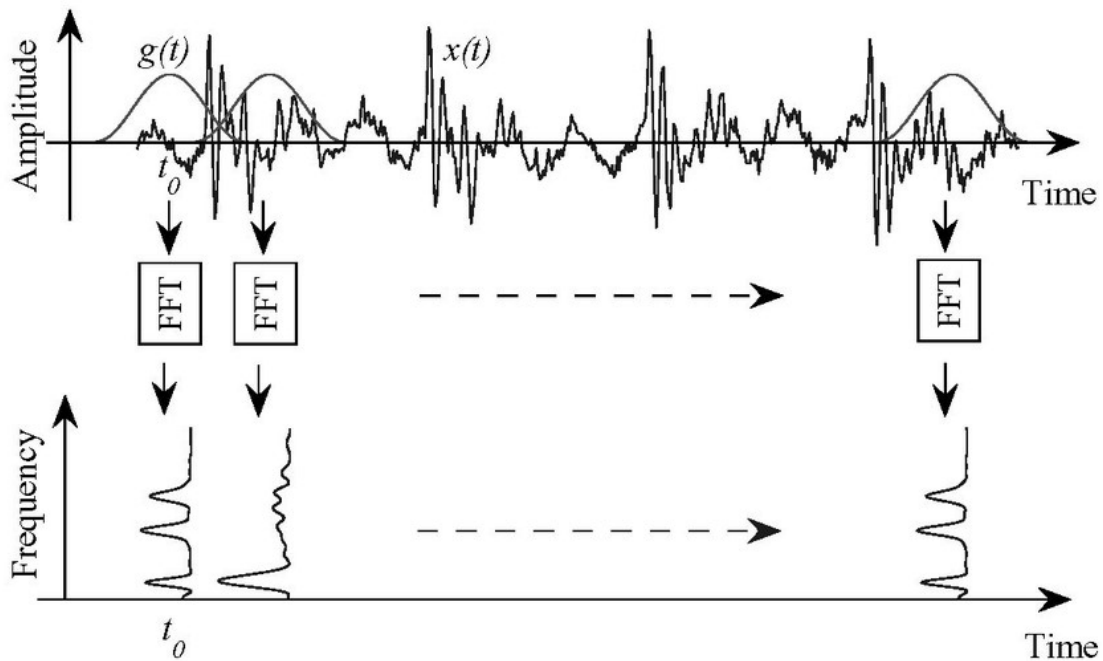
1. Φιλτράρισμα - κανονικοποίηση των δεδομένων του ήχου: Για το φιλτράρισμα χρησιμοποιήθηκε η τεχνική όπου μια τιμή συγκρίνεται με το κινητό μέσο του σήματος και έτσι ουσιαστικά ότι είναι μικρότερο από τη τιμή που συγκρίνουμε, μηδενίζεται «Σχήμα 3.7».



Σχήμα 3.7 Σημά ήχου πριν και μετά το φιλτράρισμα

2. Short-Time Fourier Transform (STFT): Στη συνέχεια χρησιμοποιούμε το μετασχηματισμό Fourier σε μικρά χρονικά τμήματα του σήματος γι' αυτό ονομάζεται και STFT. Με αυτό το τρόπο σε ένα ηχητικό σήμα που δεν έχει σταθερή συχνότητα π.χ. ομιλία εξάγουμε τις συχνότητες του σήματος στη διάρκεια διαφορετικών χρονικών διαστημάτων. Κάθε μικρό τμήμα ονομάζεται παράθυρο, και εφαρμόζουμε το μετασχηματισμό Fourier σε κάθε παράθυρο ξεχωριστά. Στη συνέχεια, για κάθε παράθυρο, έχουμε ένα φάσμα συχνοτήτων που αντιπροσωπεύει την ενέργεια σε κάθε συχνότητα για αυτό το χρονικό διάστημα. Το αποτέλεσμα του STFT είναι ένας διδιάστατος πίνακας, όπου οι δύο διαστάσεις αντιστοιχούν στο χρόνο και στη συχνότητα. Συνολικά, ο STFT μας επιτρέπει να παρακολουθήσουμε πώς η

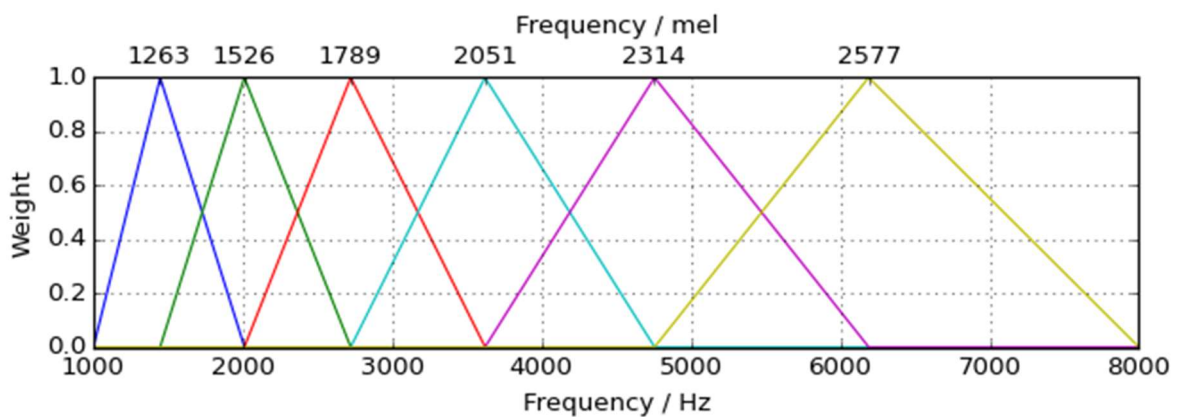
συχνότητα εξελίσσεται κατά μήκος του χρόνου στο σήμα, καθιστώντας την χρήσιμη για εφαρμογές όπως η ανάλυση ήχου και η επεξεργασία σήματος.



Σχήμα 3.8 Μετασχηματισμός STFT

[Πηγή: https://www.researchgate.net/figure/illustration-of-short-time-Fourier-transform-on-the-test-signal-xt_fig3_220650705]

3. Mel-Filterbank: Η εφαρμογή του Mel filterbank γίνεται σε κάθε μικρό τμήμα STFT και επιτρέπει τη μετατροπή του σε ένα πιο αναλυτικό πεδίο συχνοτήτων, γνωστό ως Mel κλίμακα «Σχήμα 3.9». Το Mel filterbank αποτελείται από μια σειρά φίλτρων, κάθε ένα από τα οποία καλύπτει ένα διαφορετικό εύρος συχνοτήτων στη Mel κλίμακα. Αυτή η διαδικασία επιτρέπει τη βελτιστοποίηση της αναπαράστασης της φωνητικής πληροφορίας, καθώς η Mel κλίμακα αντικατοπτρίζει καλύτερα τον τρόπο με τον οποίο αντιλαμβάνεται το ανθρώπινο αυτί τις συχνότητες.

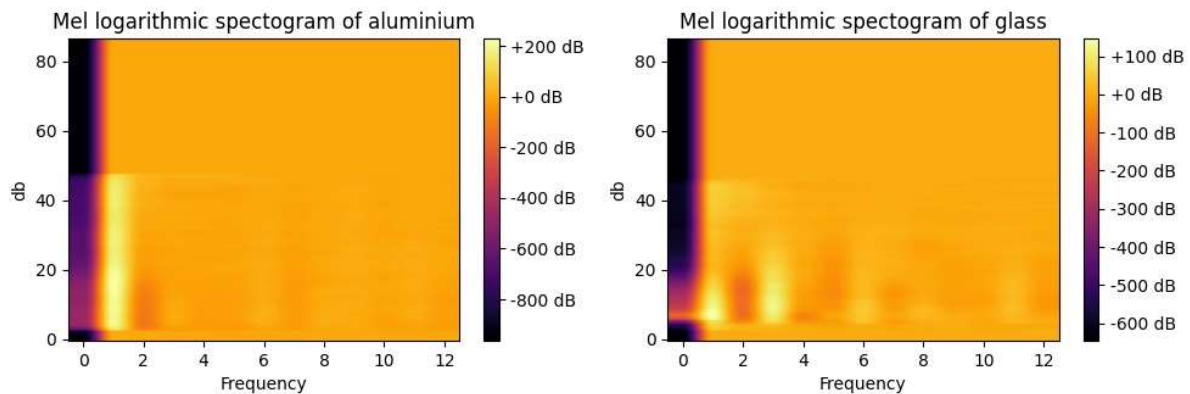


Σχήμα 3.9 Απεικόνιση του Mel-filterbank

[Πηγή: <https://siggigue.github.io/pyfilterbank/melbank.html>]

4. Υπολογισμός λογαριθμικής ενέργειας: Αφού το φάσμα περάσει από τα Mel φίλτρα, υπολογίζεται ο λογάριθμος της ενέργειας για κάθε φίλτρο.

5. Διακριτοποίηση του αποτελέσματος με τη χρήση DCT (Discrete Cosine Transform): εφαρμόζεται στα λογαριθμικά ενεργειακά σήματα που προκύπτουν από το Mel spectrogram. Αυτός ο μετασχηματισμός μετατρέπει τα δεδομένα από το πεδίο της συχνότητας στο πεδίο των συντελεστών της DCT. Οι συντελεστές που προκύπτουν από τη DCT είναι τα MFCC (Mel-Frequency Cepstral Coefficients). Τα MFCC είναι μια σειρά αριθμητικών συντελεστών που αντιπροσωπεύουν τα χαρακτηριστικά του ήχου με βάση τη mel κλίμακα συχνοτήτων και την κεφαλαϊώδη ανάλυση «». Αυτοί οι συντελεστές παρέχουν μια συμπαγή περιγραφή του ηχητικού σήματος που είναι ανθεκτική σε μικρές διαφορές, στην παραμόρφωση του ήχου και συχνά χρησιμοποιούνται σε εφαρμογές επεξεργασίας ήχου και αναγνώρισης προτύπων.



Σχήμα 3.10 MFCC από αλουμίνιο και γυαλί

3.6 Η γλώσσα προγραμματισμού Python

Η τεχνητή νοημοσύνη έχει βιώσει μια επαναστατική ανάπτυξη τα τελευταία χρόνια, με την Python να διαδραματίζει καθοριστικό ρόλο σε αυτή την πορεία. Πριν τη python οι κύριες γλώσσες προγραμματισμού για T.N ήταν αρχικά η LISP και στη συνέχεια η Prolog. Η Python είναι μια γλώσσα υψηλού επιπέδου χρησιμοποιείται ευρέως σε διάφορους σκοπούς οπότε λέμε ότι είναι και γενικού σκοπού.

Είναι πολύ διαδεδομένη καθώς έχει μεγάλη κοινότητα σε πολλά πεδία και ένα από αυτά φυσικά είναι και η T.N. Είναι εύκολη στην εκμάθηση διότι στηρίζει τις εντολές της στην Αγγλική γλώσσα αλλά είναι και πολύ απλή στη σύνταξη της. Λόγω της ευκολίας της, δεν χρειάζεται να έχεις είδη γνώση προγραμματισμού.

Είναι ευέλικτη γλώσσα προγραμματισμού καθώς υποστηρίζει πολλά στυλ δομής προγράμματος, όπως δομημένο προγραμματισμό, αντικειμενοστραφή προγραμματισμό, διαδικαστικό προγραμματισμό, τμηματοποιημένο προγραμματισμό. Λόγω της ενασχόλησης πολλών προγραμματιστών με τη συγκεκριμένη γλώσσα έχει πληθώρα από έτοιμες βιβλιοθήκες και πλατφόρμες. Έτσι οι ερευνητές και οι επαγγελματίες μπορούν να δημιουργήσουν γρήγορα και εύκολα πρωτότυπες ιδέες. Στον τομέα της T.N έχει βιβλιοθήκες η οποίες παρέχουν στο χρήστη έτοιμα μοντέλα διαφόρων αρχιτεκτονικών για αναγνώριση προτύπων, επεξεργασία φυσικής γλώσσας και πολλά άλλα.

3.7 Βιβλιοθήκες της Python που χρησιμοποιήθηκαν

Για την λήψη και την επεξεργασία των εικόνων και των ήχων καθώς και για την ανάπτυξη των ταξινομητών που δοκιμάστηκαν, χρησιμοποιήθηκαν κάποιες έτοιμες βιβλιοθήκες της γλώσσας python που αναφέρονται παρακάτω.

Ας δούμε κάποιες βασικές πληροφορίες για κάθε βιβλιοθήκη:

- PyAudio: Η πρώτη έκδοση δημιουργήθηκε από τον Hubert Pham το 2001. Η βιβλιοθήκη χρησιμοποιήθηκε για την αναπαραγωγή και εγγραφή ήχου σε Python, προσφέροντας δυνατότητες όπως αναπαραγωγή, εγγραφή, αλλαγή της έντασης, κ.λπ.
- Librosa: Δημιουργοί της βιβλιοθήκης είναι Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis και τη δημιούργησαν το 2011. Η χρήση της είναι για την ανάλυση και επεξεργασία ήχου στη Python, ειδικά για μουσική σήμανση, ανάλυση φάσματος, υπολογισμό του Mel-frequency cepstral coefficients (MFCCs) κ.ά.
- TensorFlow: Δημιουργοί είναι Ερευνητές της Google Brain Team. Πρώτη έκδοση ήταν το 2015. Είναι ένα από τα πιο δημοφιλή εργαλεία για μηχανική μάθηση και βαθιά μάθηση. Παρέχει ένα ευρύ φάσμα αλγορίθμων μηχανικής μάθησης και είναι ευέλικτο για χρήση σε διάφορα περιβάλλοντα. Η εργασία είναι δομημένη επάνω σε αυτή τη βιβλιοθήκη.
- OpenCV: Δημιουργήθηκε από την ομάδα ανάπτυξης του OpenCV.το 2000. Η OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη που παρέχει εργαλεία για επεξεργασία εικόνας και βίντεο, αναγνώριση προσώπων, εντοπισμό αντικειμένων, χαρακτηριστικά μορφολογίας και άλλα.
- PyTorch: Δημιουργήθηκε από ομάδα Ερευνητών του Facebook το 2016. Η PyTorch παρέχει εργαλεία για μηχανική μάθηση και βαθιά μάθηση, είναι ευέλικτο και ευανάγνωστο και χρησιμοποιείται ευρέως για την ανάπτυξη μοντέλων νευρωνικών δικτύων.
- Keras: Δημιουργήθηκε από τον François Chollet το 2015.Είναι μια ανοιχτού κώδικα βιβλιοθήκη που παρέχει ένα υψηλού επιπέδου API για το χτίσιμο και την εκπαίδευση μοντέλων νευρωνικών δικτύων. Πλέον έχει ενσωματωθεί στην Tensorflow ως επίσημο υψηλού επιπέδου API.
- Matplotlib.pyplot: Δημιουργήθηκε από τον John D. Hunter το 2003.Παρέχει εργαλεία γραφικής απεικόνισης δεδομένων και δημιουργίας διαγραμμάτων στη Python.
- NumPy: Δημιουργός της NumPy είναι ο Travis Oliphant, την δημιούργησε το 2006. Είναι μια βιβλιοθήκη για τη γλώσσα Python που επικεντρώνεται στους επιστημονικούς υπολογισμούς και επεξεργασία πινάκων. Παρέχει υψηλές αποδόσεις και ευέλικτες λειτουργίες για την εργασία με πίνακες δεδομένων.
- Pandas: Τη βιβλιοθήκη Pandas τη δημιούργησε ο Wes McKinney το 2008. Με αυτή τη βιβλιοθήκη κάνουμε διαχείριση και ανάλυση δεδομένων, παρέχοντας δομές δεδομένων και λειτουργίες για την εύκολη εκτέλεση πολύπλοκων λειτουργιών ανάλυσης δεδομένων.
- Sklearn: Τη δημιούργησε ομάδα ερευνητών του INRIA. Πρόκειται για μία από τις πιο δημοφιλείς βιβλιοθήκες μηχανικής μάθησης στην Python, παρέχοντας εργαλεία για την ταξινόμηση, την παλινδρόμηση, τη συσταδοποίηση και άλλες μεθόδους μηχανικής μάθησης.

3.8 Μετρήσεις και αποτελέσματα

Όπως θα δούμε παρακάτω, αρχικά οι μετρήσεις γίνανε ξεχωριστά για τα μοντέλα που είχαν ως δεδομένα εικόνες και ξεχωριστά για τα μοντέλα που είχαν ως δεδομένα ήχο, αυτό είχε σκοπό την ευρέθη του ποιοτικότερου μοντέλου για τις εικόνες και του ποιοτικότερου μοντέλου για τον ήχο . Στη συνέχεια επιλέχθηκε το καλύτερο μοντέλο που είχε ως δεδομένα εικόνες και το καλύτερο που είχε ως δεδομένα ήχο και χρησιμοποιήσαμε μία μέθοδο που χρησιμοποιεί τις προβλέψεις αυτών των δύο για την εξαγωγή πιο αξιόπιστου αποτελέσματος.

3.8.1 Υλικό που χρησιμοποιήθηκε

Για να καθιστή δυνατή η εκπαίδευση των μοντέλων σε πολλές εικόνες χρειάστηκε υπολογιστής με πολλούς πόρους. Ο προσωπικός υπολογιστής που αρχικά χρησιμοποιούνταν δεν κάλυπτε της απαιτήσεις που έχουν οι εφαρμογές αυτές. Οπότε έγινε αναζήτηση ώστε να βρεθεί ένας υπερυπολογιστής για της που να καλύπτει τις απαιτήσεις της εφαρμογής για την εκπαίδευση των

μοντέλων σε εικόνες. Η λύση στο πρόβλημα ήταν η μίσθωση μίας ειδικής πλατφόρμας στημένη σε εικονική μηχανή από την Google, η πλατφόρμα αυτή φέρει το όνομα Google Colab το οποίο με χρήση μικρών υπολογιστικών πόρων παρέχεται εντελώς δωρεάν ενώ για περισσότερους πόρους και καλύτερες κάρτες γραφικών μισθώνεται το μήνα. Η εικονική μηχανή του Google Colab που χρησιμοποιήθηκε είναι η Google Colab V100 η οποία ως μονάδα επεξεργασίας χρησιμοποιεί την κάρτα γραφικών της NVIDIA Tesla V100. Η κάρτα γραφικών Tesla V100 είναι μια από τις πιο ισχυρές GPU της NVIDIA και χρησιμοποιείται ευρέως σε εφαρμογές Deep Learning και Machine Learning λόγω της υψηλής της απόδοσης.

Η NVIDIA Tesla V100 χρησιμοποιεί την αρχιτεκτονική Volta και διαθέτει 640 Tensor Cores και 5.120 CUDA Cores με αυτή την αρχιτεκτονική γίνεται παράλληλες πράξεις σε πίνακες, παρέχοντας υψηλή απόδοση επιταχύνοντας την εκπαίδευση σε εφαρμογές Machine Learning και Deep Learning. Η NVIDIA Tesla V100 διαθέτει μνήμη HBM2 (High Bandwidth Memory) με χωρητικότητα έως και 32GB, που προσφέρει υψηλή ταχύτητα μεταφοράς δεδομένων και μείωση του bottleneck στις εφαρμογές που απαιτούν μεγάλο όγκο δεδομένων.

3.8.2 Μοντέλα και αποτελέσματα για εικόνες

Τα δεδομένα, για τα μοντέλα που εκπαιδεύτηκαν και κάνανε πρόβλεψη σε εικόνες, είναι χωρισμένα στις εξής κλάσεις Aluminium για τα αλουμινένια τενεκεδάκια, Glasss για τις γυάλινες φιάλες, Paper για τα χάρτινα ποτήρια, Plastic για πλαστικά μπουκάλια και Plastic Cups για πλαστικά ποτήρια. Οι εικόνες είναι συνολικά 5000, οι 1500 χρησιμοποιήθηκαν για επικύρωση και οι υπόλοιπες 3500 χρησιμοποιήθηκαν για εκπαίδευση.

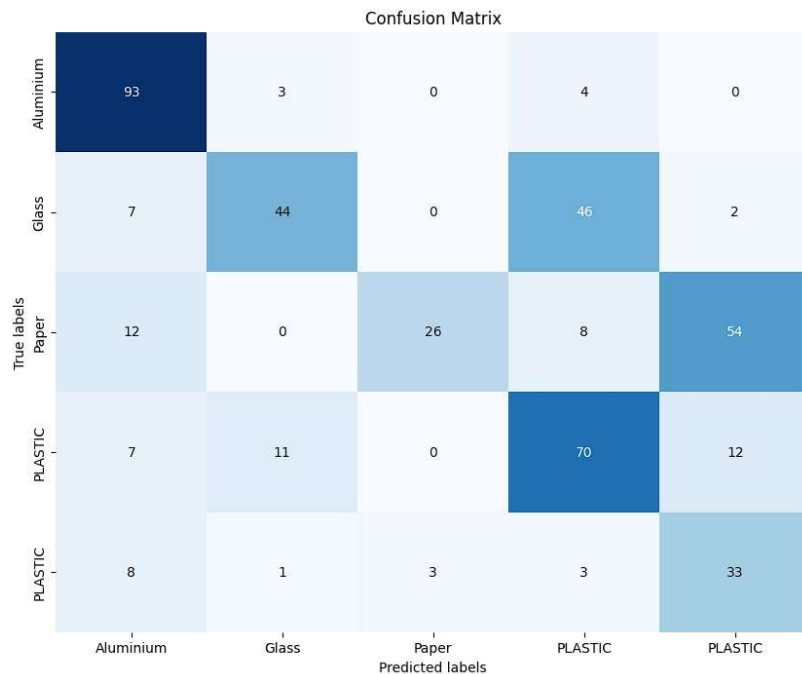
- Μοντέλο CNN1: Κατά τη διάρκεια των πρώτων δοκιμών τα μοντέλα που σχεδιάστηκαν δεν περνούσαν το 0,3 στο F1-Score. Στη συνέχεια κατορθώθηκε το πρώτο αρκετά καλό μοντέλο το οποίο έφτανε το 0,56 στο F1-Score αυτό είχε σαν αποτέλεσμα «Σχήμα 3.11» το οποίο αποτελούνταν από 1 επίπεδο εισόδου, 3 κρυφά επίπεδα και 3 πλήρως συνδεδεμένα επίπεδα «Πίνακας 3-1». Τα μοντέλα συνήθως λέμε ότι είναι αποδεκτά έχουν F1-Score 0,7 και πάνω και εξαρτάτε και από την εφαρμογή.

Πίνακας 3-1 Πρώτο μοντέλο CNN1 που σχεδιαστικέ για ταξινόμηση των εικόνων

Layer (Type)	Output Shape	Parameters
conv2d (Conv2D)	(224, 224, 16)	208
max_pooling2d (MaxPooling2D)	(112, 112, 16)	0
conv2d_1 (Conv2D)	(112, 112, 32)	2008
max_pooling2d_1 (MaxPooling2D)	(56, 56, 32)	0
conv2d_2 (Conv2D)	(56, 56, 64)	8256
max_pooling2d_2 (MaxPooling2D)	(28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	32896
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
dropout (Dropout)	(None, 14, 14, 128)	0

flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None,1500)	37633500
dense1 (Dense)	(None, 150)	225150
dropout1 (Dropout)	(None, 150)	0
dense2 (Dense)	(None, 5)	755
Total parameters		37902773
Trainable parameters		37902773
Non-trainable parameters		0

F1-score on validation set: 0.5656919373233966



Σχήμα 3.11 Confusion Matrix από το πρώτο μοντέλο

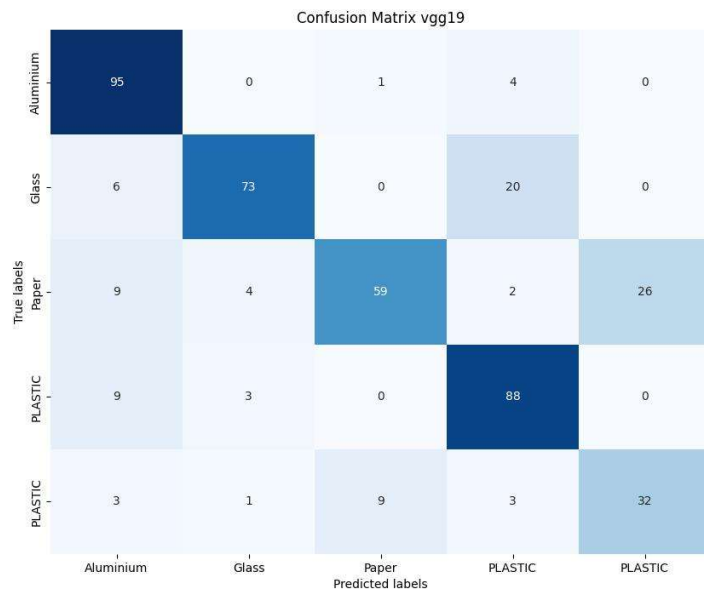
Στη συνέχεια τα υπόλοιπα 2 CNN που θα παρουσιαστούν θα είναι προ-εκπαιδευμένα μοντέλα με υψηλό F1 score. Σε αυτά τα μοντέλα για να τα εκπαιδύσουμε χρησιμοποιήσουμε την τεχνική της Μεταφοράς μάθησης (transfer learning). Η μεταφορά μάθησης (transfer learning) είναι μια τεχνική μηχανικής μάθησης όπου τα μοντέλα που έχουν εκπαιδευτεί σε ένα πρόβλημα χρησιμοποιούνται σε ένα διαφορετικό, συνήθως συναφές πρόβλημα. Αυτό γίνεται είτε με τη χρήση των προ εκπαιδευμένων βαρών από ένα ήδη εκπαιδευμένο μοντέλο, είτε με την προσαρμογή του μοντέλου σε ένα μικρότερο σύνολο δεδομένων για την επίλυση του νέου προβλήματος εδώ χρησιμοποιείται η προσαρμογή του μοντέλου σε μικρότερο σύνολο δεδομένων.

- Μοντέλο CNN Vgg19: Το μοντέλο VGG19 δημιουργήθηκε από τους Karen Simonyan και Andrew Zisserman. Το μοντέλο παρουσιάστηκε και συμμετείχε στο διαγωνισμό ImageNet Large Scale Visual Recognition Challenge (ILSVRC) το 2014. Αποτελείται από ένα επίπεδο εισόδου, 16 κρυφά επίπεδα και 3 πλήρως συνδεδεμένα επίπεδα. Κατά τη χρήση του στη δοκιμή δεν εμφανίζεται η δομή του, βλέπε «Πίνακας 3-2», αλλά οι δημιουργοί την έχουν δημοσιεύσει. Όπως θα δούμε προστέθηκαν λίγα επίπεδα για προσαρμογή του μοντέλου. Το αποτέλεσμα είναι να πετύχει ένα ικανοποιητικό F1-Score 0,76 και το αποτέλεσμα στον πίνακα σύγκρισης παρουσιάζει βελτίωση «Σχήμα 3.12 Vgg19 Confusion Matrix». Ένα δείγμα από 25 εικόνες στις οποίες έγινε πρόβλεψη βλέπουμε «Σχήμα 3.13»

Πίνακας 3-2 Δομή του μοντέλου CNN Vgg19

Layer (Type)	Output Shape	Parameters
Vgg19	(7, 7, 512)	20024383
conv2d (Conv2D)	(7, 7, 1024)	4719616
max_pooling2d_1 (MaxPooling2D)	(3, 3, 1024)	0
Dropout (Dropout)	(3, 3, 1024)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 150)	1382550
Dropout1 (Dropout)	(None, 150)	0
Dense1 (Dense)	(None, 5)	755
Total parameters		26127305
Trainable parameters		6102921
Non-trainable parameters		20024383

F1-score on validation set: 0.7560050623555469



Σχήμα 3.12 Vgg19 Confusion Matrix



Σχήμα 3.13 Προβλέψεις Vgg19

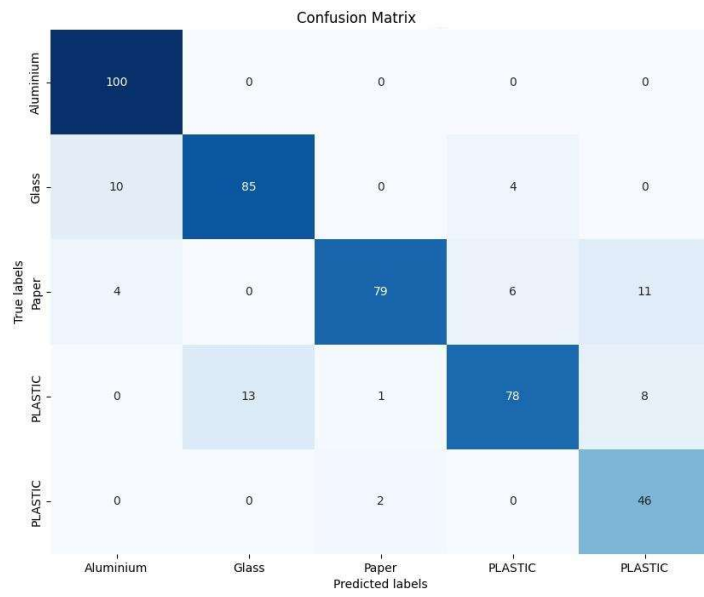
- Μοντέλο InceptionResNet. Το Inception-ResNet είναι ένα μοντέλο νευρωνικών δικτύων που συνδυάζει στοιχεία από τα μοντέλα Inception και ResNet. Η αρχιτεκτονική του σχεδιάστηκε από τους Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe και Jonathon Shlens της Google Research. Το μοντέλο σχεδιάστηκε για να βελτιώσει την απόδοση των νευρωνικών δικτύων σε διάφορες εργασίες μηχανικής μάθησης, όπως η αναγνώριση αντικειμένων και η ταξινόμηση εικόνων. Κατά τη χρήση του στη δοκιμή δεν εμφανίζεται η δομή του, βλέπε «Πίνακας 3-3», αλλά οι δημιουργοί την έχουν δημοσιεύσει. Όπως θα δούμε προστέθηκαν λίγα επίπεδα για προσαρμογή του μοντέλου. Το αποτέλεσμα είναι να πετύχει ένα ικανοποιητικό F1-Score 0,86 και το αποτέλεσμα στον πίνακα σύγκρισης παρουσιάζει βελτίωση «Σχήμα 3.14».

Πίνακας 3-3 Δομή του μοντέλου InceptionResNet

Layer (Type)	Output Shape	Parameters
Inception_resnet_v2(Functional)	(5, 5, 1536)	54336736
conv2d (Conv2D)	(7, 7, 1024)	14156800
max_pooling2d_1 (MaxPooling2D)	(3, 3, 1024)	0
Dropout (Dropout)	(3, 3, 1024)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 150)	614550
Dropout1 (Dropout)	(None, 150)	0
Dense1 (Dense)	(None, 5)	755
Total parameters		69108841
Trainable parameters		14772105
Non-trainable parameters		54336736

F1-score on validation set: 0.861920399179741

Ένα δείγμα προβλέψεων σε 25 εικόνες, από τις εικόνες που πρόβλεψε παρουσιάζεται στο «Σχήμα 3.15»



Σχήμα 3.14 Πίνακας σύγκρισης InceptionResNet



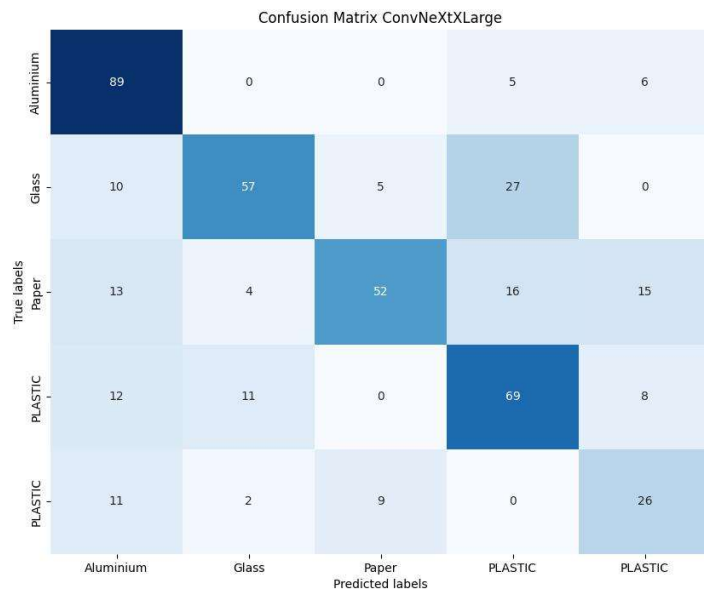
Σχήμα 3.15 Δείγμα εικόνων από την πρόβλεψη

- Μοντέλο ConvNeXtXLarge. Το μοντέλο ConvNeXtXLarge δημιουργήθηκε από τους Mingxing Tan, Bo Chen, Ruoming Pang και Quoc V. Le. Δημοσιεύτηκε για πρώτη φορά στο άρθρο "EfficientNetV2: Smaller Models and Faster Training" στις 24 Μαρτίου 2021.. Το αποτελέσματα που μας παρείχε σε σχέση με τους προηγούμενους ταξινομητές είναι υποδεέστερα παρόλα αυτά θα αναφερθούμε για να δούμε και τη διαφορά και στο πίνακα σύγκρισης «Σχήμα 3.16» αλλά και στο αποτέλεσμα της πρόβλεψης «Σχήμα 3.17». Η αρχιτεκτονική του με τα επίπεδα προσαρμογής προβάλλεται στο «Πίνακας 3-4».

Πίνακας 3-4 Αρχιτεκτονική του Convnext xlarge

Layer (Type)	Output Shape	Parameters
Convnext_xlarge (Functional)	(7, 7, 2048)	
conv2d (Conv2D)	(7, 7, 1024)	14156800
max_pooling2d_1 (MaxPooling2D)	(3, 3, 1024)	0
Dropout (Dropout)	(3, 3, 1024)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 150)	614550
Dropout1 (Dropout)	(None, 150)	0
Dense1 (Dense)	(None, 5)	755
Total parameters		69108841
Trainable parameters		14772105
Non-trainable parameters		54336736

F1-score on validation set: 0.6367422879144293



Σχήμα 3.16 Πίνακας σύγχυσης ConvNeXtXLarge



Σχήμα 3.17 Προβλέψεις από το μοντέλο ConvNeXtXLarge

3.8.3 Ταξινομητές ήχου

Για να ταξινομήσουμε τον ήχο χρησιμοποιήσαμε μόνο ίδιας αρχιτεκτονικής CNN ταξινομητές και μάλιστα ρηχής μάθησης, διότι τα τελικά δεδομένα μας μετά την επεξεργασία είχαν διαστάσεις 200x20x1 τις οποίες δεν μπορούσαν να επεξεργαστούν από βαθιά μοντέλα καθώς λόγω μεγέθους

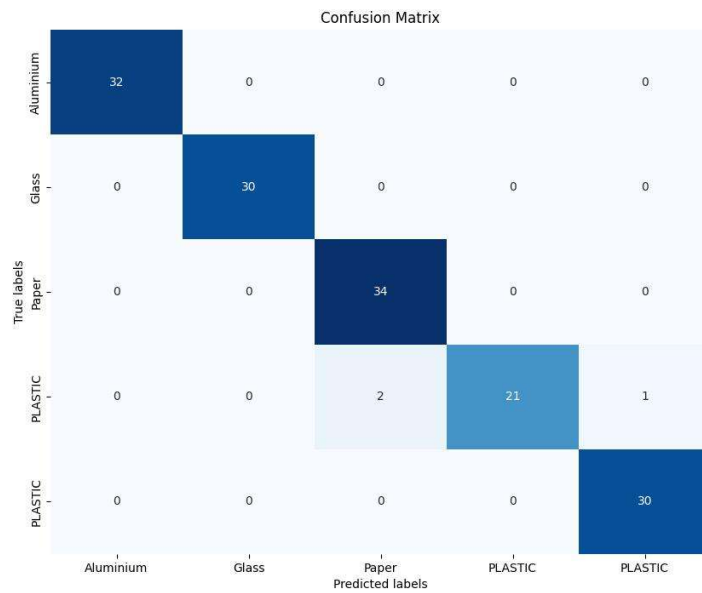
μηδένιζαν τα χαρακτηριστικά. Επίσης τα έτοιμα μοντέλα CNN δεν υποστηρίζουν την επεξεργασία αυτού του μεγέθους, η πιο μικρή διάσταση που υποστηρίζουν κάποια από αυτά, είναι 32x32x3. . Παρακάτω θα περιγραφούν δύο ταξινομητές ήχου και θα αποδοθούν και τα αποτελέσματα τους.

- Πρώτος ταξινομητής ήχου αποτελείται όπως θα δούμε αναλυτικά στο «Πίνακας 3-5» από 1 επίπεδο εισόδου, 2 κρυφά επίπεδα, 3 πλήρως συνδεδεμένα επίπεδα και 1 έξοδο. Ο ταξινομητής έβγαλε F1-Score 0,96 και ο πίνακας σύγχυσης. «»

Πίνακας 3-5 Δομή CNN για ήχο

Layer (Type)	Output Shape	Parameters
Conv2d (Conv2d)	(198, 18, 64)	640
Dropout (Dropout)	(198, 18, 64)	0
max_pooling2d_1 (MaxPooling2D)	(99, 18, 64)	0
Conv2d_1 (Conv2d)	(97, 16, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(48, 16, 128)	0
Conv2d_2 (Conv2d)	(46, 14, 512)	590336
max_pooling2d_1 (MaxPooling2D)	(23, 7, 512)	0
Flatten (Flatten)	(None, 82432)	0
Dence (dense)	(none, 1000)	82433000
Dence (dense)	(none, 512)	512512
Dence (dense)	(none, 128)	65664
Dence (dense)	(none, 5)	645
Total parameters		83676653
Trainable parameters		83676653
Non-trainable parameters		0

F1-score on validation set: 0.9614722879124291



Σχήμα 3.18 Πίνακας σύγκρισης πρώτου ταξινομητή ήχου

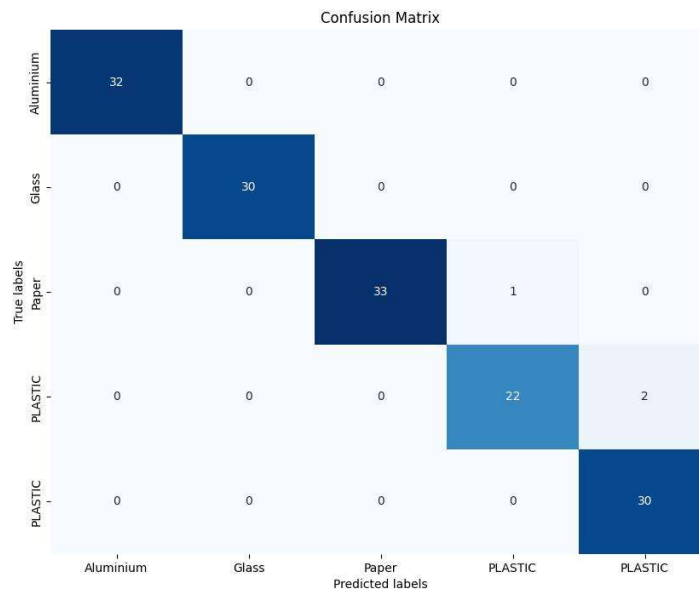
- Αυτός ο ταξινομητής είναι παρόμοιος με τον προηγούμενο αλλά έχει 1 κρυφό επίπεδο περισσότερο δηλαδή, έχει 1 επίπεδο εισόδου, 3 κρυφά επίπεδα, 3 πλήρως συνδεδεμένα επίπεδα και 1 έξοδο αναλυτικά παρουσιάζονται «Πίνακας 3-6». Ο ταξινομητής πέτυχε 0,97 F1-Score και ο πίνακας σύγκρισης μας δείχνει τη ταξινόμηση που πέτυχε.

Πίνακας 3-6 Δεύτερο CNN για ήχο

Layer (Type)	Output Shape	Parameters
conv2d (Conv2D)	(198, 18, 32)	320
Dropout (Dropout)	(198, 18, 32)	0
max_pooling2d (MaxPooling2D)	(99, 18, 32)	0
Conv2d_1 (Conv2d)	(97, 16, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(48, 16, 64)	0
Conv2d_2 (Conv2d)	(46, 14, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(23, 14, 128)	0
conv2d_3 (Conv2D)	(21, 12, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(10, 6, 128)	0
Flatten (Flatten)	(None, 82432)	0
Dence (dense)	(none, 1000)	7681000
Dence (dense)	(none, 512)	512512
Dence (dense)	(none, 128)	65664

Dence (dence)	(none, 5)	645
Total parameters		8500077
Trainable parameters		8500077
Non-trainable parameters		0

F1-score on validation set: 0.97345267879101251



Σχήμα 3.19 Πίνακας σύγχυσης δεύτερου ταξινομητή ήχου

3.8.4 Συνδυασμός των δύο αισθήσεων

Για τον συνδυασμό των προβλέψεων των δύο ταξινομητών, η τεχνική που χρησιμοποιήθηκε είναι ο μέσος όρος συνδυασμών πιθανοτήτων πρόβλεψης μοντέλων. Σε αυτή τη μέθοδο πρώτα παίρνουμε την πρόβλεψη από τον ένα ταξινομητή στη συγκεκριμένη περίπτωση του InceptionResnet και στη συνέχεια παίρνουμε από τον δεύτερο ταξινομητή που είναι του ήχου και τα προσθέτουμε τα διαιρούμε με το δύο (από δυο ταξινομητές προβλέψεις) και στη συνέχεια κρατάμε το μέγιστο και από τις 5 κλάσης υλικών. Για παράδειγμα ο ταξινομητής εικόνας, σε μία εικόνα από ένα πλαστικό μπουκάλι προβλέπει, οι πιθανότητες να είναι αλουμίνιο είναι 0.1, οι πιθανότητες να είναι γυαλί 0.2, οι πιθανότητες να είναι πλαστικό 0.5 και 0,2 να είναι πλαστικό ποτήρι. Ο ταξινομητής ήχου από την άλλη προβλέπει 0 πιθανότητες σε όλα εκτός από το πλαστικό που προβλέπει 1. Ο συνδυασμός αυτός των αποτελεσμάτων επιτυγχάνεται προσθέτοντας τις εκάστοτε πιθανότητες μεταξύ τους και διαιρώντας τη κάθε μια με το 2 έτσι αυξάνει η μέση επίδοση του συστήματος «Πίνακας 3-7». Σε 50 δοκιμές που έκανε το σύστημα οι ταξινομητές κατόρθωσαν το εξής F1-score:

F1-score for images: 0.6281481481481481

F1-score for sound: 1.0

Combined F1-score: 0.7959905660377359

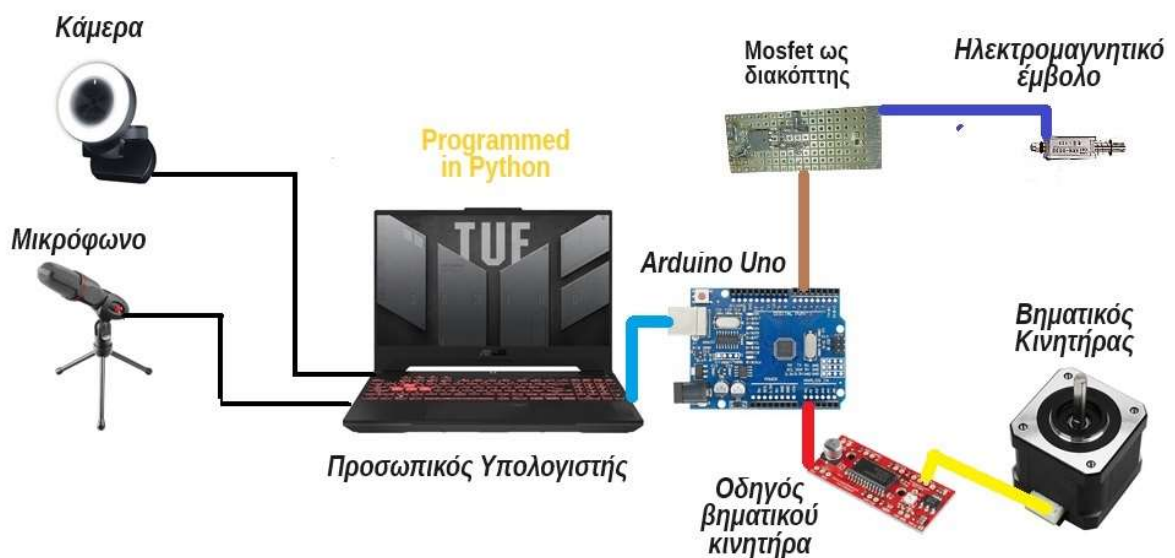
Όπως παρατηρούμε το μοντέλου του ήχου πέτυχε το απόλυτο, αλλά σε συνδυασμό με το μοντέλο της εικόνας το τελικό αποτέλεσμα παρουσιάζει βελτίωση.

Πίνακας 3-7 Παράδειγμα συνδυασμού πιθανοτήτων πρόβλεψης μοντέλων

Κατηγορία Υλικού	Προβλέψεις Ταξ. Εικόνας	Προβλέψεις Ταξ. ήχου	Μέση τιμή
Αλουμίνιο	0,1	0	0,05
Γυαλί	0,2	0	0,1
Πλαστικό	0,5	1	0,75
Πλαστικό ποτήρι	0,2	0	0,1
Μέγιστη μέση τιμή			0,75 κατηγορία πλαστικό

3.9 Κατασκευή για φυσική αναγνώριση

Το σύστημα το οποίο κατασκευάστηκε για να γίνει η φυσική αναγνώριση των υλικών μας αποτελείται από δύο κύρια μέρη, τον Ηλεκτρονικό υπολογιστή όπου είναι συνδεδεμένα η κάμερα και το μικρόφωνο και την αναπτυξιακή πλακέτα arduino που σε συνδυασμό με το σερβοκινητήρα και το ηλεκτρομαγνητικό έμβολο κάνει τις απαραίτητες κινήσεις «Σχήμα 3.20».



Σχήμα 3.20 Το σύστημα διάλογης

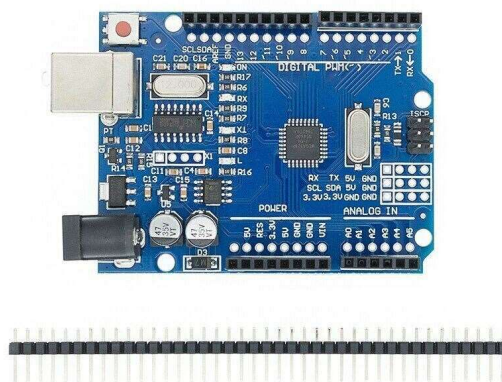
3.9.1 Arduino & Arduino Uno

Το όνομα "Arduino" προέρχεται από το όνομα του από ένα μπαρ στην Ivrea της Ιταλίας, όπου ο Massimo Banzi και οι David Cuartielles, David Mellis, Tom Igoe, Gianluca Martino και Casey Reas. συναντήθηκαν για να σχεδιάσουν το πρώτο πρωτότυπο του Arduino. Αυτό το μπαρ είχε το όνομα "Bar di Re Arduino," που μεταφράζεται στα αγγλικά ως "Arduino's Bar" ή απλά "Arduino" στο ευρύτερο κοινό .

Το Arduino ξεκίνησε ως μία εργασία στο πανεπιστήμιο Ivrea Interaction Design Institute της Ιταλίας. Η ομάδα που ανέλαβε το έργο αποτελούνταν από τους Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe, Gianluca Martino και Casey Reas. Η πρώτη πλακέτα Arduino παρουσιάστηκε το 2005 και από τότε έχει γνωρίσει μεγάλη επιτυχία και δημοτικότητα στον κόσμο του DIY, των ηλεκτρονικών και του προγραμματισμού.

Η φιλοσοφία του Arduino βασίζεται στην ευκολία χρήσης και στην ανοικτή πρόσβαση. Οι δημιουργοί του επιθυμούσαν να δημιουργήσουν ένα εργαλείο που θα έκανε τον προγραμματισμό και την ανάπτυξη ηλεκτρονικών συσκευών προσιτή σε όλους, ακόμη και σε αρχάριους. Έτσι, το Arduino παρέχει μια φιλικό προς τον χρήστη περιβάλλον, το Arduino IDE. Το περιβάλλον αυτό είναι προσαρμοσμένο για τους αρχάριους χρήστες έτσι η γλώσσα προγραμματισμού που χρησιμοποιείται είναι απλοποιημένη C/C++. Παρέχει επίσης ένα ευρύ φάσμα βιβλιοθηκών κώδικα, και μια ενεργή κοινότητα που μοιράζεται γνώσεις και έργα. Αυτή η προσέγγιση έχει κάνει το Arduino ιδιαίτερα δημοφιλές σε φοιτητές, ερασιτέχνες, καλλιτέχνες, επαγγελματίες και εκπαιδευτικούς παγκοσμίως.

Το Arduino Uno «Σχήμα 3.21» κυκλοφόρησε για πρώτη φορά το 2010 από την Arduino LLC. Ήταν μια αναβάθμιση του προηγούμενου μοντέλου Arduino Diecimila και αντικατέστησε το Arduino Duemilanove. Το Arduino Uno έγινε αμέσως ευρέως αποδεκτό λόγω της απλότητάς του στη χρήση, λόγω των χαρακτηριστικών του «Πίνακας 3-8» που υπερτερούν σε σχέση με τα προηγούμενα μοντέλα, της ευελιξίας του και της μεγάλης κοινότητας που το υποστήριξε. Από τότε έχει παραμείνει ένα από τα πιο δημοφιλή μοντέλα Arduino στην αγορά.



Σχήμα 3.21 Arduino uno

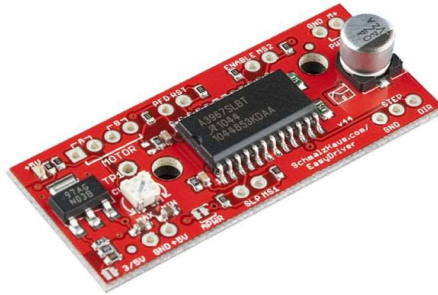
(Πηγή: <https://www.ebay.co.uk/>)

Πίνακας 3-8 Χαρακτηριστικά του Arduino uno

Microcontroller:	ATmega328p
Operating Voltage:	5V
Input Voltage (recommended):	7-12V
Inout Voltage (limit):	6-20V
Digital I/O Pins:	14 and 6 can be used as PWM
PWM Digital I/O Pins:	6 PWM pins
Analog Input Pins:	6 pins
Serial Ports:	1 port
I2C Ports:	1 port
SPI	1 port
DC Current per I/O Pin:	20mA
DC current for 3.3V Pin:	50mA
Flash Memory:	32 KB of which 0.5 KB used by bootloader
SRAM:	2kB
EEPROM:	1kB
Clock Speed:	16 MHz
Length:	68.6mm
Width:	58.4mm
Weight:	25gr

3.9.2 Οδηγός βηματικού κινητήρα

Ο οδηγός του βηματικού κινητήρα που χρησιμοποιήθηκε στη παρούσα εργασία είναι ο EasyDrive «Σχήμα 3.22» της Sparkfun. Ο EasyDriver είναι ένας οδηγός βηματικών κινητήρων που χρησιμοποιείται για τον έλεγχο τους, είναι ένας δημοφιλής οδηγός βηματικών κινητήρων στην κοινότητα των ερασιτεχνών και των μηχανικών που ασχολούνται με τα ρομπότ, τις εκτυπώσεις 3D, τα CNC και άλλες εφαρμογές που απαιτούν ακριβή κίνηση. Παρέχει μια εύκολη λύση για τον έλεγχο βηματικών κινητήρων, χωρίς την ανάγκη για πολύπλοκα κυκλώματα ή προηγμένες γνώσεις ηλεκτρονικής.



Σχήμα 3.22 EasyDriver

(Πηγή: <https://grobotronics.com/easydriver-stepper-motor-driver.html>)

Ο EasyDriver υποστηρίζει τρεις κύριες λειτουργίες:

- **Ελέγχος Βημάτων:** Ο EasyDriver μπορεί να ελέγχει τον αριθμό των βημάτων που κάνει ο βηματικός κινητήρας. Μπορούμε να προγραμματίσουμε τον ακριβή αριθμό των βημάτων που θέλουμε να κάνει ο κινητήρας, ελέγχοντας την ταχύτητα και την κατεύθυνσή του.
- **Ρύθμιση Ρεύματος:** Ο EasyDriver σας επιτρέπει να ρυθμίσουμε το ρεύμα που διέρχεται από τον βηματικό κινητήρα. Αυτό είναι χρήσιμο για την αντιστοίχιση του ρεύματος με τις απαιτήσεις του κινητήρα και την αποφυγή υπερθέρμανσης.
- **Μικροβηματικό Βήμα:** Ο EasyDriver υποστηρίζει επίσης τη λειτουργία μικροβηματικού βήματος, που επιτρέπει στον κινητήρα να κινείται με μεγαλύτερη ακρίβεια και ομαλότητα.

Βασίζεται στο ολοκληρωμένο κύκλωμα A3967 της Allegro Microsystems. Αυτός ο ολοκληρωμένο χρησιμοποιείται ευρέως για τον έλεγχο βηματικών κινητήρων και προσφέρει πολλές ρυθμίσεις και επιλογές για τον έλεγχο της κίνησης και του ρεύματος. Έχει υιοθετεί την τεχνολογία ελέγχου ρεύματος. Παρέχει δυνατότητες ρύθμισης ρεύματος και μπορεί να λειτουργήσει σε ρεύματα έως 750mA.

3.9.3 Ο βηματικός κινητήρας

Ο βηματικός κινητήρας που χρησιμοποιήθηκε είναι ο Nema 17 «Σχήμα 3.23». Ο οποίος είναι ένας ηλεκτρομηχανικός κινητήρας συνεχούς ρεύματος, ονομάστηκε "βηματικός" κινητήρας λόγω του τρόπου λειτουργίας του. Κάθε βήμα του κινητήρα αντιστοιχεί σε ένα συγκεκριμένο γωνιακό ή γραμμικό μετατοπισμό που συμβαίνει λόγω του μαγνητικού πεδίου που δημιουργείται κατά τη διέλευση του ρεύματος στο επιλεγμένο ή επιλεγμένα τυλίγματα του .

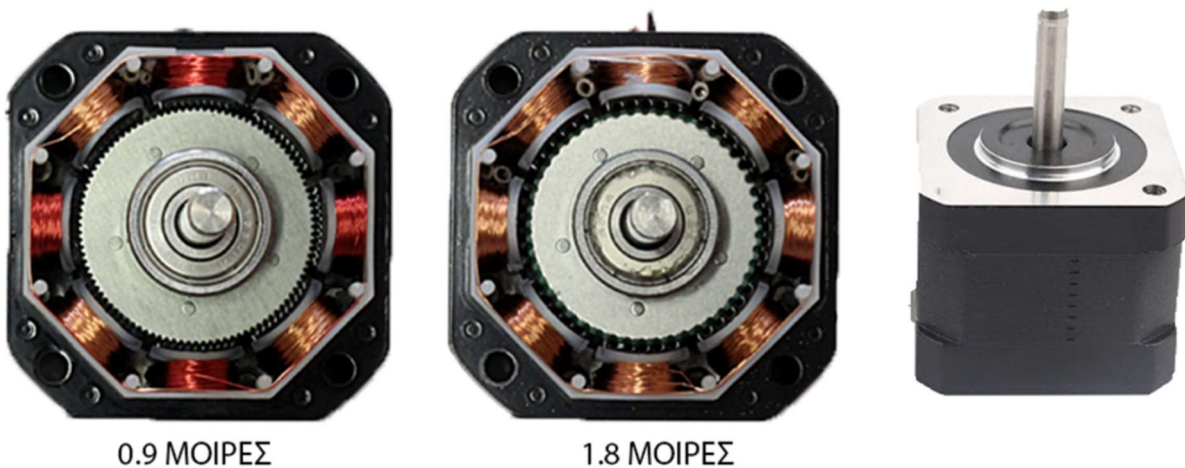


Σχήμα 3.23 Nema 17

(Πηγή: <https://aus3d.com.au/products/nema-17-e3d-titan-slimline-stepper-motor>)

Η δομή του βηματικού κινητήρα χωρίζεται σε δυο μέρη «Σχήμα 3.24»:

- Το στάτη που έχει τους ηλεκτρομαγνήτες, δηλαδή περιέχει την περιέλιξη των πηνίων που δέχονται τη τάση.
- Το ρότορα που έχει απάνω του μόνιμους μαγνήτες σε οδοντωτό σχήμα.



0.9 ΜΟΙΠΕΣ

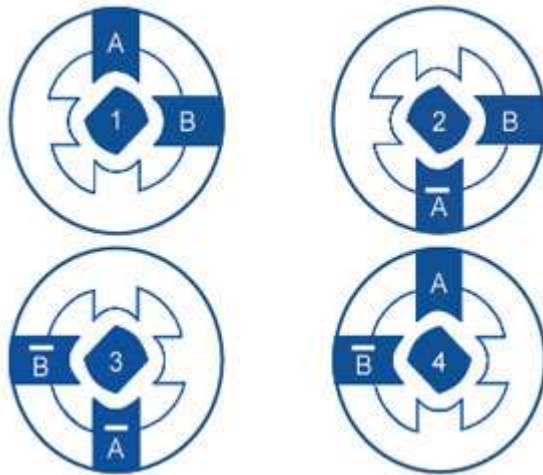
1.8 ΜΟΙΠΕΣ

Σχήμα 3.24 Εσωτερικό βηματικού κινητήρα

(Πηγή: <https://fuselab.gr/ti-einai-vimatikos-kinitiras-stepper-motor/>)

Η περιστροφή του βηματικού κινητήρα γίνεται ενεργοποιώντας και απενεργοποιώντας τους ηλεκτρομαγνήτες ένα προς ένα. Για παράδειγμα στο σχήμα«» όταν ρεύμα διέλθει του πηνίου Α τότε ο ρότορας περιστρέφεται προς αυτό στη συνέχεια κόβοντας τη τροφοδοσία στο Α και τροφοδοτώντας το Β ο ρότορας περιστρέφεται προς το Β, το ίδιο για το Α ανάστροφο και Β ανάστροφο. Ανάλογα με το πως είναι τοποθετημένα τα πηνία στο στάτη του βηματικού κινητήρα, αλλά και με τί σειρά και ταχύτητα ενεργοποιούνται, πραγματοποιούν τα βήματα τους και τη φορά περιστροφής τους. Υπάρχουν τρεις κατηγορίες βηματισμού και είναι πλήρους βήματος (Full step), μισού βήματος (Half step) και μικροβηματισμού (Micro-step).

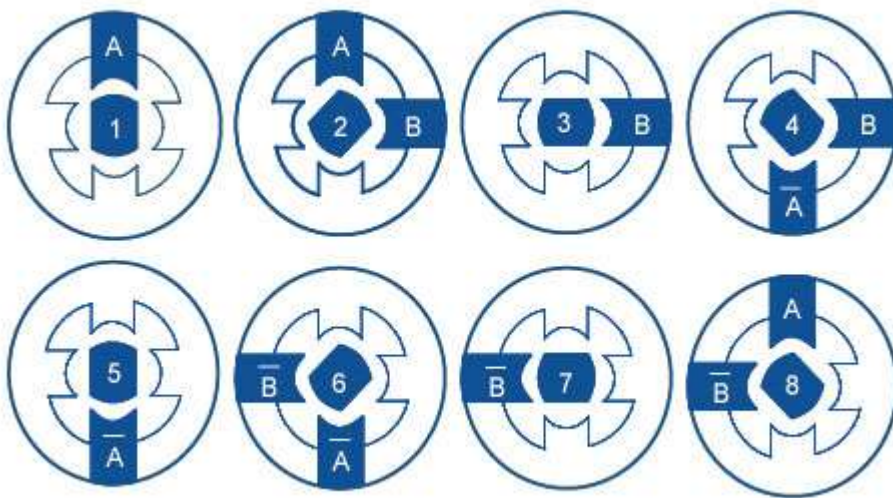
- Πλήρους βηματισμού (Full step), σε αυτό το βηματισμό ενεργοποιούνται πάντα 2 πηνία «Σχήμα 3.25»



Σχήμα 3.25 Full step

(Πηγή: <https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html>)

- Μισού βήματος (Half step),εδώ ο βηματικός κινητήρας ενεργοποιεί το κάθε πηνίο μόνο του αλλά και τα δύο μαζί αυτό σε σχέση με πριν επιτυγχάνει ενδιάμεσα βήματα «
- Σχήμα 3.26»



Σχήμα 3.26 Half step

(Πηγή: <https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html>)

- Μικροβηματισμού (Micro step) είναι μια τεχνική που χρησιμοποιείται για την εξομάλυνση της κίνησης ενός βηματικού κινητήρα και την αύξηση της ακρίβειας της θέσης του. Κανονικά, ένας βηματικός κινητήρας κινείται από ένα βήμα στο επόμενο όταν ενεργοποιείται ένας από τους ηλεκτρομαγνήτες ή και οι δύο . Ωστόσο, με τον μικροβηματισμό, το ρεύμα που δίνεται στους ηλεκτρομαγνήτες διαμορφώνεται σε έναν τρόπο που δημιουργεί ενδιάμεσα

βήματα μεταξύ των βασικών θέσεων του κινητήρα. Αν πούμε αρχικά ότι το πηνίο A διαρρέετε από το μέγιστο ρεύμα και σιγά σιγά μειώνεται και αυξάνεται στο πηνίο B τότε δημιουργούνται ενδιάμεσα βήματα.

Τις μοίρες που περιστροφής που αναφέρεται στο βηματικό κινητήρα, τις ορίζουν τα δοντάκια στον ρότορα και αναφέρονται πάντα σε Full Step βηματισμό«», αν χρησιμοποιηθεί Half step βηματισμός σε ένα βηματικό κινητήρα 1.8° τότε ο κινητήρας θα κάνει και ενδιάμεσα βήματα 0.9°.

Ο Nema 17 είναι ένας πολύ χρησιμοποιημένος βηματικός κινητήρας σε κατασκευές 3D printer, plotter mini CNC και laser engravers και σε άλλες εφαρμογές. Τα χαρακτηριστικά του συμφωνά με το φύλλο δεδομένων του είναι «Πίνακας 3-9».

Πίνακας 3-9 Χαρακτηριστικά Nema 17

Step angle	1.8°
Rated Voltage	12V
Current per phase	350 mA
Holding torque	1.6kg/cm
No. of leads	4
Shaft diameter	5mm
Shaft length	±24mm
Weight	220gr

3.9.4 Ηλεκτρομαγνητικό έμβολο

Η λειτουργία ενός ηλεκτρομαγνητικού εμβόλου έχει ως θεμελιώδη αρχή τη δύναμη Lorentz, που μας λέει ότι όταν ένας αγωγός διαρρέεται από ρεύμα δημιουργείται ένα μαγνητικό πεδίο γύρω του. Όταν αυτός ο αγωγός τοποθετείται μέσα σε ένα εξωτερικό μαγνητικό πεδίο, η αλληλεπίδραση μεταξύ των δύο πεδίων παράγει μια δύναμη που ενεργεί στον αγωγό, οδηγώντας σε κίνηση. Αυτή η δύναμη, γνωστή ως δύναμη Lorentz, είναι ευθέως ανάλογη με το ρεύμα που διαρρέει τον αγωγό και την ισχύ του εξωτερικού μαγνητικού πεδίου.

Το ηλεκτρομαγνητικό έμβολο που χρησιμοποιήθηκε για να χτύπα τα αντικείμενα έτσι ώστε να παράγουν ήχο είναι το JF-0530B «Σχήμα 3.27» τα χαρακτηριστικά του αναφέρονται «Πίνακας 3-10».



Σχήμα 3.27 Ηλεκτρομαγνητικό έμβολο

Πίνακας 3-10 Χαρακτηριστικά ηλεκτρομαγνητικού εμβόλου

Operating voltage $t < 5s$	12V
Operating current	1A
Shaft length	10mm
Force	5N
Weight	30gr

3.9.5 Mosfet σε λειτουργία διακόπτη

Για να παρέχουμε την απαιτούμενη τάση και ρεύμα για να ενεργοποιήσουμε το ηλεκτρομαγνητικό έμβολο χρησιμοποιήσαμε ένα mosfet σε συνδεσμολογία διακόπτη «Σχήμα 3.28». Οι αναλογικοί διακόπτες όπως αναφέρονται είναι ηλεκτρονικά εξαρτήματα που χρησιμοποιούνται για τον έλεγχο της ροής ή της διέλευσης αναλογικών σημάτων, όπως τάση και το ρεύμα, σε ένα κύκλωμα. Οι αναλογικοί διακόπτες επιτρέπουν τον έλεγχο αναλογικών σημάτων με βάση κάποια είσοδο ελέγχου.

Λειτουργούν συνήθως με δύο κύριους τρόπους: μεταβολή της αντίστασης ή το πλήρες άνοιγμα - κλείσιμο, ανάλογα με το σχεδιασμό τους. Αυτό επιτρέπει στο ρεύμα ή την τάση να περάσουν ανάλογα με την εφαρμογή.

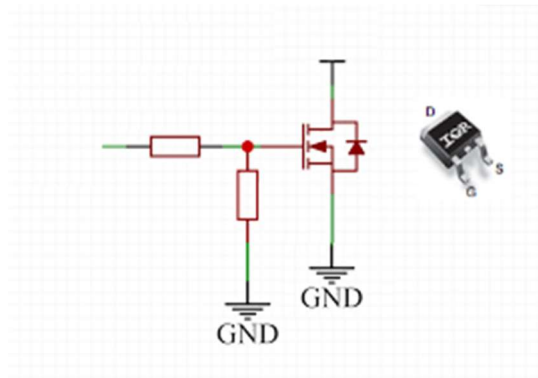
Οι αναλογικοί διακόπτες χρησιμοποιούνται σε πολλές εφαρμογές, όπως η ρύθμιση της φωτεινότητας σε οπίσθιους φωτισμούς οθονών, ο έλεγχος της έντασης ή του ηχητικού όγκου σε ηχοσυστήματα, και η ρύθμιση της έντασης σε εφαρμογές σήμανσης, για standby κ.α.

Το mosfet που χρησιμοποιήθηκε είναι N channel και συγκεκριμένα είναι το IRLR3636U το οποίο έχει τα εξής χαρακτηριστικά «».

Πίνακας 3-11 Τα χαρακτηριστικά του IRLR3636U

Type	N-Channel
V_{DS}	60V
I_{DS}	50A
$R_{DS\ on\ max}$	6.8mΩ

V_{GS}	$\pm 16V$
I_{GS}	100nA
$V_{GS(th)}$	2.5V
Power Consumption	143W



Σχήμα 3.28 Mosfet ως διακόπτης

3.9.6 Κάμερα

Η όραση του συστήματος παρέχεται από μία κάμερα με υψηλή ανάλυση και έχει και φωτισμό. Με τη κάμερα το σύστημα μετατρέπει την εικόνα που βλέπει από αναλογική σε ψηφιακή. Αυτό που κάνει όμως πιο πρακτικά στο σύστημα είναι να τραβά φωτογραφίες από το αντικείμενο που είναι προς πρόβλεψη. Η κάμερα που χρησιμοποιείται στο σύστημα είναι μια usb web κάμερα και συγκεκριμένα η Razer KIYO«. Τα χαρακτηριστικά της αναγράφονται στο «Πίνακας 3-12»



Σχήμα 3.29 Κάμερα Razer Kiyo

Πίνακας 3-12 Χαρακτηριστικά της κάμερας

Optical sensor analyses	4megapixel
Focus settings	Auto-Focu
Max frame rate	60 fps
Rotation	180°

Σύνδεση	USB
---------	-----

3.9.7 Μικρόφωνο

Η ακοή του συστήματος είναι βασισμένη στο μικρόφωνο το οποίο συλλέγει τον αναλογικό ήχο από τα αντικείμενα και τον μετατρέπει σε αναλογική τάση. Στη συνέχεια μετατρέπεται σε ψηφιακό σήμα από την κάρτα ήχου. Το μικρόφωνο που χρησιμοποιήσαμε είναι το TRUST GX212 «».

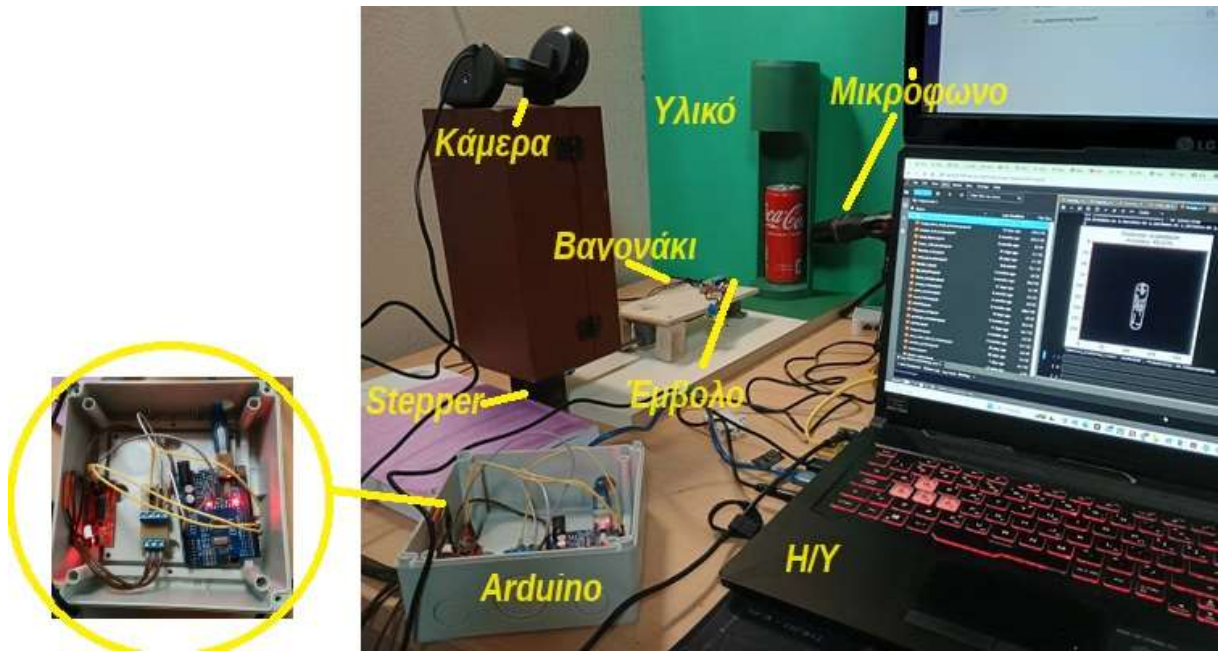


Σχήμα 3.30 Το μικρόφωνο Trust GX212

3.9.8 Λειτουργία του συστήματος.

Το σύστημα μας όπως φαίνεται «Σχήμα 3.31», αποτελείται :

- Από το σημείο που τοποθετούμε τα υλικά και ουσιαστικά προσημειώνουμε τον κάδο.
- Το βαγονάκι, το οποίο μετακινείται από τον βηματικό κινητήρα, και επάνω του έχει το ηλεκτρομαγνητικό έμβολο.
- Τη κάμερα που είναι τοποθετημένη απέναντι από το σημείο που έχουμε τοποθέτηση το υλικό προς πρόβλεψη και ανακύκλωση.
- Το μικρόφωνο που είναι τοποθετημένο δίπλα από το ανακυκλώσιμο υλικό.
- Σε πλαστικό κουτί δίπλα από τον Η/Υ έχουμε τοποθετημένο το Arduino με το easydrive και το αναλογικό διακόπτη που ελέγχει το ηλεκτρομαγνητικό έμβολο.



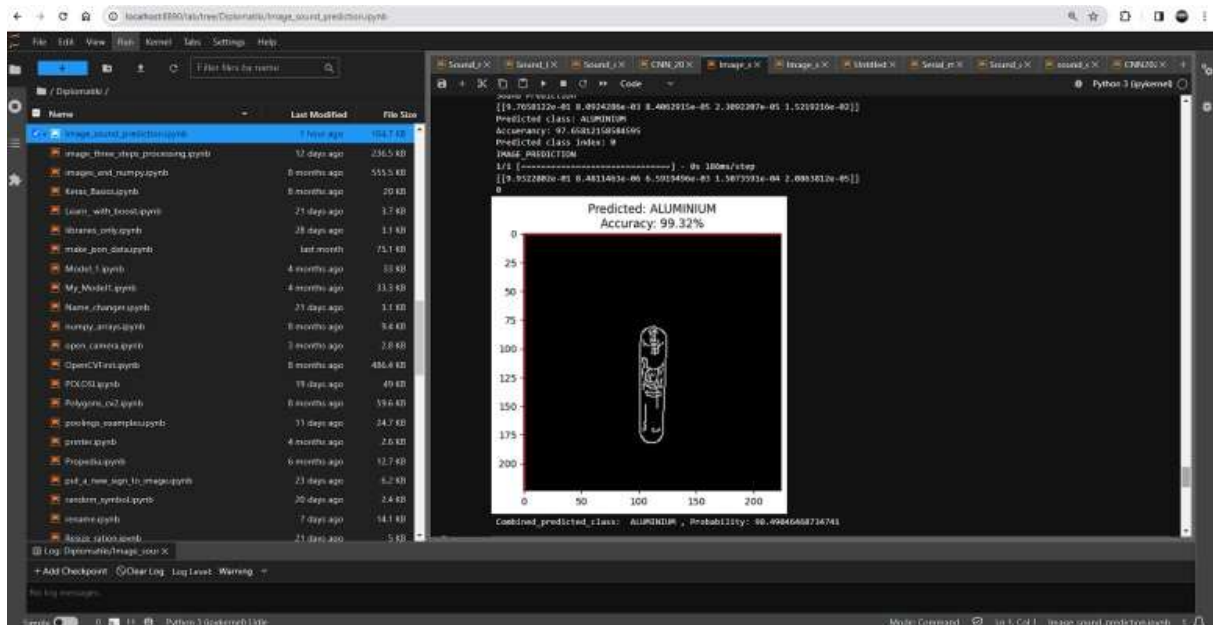
Σχήμα 3.31 Το σύστημα διαλογής

Αφού θέσουμε το σύστημα μας σε λειτουργία, το βαγονάκι είναι σταματημένο προ την κάμερα, ενώ έχουμε τοποθετήσει το υλικό προς αναγνώριση στο σωλήνα πατάμε το πλήκτρο S από το πληκτρολόγιο του Η/Υ, το πλήκτρο S προσημειώνει το φωτοκύτταρο που είδε το υλικό στο σωλήνα και το σύστημα ξεκινά. Ανοίγει την κάμερα τραβά φωτογραφία από το υλικό το οποίο είναι τοποθετημένο στο σημείο που προσομοιώνει την υποδοχή του κάδου «Σχήμα 3.32».



Σχήμα 3.32 Κάμερα απέναντι από το υλικό

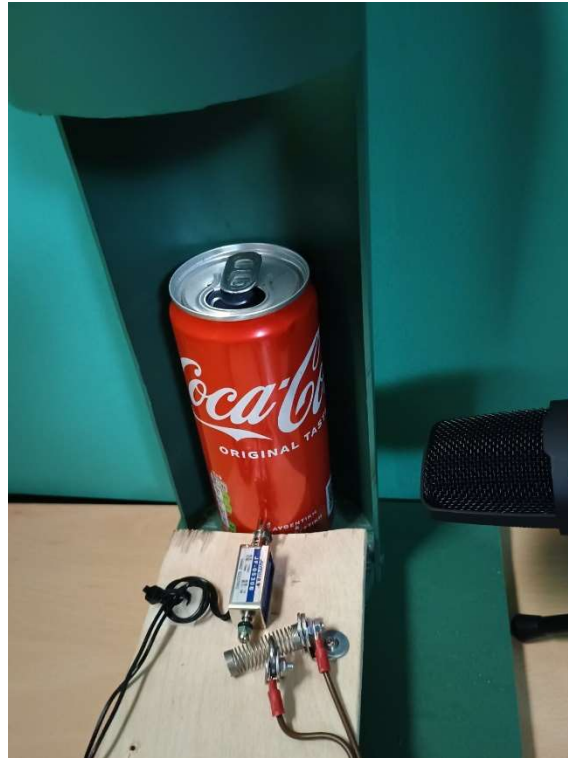
Αποθηκεύει την εικόνα στον Η/Υ, εκεί την ανακαλεί η εφαρμογή, την επεξεργάζεται, την τροφοδοτεί στον ταξινομητή και βγάζει μια πρόβλεψη, βάση της εκπαίδευσης του στη μηχανική όρασης «Σχήμα 3.33».



Σχήμα 3.33 Πρόβλεψη στην εικόνα από τον ταξινομητή

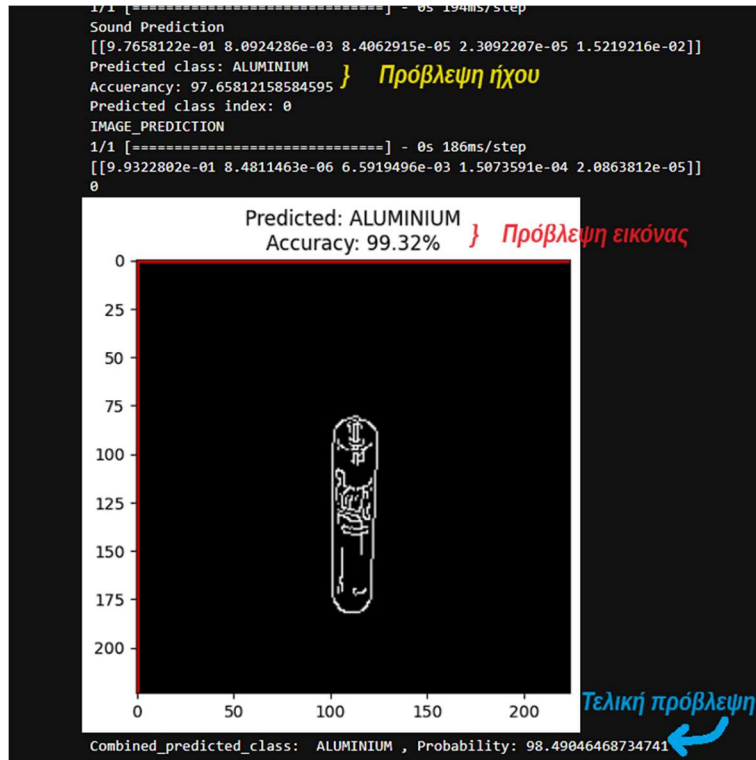
Στη συνέχεια μέσω της σειριακής ειδοποιεί το Arduino και εκείνο δίνει εντολή για να μετακινηθεί το βαγονάκι με το έμβολο προς το υλικό. Μόλις πλησιάσει αρκετά ενεργοποιείται ένας διακόπτης ελατηρίου που είναι συνδεδεμένος με το Arduino και το Arduino σταματά την κίνηση του βαγονιού

ειδοποιεί την εφαρμογή στον Η/Υ ότι είναι έτοιμο και εκείνη δίνει εντολή στο Arduino να ξεκινήσει το έμβολο και ξεκινά την εγγραφή του ήχου «Σχήμα 3.34».



Σχήμα 3.34 Συλλογή του ήχου

Το έμβολο χτυπά τρεις φορές το αντικείμενο για να είμαστε σίγουροι ότι έχει καταγράψει τον ήχο, στη συνέχεια τον αποθηκεύει στον Η/Υ, εκεί τον ανακαλεί η εφαρμογή τον επεξεργάζεται, την τροφοδοτεί στον ταξινομητή και βγάζει μια πρόβλεψη, βάση της εκπαίδευσης του στη μηχανική ακοή. «Σχήμα 3.35». Στο τέλος εμφανίζει και τις τρεις προβλέψεις μαζί, δηλαδή την πρόβλεψη του ταξινομητή εικόνας, την πρόβλεψη του ταξινομητή ήχου και την συνδυασμένη πρόβλεψη.



Σχήμα 3.35 Πρόβλεψη ήχου, εικόνας και συνδυαστική

3.10 Επίλογος

Σε αυτό το κεφάλαιο είδαμε κάποιες από τις μετρήσεις και τις υλοποιήσεις που έγιναν κατά τη διάρκεια της πειραματικής διαδικασίας. Αξίζει να σημειωθεί ότι και κατά την εκπαίδευση αλλά και στο στατικό έλεγχο που έγινε στους ταξινομητές είχαμε καλά αποτελέσματα. Αλλά και στο φυσικό σύστημα αναγνώρισης και διαλογής είχαμε πολύ καλά αποτελέσματα. Είδαμε επίσης τα μέρη που απαρτίζουν το σύστημα.

Κεφάλαιο 4ο: Συμπεράσματα

Από την ερευνά και την ενασχόληση με το αντικείμενο βαθιάς μάθησης με τη χρήση μηχανικής όρασης και ακοής, αρχικά αντιμετωπίσαμε τη δυσκολία της συλλογής των συγκεκριμένων δεδομένων στη συνέχεια μέσα από τα πειράματα αποδείχθηκε ότι οι εικόνες και ο ήχος, τουλάχιστον κατά την εκπαίδευση θα πρέπει να είναι καθαρά (Χωρίς φόντο και θόρυβο). Είδαμε ότι χρειάζονται να γίνουν πάρα πολλά πειράματα για να καταλήξουμε να έχουμε ένα ταξινομητή που να έχει καλές επιδόσεις και γενίκευση πράγμα που το καθιστά δύσκολο για ένα άτομο συνήθως τα μοντέλα δουλεύονται από ομάδες επιστημόνων. Επίσης για να έχουμε καλή επίδοση ο ταξινομητής θα πρέπει να εκπαιδεύεται σε πολλές κλάσεις και η κάθε κλάση να έχει πολλά δεδομένα. Αυτό βοηθά και στην εκπαίδευση αλλά και στη γενίκευση. Όμως όσα περισσότερα δεδομένα τόσο περισσότερος χρόνος χρειάζεται για την εκπαίδευση και τόσο περισσότερη υπολογιστική ισχύ χρειαζόμαστε.

Ο τελικός ταξινομητής που χρησιμοποιήθηκε είδαμε ότι αυξάνει κατά μικρό ποσοστό την επιτυχή πρόβλεψη και δεν εξαρτάται αποκλειστικά από ένα παράγοντα. Ο ταξινομητής θα μπορούσε να έχει πιθανός καλύτερα αποτελέσματα, προσθέτοντας δύο ακόμα αισθητήρες έναν αισθητήρα που να αντιλαμβάνεται το μέταλλο, δυναμοκυψέλες για τον εντοπισμό της σκληρότητας του αντικειμένου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] R. j. Satler και W. Sternberg, *Handbook of Human Intelligence.*, Cambridge: Cambridge University Press, 1982.
- [2] S. Rothman και M. Snyderman, «Survey of Expert Opinion on Intelligence and Aptitude Testing,» *American Psychologist*, τόμ. 42, αρ. 2, pp. 137-142, 1987.
- [3] S. Legg και M. Hutter, «A Collection of Definitions of Intelligence,» *Frontiers in Artificial Intelligence and Applications*, τόμ. 157, αρ. 2, pp. 17-24, 15 JUNE 2007.
- [4] Φ. Κόκκορας, Ν. Βασιλειάδης, Η. Σακελλαρίου, Ι. Βλαχάβα και Π. Κεφαλάς, *Τεχνητή Νοημοσύνη*, Θεσσαλονίκη: Πανεπιστημίου Μακεδονίας, 2020.
- [5] M. M. Maad, «History of Artificial Intelligence.,» *To be published.*, Accessed: April 2015 [Online]. Available: https://www.researchgate.net/publication/322234922_History_of_Artificial_Intelligence.
- [6] A. W. Melton, *Categories of human learning*, London: Academic Press INC, 1964.
- [7] Κ. Διαμαντάρας και Δ. Α. Μπότσης, *Μηχανική Μάθηση*, Αθήνα: Κλειδάριθμος, 2019.

ΠΑΡΑΡΤΗΜΑ Α : Χρήσιμα προγράμματα σε python

Πρόγραμμα για την αλλαγή διαστάσεων σε εικόνες χωρίς την αλλοίωση της αρχικής.

```
from PIL import Image
import os
# Διαδρομή προς τον φάκελο με τα αρχεία
folder_path = 'folders path (c:/ user/blablabla)'
# Λίστα με τα ονόματα όλων των αρχείων στο φάκελο
file_names = os.listdir(folder_path)
# Αρχικοποίηση του μέγιστου αριθμού
max_number = 0
# Ελέγχουμε κάθε όνομα αρχείου στο φάκελο
for file_name in file_names:
    # Ελέγχουμε αν το όνομα αρχείου αποτελείται μόνο από αριθμούς
    if file_name[:-4].isdigit():
        # Αντικαθιστούμε το πρόθεμα και το επίθημα για να πάρουμε τον αριθμό
        number = int(file_name[:-4])
        # Αν ο αριθμός είναι μεγαλύτερος από τον μέχρι τώρα μέγιστο, τον ενημερώνουμε
        if number > max_number:
            max_number = number
# Εκτύπωση του μέγιστου αριθμού
print(f'Το μεγαλύτερο όνομα-αριθμός αρχείου είναι το {max_number}.jpg')
def resize_image_with_padding(input_image_path, output_image_path, target_size):
    # Φορτώστε την αρχική εικόνα
    original_image = Image.open(input_image_path)
    # Υπολογίστε το μέγεθος της νέας εικόνας διατηρώντας την αναλογία διαστάσεων
    original_size = original_image.size
    ratio = min(target_size[0] / original_size[0], target_size[1] / original_size[1])
    new_dimensions = (int(original_size[0] * ratio), int(original_size[1] * ratio))
    # Κάντε resize την αρχική εικόνα
    resized_image = original_image.resize(new_dimensions, Image.BILINEAR)
    # Δημιουργήστε μια νέα λευκή εικόνα με το επιθυμητό μέγεθος
```

```

new_image = Image.new("RGB", target_size, "white")
# Τοποθετήστε την αρχική εικόνα στο κέντρο της νέας
new_image.paste(resized_image, ((target_size[0] - new_dimensions[0]) // 2,
                                (target_size[1] - new_dimensions[1]) // 2))
# Αποθηκεύστε τη νέα εικόνα
new_image.save(output_image_path)
for i in range(1,(max_number+1),1):
# Παράδειγμα χρήσης\
    resize_image_with_padding("C:\\Users\\blabla"+str(i)+".jpg","C:\\Users\\blabla"+str(i)+".jpg",
(224, 224))

```

Με το παρακάτω πρόγραμμα κάνουμε μαζική μετονομασία σε αρχεία

```

import os

def rename_files():
    directory = 'C:/Users/blablal' # Αντικατάστησε το με τον κατάλληλο διαδρομής του φακέλου που
περιέχει τα αρχεία
    counter = 9
    for filename in os.listdir(directory):
        if filename.endswith('.jpg'):
            original_path = os.path.join(directory, filename)
            new_filename = f'{counter}.jpg'
            new_path = os.path.join(directory, new_filename)
            os.rename(original_path, new_path)
            print(f'Renamed {filename} to {new_filename}')
            counter += 1
rename_files()

```

Πρόγραμμα για την αφαίρεση του φόντο

```

from rembg import remove
from PIL import Image
import cv2
import numpy as np
import os

```

```

#ALUMINIUM','GLASS','PET','PAPER',PET_CUPS
labels =[your last files name]
for j in labels:
    for i in range (1,63,1):
        img = Image.open("C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+".jpg")
        R = remove(img)
        R.save("C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+".png")
        os.remove("C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+".jpg")
    for i in range (1,63,1):
        # Example usage:
        png_file_path = 'C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+'.png'
        jpg_file_path = 'C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+'.jpg'
        background_color = (255, 255, 255)

        # Read the PNG image
        img = cv2.imread(png_file_path, cv2.IMREAD_UNCHANGED)

        # Create a new image with the desired background color
        new_img = np.full_like(img, background_color + (255,), dtype=np.uint8)

        # Extract the alpha channel from the original image
        alpha_channel = img[:, :, 3]

        # Composite the original image onto the new image using the alpha channel
        for c in range(0, 3):
            new_img[:, :, c] = new_img[:, :, c] * (1 - alpha_channel / 255) + img[:, :, c] * (alpha_channel /
255)
        # Save the image in JPG format
        cv2.imwrite(jpg_file_path, new_img, [int(cv2.IMWRITE_JPEG_QUALITY), 100])
        os.remove("C:\\Users\\blabla\\"+str(j)+"\\"+str(i)+".png")

```

Πρόβλεψη ήχου από τον ταξινομητή.

```

import pyaudio
import wave
import numpy as np
import librosa
from tensorflow.keras.models import load_model
from tqdm import tqdm
import keyboard
import serial
import time
#Εισαγωγή βιβλιοθηκών
import os
from tqdm import tqdm
import pandas as pd
from pathlib import Path
import numpy as np
import matplotlib.pyplot as plt
import tensorflow_io as tfio
import tensorflow as tf
from tensorflow.keras import layers, models
%matplotlib inline
from scipy.io import wavfile
from python_speech_features import mfcc, logfbank
import librosa
import librosa.display
import soundfile as sf
import random
from cycler import cycler
from scipy.ndimage import zoom

csv_file_path = "C:\\Users\\blabla\\audio_filenames.csv"

# Ορίζουμε τις στήλες που θέλουμε να επιλέξουμε
selected_columns = ['FILENAME', 'FOLDER']

```

```

# Διαβάζουμε το αρχείο CSV και επιλέγουμε τις επιθυμητές στήλες
data = pd.read_csv(csv_file_path, usecols=selected_columns)

category_mapping = {'ALUMINIUM': 0, 'GLASS': 1, 'PAPER': 2, 'PLASTIC': 3, 'PLASTIC_CUPS':
4}

# Εφαρμόζουμε τον αντιστοιχισμό στη στήλη 'FOLDER'
data['FOLDER'] = data['FOLDER'].map(category_mapping)

# Εμφανίζουμε τα πρώτα πέντε στοιχεία του DataFrame
# Αποθηκεύουμε τα δεδομένα σε μια μεταβλητή
selected_data = data[selected_columns]
#print(selected_data)

datas = pd.read_csv(csv_file_path)
classes=list(np.unique(datas.FOLDER))
print(classes)

#####
#####

#####
#####

#!!!!!!!!!!!!!!Δημιουργία κλάσης για το άνοιγμα και επεξεργασία του ήχου!!!!!!!!!!!!!!#

#####
#####

#####
#####

class AudioActivities():

#####
#####

#Βήμα 1ο ανοίγουμε το αρχείο ήχου. #

#####
#####

    @staticmethod
    def open_audio(audio_file):
        signal,sr = librosa.load(audio_file, sr=44100)
        return (signal,sr)

```

```

# -----Τέλος open_audio-----#

#####

#Βήμα 2ο απεικόνιση των χαρακτηριστικών του σήματος #
#####

    @staticmethod
    def get_audio_info(audio):
        signal, sr = audio
        num_samples = len(signal)
        sample_rate = sr
        bit_depth = signal.dtype.itemsize * 8
        num_channels = signal.shape[0] if signal.ndim > 1 else 1
        duration = librosa.get_duration(y=signal, sr=sr)
        num_frames = int(duration * sample_rate)
        # Εκτύπωση πληροφοριών
        print("Συχνότητα δειγματοληψίας:", sample_rate, "Hz")
        print("Πλήθος bits ανά δείγμα:", bit_depth)
        print("Μήκος σε πλαίσια (frames):", num_frames)
        print("Κανάλια ήχου:", num_channels)
        print("Διάρκεια ήχου:", duration, "seconds")
        return (sample_rate, bit_depth, num_frames, num_channels, duration)
# -----Τέλος get_audio_info-----#

#####

#Βήμα 3ο απεικόνιση των διαγραμμάτων ήχου με άξονα ψ πλάτος, χ χρόνος #
#####

    @staticmethod
    def plot_audio(audio, classes, ax):
        signal, sr = audio
        classes=classes.lower()

```

```

librosa.display.waveshow(signal, sr=sr, ax=ax)
ax.set_title('Plot of sound amplitude at time domain for ' +classes) # Τίτλος γραφήματος
ax.set_xlabel("Time")
ax.set_ylabel("Amplitude")
# -----Τέλος plot_audio-----#

#####

#Βήμα 4ο μετατροπή του αρχείου ήχου από mono σε στέρεο #
#####

def rechannel_audio(audio,channels):
    signal, sr = audio
    if (signal.ndim == channels):
        # print("Δεν εγινε αλλαγη στα καναλια,καθως το αρχειο ηχου εχει καναλια: ",channels)
        return audio
#Προσθέστε μια δεύτερη διάσταση στο μονοκαναλικό ήχο για να δημιουργήσετε διπλοκαναλικό ήχο
elif (signal.ndim != channels and channels == 1):
    channels_audio = signal.ndim
    signal_mono = librosa.to_mono(signal)
    # print("Εγινε αλλαγή σε μονοκάναλο, καθώς το αρχείο ήχου έχει κανάλια: ",channels_audio)
    audio=signal_mono,sr
    return(audio)
elif (signal.ndim != channels and channels == 2):
    channels_audio = signal.ndim
    audio_stereo = np.vstack([signal, signal])
    # print("Εγινε αλλαγή σε διπλοκαναλο, καθώς το αρχείο ήχου έχει κανάλια: ",channels_audio)
    audio=audio_stereo,sr
    return(audio)
# -----Τέλος rechannel_audio-----#

#####

#Βήμα 5ο Κάνουμε τη δειγματοληψία ίση με την τιμή newsr (πχ.44100) #

```

```
#####
#####

@staticmethod
def resample(audio, newsr):
    signal, sr = audio
    if sr==newsr:
        return audio
    else:
        new_length = int(len(sig) * (newsr / sr))
        # Εφαρμογή αναδειγμένης δειγματοληψίας με τη χρήση της tf.signal.resample
        resig = tf.signal.resample(sig, new_length)
        return (resig, newsr)

# -----Τέλος resample-----#
```

```
#####
#####

#Βήμα 6ο Κάνουμε το χρόνο του αρχείου ήχου όσο επιθυμούμε προσθέτοντας σιγή (μηδενικά)#
#####
#####
```

```
@staticmethod
def add_silence(audio_file, target_duration):
    # Φόρτωση του ήχου
    y, sr = audio_file

    # Υπολογισμός της τρέχουσας διάρκειας του ήχου σε δευτερόλεπτα
    current_duration = librosa.get_duration(y=y, sr=sr)
    #print(current_duration)

    # Υπολογισμός της διαφοράς στο χρόνο (σε δευτερόλεπτα) που λείπει για να φτάσουμε στον
    στόχο
    silence_duration = target_duration - current_duration

    # Δημιουργία ενός πίνακα με σιγή (μηδενικά) για τη διάρκεια που λείπει
    silence = np.zeros(int(silence_duration * sr))
```

```

# Προσθήκη της σιγής στο τέλος του ήχου
y_extended = np.concatenate([y, silence])

# Επιστροφή του νέου ήχου και του sample rate
return y_extended, sr

# -----Τέλος pad_trunc-----#

#####
#####
#Βήμα 7ο κάνουμε μετατόπιση στο σήμα το ήχου για χρήση στην εκπαίδευση      #
#####
#####

@staticmethod
def time_shift(audio, shift_limit):
    sig, sr = audio
    _, sig_len = sig.shape
    shift_amt = int(random.random() * shift_limit * sig_len)
    shifted_sig = tf.roll(sig, shift_amt, axis=1)
    return shifted_sig, sr

# -----Τέλος time_shift-----#

#####
#####
#Βήμα 8ο Δημιουργία mel spectrogram      #
#####
#####

@staticmethod
def mel_spectro_gram(aud, n_mels=13, n_fft=2048, hop_len=512):
    sig, sr = aud
    # Compute spectrogram using librosa
    spec = librosa.feature.melspectrogram(y=sig, sr=sr, n_fft=n_fft, hop_length=hop_len,
n_mels=n_mels)
    # Convert to decibels using librosa

```

```

# spec_db = librosa.power_to_db(spec, ref=np.max)

# Convert to TensorFlow tensor

# spec_db = tf.convert_to_tensor(spec_db)

# return spec_db

return spec

# -----Τέλος mel_spectro_gram-----#

#####

#####

#Βήμα 9ο δημιουργία Mel Frequency Cepstral Coefficients #

#####

#####

@staticmethod

def mfcc_spegram(signal):

    signal, sr = signal

    mfccs =librosa.feature.mfcc(y=signal,n_fft=2048, n_mfcc=13 ,sr=sr, hop_length=512)

    return mfccs

# -----Τέλος mfcc_spegram-----#

#####

#####

#Βήμα 10ο Spectrogram απο σε db #

#####

#####

@staticmethod

def mfcc_spegram_db(mel_spectrogram,ref=np.max):

    mel_spectrogram_db = librosa.power_to_db(mel_spectrogram, ref=ref)

    mel_spectrogram_db = tf.convert_to_tensor(mel_spectrogram_db)

    return mel_spectrogram_db

#####

#####

#Βήμα 10ο Spectrogram αλλαγή θέσης στο πεδίο του χρόνου #

#####

#####

@staticmethod

```

```

def spec_augment(spec, max_freq_mask_pct=0.1, max_time_mask_pct=0.1, n_freq_masks=1,
n_time_masks=1):

    batch_size, num_mel_bins, num_time_steps = tf.shape(spec)[0], tf.shape(spec)[1],
tf.shape(spec)[2]

    # Frequency masking

    max_freq_mask_len = tf.maximum(1, tf.cast(max_freq_mask_pct * tf.cast(num_mel_bins,
tf.float32), tf.int32))

    for _ in range(n_freq_masks):

        mask_len = tf.random.uniform([], 0, max_freq_mask_len, dtype=tf.int32) # Ensure maxval is
a scalar

        mask_start = tf.random.uniform([], 0, num_mel_bins - mask_len, dtype=tf.int32)

        mask_len = mask_len.numpy() # Extract value as a Python scalar

        mask_start = mask_start.numpy() # Extract value as a Python scalar

        mask = tf.pad(tf.ones([mask_len, num_time_steps], dtype=tf.float32), [[mask_start,
num_mel_bins - mask_start - mask_len], [0, 0]])

        spec *= mask

    # Time masking

    max_time_mask_len = tf.maximum(1, tf.cast(max_time_mask_pct * tf.cast(num_time_steps,
tf.float32), tf.int32))

    for _ in range(n_time_masks):

        mask_len = tf.random.uniform([], 0, max_time_mask_len, dtype=tf.int32) # Ensure maxval is
a scalar

        mask_start = tf.random.uniform([], 0, num_time_steps - mask_len, dtype=tf.int32)

        mask_len = mask_len.numpy() # Extract value as a Python scalar

        mask_start = mask_start.numpy() # Extract value as a Python scalar

        mask = tf.pad(tf.ones([num_mel_bins, mask_len], dtype=tf.float32), [[0, 0], [mask_start,
num_time_steps - mask_start - mask_len]])

        spec *= mask

    return spec

# -----Τέλος pec_augment-----#

#####
#####

```

```

#Βήμα 11ο Καθαρισμός του σήματος ήχου με χρήση envelope #
#####
#####

@staticmethod
def envelope(signal, rate, threshold):
    mask = []
    y = pd.Series(signal).apply(np.abs)
    y_mean = y.rolling(window=int(rate/10), min_periods=1, center=True).mean()
    for mean in y_mean:
        if mean > threshold:
            mask.append(True)
        else:
            mask.append(False)
    return mask
# -----Τέλος envelope-----#

#####
#####

#Βήμα 12ο Plot το mel spectrogram #
#####
#####

@staticmethod
def plot_melspectrogram(spec, classes, ax):
    if spec.ndim == 3: #Ελέγχουμε αν ο πίνακας είναι τριών διαστάσεων
        spec = spec[0, :, :]
    spec = np.transpose(spec)
    ax.imshow(spec, aspect='auto', origin='lower', cmap='inferno')
    plt.colorbar(ax.imshow(spec, aspect='auto', origin='lower', cmap='inferno'), ax=ax,
format='%+2.0f')
    ax.set_title('Mel Spectrogram of ' + classes.lower())
    ax.set_xlabel('Frequency')
    ax.set_ylabel('Amplitude')
# -----Τέλος plot_melspectrogram-----#

```

```

#####
#####
#Βήμα 13ο Plot το Mel Frequency Cepstral Coefficients #
#####
#####

    @staticmethod
    def plot_mfcc_spegram(mfccs,classes,ax):
        if mfccs.ndim == 3: #Ελέγχουμε αν ο πίνακας είναι τριών διαστάσεων
            mfccs = mfccs[0, :, :]#παίρνουμε μόνο τις δύο διαστάσεις του πίνακα αγνοώντας τη πρώτη
            # Εμφανίζουμε το πίνακα MFCC ως εικόνα, ρυθμίζοντας τις παραμέτρους aspect, origin και
            colormap
            mfccs = np.transpose(mfccs)
            ax.imshow(mfccs, aspect='auto', origin='lower', cmap='inferno')
            # Προσθέτουμε μια γραμμή χρωματικής κλίμακας για να ερμηνεύσουμε τις τιμές του MFCC
            plt.colorbar(ax.imshow(mfccs, aspect='auto', origin='lower', cmap='inferno'), ax=ax,
            format='%+2.0f dB')
            ax.set_title('Mel logarithmic spectrogram of ' +classes.lower())# Τίτλος γραφήματος
            ax.set_xlabel('Frequency') # Ορισμός της ετικέτας του άξονα X
            ax.set_ylabel('db')# Ορισμός της ετικέτας του άξονα Y
# -----Τέλος plot_mfcc_spegram-----#

#####
#####
#Βήμα 14 Plot το envelope #
#####
#####

    @staticmethod
    def plot_envelope(signal,classes,ax):
        classes=classes.lower()# Κάνουμε τη μεταβλητή classes πεζά για την εμφάνιση
        librosa.display.waveshow(signal, sr=sr, ax=ax) # Εμφάνιση του σήματος στο πεδίο του χρόνου
        χρησιμοποιώντας την librosa
        ax.set_title('Clear sound Aplitude at time domain of ' +classes) # Τίτλος γραφήματος
        ax.set_xlabel("Time") # Ορισμός της ετικέτας του άξονα X
        ax.set_ylabel("Amplitude")# Ορισμός της ετικέτας του άξονα Y
# -----Τέλος plot_envelope-----#

```

```

# Ανοίγουμε μια σειριακή σύνδεση με την Arduino
ser = serial.Serial(port='COM3', baudrate=9600, timeout=.1)

# Επιλέξτε το μοντέλο που έχετε αποθηκευμένο
model_path = "Sound_CNN2"
sound_path = "record_sound/output.wav"
filtered_sound_path="record_sound/clear_output.wav"

# Φορτώστε το μοντέλο
model = load_model('C:\\Users\\blabla\\Sound_CNN2')

# Ανιχνεύστε το πάτημα του πλήκτρου "space" για να ξεκινήσει η ηχογράφηση
print("Press space to start recording...")

keyboard.wait("a")

# Ορίστε τις παραμέτρους της καταγραφής ήχου
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = int(RATE / 10) # 100ms buffer
RECORD_SECONDS = 2
WAVE_OUTPUT_FILENAME = "record_sound/output.wav"
input_device_index = 2

# Ξεκινήστε την καταγραφή ήχου από το μικρόφωνο
ser.write(b'a')

time.sleep(0.5)

audio = pyaudio.PyAudio()

stream = audio.open(format=FORMAT, channels=CHANNELS,
                    rate=RATE, input=True,
                    input_device_index=input_device_index,
                    frames_per_buffer=CHUNK)

```

```

# Στέλνουμε το 'a' στην Arduino για να ενεργοποιήσει το servo

print("Recording...")

frames = []

# Χρησιμοποιήστε το tqdm για να εμφανίσετε μια μπάρα προόδου
for i in tqdm(range(0, int(RATE / CHUNK * RECORD_SECONDS)), desc="Recording",
unit="chunks"):
    data = stream.read(CHUNK)
    frames.append(data)

print("Finished recording.")

stream.stop_stream()
stream.close()
audio.terminate()

# Αποθηκεύστε την καταγραφή ήχου σε ένα αρχείο WAV
waveFile = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
waveFile.setnchannels(CHANNELS)
waveFile.setsampwidth(audio.get_sample_size(FORMAT))
waveFile.setframerate(RATE)
waveFile.writeframes(b''.join(frames))
waveFile.close()

# Φορτώστε τον ήχο από το αρχείο WAV που καταγράψατε
signal,sr = librosa.load(sound_path, sr=44100)
envelope = AudioActivities.envelope(signal, sr, 0.0002)
filtered_signal = signal[envelope]
sf.write(filtered_sound_path, filtered_signal, sr)
signal= librosa.load(filtered_sound_path, sr=44100)
target_duration = 2.5

```

```

signal = AudioActivities.add_silence(signal, target_duration)
# Εξαγωγή χαρακτηριστικών MFCC
mfccs = AudioActivities.mfcc_spegram(signal)
print(mfccs.shape[0])
mfccs= mfccs.reshape(1,mfccs.shape[0],mfccs.shape[1],1)
mfccs = np.transpose(mfccs)
# Κάντε πρόβλεψη χρησιμοποιώντας το μοντέλο
prediction = model.predict(mfccs)
print(prediction)
# Εκτυπώστε την πρόβλεψη
predicted_class = np.argmax(prediction)
Labels=['Aluminium','Glass','Paper','PLASTIC','PLASTIC']
predicted_class_index = tf.argmax(prediction, axis=1)[0].numpy() # Get the scalar value
predicted_class_label = Labels[predicted_class_index]
predicted_accuracy = prediction[0][predicted_class_index] * 100
print("Predicted class:", predicted_class_label)
print("Accuerancy:", predicted_accuracy)
print("Predicted class index:", predicted_class)
ser.close()

```

Πρόβλεψη από ταξινομητή σε εικόνα.

```

# Ορίζουμε τον φάκελο όπου θα αποθηκεύσουμε τις φωτογραφίες
output_folder = 'C:/Users/Xristoforos/blabla/'
# Δημιουργούμε τον φάκελο αν δεν υπάρχει
os.makedirs(output_folder, exist_ok=True)
# Αριθμός της κάμερας USB (μπορεί να διαφέρει ανάλογα με τη συνδεδεμένη κάμερα)
camera_number = 0
# Δημιουργούμε ένα αντικείμενο VideoCapture για την κάμερα USB
def take_photo():
    cap = cv2.VideoCapture(camera_number)
    ret, frame = cap.read()

```

```

print (ret)
filename = os.path.join(output_folder, 'input_photo.jpg')
cv2.imwrite(filename, frame)
print(f"Photo saved as {filename}")
cap.release()

while True:
    keyboard.wait('s')
    take_photo()
    break
#-----#
#-----Resize to 224x224x3-----#
#-----#
def resize_image_with_padding(input_image_path, output_image_path, target_size):
    # Load the image using cv2
    image = cv2.imread(input_image_path)

    # Get the dimensions of the original image
    original_size = image.shape[:2]

    # Compute the resizing ratio
    ratio = min(target_size[0] / original_size[0], target_size[1] / original_size[1])

    # Compute the new dimensions
    new_dimensions = (int(original_size[0] * ratio), int(original_size[1] * ratio))

    # Resize the image
    resized_image = cv2.resize(image, new_dimensions, interpolation=cv2.INTER_CUBIC)

    # Create a white background canvas with the target size
    canvas = np.ones((target_size[1], target_size[0], 3), dtype=np.uint8) * 255

    # Compute the coordinates to paste the resized image onto the canvas

```

```

x_offset = (target_size[0] - new_dimensions[0]) // 2
y_offset = (target_size[1] - new_dimensions[1]) // 2

# Paste the resized image onto the canvas
canvas[y_offset:y_offset + new_dimensions[1], x_offset:x_offset + new_dimensions[0]] =
resized_image

# Save the new image
cv2.imwrite(output_image_path, canvas)

resize_image_with_padding("C:\\Users\\blabla\\input_photo.jpg",
"C:\\Users\\blabla\\input_photo.jpg", (224, 224))
#-----End of resizing-----#

#-----#
#-----Remove Background-----#
#-----#

# Διαβάζουμε την εικόνα χρησιμοποιώντας την OpenCV
img = cv2.imread("C:\\Users\\blabla\\input_photo.jpg", cv2.IMREAD_UNCHANGED)

# Καλούμε τη συνάρτηση remove() της rembg για την αφαίρεση του φόντου
R = remove(img)

# Αποθηκεύουμε την αφαιρεμένη εικόνα
cv2.imwrite("C:\\Users\\blabla\\output_photo.png", R)

# Εφαρμόζουμε τις απαραίτητες μετατροπές για να μετατρέψουμε το PNG σε JPEG
background_color = (255, 255, 255)

# Δημιουργούμε μια νέα εικόνα με το επιθυμητό χρώμα φόντου
new_img = np.full_like(R, background_color + (255,), dtype=np.uint8)

# Αντιγράφουμε τον κόκκινο, πράσινο και μπλε κανάλι από την αφαιρεμένη εικόνα στη νέα

```

```

for c in range(0, 3):
    new_img[:, :, c] = new_img[:, :, c] * (1 - R[:, :, 3] / 255) + R[:, :, c] * (R[:, :, 3] / 255)

# Αποθηκεύουμε την εικόνα σε μορφή JPEG
cv2.imwrite("C:\\Users\\blabla\\output_photo.jpg", new_img, [int(cv2.IMWRITE_JPEG_QUALITY),
100])

# Διαγράφουμε τα αρχεία που δημιουργήθηκαν
os.remove("C:\\Users\\blabla\\output_photo.png")

#-----End of removing Background-----#

#-----#
#-----Canny Filter-----#
#-----#

image = cv2.imread('C:/Users/blabla/output_photo.jpg', cv2.IMREAD_GRAYSCALE)
    # Εφαρμογή αλγορίθμου Canny για την ανίχνευση ακμών
edges = cv2.Canny(image, 10, 150)
    # Δημιουργία νέου ονόματος αρχείου
new_filename = 'C:/Users/blabla/output_photo.jpg'
print(new_filename )
    # Αποθήκευση της νέας εικόνας
cv2.imwrite(new_filename, cv2.Canny(image, 100, 200))

#-----END OF CANNY-----#
image_path = 'C:/Users/blabla/output_photo.jpg'
from keras.models import load_model
#loaded_model = load_model('C:\\Users\\blabla\\ResNet2_2024-03-03_10-38-18')
def predict_and_visualize(image_path):
    image = tf.keras.preprocessing.image.load_img(image_path, target_size=(224, 224,3))
    image_array = tf.keras.preprocessing.image.img_to_array(image)
    image_array = image_array / 255.0 # Normalize pixel values

```

```

input_data = tf.expand_dims(image_array, axis=0)
predictions = loaded_model.predict(input_data)
print(predictions)
class_labels = ['ALUMINIUM', 'GLASS', 'PAPER', 'PLASTIC', 'PLASTIC']
predicted_class_index = tf.argmax(predictions, axis=1)[0].numpy() # Get the scalar value
print(predicted_class_index)
predicted_class_label = class_labels[predicted_class_index]
predicted_accuracy = predictions[0][predicted_class_index] * 100

plt.imshow(image)
plt.title(f'Predicted: {predicted_class_label}\nAccuracy: {predicted_accuracy:.2f}%')
plt.gca().add_patch(plt.Rectangle((0, 0), 250, 300, fill=False, color='red', linewidth=2))
plt.show()

predict_and_visualize(image_path)

```