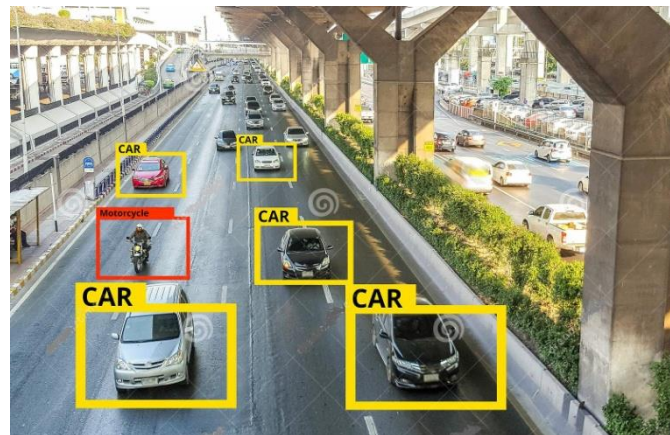


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

# ΑΝΑΓΝΩΡΙΣΗ ΕΙΚΟΝΑΣ ΜΕΣΩ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΓΙΑ ΡΟΜΠΟΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ



ΦΟΙΤΗΤΗΣ: ΤΑΣΙΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ (Α.Μ.: 7473)  
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΑΡΑΒΑΝΗΣ ΘΕΟΦΑΝΗΣ

ΠΑΤΡΑ 2023

## ΠΡΟΛΟΓΟΣ

Η συγκεκριμένη εργασία αποτελεί την Διπλωματική Εργασία που εκπονήθηκε στο Τμήμα Μηχανολόγων Μηχανικών της σχολής Μηχανικών του Πανεπιστημίου Πελοποννήσου και επεξηγεί ένα σύστημα αναγνώρισης εικόνας με την βοήθεια ενός Τεχνητού Νευρωνικού Δικτύου, το οποίο αποτελεί αντικείμενο του μεγάλου τομέα της Τεχνητής Νοημοσύνης. Τα τελευταία χρόνια, η επιστήμη της Τεχνητής Νοημοσύνης, η οποία πολλές φορές επικουρείται από την επιστήμη της Ρομποτικής, εξελίσσεται όλο και περισσότερο, με σκοπό την διευκόλυνση αλλά και την αυτοματοποίηση της καθημερινότητας μας.

Αρχικά στην εργασία παρουσιάζονται γενικές εφαρμογές και η χρησιμότητα της Τεχνητής Νοημοσύνης. Μετέπειτα αναλύονται ορολογίες και μαθηματικές σχέσεις της Βαθείας Μάθησης, δηλαδή ενός τυπικού συστήματος Τεχνικού Νευρωνικού Δικτύου, καθώς και η τεχνική επίτευξης της αναγνώρισης εικόνας. Ο πυρήνας της εργασίας όμως βρίσκεται στο τρίτο Κεφάλαιο, όπου γίνεται πλήρης ανάλυση ενός κώδικα Python, ο οποίος αποτελεί το μοντέλο εκμάθησης ενός drone με Τεχνητή Νοημοσύνη. Μετά την εφαρμογή του κώδικα, το drone θα περιπολεί δάση ή φυσικά μέρη και θα ανιχνεύει-αναγνωρίζει τυχόν πυρκαγιές με ποσοστό επιτυχίας στην αναγνώριση (accuracy) 97,06%. Τότε το drone θα ειδοποιεί άμεσα την πυροσβεστική, με αποτέλεσμα την αποφυγή εξάπλωσης της πυρκαγιάς σε μεγαλύτερη έκταση και την πρόληψη μη ελεγχόμενης κατάστασης της.

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου κ. Θεοφάνη Αραβανή, Ακαδημαϊκό Υπότροφο του Τμήματος Μηχανολόγων Μηχανικών, για την βοήθεια του και τον χρόνο που αφιέρωσε για την επιτυχία της Εργασίας.

Τασιόπουλος Δημήτριος  
Φεβρουάριος 2023

**Υπεύθυνη Δήλωση Φοιτητή:** Ο κάτωθι υπογεγραμμένος Φοιτητής έχω επίγνωση των συνεπειών του Νόμου περί λογοκλοπής και δηλώνω υπεύθυνα ότι είμαι συγγραφέας αυτής της Διπλωματικής Εργασίας, έχω δε αναφέρει στην Βιβλιογραφία μου όλες τις πηγές τις οποίες χρησιμοποίησα και έλαβα ιδέες ή

δεδομένα. Δηλώνω επίσης ότι, οποιοδήποτε στοιχείο ή κείμενο το οποίο έχω ενσωματώσει στην εργασία μου προερχόμενο από Βιβλία ή άλλες εργασίες ή το διαδίκτυο, γραμμένο ακριβώς ή παραφρασμένο, το έχω πλήρως αναγνωρίσει ως πνευματικό έργο άλλου συγγραφέα και έχω αναφέρει ανελλιπώς το όνομά του και την πηγή προέλευσης.

Ο Φοιτητής  
(Ονοματεπώνυμο)  
ΤΑΣΙΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ  
(Υπογραφή)



## ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία χωρίζεται σε δύο μέρη. Στο πρώτο μέρος αναπτύσσονται γενικές έννοιες και εφαρμογές της τεχνητής νοημοσύνης και των τεχνητών νευρωνικών δικτύων, που προορίζονται για την επιστήμη της ρομποτικής. Παράλληλα στο δεύτερο μέρος επιτυγχάνεται η ανάπτυξη ενός προγράμματος, που εφαρμόζει στην πράξη τα θεωρητικά αποσπάσματα του πρώτου μέρους. Έχει σκοπό να μεταδώσει στον αναγνώστη το κυρίως θέμα των τεχνητών νευρωνικών δικτύων, δηλαδή την δομή τους, το πώς επεξεργάζονται τα δεδομένα, την εκπαίδευσή τους, αλλά και τι προσφέρουν στην επιστήμη και στην κοινωνία.

Το κυρίως θέμα του πρώτου κεφαλαίου είναι τα τεχνητά νευρωνικά δίκτυα. Αρχικά παρουσιάζεται ο ορισμός των τεχνητών νευρωνικών δικτύων, που βασίζονται και τι στόχο έχουν. Στην συνέχεια αναφέρονται τα ιστορικά στοιχεία τους. Πιο συγκεκριμένα, αναφέρονται οι χρονολογίες και τα ονόματα των επιστημόνων που εφηύραν και εξέλιξαν το αντικείμενο των τεχνητών νευρωνικών δικτύων, καθώς και τις επιτυχίες του καθενός. Ακολουθεί πλήρης επεξήγηση του μοντέλου ενός τεχνητού νευρώνα, αλλά και του μοντέλου ενός τεχνητού νευρωνικού δικτύου. Δηλαδή η δομή τους, πώς επεξεργάζονται τα δεδομένα που τροφοδοτούνται, η διαδικασία μεταφοράς του σήματος μέσα στο δίκτυο, αλλά και η διαδικασία εξόδου των τιμών του δικτύου, μέσω των συναρτήσεων μεταφοράς. Ακόμη, παρουσιάζονται οι τρόποι με τους οποίους το δίκτυο αυτό εκπαιδεύεται έτσι ώστε να ληφθεί η επιθυμητή τιμή εξόδου. Τέλος γίνεται μια μικρή αναφορά στα βιολογικά νευρωνικά δίκτυα και στις δυνατότητές τους, όπου στη συνέχεια συγκρίνονται με τα τεχνητά.

Το δεύτερο κεφάλαιο, βασίζεται στην σύνδεση της τεχνητής νοημοσύνης με την επιστήμη της ρομποτικής. Αρχικά γίνεται μία εισαγωγή στο αντικείμενο της τεχνητής νοημοσύνης, τι φιλοδοξίες έχει, αλλά και των συνεπειών της. Στην συνέχεια δίνονται παραδείγματα από τα βασικά εξαρτήματα που χρειάζονται για κατασκευαστεί και να λειτουργήσει επιτυχώς, ένα μοντέλο ρομπότ με τεχνητή νοημοσύνη. Ακολουθεί η επεξήγηση, πώς η μηχανική μάθηση αποτελεί την αιτία που οι άνθρωποι και τα ρομπότ έχουν αποκτήσει τρόπο επικοινωνίας και αναφέρονται ορισμένες πληροφορίες για τις γλώσσες προγραμματισμού που χρησιμοποιούνται για ρομποτικές εφαρμογές. Επίσης παρουσιάζεται ο ορισμός της μηχανικής όρασης και τα στάδια των διαδικασιών που εφαρμόζονται έτσι ώστε να επιτευχθεί η αναγνώριση της εικόνας που επιθυμεί ο χειριστής του ρομπότ. Τέλος αναφέρονται παραδείγματα από χώρους ή επιστήμες, που λαμβάνουν θέση ρομπότ με τεχνητή νοημοσύνη.

Το τρίτο και τελευταίο κεφάλαιο παρουσιάζει την γλώσσα προγραμματισμού Python. Πρώτα από όλα, δίνονται πληροφορίες για την γλώσσα αυτή, δηλαδή ποιος

την ίδρυσε και πότε, ποιες είναι οι αιτίες που βρίσκεται στην κορυφή των υπόλοιπων γλωσσών προγραμματισμού, αλλά και πως επιτυγχάνεται η γραφή της. Ύστερα δίνεται ένα παράδειγμα κώδικα προγραμματισμού, μαζί με το αντίστοιχο διάγραμμα ροής, που αναλύει την διαδικασία εκμάθησης ενός μοντέλου ρομπότ, με την μορφή ενός drone, να αντιλαμβάνεται την ύπαρξη πυρκαγιάς στο φυσικό περιβάλλον. Εάν το drone εντοπίσει πυρκαγιά, τότε θα υπολογίζει τις ακριβείς συντεταγμένες της με την χρήση ενός GPS και θα ειδοποιεί την πυροσβεστική. Ταυτόχρονα με τον κώδικα, δίνονται επεξηγήσεις του κάθε σταδίου εφαρμογής, αλλά και ορισμένων εντολών.

Μετά την εφαρμογή του προγράμματος, προκύπτουν τα συμπεράσματα και η αξιολόγηση του, ενώ στο τελευταίο μέρος της εργασίας γίνεται ανακεφαλαίωση της. Πιο συγκεκριμένα, εμφανίζονται τα ποσοστά επιτυχίας αναγνώρισης της φωτιάς κατά την εκτέλεση του προγράμματος εκμάθησης, όταν θα υπάρξει κυριολεκτικά φωτιά, δηλαδή 97,06% επιτυχία αναγνώρισης, με το μοντέλο να είναι έτοιμο για χρήση.

# ΠΕΡΙΕΧΟΜΕΝΑ

**ΠΡΟΛΟΓΟΣ**

**ΠΕΡΙΛΗΨΗ**

**ΠΕΡΙΕΧΟΜΕΝΑ**

**ΕΙΣΑΓΩΓΗ**

1. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ .....	1
2. ΣΥΓΧΡΟΝΕΣ ΕΦΑΡΜΟΓΕΣ .....	2
3. ΟΠΤΙΚΟ ΣΥΣΤΗΜΑ .....	3

**1. ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**

1.1 ΤΙ ΕΙΝΑΙ ΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ; .....	5
1.1.1 Ορισμός .....	5
1.1.2 Ιστορικά στοιχεία .....	5
1.1.3 Μοντέλο νευρωνικών δικτύων .....	7
1.2 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ .....	9
1.2.1 Βηματική Συνάρτηση .....	9
1.2.2 Σιγμοειδής Συνάρτηση .....	10
1.2.3 Υπερβολική Εφαπτομένη Συνάρτηση Ενεργοποίησης .....	10
1.3 ΕΚΠΑΙΔΕΥΣΗ ΤΕΧΝΗΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ .....	11
1.4 ΣΤΟΙΧΕΙΑ ΒΙΟΛΟΓΙΑΣ .....	12
1.4.1 Βιολογικός Νευρώνας .....	12
1.4.2 Σύγκριση με τεχνητά νευρωνικά δίκτυα .....	13

**2. ΧΡΗΣΗ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ**

2.1 ΓΕΝΙΚΑ .....	15
2.2 ΣΥΝΕΠΕΙΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ .....	16
2.3 HARDWARE ΓΙΑ ΡΟΜΠΟΤ ΜΕ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ .....	17
2.3.1 Επεξεργαστές .....	17
2.3.2 Συσσκευές Εισόδου .....	20

2.4 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ .....	21
2.5 ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ ΚΑΙ ΑΝΑΓΝΩΡΗΣΗ ΕΙΚΟΝΑΣ .....	22
2.5.1 Αρχή Λειτουργίας .....	22
2.5.2 Χρησιμότητα Μηχανικής Όρασης .....	23
2.6 ΠΕΡΙΣΣΟΤΕΡΕΣ ΕΦΑΡΜΟΓΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΣΤΗ ΡΟΜΠΟΤΙΚΗ .....	23
2.6.1 Στην Ιατρική.....	23
2.6.2 Στη Γεωργία .....	24
2.6.3 Στην Αυτοκινητοβιομηχανία.....	24
<b>3. ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ</b>	
3.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΡΥΤΗΘΝ .....	27
3.2 ΠΑΡΑΔΕΙΓΜΑ ΚΩΔΙΚΑ ΡΥΤΗΘΝ ΓΙΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....	28
3.2.1 Διάγραμμα Ροής.....	28
3.2.2 Μοντέλο εκμάθησης ρομπότ αναγνώρισης εικόνας με πρόγραμμα Python.....	29
3.2.3 Εκτέλεση προγράμματος του μοντέλου εκμάθησης ρομπότ αναγνώρισης εικόνας και λήψη συμπερασμάτων.....	34

#### **4. ΣΥΜΠΕΡΑΣΜΑΤΑ**

#### **ΒΙΒΛΙΟΓΡΑΦΙΑ**

## ΣΥΜΒΟΛΙΣΜΟΙ

$\theta$  = τιμή κατωφλιού  
 $u$  = έξοδος αθροιστή  
 $f$  = συνάρτηση ενεργοποίησης  
 $S$  = αθροιστής  
 $w_i$  = συναπτικά βάρη νευρώνων  
 $x_i$  = σήμα εισόδου  
 $y_i$  = σήμα εξόδου νευρώνων

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΤΝΔ = Τεχνητό Νευρωνικό Δίκτυο  
ΝΔ = Νευρωνικό Δίκτυο





# ΕΙΣΑΓΩΓΗ

## 1. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Ο ανθρώπινος εγκέφαλος επειδή είναι ανακριβής και πιο αργός σε σχέση με κάποιον ηλεκτρονικό υπολογιστή, τότε για να μειωθεί η πιθανότητα σοβαρών αποκλίσεων (σφαλμάτων) στα αποτελέσματα της καθημερινότητας αλλά και στους χώρους εργασίας, η επιστήμη κατέληξε στην δημιουργία της *Τεχνητής νοημοσύνης (Artificial Intelligence)*. Ανήκει στον ευρύτερο κλάδο της *Ρομποτικής*. Ρομπότ είναι μία μηχανή που διενεργεί ανθρώπινα καθήκοντα με την χρήση προγραμματισμένων εντολών από ανθρώπους, με στόχο την προσέγγιση της ανθρώπινης συμπεριφοράς και την υποκατάσταση της ανθρώπινης παρουσίας.

Η Τεχνητή νοημοσύνη βασίζεται στην εκπαίδευση των υπολογιστών να λειτουργήσουν έχοντας σαν πρότυπο την ανθρώπινη νοημοσύνη, δηλαδή ένας υπολογιστής να αποκτάει ανθρώπινες αισθήσεις και αντιλήψεις όπως όραση, ακοή, λήψη αποφάσεων κλπ. Η Τεχνητή νοημοσύνη χωρίζεται σε ορισμένες υποκατηγορίες. Στην περίπτωση της συγκεκριμένης εργασίας μας ενδιαφέρει η υποκατηγορία της Μηχανικής μάθησης. Η Μηχανική μάθηση έχει το τέχνασμα να αυτοματοποιεί τις καταστάσεις χωρίς να παρέμβει κάποια ανθρώπινη βοήθεια. Μια μορφή Μηχανικής μάθησης ονομάζεται Βαθιά μάθηση (*Deep Learning*), η οποία ουσιαστικά παραπέμπει σε Τεχνητά Νευρωνικά Δίκτυα. Η Βαθιά μάθηση αναλύει με μεγάλη λεπτομέρεια τα δεδομένα, έτσι ώστε να υπάρξει αποδοτικότερη λύση των προβλημάτων, με μεγαλύτερη ακρίβεια σε σχέση με τις περισσότερες τεχνικές της Μηχανικής μάθησης .

Η ανθρώπινη νοημοσύνη αποτελείται από φυσικά χαρίσματα όπως η φαντασία, η κοινή λογική, η κρίση του κάθε ανθρώπου, η εμπειρία που αποκτά και η ικανότητα να χειρίζεται και να επεξεργάζεται δεδομένα και στοιχεία διαφορετικών καταστάσεων. Έτσι λοιπόν και η Βαθιά μάθηση, με την βοήθεια των υπολογιστών και των αλγορίθμων της, προσεγγίζει όσο το δυνατόν περισσότερο την ανθρώπινη νοημοσύνη και έρχεται αντιμέτωπη με τον μεγάλο όγκο δεδομένων που έχει να επεξεργαστεί. Επιπλέον οι αλγόριθμοι Βαθιάς μάθησης είναι στην ακμή τους, λόγω της δυναμικότητας που παρέχεται πλέον από τους σύγχρονους υπολογιστές, κάτι που τα προηγούμενα χρόνια ήταν αδύνατον να συμβεί. Αυτό συνέβη, επειδή οι υπολογιστές παλαιότερης τεχνολογίας δεν ανταποκρίνονταν στις προσδοκίες που απαιτούνταν για το επιθυμητό αποτέλεσμα.

## 2. ΣΥΓΧΡΟΝΕΣ ΕΦΑΡΜΟΓΕΣ

Οι εφαρμογές της Βαθιάς μάθησης, δηλαδή εφαρμογές νευρωνικών δικτύων, έχουν συμβάλει στην εύρεση λύσεων καθώς και στην λήψη αποφάσεων στους περισσότερους επιστημονικούς κλάδους. Στο “σήμερα” το πεδίο αυτό βρίσκεται σε πρώιμο στάδιο, γεγονός που τις επόμενες δεκαετίες θα συνέβη μια κατακόρυφη εξέλιξη στις εφαρμογές αυτές. Αυτό οφείλεται στην ανακάλυψη νέων μηχανισμών, νέων υλικών πόρων αλλά και νέων τεχνολογιών, δηλαδή στην ραγδαία ανάκαμψη της τεχνολογίας. Στην συνέχεια θα αναφερθούν ορισμένα από τα πολυάριθμα πεδία εφαρμογής των Νευρωνικών Δικτύων αλλά και αντίστοιχα παραδείγματα.

Πρώτο πεδίο εφαρμογής είναι τα διασκεδαστικά σε όλους μας Ηλεκτρονικά παιχνίδια. Υπάρχουν παιχνίδια σε ηλεκτρονικές συσκευές-κονσόλες που ο εικονικός χαρακτήρας μας χρειάζεται έναν αντίπαλο και συνήθως για λόγους ευκολίας και λιγότερης αναμονής, διαλέγουμε να παίξουμε εναντίον του ίδιου του υπολογιστή. Ο υπολογιστής λοιπόν είναι “εκπαιδευμένος” και εξετάζει όλες τις πιθανές απαντήσεις που διαθέτει, έτσι ώστε να επιλέξει την ιδανικότερη πιθανή απάντηση. Η αρχή λειτουργίας του είναι τόσο καλά προγραμματισμένη που μπορεί να κοντράρει τις ικανότητες του ανθρώπινου νου.

Δεύτερο πεδίο εφαρμογής βρίσκεται στον χώρο του Στρατού και συγκεκριμένα στα γνωστά “radar”. Είναι μία συσκευή η οποία ανιχνεύει στόχους είτε ακίνητους, είτε εν κινήσει και μετράει την απόσταση από αυτόν σε συνθήκες φωτισμού που η ανθρώπινη όραση είναι ανίκανη να εντοπίσει. Η διαδικασία της επιστροφής του σήματος, δίνει αποτελέσματα με μεγάλη ακρίβεια.

Η επιστήμη της Ιατρικής είναι ακόμη ένα πεδίο που εφαρμόζεται η Βαθιά μάθηση. Για παράδειγμα, στις ακτινογραφίες διακρίνουμε τα οστά μας με την μορφή σκαναρίσματος εξ’ αιτίας της ακτινοβολίας Χ, η οποία βρίσκει μεγαλύτερη αντίσταση στα περισσότερα συμπαγή μόρια, δηλαδή τα οστά και με την βοήθεια ενός υπολογιστή τα αποθηκεύει και τα αποτυπώνει στο φιλμ. Έτσι ο γιατρός καταλήγει ορθά στην διάγνωση του προβλήματος υγείας του ασθενή, αντιμετωπίζοντάς το με την κατάλληλη αγωγή.

Στις επιστήμες τεχνολογικών εφαρμογών και τηλεπικοινωνιών, η Βαθιά μάθηση κυριαρχεί και έχει αναδείξει τις δυνατότητές της με μεγάλη επιτυχία. Στη συγκεκριμένη περίπτωση έχουμε αυτόματη μετάφραση κειμένων οποιασδήποτε γλώσσας ή διαλέκτου, αναγνώριση φωνής, πλατφόρμες αναζήτησης, ανάλυση ομιλίας σε ακουστικά βαρηκοΐας, οχήματα χωρίς την παρουσία ανθρώπου στον ρόλο του οδηγού και η μηχανική όραση, που είναι το κύριο θέμα αυτής της εργασίας. Η μηχανική όραση έχει την δυνατότητα να αναγνωρίζει και να διακρίνει αντικείμενα, πρόσωπα, πρότυπα και καταστάσεις με μέσα εισόδου (κάμερα ή αισθητήρας). Ένα πρόβλημα που υπάρχει προς το παρόν είναι ότι τα περισσότερα διαθέσιμα προγράμματα λειτουργούν με τρόπο, ώστε το αποτέλεσμα να διατυπωθεί σε δύο διαστάσεις, ενώ σε ορισμένες περιπτώσεις όρασης υπολογιστών είναι επιθυμητή η λήψη τρισδιάστατων πληροφοριών.

Οι παραπάνω εφαρμογές είναι ένα πολύ μικρό ποσοστό στον συνολικό όγκο των πεδίων εφαρμογών της Βαθιάς μάθησης. Μέσα σε μόλις λίγα χρόνια, χρειάστηκαν να γίνουν πολλά πειράματα και μελέτες καθώς και μεγάλη αφοσίωση στο έργο από την συνεργασία μηχανολόγων και ηλεκτρολόγων μηχανικών, προγραμματιστών και μαθηματικών έτσι ώστε η τεχνολογία να καταφέρει να εξελίξει και να ανεβάσει το επίπεδο της Τεχνητής Νοημοσύνης σε σημείο που ούτε ο ίδιος ο άνθρωπος θα φανταζόταν.

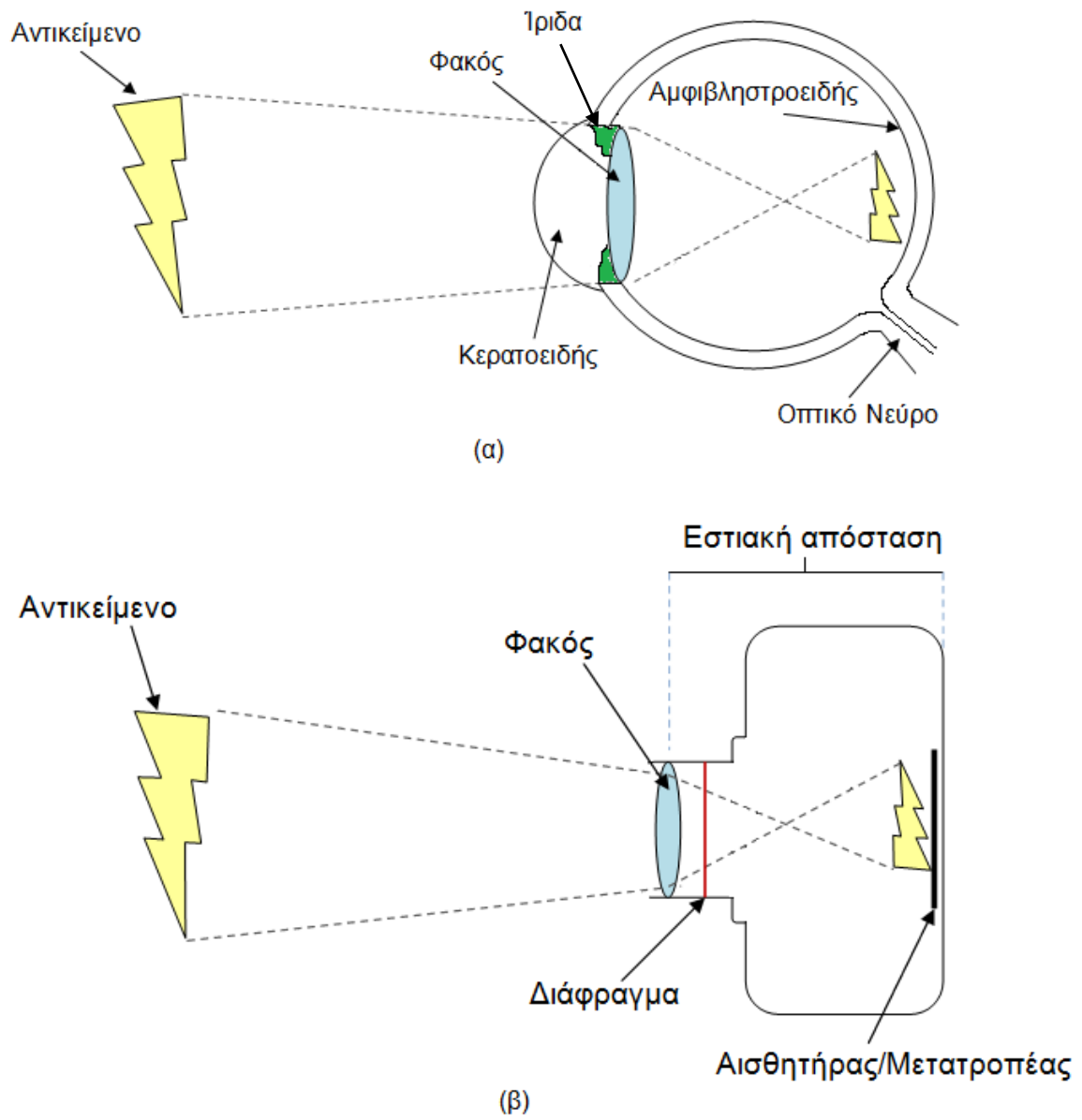
### 3. ΟΠΤΙΚΟ ΣΥΣΤΗΜΑ

Το νόημα της εργασίας αυτής, βασίζεται σε μία από τις πέντε αισθήσεις του ανθρώπου, αυτήν της όρασης. Η *όραση* είναι το μέσο αλληλεπίδρασης και επικοινωνίας του ανθρώπου με τον χώρο και τα αντικείμενα που τον περιβάλλουν. Η αίσθηση αυτή επιτυγχάνεται και βρίσκεται σε ισχύ με την βοήθεια του οπτικού συστήματος που το πιο σημαντικό όργανο στην λειτουργία του, είναι τα μάτια. Τα μάτια θεωρούνται η είσοδος του συστήματος, τα οποία αποτυπώνουν την εικόνα που δέχονται και καταλήγει στα μέρη του εγκεφάλου, όπου και επεξεργάζεται. Το παραπάνω σύστημα λοιπόν, είναι το πρότυπο που ακολουθούν οι επιστήμονες για την δημιουργία του οπτικού συστήματος ενός ρομπότ, αντικαθιστώντας τα βιολογικά χαρακτηριστικά του με νέα τεχνητά, όπως για παράδειγμα, μία κάμερα παίρνει τον ρόλο των ματιών στην μηχανική όραση, όπως αναλύεται παρακάτω και φαίνεται στο Σχ. 1.

Αρχικά, χρειάζεται ένας *φακός* (είτε γυάλινου υλικού, είτε διαφανής πλαστικού), του οποίου το καθήκον είναι να συλλέγει από το πεδίο παρατήρησης τις προσπίπτουσες *δέσμες φωτός*, με βασική προϋπόθεση την ύπαρξη του κατάλληλου φυσικού ή τεχνητού φωτισμού. Το *διάφραγμα* αποτελεί τον ελεγκτή της ποσότητας του φωτός που διαπερνάται από τον φακό για τις ανάγκες της φωτο-μετατροπής. Αντίστοιχη λειτουργία έχει και το ανθρώπινο μάτι, με την διαφορά ότι τον ρόλο του διαφράγματος τον εκτελεί η *ίριδα*. Η ίριδα είναι ένας λείος μυς που έχει την δυνατότητα να συστέλλεται ή να διαστέλλεται, ανάλογα με την ένταση του φωτισμού.

Σε επόμενο στάδιο, οι συλλεγμένες δέσμες φωτός μετατρέπονται σε σήματα, όπου επεξεργάζονται και αποτυπώνονται σε μια φωτο-ευαίσθητη επιφάνεια. Η διαδικασία αυτή ονομάζεται *φωτο-μετατροπή*. Φωτο-ευαίσθητη επιφάνεια θεωρείται ένα φιλμ ή ένας αισθητήρας για την περίπτωση της κάμερας και ο *αμφιβληστροειδής χιτώνας* για την περίπτωση του ματιού. Η απόσταση ανάμεσα στον φακό και στον μετατροπέα ονομάζεται *εστιακό μήκος*. Σε αυτήν την απόσταση γίνεται η διαδικασία της φωτο-μετατροπής. Στον αμφιβληστροειδή, καθώς και στο φιλμ οι εικόνες προβάλλονται ανάποδα, δηλαδή αντιστρέφονται.

Ο αμφιβληστροειδής περιλαμβάνει δύο είδη υποδοχέων, τα *ραβδία* και τα *κωνία*. Αυτοί οι δύο υποδοχείς βρίσκονται σε ένα από τα πολλά νευρικά στρώματα του αμφιβληστροειδή και συνδέονται με τις απολήξεις του οπτικού νεύρου. Τα ραβδία είναι ευαίσθητα στην ένταση του φωτός και ο αριθμός τους είναι πολύ μεγαλύτερος από τον αριθμό των κωνίων, τα οποία έχουν ευαισθησία και στα χρώματα. Αυτό έχει σαν αποτέλεσμα, όταν ο φωτισμός ελαττώνεται, το μάτι να αδυνατεί να ξεχωρίσει χρώματα και να βλέπει όλο το περιβάλλον γύρω του γκρι. Ο αμφιβληστροειδής μετατρέπει τις εικόνες σε ηλεκτρικά σήματα, που ξεκινάνε να μεταφέρονται από τα ραβδία και τα κωνία και να καταλήγουν στον εγκέφαλο με την βοήθεια των *οπτικών νεύρων*. Το οπτικό νεύρο διαθέτει ένα εκατομμύριο οπτικές ίνες. Τέλος το μέρος του εγκεφάλου που καταλήγουν τα σήματα ονομάζεται *ινιακός λοβός*, βρίσκεται στο πίσω μέρος του εγκεφάλου και επεξεργάζεται τα οπτικά δεδομένα.



**Σχήμα 1:** Αντιστοιχία χαρακτηριστικών ενός ανθρώπινου οφθαλμού (ματιού) (α) με τα χαρακτηριστικά μιας κάμερας (β)

# 1. ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

## 1.1 ΤΙ ΕΙΝΑΙ ΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ;

### 1.1.1 Ορισμός

Τα Τεχνητά Νευρωνικά Δίκτυα αποτελούν μία νέα ανακάλυψη και η ανάπτυξή τους λαμβάνει χώρο σε παγκόσμιο επίπεδο. Η έμπνευσή τους προήλθε από τα νευρωνικά συστήματα που διαθέτουν οι ζωτικοί οργανισμοί, κυρίως οι ανθρώπινοι, (οι φυτικοί οργανισμοί δεν διαθέτουν), οι οποίοι έχουν την ικανότητα να μαθαίνουν, να ξεχνούν, να θυμούνται, κλπ. Είναι γεγονός ότι η ανάπτυξη των ΤΝΔ έχει ξεπεράσει σε μεγάλο βαθμό τις ικανότητες των βιολογικών νευρωνικών δικτύων και πλέον θεωρούνται ανεξάρτητο κομμάτι της τεχνολογίας, αλλά όχι της βιολογίας, ούτε της κλασσικής λειτουργίας των υπολογιστών.

Ουσιαστικά είναι ένα συγκρότημα μονάδων επεξεργασίας ή κόμβων που υπάρχουν διασυνδέσεις μεταξύ τους, το οποίο επιλύει διαφόρων ειδών προβλήματα με την χρήση του ηλεκτρονικού υπολογιστή. Βασίζεται στον συνδυασμό του τρόπου σκέψης ενός ανθρώπινου εγκεφάλου και σε γνώσεις μαθηματικής Ανάλυσης, εκμεταλλεύοντας τον τεράστιο όγκο δεδομένων που λαμβάνει στην είσοδο. Έτσι με τις αντίστοιχες μαθηματικές πράξεις και τους κατάλληλους αλγορίθμους εκπαίδευσης, καταλήγει στο επιθυμητό αποτέλεσμα. Ένα απλό παράδειγμα είναι, μία κάμερα και πρέπει να αναγνωρίσει και να διακρίνει από μία λήψη της, μία γάτα και έναν σκύλο με την βοήθεια μαθηματικών πράξεων, πολύπλοκων συναρτήσεων και των αντίστοιχων αλγορίθμων.

### 1.1.2 Ιστορικά στοιχεία

Τα νευρωνικά δίκτυα και η Βαθιά μάθηση, δεν θεωρούνται ότι έχουν μακροβιότερη ιστορία σε σχέση με άλλες επιστήμες, επειδή ουσιαστικά είναι μία ανακάλυψη των τελευταίων δεκαετιών. Η ανάπτυξη των ΝΔ κορυφώθηκε σε παγκόσμιο επίπεδο από το 1980 και έπειτα. Η κορύφωση αυτή οφείλεται στην ανάπτυξη των λογισμικών των Η/Υ αλλά και των αλγορίθμων εκπαίδευσης.

Η συνεργασία του νευροφυσιολόγου McCulloch και του φοιτητή μαθηματικών Pitts το 1943, οδήγησε στην ανακάλυψη του πρώτου ΝΔ που παρουσιάζει την

ύπαρξη πολυάριθμων νευρώνων και ερμηνεύει την λειτουργία τους, μέσω των διασυνδέσεών τους. Η ιδέα αυτή βασίστηκε σε ένα ηλεκτρικό κύκλωμα. Το 1947, κατάφεραν να προχωρήσουν το έργο τους στο να αναγνωρίζει σχήματα και θεώρησαν ότι οι νευρώνες μπορούν να δεχτούν πολλές εισόδους, αλλά μόνο μία έξοδο. Η διαδικασία της αποστολής των πληροφοριών από τον έναν νευρώνα στον άλλον, γίνεται με την μορφή παλμών και σε συγχρονισμένο ρυθμό. Στην συνέχεια, οι ίδιοι συγγραφείς εξηγούν έναν πιθανό μηχανισμό μνήμης. Πιο συγκεκριμένα, αν υπάρξει μία κλειστή διαδρομή του σήματος μέσα στο δίκτυο, δηλαδή μία ίνα ενώσει την έξοδο του νευρώνα με το σημείο εισόδου του, τότε το σήμα θα ακολουθεί την κυκλική τροχιά και θα επανατροφοδοτεί τον νευρώνα με τις ίδιες πληροφορίες. Λίγα χρόνια αργότερα, τις παραπάνω γνώσεις αξιοποίησε ο J. Von Neumann ως παράδειγμα για υπολογιστικές μηχανές και τότε τα βιολογικά δίκτυα αποτέλεσαν πρώτη φορά πηγή έμπνευσης ώστε να δημιουργηθούν τα ΤΝΔ.

Το 1949 ο D. Hebb εκδίδει το βιβλίο με τίτλο “The organization of behavior”, το οποίο εξηγεί τον κανόνα μάθησης του Hebb με κύριο θέμα τις συνδέσεις των νευρώνων. Μετά από πολλές μελέτες κατέληξε στο ότι κάθε φορά που το δίκτυο χρησιμοποιεί τις νευρωνικές συνδέσεις του, αυτό εξειδικεύεται ολοένα και πιο πολύ και μαθαίνει το πρότυπο.

Ο F. Rosenblatt ανακοίνωσε το 1957 το πρώτο μοντέλο αισθητήρα (*perceptron*). Αρχικά το μοντέλο αυτό διαθέτει δύο επίπεδα, το επίπεδο της εισόδου και το επίπεδο της εξόδου, ενώ το σήμα κατευθύνεται μονόδρομα. Παρόλο που το συγκεκριμένο πρότυπο είχε αρκετές επιτυχίες, το 1969 μέσω του βιβλίου “*perceptrons*” των Minsky και Papert, το οποίο εκτιμά την χρησιμότητα του μοντέλου, αλλά ταυτόχρονα αποδεικνύει με αναλυτικά μαθηματικά ότι περιέχει αυστηρούς περιορισμούς. Αυτό είχε σαν αποτέλεσμα, να απωθήσει ερευνητές και ειδικούς στο να ασχοληθούν με το αντικείμενο αυτό για τα επόμενα 20 χρόνια, μέχρι ο βιολόγος Horfield το 1982 να αποδείξει ότι τα ΝΔ μπορούν να χρησιμοποιηθούν σαν αποθηκευτικοί χώροι και επίσης μπορούν να αναστήσει την πληροφορία ενός συστήματος, μόνο με μερικά από τα τμήματά του.

Λίγα χρόνια αργότερα η ιδέα της μελέτης των ΝΔ ξαναβγήκε στο προσκήνιο και η διαδικασία της εκπαίδευσης των δικτύων κορυφώθηκε όταν παρουσιάστηκε ο κανόνας *διόρθωσης του σφάλματος*. Ο κανόνας αυτός, κατά την εκπαίδευση, εστιάζει στην διαφορά της τιμής που λαμβάνουμε από την επιθυμητή τιμή, σε όποια κατάσταση και αν βρίσκεται μία δεδομένη στιγμή. Η διαφορά αυτή ονομάζεται σφάλμα και μέσω αυτού, το δίκτυο μαθαίνει στον έλεγχο και στην αυτοβελτίωση διορθώνοντας τα βάρη  $w$  των νευρώνων.

Το 1986 το μοντέλο *perceptron* παίρνει άλλη διάσταση, όταν οι McClelland και Rumelhart εκδίδουν το βιβλίο τους με τίτλο “*Parallel Distributed Processing*” στο οποίο εξηγούν την ιδέα, πως ένα ΝΔ μπορεί να θεωρηθεί ως παράλληλος επεξεργαστής. Βασισμένο στο μοντέλο *perceptron*, με την διαφορά να δίνει την δυνατότητα να υπάρξουν περισσότερα στρώματα νευρώνων στην δομή του δικτύου, τα οποία θα επεξεργάζονται τα δεδομένα. Προωθούν μία νέα μέθοδο εκπαίδευσης, την *μέθοδο της οπισθοδιάδοσης (back-propagation)* που από το 1986 (που παρουσιάστηκε τεκμηριωμένα) μέχρι και σήμερα είναι η πιο χρήσιμη τεχνική εκπαίδευσης δικτύων.

Πλέον τα ΝΔ αποτελούν ένα ώριμο και ανεξάρτητο μέρος της τεχνολογίας με εκατοντάδες επιστήμονες να στρέφονται προς το μέρος αυτό. Από το 1986 και έπειτα, τα πρώτα συνέδρια και οι πρώτες επαγγελματικές εταιρείες, που ασχολούνται με τα ΝΔ, κάνουν την εμφάνισή τους. Επίσης λίγο πριν το 1990 παρουσιάζονται και τα πρώτα περιοδικά με θέμα τα ΝΔ. Τα τελευταία δεκαπέντε χρόνια ιδρύθηκαν

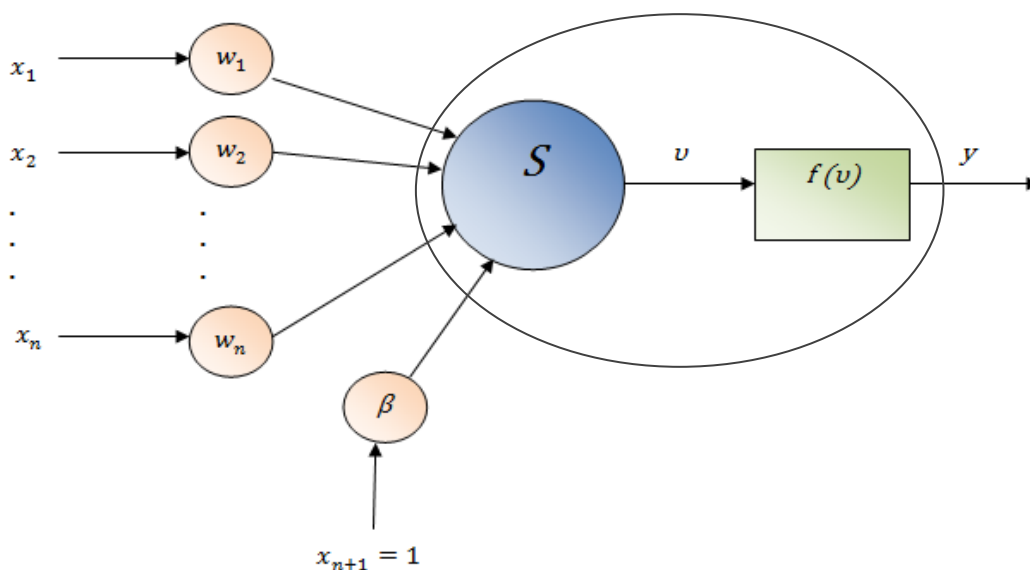
μεγάλες εμπορικές εταιρείες στις ΗΠΑ που παράγουν προγράμματα επίλυσης προβλημάτων αποκλειστικά με την χρήση ΝΔ.

### 1.1.3 Μοντέλο νευρωνικών δικτύων

Όπως αναφέρθηκε και προηγουμένως, τα ΤΝΔ είναι μία προσέγγιση των ΝΔ των ζωτικών οργανισμών. Αυτό συνεπάγεται με το ότι οι επιστήμονες έχουν καταλήξει στην δημιουργία μοντέλου ενός τεχνητού νευρώνα. Ο κάθε νευρώνας αποτελείται από τους δενδρίτες του. Οι δενδρίτες δέχονται κάποια σήματα εισόδου  $(x_1, x_2, \dots, x_n)$  από τους δενδρίτες του προηγούμενου νευρώνα (εφόσον υπάρχει). Στην συνέχεια, οι τιμές αυτών των σημάτων πολλαπλασιάζονται με τα αντίστοιχα βάρη ή αλλιώς συνάψεις  $(w_1, w_2, \dots, w_n)$ , τα οποία βάρη δηλώνουν πόσο επηρεάζει ο ένας νευρώνας τον άλλον ανάλογα με την τιμή του αντίστοιχου βάρους. Σε επόμενο στάδιο, η επεξεργασία των σημάτων γίνονται στον πυρήνα του νευρώνα. Πιο συγκεκριμένα, τα παραπάνω γινόμενα αθροίζονται με την συσκευή του αθροιστή  $S$  και η συνάρτηση ενεργοποίησης  $f$  υπολογίζει την τιμή του σήματος εξόδου του νευρώνα  $y$ , συναρτήσει του αθροίσματος  $S$  και της τιμής κατωφλιού  $\theta$ . Η τιμή κατωφλιού περιγράφει την κατάσταση ενός τεχνητού νευρώνα, ο οποίος περιγράφεται από έναν δυαδικό αριθμό  $y$  που όταν  $y=0$  τότε ο νευρώνας βρίσκεται σε αδρανή κατάσταση, ενώ όταν  $y=1$  ο νευρώνας βρίσκεται σε ενεργή κατάσταση. Ο αθροιστής λαμβάνει ακόμα μία είσοδο με σταθερή τιμή, δηλαδή  $x_{n+1} = 1$  και στην συνέχεια πολλαπλασιάζεται με την σταθερά πόλωσης  $\beta$  που ισούται με το  $n+1$  βάρος, δηλαδή  $w_{n+1} = \beta$ . Επομένως καταλήγουμε στο ότι η τιμή της εξόδου του αθροιστή  $v$  υπολογίζεται από την παρακάτω σχέση:

$$v = \sum_{i=1}^n w_i x_i + \beta = \sum_{i=1}^{n+1} w_i x_i \quad (1.1)$$

Συνεπώς στο Σχ. 1.1 απεικονίζεται το μοντέλο ενός τεχνητού νευρώνα.



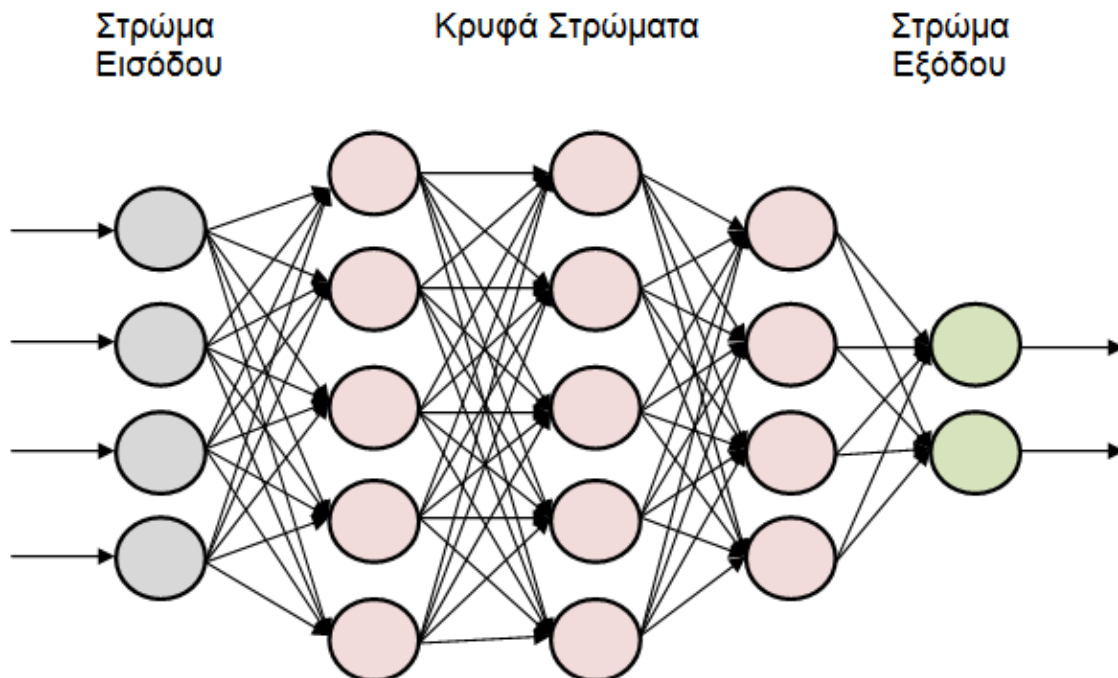
Σχήμα 1.1: Μοντέλο τεχνητού νευρώνα.



Όταν πολλοί νευρώνες μαζί συνδέονται με συνάψεις, τότε δημιουργείται ένα νευρωνικό δίκτυο, όπου ο κάθε νευρώνας χρησιμοποιεί την κατάλληλη συνάρτηση ενεργοποίησης. Οι νευρώνες ταξινομούνται σε στρώματα-επίπεδα ανάλογα με τον αλγόριθμο που χρησιμοποιούν για την εκπαίδευση του δικτύου. Οι τύποι των νευρώνων είναι οι εξής: *νευρώνας εισόδου*, *νευρώνας εξόδου* και *κρυφός νευρώνας*, οι οποίοι κατατάσσονται και στα αντίστοιχα στρώματα. Επισημαίνεται, ότι η τιμή της εξόδου ενός νευρώνα σε ένα στρώμα θεωρείται ίση με την τιμή της εισόδου ενός νευρώνα στο διαδοχικό στρώμα. Οι κρυφοί νευρώνες επεξεργάζονται τα δεδομένα.

Σε όλα τα ΤΝΔ δεν είναι απαραίτητο να συνδέονται όλοι οι νευρώνες ενός επιπέδου με όλους τους νευρώνες των γειτονικών επιπέδων. Τα δίκτυα που συμβαίνει αυτό ονομάζονται *πλήρως συνδεδεμένα*, ενώ τα δίκτυα που δεν συνδέονται όλοι οι νευρώνες μεταξύ τους ονομάζονται *μερικώς συνδεδεμένα*. Ο διαχωρισμός των ΤΝΔ γίνεται με βάση την αρχιτεκτονική του κάθε δικτύου. Η αρχιτεκτονική του δικτύου φανερώνει τον αριθμό και την σειρά των νευρώνων, δηλαδή τον τρόπο διάταξης της σύνδεσης των νευρώνων.

Η πιο απλή μορφή αρχιτεκτονικής που μπορεί να έχει ένα ΤΝΔ έχει την λογική ότι το σήμα που μεταφέρει τις πληροφορίες του δικτύου, πρέπει να ξεκινήσει να τροφοδοτεί τους νευρώνες από το επίπεδο εισόδου και να καταλήξει μονόδρομα στο επίπεδο εξόδου, χωρίς την ανάδρασή του. Αυτά τα δίκτυα ονομάζονται *Δίκτυα πρόσθιας τροφοδότησης* ή αλλιώς *feedforward νευρωνικά δίκτυα*. Αντιθέτως, σε περίπτωση που η ανάδραση του σήματος είναι επιτρεπτή, τότε τα δίκτυα αυτά δημιουργούν βρόχους μέσα στο δίκτυο και ονομάζονται *Αναδρομικά Δίκτυα* ή *recurrent neural networks (RNN)*. Στο Σχ. 1.2 παρατίθεται ένα παράδειγμα ενός απλού ΤΝΔ.



**Σχήμα 1.2:** Μοντέλο τεχνητού νευρωνικού δικτύου, εμπρόσθιας τροφοδότησης πολλών επιπέδων, πλήρως συνδεδεμένων νευρώνων.

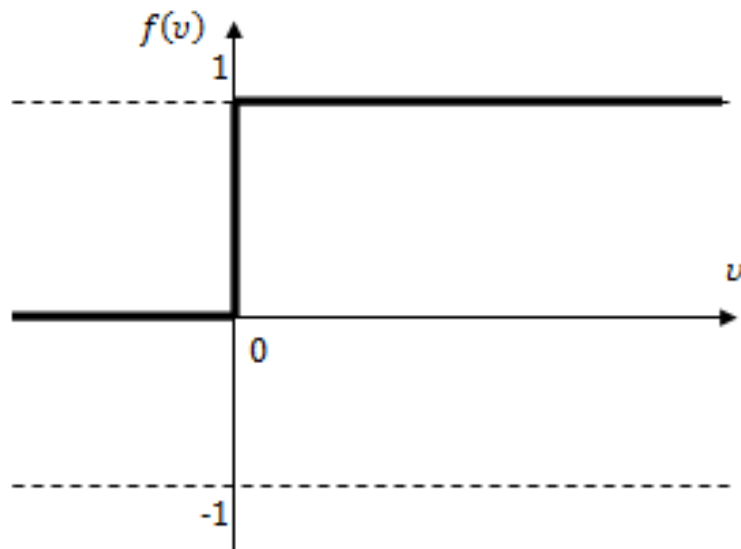
## 1.2 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ

Μέσω των συναρτήσεων ενεργοποίησης (*activation function*) ή αλλιώς συναρτήσεων μεταφοράς (*transfer function*), προσδιορίζεται η τιμή της εξόδου των νευρώνων, καθώς και η μεταφορά του σήματος μέσα στο δίκτυο αυτό. Υπάρχουν δύο είδη συναρτήσεων μεταφοράς, οι οποίες είναι οι γραμμικές και οι μη-γραμμικές. Οι γραμμικές συναρτήσεις πλέον είναι οι λιγότερο σημαντικές και χρήσιμες, επειδή δεν μπορούν να ανταπεξέλθουν στα σημερινά δεδομένα των ΤΝΔ, λόγω της πολυπλοκότητάς τους. Παρακάτω αναφέρονται κάποιες βασικές μη-γραμμικές συναρτήσεις μεταφοράς.

### 1.2.1 Βηματική Συνάρτηση

Η συνάρτηση αυτή μεταδίδει το σήμα δυαδικά, δηλαδή έχει δύο πιθανά αποτελέσματα. Τα αποτελέσματα καθορίζονται από την σταθερά κατωφλιού  $\theta$  (*threshold*). Ειδικότερα, ένας νευρώνας δέχεται πληροφορίες (είσοδοι) και υπολογίζει ένα αποτέλεσμα  $v$ . Στην συνέχεια η τιμή αυτή συγκρίνεται με την σταθερά κατωφλιού του νευρώνα. Αν η τιμή του  $v$  ξεπερνάει την τιμή της σταθεράς  $\theta$ , τότε η συνάρτηση παίρνει την τιμή 1. Άρα λέμε ότι ο νευρώνας είναι ενεργός και μεταφέρει το σήμα του. Αντιθέτως αν δεν ξεπεράσει την σταθερά, τότε η συνάρτηση παίρνει την τιμή 0 και λέμε ότι ο νευρώνας είναι αδρανής. Η βηματική συνάρτηση ή συνάρτηση κατωφλιού απεικονίζεται στο Σχ. 1.3 και με αλγεβρικούς όρους παίρνει την εξής μορφή:

$$f(v) = \begin{cases} 0, & \text{αν } v < \theta \\ 1, & \text{αν } v \geq \theta \end{cases} \quad (1.2)$$

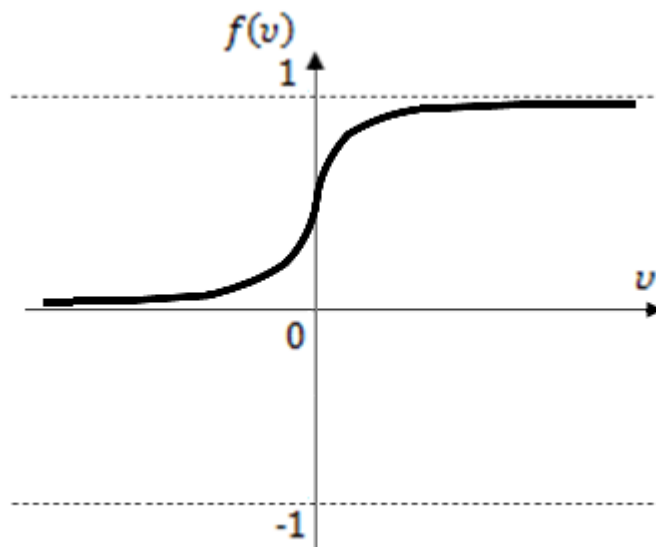


**Σχήμα 1.3:** Γραφική παράσταση βηματικής συνάρτησης μεταφοράς.

### 1.2.2 Σιγμοειδής Συνάρτηση

Η σιγμοειδής συνάρτηση μεταφοράς (*sigmoid function*) έχει γίνει ευρέως η πιο καθιερωμένη συνάρτηση στα ΤΝΔ. Η διαφορά της με την συνάρτηση κατωφλιού είναι ότι εδώ δεν λαμβάνουμε μόνο δύο τιμές στην έξοδο, αλλά και τις ενδιάμεσες τιμές τους. Καθώς εισέρχονται τα δεδομένα, τότε ο σιγμοειδής νευρώνας έχει την δυνατότητα της επιτρεπτής ρύθμισης των βαρών των συνδέσεων ( $w_1, w_2, \dots, w_n$ ) με σκοπό την καλύτερη προσέγγιση της επιθυμητής τιμής της εξόδου. Η σιγμοειδής συνάρτηση καθορίζεται από την παρακάτω σχέση και η γραφική της παράσταση παρουσιάζεται στο Σχ. 1.4.

$$f(v) = \frac{1}{1 + e^{-v}} \quad (1.3)$$

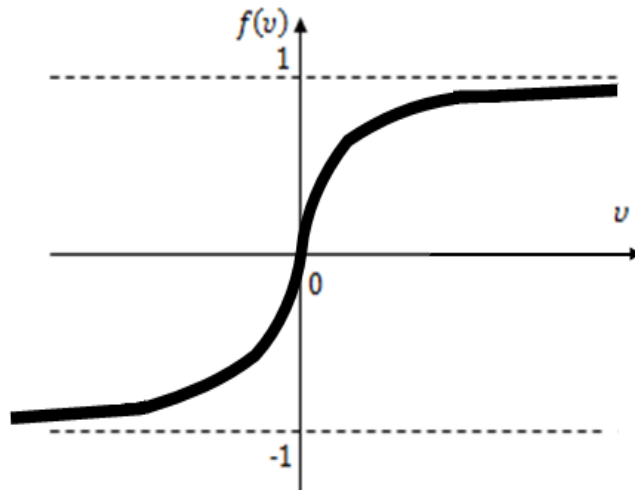


**Σχήμα 1.4:** Γραφική παράσταση σιγμοειδής συνάρτησης μεταφοράς

### 1.2.3 Υπερβολική Εφαπτομένη Συνάρτηση Ενεργοποίησης

Η υπερβολικής εφαπτομένης (*hyperbolic tangent*) συνάρτηση μεταφοράς, ουσιαστικά είναι μία σιγμοειδής συνάρτηση με την διαφορά ότι η συγκεκριμένη, δίνει τιμές εξόδου στο διάστημα (-1,1) και έχει επίκεντρο το μηδέν. Το μειονέκτημα της είναι ότι υπάρχει πιθανότητα το δίκτυο να καταλήξει με ελλιπής εκπαίδευση, εξ' αιτίας των αρνητικών τιμών εισόδου. Στην συνέχεια εμφανίζεται η γραφική της παράσταση στο Σχ.1.5, καθώς και η αλγεβρική της μορφή.

$$f(v) = \tanh v = \frac{\sin v}{\cos v} = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad (1.4)$$



**Σχήμα 1.5:** Γραφική παράσταση συνάρτησης μεταφοράς τύπου υπερβολικής εφαπτομένης

### 1.3 ΕΚΠΑΙΔΕΥΣΗ ΤΕΧΝΗΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ

Όπως έχει προαναφερθεί, τα ΤΝΔ υπάρχουν για να επιλύουν καθημερινά προβλήματα που τους επιβάλλονται ή να τα επιλύουν χωρίς την ανθρώπινη παρέμβαση. Βασικό χαρακτηριστικό των ΤΝΔ είναι ότι μαθαίνουν-εκπαιδεύονται κατάλληλα, έτσι ώστε να βελτιώνουν ολοένα και περισσότερο την απόδοσή τους, ώστε να καταλήξουν στο επιθυμητό αποτέλεσμα. Ο τρόπος με τον οποίο επιτυγχάνεται η μάθηση γίνεται με την βοήθεια αλγορίθμων και είναι περιοδικός μέχρις ότου, τα σφάλματα να εξαιρεθούν όσο το δυνατόν περισσότερο. Έτσι η έξοδος παίρνει την πλησιέστερη τιμή της επιθυμητής. Τότε ένα ΤΝΔ μπορεί να θεωρηθεί κατάλληλα εκπαιδευμένο με κάποιο συγκεκριμένο πρότυπο. Η τιμή της εξόδου, κάθε επανάληψης, καθορίζεται από τις τιμές των βαρών των συνδέσεων  $w$ , οι οποίες μεταβάλλονται μέχρι το δίκτυο να εκπαιδευτεί πλήρως.

Δύο είναι οι τρόποι με τους οποίους τα βάρη αλλάζουν και αναλύονται στην συνέχεια. Πρώτος τρόπος είναι η εκπαίδευση *με επίβλεψη* (*supervised learning*). Είναι ο πιο καθιερωμένος τρόπος μάθησης ενός δικτύου. Αρχικά το δίκτυο τροφοδοτείται με πρότυπα, όπως είναι οι τιμές των εισόδων και των επιθυμητών εξόδων. Τότε τα βάρη των συνδέσεων  $w$  και οι πολώσεις, κατά την εκπαίδευση, προσαρμόζονται αναλόγως, με στόχο την ελαχιστοποίηση του σφάλματος μεταξύ της θεωρητικής και της πραγματικής τιμής εξόδου. Η διαδικασία γίνεται σταδιακά μέχρι η πρακτική τιμή της εξόδου να προσεγγίσει την θεωρητική.

Δεύτερος τρόπος είναι η εκπαίδευση *χωρίς επίβλεψη* (*unsupervised learning*). Είναι λιγότερο διαδεδομένη η χρήση του σε σχέση με τον προηγούμενο τρόπο, αλλά εξίσου σημαντικός. Η διαφορά τους είναι ότι πλέον το δίκτυο, δεν τροφοδοτείται με τιμές εξόδων και έτσι δεν υπάρχει σύγκριση με κάποια θεωρητική τιμή. Επιπλέον διαθέτει μία αυτόνομη παράμετρο αλλαγής βαρών και ακολουθεί μία συγκεκριμένη διαδικασία, που διορθώνει τα σφάλματα με έναν μηχανισμό ανάδρασης (*feedback*). Επίσης το δίκτυο χρησιμοποιεί έναν εσωτερικό έλεγχο και προσπαθεί να

προσδιορίσει τα χαρακτηριστικά των τιμών εξόδου, έτσι ώστε να ταυτίζονται με τα χαρακτηριστικά των τιμών εισόδου. Όταν οι τιμές των βαρών πάψουν να αλλάζουν, τότε το δίκτυο έχει εκπαιδευτεί πλήρως.

## 1.4 ΣΤΟΙΧΕΙΑ ΒΙΟΛΟΓΙΑΣ

Οι ζωτικοί εγκέφαλοι (ανθρώπινοι και ζωικοί), έχουν για μοναδικό δομικό υλικό τους νευρώνες, οι οποίοι διαχωρίζονται σε 100 διαφορετικές κατηγορίες και όλοι είναι διαφορετικοί μεταξύ τους. Ο ανθρώπινος εγκέφαλος είναι ο πιο πολύπλοκος και αποτελείται από δισεκατομμύριους νευρώνες ( $\cong 10^{10}$ ). Οι νευρώνες δεν αναπαράγονται, ούτε πολλαπλασιάζονται παρά μόνο καταστρέφεται ένας αριθμός τους σε καθημερινή βάση, ανάλογα με την κατανάλωση αλκοόλ και την ηλικία του οργανισμού. Ο κάθε νευρώνας συνδέεται με περίπου 10.000 άλλους νευρώνες με την βοήθεια συνδέσεων που ονομάζονται συνάψεις. Οι συνάψεις αυτές βρίσκονται σε ισορροπία, δηλαδή όταν καταστρέφονται οι παλιές δημιουργούνται αυτόματα νέες. Αυτό συμβαίνει επειδή ο οργανισμός αποκτάει γνώσεις και εμπειρία. Στην καταστροφή των συνδέσεων οφείλονται οι περισσότερες ασθένειες των ηλικιωμένων.

Ορισμένοι νευρώνες μαζί με τις αντίστοιχες συνάψεις τους αποτελούν ένα ΝΔ και όλα τα ΝΔ αποτελούν το Κεντρικό Νευρωνικό Σύστημα που το μεγαλύτερο τμήμα του καταλαμβάνει θέση στον εγκέφαλο και στην σπονδυλική στήλη του οργανισμού. Πολλά τμήματα του εγκεφάλου εξειδικεύονται στις αισθήσεις. Η πιο μελετημένη αίσθηση είναι αυτή της όρασης, με την οποία θα ασχοληθούμε και στην συγκεκριμένη εργασία.

### 1.4.1 Βιολογικός Νευρώνας

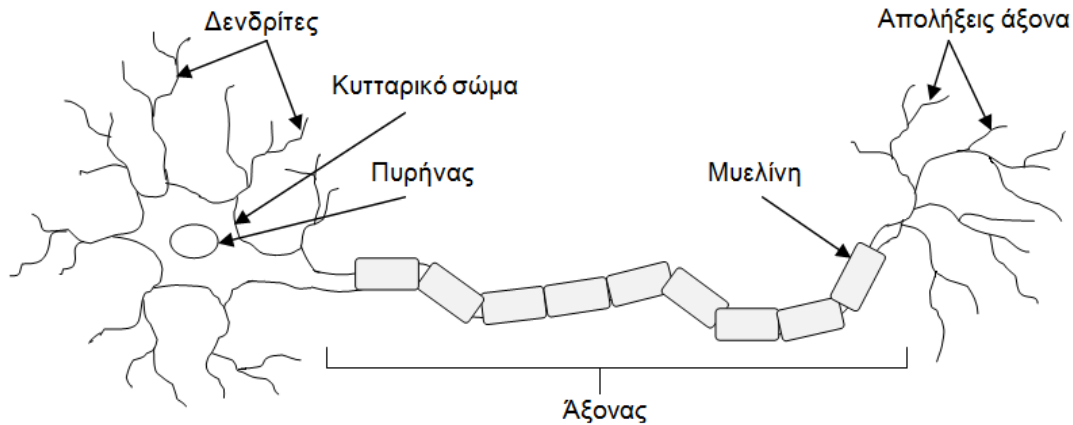
Ένας νευρώνας υποδιαιρείται στο σώμα, στους δενδρίτες, στον άξονα και στις συνάψεις όπως φαίνεται στο Σχ. 1.6. Το σώμα είναι κυρίως μέρος του νευρώνα, όπου περιλαμβάνει τον πυρήνα του. Μέσα στον πυρήνα βρίσκονται όλες οι πληροφορίες του γενετικού υλικού του οργανισμού και εκτελούνται όλες οι χημικές διαδικασίες για την επιβίωση του κυττάρου.

Οι δενδρίτες παρομοιάζονται με διακλαδώσεις δέντρων και μέσω αυτών, εισέρχεται η ροή δεδομένων στον νευρώνα από τον προηγούμενό του με την μορφή ηλεκτρικών παλμών. Το ηλεκτρικό σήμα εξέρχεται μέσω του άξονα του νευρώνα προς τους δενδρίτες των γειτονικών νευρώνων. Ορισμένοι άξονες έχουν επικάλυψη μιας ουσίας, της μυελίνης. Στα κύτταρα με τέτοιους άξονες, η ταχύτητα του ηλεκτρικού σήματος φτάνει τα 100 *m/s*. Διαφορετικά, οι άξονες δεν έχουν καμία επικάλυψη και η ταχύτητα του σήματος μέσα στον νευρώνα κυμαίνεται από 10-20 *m/s*.

Μία σύναψη είναι ένα πολύ μικρό διάκενο στα σημεία σύνδεσης-επαφής των αξόνων με τους δενδρίτες. Το πλάτος της σύναψης και η πυκνότητα του ηλεκτροχημικού υλικού της εξόδου του άξονα, καθορίζουν το πόσο άμεσα θα μεταδοθεί το ηλεκτρικό σήμα από τον έναν νευρώνα στον άλλο. Οι ποσότητες των σημάτων αυτών ονομάζονται συναπτικά βάρη.

Ακόμη, ένας νευρώνας μπορεί να θεωρηθεί ενεργός ή μη-ενεργός, δηλαδή είναι ένα δυαδικό στοιχείο. Πιο συγκεκριμένα, όταν είναι ενεργός, τότε πυροδοτεί

ηλεκτρικό σήμα και μέσω αυτού του σήματος μεταφέρει δεδομένα σε κάποιον γειτονικό νευρώνα. Για να γίνει αυτό, πρέπει το συνολικό άθροισμα των φορτίων της εισόδου, να είναι μεγαλύτερο από ένα κατώφλι. Αντίθετα μη-ενεργός νευρώνας σημαίνει ότι δεν παράγει ηλεκτρικό σήμα, δηλαδή είναι αδρανής.



**Σχήμα 1.6:** Βιολογικός νευρώνας.

#### 1.4.2 Σύγκριση με τεχνητά νευρωνικά δίκτυα

Ο ανθρώπινος εγκέφαλος έχει εξαιρετική πολυπλοκότητα και πραγματοποιεί διεργασίες που ένα ΤΝΔ αδυνατεί να προσδιορίσει. Αναφέρθηκε παραπάνω ο αριθμός των νευρώνων και των συνδέσεων ενός ανθρώπινου εγκεφάλου και διακρίνεται ότι είναι πολύ μεγαλύτερος από αυτόν μπορεί να χειριστεί ένας υπολογιστής, παρόλο που τα ΤΝΔ μπορούν να λύσουν ευκολότερα ορισμένα από τα προβλήματα που ο άνθρωπος δυσκολεύεται, όπως για παράδειγμα μαθηματικές πράξεις.

Ίσως η μεγαλύτερη διαφορά τους να βρίσκεται στην συνδεσμολογία. Στα ΤΝΔ οι συνδεσμολογίες είναι πολύ απλές, ενώ στα βιολογικά πολύ περίπλοκες. Παρόλο που η ταχύτητα διάδοσης του σήματος στους υπολογιστές είναι μεγαλύτερη κατά χιλιάδες από αυτή του ανθρώπινου εγκεφάλου, δεν αρκεί για να καλύψει την διαφορά στην πολυπλοκότητα.

Το σώμα του βιολογικού νευρώνα συγκρίνεται με μία μονάδα νευρώνα ενός ΤΝΔ. Και οι δύο μονάδες διαθέτουν αθροιστή και κατώφλι. Επιπλέον ο άξονας του βιολογικού νευρώνα παρομοιάζεται με την μονάδα εξόδου  $y$  του τεχνητού νευρώνα.



## 2. ΧΡΗΣΗ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ

### 2.1 ΓΕΝΙΚΑ

Ανέκαθεν τα ρομπότ είχαν μοναδικό στόχο την εκτέλεση συγκεκριμένων καθηκόντων που αντικαθιστούσαν την ανθρώπινη εργασία, είτε για λόγους ασφαλείας και ευκολίας, είτε για λόγους αξιοπιστίας και ακρίβειας. Τα πρώτα ρομπότ που εφευρέθηκαν ήταν προγραμματισμένα με τέτοιο τρόπο έτσι ώστε να εκτελούνται επαναλαμβανόμενες και προκαθορισμένες ενέργειες για την επιτυχία των εργασιών. Με την εξέλιξη της τεχνολογίας όμως, ενσωματώθηκε και η τεχνητή νοημοσύνη, εκτοξεύοντας το επίπεδο της ρομποτικής επιστήμης όπου ακόμη και σήμερα αναπτύσσεται όλο και περισσότερο. Ειδικότερα, πλέον η τεχνητή νοημοσύνη έχει ενισχύσει τα ρομπότ έτσι ώστε να βλέπουν, να μιλούν, να υπολογίζουν, να μαθαίνουν και γενικά να έχουν αντίληψη και να είναι εξυπνότερα σε σχέση με τα ρομπότ παλαιότερης τεχνολογίας.

Η πορεία της εξέλιξης αυτής, έχει οδηγήσει την ανθρωπότητα σε μεγάλο ενθουσιασμό αλλά και σε μεγάλο φόβο για τυχόν ανεξέλεγκτη κατάσταση και ότι τα ρομπότ θα ξεπεράσουν τους ανθρώπους, φέρνοντας έτσι κοινωνική και οικονομική κρίση. Προς το παρόν, το στάδιο που βρίσκεται η τεχνητή νοημοσύνη σήμερα υστερεί σε συναισθήματα και δεν διαθέτει την “κοινή λογική” των ανθρώπων. Ακόμη δεν υπάρχει πλήρης τεχνητή νοημοσύνη, αλλά ουσιαστικά είναι ένας τρόπος μίμησης μίας μη ολοκληρωμένης ανθρώπινης νοημοσύνης. Βασικές λειτουργίες των οποίων οι άνθρωποι επεξεργάζονται τα δεδομένα τους είναι η δημιουργία, η κωδικοποίηση και η αποθήκευση. Με βάση αυτές τις λειτουργίες δημιουργήθηκαν και οι πρώτοι λογικοί υπολογιστές.

Η τεχνητή νοημοσύνη υποδιαιρείται σε ορισμένες κατηγορίες, όπως είναι τα τεχνητά νευρωνικά δίκτυα, η ασαφής λογική, οι πιθανολογικές μέθοδοι, κλπ. Τα τεχνητά νευρωνικά δίκτυα όπως αναλύθηκαν στο προηγούμενο κεφάλαιο, ήταν η αιτία για την κατασκευή των ρομπότ με αντίληψη, τα οποία μπορούν να χρησιμοποιηθούν με μεγάλη επιτυχία σε εφαρμογές π.χ. αναγνώρισης ομιλίας-ήχου-μουσικής, αναγνώρισης προτύπων, μηχανικής όρασης σε πραγματικό χρόνο. Οι κατηγορίες αυτές συνθέτουν την διαδικασία της εκμάθησης των ρομπότ.

Υπάρχουν διάφορα είδη ρομπότ. Ορισμένα από αυτά είναι τα ρομπότ που φροντίζουν, ρομπότ έρευνας και διάσωσης, ρομπότ που συναρμολογούν, ειδικά για στρατιωτικούς σκοπούς καθώς υπάρχουν στην αγορά και παιχνίδια ρομπότ. Ένα παράδειγμα μίας πρόσφατης ανακάλυψης στην ρομποτική με τεχνητή νοημοσύνη, είναι τα οχήματα με αυτόματο σύστημα διεύθυνσης, κοινώς και ως αυτόνομα οχήματα ή οχήματα χωρίς οδηγό. Αυτή η επιτυχία οφείλεται στην τοποθέτηση



καμερών και αισθητήρων, οι οποίοι αναγνωρίζουν το περιβάλλον γύρω από το όχημα και έτσι ο εγκέφαλος παίρνει τις σωστές αποφάσεις για πιθανούς ελιγμούς ή φρεναρίσματα στην κυκλοφορία του οδικού δικτύου.

Όπως αναφέρθηκε παραπάνω, η εκθετική εξέλιξη της τεχνητής νοημοσύνης παραπέμπει σε φόβο. Γι' αυτό είναι αφορμή να αποτυπωθούν οι αρνητικές συνέπειες και να βρεθούν λύσεις σε τεχνολογικά, κοινωνικά και νομικά ζητήματα.

## **2.2 ΣΥΝΕΠΕΙΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ**

Όλη αυτή η ανάπτυξη της τεχνητής νοημοσύνης δεν επιφέρει απολύτως θετικά αποτελέσματα στην κοινωνία. Ειδικοί έχουν μελετήσει και έχουν καταλήξει σε ορισμένες συνέπειες της τεχνητής νοημοσύνης στην ανθρωπότητα, είτε αρνητικές είτε θετικές, αφορώντας τεχνολογικά, ηθικά, νομικά και οικονομικά ζητήματα. Μερικά από τα ζητήματα αυτά αναφέρονται παρακάτω.

Σημαντικό ζήτημα αποτελεί το ότι μερικά συστήματα τεχνητής νοημοσύνης οφείλουν να προστατέψουν τα προσωπικά δεδομένα και το απόρρητο του χρήστη, γεγονός που δεν σέβεται την ιδιωτική ζωή του. Ακόμη ένα ηθικό ζήτημα είναι το πόσο η τεχνητή νοημοσύνη λειτουργεί εις βάρος των ανθρώπων που συναναστρέφεται και περιορίζει την ελευθερία και την δημοκρατία τους. Για παράδειγμα μέσω κακόβουλων λογισμικών ή μολυσμένες συσκευές, αναπαράγεται ο κίνδυνος να κλατούν προσωπικά στοιχεία. Ως επακόλουθες αναμένονται να είναι οι νομικές κυρώσεις που αντιστοιχούν. Γι' αυτό πρέπει να προσδιοριστούν νόμοι και περιορισμοί που θα οδηγούν στη μη παράνομη και δίκαιη χρήση της τεχνητής νοημοσύνης ως προς τους ανθρώπους.

Πλέον, ο οπλισμός στρατιωτικών δυνάμεων ενισχύεται με τεχνολογίες βασισμένες στην τεχνητή νοημοσύνη. Έξυπνα όπλα, ρομπότ ελεύθεροι σκοπευτές και drones αποτελούν αντικείμενα πολέμου. Επομένως πρέπει να οριστούν διεθνείς συμφωνίες για επιτρεπτά και ασφαλή όρια χρήσης τους.

Από μέρα σε μέρα, οι θέσεις εργασίας μειώνονται και η ανεργία αυξάνεται. Πολλά από τα επαγγέλματα έχουν εξαφανιστεί ή τείνουν να εξαφανιστούν, εξ' αιτίας της αυτοματοποίησης και της αντικατάστασης του ανθρώπινου δυναμικού από μηχανές ρομπότ. Αυτό επιφέρει οικονομικές καταστροφές, με τις επόμενες γενιές να αντιμετωπίσουν κρίση σε πολλές πτυχές της ζωής τους. Πρέπει λοιπόν η αντικατάσταση αυτή, να ελέγχεται και να προβλέπεται από ειδικούς για την αποφυγή του παραπάνω συμβάντος. Παρόμοια ανησυχία προκαλεί και η απώλεια ανθρώπινου ελέγχου των ρομπότ με τεχνητή νοημοσύνη, αλλά ο τρόπος για την αντιμετώπιση του προβλήματος αυτού, είναι υπό διερεύνηση.

Όταν η τεχνητή νοημοσύνη φτάσει στο επίπεδο της συνείδησης, των συναισθημάτων και της αυτογνωσίας, τότε θα ανοίξουν νέοι ορίζοντες στην επιστήμη της ψυχολογίας που θα βοηθήσουν στην κατανόηση των ψυχολογικών προβλημάτων των ρομπότ και στην αρμονική διαβίωση τους ανάμεσα στους ανθρώπους.

## 2.3 HARDWARE ΓΙΑ ΡΟΜΠΟΤ ΜΕ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Για την υλοποίηση ενός συστήματος ρομπότ με τεχνητή νοημοσύνη απαιτούνται ορισμένα βασικά εξαρτήματα. Την πρώτη ανακάλυψη την έκανε ο Gordon Moore που εξηγεί ότι σε ένα ολοκληρωμένο κύκλωμα, ο αριθμός των τρανζίστορ διπλασιάζεται με το πέρασμα των χρόνων. Στη συνέχεια, το κύκλωμα αυτό απέκτησε την δυνατότητα κατασκευής του σε διαστάσεις ενός ατόμου της ύλης. Αυτή η αρχή λειτουργίας χρησιμοποιείται ακόμη και σήμερα για την κατασκευή των πυρήνων των επεξεργαστών δεδομένων. Αλλά, επειδή με αυτόν τον τρόπο η συχνότητα και η ισχύς ήταν σε στάσιμο επίπεδο και οι υπολογιστές δεν απέδιδαν στα σημερινά δεδομένα, ανακαλύφθηκαν οι πολυπύρρηνοι υπολογιστές. Επόμενο βήμα ήταν το λογισμικό που διέθετε, να διαμορφωθεί ως λογισμικό πολλαπλών νημάτων. Εκτός από τους επεξεργαστές, απαραίτητα εξαρτήματα για τη μηχανική όραση είναι οι κοινές κάμερες και οι αισθητήρες που βοηθούν τα ρομπότ να επικοινωνούν με το εξωτερικό τους περιβάλλον.

### 2.3.1 Επεξεργαστές

Η κεντρική μονάδα επεξεργασίας ή αλλιώς επεξεργαστής ή *CPU (Central Processing Unit)*, είναι το βασικό υλικό που χρησιμοποιείται στους σύγχρονους υπολογιστές της τεχνητής νοημοσύνης για σειριακή επεξεργασία πληροφοριών. Μόνο ένας επεξεργαστής στο υπολογιστικό σύστημα δεν αρκεί με βάση τις σημερινές απαιτήσεις. Πλέον υπάρχουν αρκετοί, ισχυροί και πολυπύρρηνοι επεξεργαστές σε ένα μόνο σύστημα. Ένας τυπικός επεξεργαστής χωρίζεται σε τρία μέρη. Αρχικά η πληροφορία εισέρχεται από την μονάδα εισόδου και καταλήγει στην μονάδα εξόδου. Οι μονάδες αυτές βοηθούν στην επικοινωνία του επεξεργαστή με το εξωτερικό του περιβάλλον. Ενδιάμεσα από τις μονάδες αυτές βρίσκεται ο μικροεπεξεργαστής που περιλαμβάνει την αριθμητική και την λογική μονάδα, τη μονάδα ελέγχου και τους καταχωρητές. Το καθένα από αυτά τα μέρη συμβάλλουν στην επεξεργασία των δεδομένων, τον συντονισμό της κάθε μονάδας του συστήματος ξεχωριστά, καθώς και στην προσωρινή αποθήκευση αντίστοιχα. Τέλος περιλαμβάνει την κύρια μνήμη που αποθηκεύει το απαιτούμενο πρόγραμμα και τα προσωρινά αποτελέσματα. Μία τυπική κεντρική μονάδα επεξεργασίας εμφανίζεται στην Εικ. 2.1.

Εκτός της CPU υπάρχει και ένα άλλο είδος επεξεργαστή, η μονάδα επεξεργασίας γραφικών ή *GPU (Graphics Processing Unit)* (Εικ. 2.2), που αντίστοιχά της είναι η κάρτα γραφικών των Η/Υ. Η GPU είναι ένα είδος επεξεργαστή, τύπου τσιπ το οποίο έχει σχεδιαστεί έτσι ώστε να επεξεργάζεται πολυδιάστατα δεδομένα. Ουσιαστικά, μία GPU βοηθάει την CPU του ίδιου συστήματος στην εργασία της επεξεργασίας. Σε σχέση με μία CPU περιλαμβάνει χιλιάδες μικρότερους πυρήνες και σε αυτό οφείλεται η επίλυση δυσκολότερων υπολογισμών και η δυνατότητα επαναλαμβανόμενης λειτουργίας σε ταχύτατο χρόνο. Η παραπάνω βασική διαφορά τους φαίνεται στο Σχ. 2.1. Για παράδειγμα, η επεξεργασία μιας εικόνας ή η προβολή ενός βίντεο, ακόμη και ένα μαθηματικό πρόβλημα, αποκρίνονται με GPU επεξεργαστή.

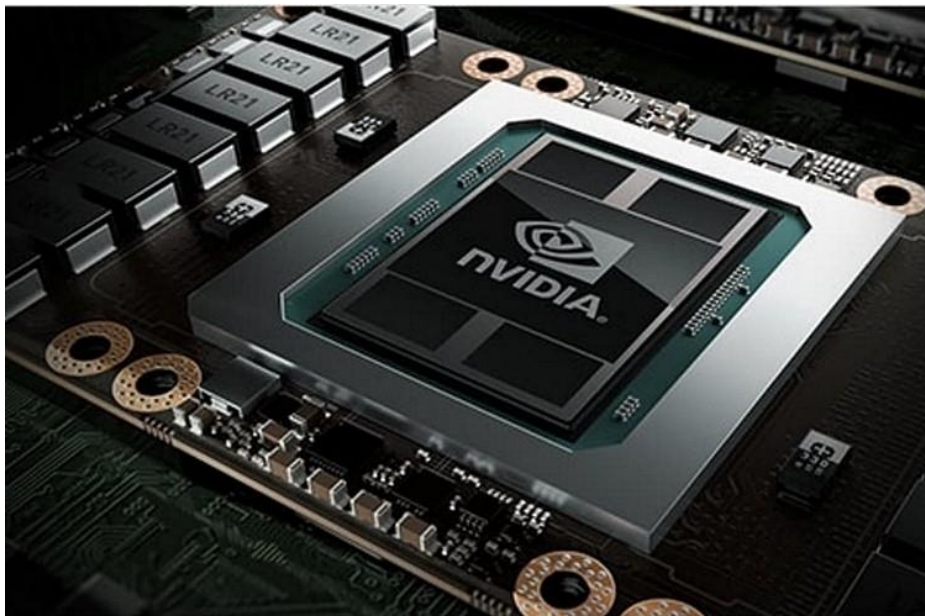
Ένα ακόμη εξάρτημα που παίρνει θέση σε συστήματα τεχνητής νοημοσύνης είναι το *FPGA (Field Programmable Gate Array)* (Εικ. 2.3). Το FPGA είναι ένα κύκλωμα με προγραμματισμένα λογικά μπλοκ που έχει ενσωματωμένες πολυάριθμες τυποποιημένες πύλες και συγκεκριμένες ψηφιακές λειτουργίες. Αλλά επειδή δεν είναι επεξεργαστής, δεν εκτελεί πρόγραμμα που να είναι αποθηκευμένο στην μνήμη,

οπότε ο τρόπος προγραμματισμού του διαμορφώνεται από γλώσσες περιγραφής υλικού (HDL). Επιπλέον λειτουργεί παράλληλα, δηλαδή έχει την δυνατότητα να επεξεργάζεται ταυτόχρονα πολλές διαφορετικές πληροφορίες και άρα συμβάλλει στην μείωση του συνολικού λειτουργικού κόστους.



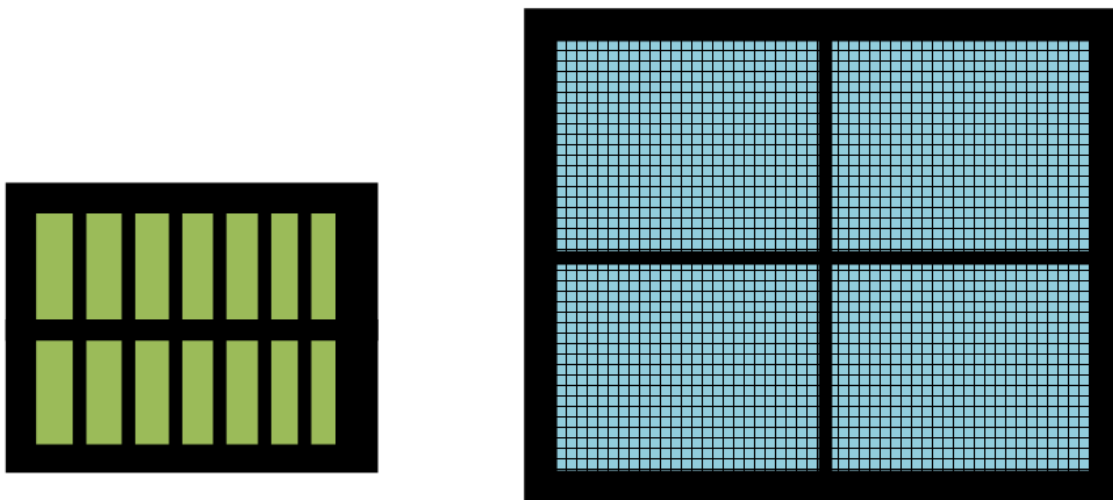
**Εικόνα 2.1:** Κεντρική Μονάδα Επεξεργασίας (CPU)

(Πηγή: <https://www.techspot.com/news/76824-intel-introduces-28-core-xeon-w-3175x-cpu.html>)



**Εικόνα 2.2:** Μονάδα Επεξεργασίας Γραφικών (GPU)

(Πηγή: <https://www.cienciaedados.com/gpu-e-deep-learning/>)



**Σχήμα 2.1:** Διαφορές μίας CPU (αριστερά) και με μίας GPU (δεξιά) με βάση την διάταξη (σειριακή – παράλληλη) και το πλήθος των πυρήνων τους.



**Εικόνα 2.3:** Συστοιχία Προγραμματιζόμενης Πύλης Πεδίου (FPGA).  
(Πηγή: <https://www.anandtech.com/show/7334/a-look-at-alteras-opencl-sdk-for-fpgas>)

### 2.3.2 Συσκευές Εισόδου

Ως μέσο εισόδου πληροφοριών στα ρομπότ με τεχνητή νοημοσύνη, ορίζονται οι κοινές κάμερες και κυρίως οι αισθητήρες (*sensors*). Με την χρήση των συσκευών αυτών, το ρομπότ επικοινωνεί με το εξωτερικό περιβάλλον του, από το οποίο περιβάλλον αισθάνονται ορισμένα ερεθίσματα ανάλογα τον τύπο του αισθητήρα ή της κάμερας, για παράδειγμα: οπτικό, ηχητικό, κλπ. Στην συνέχεια το ερέθισμα μετατρέπεται σε ηλεκτρικό σήμα και η ποσότητά του δίνει μία τιμή στην μεταβλητή της εξόδου, μέσω της συνάρτησης μεταφοράς που χαρακτηρίζει τον εκάστοτε αισθητήρα.

Γενικά, οι αισθητήρες χωρίζονται σε ενεργούς και παθητικούς. Οι ενεργοί αισθητήρες είναι ικανοί να λειτουργήσουν χωρίς την παροχή ηλεκτρικού ρεύματος, ενώ οι παθητικοί αισθητήρες δεν έχουν αυτή την δυνατότητα. Ακόμη, οι αισθητήρες χωρίζονται σε αναλογικούς και διακριτούς. Οι αναλογικοί αισθητήρες παράγουν συνεχές σήμα, ενώ το σήμα που παράγουν οι διακριτοί αισθητήρες παίρνει συγκεκριμένες τιμές. Ο πιο συνηθισμένος διακριτός αισθητήρας είναι ο δυαδικός, ο οποίος δίνει μόνο 2 τιμές.

Στην ρομποτική όμως χρησιμοποιούνται συγκεκριμένοι τύποι αισθητήρων. Μερικοί από αυτούς απεικονίζονται στην Εικ. 2.4. Πιο συγκεκριμένα είναι οι εξής: Οπτικοί αισθητήρες χρόνου πτήσης (*Time-of-Flight*), αισθητήρες θερμοκρασίας και υγρασίας, αισθητήρες ανίχνευσης φωτός και εμβέλειας (*LIDAR*), αισθητήρες υπερήχων, αισθητήρες κραδασμών και αισθητήρες κυμάτων-χιλιοστών. Στην σημερινή εποχή, τα χαρακτηριστικά των παραπάνω αισθητήρων, έχουν αναβαθμιστεί σε τέτοιο βαθμό που δίνουν στα ρομπότ περισσότερες ικανότητες και καλύτερη αντίληψη για την σωστή λήψη αποφάσεων στις δυσκολίες που έχουν να αντιμετωπίσουν.



**Εικόνα 2.4:** Διαφορετικοί τύποι αισθητήρων για ρομποτικές εφαρμογές

(Πηγή: [https://el.lambdageeks.com/robot-sensors-characteristics/?\\_ga=2.225153818.1099420019.1668123160-1431487164.1668123160#open](https://el.lambdageeks.com/robot-sensors-characteristics/?_ga=2.225153818.1099420019.1668123160-1431487164.1668123160#open))

Στην περίπτωση της συγκεκριμένης εργασίας όμως, μας ενδιαφέρουν οι αισθητήρες όρασης. Οι αισθητήρες όρασης που χρησιμοποιούνται συχνότερα σε εφαρμογές ρομποτικής, είναι οι έξυπνες κάμερες. Μία βασική εφαρμογή τους λαμβάνει χώρο σε βιομηχανίες και επιχειρήσεις, οι οποίες κάμερες παρακολουθούν

το προσωπικό και την σωστή λειτουργία της παραγωγής, καθώς ελέγχουν και την ασφάλεια των προϊόντων. Οι αισθητήρες όρασης επιτρέπουν δύο είδη προβολών. Αρχικά είναι ο ορθογραφικός τύπος προβολής που έχει ορθογωνικό οπτικό πεδίο και χρησιμεύει σε δραστηριότητες ανίχνευσης. Ο δεύτερος τύπος προβολής είναι ο προοπτικός τύπος προβολής που έχει τραπεζοειδές οπτικό πεδίο και παίρνει θέση στις κοινές κάμερες.

## 2.4 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η Μηχανική Μάθηση είναι η αιτία ανάπτυξης της τεχνητής νοημοσύνης. Χάρης αυτήν τα ρομπότ έχουν αποκτήσει το τέχνασμα να εκπαιδεύονται στο να εκτελούν συγκεκριμένες ενέργειες που επιθυμεί ο χειριστής. Αυτή η επικοινωνία, μεταξύ ρομπότ και ανθρώπου επιτυγχάνεται μέσω αλγορίθμων και εντολών.

Ο κάθε αλγόριθμος στέλνει ένα σύνολο δεδομένων εκπαίδευσης στον εγκέφαλο του ρομπότ, μεγάλης κλίμακας, έτσι ώστε να έχουν μεγαλύτερη ακρίβεια και αποδοτικότητα στα καθήκοντα που τους έχουν ανατεθεί. Οι αλγόριθμοι και οι εντολές αυτές κωδικοποιούνται στη γλώσσα των υπολογιστών, τις λεγόμενες γλώσσες Προγραμματισμού.

Όλες οι γλώσσες Προγραμματισμού αποτελούνται από σύμβολα, αριθμούς και λογικούς χαρακτήρες που είναι εύκολο να κατανοήσουν τα ρομπότ. Στα χρονικά έχουν αναπτυχθεί αρκετές γλώσσες. Μερικές έχουν σχεδόν ξεπεραστεί, επειδή έχουν ανακαλυφθεί νέες γλώσσες με περισσότερες δυνατότητες. Ακόμη, καμία από αυτές, δεν αντιστοιχεί σε όλες τις εργασίες που θέλουμε να εκτελέσουν τα ρομπότ. Αυτό συμβαίνει διότι μία από τις γλώσσες τυχαίνει να είναι η εξέλιξη μίας άλλης, δηλαδή η καθεμία γλώσσα προσφέρει διαφορετικά αποτελέσματα.

Στη συνέχεια αναφέρονται ορισμένες από τις γλώσσες Προγραμματισμού (από την παλαιότερη στην νεότερη), που συνέβαλαν στην ανάπτυξη της τεχνητής νοημοσύνης, καθώς σε μερικές θα αναφερθεί και η χρησιμότητά τους. Η πρώτη πετυχημένη γλώσσα που αναπτύχθηκε ονομάζεται *LISP (List Processing)*. Η επόμενη γλώσσα ονομάζεται *Prolog (Programming in Logic)*. Την ίδια χρονική περίοδο αναπτύχθηκαν οι *C/C++* και *Fortran*, οι οποίες χρησιμοποιούνται σε εργασίες ανάλυσης δεδομένων, λόγω της εξειδίκευσής τους στην δημιουργία συναρτήσεων και βιβλιοθηκών για επιστημονικούς υπολογισμούς. Το επίπεδο της τεχνητής νοημοσύνης το ανέβασε σε μεγάλο βαθμό η γλώσσα Προγραμματισμού *Java* που δημιούργησε τους διακομιστές cloud. Πλέον η πιο διαδεδομένη γλώσσα Προγραμματισμού στην τεχνητή νοημοσύνη, η οποία είναι και το κυρίως θέμα της συγκεκριμένης εργασίας, ονομάζεται *Python*. Η Python θεωρείται η λιγότερο χρονοβόρα (στη γραφή της), δαπανηρή γλώσσα καθώς εμφανίζει λιγότερα σφάλματα, αλλά διαθέτει και κάποια αρνητικά χαρακτηριστικά σε σχέση με άλλες γλώσσες. Επίσης, προσπαθεί να προσεγγίσει τις ικανότητες του *Matlab* για επιστημονικούς υπολογιστές. Το μέλλον επιφυλάσσει την ανακάλυψη νέων γλωσσών, καθώς και την εξέλιξη ήδη υπαρχόντων γλωσσών, ανεβάζοντας το επίπεδο της τεχνητής νοημοσύνης ακόμα περισσότερο.

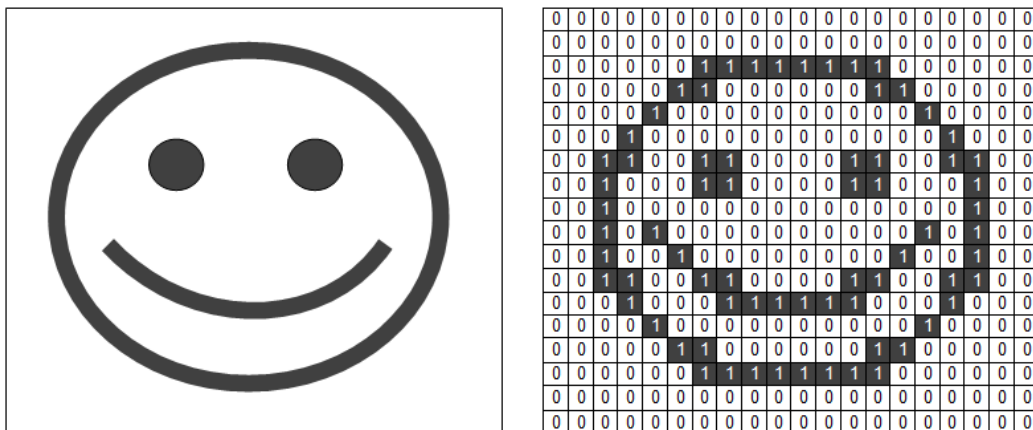
Οι βασικές επιθυμητές ενέργειες που εκτελούν τα ρομπότ από εντολές μηχανικής μάθησης, είναι η αναγνώριση προτύπων και η μετακίνησή τους από μία θέση σε μία άλλη. Η αναγνώριση προτύπων-αντικειμένων αποτελεί δεξιότητα της μηχανικής όρασης και πραγματοποιείται χάρις αυτήν.

## 2.5 ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ ΚΑΙ ΑΝΑΓΝΩΡΗΣΗ ΕΙΚΟΝΑΣ

### 2.5.1 Αρχή Λειτουργίας

Η μηχανική όραση (ή υπολογιστική όραση) επιτρέπει στα ρομπότ να έχουν την αίσθηση της όρασης. Είναι ένας τρόπος προσέγγισης της ανθρώπινης όρασης με την δεξιότητα να αναγνωρίζει σχήματα, μοτίβα και μορφές και χρησιμοποιείται συνήθως για ποιοτικό έλεγχο σε γραμμές παραγωγής και όχι μόνο. Ικανότητά της είναι να λειτουργεί 24 ώρες το 24ωρο χωρίς να χάνει την συγκέντρωσή της λόγω κούρασης, κάτι που ο άνθρωπος αδυνατεί. Έτσι υπάρχει μεγαλύτερη παραγωγή και πιθανότερη αποφυγή σφαλμάτων. Η αναγνώριση εικόνας αποτελεί αντικείμενο της μηχανικής όρασης. Η αρχή λειτουργίας της βασίζεται σε τρία στάδια.

Αρχικό στάδιο είναι η καταγραφή της εικόνας, συνήθως πραγματοποιείται με την χρήση μίας κάμερας, που στην συνέχεια ψηφιοποιείται. Ειδικότερα χωρίζει την εικόνα σε *pixels*, όπου στο καθένα από αυτά αντιστοιχεί μία τιμή ανάλογη του χρώματος. Η τιμή αυτή προσδιορίζεται από το ποσοστό του χρώματος του pixel, λαμβάνοντας σαν όριο την τιμή κατωφλιού που έχει καθοριστεί στο συγκεκριμένο σύστημα. Σαν ένα εύκολο παράδειγμα, ασπρόμαυρης εικόνας, λαμβάνουμε ως τιμή κατωφλιού ίση με 75%. Άρα αν το ποσοστό του μαύρου ενός pixel το ξεπερνάει, τότε παίρνει την τιμή 1 (Σχ. 2.2).



Σχήμα 2.2: Ψηφιοποίηση ασπρόμαυρης εικόνας σε pixels.

Επόμενο στάδιο είναι η ανάλυση πληροφοριών της εικόνας, που αναγνωρίζει και τμηματοποιεί σημαντικά στοιχεία της εικόνας και στέλνει τα δεδομένα στον εγκέφαλο του υπολογιστή, όπου εκεί θα πραγματοποιηθεί το τελικό στάδιο. Η τμηματοποίηση αυτή γίνεται είτε με την τεχνική του κατωφλιού που αναλύθηκε στην προηγούμενη παράγραφο, είτε με την τεχνική της αναγνώρισης της εξαγωγής χαρακτηριστικών του αντικειμένου, όπως είναι για παράδειγμα η περίμετρος, το εμβαδόν, το κέντρο βάρους, κλπ.

Τρίτο και τελευταίο στάδιο για να ολοκληρωθεί η διαδικασία της αναγνώρισης, είναι ο προσδιορισμός των αποτελεσμάτων ή αλλιώς η ερμηνεία της εικόνας. Υπάρχουν δύο τρόποι για να ολοκληρωθεί η αναγνώριση. Ο πρώτος τρόπος βασίζεται στην θεωρία των προτύπων, δηλαδή όταν γίνεται μία λήψη ενός

αντικειμένου, τότε ο εγκέφαλος συγκρίνει τα pixels του ληφθέντος αντικειμένου με τα pixels του προτύπου, που του έχει πληροφορήσει ο αλγόριθμος εκμάθησης στην μνήμη. Στη συνέχεια καταλήγει σε συμπεράσματα με χρήση στατιστικών μεθόδων. Μειονέκτημα αυτής της θεωρίας είναι ότι υπάρχει το ενδεχόμενο να μην αναγνωρίζει παραλλαγές του αντικειμένου, λόγω της ανομοιότητάς τους με το πρότυπο. Ο δεύτερος τρόπος βασίζεται στην θεωρία των χαρακτηριστικών του αντικειμένου, όπως μήκη γραμμών, σχήματα, εξαρτήματα. Για παράδειγμα, αν το αντικείμενο αναγνώρισης είναι ένα ζώο, τότε το σύστημα γνωρίζει ότι το ζώο έχει κεφάλι, σώμα, τέσσερα πόδια και ουρά και έτσι καταλήγει στην αναγνώρισή του. Η διάκριση του ζώου γίνεται με πιο εξειδικευμένα πρότυπα χαρακτηριστικών.

## **2.5.2 Χρησιμότητα Μηχανικής Όρασης**

Τα πεδία εφαρμογής της Μηχανικής όρασης είναι τρία, ο ποιοτικός έλεγχος μίας γραμμής παραγωγής, η αναγνώριση αντικειμένων και ψηφιακή καθοδήγηση του ρομπότ.

Στον ποιοτικό έλεγχο, η μηχανική όραση αναλαμβάνει τον ρόλο του επιθεωρητή. Για παράδειγμα σε μία κατασκευαστική βιομηχανία, με την βοήθεια συστημάτων όρασης, επιβλέπει τις κατεργασίες, με σκοπό τον σωστό προσδιορισμό διαστάσεων και τον εντοπισμό ατελειών ή ελαττωματικών προϊόντων.

Η αναγνώριση αντικειμένων στον χώρο της βιομηχανίας, καταμετρεί και ταξινομεί τα αντικείμενα ανάλογα με τις προδιαγραφές τους. Επίσης χρησιμοποιείται για την ανάγνωση του γραμμωτού κώδικα (bar-code) των προϊόντων. Μία εφαρμογή στην καθημερινότητα μας μπορεί να θεωρηθεί το ξεκλείδωμα μίας κινητής συσκευής με την αναγνώριση του προσώπου ή των ματιών του κατόχου της.

Τέλος στην ψηφιακή καθοδήγηση υπάρχουν καθοδηγούμενα ρομπότ, τα οποία είναι εκπαιδευμένα στο να μετακινούνται από μία τοποθεσία σε μία άλλη, αποφεύγοντας τα εμπόδια χωρίς να συγκρούονται (αντιλαμβάνονται τα εμπόδια). Συνήθως παίρνουν θέση σε αποθήκες για την μεταφορά μεγάλων όγκων προϊόντων.

## **2.6 ΠΕΡΙΣΣΟΤΕΡΕΣ ΕΦΑΡΜΟΓΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΣΤΗ ΡΟΜΠΟΤΙΚΗ**

Η τεχνητή νοημοσύνη δίνει μεγαλύτερη αποτελεσματικότητα στις μηχανές ρομπότ, καθώς και την επιθυμητή ακρίβεια που οι άνθρωποι να μην μπορούν να αποδώσουν. Παρακάτω θα αναφερθούν ορισμένα από τα πεδία εφαρμογής της ρομποτικής με τεχνητή νοημοσύνη με αντίστοιχα παραδείγματα.

### **2.6.1 Στην Ιατρική**

Η ιατρική αποτελεί έναν κλάδο εφαρμογής αυτοματοποιημένων μηχανών με τεχνητή νοημοσύνη που συμβάλλουν στην περίθαλψη ασθενών.

Παραδείγματα εφαρμογών είναι οι έξυπνες κάψουλες ενδοσκοπήσης, οι οποίες είναι λιγότερο επώδυνες για τους ασθενείς σε σύγκριση με τους παραδοσιακούς τρόπους. Πλέον η ρομποτική έχει ανεβάσει τον πήχη της επιτυχίας



στις χειρουργικές επεμβάσεις. Πιο συγκεκριμένα έχουν εφευρεθεί ρομπότ με τεχνητή νοημοσύνη που πραγματοποιούν λεπτές χειρουργικές επεμβάσεις σε σημεία του σώματος που ο χειρουργός γιατρός είναι αδύνατον να παρέμβει. Ένα άλλο παράδειγμα είναι η ευφυής προσθετική. Δημιουργήθηκε για να διευκολύνει τις κινήσεις σε άτομα χωρίς άκρα και έχει τη μορφή βραχιόνων ή ποδιών, με σύστημα ελέγχου παρόμοιο με τον φυσικό. Ακόμη ένα παράδειγμα είναι τα συστήματα παρακολούθησης, όταν οι ασθενείς πάσχουν από ασθένειες που χρειάζονται επίβλεψη και χρειάζονται άμεση παρέμβαση ή επικοινωνία με τον γιατρό, σε τυχόν κρίσιμες καταστάσεις.

Ο τομέας της υγείας είναι ένα πεδίο όπου τα ρομπότ με τεχνητή νοημοσύνη βρίσκονται ακόμη σε πρώιμο επίπεδο και η ανθρωπότητα είναι αισιόδοξη σε αυτό το κομμάτι της τεχνολογίας για το τι της επιφυλάσσει στο μέλλον.

### 2.6.2 Στη Γεωργία

Επόμενος κλάδος εφαρμογής είναι η γεωργία. Μέσω αυτοματοποιημένων μηχανών, οι αγρότες καταφέρνουν να αυξήσουν την παραγωγικότητα και λαμβάνουν βοήθεια για αποτελεσματικότερη καλλιέργεια. Παραδείγματα είναι το ψέκασμα φυτοφαρμάκων, το αυτόματο πότισμα, ο εντοπισμός βλαβερών ζωντανών οργανισμών που καταστρέφουν την υγεία των φυτών, καθώς και η συγκομιδή. Στην Εικ. 2.5 παρουσιάζονται μηχανήματα ρομπότ με τεχνητή νοημοσύνη που βρίσκουν θέση στον τομέα της γεωργίας.



**Εικόνα 2.5:** Εφαρμογές ρομποτικής με τεχνητή νοημοσύνη στην γεωργία.

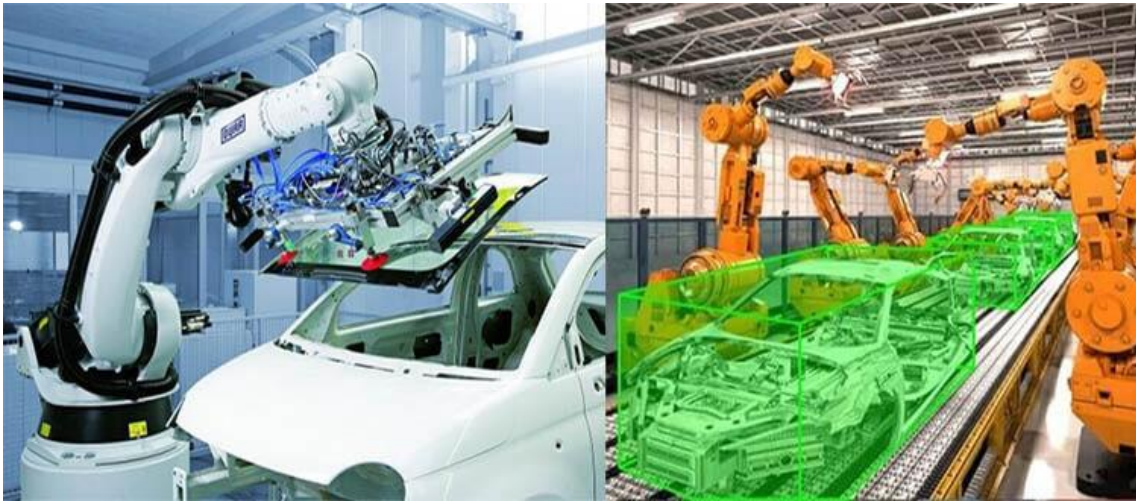
(Πηγή: <https://medium.com/vsinghbisen/ai-in-robotics-use-of-artificial-intelligence-in-robotics-726a4e9ade18>)

### 2.6.3 Στην Αυτοκινητοβιομηχανία

Στον τομέα της αυτοκινητοβιομηχανίας υπάρχει τεράστια ποικιλία από ρομπότ, διότι απαιτείται ταχύτατος ρυθμός παραγωγής και μόνο η δυναμικότητα του ανθρώπινου προσωπικού δεν αρκεί. Το καθένα από αυτά τα ρομπότ διαφέρει στα καθήκοντα του.

Τυχαίο παράδειγμα είναι τα αυτόματα μηχανήματα κατεργασίας υλικού, τα οποία έχουν προγραμματιστεί για να διαμορφώνουν εξαρτήματα αυτοκινήτου και στοιχεία των μηχανών.

Ένα ακόμη παράδειγμα είναι οι διεργασίες της συναρμολόγησης, της βαφής και της συσκευασίας του οχήματος (Εικ.2.6). Τα κατάλληλα προγραμματισμένα ρομπότ, με την μορφή βραχίονα, φημίζονται για την ευελιξία τους και την άριστη αποτελεσματικότητά τους. Χάρης αυτά πλέον, στο μέρος της βαφής, το χρώμα είναι ομοιόμορφο λόγω της επαναληψιμότητά τους, αλλά και μειώνει την άσκοπη κατανάλωση υλικού βαφής. Παράλληλα, στο μέρος της συναρμολόγησης, οι βραχίονες εκτελούν εργασίες όπως η τοποθέτηση των εξαρτημάτων στον σκελετό του αμαξώματος και στη συνέχεια η συναρμογή τους, μέσω συγκολλήσεων ή με τις κατάλληλες βίδες.



**Εικόνα 2.6:** Συνεργασία ρομποτικών βραχιόνων για την παραγωγή αυτοκινήτων.  
(Πηγή: <https://medium.com/vsinghbisen/ai-in-robotics-use-of-artificial-intelligence-in-robotics-726a4e9ade18>)



## 3. ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ

### 3.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΡΥΘΟΝ

Η γλώσσα προγραμματισμού *Python*, σύμφωνα με τον *Δείκτη Δημοφιλίας Προγραμματιστικών Γλωσσών (Popularity of Programming Languages Index)*, είναι η πιο χρησιμοποιούμενη και δημοφιλέστερη γλώσσα προγραμματισμού παγκοσμίως.

Εφευρέθηκε στο Άμστερνταμ από τον Guido van Rossum του Εθνικού Ινστιτούτου Έρευνών για εφαρμογές μαθηματικών και πληροφορικής όπου και κυκλοφόρησε για πρώτη φορά το 1991.

Η Python αναπτύχθηκε ταχύτατα και στην σημερινή εποχή διακρίθηκε ως η πιο δημοφιλής γλώσσα στην εκπαιδευτική και επιστημονική πληροφορική. Ορισμένα βασικά αίτια της διάκρισης αυτής είναι τα εξής:

- Η σύνταξή της και η εκμάθησή της είναι ευκολότερη σε αρχάριους, σε αντίθεση με άλλες γλώσσες όπως C, C++, Java, κλπ. Ακόμη και οι έμπειροι προγραμματιστές ενημερώνονται ευκολότερα και γρηγορότερα.
- Ευκολότερη είναι επίσης η ανάγνωση του κώδικα και των αποτελεσμάτων.
- Με την εισαγωγή βιβλιοθηκών ανοικτού κώδικα από τρίτους, η Python βοηθάει τους προγραμματιστές, έτσι ώστε να ολοκληρώνουν εργασίες (μεγάλης δυσκολίας) σε λιγότερο χρόνο και με μικρότερο όγκο προγράμματος. Δηλαδή οι προγραμματιστές γίνονται παραγωγικότεροι, σε σύγκριση με όταν χρησιμοποιούν άλλες γλώσσες.
- Αναπτύσσει ιστότοπους.
- Η απόδοση της βελτιώνεται με πολλούς τρόπους.
- Δημιουργεί εφαρμογές, από την πιο απλή, έως την πιο πολύπλοκη με μεγάλο αριθμό χρηστών, όπως είναι για παράδειγμα το Facebook.
- Η ραγδαία ανάπτυξη της τεχνητής νοημοσύνης, η οποία ειδικά σχετίζεται με την επιστήμη δεδομένων, κάνει ακόμη πιο δημοφιλή την Python.

Και πολλά ακόμη αίτια.

Για να πραγματοποιηθεί η γραφή και τα αποτελέσματα ενός προγράμματος, είναι απαραίτητο ένα λογισμικό που να αναπτύσσει εφαρμογές από αντίστοιχους κώδικες. Αυτό το λογισμικό ονομάζεται και *διερμηνευτής* ή *ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment)*. Αντίστοιχα λογισμικά που χρησιμοποιούνται αποκλειστικά για προγράμματα σε python είναι το *Python IDLE*, το *JupyterLab*, κλπ, τα οποία μας δίνουν την δυνατότητα για παράδειγμα, να

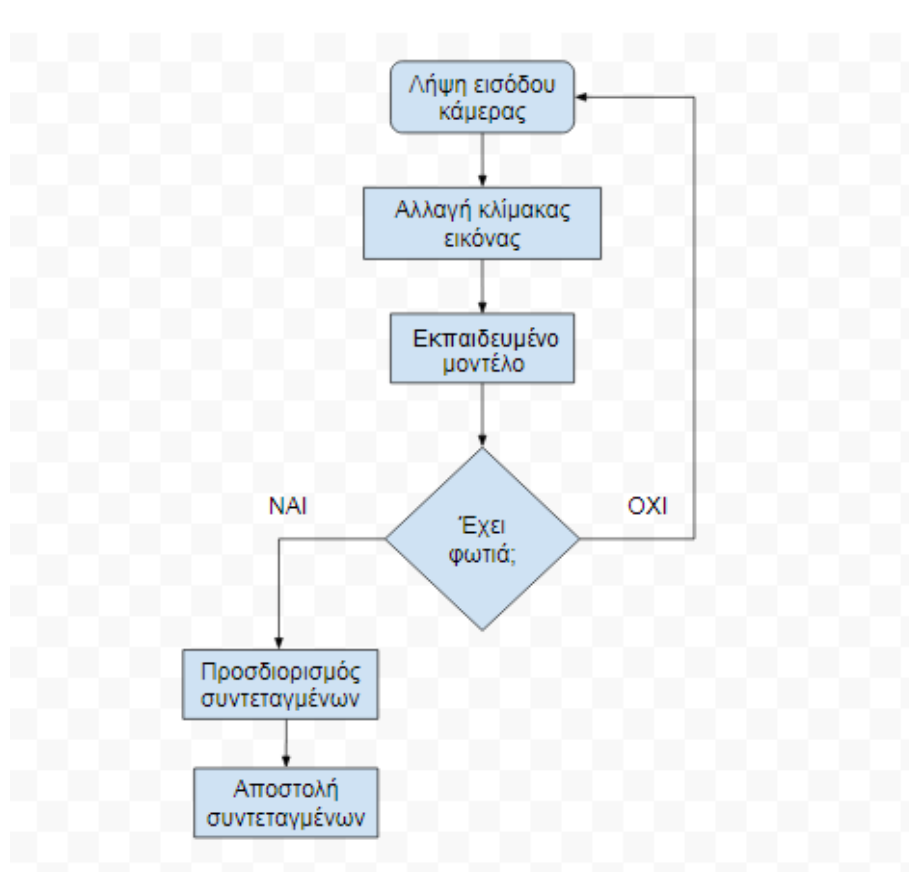
επεξεργαστούμε προγράμματα, να αποθηκεύσουμε σε αρχεία κ.ά. Το JupyterLab περιλαμβάνει τα Τετράδια Jupyter (*Jupyter Notebooks*), τα οποία επιτρέπουν να εισάγουμε εικόνες, βίντεο, ήχο, κλπ.

## 3.2 ΠΑΡΑΔΕΙΓΜΑ ΚΩΔΙΚΑ ΡΥΘΜΩΝ ΓΙΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Το πρόγραμμα που θα ακολουθήσει, εκτελεί γενικές εντολές στην Python, καθώς και εντολές που αναφέρονται αποκλειστικά στα ΤΝΔ. Το πρόγραμμα αυτό θα αποτελεί το μοντέλο εκμάθησης ενός drone με τεχνητή νοημοσύνη, που με την όρασή του θα επιβλέπει το εξωτερικό του περιβάλλον και θα είναι εκπαιδευμένο στο να διακρίνει την εικόνα της φωτιάς. Μετά από την ολοκλήρωση της εκμάθησης, το μοντέλο θα είναι έτοιμο για να περιπολεί δάση ή μέρη με υψηλό κίνδυνο πυρκαγιάς και θα αναγνωρίζει αν στο μέρος αυτό υπάρχει πυρκαγιά ή όχι. Αν εντοπίσει πυρκαγιά, τότε στο drone θα υπάρχει ήδη ένας επιπλέον αλγόριθμος (εν ακολουθία του παρακάτω), που θα ενημερώνει άμεσα τις αρχές της πυροσβεστικής για την ύπαρξη της πυρκαγιάς αλλά και την τοποθεσία της με τις ακριβείς συντεταγμένες της, με την βοήθεια ενσωματωμένου GPS που θα έχει το drone. Ο επιπλέον αυτός αλγόριθμος διαφοροποιείται ανάλογα με το λογισμικό του εκάστοτε drone.

### 3.2.1 Διάγραμμα Ροής

Όλοι οι κώδικες, οποιασδήποτε γλώσσας προγραμματισμού, μπορούν να αναπαρασταθούν από *Διαγράμματα Ροής*. Τα Διαγράμματα Ροής αποτυπώνουν την διαδικασία του κώδικα με την μορφή σχημάτων (που συμβολίζουν τα δεδομένα), αλλά και με βέλη (που συμβολίζουν την ροή των δεδομένων). Έχουν στόχο την ευκολότερη ανάγνωση του προγράμματος και συμβάλλουν στο να βρεθεί λύση σε προκύπτον πρόβλημα με περισσότερη ευκολία. Στην συγκεκριμένη περίπτωση το Διάγραμμα Ροής του συνολικού προγράμματος εμφανίζεται παρακάτω στο Σχ. 3.1, συμπεριλαμβάνοντας και το μέρος του επιπλέον αλγόριθμου αποστολής των συντεταγμένων της πυρκαγιάς.



**Σχήμα 3.1:** Διάγραμμα Ροής προγράμματος αναγνώρισης πυρκαγιάς και εντοπισμού των συντεταγμένων της.

### 3.2.2 Μοντέλο εκμάθησης ρομπότ αναγνώρισης εικόνας με πρόγραμμα Python

Το πρόγραμμα με τίτλο «**Ανίχνευση δασικών πυρκαγιών**» αναπτύχθηκε σε Google Colab Notebook με Python 3.7.15 και με τις εξής βιβλιοθήκες:

- tensorflow==2.9.2, για την εκπαίδευση του μοντέλου
- matplotlib==3.2.2, για την προβολή εικόνων και διαγραμμάτων
- os, για την δημιουργία ροής αρχείων

Σε αρχικό στάδιο λαμβάνονται δεδομένα από βιβλιοθήκες και ορίζονται οι τιμές των μεταβλητών:

```

import os

from matplotlib import pyplot as plt

import tensorflow as tf
from tensorflow.keras.utils import plot_model
from tensorflow.math import count_nonzero
from tensorflow.keras import Sequential
from tensorflow.keras.models import load_model
  
```

```

from tensorflow.keras.losses import SparseCategoricalCrossentropy
from tensorflow.keras.layers import Flatten, Dense, Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator

im_height = 64 #ύψος εικόνας
im_width = 64 #πλάτος εικόνας

batch_size = 64 #για εκπαίδευση του μοντέλου βλέπει 64 εικόνες
την φορά

base_path = '/content/drive/MyDrive/forest_fire_detector/dataset/forest_fire'
train_path = os.path.join(base_path, 'Training and Validation'
)
test_path = os.path.join(base_path, 'Testing')

```

Στην συνέχεια δημιουργούνται γεννήτριες εικόνων από τα αρχεία που είναι ήδη αποθηκευμένα στον υπολογιστή και προορίζονται για την εισαγωγή τους στο πρόγραμμα για την επιτυχία της εκμάθησης (*dataset*). Οι γεννήτριες αυτές θα στέλνουν εικόνες στο μοντέλο, κάνοντας διάφορες μετατροπές. Ειδικότερα, στέλνουν εικόνες για την εκπαίδευση του μοντέλου, για επικαιροποίηση (*Validation*) και για τον τελικό έλεγχο του εκπαιδευμένου μοντέλου.

Το *Validation* θεωρείται μία από τις πιο αποτελεσματικές μεθόδους εκμάθησης των ΤΝΔ που χρησιμοποιείται για έλεγχο της ακριβείας, όταν το μοντέλο πρόκειται να προβλέψει για κάτι όπως και στην περίπτωση μας. Πρακτικά, από την δειγματοληψία των δεδομένων, αποσπάται ένα μικρό τμήμα, ενώ το μοντέλο μαθαίνει με το υπόλοιπο και μεγαλύτερο τμήμα. Το μικρό τμήμα χρησιμοποιείται στο τέλος για να ελέγξει την αξιοπιστία της πρόβλεψης που έκανε το μοντέλο. Χρησιμοποιείται σε περιπτώσεις μη διαχωρισμού των δεδομένων (εκπαίδευσης, ελέγχου) ή όταν είναι αβέβαιο το πλήθος των δεδομένων που απαιτούνται για την αποφυγή της υπερπροσαρμογής (*Overfitting*).

Το *overfitting* συμβαίνει όταν το μοντέλο είναι πολύπλοκο ανάλογα με τα δεδομένα εκμάθησής του. Αν παραληφθεί και εισαχθούν δεδομένα που ταυτίζονται με τα δεδομένα εκμάθησης, τότε το μοντέλο θα κάνει άριστες προβλέψεις, εκτός από τα δεδομένα που δεν έχει ξαναδεί.

```

data_generator = ImageDataGenerator(
    rescale=1./255, #διαιρώ όλα τα rgb values με το 255 για να έχουν την τιμή μεταξύ 0 και 1
    shear_range=0.3,
    zoom_range=0.4,
    width_shift_range=0.0,
    height_shift_range=0.0,
    horizontal_flip=True,
    validation_split=0.3)

test_datagen = ImageDataGenerator(rescale=1./255)

```

```

train_generator = data_generator.flow_from_directory(
    train_path,
    target_size=(im_height, im_width),
    batch_size=batch_size,
    class_mode='binary', #δυναδικό, δηλαδή 2 αποτελέσματα
    shuffle=True, #τυχαία σειρά στις εικόνες
    seed=42, #ίδια τυχαία σειρά κάθε φορά
    subset='training')

validation_generator = data_generator.flow_from_directory(
    train_path,
    target_size=(im_height, im_width),
    batch_size=batch_size,
    class_mode='binary',
    subset='validation')

test_generator = test_datagen.flow_from_directory(
    test_path,
    target_size=(im_height, im_width),
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False)

images, labels = next(train_generator)
fig, axes = plt.subplots(4, 4, figsize = (16, 16))
for (image, label, ax) in zip(images, labels, axes.flatten()):
    ax.imshow(image)
    ax.set_title('Not Fire' if label else 'Fire')
    ax.axis('off')

```

Στην συνέχεια ακολουθούν οι εντολές για την δημιουργία και την αποτύπωση του μοντέλου του ΤΝΔ. Τα είδη των στρώματων που χρησιμοποιούνται είναι οι εξής:

- Conv2D (*Convolution Layer* ή *Στρώμα Συνέλιξης*), συλλέγει ορισμένα-χρήσιμα χαρακτηριστικά των εικόνων από μικρές περιοχές τους (*πυρήνες*).
- MaxPooling2D (*Στρώμα Συγκέντρωσης*), είναι ένα στρώμα 2x2 που μικραίνει την κλίμακα της εικόνας, διατηρώντας τα μέγιστα και πιο σημαντικά χαρακτηριστικά.
- Flatten, μετατρέπει το τρισδιάστατο αποτέλεσμα του προηγούμενου στρώματος σε ένα μονοδιάστατο.
- Dense (*Πυκνό Στρώμα*), δέχεται μόνο μονοδιάστατες εισόδους. Αντιστοιχεί τα χαρακτηριστικά-ψηφία με την ανάλογη εικόνα. Στην συγκεκριμένη περίπτωση δημιουργεί 64 νευρώνες που μαθαίνουν από τους 2304 του προηγούμενου, ενώ στο τέλος καταλήγει σε έναν μόνο νευρώνα που αντιστοιχεί στην πιθανότητα να περιλαμβάνει φωτιά η λήψη.

```

model = Sequential([
    # 1st Convolutional Block-Στρώμα εισόδου
    Conv2D(32, (3, 3), activation='relu', input_shape=(im_height, im_width, 3)), MaxPooling2D((2, 2)),

```



```

# 2nd Convolutional Block-Κρυφό στρώμα
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D((2, 2)),
# 3rd Convolutional Block-Στρώμα εξόδου
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D((2, 2)),
# Dense Layers
Flatten(),
Dense(64, activation='relu'),
Dense(1, Dense(1, activation='sigmoid'))
]
)

model.compile(optimizer='Adam', #μου λέει με ποιον τρόπο κάνει
update τα βάρη του μοντέλου
              loss='binary_crossentropy', #με ποιον τρόπο θα
μετράει τις απώλειες
              metrics=['accuracy']) #ποιο metric θα μετράει γι
α να βελτιώνεται

model.summary()

plot_model(model)

```

Επόμενο βήμα είναι η εκπαίδευση του μοντέλου στο να ξεχωρίζει στο αν η εικόνα περιλαμβάνει φωτιά ή όχι:

```

callbacks = [
    EarlyStopping(monitor='val_loss',
                  patience=5,
                  verbose=0,
                  restore_best_weights=False)
] #Αν για 5 συνεχόμενους γύρους δεν βελτιωθούν τα
σφάλματα του μοντέλου στο validation set τότε σταματάει η διαδ
ικασία. Αυτό το κάνουμε για να αποφύγουμε το overfitting.

history = model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=50)

model.save('fire_detection_model')

```

Ακόμη ένα βήμα είναι οι εντολές εμφάνισης των διαγραμμάτων της διαδικασίας εκπαίδευσης του μοντέλου, αρχικά της ακρίβειας και ύστερα των απωλειών:

```

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')

```

```
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

Τέλος, γίνεται η αξιολόγηση του μοντέλου που εφευρέθηκε:

```
scores = model.evaluate(test_generator)
y_pred = model.predict(test_generator).flatten().round()
y_true = test_generator.classes.astype('float32')

TP = sum(y_pred * y_true == 1) #υπάρχει φωτιά --
    προέβλεψε ότι υπάρχει φωτιά
TN = sum((1-y_pred) * (1-y_true) == 1) #δεν υπάρχει φωτιά --
    προέβλεψε ότι δεν υπάρχει φωτιά
FP = sum(y_pred * (1-y_true) == 1) #δεν υπάρχει φωτιά --
    προέβλεψε ότι υπάρχει φωτιά
FN = sum((1-y_pred) * y_true == 1) #υπάρχει φωτιά --
    προέβλεψε ότι δεν υπάρχει φωτιά

accuracy = scores[1] #πόσες φορές έκανε σωστή πρόβλεψη
loss = scores[0] #σφάλμα μοντέλου
precision = TP / (TP + FP) #πόσες φορές προέβλεψε σωστά ότι υπ
    άρχει φωτιά
recall = TP / (TP + FN) #από όλες τις φωτιές, πόσες από αυτές
    προέβλεψε σωστά
specificity = TN / (TN + FP) #πόσες φορές προέβλεψε σωστά ότι
    δεν υπάρχει φωτιά
f1 = 2 * precision * recall / (precision + recall) #αρμονικός
    μέσος μεταξύ precision και recall

print('Model Results on Test Data')
print(f'* Accuracy: {round(100*accuracy,2)}%')
print(f'* Loss: {round(loss,4)}')
print(f'* Precision: {round(100*precision,2)}%')
print(f'* Recall: {round(100*recall,2)}%')
print(f'* Specificity: {round(100*specificity,2)}%')
print(f'* F1: {round(100*f1,2)}%')
```

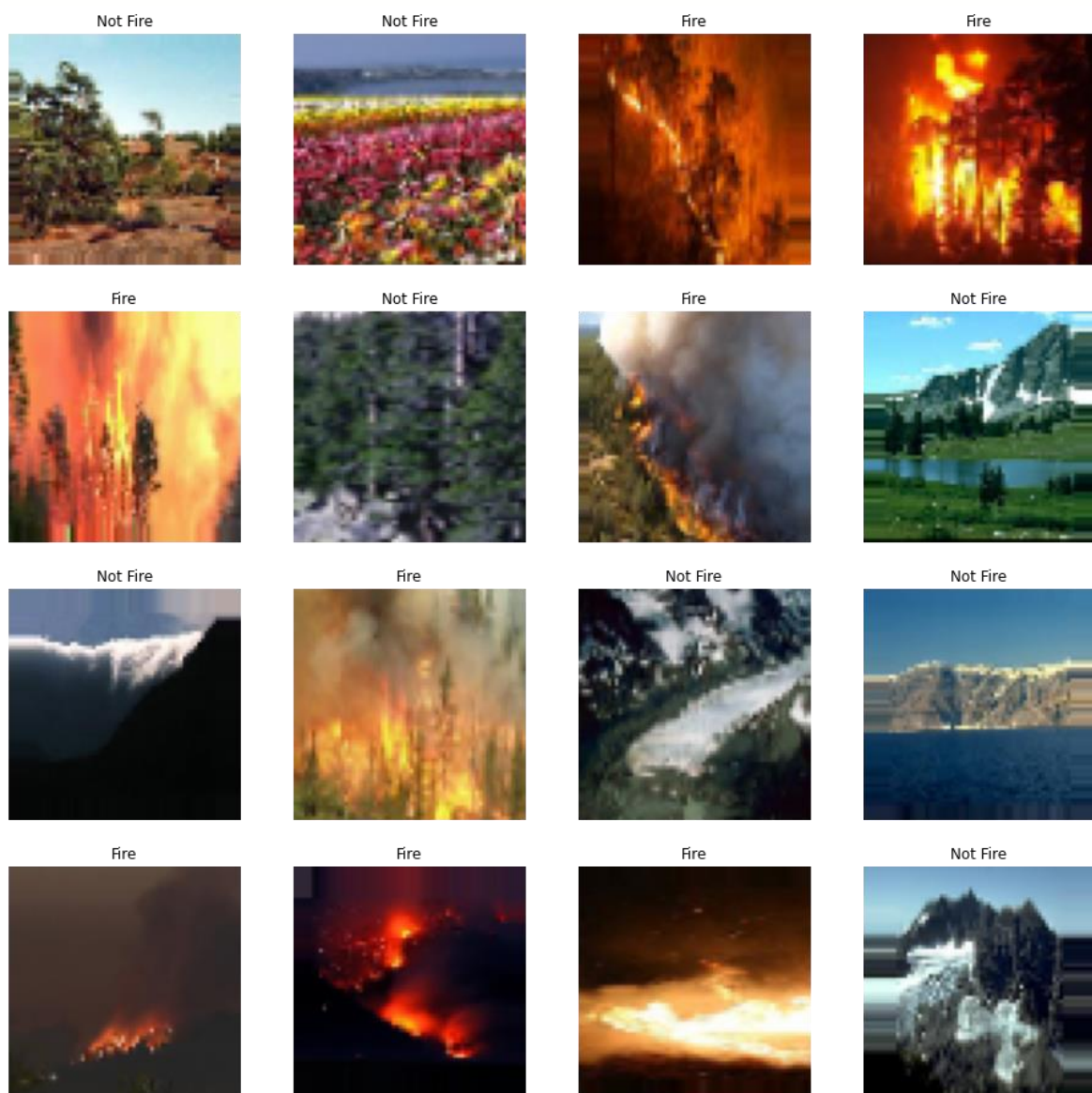
### 3.2.3 Εκτέλεση προγράμματος του μοντέλου εκμάθησης ρομπότ αναγνώρισης εικόνας και λήψη συμπερασμάτων

Παρακάτω παρουσιάζονται τα αποτελέσματα που προέκυψαν από τους αλγορίθμους που προηγήθηκαν. Αρχικά, λήφθηκαν τα παρακάτω δεδομένα, που δείχνουν τον αριθμό των εικόνων που χρησιμοποιήθηκαν από το dataset, αλλά και ότι διαχωρίζονται σε 2 κατηγορίες (υπάρχει φωτιά – δεν υπάρχει φωτιά). Πρώτα εμφανίζεται το πλήθος των εικόνων που έγιναν εισαγωγή στο πρόγραμμα για εκπαίδευση του μοντέλου, μετά για το validation και ύστερα για το τελικό τεστ. Ακόμη εμφανίζεται ένα πλήθος από τις εικόνες που αναφέρθηκαν (Εικ. 3.1).:

Found 1283 images belonging to 2 classes.

Found 549 images belonging to 2 classes.

Found 68 images belonging to 2 classes.

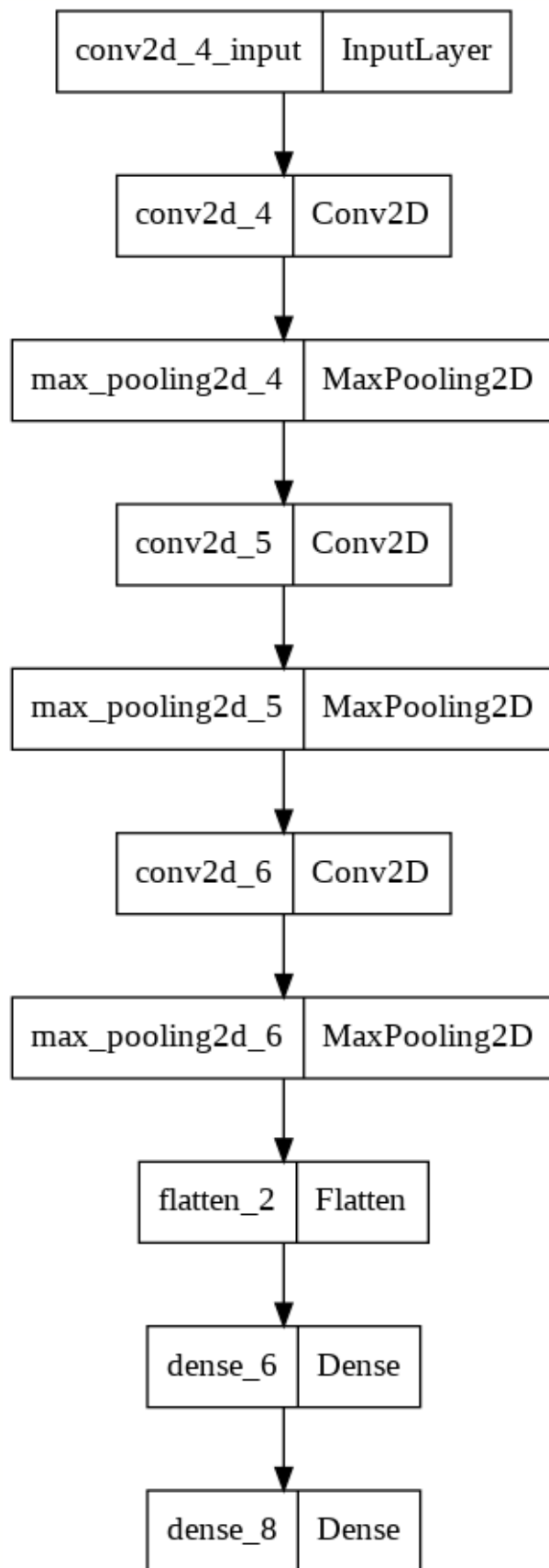


**Εικόνα 3.1:** Φωτογραφίες εκπαίδευσης μοντέλου ΤΝΔ που συμπεριλαμβάνουν ή δεν συμπεριλαμβάνουν πυρκαγιά.

Σε δεύτερη φάση αποτυπώνονται σε μορφή πίνακα οι πληροφορίες για την αρχιτεκτονική του μοντέλου του ΤΝΔ, ενώ στο Σχ. 3.2 απεικονίζονται τα στρώματα του δικτύου. Η πρώτη στήλη δείχνει το είδος του στρώματος, η δεύτερη στήλη τις διαστάσεις του αποτελέσματος που εξάγει το κάθε στρώμα, ενώ η τελευταία στήλη δείχνει το πλήθος των παραμέτρων της εκμάθησης του κάθε στρώματος.

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_5 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_6 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten_2 (Flatten)	(None, 2304)	0
dense_6 (Dense)	(None, 64)	147520
dense_8 (Dense)	(None, 1)	67
=====		
Total params: 203,907		
Trainable params: 203,907		
Non-trainable params: 0		



**Σχήμα 3.2:** Αποτύπωση του μοντέλου του ΤΝΔ κατά την εκτέλεση του παραπάνω αλγορίθμου.

Ακολουθούν οι τιμές, 50 επαναλήψεων, της ακρίβειας και των απωλειών που λήφθηκαν από την εκμάθηση του μοντέλου στο να αναγνωρίζει την φωτιά για κάθε επανάληψη. Σε κάθε επανάληψη, το μοντέλο εισάγει τις εικόνες από το dataset, οι οποίες χωρίζονται σε 21 μέρη των 64 εικόνων. Επομένως οι τιμές αυτές αποτυπώνονται στο διάγραμμα ακρίβειας (model accuracy) (Σχ. 3.3) και στο διάγραμμα των σφαλμάτων-αποκλίσεων (model loss) (Σχ. 3.4) αντίστοιχα. Στα διαγράμματα, τα πορτοκαλί γραφήματα δείχνουν το validation set, ενώ τα μπλε γραφήματα δείχνουν τις τιμές κατά την διάρκεια της εκμάθησης:

```
Epoch 1/50
21/21 [=====] - 7s 311ms/step - loss: 0.4271
- accuracy: 0.8153 - val_loss: 0.3053 - val_accuracy: 0.8798
Epoch 2/50
21/21 [=====] - 6s 288ms/step - loss: 0.2231
- accuracy: 0.9127 - val_loss: 0.2494 - val_accuracy: 0.9107
Epoch 3/50
21/21 [=====] - 6s 288ms/step - loss: 0.2707
- accuracy: 0.8940 - val_loss: 0.2839 - val_accuracy: 0.9144
Epoch 4/50
21/21 [=====] - 6s 301ms/step - loss: 0.1700
- accuracy: 0.9345 - val_loss: 0.2090 - val_accuracy: 0.9235
Epoch 5/50
21/21 [=====] - 6s 288ms/step - loss: 0.1416
- accuracy: 0.9470 - val_loss: 0.1969 - val_accuracy: 0.9399
Epoch 6/50
21/21 [=====] - 6s 288ms/step - loss: 0.1228
- accuracy: 0.9602 - val_loss: 0.1873 - val_accuracy: 0.9435
Epoch 7/50
21/21 [=====] - 6s 287ms/step - loss: 0.1296
- accuracy: 0.9564 - val_loss: 0.2013 - val_accuracy: 0.9308
Epoch 8/50
21/21 [=====] - 6s 290ms/step - loss: 0.1322
- accuracy: 0.9493 - val_loss: 0.2132 - val_accuracy: 0.9290
Epoch 9/50
21/21 [=====] - 6s 287ms/step - loss: 0.1124
- accuracy: 0.9641 - val_loss: 0.2350 - val_accuracy: 0.9326
Epoch 10/50
21/21 [=====] - 6s 288ms/step - loss: 0.0975
- accuracy: 0.9719 - val_loss: 0.2223 - val_accuracy: 0.9290
Epoch 11/50
21/21 [=====] - 6s 289ms/step - loss: 0.1078
- accuracy: 0.9595 - val_loss: 0.1815 - val_accuracy: 0.9454
Epoch 12/50
21/21 [=====] - 6s 291ms/step - loss: 0.0926
- accuracy: 0.9719 - val_loss: 0.2245 - val_accuracy: 0.9271
Epoch 13/50
21/21 [=====] - 6s 298ms/step - loss: 0.1088
- accuracy: 0.9595 - val_loss: 0.1818 - val_accuracy: 0.9417
Epoch 14/50
21/21 [=====] - 7s 344ms/step - loss: 0.0836
- accuracy: 0.9751 - val_loss: 0.1445 - val_accuracy: 0.9636
```

Epoch 15/50  
21/21 [=====] - 6s 300ms/step - loss: 0.0739  
- accuracy: 0.9766 - val\_loss: 0.1548 - val\_accuracy: 0.9508  
Epoch 16/50  
21/21 [=====] - 6s 286ms/step - loss: 0.0556  
- accuracy: 0.9805 - val\_loss: 0.2064 - val\_accuracy: 0.9381  
Epoch 17/50  
21/21 [=====] - 6s 290ms/step - loss: 0.0816  
- accuracy: 0.9751 - val\_loss: 0.1580 - val\_accuracy: 0.9490  
Epoch 18/50  
21/21 [=====] - 6s 287ms/step - loss: 0.0616  
- accuracy: 0.9790 - val\_loss: 0.1796 - val\_accuracy: 0.9472  
Epoch 19/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0644  
- accuracy: 0.9774 - val\_loss: 0.1513 - val\_accuracy: 0.9581  
Epoch 20/50  
21/21 [=====] - 6s 288ms/step - loss: 0.0515  
- accuracy: 0.9797 - val\_loss: 0.1596 - val\_accuracy: 0.9599  
Epoch 21/50  
21/21 [=====] - 6s 288ms/step - loss: 0.0408  
- accuracy: 0.9829 - val\_loss: 0.1431 - val\_accuracy: 0.9672  
Epoch 22/50  
21/21 [=====] - 6s 288ms/step - loss: 0.0526  
- accuracy: 0.9844 - val\_loss: 0.1682 - val\_accuracy: 0.9399  
Epoch 23/50  
21/21 [=====] - 6s 290ms/step - loss: 0.0544  
- accuracy: 0.9813 - val\_loss: 0.2349 - val\_accuracy: 0.9399  
Epoch 24/50  
21/21 [=====] - 6s 290ms/step - loss: 0.0474  
- accuracy: 0.9821 - val\_loss: 0.1669 - val\_accuracy: 0.9490  
Epoch 25/50  
21/21 [=====] - 6s 288ms/step - loss: 0.0365  
- accuracy: 0.9875 - val\_loss: 0.2125 - val\_accuracy: 0.9563  
Epoch 26/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0409  
- accuracy: 0.9860 - val\_loss: 0.1912 - val\_accuracy: 0.9472  
Epoch 27/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0348  
- accuracy: 0.9875 - val\_loss: 0.1705 - val\_accuracy: 0.9599  
Epoch 28/50  
21/21 [=====] - 6s 288ms/step - loss: 0.0341  
- accuracy: 0.9867 - val\_loss: 0.1572 - val\_accuracy: 0.9581  
Epoch 29/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0275  
- accuracy: 0.9906 - val\_loss: 0.1774 - val\_accuracy: 0.9617  
Epoch 30/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0359  
- accuracy: 0.9883 - val\_loss: 0.2485 - val\_accuracy: 0.9344  
Epoch 31/50  
21/21 [=====] - 7s 340ms/step - loss: 0.0353  
- accuracy: 0.9899 - val\_loss: 0.2065 - val\_accuracy: 0.9472

Epoch 32/50  
21/21 [=====] - 6s 285ms/step - loss: 0.0382  
- accuracy: 0.9852 - val\_loss: 0.1511 - val\_accuracy: 0.9672

Epoch 33/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0334  
- accuracy: 0.9891 - val\_loss: 0.1491 - val\_accuracy: 0.9654

Epoch 34/50  
21/21 [=====] - 6s 291ms/step - loss: 0.0318  
- accuracy: 0.9891 - val\_loss: 0.2414 - val\_accuracy: 0.9381

Epoch 35/50  
21/21 [=====] - 6s 291ms/step - loss: 0.0332  
- accuracy: 0.9875 - val\_loss: 0.1688 - val\_accuracy: 0.9581

Epoch 36/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0346  
- accuracy: 0.9860 - val\_loss: 0.1964 - val\_accuracy: 0.9490

Epoch 37/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0199  
- accuracy: 0.9930 - val\_loss: 0.2246 - val\_accuracy: 0.9599

Epoch 38/50  
21/21 [=====] - 6s 314ms/step - loss: 0.0256  
- accuracy: 0.9891 - val\_loss: 0.1700 - val\_accuracy: 0.9526

Epoch 39/50  
21/21 [=====] - 6s 295ms/step - loss: 0.0211  
- accuracy: 0.9914 - val\_loss: 0.1702 - val\_accuracy: 0.9563

Epoch 40/50  
21/21 [=====] - 6s 292ms/step - loss: 0.0260  
- accuracy: 0.9891 - val\_loss: 0.2530 - val\_accuracy: 0.9545

Epoch 41/50  
21/21 [=====] - 6s 312ms/step - loss: 0.0287  
- accuracy: 0.9914 - val\_loss: 0.2260 - val\_accuracy: 0.9581

Epoch 42/50  
21/21 [=====] - 6s 302ms/step - loss: 0.0160  
- accuracy: 0.9945 - val\_loss: 0.2021 - val\_accuracy: 0.9490

Epoch 43/50  
21/21 [=====] - 6s 293ms/step - loss: 0.0168  
- accuracy: 0.9922 - val\_loss: 0.2596 - val\_accuracy: 0.9563

Epoch 44/50  
21/21 [=====] - 6s 311ms/step - loss: 0.0200  
- accuracy: 0.9938 - val\_loss: 0.1927 - val\_accuracy: 0.9508

Epoch 45/50  
21/21 [=====] - 6s 290ms/step - loss: 0.0223  
- accuracy: 0.9899 - val\_loss: 0.1866 - val\_accuracy: 0.9617

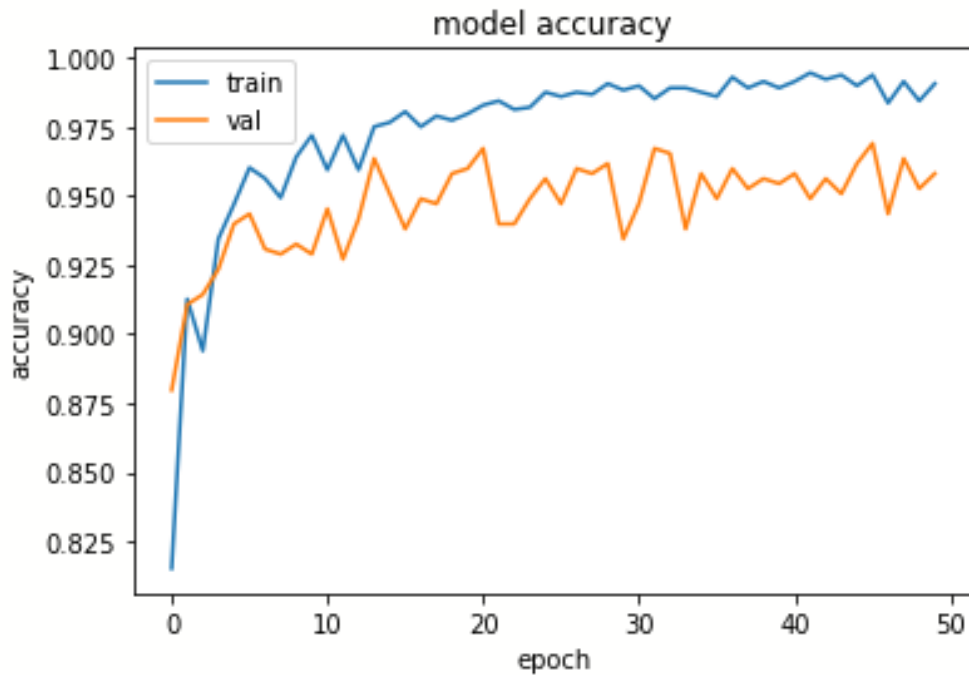
Epoch 46/50  
21/21 [=====] - 6s 290ms/step - loss: 0.0131  
- accuracy: 0.9938 - val\_loss: 0.2088 - val\_accuracy: 0.9690

Epoch 47/50  
21/21 [=====] - 6s 289ms/step - loss: 0.0428  
- accuracy: 0.9836 - val\_loss: 0.1831 - val\_accuracy: 0.9435

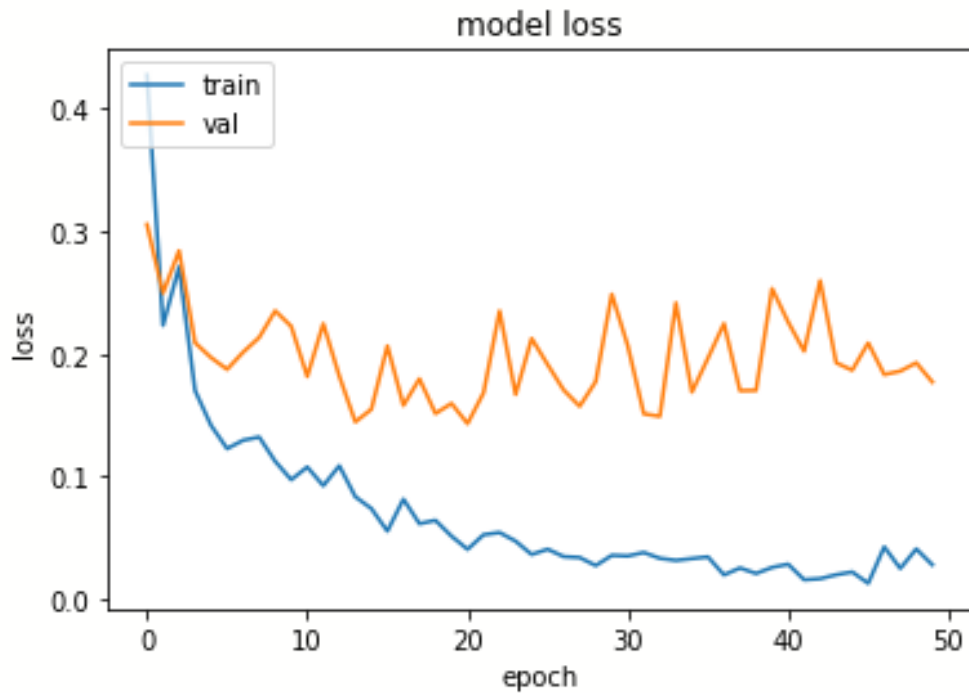
Epoch 48/50  
21/21 [=====] - 8s 379ms/step - loss: 0.0252  
- accuracy: 0.9914 - val\_loss: 0.1859 - val\_accuracy: 0.9636



Epoch 49/50  
21/21 [=====] - 7s 331ms/step - loss: 0.0411  
- accuracy: 0.9844 - val\_loss: 0.1926 - val\_accuracy: 0.9526  
Epoch 50/50  
21/21 [=====] - 6s 291ms/step - loss: 0.0281  
- accuracy: 0.9906 - val\_loss: 0.1770 - val\_accuracy: 0.9581



Σχήμα 3.3: Διάγραμμα ακρίβειας



Σχήμα 3.4: Διάγραμμα σφαλμάτων

Τέλος, γίνεται η λήψη στατιστικών και η αξιολόγηση του μοντέλου στο τεστ, μετά από την εισαγωγή των φωτογραφιών του τελικού τεστ σε 2 μέρη:

```
2/2 [=====] - 0s 37ms/step - loss: 0.1980 -  
accuracy: 0.9706  
Model Results on Test Data  
* Accuracy: 97.06%  
* Loss: 0.198  
* Precision: 97.83%  
* Recall: 97.83%  
* Specificity: 95.45%  
* F1: 97.83%
```

Συμπεραίνοντας, το μοντέλο έχει αποκτήσει 97,06% ακρίβεια (accuracy), το οποίο είναι και το πιο σημαντικό, καθώς και το σφάλμα του (loss) που είναι μόλις 0,198. Το ποσοστό ορθής πρόβλεψης ότι υπάρχει φωτιά είναι 97,83%, ενώ ότι δεν υπάρχει φωτιά 95,45%. Το ποσοστό επιτυχίας του τεστ για οποιαδήποτε πρόβλεψη βρίσκεται στο 97,83%. Τέλος ο αρμονικός μέσος μεταξύ precision και recall είναι 97,83%.



## 4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτό το κεφάλαιο θα γίνει η ανακεφαλαίωση της εργασίας και θα αναπτυχθούν τα συμπεράσματά της. Επίσης θα σχολιαστούν και τα συμπεράσματα που προκύπτουν από την εκτέλεση του προγράμματος εκμάθησης.

Αρχικά, όπως έχει αναφερθεί, η τεχνητή νοημοσύνη, η ρομποτική και γενικά η επεξεργασία δεδομένων είναι αντικείμενα με μικρή ιστορία, που βρίσκονται σε πρώιμο στάδιο. Ως εκ τούτου, η εξέλιξή τους και η επίδρασή τους σε όλες τις επιστήμες, πρόκειται να γιγαντωθεί από χρόνο σε χρόνο. Η εξέλιξη αυτή φυσικά βασίζεται και είναι ανάλογη με αυτήν των υπολογιστών και των λογισμικών.

Αντιλαμβάνοντας την λογική των ΝΔ, καταλαβαίνουμε ότι είναι ένα πολύπλοκο σύστημα, που για την ανάπτυξή τους λαμβάνουν θέση στοιχεία από πολλές διαφορετικές επιστήμες. Για την επιτυχή δημιουργία ενός πιστού αντίγραφου του βιολογικού νευρώνα ή ΝΔ, οφείλεται ο συνδυασμός των διαφόρων διαδικασιών-βημάτων που διασχίζει το σήμα, όπως το σύστημα όρασης, οι συναρτήσεις ενεργοποίησης εξόδου, η εκπαίδευση του δικτύου, κλπ. Όλο το νόημα των ΤΝΔ βρίσκεται στην εκπαίδευσή τους, δηλαδή στην σταδιακή βελτίωση και τη μέγιστη δυνατή εξάλειψη των σφαλμάτων της αναγνώρισης για τις επιθυμητές προβλέψεις που θα κάνει το μοντέλο.

Επιπλέον, γίνεται κατανοητό το ότι η χρήση της τεχνητής νοημοσύνης δεν γίνεται απαραίτητα για την βοήθεια των ανθρώπων και την αντιμετώπιση των προβλημάτων τους, που αυτός είναι και ο πρωταρχικός στόχος της, αλλά είναι εφικτό να χρησιμοποιηθεί εναντίον τους. Ακόμη συμπεραίνουμε ότι για να δημιουργηθεί ένα σύστημα τεχνητής νοημοσύνης, δεν χρειάζονται πολλά ή εξεζητημένα εξαρτήματα για την υλική συναρμολόγηση του. Το σημαντικότερο είναι να κωδικοποιηθούν σωστοί αλγόριθμοι με τις κατάλληλες εντολές, έτσι ώστε να δημιουργηθεί μία ορθή αντίληψη στο ρομπότ.

Ύστερα από την συγγραφή και την εκτέλεση του προγράμματος στο τρίτο κεφάλαιο της εργασίας, προέκυψαν συμπεράσματα που δείχνουν την επιτυχία της εκμάθησης του μοντέλου του δικτύου.

Στο τελευταίο στάδιο της εκτέλεσης φαίνονται τα στατιστικά αποτελέσματα του μοντέλου. Τα σημαντικότερα αποτελέσματα είναι η ακρίβεια της σωστής πρόβλεψης που έκανε το μοντέλο στην αναγνώριση της φωτιάς (accuracy), με 97,06% επιτυχία και το σφάλμα του (loss) που είναι μόλις 0,198.

Αυτά τα στατιστικά αποδεικνύουν ότι το μοντέλο είναι έτοιμο και ικανό να αναγνωρίσει από το περιβάλλον του, με την χρήση της κάμεράς του, την φωτιά σε οποιαδήποτε μορφή της (έκταση που καταλαμβάνει, σχήμα, κλπ) και σε οποιοδήποτε

μέρος και αν βρίσκεται. Αφού προσδιοριστεί και ο κώδικας για τον εντοπισμό των συντεταγμένων της τοποθεσίας που βρίσκεται η εστία της φωτιάς, τότε το drone θα είναι ένα έμπιστο μηχάνημα, έτοιμο να παραχωρηθεί στις αρμόδιες αρχές για την λειτουργία του.

Σε μια πιο γενική άποψη, το συμπέρασμα είναι ότι η τεχνητή νοημοσύνη κατάφερε να εκπληρώσει έναν από τους στόχους που επιδίωκε στον τομέα της όρασης των ρομπότ. Δηλαδή κατάφερε αποδεδειγμένα να δημιουργήσει την προσεγγιστική απομίμηση της ανθρώπινης όρασης-αντίληψης με μεγάλη επιτυχία.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Αργυράκης Π., *Νευρωνικά Δίκτυα και Εφαρμογές*, Τόμος Β', Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα, 2001
2. Γεωργούλη Α., *Τεχνητή Νοημοσύνη*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, Αθήνα, 2015
3. Γιαννάτσης Ι., Δεδούσης Β., Κανελλίδης Β., *Σύγχρονες Τεχνολογίες Κατασκευής με τη βοήθεια Η/Υ*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, 2015
4. Δημουλάς Χ., *Τεχνολογίες Συγγραφής και Διαχείρισης Πολυμέσων – Τεχνικές Μη Γραμμικής Αφήγησης στα Νέα Ψηφιακά Μέσα*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, Θεσσαλονίκη, 2015
5. Καμπουρλάζος Β., Παπακώστας Γ., *Εισαγωγή στην Υπολογιστική Νοημοσύνη*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, Καβάλα, 2015
6. Κιάτος Μ., *Αναγνώριση Αντικειμένων με Τεχνητή Όραση*, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη, 2015
7. Λέπουρας Γ., Αντωνίου Α., Πλατής Ν., Χαρίτος Δ., *Ανάπτυξη Συστημάτων Εικονικής Πραγματικότητας*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, Αθήνα, 2015
8. Μουστάκας Κ., Παλιόκας Ι., Τσακίρης Α., Τζοβάρας Δ., *Γραφικά και Εικονική Πραγματικότητα*, Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδόσεις, 2015
9. Πήτας Δ., *Τεχνητά Νευρωνικά Δίκτυα και Εφαρμογή (Neural Networks and Application)*, Πανεπιστήμιο Αιγαίου, Καρλόβασι, 2018
10. Τριανταφυλλίδης Θ., *Μέθοδοι Εκπαίδευσης Βαθιών Συνελικτικών Νευρωνικών Δικτύων στην Ταξινόμηση Εικόνων Προτύπων Προσώπου και Ανίχνευση/Ταυτοποίηση Αντικειμένων, μέσα από το λογισμικό MATLAB*, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη, 2018
11. Deitel P., Deitel H., Απόδοση: Τροχάνης Α., *Python για Προγραμματιστές*, Εκδόσεις Μ. Γκιούρδας, Αθήνα, 2019
12. Robert Bogue, *The Role of Artificial Intelligence in Robotics*, Industrial Robot, Vol. 41 No. 2, pp. 119-123, 2014
13. Perez JA, Deligianni F., Ravi D., Yang G., *Artificial Intelligence and Robotics*, Engineering and Physical Sciences Research Council (EPSRC) UK-RAS Network, London, 2017
14. <https://medium.com/vsinghbisen/ai-in-robotics-use-of-artificial-intelligence-in-robotics-726a4e9ade18>
15. <https://intelligenttransport.wordpress.com/2011/02/17/%CE%B7-%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE-%CF%83%CF%84%CE%B7->

- [%CE%B2%CE%B9%CE%BF%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%AF%CE%B1/](#)
16. <https://www.techspot.com/news/76824-intel-introduces-28-core-xeon-w-3175x-cpu.html>
  17. <https://www.cienciaedados.com/gpu-e-deep-learning/>
  18. <https://www.anandtech.com/show/7334/a-look-at-alteras-openc1-sdk-for-fpgas>
  19. [https://el.lambdageeks.com/robot-sensors-characteristics/?\\_ga=2.225153818.1099420019.1668123160-1431487164.1668123160#open](https://el.lambdageeks.com/robot-sensors-characteristics/?_ga=2.225153818.1099420019.1668123160-1431487164.1668123160#open)
  20. [https://el.lambdageeks.com/robotic-vision-important-features/?\\_ga=2.220189275.31742999.1668123492-1646903395.1668123492#](https://el.lambdageeks.com/robotic-vision-important-features/?_ga=2.220189275.31742999.1668123492-1646903395.1668123492#)
  21. <https://www.tensorflow.org/>
  22. [https://www.youtube.com/watch?v=aircArvvnKk&ab\\_channel=3Blue1Brown](https://www.youtube.com/watch?v=aircArvvnKk&ab_channel=3Blue1Brown)
  23. [https://www.youtube.com/watch?v=wQ8BIBpya2k&t=2s&ab\\_channel=sentdex](https://www.youtube.com/watch?v=wQ8BIBpya2k&t=2s&ab_channel=sentdex)
  24. [https://www.youtube.com/watch?v=WvoLTXljBYU&ab\\_channel=sentdex](https://www.youtube.com/watch?v=WvoLTXljBYU&ab_channel=sentdex)
  25. [https://www.youtube.com/watch?v=uqomO\\_BZ44g&ab\\_channel=WhenMathsMetCoding](https://www.youtube.com/watch?v=uqomO_BZ44g&ab_channel=WhenMathsMetCoding)
  26. [https://www.youtube.com/watch?v=bns5ELvbzVk&ab\\_channel=codebasics](https://www.youtube.com/watch?v=bns5ELvbzVk&ab_channel=codebasics)