

**Ανάπτυξη προηγμένης μηχανής
δοκιμών κόπωσης σε καταπόνηση
κάμψης 3 σημείων
(Development of an advanced fatigue
testing machine for 3 point bending)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΑΒΑΡΗΣ ΝΙΚΟΛΑΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΒΟΣΝΙΑΚΟΣ ΓΕΩΡΓΙΟΣ-ΧΡΙΣΤΟΦΟΡΟΣ

Ευχαριστίες

Ολοκληρώνοντας αυτή την εργασία θέλω να ευχαριστήσω τους ανθρώπους που με βοήθησαν να την φέρω εις πέρας. Αρχικά τον καθηγητή μου κ. Βοσνιάκο που με καθοδήγησε όλους αυτούς τους μήνες. Τον κ. Κερασιώτη ο οποίος ήταν δίπλα μου σε όποιο πρόβλημα προέκυπτε και έτοιμος να με βοηθήσει. Τους κ. Τζίμα και κ. Νιδριώτη για την πολύτιμη βοήθεια τους στα ηλεκτρονικά και τον κ. Σταθάτο για τις πολύτιμες συμβουλές του πάνω στα νευρωνικά δίκτυα. Τέλος θέλω να ευχαριστήσω την οικογένεια μου, την Ευγενία και τους φίλους μου που ήταν δίπλα μου και με στήριξαν όλη αυτή τη περίοδο.

Περιεχόμενα

1.	Περίληψη.....	4
2.	Εισαγωγή.....	5
3.	Βιβλιογραφική ανασκόπηση.....	7
3.1.	Κατηγοριοποίηση μηχανών κόπωσης.....	7
3.2.	Παρόμοιες εργασίες.....	8
4.	Τροποποίηση εργαστηριακής μηχανής.....	14
4.1	Λειτουργία υπάρχουσας μηχανής.....	14
4.2	Σχεδιασμός και έλεγχος μετρητικών οργάνων.....	14
4.2.1	Ζυγαριά.....	14
4.2.2	Έλεγχος.....	19
4.2.3	Μετρητής κύκλων.....	19
4.2.4	Κάμερα.....	21
4.2.5	Κώδικας.....	21
5.	Νευρωνικό Δίκτυο.....	23
5.1	Dataset (σύνολο δεδομένων).....	23
5.1.1	Καθορισμός αρχικού δείγματος.....	23
5.1.2	Μέθοδος επαύξησης συνόλου δεδομένων (Data augmentation).....	26
5.1.3	Εφαρμογή.....	27
5.2	Αρχιτεκτονική νευρωνικού δικτύου και εκπαίδευση.....	28
5.2.1	Εισαγωγή.....	28
5.2.2	Τρόπος λειτουργίας νευρωνικών δικτύων.....	28
5.3	Δίκτυα συνέλιξης.....	33
5.4	Κατασκευή δικτύων αναγνώρισης.....	34
5.4.1	Συνελικτικό δίκτυο.....	34
5.4.2	Εναλλακτικά δίκτυα συνέλιξης.....	35
6	Αποτελέσματα.....	36
7	Συμπεράσματα για μελλοντική εργασία.....	44
7.1	Συμπεράσματα.....	44
7.2	Μελλοντική εργασία.....	44
8	Παραρτήματα.....	45
8.1	Κώδικας επαύξησης των δεδομένων.....	45
8.2	Κώδικες νευρωνικών δικτύων.....	46
8.2.1	Κώδικας νευρωνικού δικτύου ενότητας 4.4.1.....	46
8.2.2	Κώδικας νευρωνικού δικτύου μειωμένου βάθους.....	48

8.2.3	Κώδικας νευρωνικού δικτύου μειωμένου βάθους με early stopping	50
8.2.4	Κώδικας νευρωνικού δικτύου ResNet.....	52
8.3	Κώδικας ελέγχου Arduino	54
8.4	Κώδικας Python.....	58
8.5	Βαθμονόμηση ζυγαριάς	59
9	Βιβλιογραφία.....	65

1. Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται τη βελτίωση μίας μηχανής κόπωσης με την προσθήκη συστήματος για την καταγραφή του φορτίου, των κύκλων κόπωσης και την χρήση νευρωνικού δικτύου για τον προσδιορισμό της αστοχίας του δοκιμίου. Αρχικά γίνεται μία βιβλιογραφική ανασκόπηση που περιλαμβάνει εργασίες κατασκευής μηχανών κόπωσης και εργασίες εντοπισμού αστοχιών μέσω μηχανικής όρασης. Κατόπιν παρουσιάζονται τα επιμέρους συστήματα (ζυγαριά, διάταξη ελέγχου, μετρητής κύκλων και κάμερα) και ο τρόπος λειτουργίας τους, ενώ αναλύεται και η αρχή λειτουργίας της μηχανής συνολικά. Ύστερα πραγματοποιείται μία ανάλυση για τη δημιουργία και την επαύξηση του συνόλου δεδομένων. Στη συνέχεια αναλύεται ο τρόπος λειτουργίας και κατασκευής των νευρωνικών δικτύων και των δικτύων συνέλιξης. Τέλος παρουσιάζονται τα αποτελέσματα της εργασίας, τα συμπεράσματα που προέκυψαν και ορισμένες βελτιώσεις για μία μελλοντική εργασία.

2. Εισαγωγή

Σκοπός της παρούσας εργασίας είναι η αναβάθμιση μίας εργαστηριακής μηχανής κόπωσης. Η μηχανή εκτελούσε απλά μία παλινδρομική κίνηση. Στο πλαίσιο της εργασίας της προστέθηκαν δυναμοκυψέλες για την καταγραφή των φορτίων, ένας μετρητής για τους κύκλους κόπωσης και μία κάμερα με ένα νευρωνικό δίκτυο για τον εντοπισμό της ρωγμής στο δοκίμιο. Τελικός στόχος αυτής της κατασκευής είναι η αυτοματοποιημένη εξαγωγή διαγραμμάτων κόπωσης για διάφορα υλικά. Στην παρούσα εργασία πρόκειται για σύνθετα υλικά πολυμερικής μήτρας, αλλά αυτό είναι μόνο ενδεικτικό. Έτσι ο χειριστής θα απελευθερωθεί από τη διαδικασία παρατήρησης του πότε θα καταστραφεί το δοκίμιο.

Η ύπαρξη μηχανών κόπωσης είναι ιδιαίτερα σημαντική για τη βιομηχανία καθώς βελτιώνει την απόδοση και την ασφάλεια των κατασκευών. Η βελτίωση της αξιοπιστίας επιτυγχάνεται μέσω της δοκιμής των υλικών σε πραγματικές συνθήκες. Έτσι οι κατασκευαστές μπορούν να δοκιμάσουν την αντοχή και την ανθεκτικότητά τους. Παράλληλα μεγάλα είναι και τα οικονομικά οφέλη καθώς από την όλη διαδικασία οι κατασκευαστές μπορούν να περιορίσουν έξοδα που σχετίζονται με ανακλήσεις προϊόντων και καταβολή εγγυήσεων. Οι σύγχρονες μηχανές κόπωσης έχουν τη δυνατότητα να προσομοιώσουν ένα μεγάλο εύρος τρόπων φόρτισης και φορτίων. Ταυτόχρονα τα αποτελέσματα τους θα χαρακτηρίζονται από υψηλή ακρίβεια, καθώς οι αισθητήρες μπορούν να ανιληφθούν μέχρι και τη μικρότερη μεταβολή. Τέλος οι σύγχρονες μηχανές κόπωσης χαρακτηρίζονται από ασφάλεια στον τρόπο λειτουργίας τους και έχουν τη δυνατότητα να δώσουν γρήγορα αποτελέσματα. [1]

Βασική προϋπόθεση για την κατασκευή της μηχανής ήταν η έρευνα παρόμοιων μηχανών κόπωσης που κυκλοφορούν στο εμπόριο. Ένα μοντέλο που κυκλοφορεί από το 2007 είναι η ElectroPuls της Instron το οποίο προσφέρει δυνατότητα για πολύπλευρες δοκιμές κόπωσης σε ένα μεγάλο εύρος ταχυτήτων με τη χρήση ενός γραμμικού ηλεκτροκινητήρα. Επίσης ανάμεσα στα τέσσερα μοντέλα που κυκλοφορούν ο αγοραστής μπορεί να επιλέξει αυτό που ταιριάζει καλύτερα στις απαιτήσεις του, με δυνατότητα μέγιστου φορτίου ως 20kN και μέγιστης ροπής ως 130Nm. Το μοντέλο 5/10kN Fatigue Testing Machine της WTS παρουσιάζει δυνατότητες εκτέλεσης θλίψης, εφελκυσμού και κάμψης σε ένα μεγάλο εύρος κύκλων κόπωσης. Παράλληλα δίνει τη δυνατότητα στον χρήστη να επιλέξει μία από τις συνήθεις μορφές φορτίων (ημιτονοειδές, παλμικό κλπ) ή να σχεδιάσει ένα δικό του φορτίο. Το συγκεκριμένο μοντέλο είναι συμβατό με διάφορα περιφερειακά συστήματα, όπως θαλάμους χαμηλής/υψηλής θερμοκρασίας και θαλάμους διάβρωσης. Η μηχανή κυκλοφορεί σε δύο μοντέλα το πρώτο που μπορεί να προσφέρει μέγιστο δυναμικό φορτίο $\pm 10kN$ και το δεύτερο με $\pm 20kN$. Τέλος η ακρίβεια τους στην καταγραφή της δύναμης και της μετατόπισης είναι $\pm 0.5\%$.

Οι απαιτήσεις του χρήστη μιας μηχανής κόπωσης είναι οι ακόλουθες:

- I. Συνεχής καταγραφή του φορτίου.
- II. Αυτόματος προσδιορισμός της έναρξης αστοχίας του δοκιμίου μέσω πτώσης φορτίου και εμφάνισης γεωμετρικής ασυνέχειας (πχ ρωγμής για μεταλλικά υλικά, delamination και αποκολλήσεων στρώσεων για στρωματικά σύνθετα υλικά κλπ).
- III. Αυτόματος προσδιορισμός της ολοκλήρωσης της αστοχίας, ανάλογα με το υλικό.
- IV. Αυτόματη καταγραφή της συμπεριφοράς (πχ παραμόρφωσης) του δοκιμίου σε κάθε κύκλο έως την αστοχία.
- V. Δυνατότητα επιβολής διαφορετικών φορτίων ή βέλους κάμψης.
- VI. Δυνατότητα χρήσης δοκιμίων διαφορετικών διαστάσεων.
- VII. Δυνατότητα αυξομείωσης της ταχύτητας κύκλων φόρτισης.

- VIII. Δυνατότητα επιβολής διαφορετικών τύπων φόρτισης σε κόπωση, πχ εφελκυσμός, θλίψη, κάμψη, στρέψη, ή και συνδυασμός αυτών.
- IX. Δυνατότητα μεταβολής των τιμών φόρτισης στον τυπικό κύκλο.

Η παρούσα εργασία εστίασε στην φόρτιση κάμψης τριών σημείων και μόνο, δεδομένου ότι η κάλυψη και άλλων τύπων είναι εξαιρετικά δύσκολη ακόμη και για μεγάλου κόστους εμπορικά διαθέσιμες μηχανές. Από τις λοιπές απαιτήσεις, η εργασία αντιμετώπισε τις υπ' αριθμό I, III,V,IX. Μελλοντικά θεωρείται ότι θα καλυφθούν περισσότερες απαιτήσεις.

Η εργασία διαρθρώνεται ως εξής. Αρχικά γίνεται μία εισαγωγή των διάφορων ειδών μηχανών κόπωσης που υπάρχουν και των γενικών τους χαρακτηριστικών. Έπειτα παρουσιάζονται και σχολιάζονται ορισμένες εργασίες που αφορούν μηχανές κόπωσης για κάμψη. Κατόπιν γίνεται μια λεπτομερής περιγραφή του εξοπλισμού που χρησιμοποιήθηκε καθώς και της λογικής με βάση την οποία προγραμματίστηκαν τα διάφορα εξαρτήματα. Στη συνέχεια πραγματοποιείται παρουσίαση του νευρωνικού δικτύου με το οποίο γίνεται αναγνώριση αστοχίας, ξεκινώντας με μία εισαγωγή στην επαύξηση δεδομένων και πως αξιοποιήθηκε στην παρούσα εργασία και συνεχίζοντας με βασικές έννοιες των νευρωνικών δικτύων συνέλιξης και ανάλυση της λογικής με βάση την οποία έγινε η εκπαίδευση του συγκεκριμένου νευρωνικού δικτύου. Τέλος παρουσιάζονται και σχολιάζονται αποτελέσματα. Στο τελικό κεφάλαιο δίδονται συμπεράσματα και μελλοντική εργασία βελτίωσης της μηχανής.



Εικόνα 1 Η μηχανή

3. Βιβλιογραφική ανασκόπηση

3.1. Κατηγοριοποίηση μηχανών κόπωσης

Όλες οι μηχανές κόπωσης δεν χρησιμοποιούνται για τον ίδιο σκοπό, κάποιες φορές ο κατασκευαστής ενδιαφέρεται να μελετήσει την κόπωση ενός φτερού αεροσκάφους ενώ κάποιες άλλες την κόπωση ενός περιστρεφόμενου άξονα που χρησιμεύει στη μεταφορά ισχύος. Υπάρχουν μηχανές γενικού σκοπού οι οποίες μπορούν να δώσουν αποτελέσματα για ένα σύνολο εφαρμογών και οι ειδικού σκοπού που κατασκευάζονται για την μελέτη ενός συγκεκριμένου πειράματος. Παράλληλα μία ακόμα διάκριση που μπορεί να γίνει είναι με βάση το μέγεθος του δοκιμίου ή συναρμολογήματος. [2]

Το είδος φόρτισης στο οποίο θα εκτεθεί το δοκίμιο είναι ο βασικός πυλώνας μίας μηχανής κόπωσης. Στο παρόν σημείο θα γίνει μία συνοπτική παρουσίαση των ειδών και των τρόπων δημιουργίας διάφορων φορτίων. Αρχικά η κλασική εναλλαγή θλίψης-κάμψης που εμφανίζεται για αξονική φόρτιση. Κατόπιν η φόρτιση για κάμψη του δοκιμίου, όπως συμβαίνει στο παρόν πείραμα, μπορεί να είναι μονής κατεύθυνσης ή εναλλασσόμενη. Η στρεπτική φόρτιση μπορεί να βρει εφαρμογή στην μελέτη κατασκευών που τείνουν να περιστραφούν λόγω του αέρα. Τέλος μπορεί να υπάρξει διαξονική και πολυαξονική φόρτιση ενός δοκιμίου σε πολυπλοκότερες εφαρμογές.

Ο κλασικότερος τρόπος δημιουργίας φορτίου είναι με τη χρήση ενός ηλεκτροκινητήρα και ενός διωστήρα, έτσι η περιστροφική κίνηση μετατρέπεται σε μονοαξονική και μπορεί να χρησιμοποιηθεί για κάμψη, όπως στην παρούσα εργασία. Είναι σημαντικό να σημειωθεί ότι με αυτό τον τρόπο το πλάτος της ταλάντωσης παραμένει σταθερό. Αντίθετα αν είναι απαραίτητο το φορτίο να μείνει σταθερό τότε αποτελεί συνήθης πρακτική το κρέμασμα μίας ορισμένης μάζας από το ένα ή και τα δύο άκρα του δοκιμίου και η ταυτόχρονη περιστροφή του δοκιμίου με τη χρήση ενός ηλεκτροκινητήρα. Παράλληλα ιδιαίτερα χρήσιμη εφαρμογή για τη φόρτιση ενός δοκιμίου είναι η χρήση φυγόκεντρων δυνάμεων. Η φυγόκεντρο δύναμη παράγεται μέσω ενός περιστρεφόμενου αζυγοσταθμισμένου δίσκου. Η φυγόκεντρος δύναμη μπορεί να κατευθυνθεί με τη χρήση κατάλληλης διάταξης και ο έλεγχος του μεγέθους της με τη χρήση ελατηρίων.

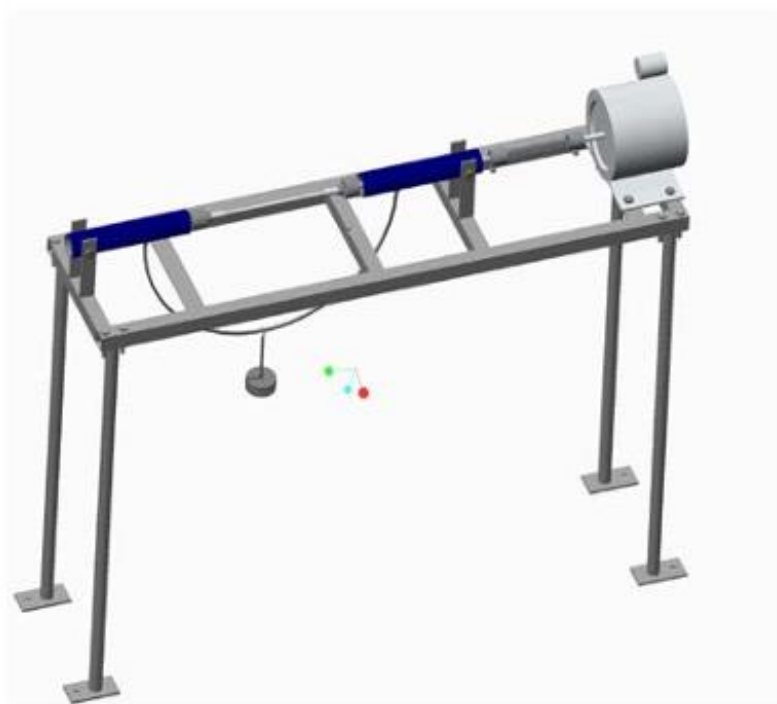
Η δύναμη που παράγεται με υδραυλικά μέσα είναι ιδιαίτερα χρήσιμη καθώς δίνει τη δυνατότητα τον μηχανικό να πειραματιστεί με μεγάλα φορτία, ωστόσο βασικό μειονέκτημα της μεθόδου είναι η χαμηλή συχνότητα στην οποία μπορεί να λειτουργήσει ένα υδραυλικό σύστημα. Η υδραυλική φόρτιση μπορεί με κατάλληλη διάταξη να δώσει τη δυνατότητα για πολυαξονική φόρτιση (αξονική, στρεπτική και εξαιτίας της διαφοράς πίεσης που μπορεί να υπάρχει μεταξύ ενός τοιχώματος). Οι δυνάμεις από πνευματικές διατάξεις έχουν την εξής αρχή λειτουργίας, ένας συμπιεστής τροφοδοτεί με συμπιεσμένο αέρα έναν κύλινδρο μετατοπίζοντας κατά αυτό τον τρόπο το πιστόνι. Ύστερα με έναν δεύτερο συμπιεστή μπορεί να συντελείται η αποσυμπίεση. Τέλος υπάρχουν οι μέθοδοι της θερμικής διαστολής για τις οποίες η συχνότητα των θερμικών κύκλων είναι πολύ μικρή (μπορεί και δύο κύκλους το λεπτό) και η χρήση ηλεκτρομαγνητικών δυνάμεων.

Εν κατακλείδι πριν από την δημιουργία μίας μηχανής κόπωσης πρέπει να έχει προσδιοριστεί ο σκοπός της καθώς και το είδος φόρτισης του δοκιμίου. Έπειτα αξιολογώντας τα τεchnοοικονομικά του μέσα ο μηχανικός καλείται να επιλέξει τον τρόπο φόρτισης του.

3.2. Παρόμοιες εργασίες

Κατόπιν μελέτης της σχετική βιβλιογραφίας βρέθηκαν διάφορες εργασίες κατασκευής μηχανών κόπωσης. Η αρχή λειτουργίας τους βασίζεται στην πρώτη κατηγορία δημιουργίας φορτίου, χρήση ηλεκτροκινητήρα-εμβόλου ή ηλεκτροκινητήρα-σταθερού βάρους, επομένως προκύπτουν δύο μεγάλες υποκατηγορίες. Η πρώτη αποτελείται από τα μηχανήματα σταθερού φορτίου και η δεύτερη από τα μηχανήματα σταθερής μετατόπισης.

Εν συνεχεία θα παρουσιασθούν και θα σχολιαστούν ορισμένες χαρακτηριστικές εργασίες. Η εργασία των S.Rajesh και N.Saravanan [3] ανήκει στην κατηγορία του σταθερού φορτίου. Στο σχήμα φαίνεται ο ηλεκτροκινητήρας καθώς και το σταθερό φορτίο που κρέμεται με αποτέλεσμα να κάμπει το δοκίμιο. Έτσι χάρη στην περιστροφή του δοκιμίου όλες οι ίνες του θα υποστούν διαδοχικά εφελκυσμό και θλίψη, επομένως το είδος της δύναμης που ασκείται είναι αυτό του εναλλασσόμενου φορτίου. Αξίζει να σημειωθεί ότι στην παρούσα εργασία δεν χρησιμοποιήσαν κάποιον αισθητήρα για να μετρήσουν τους συνολικούς κύκλους κόπωσης. Αλλά χρονομέτρησαν τη διάρκεια του πειράματος και πολλαπλασίασαν τον χρόνο με τις ονομαστικές στροφές του κινητήρα για να υπολογίσουν τους συνολικούς κύκλους. Είναι εύκολα κατανοητό ότι με τη συγκεκριμένη μεθοδολογία θα δημιουργηθεί κάποια απόκλιση στο τελικό αποτέλεσμα. Παράλληλα η τάση για τα διαγράμματα Wöhler υπολογίστηκε με την κλασική σχέση $\frac{\text{φορτίο}}{\text{εμβαδόν κάθετης διατομής}}$. Τέλος πρέπει να σημειωθεί ότι δεν έχουν προνοήσει κάποια αυτοματοποιημένη διαδικασία με την οποία θα μπορεί να γνωρίζει ο χειριστής αν έχει επέλθει η καταστροφή του δοκιμίου, αλλά πρέπει ο ίδιος να το παρατηρεί.

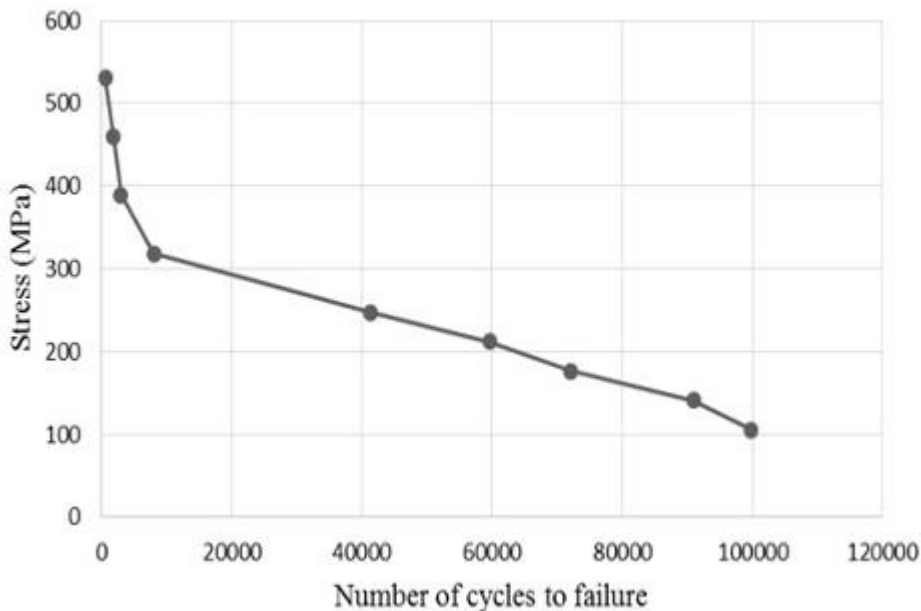


Εικόνα 2 Απεικόνιση μηχανής κόπωσης (Πηγή: S.Rajesh & N.Saravanan (2016))



Εικόνα 3 Η μηχανή κόπωσης (Πηγή: S.Rajesh & N.Saravanan (2016))

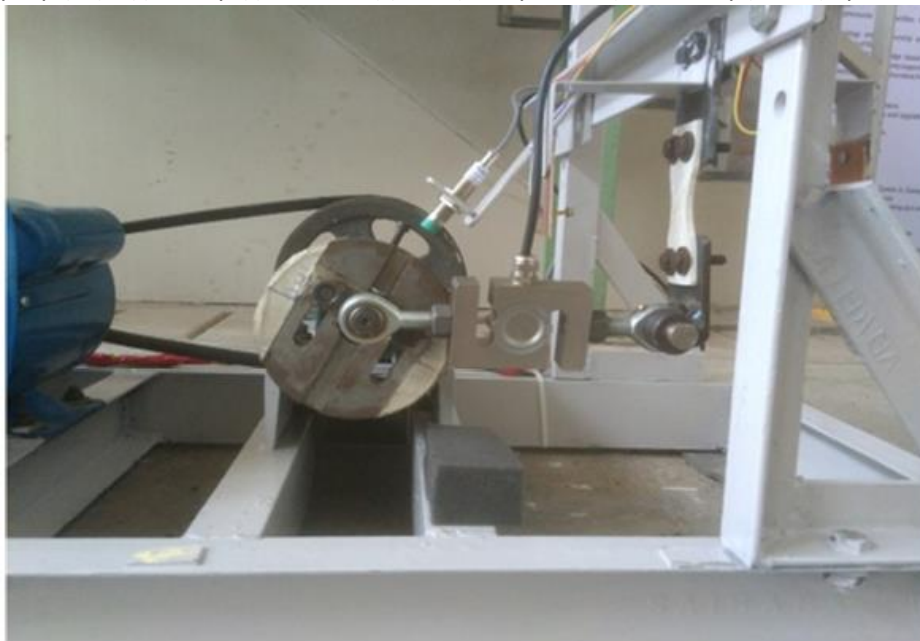
Η εργασία των Martin K. Vincent, Vipin Varghese V. και Suneeth Sukumaran είναι χτισμένη με παρόμοια λογική με την προηγούμενη, δηλαδή η μηχανή είναι σταθερού φορτίου με πανομοιότυπη κατασκευή. Ωστόσο, εδώ χρησιμοποιείται αισθητήρας για τον υπολογισμό των κύκλων κόπωσης, μία πολύ σημαντική βελτίωση για την αξιοπιστία της μηχανής. Παρακάτω παρουσιάζεται το διάγραμμα Wöhler για το απλό χάλυβα 1080 από την εργασία τους. Το όριο διαρκούς αντοχής σύμφωνα με το παρόν διάγραμμα για το συγκεκριμένο υλικό είναι 106.1 MPa, ωστόσο στη σχετική βιβλιογραφία δίνεται 145 MPa. Η σημαντική αυτή απόκλιση οφείλεται σύμφωνα με τους συγγραφείς στη μέθοδο με την οποία το φορτίο επηρεάζει το δοκίμιο, καθώς το φορτίο δεν είναι μόνο εναλλασσόμενο (όπως θα ήταν το επιθυμητό). Σημαντική επίδραση στο δοκίμιο έχει η περιστροφή του και η διακύμανση του φορτίου που σαρώνει ολόκληρο το δοκίμιο. Επίσης αρνητική επίδραση μπορεί να έχουν οι ατμοσφαιρικές συνθήκες και η ανομοιογένεια του δοκιμίου. Από αυτή την εργασία είναι φανερό ότι ο συγκεκριμένος τρόπος εκτέλεσης ενός πειράματος κάμψης μπορεί να αλλοιώσει σημαντικά τα τελικά αποτελέσματα λόγω της μεθόδου εναλλαγής του φορτίου. [4]



Διάγραμμα 1 Το διάγραμμα Wöhler (Πηγή: Martin K. Vincent, Vipin Varghese V., Suneeth Sukumaran (2016))

Η επόμενη εργασία που θα σχολιαστεί ανήκει στην κατηγορία τους σταθερού βέλους κάμψης. Η εργασία των Pritam.V.Kulkarni, P. J. Sawant και V.V.Kulkarni χρησιμοποιεί έναν ηλεκτροκινητήρα για να θέσει σε περιστροφή μία κάμα το άκρο της οποίας είναι συνδεδεμένο με τη μία άκρη του δοκιμίου. Παράλληλα μεταξύ κάμας και δοκιμίου βρίσκεται

μία δυναμοκυψέλη για την μέτρηση της δύναμης. Τέλος χρησιμοποιείται αισθητήρας για την μέτρηση της δύναμης. Η διάταξη αυτή επιτρέπει την εναλλαγή του φορτίου στο δοκίμιο. [5]

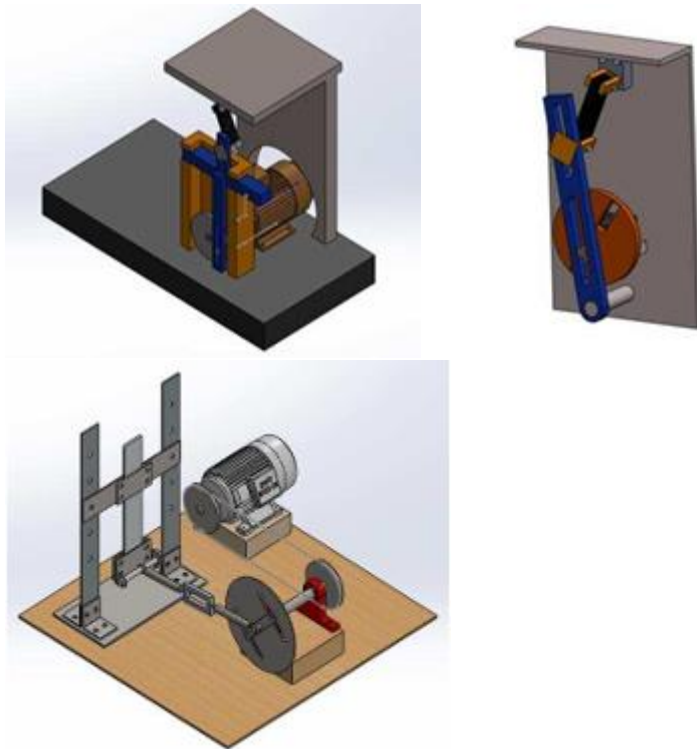


Εικόνα 4 Η πειραματική διάταξη (Πηγή: Pritam. V. Kulkarnia, P. J. Sawantb , V. V. Kulkarnic (2017))

Η εργασία των Ikechukwu, Clementina και Onyebuchi [6] ανήκει στην κατηγορία του σταθερού φορτίου. Ο χρήστης διαλέγει ανάμεσα σε ένα πλήθος σωμάτων αυτό που έχει το κατάλληλο βάρος για την κάθε εφαρμογή. Η παρατήρηση των κύκλων κόπωσης γίνεται με τη βοήθεια ενός μετρητή ο οποίος εμφανίζει το αποτέλεσμα σε μία ψηφιακή οθόνη και έχει τη δυνατότητα να μετράει μέχρι έναν συγκεκριμένο αριθμό. Έτσι σε πειράματα μεγάλης διάρκειας θα χρειαστεί να τοποθετηθεί και κάποιο μέσο για την καταγραφή των κύκλων. Τέλος έχει τοποθετηθεί διακόπτης ο οποίος διακόπτει το ρεύμα στον ηλεκτροκινητήρα όταν το ρουλεμάν δεχθεί μία κλίση 3ο μοιρών.

Οι McAlorum, Rubert, Fusiek, Niewczas και Zorzi [7] κατασκεύασαν μία μηχανή κόπωσης για την μελέτη κομματιών τσιμέντου συγκεκριμένων διαστάσεων στα οποία προϋπάρχει μία ρωγμή, επομένως βασική απαίτηση για τη συγκεκριμένη εργασία είναι το κλείσιμο και το άνοιγμα της. Στην εργασία υπάρχει και η απαίτηση (σύμφωνα με τις Βρετανικές προδιαγραφές) το φορτίο που εφαρμόζεται να είναι ρεαλιστικό, δηλαδή πρέπει να μπορεί να πάρει τη μορφή οποιουδήποτε κύματος με μεταβαλλόμενο πλάτος και συχνότητα. Για την ικανοποίηση αυτών των απαιτήσεων επιλέχθηκε σερβοκινητήρας. Η χρήση υδραυλικού επενεργητή απορρίφθηκε λόγω κόστους και ενός πνευματικού λόγω χαμηλής απόδοσης. Σε αυτούς τους παράγοντες πρέπει να προστεθεί και η πολυπλοκότητα των συστημάτων αυτών.

Το ζήτημα του υψηλού κόστους αγοράς και συντήρησης των εμπορικών μηχανών κόπωσης αναλύεται στην εργασία των Landge, Patil, Talekar, Ghosh και Bhaumik [8] και παρουσιάζονται ένα σύνολο διατάξεων για την κατασκευή μηχανών κόπωσης χαμηλού κόστους που ανήκουν στην κατηγορία της σταθερής μετατόπισης.



Εικόνα 5 Οι διατάξεις χαμηλού κόστους που προτάθηκαν

Η εργασία των Kattimani, Khatib, Ghor, Sajjad, Jahagir, Baqtaiyan και Sadiq [9] αφορά την κατασκευή μίας μηχανής κόπωσης σταθερού βάρους. Η έδραση και η περιστροφή του δοκιμίου επιτυγχάνεται με την χρήση εδράνων κύλισης και ενός ηλεκτροκινητήρα. Το σταθερό φορτίο που κρεμάτε από τη διάταξη και κάμπτε το δοκίμιο έχει ως αποτέλεσμα την εναλλασσόμενη θλίψη-εφελκυσμό του ήμισυ του δοκιμίου. Η καταμέτρηση των κύκλων κόπωσης επιτυγχάνεται με την χρήση ενός αισθητήρα Hall και η αποτύπωση τους γίνεται σε μία ψηφιακή οθόνη. Η τιμή της τάσης για την εξαγωγή των διαγραμμάτων προκύπτει από τις κατάλληλες εξισώσεις.

Η εργασία του Handi [10] λαμβάνει ως αφορμή το υψηλό κόστος απόκτησης μία μηχανής κόπωσης στη βιομηχανία, το κόστος μπορεί να υπερβαίνει τις 150000\$, και κατασκευάζει μία μηχανή κόπωσης σταθερού βάρους. Ο τερματισμός της λειτουργίας του ηλεκτροκινητήρα γίνεται με τη χρήση ενός διακόπτη, μόλις το δοκίμιο σπάσει το ένα άκρο που το κρατάει πέφτει και πατάει το κουμπί του διακόπτη τερματίζοντας την λειτουργία του ηλεκτροκινητήρα.

Η εργασία των Nagabhooshnam, Baskar, Nagarajan και Kumar [11] έχει ως στόχο την κατασκευή και αξιολόγηση μίας μηχανής κόπωσης για ένα σύνθετο υλικό με ίνες από λινάρι και μήτρα από εποξική ρητίνη. Η μηχανή που κατασκευάστηκε είναι σταθερής μετατόπισης του δοκιμίου. Η κίνηση της διάταξης γίνεται με τη χρήση ενός ηλεκτροκινητήρα. Η τελική διάταξη είναι ιδιαίτερα χαμηλού κόστους και μπορεί να χρησιμοποιηθεί με ευκολία για πανεπιστημιακούς σκοπούς.

Η εργασία των Wang, Zhang, Zhou, Wei, Ding και Li [12] πραγματεύεται τον εντοπισμό ρωγμών λόγω κόπωσης με την χρήση μηχανικής όρασης. Οι αρθογράφοι συνέλεξαν φωτογραφίες με ρωγμές από διάφορες πηγές και τις κανονικοποίησαν ώστε να μην επηρεαστεί αρνητικά η απόδοση του δικτύου. Κατόπιν η επαύξηση των δεδομένων έγινε με φυσικές μεθόδους, δηλαδή μετακίνηση της εικόνας, αντιστροφή και συνδυασμό των δύο μεθόδων.

Το ζήτημα του εντοπισμού επιφανειακών ατελειών με τη χρήση δικτύων συνέλιξης πραγματεύεται η εργασία των Singh και Desai [13]. Οι συγγραφείς χρησιμοποιούν ένα προεκπαιδευμένο δίκτυο συνέλιξης ResNet το οποίο έχει μικρότερη πολυπλοκότητα από άλλα δίκτυα. Η επαύξηση των δεδομένων σε αυτή την περίπτωση γίνεται μέσω της περιστροφής, αυτό διότι στη συγκεκριμένη εφαρμογή η θέση της κάμερας και ο φωτισμός είναι σταθερά. Σε μία δεύτερη εργασία τους, με την συμμετοχή του Kumar, [14] προτείνουν τη δημιουργία dataset με την χρήση αφινικού μετασχηματισμού για την επαύξηση ενός συνόλου δεδομένων.

Η εργασία των Kang Park, Kwon, Hyub Park και Kang [15] πραγματεύεται την απόδοση διαφόρων μεθόδων εντοπισμού ατελειών για διάφορες επιφάνειες υλικών. Οι συγγραφείς δημιούργησαν ένα δίκτυο συνέλιξης και το συνέκριναν με το φίλτρο Gabor, με τη μέθοδο ICA και τη μέθοδο του τυχαίου δάσους (Random Forest). Στην αρχή σε κάθε επιφάνεια υλικού εφαρμόστηκε η κάθε μέθοδος. Ύστερα τα μοντέλα εκπαιδεύθηκαν ώστε να εντοπίζουν τις ατέλειες ανά ζευγάρια υλικών. Στο πρώτο σενάριο το σφάλμα του δικτύου συνέλιξης ήταν ιδιαίτερα χαμηλό. Αντίθετα στο δεύτερο η απόδοσή του ήταν πολύ χαμηλότερη. Οι συγγραφείς αναφέρουν ότι αυτό συμβαίνει εξαιτίας της χαμηλής ποσότητας δεδομένων για την εκπαίδευση. Παράλληλα πρέπει να ληφθεί υπόψιν ότι αν οι φωτογραφίες έχουν μεγάλη πυκνότητα σε πληροφορία τότε το βάθος δικτύου πρέπει να αυξηθεί, έτσι αυξάνεται και το υπολογιστικό κόστος. Επίσης δίχως κάποια ανάλυση δεδομένων είναι πιθανό να μην παρατηρηθεί κάποια σημαντική βελτίωση στην αξιοπιστία του δικτύου. Τέλος τα αποτελέσματα του δικτύου συγκρίνονται με την ακρίβεια που θα είχε η αξιολόγηση των υλικών από άνθρωπο.

Η εργασία των Weimer, Scholz-Reiter και Shpitalni [16] συγκρίνει την απόδοση δικτύων συνέλιξης με διαφορετικό αριθμό φίλτρων και στρωμάτων. Παρατηρείται ότι η αύξηση των στρωμάτων βελτιώνει την απόδοση, ενώ η αύξηση του αριθμού των φίλτρων δεν παρουσιάζει σημαντική βελτίωση. Αυτό συμβαίνει διότι οι ατέλειες έχουν χαμηλή γεωμετρική πολυπλοκότητα.

Η εργασία των Xu, Lv, Deng και Li [17] πραγματεύεται τις αποδόσεις διαφόρων αρχιτεκτονικών νευρωνικών δικτύων για τον εντοπισμό ατελειών. Οι συγγραφείς μεταβάλλουν τον αριθμό και το είδος των στρωμάτων (pooling, συνέλιξης), το μέγεθος των φίλτρων και την αρχιτεκτονική.

Η εργασία των Sundaram και Zeid [18] αφορά τον έλεγχο ποιότητας στη χύτευση με διάφορες μεθόδους. Οι συγγραφείς χρησιμοποιούν μία σειρά μεθόδων όπως δίκτυα συνέλιξης και προεκπαιδευμένα δίκτυα. Παράλληλα προτείνουν μία μεθοδολογία έξυπνου ελέγχου ποιότητας ο οποίος θα βελτιώσει την διαδικασία.

Η εργασία των Tong, Lin, Hua, Gao και Zhang [19] προτείνει την χρήση ολικών δικτύων συνέλιξης (FNN) και τη δημιουργία χαρτών μέσω αποσυνέλιξης για την κατηγοριοποίηση των εικόνων. Αρχικά τρέχουν διάφορα κλασσικά δίκτυα συνέλιξης για περιοχές της εικόνας, αν αποφανθούν ότι πρόκειται για ρωγμή τότε η έξοδος τους γίνεται είσοδος για το δίκτυο FNN. Η διαδικασία αυτή μπορεί να μειώσει το κίνδυνο λάθους κατηγοριοποίησης από εξωτερικούς παράγοντες, όπως σκιές, σημάδια στο υλικό και συγκολλήσεις.

Η εργασία των Zhang, Hao, Wang, Qin και Zhao [20] προτείνει τη μέθοδο της οργανωμένης εστίασης (coordinated attention) για τον εντοπισμό επιφανειακών ατελειών. Αυτό το πετυχαίνει με τη δημιουργία κάθετων και οριζόντιων πινάκων άθροισης και την κατάλληλη κωδικοποίηση τους μέσω πράξεων συνέλιξης, κανονικοποίησης και hard-swiss. Έτσι ο καθένας από τους δύο πίνακες εστιάζει στην αντίστοιχη κατεύθυνση. Έπειτα οι δύο αυτοί πίνακες πολλαπλασιάζονται με την είσοδο και προκύπτει ένας νέος πίνακας που δίνει έμφαση

στις περιοχές ενδιαφέροντος του χρήστη. Απότοκο αυτής της διαδικασίας είναι η σημαντική μείωση του υπολογιστικού κόστους.

Η εργασία των Sik, Atak, Yanvz και Ozdemir [21] παρουσιάζει συνοπτικά τις βασικές αρχές λειτουργίας μίας μηχανής κόπωσης και τους του λόγους για τους οποίους είναι επικτακτική ανάγκη η κατασκευή της.

Η εργασία των Xiong, Huang και Tian [22] λαμβάνει ως αφορμή ρωγμές που υπάρχουν σε κτήρια και παρουσιάζει μία μεθοδολογία για την επεξεργασία της εικόνας, ώστε να χρησιμοποιηθεί για τον εντοπισμό ρωγμών. Συγκεκριμένα προτείνει την χρήση ασπρόμαυρης φωτογραφίας, διαφόρων μεθόδων καθαρισμού της εικόνας από θόρυβο και επίλυσης ζητημάτων που σχετίζονται με τον άνισο φωτισμό του χώρου.

Η εργασία των N. Ghalyan, I. Ghalyan και Ray [23] πραγματεύεται τον εντοπισμό ρωγμών με τη χρήση της μεθόδου Bag-of-Words (BoW). Η διαδικασία ξεκινάει με την εξαγωγή των χαρακτηριστικών των φωτογραφιών, κατόπιν η εξαγωγή των ιστογραμμάτων και τέλος με βάση τα διάφορα ιστογράμματα μπορεί να γίνει η κατηγοριοποίηση. Ο χρήστης μπορεί να χρησιμοποιήσει όποια μέθοδο κατηγοριοποίησης επιθυμεί, όπως ένα δέντρο αποφάσεων.

Η εργασία των Alqahtani, Bharadwaj και Ray [24] παρουσιάζει μία μέθοδο κατηγοριοποίησης των ρωγμών για πολυκρυσταλλικά υλικά. Η κατηγοριοποίηση των δοκιμών σε υγιή, χαμηλού ρίσκου και υψηλού ρίσκου γίνεται με βάση την εξής διαδικασία. Αρχικά οι συγγραφείς υπολογίζουν το κρίσιμο μήκος της ρωγμής. Έπειτα παρατηρούν ότι υπάρχει ισχυρή συσχέτιση μεταξύ της μεταβολής του ανοίγματος της κορυφής της ρωγμής (Crack Tip Opening Displacement – CTOD) και του μήκους της ρωγμής. Έτσι μπορούν να κάνουν την κατηγοριοποίηση με βάση το CTOD. Οι φωτογραφίες του συνόλου των δεδομένων αποθηκεύτηκαν ως ασπρόμαυρες και μειώθηκαν οι διαστάσεις τους. Εν συνεχεία έγινε μία επαύξηση του συνόλου δεδομένων με τετριμένες μεθόδους (περιστροφή, μετατόπιση, αναστροφή και μεγέθυνση). Τέλος πραγματοποιήθηκε η κατασκευή και η εκπαίδευση του νευρωνικού.

Η εργασία των Min, Vandenberg, Dowling, Holloway και Haworth [25] πραγματεύεται τις διάφορες μεθόδους για την επαύξηση ενός συνόλου δεδομένων εικόνων, μέσω τόσο συμβατικών μεθόδων (μεταβολές στις εικόνες) όσο και πιο προχωρημένων (Παραγωγικά Αντιπαραθετικά Δίκτυα – GAN).

4. Τροποποίηση εργαστηριακής μηχανής

4.1 Λειτουργία υπάρχουσας μηχανής

Η μηχανή βρίσκεται στο Εργαστήριο Τεχνολογίας των Κατεργασιών της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ. Πρόκειται για μηχανική πρέσσα εκκέντρου απλής ενέργειας με μέγιστη διαδρομή 15 mm. Απαρτίζεται από έναν ηλεκτροκινητήρα μέσω του οποίου και με χρήση κατάλληλης μηχανικής διάταξης κινείται το έμβολο εκτελώντας μία κατακόρυφη κίνηση. Η συχνότητα του οποίου ελέγχεται με τη βοήθεια ενός inverter και η επιλογή της γίνεται μέσω ενός ποτενσιόμετρου το οποίο ο χρήστης ρυθμίζει κατάλληλα. Ο inverter που χρησιμοποιήθηκε είναι ο CDS 75, ο οποίος λειτουργεί με τάση εισόδου 200-240V AC και συχνότητα 48-62 Hz και κυκλοφορεί σε ένα εύρος ισχύος από 0.75 ως 2.2 kW. Στο κρεβάτι υπάρχει ειδική βάση για την τοποθέτηση των δοκιμών και την πραγματοποίηση κάμψης τριών σημείων. Το πείραμα πραγματοποιείται χωρίς τα άκρα του δοκιμίου να είναι πακτωμένα. Τέλος υπάρχει και ένας μαγνητικός αισθητήρας συνδεδεμένος με μετρητή για την καταμέτρηση των κύκλων της μηχανής. Η παρούσα διάταξη χρησιμοποιούταν από το εργαστήριο για εκτέλεση πειραμάτων κόπωσης σε κάμψη τριών σημείων.

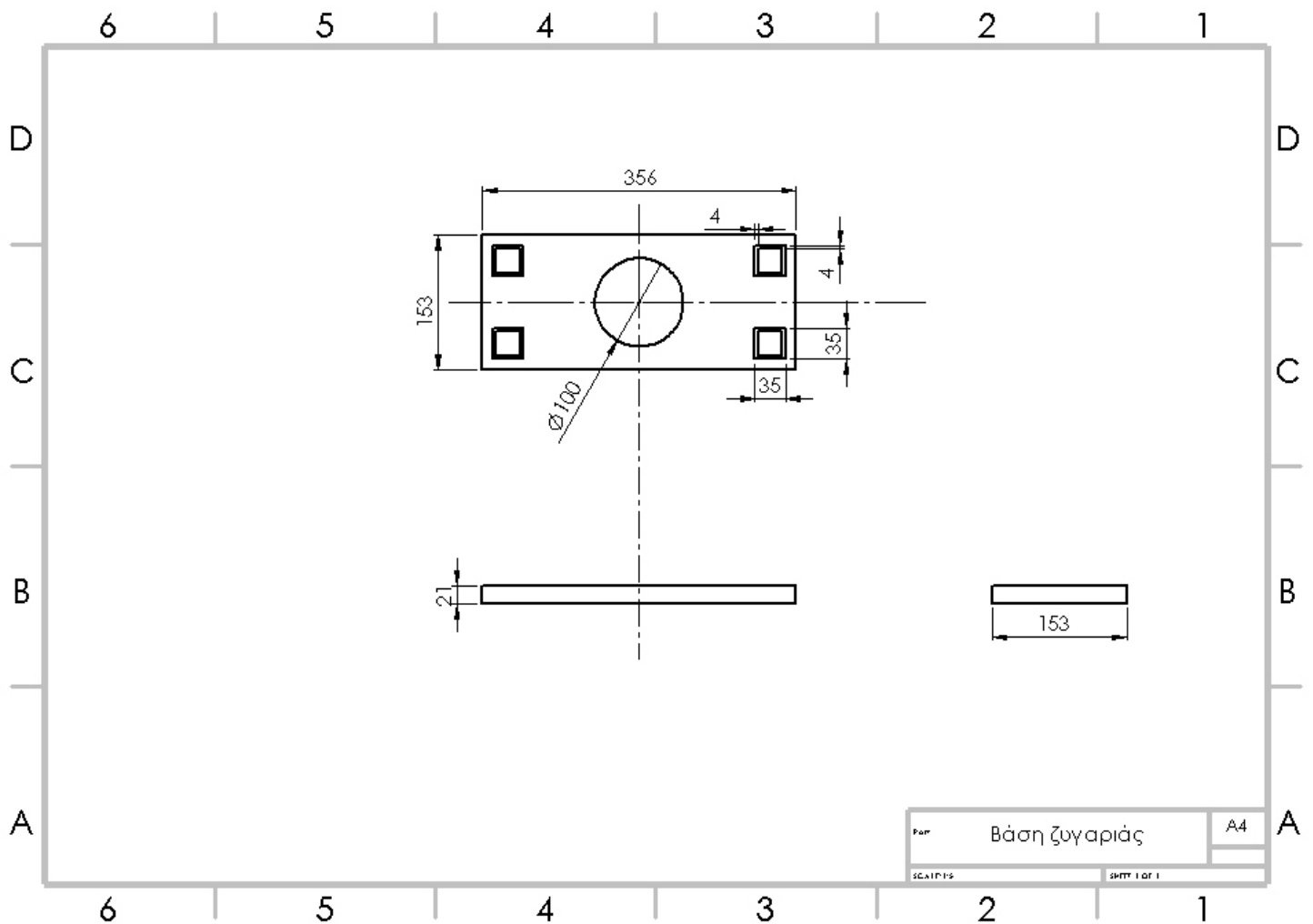
4.2 Σχεδιασμός και έλεγχος μετρητικών οργάνων

Τα ηλεκτρολογικά μέρη που προστέθηκαν για την ορθή λειτουργία της μηχανής είναι τρία. Αρχικά σχεδιάστηκε ένας μετρητής κύκλων της μηχανής, κάνοντας χρήση του αισθητήρα Hall, ο οποίος έχει τη δυνατότητα αποθήκευσης των δεδομένων σε αρχείο .txt. Κατόπιν κατασκευάστηκε ζυγαριά για τη μέτρηση των κάθετων δυνάμεων που επιδρούν στο δοκίμιο και αποθήκευση των τιμών τους στον υπολογιστή. Στη συνέχεια τοποθετήθηκε κάμερα η οποία με τη βοήθεια νευρωνικού δικτύου μπορεί να εντοπίζει την εμφάνιση της ρωγμής. Τέλος χρειάστηκε και ο σχεδιασμός ενός ελεγκτή για τη διακοπή και εκκίνηση της λειτουργίας της μηχανής. Στο παρόν κεφάλαιο θα παρουσιαστεί η λειτουργία, σχεδιασμός και εγκατάσταση του καθενός από τα προαναφερόμενα μέρη.

4.2.1 Ζυγαριά

Κατασκευή βάσης

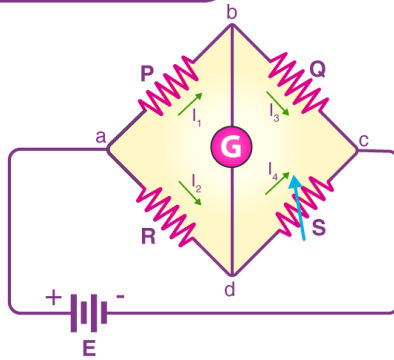
Η κατασκευή της βάσης της ζυγαρίας έγινε με κατάλληλη κατεργασία πλάκας χάλυβα υψηλής αντοχής τύπου Sverker. Η συγκεκριμένη επιλογή έγινε με γνώμονα τη στιβαρότητα που θα προσέφερε το συγκεκριμένο έλασμα όταν θα δεχόταν το φορτίο. Ήταν απαραίτητη η δημιουργία ποκετών στις τέσσερις γωνίες της πλάκας για την τοποθέτηση των αισθητήρων βάρους. Παράλληλα ανοίχτηκε οπή στο κέντρο της πλάκας κατάλληλης διαμέτρου, για την τοποθέτηση της κάμερας.



Εικόνα 6 Κατασκευαστικό σχέδιο βάσης ζυγαριάς

Θεωρία γέφυρας Wheatstone

Η χρήση των ζυγαριών με αρχή λειτουργίας ηλεκτρομηκνιομέτρου απαντάται στην καθημερινότητα, από οικιακή χρήση ως ζυγαριές για τη μέτρηση εμπορευμάτων. Ωστόσο λίγοι γνωρίζουν τον μηχανισμό της γέφυρας Wheatstone με τον οποίο οι ζυγαριές λειτουργούν και παρέχουν έγκυρες μετρήσεις. Η διάταξη εφευρέθηκε από τον Samuel Hunter Christie το 1833 και δημοσιοποιήθηκε μία δεκαετία αργότερα από τον Charles Wheatstone. Η αρχή λειτουργίας βασίζεται στα ηλεκτρομηκνισιόμετρα (strain gauges), τα οποία έχουν την ιδιότητα να μεταβάλλουν την αντίστασή τους όταν μεταβάλλεται το μήκος τους. Έτσι τοποθετώντας σε μία διάταξη, σαν την παρακάτω, ένα strain gauge ως μεταβλητή αντίσταση και τρεις σταθερές γνωστές αντιστάσεις όταν εφαρμοστεί ένα μηχανικό φορτίο στο στοιχείο το μήκος του strain gauge θα μεταβληθεί και κατ' επέκταση η τάση που μετρείται μεταξύ των σημείων bd θα αλλάξει. Επομένως με κατάλληλη βαθμονόμηση μπορεί πολύ εύκολα να μετρείται η δύναμη που ασκείται στο στοιχείο.



Εικόνα 7 Σχήμα γέφυρας Wheatstone (Πηγή ιστοσελίδα: <https://byjus.com/physics/wheatstone-bridge/>)

Επιλογή αισθητήρα

Η επιλογή των κατάλληλων αισθητήρων έγινε κατόπιν πειράματος κάμψης τριών σημείων στη μηχανή δοκιμών Instron 4482 του εργαστηρίου, έτσι ώστε να προσδιοριστούν οι απαιτήσεις των αισθητήρων. Το μέγιστο βέλος κάμψης προέκυψε 32mm και το μέγιστο φορτίο ίσο με 500N. Τελικά επιλέχθηκαν να τοποθετηθούν τέσσερις αισθητήρες φορτίου (load cells) των 50 kg τόσο για λόγους ισορροπίας της ζυγαριάς όσο και για την επίτευξη συμβατότητας με τον ενισχυτή. Η βαθμονόμηση της διάταξης έγινε με τη χρήση γνωστού βάρους και ανοιχτού κώδικα.

Capacity	kg	40-50
Comprehensive Error	mV/V	0.05
Output Sensitivity	mV/V	1.0 ± 0.1
Nonlinearity	%FS	0.03
Repeatability	%FS	0.03
Hysteresis	%FS	0.03
Creep	(3min)%FS	0.03
Zero Drift	(1min)%FS	0.03
Temp. Effect on Zero	%FS/10°C	1
Temp. Effect on Output	%FS/10°C	0.05
Zero Output	mV/V	± 0.1
Input Resistance	Ω	1000 ± 20
Output Resistance	Ω	1000 ± 20
Insulation Resistance	M Ω	≥ 5000
Excitation Voltage	V	≤ 10
Operation Temp. Range	°C	0--+50
Overload Capacity	%FS	150

Πίνακας 1 Τα χαρακτηριστικά των δυναμοκυψελών

Στον Πίνακα 1 αναγράφονται τα ποιοτικά χαρακτηριστικά του αισθητήρα που επιλέχθηκε. Επιγραμματικά θα σχολιαστούν ορισμένα από αυτά. Ο κάθε αισθητήρας έχει ικανότητα να μετρήσει μέγιστο βάρος 40-50 kg. Η μέγιστη τάση εισόδου που μπορεί να δεχθεί για την ομαλή λειτουργία του είναι 10V, ενώ η τάση εξόδου που μπορεί να δώσει σε σχέση με την είσοδο είναι 1mV/V (Output Sensitivity). Το συγκεκριμένο χαρακτηριστικό δίνει μία πρώτη εικόνα στον χρήστη για την ποιότητα του αισθητήρα που επιλέγει. Πολύ σημαντικό χαρακτηριστικό είναι το

creep rate, ορίζεται ως το ποσοστό απόκλισης

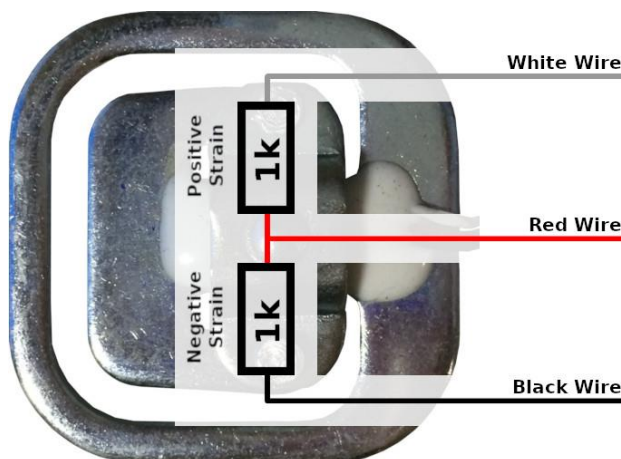
του output sensitivity όταν στον αισθητήρα εφαρμόζεται το 90-100% του μέγιστου φορτίου για ορισμένο χρονικό διάστημα. Στην περίπτωση της παρούσας διάταξης το μέγιστο φορτίο (ή έστω κοντά σε αυτό) δεν θα το υποστεί, αφού ο κάθε αισθητήρας θα δεχθεί περίπου το πολύ $50/4=12.5\text{kg}$, επομένως και το φαινόμενο του creep rate θα περιοριστεί σημαντικά. Παράλληλα το creep rate επηρεάζει κυρίως ζυγαριές οι οποίες δέχονται φορτίο για μεγάλα χρονικά διαστήματα, όπως σε εφαρμογές ζυγιστικών σιλό. Το zero drift αναφέρεται στο φαινόμενο κατά το οποίο η τάση εξόδου μεταβάλλεται στο μηδενικό φορτίο. Αυτό μπορεί να οφείλεται σε κατασκευαστικά λάθη, τη θερμοκρασία και άλλους παράγοντες. Είναι πιθανό

ωστόσο το drift γενικότερα να μην εμφανίζεται μόνο στο μηδενικό φορτίο, αλλά να επεκταθεί σε οποιοδήποτε σταθερό φορτίο [26],[27].



Εικόνα 8 Το load cell που χρησιμοποιήθηκε (Πηγή ιστοσελίδα: <https://grobotronics.com/load-sensor-50kg.html?sl=en>)

Σύμφωνα με τη παρουσίαση που έγινε στην προηγούμενη ενότητα τα load cells που λειτουργούν με τη γέφυρα Wheatstone χρειάζονται στη γενική τους περίπτωση τέσσερα άκρα (δύο για τη τροφοδοσία και δύο για την έξοδο). Ωστόσο η διάταξη του load cell που χρησιμοποιήθηκε είναι λίγο διαφορετική. Έχει μόνο δύο αντιστάσεις εκ των οποίων η μία είναι μεταβλητή και η έξοδος δίνεται μόνο από ένα (κόκκινο) καλώδιο. Η μορφή αυτή είναι ουσιαστικά μία ημι-γέφυρα.

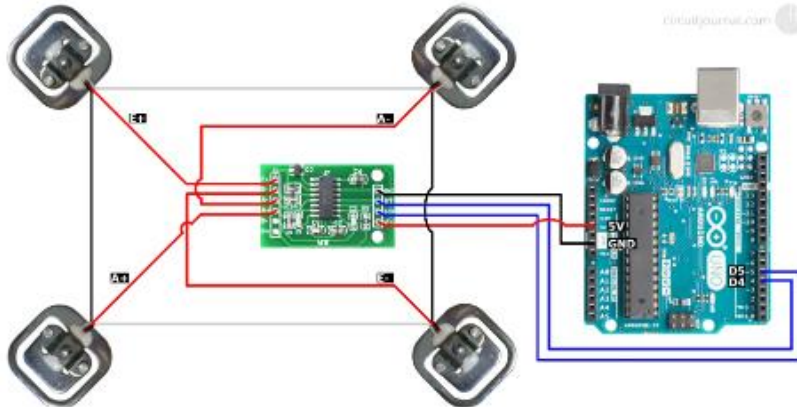


 circuitjournal.com

Εικόνα 9 Η διάταξη του load cell (Πηγή ιστοσελίδα: <https://circuitjournal.com/50kg-load-cells-with-HX711>)

Επιλογή μετατροπέα

Η σύνδεση ενός αισθητήρα με τόσο μικρές τάσεις εξόδου δεν θα μπορούσε να γίνει με κάποια ηλεκτρονική ιδιοκατασκευή αλλά χρειάζεται τη χρήση ειδικού ενισχυτή προορισμένου για την ανάγνωση τέτοιου τύπου load cells. Έτσι επιλέχθηκε ο αναλογοψηφιακός μετατροπέας HX711 ο οποίος χρησιμοποιείται κυρίως σε εφαρμογές ζυγαριών. Η διάταξη με την οποία έγινε η σύνδεση αισθητήρων, μετατροπέα και μικροελεγκτή φαίνεται στην παρακάτω εικόνα. [28]



Εικόνα 10 Το σχέδιο σύνδεσης load cells-HX711-Arduino (Πηγή ιστοσελίδα: <https://circuitjournal.com/50kg-load-cells-with-HX711>)

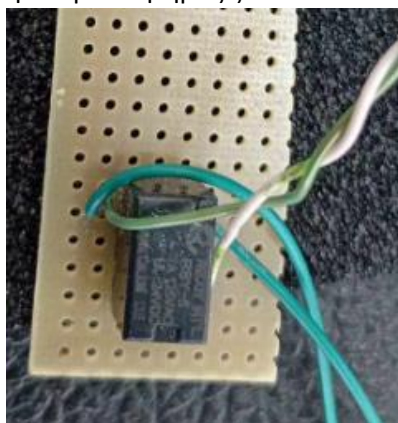


Εικόνα 11 Η ζυγαριά

Η βαθμονόμηση και η επικοινωνία της διάταξης έγινε με κατάλληλο κώδικα που φαίνεται στα αντίστοιχα παράρτηματα 7.5 και 7.3.

4.2.2 Έλεγχος

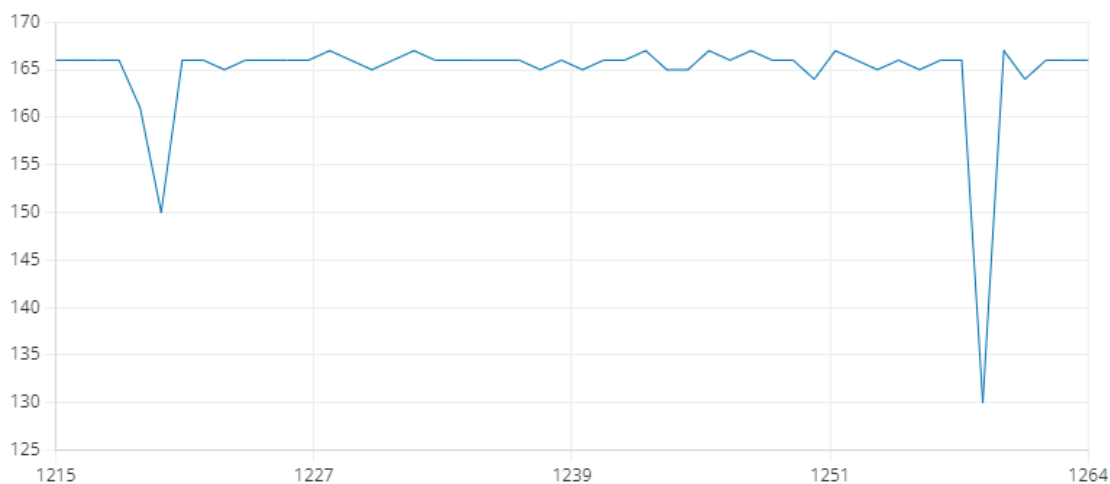
Η δυνατότητα παύσης της διαδικασίας, κατά τη λήψη της εικόνας και επεξεργασίας της από το νευρωνικό δίκτυο, και κατόπιν η επανεκκίνηση της θεωρήθηκε απαραίτητη για την εύρυθμη λειτουργία της διάταξης. Η ανάγκη αυτή ικανοποιήθηκε με τη χρήση ρελέ και τον έλεγχο του από μικροελεγκτή arduino. Όταν το πλήθος των κύκλων κόπωσης γίνει πολλαπλάσιο του αριθμού των κύκλων για τον οποίο έχει επιλεγθεί να γίνεται χρήση του νευρωνικού δικτύου, η κατάσταση της εξόδου του μικροελεγκτή που ευθύνεται για τον έλεγχο του ρελέ αλλάζει. Γίνεται υψηλή (High), ανοίγοντας το ρελέ και διακόπτοντας τη λειτουργία του inverter και κατ' επέκταση του ηλεκτροκινητήρα. Αφού ολοκληρωθεί η διαδικασία του νευρωνικού δικτύου τότε η κατάσταση της εξόδου γίνεται χαμηλή (Low) και ο ηλεκτροκινητήρας ξεκινάει πάλι την λειτουργία του.



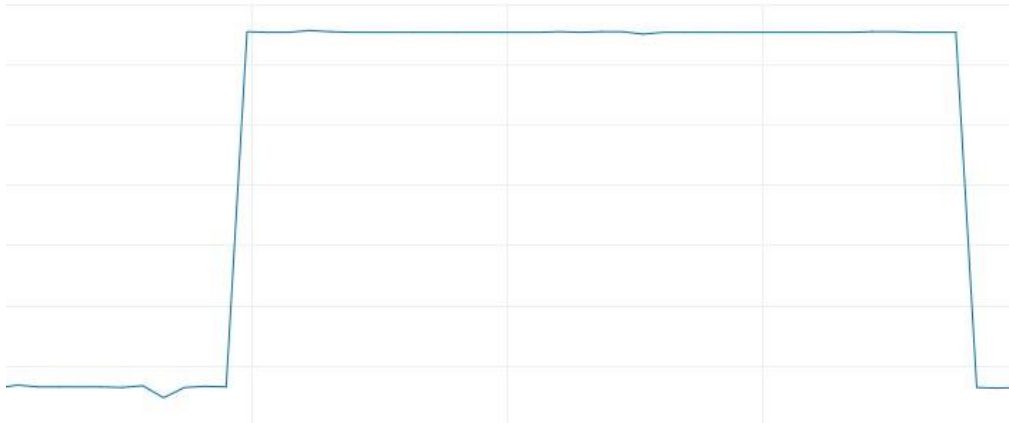
Εικόνα 12 Το ρελέ για τον έλεγχο της μηχανής

4.2.3 Μετρητής κύκλων

Βασικό σκέλος της παρούσας διπλωματικής είναι η ορθή παρακολούθηση και καταγραφή των κύκλων της μηχανής. Για τον λόγο αυτό χρησιμοποιήθηκε αισθητήρας Hall τοποθετημένος κατάλληλα έτσι ώστε να ενεργοποιείται όταν το έμβολο φτάνει στο άνω σημείο του. Ωστόσο λόγω του ηλεκτρικού θορύβου το σήμα που έφτανε στον μικροελεγκτή ήταν αρκετά παραμορφωμένο και απείχε αρκετά από ένα ιδανικό παλμικό σήμα. Έτσι αποφασίστηκε η κατασκευή ενός φίλτρου πρώτης τάξης και με αυτό το τρόπο το σήμα πήρε τη σωστή του μορφή.

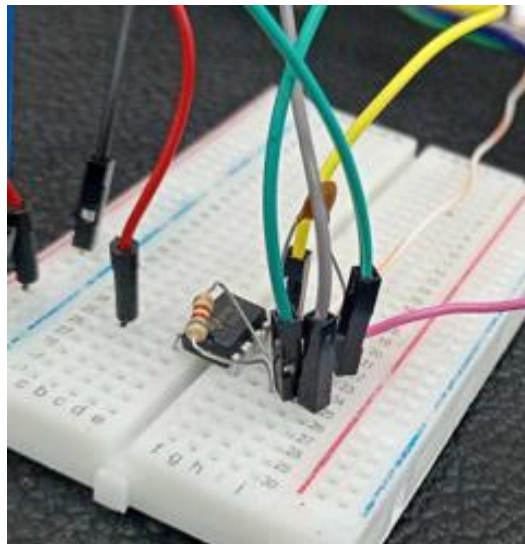


Εικόνα 13 Το σήμα χωρίς την ύπαρξη φίλτρου



Εικόνα 14 Το σήμα μετά την χρήση φίλτρου

Ο κώδικας παρουσιάζεται στο αντίστοιχο παράρτημα (βλ. Κώδικας ελέγχου Arduino), έγινε χρήση ενός interrupt function όταν το σήμα κινούταν από το χαμηλό στο υψηλό του. Η μεταβλητή threshold υπάρχει για να προσδιορίζει τι θα θεωρεί ο μικροελεγκτής ως κύκλο μηχανής και να διασφαλίζει ότι τυχαίες μεταβολές στο σήμα δεν θα υπονομεύσουν την απόδοση του μετρητή. Η κατάλληλη τιμή για τη μεταβλητή threshold (200) επιλέχθηκε ύστερα από πειραματισμούς και επιβεβαίωση των αποτελεσμάτων με μετρητή που χρησιμοποιείται στη βιομηχανία. Τέλος υπάρχει μία έξοδος στη σειριακή η οποία αξιοποιείται ύστερα από το πρόγραμμα στον υπολογιστή.



Εικόνα 15 Το φίλτρο και η ενίσχυση του μετρητή των κύκλων κόπωσης

4.2.4 Κάμερα

Χρησιμοποιήθηκε η κάμερα DFK MKU130-10x22 του εργαστηρίου. Ο φωτισμός του δωματίου υπήρξε ικανοποιητικός και έτσι δεν χρησιμοποιήθηκε επιπλέον φως (π.χ. ταινία LED). Ο έλεγχος της κάμερας καθώς και μία απαραίτητη επεξεργασία της φωτογραφίας πριν αυτή εισαχθεί στο νευρωνικό έγιναν με τη χρήση της Python. Όταν χρειαστεί το πρόγραμμα να μπει στο σκέλος στο οποίο καλεί το νευρωνικό δίκτυο τότε γίνεται η λήψη της εικόνας. Ωστόσο η εικόνα αυτή δεν είναι σε ίδια μορφή με αυτή που έγινε η εκπαίδευση του νευρωνικού δικτύου (μαύρο φόντο και να φαίνεται μόνο το δοκίμιο), αλλά περιέχει πληροφορία από το περιβάλλον. Επομένως μία περικοπή είναι απαραίτητη. Ο κώδικας κρατάει μόνο το δοκίμιο σύμφωνα με τις τιμές για τη θέση του στην εικόνα ύστερα από δοκιμές. Κατόπιν γίνεται μία σμίκρυνση της εικόνας, αυτό ήταν απαραίτητο ώστε να μειωθεί η κατανάλωση της μνήμης και το υπολογιστικό κόστος, αφού οι δυνατότητες του υπολογιστή ήταν περιορισμένες. Παράλληλα η σταθερή είσοδος στο νευρωνικό δίκτυο βελτιώνει την ταχύτητα σύγκλισης, τον υπολογισμό της παραγώγου του νευρωνικού και διασφαλίζει ότι οποιαδήποτε είσοδος θα είναι παρόμοια με την προηγούμενη. Η αναπροσαρμογή (rescale) των τιμών των εικονοστοιχείων (pixel) είναι μία συνήθης διαδικασία η οποία βελτιώνει την απόδοση της παραγώγου, αποτρέπει αστάθειες στη συνάρτηση σφάλματος, κατά τη διάρκεια της εκπαίδευσης και κανονικοποιεί τη διαδικασία. Έτσι αφού ολοκληρωθεί και αυτό το βήμα η εικόνα είναι έτοιμη να εισαχθεί στο νευρωνικό.

4.2.5 Κώδικας

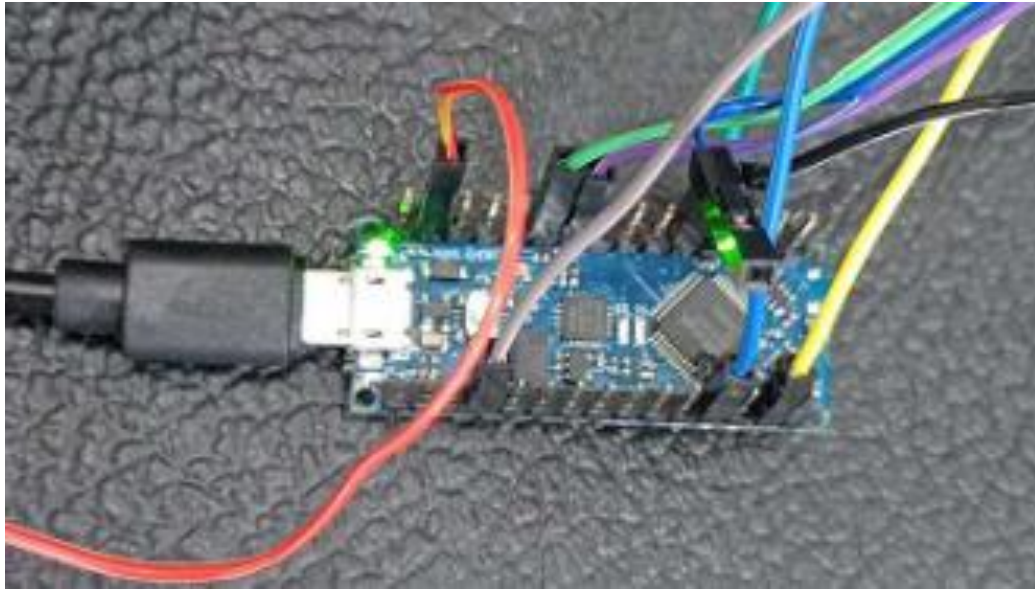
Ο έλεγχος των μετρητικών οργάνων έγινε μέσω ενός μικροελεγχτή Arduino και η σύνδεση του συστήματος με τον Η/Υ και το νευρωνικό δίκτυο επιτεύχθηκε με τη χρήση της Python. Οι δύο κώδικες λειτουργούν παράλληλα και η επεξήγηση τους θα γίνει με τη σειρά που συμβαίνουν τα γεγονότα που περιγράφουν.

Αρχικά η εντολή `interrupt` παρατηρεί πότε θα ολοκληρωθεί ένας κύκλος κόπωσης, μόλις γίνει αυτό ανεβάζει τη μεταβλητή `counter` (πλήθος κύκλων) και προσθέτει στην μεταβλητή `max_force` (το άθροισμα των μέγιστων φορτίων κάθε κύκλου για ένα συγκεκριμένο πλήθος κύκλων). Η μεταβλητή `max_weight` ορίζεται ως το μέγιστο φορτίο ενός κύκλου και προκύπτει μέσω της ακόλουθης διαδικασίας. Κατά τη διάρκεια του πειράματος ο ADC μετατροπέας HX711 παρατηρεί την αλλαγή του φορτίου και ενημερώνει τον μικροελεγχτή, το οποίο με τη χρήση κατάλληλου αλγορίθμου βρίσκει τη τιμή `max_weight` κάθε φορά. Τέλος ο μηδενισμός του `max_weight` γίνεται εντός του `interrupt`.

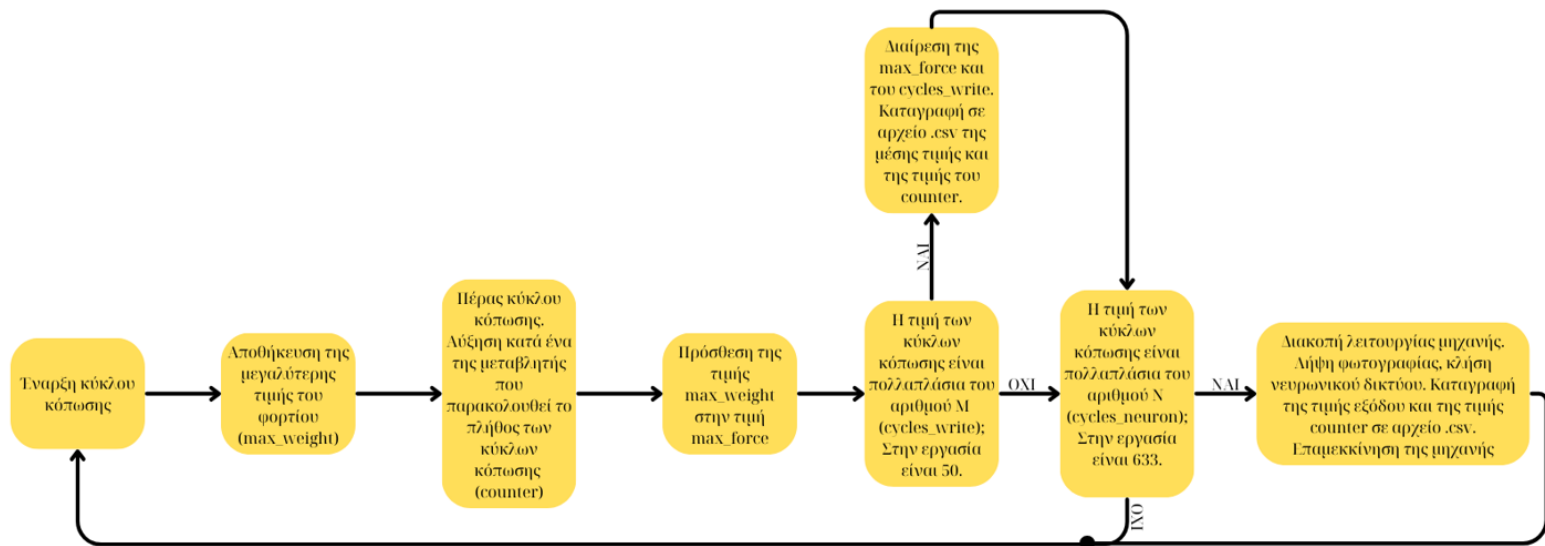
Βασικό στοιχείο για την ορθή λειτουργία του συστήματος είναι η ύπαρξη των δύο `if-statement`. Το πρώτο ελέγχει αν έχει συμπληρωθεί το πλήθος κύκλων για το οποίο γίνεται καταγραφή στο `.csv` αρχείο. Τότε ο κώδικας υπολογίζει τον μέσο όρο των μέγιστων καταγραφών και στέλνει στη σειριακή την τιμή αυτή μαζί με το πλήθος των κύκλων. Ύστερα ο κώδικας στη Python είναι σε αναμονή για να δεχθεί κάποια τιμή στη σειριακή από το Arduino, μόλις γίνει αυτό αποθηκεύει τη δυάδα των τιμών στο αρχείο `.csv`. Ο σκοπός του δεύτερου `if-statement` είναι όταν συμπληρωθεί το πλήθος κύκλων για το οποίο ο χρήστης επιθυμεί να γίνεται η χρήση του νευρωνικού και η καταγραφή του αποτελέσματος, να διακόπτει τη μηχανή μέσω του `ρελέ` και να στέλνει εντολή στην Python να κάνει χρήση του νευρωνικού. Ο κώδικας της Python με τη σειρά του κάνει μία λήψη όπως περιγράφηκε στην αντίστοιχη ενότητα. Ύστερα εισάγει εικόνα στο νευρωνικό και το συμπέρασμα για την υγεία του δοκιμίου μαζί με την τιμή εξόδου τα αποθηκεύει στην τρίτη και την τέταρτη στήλη του αρχείου `.csv`.

Τέλος στέλνει εντολή στον μικροελεγχτή μέσω της σειριακής για να συνεχίσει τη λειτουργία του.

Έτσι μέσω αυτών των δύο αλγορίθμων που λειτουργούν σε βρόγχο μπορεί να διασφαλιστεί η ορθή λειτουργία της πειραματικής διάταξης καθώς και η έγκυρη καταγραφή των αποτελεσμάτων.



Εικόνα 16 Ο μικροελεγχτής για τον έλεγχο της μηχανής



Εικόνα 17 Διάγραμμα ροής αλγορίθμου

5. Νευρωνικό Δίκτυο

5.1 Dataset (σύνολο δεδομένων)

Βασικό συστατικό για την υλοποίηση ενός νευρωνικού δικτύου είναι η ύπαρξη ενός αρκετά μεγάλου συνόλου δεδομένων (dataset) για την εκπαίδευση του. Παράλληλα οι πιθανές εφαρμογές της τεχνητής νοημοσύνης είναι ποικίλες. Έτσι είναι συχνό φαινόμενο ο μηχανικός να μην μπορεί να βρει ένα προκατασκευασμένο dataset για το εκάστοτε πρόβλημα που μελετάει ή το μέγεθος του dataset να μην είναι επαρκές για την εκπαίδευση του νευρωνικού δικτύου. Σε αυτή την περίπτωση καλείται να δημιουργήσει ένα.

Ωστόσο η δημιουργία ενός dataset δεν είναι απλό ζήτημα. Η εύρεση και η κατηγοριοποίηση των δεδομένων είναι μία διαδικασία που πολλές φορές υπόκεινται σε διάφορους τεχνοοικονομικούς περιορισμούς, όπως η έλλειψη αρκετών δειγμάτων στον φυσικό κόσμο ή η ανάγκη για πολλές εργατώρες ώστε να γίνει από κάποιον άνθρωπο η απαραίτητη κατηγοριοποίηση. Η λύση που προτείνεται είναι αύξηση ενός μικρού συνόλου δεδομένων με αυτοματοποιημένο τρόπο (data augmentation).

5.1.1 Καθορισμός αρχικού δείγματος

Η διαδικασία του καθορισμού των δεδομένων, η δημιουργία του dataset, η επαύξηση του και η διαδικασία του σχεδιασμού του νευρωνικού δικτύου είναι τα ζητήματα που θα αναπτυχθούν στο παρόν κεφάλαιο.

Η έρευνα για κάποιο δημόσιο dataset με φωτογραφίες μηχανικών αστοχιών δοκιμών κόπωσης σε πολύστρωτα σύνθετα υλικά αποδείχθηκε σύντομα άκαρπη. Έτσι δημιουργήθηκε η ανάγκη για την κατασκευή ενός dataset. Τα πρώτα δείγματα πάρθηκαν από το εργαστήριο και ήταν δοκίμια σύνθετων υλικών από προηγούμενα πειράματα. Πρέπει να σημειωθεί ότι στην πλειοψηφία τους δεν είχαν κάποια αστοχία.

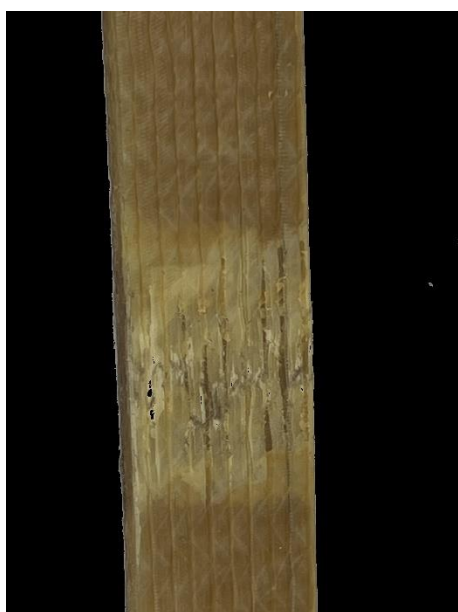
Η φωτογράφιση τους έγινε με την κάμερα του iPhone 13 Pro Max οποία έχει ανάλυση 12MP (4000X2800) εντός ειδικού κουτιού που είναι κατάλληλο για φωτογραφίες κοσμημάτων, έτσι επιτεύχθηκε ένας διπλός στόχος· οι φωτογραφίες τραβήχτηκαν με πολύ καλό φωτισμό και σε λευκό φόντο. Ο λόγος που ήταν αναγκαία η ύπαρξη λευκού φόντου θα γίνει ξεκάθαρος στην συνέχεια. Αρχικά φωτογραφήθηκαν τα υγιή δοκίμια και εν συνέχεια συκρατήθηκαν σε μία μέγγενη όπου με παλινδρομικές κινήσεις έσπασαν ώστε να αναπαρασταθεί όσο είναι δυνατόν καλύτερα η αστοχία που θα επέφερε η κόπωση στο υλικό. Οι Εικόνες 20 και 21 αποδεικνύουν ότι η πραγματικότητα με την προσομοίωση δεν διαφέρουν. Ακολούθησε η φωτογράφιση των πλέον μη υγιών δοκιμών. Εν συνέχεια η διαδικασία που θα παρουσιαστεί αφορά και τις δύο κατηγορίες φωτογραφιών.



Εικόνα 18 Τα δοκίμια που χρησιμοποιήθηκαν για την δημιουργία του dataset



Εικόνα 19 Το στάδιο της δημιουργίας ρωγμής στα δοκίμια



Εικόνα 20 Σπασμένο δοκίμιο στη μέγγενη



Εικόνα 21 Σπασμένο δοκίμιο ύστερα από κάμψη τριών σημείων

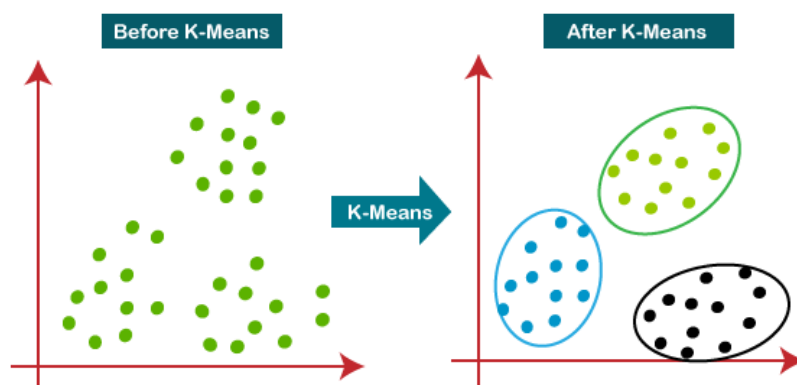
Η βέλτιστη εκπαίδευση του νευρωνικού μπορούσε να γίνει μόνο αν το αρχικό dataset ήταν επαρκώς καθαρό, δηλαδή αν είχε απομονωθεί όλο το περιττό φόντο από τη φωτογραφία. Ο λόγος για τον οποίο θέλουμε κάτι τέτοιο είναι ώστε το νευρωνικό δίκτυο να μην συσχετίσει την ύπαρξη κάποιου αντικειμένου ή διαφορετικού χρώματος με την ύπαρξη της ρωγμής, κατά τη διάρκεια της εκπαίδευσης ή των πειραμάτων. Αν γινόταν κάτι τέτοιο θα πετυχαίναμε ένα ικανοποιητικό accuracy με βάση αυτό το dataset, αλλά μόλις χρησιμοποιούταν πάνω στην μηχανή κόπωσης όπου και το φόντο αλλάζει δραματικά δεν θα παίρναμε τα ίδια αποτελέσματα.

Έτσι δημιουργήθηκε η ανάγκη να αποκοπεί η φωτογραφία από το περιβάλλον της. Για τον σκοπό αυτό υπάρχουν αρκετά διαθέσιμα προγράμματα τα οποία είναι συνήθως με πλήρωμή. Ενώ παράλληλα θα χρειαζόταν αρκετός χρόνος ώστε να αποκοπεί μία μία φωτογραφία. Κάτι τέτοιο θα ήταν αρκετά χρονοβόρο και θα έφερε μαζί του ένα περιττό κόστος. Έτσι αποφασίστηκε να χρησιμοποιηθεί η μέθοδος του k-means clustering για την αυτοματοποιημένη περικοπή τους.

Η μέθοδος k-means clustering είναι από τους απλούστερους τρόπους για μη επιβλεπόμενη μάθηση. Στόχος της είναι η κατηγοροποίηση των δεδομένων σε k ομάδες (clusters). Η μέθοδος εκκινεί με την επίλογη του πλήθους των ομάδων και την επιλογή k τυχαίων σημείων στον χώρο. Κατόπιν η εικόνα που έχουμε είναι X_n σημεία δεδομένων και C_k κέντρα ομάδων. Εν συνεχεία ακολουθείται η παρακάτω διαδικασία ωστόσο να έχουν σαρωθεί όλα τα σημεία δεδομένων. Για κάθε σημείο υπολογίζεται η ευκλείδεια απόσταση μεταξύ του κάθε σημείου και των κέντρων με τη γνωστή σχέση.

$$d(x, y) = \sqrt{(x_i - C_j)^2}$$

Κατόπιν το κάθε σημείο κατατάσσεται στο κέντρο με το οποίο είχε τη μικρότερη απόσταση. Ωστόσο τα κέντρα επιλέχθηκαν τυχαία όταν ξεκίνησε η διαδικασία και αν μείνουν έτσι τα αποτελέσματα που θα προκύψουν θα είναι προφανώς λανθασμένα. Για τον λόγο αυτό είναι αναγκαίο αφού ολοκληρώνεται μία κατηγοροποίηση να γίνεται επανυπολογισμός του κέντρου. Το νέο κέντρο είναι ο μέσος όρος όλων των σημείων της ομάδας.

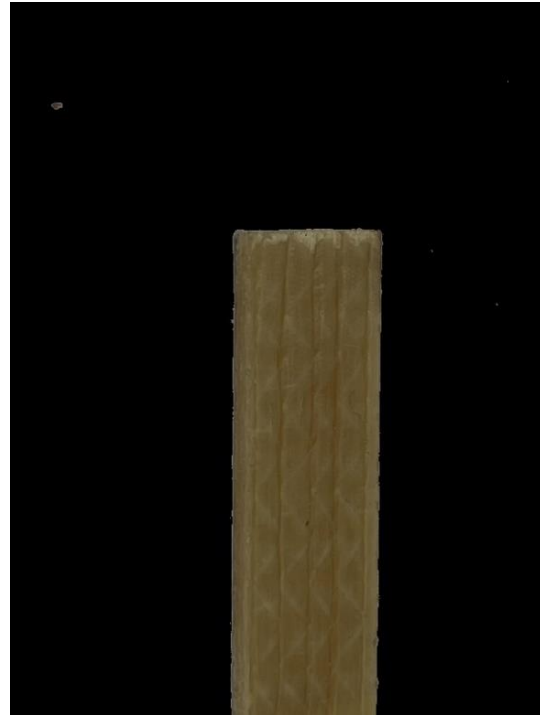


Εικόνα 22 Αναπαράσταση του k-means clustering (Πηγή ιστοσελίδα: <https://av-eks-blogoptimized.s3.amazonaws.com/46668k-means-clustering-algorithm-in-machine-learning.png>)

Σε αυτό το σημείο θα εξηγηθεί ποια ήταν η αναγκαιότητα για την ύπαρξη λευκού φόντου και πώς αυτή σχετίζεται με την περικοπή. Κάθε φωτογραφία που αποθηκεύεται σε ψηφιακή μορφή αποτελείται από τα τρία κανάλια RGB (Κόκκινο-Πράσινο-Μπλε) και με την χρήση αυτών των 3 καναλιών κάθε φωτογραφία μπορεί να πάρει όποιο χρώμα χρειάζεται. Κάθε κανάλι αποτελείται από ένα byte και άρα μπορεί να λάβει τιμές από το 0 ως και το 255, με το 0 να είναι η ανυπαρξία του εκάστοτε χρώματος. Το λευκό χρώμα δημιουργείται όταν και τρία κανάλια λαμβάνουν τις μέγιστες τιμές τους, άρα σε έναν τρισδιάστατο χώρο θα ήταν το σημείο (255,255,255) και όσα σημεία βρίσκονται κοντά σε αυτό είναι πολύ κοντά στο λευκό. Επομένως δεδομένου ότι τα δοκίμια στο μεγαλύτερο μέρος τους έχουν διάφορες αποχρώσεις του καφέ είναι αρκετά εύκολο με τη χρήση μίας μεθόδου k-clustering να εντοπιστούν τα pixel που ανήκουν στον φόντο (είναι λευκά) και κατόπιν αυτά τα pixel να λάβουν τις τιμές να λαβουν την τιμή (0,0,0). Έτσι με την ανασύνθεση της φωτογραφίας θα έχει απομονωθεί μόνο το δοκίμιο.



Εικόνα 23 Δοκίμιο με αστοχία ύστερα από την αποκοπή της εικόνας



Εικόνα 24 Υγιές δοκίμιο ύστερα από την αποκοπή της εικόνας

Από την διαδικασία που περιγράφηκε έχουν προκύψει 73 φωτογραφίες μη υγιών δοκιμίων και 67 υγιών. Ένα σύνολο το οποίο είναι ανεπαρκές για την εκπαίδευση ενός νευρωνικού δικτύου. Παρακάτω θα παρουσιαστεί η μέθοδος για την επαύξηση του συνόλου δεδομένων.

5.1.2 Μέθοδος επαύξησης συνόλου δεδομένων (Data augmentation)

Οι κύριοι λόγοι για τους οποίους χρειάζεται η μέθοδος της αύξησης των δεδομένων είναι η αποτροπή της υπερμοντελοποίησης (overfitting) στο δίκτυο, δημιουργώντας παρόμοια δεδομένα τα οποία διατηρούν την αρχική ιδέα και την κάνουν πιο ξεκάθαρη για το δίκτυο. Αυτό με τη σειρά του μπορεί να βελτιώσει την ακρίβεια του νευρωνικού. Ένας δεύτερος λόγος είναι η σημαντική μείωση κόστους για τη δημιουργία του dataset όπως έχει ήδη αναφερθεί παραπάνω.

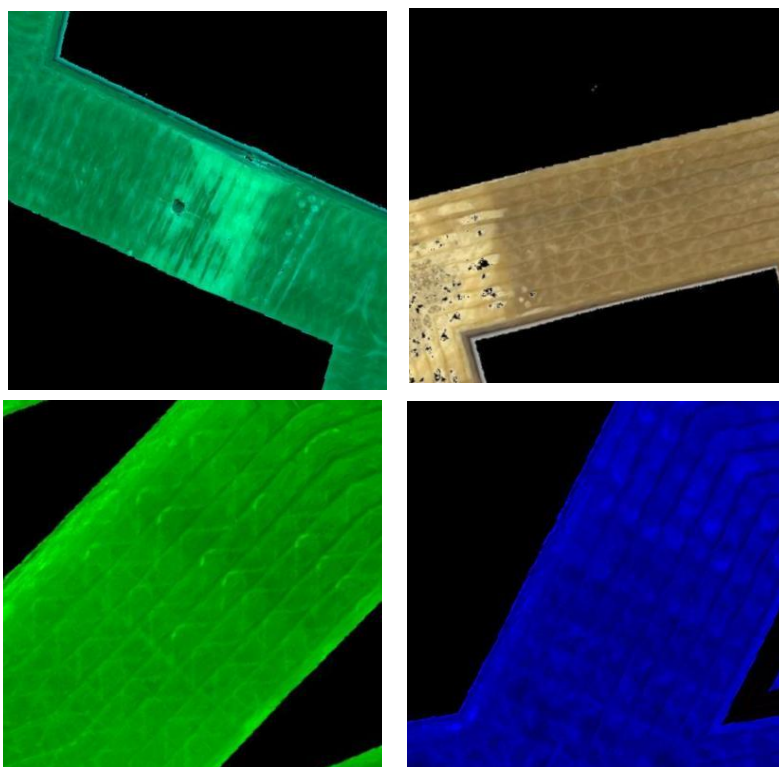
Οι πιο συνηθισμένες μέθοδοι για data augmentation στις εικόνες θα αναφερθούν παρακάτω. Αρχικά είναι οι γεωμετρικές αλλαγές οι οποίες μπορούν να γίνουν σε μία εικόνα όπως περιστροφή, αποκοπή, εστίαση και επιμήκυνση. Οι αλλαγές αυτές ωστόσο πρέπει να γίνονται με μεγάλη προσοχή καθώς μπορούν να αλλοιώσουν το δείγμα και κατ' επέκταση την απόδοση του νευρωνικού δικτύου. Παράλληλα υπάρχει η μέθοδος της αλλαγής των χρωμάτων κάτι τέτοιο μπορεί να γίνει αναμιγνύοντας τα χρωματικά κανάλια για να προκύψουν νέες εικόνες. Επίσης μπορεί να αυξομειωθεί η αντίθεση και η φωτεινότητα στις εικόνες. Η μέθοδος των φίλτρων kernel είναι αρκετά διαδεδομένη και βασίζεται στη σάρωση της εικόνα με τους γνωστούς πίνακες kernel, οι οποίοι είναι μικροί σε μέγεθος (π.χ. 2X2, 3X3), ώστε να μετάβαλλουν την οξύτητα στα σημεία της εικόνας που υπάρχει αλλαγή χρώματος και αυτές οι αλλαγές να γίνουν εντονότερες. Προφανώς μπορεί να ισχύσει το αντίθετο θολώνοντας τις φωτογραφίες και ανακατεύοντας κατ' αυτό τον τρόπο τα χρώματα. Πρέπει να σημειωθεί το πόσο σημαντικό είναι το μέγεθος των πινάκων kernel να διατηρείται σε λογικά επίπεδα έτσι ώστε να μην υπάρξει μεγάλη αλλοίωση της πληροφορίας. Οι τελευταίες βασικές

μέθοδοι για την επέκταση του συνόλου των δεδομένων είναι η η διαγραφή ορισμένων κομματιών της φωτογραφίας ή η τυχαία ανάμειξη τους.

Περισσότερο προχωρημένες μέθοδοι για την επαύξηση του συνόλου των δεδομένων σε εικόνες είναι τα GANs (Generative Adversarial Networks), τα οποία δεν απαιτούν την ύπαρξη πραγματικών δεδομένων για τη δημιουργία συνθετικών. Εξίσου σημαντικά είναι τα Neural Transfer Networks τα οποία χρησιμοποιούν μία σειρά επιπέδων συνέλιξης με σκοπό να διαχωρίσουν την τεχνοτροπία από την κεντρική ιδέα. [29]

5.1.3 Εφαρμογή

Η λειτουργία του κώδικα που θα περιγραφεί και αφορά το data augmentation βρίσκεται στο παράρτημα 7.1 . Αρχικά έγινε μία σμίκρυνση των εικόνων και μία βαθμονόμηση τους έτσι ώστε να επιτευχθεί υπολογιστική οικονομία και να μην αλλοιωθεί η πληροφορία. Κατόπιν κάθε εικόνα περιστράφηκε περίξ των δύο αξόνων και τέλος ακολούθησαν τυχαίες περιστροφές κατά 0.4 rad γύρω από τον κάθετο άξονα. Σειρά έχει ο διαχωρισμός τους στα τρία χρωματικά κανάλια και η ανάμειξη όλων των χρωματικών καναλιών μεταξύ τους. Το τελευταίο βήμα είναι το τυχαίο ανακάτεμα της κάθε εικόνας όπως περιγράφηκε πιο πάνω. Ακολουθώντας αυτή τη μεθοδολογία έγινε δυνατό το αρχικό δείγμα των υγιών φωτογραφιών να φτάσει τις 2816, ενώ των μη υγιών τις 2856. Εν κατακλείδι με αυτή τη μεθοδολογία γίνεται να πραγματοποιηθεί η εκπαίδευση του νευρωνικού δικτύου με 5672 φωτογραφίες.



Εικόνα 25 Εικόνες δοκιμών ύστερα από την επαύξηση των δεδομένων

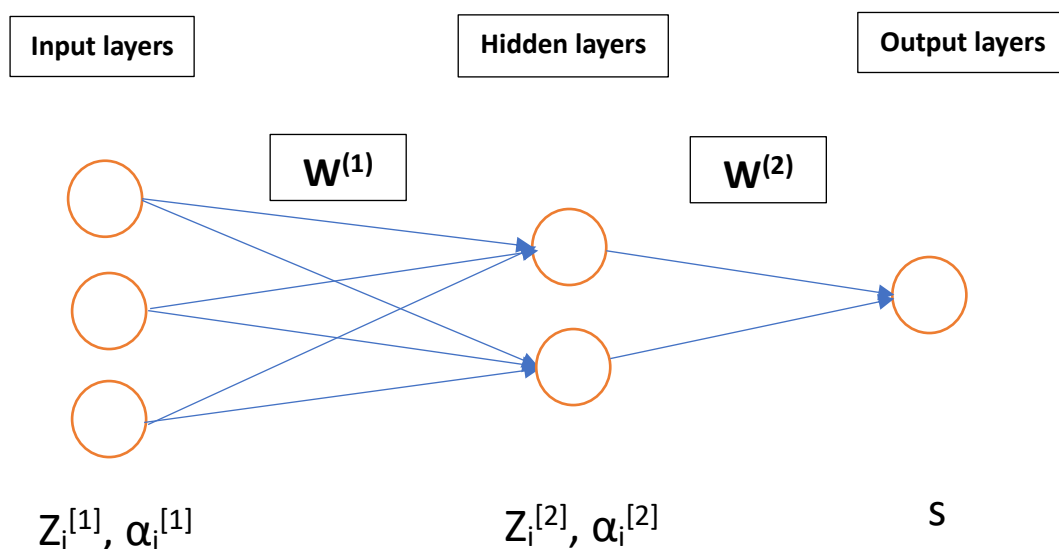
5.2 Αρχιτεκτονική νευρωνικού δικτύου και εκπαίδευση

5.2.1 Εισαγωγή

Η κατηγοροποίηση της ύπαρξης ή ανυπαρξίας ρωγμής στο δοκίμιο έγινε με τη βοήθεια νευρωνικών δικτύων. Στη συνέχεια θα γίνει μία συνοπτική παρουσίαση του τρόπου λειτουργίας ενός δικτύου καθώς και διαφόρων σημαντικών παραγόντων που καθορίζουν τη λειτουργία του. Τέλος θα παρουσιαστεί η αρχιτεκτονική του νευρωνικού δικτύου που δημιουργήθηκε για τις ανάγκες της παρούσας εργασίας.

5.2.2 Τρόπος λειτουργίας νευρωνικών δικτύων

Η αρχή λειτουργίας του νευρωνικού δικτύου βασίζεται στον ανθρώπινο εγκέφαλο και στον τρόπο με τον οποίο αυτός αναλύει τα ερεθίσματα που δέχεται. Βασικό στοιχείο κάθε νευρωνικού δικτύου είναι ο κόμβος (ή νευρώνας ή node) ο ρόλος του οποίου είναι να δέχεται πληροφορία από τους συνδεδεμένους με αυτόν κόμβους να την επεξεργάζεται και να την μεταφέρει στα επόμενα στρώματα του δικτύου (layers) ως μοναδική έξοδο.



Το ανωτέρω σχήμα δείχνει τη διάταξη και τα βασικά χαρακτηριστικά ενός απλού νευρωνικού δικτύου. Κάθε δίκτυο αποτελείται από ένα στρώμα εισόδου (input layer) μέσω του οποίου εισάγονται τα δεδομένα. Κατόπιν υπάρχουν τα ενδιάμεσα ή κρυφά στρώματα (hidden layers) τα οποία αναλαμβάνουν την επεξεργασία του μεγαλύτερου όγκου της πληροφορίας. Τέλος είναι το στρώμα εξόδου (output layer) το οποίο δίνει και το τελικό αποτέλεσμα.

Κάθε κόμβος περιέχει μία συνάρτηση ενεργοποίησης, με βάση αυτή και την είσοδο $Z_i^{[l]}$ που δέχεται προκύπτει η έξοδος $\alpha_i^{[l]}$. Ανάλυση των συναρτήσεων ενεργοποίησης θα γίνει αργότερα. Η τιμή της εισόδου του κάθε κόμβου προκύπτει από τη σχέση:

$$Z_i^{[j]} = \sum_{i=0}^k (W_k^{(j-1)} \cdot a_k^{[j-1]}) + b_i$$

$W_k^{(j-1)}$, είναι τα βάρη του νευρωνικού δικτύου

b_i , είναι ο συντελεστής bias

Στόχος της διαδικασίας εκπαίδευσης του δικτύου είναι η κατάλληλη τροποποίηση του συντελεστή bias και των βαρών έτσι ώστε να επιτευχθεί το μικρότερο δυνατό σφάλμα για τα αποτελέσματα του δικτύου. Η τιμή $Z_i^{[j]}$ είναι η είσοδος της συνάρτησης ενεργοποίησης για την οποία ισχύει η σχέση $a_k^{[j]} = f(Z_i^{[j]})$.

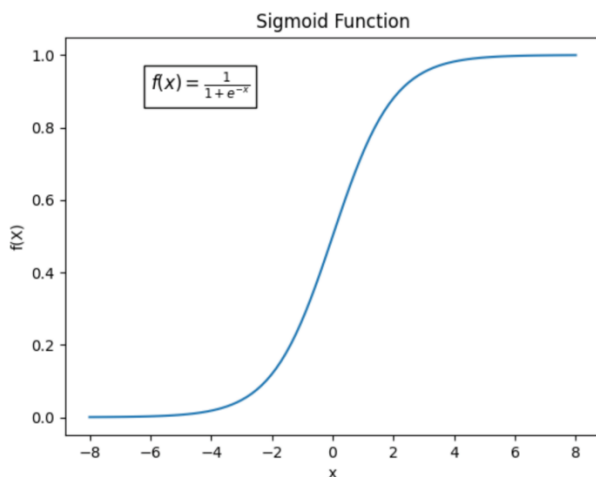
Συναρτήσεις ενεργοποίησης

Βασικό συστατικό ενός νευρωνικού δικτύου είναι η συνάρτηση ενεργοποίησης. Οι συναρτήσεις ενεργοποίησης πρέπει να έχουν ορισμένα χαρακτηριστικά έτσι ώστε να επιτευχθεί η εκπαίδευση του νευρωνικού δικτύου. Απαραίτητη προϋπόθεση είναι η συνάρτηση ενεργοποίησης να είναι μη-γραμμική, διαφορετικά τα δεδομένα θα περνούσαν μέσα από τους κόμβους χωρίς να επεξεργαστούν καθόλου. Δεύτερη πολύ σημαντική ιδιότητα είναι η συνάρτηση ενεργοποίησης να είναι συνεχώς παραγωγίσιμη σε όλο το εύρος της. Διαφορετικά οι μέθοδοι βελτιστοποίησης οι οποίες βασίζονται στην παραγωγή δεν θα είναι σε θέση να κάνουν καμία ουσιαστική βελτίωση. Παράλληλα καίρια είναι η επίδραση του εύρους της συνάρτησης και υπάρχουν δύο εξίσου αποδεκτές προσεγγίσεις. Αν το εύρος της συνάρτησης είναι πεπερασμένο τότε το μοντέλο τείνει να είναι σταθερότερο, αφού τα μοτίβο με τη σειρά τους θα μπορούν να επηρεάσουν ένα πεπερασμένο πλήθος βαρών. Στην αντίθετη περίπτωση του μη πεπερασμένου εύρους το μοντέλο μπορεί να γίνει αποδοτικότερο καθώς στη διάρκεια της εκπαίδευσης τα περισσότερα βάρη θα έχουν επηρεαστεί από το δείγμα των δεδομένων. Συγχρόνως η συνάρτηση ενεργοποίησης πρέπει να είναι μονότονη και για λόγους ευκολίας που σχετίζονται με τις τιμές εκκινήσεις των βαρών πρέπει να τείνει στην ταυτοτική, κοντά στο μηδέν. [30]

Υπάρχουν αρκετές συναρτήσεις ενεργοποίησης και η κάθε μία έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Έτσι ο κάθε σχεδιαστής καλείται να αξιολογήσει τη φύση του προβλήματος και να επιλέξει. Παρακάτω θα παρουσιαστούν οι πιο γνωστές.

➤ Σιγμοειδής συνάρτηση

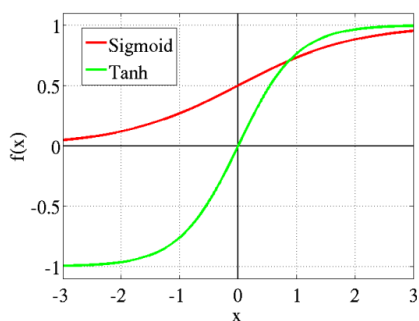
Η σιγμοειδής δίνεται από τον τύπο $S(t) = \frac{1}{1+e^{-t}}$. Το όνομα της οφείλεται στο χαρακτηριστικό της σχήμα που μοιάζει με το γράμμα S. Βασική ιδιότητα της είναι ότι μπορεί να λάβει τιμές μόνο μεταξύ του 0 και του 1, κάτι το οποίο είναι ιδιαίτερο βοηθητικό σε περιπτώσεις δυαδικής ταξινόμησης. Ωστόσο σε άλλα προβλήματα αυτός ο μεγάλος περιορισμός στο εύρος είναι κάτι το οποίο μπορεί να λειτουργήσει ανασταλτικά για την απόδοση του δικτύου. Τέλος πρέπει να δωθεί ιδιαίτερη προσοχή στο γεγονός ότι η παράγωγος της σιγμοειδούς τείνει να λάβει πολύ μικρές τιμές στα άκρα της, με αποτέλεσμα ο αλγόριθμος βελτιστοποίησης να μην είναι σε θέση να προσαρμόσει επαρκώς τα βάρη και τα bias. [32]



Εικόνα 26 Η σιγμοειδής συνάρτηση (Πηγή ιστοσελίδα: <https://www.codecademy.com/resources/docs/ai/neural-networks/sigmoid-activation-function>)

➤ Συνάρτηση tanh

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



Η συνάρτηση tanh είναι παρόμοια με τη σιγμοειδή ωστόσο το γεγονός ότι κυμαίνεται από -1 ως 1 μπορεί να την καταστήσει αρκετά καλύτερη. Αυτό γιατί αν οι εισοδοι είναι αρνητικοί τότε η έξοδος θα είναι εντόνως αρνητική.

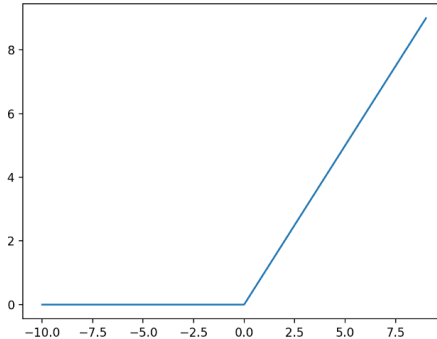
Εικόνα 27 Η συνάρτηση tanh (Πηγή ιστοσελίδα: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>)

➤ Συνάρτηση ReLU

Η συνάρτηση ReLU έχει την ιδιότητα ότι αν η είσοδος είναι αρνητική η έξοδος θα είναι μηδέν, ενώ αν η είσοδος είναι θετική τότε η συνάρτηση γίνεται ταυτοτική. Η γενική σχέση είναι η $\text{Relu}(x) = \max(0, x)$. Βασικό πλεονέκτημα της ReLU (και των παραλλαγών της) είναι ότι είναι ιδιαίτερα απλή στον υπολογισμό της και κατ' επέκταση η διαδικασία εκπαίδευσης μπορεί να επιταχυνθεί σημαντικά.

Παράλληλα πολύ σημαντικό της πλεονέκτημα είναι το sparsity (sparse είναι ένα μητρώο του οποίου τα περισσότερα στοιχεία είναι μηδενικά) εξαιτίας της ιδιότητας της να μηδενίζει στοιχεία. Έτσι τα μοντέλα είναι πιο συνεκτικά με αποτέλεσμα να βελτιώνεται η προγνωστική του ικανότητα και να μειώνεται η υπερμοντελοποίηση. Εξίσου σημαντικό χαρακτηριστικό της είναι ότι δεν αντιμετωπίζει το πρόβλημα της εξαφάνισης της παραγώγου (vanishing gradient problem) εν αντιθέσει με τις

συναρτήσεις tanh και τη σιγμοειδή αφού δεν έχει σημεία κορεσμού. Η ReLU όμως αντιμετωπίζει ένα αρκετά διαφορετικό πρόβλημα χάρη στην ιδιότητα της να μηδενίζει τις τιμές του αρνητικού ημίξονα, μπορεί να οδηγήσει ολόκληρους κόμβους να νεκρώσουν αν αυτοί λαμβάνουν αρνητικές τιμές (γνωστό και ως dying ReLU).[36]



Εικόνα 28 Η συνάρτηση Relu (Πηγή ιστοσελίδα: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>)

➤ Leaky ReLU

Η leaky ReLU έρχεται για να επιλύσει το πρόβλημα της dying ReLU με μία πολύ μικρή τροποποίηση. Πλέον οι αρνητικές εισοδοι δεν μηδενίζονται αλλά γίνονται πολύ μικρές. Έτσι η τελικά σχέση είναι $f(x)=\max(0.01x,x)$. Τέλος πρέπει να σημειωθεί ότι καμία από τις δύο μεθόδους δεν είναι σε θέση να εξαφανίσει το πρόβλημα της εκθετικής παραγωγού (exploding gradient problem) κάτι το οποίο ταλαιπωρεί πολύ τη συγκεκριμένη συνάρτηση ενεργοποίησης.

Συνάρτηση κόστους και αλγόριθμοι βελτιστοποίησης

Ένα σημαντικό σκέλος των νευρωνικών δικτύων το οποίο πρέπει να αναλυθεί πριν περιγραφεί ο τρόπος με τον οποίο εκπαιδεύεται το δίκτυο είναι η συνάρτηση κόστους (cost function). Ο ρόλος της είναι να ποσοτικοποιήσει το πόσο μακριά είναι η πρόβλεψη (S) από την πραγματική τιμή. Στόχος είναι αυτή η διάφορα να ελαχιστοποιηθεί. Υπάρχουν διάφοροι τύποι συναρτήσεων κόστους εδώ παρουσιάζονται μόνο δύο. [33],[34]

➤ Μέσο τετραγωνικό σφάλμα

$MSE = \frac{1}{n} \sum (y - \hat{y})^2$, όπου n είναι το πλήθος των δειγμάτων στο σύνολο δεδομένων. Το MSE χρησιμοποιείται κυρίως σε προβλήματα παλινδρόμησης.

➤ Binary cross-entropy (BCE)

Η μέθοδος BCE χρησιμοποιείται κυρίως σε προβλήματα δυαδικής ταξινόμησης.

$$BCE = -\frac{1}{n} \sum y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})$$

Το τελευταίο ζήτημα που μένει να εξηγηθεί για τα νευρωνικά δίκτυα είναι ο μηχανισμός με τον οποίο γίνεται η ανανέωση των βαρών και των bias. Η διαδικασία κατά την οποία το δίκτυο κινείται από την είσοδο προς την έξοδο και εξάγει αποτέλεσμα, ονομάζεται εμπρόσθια διάδοση (forward propagation). Κατόπιν ακολουθεί ο αλγόριθμος οπισθοδιάδοσης (back propagation) με τη χρήση του οποίου γίνεται η ανανέωση των σταθερών. Ισχύει η γενική

σχέση $w = w - a * \frac{\partial C}{\partial w}$, είναι εμφανές ότι η ανανέωση κάθε τιμής γίνεται με τη λογική πόσο επηρεάζει το εκάστοτε βάρος τη συνάρτηση κόστους. Η σταθερά a μετριάζει το πως θα κινηθεί τελικά η νέα τιμή του βάρους και ονομάζεται ρυθμός εκμάθησης (learning rate). Τέλος ο υπολογισμός της παραγώγου $\frac{\partial C}{\partial w}$ γίνεται με τον κανόνα της αλυσίδας. Η διαδικασία βελτιστοποίησης με βάση τη μέθοδο κλίσης κατάβασης (gradient descent) είναι από τις πιο συνηθισμένες, ωστόσο παρουσιάζει κάποιες αδυναμίες.

Μία εξαιρετική εναλλακτική είναι ο αλγόριθμος βελτιστοποίησης Adam. Ο αλγόριθμος Adam είναι ο συνδυασμός δύο άλλων μεθόδων βελτιστοποίησης: της μεθόδου κλίσης κατάβασης με ορμή (gradient descent with momentum) και του αλγόριθμου RMS prop. Η κλασική gradient descent έχει το εξής μειονέκτημα, εξαρτάται αποκλειστικά από την κλίση του παρόντος βήματος. Αποτέλεσμα αυτού είναι ότι αν το σημείο βρεθεί σε περιοχή με μικρή κλίση η σύγκλιση του αλγορίθμου θα γίνει πολύ αργά. Παράλληλα σε αυτή τη περίπτωση η αύξηση του ρυθμού εκμάθησης δεν ενδείκνυται διότι η μέθοδος αυτή χαρακτηρίζεται από σημαντικές ταλαντώσεις και η ενέργεια αυτή μπορεί να οδηγήσει σε αστάθεια όλου του συστήματος. Έτσι εισάγεται η παραλλαγή gradient descent with momentum η οποία αξιοποιεί τις κλίσεις προηγούμενων βημάτων για να υπερβεί περιοχές μικρής παραγώγου και να υπάρξει ταχύτερη σύγκλιση. Ο τύπος είναι:

$$v(t) = (1 - \beta) \cdot v(t - 1) + (1 - \beta) \delta(t)$$

$v(t)$: είναι το βάρος του βήματος t

β : είναι η σταθερά ορμής και λαμβάνει τιμές από 0 ως 1. Μεγαλύτερες τιμές της σταθεράς δίνουν μεγαλύτερο βάρος σε παρελθοντικές τιμές.

$\delta(t)$: είναι η παράγωγος κόστους ως προς το βάρος του βήματος t .

Εναλλακτικά υπάρχει και η γενική σχέση:

$$v(n) = (1 - \beta) \sum_{t=1}^n \beta^{n-t} \cdot \delta(t)$$

Το γενικότερο πρόβλημα των αλγορίθμων κατάβασης είναι οι ταλαντώσεις που εμφανίζονται και μπορούν να οδηγήσουν σε αστάθεια, έτσι προτάθηκε η χρήση της μεθόδου RMS prop. Οι σχέσεις της μεθόδου είναι

$$W = W - \alpha \frac{1}{\sqrt{S_{dw} + \epsilon}} \frac{dC}{dW} S_{dw} = \beta \quad \text{και} \quad S_{dw} = \beta \cdot S_{dw,previous} + (1 - \beta) \cdot \delta(t)^2$$

S_{dw} : είναι ο κινητός μέσος όρος της παραγώγου υψομένος στο τετράγωνο.

Παρατηρώντας τις σχέσεις είναι εμφανές ότι η συγκεκριμένη μέθοδος *τιμωρεί* τις μεγάλες τιμές της παραγώγου ενώ *επιβραβεύει* τις μικρές, με αυτό τον τρόπο περιορίζονται σημαντικά οι ταλαντώσεις της συνάρτησης κόστους και δίνεται η δυνατότητα για χρήση μεγαλύτερων τιμών για τον βαθμό εκμάθησης. Τέλος πρέπει να σημειωθεί ότι η τιμή ϵ υπάρχει ώστε να διασφαλιστεί πως δεν θα υπάρξει μηδενισμός του παρονομαστή και λαμβάνει τιμές της τάξεως του 10^{-8} .

Έχοντας εικόνα των δύο προηγούμενων μεθόδων και γνωρίζοντας τον σκοπό που εξυπηρετεί η κάθε μία γίνεται να παρουσιαστεί η μέθοδος Adam ως ένα υβρίδιο τους. Η μέθοδος αναπτύχθηκε από τους Diederik P. Kingma and Jimmy Ba το 2014. Αρχικά πρέπει να οριστούν τα m_t , u_t τα οποία συμβολίζουν τον σταθμισμένο μέσο όρο των προηγούμενων βημάτων και της παρούσης παραγώγου. Η τιμή εκκίνησης τους είναι το μηδέν και δίνονται από τους τύπους:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \delta(t) \quad \text{και} \quad u_t = \beta_2 u_{t-1} + (1 - \beta_2) \delta(t)^2$$

Είναι σημαντικό να σημειωθεί ότι επειδή οι m_t, u_t εκκινούν από το μηδέν έχει ως αποτέλεσμα να έχουν μία τάση προς το μηδέν, ειδικά για τα πρώτα βήματα. Έτσι για να λυθεί το πρόβλημα χρησιμοποιούνται οι παρακάτω διορθώσεις:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

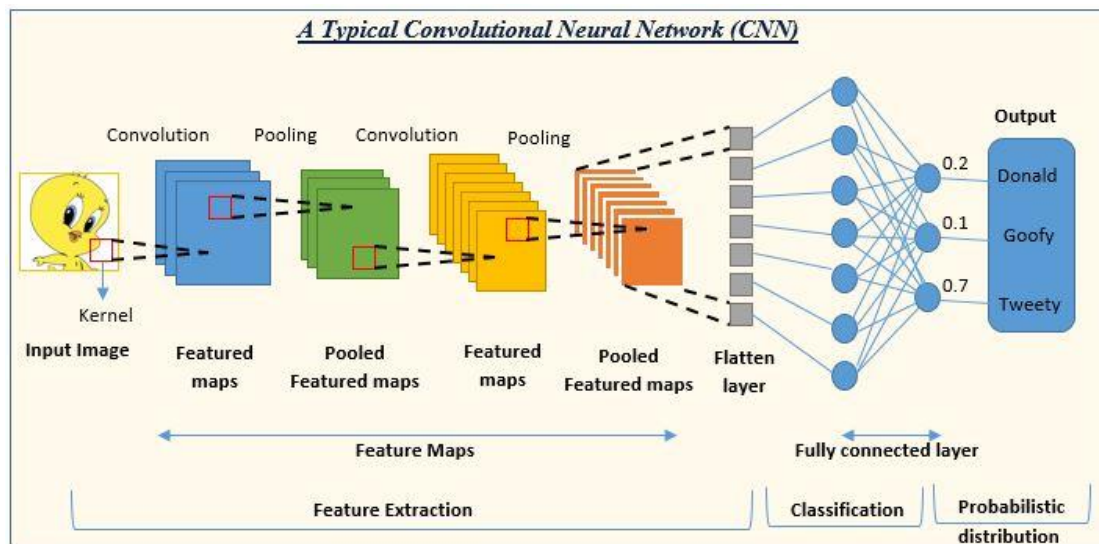
Τέλος η σχέση ανανέωσης είναι:

$$W_t = W_{t-1} - a \cdot \frac{\hat{m}_t}{\sqrt{\hat{u}_t + \epsilon}}$$

Οι μεθοδολογίες που παρουσιάστηκαν για τη βελτιστοποίηση του βάρους ισχύουν και για τις σταθερές bias του δικτύου.

5.3 Δίκτυα συνέλιξης

Τα κλασικά νευρωνικά δίκτυα εκπαιδεύονται με τα δεδομένα να εισέρχονται κάθε φορά στο δίκτυο σειριακά. Αυτό είναι βολικό για ορισμένα προβλήματα, όπως κείμενο ή ιατρικές διαγνώσεις, ωστόσο υπολείπεται σημαντικά όταν το αντικείμενο ενδιαφέροντος του νευρωνικού είναι εικόνες. Αν η εικόνα αναπαρασταθεί στο νευρωνικό ως μία σειρά αριθμών είναι εμφανές ότι χάνεται σημαντική πληροφορία από την ενέργεια αυτή. Από αυτή την ανάγκη δημιουργήθηκαν τα Νευρωνικά Δίκτυα Συνέλιξης (Convolutional Neural Networks-CNN).



Εικόνα 29 Αναπαράσταση ενός δικτύου CNN (Πηγή ιστοσελίδα:

<https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/>)

Τα νευρωνικά δίκτυα αυτού του τύπου έχουν την ικανότητα να αντιληφθούν καλύτερα τον χώρο γύρω τους με τη βοήθεια των φίλτρων kernel. Τα φίλτρα kernel είναι μητρώα μικρών διαστάσεων τα οποία σαρώνουν τη φωτογραφία με σκοπό να αναδείξουν ορισμένα γενικά

χαρακτηριστικά της, στα πρώτα hidden layers (π.χ. άκρες στη φωτογραφία) ενώ στα επόμενα hidden layers μπορούν να εμφανίσουν πιο εκλεπτυσμένα χαρακτηριστικά όπως περιγράμματα αντικειμένων. Αφού ολοκληρωθεί το βήμα της συνέλιξης ακολουθεί η εφαρμογή της συνάρτησης ενεργοποίησης στο μητρώο και το βήμα του Pooling. Η διαδικασία Pooling είναι αρκετά παρόμοια με τη συνέλιξη, πάλι γίνεται σάρωση του μητρώου αλλά με σκοπό να μειωθεί το δείγμα μειώνοντας την υπολογιστική δύναμη και κάνοντας το περιεχόμενο της εικόνας πιο αφηρημένο. Το τελευταίο χαρακτηριστικό ειδικά, είναι πολύ βασικό για τα νευρωνικά δίκτυα. Υπάρχουν διάφοροι τύποι pooling, όπως max pooling, average pooling κ.α. Τέλος για να μπορεί να γίνει σωστή κατηγοροποίηση δύναται να τοποθετηθεί και ένα κλασσικό στρώμα νευρωνικού δικτύου που θα κάνει το τελικό μητρώο σειριακό.

5.4 Κατασκευή δικτύων αναγνώρισης

5.4.1 Συνελκτικό δίκτυο

Το δίκτυο το οποίο κατασκευάστηκε για να καλυφθούν οι ανάγκες της παρούσας διπλωματικής εργασίας είναι ένα δίκτυο συνέλιξης. Η είσοδος του δικτύου είναι μία φωτογραφία διαστάσεων 200X200, όπως αυτή προέκυψε από τη διαδικασία της σμίκρυνσης που αναφερθηκε προηγουμένως. Τα συνολικά στρώματα του δικτύου είναι 4 με κάθε στρώμα συνέλιξης να το διαδέχεται ένα στρώμα pooling. Το πρώτο στρώμα συνέλιξης περιέχει 64 φίλτρα διαστάσεων 3X3, ενώ στα υπόλοιπα το πλήθος των φίλτρων είναι 256. Η συνάρτηση ενεργοποίησης που επιλέχθηκε είναι η ReLU, καθώς έχει αποδειχτεί ότι εμφανίζει πολύ καλή συμπεριφορά τόσο από πλευρά σύγκλισης όσο και από από πλευρά υπολογισμού, εξαιτίας της απλότητας της. Ο τύπος pooling που χρησιμοποιήθηκε είναι ο max pooling διαστάσεων 2X2 ο οποίος σαρώνει το μητρώο και κρατάει τη μέγιστη τιμή μόνο.

Ένα πολύ σημαντικό χαρακτηριστικό του δικτύου το οποίο επηρεάζει σημαντικά την απόδοση του είναι το dropout layer το οποίο ακολουθεί το layer που μετατρέπει το τελικό μητρώο σε σειριακή αλληλουχία. Ο ρόλος ενός dropout layer είναι να προστατέψει το δίκτυο από το να αντιληφθεί τον στατιστικό θόρυβο που υπάρχει πάντα σε κάθε σύνολο δεδομένων ως χαρακτηριστικό του συνόλου (πρόβλημα υπερμοντελοποίησης), έτσι το δίκτυο θα έχει πολύ καλή ακρίβεια για το σύνολο με το οποίο έγινε η εκπαίδευση αλλά θα αποδίδει πολύ φτωχά όταν χρειαστεί να κάνει κάποια πρόβλεψη εκτός του συνόλου αυτού. Η λειτουργία του είναι ότι σε κάθε διαδικασία εμπρόσθιας διάδοσης απενεργοποιεί τυχαία ένα ποσοστό των κόμβων που υπάρχει στο συγκεκριμένο στρώμα. Έτσι περιορίζει τυχαία την πληροφορία που χρησιμοποιείται για την εξαγωγή των αποτελεσμάτων. Στο συγκεκριμένο δίκτυο επιλέχθηκε ένα ποσοστό dropout της τάξης του 40%. Τέλος το στρώμα εξόδου αποτελείται από ένα κλασσικό κόμβο με συνάρτηση ενεργοποίησης τη σιγμοειδή, δίνοντας έτσι μία έξοδο μεταξύ του 0 και του 1. Ο λόγος που επιλέχθηκε αυτή η συνάρτηση είναι γιατί το πρόβλημα είναι δυαδικής κατηγοροποίησης.

Το σύνολο δεδομένων χωρίστηκε σε μία αναλογία 80-20 για εκπαίδευση και έλεγχο του νευρωνικού. Ο διαχωρισμός του συνόλου έγινε με συγκεκριμένο τρόπο ώστε τα δύο υποσύνολα να είναι κάθε φορά τα ίδια (βλ. Παράρτημα νευρωνικού – seed=2024). Ο αλγόριθμος βελτιστοποίησης που επιλέχθηκε είναι ο Adam λόγω της πολύ καλής συμπεριφοράς του. Ενώ για συνάρτηση κόστους χρησιμοποιήθηκε η συνάρτηση binary crossentropy που είναι κατάλληλα για προβλήματα δυαδικής ταξινόμησης. Καίριο χαρακτηριστικό για την εκπαίδευση του δικτύου είναι το μέγεθος του batch. Μέσω αυτού ελέγχεται το πλήθος των δειγμάτων που θα περάσουν μέσα από το δίκτυο πρίν γίνει η

ανανέωση των εσωτερικών σταθερών του. Έτσι μπορεί να επηρεαστεί το μέγεθος των βημάτων που γίνονται για την ανανέωση τους. Τέλος πολύ σημαντική είναι και η συνεισφορά των εποχών (epoch) για την εκπαίδευση του νευρωνικού. Μέσω των εποχών γίνεται ξεκάθαρο στον κώδικα πόσες φορές το σύνολο των δεδομένων για την εκπαίδευση πρέπει να περάσει μέσα από το δίκτυο πριν ολοκληρωθεί η όλη διαδικασία. Χάρη σε αυτή την υπερπαράμετρο μπορεί να αυξηθεί σημαντικά η ακρίβεια του δικτύου, ωστόσο χρειάζεται προσοχή για να μην οδηγηθεί το δίκτυο σε κατάσταση υπερμοντελοποίησης. Στο συγκεκριμένο πρόβλημα χρησιμοποιήθηκαν, μέγεθος batch ίσο με 25 και πλήθος εποχών ίσο με 25.

Τελικώς το νευρωνικό που προέκυψε έφερε μία ακρίβεια 98% κατά την εκπαίδευση, η οποία θεωρείται πολύ καλή, ο έλεγχος έγινε με το αντίστοιχο υποσύνολο. Εδώ χρειάζεται λίγη προσοχή γιατί αν το αρχικό σύνολο δεδομένων είχε μια αναλογία υγιή προς μη υγιή 90-10 τότε θα ήταν πιθανό το δίκτυο να μην προέβλεπε αλλά απλά να μάντευε χωρίς κάποια λογική. Αυτός ο κίνδυνος προέρχεται από κάποια ασυμμετρία στο σύνολο των δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση και μπορεί να αποτραπεί πολύ εύκολα φροντίζοντας το δείγμα να είναι συμμετρικό. Σε αυτή τη περίπτωση η αναλογία ήταν περίπου 50-50. Ωστόσο θα ήταν ελλιπές και πιθανόν λάθος αν ο έλεγχος της ποιότητας του δικτύου έμενε ως εδώ. Το υποσύνολο το οποίο επιλέχθηκε στην αρχή χωρίστηκε ξανά σε δύο ίσα υποσύνολα. Το πρώτο είναι το validation υποσύνολο, ο ρόλος του είναι να γίνεται έλεγχος της ποιότητας του δικτύου σε κάθε κύκλο εκπαίδευσης και να γίνεται αλλαγή των παραμέτρων με στόχο την αύξηση της ακρίβειας. Το δεύτερο υποσύνολο είναι το testing, ο ρόλος του είναι να γίνει έλεγχος του νευρωνικού αφού έχει ολοκληρωθεί η εκπαίδευση και αυτό είναι έτοιμο για χρήση, με αυτό τον τρόπο αποφεύγεται ένα φαινόμενο που ονομάζεται human bias. Τα δύο αυτά υποσύνολα παρουσίασαν ακρίβεια 89.4% και 85% αντίστοιχα.

5.4.2 Εναλλακτικά δίκτυα συνέλιξης

Ωστόσο το δίκτυο αυτό όταν χρησιμοποιήθηκε για τη λειτουργία της μηχανής δεν λειτούργησε όπως θα έπρεπε, σημαντικός παράγοντας για αυτό το αποτέλεσμα ήταν η φτωχή ανάλυση της κάμερας της πειραματικής διάταξης. Επομένως για να δοκιμαστεί η ποιότητα και η αξιοπιστία του νευρωνικού δικτύου ακολουθήθηκε η εξής διαδικασία.

Αρχικά το σύνολο των δεδομένων χωρίστηκε σε 65 φωτογραφίες υγιών και 65 φωτογραφίες μη υγιών δοκιμών. Κατόπιν από κάθε υποσύνολο πάρθηκαν 5 φωτογραφίες οι οποίες χρησιμοποιήθηκαν για έλεγχο (testing) ενώ οι υπόλοιπες για την εκπαίδευση (training) του νευρωνικού δικτύου. Το σύνολο των φωτογραφιών που κρατήθηκαν για εκπαίδευση υποβλήθηκαν σε μία διαδικασία επαύξησης του συνόλου τους. Το νέο σύνολο για την εκπαίδευση αποτελείται από 540 φωτογραφίες υγιών και 540 φωτογραφίες μη υγιών (εννιαπλασιασμός του συνόλου δεδομένων). Τέλος πρέπει να σημειωθεί ότι όλες οι φωτογραφίες μετατράπηκαν σε ασπρόμαυρες.

Η άνωθεν διαδικασία επαναλήφθηκε δέκα φορές δημιουργώντας δέκα διαφορετικά σύνολα δεδομένων φροντίζοντας πάντα να λαμβάνεται διαφορετικός συνδυασμός φωτογραφιών που θα χρησιμοποιηθούν για τον έλεγχο της ακρίβειας του νευρωνικού δικτύου και να μην περιλαμβάνονται στο υποσύνολο της εκπαίδευσης.

Τα νέα νευρωνικά δίκτυα που δοκιμάστηκαν και αξιολογήθηκαν είναι δύο. Το πρώτο είναι ένα δίκτυο παρόμοιας δομής με το αρχικό αλλά μικρότερου βάθους. Ειδικότερα οι κρυφές στρώσεις (hidden layers) μειώθηκαν σε δύο. Το δεύτερο είναι ένα προεκπαιδευμένο δίκτυο ResNet. Στα νέα δίκτυα δοκιμάστηκε ένας ρυθμός εκμάθησης (learning rate) ίσος με 0.0001, δηλαδή υποδεκαπλάσιος του προηγούμενου. Επίσης και στα δύο δίκτυα χρησιμοποιήθηκε και η μέθοδος early stopping για την βελτίωση των αποτελεσμάτων. Τέλος οι λοιπές μεταβλητές της ενότητας 4.4.1 παρέμειναν ίδιες.

6 Αποτελέσματα

Στο παρόν κεφάλαιο γίνεται παρουσίαση των πειραματικών δεδομένων και σχολιασμός τους. Για την αξιολόγηση της μηχανής έγιναν τρία πειράματα κόπωσης με δοκίμια σύνθετου υλικού και το ίδιο βέλος κάμψης. Τα τρία δοκίμια είναι διαστάσεων 130X23X3 mm και 4 στρώσεων.



Εικόνα 32 Πείραμα 1



Εικόνα 31 Πείραμα 2



Εικόνα 30 Πείραμα 3

Αρχικά παρουσιάζονται τα διαγράμματα δύναμης-κύκλων κόπωσης για τα τρία πειράματα. Τα τρία δοκίμια παρουσιάζουν κοντινά μέγιστα φορτία κοντά στα 180 N, ωστόσο η μεγάλη τους διαφορά αφορά την αντοχή τους στην κόπωση. Χειρότερη απόδοση έχει το πρώτο δοκίμιο για το οποίο η κάθετη δύναμη μειώνεται δραματικά στους 550 κύκλους, παρουσιάζει ένα σταθερό μικρό φορτίο για ακόμα 3000 κύκλους και μετά μηδενίζει. Παρόμοια κατάσταση εμφανίζεται και στο δεύτερο πείραμα το οποίο εμφανίζει ένα γόνατο στους 1250 κύκλους, έπειτα από αυτό το σημείο το φορτίο μειώνεται σταθερά. Τέλος η πιο ξεκάθαρη εικόνα παρουσιάζεται στο τρίτο πείραμα όπου η πλήρης καταστροφή του δοκιμίου έχει επέλθει ύστερα από 750 κύκλους.

Τα τρία διαγράμματα παρουσιάζουν μία αρκετά διαφορετική εικόνα μεταξύ τους. Όστε να εντοπιστεί η αιτία της απόκλισης τα τρία αυτά δοκίμια τοποθετήθηκαν σε μικροσκόπιο και παρατηρήθηκε η δομή τους. Δύο σημαντικά στοιχεία εντοπίστηκαν τα οποία μπορούν να δικαιολογήσουν αυτές τις αποκλίσεις στο τελικό αποτέλεσμα. Πρώτον παρατηρήθηκε ότι στο Δοκίμιο 2 μία εσωτερική στρώση του σύνθετου ήταν πολύ πιο παχιά από ότι στα άλλα δοκίμια, γεγονός που επηρεάζει την τελική του αντοχή. Δεύτερον στο Δοκίμιο 3 οι στρώσεις του συνθέτου δεν ήταν σωστά προσανατολισμένες και είχαν συνημιτονοειδή μορφή.

Η απόδοση του μηχανισμού μέτρησης κύκλων κόπωσης και φορτίου θεωρείται ικανοποιητική, αφού έγινε σύγκριση με έναν βιομηχανικό μετρητή κύκλων. Τα διαγράμματα

παρουσιάζουν συνοχή και συνέπεια. Ωστόσο πρέπει να σημειωθεί ότι οι δυναμοκυψέλες και ο αναλογοψηφιακός μετατροπέας που χρησιμοποιήθηκαν είναι για οικιακή χρήση και δεν προτείνονται για βιομηχανική χρήση. Βασικό πρόβλημα που υπήρξε με τη χρήση της συγκεκριμένης διάταξης ήταν η ύπαρξη θορύβου που αλλοίωνε τα αποτελέσματα. Το πρόβλημα αυτό περιορίστηκε σε μεγάλο βαθμό με την εισαγωγή κατάλληλης μόνωσης για τη βάση της ζυγαριάς και την ξεχωριστή γείωση της. Στον αντίποδα ο αισθητήρας βιομηχανικού τύπου παρουσίασε ένα μικρό θόρυβο ο οποίος αντιμετωπίστηκε με την χρήση ενός φίλτρου πρώτης τάξης και λίγο προγραμματισμό με βάση το σήμα εξόδου στον μικροελεγκτή. Εν κατακλείδι μία σημαντική πρόταση για βελτίωση είναι η αγορά ενός καλύτερου συστήματος δυναμοκυψελών-αναλογοψηφιακού μετατροπέα.



Διάγραμμα 3 Διάγραμμα δύναμης-κύκλων κόπωσης για το πρώτο πείραμα



Διάγραμμα 4 Διάγραμμα δύναμης-κύκλων κόπωσης για το δεύτερο πείραμα



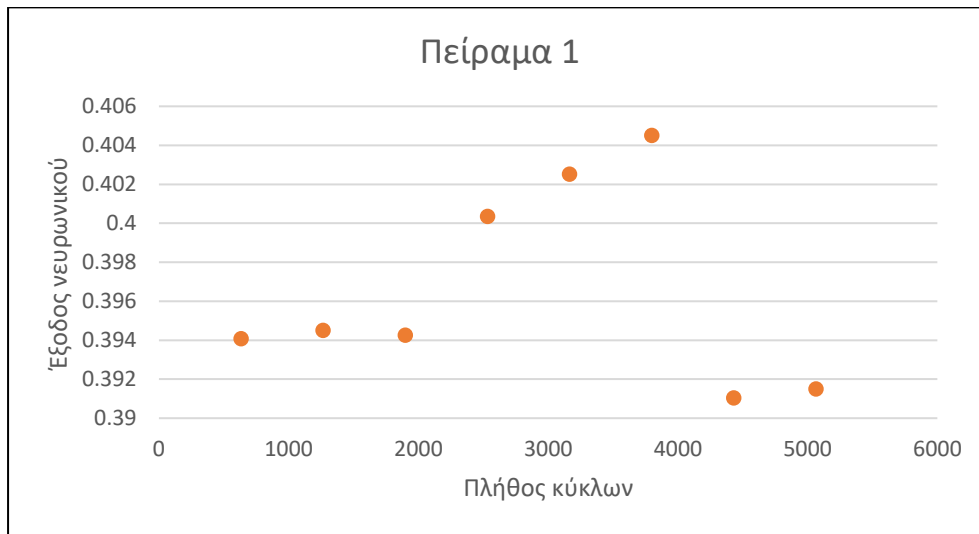
Διάγραμμα 5 Διάγραμμα δύναμης-κύκλων κόπωσης για το τρίτο πείραμα



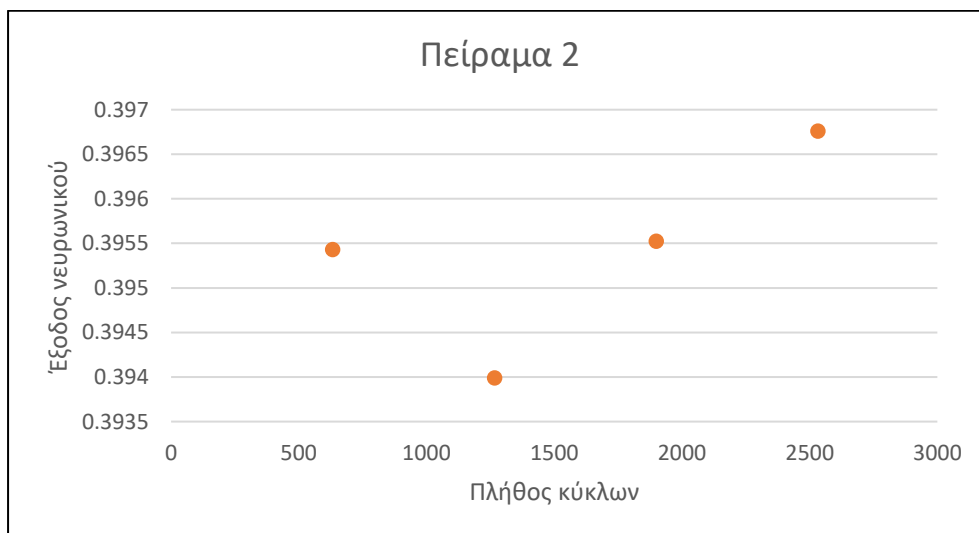
Εικόνα 33 Η μόνωση και η γείωσης της ζυγαριάς

Όσο αφορά στον εντοπισμό της αστοχίας-θραύσης του δοκιμίου για τα τρία πειράματα που διεξήχθησαν, η διακρίση μεταξύ υγιούς και κατεστραμμένου δοκιμίου γίνεται με βάση την τιμή στην έξοδο του νευρωνικού. Από την εκπαίδευση καθορίστηκε ότι για τιμές μικρότερες του 0.5 το νευρωνικό κατηγοριοποιεί την εικόνα ως υγιή, διαφορετικά το δοκίμιο θεωρείται κατεστραμμένο. Γενικότερα οι τιμές της εξόδου του νευρωνικού κυμαίνονται στο 0.4 ανεξαρτήτως της κατάστασης του δοκιμίου. Επομένως η ικανοποιητική απόδοση που υπήρχε κατά την κατασκευή και τον έλεγχο του δικτύου απέχει παρασάγγας από τα τελικά πειράματα. Το πρόβλημα εντοπίστηκε στην κάμερα που χρησιμοποιείται στην τελική διάταξη η οποία δίνει μία εικόνα που δεν έχει την ίδια πυκνότητα σε πληροφορία όπως οι εικόνες που χρησιμοποιήθηκαν για τη δημιουργία του νευρωνικού δικτύου. Απότοκο αυτού είναι η κακή τελική απόδοση του νευρωνικού δικτύου. Επομένως είναι απαραίτητη η χρήση μίας κάμερας

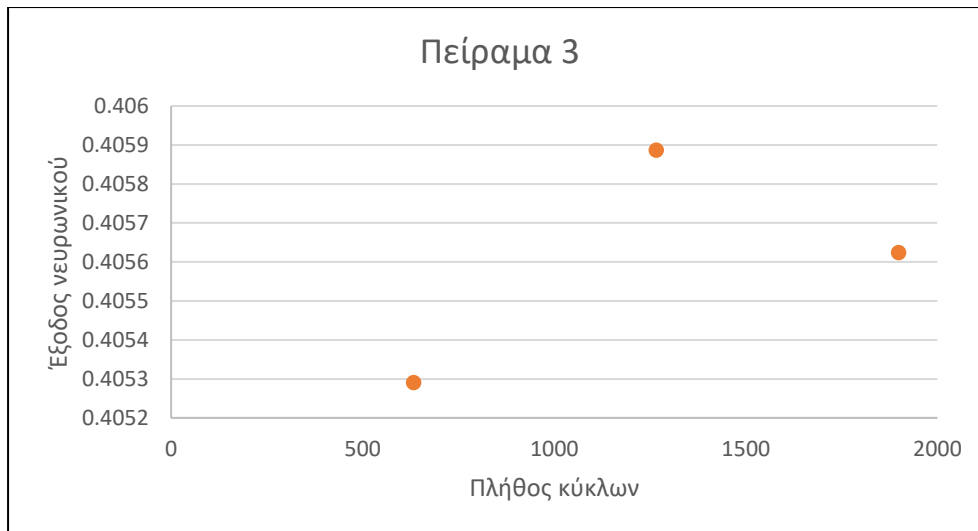
που θα έχει καλή ευκρίνεια και θα διασφαλίζει ότι δεν θα υπάρχει αυτή η απώλεια στην πληροφορία.



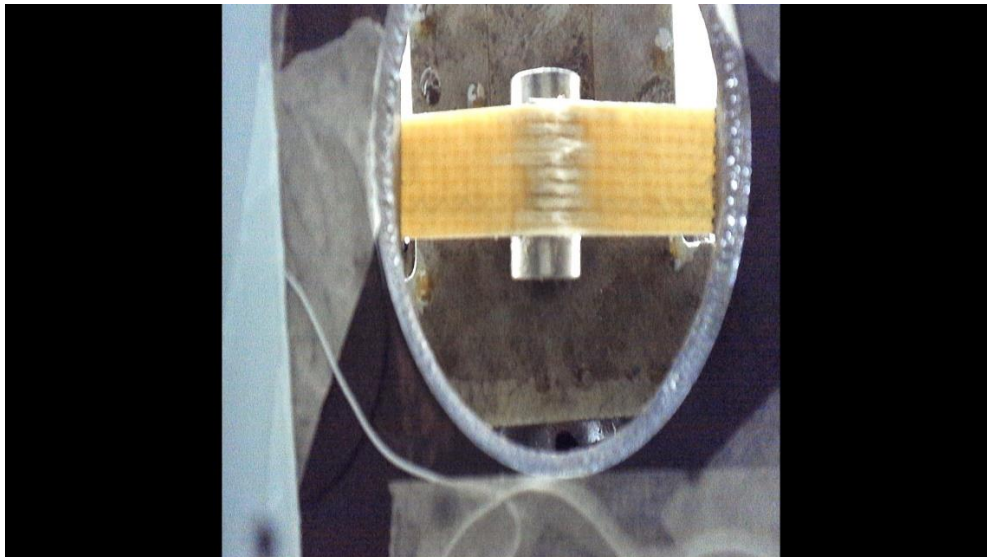
Διάγραμμα 6 Διάγραμμα των τιμών του νευρωνικού δικτύου και των κύκλων κόπωσης για το πρώτο πείραμα



Διάγραμμα 7 Διάγραμμα των τιμών του νευρωνικού δικτύου και των κύκλων κόπωσης για το πρώτο πείραμα



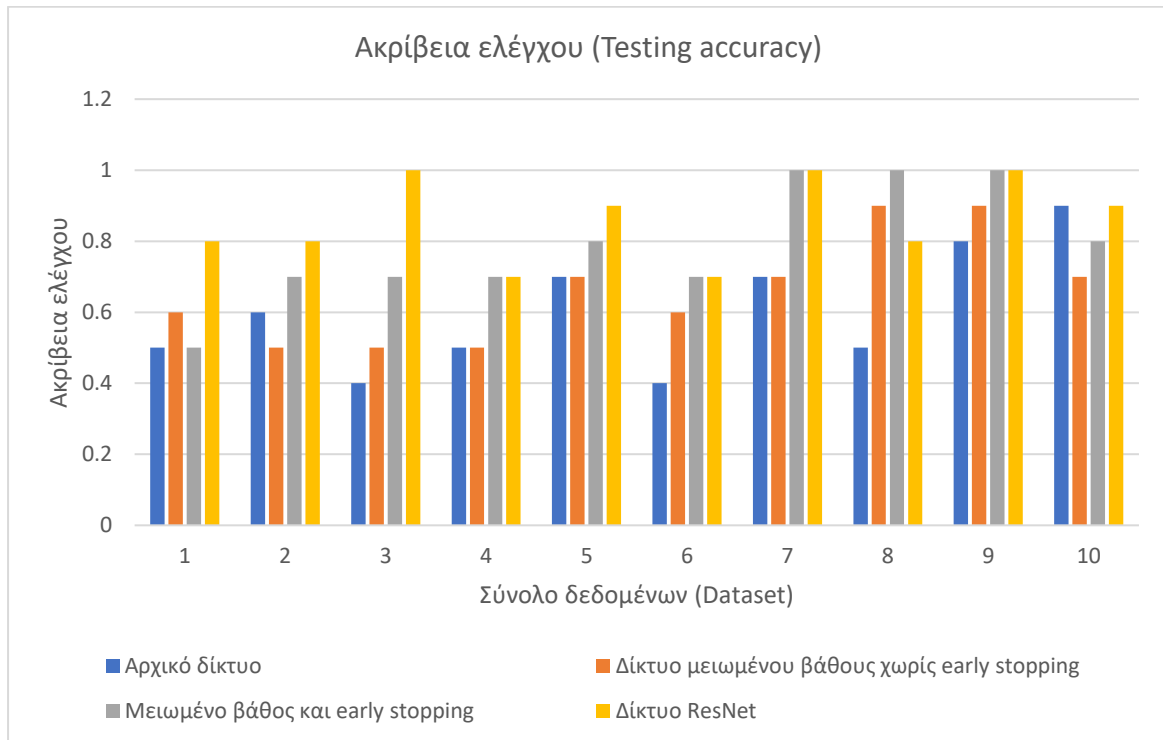
Διάγραμμα 8 Διάγραμμα των τιμών του νευρωνικού δικτύου και των κύκλων κόπωσης για το τρίτο πείραμα



Εικόνα 34 Το δοκίμιο από την κάμερα του εργαστηρίου

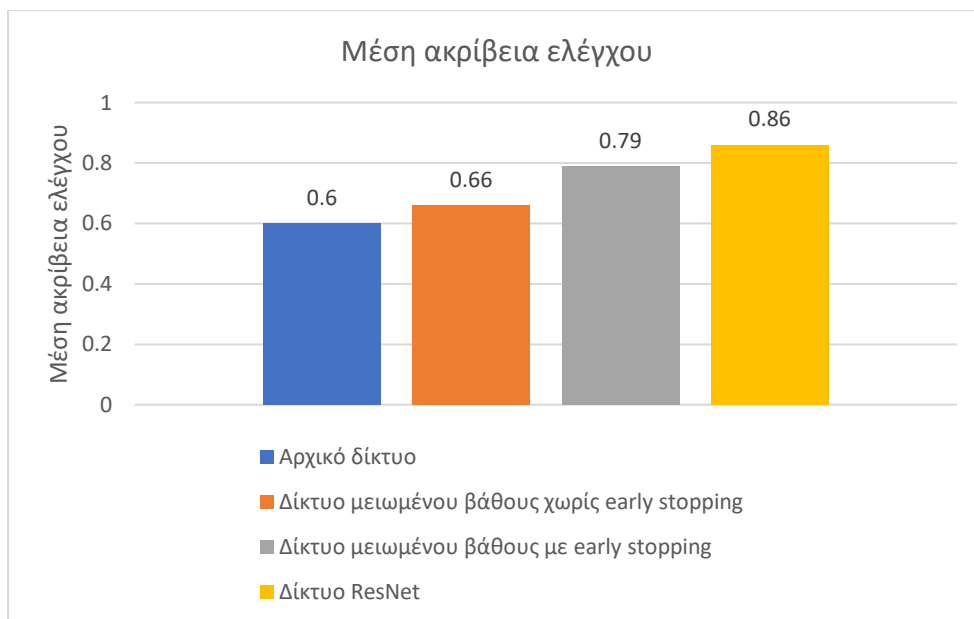
Κατόπιν έγινε μία μελέτη βελτίωσης και αξιολόγησης του νευρωνικού δικτύου με τη χρήση διαφορετικών μοντέλων και συνόλου δεδομένων. Τα αποτελέσματα που παρουσιάζονται προέκυψαν ύστερα από εκπαίδευση και έλεγχο με τα νέα σύνολα δεδομένων όπως αυτά προέκυψαν από τη διαδικασία της 4.4.2 για τέσσερις δομές δικτύων συνέλιξης. Η πρώτη δομή αφορά το αρχικό δίκτυο που αναλύθηκε στην ενότητα 4.4.1. Το δίκτυο μειωμένου βάθους χωρίς early stopping είναι αυτό που παρουσιάστηκε στη ενότητα 4.4.2, το οποίο όμως έχει μία σημαντική διαφορά με το προηγούμενο πέρα από το μικρότερο πλήθος hidden layers. Η εκπαίδευση του έγινε με μόλις τρεις εποχές, αυτή η επιλογή έγινε γιατί ύστερα από δοκιμές παρατηρήθηκε ότι το δίκτυο κατά τον έλεγχο σύγκλινε πολύ γρήγορα σε μικρές τιμές απώλειας (loss) και σε μεγάλες τιμές ακρίβειας (accuracy) κάτι τέτοιο δείχνει μία υπερμοντελοποίηση. Το τρίτο δίκτυο είναι ίδιο με το δεύτερο, αλλά χρησιμοποιεί τη μέθοδο early stopping παρατηρώντας τη μεταβλητή της ακρίβειας του υποσυνόλου του ελέγχου σε κάθε εποχή. Παράλληλα το δίκτυο έχει τη δυνατότητα να τρέξει μέχρι και είκοσι εποχές. Το

τέταρτο δίκτυο είναι το προεκπαιδευμένο ResNet με την ίδια μέθοδο early stopping και την ίδια δυνατότητα εποχών.



Διάγραμμα 9 Η ακρίβεια των δικτύων για κάθε σύνολο δεδομένων

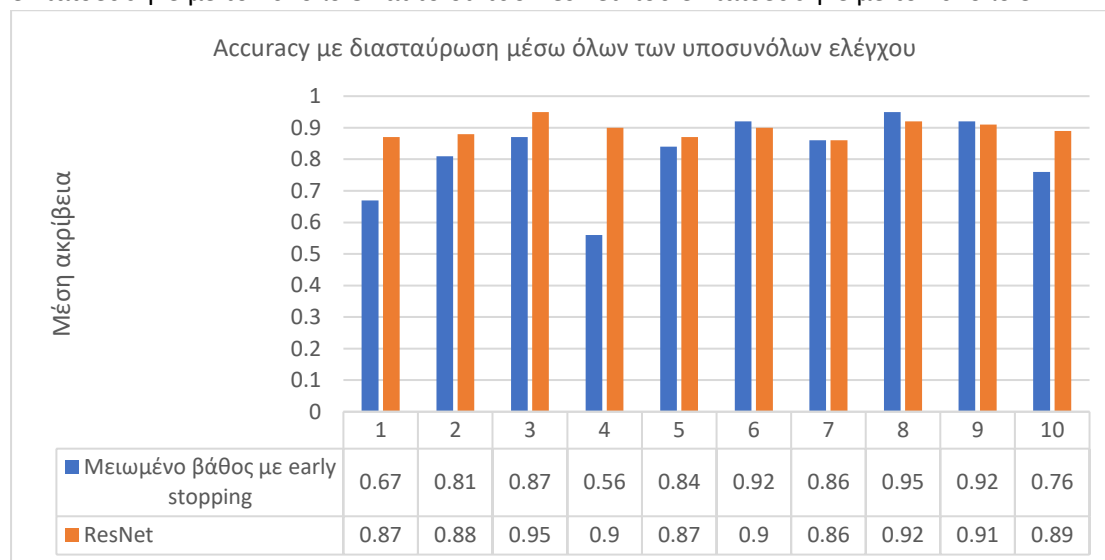
Στο Διάγραμμα 9 είναι εμφανής η ανωτερότητα του δικτύου μειωμένου βάθους με early stopping και του δικτύου ResNet. Η συγκεκριμένη παρατήρηση φαίνεται καλύτερα στο Διάγραμμα 10 με το αρχικό δίκτυο να έχει την χειρότερη μέση ακρίβεια κατά τον έλεγχο (60%) και το ResNet την καλύτερη (86%).



Διάγραμμα 10 Η μέση ακρίβεια ελέγχου για τα τέσσερα δίκτυα

Στο σημείο αυτό γεννάται το ζήτημα πιο από τα είκοσι επικρατέστερα μοντέλα (τα μοντέλα μειωμένου βάθους με early stopping και ResNet) θα επιλεγεί για την εφαρμογή. Το πρόβλημα του μικρού διαθέσιμου συνόλου δεδομένων και του ακόμα μικρότερου υποσυνόλου που χρησιμοποιήθηκε για τον έλεγχο δημιουργεί μία σημαντική αναξιοπιστία για τα αποτελέσματα. Ένα παράδειγμα είναι η ακρίβεια του δικτύου ResNet για το Σύνολο Δεδομένων 3. Το 100% της ακρίβειας μπορεί να προέκυψε τυχαία δεδομένου ότι το πείραμα επαναλήφθηκε μόλις 10 φορές. Η λύση που προτάθηκε δίχως να είναι βέλτιστη, αλλά δεδομένου του μικρού όγκου των δεδομένων δεν υπάρχει καλύτερη, είναι στο κάθε δίκτυο να εισαχθούν όλα τα υποσύνολα που είναι για έλεγχο (testing datasets) και από αυτά να προκύψει η ακρίβεια του κάθε δικτύου. Το πρόβλημα με τη συγκεκριμένη διαδικασία είναι ότι 9/10 υποσύνολα ελέγχου περιέχονται στο υποσύνολο εκπαίδευσης κάθε νευρωνικού δικτύου (έστω και αν είναι τροποποιημένα εξαιτίας της επαύξησης του συνόλου), απότοκο αυτού είναι ότι ο κίνδυνος διαρροής των δεδομένων (data leakage) να είναι υπαρκτός. Το γεγονός αυτό ενδέχεται να δώσει πλασματικά αποτελέσματα.

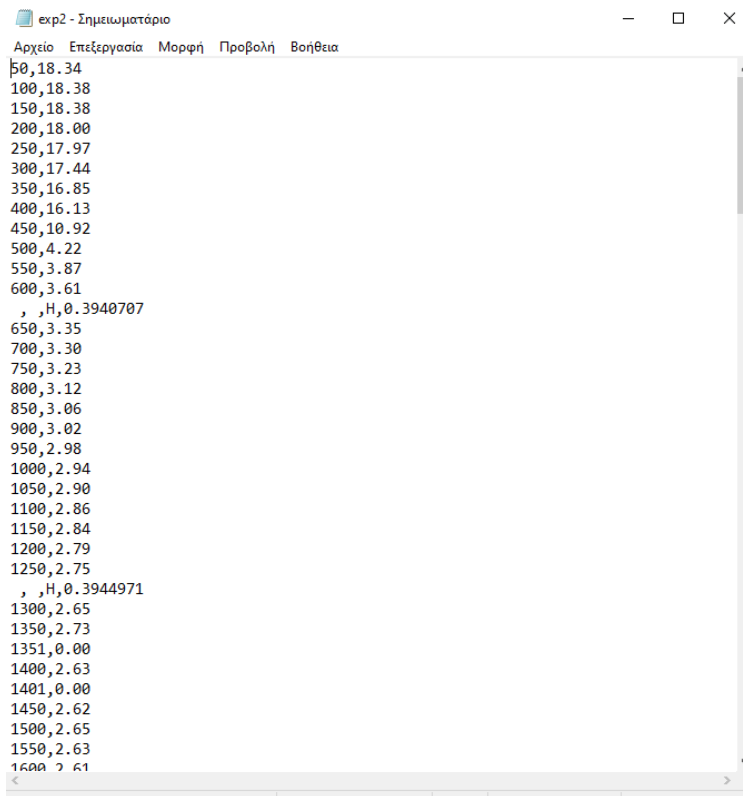
Εν κατακλείδι σύμφωνα με το Διάγραμμα 11 τα δύο καλύτερα νευρωνικά δίκτυα για τη συγκεκριμένη εφαρμογή είναι το δίκτυο μειωμένου βάθους με early stopping που εκπαιδεύθηκε με το Σύνολο 8 και το δίκτυο ResNet που εκπαιδεύθηκε με το Σύνολο 3.



Διάγραμμα 11 Η ακρίβεια των δύο δικτύων μέσω διασταύρωσης των υποσυνόλων ελέγχου

Η αρχή λειτουργίας της μηχανής κόπωσης είναι συνολικά σωστή και κάνει αυτά για τα οποία δημιουργήθηκε. Ωστόσο για την βέλτιστη λειτουργία της είναι αναγκαία η αναβάθμιση του εξοπλισμού. Η συγκεκριμένη κατασκευή έγινε με χαμηλό προϋπολογισμό καθώς πρόκειται για ένα πρωτότυπο και μία πρώτη προσέγγιση στο συγκεκριμένο πρόβλημα. Ενδεικτικά η μία από τις τέσσερις δυναμοκυψέλες κοστίζει 6.8€ και ο αναλογισμιακός μετατροπέας 1.8€.

Η χρήση του συγκεκριμένου μηχανήματος από κάποιον χειριστή είναι εξαιρετικά απλή καθώς από εκείνον απαιτείται μόνο η ρύθμιση του βέλους κάμψης και να εκτελέσει το αντίστοιχο πρόγραμμα. Τα τελικά δεδομένα του παρέχονται σε ένα αρχείο .csv μέσω του οποίου μπορεί να κάνει όποια επεξεργασία θέλει σε ένα φύλλο excel.



Εικόνα 35 Το αρχείο .csv με τα αποτελέσματα

7 Συμπεράσματα για μελλοντική εργασία

7.1 Συμπεράσματα

- Ο έλεγχος της μηχανής μπορεί να γίνει πολύ καλά με εξοπλισμό χαμηλού κόστους.
- Η χρήση δυναμοκυψελών και αναλογοψηφιακού μετατροπέα χαμηλού κόστους δεν μπορεί να δώσει μετρήσεις υψηλής ποιότητας.
- Ο ηλεκτρικός θόρυβος είναι δύσκολο να εντοπιστεί και να περιοριστεί πλήρως.
- Η κάμερα που χρησιμοποιείται για την εκπαίδευση ενός CNN πρέπει να είναι παρόμοια με αυτή της εφαρμογής και καλής ανάλυσης, ενώ και οι συνθήκες πρέπει να είναι και αυτές παρόμοιες.
- Τα νευρωνικά δίκτυα μπορούν να εκπαιδευτούν με ένα σχετικά μικρό πλήθος δεδομένων χάρη στα δίκτυα συνέλιξης.
- Η επαύξηση του συνόλου των δεδομένων είναι καλό να μην είναι πολλές φορές πολλαπλάσια του αρχικού συνόλου, καθώς ένα μικρότερο σύνολο μπορεί να δώσει καλύτερα αποτελέσματα.
- Η δομή του δικτύου πρέπει να ξεκινάει από μικρό βάθος (μικρό πλήθος hidden layers) και αν υπάρχει αιτία να αυξάνεται, έτσι ώστε το δίκτυο να είναι όσο πιο ευέλικτο γίνεται.
- Τα δεδομένα που χρησιμοποιούνται για τον έλεγχο του δικτύου (testing) δεν πρέπει να περιέχονται με καμία τους μορφή στο σύνολο εκπαίδευσης (training) διότι μπορούν να αλλοιώσουν τα αποτελέσματα.
- Η μέθοδος early stopping σε συνδυασμό με callbacks πρέπει να χρησιμοποιείται κάθε φορά για να εξάγεται το βέλτιστο δίκτυο και να μην απασχολείται ο χρήστης σε μεγάλο βαθμό με το πλήθος των εποχών και την απόδοση του δικτύου.
- Το βέλτιστο από οικονομικής σκοπιάς είναι η χρήση ενός μικροελεγκτή, αφού ο έλεγχος μπορεί να γίνει από έναν μικροελεγκτή εξαιτίας των σχετικά μεγάλων χρόνων μεταξύ των ενεργειών.

7.2 Μελλοντική εργασία

Πρέπει να σημειωθεί ότι οι δυναμοκυψέλες και ο αναλογοψηφιακός μετατροπέας που χρησιμοποιήθηκαν είναι για ερασιτεχνική (DIY) χρήση και δεν προτείνονται για βιομηχανική χρήση. Βασικό πρόβλημα που υπήρξε με τη λειτουργία της συγκεκριμένης διάταξης ήταν η ύπαρξη θορύβου που αλλοίωνε τα αποτελέσματα. Το πρόβλημα αυτό περιορίστηκε σε μεγάλο βαθμό με την εισαγωγή κατάλληλης μόνωσης για τη βάση της ζυγαριάς και την ξεχωριστή γείωση της. Στον αντίποδα ο αισθητήρας βιομηχανικού τύπου παρουσίασε ένα μικρό θόρυβο ο οποίος αντιμετωπίστηκε με την χρήση ενός φίλτρου πρώτης τάξης και λίγο προγραμματισμό με βάση το σήμα εξόδου στον μικροελεγκτή. Εν κατακλείδι μία σημαντική πρόταση για βελτίωση είναι η αγορά ενός καλύτερου συστήματος δυναμοκυψελών-αναλογοψηφιακού μετατροπέα. Μία ακόμα βελτίωση είναι η χρήση κάμερας υψηλότερης ανάλυσης για την είσοδο του νευρωνικού δικτύου. Έτσι ώστε αυτό να μπορεί να εξάγει καλύτερα αποτελέσματα.

8 Παραρτήματα

8.1 Κώδικας επαύξησης των δεδομένων

```
from PIL import Image, ImageEnhance, ImageFilter
import numpy as np
import cv2
from sklearn.cluster import KMeans
import os
import rawpy
import tensorflow as tf
from keras import layers
import keras
import matplotlib.pyplot as plt
from image_shuffler import Shuffler

i=3
string = "H:/A Σχολή/Διπλωματική/Unhealthy Dataset/UNHEALTHY_{0}.jpg".format(i)
file = string
image = Image.open(file)
data_augmentation = keras.Sequential([
    #resize και rescale για οικονομία και σύγκλιση
    layers.Resizing(400,400),
    layers.Rescaling(1./255),
])
#flip and rotate
data_raven = keras.Sequential([
    layers.RandomFlip("horizontal_and_vertical"),
    layers.RandomRotation(0.4),
])

#RGB, RG, GB, RB, R, G, B generator
def Color_generator(IMAGE):
    IMG = data_augmentation(IMAGE)
    IMG = np.array(IMG)
    R = IMG[:, :, 0]; G = IMG[:, :, 1]; B = IMG[:, :, 2]
    Z = tf.zeros((400,400))
    RG = np.stack((R,G,Z), axis=-1); GB = np.stack((Z,G,B), axis=-1); RB = np.stack((R,Z,B),\
axis=-1)
    R = np.stack((R,Z,Z), axis=-1); G = np.stack((Z,G,Z), axis=-1); B = np.stack((Z,Z,B), axis=-1);
    return IMG, RG, GB, RB, R, G, B
#rotate images

folder_path = 'H:/A Σχολή/Διπλωματική/Healthy Dataset Processed'
file_list = os.listdir(folder_path)
for i in range(0,len(file_list)):
```

```

string = "H:/A Σχολή/Διπλωματική/Healthy Dataset
Processed/HEALTHY_{0}.jpg".format(i+1); image = Image.open(string)
for j in range(6):
    image = data_raven(image)
    tf.keras.preprocessing.image.save_img('H:/A
Σχολή/Διπλωματική/Spare/HEALTHY_{0}.jpg'.format(6*i+j+1), image)

#shuffle images
folder_path = 'H:/A Σχολή/Διπλωματική/Healthy Dataset Processed'
file_list = os.listdir(folder_path)
os.chdir("H:/A Σχολή/Διπλωματική/Healthy Dataset Processed")
for i in range(0,len(file_list)):
    string = "H:/A Σχολή/Διπλωματική/Healthy Dataset
Processed/HEALTHY_{0}.jpg".format(i+1); image = Image.open(string)
    image = Shuffler("HEALTHY_{0}.jpg".format(i+1))
    for j in range(1):
        image.shuffle(matrix=(2, 2))
        tf.keras.preprocessing.image.save_img('H:/A
Σχολή/Διπλωματική/Spare/HEALTHY_{0}.jpg'.format(2*i+j+1), image)

```

8.2 Κώδικες νευρωνικών δικτύων

8.2.1 Κώδικας νευρωνικού δικτύου ενότητας 4.4.1

```

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models
import time
from PIL import Image
import numpy as np

image_size = (200, 200)
batch_size = 25
train_ds, val_ds = keras.utils.image_dataset_from_directory(
    "H:/Dataset",
    labels="inferred",
    label_mode="binary",
    validation_split=0.2,
    subset="both",
    seed=2024,
    image_size=image_size,
    shuffle=True,
    batch_size=batch_size,
)

val_batches = tf.data.experimental.cardinality(val_ds)
test_batches = val_batches//2

```

```

test_ds = val_ds.take(test_batches)
val_ds = val_ds.skip(test_batches)

start_time = time.time()
"""
model = models.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(200, 200, 3)),
    layers.AveragePooling2D((2, 2)),
    layers.Conv2D(254, (3, 3), activation='relu'),
    layers.AveragePooling2D((2, 2)),
    layers.Conv2D(254, (3, 3), activation='relu'),
    layers.AveragePooling2D((2, 2)),
    layers.Conv2D(254, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.4),
    layers.Dense(1, activation='sigmoid') # Binary classification output
])

BIN = keras.metrics.BinaryAccuracy(threshold=0.5)
opt = keras.optimizers.Adam(learning_rate=0.002)

model.compile(
    optimizer=opt,
    loss='binary_crossentropy',
    metrics=[BIN])

model.fit(train_ds,
    epochs=25,
    verbose=1,
    validation_data=val_ds,
)
end_time = time.time()
#test_loss, test_acc = model.evaluate(val_ds)

elapsed_time = end_time - start_time
print("Total time elapsed:", elapsed_time/3600, "hours")

save_path = 'H:/NEURAL NETWORK/Brain'
model.save(save_path)
"""

loaded_model = tf.keras.models.load_model('H:/NEURAL NETWORK/Brain')

test_loss, test_accuracy = loaded_model.evaluate(test_ds)

```



```

print("Test Loss:", test_loss)
print("Test Accuracy:", test_accuracy)
"""
string="H:/A Σχολή/Διπλωματική/Healthy Dataset/HEALTHY_18.jpg"
image = keras.utils.load_img(string,
    target_size=(200,200),
    interpolation="nearest",
    keep_aspect_ratio=False,
    )
input_arr = keras.utils.img_to_array(image)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predict=loaded_model(input_arr)
print(predict)
"""

```

8.2.2 Κώδικας νευρωνικού δικτύου μειωμένου βάθους

```

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models
import time
from PIL import Image
import numpy as np

image_size = (200, 200)
batch_size = 25
train_ds, val_ds = keras.utils.image_dataset_from_directory(
    "H:/Dataset",
    labels="inferred",
    label_mode="binary",
    validation_split=0.2,
    subset="both",
    seed=2024,
    image_size=image_size,
    shuffle=True,
    batch_size=batch_size,
)

val_batches = tf.data.experimental.cardinality(val_ds)
test_batches = val_batches//2
test_ds = val_ds.take(test_batches)
val_ds = val_ds.skip(test_batches)

start_time = time.time()
"""
model = models.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(200, 200, 3)),

```

```

layers.AveragePooling2D((2, 2)),
layers.Conv2D(254, (3, 3), activation='relu'),
layers.Flatten(),
layers.Dense(128, activation='relu'),
layers.Dropout(0.4),
layers.Dense(1, activation='sigmoid') # Binary classification output
])

BIN = keras.metrics.BinaryAccuracy(threshold=0.5)
opt = keras.optimizers.Adam(learning_rate=0.002)

model.compile(
optimizer=opt,
loss='binary_crossentropy',
metrics=[BIN])

model.fit(train_ds,
epochs=25,
verbose=1,
validation_data=val_ds,
)
end_time = time.time()
#test_loss, test_acc = model.evaluate(val_ds)

elapsed_time = end_time - start_time
print("Total time elapsed:", elapsed_time/3600, "hours")

save_path = 'H:/NEURAL NETWORK/Brain'
model.save(save_path)
"""

loaded_model = tf.keras.models.load_model('H:/NEURAL NETWORK/Brain')

test_loss, test_accuracy = loaded_model.evaluate(test_ds)

print("Test Loss:", test_loss)
print("Test Accuracy:", test_accuracy)
"""

string="H:/Α Σχολή/Διπλωματική/Healthy Dataset/HEALTHY_18.jpg"
image = keras.utils.load_img(string,
target_size=(200,200),
interpolation="nearest",
keep_aspect_ratio=False,
)
input_arr = keras.utils.img_to_array(image)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predict=loaded_model(input_arr)

```

```
print(predict)
"""
```

8.2.3 Κώδικας νευρωνικού δικτύου μειωμένου βάθους με early stopping

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models
import time
from PIL import Image
import numpy as np
import tensorflow as tf
from tensorflow.keras.layers import Rescaling, Conv2D, BatchNormalization, ReLU
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import ResNet50

start_time = time.time()

for file in range(1,11):
    tf.keras.backend.clear_session()

    BIN = keras.metrics.BinaryAccuracy(threshold=0.5)
    opt = keras.optimizers.Adam(learning_rate=0.0001)

    print("Beggining of Dataset_{0}".format(file))
    image_size = (200, 200)
    batch_size = 19
    train_ds = image_dataset_from_directory(
        'H:/Last_Stand/Dataset_{0}/Training/Augmented'.format(file),
        labels="inferred",
        label_mode="binary",
        seed=2024,
        image_size=image_size,
        shuffle=True,
        batch_size=batch_size,
    )

    test_ds = image_dataset_from_directory(
        'H:/Last_Stand/Dataset_{0}/Testing/Test'.format(file),
        shuffle=False,
        batch_size=batch_size,
        image_size=image_size,
        label_mode='binary'
    )
```

```

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(200, 200,
3))
base_model.trainable = False

model = models.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(200, 200, 3)),
    layers.AveragePooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.4),
    layers.Dense(1, activation='sigmoid') # Binary classification output
])

model.compile(
optimizer=opt,
loss='binary_crossentropy',
metrics=[BIN])

early_stopping = EarlyStopping(
monitor='val_binary_accuracy', # Metric to monitor
min_delta=0.15,
patience=3,
mode='max', # Number of epochs to wait for improvement
restore_best_weights=True # Restore model weights from the epoch with the best value of
the monitored metric
)

model_checkpoint = ModelCheckpoint(
'H:/Last_Stand/Dataset_{0}/Brain_{0}'.format(file), # Path where the model will be
saved
monitor='val_binary_accuracy', # Metric to monitor
save_best_only=True, # Save only the best model (based on `monitor`)
mode='max', # Mode for monitoring (minimize the monitored metric)
verbose=1 # Verbosity mode (1 = progress messages)
)

model.fit(train_ds,
epochs=20,
verbose=1,
validation_data=test_ds,
callbacks=[early_stopping, model_checkpoint] # List of callbacks
)
#save_path = 'H:/Last_Stand/Dataset_{0}/Brain_{0}'.format(file)
#model.save(save_path)

end_time = time.time()

```

```
elapsed_time = end_time - start_time
print("Total time elapsed:", elapsed_time/3600, "hours")
```

8.2.4 Κώδικας νευρωνικού δικτύου ResNet

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models
import time
from PIL import Image
import numpy as np
import tensorflow as tf
from tensorflow.keras.layers import Rescaling, Conv2D, BatchNormalization, ReLU
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import ResNet50

start_time = time.time()

for file in range(1,11):
    tf.keras.backend.clear_session()

    BIN = keras.metrics.BinaryAccuracy(threshold=0.5)
    opt = keras.optimizers.Adam(learning_rate=0.0001)

    print("Beggining of Dataset_{0}".format(file))
    image_size = (200, 200)
    batch_size = 19
    train_ds = image_dataset_from_directory(
        'H:/Last_Stand/Dataset_{0}/Training/Augmented'.format(file),
        labels="inferred",
        label_mode="binary",
        seed=2024,
        image_size=image_size,
        shuffle=True,
        batch_size=batch_size,
    )

    test_ds = image_dataset_from_directory(
        'H:/Last_Stand/Dataset_{0}/Testing/Test'.format(file),
        shuffle=False,
        batch_size=batch_size,
        image_size=image_size,
        label_mode='binary'
    )
```

```

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(200, 200,
3))
base_model.trainable = False

model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(), # Pooling layer to reduce output
    layers.Dense(128, activation='relu'), # Tune size
    layers.Dropout(0.3), # Tune %
    layers.Dense(1, activation='sigmoid')
])

model.compile(
optimizer=opt,
loss='binary_crossentropy',
metrics=[BIN])

early_stopping = EarlyStopping(
monitor='val_binary_accuracy', # Metric to monitor
min_delta=0.15,
patience=3,
mode='max', # Number of epochs to wait for improvement
restore_best_weights=True # Restore model weights from the epoch with the best value of
the monitored metric
)

model_checkpoint = ModelCheckpoint(
'H:/Last_Stand/Dataset_{0}/Brain_{0}'.format(file), # Path where the model will be
saved
monitor='val_binary_accuracy', # Metric to monitor
save_best_only=True, # Save only the best model (based on `monitor`)
mode='max', # Mode for monitoring (minimize the monitored metric)
verbose=1 # Verbosity mode (1 = progress messages)
)

model.fit(train_ds,
epochs=20,
verbose=1,
validation_data=test_ds,
callbacks=[early_stopping, model_checkpoint] # List of callbacks
)
#save_path = 'H:/Last_Stand/Dataset_{0}/Brain_{0}'.format(file)
#model.save(save_path)

end_time = time.time()
elapsed_time = end_time - start_time

```

```
print("Total time elapsed:", elapsed_time/3600, "hours")
```

8.3 Κώδικας ελέγχου Arduino

```
/*
-----
HX711_ADC
Arduino library for HX711 24-Bit Analog-to-Digital Converter for Weight Scales
Olav Kallhovd sept2017
-----
*/

/*
  Settling time (number of samples) and data filtering can be adjusted in the config.h file
  For calibration and storing the calibration value in eeprom, see example file
  "Calibration.ino"

  The update() function checks for new data and starts the next conversion. In order to
  acheive maximum effective
  sample rate, update() should be called at least as often as the HX711 sample rate;
  >10Hz@10SPS, >80Hz@80SPS.
  If you have other time consuming code running (i.e. a graphical LCD), consider calling
  update() from an interrupt routine,
  see example file "Read_1x_load_cell_interrupt_driven.ino".

  This is an example sketch on how to use this library
*/
// Start Calibration setup//
volatile float max_force = 0;// the max force for a number of cycles
#include <HX711_ADC.h>
#if defined(ESP8266) || defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif

//pins:
const int HX711_dout = 6; //mcu > HX711 dout pin
const int HX711_sck = 7; //mcu > HX711 sck pin
volatile float max_weight = 0.0; // the variable for the max force per cycle
const int data_sample_time = 70; //the speed of data sample

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_calVal_eepromAdress = 0;
unsigned long t = 0;
// End Calibration setup//
```

```

//Start Counter setup//
volatile int counter = 0;
int a = 200; // το 200 επιλέχθηκε από το διάγραμμα
int b = 0;
int cycles_write = 50; //the number of cycles we print weight and number of cycles
int cycles_neuron = 633; //the number of cycles we stop the machine and activate the
neuron
//End Counter setup//

// Start Control setup//
int control_pin=11;
char x; //for the communication between python and microcontroller
//End Control pin

void setup() {
//Start Calibration setup//
Serial.begin(2400); delay(10);
//Serial.println();
//Serial.println("Starting...");

float calibrationValue; // calibration value
calibrationValue = 22855.54; // uncomment this if you want to set this value in the sketch
25304.27
#ifdef ESP8266 || defined(ESP32)
//EEPROM.begin(512); // uncomment this if you use ESP8266 and want to fetch this value
from eeprom
#endif

LoadCell.begin();
//LoadCell.setReverseOutput();
unsigned long stabilizingtime = 2000; // tare preciscion can be improved by adding a few
seconds of stabilizing time
boolean _tare = true; //set this to false if you don't want tare to be performed in the next
step
LoadCell.start(stabilizingtime, _tare);
if (LoadCell.getTareTimeoutFlag()) {
}
else {
LoadCell.setCalFactor(calibrationValue); // set calibration factor (float)
}
while (!LoadCell.update());

if (LoadCell.getSPS() < 7) {
Serial.println("!!Sampling rate is lower than specification, check MCU>HX711 wiring and
pin designations");
}
}

```



```

else if (LoadCell.getSPS() > 100) {
  Serial.println("!!Sampling rate is higher than specification, check MCU>HX711 wiring and
pin designations");
}
//End Calibration setup//

//Start Counter setup//
pinMode(A0, INPUT);
pinMode(2, INPUT);
attachInterrupt(digitalPinToInterrupt(2), pin_ISR, RISING);
//End Counter setup//

//Start Control setup
pinMode(control_pin,OUTPUT);
//End Control setup
}

void loop() {
  restart:
  if (counter%cycles_neuron==0 && counter!=0 && counter!=1){ //when we want to stop
the machine and activate the neuron
  // Serial.println("We will stop");
  Serial.println("Y");
  delay(1000);
  digitalWrite(control_pin,HIGH);
  // Serial.println("We will restart");
  // delay(2000);
  // Serial.println(Serial.available());
  // Serial.println(x);
  x=Serial.read();
  while(Serial.available()==0){
  }
  digitalWrite(control_pin,LOW);
  //Serial.println("Restart");
  delay(2000);
  goto restart;
  }

  if (counter%cycles_write==0 && counter!=0 && counter!=1){ //when we want to print the
average weight and the number of cycles
  max_force=max_force/cycles_write;
  Serial.println(max_force);
  delay(150);
  Serial.println(counter);
  delay(150);
  max_force=0;
  //max_weight=0;
  delay(600); //if it double prints raise the delay time

```

```
}
```

```
static boolean newDataReady = 0;  
const int serialPrintInterval = 50; //increase value to slow down serial print activity
```

```
// check for new data/start next conversion:  
if (LoadCell.update()) newDataReady = true;
```

```
// get smoothed value from the dataset:
```

```
if (newDataReady) {  
  if (millis() > t + serialPrintInterval) {  
    float i = LoadCell.getData();  
    i=i+4.23; // offset for load  
    if (i<0) i=0;
```

```
    //Serial.println(i);  
    if (max_weight<i) max_weight=i;  
    //Serial.println(max_weight);  
    delay(data_sample_time);  
    newDataReady = 0;  
    t = millis();
```

```
  }  
}  
}
```

```
void pin_ISR(){  
  b = analogRead(A0);  
  if (abs(a-b)>500){ //treshhold  
    counter++;  
    max_force = max_force+max_weight;  
    max_weight=0;  
    //Serial.println(counter);  
  }  
}
```

8.4 Κώδικας Python

```
import time
import serial
import csv
from keras.models import load_model
import cv2 as cv
import keras
from keras import layers
import tensorflow as tf
import numpy as np

model = load_model('C:/Users/USER/Desktop/Brain')
ser = serial.Serial("COM9",2400)
time.sleep(1)
data=[]

def neuron():
    cam = cv.VideoCapture(0) #cv.CAP_DSHOW
    result, image = cam.read()
    for k in range(0,3):
        for i in range(0,480):
            for j in range(0,640):
                if i>=240 and i<=285 and j<=430 and j>=163:
                    pass
                else:
                    image[i][j][k]=0

#resize και rescale για οικονομία και σύγκλιση
image_reform = keras.Sequential([
    layers.Resizing(200,200),
    layers.Rescaling(1./255),
])
image = image_reform(image)
image = np.expand_dims(image, axis=0)
predictions = model(image)
if predictions<0.5:
    state = "H"
else:
    state = "U"
predictions = predictions.numpy()
data.append(" "); data.append(" "); data.append(state); data.append(predictions[0][0])
with open('C:/Users/USER/Desktop/ZY/data.csv','a',encoding='UTF8',newline='') as f:
    writer=csv.writer(f)
    writer.writerow(data)
    data.clear()
print(predictions)
```

```

while True:
    if ser.inWaiting(>0):
        with open('C:/Users/USER/Desktop/ZY/data.csv','a',encoding='UTF8',newline='') as f:
            writer=csv.writer(f)
            force = ser.readline()
            force = str(force, 'utf-8')
            force = force.strip('\r\n')
            if force=="Y":
                time.sleep(5)
                neuron()
                ser.write(str.encode(' '))
                # ser.write(str.encode(' '))
                # print('We have send the message')
                time.sleep(5)
                continue
                # print("Continue is no working")
            print(f"The force is {force}")
            counter = ser.readline()
            counter = str(counter, 'utf-8')
            counter = counter.strip('\r\n')
            print(f"The counter is {counter}")
            data.append(counter); data.append(force)
            writer.writerow(data)
            data.clear()

```

8.5 Βαθμονόμηση ζυγαριάς

```
/*
```

```

-----
HX711_ADC
Arduino library for HX711 24-Bit Analog-to-Digital Converter for Weight Scales
Olav Kallhovd sept2017
-----

```

```
*/
```

```
/*
```

This example file shows how to calibrate the load cell and optionally store the calibration value in EEPROM, and also how to change the value manually.
The result value can then later be included in your project sketch or fetched from EEPROM.

To implement calibration in your project sketch the simplified procedure is as follow:

```

LoadCell.tare();
//place known mass
LoadCell.refreshDataSet();
float newCalibrationValue = LoadCell.getNewCalibration(known_mass);

```

```

*/

#include <HX711_ADC.h>
#if defined(ESP8266) || defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif

//pins:
const int HX711_dout = 4; //mcu > HX711 dout pin
const int HX711_sck = 5; //mcu > HX711 sck pin

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_eeepromAdress = 0;
unsigned long t = 0;

void setup() {
  Serial.begin(57600); delay(10);
  Serial.println();
  Serial.println("Starting...");

  LoadCell.begin();
  //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to positive
  unsigned long stabilizingtime = 2000; // preciscion right after power-up can be improved by
  adding a few seconds of stabilizing time
  boolean _tare = true; //set this to false if you don't want tare to be performed in the next
  step
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag() || LoadCell.getSignalTimeoutFlag()) {
    Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
    while (1);
  }
  else {
    LoadCell.setCalFactor(1.0); // user set calibration value (float), initial value 1.0 may be used
    for this sketch
    Serial.println("Startup is complete");
  }
  while (!LoadCell.update());
  calibrate(); //start calibration procedure
}

void loop() {
  static boolean newDataReady = 0;
  const int serialPrintInterval = 0; //increase value to slow down serial print activity

  // check for new data/start next conversion:
  if (LoadCell.update()) newDataReady = true;

```

```

// get smoothed value from the dataset:
if (newDataReady) {
  if (millis() > t + serialPrintInterval) {
    float i = LoadCell.getData();
    Serial.print("Load_cell output val: ");
    Serial.println(i);
    newDataReady = 0;
    t = millis();
  }
}

// receive command from serial terminal
if (Serial.available() > 0) {
  char inByte = Serial.read();
  if (inByte == 't') LoadCell.tareNoDelay(); //tare
  else if (inByte == 'r') calibrate(); //calibrate
  else if (inByte == 'c') changeSavedCalFactor(); //edit calibration value manually
}

// check if last tare operation is complete
if (LoadCell.getTareStatus() == true) {
  Serial.println("Tare complete");
}

}

void calibrate() {
  Serial.println("***");
  Serial.println("Start calibration:");
  Serial.println("Place the load cell on a level stable surface.");
  Serial.println("Remove any load applied to the load cell.");
  Serial.println("Send 't' from serial monitor to set the tare offset.");

  boolean _resume = false;
  while (_resume == false) {
    LoadCell.update();
    if (Serial.available() > 0) {
      if (Serial.available() > 0) {
        char inByte = Serial.read();
        if (inByte == 't') LoadCell.tareNoDelay();
      }
    }
    if (LoadCell.getTareStatus() == true) {
      Serial.println("Tare complete");
      _resume = true;
    }
  }
}

```

```
Serial.println("Now, place your known mass on the loadcell.");
Serial.println("Then send the weight of this mass (i.e. 100.0) from serial monitor.");
```

```
float known_mass = 0;
_resume = false;
while (_resume == false) {
  LoadCell.update();
  if (Serial.available() > 0) {
    known_mass = Serial.parseFloat();
    if (known_mass != 0) {
      Serial.print("Known mass is: ");
      Serial.println(known_mass);
      _resume = true;
    }
  }
}
```

```
LoadCell.refreshDataSet(); //refresh the dataset to be sure that the known mass is
measured correct
float newCalibrationValue = LoadCell.getNewCalibration(known_mass); //get the new
calibration value
```

```
Serial.print("New calibration value has been set to: ");
Serial.print(newCalibrationValue);
Serial.println(", use this as calibration value (calFactor) in your project sketch.");
Serial.print("Save this value to EEPROM adress ");
Serial.print(calVal_eepromAdress);
Serial.println("? y/n");
```

```
_resume = false;
while (_resume == false) {
  if (Serial.available() > 0) {
    char inByte = Serial.read();
    if (inByte == 'y') {
#if defined(ESP8266) || defined(ESP32)
      EEPROM.begin(512);
#endif
      EEPROM.put(calVal_eepromAdress, newCalibrationValue);
#if defined(ESP8266) || defined(ESP32)
      EEPROM.commit();
#endif
      EEPROM.get(calVal_eepromAdress, newCalibrationValue);
      Serial.print("Value ");
      Serial.print(newCalibrationValue);
      Serial.print(" saved to EEPROM address: ");
      Serial.println(calVal_eepromAdress);
      _resume = true;
    }
  }
}
```

```

    }
    else if (inByte == 'n') {
        Serial.println("Value not saved to EEPROM");
        _resume = true;
    }
}
}

Serial.println("End calibration");
Serial.println("****");
Serial.println("To re-calibrate, send 'r' from serial monitor.");
Serial.println("For manual edit of the calibration value, send 'c' from serial monitor.");
Serial.println("****");
}

void changeSavedCalFactor() {
    float oldCalibrationValue = LoadCell.getCalFactor();
    boolean _resume = false;
    Serial.println("****");
    Serial.print("Current value is: ");
    Serial.println(oldCalibrationValue);
    Serial.println("Now, send the new value from serial monitor, i.e. 696.0");
    float newCalibrationValue;
    while (_resume == false) {
        if (Serial.available() > 0) {
            newCalibrationValue = Serial.parseFloat();
            if (newCalibrationValue != 0) {
                Serial.print("New calibration value is: ");
                Serial.println(newCalibrationValue);
                LoadCell.setCalFactor(newCalibrationValue);
                _resume = true;
            }
        }
    }
    _resume = false;
    Serial.print("Save this value to EEPROM adress ");
    Serial.print(calVal_eeepromAdress);
    Serial.println("? y/n");
    while (_resume == false) {
        if (Serial.available() > 0) {
            char inByte = Serial.read();
            if (inByte == 'y') {
                #if defined(ESP8266) || defined(ESP32)
                    EEPROM.begin(512);
                #endif
                EEPROM.put(calVal_eeepromAdress, newCalibrationValue);
                #if defined(ESP8266) || defined(ESP32)

```



```
EEPROM.commit();
#endif
EEPROM.get(calVal_eeepromAdress, newCalibrationValue);
Serial.print("Value ");
Serial.print(newCalibrationValue);
Serial.print(" saved to EEPROM address: ");
Serial.println(calVal_eeepromAdress);
_resume = true;
}
else if (inByte == 'n') {
  Serial.println("Value not saved to EEPROM");
  _resume = true;
}
}
Serial.println("End change calibration value");
Serial.println("***");
}
```

9 Βιβλιογραφία

- [1] The Advantages of Using Fatigue Testing Machine. Horizon Technology. Ανακτήθηκε από ιστοσελίδα: <https://www.horizontester.com/the-advantages-of-using-the-fatigue-testing-machine-a.html>
- [2] Shawki, Galal SA. "A review of fatigue testing machines." (1990).
- [3] Rajesh, S., and N. Saravanan. "Design and Fabrication of Low Cost Fatigue Test Rig." *Imperial Journal of Interdisciplinary Research* 2.10 (2016): 684-688.
- [4] Vincent, Martin K., Vipin Varghese, and Suneeth Sukumaran. "Fabrication and analysis of fatigue testing machine." *Int. J. Eng. Sci* 5 (2016): 15-19.
- [5] Kulkarni, Pritam V., P. J. Sawant, and V. V. Kulkarni. "Design and development of plane bending fatigue testing machine for composite material." *Materials Today: Proceedings* 5.5 (2018): 11563-11568.
- [6] Ikechukwu, Gbasouzor Austine, Okeke Ogochukwu Clementina, and Chima Lazarus Onyebuchi. "Design and characterization of a fatigue testing machine." *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 1. 2013.
- [7] McAlorum, Jack, et al. "Design and demonstration of a low-cost small-scale fatigue testing machine for multi-purpose testing of materials, sensors and structures." *Machines* 6.3 (2018): 30.
- [8] Landge, Sagar Madhukar, et al. "Design and development of bending fatigue testing machine for composite materials." *Int Res J Eng Technol (IRJET)* 5.04 (2018).
- [9] Kattimani, Mohammed Asif, et al. "Design and fabrication of fatigue testing machine." *International Journal of Scientific Research in Science, Engineering and Technology* 7.1 (2020): 295-304.
- [10] Hadi, S., et al. "Testing of the ability of fatigue test machine prototype and fatigue test for nylon and cast iron specimens." *IOP Conference Series: Materials Science and Engineering*. Vol. 732. No. 1. IOP Publishing, 2020.
- [11] Nagabhooshanam, N., S. Baskar, and P. K. Nagarajan. "Design and fabrication of fatigue test rig and preliminary investigation on flax composite beam." *Materials Today: Proceedings* 5.5 (2018): 11771-11779.
- [12] Wang, S. Y., et al. "A computer vision based machine learning approach for fatigue crack initiation sites recognition." *Computational materials science* 171 (2020): 109259.
- [13] Singh, Swarit Anand, and Kaushal A. Desai. "Automated surface defect detection framework using machine vision and convolutional neural networks." *Journal of Intelligent Manufacturing* 34.4 (2023): 1995-2011.
- [14] Singh, Swarit Anand, Aitha Sudheer Kumar, and K. A. Desai. "Comparative assessment of common pre-trained CNNs for vision-based surface defect detection of machined components." *Expert Systems with Applications* 218 (2023): 119623.
- [15] Park, Je-Kang, et al. "Machine learning-based imaging system for surface defect inspection." *International Journal of Precision Engineering and Manufacturing-Green Technology* 3 (2016): 303-310.
- [16] Weimer, Daniel, Bernd Scholz-Reiter, and Moshe Shpitalni. "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection." *CIRP annals* 65.1 (2016): 417-420.

- [17] Xu, Liang, et al. "A weakly supervised surface defect detection based on convolutional neural network." *IEEE Access* 8 (2020): 42285-42296.
- [18] Sundaram, Sarvesh, and Abe Zeid. "Artificial intelligence-based smart quality inspection for manufacturing." *Micromachines* 14.3 (2023): 570.
- [19] Tong, Tong, et al. "Crack identification for bridge condition monitoring using deep convolutional networks trained with a feedback-update strategy." *Maintenance, Reliability and Condition Monitoring* 1.2 (2021): 37-51.
- [20] Zhang, Dehua, et al. "An efficient lightweight convolutional neural network for industrial surface defect detection." *Artificial Intelligence Review* 56.9 (2023): 10651-10677.
- [21] Şik, A. Y. D. I. N., et al. "The design of fatigue strength machine being one of the methods for determining the mechanical properties of the materials used in the industry." *Gazi University Journal of Science Part A: Engineering and Innovation* 5.2 (2018): 79-88.
- [22] Xiong, Xin, Li Huang, and Jinrong Tian. "Application of Image Processing Technology in Building Surface Crack Detection." *IOP Conference Series: Materials Science and Engineering*. Vol. 612. No. 3. IOP Publishing, 2019.
- [23] Ghalyan, Najah F., Ibrahim F. Ghalyan, and Asok Ray. "Modeling of microscope images for early detection of fatigue cracks in structural materials." *The International Journal of Advanced Manufacturing Technology* 104 (2019): 3899-3913.
- [24] Alqahtani, Hassan, Skanda Bharadwaj, and Asok Ray. "Classification of fatigue crack damage in polycrystalline alloy structures using convolutional neural networks." *Engineering Failure Analysis* 119 (2021): 104908.
- [25] Chlap, Phillip, et al. "A review of medical image data augmentation techniques for deep learning applications." *Journal of Medical Imaging and Radiation Oncology* 65.5 (2021): 545-563.
- [26] (2021), Creep, Drift and Your Load Cell. Adam Equipment. Ανακτήθηκε από ιστοσελίδα: <https://adamequipment.co.uk/content/post/creep-and-drift>
- [27] How to Test a Load Cell? Omega Engineering. Ανακτήθηκε από ιστοσελίδα: <https://www.omega.com/en-us/resources/load-cell-troubleshooting#>
- [28] Indrek Luuk, 50kg Load Cells with HX711 and Arduino. 4x, 2x, 1x Diagrams. circuitjournal.com Ανακτήθηκε από ιστοσελίδα: <https://circuitjournal.com/50kg-load-cells-with-HX711>
- [29] Abid All Awan (2022), A Complete Guide to Data Augmentation. Datacamp. Ανακτήθηκε από ιστοσελίδα: <https://www.datacamp.com/tutorial/complete-guide-data-augmentation>
- [30] Deval Shah (2017), Activation Functions. Medium. Ανακτήθηκε από ιστοσελίδα: <https://medium.com/@devalshah1619/activation-functions-in-neural-networks-58115cda9c96>
- [31] <https://medium.com/@devalshah1619/activation-functions-in-neural-networks-58115cda9c96>
- [32] Swapnil Vishwakarma (2023), How to Understand Sigmoid Function in Artificial Neural Networks? Analytics Vidhya. Ανακτήθηκε από ιστοσελίδα: <https://www.analyticsvidhya.com/blog/2023/01/why-is-sigmoid-function-important-in-artificial-neural-networks/>
- [33] (2023), What is Cost Function in Neural Network? Saturn Cloud. Ανακτήθηκε από ιστοσελίδα: <https://saturncloud.io/blog/what-is-cost-function-in-neural-network/>

- [34] Talha Quddoos (2021), Neural Network Basics: Loss and Cost Functions. Medium. Ανακτήθηκε από ιστοσελίδα: <https://medium.com/artificialis/neural-network-basics-loss-and-cost-functions-9d089e9de5f8>
- [35] Saily Shah (2022), Convolutional Neural Network: An Overview. Analytics Vidhya. Ανακτήθηκε από ιστοσελίδα: <https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/>
- [36] Great Learning Team (2024), What is Rectified Linear Unit (ReLU)? | Introduction to ReLU Activation Function. Great Learning. Ανακτήθηκε από ιστοσελίδα: <https://www.mygreatlearning.com/blog/relu-activation-function/>