



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ**

**Ανίχνευση προσχεδίων σε πίνακες  
ζωγραφικής με χρήση τεχνικών  
επεξεργασίας εικόνας και νευρωνικών  
δικτύων βαθιάς μάθησης**

Detection of Under Drawings in Paintings Using Image  
Processing Techniques and Deep Neural Networks

**Λυκούδη Δέσποινα**  
A.M. 18390151

**Εισηγητής: Νικόλαος Βασιλάς, Καθηγητής**

Αιγάλεω, Οκτώβριος 2023



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανίχνευση προσχεδίων σε πίνακες ζωγραφικής με χρήση τεχνικών επεξεργασίας  
εικόνας και νευρωνικών δικτύων βαθιάς μάθησης**

Η παρούσα διπλωματική εργασία παρουσιάστηκε από την

**Λυκούδη Δέσποινα**

18390151

την Πέμπτη, 12 Οκτωβρίου 2023

**Επιβλέπων Καθηγητής: Νικόλαος Βασιλάς**

**Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:**

Νικόλαος Βασιλάς,  
Καθηγητής

Γεώργιος Μπαρδής,  
Επίκουρος Καθηγητής

Χριστίνα Γεωργουλάκη,  
ΕΔΠ



## Δήλωση Συγγραφέα Διπλωματικής

Η κάτωθι υπογεγραμμένη Λυκούδη Δέσποινα του Κωνσταντίνου με αριθμό μητρώου 18390151 φοιτήτρια του Τμήματος Μηχανικών Πληροφορικής & Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών δηλώνω υπεύθυνα ότι:

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

Η Δηλούσα



Δέσποινα Λυκούδη



## Ευχαριστίες

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια σπουδών μου στο Πανεπιστήμιο Δυτικής Αττικής, στο τμήμα Μηχανικών Πληροφορικής και Υπολογιστών, κατά το έτος 2023. Η ολοκλήρωση της διπλωματικής εργασίας έγινε με την υποστήριξη και την καθοδήγηση του επιβλέπων καθηγητή Νικόλαου Βασιλά, τον οποίο θα ήθελα να ευχαριστήσω. Θέλω επίσης, να ευχαριστήσω τις φίλες μου, για την υποστήριξη και τη συμπαράσταση τους, σε όλη αυτή την περίοδο. Τέλος, θέλω να ευχαριστήσω πολύ την οικογένεια μου, και ιδιαίτερα τον αδελφό μου, οι οποίοι υπήρξαν πάντα ένα ανεκτίμητο στήριγμα για εμένα, πόσο μάλλον στη διάρκεια των σπουδών μου.





## Περίληψη

Στην παρούσα διπλωματική υλοποιείται η ανάπτυξη και εκπαίδευση ενός νευρωνικού δικτύου βαθιάς μάθησης για την ταξινόμηση των εικονοστοιχείων υπερφασματικών υπέρυθρων εικόνων μαζί με το χωρικό τους πλαίσιο είτε σε γραμμές προσχεδίων είτε στο φόντο. Το αποτέλεσμα θα είναι η δημιουργία ενός χάρτη εμπιστοσύνης για κάθε νέα υπερφασματική εικόνα. Στη συνέχεια, θα χρησιμοποιηθούν τεχνικές επεξεργασίας εικόνας και κατωφλίου, ώστε να τμηματοποιηθεί η εικόνα εισόδου, να προσδιοριστούν οι γραμμές του προσχεδίου και να δημιουργηθεί η αντίστοιχη δυαδική εικόνα του προσχεδίου. Η ανάπτυξη της εφαρμογής έγινε με χρήση του Matlab, που είναι αρκετά διαδεδομένο στον τομέα της Μηχανικής Μάθησης.

Λέξεις Κλειδιά :

Έργα Τέχνης, Προσχέδια, Επεξεργασία Εικόνας, Νευρωνικά Δίκτυα, Βαθιά Μάθηση

## **Abstract**

In this thesis an application is implemented, which develops and trains a deep learning neural network for classifying pixels of hyper spectral infrared images along with their spatial context either in line drawings or in the background. The result will be the generation of a confidence map for each new hyper spectral image. Image and thresholding techniques will then be used to segment the input image, identify the under drawings and generate the corresponding binary image. The development of this application was done using Matlab, which is quite widespread in machine learning.

Keywords:

Paintings, Underdrawings, Image Processing, Neural Networks, Deep Learning

## Περιεχόμενα

Δήλωση Συγγραφέα Διπλωματικής.....	iv
Περίληψη.....	viii
Abstract.....	ix
Κατάλογος Εικόνων.....	xii
Συνοτομογραφίες.....	xiii
1 Εισαγωγή.....	1
1.1 Κίνητρα.....	2
2 Θεωρητικό Υπόβαθρο.....	4
2.1 Επεξεργασία Εικόνας (Image Processing).....	4
2.1.1 Ανίχνευση Ακμών (Edge Detection).....	5
2.1.2 Κατωφλίωση (Thresholding).....	7
2.2 Όραση Υπολογιστών (Computer Vision).....	8
2.3 Μηχανική Μάθηση (Machine Learning).....	9
2.3.1 Νευρωνικά Δίκτυα (Neural Network).....	11
2.3.2 Πολυστρωματικά Νευρωνικά Δίκτυα (Multilayer Neural Networks).....	14
2.3.3 Συνελεκτικά Νευρωνικά Δίκτυα (Convolutional Neural Network).....	17
2.3.4 Μηχανική Μάθηση στα έργα τέχνης.....	18
2.4 Βαθιά Μάθηση (Deep Learning).....	18
2.4.1 Εφαρμογές Βαθιάς Μάθησης.....	19
2.5 Μηχανική Μάθηση και Βαθιά Μάθηση.....	21
3 Περιγραφή της Γλώσσας Matlab.....	23
3.1 Matlab και Μηχανική Μάθηση.....	24
4 Υπερφασματικά Δεδομένα.....	25
5 Προ-επεξεργασία δεδομένων.....	27
5.1 Ανίχνευση Γραμμών (Line Detection) με Hough Transform.....	27
5.1.1 Μεθοδολογία μετασχηματισμού Hough για εντοπισμό γραμμών.....	28
5.2 Δοκιμές.....	29

5.2.1 Δοκιμή 1 <sup>η</sup> .....	29
5.2.2 Δοκιμή 2 <sup>η</sup> .....	31
5.2.3 Δοκιμή 3 <sup>η</sup> .....	32
5.2.4 Δοκιμή 4 <sup>η</sup> .....	33
5.3 Εξαγωγή Patches.....	34
6 Πειραματικό Μέρος.....	36
7 Συμπεράσματα.....	41
Βιβλιογραφία – Αναφορικές – Διαδικτυακές Πηγές.....	42
Παράρτημα I.....	46
I.I.....	46
I.II.....	46
I.III.....	47
I.IV.....	47
I.V.....	48
I.VI.....	50
I.VII.....	50
I.VIII.....	50
I.IX.....	51
I.X.....	51

## Κατάλογος Εικόνων

Εικόνα 1: Τα τρία επίπεδα της μηχανικής μίας εικόνας [3] .....	5
Εικόνα 2: Sobel Convolutional Kernels .....	7
Εικόνα 3: Μηχανική εικόνας και συναφή θέματα [3].....	9
Εικόνα 4: Μηχανική Μάθηση συγκριτικά με τον συνηθισμένο Προγραμματισμό [10] .....	9
Εικόνα 5: Διαφορετικές κατηγορίες Μηχανικής Μάθησης και αλγόριθμοι [11] .....	11
Εικόνα 6: Στρώματα Τεχνητών Νευρωνικών Δικτύων [12].....	12
Εικόνα 7: Τοπολογία MFL νευρωνικού δικτύου [39].....	15
Εικόνα 8: Μονοπολικό σιγμοειδές, συνάρτηση ενεργοποίησης [36].....	16
Εικόνα 9: multi-unit perceptron [36].....	17
Εικόνα 10: Τεχνητή Νοημοσύνη, Μηχανική Μάθηση και Βαθιά Μάθηση [20].....	22
Εικόνα 11: Τα πρότυπα δείγματα που χρησιμοποιήθηκαν.....	25
Εικόνα 12: Τα χρώματα του δείγματος .....	25
Εικόνα 13: Σχέσεις μεταξύ $(x, y)$ , $\rho$ και $\theta$ [26] .....	28
Εικόνα 14: Εικόνα δοκιμών.....	29
Εικόνα 15: Ανίχνευση Ακμών με τη μέθοδο Canny .....	30
Εικόνα 16: Line detection with Canny method.....	30
Εικόνα 17: Ανίχνευση Ακμών με τη μέθοδο Sobel .....	31
Εικόνα 18: Line detection with Sobel method.....	31
Εικόνα 19: Edge image .....	32
Εικόνα 20: Line detection.....	33
Εικόνα 21: Αρχική εικόνα συγκριτικά με την εικόνα με της ανιχνευμένες ακμές .....	34
Εικόνα 22: Ανίχνευση Γραμμών.....	34
Εικόνα 23: Feed-Forward Neural Network .....	40

## Συντομογραφίες

- ΜΜ** Μηχανική Μάθηση
- ΝΔ** Νευρωνικά Δίκτυα
- ΤΝΔ** Τεχνητά Νευρωνικά Δίκτυα
- ΣΝΔ** Συνελεκτικά Νευρωνικά Δίκτυα
- ΒΜ** Βαθιά Μάθηση
- ΠΝΔ** Πολυστρωματικά Νευρωνικά Δίκτυα
- MLF** Multilayer Feedforward

## 1 Εισαγωγή

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανίχνευση προσχεδίων σε πίνακες ζωγραφικής. Για την εξαγωγή και την ανίχνευση προτύπων πίσω από έργα τέχνης, χρησιμοποιούνται τεχνικές επεξεργασίας εικόνας, όρασης υπολογιστών και βαθιάς μάθησης. Οι προαναφερόμενες μέθοδοι σε συνδυασμό με την πληθώρα διαθέσιμων ψηφιακών συλλογών έργων τέχνης, δίνουν τη δυνατότητα σε ερευνητές της επιστήμης των υπολογιστών να συνδράμουν στην κοινότητα των καλών τεχνών, παρέχοντας αυτόνομα εργαλεία, για την πληρέστερη ερμηνεία των έργων τέχνης [1].

Ειδικότερα, στα επόμενα κεφάλαια αναπτύσσεται το θεωρητικό υπόβαθρο που χρειάστηκε για την υλοποίηση, καθώς και μεθοδολογία και τα εργαλεία που πάρθηκαν για την επίτευξη του στόχου. Στο πρώτο κεφάλαιο της εργασίας παρουσιάζονται ορισμένες γενικές πληροφορίες για το αντικείμενο της διπλωματικής, καθώς μία σύντομη αναφορά στα κίνητρα που αποτέλεσαν το έναυσμα για την υλοποίηση της.

Το δεύτερο κεφάλαιο αφορά το θεωρητικό υπόβαθρο της εργασίας. Το θεωρητικό μέρος συγκροτείται σε πέντε μεγάλες ενότητες την Επεξεργασία Εικόνας, την Όραση Υπολογιστών, τη Μηχανική Μάθηση, τη Βαθιά Μάθηση και Μηχανική Μάθηση συγκριτικά με τη Βαθιά Μάθηση. Στις τέσσερις πρώτες ενότητες παρουσιάζονται ορισμένες χρήσιμες πληροφορίες για τον εκάστοτε τομέα, κάποια χαρακτηριστικά μοντέλα, καθώς και ορισμένες ενδιαφέρουσες εφαρμογές. Συνεπακόλουθα, στην πέμπτη ενότητα του κεφαλαίου παρουσιάζονται συγκριτικά ορισμένες πληροφορίες για την Μηχανική Μάθηση και τη Βαθιά Μάθηση.

Στο τρίτο κεφάλαιο γίνεται αναφορά στη προγραμματιστική γλώσσα που χρησιμοποιήθηκε για την υλοποίηση της εργασίας, η οποία είναι η γλώσσα Matlab, έκδοσης R2023b. Η ενότητα ξεκινάει με μία ιστορική αναδρομή της προγραμματιστικής γλώσσας. Στη συνέχεια, αναγράφονται κάποια σημαντικά χαρακτηριστικά της, και τέλος επισημαίνονται ορισμένα από τα προτερήματα της σχετικά με τη Μηχανική Μάθηση.

Στο τέταρτο κεφάλαιο αναγράφονται πληροφορίες σχετικά με τα υπερφασματικά δεδομένα με τα γραμμικά σχέδια, που χρησιμοποιήθηκαν για τη δημιουργία του συνόλου δεδομένων. Τα υπερφασματικά δεδομένα με τα γραμμικά σχέδια, πάρθηκαν από το Τμήμα Συντήρησης Αρχαιοτήτων και Έργων Τέχνης του Πανεπιστημίου Δυτικής Αττικής και πρόκειται για εικόνες που έχουν ληφθεί σε διαφορετικές και σαφώς ορισμένες περιοχές του φάσματος, με διαφορετικούς ανιχνευτές από πρότυπα δείγματα.

Το πέμπτο κεφάλαιο αφοσιώνεται στην προ-επεξεργασία των δεδομένων. Αρχικά, περιγράφονται αναλυτικά όλες οι τεχνικές που χρησιμοποιήθηκαν για την ανίχνευση των

γραμμών στα υπερφασματικά δεδομένα. Η ανίχνευση γραμμών υλοποιήθηκε με χρήση του μετασχηματισμού Hough, ωστόσο έγιναν ποικίλες δοκιμές για τον εντοπισμό όσο το δυνατόν περισσότερων γραμμών. Στη συνέχεια, ακολουθεί η μεθοδολογία για την εξαγωγή των απαιτούμενων τμημάτων της εικόνας και η κατάταξη τους σε συγκεκριμένους καταλόγους, με στόχο τη δημιουργία ενός μοναδικού συνόλου δεδομένων. Το μοναδικό σύνολο δεδομένων που δημιουργήθηκε, αποτελείται από τρεις κατηγορίες. Στην πρώτη υπάρχουν τμήματα εικόνων που το κεντρικό σημείο βρίσκεται σε γραμμή προσχεδίου, στη δεύτερη κατηγορία τμήματα όπου περιέχουν γραμμή προσχεδίου αλλά όχι στο κέντρο, ενώ στην τρίτη και τελευταία κατηγορία τμήματα που δεν έχουν καμία γραμμή προσχεδίου, δηλαδή είναι υπόβαθρο.

Το έκτο κεφάλαιο εμπεριέχει την εκπαίδευση του νευρωνικού δικτύου, δηλαδή το πειραματικό μέρος. Αναγράφονται αναλυτικά ποια μοντέλα χρησιμοποιήθηκαν και τι δοκιμές έγιναν.

Ολοκληρώνοντας, στο έβδομο και τελευταίο κεφάλαιο παρουσιάζεται μία ανασκόπηση της παρούσας διπλωματικής, με συμπεράσματα και ορισμένες προοπτικές.

## 1.1 Κίνητρα

Οι ανθρώπινοι οργανισμοί, μέσω της ανθρώπινη αντίληψης, είναι σε θέση να αναγνωρίζουν πρότυπα ή επαναλαμβανόμενα μοτίβα σε οπτικά έργα τέχνης. Για την ακρίβεια, η ικανότητα αναγνώρισης υφολογικών και σημασιολογικών χαρακτηριστικών σε έργα τέχνης, προκύπτει από τη σύνθεση των διαφορετικών χαρακτηριστικών που πλαισιώνουν ένα έργο τέχνης, όπως του χρώματος, της υφής και του σχήματος, τα οποία αντιλαμβάνεται το ανθρώπινο μάτι.

Ωστόσο, με μοναδικό εργαλείο την ανθρώπινη αντίληψη, οι πρώτες προσπάθειες εξαγωγής χαρακτηριστικών εμφάνισαν δυσκολίες. Αυτό συμβαίνει διότι μετά τη σύλληψη της ρητής γνώσης ως προς τα χαρακτηριστικά, ακολουθεί το στάδιο της συσχέτισης τους, με έναν συγκεκριμένο καλλιτέχνη ή με ένα συγκεκριμένο έργο τέχνης. Ως αποτέλεσμα, οι ανθρώπινοι παρατηρητές μπορεί να δυσκολεύονται να διατυπώσουν και να εννοιολογήσουν, καθώς η γνώση συνδέεται με την έμμεση και υποκειμενική εμπειρογνωμοσύνη.

Αντιθέτως, προσεγγίσεις όπως αυτές που προσφέρουν τα μοντέλα βαθιάς μάθησης ενδέχεται να αποτελέσουν το κλειδί της επιτυχίας για την εξαγωγή χρήσιμων στοιχείων. Με κίνητρο λοιπόν, το ολοένα και αυξανόμενο ενδιαφέρον σε αυτόν τον ερευνητικό τομέα, η παρούσα διπλωματική έχει ως στόχο να παρουσιάσει τεχνικές που βασίζονται στην επεξεργασία εικόνας και στη βαθιά μάθηση, οι οποίες θα συνεισφέρουν στην εξαγωγή και αναγνώριση προτύπων σε πίνακες ζωγραφικής.



Τέλος, η παρούσα διπλωματική, αν και απευθύνεται κυρίως σε ερευνητές ή επαγγελματίες της πληροφορικής, επιδιώκει να προσελκύσει και ανθρωπιστές, οι οποίοι θα μπορούσαν να ανακαλύψουν τις προόδους στη βαθιά μάθηση, την επεξεργασία εικόνας και την όραση υπολογιστών και να συμβάλλουν στην υποστήριξη των δραστηριοτήτων τους [1].

## 2 Θεωρητικό Υπόβαθρο

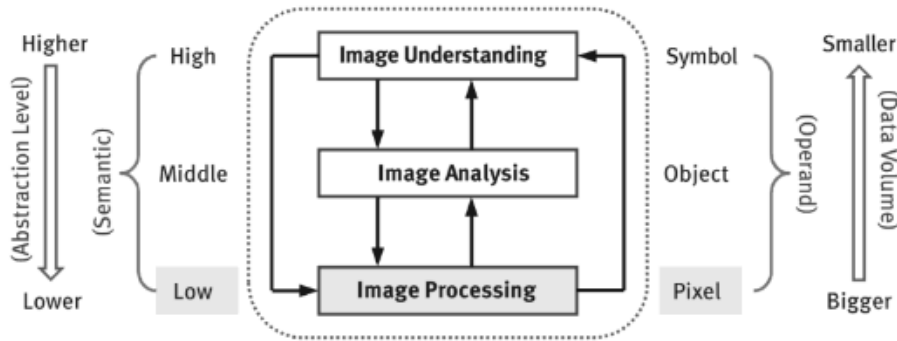
Το κεφάλαιο που ακολουθεί οργανώνεται σε πέντε ευρύτερες ενότητες, και έχει ως στόχο να παραθέσει στον αναγνώστη το προαπαιτούμενο γνωστικό υπόβαθρο για την είσοδο του σε αυτό το γνωστικό πεδίο.

### 2.1 Επεξεργασία Εικόνας (Image Processing)

Οι εικόνες αποτελούν σημαντικό μέσο για την πλειονότητα πληροφοριών που λαμβάνουν τα ανθρώπινα όντα, από τον πραγματικό κόσμο. Η λήψη των εικόνων μπορεί να γίνει μέσω διάφορων συστημάτων παρατήρησης και σύλληψης. Ένα χαρακτηριστικό σύστημα είναι το ανθρώπινο οπτικό σύστημα. Οι άνθρωποι αναγνωρίζουν αντικείμενα και σκηνές παρατηρώντας εικόνες. Οι εικόνες στον πραγματικό κόσμο είναι τρισδιάστατες (3-D), ωστόσο οι ψηφιακές εικόνες αποτυπώνονται σε δισδιάστατη μορφή (2-D). Επομένως, μία εικόνα μπορεί να οριστεί ως μία δισδιάστατη συνάρτηση  $f(x, y)$ , όπου οι τιμές  $x$  και  $y$  είναι πεπερασμένες και διακριτές ποσότητες. Σε αυτή τη μορφή η εικόνα μπορεί να επεξεργαστεί.

Η επεξεργασία εικόνας, με την ευρύτερη έννοια, εκτείνεται σε μία σειρά από τεχνικές επεξεργασίας εικόνας, οι οποίες αναπτύσσονται εδώ και αρκετά χρόνια και ως στόχο έχουν την απόκτηση πληροφοριών από την εικόνα. Η όλη διαδικασία της επεξεργασίας εικόνας δομείται από τρία στρώματα, όπως φαίνεται στην Εικόνα 1. Στο πρώτο επίπεδο είναι η επεξεργασία εικόνας (image processing), έπειτα η ανάλυση εικόνας (image analysis) και τέλος η κατανόηση εικόνας (image understanding). Η επεξεργασία εικόνας αφορά την τροποποίηση μίας αρχικής εικόνας, έτσι ώστε να παραχθεί μία βελτιωμένη εικόνα, ενώ η ανάλυση της εικόνας σχετίζεται με την εξαγωγή των χρήσιμων πληροφοριών της εικόνας. Τέλος, η κατανόηση της εικόνας αναφέρεται στην ορθή αξιοποίηση των εξαγόμενων δεδομένων.

Οι διεργασίες της επεξεργασίας εικόνας μπορούν να κατανεμηθούν σε τρία επίπεδα, όπως αυτά απεικονίζονται στην Εικόνα 1. Αρχικά, υπάρχουν οι χαμηλές επιπέδου διεργασίες, οι οποίες περιλαμβάνουν λειτουργίες όπως μείωση του θορύβου, εξομάλυνση και όξυνση της εικόνας. Στις συγκεκριμένες διεργασίες τόσο η είσοδος όσο και η έξοδος είναι εικόνες. Συνεπακόλουθα, οι διεργασίες μεσαίου επιπέδου έχουν ως είσοδο μία εικόνα, οι έξοδοι τους όμως είναι χαρακτηριστικά που εξάγονται από την αρχική εικόνα. Για παράδειγμα, ορισμένες μεσαίου επιπέδου διεργασίες είναι η ανίχνευση γραμμών, ακμών, η τμηματοποίηση και η κατωφλίωση. Ολοκληρώνοντας, οι διεργασίες υψηλότερου επιπέδου χαρακτηρίζονται ως οι πιο σύνθετες, αφού τα εξαγόμενα χαρακτηριστικά συνήθως είναι αναγνωρισμένα αντικείμενα, για τα οποία χρειάζεται η ανθρώπινη οπτική για να αποδοθεί νόημα στην ύπαρξη τους [3],[4].



Εικόνα 1: Τα τρία επίπεδα της μηχανικής μίας εικόνας [3]

### 2.1.1 Ανίχνευση Ακμών (Edge Detection)

Η ανίχνευση ακμών αποτελεί βασικό εργαλείο για τη μηχανική όραση και την επεξεργασία εικόνας. Στην Επεξεργασία Εικόνας, μια ακμή είναι το όριο μεταξύ ενός αντικειμένου και του φόντου του. Αντιπροσωπεύει το σύνορο για μεμονωμένα αντικείμενα. Επομένως, εάν οι ακμές των αντικειμένων της εικόνας μπορούν να προσδιοριστούν με ακρίβεια, όλα τα αντικείμενα μπορούν να εντοπιστούν και να υπολογιστούν οι ιδιότητές τους, όπως το εμβαδόν, η περίμετρος ή το σχήμα. Η ανίχνευση ακμών είναι η διαδικασία εντοπισμού των εικονοστοιχείων των ακμών. Στη συνέχεια, μια ενίσχυση ακμών θα αυξήσει την αντίθεση μεταξύ των ακμών και του φόντου με τέτοιο τρόπο ώστε οι ακμές να γίνουν πιο ορατές. Επιπλέον, η ανίχνευση ακμών είναι η διαδικασία παρακολούθησης των ακμών, συγκεντρώνοντας συνήθως τα εικονοστοιχεία των ακμών σε έναν κατάλογο. Υπάρχουν διάφορα μοντέλα που χρησιμοποιούνται για την ανίχνευση ακμών [27]. Στη συνέχεια, θα γίνει αναφορά σε δύο αλγόριθμους που χρησιμοποιούνται για ανίχνευση ακμών.

Καταρχάς, ένα ισχυρό μοντέλο για ανίχνευση ακμών είναι ο ανιχνευτής ακμών Canny. Η προσέγγιση του Canny αποσκοπεί στην υλοποίηση τριών βασικών στόχων όπως αυτοί καταγράφονται στη συνέχεια. Πρώτα, επιδιώκει το χαμηλό ρυθμό σφαλμάτων, με άλλα λόγια ο αλγόριθμος θα πρέπει να είναι σε θέση να εντοπίσει όλες τις ακμές χωρίς να υπάρχουν πλασματικές αποκρίσεις. Δεύτερον, τα σημεία των ακμών θα πρέπει να είναι πολύ καλά εντοπισμένα, δηλαδή οι ακμές που έχουν εντοπιστεί θα πρέπει να είναι όσο τον δυνατόν πιο κοντά στις πραγματικές ακμές και τρίτον, επιδιώκει στην απόκριση ενός μοναδικού σημείου. Ο ανιχνευτής θα πρέπει να επιστρέψει μόνο ένα σημείο για το κάθε σημείο της πραγματικής ακμής [4]. Το πρώτο βήμα του παραδοσιακού αλγόριθμου Canny είναι η εξομάλυνση της εικόνας. Ο Canny συμπέρανε την πρώτη παράγωγο της γκαουσιανής συνάρτησης, η οποία είναι η καλύτερη προσέγγιση του βέλτιστου τελεστή ανίχνευσης ακμών. Προκειμένου να γενικευτεί το αποτέλεσμα σε δύο διαστάσεις, ο αλγόριθμος Canny χρησιμοποιεί γενικά δισδιάστατη συνάρτηση Gauss που ακολουθεί,  $G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ , όπου το  $\sigma$  αντιπροσωπεύει την παράμετρο του φίλτρου Gauss και ελέγχει την έκταση της

εξομάλυνσης της εικόνας. Έπειτα, μπορεί να κατασκευαστεί μία εξομαλυμένη εικόνα  $f_s(x, y) = G(x, y) * f(x, y)$ . Το δεύτερο βήμα είναι ο υπολογισμός του μεγέθους και της κατεύθυνσης της κλίσης της εικόνας. Ο παραδοσιακός αλγόριθμος Canny υιοθετεί την περιορισμένη διαφορά  $2 \times 2$  γειτονικών περιοχών για τον υπολογισμό της τιμής και της κατεύθυνσης της κλίσης της εικόνας. Το μέγεθός και η κατεύθυνση της κλίσης μπορούν να υπολογιστούν βάσει την επόμενη συνάρτηση,  $M(x, y) = \|\nabla f_s(x, y)\| = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}$  και  $\alpha(x, y) = \tan^{-1} \left[ \frac{g_y(x, y)}{g_x(x, y)} \right]$ , όπου  $g_x(x, y) = \frac{\partial f_s(x, y)}{\partial x}$  και  $g_y(x, y) = \frac{\partial f_s(x, y)}{\partial y}$ . Αφού αποκτηθεί η εικόνα μεγέθους κλίσης  $f_s(x, y)$ , απαιτείται η εκτέλεση μη μέγιστης καταστολής στην εικόνα για την ακριβή τοποθέτηση των ακμών. Τέλος, Ο αλγόριθμος Canny υιοθετεί τη μέθοδο διπλού κατώφλιου για την επιλογή των σημείων ακμής μετά τη διενέργεια μη μέγιστης καταστολής. Τα εικονοστοιχεία των οποίων το μέγεθος κλίσης είναι πάνω από το υψηλό κατώφλι θα επισημανθούν ως σημεία ακμής, και εκείνα των οποίων το μέγεθος κλίσης είναι κάτω από το χαμηλό κατώφλι θα επισημανθούν ως σημεία μη ακμής, και τα υπόλοιπα θα επισημανθούν ως υποψήφια σημεία ακμής. Τα υποψήφια σημεία ακμής που συνδέονται με σημεία ακμής θα σημειωθούν ως σημεία ακμής. Αυτή η μέθοδος μειώνει την επίδραση του θορύβου στην ακμή της τελικής εικόνας ακμής [27], [28].

Ένα επιπλέον ισχυρό μοντέλο είναι ο ανιχνευτής ακμών Sobel. Η μέθοδος ανίχνευσης ακμών Sobel προτάθηκε από τον Sobel το 1970. Ο τελεστής Sobel εκτελεί μια διδιάστατη μέτρηση χωρικής κλίσης σε μια εικόνα και έτσι δίνει έμφαση σε περιοχές υψηλής χωρικής συχνότητας που αντιστοιχούν σε ακμές. Συνήθως χρησιμοποιείται για την εύρεση του κατά προσέγγιση απόλυτου μεγέθους της κλίσης σε κάθε σημείο μιας εικόνας εισόδου σε κλίμακα του γκρι. Θεωρητικά τουλάχιστον, ο τελεστής αποτελείται από ένα ζεύγος πυρήνων συνέλιξης  $3 \times 3$ , όπως απεικονίζονται στην Εικόνα 2. Ο ένας από τους δύο πυρήνες έχει μέγιστη απόκριση στην κάθετη ακμή και ο άλλος έχει μέγιστη απόκριση στην ακμή επιπέδου, για αυτό ο ένας πυρήνας είναι απλώς ο άλλος περιστρεφόμενος κατά  $90^\circ$ . Αυτοί οι πυρήνες έχουν σχεδιαστεί έτσι ώστε να ανταποκρίνονται στο μέγιστο βαθμό στις ακμές που διατρέχουν κάθετα και οριζόντια σε σχέση με το πλέγμα εικονοστοιχείων, ένας πυρήνας για κάθε έναν από τους δύο κάθετους προσανατολισμούς. Οι πυρήνες μπορούν να εφαρμοστούν χωριστά στην εικόνα εισόδου, για να παράγουν χωριστές μετρήσεις της συνιστώσας κλίσης σε κάθε προσανατολισμό, οι οποίες ονομάζονται  $G_x$  και  $G_y$ . Αυτές μπορούν στη συνέχεια να συνδυαστούν μαζί για να βρεθεί το απόλυτο μέγεθος της κλίσης σε κάθε σημείο και ο προσανατολισμός αυτής της κλίσης. Ο ανιχνευτής ακμών Sobel δίνει μεγαλύτερη έμφαση στα εικονοστοιχεία της εικόνας που βρίσκονται πιο κοντά στο κέντρο της μάσκας. Αυτός ο χειριστής προηγείται των ακμών όπου η κλίση είναι υψηλότερη σε εικόνες κλίμακας του γκρι, αυτή η μέθοδος χρησιμοποιείται επίσης για την εύρεση των

εκτιμώμενων τιμών απόλυτου μεγέθους κλίσης σε κάθε σημείο. Το μέγεθος της κλίσης υπολογίζεται από τον τύπο,  $|G| = \sqrt{G_x^2 + G_y^2}$ , ενώ η γωνία προσανατολισμού της ακμής, σε σχέση με το πλέγμα των εικονοστοιχείων, που προκαλεί τη χωρική κλίση δίνεται από τον τύπο  $\theta = \tan^{-1} \frac{G_y}{G_x}$  [29], [30].

-1	0	+1
-2	0	+2
-1	0	+1

G<sub>x</sub>

+1	+2	+1
0	0	0
-1	-2	-1

G<sub>y</sub>

Εικόνα 2: Sobel Convolutional Kernels

### 2.1.2 Κατωφλίωση (Thresholding)

Η κατωφλίωση είναι μία θεμελιώδης τεχνική στην επεξεργασία εικόνας, η οποία κατατάσσεται στις τεχνικές τμηματοποίησης της εικόνας. Αποτελεί μία από τις πιο συνηθισμένες χρησιμοποιούμενες μεθόδους για την τμηματοποίηση εικόνας, εξαιτίας της διαισθητικότητας της, της απλότητας που χαρακτηρίζει την υλοποίηση της και της υπολογιστικής ταχύτητας της. Στην προηγούμενη υποενότητα, τα τμήματα της εικόνας που προσδιορίστηκαν ως τμήματα ακμών, συνδέθηκαν δημιουργώντας περιγράμματα. Ζητούμενο της ενότητας που ακολουθεί, αποτελεί η άμεση κατάτμηση εικόνων σε περιοχές, με τη βοήθεια τιμών έντασης [4].

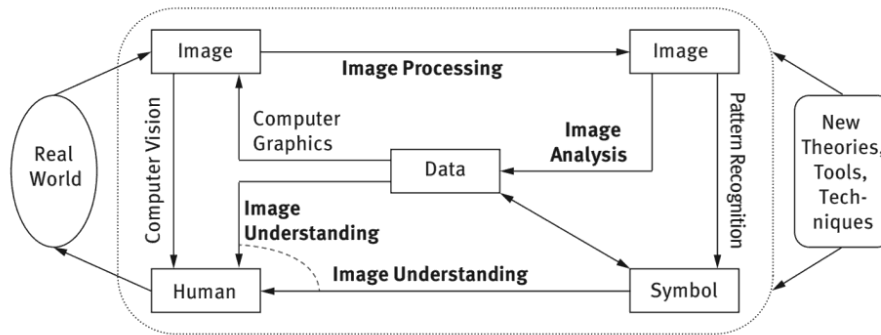
Πιο συγκεκριμένα, η τμηματοποίηση μίας εικόνας είναι η διαδικασία διαχωρισμού της, σε πολλαπλές περιοχές ή τμήματα, όπου κάθε τμήμα αντιπροσωπεύει ένα ξεχωριστό αντικείμενο ή μέρος ενός αντικειμένου εντός της εικόνας. Στόχος είναι η απλοποίηση της αναπαράστασης μιας εικόνας σε περιοχές με νόημα για περαιτέρω ανάλυση ή επεξεργασία. Έστω ότι υπάρχει μία εικόνα  $f(x, y)$ , η οποία αποτελείται από φωτεινά αντικείμενα που βρίσκονται σε ένα σκοτεινό υπόβαθρο, με τρόπο τέτοιο ώστε, τα εικονοστοιχεία των αντικειμένων και του υπόβαθρου να έχουν τιμές έντασης, οι οποίες να είναι ομαδοποιημένες σε δύο βασικές ομάδες. Οι τιμές έντασης της εικόνας συμβολίζονται από τις συντεταγμένες  $(x, y)$ , της συνάρτησης  $f(x, y)$ . Ένας προφανής τρόπος εξαγωγής των αντικειμένων από το υπόβαθρο, είναι να επιλεγεί ένα κατώφλι  $T$ , το οποίο θα επιτρέπει το διαχωρισμό των ομάδων. Στην περίπτωση αυτή, το κάθε σημείο  $(x, y)$  της εικόνας για το οποίο ισχύει  $f(x, y) > T$ , ονομάζεται σημείο του αντικειμένου (ή αλλιώς προσκήνιο), ενώ στην αντίθετη περίπτωση είναι γνωστό ως σημείο υπόβαθρου. Το τελικό προϊόν, είναι η κατατεμημένη εικόνα  $g(x, y)$ , η οποία περιγράφεται ως εξής:  $g(x, y) = \begin{cases} 1, & \text{εάν είναι } f(x, y) > T \\ 0, & \text{εάν είναι } f(x, y) < T \end{cases}$  [4].

Σε περιπτώσεις κατά τις οποίες το  $T$  είναι μία σταθερά, η οποία εφαρμόζεται σε όλη την έκταση της εικόνας, η διαδικασία που περιγράφεται από την προηγούμενη εξίσωση είναι γνωστή ως ολική καταφλίσωση. Σε αντίθετη περίπτωση, όπου η τιμή του  $T$  είναι μεταβλητή για διάφορες περιοχές της εικόνας, χρησιμοποιείται ο όρος μεταβλητή καταφλίσωση. Ειδικότερα, σε καταστάσεις εφαρμογής μεταβλητού καταφλίου, στις οποίες η τιμή του  $T$  σε κάθε σημείο  $(x, y)$  της εικόνας εξαρτάται από τις ιδιότητες της γειτονίας του σημείου  $(x, y)$ . Συνεπακόλουθα, η επιλογή της κατάλληλης τιμής καταφλίου μπορεί να αποτελέσει πρόκληση και συχνά εξαρτάται από τα χαρακτηριστικά της εικόνας και τη συγκεκριμένη εργασία. Η επίτευξη της επιλογής κατάλληλης τιμής καταφλίου μπορεί να γίνει μέσω διάφορων αυτοματοποιημένων τεχνικών, όπως της μεθόδου Otsu [4].

## 2.2 Όραση Υπολογιστών (Computer Vision)

Όπως έγινε αντιληπτό από την προηγούμενη ενότητα, είναι σημαντικό οι εικόνες που συλλέγονται από τον πραγματικό κόσμο, να υποβάλλονται σε διαδικασία επεξεργασίας, έτσι ώστε να εξαχθούν βασικά χαρακτηριστικά ή να αναγνωριστούν μοτίβα. Η ικανότητα των υπολογιστών να κατανοούν περιβάλλοντα από οπτικές πληροφορίες, είναι ο σκοπός της όρασης υπολογιστών [3], [6].

Πιο συγκεκριμένα, η όραση υπολογιστών είναι ένας διεπιστημονικός τομέας που επικεντρώνεται στο να δίνει τη δυνατότητα στους υπολογιστές, να ερμηνεύουν και να κατανοούν τις οπτικές πληροφορίες του κόσμου, παρόμοια με τον τρόπο με τον οποίο οι άνθρωποι αντιλαμβάνονται και επεξεργάζονται τα οπτικά δεδομένα. Περιλαμβάνει την ανάπτυξη αλγορίθμων, μοντέλων και συστημάτων που επιτρέπουν στους υπολογιστές να εξάγουν σημαντικές πληροφορίες από εικόνες, βίντεο και άλλες πηγές οπτικών δεδομένων. Η όραση υπολογιστών αποσκοπεί στην αναπαραγωγή και την αυτοματοποίηση των ανθρώπινων διαδικασιών όρασης, με τη χρήση ψηφιακής τεχνολογίας. Ακόμα, η όραση υπολογιστών έχει σημειώσει σημαντική πρόοδο, χάρη στη διαθεσιμότητα μεγάλων συνόλων δεδομένων, ισχυρού υπολογιστικού υλικού και βελτιωμένων αλγορίθμων. Διαδραματίζει κρίσιμο ρόλο σε διάφορες βιομηχανίες, προωθώντας καινοτομίες σε τομείς όπως η υγειονομική περίθαλψη, η τεχνολογία αυτοκινήτων, η ψυχαγωγία και η επιτήρηση, μεταξύ άλλων. Καθώς η τεχνολογία συνεχίζει να εξελίσσεται, η όραση υπολογιστών αναμένεται να έχει ολοένα και πιο σημαντικό ρόλο στην καθημερινή μας ζωή και στις βιομηχανίες. Στο σχήμα που αναπαριστάται στην Εικόνα 3, γίνεται αντιληπτή η μηχανική της εικόνας, δηλαδή πως από δεδομένα του πραγματικού κόσμου προκύπτουν εικόνες, δεδομένα, καινούριες θεωρίες και τεχνικές.

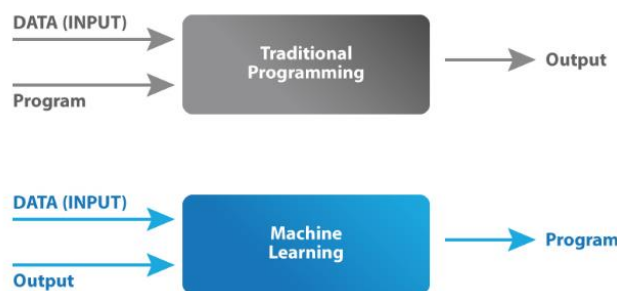


Εικόνα 3: Μηχανική εικόνας και συναφή θέματα [3]

## 2.3 Μηχανική Μάθηση (Machine Learning)

Οι άνθρωποι, από την αρχή της εξέλιξης τους, χρησιμοποιούν διάφορους τύπους εργαλείων, ώστε να απλοποιήσουν τις καθημερινές τους εργασίες. Αργότερα, η εφευρετικότητα του ανθρώπινου εγκεφάλου οδήγησε στη δημιουργία μηχανών, οι οποίες διευκόλυναν ποικίλες ανάγκες τους. Έτσι, και η Μηχανική Μάθηση (ΜΜ) είναι μία από αυτές τις μεταγενέστερες μηχανές, που επιδιώκουν να διευκολύνουν την ανθρώπινη ζωή. Σύμφωνα με τον Arthur Samuel, ο οποίος επινόησε τον όρο «Μηχανική Μάθηση» το 1959, ως ΜΜ ορίζεται το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν χωρίς να προγραμματίζουν ρητά [7].

Συγκεκριμένα, η ΜΜ είναι μία συνδυαστική προσέγγιση της Τεχνητής Νοημοσύνης και της επιστήμης των υπολογιστών. Βασίζεται σε τεχνικές οι οποίες επιτρέπουν στους υπολογιστές, να αποκτήσουν γνώση και να λαμβάνουν αποφάσεις, χωρίς να χρειάζεται να προγραμματιστούν εξαρχής για κάθε συγκεκριμένη εργασία. Απλώς τους παρέχονται δεδομένα και στη συνέχεια, αναγνωρίζουν μοτίβα και λαμβάνουν αποφάσεις με ελάχιστη επέμβαση από τον ανθρώπινο παράγοντα. Σε αντίθεση, ο παραδοσιακός προγραμματισμός απαιτεί, να κατασκευαστούν εκ των προτέρων καλά προγραμματισμένα προγράμματα, που να δέχονται δεδομένα και να εκτελούν συγκεκριμένες εργασίες, βασιζόμενα σε αυστηρές οδηγίες. Μια αναπαράσταση για το πώς λειτουργεί η ΜΜ συγκριτικά με τον παραδοσιακό προγραμματισμό απεικονίζεται στην Εικόνα 4.



Εικόνα 4: Μηχανική Μάθηση συγκριτικά με τον συνηθισμένο Προγραμματισμό [10]

Συνεπακόλουθα, η MM βασίζεται σε διαφορετικούς αλγόριθμους για την επίλυση προβλημάτων δεδομένων. Επισημαίνεται ότι, το είδος του αλγορίθμου που εφαρμόζεται τελικά, είναι ανάλογο του είδους του προβλήματος, του αριθμού των μεταβλητών ή του μοντέλου που θα ταίριαζε καλύτερα. Οπότε, δεν υπάρχει ένας τύπος αλγορίθμου που να επιλύει όλα τα προβλήματα. Όλοι οι αλγόριθμοι που χρησιμοποιούνται για τη MM, μπορούν να ομαδοποιηθούν σε τέσσερις ευρύτερες κατηγορίες, όπως αυτές καταγράφονται στην Εικόνα 5. Έτσι, προκύπτουν οι Αλγόριθμοι Μάθησης με Επίβλεψη (Supervised Learning Algorithms), οι Αλγόριθμοι Μάθησης Χωρίς Επίβλεψη (Unsupervised Learning Algorithms), οι Αλγόριθμοι Μάθησης με ημιεπίβλεψη (Semi-Supervised Learning Algorithms) και Αλγόριθμοι Ενισχυτικής Μηχανικής Μάθησης (Reinforcement Machine Learning Algorithms).

Αρχικά, η επιβλεπόμενη μάθηση ορίζεται ως η εργασία MM της εκμάθησης μίας συνάρτησης, η οποία προκύπτει από την αντιστοιχία της εισόδου σε μία έξοδο με βάση παραδείγματα ζευγών εισόδου-εξόδου. Πιο συγκεκριμένα, καθώς τα δεδομένα εισόδου τροφοδοτούνται στο μοντέλο, το μοντέλο ρυθμίζει τα βάρη του μέχρι να προσαρμοστεί κατάλληλα. Αυτό συμβαίνει στο πλαίσιο της διαδικασίας διασταυρούμενης επικύρωσης για να διασφαλιστεί ότι το μοντέλο αποφεύγει την υπερπροσαρμογή ή την υποπροσαρμογή. Ορισμένοι αλγόριθμοι που εφαρμόζονται στην επιβλεπόμενη μάθηση περιλαμβάνουν τα Νευρωνικά Δίκτυα (Neural Networks), Δέντρα Απόφασης (Decision Trees), τις Μηχανές Υποστήριξης Διανυσμάτων (Support Vector Machines), Γραμμική Παλινδρόμηση (Linear Regression) και άλλα.

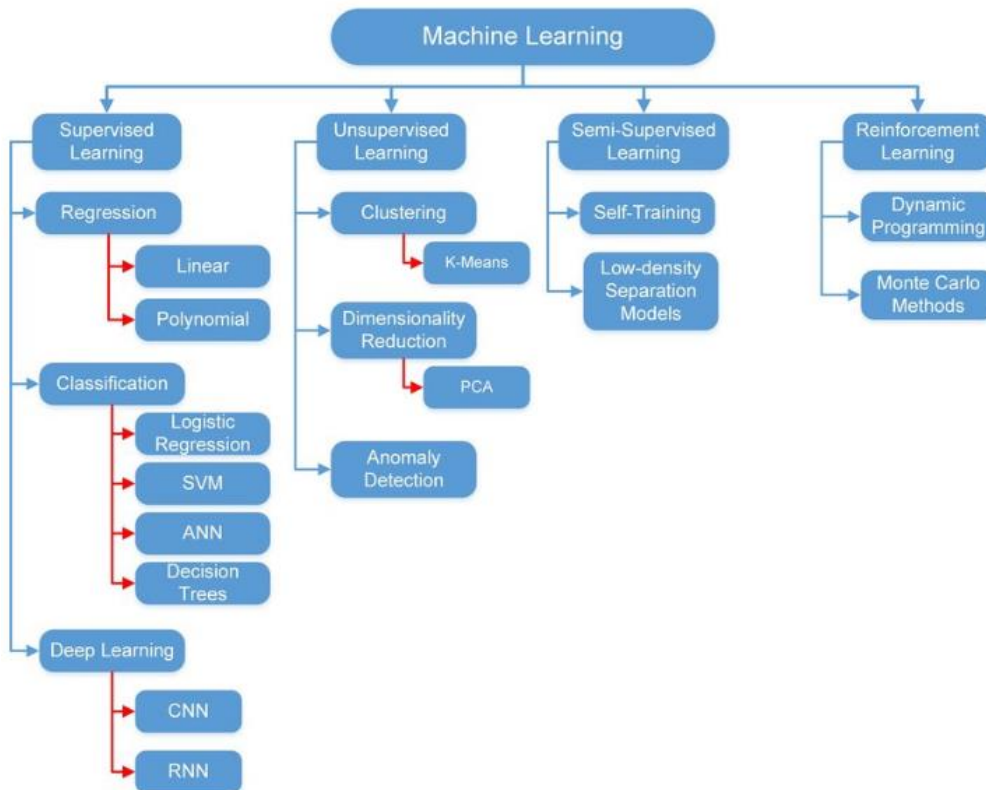
Στη συνέχεια, η μη επιβλεπόμενη μάθηση ανήκει στο κλάδο της MM στον οποίο μία συνάρτηση-πρότυπο συνάγεται με βάση μη επισημασμένα δεδομένα εκπαίδευσης. Με άλλα λόγια, τα δεδομένα εκπαίδευσης αποτελούνται μόνο από εισόδους και όχι από γνωστές εξόδους. Ως εκ τούτου, οι αλγόριθμοι μάθησης χωρίς επίβλεψη στοχεύουν στο να δώσουν νόημα στα δεδομένα εκπαίδευσης βρίσκοντας σχέσεις και μοτίβα μέσα σε αυτά. Οι αλγόριθμοι που εφαρμόζονται στη μη επιβλεπόμενη μάθηση μπορούν να χωριστούν σε τρεις ευρύτερες κατηγορίες, τους αλγόριθμους ομαδοποίησης, μειώσεις διαστάσεων και ανίχνευσης ανωμαλιών. Ο κατάλογος των αλγορίθμων που χρησιμοποιούνται για αυτού του είδους τα προβλήματα περιλαμβάνει τον αλγόριθμο Apriori και τους αλγόριθμους K-Means και Association Rules.

Έπειτα, ακολουθεί η μάθηση με ημιεπίβλεψη στην οποία μία συνάρτηση- πρότυπο προκύπτει από τη χρήση μερικώς επισημασμένων δεδομένων εκπαίδευσης. Ειδικότερα, συνδυάζει στοιχεία από την επιβλεπόμενη και τη μη επιβλεπόμενη μάθηση. Ο στόχος της μάθησης με ημιεπίβλεψη είναι να προσπαθήσει να αξιοποιήσει τόσο τα επισημασμένα όσο και τα μη επισημασμένα δεδομένα για να αποκτήσει καλύτερα μοντέλα μάθησης. Προκειμένου να επιτευχθεί αυτό, οφείλουν να γίνουν κάποιες παραδοχές, ώστε να υπάρξει



κάποια δομή στην κατανομή των δεδομένων, όπως σημεία που βρίσκονται κοντά το ένα στο άλλο είναι πιθανότερο να ανήκουν στην ίδια κατηγορία ή αντίστοιχα σημεία της ίδιας συστάδας είναι πιθανότερο να μοιράζονται ίδια χαρακτηριστικά.

Ολοκληρώνοντας, η ενισχυτική μάθηση λειτουργεί με τέτοιο τρόπο ώστε να ανακαλυφθούν οι ενέργειες με τις υψηλότερες ανταμοιβές. Οι αλγόριθμοι της ενισχυτικής MM δεν εκπαιδεύονται με τη χρήση δειγματικών δεδομένων, αντίθετα το μοντέλο μαθαίνει στην πορεία χρησιμοποιώντας τη δοκιμή και το σφάλμα [7]-[11].



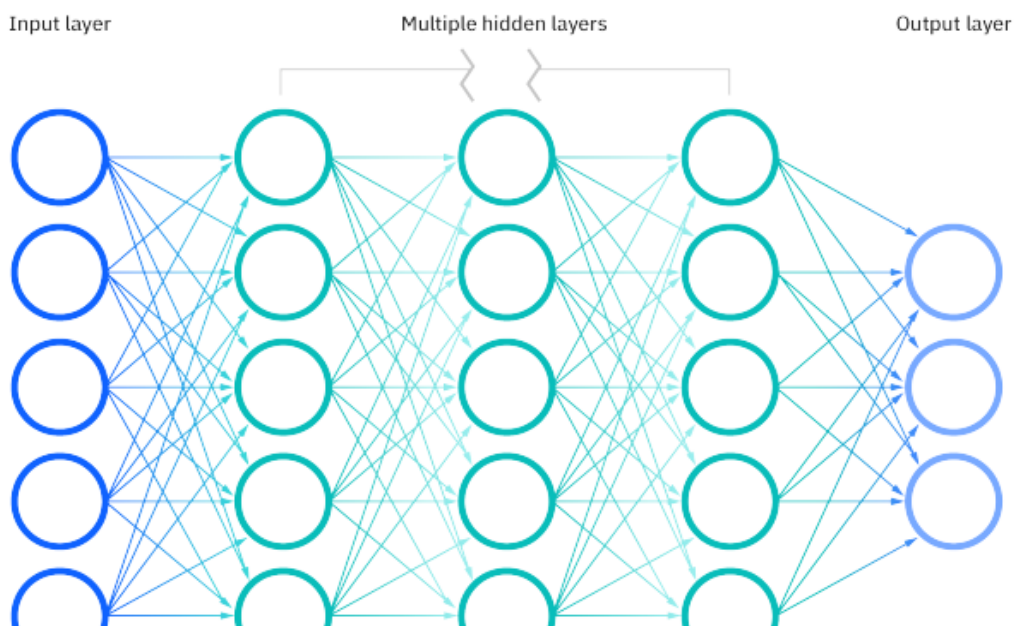
Εικόνα 5: Διαφορετικές κατηγορίες Μηχανικής Μάθησης και αλγόριθμοι [11]

### 2.3.1 Νευρωνικά Δίκτυα (Neural Network)

Υπάρχουν πολλοί διαθέσιμοι αλγόριθμοι MM, οι οποίοι χρησιμοποιούνται σε εφαρμογές αναγνώρισης προτύπων σε έργα τέχνης. Η υποενοότητα που ακολουθεί εστιάζει στα Νευρωνικά Δίκτυα (ΝΔ ή Neural Networks) ή γνωστά και ως Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ ή Artificial Neural Networks). Ένα ΝΔ περιλαμβάνει μία σειρά αλγορίθμων, οι οποίοι προσπαθούν να αναγνωρίσουν τις κρυμμένες σχέσεις μεταξύ ενός συνόλου δεδομένων. Η όλη διαδικασία λειτουργίας των ΝΔ είναι εμπνευσμένη από τον τρόπο με τον οποίο λειτουργεί ο ανθρώπινος εγκέφαλος, δηλαδή μιμούνται τον τρόπο με τον οποίο οι βιολογικοί νευρώνες στέλνουν σήματα ο ένας στον άλλο [7], [12]. Το κύριο πλεονέκτημα των νευρωνικών δικτύων είναι το γεγονός, ότι είναι σε θέση να χρησιμοποιήσουν κάποια εκ των προτέρων άγνωστη πληροφορία που κρύβεται στα δεδομένα, αλλά δεν είναι σε θέση να την εξάγουν. Η διαδικασία «σύλληψης» της άγνωστης πληροφορίας ονομάζεται «εκμάθηση του

νευρωνικού δικτύου» ή «εκπαίδευση του νευρωνικού δικτύου». Στον μαθηματικό φορμαλισμό μάθηση σημαίνει προσαρμογή των συντελεστών βάρους κατά τρόπο ώστε να πληρούνται ορισμένες προϋποθέσεις [34].

Τα ΤΝΔ αποτελούνται από στρώματα κόμβων, όπως αυτά απεικονίζονται στην Εικόνα 6, τα οποία περιλαμβάνουν ένα στρώμα εισόδου, ένα ή περισσότερα κρυφά στρώματα και ένα στρώμα εξόδου. Κάθε κόμβος, ή τεχνητός νευρώνας, συνδέεται με έναν άλλο και έχει ένα σχετικό βάρος και κατώφλι. Εάν η έξοδος οποιουδήποτε μεμονωμένου κόμβου είναι πάνω από την καθορισμένη τιμή κατωφλίου, ο εν λόγω κόμβος ενεργοποιείται, στέλνοντας δεδομένα στο επόμενο στρώμα του δικτύου. Διαφορετικά, δεν διαβιβάζονται δεδομένα στο επόμενο επίπεδο του δικτύου. Κύριο χαρακτηριστικό των ΤΝΔ είναι η δυνατότητα τους, να προσαρμόζονται σε μεταβαλλόμενες εισόδους, έτσι ώστε το δίκτυο να παράγει το βέλτιστο αποτέλεσμα χωρίς να χρειάζεται να επανασχεδιάσει τα κριτήρια εξόδου. Με αυτόν τον τρόπο καταφέρνουν να λειτουργήσουν ως ισχυρά εργαλεία στην επιστήμη των υπολογιστών και την τεχνητή νοημοσύνη, τα οποία επιτρέπουν την ταξινόμηση και την ομαδοποίηση δεδομένων με μεγάλη ταχύτητα. Ένα από τα πιο γνωστά νευρωνικά δίκτυα είναι ο αλγόριθμος αναζήτησης της Google [7], [12].



Εικόνα 6: Στρώματα Τεχνητών Νευρωνικών Δικτύων [12]

Κάθε μεμονωμένος κόμβος λειτουργεί ως αυτόνομο μοντέλο γραμμικής παλινδρόμησης, δηλαδή μία τιμή εκτιμάται βάσει την τιμή μίας άλλης μεταβλητής. Το μοντέλο αποτελείται από δεδομένα εισόδου (input data), βάρη (weights), bias ή τιμή κατωφλίου (threshold) και την έξοδο (output). Τα βάρη βοηθούν στον καθορισμό της βαρύτητας κάθε μίας από τις δοσμένες μεταβλητές. Οι μεταβλητές με τα μεγαλύτερα βάρη συμβάλλουν ουσιαστικότερα στην έξοδο συγκριτικά με τις άλλες μεταβλητές, οπότε κάθε είσοδος

πολλαπλασιάζεται με τα αντίστοιχα βάρη. Στη συνέχεια, τα αποτελέσματα συναθροίζονται και περνάνε από μία συνάρτηση ενεργοποίησης, η οποία καθορίζει την έξοδο. Εάν η έξοδος, υπερβαίνει ένα δεδομένο κατώφλι, τότε ο κόμβος ενεργοποιείται, διοχετεύοντας τα δεδομένα στο επόμενο επίπεδο του δικτύου. Αυτό έχει ως αποτέλεσμα η έξοδος ενός κόμβου να γίνεται η είσοδος του επόμενου κόμβου. Αυτή η διαδικασία της μετάδοσης δεδομένων από ένα επίπεδο στο επόμενο επίπεδο ορίζει αυτό το νευρωνικό δίκτυο ως δίκτυο τροφοδότησης. Οι εξισώσεις που περιγράφουν τις παραπάνω σχέσεις, αναγράφονται παρακάτω [12].

$$\text{Εξίσωση 1: } \sum wixi + bias = w1x1 + w2x2 + w3x3 + bias \quad [12]$$

$$\text{Εξίσωση 2: } Output = f(x) = \begin{cases} 1, & \text{αν } \sum w1x1 + b \geq 0 \\ 0, & \text{αν } \sum w1x1 + b < 0 \end{cases} \quad [12]$$

Είναι προφανές ότι, ένα ΝΔ λόγω της παράλληλης και κατανεμημένης δομής του και της ικανότητας του να μαθαίνει και να γενικεύει, προσφέρει πλήθος πλεονεκτημάτων. Καταρχάς, τα ΝΔ παρέχουν προσαρμοστικότητα, αφού είναι σε θέση να προσαρμόζουν τα βάρη τους ανάλογα με τις μεταβολές που γίνονται στο περιβάλλον τους. Πιο συγκεκριμένα, ένα ΝΔ εκπαιδευμένο να λειτουργεί σε συγκεκριμένο περιβάλλον, μπορεί εύκολα να επανεκπαιδευτεί ώστε να διαχειριστεί τυχόν υποδεέστερες μεταβολές στις συνθήκες του περιβάλλοντος λειτουργίας του. Ακόμα, μία θετική ιδιότητα ενός ΤΝΔ είναι η μη γραμμικότητα του. Ένας τεχνητός νευρώνας μπορεί να είναι είτε γραμμικός, είτε μη γραμμικός. Ένα ΝΔ που αποτελείται από διασυνδεδεμένους μη γραμμικούς νευρώνες είναι, εκ φύσεως, μη γραμμικό. Επιπλέον, ένα ΝΔ στο πλαίσιο ταξινόμησης προτύπων μπορεί να σχεδιαστεί με τρόπο τέτοιο ώστε, να παρέχει πληροφορίες όχι μόνο για το ποιο συγκεκριμένο πρότυπο θα επιλεγεί, αλλά και για το βαθμό εμπιστοσύνης του. Με τον τρόπο αυτό, θα απορρίπτονται διαφορούμενα μοτίβα, εάν αυτά προκύψουν, και τα βελτιώνεται η απόδοση του δικτύου. Επιπρόσθετα, η γνώση αντιπροσωπεύεται από την ίδια τη δομή και την κατάσταση ενεργοποίησης ενός ΝΔ. Κάθε νευρώνας στο δίκτυο μπορεί να επηρεαστεί από τη συνολική δραστηριότητα όλων των άλλων νευρώνων του δικτύου. Αυτό σημαίνει, ότι ένα ΝΔ χειρίζεται με φυσικό τρόπο, τη σχετική με το περιεχόμενο πληροφορία. Ολοκληρώνοντας, ως γνωστόν η σχεδίαση ενός ΝΔ δανείζεται στοιχεία από τη λειτουργία του ανθρώπινου εγκεφάλου, για το λόγο αυτό οι νευροβιολόγοι αντιμετωπίζουν τα ΤΝΔ ως ένα ερευνητικό εργαλείο για την ερμηνεία νευροβιολογικών φαινομένων [19].

### 2.3.1.1 Perceptron

Το Perceptron είναι η πιο βασική μορφή αρχιτεκτονικής νευρωνικών δικτύων με μηδενικά κρυφά στρώματα, που χρησιμοποιείται κυρίως για την ταξινόμηση γραμμικά διαχωρίσιμων προτύπων. Το μοντέλο perceptron εμπίπτει στην κατηγορία της μάθησης με επίβλεψη και μπορεί να αναφέρεται ως δυαδικός ταξινομητής. Ο αλγόριθμος perceptron πλεονεκτεί στο σενάριο όπου θέλουμε να δώσουμε προτεραιότητα στην επίδραση

επιλεγμένων χαρακτηριστικών έναντι των άλλων, χρησιμοποιώντας τροποποιήσιμα βάρη. Αποτελείται από έναν μόνο νευρώνα McCulloch-Pitts που περιέχει ρυθμιζόμενα συναπτικά βάρη (weights) και bias. Ο αλγόριθμος perceptron είναι η απλούστερη διαμόρφωση του ΤΝΔ που δημιουργήθηκε ποτέ. Εφευρέθηκε στο Αεροναυτικό Εργαστήριο Cornell το έτος 1957 από τον Frank Rosenblatt. Το perceptron υλοποιήθηκε για πρώτη φορά σε λογισμικό IBM704. Επιπλέον, κατασκευάστηκε σε ειδικά κατασκευασμένο υλικό ως Mark 1 perceptron, σχεδιασμένο κυρίως για να εκτελεί αναγνώριση εικόνων με τη βοήθεια μιας συστοιχίας 400 φωτοκυττάρων, τυχαία συνδεδεμένων με τους νευρώνες. Το Perceptron μονού στρώματος, δεν μπορεί να εκπαιδευτεί για την αναγνώριση πολλαπλών κλάσεων μοτίβων, είναι ικανό μόνο για την εκμάθηση γραμμικά διαχωρίσιμων μοτίβων. Αργότερα, αναγνωρίστηκε ότι ένα νευρωνικό δίκτυο τροφοδότησης προς τα εμπρός που διαθέτει δύο ή περισσότερα κρυφά νευρωνικά στρώματα έχει ταχύτερη επεξεργαστική ισχύ από οποιοδήποτε perceptron ενός στρώματος [35].

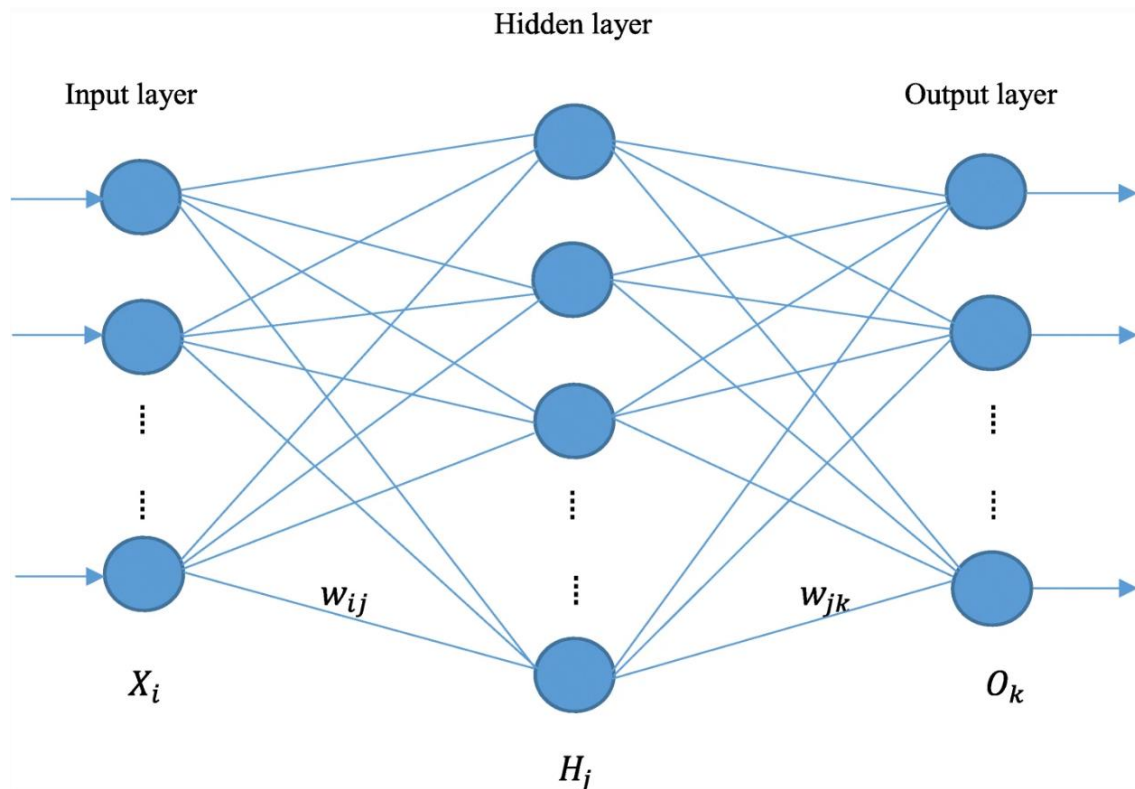
### 2.3.2 Πολυστρωματικά Νευρωνικά Δίκτυα (Multilayer Neural Networks)

Τις τελευταίες δεκαετίες, τα Πολυστρωματικά Νευρωνικά Δίκτυα (ΠΝΔ) έχουν λάβει σημαντική προσοχή και έχουν εφαρμοστεί με επιτυχία σε πολλούς τομείς, όπως η συνδυαστική βελτιστοποίηση, η επεξεργασία σήματος, η αναγνώριση προτύπων και η τεχνητή νοημοσύνη. Ένα ΠΝΔ είναι ένα συζευγμένο σύστημα εξισώσεων ΝΔ [37]. Μια από τις πιο δημοφιλείς και πιο ισχυρές αρχιτεκτονικές είναι το πολυστρωματικό perceptron.

Συχνά θεωρείται ότι τα perceptrons χρησιμοποιούνται γενικά για την ταξινόμηση γραμμικά διαχωρίσιμων τιμών που μπορούν να διαχωριστούν με μια ευθεία γραμμή. Αλλά, όταν η ταξινόμηση δεν είναι γραμμικά διαχωρίσιμη και αν τέτοια προβλήματα υλοποιούνται στο perceptron, αυτό θα επαναλαμβάνεται άπειρες φορές χωρίς άμεσο αποτέλεσμα. Τότε είναι που το perceptron πολλαπλών στρωμάτων είναι χρήσιμο. Το perceptrons πολλαπλών στρωμάτων, ή επίσης γνωστό ως νευρωνικό δίκτυο τροφοδότησης (Feedforward Neural Network), είναι μια από τις πιο συχνά χρησιμοποιούμενες τεχνικές προσέγγισης συναρτήσεων και έχει εφαρμοστεί σε μια μεγάλη ποικιλία προβλημάτων που προκύπτουν από διάφορους κλάδους [33]. Τα νευρωνικά δίκτυα multilayer feedforward (MLF), τα οποία εκπαιδεύονται με αλγόριθμο μάθησης οπισθοδιάδοσης, είναι από τα πιο δημοφιλή νευρωνικά δίκτυα. Το νευρωνικό δίκτυο πολλαπλών επιπέδων τροφοδότησης προς τα εμπρός, αποτελείται από νευρώνες, οι οποίοι είναι ταξινομημένοι σε επίπεδα.

Ένα πολυεπίπεδο δίκτυο perceptron περιλαμβάνει την οπισθοδιάδοση, μια διαδικασία για την επανειλημμένη προσαρμογή των σταθμισμένων και των τιμών κατωφλίου, αναλόγως για την ελαχιστοποίηση της διαφοράς μεταξύ της επιθυμητής-στοχευμένης εξόδου και της ληφθείσας εξόδου. Συνδέεται επίσης με κρυφά στρώματα ή νευρωνικούς κόμβους,

που στοιβάζονται μεταξύ των εισόδων και της εξόδου. Τα κρυφά στρώματα βοηθούν συχνά στην εκμάθηση πιο περίπλοκων χαρακτηριστικών. Το πλήθος των δεδομένων εισόδου που τροφοδοτούνται στο δίκτυο είναι γνωστό ως χαρακτηριστικό. Έτσι, αν υποθέσουμε ότι έχουμε τα δεδομένα εισόδου  $x_1 x_2 \dots x_n$ , διαθέτουμε «n» χαρακτηριστικά. Ένα χαρακτηριστικό είναι μια μεταβλητή της οποίας η επιρροή γίνεται αισθητή σε μια έξοδο. Αυτά τα χαρακτηριστικά πολλαπλασιάζονται στη συνέχεια με τα βάρη τους για να πραγματοποιηθεί το άθροισμα όλων των σταθμισμένων χαρακτηριστικών, το οποίο μας δίνει το γινόμενο σημείου [35].

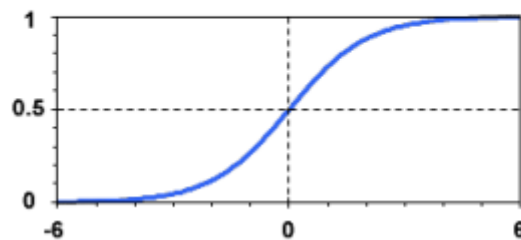


Εικόνα 7: Τοπολογία MFL νευρωνικού δικτύου [39]

Στην προηγούμενη εικόνα, αναπαριστάται η αρχιτεκτονική ενός MFL νευρωνικού δικτύου. Τα στρώματα που δεν συνδέονται άμεσα με το περιβάλλον ονομάζονται κρυφά. Το πρώτο στρώμα, το στρώμα εισόδου, μπορεί να θεωρηθεί ως ένα αυτόνομο στρώμα στο δίκτυο, δεδομένου ότι η μόνη του λειτουργία είναι να μεταδίδει τα σήματα εισόδου στα ανώτερα στρώματα, χωρίς καμία επεξεργασία στις εισόδους. Όπως έγινε αντιληπτό, ένα perceptron με ένα στρώμα και μία είσοδο παράγει περιοχές απόφασης υπό τη μορφή ημιεπιπέδων. Με την προσθήκη ενός ακόμη στρώματος, κάθε νευρώνας λειτουργεί ως τυπικό perceptron για τις εξόδους των νευρώνων στο πρόσθιο στρώμα, οπότε η έξοδος του δικτύου μπορεί να εκτιμήσει κυρτές περιοχές απόφασης, που προκύπτουν από την τομή των ημιεπιπέδων που παράγονται από τους νευρώνες. Με τη σειρά του, ένα perceptron τριών στρωμάτων μπορεί να παράγει αυθαίρετες περιοχές απόφασης. Όσον αφορά τη συνάρτηση

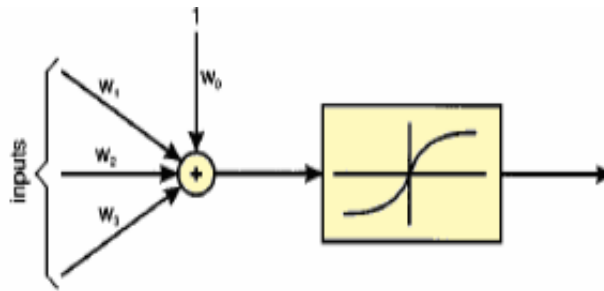
ενεργοποίησης των νευρώνων, έχει διαπιστωθεί ότι τα πολυστρωματικά δίκτυα δεν παρέχουν αύξηση της υπολογιστικής ισχύος σε σύγκριση με τα δίκτυα με ένα στρώμα, εάν οι συναρτήσεις ενεργοποίησης είναι γραμμικές, διότι μια γραμμική συνάρτηση γραμμικών συναρτήσεων είναι επίσης γραμμική συνάρτηση. Η ισχύς του πολυστρωματικού perceptron προέρχεται ακριβώς από τις μη γραμμικές συναρτήσεις ενεργοποίησης. Σχεδόν οποιαδήποτε μη γραμμική συνάρτηση μπορεί να χρησιμοποιηθεί για το σκοπό αυτό, εκτός από τις πολωνυμικές συναρτήσεις. Επί του παρόντος, οι συναρτήσεις που χρησιμοποιούνται πιο συχνά σήμερα είναι το μονοπολικό (ή λογιστικό) και το διπολικό σιγμοειδές. Μπορεί να σημειωθεί ότι, οι σιγμοειδείς συναρτήσεις δρουν περίπου γραμμικά για μικρές απόλυτες τιμές του επιχειρήματος και κορεσμένες, αναλαμβάνοντας κατά κάποιο τρόπο το ρόλο του κατωφλίου για υψηλές απόλυτες τιμές του επιχειρήματος. Επιπρόσθετα, έχει αποδειχθεί ότι ένα δίκτυο με ένα κρυφό στρώμα είναι σε θέση να προσεγγίσει οποιαδήποτε συνεχή συνάρτηση [36].

$$f(s) = \frac{1}{1 + e^{-s}}.$$



Εικόνα 8: Μονοπολικό σιγμοειδές, συνάρτηση ενεργοποίησης [36]

Οι προηγούμενες επισημάνσεις, δικαιολογούν την ιδιότητα του πολυστρωματικού perceptron να λειτουργεί ως καθολικός προσεγγιστής. Επίσης, με την εφαρμογή του θεωρήματος Stone-Weierstrass στο νευρωνικό δίκτυο, αποδείχθηκε ότι μπορούν να υπολογίζουν ορισμένες πολωνυμικές εκφράσεις: αν υπάρχουν δύο δίκτυα που υπολογίζουν ακριβώς δύο συναρτήσεις  $f_1$ , δηλαδή  $f_2$ , τότε υπάρχει ένα μεγαλύτερο δίκτυο που υπολογίζει ακριβώς μια πολωνυμική έκφραση των  $f_1$  και  $f_2$ . Το Multi Perceptron είναι ο πιο γνωστός και πιο συχνά χρησιμοποιούμενος τύπος νευρωνικών δικτύων είναι εκπαιδευμένες μονάδες του τύπου που φαίνεται στην Εικόνα 9. Κάθε μία από αυτές τις μονάδες σχηματίζει ένα σταθμισμένο άθροισμα των εισόδων της στις οποίες προστίθεται μια σταθερά. Το ποσό αυτό περνάει στη συνέχεια μέσα από μια μη γραμμική συνάρτηση, η οποία ονομάζεται συνάρτηση ενεργοποίησης. Οι περισσότερες μονάδες διασυνδέονται κατά «feed forward», δηλαδή διασυνδέσεις οι οποίες σχηματίζουν ένα βρόχο όπως φαίνεται στην Εικόνα 7 [36].



Εικόνα 9: multi-unit perceptron [36]

### 2.3.3 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Network)

Ένα Συνελκτικό Νευρωνικό Δίκτυο (ΣΝΔ ή Convolutional Neural Network ή CNN) είναι μια ειδική περίπτωση του ΤΝΔ που συνήθως χρησιμοποιείται για την αναγνώριση εικόνων, την αναγνώριση προτύπων ή και την όραση υπολογιστών. Αυτά τα δίκτυα αξιοποιούν αρχές από τη γραμμική άλγεβρα, ιδίως τον πολλαπλασιασμό πινάκων, για τον εντοπισμό μοτίβων μέσα σε μια εικόνα. Τα ΣΝΔ χαρακτηρίζονται από τη χρήση συνελκτικών στρώματων. Αυτά τα στρώματα εφαρμόζουν λειτουργίες συνελίξεων στα δεδομένα εισόδου, χρησιμοποιώντας μικρά φίλτρα, ή αλλιώς πυρήνες, για τη σάρωση της εισόδου με εντοπισμένο τρόπο. Τα συνελκτικά στρώματα χρησιμοποιούνται για την αυτόματη εκμάθηση ιεραρχικών χαρακτηριστικών από τα δεδομένα. Επιπρόσθετα, ένα ΣΝΔ αντί να χρησιμοποιεί μεμονωμένα εικονοστοιχεία ως είσοδο, τροφοδοτεί τμήματα μιας εικόνας σε συγκεκριμένους κόμβους στο επόμενο επίπεδο κόμβων (και όχι σε όλους τους κόμβους), διατηρώντας έτσι το χωρικό πλαίσιο από το οποίο εξήχθη ένα χαρακτηριστικά. Αυτά τα τμήματα κόμβων μαθαίνουν να εξάγουν συγκεκριμένα χαρακτηριστικά και είναι γνωστά ως φίλτρα συνελκτικής τροφοδότησης [12], [14]. Τα ΣΝΔ έχουν σημειώσει εξαιρετική επιτυχία σε εφαρμογές υπολογιστικής όρασης, όπως η αναγνώριση προσώπων, η ανίχνευση αντικειμένων και η τροφοδότηση της όρασης στη ρομποτική [5]. Ακόμα, χρησιμοποιούνται για σημασιολογική κατάτμηση, όπου κάθε εικονοστοιχείο σε μια εικόνα ταξινομείται σε μια συγκεκριμένη κατηγορία αντικειμένων. Αυτό είναι πολύτιμο σε εφαρμογές όπως η αυτόνομη οδήγηση και η ανάλυση ιατρικών εικόνων. Ολοκληρώνοντας, η εκπαίδευση βαθιών ΣΝΔ απαιτεί σημαντικούς υπολογιστικούς πόρους και μεγάλα σύνολα δεδομένων. Η υπερπροσαρμογή μπορεί επίσης να αποτελέσει πρόβλημα, γι' αυτό και χρησιμοποιούνται συχνά τεχνικές όπως η εγκατάλειψη και η αύξηση δεδομένων. Τα ΣΝΔ έχουν φέρει επανάσταση στον τομέα της όρασης υπολογιστών και έχουν εφαρμογές σε ένα ευρύ φάσμα τομέων, από την υγειονομική περίθαλψη και την αυτοκινητοβιομηχανία έως την ψυχαγωγία και την ασφάλεια. Η ικανότητά τους να μαθαίνουν αυτόματα ιεραρχικά χαρακτηριστικά τα καθιστά ανεκτίμητα για την εξαγωγή σημαντικών πληροφοριών από οπτικά δεδομένα.

### 2.3.4 Μηχανική Μάθηση στα έργα τέχνης

Ο άνθρωπος στην καθημερινότητα του συναντά πληθώρα εφαρμογών μηχανικής μάθησης. Μερικά ευρέως γνωστά παραδείγματα MM είναι η αναγνώριση ομιλίας όπου χρησιμοποιείται στις φωνητικές αναζητήσεις, η αυτόματη εξυπηρέτηση πελατών μέσω των γνωστών “chatbots” ή ακόμα και η όραση υπολογιστών, η οποία επιτρέπει στους υπολογιστές να αντλούν σημαντικές πληροφορίες από ψηφιακές εικόνες, βίντεο και άλλες οπτικές εισροές και στη συνέχεια να αναλαμβάνουν την κατάλληλη δράση. Καθώς λοιπόν, οι αλγόριθμοι MM έχουν εξελιχθεί την τελευταία δεκαετία, έχει αυξηθεί το ενδιαφέρον τους για την εφαρμογή τους σε ποικίλα προβλήματα Όρασης Υπολογιστών, ένα εκ των οποίων αφορά την μελέτη έργων τέχνης. Είναι γεγονός πως, η μελέτη των έργων τέχνη χαρακτηρίζεται ως μία σχετικά δύσκολη δραστηριότητα, η οποία θα μπορούσε να επωφεληθεί από τους αλγόριθμους MM. Με δεδομένα τα παραπάνω, ακολουθεί ένα παράδειγμα εφαρμογής MM σε έργα τέχνης.

Η επιτυχημένη εφαρμογή αφορά την ανάλυση των πινάκων ενός καλλιτέχνη. Στόχος της συγκεκριμένης εφαρμογής είναι να συνεισφέρει στο έργο των ιστορικών τέχνης και των επιστημών, οι οποίοι επιδιώκουν να κατανοήσουν κρυμμένα μηνύματα πίσω από τα έργα τέχνης, όπως για παράδειγμα κοινές επιρροές σε πίνακες διαφορετικών καλλιτεχνών, αντικείμενα και χαρακτηριστικά που επαναλαμβάνονται. Ακόμα, μέσω τεχνικών MM για την ανάλυση ενός πίνακα ζωγραφικής μπορεί να πιστοποιηθεί η αυθεντικότητα ενός έργου τέχνης. Για την επίτευξη του έργου αυτού, πρώτο βήμα αποτέλεσε η δημιουργία κατάλληλου συνόλου δεδομένων, το οποίο αποτελούταν από ένα σύνολο έργων ζωγραφικής από επτά διαφορετικού καλλιτέχνης. Μετά από διάφορες δοκιμές για την ταξινόμηση, χρησιμοποιήθηκε ο αλγόριθμος Support Vector Machine. Έπειτα, για την ανάδειξη της ομοιότητας ανάμεσα στους καλλιτέχνης εφαρμόστηκαν διαφορετικές μετρικές ομοιότητας. Τα αποτελέσματα που πάρθηκαν, ανέδειξαν την ομοιότητα μεταξύ διαφορετικών καλλιτεχνών. Συγκρίνοντας την απόδοση των διαφόρων χαρακτηριστικών, διαπιστώθηκε ότι το HOG2x2 έχει την καλύτερη απόδοση, εφόσον μπορεί να καταγράψει πιο ισορροπημένα σημεία κλειδιά σε όλες τις εικόνες. Ολοκληρώνοντας διαπιστώθηκε ότι, οι επιδόσεις των διαφορετικών χαρακτηριστικών μπορεί να υποδηλώνουν τις προτιμήσεις και το στυλ των καλλιτεχνών [13].

## 2.4 Βαθιά Μάθηση (Deep Learning)

Η Βαθιά Μάθηση (BM) είναι ένας υποκλάδος της Μηχανικής Μάθησης και της Τεχνητής Νοημοσύνης. Πιο συγκεκριμένα, η BM αναφέρεται σε μια κατηγορία τεχνικών MM που εκμεταλλεύονται ιεραρχικές αρχιτεκτονικές επιπέδων επεξεργασίας πληροφοριών, για την εκμάθηση χαρακτηριστικών και την αναγνώριση προτύπων. Με άλλα λόγια, είναι



μια προσέγγιση μηχανικής μάθησης που μπορεί να μαθαίνει από μη δομημένα ή μη επισημασμένα δεδομένα και η αναπαράσταση να γίνεται βασισμένη στη γνώση του ανθρώπινου εγκεφάλου. Για το λόγο αυτό, βασίζεται στα Νευρωνικά Δίκτυα, τα οποία μπορούν να μιμηθούν τον ανθρώπινο εγκέφαλο και να εκτελέσουν αναλυτική μάθηση αναλύοντας δεδομένα όπως κείμενο, εικόνες και ήχο. Τα μοντέλα ΒΒ δομούνται από πολλαπλά συνδεδεμένα στρώματα, ως αποτέλεσμα μπορούν να εκφράζουν αποτελεσματικότερα συναρτήσεις επαυξημένης πολυπλοκότητας [1], [15].

Τα μοντέλα ΒΜ διαθέτουν πληθώρα πλεονεκτημάτων έναντι των κλασικών αλγορίθμων ΜΜ. Το βασικό τους πλεονέκτημα αφορά την ικανότητα τους να μαθαίνουν τα σχετικά χαρακτηριστικά απευθείας από τα δεδομένα. Επιπλέον, το γεγονός ότι οι αλγόριθμοι ΒΜ είναι σε θέση χειριστούν μεγάλα και πολύπλοκα σύνολα δεδομένων, καθιστά τη ΒΜ χρήσιμο εργαλείο για την εξαγωγή συμπερασμάτων από τέτοιου είδους δεδομένα. Επιπρόσθετα, έχει αποδειχθεί ότι οι αλγόριθμοι βαθιάς μάθησης επιτυγχάνουν κορυφαίες επιδόσεις σε ένα ευρύ φάσμα προβλημάτων, όπως η αναγνώριση εικόνας και ομιλίας, η επεξεργασία φυσικής γλώσσας και η όραση υπολογιστών. Ακόμα, η ΒΜ μπορεί να χρησιμοποιηθεί για την πραγματοποίηση προβλέψεων σχετικά με μελλοντικά γεγονότα ή τάσεις, οι οποίες μπορούν να βοηθήσουν τους οργανισμούς να σχεδιάσουν το μέλλον και να λάβουν στρατηγικές αποφάσεις. Τα πλεονεκτήματα που προαναφέρθηκαν είναι μερικά από αυτά που συγκεντρώνει η ΒΜ. Συνολικά, επισημαίνεται ότι η ΒΜ διαθέτει πλεονεκτήματα που αφορούν την αυτόματη μάθηση χαρακτηριστικών, το χειρισμό μεγάλων και πολύπλοκων δεδομένων, τη βελτιωμένη απόδοση, το χειρισμό μη γραμμικών σχέσεων, δομημένων και μη δομημένων δεδομένων, την προγνωστική μοντελοποίηση, το χειρισμό ελλειπών δεδομένων και διαδοχικών δεδομένων, την επεκτασιμότητα και την ικανότητα γενίκευσης [16].

Συνεπακόλουθα, η ΒΜ παρά τα θετικά της εμφανίζει και ορισμένα μειονεκτήματα, τα οποία οφείλουν να ληφθούν υπόψη. Αρχικά, η εκπαίδευση μοντέλων ΒΜ απαιτεί σημαντικούς υπολογιστικούς πόρους, όπως ισχυρές κάρτες γραφικών μεγάλα ποσοστά μνήμης, ως αποτέλεσμα το υπολογιστικό κόστος είναι υψηλό. Στη συνέχεια, εάν ένα μοντέλο εκπαιδευτεί εξαιρετικά καλά στα δεδομένα εκπαίδευσης, αλλά αποδίδει ελλιπώς σε νέα δεδομένα τότε παρατηρείται η υπερπροσαρμογή. Επίσης, μοντέλα με πολλά επίπεδα ενδέχεται να είναι πολύπλοκα και δύσκολα ερμηνεύσιμα. Τέλος, εάν τα δεδομένα δεν είναι ποιητικά, η απόδοση των αλγορίθμων ΒΜ επηρεάζεται αρνητικά [16].

#### **2.4.1 Εφαρμογές Βαθιάς Μάθησης**

Στην υποενότητα που ακολουθεί, αναπτύσσονται σε παραδείγματα τρεις σχετικά πρόσφατες εφαρμογές της ΒΜ, οι οποίες επιβεβαιώνουν την αναγκαιότητα και τη χρησιμότητα τεχνικών ΒΜ. Η πρώτη εφαρμογή αφορά αυτόματη ανίχνευση και αναγνώριση

ελαττωμάτων μεταλλικών επιφανειών με χρήση συνελκτικών νευρωνικών δικτύων. Η ανίχνευση μεταλλικών ελαττωμάτων πραγματοποιείται συνήθως σε πολύπλοκα βιομηχανικά σενάρια, παρουσιάζοντας ένα ενδιαφέρον αλλά δύσκολο πρόβλημα. Οι παραδοσιακές μέθοδοι βασίζονται σε τεχνικές επεξεργασίας εικόνας ή ρηχής μηχανικής μάθησης, αλλά αυτές μπορούν να ανιχνεύσουν ατέλειες μόνο υπό συγκεκριμένες συνθήκες ανίχνευσης, όπως εμφανή περιγράμματα ατελειών με ισχυρή αντίθεση και χαμηλό θόρυβο, σε συγκεκριμένες κλίμακες ή υπό συγκεκριμένες συνθήκες φωτισμού. Όμως στην συγκεκριμένη εφαρμογή, η αυτόματη ανίχνευση μεταλλικών ελαττωμάτων επιτεύχθηκε μέσω μίας διπλής διαδικασίας που εντοπίζει με ακρίβεια και ταξινομεί τα ελαττώματα που εμφανίζονται σε εικόνες εισόδου που έχουν ληφθεί από πραγματικά βιομηχανικά περιβάλλοντα. Για το λόγο αυτό, σχεδιάστηκε μια νέα αρχιτεκτονική αυτοκωδικοποιητή με καταιγισμό (CASAE) για την τμηματοποίηση και τον εντοπισμό ελαττωμάτων. Το κλιμακωτό δίκτυο μετασχηματίζει την εικόνα ελαττωμάτων εισόδου σε μια μάσκα πρόβλεψης ανά εικονοστοιχείο με βάση τη σημασιολογική κατάτμηση. Οι περιοχές ελαττωμάτων των τμηματοποιημένων αποτελεσμάτων ταξινομούνται στις συγκεκριμένες κλάσεις τους μέσω ενός συμπαγούς νευρωνικού δικτύου συνελκτικής ανάλυσης (ΣΝΔ). Τα μεταλλικά ελαττώματα υπό διάφορες συνθήκες μπορούν να ανιχνευθούν με επιτυχία χρησιμοποιώντας ένα βιομηχανικό σύνολο δεδομένων [17].

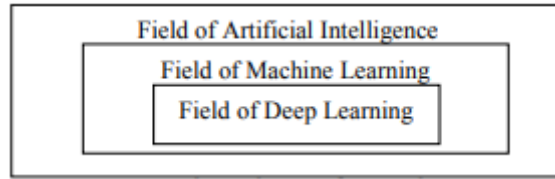
Η δεύτερη εφαρμογή αφορά την ανίχνευση ρωγμών σε επίπεδο pixel με χρήση βαθιάς πρόβλεψης τοπικών μοτίβων για εφαρμογή σε ρομπότ. Η ρομποτική ανίχνευση ρωγμών σε γέφυρες από τσιμέντο με βάση την όραση είναι ένα σημαντικό έργο για τη διατήρηση αυτών των περιουσιακών στοιχείων, καθώς και της ανθρώπινης ασφάλειάς τους. Η συμβατική μέθοδος οπτικής επιθεώρησης από τον άνθρωπο είναι χρονοβόρα και οικονομικά αναποτελεσματική. Ειδικότερα στην πράξη, η ανίχνευση ρωγμών εξακολουθεί να αποτελεί πρόκληση από τις ακόλουθες δύο απόψεις. Πρώτον, η απόδοση ανίχνευσης διαταράσσεται από θορύβους και ακαταστασίες του περιβάλλοντος και δεύτερον η απαίτηση υψηλής ακρίβειας ανά εικονοστοιχείο είναι δύσκολο να επιτευχθεί. Για την αντιμετώπιση αυτών των περιορισμών, στην συγκεκριμένη εφαρμογή εξετάστηκαν τρία βήματα. Πρώτον, κατασκευάστηκε ένας τοπικός προγνωστικός δείκτης μοτίβου (LPP) με τη χρήση νευρωνικών δικτύων συνελίξεων (ΣΝΔ), ο οποίος μπορεί να εξάγει διακριτικά χαρακτηριστικά των εικόνων. Δεύτερον, κάθε εικονοστοιχείο ταξινομήθηκε αποτελεσματικά σε κατηγορίες ρωγμών ή μη ρωγμών από τον LPP, χρησιμοποιώντας ως πλαίσιο ένα patch με κέντρο το εικονοστοιχείο. Τέλος, η έξοδος του ΣΝΔ, δηλαδή ο χάρτης εμπιστοσύνης, επεξεργάζεται εκ των υστέρων για να προκύψουν οι περιοχές ρωγμών. Τα πειραματικά αποτελέσματα της εφαρμογής απέδειξαν την καλή απόδοση της μεθόδου [18].

Η τρίτη και τελευταία εφαρμογή αναφέρεται στην ανίχνευση ακμών σε πίνακες ζωγραφικές με τη χρήση συνελκτικών νευρωνικών δικτύων. Η ακριβής ανίχνευση ρωγμών

σε πίνακες ζωγραφικής, οι οποίοι γενικά απεικονίζουν πλούσιο και ποικίλο περιεχόμενο, είναι ένα δύσκολο έργο. Αντίστοιχα με τις προηγούμενες εφαρμογές έτσι και εδώ, οι παραδοσιακές μέθοδοι ανίχνευσης ρωγμών είναι συχνά ελλιπείς σε πρόσφατες αποκτήσεις ζωγραφικών έργων, καθώς δεν είναι επαρκώς προσαρμοσμένες σε υψηλές αναλύσεις και δεν χρησιμοποιούν τις άλλες μεθόδους απεικόνισης που συχνά είναι διαθέσιμες. Επιπλέον, πολλοί πίνακες απεικονίζουν μια σύνθετη ή ακατάστατη σύνθεση, περιπλέκοντας σημαντικά την ακριβή ανίχνευση ρωγμών όταν χρησιμοποιείται μόνο φωτογραφικό υλικό. Στην παρούσα εφαρμογή προτείνεται ένας γρήγορος αλγόριθμος ανίχνευσης ρωγμών που βασίζεται σε βαθιά νευρωνικά δίκτυα συνελίξεων (ΣΝΔ), ο οποίος είναι ικανός να συνδυάζει διάφορες μεθόδους απεικόνισης, όπως κανονικές φωτογραφίες, υπέρυθρες φωτογραφίες και εικόνες ακτινών Χ. Επιπλέον, προτείνεται μια αποτελεσματική λύση για τη βελτίωση του εντοπισμού των πραγματικών ορίων της ρωγμής με βάση το ΣΝΔ και επεκτείνεται η αρχιτεκτονική του ΣΝΔ έτσι ώστε να παραλείπονται οι περιοχές όπου δεν έχει νόημα να εκτελούνται ακριβά μοντέλα μάθησης. Η on-line μέθοδος είναι ικανή να μαθαίνει συνεχώς από τα νεοαποκτηθέντα οπτικά δεδομένα, βελτιώνοντας έτσι περαιτέρω τα αποτελέσματα ταξινόμησης καθώς περισσότερα δεδομένα γίνονται διαθέσιμα. Ολοκληρώνοντας, η μελέτη παρουσίασε βελτιώσεις σε σχέση με την τελευταία λέξη της τεχνολογίας στις μεθόδους ανίχνευσης ρωγμών και απέδειξε τις δυνατότητες της προτεινόμενης μεθόδου μας για την υποβοήθηση των συντηρητών έργων τέχνης [21].

## 2.5 Μηχανική Μάθηση και Βαθιά Μάθηση

Πράγματι, η MM είναι ένα από τα πεδία του σύγχρονου κόσμου της πληροφορικής. Η μάθηση είναι μια φυσική ανθρώπινη συμπεριφορά, η οποία έχει γίνει βασική πτυχή και των μηχανών. Οι παραδοσιακοί αλγόριθμοι MM έχουν εφαρμοστεί σε πολλούς τομείς εφαρμογών. Οι ερευνητές όμως, έχουν καταβάλει πολλές προσπάθειες για τη βελτίωση της ακρίβειας αυτών των αλγορίθμων μηχανικής μάθησης. Με στόχο τα βέλτιστα αποτελέσματα, δόθηκε μια άλλη διάσταση που οδηγεί στην έννοια της βαθιάς μάθησης. Με άλλα λόγια, Η βαθιά μάθηση αποτελεί ένα υποσύνολο της μηχανικής μάθησης. Όπως φαίνεται και στην Εικόνα 10, η BM είναι υποσύνολο της MM, η οποία με τη σειρά της είναι υποσύνολο της Τεχνητής Νοημοσύνης. Ανεπίσημα, ο όρος «τεχνητή νοημοσύνη» εφαρμόζεται όταν μια μηχανή είναι σε θέση να εκτελεί λειτουργίες που οι άνθρωποι συνδέουν με άλλα ανθρώπινα μυαλά, όπως η «μάθηση» και η «επίλυση προβλημάτων». Η μάθηση αποτελεί ζωτική πτυχή των μηχανών. Ως εκ τούτου, η μηχανική μάθηση είναι ένα υποπεδίο της TN [20].



**Εικόνα 10: Τεχνητή Νοημοσύνη, Μηχανική Μάθηση και Βαθιά Μάθηση [20]**

Συνεπακόλουθα, η ΒΜ και τα νευρωνικά δίκτυα έχουν τη τάση να χρησιμοποιούνται εναλλακτικά στη συζήτηση, γεγονός που μπορεί να δημιουργήσει σύγχυση. Αξίζει να επισημανθεί ότι, η ΒΜ αναφέρεται στα βαθιά τεχνητά νευρωνικά δίκτυα. Για την ακρίβεια, ο όρος «Βαθιά» στη ΒΜ αναφέρεται στο βάθος των επιπέδων που έχει ένα ΝΔ. Ένα νευρωνικό δίκτυο που αποτελείται από περισσότερα από τρία στρώματα - τα οποία περιλαμβάνουν τις εισόδους και την έξοδο - μπορεί να θεωρηθεί αλγόριθμος βαθιάς μάθησης. Το βαθύ δίκτυο έχει περισσότερα από ένα κρυφά στρώματα, ενώ ένα ρηχό δίκτυο έχει μόνο ένα. Ένα νευρωνικό δίκτυο που έχει μόνο δύο ή τρία στρώματα είναι απλώς ένα βασικό νευρωνικό δίκτυο [12], [20].

### 3 Περιγραφή της Γλώσσας Matlab

Η Προγραμματιστική Γλώσσα που επιλέχθηκε για την υλοποίηση της παρούσας εργασίας, είναι η γλώσσα Matlab, έκδοσης R2023b. Η πρώτη έκδοση του Matlab ολοκληρώθηκε στα τέλη του 1970 και εφευρέτης της συγκεκριμένης προγραμματιστικής γλώσσας είναι μαθηματικός και προγραμματιστής Cleve Moler. Αρχικά (πριν από την έκδοση 1.0), το Matlab δεν ήταν μια γλώσσα προγραμματισμού, ήταν μια απλή διαδραστική αριθμομηχανή πινάκων. Δεν υπήρχαν προγράμματα, ούτε εργαλειοθήκες, ούτε γραφικά. Η πρώτη πρόμη έκδοση του Matlab ολοκληρώθηκε στα τέλη της δεκαετίας του '70. Το λογισμικό αποκαλύφθηκε για πρώτη φορά στο κοινό τον Φεβρουάριο του 1979. Εκείνη την εποχή, το Matlab διανεμήθηκε δωρεάν σε πανεπιστήμια. Ο Moler άφησε αντίγραφα στα πανεπιστήμια που επισκεπτόταν και το λογισμικό ανέπτυξε μεγάλη απήχηση στα μαθηματικά τμήματα των πανεπιστημίων. Αργότερα, σε συνεργασία με τον John N. Little, ο Cleve Moler αποφάσισε να επαναπρογραμματίσει το Matlab σε γλώσσα C και να το διαθέσει στην αγορά. Έτσι, αναπτύχθηκε η γλώσσα προγραμματισμού Matlab και διαμορφώθηκαν χαρακτηριστικά και εργαλειοθήκες [32]. Έκτοτε έχει εξελιχθεί σε ένα ολοκληρωμένο εργαλείο λογισμικού που αναπτύσσεται και συντηρείται από την MathWorks.

Το Matlab, λοιπόν, δημιουργήθηκε για να διευκολύνει τους μηχανικούς και του επιστήμονες. Η πλατφόρμα προγραμματισμού Matlab έχει πολλά πλεονεκτήματα έναντι άλλων τεχνικών ή γλωσσών. Η θεμελιώδης δομή έχει ως βασικό στοιχείο δεδομένων έναν πίνακα. Ένας απλός ακέραιος αριθμός αναγνωρίζεται ως πίνακας μιας γραμμής και μιας στήλης. Επιπρόσθετα, χρησιμοποιεί συναρτήσεις, γνώστες και εύκολες για απομνημόνευση. Το Matlab ειδικεύεται στα μαθηματικά, όποτε αβίαστα μπορούν να αποτυπωθούν και οι πιο σύνθετες μαθηματικές εκφράσεις. Επιπλέον, το Matlab διαθέτει εργαλειοθήκες, οι οποίες παρέχουν επαγγελματικά αναπτυγμένες, αυστηρά δοκιμασμένες, δοκιμασμένες στο πεδίο και πλήρως τεκμηριωμένες λειτουργίες για ένα ευρύ φάσμα επιστημονικών και μηχανολογικών εφαρμογών. Ακόμα, το Matlab παρέχει ένα διαδραστικό περιβάλλον, επιτρέποντας στους χρήστες να εκτελούν κώδικα μία γραμμή τη φορά και να βλέπουν τα αποτελέσματα αμέσως. Αυτό το καθιστά εξαιρετικό για την κατασκευή πρωτοτύπων και τη διερεύνηση δεδομένων. Τέλος, το Matlab διαθέτει δικές του εφαρμογές, οι οποίες μπορούν αυξήσουν σημαντικά τη λειτουργικότητα του. Οι εφαρμογές του Matlab είναι διαδραστικές εφαρμογές που συνδυάζουν άμεση πρόσβαση σε μεγάλες συλλογές αλγορίθμων με άμεση οπτική ανατροφοδότηση, αφού προσφέρει ισχυρά εργαλεία για τη δημιουργία δισδιάστατων και τρισδιάστατων γραφικών παραστάσεων [22], [23].

### 3.1 Matlab και Μηχανική Μάθηση

Το Matlab παρέχει ένα ιδανικό περιβάλλον για τη MM. Αυτό συμβαίνει λόγω ότι, στο Matlab χρειάζονται λιγότερες γραμμές κώδικα για να δημιουργηθεί ένα μοντέλο μηχανικής μάθησης ή βαθιάς μάθησης, χωρίς να χρειάζεται το άτομο που θα τις χρησιμοποιήσει να είναι ειδικός στις τεχνικές αυτές. Η σύνταξη του κώδικα, στη συγκεκριμένη προγραμματιστική γλώσσα, είναι διαισθητική και φιλική προς το χρήστη, ιδιαίτερα για μηχανικούς, επιστήμονες και ερευνητές που μπορεί να μην έχουν εκτεταμένη εμπειρία στον προγραμματισμό. Το διαδραστικό του περιβάλλον επιτρέπει στους χρήστες να πειραματιστούν με αλγορίθμους και να οπτικοποιήσουν γρήγορα τα αποτελέσματα. Ακόμα, Το Matlab παρέχει ένα ευρύ φάσμα εργαλειαθικών και βιβλιοθηκών για τη μηχανική μάθηση, συμπεριλαμβανομένων των εργαλειαθικών στατιστικής και μηχανικής μάθησης, της εργαλειαθικής βαθιάς μάθησης και άλλων. Πιο συγκεκριμένα, η εργαλειαθήκη Deep Learning Toolbox ενσωματώνει ένα πλαίσιο για τη σύνθεση και εκτέλεση βαθιών νευρωνικών δικτύων με αλγορίθμους, εκπαιδευμένα μοντέλα και εφαρμογές. Αυτά τα εργαλεία καλύπτουν ένα ευρύ φάσμα αλγορίθμων και τεχνικών μηχανικής μάθησης. Επιπροσθέτως, οι εφαρμογές και τα διαγράμματα υποστηρίζουν τους χρήστες να οπτικοποιούν τις ενεργοποιήσεις, να επεξεργάζονται τις αρχιτεκτονικές των δικτύων και να παρακολουθούν την πρόοδο της προετοιμασίας. Η οπτικοποίηση των δεδομένων είναι καθοριστικής σημασίας για την κατανόηση της συμπεριφοράς των μοντέλων μηχανικής μάθησης και την ανάλυση δεδομένων [23]. Τέλος, το Matlab παρέχει προ-εκπαιδευμένα μοντέλα βαθιάς μάθησης που μπορούν να ρυθμιστούν λεπτομερώς για συγκεκριμένες εργασίες. Αυτό είναι ιδιαίτερα χρήσιμο για όσους θέλουν να αξιοποιήσουν μοντέλα τελευταίας τεχνολογίας χωρίς να ξεκινήσουν από το μηδέν.

## 4 Υπερφασματικά Δεδομένα

Τα υπερφασματικά δεδομένα με τα γραμμικά σχέδια, τα οποία χρησιμοποιήθηκαν για τη δημιουργία του dataset, πάρθηκαν από το Τμήμα Συντήρησης Αρχαιοτήτων και Έργων Τέχνης του Πανεπιστημίου Δυτικής Αττικής. Στην ουσία πρόκειται για εικόνες που έχουν ληφθεί σε διαφορετικές και σαφώς ορισμένες περιοχές του φάσματος, με διαφορετικούς ανιχνευτές από πρότυπα δείγματα. Τα πρότυπα δείγματα κατασκευάστηκαν σύμφωνα με συγκεκριμένα πρωτόκολλα και απεικονίζονται στην Εικόνα 11.



Εικόνα 11: Τα πρότυπα δείγματα που χρησιμοποιήθηκαν

Καταρχάς, πραγματοποιήθηκαν δύο διαφορετικοί τρόποι λήψης των δειγμάτων. Ο πρώτος τρόπος λήψης έγινε σε απόσταση 58 εκατοστών, του φακού από το καβαλέτο, με πάχος δείγματος 8mm. Κατά συνέπεια, η απόσταση του φακού από την επιφάνεια της ζωγραφικής είναι 57,2 εκατοστά. Ο δεύτερος τρόπος λήψης ολοκληρώθηκε με δύο κουτάκια δειγμάτων σε κάθε λήψη, οπότε η απόσταση είναι ανάλογη σε κάθε περίπτωση. Οι λήψεις αφορούν τα έξι πρώτα χρώματα, τα οποία απεικονίζονται στην Εικόνα 12, με τη σειρά που φαίνονται από κάτω προς τα πάνω, τα χρώματα είναι κάδμιο, κιννάβαρι, μίνιο, αιματίτης, κίτρινη όχρα και κόκκινη όχρα.



Εικόνα 12: Τα χρώματα του δείγματος

Όσον αφορά τις κάμερες που χρησιμοποιήθηκαν αναγράφονται στη συνέχεια. Η πρώτη κάμερα είναι η DSLR NIKON D-70s με φακό Nikon MicroNikkor 60mm, 1:2.8 D, η δεύτερη είναι η υπερφασματική MuSIS HS του ιδρύματος με το δικό της φακό, η τρίτη είναι η θερμογραφική κάμερα InGaAs AVT Goldeye P-008 SWIR Cool, με φακό Nikon MicroNikkor 60mm, 1:2.8 D, τέταρτη και πέμπτη κάμερα είναι η κάμερα MuSIS η καινούργια και η παλιά αντίστοιχα, τις οποίες διαθέτει η Εθνική Πινακοθήκη Μουσείο Αλέξανδρου Σούτσου (ΕΠΜΑΣ).



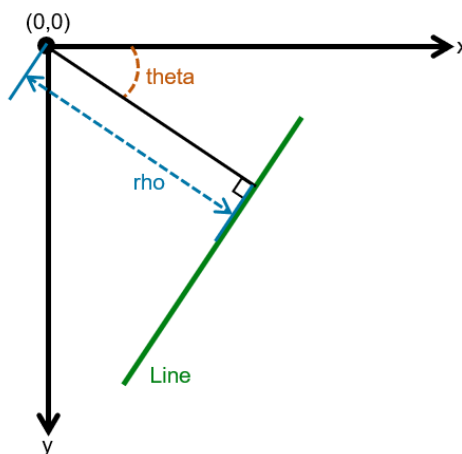
## 5 Προ-επεξεργασία δεδομένων

Πρώτο βήμα για την υλοποίηση της παρούσας διπλωματικής, αποτέλεσε η δημιουργία ενός μοναδικού συνόλου δεδομένων (dataset). Στο κεφάλαιο που ακολουθεί, θα μελετηθούν εις βάθος όλες οι τεχνικές που χρησιμοποιήθηκαν, έτσι ώστε από τα υπερφασματικά δεδομένα με τα γραμμικά προσχέδια, να δημιουργηθούν τρεις κατηγορίες με τμήματα εικόνων (ή αλλιώς patches). Τα patches είναι υπομήματα των αρχικών εικόνων, τα οποία χαρακτηρίζονται από συγκεκριμένα γνωρίσματα, όπως αυτά υποδηλώνονται από τα τρία είδη κατηγοριών που ακολουθούν. Οι τρεις κατηγορίες αφορούν, τμήματα εικόνων όπου το κεντρικό pixel βρίσκεται πάνω σε γραμμή προσχεδίου, τμήματα που περιέχουν γραμμή προσχεδίου και τμήματα όπου δεν περιέχουν καμία γραμμή. Για να κατηγοριοποιηθούν τα τμήματα των υπερφασματικών δεδομένων στις τρεις κατηγορίες, θα πρέπει πρώτα να ανιχνευθούν οι κάθετες και οι οριζόντιες γραμμές.

### 5.1 Ανίχνευση Γραμμών (Line Detection) με Hough Transform

Η ανίχνευση γραμμικών δομών από ψηφιακές εικόνες έχει μακρά ιστορία στην όραση υπολογιστών. Η οργάνωση των γραμμικών δομών είναι ένα πρώιμο αλλά ουσιαστικό βήμα για τη μετατροπή του οπτικού σήματος σε χρήσιμες ενδιάμεσες έννοιες [24]. Ο μετασχηματισμός Hough είναι ένας δημοφιλής ισχυρός στατιστικός αλγόριθμος για την εξαγωγή σφαιρικών χαρακτηριστικών, όπως ευθείες γραμμές, κύκλοι, ελλείψεις κ.λπ., από μια εικόνα, ο οποίος χρησιμοποιείται ευρέως στην όραση υπολογιστών και την αναγνώριση προτύπων. Ο αλγόριθμος αυτός είναι ουσιαστικά μια διαδικασία ψηφοφορίας όπου κάθε σημείο που ανήκει στα πρότυπα ψηφίζει για όλα τα πιθανά πρότυπα που διέρχονται από το σημείο αυτό. Αυτές οι ψήφοι συσσωρεύονται σε έναν συσσωρευτικό πίνακα που ονομάζεται bins, και το μοτίβο που λαμβάνει τις μέγιστες ψήφους αναγνωρίζεται ως το επιθυμητό μοτίβο [25]. Δεδομένης μιας δυαδικής εικόνας ακμών  $N \times N$ , οι ευθείες στο τυπικό μετασχηματισμό Hough, ορίζονται από την εξίσωση  $\rho = x \cos \theta + y \sin \theta$ . Όπως φαίνεται στην Εικόνα 13, η αρχή του συστήματος συντεταγμένων θεωρείται ότι βρίσκεται στο κέντρο του εικονοστοιχείου της άνω αριστερής γωνίας. Επιπλέον, μεταβλητή  $\rho$  είναι η κάθετη απόσταση από την αρχή στην ευθεία. Τέλος, η μεταβλητή  $\theta$  είναι η γωνία της κάθετης προβολής από την αρχή στη γραμμή, μετρούμενη σε μοίρες δεξιόστροφα από τον θετικό άξονα  $x$ . Το εύρος του θήτα είναι  $-90^\circ \leq \theta < 90^\circ$ . Η γωνία της ίδιας της ευθείας είναι  $\theta + 90^\circ$ , επίσης μετρούμενη δεξιόστροφα ως προς τον θετικό άξονα  $x$  [26]. Στον τυπικό μετασχηματισμό Hough (SHT), το  $(x, y)$  δηλώνει τις συντεταγμένες των σημείων της ευθείας, δηλαδή τον χώρο των εικονοστοιχείων, ενώ τα  $P$  και  $\theta$  ορίζονται ως ο χώρος των παραμέτρων. Για όλα τα πεδία παραμέτρων  $(P, \theta)$ , ο αλγόριθμος προσδιορίζει εάν υπάρχουν αρκετές ενδείξεις για μια

πραγματική ευθεία στο συγκεκριμένο εικονοστοιχείο, το οποίο υπολογίζει τις τιμές των παραμέτρων και συσσωρεύει τα πεδία στο χώρο παραμέτρων σύμφωνα με όλα τα εικονοστοιχεία. Ο τυπικός υπολογισμός του μετασχηματισμού Hough αποτελείται από τρία μέρη: (1) υπολογισμός των τιμών των παραμέτρων και συσσωρευση των bins στο χώρο των παραμέτρων- (2) εύρεση των τοπικών μεγίστων που αντιπροσωπεύουν τμήματα γραμμής- και (3) εξαγωγή των τμημάτων γραμμής χρησιμοποιώντας τη γνώση από τις θέσεις των μεγίστων. Επισκέπτεται κάθε εικονοστοιχείο της εικόνας μία φορά. Επομένως, η χρονική πολυπλοκότητά του είναι γραμμική με τον συνολικό αριθμό εικονοστοιχείων, ο οποίος είναι το γινόμενο του πλάτους και του ύψους της εικόνας. Ταυτόχρονα, δεδομένου ότι ο αλγόριθμος πρέπει να συσσωρεύει τις ψήφους κάθε εικονοστοιχείου, η απαίτηση αποθήκευσης μεταβάλλεται μαζί με τον αριθμό των εικονοστοιχείων. Έτσι, η υπολογιστική πολυπλοκότητα του αλγορίθμου και οι απαιτήσεις αποθήκευσης αποτελούν τα κύρια σημεία συμφόρησης του τυπικού μετασχηματισμού Hough. Ως εκ τούτου, έχουν προταθεί πολλές παραλλαγές του αρχικού μετασχηματισμού Hough για την ελάφρυνση του υπολογιστικού και αποθηκευτικού φόρτου [25].



Εικόνα 13: Σχέσεις μεταξύ  $(x, y)$ ,  $\rho$  και  $\theta$  [26]

### 5.1.1 Μεθοδολογία μετασχηματισμού Hough για εντοπισμό γραμμών

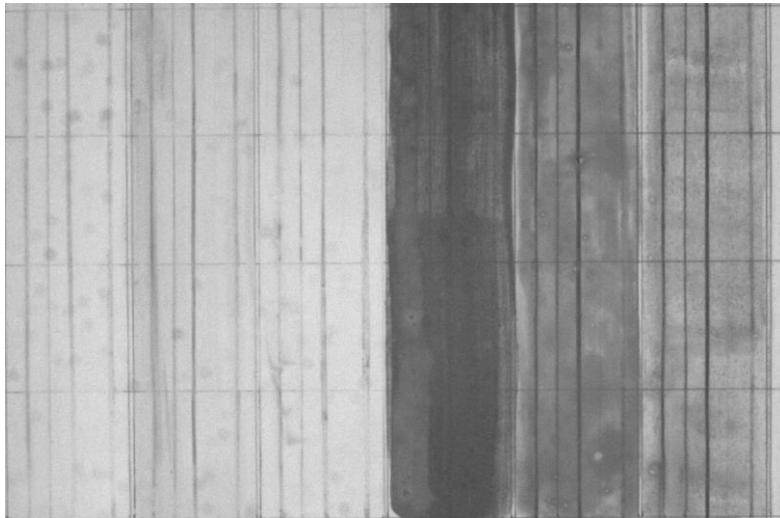
Ο Μετασχηματισμός Hough εφαρμόζεται ως εξής:

- 1° Βήμα: Το πρόγραμμα διαβάζει την επιθυμητή εικόνα από το καθορισμένο αρχείο.
- 2° Βήμα: Μετατρέπει την καθορισμένη εικόνα πραγματικού χρώματος RGB σε εικόνα έντασης σε κλίμακα του γκρι.
- 3° Βήμα: Γίνεται ανίχνευση των ακμών της ασπρόμαυρης εικόνας.
- 4° Βήμα: Γίνεται ο υπολογισμός του μετασχηματισμού Hough της εικόνας όπου έχουν ανιχνευτεί οι ακμές.
- 5° Βήμα: Εντοπίζονται οι κορυφές στον πίνακα μετασχηματισμού του Hough. Επιστρέφεται ένας πίνακας peaks που περιέχει τις συντεταγμένες γραμμής και στήλης των κορυφών.

6<sup>ο</sup> Βήμα: Εξάγονται πληροφορίες σχετικά με κάθε γραμμή, βάσει των κορυφών που εντοπίστηκαν στο προηγούμενο βήμα.

## 5.2 Δοκιμές

Για τον εντοπισμό όσο το δυνατόν περισσότερων γραμμών, έγιναν ποικίλες δοκιμές. Αρχικά, οι δοκιμές έγιναν στην Εικόνα 14 έως ότου βρεθούν οι παράμετροι που αποδίδουν καλύτερα. Έπειτα, οι τεχνικές εφαρμόστηκαν σε επιλεγμένες εικόνες έτσι ώστε να ληφθούν τα επιθυμητά υποτιμήματα εικόνων και στη συνέχεια να ταξινομηθούν.

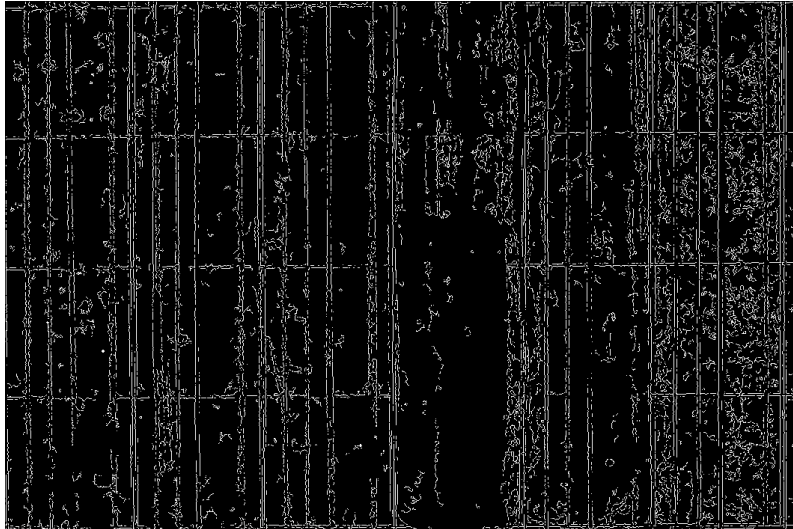


Εικόνα 14: Εικόνα δοκιμών

### 5.2.1 Δοκιμή 1<sup>η</sup>

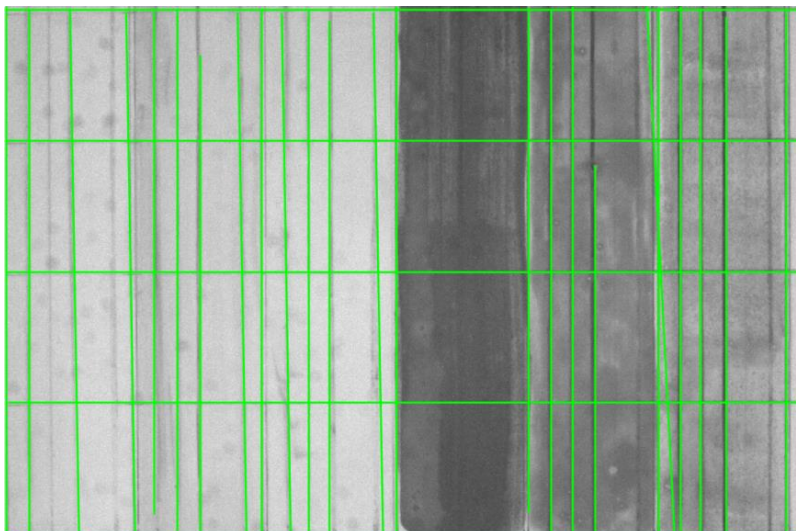
Η πρώτη δοκιμή του μετασχηματισμού Hough έγινε με χρήση του Canny filter, ο κώδικας που υλοποιεί την πρώτη δοκιμή βρίσκεται στο Παράρτημα I.I. Η συνάρτηση **edge()** δέχεται ως ορίσματα την ασπρόμαυρη εικόνα και τον αλγόριθμο ανίχνευσης ακμών Canny και επιστρέφει την εικόνα edges με ανιχνευμένες τις ακμές, όπως φαίνεται στην Εικόνα 15. Έπειτα, εφαρμόζεται ο μετασχηματισμός Hough, μέσω της συνάρτησης **hough()**, έχοντας ως μοναδικό όρισμα την εικόνα με τις ανιχνευμένες ακμές. Ως αποτελέσματα εξόδου προκύπτουν, ένας πίνακας μετασχηματισμού H, όπου οι γραμμές και οι στήλες αντιστοιχούν στις τιμές  $\rho$  και  $\theta$ , και δύο αριθμητικοί πίνακες, ο πίνακας T που περιέχει τις τιμές των γωνιών σε μοίρες μεταξύ του άξονα  $x$  και του διανύσματος  $\rho$  και ο πίνακας R που επιστρέφει την απόσταση από την αρχή έως την ευθεία κατά μήκος ενός διανύσματος κάθετοι στην ευθεία. Επόμενο βήμα, είναι η ανίχνευση κορυφών μέσω της συνάρτησης **houghpeaks()**. Ως ορίσματα δέχεται, το μετασχηματισμό H, τον μέγιστο αριθμό κορυφών που πρέπει να εντοπιστούν και ένας ζεύγος threshold και της τιμής 200, όπου καθορίζει την ελάχιστη τιμή που μπορεί ένα σημείο να θεωρηθεί κορυφή. Η έξοδος που προκύπτει και αποθηκεύεται στην μεταβλητή P, είναι ένας πίνακας Q επί 2, συντεταγμένων γραμμής και στήλης των κορυφών

που βρέθηκαν, όπου η τιμή  $Q$  μπορεί να κυμαίνεται από μηδέν έως το μέγιστο αριθμό κορυφών που δόθηκε ως όρισμα. Τέλος, η `houghlines()` εξάγει πληροφορίες σχετικά με τμήματα γραμμών στην εικόνα `edges`. Τα ορίσματα της είναι η εικόνα `edges`, οι πίνακες  $T$ ,  $R$  και  $P$ , που υπολογίστηκαν προηγουμένως, ένας ζεύγος `FillGap` με τιμή 2000, η οποία καθορίζει πως αν η απόσταση μεταξύ των τμημάτων γραμμής είναι μικρότερη από την καθορισμένη τιμή, η συνάρτηση συγχωνεύει τα τμήματα γραμμής σε ένα ενιαίο τμήμα γραμμής και τέλος, ένα ζεύγος `MinLength` με τιμή 1, το οποίο δηλώνει πως τον ελάχιστο μέγεθος γραμμής για να αποθηκευτεί αρκεί να είναι μεγέθους ενός pixel.



Εικόνα 15: Ανίχνευση Ακμών με τη μέθοδο Canny

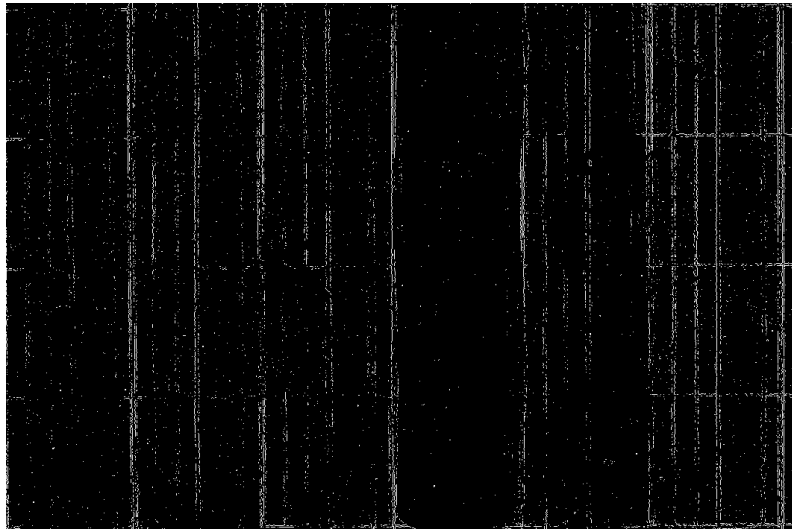
Ολοκληρώνοντας την πρώτη δοκιμή, σχεδιάστηκαν οι γραμμές στην αρχική εικόνα και τα αποτελέσματα καταγράφονται στην Εικόνα 16. Οι γραμμές που εν τέλει καταγράφηκαν, δεν κρίθηκαν ικανοποιητικές. Εκτός του ότι δεν εντοπίστηκαν οι περισσότερες, κάποιες ήταν ημιτελείς ή και διαγώνιες.



Εικόνα 16: Line detection with Canny method

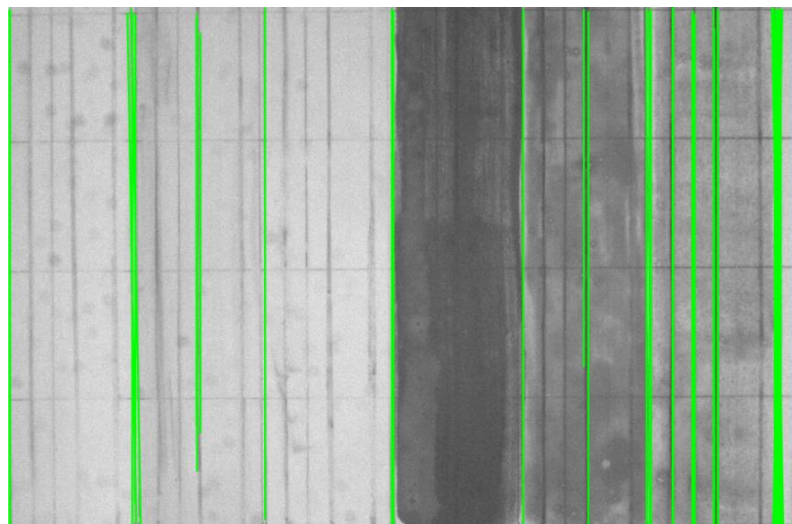
### 5.2.2 Δοκιμή 2<sup>η</sup>

Στη δεύτερη δοκιμή χρησιμοποιήθηκε για την ανίχνευση των ακμών η μέθοδος Sobel. Ο κώδικας με τον οποίο υλοποιήθηκε η δεύτερη δοκιμή καταγράφεται στο Παράρτημα Ι.ΙΙ. Σε αυτή τη δοκιμή ως παράμετροι στην στη συνάρτηση **edge()** δόθηκαν η ασπρόμαυρη εικόνα **img**, η μέθοδος **sobel**, η τιμή 0.04 για **threshold**, όσες ακμές δεν είναι ισχυρότερες από αυτή την τιμή αγνοούνται και η παράμετρος **both** η οποία υποδηλώνει την οριζόντια και κατακόρυφη κατεύθυνση των ακμών που θα ανιχνευθούν. Η εικόνα με τις ακμές εμφανίζεται στην Εικόνα 17. Στη συνάρτηση **houghpeaks()** δόθηκε μία ακόμα παράμετρος ως όρισμα, το **NHoodSize** με τιμή [1, 1], δηλαδή ένα διάνυσμα δύο στοιχείων θετικών περιττών ακεραίων το οποίο αντιπροσωπεύει το μέγεθος της γειτονίας καταστολής. Η γειτονιά καταστολής είναι η γειτονιά γύρω από κάθε κορυφή που μηδενίζεται μετά τον εντοπισμό της κορυφής.



Εικόνα 17: Ανίχνευση Ακμών με τη μέθοδο Sobel

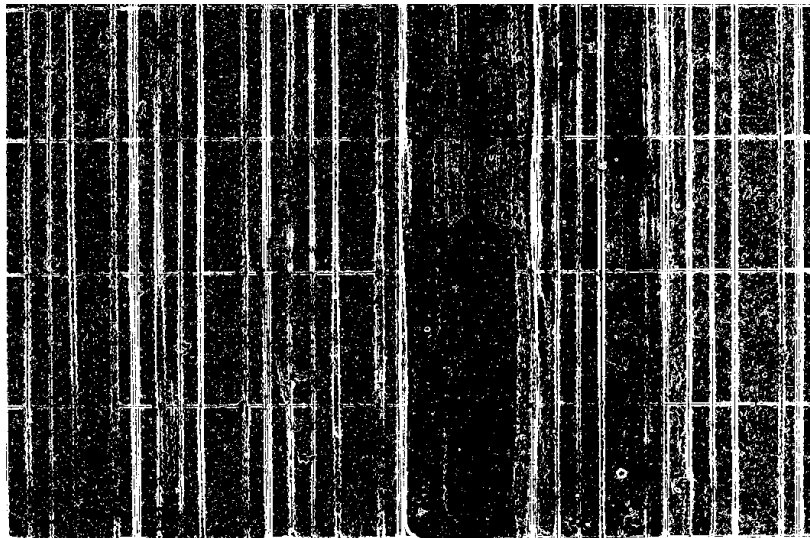
Τα τελικά αποτελέσματα απεικονίζονται στην Εικόνα 18 και σε αυτήν την προσπάθεια τα αποτελέσματα δεν ήταν τα επιθυμητά.



Εικόνα 18: Line detection with Sobel method

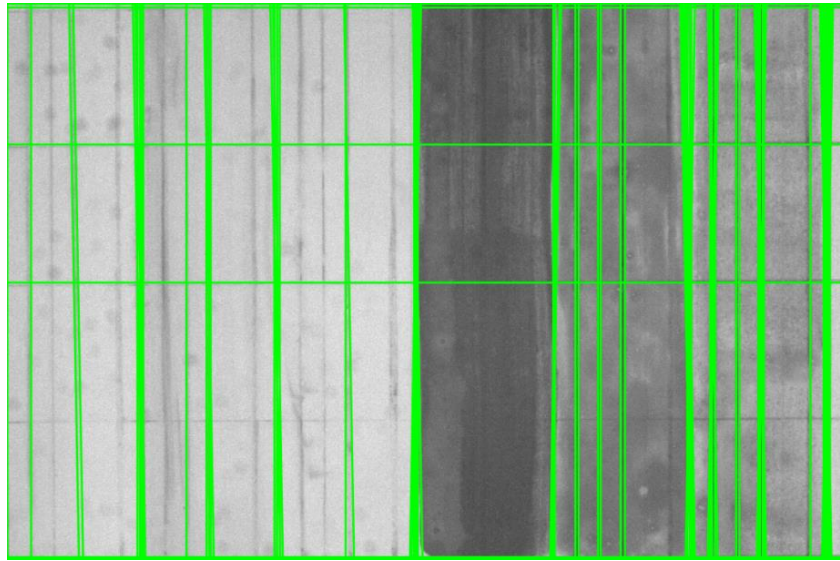
### 5.2.3 Δοκιμή 3<sup>η</sup>

Εφόσον οι προηγούμενες προσπάθειες δεν έδωσαν τα επιθυμητά αποτελέσματα, στην τρίτη δοκιμή η ανίχνευση ακμών έγινε βήμα-βήμα, εφαρμόζοντας τη μέθοδο Sobel αναλυτικά, όπως αναγράφεται στο Παράρτημα I.ΠΙ. Αρχικά, προηγήθηκε εξομάλυνση της εικόνας με ένα γκαουσιανό φίλτρο πριν την εξαγωγή των ακμών, έτσι ώστε να αποφθεχθούν ψευδοακμές που οφείλονται σε θόρυβο. Στη συνέχεια, έγινε υπολογισμός των δύο μασκών Sobel και έπειτα, ακολούθησε η εφαρμογή των δύο μασκών για τον υπολογισμό των μερικών παραγώγων της  $\text{sim}$  ως προς  $x$  και ως προς  $y$ . Μετά, οι δύο παράγωγοι συνδυάστηκαν με την ευκλείδεια μετρική για τον υπολογισμό της κλίσης (gradient). Αφού, έγινε μία κανονικοποίηση των gradients στο  $[0,1]$ , ακολούθησε η εύρεση κατάλληλου κατωφλίου μέσω της συνάρτησης **graythresh()** και της μεθόδου του Otsu. Η εικόνα μετατράπηκε σε δυαδική εικόνα ακμών, όπως φαίνεται στην Εικόνα 19 και ακολούθησε κανονικά η εφαρμογή του μετασχηματισμού Hough.



Εικόνα 19: Edge image

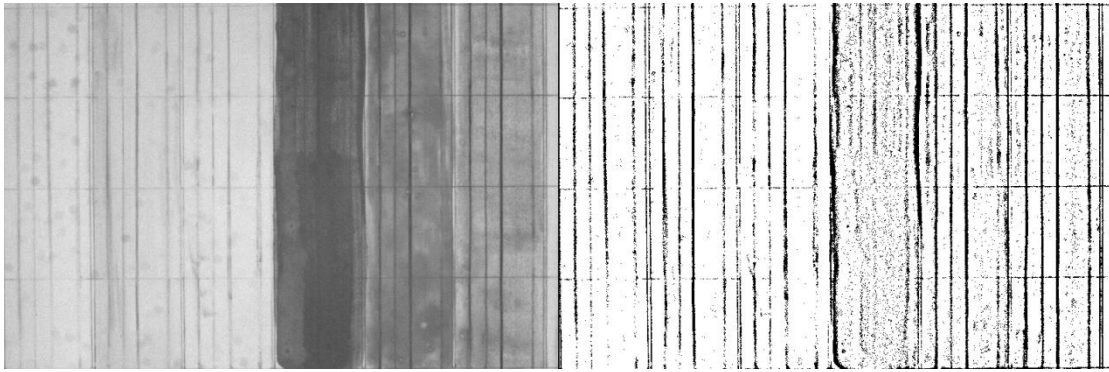
Έγιναν δοκιμές με διαφορετικές τιμές για την μεταβλητή  $c$ , ωστόσο για τις τιμές που αποτυπώνονται στον κώδικα του Παραρτήματος I.ΠΙ, τα αποτελέσματα απεικονίζονται στην Εικόνα 20. Από τα αποτελέσματα γίνεται αντιληπτό πως για προσαρμοζόμενο threshold, οι ακμές και οι γραμμές ανιχνεύονται καλύτερα. Ωστόσο, παρατηρείται δυσκολία στην ανίχνευση ακμών και γραμμών στα σκουρόχρωμα τμήματα της εικόνας.



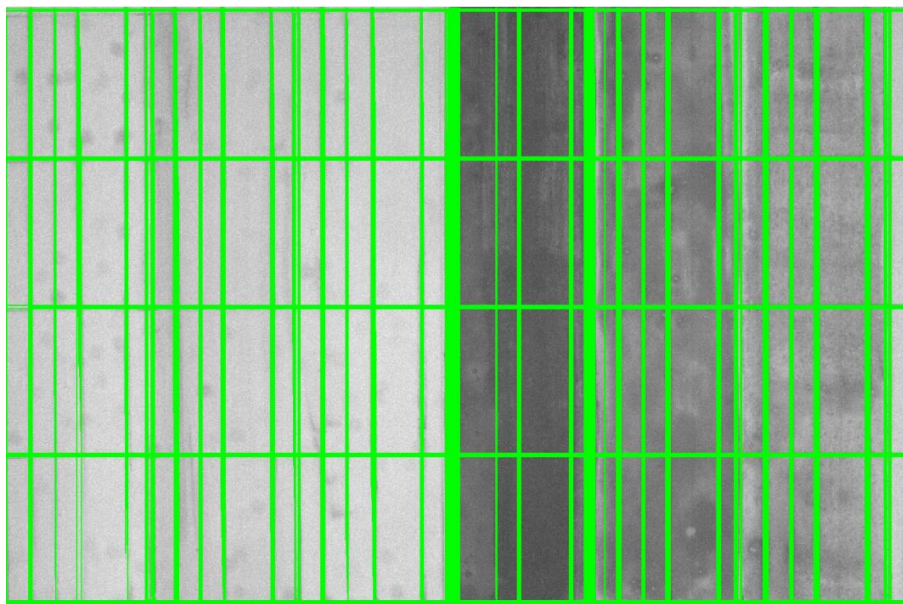
Εικόνα 20: Line detection

#### 5.2.4 Δοκιμή 4<sup>η</sup>

Η τελική τεχνική που εφαρμόστηκε, περιλαμβάνεται στον κώδικα του Παραρτήματος I.IV. Οι διορθώσεις που επικράτησαν αναγράφονται στη συνέχεια. Αρχικά, ο μετασχηματισμός Hough εντοπίζει γραμμές σε δυαδικές εικόνες, στην περίπτωση της εφαρμογής της εικόνας BW, θεωρώντας ότι τα pixels του προσκηνίου είναι άσπρα. Συνεπώς, ως όρισμα στο Hough δόθηκε το αρνητικό της εικόνας BW. Έπειτα, η γκαουσιανή εξομάλυνση απορρίφθηκε, καθώς ίσως επηρέαζε το πάχος των γραμμών. Επιπλέον, δεν χρειάζεται να υπολογιστεί η κλίση της εικόνας, επειδή προέκυπταν διπλές ακμές, είτε υπήρχε μία μετατόπιση της γραμμής, που δημιουργούταν από τις ακμές σε σχέση με τις πραγματικές θέσεις των γραμμών. Ακόμα, κρίθηκε καλύτερη η τεχνική του προσαρμοσμένου threshold, μέσω της συνάρτησης **graythresh()**, για την μετατροπή σε δυαδική εικόνα αποφασίζοντας αν κάποιο pixel είναι μαύρο ή άσπρο από μία γειτονιά γύρω από το τρέχον pixel και έλεγχο για το εάν είναι πάνω ή κάτω από τον μέσο όρο των gray levels της γειτονιάς του. Η Εικόνα 21 απεικονίζει τις ανιχνευμένες ακμές. Μια προσθήκη που εφαρμόστηκε αφορά τον περιορισμό του accumulator array H, έτσι ώστε να έχει μόνο τις οριζόντιες και κατακόρυφες γραμμές. Στον κώδικα οι διευθύνσεις των 0 και 90 μοιρών αντιστοιχούν στις τιμές  $\theta = 1(-90^\circ)$  και  $\theta = 91(0^\circ)$ , με το  $\theta$  ή αλλιώς  $th$ , να είναι η δεύτερη, και όχι η πρώτη, συντεταγμένη στο μετασχηματισμό Hough. Ολοκληρώνοντας, αγνοήθηκαν τα μικρά μήκη κύματος και λήφθηκαν υπόψη τα μεγάλα μήκη κύματος, κοντά στα 1000 nm, καθώς γίνεται καλύτερη διείσδυση στα χρωματικά στρώματα, με αποτέλεσμα να φαίνονται οι από κάτω γραμμές. Τα τελικά αποτελέσματα εμφανίζονται στην Εικόνα 22, οι γραμμές στην εμφανίζονται με πάχος 1 pixel.



Εικόνα 21: Αρχική εικόνα συγκριτικά με την εικόνα με της ανιχνευμένες ακμές



Εικόνα 22: Ανίχνευση Γραμμών

Ολοκληρώνοντας, η παραπάνω τεχνική είναι αυτή που εν τέλει εφαρμόστηκε για την ανίχνευση των γραμμών στα υπερφασματικά δεδομένα. Ωστόσο, με στόχο τα καλύτερα δυνατά αποτελέσματα, τα αρχικά δεδομένα ομαδοποιήθηκαν σε τρεις κατηγορίες. Σε κάθε κατηγορία η συνάρτηση **houghpeaks()** εφαρμόστηκε με διαφορετικές παραμέτρους. Η ομαδοποίηση των εικόνων έγινε βάσει το πόσο εμφανή είναι τα προσχέδια.

### 5.3 Εξαγωγή Patches

Εφόσον ανιχνεύτηκαν οι γραμμές στα υπερφασματικά δεδομένα, επόμενο βήμα αποτέλεσε η εξαγωγή των Patches και η κατηγοριοποίηση τους. Η κατηγοριοποίηση έγινε σε τρεις κατηγορίες, όπως αυτές προαναφέρθηκαν στην εισαγωγή του κεφαλαίου, τμήματα εικόνων όπου το κεντρικό pixel βρίσκεται πάνω σε γραμμή προσχεδίου, τμήματα που περιέχουν γραμμή προσχεδίου και τμήματα όπου δεν περιέχουν καμία γραμμή. Πριν από αυτό όμως και έχοντας εντοπίσει τις γραμμές στις εικόνες, δημιουργήθηκε μία δυαδική εικόνα στην, οποία οι γραμμές έχουν αποτυπωθεί με μαύρο χρώμα, ενώ το υπόβαθρο με



λευκό. Η υλοποίηση έγινε μέσω της συνάρτησης **imcomplement()**. Η δυαδική εικόνα που δημιουργείται, έχει ως σκοπό να βοηθήσει στην εύκολη κατηγοριοποίηση των υπο-τμημάτων της αρχικής εικόνας. Έπειτα, η κάθε δυαδική εικόνα σαρώνεται σε τμήματα 7\*7 pixels, από πάνω προς τα κάτω και από δεξιά προς τα αριστερά. Για κάθε τμήμα που προκύπτει, ακολουθεί ο υπολογισμός της θέσης των κεντρικών εικονοστοιχείων και στη συνέχεια πραγματοποιούνται τρεις έλεγχοι. Στον πρώτο έλεγχο, δίνονται οι συντεταγμένες του κεντρικού εικονοστοιχείου και γίνεται σύγκριση με την τιμή 1, η οποία αναπαριστά το χρώμα μαύρο. Εάν επιστραφεί η λογική τιμή TRUE, τότε υπάρχει γραμμή στο κεντρικό σημείο του τμήματος. Συνεπακόλουθα, γίνεται αντιστοίχιση των συντεταγμένων του τμήματος της δυαδικής εικόνας, με την αρχική εικόνα και αποθηκεύεται το τμήμα της αρχικής εικόνας στην πρώτη κατηγορία, ο κώδικας σταμάτα και η επανάληψη συνεχίζει με το επόμενο τμήμα. Στην περίπτωση που επιστραφεί η λογική τιμή FALSE τότε ακολουθεί ο δεύτερος έλεγχος. Εάν οποιοδήποτε σημείο του τμήματος της δυαδικής εικόνας έχει την τιμή 1, τότε σημαίνει πως υπάρχει σημείο γραμμής στο τμήμα και το αντίστοιχο τμήμα της αρχικής εικόνας μπορεί να αποθηκευθεί στη δεύτερη κατηγορία. Τέλος, εάν και οι δύο προηγούμενοι έλεγχοι έχουν επιστρέψει τη λογική τιμή FALSE, ακολουθεί η τρίτη περίπτωση, όπου όλα τα σημεία του τμήματος της δυαδικής εικόνας έχουν την τιμή 0, οπότε πρόκειται για υπόβαθρο. Το αντίστοιχο τμήμα της αρχικής εικόνας αποθηκεύεται στη τελευταία κατηγορία. Μετά την ολοκλήρωση όλων των επαναλήψεων, έχει δημιουργηθεί το προ απαιτούμενο σύνολο δεδομένων που θα εκπαιδευθεί στο επόμενο κεφάλαιο. Ο κώδικας που υλοποιεί όλα τα παραπάνω, περιλαμβάνεται στο Παράρτημα I.V.

## 6 Πειραματικό Μέρος

Η έκτη ενότητα αφορά την εκπαίδευση του πολυστρωματικού νευρωνικού δικτύου. Εφόσον, ήδη έχει δημιουργηθεί το μοναδικό σύνολο δεδομένων με τις τρεις κατηγορίες, ακολουθεί οργάνωση των δεδομένων ως εξής. Δημιουργείται ο κατάλογος `trndata`, ο οποίος αποτελείται από δύο κατηγορίες, `Positive` και `Negative`. Η κατηγορία `Positive` εμπεριέχει, τα τμήματα εικόνων που στο κεντρικό σημείο τους έχουν γραμμή και τα τμήματα όπου εμπεριέχουν γραμμή αλλά όχι στο κέντρο, δηλαδή τις δύο πρώτες κατηγορίες του συνόλου δεδομένων, οι οποίες στην ουσία είναι αυτές που έχουν γραμμές. Αντίθετα, η κατηγορία `Negative` περιέχει την τρίτη κατηγορία από το αρχικό σύνολο δεδομένων, στην οποία βρίσκονται τα τμήματα που δεν έχουν καμία γραμμή, αρά αποτελούν υπόβαθρο.

Πρώτο στάδιο της εκπαίδευσης αποτελεί η σωστή φόρτωση των δεδομένων που βρίσκονται στο κατάλογο `trndata`. Για τη διαχείριση του συνόλου δεδομένων εικόνων χρησιμοποιήθηκε η συνάρτηση **`ImageDatastore(location, Name, Value)`**, η οποία καθορίζει πρόσθετες παραμέτρους και ιδιότητες, για τα δεδομένα εικόνων που αποθηκεύονται, χρησιμοποιώντας ένα ή περισσότερα ορίσματα ζεύγους ονομάτων-τιμών (ή `name-value`). Οι παράμετροι που δόθηκαν ως ορίσματα στη συνάρτηση είναι το μονοπάτι του καταλόγου `trndata`, το ζευγάρι `name-value IncludeSubfolders=true`, με το οποίο συμπεριλαμβάνονται όλα τα αρχεία και όλοι οι υποφάκελοι (εάν δεν συμπεριληφθεί η τιμή προεπιλογής είναι `false`) και τέλος δόθηκε ως όρισμα το ζευγάρι `LabelSource-foldernames`, με στόχο οι ετικέτες (ή αλλιώς `labels`) να αντιστοιχίζουν σύμφωνα με τα ονόματα των φακέλων και να αποθηκεύονται στην ιδιότητα `Labels`. Κάνοντας χρήση της συγκεκριμένης συνάρτησης, επιτυγχάνεται η φόρτωση όλων των εικόνων σε όλους τους φακέλους και η δημιουργία ετικετών, ίδιων με τα ονόματα των υποφακέλων ( τα `Labels` μπορούν να τροποποιηθούν και να πάρουν όποια μορφή είναι χρησιμότερη). Ο κώδικας `matlab` που υλοποιεί όλα τα παραπάνω, περιλαμβάνεται στο Παράρτημα I.VI. Στη μεταβλητή `numObs` αποθηκεύεται ο συνολικός αριθμός των δεδομένων, ενώ στη μεταβλητή `numObsPerClass` ο αριθμός των κλάσεων.

Επόμενο στάδιο είναι η προετοιμασία των δεδομένων για την εκπαίδευση. Επομένως, πρέπει να αρχικοποιηθούν οι μεταβλητές `train_x` και `train_y`. Όσον αφορά τη μεταβλητή `train_x`, αρχικοποιείται ως ένας πίνακας μηδενικών, μέσω της συνάρτησης **`zeros()`**, με διαστάσεις (`prod(patchSize), numObs`). Αυτό υποδηλώνει ότι, πρόκειται να αποθηκευτούν δεδομένα ισοπεδωμένων εικόνων στο `train_x`, όπου το **`prod(patchSize)`** υπολογίζει το γινόμενο των στοιχείων του διανύσματος `patchSize`. Σε αυτή την περίπτωση, το `patchSize` ορίζεται ως `[7, 7]`, οπότε το **`prod(patchSize)`** ισούται με 49 και το `numObs` είναι ο συνολικός αριθμός των παρατηρήσεων (εικόνων), ο οποίος αντιπροσωπεύει τον αριθμό των στηλών του πίνακα `train_x`. Το τελικό αποτέλεσμα είναι ένας πίνακας 49 γραμμών και 26640 στηλών.

Έπειτα, μέσα σε έναν βρόγχο επανάληψης για το συνολικό αριθμό παρατηρήσεων, γίνεται ανάγνωση της κάθε εικόνας από το αρχείο δεδομένων (trainImds) και φορτώνεται στη μεταβλητή `img`. Αφού η εικόνα φορτωθεί, ισοπεδώνεται μέσω της εντολής `img(:)`, η οποία αναδιαμορφώνει τη δισδιάστατη εικόνα σε ένα μονοδιάστατο διάνυσμα στήλης. Στη συνέχεια, αυτή η ισοπεδωμένη εικόνα αποθηκεύεται στον πίνακα `train_x`. Συγκεκριμένα, αποθηκεύεται στην εκάστοτε *i*-οστή στήλη του `train_x`. Μετά την ολοκλήρωση της επανάληψης του βρόγχου, το `train_x` θα περιέχει ένα σύνολο από πεπλατυσμένα patches εικόνας, όπου κάθε στήλη αντιστοιχεί σε μια παρατήρηση (εικόνα) στο σύνολο δεδομένων εκπαίδευσης και κάθε γραμμή αντιπροσωπεύει μια τιμή εικονοστοιχείου από το πεπλατυσμένο patch εικόνας. Συνεπακόλουθα, η αρχικοποίηση της μεταβλητής `train_y` υλοποιείται ως εξής. Στην αρχή, το `train_y` δημιουργείται με τη μετατροπή των ετικετών από το `trainImds.Labels` σε έναν κατηγορικό πίνακα, μέσω της συνάρτησης `categorical()`. Μετά τη μετατροπή των ετικετών σε κατηγορική μορφή, χρησιμοποιείται η συνάρτηση `dummyvar()`, η οποία μετατρέπει τους κατηγορικούς δείκτες, σε δυαδικούς δείκτες. Κάθε μοναδική κατηγορία στο `train_y` θα μετατραπεί σε ένα δυαδικό διάνυσμα όπου κάθε στοιχείο υποδηλώνει ποια εκ των δύο κατηγοριών εμφανίζεται. Τέλος, ακολουθεί αντιμετάθεση του τον πίνακα `train_y_binary`, έτσι ώστε να έχει την επιθυμητή μορφή-μέγεθος. Ο κώδικας matlab που υλοποιεί όλα τα παραπάνω, περιλαμβάνεται στο Παράρτημα I.VII.

Τρίτο στάδιο είναι, η δημιουργία ενός πολυστρωματικού νευρωνικού δικτύου με τροφοδότηση χρησιμοποιώντας την εργαλειοθήκη νευρωνικών δικτύων του MATLAB και συγκεκριμένα τη συνάρτηση `feedforwardnet()`. Το νευρωνικό δίκτυο θα έχει δύο κρυφά στρώματα, `hiddenLayerSize1` και `hiddenLayerSize2`, με 100 και 20 νευρώνες αντίστοιχα. Στη συνέχεια, δημιουργείται ένα νευρωνικό δίκτυο τροφοδότησης (`net`) με δύο κρυφά στρώματα. Η αρχιτεκτονική του δικτύου ορίζεται από τον πίνακα `[hiddenLayerSize1, hiddenLayerSize2]`. Αφού δημιουργηθεί το νευρωνικό δίκτυο (`net`), μπορούν να υλοποιηθούν εργασίες όπως η διαμόρφωση των επιλογών εκπαίδευσης, η εκπαίδευση του δικτύου χρησιμοποιώντας τα προετοιμασμένα δεδομένα εκπαίδευσης (`train_x` και `train_y_binary`) και η αξιολόγηση του εκπαιδευμένου μοντέλου στα δεδομένα δοκιμής. Ο κώδικας matlab που υλοποιεί τα παραπάνω, περιλαμβάνεται στο Παράρτημα I.VIII.

Αφού, δημιουργηθεί το `net`, μπορούν να ρυθμιστούν-τροποποιηθούν ορισμένοι υπερμαραμέτροι ως εξής:

- Αρχικά, με την εντολή `net.layers{end}.transferFcn = 'softmax'`, καθορίζεται η συνάρτηση ενεργοποίησης για το επίπεδο εξόδου. Στην ταξινόμηση πολλαπλών κλάσεων, η συνάρτηση ενεργοποίησης `softmax` χρησιμοποιείται συνήθως για την παραγωγή πιθανοτήτων κλάσεων. Πιο συγκεκριμένα, η συνάρτηση ενεργοποίησης `'softmax'` εφαρμόζεται στις εξόδους των νευρώνων στο τελευταίο στρώμα (εξόδου) του νευρωνικού δικτύου. Επιπλέον, κανονικοποιεί τις τιμές εξόδου, εξασφαλίζοντας ότι το

άθροισμά τους είναι 1. Αυτή η ιδιότητα κανονικοποίησης σας επιτρέπει να ερμηνεύσετε τις τιμές εξόδου ως πιθανότητες. Κάθε νευρώνας εξόδου αντιπροσωπεύει την πιθανότητα της αντίστοιχης κλάσης. Η έξοδος του στρώματος μπορεί να ερμηνευτεί ως πιθανότητες κλάσης.

- Έπειτα, με την εντολή **net.trainFcn = 'traingdx'**, ορίστηκε η συνάρτηση εκπαίδευσης σε 'traingdx'. Το 'traingdx' αντιστοιχεί στο Gradient Descent with Momentum and Adaptive Learning Rate. Η συνάρτηση traingdx συνδυάζει προσαρμοστικό ρυθμό μάθησης με εκπαίδευση με momentum. Ακόμα, μπορεί να εκπαιδεύσει οποιοδήποτε δίκτυο εφόσον οι συναρτήσεις βάρους, καθαρής εισόδου και μεταφοράς του έχουν συναρτήσεις παραγώγου. Κάθε μεταβλητή προσαρμόζεται σύμφωνα με την κάθοδο κλίσης με ορμή,  $dx = mc * dXprev + lr * mc * dperf/dX$ , όπου  $dXprev$  είναι η προηγούμενη μεταβολή του βάρους ή του bias. Για κάθε εποχή, εάν η απόδοση μειώνεται προς τον στόχο, τότε ο ρυθμός μάθησης αυξάνεται κατά τον παράγοντα *lr\_inc*. Εάν η απόδοση αυξάνεται περισσότερο από τον παράγοντα *max\_perf\_inc*, ο ρυθμός μάθησης προσαρμόζεται κατά τον παράγοντα *lr\_dec* και η αλλαγή που αύξησε την απόδοση δεν πραγματοποιείται. Η εκπαίδευση σταματά όταν εμφανιστεί οποιαδήποτε από αυτές τις συνθήκες, έχει επιτευχθεί ο μέγιστος αριθμός εποχών (επαναλήψεων), υπερβαίνεται το μέγιστο χρονικό διάστημα, η απόδοση ελαχιστοποιείται στο στόχο, η κλίση της απόδοσης πέφτει κάτω από το *min\_grad* και εάν η απόδοση επικύρωσης (σφάλμα επικύρωσης) έχει αυξηθεί περισσότερο από *max\_fail* φορές από την τελευταία φορά που μειώθηκε (όταν χρησιμοποιείται επικύρωση) [40].
- Με την εντολή **net.performFcn = 'crossentropy'**, καθορίζεται η συνάρτηση απόδοσης να είναι 'crossentropy', η οποία είναι η κατηγορική απώλεια cross-entropy.
- Με την εντολή **net.trainParam.epochs = 10**, ορίζεται ο αριθμός των εποχών εκπαίδευσης σε 10.
- Οι εντολές **net.divideParam.trainRatio = 0.7**, **net.divideParam.valRatio = 0.15** και **net.divideParam.testRatio = 0.15**, καθορίζουν την αναλογία των δεδομένων που χρησιμοποιούνται για την εκπαίδευση (70%), για την επικύρωση (15%) και για τεστ (15%). Αυτός ο διαχωρισμός είναι χρήσιμος για την παρακολούθηση της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης και την αποφυγή της υπερπροσαρμογής.
- Οι εντολές **net.trainParam.lr = 0.3**, **net.trainParam.mc = 1.0** και **net.trainParam.goal = 0.95**, είναι οι παράμετροι που ελέγχουν το ρυθμό μάθησης (lr), το momentum (mc) και ορίζουν το στόχο εκπαίδευσης. Στο ρυθμό μάθησης (ή αλλιώς learning rate) η default τιμή είναι 0.01, οι τιμές που ορίστηκαν είναι μεγαλύτερες της τιμής προεπιλογής. Το ίδιο συνέβη και στη τιμή του momentum, η τιμή προεπιλογής είναι 0.9 και τροποποιήθηκε σε μεγαλύτερες τιμές. Όσον αφορά την τιμή του στόχου εκπαίδευσης, εξαρτάται από το

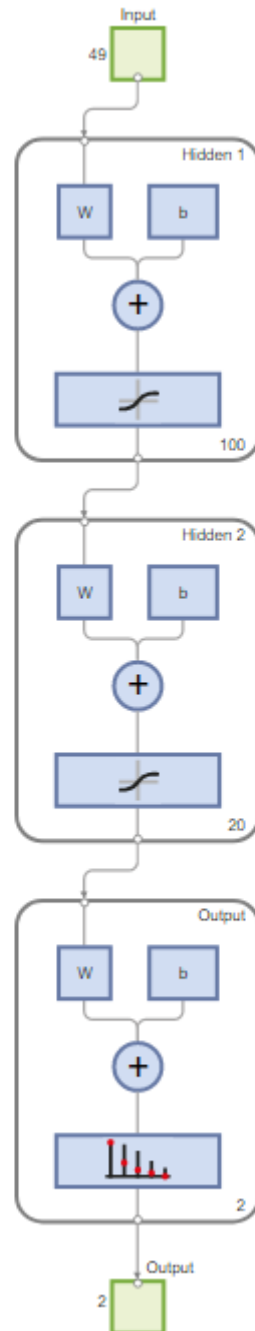
είδος του προβλήματος και τη μετρική απόδοσης έχει νόημα να εφαρμοστεί. Στο συγκεκριμένο παράδειγμα ταξινόμησης εικόνων, τέθηκε ο στόκος επίτευξης ακρίβειας ταξινόμησης στο 95 της εκατό.

- Οι εντολές `net.trainParam.showCommandLine=true` και `net.trainParam.showWindow = true`, είναι ρυθμίσεις που ελέγχουν αν θα εμφανίζεται η πρόοδος της εκπαίδευσης στη γραμμή εντολών και σε ένα παράθυρο προόδου της εκπαίδευσης.

Οι τροποποιήσεις των παραμέτρων βρίσκονται στον κώδικα που περιλαμβάνεται στο Παράρτημα Ι.ΙΧ.

Τελικό βήμα είναι η εκπαίδευση, η οποία επιτυγχάνεται μέσω της συνάρτησης `train()`, η οποία είναι ειδική για τεχνητά νευρωνικά δίκτυα. Για την εκπαίδευση χρειάζονται το `net` (το νευρωνικό δίκτυο που διαμορφώθηκε προηγουμένως), το `train_x` και το `train_y_binary`. Μετά την εκπαίδευση, η συνάρτηση επιστρέφει δύο σημαντικές εξόδους, την ενημερωμένη μεταβλητή `net` με τα μαθημένα βάρη και τις παραμέτρους κατά την εκπαίδευση τη μεταβλητή `tr`, που περιέχει πληροφορίες σχετικά με τη διαδικασία εκπαίδευσης, συμπεριλαμβανομένων των μετρικών απόδοσης εκπαίδευσης, του σφάλματος εκπαίδευσης και του σφάλματος επικύρωσης σε εποχές. Στην Εικόνα 23 αναπαριστάται το διάγραμμα του δικτύου που δημιουργείται.

Ολοκληρώνοντας, ακολουθεί αξιολόγηση του μοντέλου σε ένα σύνολο δοκιμής, όπως υλοποιείται στον κώδικα του Παραρτήματος Ι.ΙΧ. Το `tr.testInd` περιέχει τους δείκτες των δειγμάτων που προορίζονται για δοκιμή από τη ρύθμιση `divideParam.testRatio` στο νευρωνικό δίκτυο. Στόχος είναι ο υπολογισμός της ακρίβειας του νευρωνικού δικτύου στα δεδομένα δοκιμής. Ουσιαστικά, αυτός ο κώδικας αξιολογεί πόσο καλά αποδίδει το εκπαιδευμένο νευρωνικό σας δίκτυο σε ένα ξεχωριστό σύνολο δεδομένων δοκιμής και αναφέρει την ακρίβεια των προβλέψεών του. Όσο υψηλότερη είναι η ακρίβεια, τόσο καλύτερη είναι η απόδοση του δικτύου σας στα δεδομένα δοκιμής. Στις δοκιμές που έγιναν τα αποτελέσματα ήταν αρκετά υψηλά, της τάξεως 85.59%.



Εικόνα 23: Feed-Forward Neural Network

## 7 Συμπεράσματα

Έναυσμα της παρούσας διπλωματική εργασίας αποτέλεσε τον ενδιαφέρον γύρω από τους τομείς της Επεξεργασίας Εικόνας, της Μηχανικής Μάθησης και της Βαθιάς Μάθησης. Σε αυτή τη μελέτη, εξερευνήθηκαν οι προηγούμενοι τομείς σε συνδυασμό με τη τέχνη, με σκοπό την απόκτησή του κατάλληλου θεωρητικού υπόβαθρου. Κατά τη διάρκεια του σταδίου της έρευνας, μελετήθηκαν σπουδαίες εφαρμογές πάνω στα έργα τέχνης και διαπιστώθηκε πως ο συγκεκριμένος κλάδος μπορεί να επωφεληθεί κατά πολύ από τις τεχνικές Μηχανικής Μάθησης και Βαθιάς Μάθησης.

Εφόσον, δημιουργήθηκε το κατάλληλο θεωρητικό υπόβαθρο, σειρά είχε η υλοποίηση της εφαρμογής. Πρώτο και καθοριστικό στάδιο, αποτέλεσε η ανίχνευση των γραμμών, στα υπερφασματικά δεδομένα που προσφέρθηκαν, για τη δημιουργία ενός μοναδικού συνόλου δεδομένων. Για την ανίχνευση των γραμμών εφαρμόστηκε ο μετασχηματισμός Hough. Ωστόσο, για να προκύψουν όσο το δυνατόν καλύτερα αποτελέσματα έγιναν πολλές δοκιμές. Εν τέλει, δύο ενέργειες ωφέλησαν αρκετά. Η πρώτη αφορά την εφαρμογή του προσαρμοσμένου threshold και η δεύτερη αφορά την διαφορετική εφαρμογή παραμέτρων για την εύρεση των ακμών, του μετασχηματισμού Hough, στις διαφορετικές ομάδες εικόνων που δημιουργήθηκαν. Παρά όλες τις προσπάθειες η ανίχνευση ακμών δεν ήταν η βέλτιστη, για όλες τις εικόνες. Το να δουλεύεις με υπερφασματικά δεδομένα αποτελεί μία πρόκληση, για αυτό και προηγήθηκαν τόσες δοκιμές. Ως, επιπρόσθετο βήμα θα μπορούσε να εμπλουτιστεί το σύνολο δεδομένων, με τμήματα εικόνων που θα είχαν περιστραφεί σε τυχαίο βαθμό.

Το τελικό βήμα της εφαρμογής ήταν η εκπαίδευση του πολυστρωματικού νευρωνικού δικτύου. Από τη στιγμή που χρησιμοποιήθηκαν δύο κρυφά νευρωνικά στρώματα με 100 και 20 νευρώνες το καθένα, πρόκειται για εκπαίδευση νευρωνικού δικτύου βαθιάς μάθησης. Έγιναν διάφορες τροποποιήσεις στις υπερπαραμέτρους και μελετήθηκε η ακρίβεια του μοντέλου πάνω στην εκπαίδευση και στο τεστ. Τα ποσοστά ακρίβειας ήταν καλά, ωστόσο θα μπορούσαν να έχουν επιλεγθεί διαφορετικές μέθοδοι, με στόχο να αναδειχθούν οι δυνατότητες διαφορετικών μοντέλων απέναντι σε αυτά τα δεδομένα.

Συμπερασματικά, η θεματολογία της παρούσας διπλωματικής είχε μεγάλο ενδιαφέρον, ήταν επιμορφωτική και δημιουργική. Η υλοποίηση της, όμως, θα μπορούσε να χαρακτηριστεί ως μία πρόκληση. Ολοκληρώνοντας, το έργο που πραγματοποιήθηκε έχει ως στόχο να συμβάλλει στον τομέα της ανάλυσης έργων τέχνης, αλλά επίσης να αναδείξει τις τεράστιες δυνατότητες των διεπιστημονικών προσεγγίσεων. Καθώς λοιπόν, προχωράμε προς το μέλλον, η συγχώνευση της τέχνης και της τεχνολογίας θα συνεχίσει να αποκαλύπτει τους κρυμμένους θησαυρούς μέσα στον καμβά, εμπλουτίζοντας την κατανόηση της παγκόσμιας καλλιτεχνικής κληρονομιάς.

## Βιβλιογραφία – Αναφορικές – Διαδικτυακές Πηγές

- [1] G. Castellano and G. Vessio, “Deep learning approaches to pattern extraction and recognition in paintings and drawings: an overview,” *Neural Computing and Applications*, vol. 33, no. 19, pp. 12263–12282, Apr. 2021, doi: <https://doi.org/10.1007/s00521-021-05893-z>.
- [2] A. Lecoutre, B. Negrevergne, and F. Yger, “Recognizing Art Style Automatically in Painting with Deep Learning,” *proceedings.mlr.press*, Nov. 11, 2017.
- [3] Y. Zhang, *Image Processing*. Walter de Gruyter GmbH & Co KG, 2017.
- [4] R. C. Gonzalez and R. E. Woods, *Ψηφιακή Επεξεργασία Εικόνας*, 4th ed. Εκδόσεις ΤΖΙΟΛΑ, 2018.
- [5] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Computational Intelligence and Neuroscience*, Feb. 01, 2018. <https://www.hindawi.com/journals/cin/2018/7068349/>
- [6] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Nature, 2022.
- [7] B. Mahesh, “Machine Learning Algorithms - A Review,” *International Journal of Science and Research*, no. 2319–7064. pp. 381–386, Jan. 2020. doi: <https://doi.org/10.21275/ART20203995>.
- [8] Z.-H. Zhou, *Machine Learning*. Springer Nature, 2021.
- [9] IBM, “What is Machine Learning?,” *www.ibm.com*, 2023. <https://www.ibm.com/topics/machine-learning>
- [10] V. Advani, “What is Machine Learning? How Machine Learning Works and future of it?,” *GreatLearning*, Apr. 29, 2020. <https://www.mygreatlearning.com/blog/what-is-machine-learning/>
- [11] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, “E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics,” *IEEE Access*, vol. 6, pp. 39117–39138, Jul. 2018, doi: <https://doi.org/10.1109/access.2018.2851790>.
- [12] IBM, “What are Neural Networks? | IBM,” *www.ibm.com*, 2021. <https://www.ibm.com/topics/neural-networks>



- [13] A. Blessing and K. Wen, “Using Machine Learning for Identification of Art Paintings,” 2010.
- [14] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, “Introduction to Machine Learning, Neural Networks, and Deep Learning,” *Translational Vision Science & Technology*, vol. 9, no. 2, pp. 14–14, Jan. 2020, doi: <https://doi.org/10.1167/tvst.9.2.14>.
- [15] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Machine Learning and Deep Learning Approaches for CyberSecurity: A Review,” *IEEE Access*, vol. 10, pp. 19572–19585, 2022, doi:<https://doi.org/10.1109/access.2022.3151248>.
- [16] akashmomale, “Advantages and Disadvantages of Deep Learning,” *GeeksforGeeks*, Jan. 27, 2023. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-deep-learning/>
- [17] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, “Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks,” *Applied Sciences*, vol. 8, no. 9, p. 1575, Sep. 2018, doi: <https://doi.org/10.3390/app8091575>.
- [18] Y. Li, H. Li, and H. Wang, “Pixel-Wise Crack Detection Using Deep Local Pattern Predictor for Robot Application,” *Sensors*, vol. 18, no. 9, p. 3042, Sep. 2018, doi: <https://doi.org/10.3390/s18093042>.
- [19] S. Haykin, *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, 3η Έκδοση. Εκδόσεις Παπασωτηρίου, 2010.
- [20] P. P. Shinde and S. Shah, “A Review of Machine Learning and Deep Learning Applications,” *IEEE Xplore*, Aug. 01, 2018. <https://ieeexplore.ieee.org/document/8697857>
- [21] R. Sizyakin *et al.*, “Crack Detection in Paintings Using Convolutional Neural Networks,” *IEEE Access*, vol. 8, no. 2169–3536, pp. 74535–74552, 2020, doi: <https://doi.org/10.1109/access.2020.2988856>.
- [22] “Why MATLAB - MathWorks,” *uk.mathworks.com*. <https://uk.mathworks.com/products/matlab/why-matlab.html>
- [23] B. Adibhatla, “Understanding Why MATLAB Is Best Suited For Deep Learning,” *Analytics India Magazine*, Feb. 15, 2019.

<https://analyticsindiamag.com/understanding-why-matlab-is-best-suited-for-deep-learning/>

- [24] K. Zhao, Q. Han, C.-B. Zhang, J. Xu, and M.-M. Cheng, “Deep Hough Transform for Semantic Line Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021, doi: <https://doi.org/10.1109/TPAMI.2021.3077129>.
- [25] D. Duan, M. Xie, Q. Mo, Z. Han, and Y. Wan, “An improved Hough transform for line detection,” *IEEE Xplore*, Oct. 01, 2010. <https://ieeexplore.ieee.org/abstract/document/5620827>
- [26] “Hough transform - MATLAB hough - MathWorks United Kingdom,” *uk.mathworks.com*. <https://uk.mathworks.com/help/images/ref/hough.html>
- [27] A. Martin and S. Tosunoglu, “IMAGE PROCESSING TECHNIQUES FOR MACHINE VISION,” 2000.
- [28] W. Rong, Z. Li, W. Zhang, and L. Sun, “An improved Canny edge detection algorithm,” *IEEE Xplore*, Aug. 01, 2014. <https://ieeexplore.ieee.org/document/6885761>
- [29] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, “Feature Detectors - Sobel Edge Detector,” *homepages.inf.ed.ac.uk*, 2003. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>
- [30] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu, “An improved Sobel edge detection,” *IEEE Xplore*, Jul. 01, 2010. <https://ieeexplore.ieee.org/abstract/document/5563693/>
- [31] A. Abdulrahman and S. Varol, “A Review of Image Segmentation Using MATLAB Environment,” *ieeexplore.ieee.org*, Jun. 15, 2020. <https://ieeexplore.ieee.org/abstract/document/9116191>
- [32] Wikipedia Contributors, “MATLAB,” *Wikipedia*, Mar. 21, 2019. <https://en.wikipedia.org/wiki/MATLAB>
- [33] S. Razavi and B. A. Tolson, “A New Formulation for Feedforward Neural Networks,” *ieeexplore.ieee.org*, Oct. 2011. <https://ieeexplore.ieee.org/document/5993545> (accessed Oct. 07, 2023).
- [34] D. Svozil, V. Kvasnicka, and PospichalJ., “Introduction to multi-layer feed-forward neural networks,” *Chemometrics and Intelligent Laboratory Systems*, vol.

- 39, no. 1, pp. 43–62, Nov. 1997, doi: [https://doi.org/10.1016/s0169-7439\(97\)00061-0](https://doi.org/10.1016/s0169-7439(97)00061-0).
- [35] J. Singh and R. Banerjee, “A Study on Single and Multi-layer Perceptron Neural Network,” *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, doi: <https://doi.org/10.1109/ICCMC.2019.8819775>.
- [36] M.-C. POPESCU, V. E. BALAS, L. PERESCU-POPESCU, and N. MASTORAKIS, “Multilayer Perceptron and Neural Networks ,” no. 1109–2734. Jul. 2009..
- [37] C.-H. Chang, “Deep and Shallow Architecture of Multilayer Neural Networks,” *ieeexplore.ieee.org*, Oct. 2015.  
<https://ieeexplore.ieee.org/abstract/document/7010967>
- [38] R. P. Brent, “Fast Training Algorithms for Multi-layer Neural Nets,” *Transactions on Neural Networks 2* , pp. 346–354, 1991.
- [39] X. Zhong and D. Enke, “Predicting the daily return direction of the stock market using hybrid machine learning algorithms,” *Financial Innovation*, vol. 5, no. 1, Jun. 2019, doi: <https://doi.org/10.1186/s40854-019-0138-0>.
- [40] “Gradient descent with momentum and adaptive learning rate backpropagation - MATLAB traingdx - MathWorks United Kingdom,” *uk.mathworks.com*.  
<https://uk.mathworks.com/help/deeplearning/ref/traingdx.html>

## Παράρτημα Ι

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

### I.I

```
% extract edges with canny
edges = edge(img, 'canny');
figure(3)
imshow(edges);
% hough line
[H,T,R] = hough(edges);
%P = houghpeaks(H,30,'threshold',1);
P = houghpeaks(H,30,'threshold',200);
lines = houghlines(edges,T,R,P,'FillGap',2000,'MinLength',1);
figure(6), imshow(img), hold on

for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

    % Compute the bounding box for the line segment
    x_min = min(xy(:,1));
    x_max = max(xy(:,1));
    y_min = min(xy(:,2));
    y_max = max(xy(:,2));

end
```

### I.II

```
edges = edge(img, 'sobel', 0.04, 'both');
figure(3)
imshow(edges);
% hough line
[H,T,R] = hough(edges);
P = houghpeaks(H, 40, 'NHoodSize', [1,1],'threshold',100);
lines = houghlines(edges,T,R,P,'FillGap',1600,'MinLength',1);
figure(6), imshow(img), hold on

for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

    x_min = min(xy(:,1));
    x_max = max(xy(:,1));
    y_min = min(xy(:,2));
    y_max = max(xy(:,2));

end
```

### I.III

```
% remove noise
h = fspecial('gaussian',[5 5],1);
sim = conv2(double(image2),h,'same');
h = fspecial('sobel'); %horizontal Sobel mask
v = -h';
Iy = imfilter(sim, h, 'replicate');
Ix = imfilter(sim, v, 'replicate');
gradim = sqrt(Ix.^2 + Iy.^2); %Euclidean metric
%gradim = abs(Ix) + abs(Iy); %Manhattan metric
mxgrad = max(gradim(:));
mngrad = min(gradim(:));
gradim = (gradim-mngrad)/(mxgrad-mngrad);
%imtool(gradim,[])
th = graythresh(gradim);
c = 0.4;
edges = gradim > c*th;
figure(3)
imshow(edges);
% hough line
[H,T,R] = hough(edges);
P = houghpeaks(H, 80, 'NHoodSize', [5,5],'threshold',100);
lines = houghlines(edges,T,R,P,'FillGap',2000,'MinLength',2);
figure(4), imshow(image2), hold on
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

    x_min = min(xy(:,1));
    x_max = max(xy(:,1));
    y_min = min(xy(:,2));
    y_max = max(xy(:,2));
end
```

### I.IV

```
% convert to grayscale
gim = im2gray(im);
%imtool(gim);
figure(1)
imshow(gim);
NH = 2*floor(size(gim)/100)+1;
T = adaptthresh(gim,0.67,'NeighborhoodSize',NH);
BW = imbinarize(gim,T);
figure(2), imshowpair(gim,BW,'montage')
figure(3)
imshow(BW);
% Hough transform
[H,th,rho] = hough(~BW);
H1 = zeros(size(H));
H1(:,[1,91]) = H(:,[1,91]);
%P = houghpeaks(H, 50, 'NHoodSize', [9,9],'threshold',100);
%group1
P = houghpeaks(H1, 300, 'NHoodSize', [1,1],'threshold',100);
%P = houghpeaks(H1, 40, 'NHoodSize', [1,1],'threshold',100);
```

```
%group2
%P = houghpeaks(H1, 60, 'NHoodSize', [7,7],'threshold',100);
%group3
%P = houghpeaks(H1, 20, 'NHoodSize', [5,5],'threshold',100);
%group4
%P = houghpeaks(H1, 100, 'NHoodSize', [3,3],'threshold',100);
%group5
%P = houghpeaks(H1, 120, 'NHoodSize', [5,5],'threshold',100);

lines = houghlines(~BW,th,rho,P,'FillGap',2000,'MinLength',1);

% imshow(BW)
figure(4), imshow(gim), hold on
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',1,'Color','green');
end
```

## I.V

```
% Define directory paths for each category
dirA = 'data/ToLine';
dirB = 'data/NearLine';
dirC = 'data/NoLine';
dirD = 'trndata/positive';
dirE = 'trndata/negative';
% Create the directories if they don't exist
if ~isfolder(dirA)
    mkdir(dirA);
end
if ~isfolder(dirB)
    mkdir(dirB);
end
if ~isfolder(dirC)
    mkdir(dirC);
end
if ~isfolder(dirD)
    mkdir(dirD);
end
if ~isfolder(dirE)
    mkdir(dirE);
end
% counters
a = 0 ;
b = 0 ;
c = 0 ;

filelist = dir(fullfile(dirname, '*.bmp'));

% Loop over all the bmp files in the directory
for l = 1:length(filelist)

    % Read the image
    filename = fullfile(dirname, filelist(l).name);
    im = imread(filename);
```

```
[rows, cols, ~] = size(im);

% convert to grayscale
gim = im2gray(im);

NH = 2*floor(size(gim)/100)+1;

T = adaptthresh(gim,0.67,'NeighborhoodSize',NH);
BW = imbinarize(gim,T);

% Hough transform
[H,th,rho] = hough(~BW);
H1 = zeros(size(H));
H1(:,[1,91]) = H(:,[1,91]);

P = houghpeaks(H1, 300, 'NHoodSize', [1,1],'threshold',100);

lines = houghlines(~BW,th,rho,P,'FillGap',2000,'MinLength',1);

% Create an empty white image (all zeros)
imageSize = [rows, cols];
binaryImage = zeros(imageSize);

for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    binaryImage = insertShape(binaryImage, 'line', xy, 'Color', [1 1
1], 'LineWidth', 1);
end

% Invert the binary image to make lines black and background white
binaryImage = imcomplement(binaryImage);

block_size = 7; % Size of the sliding window

for col1 = 1 : 7 : 1596
    col2 = col1 + block_size - 1;
    if col2 > 1596
        col2 = 1596;
    end
    for row1 = 1 : 7 : 1057
        row2 = row1 + block_size - 1;
        if row2 > 1057
            row2 = 1057;
        end

        % Get the subimage.
        subImage = im(row1 : row2, col1 : col2);

        block = binaryImage(row1 : row2, col1 : col2);

        centroidRow = floor(block_size/2)+1;
        centroidCol = floor(block_size/2)+1;

        if block(centroidRow, centroidCol) == 1
            %disp('The point belongs to a line segment in the image.');
```

```
            imwrite(subImage, fullfile(dirA,
sprintf('line_patch_1_GR1_%d.png', a)));
            imwrite(subImage, fullfile(dirD,
sprintf('line_patch_1_GR1_%d.png', a)));
```

```
        a = a + 1;
        break;
    elseif any(block(:) == 1)
        %disp('The block contains a line.');
```

```
        imwrite(subImage, fullfile(dirB,
sprintf('line_patch_2_GR1_%d.png', b)));
        imwrite(subImage, fullfile(dirD,
sprintf('line_patch_2_GR1_%d.png', b)));
        b = b + 1;
    elseif all(block(:) == 0)
        %disp('The block does not contain a line.');
```

```
        imwrite(subImage, fullfile(dirC,
sprintf('no_line_patch_3_GR1_%d.png', c)));
        imwrite(subImage, fullfile(dirE,
sprintf('no_line_patch_3_GR1_%d.png', c)));
        c = c + 1;
    end
end
end
end
```

## I.VI

```
dataFolder = '../Final/trndata';
trainImds = imageDatastore(dataFolder, ...
    'IncludeSubfolders',true, ... %Include images from subfolders
    'LabelSource','foldernames'); %Use subfolder names as labels
numObs = length(trainImds.Labels);
numObsPerClass = countEachLabel(trainImds);
%numClasses = numel(categories(trainImds.Labels));
disp(numObsPerClass);
```

## I.VII

```
patchSize = [7, 7]; % patch size

train_x = zeros(prod(patchSize), numObs);
train_y = categorical(trainImds.Labels); % Convert labels to categorical
% Convert categorical labels to binary indicators
train_y_binary = dummyvar(train_y);
train_y_binary = train_y_binary'; %Transpose train_y_binary

for i = 1:numObs
    img = readimage(trainImds, i);
    train_x(:, i) = img(:); % Store the flattened image
end
```

## I.VIII

```
% Example using feedforwardnet
hiddenLayerSize1 = 100;
hiddenLayerSize2 = 20;

%returns a feedforward neural network with two hidden layers
net = feedforwardnet([hiddenLayerSize1, hiddenLayerSize2]);
```



## I.IX

```
numClasses = 2;

net.layers{end}.size = numClasses;
net.layers{end}.transferFcn = 'softmax'; % Softmax activation for multi-
class

% Scaled Conjugate Gradient
%net.trainFcn = 'trainscg';
% Gradient Descent with Momentum and Adaptive Learning Rate
net.trainFcn = 'traingdx';
net.performFcn = 'crossentropy'

net.trainParam.epochs = 10;
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 0.7;
net.divideParam.valRatio = 0.15;
net.divideParam.testRatio = 0.15;

net.trainParam.lr = 0.3;
net.trainParam.mc = 1.0;
%net.trainParam.goal = 0.95;

net.trainParam.showCommandLine = true;
net.trainParam.showWindow = true;
```

## I.X

```
test_x = train_x(:, tr.testInd);
test_y = train_y_binary(:, tr.testInd);

% Use the trained neural network for testing
Y_pred = net(test_x);
Y_pred_labels = vec2ind(Y_pred);
Y_test_labels = vec2ind(test_y);

% Calculate accuracy
accuracy = sum(Y_pred_labels == Y_test_labels) / numel(Y_test_labels);
fprintf('Accuracy on test set: %.2f%%\n', accuracy * 100);
```