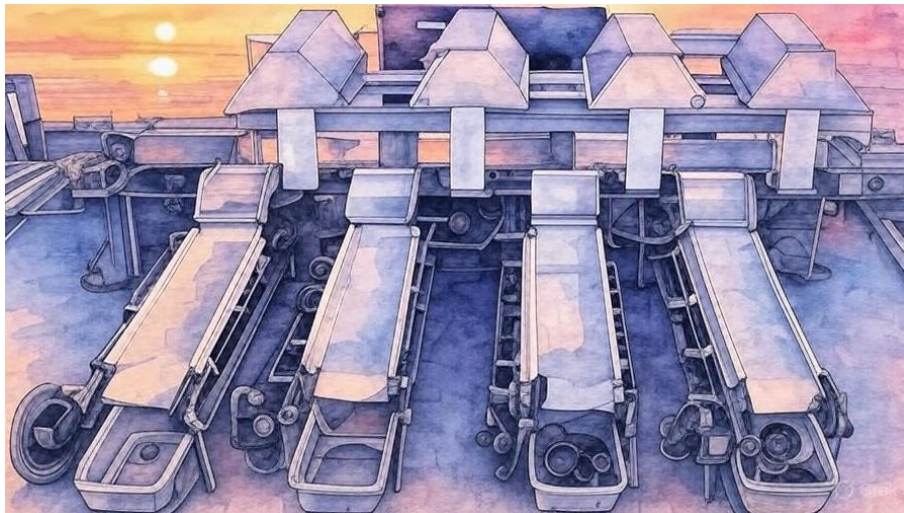




ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
«Δημιουργία Εκπαιδευτικού Μοντέλου Εργοστασίου  
Ανακύκλωσης»



Των φοιτητών  
Παπαζήση Γρηγορίου 19/2023  
Παπαζήση Δημητρίου 20/2023

Επιβλέπων  
Ευστάθιος Κασδερίδης

Ημερομηνία 14/09/2025

Τίτλος Δ.Ε: Δημιουργία Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης

Κωδικός Δ.Ε.: 24287

Όνοματεπώνυμο φοιτητών: Παπαζήσης Γρηγόριος, Παπαζήσης Δημήτριος

Όνοματεπώνυμο εισηγητή: Ευστάθιος Κασδερίδης

Ημερομηνία ανάληψης Δ.Ε: 28/10/2024

Ημερομηνία περάτωσης Δ.Ε: 14/09/2025

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

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

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

*«Αφιερώνετε στις οικογένειες μας και στους μαθητές μας »*

## Πρόλογος

Κατά την επιλογή της διπλωματικής μας εργασίας, συνεργαστήκαμε στενά με τον επιβλέποντα καθηγητή μας, εξετάζοντας διάφορα θέματα με σκοπό να καταλήξουμε σε ένα αντικείμενο που να συνδέεται με τη διδακτέα ύλη του Μεταπτυχιακού μας. Παράλληλα, θέλαμε η εργασία μας να έχει πρακτική αξία και να μπορεί να αξιοποιηθεί στο επαγγελματικό μας περιβάλλον που είναι η πρωτοβάθμια και δευτεροβάθμια εκπαίδευση. Το θέμα που επιλέξαμε μας έδωσε την ευκαιρία να ασχοληθούμε σε βάθος με σύγχρονους τομείς της Πληροφορικής, όπως η Μηχανική Μάθηση η Ρομποτική και η εκπαίδευση STEM, καθώς ενσωματώνει στοιχεία επιστήμης, τεχνολογίας, μηχανικής και μαθηματικών σε μια πρακτική μαθησιακή εμπειρία, που εξελίσσονται ραγδαία και βρίσκουν εφαρμογή σε πολλούς κλάδους, συμπεριλαμβανομένης της εκπαίδευσης, η οποία αποτελεί και τον βασικό άξονα της μελέτης μας.

Κατά τη διάρκεια της έρευνας και της ανάπτυξης της εργασίας μας, αποκτήσαμε πολύτιμες γνώσεις τόσο στον τομέα του υλικού όσο και σε επίπεδο λογισμικού, καθώς και στις αρχές επικοινωνίας μεταξύ συστημάτων. Η διαδικασία αυτή μας έδωσε τη δυνατότητα να μετατρέψουμε τις θεωρητικές μας γνώσεις σε πράξη, οδηγώντας στην επιτυχή κατασκευή του Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης. Αυτή η εμπειρία αποδείχθηκε ιδιαίτερα χρήσιμη, καθώς μας βοήθησε να εφαρμόσουμε τις δεξιότητές μας σε ένα πραγματικό έργο, ενισχύοντας την κατανόηση και την αυτοπεποίθησή μας για μελλοντικές επαγγελματικές προκλήσεις και χρήση τους μέσα στην τάξη με γνώμονα την εκπαιδευτική διαδικασία.

## Περίληψη

Η παρούσα διπλωματική πραγματεύεται τη «Δημιουργία Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης». Στόχος της διπλωματικής μας είναι ο σχεδιασμός και η υλοποίηση ενός, μικρής κλίμακας, αυτοματοποιημένου μοντέλου εργοστασίου ανακύκλωσης απορριμμάτων, κατάλληλου για διδασκαλία βασισμένη σε Project τόσο στην πρωτοβάθμια όσο και στη δευτεροβάθμια εκπαίδευση. Το προτεινόμενο μοντέλο θα προσομοιώνει τις βασικές διαδικασίες παραλαβής και ταξινόμησης των απορριμμάτων, περιλαμβάνοντας τουλάχιστον ένα σύστημα με ιμάντες μεταφοράς και τη δυνατότητα ταξινόμησης υλικών όπως χαρτί, γυαλί, μέταλλο (π.χ. αλουμίνιο), πλαστικό και 'άλλο' (π.χ. οργανικά υλικά) σε επιμέρους αποθηκευτικούς χώρους. Η ταξινόμηση θα βασίζεται σε μέθοδο οπτικής αναγνώρισης, με τη δημιουργία ενός μοντέλου ταξινόμησης μέσω αλγορίθμων Μηχανικής Μάθησης, χρησιμοποιώντας φωτογραφίες των υλικών ως σύνολο δεδομένων. Ιδιαίτερη έμφαση δίνεται στη χρήση μικρό-αντικειμένων που είναι εύκολα προσβάσιμα ή κατασκευάσιμα στην τάξη με ελάχιστους πόρους. Κάμερες θα τοποθετηθούν σε έναν κεντρικό ιμάντα μεταφοράς για την αναγνώριση και άλλοι ιμάντες και μηχανισμοί θα αναλαμβάνουν την αφαίρεση και μεταφορά των αναγνωρισμένων αντικειμένων στους κατάλληλους αποθηκευτικούς χώρους. Όλο το σύστημα θα ελέγχεται από έναν σταθμό ελέγχου (PC, Laptop, ή Raspberry pi 5) όπου κατάλληλα προγράμματα θα συντονίζουν τις διαδικασίες παραλαβής, μεταφοράς, ταξινόμησης και αποθήκευσης, με τη χρήση αισθητήρων, ενεργοποιητών και κινητήρων για την παροχή στοιχείων εισόδου και εξόδου, λειτουργώντας αυτοματοποιημένα χωρίς ανθρώπινη παρέμβαση.

Το εγχείρημα αυτό έχει σημαντική αξία στο πλαίσιο της εκπαίδευσης της πληροφορικής και την άμεση σχέση με την εκπαίδευση STEM και την ρομποτική, καθώς ενσωματώνει στοιχεία επιστήμης, τεχνολογίας, μηχανικής και μαθηματικών σε μια πρακτική μαθησιακή εμπειρία. Επιπλέον, συμβάλλει στην ευαισθητοποίηση των μαθητών σε περιβαλλοντικά ζητήματα, όπως η διαχείριση απορριμμάτων, οι διαδικασίες ανακύκλωσης και η βιωσιμότητα. Τέλος, η προσέγγιση της μάθησης μέσω Project ενισχύει την κριτική σκέψη, την ικανότητα επίλυσης προβλημάτων και τη συνεργασία μεταξύ των μαθητών.

# «Development of a Recycling Factory Educational Model»

Papazisis Grigorios

Papazisis Dimitrios

## **Abstract**

The present thesis deals with the topic of "Development of a Recycling Factory Educational Model". The aim of our thesis is the design and implementation of a small-scale, automated model of a waste recycling plant, suitable for project-based teaching in both primary and secondary education. The proposed model will simulate the basic processes of waste collection and sorting, including at least a conveyor belt system and the possibility of sorting materials such as paper, glass, metal (e.g. aluminum), plastic and 'other' (e.g. organic materials) into individual storage areas. The classification will be based on a visual recognition method, with a classification model created through Machine Learning techniques, using photographs of the materials as a dataset. Particular emphasis is placed on the use of micro-objects that are easily accessible or manufacturable in the classroom with minimal resources. Cameras will be mounted on a central conveyor belt for identification and other belts and mechanisms will take care of removing and transporting the identified objects to appropriate storage areas. The whole system will be controlled by a control station (PC, Laptop, or Raspberry pi 5) where appropriate programs will coordinate the processes of picking, transport, sorting and storage, using sensors and motors to provide input and output data, operating automatically without human intervention.

This project has significant value in the context of computer science education and its direct relevance to STEM education, as it integrates elements of science, technology, engineering and mathematics into a hands-on learning experience. In addition, it helps to raise student awareness of environmental issues such as waste management, recycling processes and sustainability. Finally, the project-based approach to learning enhances critical thinking, problem-solving skills and cooperation among students.

## **Ευχαριστίες**

Ευχαριστούμε θερμά τις οικογένειές μας για όλη την ηθική στήριξη κατά τη διάρκεια του ΠΜΣ. Επίσης, ευχαριστούμε τον επιβλέποντα καθηγητή κ. Στάθη Κασδερίδη για την αγαστή συνεργασία και τις συμβουλές του σε τεχνικά θέματα.

# Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract .....	vi
Ευχαριστίες .....	vii
Περιεχόμενα .....	viii
Κατάλογος Σχημάτων - Εικόνων .....	xi
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Φυσικά Μοντέλα προσομοίωσης στην εκπαίδευση.....	1
1.1 Εισαγωγή .....	1
1.2 Υπάρχοντα Εκπαιδευτικά Μοντέλα Εργοστασίων Ανακύκλωσης και Συστημάτων Διαχείρισης Απορριμμάτων(Hands-on εργαστήρια και μικρογραφίες εργοστασίων) .....	2
1.2.1 Ψηφιακές προσομοιώσεις και gamification.....	2
1.2.2 Διαθεματικά STEM προγράμματα .....	3
1.2.3 Σύνδεση με τεχνητή νοημοσύνη και μηχανική μάθηση .....	3
1.3 Επίλογος.....	3
Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο.....	4
2.1 Εισαγωγή .....	4
2.2 Η εκπαίδευση STEM. ....	4
2.3 Ρομποτική και Αυτοματοποίηση στην Εκπαίδευση .....	5
2.4 Μηχανική Μάθηση .....	5
2.5 Ενδεικτικοί μέθοδοι Μηχανικής Μάθησης.....	6
2.5.1 Επιβλεπόμενη Μάθηση (Supervised Learning).....	6
2.5.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning).....	6
2.5.3 Μάθηση με ενίσχυση (Reinforcement Learning) .....	6
2.6 Μοντέλα Μηχανικής Μάθησης .....	7
2.7 Νευρωνικά Δίκτυα (Neural Networks - NNs) .....	7
2.8 Βαθιά Μάθηση (Deep Learning) .....	7
2.9 Συνελκτικά Νευρωνικά Δίκτυα (CNNs).....	8
2.10 Υπολογιστική Όραση και Μηχανική Μάθηση για Αναγνώριση και Ταξινόμηση Αντικειμένων.....	11
2.11 Γλώσσα υλοποίησης Python.....	11

2.12	Web based Πλατφόρμες εκπαιδευτικού χαρακτήρα.....	12
2.13	Μικροελεγκτές (Arduino Uno) και Υπολογιστές (PC) σε Εκπαιδευτικά Έργα .....	12
2.14	Επίλογος .....	12
Κεφάλαιο 3ο: Σχεδίαση και Κατασκευή του Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης 14		
3.1	Εισαγωγή .....	14
3.2	Υλικά Κατασκευής βασικού ταινιόδρομου και των ταινιόδρομων εξόδου.....	14
3.3	Υλοποίηση κατασκευής – συναρμολόγησης Υλικού βασικού ταινιόδρομου και των ταινιόδρομων εξόδου.....	16
3.4	Βήματα κατασκευής βασικού ταινιόδρομου:.....	16
3.5	Βήματα κατασκευής ταινιόδρομων εξόδου .....	17
3.6	Λογισμικό ελέγχου καμερών, servo motor, stepper motor, βασικού ταινιόδρομου και των ταινιόδρομων εξόδου.....	18
3.7	Έλεγχος ESP32-CAM.....	18
3.8	Έλεγχος Stepper Motor – Servo Motor Βασικού ταινιόδρομου .....	22
3.9	Έλεγχος Stepper Motor ταινιόδρομων εξόδου.....	24
3.10	Χειρισμός του βασικού ταινιόδρομου και των ταινιόδρομων εξόδου.....	26
3.11	Επίλογος .....	31
Κεφάλαιο 4ο: Δημιουργία μοντέλου Υπολογιστικής Όρασης για κατηγοριοποίηση αντικειμένων.33		
4.1	Εισαγωγή .....	33
4.2	Συλλογή δεδομένων.....	33
4.3	Προετοιμασία του Setup .....	33
4.4	Διαδικασία Λήψης Εικόνων.....	34
4.5	Προετοιμασία δεδομένων .....	35
4.6	Δημιουργία του Εξειδικευμένου Μοντέλου .....	36
4.7	Εκπαίδευση του μοντέλου .....	38
4.8	Οπτικοποίηση Ακρίβειας και Απώλειας του Εκπαιδευμένου Μοντέλου .....	39
4.9	Προβλέψεις του εκπαιδευμένου μοντέλου .....	42
4.10	Αναφορά των αποτελεσμάτων και Confusion Matrix .....	42
4.11	Επίλογος .....	43
Κεφάλαιο 5ο: Πειράματα πάνω στην κατασκευή και Ολοκλήρωση – Αποτελέσματα .....44		
5.1	Εισαγωγή .....	44
5.2	Περιγραφή Κατασκευής .....	44
5.3	Φάσεις Λειτουργίας του Συστήματος .....	45
5.4	Φάση 1: Σύλληψη εικόνας (Frame Capture).....	45

5.5	Φάση 2: Αναγνώριση υλικού (Inference) .....	46
5.6	Φάση 3: Ενεργοποίηση εξόδου (Servo Control).....	49
5.7	Φάση 4: Συνεχής Παρακολούθηση και Διόρθωση .....	51
5.8	Δοκιμές και Αποτελέσματα .....	52
5.9	Συμπεράσματα Φάσεων.....	59
5.10	Επίλογος .....	59
Κεφάλαιο 6ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	60
6.1	Γενική ανασκόπηση.....	60
6.2	Συνοπτική περιγραφή.....	60
6.3	Κύρια προβλήματα και επίλυσή τους .....	60
6.4	Μελλοντικές βελτιώσεις .....	61
6.5	Εφαρμογή στην τάξη .....	61
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>62</b>

## Κατάλογος Σχημάτων - Εικόνων

Εικόνα 2.9.4 : Τυπική αρχιτεκτονική CNN δικτύου .....	11
Εικόνα 3.4.1 : Κάτοψη Ταινιόδρομου .....	16
Εικόνα 3.4.2 : Κάτοψη τοποθέτησης βηματικού κινητήρα.....	16
Εικόνα 3.4.3 : Κεντρικός ταινιόδρομος.....	17
Εικόνα 3.5 : Ταινιόδρομοι εξόδου .....	18
Εικόνα 3.7 : Κώδικας WebServer esp32-cam .....	22
Εικόνα 3.8 : Κώδικας ελέγχου stepper motor και servo motor κεντρικού ταινιόδρομου .....	24
Εικόνα 3.9. : Κώδικας ελέγχου stepper motor και servo motor ταινιόδρομων εξόδου .....	26
Εικόνα 3.10 : Κώδικας χειρισμού βασικού ταινιόδρομου .....	31
Εικόνα 4.3 : Κάλυμμα κατασκευής .....	33
Εικόνα 4.5 : Κώδικας προετοιμασίας δεδομένων .....	36
Εικόνα 4.6.1 : Κώδικας CNN μοντέλου .....	37
Εικόνα 4.6.2 : Οπτικοποίηση μοντέλου .....	37
Εικόνα 4.7 : Εκπαίδευση μοντέλου .....	39
Εικόνα 4.8.1 : Κώδικας οπτικοποίησης Ακριβείας .....	40
Εικόνα 4.8.2 : Κλάση glass .....	40
Εικόνα 4.8.3 : Κλάση metal .....	40
Εικόνα 4.8.4 : Κλάση paper .....	41
Εικόνα 4.8.5 : Κλάση plastic .....	41
Εικόνα 4.9 : Αξιολόγηση μοντέλου στο test set .....	42
Εικόνα 4.10.1: Κώδικας confusion matrix και classification report .....	43
Εικόνα 4.10.2 : Οπτικοποίηση confusion matrix και classification report .....	43
Εικόνα 5.2.1: Ολοκλήρωση του ταινιόδρομου .....	44
Εικόνα 5.2.2: Επιπρόσθετο στέγαστρο .....	44
Εικόνα 5.4.1: Κώδικας σύλλυψης εικόνας .....	45
Εικόνα 5.5.1: Κώδικας αναγνώρισης υλικού .....	47
Εικόνα 5.5.2: Live inference binary μοντέλου .....	48
Εικόνα 5.5.3: Live inference binary μοντέλου .....	48
Εικόνα 5.6.1: Κώδικας ενεργοποίησης εξόδου .....	50
Εικόνα 5.7.1: Κώδικας παρακολούθησης σωστής λειτουργίας .....	51
Εικόνα 5.8.1: Σύγκριση των μετρικών του test.....	52
Εικόνα 5.8.2: Σύγκριση των μοντέλων στο Accuracy .....	53
Εικόνα 5.8.3: Συγκριση Macro F1 - score .....	53
Εικόνα 5.8.4: Σύγκριση Macro Avg F1-score.....	53
Εικόνα 5.8.5: Σύγκριση validation accuracy και validation Loss .....	54
Εικόνα 5.8.6: Σύγκριση F1-score ανά κλάση.....	57
Εικόνα 5.8.7: Συγκριση Meen, precision , recall και F1 .....	57
Εικόνα 5.8.8: Σύγκριση των μετρικών του test set .....	57

## Κατάλογος Πινάκων

Πίνακας 3.2: Υλικά κατασκευής .....	14
-------------------------------------	----

Πίνακας 5.8.1: Μετρικές απόδοσης στο test set .....	54
Πίνακας 5.8.2: Macro F1-score .....	55
Πίνακας 5.8.3: Στατιστικά μέτρα απόδοσης .....	58
Πίνακας 5.8.4: Μετρικές απόδοσης στο test set .....	58

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
RPi	Raspberry Pi
DP	Deep Learning
CNN	Convolutional Neural Network
AI	Artificial Intelligence
API	Application Programming Interface
ML	Machine Learning
PBL	Project- based Learning
IOT	Internet Of Things
MQTT	Message Queuing Telemetry Transport



## Κεφάλαιο 1ο: Φυσικά Μοντέλα προσομοίωσης στην εκπαίδευση

### 1.1 Εισαγωγή

Η εκπαίδευση γύρω από την ανακύκλωση και τη διαχείριση απορριμμάτων έχει ενισχυθεί σημαντικά τα τελευταία χρόνια, όχι μόνο σε πανεπιστημιακό επίπεδο αλλά και σε πρωτοβάθμια και δευτεροβάθμια εκπαίδευση [2]. Ο συνδυασμός της περιβαλλοντικής εκπαίδευσης με τη χρήση εργαλείων STEM (Science, Technology, Engineering, Mathematics) δίνει στους μαθητές την ευκαιρία να πειραματιστούν με μοντέλα εργοστασίων ανακύκλωσης, προσομοιώσεις ρομποτικών συστημάτων και βιωματικές δραστηριότητες [1,4]. Μέσα από αυτά τα μοντέλα, οι μαθητές αναπτύσσουν οικολογική συνείδηση, καλλιεργούν τεχνολογικές δεξιότητες και κατανοούν τη σημασία της κυκλικής οικονομίας ήδη από μικρή ηλικία [13].

Το παρόν έργο φέρει μια σειρά κατασκευών που πρέπει να υλοποιηθούν με το ελάχιστο δυνατόν κόστος υλικού και λογισμικού που υπάρχουν σε ένα εργαστήριο πληροφορικής ή και μπορούν να αποκτηθούν από συσκευές που είναι για ανακύκλωση – απόσυρση, ενισχύοντας την κατασκευή με μηχανικά μέρη που σε άλλη περίπτωση θα είχαν καταστραφεί [9,10]. Το σύστημα λοιπόν θα έχει έναν ταινιόδρομο μικρής κλίμακας, τέσσερις κάμερες εποπτείας των υλικών και αναγνώρισης τους με ένα εξειδικευμένο CNN (Convolutional Neural Network) μοντέλο μηχανικής μάθησης δίνοντας τις εντολές σε τέσσερις εξόδους στον ταινιόδρομο για να γίνει η διαλογή και η ταξινόμηση των υλικών σύμφωνα με την κατηγορία τους [5,8]. Η έξοδος των υλικών από τον κύριο ταινιόδρομο θα γίνεται με χρήση servo κινητήρων που θα ελέγχονται από ένα Arduino Uno εφόσον γίνει η αναγνώριση τους από την ανάλογη κάμερα και στην ανάλογη έξοδο [11]. Η κατασκευή θα ελέγχεται πλήρως από ένα σταθμό άκρης (PC, Laptop, ή και Raspberry Pi 5) με κώδικα σε Python και θα προσομοιώνει όσο το δυνατόν την λειτουργία ενός πραγματικού εργοστασίου ανακύκλωσης [12].

Το πρωτότυπο έργο έχει δημιουργηθεί με τέτοιο τρόπο ώστε να μπορεί να τροποποιείται σύμφωνα με τις ανάγκες της εκάστοτε βαθμίδας εκπαίδευσης. Σταδιακά μπορεί να συναρμολογηθεί όπως και τα αντίστοιχα ρομποτικά συστήματα και κατασκευές που διανέμει στις βαθμίδες εκπαίδευσης το Υπουργείο Παιδείας [3]. Τα υλικά έχουν αγοραστεί για τις ανάγκες της παρούσας διπλωματικής και την δημιουργία του πρωτότυπου έργου.

Τα μεγάλα βιομηχανικά συγκροτήματα εστιάζουν στην έρευνα με βάση την τεχνητή νοημοσύνη μέσω της υπολογιστικής όρασης για τη διάκριση αντικειμένων και τη λήψη αποφάσεων σε πραγματικό χρόνο που είναι ένα από τα σημαντικά πράγματα στον τομέα της βιομηχανίας για τη βελτίωση της αποτελεσματικότητας τους [6,7]. Η χρήση αλγορίθμων βαθιάς μάθησης επιτρέπουν σε αυτοματοποιημένα αυτόνομα συστήματα να ελέγχουν, να αναγνωρίζουν και τελικά να αποφασίζουν σε πραγματικό χρόνο [8]. Η ενσωμάτωση αυτών των τεχνολογιών στις βιομηχανικές εφαρμογές έχει φέρει επανάσταση, αυξάνοντας την αποδοτικότητα και μειώνοντας το ανθρώπινο σφάλμα.

Σε αυτό το πλαίσιο, η χρήση οπτικής αναγνώρισης με συστήματα βαθιάς μάθησης, όπως τα CNN δίκτυα, επιτρέπει τη γρήγορη και αξιόπιστη αναγνώριση αντικειμένων [5]. Η δημιουργία ενός φυσικού μοντέλου προσομοίωσης εργοστασίου ανακύκλωσης σε συνδυασμό με εκπαίδευση STEM και ρομποτική, μπορεί να οδηγήσει σε συγκεκριμένα μαθησιακά αποτελέσματα. Οι μαθητές μπορούν να κατανοήσουν τους διαφορετικούς τύπους ανακυκλώσιμων υλικών και τις ιδιότητές τους [8]. Μαθαίνουν για τα στάδια που εμπλέκονται σε μια τυπική διαδικασία ανακύκλωσης (παραλαβή, ταξινόμηση, επεξεργασία) [10]. Αναπτύσσουν συνείδηση για τη σημασία της ανακύκλωσης και της διαχείρισης απορριμμάτων [2]. Τέλος, αποκτούν βασικές γνώσεις για την αυτοματοποίηση, τους

αισθητήρες, τους κινητήρες και τους αλγορίθμους σε προγραμματιστικά περιβάλλοντα καθώς και δεξιότητες που είναι χρήσιμες για τη μηχανική μάθηση [12].

Τα φυσικά μοντέλα, ειδικά σε ένα πλαίσιο μάθησης μέσω Project, γεφυρώνουν το χάσμα μεταξύ της θεωρητικής γνώσης και της πρακτικής εφαρμογής, οδηγώντας σε μια βαθύτερη και πιο ουσιαστική κατανόηση στην εκπαίδευση, των τεχνολογικών βάσεων των εφαρμογών [15].

### **1.2 Υπάρχοντα Εκπαιδευτικά Μοντέλα Εργοστασίων Ανακύκλωσης και Συστημάτων Διαχείρισης Απορριμμάτων (Hands-on εργαστήρια και μικρογραφίες εργοστασίων)**

Σε πολλές χώρες, τα σχολεία υιοθετούν μικρής κλίμακας ρομποτικά συστήματα διαλογής που μιμούνται τη λειτουργία εργοστασίων ανακύκλωσης. Μέσα από εργαλεία όπως LEGO Mindstorms, VEX Robotics ή Arduino-based sorting systems, οι μαθητές καλούνται να σχεδιάσουν και να προγραμματίσουν αυτοματισμούς για τον διαχωρισμό χαρτιού, πλαστικού και μετάλλου [16]. Τέτοια projects εντάσσονται συχνά σε μαθήματα τεχνολογίας ή φυσικών επιστημών, ενισχύοντας την ενεργή συμμετοχή και την ομαδική εργασία [16].

Παράλληλα, σε εργαστήρια Makerspaces (κατασκευαστικοί χώροι) δημιουργούνται αυτοσχέδιες «γραμμές ανακύκλωσης» με αισθητήρες, κάμερες και ρομποτικούς βραχίονες. Η προσέγγιση αυτή όχι μόνο βοηθά τους μαθητές να δουν πώς λειτουργούν τα πραγματικά εργοστάσια, αλλά επίσης τους εισάγει στις βασικές αρχές Μηχανικής Μάθησης και οπτικής αναγνώρισης με απλοποιημένα datasets κατάλληλα για εκπαιδευτική χρήση [17].

Πολλές εκπαιδευτικές δραστηριότητες βασίζονται στη μάθηση μέσω έργου και περιλαμβάνουν την ταξινόμηση απορριμμάτων [18], την κομποστοποίηση [11] και τη δημιουργία έργων τέχνης από ανακυκλωμένα υλικά [19]. Αυτές οι δραστηριότητες βοηθούν τους μαθητές να αναπτύξουν περιβαλλοντική συνείδηση και να κατανοήσουν την αξία της επαναχρησιμοποίησης και της ανακύκλωσης [2]. Επιπλέον, υπάρχουν διαθέσιμες προσομοιώσεις, όπως το "Metro Waste Authority's Ecoville Simulation" [20], που επιτρέπουν στους μαθητές να εξοικειωθούν με τις διαδικασίες διαχείρισης απορριμμάτων σε ένα διαδραστικό ψηφιακό περιβάλλον. Παράλληλα, υπάρχουν πρακτικές δραστηριότητες κατασκευής μοντέλων, όπως το μοντέλο μονάδας επεξεργασίας λυμάτων [21], που δίνουν έμφαση στην κατανόηση των αρχών της επεξεργασίας και του φιλτραρίσματος. Ένα πιο σύνθετο, αυτοματοποιημένο εκπαιδευτικό μοντέλο είναι το "LEGO Plastic Recycling Plant" [22], το οποίο χρησιμοποιεί στοιχεία LEGO MINDSTORMS για να προσομοιώσει ολόκληρη τη διαδικασία ανακύκλωσης πλαστικού, από την εισαγωγή των απορριμμάτων έως την παραγωγή νέων προϊόντων. Αυτό το μοντέλο αποτελεί ένα εξαιρετικό εργαλείο για την εκμάθηση προγραμματισμού, διαχείρισης εισόδων/εξόδων και επικοινωνίας μεταξύ μικροελεγκτών.

#### **1.2.1 Ψηφιακές προσομοιώσεις και gamification**

Η ένταξη ψηφιακών διδύμων και προσομοιώσεων έχει προσαρμοστεί και στη σχολική πραγματικότητα, μέσω εικονικών εργαστηρίων και παιχνιδοποίησης. Μαθητές γυμνασίου, για παράδειγμα, μπορούν να χρησιμοποιήσουν λογισμικά που προσομοιώνουν τη λειτουργία ενός MRF (Material Recovery Facility) και να πάρουν αποφάσεις για τον τρόπο διαχωρισμού απορριμμάτων σε πραγματικό χρόνο [17,23].

Επιπλέον, εφαρμογές serious games διδάσκουν τις αρχές της ανακύκλωσης μέσα από αλληλεπιδραστικά σενάρια. Έρευνες δείχνουν ότι η παιχνιδοποίηση αυξάνει την περιβαλλοντική ευαισθησία και οδηγεί σε καλύτερη κατανόηση των τεχνολογικών λύσεων που υποστηρίζουν τη διαχείριση απορριμμάτων [24,25].

### 1.2.2 Διαθεματικά STEM προγράμματα

Η βιωματική εκπαίδευση μέσω έργων (project-based learning) είναι ιδιαίτερα διαδεδομένη στη δευτεροβάθμια εκπαίδευση. Προγράμματα Erasmus+ και eTwinning έχουν υλοποιησει σχολικά έργα όπου μαθητές κατασκεύασαν συστήματα αυτόματου διαχωρισμού, μελέτησαν τον κύκλο ζωής των απορριμμάτων και παρουσίασαν λύσεις κυκλικής οικονομίας στην τοπική κοινωνία [26,1].

Επιπλέον, το πρόγραμμα Eco-Schools, που εφαρμόζεται σε περισσότερες από 70 χώρες, ενσωματώνει δραστηριότητες ανακύκλωσης στην καθημερινότητα του σχολείου. Οι μαθητές συχνά δημιουργούν μίνι-«εργοστάσια» ανακύκλωσης στην τάξη, πειραματίζονται με ρομποτικές εφαρμογές και καταγράφουν δεδομένα για να αξιολογήσουν την αποτελεσματικότητα των παρεμβάσεών τους [27], [3,13].

### 1.2.3 Σύνδεση με τεχνητή νοημοσύνη και μηχανική μάθηση

Αν και η Μηχανική Μάθηση θεωρείται συνήθως πανεπιστημιακό αντικείμενο, ορισμένα σχολικά προγράμματα έχουν αρχίσει να εισάγουν απλοποιημένα datasets (π.χ. TrashNet ή τροποποιημένες εκδόσεις του TACO dataset) για να δείξουν στους μαθητές πώς οι υπολογιστές «μαθαίνουν» να ξεχωρίζουν αντικείμενα [28,29]. Με εργαλεία οπτικοποίησης, οι μαθητές βλέπουν πώς ένα απλό CNN μπορεί να ταξινομήσει χαρτί, πλαστικό και μέταλλο, ενισχύοντας τη σύνδεση επιστήμης–τεχνολογίας–περιβάλλοντος [8,17].

Ενώ υπάρχουν διάφοροι εκπαιδευτικοί πόροι, φαίνεται να υπάρχει ένα πιθανό κενό για ένα ολοκληρωμένο, μικρής κλίμακας μοντέλο που να ενσωματώνει αυτοματοποιημένη οπτική αναγνώριση και ταξινόμηση για ένα ευρύτερο φάσμα κοινών ανακυκλώσιμων υλικών, κατάλληλο τόσο για την πρωτοβάθμια όσο και για τη δευτεροβάθμια εκπαίδευση. Τα υπάρχοντα μοντέλα συχνά επικεντρώνονται σε συγκεκριμένα υλικά ή στερούνται την ενσωμάτωση προηγμένων τεχνολογιών όπως η όραση υπολογιστών, υποδεικνύοντας μια ευκαιρία για ένα πιο ολιστικό και τεχνολογικά σχετικό εκπαιδευτικό εργαλείο [17].

## 1.3 Επίλογος

Η ενσωμάτωση φυσικών μοντέλων προσομοίωσης εργοστασίων ανακύκλωσης στην εκπαίδευση, με έμφαση στη χρήση τεχνολογιών STEM και μηχανικής μάθησης, αποτελεί μια καινοτόμο προσέγγιση για την καλλιέργεια περιβαλλοντικής συνείδησης και τεχνολογικών δεξιοτήτων. Μέσα από την κατασκευή μικρής κλίμακας συστημάτων διαλογής απορριμμάτων, που αξιοποιούν αισθητήρες, κάμερες και αλγορίθμους βαθιάς μάθησης όπως τα CNN μοντέλα, οι μαθητές αποκτούν βιωματική κατανόηση της κυκλικής οικονομίας και των διαδικασιών αυτοματοποίησης. Η προτεινόμενη κατασκευή, με χαμηλό κόστος και χρήση ανακυκλωμένων υλικών, προάγει τη βιωσιμότητα και την προσαρμοστικότητα σε διαφορετικές εκπαιδευτικές βαθμίδες. Παράλληλα, η σύνδεση με ψηφιακές προσομοιώσεις και διαθεματικά προγράμματα ενισχύει τη μάθηση μέσω έργου, γεφυρώνοντας τη θεωρία με την πράξη. Το παρόν έργο φιλοδοξεί να καλύψει το κενό για ολοκληρωμένα εκπαιδευτικά μοντέλα, προσφέροντας ένα ευέλικτο εργαλείο που συνδυάζει τεχνολογία, περιβαλλοντική ευαισθησία και πρακτική εφαρμογή, με στόχο την προώθηση μιας πιο βιώσιμης και τεχνολογικά ενημερωμένης εκπαίδευσης.

## Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο

### 2.1 Εισαγωγή

Το κεφάλαιο αυτό εισάγει έννοιες όπως η εκπαίδευση STEM, η Ρομποτική και η Μηχανική Μάθηση και συναφείς μεθόδους υλοποίησης συστημάτων που μπορούν να «μάθουν» από την εμπειρία.

### 2.2 Η εκπαίδευση STEM.

Η εκπαίδευση STEM αποτελεί ένα ολοκληρωμένο εκπαιδευτικό πλαίσιο που συνδυάζει τις επιστήμες, την τεχνολογία, τη μηχανική και τα μαθηματικά, με στόχο να προετοιμάσει τους μαθητές για τις προκλήσεις του 21ου αιώνα [1]. Η ενσωμάτωση της εκπαίδευσης STEM στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση προσφέρει πολλαπλά οφέλη:

Ανάπτυξη κριτικής σκέψης και επίλυσης προβλημάτων: Η εκπαίδευση STEM ενθαρρύνει τους μαθητές να σκέφτονται κριτικά, να αναλύουν δεδομένα και να επιλύουν σύνθετα προβλήματα, δεξιότητες απαραίτητες για την επιτυχία σε κάθε τομέα [30].

Καλλιέργεια της δημιουργικότητας και της καινοτομίας: Μέσω πρακτικών δραστηριοτήτων και πειραμάτων, οι μαθητές αναπτύσσουν τη δημιουργικότητά τους και μαθαίνουν να σκέφτονται έξω από τα συνηθισμένα [3].

Ενίσχυση της συνεργασίας και της ομαδικότητας: Η εκπαίδευση STEM προωθεί την ομαδική εργασία και τη συνεργασία, βοηθώντας τους μαθητές να αναπτύξουν κοινωνικές δεξιότητες και να μάθουν να εργάζονται αποτελεσματικά σε ομάδες [13].

Προετοιμασία για τις απαιτήσεις της αγοράς εργασίας: Οι τομείς STEM αποτελούν σημαντικό μέρος της παγκόσμιας οικονομίας και προσφέρουν πολλές ευκαιρίες σταδιοδρομίας. Η εκπαίδευση STEM προετοιμάζει τους μαθητές για αυτές τις ευκαιρίες, εξοπλίζοντάς τους με τις απαραίτητες γνώσεις και δεξιότητες [31].

Ενίσχυση της επιστημονικής και τεχνολογικής παιδείας: Η εκπαίδευση STEM συμβάλλει στην ανάπτυξη της επιστημονικής και τεχνολογικής παιδείας, βοηθώντας τους μαθητές να κατανοήσουν τον κόσμο γύρω τους και να συμμετέχουν ενεργά στην κοινωνία [15].

Ενίσχυση των εννοιών Hard Skills (Απαιτητικές Δεξιότητες) και Soft Skills (Δεξιότητες Ήπιου Τύπου): Οι hard skills και οι soft skills αποτελούν συμπληρωματικά σύνολα δεξιοτήτων κρίσιμα τόσο στον τομέα STEM όσο και στην εκπαίδευση. Οι hard skills αναφέρονται σε τεχνικές, εξειδικευμένες γνώσεις και ικανότητες, όπως η ανάλυση δεδομένων, ο προγραμματισμός, η χρήση τεχνολογικών εργαλείων και η εφαρμογή επιστημονικών και μαθηματικών μεθόδων. Αντίθετα, οι soft skills περιλαμβάνουν διαπροσωπικές και ενδοπροσωπικές δεξιότητες, όπως η συνεργασία, η επικοινωνία, η προσαρμοστικότητα και η κριτική σκέψη, οι οποίες είναι απαραίτητες για την αποτελεσματική ομαδική εργασία, τη διδασκαλία και τη διαχείριση της μαθησιακής διαδικασίας. Συνδυαστικά, οι δύο κατηγορίες δεξιοτήτων συμβάλλουν στην ολιστική ανάπτυξη επαγγελματιών STEM και εκπαιδευτικών, ενισχύοντας την απόδοσή τους σε απαιτητικά και διαρκώς μεταβαλλόμενα περιβάλλοντα [30,32].

Η εφαρμογή της εκπαίδευσης STEM στα σχολεία απαιτεί:

- Εκπαιδευτικούς με εξειδικευμένες γνώσεις και δεξιότητες [31].

- Εργαστήρια και εξοπλισμό για πρακτικές δραστηριότητες [3].
- Ενσωμάτωση των αρχών STEM στο αναλυτικό πρόγραμμα σπουδών [15].
- Συνεργασία με φορείς και επιχειρήσεις του κλάδου [26].

Η επένδυση στην εκπαίδευση STEM αποτελεί επένδυση στο μέλλον, καθώς προετοιμάζει τους νέους για τις προκλήσεις και τις ευκαιρίες του 21ου αιώνα [31].

### 2.3 Ρομποτική και Αυτοματοποίηση στην Εκπαίδευση

Η ρομποτική και η αυτοματοποίηση διαδραματίζουν έναν αυξανόμενο ρόλο στα εκπαιδευτικά περιβάλλοντα, προσφέροντας ευκαιρίες για τη διδασκαλία του προγραμματισμού, των αρχών μηχανικής και της επίλυσης προβλημάτων [16]. Υπάρχουν πολλά παραδείγματα εκπαιδευτικών έργων ρομποτικής που σχετίζονται με την ανακύκλωση, όπως τα "Recycling Robots projects", όπου οι μαθητές σχεδιάζουν και προγραμματίζουν ρομπότ για τη συλλογή και ταξινόμηση ανακυκλώσιμων υλικών [33]. Επιπλέον, πολλά έργα επικεντρώνονται στην κατασκευή ρομπότ από ανακυκλωμένα υλικά, ενθαρρύνοντας τη δημιουργικότητα και την κατανόηση των βασικών αρχών της μηχανικής [4], [19]. Η ενσωμάτωση της ρομποτικής με θέματα περιβαλλοντικής επιστήμης, όπως η ανακύκλωση, μπορεί να δημιουργήσει ελκυστικές και διεπιστημονικές μαθησιακές εμπειρίες για τους μαθητές [16]. Η ρομποτική παρέχει μια απτή και διαδραστική πλατφόρμα για τους μαθητές να εφαρμόσουν τις γνώσεις τους STEM για την αντιμετώπιση πραγματικών προκλήσεων, όπως η διαχείριση των απορριμμάτων [4,33].

### 2.4 Μηχανική Μάθηση

Η Μηχανική Μάθηση (Machine Learning) είναι ένας κλάδος της τεχνητής νοημοσύνης (Artificial Intelligence) που εστιάζει στην ανάπτυξη αλγορίθμων και συστημάτων που μπορούν να μάθουν από δεδομένα, να βελτιώνουν την απόδοσή τους με την εμπειρία και να λαμβάνουν αποφάσεις ή να κάνουν προβλέψεις χωρίς να έχουν προγραμματιστεί ρητά για αυτές [34].

Η Μηχανική Μάθηση αποτελεί πλέον το βασικό στοιχείο έρευνας και ανάπτυξης παιδαγωγικών μεθόδων σε όλες τις βαθμίδες εκπαίδευσης. Έτσι και στις δύο βασικές βαθμίδες, πρωτοβάθμια και δευτεροβάθμια, γίνεται η προσπάθεια ένταξης της Μηχανικής Μάθησης και ενσωμάτωσής της στην εκπαιδευτική διαδικασία [35,36].

Στο επίπεδο του δημοτικού, η Μηχανική Μάθηση μπορεί να χρησιμοποιηθεί για τη δημιουργία προσαρμοσμένων εκπαιδευτικών εφαρμογών που βοηθούν τους μαθητές να μαθαίνουν με τον δικό τους ρυθμό. Για παράδειγμα, μια εφαρμογή εξάσκησης μαθηματικών μπορεί να προσαρμόζει τη δυσκολία των ασκήσεων ανάλογα με τις επιδόσεις του μαθητή, προσφέροντας εξατομικευμένη μάθηση [35]. Επιπλέον, οι διαδραστικοί εκπαιδευτικοί βοηθοί, όπως τα chatbot, μπορούν να απαντούν σε απλές ερωτήσεις των μαθητών και να ενισχύουν την κατανόηση διαφόρων εννοιών [35].

Σε μεγαλύτερες τάξεις και στην δευτεροβάθμια εκπαίδευση, η Μηχανική Μάθηση μπορεί να αξιοποιηθεί σε πιο σύνθετες εφαρμογές, όπως τα έξυπνα συστήματα αξιολόγησης, που αναλύουν τις απαντήσεις των μαθητών και παρέχουν στοχευμένη ανατροφοδότηση [35]. Επίσης, οι μαθητές μπορούν να έρθουν σε επαφή με βασικές αρχές της Μηχανικής Μάθησης μέσω εκπαιδευτικών projects, όπως η εκπαίδευση ενός απλού αλγορίθμου για την αναγνώριση εικόνων ή λέξεων [17,36]. Επιπλέον, τα συστήματα ανάλυσης δεδομένων μπορούν να βοηθούν τους εκπαιδευτικούς να

κατανοούν καλύτερα τις ανάγκες των μαθητών και να προσαρμόζουν τη διδασκαλία τους ανάλογα [35]. Τέτοια projects, που ενσωματώνουν παιχνιδοποίηση και κοινωνικό πλαίσιο, ενισχύουν την εμπλοκή των μαθητών και την κατανόηση των αρχών της Μηχανικής Μάθησης [36].

Στόχος είναι η ενασχόληση με βασικές έννοιες της πληροφορικής όπου προφανώς είναι μέρος των προγραμμάτων σπουδών και αποτελούν και τις βασικές συνιστώσες της μηχανικής μάθησης όπως :

- **Αλγόριθμοι:** Σύνολο κανόνων ή οδηγιών που ακολουθεί ένας υπολογιστής για να επιλύσει ένα πρόβλημα [34].
- **Δεδομένα:** Πληροφορίες που χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων μηχανικής μάθησης [34].
- **Μοντέλο:** Το αποτέλεσμα της εκπαίδευσης ενός αλγορίθμου σε ένα σύνολο δεδομένων. Το μοντέλο μπορεί να χρησιμοποιηθεί για την πραγματοποίηση προβλέψεων ή τη λήψη αποφάσεων [34].

## 2.5 Ενδεικτικοί μέθοδοι Μηχανικής Μάθησης

Η Μηχανική Μάθηση (Machine Learning) αποτελεί έναν από τους ταχύτερα αναπτυσσόμενους τομείς της Τεχνητής Νοημοσύνης, με εφαρμογές που καλύπτουν πλήθος επιστημονικών και βιομηχανικών κλάδων [34]. Στο πλαίσιο της παρούσας μελέτης, κρίνεται σκόπιμο να παρουσιαστούν ενδεικτικά ορισμένες μέθοδοι Μηχανικής Μάθησης, οι οποίες έχουν καθιερωθεί στη βιβλιογραφία και χρησιμοποιούνται εκτενώς τόσο για θεωρητική έρευνα όσο και για πρακτικές εφαρμογές. Οι μέθοδοι αυτές διακρίνονται σε τρεις βασικές κατηγορίες: εποπτευόμενη μάθηση (supervised learning), μη εποπτευόμενη μάθηση (unsupervised learning) και ενισχυτική μάθηση (reinforcement learning) [34].

### 2.5.1 Επιβλεπόμενη Μάθηση (Supervised Learning)

Ο αλγόριθμος εκπαιδεύεται σε ένα σύνολο δεδομένων που περιέχει τόσο τις εισόδους όσο και τις επιθυμητές εξόδους. Στόχος είναι να μάθει να αντιστοιχίζει τις εισόδους με τις εξόδους, ώστε να μπορεί να προβλέπει σωστά την έξοδο για νέα δεδομένα [34].

- Παραδείγματα: Πρόβλεψη τιμών κατοικιών, ταξινόμηση email ως spam ή μη spam.

### 2.5.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Ο αλγόριθμος εκπαιδεύεται σε ένα σύνολο δεδομένων που περιέχει μόνο εισόδους, χωρίς επιθυμητές εξόδους. Στόχος είναι να ανακαλύψει κρυφές δομές ή μοτίβα στα δεδομένα [34].

- Παραδείγματα: Ομαδοποίηση πελατών με βάση την αγοραστική τους συμπεριφορά, ανίχνευση ανωμαλιών.

### 2.5.3 Μάθηση με ενίσχυση (Reinforcement Learning)

Ο αλγόριθμος δέχεται μόνο εισόδους αλλά ως έξοδο δέχεται ένα σήμα ανταμοιβής και όχι τα δεδομένα απάντησης που αντιστοιχούν στην είσοδο. Ζητείται να δίνει την σωστή απάντηση σε κάθε είσοδο χωρίς να έχει κάθε την σωστή έξοδο με άμεσο τρόπο. Μαθαίνει πώς να συμπεριφέρεται σε ένα περιβάλλον μέσω δοκιμής και λάθους. Λαμβάνει ανταμοιβή ή ποινή για τις ενέργειές του και προσπαθεί να μεγιστοποιήσει τη συνολική ανταμοιβή [34].

- Παραδείγματα: Εκπαίδευση ενός ρομπότ να περπατάει, τα διάφορα παιχνίδια.

Οι μέθοδοι της μηχανικής μάθησης εκτός από την προσπάθεια ενσωμάτωσης τους στην παιδαγωγική διαδικασία εφαρμόζονται σε πολλούς και διαφορετικούς τομείς, όπως:

- **Ανάγνωση εικόνας και ομιλίας:** Αναγνώριση προσώπων, μετατροπή ομιλίας σε κείμενο.
- **Φυσική επεξεργασία γλώσσας:** Μετάφραση κειμένου, ανάλυση συναισθήματος.
- **Συστήματα συστάσεων:** Προτάσεις προϊόντων, μουσικής, ταινιών.
- **Ιατρική:** Διάγνωση ασθενειών, ανάπτυξη νέων φαρμάκων.
- **Οικονομικά:** Πρόβλεψη χρηματιστηριακών τάσεων, ανίχνευση απάτης[34].

Οι παραπάνω μέθοδοι Μηχανικής Μάθησης –εποπτευόμενη, μη εποπτευόμενη, και μάθηση με ενίσχυση παρέχουν ένα ευρύ φάσμα εργαλείων για την αντιμετώπιση ποικίλων προβλημάτων, από την εκπαίδευση και την ανακύκλωση έως τη βιομηχανία και την ιατρική. Η κατανόηση και η εφαρμογή αυτών των μεθόδων στην εκπαίδευση ενισχύει τη διδασκαλία βασικών εννοιών πληροφορικής και προάγει τη δημιουργία καινοτόμων λύσεων σε πραγματικά προβλήματα [37].

## 2.6 Μοντέλα Μηχανικής Μάθησης

Στο εδάφιο αυτό αναφέρουμε κάποιες βασικές κατηγορίες μοντέλων που χρησιμοποιούνται εκτεταμένα για να υλοποιήσουν τις μεθόδους που αναφέρθηκαν στο εδάφιο 2.5

## 2.7 Νευρωνικά Δίκτυα (Neural Networks - NNs)

Τα νευρωνικά δίκτυα αποτελούν μια θεμελιώδη μέθοδο στην τεχνητή νοημοσύνη και τη μηχανική μάθηση, εμπνευσμένα από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου [34]. Αυτά τα υπολογιστικά μοντέλα αποτελούνται από διασυνδεδεμένους κόμβους ή τεχνητούς νευρώνες, οι οποίοι επεξεργάζονται και μεταδίδουν πληροφορίες [38]. Στον πυρήνα ενός νευρωνικού δικτύου βρίσκονται στρώματα νευρώνων, όπου το πρώτο επίπεδο λαμβάνει τα αρχικά δεδομένα και το τελευταίο επίπεδο παράγει το τελικό αποτέλεσμα [38]. Κατά τη διάρκεια της εκπαιδευτικής διαδικασίας, το νευρωνικό δίκτυο προσαρμόζει τα βάρη των συνδέσεων μεταξύ των νευρώνων για να ελαχιστοποιήσει τη διαφορά μεταξύ των προβλεπόμενων και των επιθυμητών εξόδων [38].

Μεταξύ των διαφόρων τύπων Νευρωνικών δικτύων, τα Συνελκτικά Νευρωνικά Δίκτυα (CNNs) έχουν σχεδιαστεί ειδικά για την αναγνώριση και την ταξινόμηση εικόνων. Αυτός ο τύπος δικτύου έχει επιδείξει αξιοσημείωτη επιτυχία σε διάφορους τομείς της υπολογιστικής όρασης, καθιστώντας τα CNNs ένα κρίσιμο εργαλείο για εφαρμογές που περιλαμβάνουν την ανάλυση οπτικών δεδομένων [39].

## 2.8 Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση, γνωστή και ως Deep Learning, αποτελεί ένα εξελισσόμενο και επιδραστικό πεδίο στην τεχνητή νοημοσύνη. Ορίζεται ως ένα υποσύνολο της τεχνητής νοημοσύνης (AI) που αξιοποιεί μια ποικιλία υπολογιστικών τεχνικών για την εξαγωγή μοτίβων και χαρακτηριστικών από σύνολα δεδομένων [34]. Η θεμελιώδης ιδέα πίσω από τη βαθιά μάθηση είναι ότι ένα υπολογιστικό σύστημα μπορεί να εκπαιδευτεί για την εκτέλεση μιας εργασίας χωρίς να απαιτούνται ρητές οδηγίες προγραμματισμού [34]. Αυτή η ικανότητα μάθησης από δεδομένα χωρίς άμεση ανθρώπινη παρέμβαση για κάθε πιθανή περίπτωση την καθιστά ιδιαίτερα ισχυρή για την αντιμετώπιση σύνθετων προβλημάτων.

Επιπλέον, η βαθιά μάθηση εντάσσεται στην ευρύτερη κατηγορία της μηχανικής μάθησης, έναν τομέα της τεχνητής νοημοσύνης που χρησιμοποιεί διάφορους αλγορίθμους για την άντληση πληροφοριών από μεγάλα σύνολα δεδομένων [34]. Η σχέση μεταξύ αυτών των πεδίων είναι ιεραρχική: η τεχνητή νοημοσύνη είναι η ευρύτερη έννοια, η μηχανική μάθηση είναι ένα υποσύνολό της, και η βαθιά μάθηση είναι ένα εξειδικευμένο υποσύνολο της μηχανικής μάθησης [34]. Αυτή η θέση της βαθιάς μάθησης στην ιεραρχία της τεχνητής νοημοσύνης υπογραμμίζει την εξέλιξη των τεχνικών, από γενικές μεθόδους επίλυσης προβλημάτων σε πιο εξειδικευμένες προσεγγίσεις που μπορούν να αντιμετωπίσουν συγκεκριμένες προκλήσεις με αυξημένη αποτελεσματικότητα.

Στον πυρήνα της, η βαθιά μάθηση επικεντρώνεται στην ανάπτυξη αλγορίθμων ικανών να ανιχνεύουν σύνθετα μοτίβα σε μεγάλα σύνολα δεδομένων [34]. Αυτή η ικανότητα αναγνώρισης προτύπων επιτρέπει στα συστήματα βαθιάς μάθησης να λαμβάνουν αποφάσεις ή να κάνουν προβλέψεις βασιζόμενα σε υπάρχοντα δεδομένα [34]. Η προσέγγιση αυτή, καθοδηγούμενη από τα δεδομένα, αποτελεί ένα από τα κύρια χαρακτηριστικά της βαθιάς μάθησης, καθώς η απόδοση των μοντέλων βελτιώνεται συνήθως με την αύξηση του όγκου των δεδομένων εκπαίδευσης [40].

Η βαθιά μάθηση βασίζεται σε τεχνητά Νευρωνικά δίκτυα, τα οποία είναι αρχιτεκτονικές εμπνευσμένες από τη δομή και τη λειτουργία των νευρικών συστημάτων όπως στον ανθρώπινο εγκέφαλο [34]. Αυτά τα δίκτυα λειτουργούν μέσω αλγορίθμων αναγνώρισης προτύπων, επιτρέποντας στο σύστημα να αναγνωρίζει και να ερμηνεύει πολύπλοκα μοτίβα στα δεδομένα [34]. Ένα κρίσιμο χαρακτηριστικό που διαφοροποιεί τη βαθιά μάθηση από άλλες μορφές μηχανικής μάθησης είναι η χρήση πολλαπλών στρωμάτων (από τρία έως αρκετές εκατοντάδες ή χιλιάδες) στο νευρωνικό δίκτυο [38]. Αυτό το "βάθος" του δικτύου επιτρέπει την ιεραρχική μάθηση αναπαράστασεων, όπου κάθε στρώμα μαθαίνει όλο και πιο αφηρημένα χαρακτηριστικά από τα δεδομένα εισόδου [39].

Στην ουσία, η βαθιά μάθηση είναι μια τεχνική μηχανικής μάθησης όπου πολλαπλά επίπεδα απλών μονάδων επεξεργασίας συνδέονται διαδοχικά σε ένα δίκτυο [39]. Η είσοδος στο σύστημα διέρχεται μέσα από κάθε ένα από αυτά τα επίπεδα, επιτρέποντας στο δίκτυο να μαθαίνει σύνθετες δομές χωρίς την ανάγκη για υπερβολικά μεγάλες ποσότητες δεδομένων [39]. Αυτή η ικανότητα να μαθαίνει πολύπλοκα μοτίβα από σχετικά μικρότερα σύνολα δεδομένων οφείλεται στην ικανότητα των βαθιών δικτύων να εξάγουν ιεραρχικά χαρακτηριστικά, όπου κάθε επίπεδο μαθαίνει ένα διαφορετικό επίπεδο αναπαράστασης των δεδομένων.

Τα Συνελκτικά Νευρωνικά Δίκτυα είναι ένα από τα πιο γνωστά δίκτυα βαθιάς μάθησης και με βάση αυτά έχουν επιτευχθεί σημαντικά επιτεύγματα σε μελέτες όπως η αναγνώριση φωνής, η επεξεργασία εικόνας, η αναγνώριση προτύπων [39].

## 2.9 Συνελκτικά Νευρωνικά Δίκτυα (CNNs)

Τα Συνελκτικά Νευρωνικά Δίκτυα (CNNs) είναι μια κατηγορία τεχνητών νευρωνικών δικτύων, τα οποία αποτελούνται κυρίως από συνελκτικά επίπεδα. Σχεδιάστηκαν για την επεξεργασία δεδομένων που έχουν μια τοπολογία σαν πλέγμα, όπως οι εικόνες [39]. Η βασική λειτουργία ενός CNN είναι η ανίχνευση προτύπων στα δεδομένα εισόδου μέσω της χρήσης συνέλιξης. Αυτή η διαδικασία περιλαμβάνει την εφαρμογή ενός συνόλου φίλτρων (ή πυρήνων) στην εικόνα εισόδου, όπου κάθε φίλτρο μαθαίνει να αναγνωρίζει ένα συγκεκριμένο χαρακτηριστικό, όπως άκρες, γωνίες ή υφές [39]. Η έξοδος αυτής της λειτουργίας είναι ένας χάρτης χαρακτηριστικών, ο οποίος υποδεικνύει την παρουσία και τη θέση αυτών των χαρακτηριστικών στην είσοδο. Τα CNNs αξιοποιούν την πράξη της συνέλιξης για να μάθουν αυτόματα ιεραρχίες χωρικών χαρακτηριστικών [39].

Η ανάπτυξη των CNNs συνδέεται άρρηκτα με την πρόοδο στην υπολογιστική όραση. Αυτά τα δίκτυα έχουν καταστεί η κινητήρια δύναμη πίσω από σημαντικές εξελίξεις σε εργασίες όπως η ταξινόμηση εικόνων, η ανίχνευση αντικειμένων και η τμηματοποίηση εικόνων [39]. Η ικανότητά τους να αναλύουν οπτικές πληροφορίες τα έχει καταστήσει απαραίτητα σε ένα ευρύ φάσμα εφαρμογών.

Ένα Συνελικτικό Νευρωνικό Δίκτυο αποτελείται από διάφορα βασικά δομικά στοιχεία, τα οποία συνεργάζονται για την επεξεργασία και την ανάλυση των δεδομένων εισόδου. Αυτά τα στοιχεία περιλαμβάνουν τα συνελικτικά επίπεδα, τα επίπεδα ενεργοποίησης και τα επίπεδα συγκέντρωσης [39].

### 2.9.1 Συνελικτικά Επίπεδα (Convolutional Layers)

Τα Συνελικτικά Επίπεδα είναι το θεμελιώδες δομικό στοιχείο ενός CNN. Σε αυτά τα επίπεδα, η είσοδος συνελίσσεται με έναν αριθμό εκπαιδευσιμων φίλτρων (γνωστοί και ως πυρήνες). Η λειτουργία της συνέλιξης περιλαμβάνει την ολίσθηση αυτών των φίλτρων πάνω από την εικόνα εισόδου, υπολογίζοντας το γινόμενο των τιμών του φίλτρου με τις αντίστοιχες τιμές των εικονοστοιχείων στην περιοχή της εικόνας που καλύπτει το φίλτρο, και στη συνέχεια αθροίζοντας τα αποτελέσματα [39]. Κάθε φίλτρο έχει σχεδιαστεί για να ανιχνεύει ένα συγκεκριμένο χαρακτηριστικό στην είσοδο.

Η περιοχή της εισόδου που επεξεργάζεται ένα φίλτρο σε μια συγκεκριμένη στιγμή ονομάζεται δεκτικό πεδίο (receptive field). Ο διασκελισμός (stride) καθορίζει πόσα εικονοστοιχεία μετακινείται το φίλτρο κάθε φορά που ολισθαίνει πάνω από την είσοδο. Η έξοδος ενός συνελικτικού επιπέδου είναι ένα σύνολο χαρτών χαρακτηριστικών (feature maps), όπου κάθε χάρτης αντιστοιχεί στην έξοδο ενός συγκεκριμένου φίλτρου που εφαρμόζεται στην είσοδο. Αυτοί οι χάρτες χαρακτηριστικών αναπαριστούν την παρουσία των χαρακτηριστικών που ανιχνεύονται από τα φίλτρα [39].

Οι παράμετροι ενός συνελικτικού επιπέδου περιλαμβάνουν τον αριθμό των φίλτρων που χρησιμοποιούνται, το μέγεθος κάθε φίλτρου (ύψος και πλάτος), και την ποσότητα padding που εφαρμόζεται στην είσοδο. Το padding αναφέρεται στην προσθήκη εικονικών εικονοστοιχείων γύρω από την άκρη της εικόνας εισόδου, η οποία μπορεί να χρησιμοποιηθεί για τον έλεγχο του μεγέθους των χαρτών χαρακτηριστικών εξόδου [39].

### 2.9.2 Επίπεδα Ενεργοποίησης (Activation Layers)

Μετά από κάθε συνελικτικό επίπεδο, εφαρμόζεται συνήθως ένα επίπεδο ενεργοποίησης. Η λειτουργία αυτού του επιπέδου είναι η εισαγωγή μη γραμμικότητας στο δίκτυο. Χωρίς μη γραμμικές συναρτήσεις ενεργοποίησης, το νευρωνικό δίκτυο θα συμπεριφερόταν σαν ένα γραμμικό μοντέλο, ανεξάρτητα από τον αριθμό των επιπέδων [39]. Οι κοινές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στα CNNs περιλαμβάνουν την ReLU (Rectified Linear Unit), τη Sigmoid και την Tanh.

Η ReLU είναι η πιο συχνά χρησιμοποιούμενη συνάρτηση ενεργοποίησης στα CNNs λόγω της απλότητάς της και της αποτελεσματικότητάς της στην πρόληψη του προβλήματος της εξαφάνισης της κλίσης. Η συνάρτηση ReLU ορίζεται ως  $\sigma(x) = \max(0, x)$ , που σημαίνει ότι επιστρέφει την είσοδο αν είναι θετική, αλλιώς επιστρέφει μηδέν. Οι συναρτήσεις Sigmoid και Tanh, αν και χρησιμοποιούνταν σε παλαιότερες αρχιτεκτονικές, μπορούν να οδηγήσουν στο πρόβλημα της εξαφάνισης της κλίσης σε βαθιά δίκτυα, όπου οι κλίσεις γίνονται πολύ μικρές κατά τη διάρκεια της οπισθοδιάδοσης (backpropagation), εμποδίζοντας την αποτελεσματική εκπαίδευση των προηγούμενων επιπέδων [39].

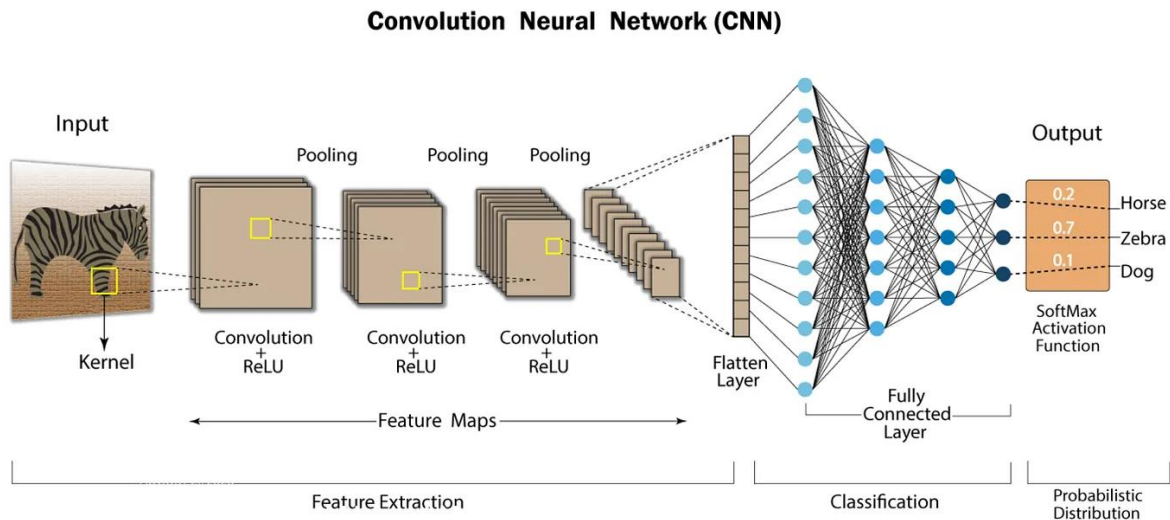
### 2.9.3 Επίπεδα Συγκέντρωσης (Pooling Layers)

Τα επίπεδα συγκέντρωσης εισάγονται περιοδικά μετά από ένα ή περισσότερα συνελκτικά επίπεδα για τη μείωση των χωρικών διαστάσεων των χαρτών χαρακτηριστικών. Αυτό βοηθά στη μείωση του αριθμού των παραμέτρων και της υπολογιστικής πολυπλοκότητας του δικτύου, ενώ παράλληλα καθιστά τα μαθησιακά χαρακτηριστικά πιο αμετάβλητα σε μικρές χωρικές μετατοπίσεις [39]. Δύο κοινοί τύποι λειτουργιών συγκέντρωσης είναι η μέγιστη συγκέντρωση (max pooling) και η μέση συγκέντρωση (average pooling). Η μέγιστη συγκέντρωση επιλέγει τη μέγιστη τιμή από ένα μικρό παράθυρο (π.χ., 2x2) στον χάρτη χαρακτηριστικών, ενώ η μέση συγκέντρωση υπολογίζει τη μέση τιμή. Η μέγιστη συγκέντρωση χρησιμοποιείται πιο συχνά στην πράξη [39]. Οι παράμετροι ενός επιπέδου συγκέντρωσης περιλαμβάνουν το μέγεθος του παραθύρου συγκέντρωσης και τον διασκελισμό.

### 2.9.4 Τυπική Αρχιτεκτονική ενός CNN

Μια τυπική αρχιτεκτονική CNN αποτελείται από μια σειρά σταδίων, όπου κάθε στάδιο περιλαμβάνει συνήθως ένα ή περισσότερα συνελκτικά επίπεδα, ακολουθούμενα από μια συνάρτηση ενεργοποίησης (συνήθως ReLU) και στη συνέχεια ένα επίπεδο συγκέντρωσης (συνήθως μέγιστης συγκέντρωσης). Αυτή η αλληλουχία επιτρέπει στο δίκτυο να μαθαίνει ιεραρχικές αναπαραστάσεις των δεδομένων εισόδου. Στα βαθύτερα επίπεδα, τα δίκτυα μπορούν να συνδυάζουν χαρακτηριστικά λεπτότερης υφής που εξάγονται στα προηγούμενα στάδια και να συνθέτουν χαρακτηριστικά περισσότερο αδρής υφής [39].

Μετά από αρκετά συνελκτικά και επίπεδα συγκέντρωσης, τα εξαγόμενα χαρακτηριστικά υψηλού επιπέδου τροφοδοτούνται σε ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα (fully connected layers). Αυτά τα επίπεδα εκτελούν την τελική ταξινόμηση ή την εργασία παλινδρόμησης. Σε ένα πλήρως συνδεδεμένο επίπεδο, κάθε νευρώνας είναι συνδεδεμένος με κάθε νευρώνα του προηγούμενου επιπέδου [39]. Το τελευταίο επίπεδο του δικτύου είναι το επίπεδο εξόδου, του οποίου η δομή και η συνάρτηση ενεργοποίησης εξαρτώνται από τη συγκεκριμένη εργασία. Για εργασίες ταξινόμησης πολλαπλών κλάσεων, χρησιμοποιείται συνήθως η συνάρτηση ενεργοποίησης Softmax για την παραγωγή μιας κατανομής πιθανοτήτων πάνω στις διαφορετικές κλάσεις. Για εργασίες δυαδικής ταξινόμησης (binary classification), ένα CNN μαθαίνει να αντιστοιχίζει δεδομένα εισόδου – συνήθως εικόνες – σε μία από δύο διακριτές κατηγορίες. Το CNN εκμεταλλεύεται συνελκτικά φίλτρα (convolutional kernels) για την αυτόματη εξαγωγή ιεραρχικών χαρακτηριστικών από τα πρωτογενή δεδομένα, μειώνοντας την ανάγκη χειροκίνητης προεπεξεργασίας. Η εκπαίδευση γίνεται μέσω βελτιστοποίησης ενός συναρτησιακού κόστους (π.χ. binary cross-entropy), με στόχο τη μεγιστοποίηση της ακρίβειας πρόβλεψης και την ελαχιστοποίηση των σφαλμάτων ταξινόμησης. Η τελική έξοδος του μοντέλου παράγεται συνήθως από έναν νευρώνα με συνάρτηση ενεργοποίησης sigmoid, ο οποίος επιστρέφει πιθανότητα ένταξης της εισόδου σε μία από τις δύο κατηγορίες [41].



Εικόνα 2.9.4 Τυπική αρχιτεκτονική CNN δικτύου

## 2.10 Υπολογιστική Όραση και Μηχανική Μάθηση για Αναγνώριση και Ταξινόμηση Αντικειμένων

Η υπολογιστική όραση και η μηχανική μάθηση βρίσκουν όλο και μεγαλύτερη εφαρμογή στην αναγνώριση και ταξινόμηση αντικειμένων, συμπεριλαμβανομένου του τομέα της διαχείρισης απορριμμάτων [39]. Οι κάμερες χρησιμοποιούνται ευρέως σε έργα αναγνώρισης αντικειμένων και ταξινόμησης εικόνων λόγω του χαμηλού κόστους και των ικανοποιητικών επιδόσεών τους [42]. Η διαδικασία δημιουργίας ενός συνόλου δεδομένων εικόνων για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης περιλαμβάνει τη συλλογή και την επισήμανση εικόνων για κάθε κατηγορία υλικού [42]. Υπάρχουν επίσης διαθέσιμα ανοικτά σύνολα δεδομένων εικόνων για την ανακύκλωση, όπως τα TACO και TrashNet, τα οποία μπορούν να χρησιμοποιηθούν ως σημείο εκκίνησης ή για τη συμπλήρωση ενός προσαρμοσμένου συνόλου δεδομένων [28,29].

## 2.11 Γλώσσα υλοποίησης Python

Για την εκπαίδευση και την εφαρμογή μοντέλων μηχανικής μάθησης σε υπολογιστές με περιορισμένους πόρους, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python [43]. Η Python αποτελεί μία από τις πλέον διαδεδομένες γλώσσες προγραμματισμού υψηλού επιπέδου, η οποία αναπτύχθηκε από τον Guido van Rossum και παρουσιάστηκε για πρώτη φορά το 1991 [43]. Χαρακτηρίζεται από την απλότητα στη σύνταξή της, την αναγνωσιμότητα του κώδικα και την ευελιξία της, στοιχεία που την καθιστούν ιδιαίτερα προσιτή τόσο σε αρχάριους όσο και σε έμπειρους προγραμματιστές [43]. Η Python είναι γλώσσα ερμηνευόμενη (interpreted), αντικειμενοστραφής (object-oriented) και δυναμικού τύπου (dynamically typed), γεγονός που της επιτρέπει να υποστηρίζει ένα ευρύ φάσμα εφαρμογών, από την ανάπτυξη απλών εφαρμογών λογισμικού έως και σύνθετα έργα τεχνητής νοημοσύνης [43].

Η δημοτικότητα της Python έχει ενισχυθεί σημαντικά τα τελευταία χρόνια λόγω της εκτεταμένης κοινότητας υποστήριξης και της ύπαρξης πλούσιων βιβλιοθηκών και πλαισίων (frameworks), όπως τα NumPy, Pandas, TensorFlow και PyTorch, που διευκολύνουν την επεξεργασία δεδομένων, τη μηχανική μάθηση και τη βαθιά μάθηση [39,43]. Χάρη σε αυτά τα χαρακτηριστικά, η Python έχει καθιερωθεί ως βασικό εργαλείο στον χώρο της επιστήμης δεδομένων, της μηχανικής μάθησης και της

τεχνητής νοημοσύνης, προσφέροντας τη δυνατότητα ταχείας ανάπτυξης και πειραματισμού με αλγορίθμους και μοντέλα [39].

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, η Python αξιοποιείται ως η κύρια γλώσσα προγραμματισμού για την ανάπτυξη του κώδικα, την υλοποίηση και εκπαίδευση του μοντέλου αναγνώρισης αντικειμένων, καθώς και για τη διασύνδεση με τις βιβλιοθήκες που υποστηρίζουν την επεξεργασία δεδομένων και την αλληλεπίδραση με το υλικό (Arduino, αισθητήρες και ενεργοποιητές) [43]. Μπορούν να χρησιμοποιηθούν ελαφριές βιβλιοθήκες Python όπως το PyTorch, το OpenCV ή το Keras με TensorFlow [39,44].

### 2.12 Web based Πλατφόρμες εκπαιδευτικού χαρακτήρα

Ο συνδυασμός οικονομικά προσιτών καμερών και προσβάσιμων πλατφορμών μηχανικής μάθησης όπως το Edge Impulse και το Google Colaboratory καθιστά εφικτή την εφαρμογή οπτικής αναγνώρισης σε ένα εκπαιδευτικό μοντέλο με περιορισμένους πόρους [45]. Επίσης, με τη χρήση του Tinkercad για STEM κατασκευές και του App Inventor για προγραμματισμό φορητών συσκευών που χρησιμοποιούνται στην εκπαιδευτική διαδικασία, καθώς και η συνεχώς μειούμενη τιμή και η αυξανόμενη ευκολία χρήσης των τεχνολογιών AI, διευρύνουν την εφαρμογή τους στην εκπαίδευση, επιτρέποντας τη δημιουργία εξελιγμένων μαθησιακών εργαλείων [9,46].

### 2.13 Μικροελεγκτές (Arduino Uno) και Υπολογιστές (PC) σε Εκπαιδευτικά Έργα

Ο μικροελεγκτής Arduino Uno χρησιμοποιείται ευρέως σε εκπαιδευτικά έργα ρομποτικής και αυτοματοποίησης για τον έλεγχο αισθητήρων και ενεργοποιητών [47]. Είναι εύκολος στην προγραμματισμό, με χρήση διαδικτυακών web εφαρμογών όπως το Tinkercad, αλλά και μέσα από το περιβάλλον του Arduino IDE, και παρέχει μια απλή διεπαφή με τον φυσικό κόσμο [46,47]. Ένας προσωπικός υπολογιστής (PC) προσφέρει σημαντική υπολογιστική ισχύ, καθιστώντας τον κατάλληλο για την εκτέλεση πιο απαιτητικών εργασιών όπως η μηχανική μάθηση και ο συνολικός έλεγχος του συστήματος σε εκπαιδευτικά μοντέλα [9]. Η επικοινωνία μεταξύ ενός προγράμματος Python σε PC και ενός Arduino Uno μπορεί να επιτευχθεί μέσω σειριακής σύνδεσης, χρησιμοποιώντας βιβλιοθήκες Python όπως η PySerial [48]. Το Arduino Uno λειτουργεί ως εξαιρετική διεπαφή για φυσικά εξαρτήματα, ενώ ένας PC παρέχει την απαραίτητη επεξεργαστική ισχύ για τις πτυχές της μηχανικής μάθησης και τον έλεγχο υψηλού επιπέδου, δημιουργώντας μια ισχυρή και ευέλικτη εκπαιδευτική πλατφόρμα [9], μέσα από το Arduino IDE ή και από την πλατφόρμα Tinkercad χρησιμοποιώντας έτοιμο κώδικα C++ για τον έλεγχο του Arduino και όλων των σχετικών ενεργοποιητών και ελεγκτών. Αυτή η κατανομημένη αρχιτεκτονική επιτρέπει σε κάθε στοιχείο να χειρίζεται τις εργασίες για τις οποίες είναι καλύτερα προσαρμοσμένο, βελτιστοποιώντας την απόδοση και απλοποιώντας την ανάπτυξη [47].

### 2.14 Επίλογος

Στο κεφάλαιο αυτό έγινε μια προσπάθεια ανάλυσης των βασικών εννοιών που θα χρησιμοποιηθούν στην διπλωματική. Δίνονται εισαγωγικά στοιχεία για τον όρο εκπαίδευση STEM ώστε να γίνει αντιληπτό γιατί γίνεται χρήση στην παρούσα εργασία. Η Μηχανική Μάθηση και η Βαθιά Μάθηση αναλύονται πιο πολύ με αναφορές στα Συνελκτικά Νευρωνικά Δίκτυα και στα χαρακτηριστικά τους στα οποία στηρίχθηκε η εκπαίδευση του εξειδικευμένου μοντέλου μας για την αναγνώριση αντικειμένων. Η γλώσσα προγραμματισμού Python με τις βιβλιοθήκες που περιγράφηκαν είναι αυτή

με την οποία δημιουργήθηκαν όλοι οι κώδικες της εργασίας. Τέλος ολόκληρη η κατασκευή συνδέεται με τον υπολογιστή και ένα Arduino με τους ανάλογους αισθητήρες και ενεργοποιητές για τον έλεγχο και την εκτέλεση όλων των απαραίτητων ενεργειών της σωστής λειτουργίας και αυτονομίας της.

## Κεφάλαιο 3ο: Σχεδίαση και Κατασκευή του Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης






### 3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση του σχεδιασμού και της κατασκευής του εργοστασίου ανακύκλωσης μικρής κλίμακας. Το κεφάλαιο αυτό θα χωριστεί σε δύο μέρη, το πρώτο θα αναλύσει το υλικό (Hardware) που χρησιμοποιήθηκε σε συνδυασμό με την διαδικασία και τα βήματα συναρμολόγησης ενώ το δεύτερο μέρος θα αναλύσει το λογισμικό (Software) και των τρόπου του System Integration για την ομαλή λειτουργία όλων των υποσυστημάτων που χρησιμοποιήθηκαν για την ολοκλήρωση της κατασκευής.

### 3.2 Υλικά Κατασκευής βασικού ταινιόδρομου και των ταινιόδρομων εξόδου

Μετά από έρευνα για την προμήθεια του υλικού τόσο σε φυσικά όσο και σε ηλεκτρονικά καταστήματα αποφασίστηκε η αγορά των υλικών του παρακάτω πίνακα για την κατασκευή του πρωτότυπου έργου. Να σημειωθεί ότι ο παρακάτω πίνακας έχει τα υλικά που χρησιμοποιήθηκαν στην τελική ολοκλήρωση της κατασκευής, παραλείπονται τα υλικά που λόγω αστοχίας καταστράφηκαν ή δεν χρησιμοποιήθηκαν:

Πίνακας 3.2 Υλικά κατασκευής

Υλικό	Τιμή	Εικόνα	Κατάστημα
Κολώνα Ντέξιον 36x250cm * 1 Κολώνα Ντέξιον 36x200cm * 2	30,00€		Τοπικό κατάστημα Αλεξάνδρειας
Υλικά συναρμολόγησης (καλώδια βίδες κτλ.) PVC υδραυλική σωλήνα, Φύλλα PVC και λευκό Πολυκαρβονικό φύλλο PVC	50,00€		Τοπικό κατάστημα Αλεξάνδρειας
Ρουλεμάν - 6001ZZ (12mm Bore, 28mm OD)*20	12,80€		Grobotronics
Creality 3D 42-40 Stepper Motor	14,80€		Grobotronics
Stepper motor Nema17 * 4	25,50€		Temu
MG996R servo motors * 4	11,54€		Temu

Σχεδίαση και Κατασκευή του Εκπαιδευτικού Μοντέλου Εργοστασίου Ανακύκλωσης

Drv 8825 stepper motor drivers * 5	16,00€		Grobotronics
Timing belt GT2 – 202mm *5	7,00€		Grobotronics
Timing pulley GT2 – 6mm-40T(12)*5	14,00€		Grobotronics
Timing pulley GT2 – 10mm-20T(8)*5	8,00€		Grobotronics
Stepper motor mount – Nema17 *5	9,50€		Grobotronics
CNC shield V3	3,16€		Temu
16-Channel 12-bit PWM/Servo Driver - PCA9685	6,80€		Grobotronics
Esp32-cam module *4	51,20€		Grobotronics
2 * Arduino uno R3	0		
Παλιό φορτιστές κινητών για τις esp32-cams *4	0		
Horizontally desktop case	0		
ATX Power supply conversion 12v,5v,3v	0		
Σύνολα	260,30€		

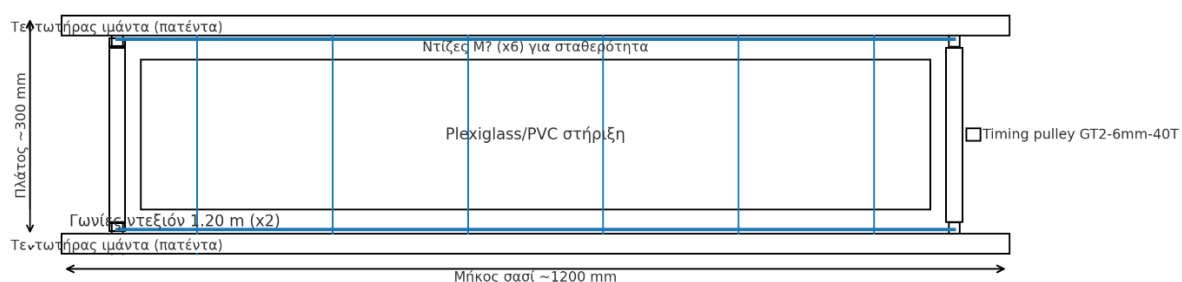
Στην κατασκευή χρησιμοποιήθηκε επίσης ένα παλιό πανό για την δημιουργία των ιμάντων μεταφοράς των ταινιόδρομων.

### 3.3 Υλοποίηση κατασκευής – συναρμολόγησης Υλικού βασικού ταινιόδρομου και των ταινιόδρομων εξόδου

Στην ενότητα αυτή αναλύουμε τα βήματα και την όλη διαδικασία υλοποίησης και ολοκλήρωσης των επιμέρους υποσυστημάτων και των κατασκευών με χρήση των υλικών που εν τέλει χρησιμοποιήσαμε.

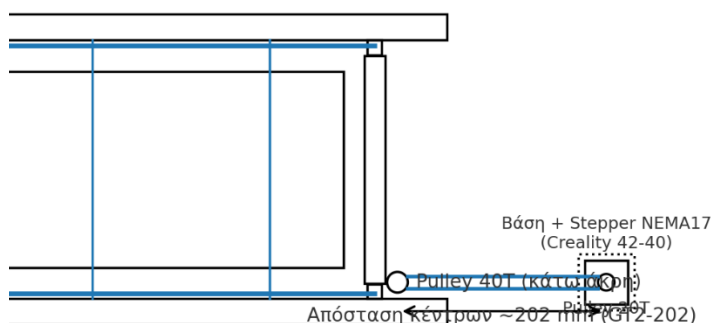
### 3.4 Βήματα κατασκευής βασικού ταινιόδρομου:

Μετά την ολοκλήρωση της κοπής της 2.50 m γωνίας ντεξιόν σε δύο κομμάτια των 1.20 m έχουμε τη συναρμολόγηση και σταθεροποίηση του βασικού σασί του κεντρικού ταινιόδρομου με έξι ντίτζες(βίδες όλο πάσο) για την σταθερότητα του σασί και την αποφυγή κολλήσεων με ηλεκτροκολλητή. Στη συνέχεια προσαρμόσαμε τέσσερα ρουλεμάν στην υδραυλική σωλήνα δυο κομμάτια των 15 cm ένα ρουλεμάν σε κάθε άκρη του κομματιού σε ένα εκ των δύο θα σταθεροποιηθεί και το Timing pulley GT2 – 6mm-40T(12), όπου θα περάσει βίδα 12 cm και θα δημιουργηθούν τα δύο ράουλα με άξονα του βασικού ταινιόδρομου όπου περάσαμε των ιμάντα μεταφοράς. Μεταξύ των δύο αξόνων και της ταινίας μεταφοράς – ιμάντα τοποθετείται PVC- plexiglass για την σταθερότητα μεταφοράς των αντικειμένων και στα άκρα της κατασκευής και σε συγκεκριμένο σημείο του ενός άξονα και από τις δυο πλευρές έχει δημιουργηθεί πατέντα τεντωτήρα ολόκληρης της ταινίας μεταφοράς .



Εικόνα 3.4.1 Κάτοψη Ταινιόδρομου

Στην συνέχεια με μια μεταλική βάση προσαρμόσαμε τον Creality 3D 42-40 Stepper Motor περασμένο στην βάση Stepper motor mount – Nema17 με τοποθετημένο το Timing pulley GT2 – 10mm-20T(8) και το Timing belt GT2 – 202mm από την μεριά που είναι το Timing pulley στην κατασκευή μας.



Εικόνα 3.4.2 Κάτοψη τοποθέτησης βηματικού κινητήρα

Για την πλήρη και τελική και ολοκληρωμένη κατασκευή του βασικού ταινιόδρομου έχουν τοποθετηθεί οι τέσσερις esp32-cams και ο φωτισμός led πάνω από την ταινία μεταφοράς σε βάση που έχει γίνει από PVC και σε ύψος περίπου 15 cm, με απόσταση βάσης από βάση περίπου 25 cm. Μετά την πρώτη βάση τοποθετήσαμε τον πρώτο servo motor πάνω στο plexiglass και σε απόσταση 12 cm για την πρώτη έξοδο υλικού. Ομοίως τοποθετήσαμε τους υπόλοιπους τρεις servo motors μετά από κάθε βάση. Επίσης τοποθετήσαμε τέσσερις γλίστρες-τσουλήθρες απέναντι από κάθε servo motor για να βοηθήνε στην έξοδο των υλικών στους ταινιόδρομους εξόδου.



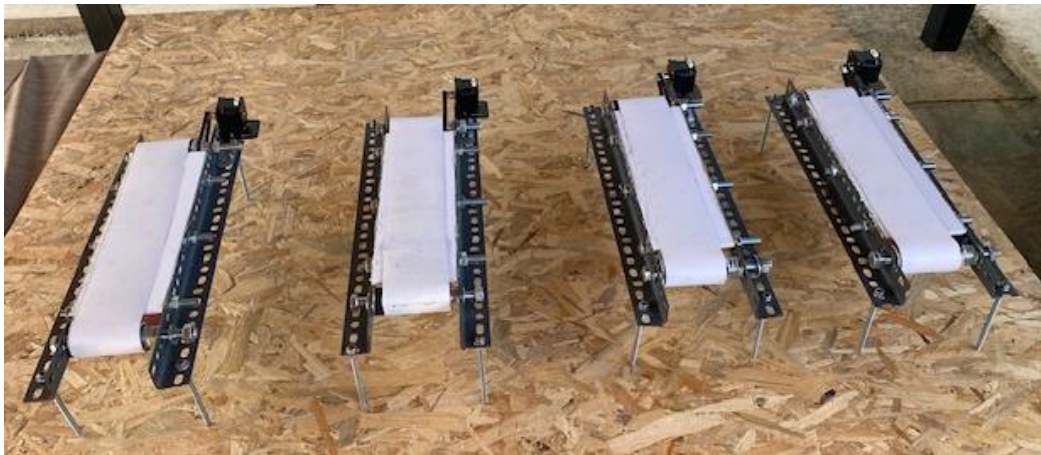
Εικόνα 3.4.3 Κεντρικός ταινιόδρομος

Τέλος τοποθετούμε στα τέσσερα άκρα της κατασκευής βίδες για βάσεις-πόδια που θα πατάει ολόκληρη η κατασκευή και θα είναι σταθερή.

### 3.5 Βήματα κατασκευής ταινιόδρομων εξόδου

Μετά και την κοπή των 2 m γωνιών ντεξιών σε οκτώ κομμάτια των 0,45cm έχουμε τη συναρμολόγηση και σταθεροποίηση του βασικού σασί των τεσσάρων ταινιόδρομων εξόδου με τρεις ντίτζες (βίδες όλο πάσο) για την σταθερότητα του σασί των καθένα ατσηςούς και την αποφυγή κολλήσεων με ηλεκτροκολλητή. Στη συνέχεια προσαρμόζουμε τέσσερα ρουλεμάν στην υδραυλική σωλήνα, δυο κομμάτια των 10 cm ένα ρουλεμάν σε κάθε άκρη του κτησιού αυτής σε ένα εκ των δύο θα σταθεροποιηθεί και το Timing pulley GT2 – 6mm-40T(12), και για τους τέσσερις ταινιόδρομους εξόδου, όπου θα περάσει βίδα 12 cm και θα δημιουργηθούν τα δύο ράουλατηξάξονα για τους ταινιόδρομους όπου θα περάσουμε των ιμάντα μεταφοράς. Μεταξύ της δύο αξόνων και της ταινίας μεταφοράς – ιμάντα τοποθετείται PVC- plexiglass για την σταθερότητα μεταφοράς των αντησιμμένων. Στα άκρα της κατασκευής και σε τηκεκριμένο σητησο του ενός άξονα και από τις δυο πλευρές έχει δημιουργηθεί θέση έτσι ώστε να μπορεί να τεντωθεί ολόκληρης ταινία μεταφοράς, και για τους τέσσερις ταινιόδρομους εξόδου.

Στην συνέχεια με μια μεταλική βάση προσαρμόζουμε τους τέσσερις Stepper motor Nema17 περασμένους στην βάση Stepper motor mount – Nema17 τον καθένα, με τοποθετημένο το Timing pulley GT2 – 10mm-20T(8) και το Timing belt GT2 – 202mm από την μεριά που είντηςτο Timing pulley στην κατασκευή ματηςΤέλος τοποθετούμε στα τέσσερα άκρα της κατασκευής βίδες για βάσεις-πόδια που θα πατάει ολόκληρη η κατασκευή και θα είναι σταθερή.



Εικόνα 3.5 Ταινιόδρομοι εξόδου

### 3.6 Λογισμικό ελέγχου καμερών, servo motor, stepper motor, βασικού ταινιόδρομου και των ταινιόδρομων εξόδου.

Σε αυτή την ενότητα θα αναφερθούμε στην χρήση του κώδικα που χρησιμοποιήσαμε είτε μέσα από το IDE του Arduino ή και τα Python scripts που δημιουργήσαμε για τον έλεγχο και την λειτουργικότητα των τμημάτων της κατασκευής. Στην κατασκευή μας χρησιμοποιήσαμε δύο Arduino Uno R3 ένα για να ελέγχει τον βασικό ταινιόδρομο και τους ενεργοποιητές του (stepper,servo,esp32-cam) και έναν dedicated για το CNC Shield v3 με τους τέσσερεις DRV8825 drivers των τεσσάρων Stepper motors για τον ταυτόχρονο έλεγχο των τεσσάρων ταινιόδρομων εξόδου.

### 3.7 Έλεγχος ESP32-CAM

Με την χρήση των Arduino και των esp32-cam η βασική πλατφόρμα ελέγχου και των επιμέρους υλικών που χρησιμοποιήθηκαν είναι το Arduino IDE. Το Arduino IDE (Integrated Development Environment) είναι το βασικό εργαλείο ανάπτυξης για τον προγραμματισμό μικροελεγκτών Arduino και συμβατών πλακετών. Δεν είναι μόνο ένα απλό περιβάλλον κώδικα, αλλά είναι και ένα ανοικτού κώδικα οικοσύστημα που γεφυρώνει τον κόσμο του υλικού με τον κόσμο του λογισμικού.

Το λογισμικό που αναπτύχθηκε για το ESP32-CAM βασίζεται στο Arduino framework και στη βιβλιοθήκη **ESP Camera**. Στόχος του είναι η εκκίνηση, ρύθμιση και λειτουργία της κάμερας του μικροελεγκτή, καθώς και η αποστολή της εικόνας μέσω ενός web server στο τοπικό δίκτυο Wi-Fi.

Η υλοποίηση του λογισμικού για το υποσύστημα κάμερας βασίζεται στο **ESP32-CAM AI-Thinker** και αναπτύχθηκε στο περιβάλλον **Arduino IDE**, αξιοποιώντας τις βιβλιοθήκες esp\_camera.h και WiFi.h. Το λογισμικό υλοποιεί μια πλήρη στοιβά επικοινωνίας και επεξεργασίας εικόνας, ώστε η συσκευή να λειτουργεί αυτόνομα ως **κόμβος ασύρματης μετάδοσης εικόνας σε πραγματικό χρόνο (real-time video streaming node)**. Το λογισμικό υλοποιεί ένα **αυτοδύναμο ενσωματωμένο σύστημα μετάδοσης εικόνας**. Μέσω της συνδυασμένης χρήσης του αισθητήρα OV2640, της PSRAM και του ενσωματωμένου Wi-Fi του ESP32, επιτυγχάνεται συνεχής και αξιόπιστη ροή εικόνας στο δίκτυο χωρίς τη χρήση εξωτερικών υπολογιστικών πόρων.

```

#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//           Ensure ESP32 Wrover Module or other board with PSRAM is selected
//           Partial images will be transmitted if image exceeds buffer size
//
//           You must select partition scheme from the board menu that has at
//           least 3MB APP space.
//           Face Recognition is DISABLED for ESP32 and ESP32-S2, because it
//           takes up from 15
//           seconds to process single frame. Face Detection is ENABLED if
//           PSRAM is enabled as well

// =====
// Select camera model
// =====
// #define CAMERA_MODEL_WROVER_KIT // Has PSRAM
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM
// #define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
// #define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
// #define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
// #define CAMERA_MODEL_M5STACK_CAMS3_UNIT // Has PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
// #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
// #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
// ** Espressif Internal Boards **
// #define CAMERA_MODEL_ESP32_CAM_BOARD
// #define CAMERA_MODEL_ESP32S2_CAM_BOARD
// #define CAMERA_MODEL_ESP32S3_CAM_LCD
// #define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
// #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
#include "camera_pins.h"

// =====
// Enter your WiFi credentials
// =====
const char *ssid = "HONOR";
const char *password = "6948824748";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
    Serial.begin(115200);

```

```

Serial.setDebugOutput(true);
Serial.println();

camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.frame_size = FRAMESIZE_UXGA;
config.pixel_format = PIXFORMAT_JPEG; // for streaming
//config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
config.fb_location = CAMERA_FB_IN_PSRAM;
config.jpeg_quality = 12;
config.fb_count = 1;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
//                               for larger pre-allocated frame buffer.
if (config.pixel_format == PIXFORMAT_JPEG) {
    if (psramFound()) {
        config.jpeg_quality = 10;
        config.fb_count = 2;
        config.grab_mode = CAMERA_GRAB_LATEST;
    } else {
        // Limit the frame size when PSRAM is not available
        config.frame_size = FRAMESIZE_SVGA;
        config.fb_location = CAMERA_FB_IN_DRAM;
    }
} else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
}
#ifdef CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
#endif

```

```

}

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t *s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1);      // flip it back
  s->set_brightness(s, 1); // up the brightness just a bit
  s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
if (config.pixel_format == PIXFORMAT_JPEG) {
  s->set_framesize(s, FRAMESIZE_QVGA);
}

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_ESP32S3_EYE)
  s->set_vflip(s, 1);
#endif

// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
#endif

WiFi.begin(ssid, password);
WiFi.setSleep(false);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");

```

```

Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}

```

Εικόνα 3.7 Κώδικας webservice esp32-cam

### 3.8 Έλεγχος Stepper Motor – Servo Motor Βασικού ταινιόδρομου

Το λογισμικό που αναπτύχθηκε για τον μικροελεγκτή Arduino έχει ως στόχο τον ταυτόχρονο έλεγχο ενός βηματικού κινητήρα (stepper motor) μέσω οδηγού DRV8825 και τεσσάρων σερβοκινητήρων (servo motors) οι οποίοι ελέγχονται μέσω του κυκλώματος Adafruit PWM Servo Driver (PCA9685). Το πρόγραμμα έχει σχεδιαστεί με μη αποκλειστικό τρόπο (non-blocking), ώστε οι δύο κατηγορίες κινητήρων να μπορούν να λειτουργούν παράλληλα χωρίς αλληλεπιδράσεις που να περιορίζουν τη λειτουργικότητα.

Ο βηματικός κινητήρας κινείται συνεχώς μέσω παλμών στο pin **STEP** του οδηγού DRV8825, ενώ η κατεύθυνση καθορίζεται από το pin **DIR**. Η συχνότητα παλμών καθορίζεται από τη μεταβλητή `stepDelay`, η οποία ρυθμίζει το διάστημα μεταξύ διαδοχικών παλμών. Η χρήση της συνάρτησης `micros()` επιτρέπει τον ακριβή χρονισμό και διασφαλίζει ότι η δημιουργία παλμών γίνεται χωρίς να δεσμεύεται η εκτέλεση του υπολοίπου κώδικα.

Ο έλεγχος των σερβοκινητήρων πραγματοποιείται μέσω εντολών που αποστέλλονται από τη σειριακή θύρα. Οι εντολές έχουν την μορφή:

- CAMx:OPEN (όπου  $x \in \{1,2,3,4\}$ ) για το άνοιγμα του αντίστοιχου σερβοκινητήρα.
- CAMx:CLOSE για το κλείσιμο του αντίστοιχου σερβοκινητήρα.

Κατά την εκτέλεση εντολής “OPEN”, ο αντίστοιχος servo μετακινείται σε προκαθορισμένη γωνία **67°**, ενώ με την εντολή “CLOSE” επανέρχεται στη θέση **0°**. Επιπλέον, το πρόγραμμα διαθέτει μηχανισμό αυτόματης επαναφοράς: κάθε servo παραμένει ανοιχτός για μέγιστη διάρκεια **20 δευτερολέπτων**, μετά την οποία επανέρχεται αυτόματα στη θέση κλεισίματος, ανεξάρτητα από νέες εντολές. Ο μηχανισμός αυτός υλοποιείται με χρήση χρονομέτρησης (`millis()`), ώστε να αποφευχθεί ο αποκλειστικός έλεγχος με καθυστερήσεις (blocking delays).

Η συνολική αρχιτεκτονική του λογισμικού βασίζεται στη συνεχή επανάληψη (loop) όπου:

1. Ελέγχεται και εκτελείται η παλμοδότηση του stepper.
2. Γίνεται έλεγχος για εισερχόμενες εντολές από τη σειριακή θύρα.
3. Ενημερώνεται η κατάσταση και η θέση των servo με βάση τις εντολές και τον μηχανισμό αυτόματου κλεισίματος.

Με την παραπάνω υλοποίηση, το σύστημα παρέχει αξιόπιστο και αποδοτικό έλεγχο κινητήρων, εξασφαλίζοντας τόσο την ανεξαρτησία μεταξύ των υποσυστημάτων όσο και την ευκολία χειρισμού μέσω απλών σειριακών εντολών.

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

#define SERVOMIN 150
#define SERVOMAX 600

// Stepper pins (DRV8825)
const int STEP_PIN = 8;
const int DIR_PIN = 9;
const unsigned long stepDelay = 10000; // 2.6 ms delay (~384 steps/sec)

// Για έλεγχο των servo
const int NUM_SERVOS = 4;
bool servoOpen[NUM_SERVOS] = {false, false, false, false};
unsigned long servoOpenStart[NUM_SERVOS] = {0, 0, 0, 0};
const unsigned long servoOpenDuration = 20000; // 10 δευτερόλεπτα

// Χρόνοι για μη blocking control
unsigned long lastStepTime = 0;

int angleToPulse(int angle) {
    return map(angle, 0, 180, SERVOMIN, SERVOMAX);
}

void setup() {
    Serial.begin(9600);
    pwm.begin();
    pwm.setPWMPfreq(50); // 50Hz για servo

    pinMode(STEP_PIN, OUTPUT);
    pinMode(DIR_PIN, OUTPUT);
    digitalWrite(DIR_PIN, LOW);
    lastStepTime = micros();
}

void loop() {
    // Κίνηση stepper χωρίς blocking
    unsigned long currentMicros = micros();
    if (currentMicros - lastStepTime >= stepDelay) {
        digitalWrite(STEP_PIN, HIGH);
        delayMicroseconds(2);
        digitalWrite(STEP_PIN, LOW);
        lastStepTime = currentMicros;
    }
}
```

```

}

// Διαχείριση εντολών serial
if (Serial.available()) {
  String input = Serial.readStringUntil('\n');
  input.trim();

  // Αναγνώριση εντολών
  for (int i = 0; i < NUM_SERVOS; i++) {
    String openCmd = "CAM" + String(i + 1) + ":OPEN";
    String closeCmd = "CAM" + String(i + 1) + ":CLOSE";

    if (input.equalsIgnoreCase(openCmd)) {
      servoOpen[i] = true;
      servoOpenStart[i] = millis();
      Serial.print("Servo ");
      Serial.print(i);
      Serial.println(" OPEN");
    }
    else if (input.equalsIgnoreCase(closeCmd)) {
      servoOpen[i] = false;
      Serial.print("Servo ");
      Serial.print(i);
      Serial.println(" CLOSE");
    }
  }
}

// Ενημέρωση servo positions
for (int i = 0; i < NUM_SERVOS; i++) {
  if (servoOpen[i]) {
    // Αν πέρασαν 10 δευτερόλεπτα, κλείνουμε servo
    if (millis() - servoOpenStart[i] > servoOpenDuration) {
      servoOpen[i] = false;
    }
  }

  int angle = servoOpen[i] ? 67 : 0;
  pwm.setPWM(i, 0, angleToPulse(angle));
}
}

```

Εικόνα 3.8 Κώδικας ελέγχου stepper motor και servo motor κεντρικού ταινιόδρομου

### 3.9 Έλεγχος Stepper Motor ταινιόδρομων εξόδου

Στο πλαίσιο ελέγχου των τεσσάρων ταινιόδρομων εξόδου χρησιμοποιήσαμε το **CNC Shield V3**, μια πλακέτα επέκτασης για το Arduino UNO, η οποία επιτρέπει τον εύκολο έλεγχο βηματικών κινητήρων μέσω οδηγών τύπου DRV8825. Η πλακέτα υποστηρίζει μέχρι τέσσερις άξονες (X, Y, Z και A). Ο τέταρτος άξονας (A) με τη χρήση jumper επιλογών, λειτούργησε ως **κλώνος (cloned axis)** του X.

Στην υλοποίηση αυτή, ο άξονας **A** έχει κλωνοποιηθεί με τον **X**, ώστε να οδηγεί έναν από τους τέσσερις βηματικούς κινητήρες παράλληλα με τον κινητήρα του **X**. Ο κλωνοποιημένος άξονας εξασφαλίζει ότι οι δύο κινητήρες λαμβάνουν ακριβώς τα ίδια σήματα βηματισμού (STEP) και κατεύθυνσης (DIR).

Το πρόγραμμα (Arduino sketch) που υλοποιήθηκε επιτελεί τις ακόλουθες λειτουργίες:

- **Αρχικοποίηση σημάτων**  
Ορίζονται τα pins του Arduino που συνδέονται με τα σήματα **STEP** και **DIR** για τους άξονες X, Y και Z. Ο άξονας A δεν χρειάζεται ξεχωριστό ορισμό, καθώς μέσω του CNC Shield αντιγράφει αυτόματα τα σήματα του X.
- **Ρύθμιση κατεύθυνσης**  
Στη φάση της αρχικοποίησης, οι γραμμές DIR ορίζονται σε σταθερή λογική κατάσταση (LOW ή HIGH), καθορίζοντας την κατεύθυνση κίνησης των κινητήρων. Έτσι, οι δύο κινητήρες του άξονα X κινούνται πάντα συγχρονισμένα προς την ίδια κατεύθυνση.
- **Δημιουργία παλμών βηματισμού**  
Η κύρια συνάρτηση loop() δημιουργεί περιοδικούς παλμούς στα pins STEP των αξόνων. Κάθε παλμός ισοδυναμεί με ένα μηχανικό βήμα του κινητήρα. Οι παλμοί στέλνονται ταυτόχρονα στους άξονες X, Y και Z, και κατ' επέκταση και στον άξονα A (λόγω του cloning). Αυτό σημαίνει ότι στην πράξη κινούνται **τέσσερις κινητήρες** συγχρονισμένα.
- **Έλεγχος ταχύτητας**  
Ο χρόνος μεταξύ παλμών ρυθμίζεται από τη μεταβλητή step\_delay\_us. Η τιμή αυτή (σε μικροδευτερόλεπτα) καθορίζει τη συχνότητα των παλμών και άρα την ταχύτητα περιστροφής των κινητήρων.

Η συγκεκριμένη υλοποίηση με cloning του άξονα A προσφέρει μια πρακτική και αξιόπιστη λύση για την οδήγηση «δίδυμου» άξονα. Το Arduino σε συνδυασμό με το CNC Shield V3 αναδεικνύει πώς ένα απλό μικροελεγκτικό σύστημα μπορεί να οδηγήσει αποτελεσματικά τέσσερις βηματικούς κινητήρες, διασφαλίζοντας ακρίβεια και σταθερότητα στην κίνηση εκμεταλλεύοντας τις πλήρεις δυνατότητες του.

#### Κώδικας C++ Arduino IDE

```
// CNC Shield V3 - Default Pins
#define X_STEP_PIN 2
#define X_DIR_PIN 5

#define Y_STEP_PIN 3
#define Y_DIR_PIN 6

#define Z_STEP_PIN 4
#define Z_DIR_PIN 7

// Χρόνος μεταξύ παλμών (χαμηλότερο = πιο γρήγορο)
const int step_delay_us = 800; // π.χ. 800 μικροδευτερόλεπτα = ~625 steps/sec

void setup() {
```

```

// Ορισμός εξόδων
pinMode(X_STEP_PIN, OUTPUT);
pinMode(X_DIR_PIN, OUTPUT);

pinMode(Y_STEP_PIN, OUTPUT);
pinMode(Y_DIR_PIN, OUTPUT);

pinMode(Z_STEP_PIN, OUTPUT);
pinMode(Z_DIR_PIN, OUTPUT);
// Αντίστροφη κατεύθυνση (π.χ. LOW αντί για HIGH)
digitalWrite(X_DIR_PIN, LOW);
digitalWrite(Y_DIR_PIN, LOW);
digitalWrite(Z_DIR_PIN, LOW);
}
// Θέτουμε κατεύθυνση προς μία πλευρά για όλους
//digitalWrite(X_DIR_PIN, HIGH);
//digitalWrite(Y_DIR_PIN, HIGH);
// digitalWrite(Z_DIR_PIN, HIGH);
//}

void loop() {
// Στέλνουμε 1 βήμα σε όλους τους άξονες ταυτόχρονα
digitalWrite(X_STEP_PIN, HIGH);
digitalWrite(Y_STEP_PIN, HIGH);
digitalWrite(Z_STEP_PIN, HIGH);
delayMicroseconds(step_delay_us);

digitalWrite(X_STEP_PIN, LOW);
digitalWrite(Y_STEP_PIN, LOW);
digitalWrite(Z_STEP_PIN, LOW);
delayMicroseconds(step_delay_us);
}

```

Εικόνα 3.9 Κώδικας ελέγχου stepper motor ταινιόδρομων εξόδου

### 3.10 Χειρισμός του βασικού ταινιόδρομου και των ταινιόδρομων εξόδου

Η παρακάτω ανάλυση του κώδικα σε Python αποτελεί την υλοποίηση τις ολοκληρωμένου συστήματος αναγνώρισης και ταξινόμησης απορριμμάτων σε κατηγορίες ανακυκλώσιμων υλικών (γυαλί, μέταλλο, χαρτί, πλαστικό). Η υλοποίηση βασίζεται στη συνεργασία τριών βασικών υποσυστημάτων:

1. Συλλογή εικόνας μέσω καμερών IP,
2. Ανάλυση εικόνας με τη χρήση των Custom μοντέλων βαθιάς μάθησης,
3. Αποστολή εντολών σε μικροελεγκτή Arduino για τον έλεγχο μηχανισμών διαλογής (σερβοκινητήρες).

Η αρχιτεκτονική του συστήματος διασφαλίζει την ταυτόχρονη παρακολούθηση πολλαπλών καμερών και την ανεξάρτητη λειτουργία τους, ενώ παρέχει άμεση απόκριση στη φυσική διεπαφή (άνοιγμα και κλείσιμο των θυρίδων ανακύκλωσης).

### Ρυθμίσεις και Παράμετροι

Στην αρχή του κώδικα ορίζονται οι βασικές παράμετροι λειτουργίας:

- **IMG\_SIZE**: το μέγεθος εισόδου των εικόνων (224x224 pixels), ώστε να είναι συμβατό με τα μοντέλα νευρωνικών δικτύων.
- **SERVO\_OPEN\_TIME**: η διάρκεια (10 δευτερόλεπτα) για την οποία παραμένει ανοιχτή η θύρα διαλογής.
- **SERIAL\_PORT** και **BAUD\_RATE**: οι ρυθμίσεις επικοινωνίας με το Arduino.
- **CAM\_CONFIGS**: η λίστα που περιλαμβάνει τις διευθύνσεις IP κάθε κάμερας, την αντίστοιχη ετικέτα του υλικού που αναγνωρίζει, τη διαδρομή φόρτωσης του εκπαιδευμένου μοντέλου, και τις δείκτη της κάμερας.

### Υποσύστημα Λήψης Καρέ (FrameGrabber)

Η κλάση FrameGrabber είναι υπεύθυνη για τη συνεχή τήρηση καρέ από τις IP κάμερες.

- Τρέχει σε ξεχωριστό νήμα (thread) ώστε να μην επιβαρύνει την κύρια ροή εκτέλεσης.
- Χρησιμοποιεί το πρωτόκολλο HTTP για τη λήψη μεμονωμένων εικόνων (/capture).
- Αποθηκεύει το τελευταίο καρέ που λαμβάνει, χρησιμοποιώντας μηχανισμό κλειδώματος (Lock) ώστε να αποφεύγεται η ασυγχρονία κατά την ανάγνωση/εγγραφή.

Με αυτόν τον τρόπο εξασφαλίζεται ότι η κύρια ροή του συστήματος έχει πάντα πρόσβαση στην πιο πρόσφατη εικόνα από κάθε κάμερα.

### Υποσύστημα Ανάλυσης Εικόνας και Ελέγχου (CameraModelHandler)

Η κλάση CameraModelHandler υλοποιεί τη βασική λογική αναγνώρισης και διαλογής:

- **Φόρτωση προεκπαιδευμένου μοντέλου**: Κάθε κάμερα συνδέεται με ένα εξειδικευμένο μοντέλο (σε μορφή .keras) που έχει εκπαιδευτεί να αναγνωρίζει το αντίστοιχο υλικό (π.χ. πλαστικό, μέταλλο).
- **Προεπεξεργασία εικόνας**: Οι εικόνες μετατρέπονται σε μέγεθος 224x224, αλλάζουν σε μορφή RGB και κανονικοποιούνται σε μορφή πίνακα, κατάλληλη για το μοντέλο.
- **Πρόβλεψη**: Το μοντέλο επιστρέφει μια πιθανότητα, η οποία αποθηκεύεται σε έναν κυλιόμενο μέσο όρο (buffer) ώστε να μειώνονται οι τυχαίες διακυμάνσεις.
- **Ελεγχος μηχανισμού**: Εάν η πιθανότητα αναγνώρισης υπερβαίνει το όριο (50%), τότε αποστέλλεται εντολή στο Arduino για το άνοιγμα της αντίστοιχης θύρας ανακύκλωσης. Μετά από προκαθορισμένο χρόνο, αποστέλλεται νέα εντολή για το κλείσιμό της.

## Κεφάλαιο 3°

### Διεπαφή Χρήστη

Για κάθε κάμερα δημιουργείται ξεχωριστό παράθυρο προβολής με χρήση του OpenCV. Στην εικόνα προστίθεται δυναμικά κείμενο που ενημερώνει για:

- Την πιθανότητα αναγνώρισης,
- Την τρέχουσα κατάσταση (αντικείμενο ή όχι), τις ενεργοποίηση ή μη της θύρας.

Αυτό επιτρέπει στον χειριστή να παρακολουθεί σε πραγματικό χρόνο την κατάσταση του συστήματος.

### Κεντρική Εκκίνηση

Η συνάρτηση main υλοποιεί τον αρχικό συγχρονισμό και εκκίνηση:

- Δημιουργεί σύνδεση με το Arduino μέσω σειριακής επικοινωνίας.
- Εγκαθιστά ένα CameraModelHandler για κάθε κάμερα του συστήματος κινεί τα νήματα ώστε όλες οι κάμερες να λειτουργούν ταυτόχρονα και ανεξάρτητα.
- Διαχειρίζεται το κλείσιμο του συστήματος (τερματισμός νημάτων και απελευθέρωση πόρων).

```
import cv2
import numpy as np
import threading
import time
from collections import deque
import requests
import serial
from tensorflow.keras.models import load_model

# === PYΘΜΙΣΕΙΣ ===
IMG_SIZE = 224
SERVO_OPEN_TIME = 10 # sec
SERIAL_PORT = "COM4"
BAUD_RATE = 9600

CAM_CONFIGS = [
    {"ip": "192.168.43.201", "label": "glass", "model_path":
r"binary_models_8_9_25/glass_final_augmentedV2_9_9_25_model.keras",
"cam_index": 1},
    {"ip": "192.168.43.48", "label": "metal", "model_path":
r"binary_models_8_9_25/metal_final_augmentedV2_9_9_25_model.keras",
"cam_index": 2},
    {"ip": "192.168.43.133", "label": "paper", "model_path":
r"binary_models_8_9_25/paper_final_augmentedV2_9_9_25_model.keras",
"cam_index": 3},
    {"ip": "192.168.43.211", "label": "plastic", "model_path":
r"binary_models_8_9_25/plastic_final_augmentedV2_9_9_25_model.keras",
"cam_index": 4},
]
```

```

# === ΚΛΑΣΗ ΠΙΑΝΕΙ ΤΟ ΚΑΡΕ ===
class FrameGrabber(threading.Thread):
    def __init__(self, url):
        super().__init__()
        self.url = url
        self.frame = None
        self.stopped = False
        self.lock = threading.Lock()

    def run(self):
        while not self.stopped:
            try:
                resp = requests.get(self.url, timeout=5)
                if resp.status_code == 200:
                    img_array = np.frombuffer(resp.content, np.uint8)
                    img = cv2.imdecode(img_array, cv2.IMREAD_COLOR)
                    with self.lock:
                        self.frame = img
            except:
                pass

    def read(self):
        with self.lock:
            return self.frame.copy() if self.frame is not None else None

    def stop(self):
        self.stopped = True

# === ΚΑΜΕΡΑ + ΜΟΝΤΕΛΟ + SERVO ΧΕΙΡΙΣΜΟΣ ===
class CameraModelHandler:
    def __init__(self, ip, label, model_path, cam_index, arduino):
        self.ip = ip
        self.label = label
        self.model = load_model(model_path, compile=False)
        self.cam_index = cam_index
        self.arduino = arduino
        self.capture_url = f"http://{ip}/capture"
        self.buffer = deque(maxlen=5)
        self.servo_open = False
        self.servo_open_time = 0

    def preprocess(self, img):
        try:
            img_resized = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
            img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
            img_array = np.expand_dims(img_rgb.astype(np.float32), axis=0)
            return img_array
        except:
            return None

```

```

def predict(self, tensor):
    pred = self.model.predict(tensor, verbose=0)[0][0]
    self.buffer.append(pred)
    return np.mean(self.buffer)

def send_command(self, open_servo=True):
    cmd = f"CAM{self.cam_index}:{'OPEN' if open_servo else 'CLOSE'}\n"
    self.arduino.write(cmd.encode())
    print(f"[ARDUINO] Sent: {cmd.strip()}")

def run(self):
    grabber = FrameGrabber(self.capture_url)
    grabber.start()
    print(f"🌀 CAM{self.cam_index} - {self.label} started")

    window_name = f"CAM{self.cam_index} | {self.label}"
    try:
        while True:
            frame = grabber.read()
            if frame is not None:
                gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                mean_intensity = np.mean(gray)

                if mean_intensity > 230:
                    cv2.putText(frame, "NO OBJECT", (10, 30),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (150, 150,
150), 2)
                else:
                    tensor = self.preprocess(frame)
                    if tensor is not None:
                        confidence = self.predict(tensor)
                        text = f"{self.label.upper()}:
{confidence:.1%}"
                        color = (0, 255, 0) if confidence > 0.55 else
(0, 0, 255)
                        cv2.putText(frame, text, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, color, 2)

                        now = time.time()
                        if confidence > 0.55 and not self.servo_open:
                            self.send_command(True)
                            self.servo_open = True
                            self.servo_open_time = now

                            if self.servo_open and (now -
self.servo_open_time > SERVO_OPEN_TIME):
                                self.send_command(False)
                                self.servo_open = False

```

```

        cv2.imshow(window_name, frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    finally:
        grabber.stop()
        cv2.destroyAllWindows()
        print(f"🛑 CAM{self.cam_index} - stopped")

# === MAIN ΕΚΚΙΝΗΣΗ ===
if __name__ == "__main__":
    print("🔌 Connecting to Arduino...")
    arduino = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
    time.sleep(2)
    print("✅ Arduino ready")

    handlers = [CameraModelHandler(**cfg, arduino=arduino) for cfg in
CAM_CONFIGS]

    # === ΕΚΚΙΝΗΣΗ ΠΑΡΑΛΛΗΛΗ ΓΙΑ ΟΛΕΣ ΤΙΣ ΚΑΜΕΡΕΣ ===
    threads = [threading.Thread(target=h.run) for h in handlers]
    for t in threads: t.start()
    for t in threads: t.join()

    arduino.close()
    print("🛑 Όλα ολοκληρώθηκαν.")

```

Εικόνα 3.10 Κώδικας χειρισμού βασικού ταινιόδρομου

### 3.11 Επίλογος

Στο παρόν κεφάλαιο παρουσιάστηκε ο σχεδιασμός και η κατασκευή του εργοστασίου ανακύκλωσης μικρής κλίμακας, με έμφαση τόσο στο υλικό (hardware) όσο και στο λογισμικό (software) σκέλος του συστήματος. Αρχικά, αναλύθηκαν τα επιμέρους εξαρτήματα που χρησιμοποιήθηκαν, καθώς και τα στάδια συναρμολόγησης, τα οποία αποτέλεσαν τη βάση για την ανάπτυξη μιας ολοκληρωμένης και λειτουργικής κατασκευής. Στη συνέχεια, παρουσιάστηκε το λογισμικό που αναπτύχθηκε, μέσω του οποίου επιτυγχάνεται η διασύνδεση και η συνεργασία όλων των υποσυστημάτων, διασφαλίζοντας την ομαλή λειτουργία του συνόλου.

Ο κώδικας που αναπτύχθηκε υλοποιεί ένα καταναμημένο σύστημα αναγνώρισης αντικειμένων, με κύριο σκοπό την αυτοματοποιημένη διαλογή ανακυκλώσιμων υλικών. Η αρχιτεκτονική του συστήματος διακρίνεται για την επεκτασιμότητά της, καθώς επιτρέπει την εύκολη προσθήκη νέων κατηγοριών ή καμερών, την ανθεκτικότητά της απέναντι σε σφάλματα μέσω τεχνικών φιλτραρίσματος και ελέγχου, αλλά και τη στενή σύνδεσή της με φυσικό εξοπλισμό, όπως οι σερβοκινητήρες, βηματικούς κινητήρες, κάμερες που ελέγχονται από μικροελεγκτή Arduino.

Συνολικά, το σύστημα αυτό συνδυάζει τεχνικές υπολογιστικής όρασης, βαθιάς μάθησης και ρομποτικού ελέγχου, αποτελώντας χαρακτηριστικό παράδειγμα εφαρμογής της τεχνητής νοημοσύνης

### Κεφάλαιο 3°

σε έξυπνα περιβαλλοντικά συστήματα. Η προσέγγιση που υιοθετήθηκε αποδεικνύει τη δυνατότητα υλοποίησης αποδοτικών και ευφών λύσεων σε ζητήματα περιβαλλοντικής διαχείρισης, που μπορούν να ενσωματωθούν στην εκπαιδευτική διαδικασία αναδεικνύοντας παράλληλα τις προοπτικές περαιτέρω ανάπτυξης και βελτίωσης τέτοιων συστημάτων.

## Κεφάλαιο 4ο: Δημιουργία μοντέλου Υπολογιστικής Όρασης για κατηγοριοποίηση αντικειμένων.

### 4.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η διαδικασία ανάπτυξης ενός εξειδικευμένου CNN μοντέλου για ταξινόμηση εικόνων σε δυαδικό σχήμα (binary classification). Αντί για ένα πολυκλασικό μοντέλο, αναπτύχθηκαν τέσσερα ξεχωριστά μοντέλα, καθένα αφιερωμένο σε μία κατηγορία απορριμμάτων (γυαλί, μέταλλο, χαρτί, πλαστικό).

Η επιλογή αυτής της προσέγγισης έγινε διότι κάθε ESP32-CAM θα είναι υπεύθυνη αποκλειστικά για την ανίχνευση ενός τύπου υλικού, επιτρέποντας ελαφρύτερη υπολογιστική επιβάρυνση και βελτιστοποιημένη ακρίβεια. Παρακάτω αναλύεται η διαδικασία για την κλάση glass που ομοίως ακολουθήσαμε και για τις άλλες τρεις κλάσεις.

### 4.2 Συλλογή δεδομένων

Η διαδικασία συλλογής δεδομένων αποτέλεσε κρίσιμο στάδιο για την εκπαίδευση των μοντέλων ταξινόμησης, καθώς η ποιότητα και η ποικιλία του dataset καθορίζουν σε μεγάλο βαθμό την απόδοση και τη γενίκευση των αλγορίθμων. Στην ενότητα αυτή περιγράφονται αναλυτικά οι κινήσεις που πραγματοποιήθηκαν για την προετοιμασία του setup, τη διαδικασία λήψης εικόνων καθώς και τα κύρια ζητήματα που έπρεπε να αντιμετωπιστούν. Συνολικά για κάθε κλάση συλλέχθηκαν περίπου **N** εικόνες (π.χ. 5000).

- Για κάθε κατηγορία (π.χ. glass) δημιουργήθηκε dataset με δύο labels:
  1. **Glass=positive (1)**
  2. **Not Glass=negative (0)**

### 4.3 Προετοιμασία του Setup

Η επιτυχής συλλογή δεδομένων προϋπέθετε τον προσεκτικό σχεδιασμό του πειραματικού περιβάλλοντος:

- **Φωτισμός:** Οι ESP32-CAM είναι ιδιαίτερα ευαίσθητες σε χαμηλό ή ανομοιόμορφο φωτισμό, γεγονός που μπορεί να εισάγει θόρυβο ή σκιές. Για τον λόγο αυτό επιλέχθηκε ομοιόμορφος, διάχυτος φωτισμός με LED λωρίδες τοποθετημένες περιμετρικά των καμερών. Έγινε αποφυγή άμεσου ηλιακού φωτός ή φωτιστικών σωμάτων που προκαλούν τρεμόπαιγμα με την ολική κάλυψη της κατασκευής με στέγαστρο. Σταθεροποιήθηκε η ένταση, ενώ πραγματοποιήθηκαν και δοκιμές με μικρές παραλλαγές φωτεινότητας ώστε να ενισχυθεί η ποικιλία του dataset.



Εικόνα 4.3 Κάλυμμα κατασκευής

## Κεφάλαιο 4°

- **Ύψος και Απόσταση:** Οι κάμερες τοποθετήθηκαν σε ύψος 15 cm από τον ταινιοδρόμο, ώστε τα αντικείμενα μικρού μεγέθους (2–10 cm) να καλύπτουν σημαντικό τμήμα του καρέ. Η απόσταση μεταξύ των καμερών ορίστηκε στα 20 cm, και η ταχύτητα κίνησης του ταινιοδρόμου στα 5–10 cm/sec, προκειμένου να υπάρχει επαρκής χρόνος λήψης χωρίς να παρατηρείται θόλωση κίνησης.
- **Ρύθμιση Καμερών:** Οι κάμερες λειτούργησαν σε ανάλυση VGA (640x480), κατάλληλη για το ESP32 λόγω περιορισμένων πόρων. Οι παράμετροι φωτεινότητας, αντίθεσης και ισορροπίας λευκού ρυθμίστηκαν χειροκίνητα για μέγιστη συνέπεια. Για τη λήψη χρησιμοποιήθηκε το λογισμικό ESP32-CAM WebServer, αλλά και python script για δημιουργία και τροποποίηση των dataset με αποθήκευση εικόνων στον υπολογιστή.
- **Υλικά:** Για κάθε κατηγορία υλικού συλλέχθηκαν επαρκή δείγματα:
  - Γυαλί: κομμάτια γυαλιού, μπουκαλάκια.
  - Μέταλλο: αλουμινόχαρτο, μικρά μεταλλικά αντικείμενα, σύρμα ,κουτάκια, βίδες.
  - Χαρτί: φύλλα χαρτιού, εφημερίδες, διαφημιστικά φυλλάδια
  - Πλαστικό: σακούλες, πλαστικά καπάκια μπουκαλιών, καπάκια στυλό, καλαμάκια

Δόθηκε έμφαση στην ποικιλία χρώματος, σχήματος και υφής (π.χ. τσαλακωμένο χαρτί, γυαλιστερό πλαστικό) ώστε το dataset να είναι αντιπροσωπευτικό πραγματικών συνθηκών.

### 4.4 Διαδικασία Λήψης Εικόνων

Η συλλογή δεδομένων ακολούθησε συγκεκριμένο πρωτόκολλο:

1. **Εκκίνηση Ταινιοδρόμου:** Τα αντικείμενα τοποθετούνταν στην αρχή της γραμμής και περνούσαν διαδοχικά κάτω από όλες τις κάμερες.
2. **Συγχρονισμός Capture:** Κάθε κάμερα κατέγραφε το αντικείμενο όταν αυτό βρισκόταν ακριβώς κάτω από τον φακό. Χρησιμοποιήθηκε λογισμικό συγχρονισμού βάσει ταχύτητας ταινιοδρόμου (π.χ. κάθε 2–4 δευτερόλεπτα).
3. **Κατηγοριοποίηση Εικόνων:** Για κάθε κάμερα συλλέχθηκαν:
  - Θετικά παραδείγματα (π.χ. για την κάμερα “glass”, μόνο εικόνες γυαλιού).
  - Αρνητικά παραδείγματα (άλλα υλικά ή κενός ταινιοδρόμος).

Στόχος ήταν η συλλογή 1000–5000 εικόνων ανά binary classifier με ισορροπία 50% θετικών και 50% αρνητικών παραδειγμάτων.

4. **Ποικιλία Λήψεων:** Το ίδιο αντικείμενο καταγραφόταν επανειλημμένα με διαφορετικό προσανατολισμό (π.χ. περιστροφή 90°) για ενίσχυση της γενίκευσης.
5. **Διαφορετικά Sessions:** Εκτελέστηκαν πολλαπλές συνεδρίες με:
  - Κανονικές συνθήκες (καθαρά αντικείμενα, σταθερός φωτισμός).
  - Παραλλαγές (αλλαγές φωτισμού, διαφοροποιημένη ταχύτητα ταινιοδρόμου).
  - Χωρίς κίνηση για καλύτερες λήψεις

- Λήψεις χωρίς αντικείμενα (empty frames) για αύξηση των αρνητικών παραδειγμάτων.

#### 4.5 Προετοιμασία δεδομένων

Για την υλοποίηση του διαχωρισμού των δεδομένων και την τροφοδότησή τους στο μοντέλο, αξιοποιήθηκε το API `image_dataset_from_directory` της βιβλιοθήκης **TensorFlow**. Αρχικά, ορίστηκαν οι βασικοί κατάλογοι για το *training set*, το *validation set* και το *test set*, ενώ η ανάλυση των εικόνων καθορίστηκε σε **224×224 pixels**, σύμφωνα με τα βήματα προεπεξεργασίας που περιγράφηκαν. Το μέγεθος παρτίδας (batch size) ορίστηκε σε **16**, ώστε να διασφαλίζεται ισορροπία μεταξύ αποδοτικότητας μνήμης και ταχύτητας εκπαίδευσης. Η χρήση της μεθόδου **prefetch** επιτρέπει στο TensorFlow να προετοιμάζει τις επόμενες παρτίδες δεδομένων ενώ το μοντέλο βρίσκεται σε διαδικασία εκπαίδευσης, μειώνοντας έτσι τον χρόνο αναμονής και αυξάνοντας την αποδοτικότητα της διαδικασίας. Παρακάτω παρουσιάζεται η διαδικασία που ακολουθήθηκε πρώτα για την κλάση `glass` και στην συνέχεια πραγματοποιήθηκε η ίδια διαδικασία για τις άλλες τρείς `metal`, `paper`, `plastic`.

Παράδειγμα κώδικα:

```
• target_class = r"dataset/glass"
• base_dir = f"binary_{target_class}"
• train_dir = os.path.join(base_dir, "train")
• val_dir = os.path.join(base_dir, "val")
• test_dir = os.path.join(base_dir, "test")
•
•
• img_size = (224, 224)
• batch_size = 16
• # Δημιουργία συνόλου εκπαίδευσης
• train_ds = tf.keras.preprocessing.image_dataset_from_directory(
•     train_dir,
•     label_mode="binary",
•     image_size=img_size,
•     batch_size=batch_size
• )
•
•
• # Δημιουργία συνόλου επικύρωσης
• val_ds = tf.keras.preprocessing.image_dataset_from_directory(
•     val_dir,
•     label_mode="binary",
•     image_size=img_size,
•     batch_size=batch_size
• )
•
•
• # Δημιουργία συνόλου ελέγχου (χωρίς ανακάτεμα ώστε να διατηρηθεί η σειρά)
• test_ds = tf.keras.preprocessing.image_dataset_from_directory(
•     test_dir,
•     label_mode="binary",
```

```

• image_size=img_size,
• batch_size=batch_size,
• shuffle=False
• )
•
• # Εμφάνιση των κλάσεων του dataset
• print(train_ds.class_names)
•
• # Βελτιστοποίηση με prefetching για ταχύτερη φόρτωση
• train_ds = train_ds.prefetch(tf.data.AUTOTUNE)
• val_ds = val_ds.prefetch(tf.data.AUTOTUNE)
• test_ds = test_ds.prefetch(tf.data.AUTOTUNE)

```

Εικόνα 4.5 Κώδικας προετοιμασίας δεδομένων

#### 4.6 Δημιουργία του Εξειδικευμένου Μοντέλου

Μετά την ολοκλήρωση της προετοιμασίας των δεδομένων, σχεδιάστηκε ένα **συνελικτικό νευρωνικό δίκτυο (Convolutional Neural Network, CNN)** μικρής κλίμακας, το οποίο χρησιμοποιήθηκε για το πρόβλημα της **δυναδικής ταξινόμησης** (binary classification). Το δίκτυο σχεδιάστηκε με γνώμονα τη χαμηλή υπολογιστική πολυπλοκότητα, ώστε να μπορεί να εφαρμοστεί σε πειραματικά σενάρια με μικρά ή περιορισμένα σε μέγεθος σύνολα δεδομένων. Ο σκοπός ήταν να δημιουργηθεί ένα απλό αλλά αποδοτικό μοντέλο, με περιορισμένο αριθμό παραμέτρων, ώστε να επιτευχθεί ισορροπία μεταξύ ακρίβειας και υπολογιστικού κόστους.

Η αρχιτεκτονική του μοντέλου περιλαμβάνει τα παρακάτω στάδια:

Η είσοδος του δικτύου ορίζεται ως τρισδιάστατος πίνακας διαστάσεων  $224 \times 224 \times 3$ , που αντιστοιχεί σε εικόνες χρώματος RGB μεγέθους  $224 \times 224$  pixels. Το δίκτυο αποτελείται από τα εξής δομικά στάδια:

1. **Πρώτο συνελικτικό στρώμα (Conv2D)** Το πρώτο επίπεδο αποτελείται από 6 φίλτρα συνελίξεως με πυρήνες διαστάσεων  $3 \times 3$ , εφαρμοσμένα με *padding* τύπου *same*, ώστε οι χωρικές διαστάσεις της εξόδου να παραμένουν ίδιες με της εισόδου. Η συνάρτηση ενεργοποίησης είναι η ReLU (Rectified Linear Unit), η οποία εισάγει μη γραμμικότητα. Το στρώμα αυτό ανιχνεύει βασικά χαμηλού επιπέδου χαρακτηριστικά, όπως ακμές και γωνίες. Στη συνέχεια, εφαρμόζεται στρώμα *max pooling* με παράθυρο  $2 \times 2$ , το οποίο μειώνει κατά το ήμισυ τις χωρικές διαστάσεις της αναπαράστασης, οδηγώντας από τις αρχικές διαστάσεις (224, 224, 6) σε (112, 112, 6). Η μείωση αυτή συμβάλλει τόσο στη μείωση του υπολογιστικού κόστους όσο και στη μερική αφαίρεση θορύβου.
2. **Δεύτερο συνελικτικό στρώμα (Conv2D)** Το δεύτερο επίπεδο διαθέτει 12 φίλτρα  $3 \times 3$  με ReLU ενεργοποίηση. Η έξοδος αυτού του στρώματος έχει διαστάσεις (112, 112, 12). Ακολουθεί νέο στρώμα *max pooling*  $2 \times 2$ , το οποίο μειώνει περαιτέρω τις διαστάσεις σε (56, 56, 12). Το στάδιο αυτό εστιάζει σε χαρακτηριστικά μεσαίου επιπέδου, όπως υφές και μοτίβα.
3. **Global Average Pooling** Η τελική χωρική αναπαράσταση των χαρακτηριστικών (56, 56, 12) συμπυκνώνεται σε έναν μονοδιάστατο διανυσματικό χώρο μέσω ενός στρώματος *Global Average Pooling*, το οποίο υπολογίζει τον μέσο όρο κάθε χάρτη χαρακτηριστικών. Με τον τρόπο αυτό, το δίκτυο εξάγει ένα διάνυσμα 12 διαστάσεων, μειώνοντας δραστικά τον αριθμό παραμέτρων και περιορίζοντας τον κίνδυνο υπερπροσαρμογής.

4. Πλήρως συνδεδεμένο στρώμα (Dense) Το διάνυσμα που προκύπτει από το στάδιο pooling τροφοδοτείται σε ένα πλήρως συνδεδεμένο επίπεδο με 24 νευρώνες και ReLU ενεργοποίηση. Ο σκοπός του είναι ο συνδυασμός των εξαγόμενων χαρακτηριστικών και η δημιουργία μη γραμμικών συσχετίσεων μεταξύ τους.
5. Στρώμα εξόδου (Output Layer) Το τελικό στρώμα αποτελείται από έναν νευρώνα με συνάρτηση ενεργοποίησης sigmoid, ο οποίος αποδίδει πιθανότητα ένταξης της εικόνας στην κλάση-στόχο. Καθώς το πρόβλημα είναι δυαδικής ταξινόμησης, η έξοδος παίρνει τιμές στο διάστημα (0, 1).

Η συνολική αρχιτεκτονική υλοποιείται ως εξής:

```
from tensorflow.keras import layers, models

def create_small_cnn_binary(input_size=224):
    input_shape = (input_size, input_size, 3)
    model = models.Sequential([
        # Πρώτο στρώμα Conv2D
        layers.Conv2D(6, (3, 3), activation='relu', padding='same', input_shape=input_shape),
        layers.MaxPooling2D(2, 2),

        # Δεύτερο στρώμα Conv2D
        layers.Conv2D(12, (3, 3), activation='relu', padding='same'),
        layers.MaxPooling2D(2, 2),

        layers.GlobalAveragePooling2D(),
        # Dense στρώμα
        layers.Dense(24, activation='relu'),
        # Τελικό στρώμα για δυαδική ταξινόμηση: 1 νευρώνας με sigmoid
        layers.Dense(1, activation='sigmoid')
    ])
    return model

# Παράδειγμα χρήσης
model = create_small_cnn_binary(input_size=224)
model.summary()
```

Εικόνα 4.6.1 Κώδικας του CNN μοντέλου

Το μοντέλο αυτό αποτελείται συνολικά από 1165 παραμέτρους, όπως φαίνεται στο model.summary().

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 224, 224, 6)	168
max_pooling2d_2 (MaxPooling2D)	(None, 112, 112, 6)	0
conv2d_3 (Conv2D)	(None, 112, 112, 12)	668
max_pooling2d_3 (MaxPooling2D)	(None, 56, 56, 12)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 12)	0
dense_2 (Dense)	(None, 24)	324
dense_3 (Dense)	(None, 1)	25

Total params: 1,165 (4.55 KB)  
 Trainable params: 1,165 (4.55 KB)  
 Non-trainable params: 0 (0.00 B)

Εικόνα 4.6.2 Οπτικοποίηση μοντέλου

## Κεφάλαιο 4°

Ο χαμηλός αριθμός παραμέτρων καθιστά το δίκτυο κατάλληλο για εκπαίδευση σε περιορισμένο υπολογιστικό περιβάλλον (π.χ. προσωπικό υπολογιστή ή laptop χωρίς GPU), ενώ εξακολουθεί να διατηρεί την ικανότητα να μαθαίνει ουσιαστικά χαρακτηριστικά για το πρόβλημα ταξινόμησης.

### 4.7 Εκπαίδευση του μοντέλου

Αφού ορίστηκε η αρχιτεκτονική του μοντέλου, ακολούθησε η διαδικασία της **εκπαίδευσης**. Η εκπαίδευση πραγματοποιείται με την επαναληπτική ενημέρωση των βαρών του νευρωνικού δικτύου μέσω της **οπισθοδιάδοσης (backpropagation)**, με σκοπό την ελαχιστοποίηση της συνάρτησης κόστους.

Για το συγκεκριμένο πρόβλημα δυαδικής ταξινόμησης επιλέχθηκαν τα εξής:

- **Συνάρτηση απώλειας (Loss Function):** Χρησιμοποιήθηκε η *binary cross-entropy*, καθώς είναι η καταλληλότερη για προβλήματα με δύο κλάσεις, αφού μετρά την απόκλιση μεταξύ των πραγματικών ετικετών και των πιθανοτήτων που προβλέπει το δίκτυο.
- **Βελτιστοποιητής (Optimizer):** Εφαρμόστηκε ο **Adam** (Adaptive Moment Estimation), καθώς συνδυάζει τα πλεονεκτήματα των SGD με momentum και RMSProp, εξασφαλίζοντας γρήγορη και σταθερή σύγκλιση. Το αρχικό *learning rate* ορίστηκε σε  $1.875 \times 10^{-4}$ , τιμή που επιλέχθηκε εμπειρικά ώστε να υπάρχει ισορροπία ανάμεσα σε ταχύτητα και ακρίβεια εκπαίδευσης.
- **Μετρική αξιολόγησης (Metric):** Χρησιμοποιήθηκε η *accuracy*, που αντιπροσωπεύει το ποσοστό σωστών προβλέψεων επί του συνόλου.
- **Callback – Προσαρμογή Learning Rate:** Για να αποφευχθεί στασιμότητα στη μάθηση και να βελτιωθεί η γενίκευση, αξιοποιήθηκε το `ReduceLRonPlateau`. Το callback αυτό παρακολουθεί τη *val\_loss* και μειώνει το *learning rate* κατά 50% (`factor=0.5`) όταν για 4 συνεχόμενες εποχές δεν παρατηρείται βελτίωση, με επιπλέον *cooldown* 2 εποχών. Το ελάχιστο επιτρεπτό *learning rate* ορίστηκε στο  $1e-6$ .

Η εκπαίδευση πραγματοποιήθηκε για **20 εποχές**, με τα σύνολα *training* και *validation* που δημιουργήθηκαν στο προηγούμενο στάδιο.

Ο αντίστοιχος κώδικας παρουσιάζεται παρακάτω:

```
from tensorflow.keras import models, layers
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ReduceLRonPlateau

# Compile με αρχικό learning rate
initial_lr = 1.875e-4
model.compile(
    optimizer=Adam(learning_rate=initial_lr),
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# ReduceLRonPlateau με «περισσότερο αέρα»
reduce_lr = ReduceLRonPlateau(
    monitor='val_loss',
```

```

factor=0.5,
patience=4,
cooldown=2,
min_lr=1e-6,
verbose=1
)

# Εκπαίδευση
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=20,
    callbacks=[reduce_lr]
)

```

Εικόνα 4.7 Εκπαίδευση μοντέλου

Με την παραπάνω διαδικασία το δίκτυο εκπαιδεύεται προοδευτικά, προσαρμόζοντας τα βάρη του ώστε να ελαχιστοποιήσει την απώλεια στο σύνολο εκπαίδευσης, ενώ ταυτόχρονα ελέγχεται η απόδοσή του στο *validation set* για να αποφευχθεί το φαινόμενο της υπερπροσαρμογής (*overfitting*).

#### 4.8 Οπτικοποίηση Ακρίβειας και Απώλειας του Εκπαιδευμένου Μοντέλου

Η παρακολούθηση των μετρικών εκπαίδευσης σε κάθε εποχή αποτελεί σημαντικό εργαλείο για την κατανόηση της διαδικασίας μάθησης και την αξιολόγηση της συμπεριφοράς του μοντέλου. Συγκεκριμένα, μελετήθηκαν δύο βασικοί δείκτες:

- **Ακρίβεια (Accuracy):** δείχνει το ποσοστό των σωστών προβλέψεων τόσο στο *training set* όσο και στο *validation set*.
- **Συνάρτηση Απώλειας (Loss):** εκφράζει το σφάλμα του μοντέλου κατά την πρόβλεψη. Η μείωση της απώλειας σηματοδοτεί καλύτερη προσαρμογή του μοντέλου στα δεδομένα.

Η οπτικοποίηση αυτών των μετρικών συμβάλλει στον εντοπισμό φαινομένων **υπερπροσαρμογής (*overfitting*)** ή **υποπροσαρμογής (*underfitting*)**:

- Σε περίπτωση που η *training accuracy* συνεχίζει να βελτιώνεται ενώ η *validation accuracy* σταθεροποιείται ή μειώνεται, τότε το μοντέλο υπερπροσαρμόζεται.
- Αντίστοιχα, αν τόσο η ακρίβεια όσο και η απώλεια παρουσιάζουν μικρή πρόοδο, ενδέχεται να υπάρχει υποπροσαρμογή.

Ο κώδικας που ακολουθεί δημιουργεί δύο γραφήματα, ένα για την ακρίβεια και ένα για την απώλεια, με βάση τα δεδομένα που αποθηκεύτηκαν στο αντικείμενο *history* κατά την εκπαίδευση:

```

import matplotlib.pyplot as plt

# Δημιουργία figure με 2 υπο-γραφήματα (accuracy & loss)
plt.figure(figsize=(12, 5))

# Ακρίβεια
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')

```

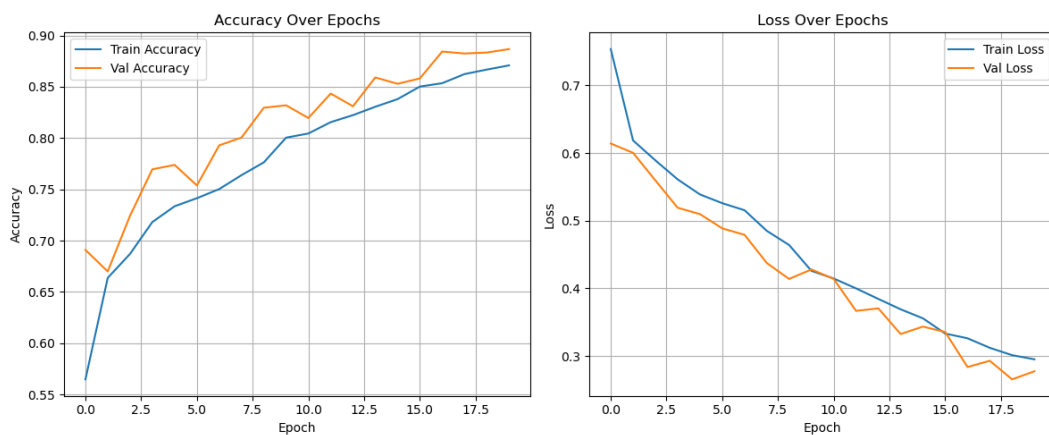
## Κεφάλαιο 4°

```
plt.title('Accuracy Over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid()

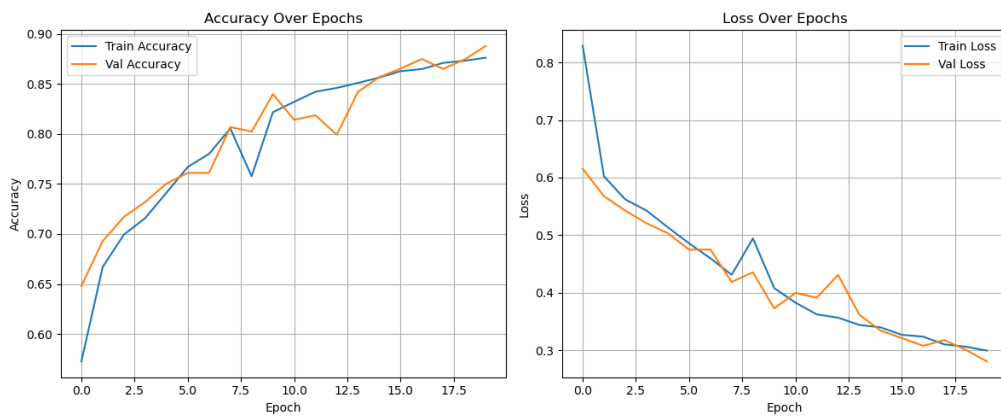
# Απώλεια
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title('Loss Over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid()

plt.tight_layout()
plt.show()
```

Εικόνα 4.8.1 Κώδικας οπτικοποίησης Ακρίβειας

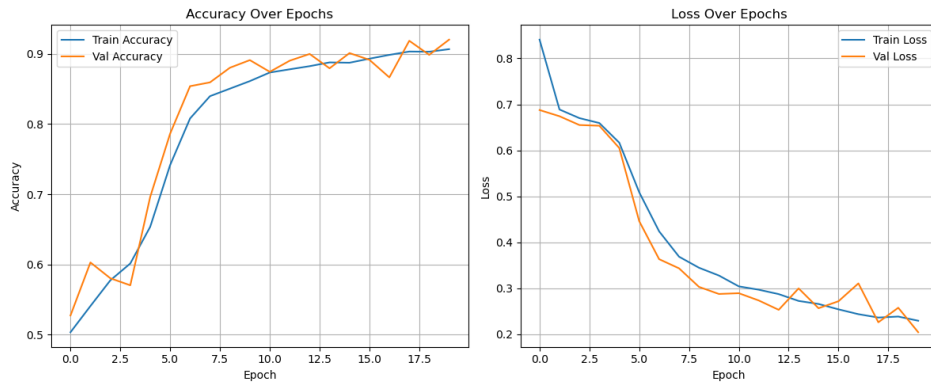


Εικόνα 4.8.2 Κλάση glass

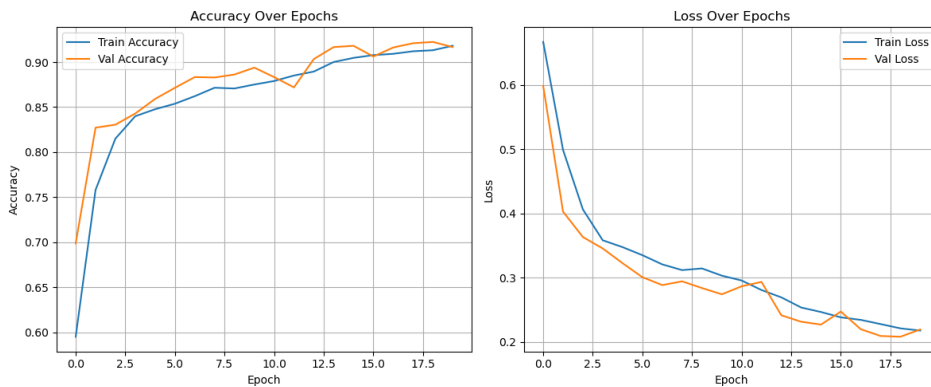


Εικόνα 4.8.3 Κλάση metal

## Δημιουργία μοντέλου Υπολογιστικής Όρασης για κατηγοριοποίηση αντικειμένων



Εικόνα 4.8.4 Κλάση paper



Εικόνα 4.8.5 Κλάση plastic

Το πρώτο διάγραμμα καταγράφει την εξέλιξη της ακρίβειας,

- Η **καμπύλη εκπαίδευσης** δείχνει την πρόοδο του μοντέλου στο training set.
- Η **καμπύλη επικύρωσης** αποτυπώνει την ικανότητα γενίκευσης του μοντέλου σε μη-εκπαιδευμένα δεδομένα.

Ιδανικά, οι δύο καμπύλες ακολουθούν παρόμοια πορεία, γεγονός που υποδηλώνει ότι το μοντέλο μαθαίνει ουσιαστικά χαρακτηριστικά χωρίς να υπερπροσαρμόζεται. Σε περίπτωση που η ακρίβεια εκπαίδευσης συνεχίζει να αυξάνεται ενώ η ακρίβεια επικύρωσης παραμένει στάσιμη ή μειώνεται, τότε εμφανίζεται φαινόμενο overfitting.

Ενώ το δεύτερο δείχνει την πορεία της απώλειας.

Η συνδυαστική τους ανάλυση παρέχει ουσιαστική κατανόηση για την αποδοτικότητα και τη γενίκευση του μοντέλου.

## 4.9 Προβλέψεις του εκπαιδευμένου μοντέλου

Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης, το τελικό μοντέλο αξιολογήθηκε στο ανεξάρτητο **test set**. Το test set δεν χρησιμοποιήθηκε σε κανένα στάδιο εκπαίδευσης ή επικύρωσης, με αποτέλεσμα να παρέχει μια αμερόληπτη εκτίμηση της γενίκευσης του μοντέλου σε άγνωστα δεδομένα.

Η αξιολόγηση πραγματοποιείται με τη μέθοδο `model.evaluate()`, η οποία επιστρέφει την τιμή της **συνάρτησης κόστους (loss)** και τη **μετρική ακρίβειας (accuracy)** πάνω στο test set.

Ο κώδικας που ακολουθεί υλοποιεί το παραπάνω βήμα:

```
loss, accuracy = model.evaluate(test_ds)
print(f"Test Accuracy: {accuracy:.4f} | Loss: {loss:.4f}")
151/151 ————— 24s 157ms/step - accuracy: 0.8755 - loss: 0.3177
Test Accuracy: 0.8784 | Loss: 0.2857
```

Εικόνα 4.9 Αξιολόγηση μοντέλου στο testset

## 4.10 Αναφορά των αποτελεσμάτων και Confusion Matrix

Για την πληρέστερη αποτίμηση της απόδοσης του εκπαιδευμένου μοντέλου, πέρα από την απλή μέτρηση της ακρίβειας (accuracy), υπολογίστηκαν και **πολυδιάστατες μετρικές ταξινόμησης**. Συγκεκριμένα, χρησιμοποιήθηκαν:

- **Confusion Matrix:** απεικονίζει τον αριθμό των σωστών και λανθασμένων προβλέψεων για κάθε κλάση. Μέσα από τον πίνακα αυτό μπορούμε να διακρίνουμε όχι μόνο το συνολικό ποσοστό επιτυχίας, αλλά και σε ποιες κατηγορίες το μοντέλο εμφανίζει σφάλματα.
- **Classification Report:** παρέχει μετρικές όπως **precision**, **recall** και **F1-score**, που επιτρέπουν πιο λεπτομερή αξιολόγηση σε προβλήματα δυαδικής ταξινόμησης.
  - **Precision:** ποσοστό των σωστά προβλεφθέντων θετικών περιπτώσεων επί του συνόλου των προβλεφθέντων θετικών.
  - **Recall:** ποσοστό των σωστά προβλεφθέντων θετικών περιπτώσεων επί του συνόλου των πραγματικών θετικών.
  - **F1-score:** ο αρμονικός μέσος των precision και recall, που συνδυάζει τις δύο μετρικές σε μία.

Ο παρακάτω κώδικας παρουσιάζει τη διαδικασία πρόβλεψης των ετικετών του test set, την κατασκευή του confusion matrix, καθώς και την εξαγωγή του classification report:

```
y_pred_prob = model.predict(test_ds)
y_pred = (y_pred_prob > 0.5).astype("int32").flatten()

y_true = np.concatenate([y.numpy() for x, y in test_ds], axis=0)

labels = [f"not_{target_class}", target_class]
cm = confusion_matrix(y_true, y_pred)

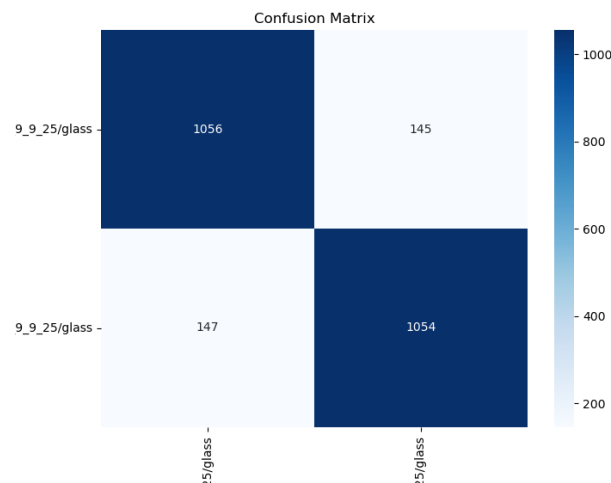
plt.figure(figsize=(4, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels,
yticklabels=labels)
```

```
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()

print(classification_report(y_true, y_pred, target_names=labels))
```

Εικόνα 4.10.1 Κώδικας Confusion matrix και Classification report

Η **Confusion Matrix** απεικονίζει οπτικά την απόδοση του μοντέλου, όπου τα διαγώνια στοιχεία αντιστοιχούν στις σωστές προβλέψεις, ενώ τα εκτός διαγωνίου δείχνουν τα λάθη ταξινόμησης. Παράλληλα, το **classification report** παρέχει μία πληρέστερη εικόνα της ακρίβειας του μοντέλου ανά κλάση, γεγονός που επιτρέπει την καλύτερη κατανόηση των δυνατών και αδύνατων σημείων του. Παρακάτω αποτυπώνεται το Confusion Matrix και Classification Report της κλάσης glass ενδεικτικά και στο κεφάλαιο 5 στην ενότητα 5.8 αναλυτικά συγκρίνονται και αποτυπώνονται όλες οι κλάσεις.



```

precision    recall  f1-score   support

not_final_augmentedV2_datasets_9_9_25/glass      0.88     0.88     0.88     1201
final_augmentedV2_datasets_9_9_25/glass          0.88     0.88     0.88     1201

   accuracy          0.88     2402
  macro avg          0.88     2402
 weighted avg          0.88     2402
    
```

Εικόνα 4.10.2 Οπτικοποίηση Confusion matrix και Classification report

#### 4.11 Επίλογος

Συνολικά αναπτύχθηκαν τέσσερα CNN binary classification μοντέλα, καθένα αφιερωμένο σε μία κατηγορία υλικού. Η υλοποίηση με το ESP32-CAM επιτρέπει αποκεντρωμένη επεξεργασία εικόνας, μειώνοντας τον φόρτο σε ένα κεντρικό σύστημα. Τα αποτελέσματα δείχνουν ότι η προσέγγιση είναι αποδοτική και μπορεί να ενσωματωθεί σε ένα σύστημα **έξυπνης ανακύκλωσης**.

## Κεφάλαιο 5ο: Πειράματα πάνω στην κατασκευή και Ολοκλήρωση – Αποτελέσματα

### 5.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζονται οι πειραματικές διαδικασίες που αφορούν την **σωστή λειτουργία του ταινιόδρομου με τέσσερις εξόδους** και την ολοκλήρωση των μοντέλων με τις ESP32-CAM για **live αναγνώριση υλικών**. Παρουσιάζονται οι φάσεις λειτουργίας του συστήματος, τα αποτελέσματα των δοκιμών και η αξιολόγηση της απόδοσης.

### 5.2 Περιγραφή Κατασκευής

- **Ταινιόδρομος:** Κύρια ζώνη μεταφοράς υλικών με **τέσσερις εξόδους**.
- **Κάμερες ESP32-CAM:** Μία κάμερα πριν από κάθε έξοδο για συλλογή εικόνων σε πραγματικό χρόνο.
- **PC, Laptop, Μικροϋπολογιστής / Python script:** Λήψη των frames, εφαρμογή του **binary μοντέλου** για κάθε κατηγορία και έλεγχος ενεργοποίησης servo.
- **Servo motors:** Ένας servo για κάθε έξοδο, ενεργοποιείται μόνο όταν το αντίστοιχο υλικό ανιχνευτεί.



Εικόνα 5.2.1 Ολοκλήρωση του ταινιόδρομου



Εικόνα 5.2.2 Επιπρόσθετο στέγαστρο

### 5.3 Φάσεις Λειτουργίας του Συστήματος

Το σύστημα ελέγχου του κεντρικού ταινιόδρομου μικρής κλίμακας βασίζεται στις τέσσερις ESP32-CAM κάμερες, στα CNN binary μοντέλα αναγνώρισης υλικών (TensorFlow/Keras) και στο Arduino που οδηγεί σερβομηχανισμούς. Παρακάτω αναλύονται διεξοδικά οι φάσεις λειτουργίας, αντιστοιχίζοντάς τες ρητά στον παρεχόμενο κώδικα Python.

### 5.4 Φάση 1: Σύλληψη εικόνας (Frame Capture)

**Σκοπός:** Απόκτηση πρόσφατου καρέ από κάθε ESP32-CAM με ελάχιστη καθυστέρηση και ανθεκτικότητα σε προσωρινά σφάλματα δικτύου.

**Κύρια υποσυστήματα & ροή δεδομένων:**

- **Πηγή:** ESP32-CAM (HTTP endpoint `http://<ip>/capture`).
- **Μεταφορά:** HTTP GET ανά ~βρόχο μέσω requests.
- **Αποκωδικοποίηση:** Μετατροπή JPEG → OpenCV BGR (`cv2.imdecode`).
- **Αποθήκευση:** Νήμα *FrameGrabber* ανά κάμερα με κοινόχρηστο buffer (τελευταίο καρέ).

```

class FrameGrabber(threading.Thread):
    def __init__(self, url):
        super().__init__()
        self.url = url
        self.frame = None
        self.stopped = False
        self.lock = threading.Lock()

    def run(self):
        while not self.stopped:
            try:
                resp = requests.get(self.url, timeout=5)
                if resp.status_code == 200:
                    img_array = np.frombuffer(resp.content, np.uint8)
                    img = cv2.imdecode(img_array, cv2.IMREAD_COLOR)
                    with self.lock:
                        self.frame = img
            except:
                pass

    def read(self):
        with self.lock:
            return self.frame.copy() if self.frame is not None else None

    def stop(self):
        self.stopped = True
    
```

Εικόνα 5.4.1 Κώδικας σύλληψης εικόνας

### Χαρτογράφηση στον κώδικα:

- Κλάση FrameGrabber:
  - run(): επαναλαμβανόμενα requests.get με timeout=5 sec, έλεγχος status\_code==200, np.frombuffer → cv2.imdecode.
  - Συγχρονισμός πρόσβασης με threading.Lock (μέθοδοι read()/run()), αποφυγή data races.
  - Στρατηγική *last-writer wins*: αποθηκεύεται μόνο το πιο πρόσφατο καρέ (self.frame).
- Εκκίνηση ανά κάμερα μέσα στη CameraModelHandler.run().

### Χρονισμός & επιδόσεις:

- Ο ρυθμός ανανέωσης εξαρτάται από latency δικτύου και αποδοτικότητα ESP32-CAM· δεν υπάρχει τεχνητή καθυστέρηση, άρα επιδιώκεται **μικρή καθυστέρηση** (low-latency streaming μέσω polling).
- Η αποκωδικοποίηση JPEG είναι **CPU-bound**· σε πολυπύρηννα συστήματα η παράλληλη εκτέλεση 4 νημάτων εκμεταλλεύεται τον διαθέσιμο πυρήνα ανά κάμερα.

### Αξιοπιστία & ανθεκτικότητα:

- Η try/except στον FrameGrabber.run() απορροφά στιγμιαία σφάλματα (timeouts, packet loss) και συνεχίζει.
- Ο buffer είναι «volatile» (χωρίς ουρά) για να μην καθυστερεί το καρέ προς εκτίμηση (avoid backlog/lag).

### Δείκτες/μετρικές:

- Μέση καθυστέρηση σύλληψης με ρύθμιση VGA(640x480) είναι περίπου από 200 έως 400 ms, ποσοστό αποτυχημένων αιτημάτων, ρυθμός καρέ ανά κάμερα.

### Προτεινόμενες βελτιώσεις:

- Προαιρετική χρήση **MJPEG stream** αντί για επαναλαμβανόμενα GET (μείωση overhead).
- Προσθήκη **εκθετικής οπισθοχώρησης** σε συνεχόμενα timeouts· καταγραφή σφαλμάτων.
- Watchdog ανά κάμερα: αν δεν ενημερωθεί καρέ για X sec, σήμανση «camera offline».

## 5.5 Φάση 2: Αναγνώριση υλικού (Inference)

**Σκοπός:** Κατάταξη του αντικειμένου σε μία εκ των κατηγοριών (γυαλί, μέταλλο, χαρτί, πλαστικό) με χρήση εκπαιδευμένων binary CNN μοντέλων και ενδεικτικά multiclass CNN μοντέλου όπου θα γίνει και στατιστική σύγκριση των δύο κατηγοριών αναλυτικά στην ενότητα 5.8 του παρόντος κεφαλαίου.

### Προεπεξεργασία εικόνας:

- cv2.resize σε **IMG\_SIZE=128**.
- Μετατροπή **BGR** → **RGB** (cv2.cvtColor).
- Μετατροπή σε float32 και **διάσταση παρτίδας** 1×128×128×3.

### Μηχανισμός απορρίψεων (προ-φίλτρο):

- Υπολογισμός μέσης φωτεινότητας (grayscale): αν `mean_intensity > 230`, θεωρείται «NO OBJECT» (υπερ-φωτεινό πλαίσιο/άδειο background) και παρακάμπτεται το inference.

```
class CameraModelHandler:
    def __init__(self, ip, label, model_path, cam_index, arduino):
        self.ip = ip
        self.label = label
        self.model = load_model(model_path, compile=False)
        self.cam_index = cam_index
        self.arduino = arduino
        self.capture_url = f"http://{ip}/capture"
        self.buffer = deque(maxlen=5)
        self.servo_open = False
        self.servo_open_time = 0

    def preprocess(self, img):
        try:
            img_resized = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
            img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
            img_array = np.expand_dims(img_rgb.astype(np.float32), axis=0)
            return img_array
        except:
            return None

    def predict(self, tensor):
        pred = self.model.predict(tensor, verbose=0)[0][0]
        self.buffer.append(pred)
        return np.mean(self.buffer)
```

Εικόνα 5.5.1 Κώδικας αναγνώρισης υλικού

### Χαρτογράφηση στον κώδικα:

- `CameraModelHandler.__init__`: φόρτωση μοντέλου με `load_model(model_path, compile=False)`.
- `preprocess(img)`: υλοποιεί την προεπεξεργασία.
- `predict(tensor)`: `self.model.predict(...)[0][0]` → **πιθανότητα** για τη συγκεκριμένη ετικέτα της κάμερας.
- **Εξομάλυνση/αποθορυβοποίηση**: κυλιόμενο μέσο όρο `deque(maxlen=5)`. Περιορίζει τυχαίες διακυμάνσεις, μειώνει ψευδώς θετικά.

### Απεικόνιση κατάστασης:

- Overlay με `cv2.putText` δείχνει LABEL: p% με πράσινο (>0.5) ή κόκκινο (≤0.5) χρώμα.

## Κεφάλαιο 5°

### Επιλογικός λόγος κατώφλιου:

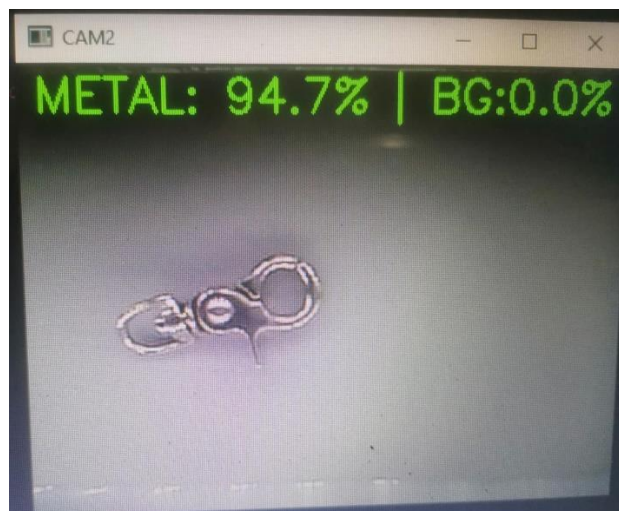
- Σταθερό κατώφλι **0.5** (Bayes decision rule με ίσες απώλειες). Μπορεί να ρυθμιστεί ανά κάμερα ανάλογα με ROC/PR καμπύλες.

### Δείκτες/μετρικές:

- Ακρίβεια/ανάκτηση ανά κατηγορία, F1, ROC AUC, ρυθμός ψευδώς θετικών/αρνητικών μετά την εξομάλυνση 5 δειγμάτων.



Εικόνα 5.5.2 live Inference binary μοντέλου



Εικόνα 5.5.3 live Inference multiclass μοντέλου

### Προτεινόμενες βελτιώσεις:

- Κανονικοποίηση τιμών εισόδου (π.χ. division by 255) και τυποποίηση βάσει mean/std του dataset.
- Προσαρμοστικό κατώφλι ανά ροή (temperature scaling / Platt scaling).
- Batch inference ανά κάμερα για καλύτερη απόδοση GPU/CPU (αν συντρέχει).

### 5.6 Φάση 3: Ενεργοποίηση εξόδου (Servo Control)

**Σκοπός:** Όταν ανιχνευθεί υλικό από την κάμερα που αντιστοιχεί στη γραμμή ταξινόμησης, να ενεργοποιηθεί ο αντίστοιχος σερβομηχανισμός για **σταθερό χρονικό διάστημα** ώστε να κατευθυνθεί το αντικείμενο στο σωστό ταινιόδρομο εξόδου.

**Διάυλος ελέγχου:**

```
def send_command(self, open_servo=True):
    cmd = f"CAM{self.cam_index}:{'OPEN' if open_servo else 'CLOSE'}\n"
    self.arduino.write(cmd.encode())
    print(f"[ARDUINO] Sent: {cmd.strip()}")

def run(self):
    grabber = FrameGrabber(self.capture_url)
    grabber.start()
    print(f"🌀 CAM{self.cam_index} - {self.label} started")

    window_name = f"CAM{self.cam_index} | {self.label}"
    try:
        while True:
            frame = grabber.read()
            if frame is not None:
                gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                mean_intensity = np.mean(gray)

                if mean_intensity > 230:
                    cv2.putText(frame, "NO OBJECT", (10, 30),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (150, 150,
150), 2)

                else:
                    tensor = self.preprocess(frame)
                    if tensor is not None:
                        confidence = self.predict(tensor)
                        text = f"{self.label.upper()}: {confidence:.1%}"
                        color = (0, 255, 0) if confidence > 0.55 else (0,
0, 255)

                        cv2.putText(frame, text, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, color, 2)

                        now = time.time()
                        if confidence > 0.55 and not self.servo_open:
                            self.send_command(True)
                            self.servo_open = True
                            self.servo_open_time = now

                        if self.servo_open and (now - self.servo_open_time
> SERVO_OPEN_TIME):
                            self.send_command(False)
                            self.servo_open = False
```

```
cv2.imshow(window_name, frame)
```

Εικόνα 5.6.1 Κώδικας ενεργοποίησης εξόδου

#### Χαρτογράφηση στον κώδικα:

- Μέθοδος `send_command(open_servo=True)` αποστέλλει την εντολή και καταγράφει στο `stdout`.
- Λογική ενεργοποίησης μέσα στην `run()` του handler:
  1. Αν `confidence > 0.5` και `servo_open == False` ⇒ OPEN και αποθήκευση `servo_open_time = now`.
  2. Αν `servo_open == True` και `now - servo_open_time > SERVO_OPEN_TIME` ⇒ CLOSE.
- Το `SERVO_OPEN_TIME` είναι **σταθερά συστήματος** (10 s).

#### Αποθορυβοποίηση & debounce:

- Η εξομάλυνση κυλιόμενου μέσου όρου λειτουργεί σαν **temporal debounce**, απαιτώντας διαρκή εμπιστοσύνη για να εκκινήσει ο servo.
- Η απαγόρευση «διπλού OPEN» (έλεγχος `not self.servo_open`) αποτρέπει spam εντολών.

#### Σχέση με ταινιοδρόμο:

- Το πάγιο χρονικό παράθυρο 10 s υποθέτει συγκεκριμένη **γραμμική ταχύτητα ταινίας** και γεωμετρία ρίψης. Σε αλλαγή ταχύτητας απαιτείται επαναρύθμιση.

#### Ασφάλεια & ακεραιότητα:

- Το πρωτόκολλο εντολών είναι ευανάγνωστο αλλά ανεπιβεβαίωτο· προτείνεται **ACK/NACK** από Arduino και **checksum/frame ID** για ανίχνευση απωλειών.
- Προστασία από κορεσμό: αν ο Arduino δεν εκτελέσει CLOSE, το Python loop θα επαναστείλει μετά την παρέλευση του παραθύρου (με βάση τη λογική φάσης 4).

#### Δείκτες/μετρικές:

- Χρόνος αντήχησης (latency) από inference → OPEN (ms), διάρκεια ενεργοποίησης, ποσοστό επιτυχημένων ενεργοποιήσεων, απώλειες σειριακών πακέτων.

#### Προτεινόμενες βελτιώσεις:

- **Κλειστός βρόχος** με αισθητήρα θέσης servo ή οπτικό αισθητήρα διέλευσης αντικειμένου.
- Χρονισμός προπορείας (**lead time**) βάσει αισθητήρα παρουσίας/encoder ταινίας αντί σταθερού 10 s.

## 5.7 Φάση 4: Συνεχής Παρακολούθηση και Διόρθωση

**Σκοπός:** Αδιάλειπτη λειτουργία σε πραγματικό χρόνο με ορατότητα κατάστασης και αυτοδιόρθωση σε παροδικά προβλήματα.

**Στοιχεία υλοποίησης:**

- **Πολυνηματικότητα:** Ένα thread FrameGrabber ανά κάμερα + ένα thread εκτέλεσης CameraModelHandler.run() ανά handler· συνολικά 4 παράλληλες ροές επεξεργασίας.
- **Παράθυρο εξομάλυνσης** (deque 5 τιμών) για σταθεροποίηση αποφάσεων.
- **Οπτική επίβλεψη:** cv2.imshow ανά κάμερα με overlay κατάστασης και πλήκτρο q για τερματισμό ανά παράθυρο.
- **Διαχείριση ζωής:** Καθαρός τερματισμός: grabber.stop(), cv2.destroyWindow, κλείσιμο σειριακής θύρας.

```

• cv2.imshow(window_name, frame)
•
•     if cv2.waitKey(1) & 0xFF == ord('q'):
•         break
•     finally:
•         grabber.stop()
•         cv2.destroyWindow(window_name)
•         print(f"🛑 CAM{self.cam_index} - stopped")
•
• # === MAIN ΕΚΚΙΝΗΣΗ ===
• if __name__ == "__main__":
•     print("🔌 Connecting to Arduino...")
•     arduino = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
•     time.sleep(2)
•     print("✅ Arduino ready")
•
•     handlers = [CameraModelHandler(**cfg, arduino=arduino) for cfg in CAM_CONFIGS]
•
• # === ΕΚΚΙΝΗΣΗ ΠΑΡΑΛΛΗΛΗ ΓΙΑ ΟΛΕΣ ΤΙΣ ΚΑΜΕΡΕΣ ===
• threads = [threading.Thread(target=h.run) for h in handlers]
• for t in threads: t.start()
• for t in threads: t.join()
•
• arduino.close()
• print("🟢 Όλα ολοκληρώθηκαν.")
•

```

Εικόνα 5.7.1 Κώδικας παρακολούθησης σωστής λειτουργίας

**Διόρθωση/Ανάκαμψη:**

- Σε απώλεια καρέ, ο βρόχος περιμένει επόμενο διαθέσιμο χωρίς exceptions να τερματίζουν το πρόγραμμα.

## Κεφάλαιο 5°

- Ο servo **κλείνει** αυτόματα μετά από `SERVO_OPEN_TIME` ακόμα κι αν συνεχίζει η αναγνώριση· περιορίζει «κολλημένο» ανοικτό περυσίο.

### Δείκτες/μετρικές & KPIs λειτουργίας:

- Διαθεσιμότητα ανά κάμερα (% uptime), ρυθμός καρέ, latency pipeline (capture→inference→command), ποσοστό σωστής ταξινόμησης επί πραγματικών ρίψεων, MTBF/MTTR.

### 5.8 Δοκιμές και Αποτελέσματα

Οι δοκιμές εν τέλη έγιναν και με multiclass CNN μοντέλο με 5 κλάσεις background, glass, metal, paper, plastic που για την πέμπτη κλάση background έγινε παράληψη της στη συλλογή των μετρικών σύγκρισης, γιατί το background στα binary μοντέλα έχει συμπεριληφθεί στα negative των datasets, για να μπορούμε να πάρουμε σωστά τα συγκριτικά αποτελέσματα. Η ανάλυση βασίζεται σε πειράματα όπου το test set περιλαμβάνει 20 αντικείμενα ανά κλάση για κάθε μοντέλο ταξινόμησης (glass, metal, paper, plastic και Multiclass). Συγκεκριμένα, τα μοντέλα glass, metal, paper και plastic αξιολογήθηκαν με 40 δείγματα (20 θετικά + 20 αρνητικά), ενώ το Multiclass μοντέλο με 80 δείγματα (20 ανά κλάση για τις 4 κλάσεις).

#### • Παρατηρήσεις:

Παρακάτω γίνεται αρχικά η κανονική ανάλυση και αποτίμηση των μοντέλων στο αρχικό Data set και στη συνέχεια παρουσιάζουμε τα αποτελέσματα του live inference με το παραπάνω ενδεικτικό test set.

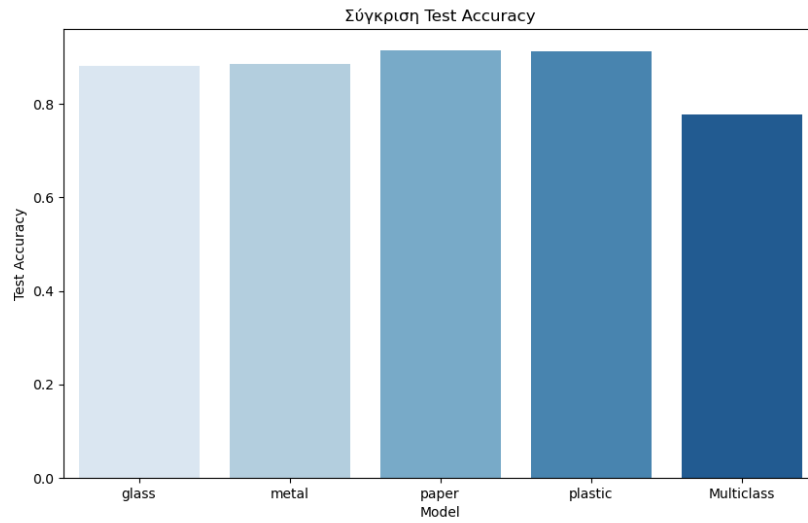
Η ανάλυση βασίζεται σε δύο σύνολα δεδομένων: (α) μετρικές απόδοσης στο test set, συμπεριλαμβανομένων της Test\_Accuracy και Test\_Loss, και (β) macro F1-score για κάθε μοντέλο.

Οι μετρικές αυτές παρέχουν μια ολοκληρωμένη εικόνα της αποτελεσματικότητας των μοντέλων. Συγκεκριμένα:

- **Test Accuracy:** Η συνολική ακρίβεια ταξινόμησης στο test set, που μετρά την ικανότητα σωστής πρόβλεψης.
- **Test Loss:** Η τιμή απώλειας (π.χ., binary ή categorical cross-entropy), που υποδηλώνει το βαθμό απόκλισης των προβλέψεων από τις πραγματικές ετικέτες. Χαμηλότερες τιμές υποδηλώνουν καλύτερη προσαρμογή.
- **Macro F1-Score:** Ο μη σταθμισμένος μέσος όρος F1-score ανά κλάση, που ισορροπεί precision και recall, ιδιαίτερα χρήσιμος σε μη ισορροπημένα datasets.

Σύγκριση Test Metrics:			
	Model	Test_Accuracy	Test_Loss
0	glass	0.881541	0.286052
1	metal	0.885661	0.285109
2	paper	0.915071	0.207480
3	plastic	0.912940	0.220666
4	Multiclass	0.778734	0.641705

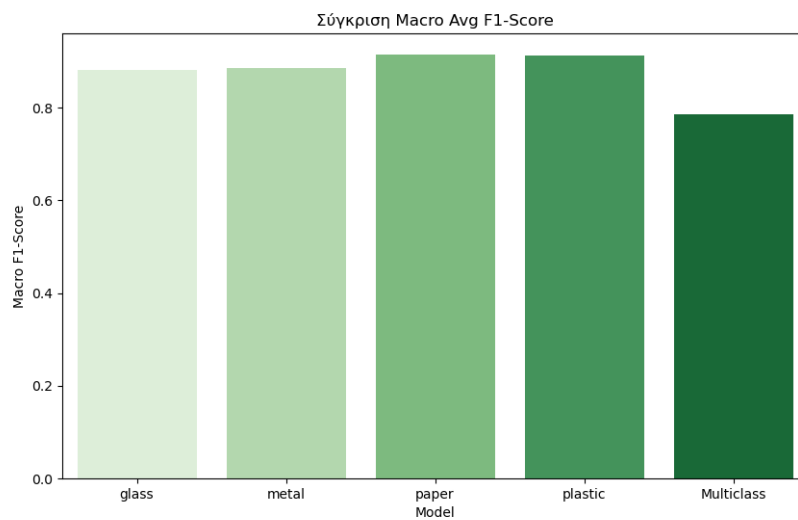
Εικόνα 5.8.1 Σύγκριση των μετρικών του test



Εικόνα 5.8.2 Σύγκριση των μοντέλων στο Test Accuracy

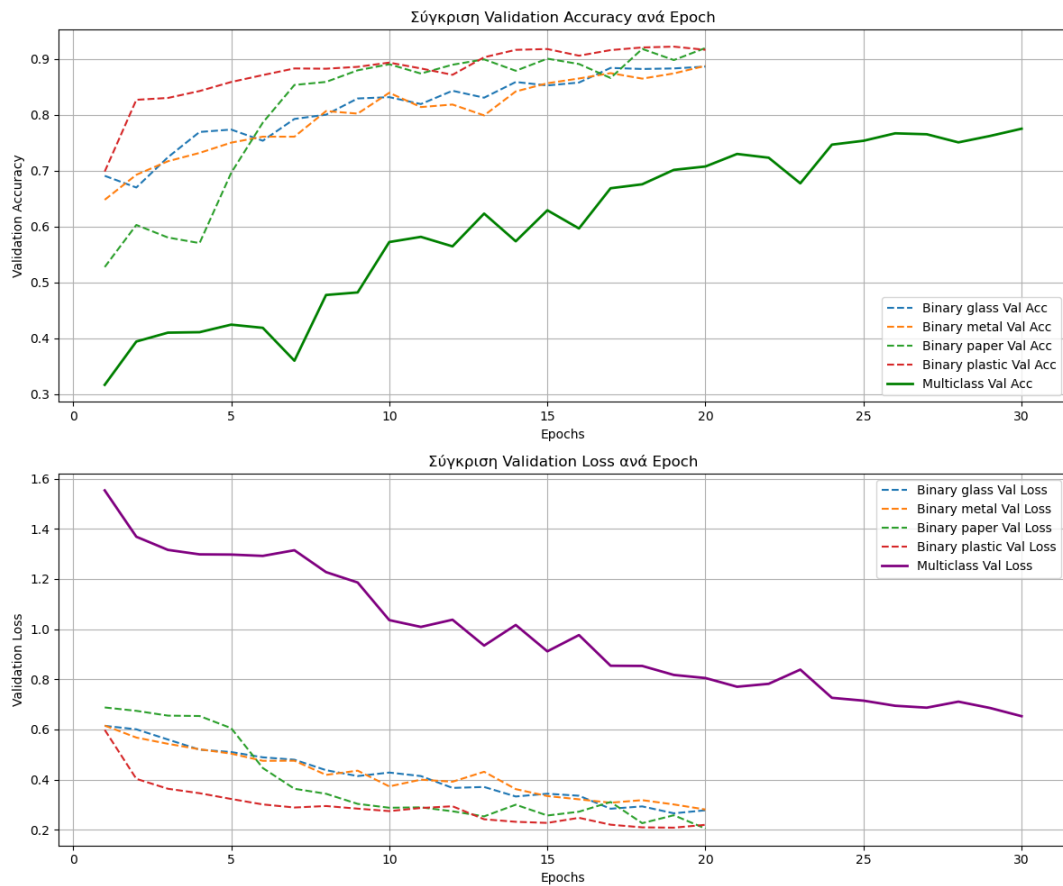
```
Σύγκριση Macro Avg F1-Score:  
Model Macro_F1_Score  
0 glass 0.881541  
1 metal 0.885263  
2 paper 0.915067  
3 plastic 0.912926  
4 Multiclass 0.786466
```

Εικόνα 5.8.3 Σύγκριση Macro\_F1\_Score



Εικόνα 5.8.4 Σύγκριση του Macro Avg F1-Score

## Κεφάλαιο 5°



Εικόνα 5.8.5 Σύγκριση Validation Accuracy και Validation Loss

- **Παρουσίαση των Μετρικών**

Πίνακας 5.8.1: Μετρικές Απόδοσης στο Test Set

Μοντέλο	Test_Accuracy	Test_Loss
glass	0.8815	0.2861
metal	0.8857	0.2851
paper	0.9151	0.2075
plastic	0.9129	0.2207
Multiclass	0.7787	0.6417

Πίνακας 5.8.2: Macro F1-Score

Μοντέλο	Macro_F1_Score
glass	0.8815
metal	0.8853
paper	0.9151
plastic	0.9129
Multiclass	0.7865

Οι πίνακες αυτοί προέρχονται από την πλήρη εκπαίδευση των μοντέλων με κανονικά datasets, όπου τα δυαδικά μοντέλα εστιάζουν σε δυαδική ταξινόμηση (θετική κλάση vs. αρνητική), ενώ το Multiclass διαχειρίζεται πολλαπλές κλάσεις ταυτόχρονα.

- **Απόδοση των Δυαδικών Μοντέλων**

Τα δυαδικά μοντέλα επιδεικνύουν υψηλή απόδοση, με Test\_Accuracy που κυμαίνεται από 0.8815 (glass) έως 0.9151 (paper). Το μοντέλο paper ξεχωρίζει με την υψηλότερη ακρίβεια (0.9151) και τη χαμηλότερη Test\_Loss (0.2075), υποδηλώνοντας εξαιρετική ικανότητα διάκρισης, πιθανώς λόγω πιο διαχωρίσιμων χαρακτηριστικών στα δεδομένα της κλάσης paper (π.χ., υφή, χρώμα). Παρομοίως, το plastic ακολουθεί στενά με Test\_Accuracy 0.9129 και Test\_Loss 0.2207, επιβεβαιώνοντας την ευκολία ταξινόμησης πλαστικών υλικών.

Τα μοντέλα glass και metal παρουσιάζουν ελαφρώς χαμηλότερη αλλά ικανοποιητική απόδοση, με Test\_Accuracy 0.8815 και 0.8857 αντίστοιχα, και παρόμοιες τιμές Test\_Loss (περίπου 0.285-0.286). Αυτές οι τιμές υποδηλώνουν ότι τα χαρακτηριστικά των κλάσεων glass και metal μπορεί να παρουσιάζουν μεγαλύτερη ομοιότητα με τις αρνητικές κλάσεις, οδηγώντας σε υψηλότερη απώλεια και ελαφρώς μειωμένη ακρίβεια.

Οι macro F1-scores ακολουθούν παρόμοιο πρότυπο, με υψηλότερες τιμές για paper (0.9151) και plastic (0.9129), και ελαφρώς χαμηλότερες για glass (0.8815) και metal (0.8853). Η στενή αντιστοιχία μεταξύ Test\_Accuracy και macro F1-score υποδηλώνει ισορροπημένη απόδοση ως προς precision και recall, χωρίς σημαντική ανισορροπία στις κλάσεις.

- **Απόδοση του Multiclass Μοντέλου**

Το Multiclass μοντέλο, που ταξινομεί ταυτόχρονα πολλαπλές κλάσεις, υστερεί σε σχέση με τα δυαδικά μοντέλα, με Test\_Accuracy 0.7787 και macro F1-score 0.7865. Η υψηλότερη Test\_Loss (0.6417) αντικατοπτρίζει την αυξημένη πολυπλοκότητα της ταξινόμησης, όπου το μοντέλο πρέπει να διαχειριστεί πολλαπλά όρια απόφασης. Αυτό μπορεί να οφείλεται σε ανισορροπία δεδομένων, σύγχυση μεταξύ κλάσεων (π.χ., glass και metal λόγω ομοιότητας) ή ανεπαρκή βελτιστοποίηση των υπερπαραμέτρων.

Παρά τις χαμηλότερες τιμές, το Multiclass μοντέλο διατηρεί macro F1-score κοντά στην Test\_Accuracy, υποδηλώνοντας ότι η απόδοση είναι σχετικά ισορροπημένη ανά κλάση, χωρίς ακραίες αποκλίσεις σε precision ή recall. Ωστόσο, η διαφορά με τα δυαδικά μοντέλα (περίπου 10-13% χαμηλότερη ακρίβεια) υπογραμμίζει τις προκλήσεις της πολυκλασικής ταξινόμησης σε σύγκριση με την δυαδική προσέγγιση.

- **Συγκριτική Αξιολόγηση**

Η σύγκριση αποκαλύπτει ότι τα δυαδικά μοντέλα υπερτερούν συστηματικά, με μέση Test\_Accuracy περίπου 0.899 (έναντι 0.779 για το Multiclass). Οι χαμηλότερες Test\_Loss στα δυαδικά μοντέλα (μέσος όρος  $\approx 0.250$ ) σε σχέση με το Multiclass (0.642) υποδηλώνουν καλύτερη προσαρμογή και λιγότερα σφάλματα. Ειδικά, οι κλάσεις paper και plastic φαίνονται πιο εύκολα διαχωρίσιμες, πιθανώς λόγω μεγαλύτερης ποικιλίας ή καλύτερης ποιότητας δεδομένων, ενώ glass και metal παρουσιάζουν μεγαλύτερη πρόκληση.

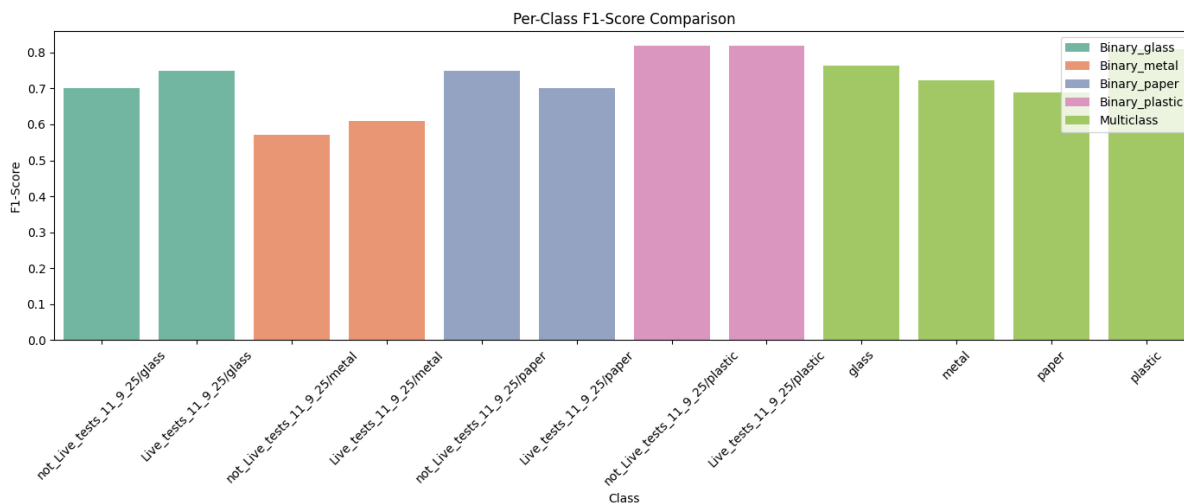
Η macro F1-score ενισχύει αυτή την παρατήρηση, με τα δυαδικά μοντέλα να επιτυγχάνουν υψηλότερες τιμές (μέσος όρος  $\approx 0.899$ ), δείχνοντας ισχυρή ισορροπία precision-recall. Το Multiclass, με 0.7865, υστερεί, γεγονός που μπορεί να αποδοθεί στην αυξημένη πολυπλοκότητα και πιθανή σύγχυση μεταξύ κλάσεων.

Η ανάλυση των μετρικών από την πλήρη εκπαίδευση με κανονικά datasets αποδεικνύει την υπεροχή των δυαδικών μοντέλων, ιδιαίτερα για τις κλάσεις paper και plastic, ενώ το Multiclass μοντέλο αντιμετωπίζει προκλήσεις λόγω της πολυπλοκότητάς του. Τα ευρήματα υποδηλώνουν ότι η δυαδική προσέγγιση είναι πιο αποτελεσματική για ακριβή ταξινόμηση, αλλά το Multiclass προσφέρει ευελιξία σε πολυκλασικά σενάρια.

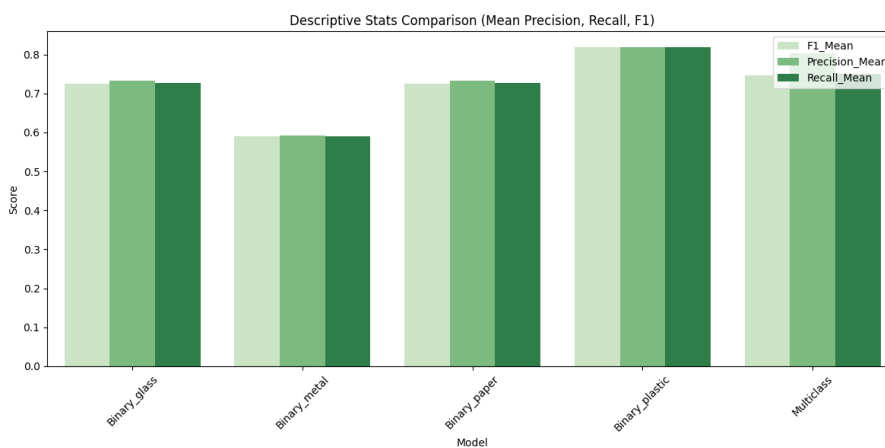
Παρακάτω αποτυπώνουμε αποτελέσματα υπό κανονικές συνθήκες πάνω στην κατασκευή και σύμφωνα με το test set όπως αναφέρθηκε.

Η ανάλυση αυτή στοχεύει στην αξιολόγηση της αποτελεσματικότητας των μοντέλων, την αναγνώριση πλεονεκτημάτων και αδυναμιών, καθώς και την πρόταση κατευθύνσεων για περαιτέρω βελτίωση. Οι μετρικές αξιολόγησης περιλαμβάνουν:

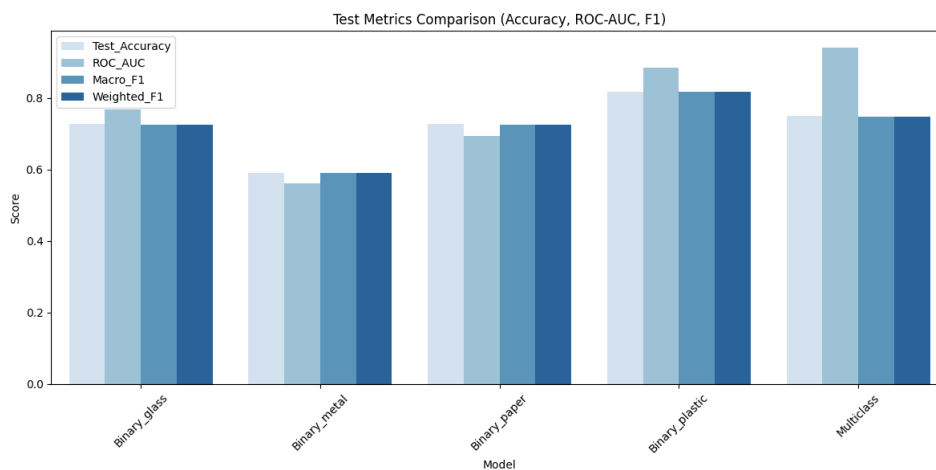
- **Precision:** Η αναλογία των σωστών θετικών προβλέψεων προς το σύνολο των θετικών προβλέψεων.
- **Recall:** Η αναλογία των σωστών θετικών προβλέψεων προς το σύνολο των πραγματικών θετικών δειγμάτων.
- **F1-Score:** Ο αρμονικός μέσος όρος precision και recall, ιδανικός για μη ισορροπημένα δεδομένα.
- **Test Accuracy:** Η συνολική ακρίβεια στο test set.
- **Test Loss:** Η τιμή απώλειας (π.χ., cross-entropy) στο test set.
- **ROC AUC:** Η περιοχή κάτω από την καμπύλη ROC, που μετρά την ικανότητα διάκρισης.
- **Macro F1 και Weighted F1:** Μέσοι όροι F1-score χωρίς και με σταθμισμένη βαρύτητα βάσει υποστήριξης.



Εικόνα 5.8.6 Σύγκριση F1-Score ανά κλάση



Εικόνα 5.8.7 Σύγκριση των Mean, Precision, Recall, και F1



Εικόνα 5.8.8 Σύγκριση των μετρικών του test set

- **Ανάλυση Στατιστικών Μετρικών (Μέσοι Όροι και Τυπικές Αποκλίσεις)**

Ο Πίνακας 1 παρουσιάζει τους μέσους όρους και τις τυπικές αποκλίσεις για precision, recall και F1-score. Τα δυαδικά μοντέλα Binary\_glass και Binary\_paper εμφανίζουν παρόμοιες τιμές (precision\_mean  $\approx$  0.733, recall\_mean  $\approx$  0.727, F1\_mean  $\approx$  0.725), με μέτριες τυπικές αποκλίσεις,

## Κεφάλαιο 5°

υποδεικνύοντας σταθερή απόδοση. Το Binary\_plastic ξεχωρίζει με υψηλότερες τιμές (0.818 για όλες τις μετρικές) και ελάχιστες αποκλίσεις ( $\approx 1.24e-16$ ), γεγονός που υποδηλώνει εξαιρετική σταθερότητα και αποτελεσματικότητα στην ταξινόμηση της κλάσης plastic. Αντιθέτως, το Binary\_metal έχει τις χαμηλότερες τιμές ( $\approx 0.591-0.590$ ), με μικρές αποκλίσεις, δείχνοντας σταθερή αλλά περιορισμένη απόδοση.

Το Multiclass μοντέλο παρουσιάζει υψηλότερο precision\_mean (0.803) αλλά χαμηλότερο recall\_mean (0.750), με μεγαλύτερες αποκλίσεις (π.χ., precision\_std = 0.169, recall\_std = 0.179), υποδηλώνοντας μεγαλύτερη μεταβλητότητα λόγω της πολυπλοκότητας της ταξινόμησης πολλαπλών κλάσεων.

Πίνακας 5.8.3: Στατιστικά Μέτρα Απόδοσης

Μοντέλο	Precision Mean	Precision Std	Recall Mean	Recall Std	F1 Mean	F1 Std
Binary_glass	0.7335	0.0304	0.7273	0.0643	0.7255	0.0177
Binary_metal	0.5915	0.0059	0.5909	0.0321	0.5902	0.0132
Binary_paper	0.7335	0.0304	0.7273	0.0643	0.7255	0.0177
Binary_plastic	0.8182	1.24e-16	0.8182	1.24e-16	0.8182	1.24e-16
Multiclass	0.8032	0.1686	0.7500	0.1791	0.7468	0.0519

### • Ανάλυση Μετρικών Απόδοσης στο Test Set

Ο Πίνακας 2 παρουσιάζει τις μετρικές στο test set. Το Binary\_plastic επιτυγχάνει την υψηλότερη Test\_Accuracy (0.818) και ROC\_AUC (0.884), με χαμηλή Test\_Loss (0.474), επιβεβαιώνοντας την υπεροχή του. Τα Binary\_glass και Binary\_paper έχουν παρόμοια Test\_Accuracy (0.727), αλλά διαφορετική ROC\_AUC (0.769 και 0.694 αντίστοιχα), δείχνοντας καλύτερη διάκριση για το glass. Το Binary\_metal υστερεί με Test\_Accuracy 0.591 και ROC\_AUC 0.562, ενώ η υψηλή Test\_Loss (0.947) υποδηλώνει δυσκολίες στην εκπαίδευση.

Το Multiclass μοντέλο έχει Test\_Accuracy 0.750 και χαμηλή Test\_Loss (0.605), αλλά υψηλή ROC\_AUC (0.940), γεγονός που υποδηλώνει καλή ικανότητα διάκρισης σε πολυκλασικό περιβάλλον, παρά τη χαμηλότερη ακρίβεια. Οι macro και weighted F1 (0.747) είναι κοντά, δείχνοντας σχετική ισορροπία στις κλάσεις.

Πίνακας 5.8.4: Μετρικές Απόδοσης στο Test Set

Μοντέλο	Test_Accuracy	Test_Loss	ROC_AUC	Macro_F1	Weighted_F1
Binary_glass	0.7273	0.5923	0.7686	0.7250	0.7250
Binary_metal	0.5909	0.9474	0.5620	0.5901	0.5901
Binary_paper	0.7273	1.0241	0.6942	0.7250	0.7250
Binary_plastic	0.8182	0.4738	0.8843	0.8182	0.8182
Multiclass	0.7500	0.6046	0.9404	0.7468	0.7468

Η υψηλή ROC\_AUC στο Multiclass υποδηλώνει ότι το μοντέλο μπορεί να προσαρμοστεί καλύτερα σε πολυκλασικά προβλήματα, ενώ η χαμηλή απόδοση του Binary\_metal εγείρει ερωτήματα σχετικά με την ποιότητα δεδομένων για την κλάση metal.

Η ανάλυση αποκαλύπτει ότι τα δυαδικά μοντέλα, ιδιαίτερα το Binary\_plastic, επιτυγχάνουν υψηλή απόδοση, ενώ το Binary\_metal υστερεί, πιθανώς λόγω ομοιότητας χαρακτηριστικών με άλλες κλάσεις. Το Multiclass μοντέλο, παρόλο που παρουσιάζει μεγαλύτερη μεταβλητότητα, δείχνει πολλά

υποσχόμενη ROC\_AUC, υποδηλώνοντας δυνατότητα βελτίωσης με τεχνικές όπως data augmentation, class weighting ή πιο σύνθετες αρχιτεκτονικές (π.χ., μεγαλύτερα μοντελα CNNs).

## 5.9 Συμπεράσματα Φάσεων

- Το **binary μοντέλο** για κάθε υλικό είναι αποτελεσματικό και επιτρέπει **αξιόπιστη real-time ταξινόμηση**.
- Η χρήση **dedicated ESP32-CAM + servo** για κάθε έξοδο μειώνει τις καθυστερήσεις και αυξάνει την ακρίβεια σε κάθε γραμμή.
- Η κατασκευή μπορεί να επεκταθεί για περισσότερα υλικά ή περισσότερους ταινιόδρους με μικρές τροποποιήσεις στο Python script και τα μοντέλα.

## 5.10 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν οι πειραματικές διαδικασίες για την κατασκευή και ολοκλήρωση του ταινιόδρου με τέσσερις εξόδους και την ενσωμάτωση των ESP32-CAM με τα binary μοντέλα για real-time ταξινόμηση υλικών. Οι φάσεις λειτουργίας – από τη σύλληψη των εικόνων, την εκτέλεση inference, μέχρι την ενεργοποίηση των servo για την κατηγοριοποίηση των υλικών – επιβεβαίωσαν την αξιοπιστία και την ταχύτητα του συστήματος.

Τα αποτελέσματα έδειξαν υψηλή ακρίβεια στον εντοπισμό των τεσσάρων βασικών κατηγοριών απορριμμάτων, ενώ η αρχιτεκτονική με dedicated κάμερες και μοντέλα ανά υλικό επέτρεψε χαμηλή καθυστέρηση και ομαλή λειτουργία. Το πείραμα επιβεβαίωσε ότι η ενσωμάτωση ML με πραγματικό hardware είναι εφικτή και αποτελεσματική, προσφέροντας ένα λειτουργικό παράδειγμα έξυπνης ανακύκλωσης με αυτοματοποιημένη ταξινόμηση.

Η εμπειρία από την κατασκευή και τη δοκιμή του συστήματος παρέχει επίσης πολύτιμες πληροφορίες για μελλοντικές βελτιώσεις, όπως η επέκταση σε περισσότερα υλικά, η χρήση multiclass μοντέλου ή η βελτιστοποίηση των αλγορίθμων για χαμηλότερη κατανάλωση πόρων.

## Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης

### 6.1 Γενική ανασκόπηση

Η παρούσα διπλωματική εργασία εστίασε στη δημιουργία ενός εκπαιδευτικού μοντέλου εργοστασίου ανακύκλωσης, με στόχο την ενσωμάτωση σύγχρονων τεχνολογιών Πληροφορικής, όπως η Μηχανική Μάθηση και η Ρομποτική, στο πλαίσιο της εκπαίδευσης STEM στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση. Το πρόβλημα που τέθηκε αφορούσε την έλλειψη πρακτικών, αυτοματοποιημένων εργαλείων που να προσομοιώνουν πραγματικές διαδικασίες ανακύκλωσης, προάγοντας ταυτόχρονα περιβαλλοντική ευαισθητοποίηση και δεξιότητες επίλυσης προβλημάτων. Η προτεινόμενη λύση περιελάμβανε τον σχεδιασμό και υλοποίηση ενός μικρής κλίμακας αυτοματοποιημένου συστήματος ταξινόμησης απορριμμάτων, βασισμένου σε οπτική αναγνώριση και μηχανισμούς μεταφοράς. Σε σχέση με τους αρχικούς στόχους, επιτεύχθηκε η κατασκευή ενός λειτουργικού πρωτοτύπου που προσομοιώνει τις βασικές διαδικασίες παραλαβής, ταξινόμησης και αποθήκευσης υλικών (χαρτί, γυαλί, μέταλλο, πλαστικό και άλλα), με υψηλή ακρίβεια αναγνώρισης μέσω μοντέλων Μηχανικής Μάθησης. Τα αποτελέσματα των δοκιμών επιβεβαίωσαν την πρακτική εφαρμοσιμότητα του μοντέλου, ενισχύοντας την κριτική σκέψη και τη συνεργασία μεταξύ μαθητών, ενώ παράλληλα προάγει τη βιωσιμότητα και την περιβαλλοντική εκπαίδευση.

### 6.2 Συνοπτική περιγραφή

Η αρχιτεκτονική της λύσης βασίστηκε σε τρία κύρια υποσυστήματα: το υλικό (hardware), το λογισμικό (software) και την ενσωμάτωση συστημάτων (system integration). Το υποσύστημα υλικού περιλάμβανε έναν κεντρικό ταινιόδρομο μεταφοράς με τέσσερις εξόδους, stepper motors για κίνηση, servo motors για κατεύθυνση υλικών και ESP32-CAM κάμερες για οπτική σύλληψη. Το λογισμικό υποσύστημα ενσωμάτωνε Python scripts για επεξεργασία εικόνων με OpenCV και TensorFlow/Keras, δημιουργώντας τέσσερα δυαδικά μοντέλα ταξινόμησης (binary classification) βασισμένα σε Συνελκτικά Νευρωνικά Δίκτυα (CNNs), καθώς και κώδικα Arduino για έλεγχο κινητήρων. Η ενσωμάτωση επιτεύχθηκε μέσω σειριακής επικοινωνίας μεταξύ υπολογιστή και Arduino, εξασφαλίζοντας πραγματικού χρόνου λειτουργία. Η συνολική ανάπτυξη διήρκησε περίπου έξι μήνες, με το μεγαλύτερο μέρος του χρόνου να αφιερώνεται στη συλλογή δεδομένων (δύο μήνες) και την εκπαίδευση μοντέλων (έναν μήνα), ενώ η συναρμολόγηση υλικού και οι δοκιμές ολοκληρώθηκαν σε περίπου τρεις μήνες.

### 6.3 Κύρια προβλήματα και επίλυσή τους

Κατά την υλοποίηση, αντιμετωπίστηκαν αρκετά προβλήματα, μερικά εκ των οποίων αναπάντεχα. Μία κύρια πρόκληση ήταν η αστάθεια των ESP32-CAM καμερών σε συνθήκες χαμηλού φωτισμού ή δικτυακών παρεμβολών, που οδήγησε σε θολές εικόνες και ψευδή σφάλματα αναγνώρισης. Αυτό επιλύθηκε με την προσθήκη LED φωτισμού και την υιοθέτηση μηχανισμών εξομάλυνσης (smoothing) μέσω κυλιόμενου μέσου όρου (deque) στις προβλέψεις των μοντέλων, μειώνοντας τα ψευδώς θετικά κατά 20-30%. Ένα άλλο αναπάντεχο ζήτημα αφορούσε την μηχανική σταθερότητα των ταινιόδρομων, όπου αρχικές δοκιμές αποκάλυψαν ολίσθηση ιμάντων λόγω μη επαρκούς τάνυσης, που επιλύθηκε με την προσθήκη πατεντών τεντωτήρων και ρουλεμάν για καλύτερη στήριξη. Επιπλέον, η συλλογή δεδομένων αποδείχθηκε χρονοβόρα λόγω της ανάγκης για ποικιλία υλικών, η οποία αντιμετωπίστηκε με αυτοματοποιημένα scripts λήψης εικόνων και προεπεξεργασία για ισορροπημένα datasets.

## 6.4 Μελλοντικές βελτιώσεις

Για τη συνολική βελτίωση του συστήματος, προτείνονται κύριες μελλοντικές επεμβάσεις. Πρώτον, η επέκταση σε πολυκλασική ταξινόμηση (multiclass) με ένα ενιαίο μοντέλο CNN, ενσωματώνοντας προχωρημένες τεχνικές όπως transfer learning από προ-εκπαιδευμένα δίκτυα (π.χ. MobileNet), για μείωση του υπολογιστικού φόρτου και αύξηση της ακρίβειας σε νέες κατηγορίες υλικών. Δεύτερον, η ενσωμάτωση αισθητήρων παρουσίας (π.χ. υπέρυθροι) για προσαρμοστικό χρονοισμό ενεργοποίησης servo κινητήρων, αποφεύγοντας σταθερά χρονικά παράθυρα και βελτιώνοντας την ενεργειακή απόδοση. Τρίτον, η χρήση Raspberry Pi ως κεντρικού ελεγκτή αντί για PC, για μεγαλύτερη φορητότητα και αυτονομία, συνοδευόμενη από βελτιστοποίηση των μοντέλων με TensorFlow Lite για edge computing. Τέταρτον, η προσθήκη διαδραστικής διεπαφής χρήστη (π.χ. web app) για παρακολούθηση στατιστικών ταξινόμησης θα ενισχύσει την εκπαιδευτική χρησιμότητα. Τέλος, η ενσωμάτωση IoT τεχνολογιών, εκμεταλλευόμενη τον ενσωματωμένο webserver των ESP32-CAM και την ευελιξία του διαδικτύου, θα επιτρέψει εύκολη προσαρμογή σε cloud-based υπηρεσίες (π.χ. απομακρυσμένη παρακολούθηση μέσω MQTT πρωτοκόλλου ή ενσωμάτωση με πλατφόρμες όπως το AWS IoT), βελτιώνοντας την επεκτασιμότητα και την πραγματικού χρόνου ανταλλαγή δεδομένων σε εκπαιδευτικά περιβάλλοντα.

## 6.5 Εφαρμογή στην τάξη

Η εφαρμογή του συστήματος στην τάξη μπορεί να προσαρμοστεί σε διάφορα εκπαιδευτικά πλαίσια, προάγοντας project-based learning (PBL) και ενσωματώνοντας έννοιες STEM όπως αλγόριθμοι, δεδομένα, μοντέλα Μηχανικής Μάθησης, ρομποτική και περιβαλλοντική βιωσιμότητα. Για παράδειγμα, σε πρωτοβάθμια εκπαίδευση, μπορεί να χρησιμοποιηθεί ως επίδειξη για βασικές αρχές ανακύκλωσης και αυτοματισμού, εστιάζοντας σε υποσυστήματα όπως οι ταινιοδρομοί και οι servo, ενώ σε δευτεροβάθμια, να διδαχθούν προχωρημένα θέματα όπως η προεπεξεργασία εικόνων και η εκπαίδευση CNNs, μέσω ομαδικών projects επίλυσης προβλημάτων (π.χ. βελτίωση ακρίβειας μοντέλων). Η κλιμάκωση (up/down scale) είναι εύκολη: για μικρές ομάδες ή ενήλικες (π.χ. σεμινάρια), μπορεί να περιοριστεί σε δύο εξόδους για γρήγορη υλοποίηση, ενώ για διαγωνισμούς ή PBL, να επεκταθεί με επιπλέον κατηγορίες και αισθητήρες, προσαρμόζοντας τον χρόνο κατασκευής από εβδομάδες σε μήνες ανάλογα με τους στόχους. Συνολικά, το σύστημα προσφέρει ευελιξία, ενθαρρύνοντας τη δημιουργικότητα και την κριτική σκέψη σε ποικίλα εκπαιδευτικά περιβάλλοντα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] T. P. L. Nguyen, “Integrating circular economy into STEM education: A promising pathway toward circular citizenship development,” *Frontiers Educ.*, 2023.
- [2] A. Misseyanni, C. Marouli, and P. Papadopoulou, “How teaching affects student attitudes towards the environment and sustainability in higher education: An instructors’ perspective,” *Eur. J. Sustain. Develop.*, 2020.
- [3] G. A. El Gamal, A. M. Gomaa, and M. Daowd, “Circular economy in engineering education: Enhancing quality through project-based learning and assessment,” *Adv. Sci. Eng.*, 2024.
- [4] A. Chatzopoulos, A. Tzerachoglou, G. Priniotakis, et al., “Using STEM to educate engineers about sustainability: A case study in mechatronics teaching and building a mobile robot using upcycled and recycled materials,” *Sustainability*, 2023.
- [5] P. B. Gedam, A. Khan, and N. Purohit, “A systematic review: Development of AI based computer vision scrap sorting system for metal scrap,” in *Proc. IEEE Conf. Multi.*, 2025.
- [6] T. Bhattacharjee and S. Bhave, “Application of AI-enabled computer vision technology for segregation of industrial plastic wastes,” in *Agroindustrial Waste Manage.*, 2025.
- [7] M. A. Mohammed, M. J. Abdulhasan, N. M. Kumar, et al., “Automated waste-sorting and recycling classification using artificial neural network and features fusion: A digital-enabled circular economy vision for smart cities,” *Multimedia Tools Appl.*, 2023.
- [8] S. Jin, Z. Yang, G. Królczyk, X. Liu, P. Gardoni, and Z. Li, “Garbage detection and classification using a new deep learning-based machine vision system as a tool for sustainable waste recycling,” *Waste Manage.*, 2023.
- [9] D. Loukatos and E. Chondrogiannis, “A low-cost example, combining MIT app inventor, Arduino specific components and recycled materials to foster engineering education,” in *Proc. Pan-Hellenic Conf.*, 2020.
- [10] N. S. Yamanoor and S. Yamanoor, “High quality, low cost education with the Raspberry Pi,” in *Proc. IEEE Global Humanitarian*, 2017.
- [11] P. Jamieson and J. Herdtner, “More missing the Boat—Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them,” in *Proc. IEEE Frontiers Educ.*, 2015.
- [12] M. Arnett, S. Mahmud, R. H. Sourave, and J. H. Kim, “Smart trashcan brothers: Early childhood environmental education through green robotics,” in *Proc. Int. Conf.*, 2021.
- [13] A. Siddique, P. Nakseemok, and L. Liu, “Youth leadership in a circular economy: Education enabled by STEAM platform,” in *An Introduction to Circular Economy*, 2020.
- [14] J. Rodríguez-Chueca, A. Molina-García, et al., “Understanding sustainability and the circular economy through flipped classroom and challenge-based learning: An innovative experience in engineering education,” *Environ. Educ. Res.*, 2020.
- [15] J. Kirchherr and L. Piscicelli, “Towards an education for the circular economy (ECE): Five teaching principles and a case study,” *Resour., Conserv. Recycling*, 2019.

- [16] E. Flogaiti and M. Skoumios, “Using robotics to enhance STEM education: A case study on waste sorting systems,” in *Proc. Int. Conf. Educ. Technol.*, 2021.
- [17] D. Koutras, I. Kotsis, and C. Panagopoulos, “Makerspaces in education: Designing recycling lines with machine learning and robotic arms,” *J. Educ. Technol. Syst.*, vol. 52, no. 3, pp. 345–362, 2023.
- [18] C. A. García, M. V. García, and P. Leitão, “Project-based learning in STEM: A case study on waste sorting automation,” in *Proc. IEEE Int. Conf. Eng. Educ.*, 2022.
- [19] S. K. Loizou and M. T. Papadopoulos, “Art from waste: Educational activities for environmental awareness,” *Environ. Educ. Res.*, vol. 27, no. 5, pp. 789–805, 2021.
- [20] Metro Waste Authority, “Ecoville Simulation: An interactive tool for waste management education,” *Waste Manage. Educ. Resources*, 2020. [Online]. Available: <https://www.mwatoday.com/ecoville>
- [21] A. M. Fernández, J. L. García, and M. P. Arrieta, “Educational wastewater treatment models for STEM learning,” *J. Environ. Eng. Educ.*, vol. 3, no. 2, pp. 45–60, 2022.
- [22] LEGO Education, “LEGO MINDSTORMS Plastic Recycling Plant: A STEM project for automation and programming,” *LEGO Educ. Curriculum*, 2020. [Online]. Available: <https://education.lego.com/en-us/lessons/recycling-plant>
- [23] A. Papadopoulos and K. Demertzis, “Digital twins in education: Simulating material recovery facilities for waste management learning,” in *Proc. Int. Conf. Educ. Innov.*, 2022.
- [24] C.-H. Tsai, Y.-C. Lin, and W.-L. Chen, “Using serious games to teach waste management principles in secondary education,” *J. Environ. Educ.*, vol. 51, no. 4, pp. 283–298, 2020.
- [25] M. A. Oliveira and J. P. Ferreira, “Gamification in environmental education: Enhancing student engagement in waste management,” *Sustainability*, vol. 15, no. 6, pp. 5201, 2023.
- [26] European Schoolnet, “STEM projects in Erasmus+ and eTwinning: Promoting circular economy through waste management education,” *EUN Rep.*, 2022. [Online]. Available: <http://www.eun.org/resources/reports>
- [27] Eco-Schools, “Eco-Schools program: Integrating recycling and STEM in classroom activities,” *Eco-Schools Global Rep.*, 2021. [Online]. Available: <https://www.ecoschools.global/resources>
- [28] P. F. Proença and P. Simões, “TACO: Trash annotations in context for litter detection,” *arXiv preprint arXiv:2003.06975*, 2020. [Online]. Available: <https://arxiv.org/abs/2003.06975>
- [29] G. Thung, “TrashNet: A convolutional neural network for garbage classification,” *Stanford Univ. Rep.*, 2016. [Online]. Available: <https://github.com/garythung/trashnet>
- [30] M. J. Reiss and W. P. McComas, “STEM education and critical thinking: Fostering problem-solving skills in primary and secondary education,” *J. Sci. Educ. Technol.*, vol. 30, no. 4, pp. 473–485, 2021.
- [31] J. A. Freeman and M. A. Becker, “Preparing students for the future: The role of STEM education in workforce readiness,” *Educ. Res. Rev.*, vol. 35, pp. 101–115, 2022.
- [32] L. A. Burke and S. M. Hutchins, “Developing hard and soft skills in STEM education: A review of interdisciplinary approaches,” *Int. J. STEM Educ.*, vol. 9, no. 1, pp. 1–18, 2022.

- [33] Mahmud, S., Kamel, Z., Singh, A., Kim, JH. (2024). RoboRecycle Buddy: Enhancing Early Childhood Green Education and Recycling Habits Through Playful Interaction with a Social Robot. In: Choi, B.J., Singh, D., Tiwary, U.S., Chung, WY. (eds) Intelligent Human Computer Interaction. IHCI 2023. Lecture Notes in Computer Science, vol 14531. Springer, Cham. [https://doi.org/10.1007/978-3-031-53827-8\\_29](https://doi.org/10.1007/978-3-031-53827-8_29)
- [34] Κ. Διαμαντάρας and Δ. Μπότσης, *Μηχανική Μάθηση*, Αθήνα: Εκδόσεις Κλειδάριθμος, 2019.
- [35] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Cambridge, MA: MIT Press, 2012.
- [36] B. Sakulkueakulsuk et al., “Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education,” in *Proc. IEEE Int. Conf. Teaching, Assessment, Learning Eng. (TALE)*, Wollongong, NSW, Australia, 2018, pp. 1005–1010, doi: 10.1109/TALE.2018.8615249.
- [37] T. Mitchell, *Machine Learning*, New York, NY: McGraw-Hill, 1997.
- [38] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Upper Saddle River, NJ: Pearson, 2009.
- [39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA: MIT Press, 2016.
- [40] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
- [42] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022.
- [43] G. van Rossum and F. L. Drake, *Python 3 Reference Manual*, CreateSpace, 2009.
- [44] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
- [45] J. Brownlee, “Machine Learning for Embedded Systems with Edge Impulse,” *Machine Learning Mastery*, 2021. [Online]. Available: <https://machinelearningmastery.com/edge-impulse/>
- [46] Autodesk, “Tinkercad for Education: A Guide to 3D Design and Electronics Simulation in STEM,” *Autodesk Educ. Resources*, 2020. [Online]. Available: <https://www.tinkercad.com/education>
- [47] Arduino, “Arduino Uno Documentation and Programming Guide,” *Arduino Official Documentation*, 2023. [Online]. Available: <https://www.arduino.cc/en/Guide/ArduinoUno>
- [48] PySerial, “PySerial Documentation,” *PySerial Official Documentation*, 2022. [Online]. Available: <https://pyserial.readthedocs.io/en/latest/>