

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Δημιουργία εφαρμογής κινητού τηλεφώνου με  
Artificial Intelligence – Object Detection και απτική  
αλληλεπίδραση ως βοηθός σε άτομα με  
προβλήματα όρασης»



**Φοιτητής:**  
**Δήμακος Ιωάννης**  
**Αριθμός Μητρώου: 164656**

**Επιβλέπων:**  
**Κοκκώνης Γεώργιος**  
**Επίκουρος Καθηγητής**

Δημιουργία εφαρμογής κινητού τηλεφώνου με Artificial Intelligence – Object Detection και απτική αλληλεπίδραση ως βοηθός σε άτομα με προβλήματα όρασης

Κωδικός: 24109

Όνοματεπώνυμο φοιτητή: Δήμακρος Ιωάννης

Όνοματεπώνυμο εισηγητή: Κοκκώνης Γεώργιος

Ημερομηνία ανάληψης Π.Ε.: 21-01-2024

Ημερομηνία περάτωσης Π.Ε: 26-05-2024

*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 πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

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

## Πρόλογος

Σε μια εποχή ραγδαίων αλλαγών, η τεχνητή νοημοσύνη αναδύεται ως μια επαναστατική δύναμη, διαμορφώνοντας τον κόσμο γύρω μας. Από την αυτοματοποίηση καθημερινών εργασιών έως την ανάπτυξη πρωτοποριακών εφαρμογών, η ΤΝ φέρνει στο προσκήνιο νέες δυνατότητες και προκλήσεις, αγγίζοντας κάθε πτυχή της ανθρώπινης δραστηριότητας. Η παρούσα πτυχιακή εργασία αγγίζει ένα επίκαιρο και ηθικά σημαντικό ζήτημα: τον ρόλο της τεχνητής νοημοσύνης στην ενίσχυση της ανεξαρτησίας και της ποιότητας ζωής των ατόμων με προβλήματα όρασης. Η ραγδαία εξέλιξη της ΤΝ φέρνει στο προσκήνιο νέες δυνατότητες για την αντιμετώπιση των προκλήσεων που αντιμετωπίζουν τα άτομα αυτά στην καθημερινότητά τους. Σε έναν κόσμο ο οποίος ψηφιοποιείται ολοένα και με γρηγορότερους ρυθμούς, η πρόσβαση στην πληροφορία και η αλληλεπίδραση με την τεχνολογία αποτελούν ζωτικής σημασίας ζητήματα για άτομα με προβλήματα όρασης. Η όραση, αν και απαραίτητη, δεν είναι η μόνη αίσθηση που μας επιτρέπει να εξερευνήσουμε τον κόσμο. Η αφή, με την πλούσια πληροφορία που προσφέρει, μπορεί να αποτελέσει ένα ισχυρό εργαλείο ώστε να ξεπεραστούν οι οπτικές δυσκολίες που μπορεί να αντιμετωπίζουν κάποιοι συνάνθρωποι μας.

## Περίληψη

Σκοπός αυτής της πτυχιακής εργασίας είναι η διερεύνηση του πώς η ΤΝ μπορεί να αξιοποιηθεί για την ανάπτυξη καινοτόμων λύσεων που θα συμβάλλουν στην αυτονομία και την κοινωνική ένταξη των ατόμων με προβλήματα όρασης. Η εργασία εστιάζει στην ανάπτυξη μιας εφαρμογής ΤΝ, η οποία ανιχνεύει αντικείμενα, αναγνωρίζοντας το σημαντικό αντίκτυπο που δύναται να έχει στην καθημερινή ζωή του εκάστοτε ατόμου. Πέρα από την καινοτομία της ΤΝ, η εφαρμογή αξιοποιεί μία ακόμα τεχνολογία, όχι τόσο διαδεδομένη στην χώρα μας, την απτική αλληλεπίδραση. Μία ακόμα υποβοηθητική λειτουργία για τα άτομα τα οποία αντιμετωπίζουν προβλήματα όρασης είναι η φωνητική περιγραφή των ανιχνευμένων αντικειμένων. Η εφαρμογή αναπτύχθηκε για συσκευές που χρησιμοποιούν το λειτουργικό σύστημα android. Έπειτα έγινε αξιολόγηση της εφαρμογής και προέκυψαν κάποια συμπεράσματα για την καινοτομία της ΤΝ και άλλων υποβοηθητικών τεχνολογιών.

« Development of a Mobile Application using Artificial Intelligence –  
Object Detection and Haptic Feedback as an Assistive Tool for  
Visually Impaired Individuals. »

Dimkaros Ioannis

**Abstract**

This thesis investigates how Artificial Intelligence can be harnessed to create solutions that empower visually impaired individuals and foster their social inclusion. The core of this thesis is the development of an AI application that detects objects, acknowledging the transformative impact it can have on their daily lives. Beyond the innovation of AI, the application incorporates haptic feedback, a technology not yet widely spread. To further assist visually impaired users, the application provides voice descriptions of detected objects. Designed for Android devices, the application was subsequently evaluated, leading to valuable insights regarding the innovation of AI and other assistive technologies like tactile feedback.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω πρώτα από όλα τον Θεό, έπειτα τον Κύριο Κοκκώνη Γεώργιο για την ανάθεση της παρούσης εργασίας και την άψογη συνεργασία μας και τέλος θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου και την κοπέλα μου που με στηρίζουν τόσα χρόνια.

# Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract .....	v
Ευχαριστίες .....	vi
Περιεχόμενα .....	vii
Κατάλογος Σχημάτων .....	x
Κατάλογος Πινάκων.....	x
Κατάλογος Εικόνων .....	x
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή και Ιστορική Αναδρομή .....	1
1.1 Εισαγωγή .....	1
1.2 Η Εργασία και η Δομή της.....	2
1.3 Επίλογος.....	2
Κεφάλαιο 2ο: Τεχνητή Νοημοσύνη.....	3
2.1 Εισαγωγή .....	3
2.2 Ορισμοί της TN.....	3
2.3 Μηχανική Μάθηση και Βαθιά Μάθηση .....	3
2.3.1 Εποπτευόμενη Μηχανική Μάθηση .....	5
2.3.2 Μη-εποπτευόμενη Μηχανική Μάθηση .....	5
2.3.3 Ενισχυτική Μάθηση .....	5
2.3.4 Ανάπτυξη Μοντέλων.....	5
2.4 Όραση Υπολογιστών .....	7
2.4.1 MobileNet.....	8
2.5 Επίλογος.....	13
Κεφάλαιο 3ο: Απτική Αλληλεπίδραση .....	14
3.1 Εισαγωγή .....	14
3.2 Απτική Τεχνολογία .....	14
3.3 Απτική Ανατροφοδότηση .....	16
3.3.1 Δερματική Ανατροφοδότηση .....	17
3.4 Τομείς Απτικής Αλληλεπίδρασης.....	17
3.5 Επίλογος.....	19
Κεφάλαιο 4ο: Υποβοηθητική Τεχνολογία .....	20

4.1	Εισαγωγή .....	20
4.2	Όραση μέσω της αφής .....	20
4.2.1	Αξιοποίηση Κινητών Συσκευών.....	20
4.3	Πρωτότυπα και εφαρμογές του μαθήματος Απτικές Διεπαφές .....	21
4.3.1	Αναπαράσταση Ψηφιακών Επιφανειών και Αντικειμένων:.....	21
4.3.2	Μοτίβα Δόνησης και Χρώματα:.....	21
4.3.3	Ανιχνευτής Εμποδίων:.....	22
4.3.4	Δόνηση ως μέσο προσανατολισμού .....	22
4.4	Επίλογος.....	23
Κεφάλαιο 5ο: Παρεμφερείς Εφαρμογές .....		24
5.1	Εισαγωγή .....	24
5.2	Be My Eyes.....	24
5.3	Seeing AI .....	25
5.4	BlindNavi.....	26
5.5	FoodTracker.....	26
5.6	Επίλογος.....	27
Κεφάλαιο 6ο: Εφαρμογή.....		28
6.1	Εισαγωγή .....	28
6.2	Έναρξη της εφαρμογής .....	28
6.2.1	Περιγραφή της Διεπαφής Χρήστη.....	30
6.3	Περιγραφή του κώδικα .....	33
6.3.1	Περιγραφή Κλάσης “CameraFragment” .....	33
6.3.2	Περιγραφή Κλάσης “PermissionFragment” .....	42
6.3.3	Περιγραφή Κλάσης “coco_translations” .....	45
6.3.4	Περιγραφή Κλάσης “MainActivity”.....	48
6.3.5	Περιγραφή Κλάσης “ObjectDetectorHelper” .....	48
6.3.6	Περιγραφή Κλάσης “OverlayView”.....	54
6.3.7	Περιγραφή Script “activity_main.xml” .....	63
6.3.8	Περιγραφή Script “fragment_camera.xml” .....	64
6.3.9	Περιγραφή Script “info_bottom_sheet.xml” .....	64
6.4	Επίλογος.....	70
Κεφάλαιο 7ο: Αξιολόγηση.....		71
7.1	Εισαγωγή .....	71
7.2	Ερωτηματολόγιο .....	71
7.2.1	Δημογραφικά Στοιχεία .....	71

7.2.2	Χρησιμότητα .....	71
7.2.3	Ευκολία Χρήσης.....	72
7.2.4	Ευκολία Μάθησης.....	73
7.2.5	Ικανοποίηση .....	73
7.2.6	Μελλοντική Βελτίωση.....	74
7.3	Στατιστικά Στοιχεία .....	74
7.3.1	Δημογραφικά Στοιχεία .....	74
7.3.2	Αραχνοειδές Διάγραμμα.....	75
7.3.3	Συγκεντρωτικός Πίνακας Αξιολόγησης Ερωτηματολογίου .....	76
7.3.4	Μελλοντική Βελτίωση.....	77
7.4	Έρευνα χρησιμότητας της Απτικής Αλληλεπίδρασης .....	77
7.5	Επίλογος.....	78
Κεφάλαιο 8ο:	Συμπεράσματα .....	79
8.1	Συμπεράσματα .....	79
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		80

## Κατάλογος Σχημάτων

Σχήμα 1.1: Σημαντικά ιστορικά επιτεύγματα στον κόσμο των υπολογιστών [1]. .....	1
Σχήμα 1.2: Χρονολογική εξέλιξη της τεχνητής νοημοσύνης [1]. .....	1
Σχήμα 2.1: Επίπεδα μηχανικής μάθησης [4]. .....	4
Σχήμα 2.2: Δομή μοντέλου [4]. .....	6

## Κατάλογος Πινάκων

Πίνακας 3.1: Ορολογίες απτικής αλληλεπίδρασης [9]. .....	15
Πίνακας 7.1: Συγκεντρωτικός πίνακας απαντήσεων ερωτηματολογίου .....	76

## Κατάλογος Εικόνων

Εικόνα 2.1: Εντοπισμός φρούτων και πλαστικού μπουκαλιού .....	10
Εικόνα 2.2: Εντοπισμός τηλεχειριστηρίου τηλεόρασης.....	10
Εικόνα 2.3: Εντοπισμός καρέκλας .....	11
Εικόνα 2.4: Εντοπισμός ψυγείου.....	11
Εικόνα 2.5: Εντοπισμός τηλεόρασης .....	12
Εικόνα 2.6: Εντοπισμός πιρουνιού και κουταλιού .....	12
Εικόνα 6.1: Εκκίνηση της εφαρμογής.....	28
Εικόνα 6.2: Διεπαφή αδειών .....	29
Εικόνα 6.3: Προεπισκόπηση κάμερας ενώ κάνει συνεχή ροή λήψης .....	30
Εικόνα 6.4: Ταμπέλα με τον χρόνο απόκρισης .....	30
Εικόνα 6.5: Επιλογή ευαισθησίας .....	30
Εικόνα 6.6: Επιλογή αριθμού ανιχνεύσιμων αντικειμένων.....	31
Εικόνα 6.7: Επιλογή αριθμού νημάτων.....	31
Εικόνα 6.8: Επιλογή Delegate .....	31
Εικόνα 6.9: Επιλογή μοντέλου ανίχνευσης αντικειμένων.....	31
Εικόνα 6.10: Διεπαφή ρυθμίσεων .....	32
Εικόνα 6.11: Imports .....	33
Εικόνα 6.12: Μεταβλητές.....	34
Εικόνα 6.13: Μέθοδος onResume .....	34
Εικόνα 6.14: Μέθοδος onDestroyView.....	35
Εικόνα 6.15: Μέθοδος onCreateView .....	35
Εικόνα 6.16: Μέθοδος onViewCreated.....	36
Εικόνα 6.17: Μέθοδος initBottomSheetControls .....	36
Εικόνα 6.18: Μέθοδος initBottomSheetControls .....	37
Εικόνα 6.19: Μέθοδος initBottomSheetControls .....	37
Εικόνα 6.20: Μέθοδος updateControlsUi .....	38
Εικόνα 6.21: Μέθοδος setUpCamera .....	38
Εικόνα 6.22: Μέθοδος bindCameraUseCases .....	39
Εικόνα 6.23: Μέθοδος bindCameraUseCases .....	40

Εικόνα 6.24: Μέθοδος bindCameraUseCases .....	40
Εικόνα 6.25: Μέθοδος detectObjects .....	41
Εικόνα 6.26: Μέθοδος onConfigurationChanged .....	41
Εικόνα 6.27: Μέθοδος onResults .....	42
Εικόνα 6.28: Μέθοδος onError .....	42
Εικόνα 6.29: Imports και μεταβλητή.....	43
Εικόνα 6.30: Permission Launcher.....	43
Εικόνα 6.31: Μέθοδος onCreate .....	44
Εικόνα 6.32: Μέθοδος navigateToCamera .....	44
Εικόνα 6.33: Μέθοδος hasPermissions .....	44
Εικόνα 6.34: Hashmap μεταφράσεων .....	45
Εικόνα 6.35: Hashmap μεταφράσεων .....	46
Εικόνα 6.36: Hashmap μεταφράσεων .....	47
Εικόνα 6.37: MainActivity .....	48
Εικόνα 6.38: Imports .....	48
Εικόνα 6.39: Μεταβλητές.....	49
Εικόνα 6.40: Μέθοδος setupObjectDetector .....	50
Εικόνα 6.41: Μέθοδος setupObjectDetector .....	51
Εικόνα 6.42: Μέθοδος detect .....	52
Εικόνα 6.43: Διασύνδεση DetectorListener .....	53
Εικόνα 6.44: Companion Object .....	54
Εικόνα 6.45: Imports .....	55
Εικόνα 6.46: Μεταβλητές.....	56
Εικόνα 6.47: Μέθοδος clear .....	56
Εικόνα 6.48: Μέθοδος initPaints.....	57
Εικόνα 6.49: Μέθοδος draw .....	57
Εικόνα 6.50: Μέθοδος draw .....	58
Εικόνα 6.51: Μέθοδος setResults.....	58
Εικόνα 6.52: Μέθοδος onTouchEvent .....	59
Εικόνα 6.53: Μέθοδος onTouchEvent .....	59
Εικόνα 6.54: Μέθοδος onTouchEvent .....	60
Εικόνα 6.55: Μέθοδος onTouchEvent .....	60
Εικόνα 6.56: Μέθοδος calculateScaledBoundingBox.....	60
Εικόνα 6.57: Μέθοδος isTouchWithinBoundingBox.....	61
Εικόνα 6.58: Μέθοδος triggerVibration .....	61
Εικόνα 6.59: Μέθοδος initTextToSpeech .....	62
Εικόνα 6.60: Μέθοδος onDetachedFromWindow .....	62
Εικόνα 6.61: Μέθοδος speakText .....	63
Εικόνα 6.62: Activity_main.xml .....	63
Εικόνα 6.63: fragment_camera.xml .....	64
Εικόνα 6.64: Info_bottom_sheet.xml .....	65
Εικόνα 6.65: Info_bottom_sheet.xml .....	65
Εικόνα 6.66: Info_bottom_sheet.xml .....	66
Εικόνα 6.67: Info_bottom_sheet.xml .....	66
Εικόνα 6.68: Info_bottom_sheet.xml .....	67
Εικόνα 6.69: Info_bottom_sheet.xml .....	67

Εικόνα 6.70: Info_bottom_sheet.xml .....	67
Εικόνα 6.71: Info_bottom_sheet.xml .....	68
Εικόνα 6.72: Info_bottom_sheet.xml .....	68
Εικόνα 6.73: Info_bottom_sheet.xml .....	68
Εικόνα 6.74: Info_bottom_sheet.xml .....	69
Εικόνα 6.75: Info_bottom_sheet.xml .....	69
Εικόνα 6.76: Info_bottom_sheet.xml .....	69
Εικόνα 6.77: Info_bottom_sheet.xml .....	70
Εικόνα 6.78: Info_bottom_sheet.xml .....	70
Εικόνα 6.79: Info_bottom_sheet.xml .....	70

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
Τ.Ν.	Τεχνητή Νοημοσύνη
Μ.Μ.	Μηχανική Μάθηση
Β.Μ.	Βαθιά Μάθηση
Υ.Τ.	Υποβοηθητική Τεχνολογία
С.N.N.	Convolutional Neural Network
Υ.Ο.Λ.Ο.	You Only Look Once
С.Ο.С.Ο.	Common Objects in COntext

## Κεφάλαιο 1ο: Εισαγωγή και Ιστορική Αναδρομή

### 1.1 Εισαγωγή

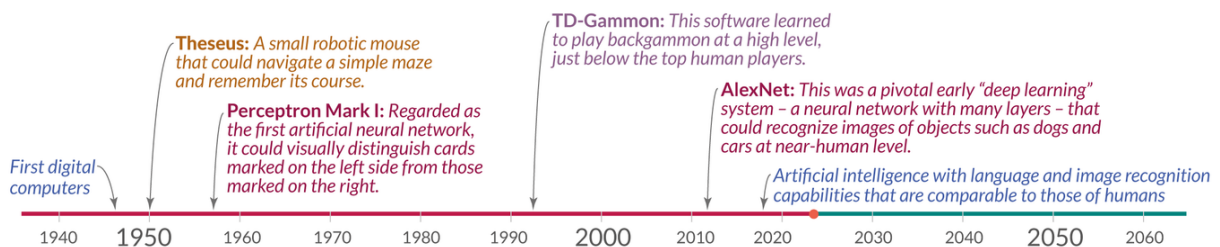
Η Τεχνητή Νοημοσύνη αποτελεί ένα συναρπαστικό και ταχύτατα αναπτυσσόμενο πεδίο που αγγίζει κάθε πτυχή της ανθρώπινης δραστηριότητας. Από την αυτοματοποίηση καθημερινών εργασιών έως την ανάπτυξη πρωτοποριακών εφαρμογών, η ΤΝ φέρνει στο προσκήνιο νέες δυνατότητες και προκλήσεις, διαμορφώνοντας το μέλλον του κόσμου μας. Η ιδέα της ΤΝ δεν είναι καινούργια. Ήδη από την αρχαιότητα, φιλόσοφοι και στοχαστές ονειρεύονταν την ύπαρξη μηχανών που θα μπορούσαν να σκέφτονται και να δρουν σαν άνθρωποι. Σταδιακά, από τις μυθολογικές ιστορίες τεχνητών όντων έως τις πρώτες επιστημονικές εξελίξεις, η ΤΝ άρχισε να παίρνει σάρκα και οστά. Συναντάμε αναφορές σε τεχνητά όντα, όπως ο Πυγμαλίων του Οβίδιου και ο Τάλως της Κρήτης, που φέρουν στοιχεία τεχνητής νοημοσύνης. [1] Μπορεί να μας φαίνεται απίστευτο, αλλά οι υπολογιστές, που σήμερα παίζουν τόσο σημαντικό ρόλο στη ζωή μας, είναι μια σχετικά νέα τεχνολογία. Μέσα σε λίγες μόνο δεκαετίες εξελίχθηκαν ραγδαία και πλέον τους χρησιμοποιούμε καθημερινά για διάφορες εργασίες. Σύμφωνα με το παρακάτω σχήμα 1.1 [1], οι πρώτοι ψηφιακοί υπολογιστές εφευρέθηκαν μόλις πριν από ογδόντα περίπου χρόνια.



Σχήμα 1.1: Σημαντικά ιστορικά επιτεύγματα στον κόσμο των υπολογιστών [1].

Η δημιουργία μηχανών με ανθρώπινη νοημοσύνη απασχολεί τους επιστήμονες υπολογιστών εδώ και δεκαετίες. Στο σχήμα 1.2 [1] που ακολουθεί, θα δούμε μερικά σημαντικά συστήματα ΤΝ και τις δυνατότητές τους. Ένα πρώτο παράδειγμα είναι ο Theseus, ένας τηλεκατευθυνόμενος ποντικός του 1950, δημιουργία του Claude Shannon. Ο Theseus μπορούσε να βρει τη διέξοδο από λαβύρινθο και να θυμηθεί τη διαδρομή του. Μέσα σε μόλις 70 χρόνια, η Τεχνητή Νοημοσύνη έχει κάνει απίστευτα άλματα.

#### A timeline of notable artificial intelligence systems



Σχήμα 1.2: Χρονολογική εξέλιξη της τεχνητής νοημοσύνης [1].

## **1.2 Η Εργασία και η Δομή της**

Το κείμενο της εργασίας αποτελείται από οκτώ κεφάλαια. Αρχικά είδαμε μια ιστορική αναδρομή για το πώς ξεκίνησε η τεχνητή νοημοσύνη στο εισαγωγικό κεφάλαιο. Στην συνέχεια θα παρατεθούν οι ορισμοί της τεχνητής νοημοσύνης, καθώς επίσης και κάποιοι από τους τομείς της. Στο επόμενο κεφάλαιο θα ερευνήσουμε μια καινοτόμο τεχνολογία την απτική αλληλεπίδραση και ορισμένους τρόπους με τους οποίους μπορούμε να την πετύχουμε. Άμεσα αλληλένδετο με το προηγούμενο κεφάλαιο είναι το επόμενο, στο οποίο θα μελετήσουμε τεχνολογίες, οι οποίες έχουν ως στόχο να βελτιώσουν την ζωή κάποιων ανθρώπων οι οποίοι αντιμετωπίζουν διαφόρων ειδών προβλήματα και δυσκολίες, όπως για παράδειγμα άνθρωποι με προβλήματα όρασης. Όπως επίσης αυτόν τον σκοπό έχει και η παρούσα εργασία την δημιουργία μιας εφαρμογής, η οποία θα λειτουργεί ως βοηθός στα άτομα αυτά, για την διευκόλυνση της ζωής τους ανιχνεύοντας αντικείμενα. Στο πέμπτο κεφάλαιο θα δούμε το τι τεχνολογίες συνδυάζει η εφαρμογή, την διεπαφή, καθώς επίσης και τον κώδικα της. Στο προτελευταίο κεφάλαιο θα δούμε το πώς αξιολογήθηκε η εφαρμογή, μέσω ενός ερωτηματολογίου και κάποια στατιστικά από τους συμμετέχοντες. Τέλος θα δούμε κάποια συμπεράσματα τα οποία προέκυψαν από την δημιουργία της εφαρμογής καθώς και την αξιολόγηση της.

## **1.3 Επίλογος**

Στο παρόν κεφάλαιο έγινε μια ιστορική αναδρομή για την τεχνητή νοημοσύνη και αναλύθηκε η δομή της εργασίας, με θέμα την δημιουργία μιας εφαρμογής ως βοηθός σε άτομα με προβλήματα όρασης, η οποία ανιχνεύει αντικείμενα, συνδυάζοντας και άλλες υποβοηθητικές τεχνολογίες.

## Κεφάλαιο 2ο: Τεχνητή Νοημοσύνη

### 2.1 Εισαγωγή

Η Τεχνητή Νοημοσύνη αποτελεί ένα απέραντο και ραγδαία αναπτυσσόμενο πεδίο, πλούσιο σε εξειδικευμένες υποκατηγορίες, όπως η μηχανική μάθηση, η βαθιά μάθηση, η επεξεργασία φυσικής γλώσσας, η όραση υπολογιστών και η ρομποτική. Κάθε υποπεδίο εστιάζει σε μοναδικές προκλήσεις και βρίσκει εφαρμογή σε πληθώρα πεδίων, συμβάλλοντας στην συνολική πρόοδο της ΤΝ. Στο παρόν κεφάλαιο, θα εστιάσουμε στις θεμελιώδεις έννοιες και τα βασικά εργαλεία της ΤΝ.

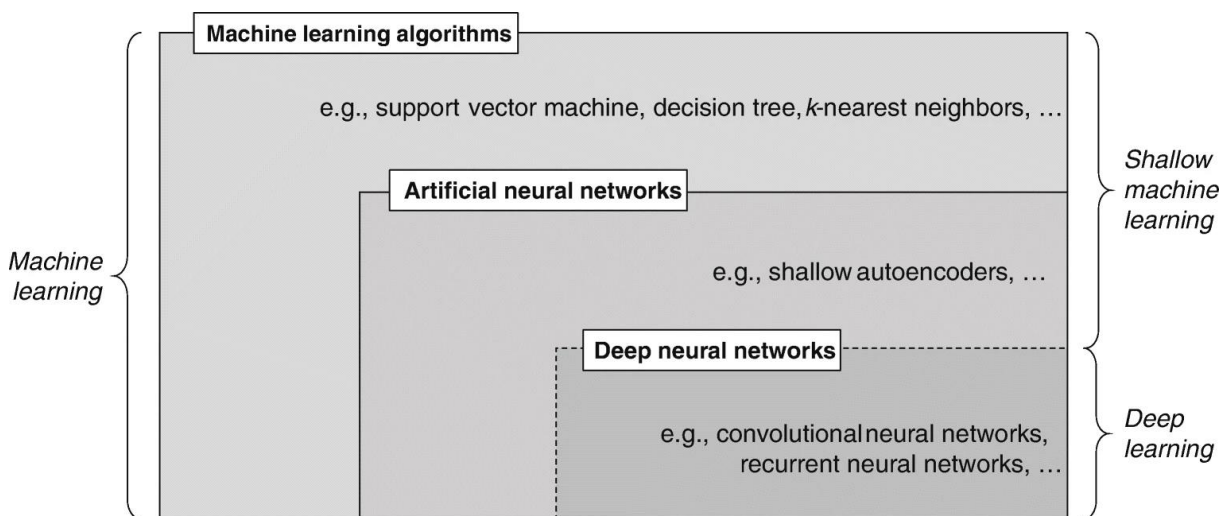
### 2.2 Ορισμοί της ΤΝ

Ο παραδοσιακός ορισμός της ΤΝ την παρουσιάζει σαν μια τέλεια απομίμηση της ανθρώπινης σκέψης από τους υπολογιστές. Ωστόσο, κάποιοι ειδικοί θεωρούν ότι οι σημερινές εφαρμογές δεν είναι αρκετά περίπλοκες για να θεωρηθούν πραγματική ΤΝ. Αν χρησιμοποιήσουμε αυτόν τον ορισμό, θα δώσουμε την εντύπωση πως η ΤΝ δεν υπάρχει καν! Πρακτικά, θα την αποκλείαμε από την συζήτηση επιλέγοντας έναν τόσο αυστηρό ορισμό. [2] Ένας συνηθισμένος ορισμός της ΤΝ την παρουσιάζει ως τεχνολογία που κάνει τα μηχανήματα να μοιάζουν με ανθρώπους, δίνοντάς τους τη δυνατότητα να εκτελούν περίπλοκες εργασίες. Ωστόσο, αυτός ο ορισμός είναι λίγο αόριστος. Στην πραγματικότητα, απλά περιγράφει την ΤΝ με άλλα λόγια, χωρίς να διευκρινίζει τι είναι πραγματικά οι "σύνθετες ανθρώπινες δεξιότητες". Όσο δεν γίνεται πιο συγκεκριμένος αυτός ο όρος, παραμένει θολό τι ακριβώς είναι η ΤΝ. Παρόμοια αοριστία υπάρχει και στον ορισμό που μιλάει για την ικανότητα των υπολογιστών να εκτελούν πολύπλοκες εργασίες σε περίπλοκα περιβάλλοντα [2]. Άλλοι ορισμοί δεν μένουν στην αοριστία και εξηγούν καλύτερα τις ικανότητες και τα καθήκοντα της ΤΝ. Για παράδειγμα, ο επιστήμονας υπολογιστών Nils John Nilsson την περιγράφει ως μια τεχνολογία που επιτρέπει στα μηχανήματα να λειτουργούν αποδοτικά και με προνοητικότητα μέσα στο περιβάλλον τους. Άλλοι επιστήμονες τονίζουν την ικανότητα αντίληψης, την επιδίωξη στόχων, την έναρξη ενεργειών και τη μάθηση από προηγούμενες εμπειρίες. Παρόμοιο πνεύμα έχει και ο ορισμός της Ομάδας Υψηλού Επιπέδου Εμπειρογνομόνων για την Τεχνητή Νοημοσύνη της Ευρωπαϊκής Επιτροπής, που χαρακτηρίζει την ΤΝ ως συστήματα ικανά να αναλύουν το περιβάλλον τους, να λαμβάνουν αυτόνομες αποφάσεις και να ενεργούν για να πετύχουν συγκεκριμένους στόχους [2].

### 2.3 Μηχανική Μάθηση και Βαθιά Μάθηση

Η μηχανική μάθηση αποτελεί έναν κλάδο της τεχνητής νοημοσύνης που εστιάζει στην ανάπτυξη αλγορίθμων υπολογιστών ικανών να βελτιώνουν την απόδοσή τους σε μια εργασία με την πάροδο του χρόνου, αξιοποιώντας την εμπειρία. Με απλά λόγια, ένα σύστημα ΜΜ "μαθαίνει" από τα δεδομένα στα οποία εκτίθεται, προσαρμόζοντας και εξελίσσοντας τις δυνατότητές του. Σε αντίθεση με τα παραδοσιακά προγράμματα, τα οποία βασίζονται σε σταθερούς κανόνες και οδηγίες, τα συστήματα ΜΜ διαθέτουν την ικανότητα αυτό-βελτίωσης και αυτό-οργάνωσης. Τα δεδομένα λειτουργούν ως "δάσκαλος", καθοδηγώντας το σύστημα προς την κατανόηση και την επίλυση προβλημάτων [3]. Σαν έξυπνοι μαθητές, οι αλγόριθμοι Μηχανικής Μάθησης εκπαιδεύονται με δεδομένα, "διαβάζοντας" και αναγνωρίζοντας μοτίβα. Χωρίς ρητές οδηγίες, οι αλγόριθμοι αυτομάτως "κατανοούν" τα δεδομένα και μπορούν να κάνουν προβλέψεις για νέα δεδομένα. Η ΜΜ επιτρέπει στους υπολογιστές να μαθαίνουν μόνοι τους σημαντικές συνδέσεις και μοτίβα μέσα από παραδείγματα, σαν να μαθαίνουν από την εμπειρία [4]. Χάρη σε αυτή την πρόοδο, αναπτύσσονται τώρα έξυπνα συστήματα που σκέφτονται σχεδόν σαν άνθρωποι και επηρεάζουν όλες τις πτυχές της ζωής μας, από τη δουλειά μέχρι τον τρόπο που αλληλεπιδρούμε στο

διαδίκτυο. Για παράδειγμα, οι επιχειρήσεις χρησιμοποιούν τη ΜΜ για να παίρνουν καλύτερες αποφάσεις που αυξάνουν την παραγωγικότητα, την ικανοποίηση των εργαζομένων και τη διατήρησή τους. Ακόμα, αναπτύσσονται βοηθοί που προσαρμόζονται στον κάθε χρήστη, ενώ ακόμα και οι χρηματιστηριακές αγορές επηρεάζονται από συστήματα συναλλαγών που βασίζονται στη ΜΜ. Αυτά τα έξυπνα συστήματα που λύνουν δύσκολα προβλήματα συνήθως τα αποκαλούμε ΤΝ. Πίσω τους βρίσκονται αναλυτικά μοντέλα που προβλέπουν, προτείνουν και δίνουν απαντήσεις. Παλιά, φτιάχναμε αυτά τα μοντέλα γράφοντας απευθείας στους υπολογιστές τους κανόνες και τη λογική που χρειάζονταν (π.χ., συστήματα γιατρών που βασίζονται σε γνώσεις ειδικών). Τώρα όμως, χάρη σε νέες τεχνολογίες και πιο ισχυρούς υπολογιστές, μπορούμε να χρησιμοποιήσουμε τη ΜΜ για να φτιάξουμε αυτά τα μοντέλα πιο εύκολα. Με τη ΜΜ δεν χρειάζεται πια να μεταφέρουμε με το χέρι τις γνώσεις μας στους υπολογιστές, κάτι που κάνει την ανάπτυξη έξυπνων συστημάτων πολύ πιο γρήγορη [4]. Στο παρακάτω σχήμα θα δούμε την σχέση ΜΜ και ΒΜ ώστε να καταλάβουμε καλύτερα πως συνδέονται μεταξύ τους. Η Βαθιά Μάθηση είναι ο τύπος αλγορίθμων που λειτουργούν καλύτερα όταν έχουν πολλά δεδομένα για να μάθουν, ειδικά για πράγματα όπως εικόνες, βίντεο και ήχο. Αλλά για πιο απλά δεδομένα, μερικές φορές οι "απλοί" αλγόριθμοι Μηχανικής Μάθησης τα καταφέρνουν καλύτερα. Επιπλέον, αυτά τα απλά μοντέλα είναι και πιο εύκολο να καταλάβουμε πώς λειτουργούν, σε αντίθεση με τη Βαθιά Μάθηση. Παρόλο που η Βαθιά Μάθηση μπορεί να κάνει πράγματα που δεν μπορούμε εμείς, δεν μπορεί ακόμα να σκεφτεί πραγματικά σαν εμάς τους ανθρώπους. Για παράδειγμα, δεν μπορεί να καταλάβει πραγματικά το νόημα των λέξεων. [4] Το σχήμα 2.1 [4] που ακολουθεί μας δείχνει τα επίπεδα ΜΜ.



Σχήμα 2.1: Επίπεδα μηχανικής μάθησης [4].

Υπάρχουν τρία είδη Μηχανικής Μάθησης:

- **Εποπτευόμενη Μάθηση:** Ο αλγόριθμος εκπαιδεύεται με δεδομένα που έχουν ήδη ταξινομηθεί ή επισημανθεί. Ο σκοπός είναι ο αλγόριθμος να μάθει να ταξινομεί νέα δεδομένα με βάση τα υπάρχοντα. [3]
- **Μη-εποπτευόμενη Μάθηση:** Ο αλγόριθμος λαμβάνει δεδομένα χωρίς ταξινομήσεις ή ετικέτες. Ο στόχος είναι ο αλγόριθμος να ανακαλύψει μόνος του τις δομές και τα μοτίβα που υπάρχουν στα δεδομένα. [3]
- **Ενισχυτική Μάθηση:** Ο αλγόριθμος μαθαίνει μέσω αλληλεπίδρασης με το περιβάλλον. Λαμβάνει θετική ή αρνητική ανατροφοδότηση αντίστοιχα για τις ενέργειές του και στόχος του

είναι να βελτιστοποιήσει την στρατηγική του έτσι ώστε να μεγιστοποιήσει την θετική ανατροφοδότηση. [4]

### 2.3.1 Εποπτευόμενη Μηχανική Μάθηση

Στην **εποπτευόμενη μάθηση**, για να μάθει ο αλγόριθμος χρειάζεται δεδομένα με "απαντήσεις". Αυτά τα δεδομένα περιλαμβάνουν τόσο τα αρχικά στοιχεία (τις "ερωτήσεις") όσο και τις σωστές απαντήσεις. Φανταστείτε ότι θέλουμε να προβλέψουμε πόσοι ενεργοί χρήστες θα εγγραφούν σε μια πλατφόρμα μέσα σε ένα μήνα. Για να το πετύχει αυτό, ο αλγόριθμος χρειάζεται παραδείγματα με στοιχεία σαν τον αριθμό των προϊόντων που πωλήθηκαν ή τις κριτικές χρηστών (αυτά είναι οι "ερωτήσεις"). Ταυτόχρονα, χρειάζεται να ξέρει και πόσοι χρήστες ενεργοποιήθηκαν τελικά (αυτή είναι η "απάντηση"). Με αυτά τα δεδομένα "ερώτηση-απάντηση", ο αλγόριθμος μαθαίνει να κάνει προβλέψεις για νέα δεδομένα.

Επιπλέον, η εποπτευόμενη μάθηση χωρίζεται σε δύο κατηγορίες: στην **παλινδρόμηση**, ο αλγόριθμος προβλέπει μια τιμή (π.χ., αριθμό χρηστών), ενώ στην **ταξινόμηση**, ο αλγόριθμος κατατάσσει τα δεδομένα σε κατηγορίες (π.χ., "αγοραστές" ή "περηγητές") [4].

### 2.3.2 Μη-εποπτευόμενη Μηχανική Μάθηση

Στη **μη-εποπτευόμενη μάθηση**, ο αλγόριθμος δεν έχει "απαντήσεις" για να μάθει. Αντίθετα, του δίνουμε μόνο τα αρχικά δεδομένα (τις "ερωτήσεις") και πρέπει να βρει μόνος του τι μοτίβα κρύβουν αυτά τα δεδομένα. Σκεφτείτε το σαν να προσπαθεί να ανακαλύψει μόνος του τις κατηγορίες σε ένα συρτάρι με διάφορα αντικείμενα. Μπορεί για παράδειγμα να βρει ομάδες αντικειμένων με κοινά χαρακτηριστικά, όπως τα ρούχα ή τα εργαλεία (ομαδοποίηση ή clustering). Επίσης, μπορεί να συμπυκνώσει τις πληροφορίες των δεδομένων ώστε να είναι πιο εύκολο να τις επεξεργαστούμε (μείωση διαστατικότητας) [4].

Ένα παράδειγμα μη-εποπτευόμενης μάθησης στην αγορά είναι να κατηγοριοποιήσουμε πελάτες σε ομάδες με βάση τις αγοραστικές τους συνήθειες. Έτσι, μπορούμε να στείλουμε πιο στοχευμένες διαφημίσεις σε κάθε ομάδα.

### 2.3.3 Ενισχυτική Μάθηση

Στην **ενισχυτική μάθηση** δεν δίνουμε στον αλγόριθμο ούτε "ερωτήσεις" ούτε "απαντήσεις". Αντίθετα, του λέμε ποια είναι η κατάσταση, ποιος είναι ο στόχος και τι ενέργειες μπορεί να κάνει. Μετά, ο αλγόριθμος μαθαίνει "παίζοντας" και προσπαθώντας να πετύχει τον στόχο με τον καλύτερο τρόπο, σαν να μαθαίνει ένα παιχνίδι με δοκιμές και λάθη. Αυτός ο τύπος μάθησης έχει χρησιμοποιηθεί με επιτυχία σε παιχνίδια στον υπολογιστή αλλά μπορεί να χρησιμοποιηθεί και σε πιο σύνθετα περιβάλλοντα, όπως στις ηλεκτρονικές αγορές [4].

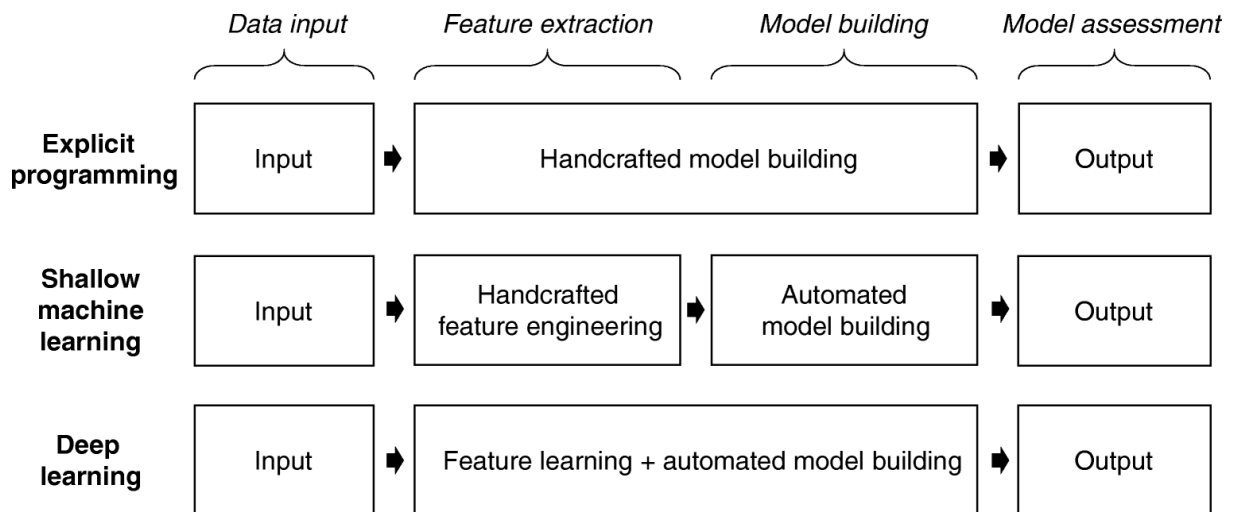
### 2.3.4 Ανάπτυξη Μοντέλων

Σε αυτήν την υποενότητα θα αναφέρουμε τους διαφορετικούς τρόπους, ώστε να δημιουργήσουμε ένα αναλυτικό μοντέλο. Υπάρχουν τρεις βασικοί τρόποι:

1. **Προγραμματισμός με κανόνες:** Σε αυτή την περίπτωση, ένας άνθρωπος ορίζει χειροκίνητα τους κανόνες που θα χρησιμοποιήσει το μοντέλο. Είναι σαν να γράφουμε οδηγίες βήμα-βήμα για το πώς να λύσει το μοντέλο το πρόβλημα.

2. **Shallow MM**: Εδώ, το μοντέλο μαθαίνει μόνο του από δεδομένα, αλλά με έναν σχετικά απλό αλγόριθμο.
3. **Βαθιά Μάθηση**: Αυτή είναι η πιο περίπλοκη προσέγγιση και χρησιμοποιεί πολύπλοκους αλγορίθμους για να μάθει από δεδομένα. [4]

Η συνέχεια του κειμένου θα επικεντρωθεί στις μεθόδους 2 και 3 (shallow ML και βαθιά μάθηση) και θα δούμε πώς συλλέγουμε δεδομένα, πώς τα προετοιμάζουμε για το μοντέλο, πώς χτίζεται το ίδιο το μοντέλο και τέλος πώς αξιολογούμε την απόδοσή του [4]. Στο παρακάτω σχήμα 2.2 [4] βλέπουμε την δομή ενός μοντέλου MM, εάν χρησιμοποιήσουμε MM, BM ή το χτίσουμε εμείς:



Σχήμα 2.2: Δομή μοντέλου [4]

Όταν χτίζουμε αυτά τα μοντέλα αυτόματα, ο αλγόριθμος ψάχνει μοτίβα στα δεδομένα που του δίνουμε. Όπως είπαμε και πριν, η shallow MM χρειάζεται καλά επιλεγμένα χαρακτηριστικά για να λειτουργήσει σωστά. Ανάλογα με το πρόβλημα που θέλουμε να λύσουμε, χρησιμοποιούμε διαφορετικούς τύπους αλγορίθμων. Για παράδειγμα, όταν θέλουμε να κατηγοριοποιήσουμε κάτι (π.χ., να πούμε αν ένα email είναι spam ή όχι), χρησιμοποιούμε αλγορίθμους σαν τα "δέντρα αποφάσεων". Αυτοί οι αλγόριθμοι χωρίζουν συνεχώς τα δεδομένα μας σε μικρότερες ομάδες, ανάλογα με τα χαρακτηριστικά τους, μέχρι να καταλήξουν σε μια τελική κατηγορία. Άλλοι αλγόριθμοι, σαν τις "μηχανές διανυσμάτων υποστήριξης", προσπαθούν να βρουν μια "γραμμή" που χωρίζει τα δεδομένα σε διαφορετικές κατηγορίες και όσο πιο καθαρά γίνεται. Ενδεχομένως να χρειαστεί ακόμα και να μετατρέψουν τα αρχικά δεδομένα σε κάτι πιο περίπλοκο για να το πετύχουν αυτό [4]. Όπως βλέπουμε, υπάρχουν πολλοί τρόποι να φτιάξουμε αυτά τα μοντέλα και ο καλύτερος τρόπος εξαρτάται από τα δεδομένα που έχουμε και το τι θέλουμε να μάθουμε. Αντίθετα η BM είναι πιο δυνατή. Μπορεί να πάρει τα "ακατέργαστα" δεδομένα μας, ακόμα και αν είναι πολύ περίπλοκα, και να μάθει μόνη της τι χαρακτηριστικά είναι σημαντικά για να λύσει το πρόβλημα. Έτσι, δεν χρειάζεται να τα προετοιμάσουμε εμείς πολύ πριν τα βάλουμε στο μοντέλο. Επιπλέον, η BM μπορεί μερικές φορές να χρησιμοποιηθεί μόνο για να βρει αυτά τα σημαντικά χαρακτηριστικά. Στη συνέχεια, αυτά τα χαρακτηριστικά μπορούν να χρησιμοποιηθούν από άλλους, πιο απλούς αλγορίθμους μηχανικής μάθησης, όπως τα δέντρα αποφάσεων ή τις "μηχανές διανυσμάτων υποστήριξης" [4].

## 2.4 Όραση Υπολογιστών

Ένας άμεσα αλληλένδετος τομέας της ΤΝ, με την μηχανική και βαθιά μάθηση είναι και ο τομέας της όρασης υπολογιστών. Την οποία χρησιμοποιούμε και στην εφαρμογή την οποία αναπτύξαμε. Με τους τομείς λοιπόν που αναφέραμε φιλοδοξούμε να χαρίσουμε στους υπολογιστές ικανότητες αντίστοιχες με αυτές του ανθρώπου. Στόχος είναι η ανίχνευση, η κατανόηση και η λήψη αποφάσεων με βάση δεδομένα, τόσο από το παρελθόν όσο και από το παρόν. Η έρευνα σε αυτόν τον τομέα είναι ακόμα σε εξέλιξη. Η όραση υπολογιστών αποτελεί βασικό συστατικό του Διαδικτύου των Πραγμάτων, του Βιομηχανικού Διαδικτύου των Πραγμάτων και των διασυνδέσεων εγκεφάλου-υπολογιστή. Χρησιμοποιώντας μηχανική μάθηση και όραση υπολογιστών, μπορούμε να αναλύσουμε και να ελέγξουμε σύνθετες ανθρώπινες δραστηριότητες σε ψηφιακές μορφές (βίντεο, εικόνες),όπως επίσης έχουμε την δυνατότητα να ανιχνεύσουμε αντικείμενα [5].Όπως αναφέραμε παραπάνω η ΜΜ λειτουργεί σε τρία βασικά βήματα:

1. **Συλλογή δεδομένων:** Η πρώτη ύλη για την εκπαίδευση των αλγορίθμων.
2. **Εκπαίδευση μοντέλου:** Ο αλγόριθμος "μαθαίνει" από τα δεδομένα.
3. **Χρήση εκπαιδευμένου μοντέλου:** Το μοντέλο χρησιμοποιείται για προβλέψεις, λήψη αποφάσεων, ανίχνευση αντικειμένων.

Η όραση υπολογιστών έχει βρει εφαρμογή σε πλήθος τομέων, με ορισμένους από τους πιο σημαντικούς να περιλαμβάνουν:

### 1. Ασφάλεια και Επιτήρηση:

- Αναγνώριση προσώπων και ανίχνευση κίνησης
- Παρακολούθηση συστημάτων

### 2. Ιατρική:

- Ανάλυση ιατρικών εικόνων
- Ρομποτική χειρουργική

### 3. Μεταφορές:

- Ανάπτυξη αυτόνομων οχημάτων
- Βελτίωση κυκλοφορίας

### 5. Γεωργία:

- Παρακολούθηση καλλιεργειών
- Βελτιστοποίηση άρδευσης

### 6. Διασκέδαση:

- Ειδικά εφέ: Η όραση υπολογιστών χρησιμοποιείται για
- Εικονική και επανυξημένη πραγματικότητα
- Διαδραστικές εμπειρίες

### 7. Βιομηχανία:

- **Επιθεώρηση προϊόντων**
- **Αυτοματοποίηση εργασιών**
- **Βελτιστοποίηση παραγωγής**

## 8. Έρευνα και Ανάπτυξη:

- **Επιστημονική έρευνα**
- **Ανάπτυξη νέων τεχνολογιών**

Αυτές είναι μόνο μερικές από τις πολλές εφαρμογές της όρασης υπολογιστών. Καθώς η τεχνολογία συνεχίζει να αναπτύσσεται, η όραση υπολογιστών αναμένεται να παίξει ακόμα πιο σημαντικό ρόλο σε διάφορους τομείς της καθημερινής μας ζωής. Για την υλοποίηση τέτοιων συστημάτων, χρησιμοποιούνται διάφορες βιβλιοθήκες λογισμικού μία από αυτές είναι η TensorFlow, την οποία χρησιμοποιήσαμε στην εφαρμογή [6]. Τα μοντέλα ανίχνευσης αξιοποιούν διάφορες τεχνικές νευρωνικών δικτύων (CNN, RNN, LSTM, GRU) και δίκτυα Bayes. [5]. Η εξαγωγή χαρακτηριστικών από εικόνες είναι απαραίτητη για την επιτυχή ανίχνευση αντικειμένων. Τα νευρωνικά δίκτυα τύπου (CNN) υπερέχουν σε αυτόν τον τομέα. Αυτά τα δίκτυα αποτελούνται από ειδικούς νευρώνες που έχουν τρία μεγέθη: πλάτος, ύψος και βάθος. Η δημοτικότητα των CNNs έχει εκτοξευθεί πρόσφατα χάρη σε τρεις βασικούς παράγοντες:

1. **Διαθεσιμότητα δεδομένων:** Υπάρχουν πλέον πολλά και εύκολα προσβάσιμα σύνολα δεδομένων εικόνων, τα οποία χρειάζονται τα CNNs για να "εκπαιδευτούν".
2. **Μονάδες επεξεργασίας γραφικών (GPUs):** Οι GPUs είναι ειδικοί επεξεργαστές που επιταχύνουν πολύ την λειτουργία των CNNs.
3. **Τεχνικές κανονικοποίησης:** Αυτές οι τεχνικές βοηθούν τα CNNs να αποφύγουν την υπερπροσαρμογή στα δεδομένα εκπαίδευσης, βελτιώνοντας έτσι την γενίκευσή τους σε νέες εικόνες [5].

Η OpenCV είναι ένα χρήσιμο εργαλείο για προγραμματιστές. Πρόκειται για μια βιβλιοθήκη λογισμικού που "συνεργάζεται" με διάφορες γλώσσες προγραμματισμού (π.χ., Android, Java) και λειτουργεί σε πολλά λειτουργικά συστήματα (π.χ., Windows, iOS), βοηθά στην επεξεργασία εικόνων, στην ανάλυση βίντεο, στην ανίχνευση αντικειμένων μέσα σε εικόνες, και γενικά σε εφαρμογές μηχανικής μάθησης που σχετίζονται με την όραση υπολογιστών. Η όραση υπολογιστών αναλύει εικόνες για να λύσει προβλήματα. Θεωρεί μια εικόνα σαν έναν πίνακα με εικονοστοιχεία (pixels) και χρησιμοποιεί μοτίβα για να αναγνωρίσει την εικόνα. Χρησιμοποιώντας την όραση υπολογιστών, μπορούμε να αυτοματοποιήσουμε εργασίες όπως η παρακολούθηση, ο έλεγχος και η επιτήρηση. Το μοντέλο που χρησιμοποιήσαμε εμείς ονομάζεται MobileNetV1 το οποίο είναι κατάλληλο για χρήση σε κινητές συσκευές και χρησιμοποιεί νευρωνικά δίκτυα τύπου CNN.

### 2.4.1 MobileNet

Το MobileNet αποτελεί μια καινοτόμος αρχιτεκτονική δικτύου βαθιάς μάθησης (CNN) που έχει σχεδιαστεί ειδικά για εφαρμογές αναγνώρισης εικόνας σε μικρές πλατφόρμες, όπως κινητά τηλέφωνα και ενσωματωμένα συστήματα. Η βασική αρχή του MobileNet έγκειται στην αξιοποίηση της τεχνικής

"διαχωρίσιμων συνελίξεων" (depthwise separable convolutions), η οποία επιτρέπει την επίτευξη σημαντικής μείωσης στο μέγεθος του μοντέλου και στο υπολογιστικό κόστος, συγκριτικά με τα κλασικά συνελκτικά δίκτυα (CNN) [7].

### Χαρακτηριστικά:

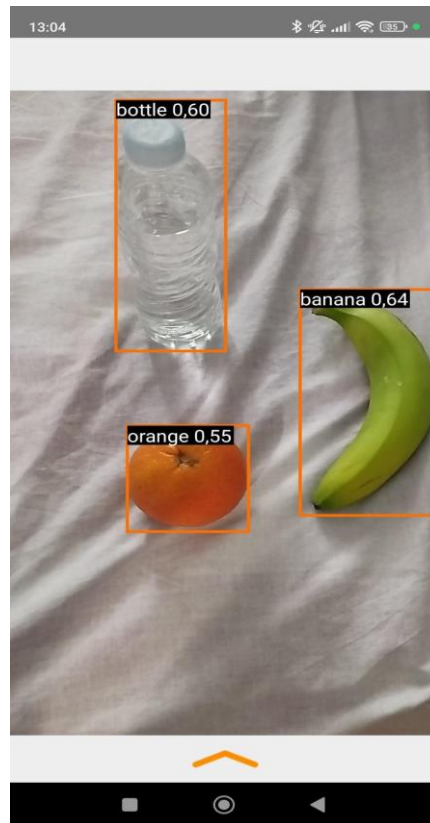
- **Μικρό Μέγεθος:** Το MobileNet διακρίνεται για το εξαιρετικά μικρό του μέγεθος, καθιστώντας το ιδανικό για συσκευές με περιορισμένους πόρους αποθήκευσης.
- **Υψηλή Ταχύτητα:** Η υλοποίηση "διαχωρίσιμων συνελίξεων" οδηγεί σε δραστική μείωση του υπολογιστικού χρόνου, εξασφαλίζοντας ταχύτερη εκτέλεση.
- **Χαμηλή Κατανάλωση Ενέργειας:** Η αποδοτική λειτουργία του MobileNet συμβάλλει στην εξοικονόμηση ενέργειας, αυξάνοντας την αυτονομία της μπαταρίας σε φορητές συσκευές.
- **Ευελιξία:** Το MobileNet δύναται να προσαρμοστεί σε ένα ευρύ φάσμα εφαρμογών, χάρη στην δυνατότητα τροποποίησης του μεγέθους και της ταχύτητας του μοντέλου μέσω των υπερπαραμέτρων "width multiplier" και "resolution multiplier" [7].

### Εφαρμογές:

- **Αναγνώριση Αντικειμένων σε Φωτογραφίες:** Το MobileNet μπορεί να ταυτοποιήσει αντικείμενα που απεικονίζονται σε φωτογραφίες, παρέχοντας χρήσιμες πληροφορίες για το περιεχόμενο της εικόνας.
- **Αναγνώριση Προσώπων:** Η δυνατότητα αναγνώρισης προσώπων αξιοποιείται σε διάφορες εφαρμογές, όπως η εξατομίκευση εμπειριών χρήστη ή η αυθεντικοποίηση ταυτότητας.
- **Ταξινόμηση Εικόνων:** Το MobileNet δύναται να κατηγοριοποιήσει εικόνες βάσει του περιεχομένου τους, διευκολύνοντας την οργάνωση και την αναζήτηση οπτικού υλικού.

Παρακάτω θα δούμε κάποιες εικόνες 2.1-2.6 τραβηγμένες από την ίδια την εφαρμογή με διάφορων ειδών ανιχνευμένα αντικείμενα.

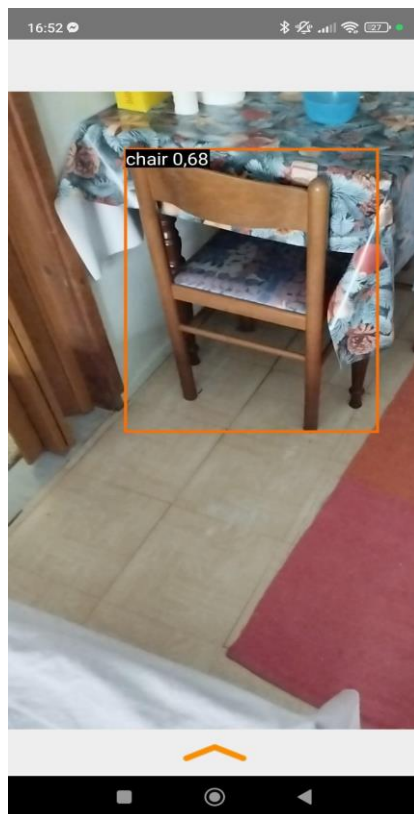
## Κεφάλαιο 2



Εικόνα 2.1: Εντοπισμός φρούτων και πλαστικού μπουκαλιού



Εικόνα 2.2: Εντοπισμός τηλεχειριστηρίου τηλεόρασης

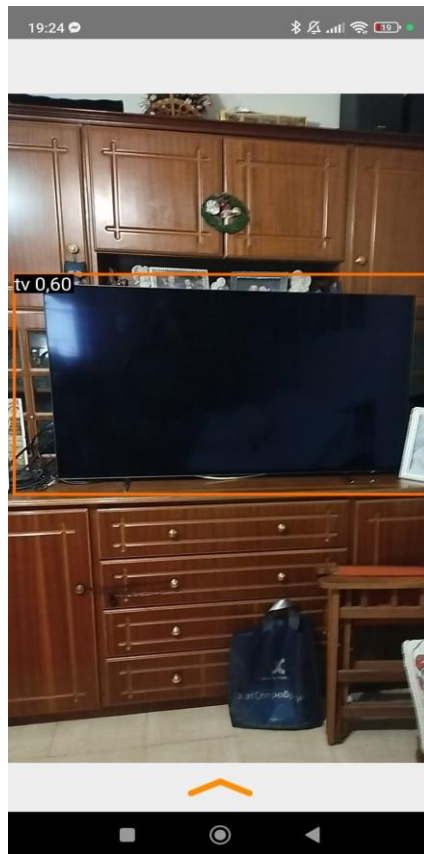


Εικόνα 2.3: Εντοπισμός καρέκλας



Εικόνα 2.4: Εντοπισμός ψυγείου

## Κεφάλαιο 2



Εικόνα 2.5: Εντοπισμός τηλεόρασης



Εικόνα 2.6: Εντοπισμός πιρουνιού και κουταλιού

## 2.5 Επίλογος

Η Τεχνητή Νοημοσύνη αναπτύσσεται με ολοένα και πιο γοργά βήματα, έχοντας ως στόχο την υλοποίηση συστημάτων που μιμούνται την ανθρώπινη σκέψη και δράση. Η μηχανική μάθηση και η βαθιά μάθηση, υποπεδία της ΤΝ, εστιάζουν στην ικανότητα των συστημάτων να "μαθαίνουν" από δεδομένα και να βελτιώνουν την απόδοσή τους με την πάροδο του χρόνου. Στην καρδιά της μηχανικής μάθησης βρίσκονται αλγόριθμοι που μπορούν να αναγνωρίζουν μοτίβα και να κάνουν προβλέψεις από δεδομένα. Η βαθιά μάθηση αποτελεί υποσύνολο της μηχανικής μάθησης που αξιοποιεί νευρωνικά δίκτυα, εμπνευσμένα από τον ανθρώπινο εγκέφαλο. Αυτά τα δίκτυα αποτελούνται από πολλαπλά στρώματα μονάδων επεξεργασίας, γνωστών ως νευρώνες. Κάθε νευρώνας λαμβάνει εισόδους από άλλους νευρώνες και παράγει μια έξοδο. Η βαθιά μάθηση έχει πετύχει εντυπωσιακά αποτελέσματα σε διάφορες εργασίες, όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας και η αναγνώριση ομιλίας. Η όραση υπολογιστών, ένα άλλο υποπεδίο της ΤΝ, δίνει έμφαση στην ικανότητα των συστημάτων να "αντιλαμβάνονται" και να ερμηνεύουν τον κόσμο γύρω τους μέσω εικόνων και βίντεο. Η όραση υπολογιστών εστιάζει στην ανάπτυξη συστημάτων που μπορούν να "αντιλαμβάνονται" και να κατανοούν τον κόσμο γύρω τους μέσω εικόνων και βίντεο. Χρησιμοποιεί αλγόριθμους για να εξάγει πληροφορίες από οπτικά δεδομένα, όπως σχήματα, αντικείμενα, κίνηση και σκηνές. Ωστόσο, η ανάπτυξη και η χρήση αυτών των τεχνολογιών έρχεται με ευθύνες. Η ηθική χρήση της ΤΝ και η προστασία της ιδιωτικής ζωής αποτελούν κρίσιμα ζητήματα που οφείλουμε να αντιμετωπίσουμε. Είναι σημαντικό να υιοθετήσουμε μια υπεύθυνη και ηθική προσέγγιση στην ανάπτυξη και χρήση της ΤΝ, λαμβάνοντας υπόψη όλες τις πτυχές και τις επιπτώσεις της.

## Κεφάλαιο 3ο: Απτική Αλληλεπίδραση

### 3.1 Εισαγωγή

Η απτική τεχνολογία, εμπνευσμένη από την αίσθηση της αφής, εισβάλλει στον κόσμο των υπολογιστών, φέρνοντας επανάσταση στον τρόπο που αλληλεπιδρούμε με πληροφορίες και ιδέες. Πλέον, αντί να περιοριζόμαστε σε οθόνες και πλήκτρα, μπορούμε να αγγίζουμε, να χειριζόμαστε και να αισθανόμαστε εικονικά αντικείμενα με τρόπο που προσομοιώνει ρεαλιστικά την αφή. Φανταστείτε να "πιάνετε" ένα εικονικό αγαλματίδιο ή να "χειρουργείτε" σε έναν εικονικό ασθενή - η απτική τεχνολογία το κάνει πραγματικότητα. Η απτική τεχνολογία συνδυάζει γνώσεις από τη βιομηχανική, την ψυχολογία, τη νευρολογία, τη μηχανική και την πληροφορική, εστιάζοντας στην αίσθηση της αφής και στην απτική ανάδραση. Μέσα από την απτική αίσθηση, οι χρήστες "αντιλαμβάνονται" τρισδιάστατα εικονικά αντικείμενα, νιώθοντας το σχήμα, το βάρος, την υφή και τη θερμοκρασία τους [8]. Η απτική τεχνολογία δεν περιορίζεται μόνο στην όραση. Επιτρέπει ειδικότερα και σε ανθρώπους με προβλήματα όρασης να "βλέπουν", να "εξερευνούν" και να "αισθάνονται" το περιβάλλον τους. Η ενεργή αφή, αμφίδρομη ροή πληροφοριών και ενέργειας, μας επιτρέπει να "αντιληφθούμε" εικονικά αντικείμενα αγγίζοντας τα, νιώθοντας δυνάμεις και αντιδράσεις. Το ανθρώπινο απτικό σύστημα αποτελείται από μηχανικά, αισθητηριακά, κινητικά και γνωστικά στοιχεία. Τα μηχανικά μέρη του σώματος ανταποκρίνονται στις εντολές του εγκεφάλου, ενώ οι αισθητηριακοί υποδοχείς του νευρικού συστήματος μεταφέρουν πληροφορίες από ερεθίσματα αφής στον εγκέφαλο. Ο εγκέφαλος επεξεργάζεται αυτές τις πληροφορίες, ενεργοποιώντας τους μύες ώστε να κινηθούν, ολοκληρώνοντας τον κύκλο της αλληλεπίδρασης. [8]

### 3.2 Απτική Τεχνολογία

Η απτική αίσθηση, η ικανότητα να αντιλαμβανόμαστε και να χειριζόμαστε αντικείμενα μέσω της αφής, αποτελεί αντικείμενο μελέτης εδώ και δεκαετίες.

- **Από τις αρχές του 20ου αιώνα:** Οι ψυχολόγοι χρησιμοποίησαν τον όρο "απτική" για να μελετήσουν την αλληλεπίδραση με απτά αντικείμενα.
- **Τέλη δεκαετίας του 1980:** Η ανάπτυξη της ρομποτικής και η εισαγωγή αισθητήρων που επέτρεψαν την τεχνητή αφή δημιούργησαν την ανάγκη για έναν όρο που να περιγράφει αυτόν τον νέο τομέα. Έτσι, ο όρος "απτική" επαναπροσδιορίστηκε.

Η σημασία της αφής στην καθημερινή μας ζωή είναι αναμφισβήτητη. Απλές κινήσεις όπως το δέσιμο των κορδονιών ή το άνοιγμα μιας βρύσης θα ήταν σχεδόν αδύνατες χωρίς την απαραίτητη απτική ανάδραση. Φανταστείτε να προσπαθήσετε να φάτε με αναισθησία στο σαγόνι σας, χωρίς αίσθηση της θέσης των μάγουλων σας. Στις σύγχρονες τεχνολογίες, οι μηχανικοί εστιάζουν στη βελτίωση της απόδοσης των χειριστηρίων σε απομακρυσμένες εφαρμογές προσομοίωσης [9]. Η απτική σχεδίαση έρχεται να καλύψει αυτό το κενό, επιτρέποντας στον χρήστη να επεκτείνει την απτική του ικανότητα πέρα από το εύρος των χεριών του. Το ανθρώπινο απτικό σύστημα αποτελείται από τα αισθητηριακά, κινητικά και γνωστικά στοιχεία του σώματος-εγκεφάλου. Όταν αγγίζουμε ένα αντικείμενο, ολόκληρο το σύστημα ενεργοποιείται. Οι νευρώνες που συνδέονται με τους μύς μας (κιναισθητική λειτουργία) και οι νευρώνες που βρίσκονται κάτω από το δέρμα του χεριού μας (δερματική λειτουργία) συνεργάζονται με τη γνωστική λειτουργία για να παρέχουν στο σύστημα ελέγχου κίνησης τις

απαραίτητες πληροφορίες για την ανάπτυξη λεπτών κινητικών δεξιοτήτων. Οι απτικές συσκευές αλληλεπιδρούν με το απτικό σύστημα καθώς οι δυνάμεις που ασκούνται στο δέρμα επηρεάζουν τους ενσωματωμένους απτικούς αισθητήρες μας. Η απτική τεχνολογία βρίσκει εφαρμογή σε διάφορους τομείς, όπως η ιατρική, η εκπαίδευση, η ψυχαγωγία, η βιομηχανία και η επιστημονική έρευνα. Χρησιμοποιείται για χειρουργικές προσομοιώσεις, εκπαίδευση σε επικίνδυνα επαγγέλματα, σχεδιασμό προϊόντων, εικονική πραγματικότητα και πολλά άλλα. Η απτική τεχνολογία ανοίγει νέες δυνατότητες για αλληλεπίδραση με ψηφιακά περιβάλλοντα, φέρνοντας την αίσθηση της αφής στον κόσμο των υπολογιστών [9]. Η ικανότητά μας να αλληλεπιδρούμε με τον κόσμο γύρω μας βασίζεται σε ένα σύνθετο δίκτυο αισθήσεων. Εκτός από τις γνωστές αισθήσεις όπως η όραση, η ακοή και η γεύση, το σώμα μας διαθέτει και άλλες εσωτερικές αισθήσεις που μας παρέχουν ζωτικές πληροφορίες για την κατάστασή του. Παρακάτω θα δούμε έναν πίνακα 3.1 [9] με κάποιους όρους, οι οποίοι χρησιμοποιούνται στην απτική τεχνολογία.

Πίνακας 3.1: Ορολογίες απτικής αλληλεπίδρασης [9]

ιδιοδεκτική αίσθηση	Μας επιτρέπει να γνωρίζουμε τη θέση και την κίνηση του σώματός μας στο χώρο, ακόμα και με κλειστά μάτια. Χάρη στην ιδιοδεκτική αίσθηση, μπορούμε να συντονίσουμε τις κινήσεις μας, να διατηρήσουμε την ισορροπία μας και να πιάσουμε αντικείμενα με ακρίβεια.
Αιθουσαία αίσθηση	Είναι υπεύθυνη για την αίσθηση της ισορροπίας και του προσανατολισμού. Χάρη στην αιθουσαία αίσθηση, μπορούμε να αντιληφθούμε την κίνηση του κεφαλιού μας, να διατηρήσουμε την ισορροπία μας και να αποφύγουμε την ζαλάδα.
Κιναισθητική αίσθηση	Μας επιτρέπει να νιώθουμε την τάση στους μυς και τις αρθρώσεις μας, καθώς και την κίνηση των μελών του σώματός μας. Χάρη στην κιναισθητική αίσθηση, μπορούμε να ελέγξουμε την δύναμη που ασκούμε, να σηκώσουμε αντικείμενα και να αλληλεπιδράσουμε με το περιβάλλον με ακρίβεια.
Δερματική αίσθηση	Είναι η αίσθηση που προέρχεται από το δέρμα μας και μας επιτρέπει να νιώθουμε την πίεση, τη θερμοκρασία και τον πόνο. Χάρη στην δερματική αίσθηση, μπορούμε να αποφύγουμε τραυματισμούς, να αντιληφθούμε την υφή των αντικειμένων και να βιώσουμε συναισθήματα αφής.
Απτική αίσθηση	Είναι η λεπτή αίσθηση αφής που μας επιτρέπει να διακρίνουμε τα σχήματα, τις υφές και τις λεπτομέρειες των αντικειμένων. Χάρη στην απτική αίσθηση, μπορούμε να διαβάσουμε Braille, να γράψουμε με στυλό και να νιώσουμε την απαλότητα ενός υφάσματος.

<p>Ανάδραση δύναμης</p>	<p>Είναι η αίσθηση της αντίθετης δύναμης που ασκείται σε ένα αντικείμενο. Χάρη στην ανάδραση δύναμης, μπορούμε να ελέγξουμε πόση δύναμη ασκείται, να χειριστούμε εύθραυστα αντικείμενα και να πιάσουμε αντικείμενα με ακρίβεια.</p>
-------------------------	---

Αυτές οι εσωτερικές αισθήσεις συνεργάζονται άψογα, δίνοντάς μας μια ολοκληρωμένη εικόνα για την κατάσταση του σώματός μας και το περιβάλλον γύρω μας. Χάρη σε αυτές, μπορούμε να κινούμαστε με αυτοπεποίθηση, να αλληλεπιδρούμε με τον κόσμο με ακρίβεια και να βιώνουμε πλούσιες αισθητηριακές εμπειρίες.

### 3.3 Απτική Ανατροφοδότηση

Η αίσθηση της αφής μπορεί να αναπαρασταθεί τεχνολογικά με δύο βασικούς τρόπους: τη δερματική και την κιναισθητική ανατροφοδότηση.

- **Κιναισθητική:** Αφορά την ικανότητά μας να αντιλαμβανόμαστε τις κινήσεις του σώματός μας. Πληροφορίες από αισθητήρες στις αρθρώσεις και τους μυς στέλνονται στον εγκέφαλο, επιτρέποντάς μας να ελέγξουμε και να αξιολογούμε τις κινήσεις μας. Η κιναισθητική αίσθηση είναι απαραίτητη για καθημερινές δραστηριότητες, όπως η εκτίμηση της δύναμης που ασκείται σε αντικείμενα. Ωστόσο, η αναπαράστασή της σε εικονικές πραγματικότητες μπορεί να είναι δύσκολη λόγω της έλλειψης πραγματικών αντικειμένων [10].
- **Δερματική:** Αναφέρεται σε όλα τα ερεθίσματα που λαμβάνουν οι αισθητήρες αφής στο δέρμα. Η αφή μας επιτρέπει να ανιχνεύουμε πίεση και κραδασμούς, με τα δάχτυλά μας να είναι ιδιαίτερα ευαίσθητα. Οι δονήσεις που προκαλούνται όταν κρατάμε ένα αντικείμενο μας βοηθούν να ρυθμίσουμε τη δύναμη που ασκούμε για να μην μας φύγει. Μελέτες έχουν δείξει ότι η ενσωμάτωση δερματικής ανατροφοδότησης σε εικονικές πραγματικότητες μπορεί να βελτιώσει σημαντικά την αίσθηση του βάρους των αντικειμένων, ακόμη και χωρίς κιναισθητική ανατροφοδότηση. Επιπλέον, η δερματική ανατροφοδότηση αποδεικνύεται χρήσιμη σε διάφορες εργασίες εντός εικονικών περιβαλλόντων, όπως η γραφή, η επίλυση γρίφων, οι ασκήσεις κίνησης των χεριών και η διάκριση καμπύλων επιφανειών [10].

Για μια πιο ρεαλιστική εμπειρία, μια ιδανική συσκευή απτικής ανατροφοδότησης θα πρέπει να συνδυάζει και τους δύο τύπους. Θα πρέπει να μπορεί να "κλειδώνει" τα χέρια και τα δάχτυλα κατά το κράτημα εικονικών αντικειμένων (κινησθητική) και να παρέχει δονήσεις ή άλλα παρόμοια ερεθίσματα στο δέρμα (δερματική). Εμείς όμως θα δούμε κυρίως τον δεύτερο τρόπο διότι αξιοποιούμε την δόνηση του κινητού τηλεφώνου για την απτική ανατροφοδότηση.

### 3.3.1 Δερματική Ανατροφοδότηση

Η απτική δόνηση (vibrotactile stimulation) αποτελεί την πιο διαδεδομένη μορφή δερματικής ανατροφοδότησης. Η οικονομική της φύση, η χαμηλή κατανάλωση ενέργειας και η διακριτική εμφάνιση επιτρέπουν την ενσωμάτωσή της σε διάφορα σημεία του χεριού. Συσκευές καθημερινής χρήσης, όπως τα smartphones και οι υπολογιστές, αξιοποιούν τις δονήσεις για ειδοποιήσεις. Επιπλέον, πολλά εμπορικά διαθέσιμα συστήματα εικονικής πραγματικότητας, όπως τα Oculus Quest και HTC Vive, χρησιμοποιούν την απτική δόνηση ως κύρια μορφή δερματικής ανατροφοδότησης [10]. Η τεχνολογία "vibrotactile phantom stimulation" αξιοποιεί σειρές μαγνητικών καρφίδων που παράγουν δονήσεις για να δημιουργήσει την ψευδαίσθηση ολίσθησης και βάρους των αντικειμένων σε εικονικά περιβάλλοντα. Λαμβάνοντας υπόψη ότι η ανθρώπινη αναγνώριση υφής σχετίζεται με μικροδονήσεις σε εύρος 1 Hz έως 500 Hz, η μίμηση αυτών των μικροδονήσεων μέσω μικρών ενεργοποιητών αποτελεί μια πιθανή λύση για την αναπαράσταση κινήσεων δακτύλων και απτικής αίσθησης. Ωστόσο, η απτική δόνηση, παρότι απλή και εύκολη στην εφαρμογή, περιορίζεται στους τύπους ερεθισμάτων που μπορεί να παράγει. Για το λόγο αυτό, συνδυάζεται συχνά με άλλα συστήματα ανατροφοδότησης [10].

### 3.4 Τομείς Απτικής Αλληλεπίδρασης

Η απτική τεχνολογία, η οποία εστιάζει στην παροχή αισθητηριακής ανατροφοδότησης μέσω της αφής, έχει καταστεί αντικείμενο αυξανόμενου ενδιαφέροντος τα τελευταία χρόνια. Η ικανότητά της να προσομοιάνει ρεαλιστικές αισθήσεις αφής ανοίγει νέες δυνατότητες σε διάφορους τομείς, όπως η πρόσβαση σε πληροφορίες, η εκπαίδευση, η ιατρική περίθαλψη και οι τέχνες.

#### Πρόσβαση σε Πληροφορίες για Άτομα με Απώλεια Όρασης

Η ενσωμάτωση οθόνης αφής σε συσκευές απτικής απεικόνισης αποτελεί σημαντική εξέλιξη για την παροχή πρόσβασης σε πληροφορίες σε άτομα με οπτική αναπηρία. Χρησιμοποιώντας την αφή, τα άτομα αυτά μπορούν να αλληλεπιδρούν με ψηφιακό περιεχόμενο, όπως μενού, εικονίδια και γραφικά στοιχεία, τα οποία γίνονται αντιληπτά μέσω της απτικής ανατροφοδότησης. Η δυνατότητα αυτή ενισχύει την ανεξαρτησία και την αυτονομία των ατόμων με τυφλότητα, επιτρέποντάς τους να αξιοποιούν στο έπακρο τα οφέλη της ψηφιακής τεχνολογίας. [8]

#### Εφαρμογές στον Τομέα των Αυτοκινήτων

Η αυξανόμενη ζήτηση για ξεχωριστούς ελέγχους κλιματισμού σε σύγχρονα αυτοκίνητα, τόσο για τον οδηγό και τον συνοδηγό όσο και για τους πίσω επιβάτες, έρχεται αντιμέτωπη με την πρόκληση της πολυπλοκότητας των σύγχρονων συστημάτων κλιματισμού. Η απτική τεχνολογία μπορεί να διαδραματίσει σημαντικό ρόλο στην αντιμετώπιση αυτής της πρόκλησης, παρέχοντας στους οδηγούς εύχρηστη και αξιόπιστη ανατροφοδότηση αφής. Η άμεση και ακριβής πληροφόρηση που προσφέρει η απτική αλληλεπίδραση μειώνει την ανάγκη για οπτική εστίαση στα χειριστήρια, βελτιώνοντας παράλληλα την οδική ασφάλεια. [8]

#### Εικονική Εκπαίδευση

Η απτική τεχνολογία έχει τη δυνατότητα να μεταμορφώσει το εκπαιδευτικό τοπίο, προσφέροντας νέες δυνατότητες για την αφομοίωση γνώσεων και την ανάπτυξη δεξιοτήτων. Η ένταξη απτικών εργαλείων στην εκπαιδευτική διαδικασία μπορεί να ωφελήσει μαθητές με

διαφορετικά στυλ μάθησης, ιδιαίτερα εκείνους που μαθαίνουν καλύτερα μέσω της κινησθητικής νόησης. Η ρεαλιστική αίσθηση αφής που προσφέρει η απτική τεχνολογία επιτρέπει στους μαθητές να εξερευνήσουν και να κατανοήσουν έννοιες με πιο ενεργό και βιωματικό τρόπο, ενισχύοντας παράλληλα την κριτική σκέψη και την επίλυση προβλημάτων. [8]

#### **Ιατρική**

Η απτική τεχνολογία έχει βρει εφαρμογή σε διάφορους τομείς της ιατρικής, συμβάλλοντας στην εκπαίδευση και την εξάσκηση ιατρών και χειρουργών. Η χρήση απτικών προσομοιωτών επιτρέπει στους ιατρούς να εκπαιδευτούν σε ελάχιστα επεμβατικές επεμβάσεις, όπως η λαπαροσκοπία, και σε τηλεχειρουργικές επεμβάσεις, εξασκώντας τις απαραίτητες δεξιότητες με ασφάλεια και υψηλή ακρίβεια. Επιπλέον, η απτική ρομποτική χρησιμοποιείται στην αποκατάσταση ασθενών με κινητικά προβλήματα, βοηθώντας στην αποκατάσταση της κινητικότητας και του συντονισμού. Η ρεαλιστική αίσθηση αφής που προσφέρει η απτική τεχνολογία συμβάλλει στην καλύτερη κατανόηση των ανατομικών δομών και στην ακριβή εκτέλεση χειρουργικών επεμβάσεων, βελτιώνοντας παράλληλα την ποιότητα ζωής των ασθενών. [8]

#### **Τέχνες και Σχεδιασμός**

Η απτική τεχνολογία έχει ανοίξει νέες δυνατότητες στον τομέα των τεχνών και του σχεδιασμού. Η ικανότητά της να προσομοιώνει ρεαλιστικά υλικά και υφές επιτρέπει στους καλλιτέχνες και τους σχεδιαστές να δημιουργούν εικονικά γλυπτά και μοντέλα με πρωτοφανή ακρίβεια και λεπτομέρεια. Η απτική ανατροφοδότηση συμβάλλει στην ανάπτυξη νέων μορφών καλλιτεχνικής έκφρασης και στην υλοποίηση καινοτόμων σχεδιαστικών λύσεων. Η απτική τεχνολογία έχει τη δυνατότητα να μεταμορφώσει τον τρόπο με τον οποίο δημιουργούμε και αλληλεπιδρούμε με την τέχνη, προσφέροντας μια πιο ολοκληρωμένη και αισθητηριακή εμπειρία. [8]

### 3.5 Επίλογος

Η απτική τεχνολογία, εμπνευσμένη από την αίσθηση της αφής, εισβάλλει στον κόσμο των υπολογιστών, φέρνοντας επανάσταση στον τρόπο που αλληλεπιδρούμε με πληροφορίες και ιδέες. Η ικανότητά της να προσομοιώνει ρεαλιστικά εικονικά αντικείμενα, αγγίγματος, χειρισμού και αίσθησης ανοίγει νέες δυνατότητες σε διάφορους τομείς, όπως η πρόσβαση σε πληροφορίες, η εκπαίδευση, η ιατρική περίθαλψη, οι τέχνες και ο σχεδιασμός. Η απτική τεχνολογία συνδυάζει γνώσεις από διάφορες επιστημονικές περιοχές, όπως η μηχανική, η ψυχολογία, η νευρολογία, η πληροφορική και η βιομηχανική, με στόχο την ανάπτυξη συστημάτων που μιμούνται την ανθρώπινη αφή. Η δερματική και η κιναισθητική ανατροφοδότηση, δύο βασικοί τύποι απτικής ανατροφοδότησης, επιτρέπουν στους χρήστες να "αντιλαμβάνονται" τρισδιάστατα εικονικά αντικείμενα, νιώθοντας το σχήμα, το βάρος, την υφή και τη θερμοκρασία τους. Η απτική τεχνολογία έχει τη δυνατότητα να βελτιώσει την ποιότητα ζωής ατόμων με οπτική αναπηρία, παρέχοντάς τους πρόσβαση σε πληροφορίες και δυνατότητες που προηγουμένως ήταν περιορισμένες. Επιπλέον, η ρεαλιστική αίσθηση αφής που προσφέρει η απτική τεχνολογία μπορεί να βελτιώσει την εκπαίδευση, την ιατρική περίθαλψη, τον σχεδιασμό και τις τέχνες, προσφέροντας νέες δυνατότητες για αλληλεπίδραση και δημιουργία. Καθώς η απτική τεχνολογία συνεχίζει να εξελίσσεται, μπορούμε να αναμένουμε ακόμη πιο ρεαλιστικές και συναρπαστικές εμπειρίες αλληλεπίδρασης με ψηφιακά περιβάλλοντα. Η απτική τεχνολογία έχει τη δυνατότητα να μεταμορφώσει τον τρόπο που ζούμε, εργαζόμαστε και αλληλεπιδρούμε με τον κόσμο γύρω μας.

## Κεφάλαιο 4ο: Υποβοηθητική Τεχνολογία

### 4.1 Εισαγωγή

Η υποβοηθητική τεχνολογία αποτελεί ένα φάρο ελπίδας για άτομα με προβλήματα όρασης, χαρίζοντάς τους εργαλεία και λύσεις που τους επιτρέπουν να αγγίξουν την ανεξαρτησία και την ενεργή συμμετοχή στην κοινωνία. Η ΥΤ αγγίζει ένα ευρύ φάσμα τεχνολογιών, από απλές συσκευές καθημερινής χρήσης έως προηγμένα λογισμικά και συστήματα. Στο προηγούμενο κεφάλαιο είδαμε τι είναι η απτική τεχνολογία και η απτική ανατροφοδότηση. Σε αυτό το κεφάλαιο θα δούμε πως αυτή η τεχνολογία σε συνδυασμό με την τεχνητή νοημοσύνη μπορεί να κάνει καλύτερη και πιο εύκολη την καθημερινότητα σε πολλούς συνανθρώπους μας, οι οποίοι αντιμετωπίζουν προβλήματα όρασης. Όμως πρώτα θα δούμε κάποια καινοτόμα επιτεύγματα πάνω σε αυτόν τον τομέα.

### 4.2 Όραση μέσω της αφής

Οι τεχνολογίες υποστήριξης, που βοηθούν άτομα τα οποία αντιμετωπίζουν προβλήματα όρασης, έχουν διαδοθεί σε μεγάλο βαθμό και τα οφέλη τους είναι αδιαμφισβήτητα. Η εξέλιξή τους με την πάροδο του χρόνου είναι εντυπωσιακή, από μια απλή γραφομηχανή του 19ου αιώνα, η οποία βοηθούσε τους τυφλούς να γράφουν με ευκρίνεια έως εφαρμογές κινητών τηλεφώνων που επιτρέπουν σε άτομα με μειωμένη όραση να "βλέπουν" και να κατανοούν το περιβάλλον τους [11]. Αυτές οι τεχνολογίες φέρνουν αισθητές βελτιώσεις στην ποιότητα ζωής των ατόμων με προβλήματα όρασης, αυξάνοντας την αυτονομία και την ασφάλειά τους. Παράλληλα, τους ενθαρρύνουν να ξεπεράσουν τα όρια του οικείου περιβάλλοντος και να αλληλεπιδράσουν κοινωνικά, μειώνοντας ταυτόχρονα τον φόβο για κοινωνική απομόνωση. Πλέον, πέρα από τις παραδοσιακές μεθόδους, όπως η γραφή Braille και το λευκό μαστούνι, οι σύγχρονες υποστηρικτικές τεχνολογίες κινητών συσκευών είναι πιο διακριτικές και αξιοποιούν ευρεία γκάμα φορητών υπολογιστικών συσκευών, όπως τα διαδεδομένα κινητά τηλέφωνα [11].

#### 4.2.1 Αξιοποίηση Κινητών Συσκευών

Οι φορητές συσκευές είναι μια πολλά υποσχόμενη κατηγορία ανάπτυξης υποστηρικτικής τεχνολογίας. Οι νέες δυνατότητες τις οποίες προσφέρουν στον τομέα των τεχνολογιών πληροφορίας, τις κάνουν έφορο έδαφος για καινοτόμες εφαρμογές οι οποίες δύνανται να αλλάξουν τις ζωές πολλών συνανθρώπων μας. Ένα ουσιαστικό πλεονέκτημα των κινητών συσκευών ως πλατφόρμας για υποστηρικτικές τεχνολογίες είναι η διακριτικότητά τους. Συσκευές με διακριτικό σχεδιασμό ή εφαρμογές ενσωματωμένες σε ευρέως διαδεδομένες συσκευές, όπως τα κινητά τηλέφωνα, μπορούν να συμβάλουν στη μείωση του στιγματισμού που μπορεί να βιώσουν οι χρήστες με αναπηρίες [11]. Η διάδοση των κινητών τηλεφώνων, ιδιαίτερα των smartphones, έχει εγκαινιάσει μια νέα εποχή διασυνδεσιμότητας, παρέχοντας στους χρήστες πρόσβαση σχεδόν παντού και κάθε ώρα και στιγμή. Οι συσκευές αυτές έχουν ξεπεράσει τον αρχικό τους ρόλο ως τηλέφωνα και προσφέρουν πλέον ένα ευρύ φάσμα λειτουργιών σε μια συμπαγή και φορητή συσκευή. Ολοένα και περισσότερα άτομα με προβλήματα όρασης αξιοποιούν τα smartphones στις καθημερινές τους δραστηριότητες. [11] Έτσι και η εφαρμογή η οποία αναπτύχθηκε για την παρούσα εργασία αξιοποιεί το κινητό ως μέσο για να προσφέρει στα άτομα με προβλήματα όρασης ένα καλύτερο μέλλον, με την ελπίδα και αυτά τα άτομα να ενταχθούν όσο το δυνατόν πιο ομαλά και χωρίς διακρίσεις.

#### 4.2.1.1 Δυσκολίες στην πλοήγηση

Για τους συνανθρώπους μας οι οποίοι αντιμετωπίζουν στην καθημερινότητα τους προβλήματα όρασης η πλοήγηση δεν είναι τόσο απλή όσο για τους υπόλοιπους ανθρώπους. Οι δυσκολίες αυτές της πλοήγησης παρόλο που υπάρχουν ήδη συστήματα τηλεβοήθειας με την καθοδήγηση ενός ατόμου, το οποίο δεν αντιμετωπίζει παρόμοια προβλήματα, μέσω βίντεο και οδηγιών, περιορίζει την αυτονομία των χρηστών. Εναλλακτικά, οι εφαρμογές πλοήγησης σε κινητές συσκευές προσφέρουν πιο ανεξάρτητη διακίνηση. Παράδειγμα αποτελεί το Voice Maps, που παρέχει οδηγίες πλοήγησης και παρακολουθεί τη θέση του χρήστη, ενημερώνοντάς τον σε περίπτωση απόκλισης. Άλλες εφαρμογές, όπως το RAMPE, εστιάζουν στην μετακίνηση με δημόσια μέσα, παρέχοντας πληροφορίες για στάσεις και διαδρομές σε πραγματικό χρόνο [11]. Ένα ακόμα ζήτημα στην πλοήγηση αυτών των ανθρώπων είναι τα εμπόδια τα οποία μπορεί να συναντήσουν και ειδικότερα στην χώρα μας όπου δεν είναι λίγες οι φορές οι οποίες κάποιος θα έχει σταθμεύσει σε ράμπα ή σε οδηγό για άτομα με οπτικές δυσκολίες, έτσι η πλοήγηση τους καθιστάτε όχι μόνο πιο δύσκολη αλλά και επικίνδυνη. Βέβαια υπάρχει η πιο διαδεδομένη και συμβατική λύση σε αυτό το ζήτημα η οποία είναι το λευκό μπαστούνι αλλά απαιτεί εκπαίδευση και συνεχή σάρωση του περιβάλλοντος ώστε να είναι αποτελεσματικό. Για να ξεπεραστούν αυτοί οι περιορισμοί, αναπτύχθηκαν συστήματα όπως το SmartVision, που συνδυάζει το μπαστούνι με αισθητήρες για την ανίχνευση εμποδίων μπροστά από τον χρήστη. Άλλα συστήματα, όπως το GuideCane, διαθέτουν τροχούς και αισθητήρες για αυτόνομη αποφυγή εμποδίων. Όμως, συστήματα όπως το NavBelt και το GuideCane είναι ογκώδη και μπορεί να στιγματίσουν τους χρήστες. Εναλλακτικές λύσεις, όπως το ειδικό smartphone του Peng ή η συσκευή haptic feedback του Amemiya, είναι πιο διακριτές. Τέλος, τα έξυπνα γυαλιά αποτελούν μια μη επεμβατική λύση που χρησιμοποιεί κάμερες και απτική ανατροφοδότηση για να ενημερώσει τους χρήστες για εμπόδια [11].

### 4.3 Πρωτότυπα και εφαρμογές του μαθήματος Απτικές Διεπαφές

Στο μάθημα των απτικών διεπαφών, εξερευνήσαμε τον κόσμο της αφής και τις δυνατότητες που προσφέρει για άτομα με προβλήματα όρασης. Μέσα από δημιουργικές ιδέες, αναπτύξαμε πρωτότυπα και εφαρμογές που στοχεύουν στην παροχή αυτονομίας, πρόσβασης σε πληροφορίες και βελτίωσης της ποιότητας ζωής των συνανθρώπων μας που στην καθημερινότητα τους αντιμετωπίζουν προβλήματα όρασης.

#### 4.3.1 Αναπαράσταση Ψηφιακών Επιφανειών και Αντικειμένων:

Χρησιμοποιώντας την απτική ανατροφοδότηση, είδαμε πως μπορούμε να αναπαραστήσουμε ψηφιακές επιφάνειες και αντικείμενα, "εκφράζοντάς τα" μέσω των μοτίβων δόνησης. Η αίσθηση της αφής μεταμορφώνεται σε όραση, καθώς διαφορετικά μοτίβα δόνησης αντιστοιχούν σε διαφορετικές επιφάνειες ή αντικείμενα. Με αυτόν τον τρόπο ο χρήστης μπορεί να λάβει πληροφορίες για μία ψηφιακή εικόνα ή ένα οποιοδήποτε ψηφιακό τρισδιάστατο αντικείμενο, ζωντανεύοντας το μπροστά του. Με αυτήν την καινοτομία ανοίγεται πλέον στον χρήστη ένας νέος ορίζοντας με νέες δυνατότητες να νιώσει τον ψηφιακό κόσμο όπως ποτέ άλλοτε.

#### 4.3.2 Μοτίβα Δόνησης και Χρώματα:

Η δημιουργική χρήση μοτίβων δόνησης άνοιξε νέους δρόμους στην αναπαράσταση χρωμάτων. Μέσω μιας κλίμακας, η αύξηση ή η μείωση των κενών παύσης και του χρόνου δόνησης "μεταφράζει" τα χρώματα, προσφέροντας μια ακουστική ερμηνεία της οπτικής πληροφορίας. Ειδικότερα στο μάθημα ο

συγγραφέας της παρούσης εργασίας ανέπτυξε μία εφαρμογή «παιχνίδι», το οποίο μπορούσε να το χειριστεί και ένα άτομο με προβλήματα όρασης, καθώς υπήρχε φωνητική περιγραφή των χρωμάτων και το κάθε χρώμα είχε το δικό του μοτίβο δόνησης. Στην εκκίνηση το παιχνίδι αναπαρήγαγε μία τυχαία δόνηση, με σκοπό ο χρήστης να ανακαλύψει πιο χρώμα ήταν βάση αυτής της δόνησης, εφόσον όμως πρώτα μελετούσε την κάθε δόνηση του εκάστοτε χρώματος.

### **4.3.3 Ανιχνευτής Εμποδίων:**

Συνδυάζοντας έναν υπερηχητικό αισθητήρα και ένα vibrator motor, κατασκευάστηκε ένα χρήσιμο ανιχνευτή εμποδίων. Ο υπερηχητικός αισθητήρας εκπέμπει σήματα προς τα εμπόδια, ενώ το vibration motor δονείται με ένταση ανάλογα με την απόσταση. Ο χρήστης, λαμβάνοντας την απτική αίσθηση, ενημερώνεται για την ύπαρξη και την απόσταση του εμποδίου, αποφεύγοντας έγκαιρα τυχόν κινδύνους. Έτσι λοιπόν με αυτό το χρήσιμο πρωτότυπο μπορούν να λυθούν μερικά από τα προβλήματα πλοήγησης που αναφέραμε πιο πάνω.

### **4.3.4 Δόνηση ως μέσο προσανατολισμού**

Στο πλαίσιο της εξερεύνησης εναλλακτικών μεθόδων προσανατολισμού για άτομα με προβλήματα όρασης, αναπτύχθηκε μία καινοτόμος εφαρμογή που αξιοποιεί την δόνηση ως αποτρεπτικό μέσο. Η εφαρμογή αυτή υλοποιήθηκε σε μορφή λαβύρινθου, με στόχο την ενίσχυση της αίσθησης αφής και την παροχή ακουστικής ανατροφοδότησης για την αποφυγή εμποδίων. Ο χρήστης, κινούμενος εντός του λαβύρινθου, ερχόταν σε επαφή με τους τοίχους του. Η επαφή ενεργοποιούσε ένα σύστημα δόνησης, το οποίο εκπέμπει αισθητές δονήσεις ως ένδειξη εμποδίου. Η ένταση και η συχνότητα των δονήσεων ενίσχυαν την αίσθηση της αφής, ενημερώνοντας τον χρήστη για την εγγύτητα του εμποδίου. Μόλις ο χρήστης απομακρυνόταν από τον τοίχο, οι δονήσεις σταματούσαν, παρέχοντας αίσθηση ελευθερίας διαδρομής.

#### 4.4 Επίλογος

Η απτική τεχνολογία, σε συνδυασμό με την τεχνητή νοημοσύνη, φέρνει επανάσταση στον τρόπο που άτομα με προβλήματα όρασης αλληλεπιδρούν με τον κόσμο γύρω τους. Ανοίγει νέες δυνατότητες για την παροχή πρόσβασης σε πληροφορίες, αυτονομία και βελτίωση της ποιότητας ζωής. Στο πλαίσιο του μαθήματος "Απτικές Διεπαφές", εξερευνήθηκαν πρωτότυπες εφαρμογές που αξιοποιούν την απτική ανατροφοδότηση για την αναπαράσταση ψηφιακών επιφανειών και αντικειμένων, την "ερμηνεία" χρωμάτων, την ανίχνευση εμποδίων και τον προσανατολισμό. Η απτική αναπαράσταση ψηφιακών πληροφοριών μεταμορφώνει την αφή σε όραση, προσφέροντας μια νέα οπτική στους χρήστες. Η χρήση μοτίβων δόνησης για την αναπαράσταση χρωμάτων ανοίγει νέους δρόμους στην ερμηνεία οπτικών ερεθισμάτων, ενώ η δημιουργία **ανιχνευτών εμποδίων** με βάση την απτική ανατροφοδότηση ενισχύει την ασφάλεια και την αυτονομία. Τέλος, η αξιοποίηση της δόνησης ως μέσο προσανατολισμού φέρνει νέες δυνατότητες στην πλοήγηση και την αποφυγή εμποδίων. Η απτική τεχνολογία, σε συνδυασμό με την τεχνητή νοημοσύνη, έχει τη δυνατότητα να βελτιώσει δραματικά την ποιότητα ζωής ατόμων με προβλήματα όρασης. Η ανάπτυξη νέων, εξελιγμένων συστημάτων και εφαρμογών, με έμφαση στην ευχρηστία, την διακριτικότητα και την προσιτή τιμή, μπορεί να φέρει την απτική τεχνολογία σε ευρεία χρήση, ενδυναμώνοντας τα άτομα με προβλήματα όρασης και προσφέροντάς τους ίσες ευκαιρίες συμμετοχής στην κοινωνία.

## Κεφάλαιο 5ο: Παρεμφερείς Εφαρμογές

### 5.1 Εισαγωγή

Η ανάπτυξη τεχνολογιών υποβοήθησης για άτομα με προβλήματα όρασης αποτελεί ένα πεδίο αδιάκοπης έρευνας και καινοτομίας. Η ολοένα αυξανόμενη συχνότητα και ποικιλία εφαρμογών σε αυτόν τον τομέα μαρτυρά την αυξανόμενη δέσμευση για την ενδυνάμωση και την ενίσχυση της αυτονομίας των ατόμων με οπτικές δυσκολίες. Στο πλαίσιο της παρούσας εργασίας, η ανίχνευση καθημερινών αντικειμένων λαμβάνει κεντρικό ρόλο, στοχεύοντας στην απεξάρτηση των χρηστών από την εξωτερική βοήθεια και την ενίσχυση της αυτονομίας τους. Η εφαρμογή που αναπτύχθηκε φιλοδοξεί να λειτουργήσει ως ο ψηφιακός βοηθός τους, παρέχοντας αξιόπιστες πληροφορίες και υποστήριξη στην καθημερινότητά τους. Στην παρούσα ενότητα θα εξερευνήσουμε ορισμένες παρεμφερείς εφαρμογές με της εργασίας.

### 5.2 Be My Eyes

Η εφαρμογή "Be My Eyes" για κινητές συσκευές Android αποτελεί ένα καινοτόμο σύστημα υποβοήθησης ατόμων με οπτικές δυσκολίες. Χρησιμοποιώντας τεχνολογία αναγνώρισης φωνής, η εφαρμογή επιτρέπει στους χρήστες να εκτελούν βασικές λειτουργίες του κινητού τους τηλεφώνου μέσω φωνητικών εντολών, χωρίς την ανάγκη χειρισμού του πληκτρολογίου. Μετά την ξεκλείδωση της συσκευής, η εφαρμογή "Be My Eyes" ενεργοποιείται αυτόματα, εξαλείφοντας την ανάγκη για πρόσθετες εντολές. Το σύστημα αναγνωρίζει τις φωνητικές εντολές του χρήστη και εκτελεί τις αντίστοιχες λειτουργίες. Για την επεξεργασία των εντολών, η εφαρμογή μετατρέπει αρχικά την ομιλία σε κείμενο και στη συνέχεια παρέχει φωνητική ανατροφοδότηση για την επιβεβαίωση της ενέργειας. [12] Η εφαρμογή "Be My Eyes" διευκολύνει σημαντικά την καθημερινή ζωή των ατόμων με οπτικές δυσκολίες, τα οποία παλαιότερα έπρεπε να απομνημονεύουν τη διάταξη των πλήκτρων στο κινητό τους τηλέφωνο για την εκτέλεση βασικών λειτουργιών όπως κλήσεις, αποστολή μηνυμάτων και διαχείριση επαφών (προσθήκη, προβολή, διαγραφή). Χάρη στην αναγνώριση φωνής, οι χρήστες μπορούν πλέον να αλληλεπιδρούν με το κινητό τους τηλέφωνο με άνεση και ευκολία. Προηγουμένως, τα άτομα με τύφλωση βασίζονταν σε άτομα με όραση για την πραγματοποίηση κλήσεων και την πρόσβαση σε άλλες λειτουργίες του κινητού τους τηλεφώνου. Η εφαρμογή "Be My Eyes" γεφυρώνει αυτό το χάσμα, επιτρέποντας την αυτονομία και την ανεξαρτησία των χρηστών στην καθημερινή τους επικοινωνία [12]. Η εφαρμογή ξεπερνά τα όρια μιας απλής εφαρμογής αναγνώρισης φωνής, προσφέροντας ένα ολοκληρωμένο σύστημα υποστήριξης για άτομα με οπτικές δυσκολίες. Εκτός από τις βασικές λειτουργίες, η εφαρμογή διαθέτει δύο σημαντικές δυνατότητες που ενισχύουν την αυτονομία και την ασφάλεια των χρηστών:

#### 1. Δίκτυο Εθελοντών:

Η εφαρμογή "Be My Eyes" έχει δημιουργήσει ένα παγκόσμιο δίκτυο εθελοντών, διαθέσιμων ανά πάσα ώρα και στιγμή, έτσι ώστε να παρέχουν υποστήριξη σε πραγματικό χρόνο. Μέσω τηλεδιάσκεψης, οι χρήστες μπορούν να συνδεθούν με εθελοντές και να λάβουν βοήθεια σε οποιαδήποτε κατάσταση που απαιτεί ανθρώπινη παρέμβαση [13]. Αυτή η δυνατότητα αποδεικνύεται ιδιαίτερα χρήσιμη σε περιπτώσεις όπου η απλή φωνητική ανατροφοδότηση δεν επαρκεί, όπως:

- **Πλοήγηση σε άγνωστα ή περίπλοκα περιβάλλοντα:** Οι εθελοντές μπορούν να παρέχουν οδηγίες και περιγραφές του περιβάλλοντος, βοηθώντας τους χρήστες να κινηθούν με αυτοπεποίθηση.

- **Εκτέλεση καθημερινών εργασιών:** Οι εθελοντές μπορούν να βοηθήσουν στην ανάγνωση εγγράφων, στην οργάνωση ραντεβού ή στην πραγματοποίηση αγορών.
- **Αντιμέτωπιση έκτακτων περιστατικών:** Σε περίπτωση έκτακτης ανάγκης, οι εθελοντές μπορούν να παρέχουν συναισθηματική υποστήριξη και να βοηθήσουν στην επικοινωνία με τις αρμόδιες υπηρεσίες. [13]

## 2. Τεχνητή Νοημοσύνη:

Χάρη στην καινοτομία της τεχνητής νοημοσύνης, η εφαρμογή "Be My Eyes" προσφέρει μια νέα δυνατότητα: την αναγνώριση του περιβάλλοντος. Με τη χρήση αλγορίθμων μηχανικής μάθησης και τεχνολογίας όρασης υπολογιστή, η εφαρμογή μπορεί να:

- **Αναγνωρίζει αντικείμενα και πρόσωπα:** Η εφαρμογή μπορεί να περιγράψει στον χρήστη τα αντικείμενα που βρίσκονται στο οπτικό του πεδίο, καθώς και να αναγνωρίσει πρόσωπα και να δώσει πληροφορίες για την ταυτότητά τους.
- **Διαβάζει κείμενο:** Η εφαρμογή μπορεί να διαβάσει κείμενο από πινακίδες, ετικέτες ή έγγραφα, παρέχοντας στους χρήστες πρόσβαση σε πληροφορίες που προηγουμένως ήταν απροσπέλαστες.
- **Περιγράφει σκηνές:** Η εφαρμογή μπορεί να περιγράψει στον χρήστη το περιβάλλον γύρω του, δίνοντάς του μια πιο ολοκληρωμένη εικόνα του χώρου. [13]

Η ενσωμάτωση της τεχνητής νοημοσύνης στην εφαρμογή "Be My Eyes" αποτελεί ένα σημαντικό βήμα προς την κατεύθυνση της παροχής αυτονομίας και ανεξαρτησίας για άτομα με οπτικές δυσκολίες.

## 5.3 Seeing AI

Η εφαρμογή Seeing AI αποτελεί ένα ακόμα καινοτόμο εργαλείο αξιοποίησης της τεχνητής νοημοσύνης για την περιγραφή του περιβάλλοντος σε άτομα με οπτικές δυσκολίες.

Χρησιμοποιώντας την κάμερα του κινητού και αλγόριθμους TN, η εφαρμογή Seeing AI "βλέπει" τον κόσμο και μεταφράζει τις οπτικές πληροφορίες σε σαφείς και περιεκτικές λεκτικές περιγραφές [14]. Η εφαρμογή προορίζεται μόνο το λειτουργικό σύστημα IOS της εταιρείας Apple.

### Λειτουργίες:

- **Περιγραφή Σκηνών:** Η εφαρμογή Seeing AI μπορεί να περιγράψει λεπτομερώς το περιβάλλον στο οποίο βρίσκεται ο χρήστης. Αυτό περιλαμβάνει την αναγνώριση αντικειμένων, προσώπων, κειμένου και τοπίου, καθώς και την περιγραφή της διάταξης και της ατμόσφαιρας του χώρου.
- **Ανάγνωση Κειμένου:** Η εφαρμογή Seeing AI μπορεί να "διαβάσει" κείμενο από διάφορες πηγές, όπως πινακίδες, ετικέτες, βιβλία και έγγραφα. Αυτό επιτρέπει στους χρήστες να έχουν πρόσβαση σε πληροφορίες που προηγουμένως ήταν απροσπέλαστες.
- **Αναγνώριση Προσώπων:** Η εφαρμογή Seeing AI μπορεί να αναγνωρίσει πρόσωπα και να δώσει πληροφορίες για την ταυτότητά τους, όπως το όνομα ή τη σχέση τους με τον χρήστη.
- **Αναγνώριση Αντικειμένων:** Η εφαρμογή Seeing AI μπορεί να αναγνωρίσει και να περιγράψει αντικείμενα που βρίσκονται στο οπτικό πεδίο του χρήστη.

- **Εντοπισμός Πληροφοριών:** Η εφαρμογή Seeing AI μπορεί να εντοπίσει και να "διαβάσει" barcodes, QR codes και άλλα οπτικά σήματα, παρέχοντας στους χρήστες πρόσβαση σε επιπλέον πληροφορίες [14].

## 5.4 BlindNavi

Η εφαρμογή BlindNavi αποτελεί ένα πρωτότυπο σύστημα πλοήγησης που σχεδιάστηκε με στόχο την ενίσχυση της αυτονομίας και της ασφάλειας ατόμων με οπτικές δυσκολίες. Η εφαρμογή βασίζεται σε τρία βασικά στοιχεία:

### 1. Απλό σχεδιασμό με αναζήτηση τριών βημάτων :

Η BlindNavi υιοθετεί ένα απλό και εύχρηστο σχεδιασμό με αναζήτηση τριών βημάτων, επιτρέποντας στους χρήστες να εντοπίζουν εύκολα τον προορισμό τους. Η δομή της εφαρμογής είναι γραμμική και κατανοητή, μειώνοντας το γνωστικό φορτίο και την πιθανότητα σύγχυσης [15].

### 2. Πολυαισθητηριακή Ανατροφοδότηση:

Η εφαρμογή BlindNavi αξιοποιεί πολυαισθητηρική ανατροφοδότηση, συνδυάζοντας φωνητικές οδηγίες με μη λεκτικές πληροφορίες, όπως δονήσεις και αισθητηριακές ενδείξεις. Αυτή η προσέγγιση προσφέρει στους χρήστες μια ολοκληρωμένη αίσθηση του περιβάλλοντος, ενισχύοντας την επίγνωσή τους και την ικανότητά τους να λαμβάνουν σωστές αποφάσεις πλοήγησης [15].

### 3. Τεχνολογία Μίκρο-Τοποθεσίας:

Η εφαρμογή BlindNavi ενσωματώνει τεχνολογία μίκρο-τοποθεσίας, η οποία παρέχει στους χρήστες ακριβείς και λεπτομερείς πληροφορίες για την τοποθεσία τους. Αυτό επιτρέπει στους χρήστες να προσανατολίζονται με άνεση και αυτοπεποίθηση, ακόμα και σε περίπλοκα ή άγνωστα περιβάλλοντα.

Συνολικά, η εφαρμογή BlindNavi αποτελεί ένα πολλά υποσχόμενο εργαλείο που μπορεί να βελτιώσει δραματικά την ποιότητα ζωής ατόμων με οπτικές δυσκολίες. Παρέχοντας εύχρηστες οδηγίες, πολυαισθητηρική ανατροφοδότηση και ακριβείς πληροφορίες τοποθεσίας, η BlindNavi μπορεί να ενδυναμώσει τα άτομα με οπτικές δυσκολίες, επιτρέποντάς τους να εξερευνούν τον κόσμο με αυτοπεποίθηση και ανεξαρτησία [15].

## 5.5 FoodTracker

Η εφαρμογή FoodTracker αναπτύχθηκε για κινητές συσκευές, η οποία αξιοποιεί τεχνολογίες βαθιάς μάθησης για την αναγνώριση πολλαπλών τροφίμων σε ένα γεύμα, απεικονιζόμενο σε μία εικόνα, και την παροχή των αντίστοιχων διατροφικών πληροφοριών σε πραγματικό χρόνο [16]. Η κατασκευή του μοντέλου διαρθρώνεται σε δύο κύρια στάδια:

### Στάδιο 1: Ανάπτυξη βαθύ δικτύου νευρώνων με σύντηξη YOLO

Αναπτύχθηκε ένα βαθύ δίκτυο νευρώνων CNN το οποίο ενσωματώνει την τεχνολογία ανίχνευσης αντικειμένων YOLO, που αποτελεί μία κορυφαία στρατηγική ανίχνευσης. Αυτή η σύντηξη επιτυγχάνει ταυτόχρονη αναγνώριση και εντοπισμό πολλαπλών αντικειμένων (τροφίμων) στην εικόνα, με μέση ακρίβεια που αγγίζει το 80% [16].

## Στάδιο 2: Προσαρμογή σε κινητή εφαρμογή με ανάλυση θρεπτικών συστατικών

Το αναπτυγμένο μοντέλο προσαρμόζεται σε μια εφαρμογή για κινητές συσκευές, η οποία εμπλουτίζεται με πρόσθετες λειτουργίες για την ανάλυση των διατροφικών συστατικών των αναγνωρισμένων τροφίμων. Μετά την εξαγωγή και αποκωδικοποίηση των αποτελεσμάτων του μοντέλου εντός της εφαρμογής, παρουσιάζονται στον χρήστη τα ανιχνευμένα αντικείμενα, συμπεριλαμβανομένης της θέσης του οροθετικού πλαισίου και της ετικέτας κατηγορίας. Η εφαρμογή λειτουργεί σε πραγματικό χρόνο και είναι κατάλληλη για κινητές συσκευές, χάρη στον αλγόριθμο βαθιάς μάθησης που έχει σχεδιαστεί για χαμηλό χρόνο εκτέλεσης (inference time) και περιορισμένες απαιτήσεις μνήμης [16].

### 5.6 Επίλογος

Στο παρόν κεφάλαιο της παρούσας εργασίας, εξετάσαμε ορισμένες παρεμφερείς εφαρμογές με την προτεινόμενη λύση ανίχνευσης καθημερινών αντικειμένων. Αναλύσαμε τα βασικά χαρακτηριστικά και τις δυνατότητες εφαρμογών όπως το Be My Eyes, το Seeing AI, το BlindNavi και το FoodTracker, θέτοντας σε έμφαση την αξιοποίηση της τεχνητής νοημοσύνης και της πολυαισθητηριακής ανατροφοδότησης για την ενίσχυση της αυτονομίας και της ποιότητας ζωής ατόμων με οπτικές δυσκολίες. Η ανάλυση παρεμφερών εφαρμογών παρέχει πολύτιμες πληροφορίες και ιδέες για την ανάπτυξη μίας καινοτόμου και αποτελεσματικής λύσης ανίχνευσης καθημερινών αντικειμένων, η οποία δύναται να συμβάλλει στην ενίσχυση της αυτονομίας και της ποιότητας ζωής ατόμων με οπτικές δυσκολίες.

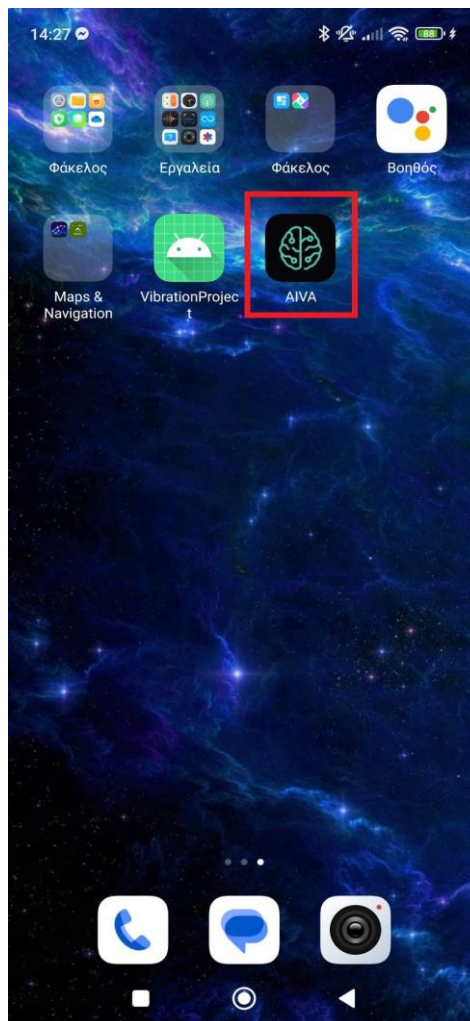
## Κεφάλαιο 6ο: Εφαρμογή

### 6.1 Εισαγωγή

Αυτό το κεφάλαιο εστιάζει στην ανάλυση του τρόπου λειτουργίας και του συνδυασμού τεχνολογιών της εφαρμογής η οποία αναπτύχθηκε στο πλαίσιο της παρούσης εργασίας, με τεχνολογίες όπως η απτική ανατροφοδότηση και η φωνητική περιγραφή. Η εφαρμογή αξιοποιεί την τεχνητή νοημοσύνη για την ανίχνευση αντικειμένων σε πραγματικό χρόνο, παρέχοντας παράλληλα στους χρήστες αισθητηριακή και ακουστική πληροφορία για τα ανιχνευμένα αντικείμενα. Η εφαρμογή απευθύνεται σε άτομα με προβλήματα όρασης, προσφέροντάς τους ένα καινοτόμο εργαλείο για την αλληλεπίδραση με τον ψηφιακό και τον φυσικό κόσμο. Επίσης να αναφέρουμε εδώ ότι η εφαρμογή ονομάστηκε AIVA και είναι τα αρχικά από τις λέξεις Artificial Intelligence Vision Assistant. Επιπλέον να πούμε ότι το μοντέλο ανίχνευσης αντικειμένων έχει «εκπαιδευτεί» σε ένα σύνολο δεδομένων το οποίο ονομάζεται COCO αποτελεί ένα εκτεταμένο σύνολο δεδομένων που εστιάζει στην αναγνώριση αντικειμένων σε ρεαλιστικές εικόνες και συμπεριλαμβάνει 91 κατηγορίες συνηθισμένων αντικειμένων [17].

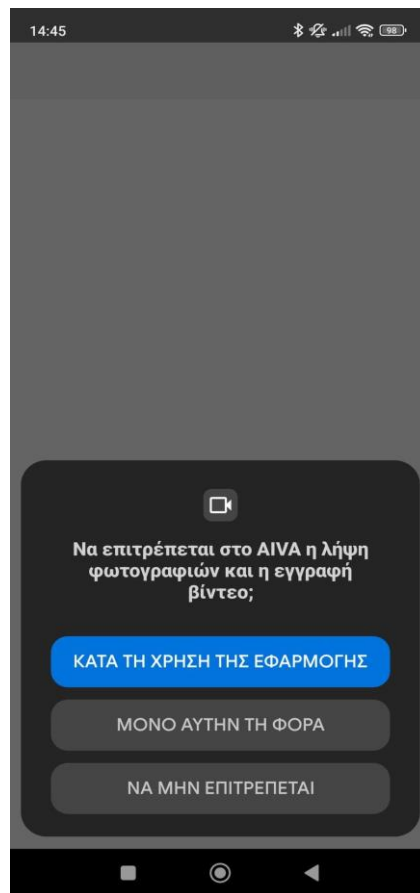
### 6.2 Έναρξη της εφαρμογής

- Ο χρήστης εκκινεί την εφαρμογή από την οθόνη της συσκευής.



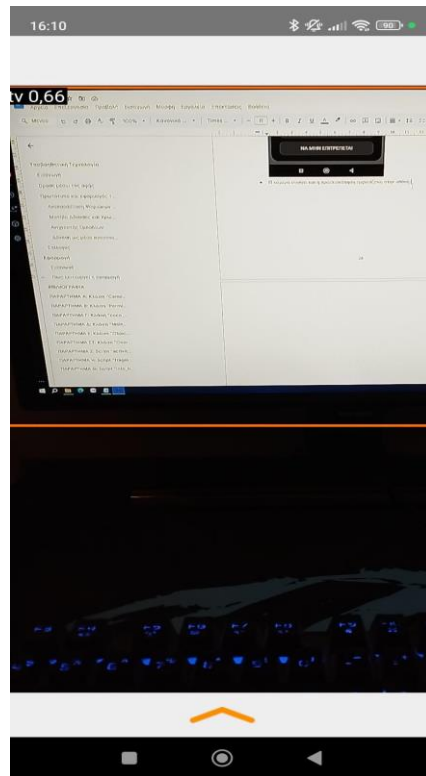
Εικόνα 6.1: Εκκίνηση της εφαρμογής

- Η εφαρμογή ζητά άδεια πρόσβασης για την κάμερα από τον χρήστη.



Εικόνα 6.2: Διεπαφή αδειών

- Με το που δοθεί η πρόσβαση κάμερα ξεκινά η λήψη καρέ σε συνεχή ροή.



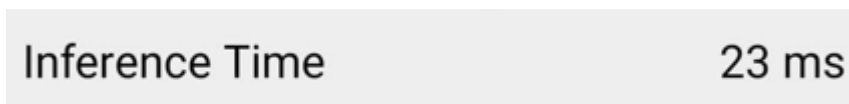
Εικόνα 6.3: Προεπισκόπηση κάμερας ενώ κάνει συνεχή ροή λήψης

### 6.2.1 Περιγραφή της Διεπαφής Χρήστη

Οι ρυθμίσεις βρίσκονται σε ένα συρόμενο προς τα πάνω μενού και με το που το ανοίξει ο χρήστης θα δει μία ετικέτα με τον χρόνο απόκρισης και 5 διαφορετικές ρυθμίσεις. Ακολουθούν λεπτομερείς περιγραφές της διεπαφής:

#### Ταμπέλα με τον χρόνο απόκρισης:

- Η ταμπέλα μας ενημερώνει σε πραγματικό χρόνο για τον χρόνο απόκρισης σε millisecond.



Εικόνα 6.4: Ταμπέλα με τον χρόνο απόκρισης

#### Επιλογή Ευαισθησίας:

- Ελέγχει πόσο αυστηρά η εφαρμογή φιλτράρει τα πιθανά αντικείμενα. Χαμηλότερο ποσοστό ευαισθησίας ανιχνεύει περισσότερα αντικείμενα (ίσως και ψευδώς θετικά), ενώ υψηλότερο ποσοστό ευαισθησίας εστιάζει σε πιο σίγουρα αντικείμενα.



Εικόνα 6.5: Επιλογή ευαισθησίας

#### Επιλογή Αριθμού Αντικειμένων:

- Η συγκεκριμένη ρύθμιση αυξομειώνει τον αριθμό των ανιχνεύσιμων αντικειμένων με μεγαλύτερο το 5 και μικρότερο το 1



Εικόνα 6.6: Επιλογή αριθμού ανιχνεύσιμων αντικειμένων

### Επιλογή Αριθμού Νημάτων:

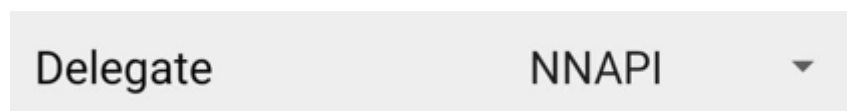
- Ο αριθμός αυτός διαδραματίζει σημαντικό ρόλο διότι η εφαρμογή πρέπει να εκτελέσει αρκετές διεργασίες, οπότε όσο μεγαλύτερος τόσο λιγότερο χρόνο απόκρισης θα έχει η εφαρμογή. Ο μέγιστος αριθμός νημάτων είναι 4 και ο ελάχιστος 1.



Εικόνα 6.7: Επιλογή αριθμού νημάτων

### Επιλογή Delegate

- Η επιλογή Delegate επιτρέπει στους χρήστες να καθορίσουν πώς θα εκτελείται η επεξεργασία ανίχνευσης αντικειμένων. Κάθε επιλογή αξιοποιεί διαφορετικούς πόρους υλικού στη συσκευή, οδηγώντας σε trade-offs μεταξύ ταχύτητας, κατανάλωσης ενέργειας και συμβατότητας.



Εικόνα 6.8: Επιλογή Delegate

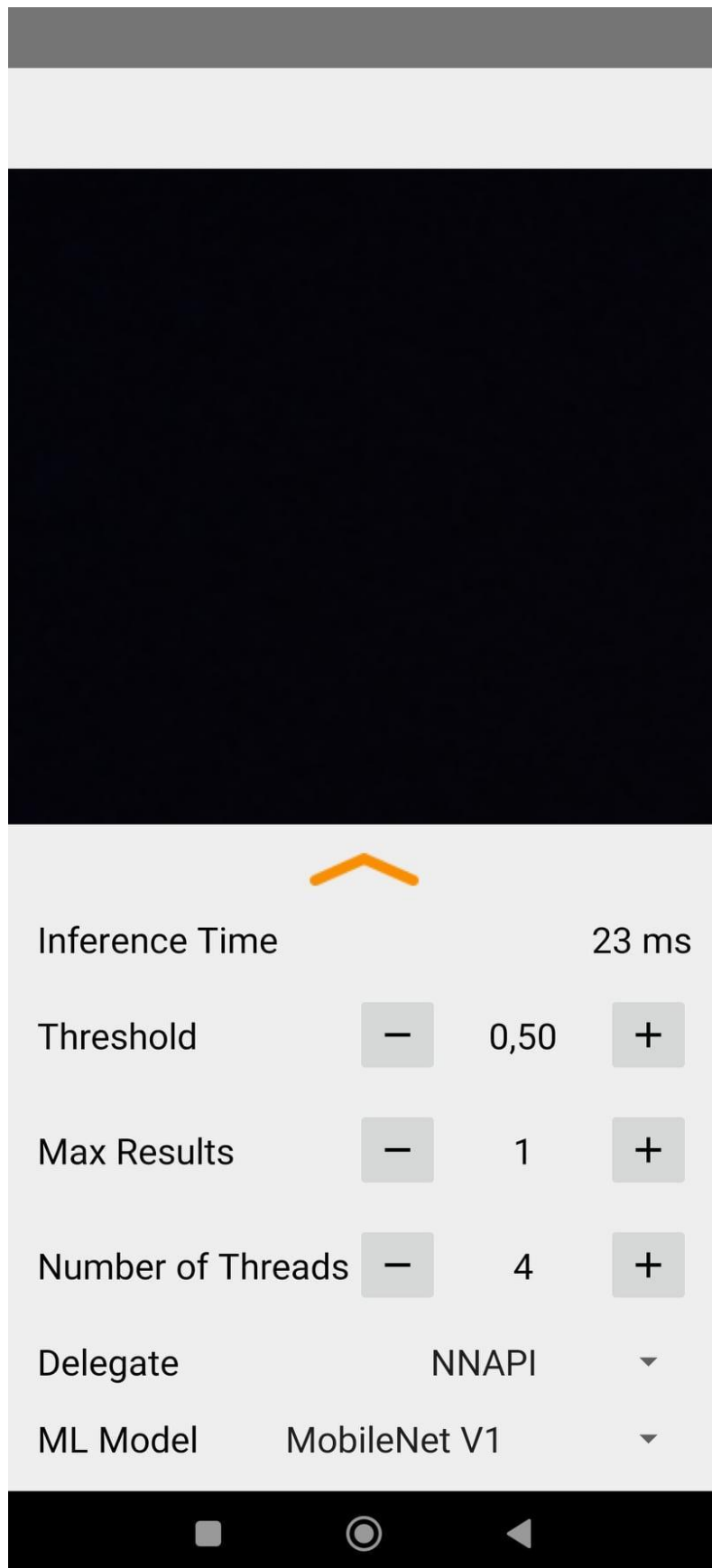
### Επιλογή Μοντέλου

- Προεπιλεγμένο μοντέλο της εφαρμογής είναι το MobileNetV1, όμως υπάρχουν και άλλα 3 μοντέλα που μπορεί να χρησιμοποιήσει ο χρήστης και αυτά είναι τα EfficientDet Lite0, EfficientDet Lite1, EfficientDet Lite2.



Εικόνα 6.9: Επιλογή μοντέλου ανίχνευσης αντικειμένων

Παρακάτω φαίνεται το στιγμιότυπο με την διεπαφή της εφαρμογής και το ανοιχτό μενού.



Εικόνα 6.10: Διεπαφή ρυθμίσεων

## 6.3 Περιγραφή του κώδικα

Αρχικά να αναφέρω ότι ένα κομμάτι της εφαρμογής ανίχνευσης αντικειμένων, η οποία αναπτύχθηκε στο πλαίσιο αυτής της εργασίας, εξήχθει από το github της πλατφόρμας ανοιχτού κώδικα της tensorflow [6] και πρόσθεσα την απτική ανατροφοδότηση και την φωνητική περιγραφή των αντικειμένων στην Ελληνική γλώσσα. Η ανάπτυξη της εφαρμογής έγινε στο περιβάλλον Android Studio επομένως η εφαρμογή είναι για κινητές συσκευές οι οποίες υποστηρίζουν το λογισμικό Android.

### 6.3.1 Περιγραφή Κλάσης “CameraFragment”

Ο κώδικας της κλάσης CameraFragment υλοποιεί την λειτουργία ανίχνευσης αντικειμένων σε πραγματικό χρόνο. Η διαδικασία αυτή αξιοποιεί την κάμερα της συσκευής για τη λήψη συνεχούς ροής εικόνων και την επεξεργασία τους με σκοπό την αναγνώριση αντικειμένων που εμπεριέχονται σε αυτές. Η αρχιτεκτονική του κώδικα βασίζεται σε διάφορες βιβλιοθήκες και διασυνδέσεις του Android Studio για την επίτευξη της ανίχνευσης αντικειμένων. Αναλυτικότερα, θα δούμε τον κώδικα τι κάνει γραμμή γραμμή.

**Δήλωση Πακέτου(γραμμή 1):** Δηλώνεται το πακέτο στο οποίο ανήκει η κλάση CameraFragment.

**Imports(γραμμές 3-30):** Σε αυτό το τμήμα, ο κώδικας εισάγει απαραίτητες βιβλιοθήκες που χρειάζεται η εφαρμογή για να λειτουργήσει.

```

1 package org.tensorflow.lite.examples.objectdetection.fragments
2
3 import android.annotation.SuppressLint
4 import android.content.res.Configuration
5 import android.graphics.Bitmap
6 import android.os.Bundle
7 import android.util.Log
8 import android.view.LayoutInflater
9 import android.view.View
10 import android.view.ViewGroup
11 import android.widget.AdapterView
12 import android.widget.Toast
13 import androidx.camera.core.AspectRatio
14 import androidx.camera.core.Camera
15 import androidx.camera.core.CameraSelector
16 import androidx.camera.core.ImageAnalysis
17 import androidx.camera.core.ImageAnalysis.OUTPUT_IMAGE_FORMAT_RGBA_8888
18 import androidx.camera.core.ImageProxy
19 import androidx.camera.core.Preview
20 import androidx.camera.lifecycle.ProcessCameraProvider
21 import androidx.core.content.ContextCompat
22 import androidx.fragment.app.Fragment
23 import androidx.navigation.Navigation
24 import java.util.LinkedList
25 import java.util.concurrent.ExecutorService
26 import java.util.concurrent.Executors
27 import org.tensorflow.lite.examples.objectdetection.ObjectDetectorHelper
28 import org.tensorflow.lite.examples.objectdetection.R
29 import org.tensorflow.lite.examples.objectdetection.databinding.FragmentCameraBinding
30 import org.tensorflow.lite.task.vision.detector.Detection
31

```

Εικόνα 6.11: Imports

**Δήλωση Κλάσης και Μεταβλητών(γραμμές 34-49):** Ορίζει την κλάση CameraFragment που κληρονομεί την κλάση Fragment και υλοποιεί τη διασύνδεση ObjectDetectorHelper.DetectorListener. Ορίζεται μια σταθερά TAG για ευκολότερο εντοπισμό μηνυμάτων log που σχετίζονται με την ανίχνευση αντικειμένων. Δηλώνεται μια μεταβλητή fragmentCameraBinding τύπου FragmentCameraBinding για πρόσβαση στα στοιχεία του interface του XML αρχείου που καθορίζει τη

διάταξη της οθόνης. Δηλώνονται άλλες μεταβλητές για διάφορα αντικείμενα που χρησιμοποιούνται στην εφαρμογή, όπως `objectDetectorHelper` για τον χειρισμό του μοντέλου ανίχνευσης αντικειμένων, `bitmapBuffer` για την αποθήκευση των εικόνων της κάμερας σε μορφή `bitmap`, `preview`, `imageAnalyzer` και `camera` για τη διαχείριση της κάμερας και της ανάλυσης εικόνας, `cameraProvider` για πρόσβαση στις λειτουργίες της κάμερας της συσκευής, και `cameraExecutor` για την εκτέλεση εργασιών της κάμερας σε ξεχωριστό `thread`.

```

32 class CameraFragment : Fragment(), ObjectDetectorHelper.DetectorListener {
33
34     private val TAG = "ObjectDetection"
35
36     private var _fragmentCameraBinding: FragmentCameraBinding? = null
37
38     private val fragmentCameraBinding
39         get() = _fragmentCameraBinding!!
40
41     private lateinit var objectDetectorHelper: ObjectDetectorHelper
42     private lateinit var bitmapBuffer: Bitmap
43     private var preview: Preview? = null
44     private var imageAnalyzer: ImageAnalysis? = null
45     private var camera: Camera? = null
46     private var cameraProvider: ProcessCameraProvider? = null
47
48     /** Blocking camera operations are performed using this executor */
49     private lateinit var cameraExecutor: ExecutorService

```

Εικόνα 6.12: Μεταβλητές

**Μέθοδος `onResume`(γραμμές 51-59):** Η μέθοδος `onResume` καλείται όταν η οθόνη της εφαρμογής ενεργοποιείται. Ελέγχει αν ο χρήστης έχει δώσει τα απαραίτητα δικαιώματα για την πρόσβαση στην κάμερα.

```

51 override fun onResume() {
52     super.onResume()
53     // Make sure that all permissions are still present, since the
54     // user could have removed them while the app was in paused state.
55     if (!PermissionsFragment.hasPermissions(requireContext())) {
56         Navigation.findNavController(requireActivity(),
57             R.id.fragment_container).navigate(CameraFragmentDirections.actionCameraToPermissions())
58     }
59 }

```

Εικόνα 6.13: Μέθοδος `onResume`

**Μέθοδος `onDestroyView`(γραμμές 61-67):** Η μέθοδος `onDestroyView` καλείται όταν η οθόνη της εφαρμογής απενεργοποιείται. Αποδεσμεύει τη μεταβλητή `fragmentCameraBinding` και τερματίζει το `thread cameraExecutor` που χρησιμοποιείται για την κάμερα.

```

61  override fun onDestroyView() {
62      _fragmentCameraBinding = null
63      super.onDestroyView()
64
65      // Shut down our background executor
66      cameraExecutor.shutdown()
67  }

```

Εικόνα 6.14: Μέθοδος onDestroyView

**Μέθοδος onCreateView(γραμμές 69-77):** Η μέθοδος onCreateView() καλείται για τη δημιουργία του layout του Fragment. Κάνει Inflate το layout του Fragment από το αντίστοιχο αρχείο xml. Αυτό δημιουργεί τα στοιχεία του Fragment στην οθόνη. Αποθηκεύει την μεταβλητή binding object η οποία παρέχει μια πιο βολική πρόσβαση στα στοιχεία του layout που ορίστηκαν στο αρχείο XML. Επιστρέφει την root view του layout.

```

69  override fun onCreateView(
70      inflater: LayoutInflater,
71      container: ViewGroup?,
72      savedInstanceState: Bundle?
73  ): View {
74      _fragmentCameraBinding = FragmentCameraBinding.inflate(inflater, container, attachToParent: false)
75
76      return fragmentCameraBinding.root
77  }

```

Εικόνα 6.15: Μέθοδος onCreateView

**Μέθοδος onViewCreated(γραμμές 80-98):** Η μέθοδος onViewCreated καλείται αφού η διάταξη της οθόνης έχει δημιουργηθεί. Δημιουργεί ένα αντικείμενο objectDetectorHelper για τον χειρισμό του μοντέλου ανίχνευσης αντικειμένων, αρχικοποιεί το νήμα για την μεταβλητή cameraExecutor και περιμένει μέχρι να οριστικοποιηθούν τα στοιχεία της οθόνης πριν ρυθμίσει την κάμερα.

```

80 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
81     super.onViewCreated(view, savedInstanceState)
82
83     objectDetectorHelper = ObjectDetectorHelper(
84         context = requireContext(),
85         objectDetectorListener = this)
86
87     // Initialize our background executor
88     cameraExecutor = Executors.newSingleThreadExecutor()
89
90     // Wait for the views to be properly laid out
91     fragmentCameraBinding.viewFinder.post {
92         // Set up the camera and its use cases
93         setUpCamera()
94     }
95
96     // Attach listeners to UI control widgets
97     initBottomSheetControls()
98 }

```

Εικόνα 6.16: Μέθοδος onViewCreated

**Μέθοδος initBottomSheetControls(γραμμές 100-177):** Η μέθοδος initBottomSheetControls προσθέτει listeners στα στοιχεία ελέγχου του bottom sheet που επιτρέπουν στον χρήστη να ρυθμίσει τις παραμέτρους ανίχνευσης αντικειμένων, όπως το ποσοστό ευαισθησίας, τον αριθμό των ανιχνευόμενων αντικειμένων, τον αριθμό των νημάτων, την επιλογή delegate και τον τύπο μοντέλου.

```

100 private fun initBottomSheetControls() {
101     // When clicked, lower detection score threshold floor
102     fragmentCameraBinding.bottomSheetLayout.thresholdMinus.setOnClickListener { it: View!
103         if (objectDetectorHelper.threshold >= 0.1) {
104             objectDetectorHelper.threshold -= 0.1f
105             updateControlsUi()
106         }
107     }
108
109     // When clicked, raise detection score threshold floor
110     fragmentCameraBinding.bottomSheetLayout.thresholdPlus.setOnClickListener { it: View!
111         if (objectDetectorHelper.threshold <= 0.8) {
112             objectDetectorHelper.threshold += 0.1f
113             updateControlsUi()
114         }
115     }
116
117     // When clicked, reduce the number of objects that can be detected at a time
118     fragmentCameraBinding.bottomSheetLayout.maxResultsMinus.setOnClickListener { it: View!
119         if (objectDetectorHelper.maxResults > 1) {
120             objectDetectorHelper.maxResults--
121             updateControlsUi()
122         }
123     }

```

Εικόνα 6.17: Μέθοδος initBottomSheetControls

```

124
125 // When clicked, increase the number of objects that can be detected at a time
126 fragmentCameraBinding.bottomSheetLayout.maxResultsPlus.setOnClickListener { it: View!
127     if (objectDetectorHelper.maxResults < 5) {
128         objectDetectorHelper.maxResults++
129         updateControlsUi()
130     }
131 }
132
133 // When clicked, decrease the number of threads used for detection
134 fragmentCameraBinding.bottomSheetLayout.threadsMinus.setOnClickListener { it: View!
135     if (objectDetectorHelper.numThreads > 1) {
136         objectDetectorHelper.numThreads--
137         updateControlsUi()
138     }
139 }
140
141 // When clicked, increase the number of threads used for detection
142 fragmentCameraBinding.bottomSheetLayout.threadsPlus.setOnClickListener { it: View!
143     if (objectDetectorHelper.numThreads < 4) {
144         objectDetectorHelper.numThreads++
145         updateControlsUi()
146     }
147 }

```

Εικόνα 6.18: Μέθοδος initBottomSheetControls

```

149 // When clicked, change the underlying hardware used for inference. Current options are CPU
150 // GPU, and NNAPI
151 fragmentCameraBinding.bottomSheetLayout.spinnerDelegate.setSelection( position: 0, animate: false)
152 fragmentCameraBinding.bottomSheetLayout.spinnerDelegate.onItemSelectedListener =
153     object : AdapterView.OnItemSelectedListener {
154         override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
155             objectDetectorHelper.currentDelegate = p2
156             updateControlsUi()
157         }
158
159         override fun onNothingSelected(p0: AdapterView<*>?) {
160             /* no op */
161         }
162     }
163
164 // When clicked, change the underlying model used for object detection
165 fragmentCameraBinding.bottomSheetLayout.spinnerModel.setSelection( position: 0, animate: false)
166 fragmentCameraBinding.bottomSheetLayout.spinnerModel.onItemSelectedListener =
167     object : AdapterView.OnItemSelectedListener {
168         override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
169             objectDetectorHelper.currentModel = p2
170             updateControlsUi()
171         }
172
173         override fun onNothingSelected(p0: AdapterView<*>?) {
174             /* no op */
175         }
176     }
177 }

```

Εικόνα 6.19: Μέθοδος initBottomSheetControls

**Μέθοδος updateControlsUi(γραμμές 180-192):** Η μέθοδος updateControlsUi ενημερώνει τα στοιχεία του UI στο bottom sheet με τις τρέχουσες τιμές των παραμέτρων ανίχνευσης αντικειμένων. Ενημερώνει το πεδίο για τον μέγιστο αριθμό αντικειμένων που μπορούν να ανιχνευθούν (maxResults), το όριο σκόρ ανίχνευσης (threshold) και τον αριθμό των threads που χρησιμοποιούνται για την ανίχνευση (numThreads). Επιπλέον, επαναρυθμίζει τον ανιχνευτή αντικειμένων (objectDetectorHelper) και διαγράφει τυχόν υπάρχοντα αποτελέσματα ανίχνευσης από την κλάση overlay.

```

179 // Update the values displayed in the bottom sheet. Reset detector.
180 private fun updateControlsUi() {
181     fragmentCameraBinding.bottomSheetLayout.maxResultsValue.text =
182         objectDetectorHelper.maxResults.toString()
183     fragmentCameraBinding.bottomSheetLayout.thresholdValue.text =
184         String.format("%.2f", objectDetectorHelper.threshold)
185     fragmentCameraBinding.bottomSheetLayout.threadsValue.text =
186         objectDetectorHelper.numThreads.toString()
187
188     // Needs to be cleared instead of reinitialized because the GPU
189     // delegate needs to be initialized on the thread using it when applicable
190     objectDetectorHelper.clearObjectDetector()
191     fragmentCameraBinding.overlay.clear()
192 }

```

Εικόνα 6.20: Μέθοδος updateControlsUi

**Μέθοδος setUpCamera(γραμμές 195-207):** Η μέθοδος setUpCamera είναι υπεύθυνη για την αρχικοποίηση της κάμερας και την προετοιμασία της σύνδεσης των περιπτώσεων χρήσης της. Πρώτα, αποκτά ένα Instance του ProcessCameraProvider μέσω της μεθόδου getInstance. Στη συνέχεια, προσθέτει έναν listener ο οποίος θα εκτελεστεί όταν η μεταβλητή CameraProvider είναι διαθέσιμη. Μέσα στον listener, αποθηκεύει το Instance της μεταβλητής CameraProvider και καλεί τη μέθοδο bindCameraUseCases για να δημιουργήσει και να συνδέσει τις περιπτώσεις χρήσης της κάμερας.

```

195 private fun setUpCamera() {
196     val cameraProviderFuture = ProcessCameraProvider.getInstance(requireContext())
197     cameraProviderFuture.addListener(
198         {
199             // CameraProvider
200             cameraProvider = cameraProviderFuture.get()
201
202             // Build and bind the camera use cases
203             bindCameraUseCases()
204         },
205         ContextCompat.getMainExecutor(requireContext())
206     )
207 }

```

Εικόνα 6.21: Μέθοδος setUpCamera

**Μέθοδος bindCameraUseCases(211-266):** Η μέθοδος bindCameraUseCases δημιουργεί και συνδέει τις περιπτώσεις χρήσης της κάμερας:

- Δημιουργεί ένα αντικείμενο CameraSelector για να επιλέξει την πίσω κάμερα της συσκευής.
- Δημιουργεί ένα αντικείμενο Preview για την προεπισκόπηση της κάμερας σε αναλογία διαστάσεων 4:3.
- Δημιουργεί ένα αντικείμενο ImageAnalysis για την ανάλυση των εικόνων της κάμερας.
  - Η μέθοδος setAnalyzer του ImageAnalysis ορίζει μια lambda συνάρτηση η οποία θα εκτελείται για κάθε νέο πλαίσιο εικόνας που λαμβάνεται από την κάμερα.
  - Η lambda συνάρτηση αυτή αντιγράφει τα δεδομένα εικόνας στον κοινόχρηστο buffer bitmapBuffer και καλεί τη μέθοδο detectObjects για να ανιχνεύσει αντικείμενα στην εικόνα.

Πριν τη σύνδεση των περιπτώσεων χρήσης, καλείται η μέθοδος unbindAll του cameraProvider για να αποσυνδεθούν τυχόν υπάρχουσες περιπτώσεις χρήσης. Στη συνέχεια, η μέθοδος bindToLifecycle του cameraProvider καλείται για να συνδέσει τις περιπτώσεις χρήσης preview και imageAnalyzer με τον κύκλο ζωής του fragment. Τέλος, ορίζεται ο surfaceProvider του αντικειμένου preview στον surfaceProvider του viewfinder της κάμερας, ο οποίος εμφανίζει την προεπισκόπηση της κάμερας στην οθόνη.

```

211 private fun bindCameraUseCases() {
212
213     // CameraProvider
214     val cameraProvider =
215         cameraProvider ?: throw IllegalStateException("Camera initialization failed.")
216
217     // CameraSelector - makes assumption that we're only using the back camera
218     val cameraSelector =
219         CameraSelector.Builder().requireLensFacing(CameraSelector.LENS_FACING_BACK).build()
220
221     // Preview. Only using the 4:3 ratio because this is the closest to our models
222     preview =
223         Preview.Builder()
224             .setTargetAspectRatio(AspectRatio.RATIO_4_3)
225             .setTargetRotation(fragmentCameraBinding.viewFinder.display.rotation)
226             .build()

```

Εικόνα 6.22: Μέθοδος bindCameraUseCases

```

228 // ImageAnalysis. Using RGBA 8888 to match how our models work
229 imageAnalyzer =
230     ImageAnalysis.Builder() ImageAnalysis.Builder
231         .setTargetAspectRatio(AspectRatio.RATIO_4_3)
232         .setTargetRotation(fragmentCameraBinding.viewFinder.display.rotation)
233         .setBackpressureStrategy(ImageAnalysis.STRATEGY_KEEP_ONLY_LATEST)
234         .setOutputImageFormat(OUTPUT_IMAGE_FORMAT_RGBA_8888)
235         .build() ImageAnalysis
236 // The analyzer can then be assigned to the instance
237 .also { it: ImageAnalysis
238     it.setAnalyzer(cameraExecutor) { image ->
239         if (!::bitmapBuffer.isInitialized) {
240             // The image rotation and RGB image buffer are initialized only once
241             // the analyzer has started running
242             bitmapBuffer = Bitmap.createBitmap(
243                 image.width,
244                 image.height,
245                 Bitmap.Config.ARGB_8888
246             )
247         }

```

Εικόνα 6.23: Μέθοδος bindCameraUseCases

```

249         detectObjects(image)
250     }
251 }
252
253 // Must unbind the use-cases before rebinding them
254 cameraProvider.unbindAll()
255
256 try {
257     // A variable number of use-cases can be passed here -
258     // camera provides access to CameraControl & CameraInfo
259     camera = cameraProvider.bindToLifecycle(lifecycleOwner.this, cameraSelector, preview, imageAnalyzer)
260
261     // Attach the viewfinder's surface provider to preview use case
262     preview?.setSurfaceProvider(fragmentCameraBinding.viewFinder.surfaceProvider)
263 } catch (exc: Exception) {
264     Log.e(TAG, msg:"Use case binding failed", exc)
265 }
266 }

```

Εικόνα 6.24: Μέθοδος bindCameraUseCases

**Μέθοδος detectObjects(image: ImageProxy)(γραμμές 268-275):** Η μέθοδος detectObjects λαμβάνει ένα αντικείμενο ImageProxy που αντιπροσωπεύει το τρέχον πλαίσιο εικόνας από την κάμερα. Αντιγράφει τα δεδομένα εικόνας RGB στον κοινόχρηστο buffer bitmapBuffer. Λαμβάνει τον περιστροφικό προσανατολισμό της εικόνας (imageRotation). Καλεί τη μέθοδο detect του αντικειμένου objectDetectorHelper για να ανιχνεύσει αντικείμενα στην εικόνα. Η μέθοδος αυτή δέχεται το bitmapBuffer και τον imageRotation ως ορίσματα.

```

268     private fun detectObjects(image: ImageProxy) {
269         // Copy out RGB bits to the shared bitmap buffer
270         image.use { bitmapBuffer.copyPixelsFromBuffer(image.planes[0].buffer) }
271
272         val imageRotation = image.imageInfo.rotationDegrees
273         // Pass Bitmap and rotation to the object detector helper for processing and detection
274         objectDetectorHelper.detect(bitmapBuffer, imageRotation)
275     }

```

Εικόνα 6.25: Μέθοδος detectObjects

**Μέθοδος onConfigurationChanged(newConfig: Configuration)(γραμμές 277-280):** Η μέθοδος onConfigurationChanged καλείται όταν ο προσανατολισμός της συσκευής αλλάζει. Ενημερώνει τον περιστροφικό προσανατολισμό του αντικειμένου imageAnalyzer για να διασφαλίσει ότι τα ανιχνευμένα αντικείμενα θα τοποθετηθούν σωστά στην οθόνη ακόμα και όταν αλλάζει ο προσανατολισμός.

```

277     override fun onConfigurationChanged(newConfig: Configuration) {
278         super.onConfigurationChanged(newConfig)
279         imageAnalyzer?.targetRotation = fragmentCameraBinding.viewFinder.display.rotation
280     }

```

Εικόνα 6.26: Μέθοδος onConfigurationChanged

**Μέθοδος onResults(results: MutableList<Detection>?, inferenceTime: Long, imageHeight: Int, imageWidth: Int)(γραμμές 284-304):** Η μέθοδος onResults καλείται από τον ObjectDetectorHelper όταν ολοκληρωθεί η ανίχνευση αντικειμένων σε ένα πλαίσιο εικόνας.

#### Λειτουργίες:

- Ενημέρωση Χρόνου Ανίχνευσης:
  - Εμφανίζει στην διεπαφή τον χρόνο που χρειάστηκε για την ανίχνευση αντικειμένων στο τρέχον πλαίσιο εικόνας.
  - Το κείμενο ενημερώνεται με το μήνυμα "Inference Time: X ms", όπου X είναι ο χρόνος σε χιλιοστά του δευτερολέπτου.
- Εμφάνιση Αποτελεσμάτων Ανίχνευσης:
  - Παίρνει μια λίστα με τα ανιχνευμένα αντικείμενα (results).
  - Καλεί τη μέθοδο setResults της κλάσης overlay για να ορίσει τα αποτελέσματα της ανίχνευσης:
    - Η λίστα results περιέχει λεπτομερείς πληροφορίες για κάθε ανιχνευμένο αντικείμενο, όπως η κατηγορία, το σκορ ανίχνευσης, το πλαίσιο περιοχής και τυχόν δεδομένα ετικέτας.
    - Το ύψος και το πλάτος της αρχικής εικόνας (imageHeight και imageWidth) χρησιμοποιούνται για να υπολογιστούν οι σωστές θέσεις και μεγέθη των οπτικοποιημένων στοιχείων (π.χ., πλαίσια περιοχής) στην οθόνη.
- Αναπαράσταση Αποτελεσμάτων:
  - Καλεί τη μέθοδο invalidate του αντικειμένου overlay για να αναγκάσει την αναπαράσταση των αποτελεσμάτων της ανίχνευσης στην οθόνη.
  - Αυτό σημαίνει ότι το overlay θα επανασχεδιάσει το περιεχόμενό του με βάση τα νέα δεδομένα ανίχνευσης που έχουν οριστεί.

```

284 override fun onResults(
285     results: MutableList<Detection>?,
286     inferenceTime: Long,
287     imageHeight: Int,
288     imageWidth: Int
289 ) {
290     activity?.runOnUiThread {
291         fragmentCameraBinding.bottomSheetLayout.inferenceTimeVal.text =
292             String.format("%d ms", inferenceTime)
293
294         // Pass necessary information to OverlayView for drawing on the canvas
295         fragmentCameraBinding.overlay.setResults(
296             detectionResults: results ?: LinkedList<Detection>(),
297             imageHeight,
298             imageWidth
299         )
300
301         // Force a redraw
302         fragmentCameraBinding.overlay.invalidate()
303     }
304 }

```

Εικόνα 6.27: Μέθοδος onResults

**Μέθοδος onError(error: String)(γραμμές 306-310):** Η μέθοδος onError καλείται από τον ObjectDetectorHelper όταν προκύψει σφάλμα κατά την ανίχνευση αντικειμένων. Λαμβάνει ένα μήνυμα σφάλματος (error). Εμφανίζει ένα μήνυμα toast με το μήνυμα σφάλματος στην διεπαφή.

```

306 override fun onError(error: String) {
307     activity?.runOnUiThread {
308         Toast.makeText(requireContext(), error, Toast.LENGTH_SHORT).show()
309     }
310 }
311 }

```

Εικόνα 6.28: Μέθοδος onError

### 6.3.2 Περιγραφή Κλάσης “PermissionFragment”

#### Σκοπός:

- Αυτή η κλάση είναι υπεύθυνη για τη διαχείριση των αδειών που απαιτεί η εφαρμογή. Συγκεκριμένα, χρειάζεται την άδεια χρήσης της κάμερας για να λειτουργήσει η ανίχνευση αντικειμένων.

#### Μεταβλητές:

- PERMISSIONS\_REQUIRED: Ένας πίνακας strings που περιέχει την άδεια χρήσης της κάμερας (Manifest.permission.CAMERA).

```

1  package org.tensorflow.lite.examples.objectdetection.fragments
2
3  import android.Manifest
4  import android.content.Context
5  import android.content.pm.PackageManager
6  import android.os.Bundle
7  import android.widget.Toast
8  import androidx.activity.result.contract.ActivityResultContracts
9  import androidx.core.content.ContextCompat
10 import androidx.fragment.app.Fragment
11 import androidx.lifecycle.lifecycleScope
12 import androidx.navigation.Navigation
13 import org.tensorflow.lite.examples.objectdetection.R
14
15 private val PERMISSIONS_REQUIRED = arrayOf(Manifest.permission.CAMERA)

```

Εικόνα 6.29: Imports και μεταβλητή

### Μέθοδοι:

- `requestPermissionLauncher`: Αυτή η μέθοδος είναι ένας launcher που χρησιμοποιείται για την έναρξη της διαδικασίας αίτησης άδειας. Αν ο χρήστης δώσει την άδεια, εμφανίζει ένα μήνυμα επιβεβαίωσης και καλεί τη `navigateToCamera`. Αν αρνηθεί, εμφανίζει ένα μήνυμα άρνησης.

```

21 class PermissionsFragment : Fragment() {
22
23     private val requestPermissionLauncher =
24         registerForActivityResult(ActivityResultContracts.RequestPermission()
25             ) { isGranted: Boolean ->
26             if (isGranted) {
27                 Toast.makeText(context, text: "Permission request granted", Toast.LENGTH_LONG).show()
28                 navigateToCamera()
29             } else {
30                 Toast.makeText(context, text: "Permission request denied", Toast.LENGTH_LONG).show()
31             }
32         }
33 }

```

Εικόνα 6.30: Permission Launcher

- `onCreate`: Αυτή η μέθοδος καλείται όταν δημιουργείται το fragment. Ελέγχει εάν η άδεια χρήσης της κάμερας έχει ήδη δοθεί:
  - Αν ναι, καλεί τη μέθοδο `navigateToCamera` για να μεταβεί στο fragment της κάμερας.
  - Αν όχι, καλεί τη `requestPermissionLauncher` για να ζητήσει την άδεια από τον χρήστη.

```

34  override fun onCreate(savedInstanceState: Bundle?) {
35      super.onCreate(savedInstanceState)
36      when {
37          ContextCompat.checkSelfPermission(
38              requireContext(),
39              Manifest.permission.CAMERA
40          ) == PackageManager.PERMISSION_GRANTED -> {
41              navigateToCamera()
42          }
43          else -> {
44              requestPermissionLauncher.launch(
45                  Manifest.permission.CAMERA)
46          }
47      }
48  }

```

Εικόνα 6.31: Μέθοδος onCreate

- `navigateToCamera`: Αυτή η μέθοδος χρησιμοποιεί το Navigation για να μεταβεί από το fragment των αδειών στο fragment της κάμερας.

```

50  private fun navigateToCamera() {
51      lifecycleScope.launchWhenStarted { this: CoroutineScope
52          Navigation.findNavController(requireActivity(), R.id.fragment_container).navigate(
53              PermissionsFragmentDirections.actionPermissionsToCamera())
54      }
55  }

```

Εικόνα 6.32: Μέθοδος navigateToCamera

- `hasPermissions`: Αυτή η static μέθοδος είναι μια βοηθητική συνάρτηση που ελέγχει εάν όλες οι απαιτούμενες άδειες έχουν δοθεί.

```

60  fun hasPermissions(context: Context) = PERMISSIONS_REQUIRED.all { it: String
61      ContextCompat.checkSelfPermission(context, it) == PackageManager.PERMISSION_GRANTED
62  }
63  }
64  }

```

Εικόνα 6.33: Μέθοδος hasPermissions

### 6.3.3 Περιγραφή Κλάσης “coco\_translations”

Η συγκεκριμένη κλάση δημιουργήθηκε από τον συγγραφέα της παρούσης πτυχιακής και αποτελείται από μία λίστα με τα ανιχνεύσιμα αντικείμενα και την ελληνική μετάφραση της κατηγορίας κάθε αντικειμένου, έτσι ώστε η φωνητική περιγραφή να γίνεται στα Ελληνικά. Η λίστα είναι ένας hashmap με τις κατηγορίες των αντικειμένων ως το κλειδί και η τιμή είναι η Ελληνική μετάφραση.

```

1  package org.tensorflow.lite.examples.objectdetection
2
3  class coco_translations {
4      object CocoTranslations {
5          val translations = mapOf(
6              "person" to "άτομο",
7              "bicycle" to "ποδήλατο",
8              "car" to "αυτοκίνητο",
9              "motorcycle" to "μοτοσικλέτα",
10             "airplane" to "αεροπλάνο",
11             "bus" to "λεωφορείο",
12             "train" to "τρένο",
13             "truck" to "φορτηγό",
14             "boat" to "βάρκα",
15             "traffic light" to "φανάρι",
16             "fire hydrant" to "πυροσβεστικός κρουνός",
17             "stop sign" to "πινακίδα στοπ",
18             "parking meter" to "παρκόμετρο",
19             "bench" to "πάγκος",
20             "bird" to "πουλί",
21             "cat" to "γάτα",
22             "dog" to "σκύλος",
23             "horse" to "άλογο",
24             "sheep" to "πρόβατο",
25             "cow" to "αγελάδα",
26             "elephant" to "ελέφαντας",
27             "bear" to "αρκούδα",
28             "zebra" to "ζέβρα",
29             "giraffe" to "καμηλοπάρδαλι",
30             "backpack" to "σακίδιο πλάτης",

```

Εικόνα 6.34: Hashmap μεταφράσεων

31		"umbrella" to "ομπρέλα",
32	💡	"handbag" to "χαρτοφύλακας",
33		"tie" to "γραβάτα",
34		"suitcase" to "βαλίτσα",
35		"frisbee" to "φρίσμπι",
36		"skis" to "σκι",
37		"snowboard" to "σνόουμπορντ",
38		"sports ball" to "σφαίρα",
39		"kite" to "χαρταετός",
40		"baseball bat" to "ρόπαλο μπέιζμπολ",
41		"tennis racket" to "ρακέτα τένις",
42		"bottle" to "μπουκάλι",
43		"spoon" to "κουτάλι",
44		"fork" to "πιρουνί",
45		"knife" to "μαχαίρι",
46		"bowl" to "μπολ",
47		"banana" to "μπανάνα",
48		"apple" to "μήλο",
49		"sandwich" to "σάντουιτς",
50		"orange" to "πορτοκάλι",
51		"hot dog" to "χότ ντογκ",
52		"pizza" to "πίτσα",
53		"cup" to "κούπα",
54		"tv" to "τηλεόραση",
55		"laptop" to "φορητός υπολογιστής",
56		"mouse" to "ποντίκι υπολογιστη",
57		"keyboard" to "πληκτρολόγιο",
58		"cellphone" to "κινητό τηλέφωνο",
59		"microwave" to "φούρνος μικροκυμάτων",
60		"oven" to "φούρνος",

Εικόνα 6.35: Hashmap μεταφράσεων

```
61     "toaster" to "φρυγανιέρα",
62     "sink" to "νιπτήρας",
63     "refrigerator" to "ψυγείο",
64     "bathtub" to "μπανιέρα",
65     "toilet" to "τουαλέτα",
66     "shower" to "ντους",
67     "hair dryer" to "πιστολάκι μαλλιών",
68     "toothbrush" to "οδοντόβουρτσα",
69     "teddy bear" to "αρκουδάκι",
70     "saucepan" to "κατσαρόλα",
71     "frying pan" to "τηγάνι",
72     "pot" to "χύτρα",
73     "bed" to "κρεβάτι",
74     "couch" to "καναπές",
75     "chair" to "καρέκλα",
76     "table" to "τραπέζι",
77     "desk" to "γραφείο",
78     "lamp" to "λάμπα",
79     "nightstand" to "κομοδίνο",
80     "curtain" to "κουρτίνα",
81     "book" to "βιβλίο",
82     "clock" to "ρολόι"
83 )
84 }
85 }
```

Εικόνα 6.36: Hashmap μεταφράσεων

### 6.3.4 Περιγραφή Κλάσης “MainActivity”

Η κλάση “MainActivity” είναι υπεύθυνη για την αρχική ρύθμιση της εφαρμογής. Κάνει inflate το αρχείο “activity\_main.xml” στον φάκελο layout, έτσι ώστε να δημιουργήσει το layout και να το τοποθετήσει στην οθόνη. Επίσης η μεταβλητή τύπου ActivityMainBinding χρησιμοποιείται για τη σύνδεση του activity με το layout της (activity\_main.xml).

```

1  package org.tensorflow.lite.examples.objectdetection
2
3  import android.os.Bundle
4  import androidx.appcompat.app.AppCompatActivity
5  import org.tensorflow.lite.examples.objectdetection.databinding.ActivityMainBinding
6
7  class MainActivity : AppCompatActivity() {
8
9      private lateinit var activityMainBinding: ActivityMainBinding
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         activityMainBinding = ActivityMainBinding.inflate(layoutInflater)
14         setContentView(activityMainBinding.root)
15     }
16 }

```

Εικόνα 6.37: MainActivity

### 6.3.5 Περιγραφή Κλάσης “ObjectDetectorHelper”

Η κλάση ObjectDetectorHelper παρέχει μεθόδους για την αρχικοποίηση, ρυθμίσεις και χρήση ενός ανιχνευτή αντικειμένων. Χρησιμοποιεί ObjectDetector για να ανιχνεύσει αντικείμενα σε εικόνες και ενημερώνει ένα αντικείμενο DetectorListener για σφάλματα και αποτελέσματα.

**Δήλωση Πακέτου και Imports (Γραμμές 1-13):** Δήλωση πακέτου και βιβλιοθηκών που χρησιμοποιούνται για τη λειτουργία της κλάσης.

```

1  package org.tensorflow.lite.examples.objectdetection
2
3  import android.content.Context
4  import android.graphics.Bitmap
5  import android.os.SystemClock
6  import android.util.Log
7  import org.tensorflow.lite.gpu.CompatibilityList
8  import org.tensorflow.lite.support.image.ImageProcessor
9  import org.tensorflow.lite.support.image.TensorImage
10 import org.tensorflow.lite.support.image.ops.Rot90Op
11 import org.tensorflow.lite.task.core.BaseOptions
12 import org.tensorflow.lite.task.vision.detector.Detection
13 import org.tensorflow.lite.task.vision.detector.ObjectDetector
14

```

Εικόνα 6.38: Imports

**Δήλωση Κλάσης και Μεταβλητών (Γραμμές 15-24):**

- **threshold (προεπιλογή 0.5f):** Ένα float που αντιπροσωπεύει το ελάχιστο σκορ που πρέπει να ξεπεράσει ένα αντικείμενο για να θεωρηθεί έγκυρη ανίχνευση (αποσιώπηση δεκαδικών).
- **numThreads (προεπιλογή 2):** Ένας ακέραιος που ορίζει τον αριθμό των νημάτων που θα χρησιμοποιηθούν για την ανίχνευση αντικειμένων.
- **maxResults (προεπιλογή 1):** Ένας ακέραιος που ορίζει τον μέγιστο αριθμό αντικειμένων που θα μπορούν να ανιχνευθούν.
- **currentDelegate (προεπιλογή 0):** Ένας ακέραιος που καθορίζει την τρέχουσα επιλογή Delegate που θα χρησιμοποιηθεί για την εκτέλεση του μοντέλου ανίχνευσης (CPU, GPU ή NNAPI).
- **currentModel (προεπιλογή MobileNetV1):** Ένας ακέραιος που καθορίζει το τρέχον μοντέλο ανίχνευσης που θα χρησιμοποιηθεί.
- **context:** Αντικείμενο τύπου Context που παρέχει πρόσβαση στο περιβάλλον της εφαρμογής.
- **objectDetector:** Αντικείμενο τύπου ObjectDetector που αντιπροσωπεύει τον ανιχνευτή αντικειμένων (αρχικά null).
- **objectDetectorListener (προαιρετικό):** Αντικείμενο που υλοποιεί τη διασύνδεση DetectorListener και ενημερώνεται για σφάλματα και αποτελέσματα ανίχνευσης.

```

15 class ObjectDetectorHelper(
16     var threshold: Float = 0.5f,
17     var numThreads: Int = 2,
18     var maxResults: Int = 1,
19     var currentDelegate: Int = 0,
20     var currentModel: Int = 0,
21     val context: Context,
22     val objectDetectorListener: DetectorListener?
23 ) {
24     private var objectDetector: ObjectDetector? = null
25

```

Εικόνα 6.39: Μεταβλητές

**Μέθοδος setupObjectDetector() (Γραμμές 38-85):** Αυτή η μέθοδος είναι υπεύθυνη για την αρχικοποίηση του ανιχνευτή αντικειμένων. Εκτελεί τα ακόλουθα βήματα:

**Δημιουργία Επιλογών Ανίχνευσης:**

- Χρησιμοποιεί τον ObjectDetector.ObjectDetectorOptions.builder για να δημιουργήσει ένα builder επιλογών ανίχνευσης.
- Ορίζει το όριο σκορ (threshold) και τον μέγιστο αριθμό αποτελεσμάτων (maxResults) στο builder.
- Χτίζει και αποθηκεύει τις επιλογές ανίχνευσης.

**Δημιουργία Βασικών Επιλογών:**

- Χρησιμοποιεί τον `BaseOptions.builder` για να δημιουργήσει ένα builder βασικών επιλογών.
- Ορίζει τον αριθμό των νημάτων (`numThreads`) στο builder.
- Χτίζει και αποθηκεύει τις βασικές επιλογές.

#### Επιλογή Delegate:

- Ελέγχει την τιμή του `currentDelegate`:
  - Αν `currentDelegate` είναι 0 (CPU), δεν γίνεται κάποια ενέργεια.
  - Αν `currentDelegate` είναι 1 (GPU):
    - Ελέγχει αν η συσκευή υποστηρίζει GPU με `CompatibilityList`.
      - Αν υποστηρίζεται, ορίζει χρήση GPU στις βασικές επιλογές.
      - Διαφορετικά, ενημερώνει τον listener για μη υποστήριξη GPU.
  - Αν `currentDelegate` είναι 2 (NNAPI), ορίζει χρήση NNAPI στις βασικές επιλογές.

#### Ρύθμιση Βασικών Επιλογών στις Επιλογές Ανίχνευσης:

- Ορίζει τις βασικές επιλογές στις επιλογές ανίχνευσης.

#### Επιλογή Αρχείου Μοντέλου:

- Ελέγχει την τιμή του `currentModel` και επιλέγει το αντίστοιχο όνομα αρχείου μοντέλου.

#### Δημιουργία Ανιχνευτή:

- Προσπαθεί να δημιουργήσει ένα αντικείμενο `ObjectDetector` από το αρχείο μοντέλου και τις επιλογές ανίχνευσης.

#### Διαχείριση Σφαλμάτων:

- Σε περίπτωση σφάλματος κατά τη δημιουργία του ανιχνευτή:
  - Ενημερώνει τον listener για το σφάλμα.
  - Καταγράφει το μήνυμα σφάλματος στο log.

```
38 fun setupObjectDetector() {
39     // Create the base options for the detector using specifies max results and score threshold
40     val optionsBuilder =
41         ObjectDetector.ObjectDetectorOptions.builder()
42             .setScoreThreshold(threshold)
43             .setMaxResults(maxResults)
44
45     // Set general detection options, including number of used threads
46     val baseOptionsBuilder = BaseOptions.builder().setNumThreads(numThreads)
47
48     // Use the specified hardware for running the model. Default to CPU
49     when (currentDelegate) {
50         DELEGATE_CPU -> {
51             // Default
52         }
53         DELEGATE_GPU -> {
54             if (CompatibilityList().isDelegateSupportedOnThisDevice) {
55                 baseOptionsBuilder.useGpu()
56             } else {
57                 objectDetectorListener?.onError(error: "GPU is not supported on this device")
58             }
59         }
60     }
61 }
```

Εικόνα 6.40: Μέθοδος `setupObjectDetector`

```

65     optionsBuilder.setBaseOptions(baseOptionsBuilder.build())
66
67     val modelName =
68         when (currentModel) {
69             MODEL_MOBILENETV1 -> "mobilenetv1.tflite"
70             MODEL_EFFICIENTDET0 -> "efficientdet-lite0.tflite"
71             MODEL_EFFICIENTDET1 -> "efficientdet-lite1.tflite"
72             MODEL_EFFICIENTDET2 -> "efficientdet-lite2.tflite"
73             else -> "mobilenetv1.tflite"
74         }
75
76     try {
77         objectDetector =
78             ObjectDetector.createFromFileAndOptions(context, modelName, optionsBuilder.build())
79     } catch (e: IllegalStateException) {
80         objectDetectorListener?.onError(
81             error: "Object detector failed to initialize. See error logs for details"
82         )
83         Log.e(tag: "Test", msg: "TFLite failed to load model with error: " + e.message)
84     }
85 }

```

Εικόνα 6.41: Μέθοδος setupObjectDetector

**Μέθοδος detect(Γραμμές 87-114):**

Αυτή η μέθοδος πραγματοποιεί την ανίχνευση αντικειμένων σε μια εικόνα. Εκτελεί τα ακόλουθα βήματα:

**Έλεγχος Ανιχνευτή:**

- Ελέγχει αν το αντικείμενο objectDetector είναι null. Αν ναι, καλεί τη setupObjectDetector() για να το αρχικοποιήσει.

**Υπολογισμός Χρόνου Εκτέλεσης:**

- Αποθηκεύει τον τρέχοντα χρόνο συστήματος (SystemClock.uptimeMillis) για να υπολογίσει τον χρόνο εκτέλεσης της ανίχνευσης.

**Δημιουργία Προεπεξεργαστή:**

- Δημιουργεί έναν επεξεργαστή εικόνας (ImageProcessor) με χρήση ImageProcessor.Builder:
  - Προσθέτει έναν επεξεργαστή Rot90Op για περιστροφή της εικόνας κατά 90 μοίρες (ανάλογα με την τιμή του imageRotation).
- Χτίζει και αποθηκεύει τον επεξεργαστή εικόνας.

**Προεπεξεργασία Εικόνας:**

- Μετατρέπει την εικόνα (image) σε TensorImage χρησιμοποιώντας TensorImage.fromBitmap.
- Προεπεξεργάζεται την εικόνα με τον επεξεργαστή που δημιουργήθηκε στο προηγούμενο βήμα.

**Ανίχνευση Αντικειμένων:**

- Χρησιμοποιεί το αντικείμενο `objectDetector` για να ανιχνεύσει αντικείμενα στην προεπεξεργασμένη εικόνα (`tensorImage`).
- Αποθηκεύει τα αποτελέσματα ανίχνευσης σε μια μεταβλητή `results`.

#### Υπολογισμός Χρόνου Εκτέλεσης:

- Υπολογίζει τον χρόνο εκτέλεσης της ανίχνευσης αφαιρώντας τον αρχικά αποθηκευμένο χρόνο συστήματος από τον τρέχοντα χρόνο.

#### Ενημέρωση Listener:

- Ενημερώνει το αντικείμενο `objectDetectorListener` (αν υπάρχει) για τα αποτελέσματα:
  - Παρέχει τη λίστα αποτελεσμάτων (`results`).
  - Παρέχει τον υπολογισμένο χρόνο εκτέλεσης (`inferenceTime`).
  - Παρέχει το ύψος και το πλάτος της εικόνας (`imageHeight` και `imageWidth`).

```
87 fun detect(image: Bitmap, imageRotation: Int) {
88     if (objectDetector == null) {
89         setupObjectDetector()
90     }
91
92     // Inference time is the difference between the system time at the start and finish of the
93     // process
94     var inferenceTime = SystemClock.uptimeMillis()
95
96     // Create preprocessor for the image.
97     // See https://www.tensorflow.org/lite/inference\_with\_metadata/
98     //     lite_support#imageprocessor_architecture
99     val imageProcessor =
100         ImageProcessor.Builder() ImageProcessor.Builder
101             .add(Rot90p( k: -imageRotation / 90)) ImageProcessor.Builder!
102             .build()
103
104     // Preprocess the image and convert it into a TensorImage for detection.
105     val tensorImage = imageProcessor.process(TensorImage.fromBitmap(image))
106
107     val results = objectDetector?.detect(tensorImage)
108     inferenceTime = SystemClock.uptimeMillis() - inferenceTime
109     objectDetectorListener?.onResults(
110         results,
111         inferenceTime,
112         tensorImage.height,
113         tensorImage.width)
114 }
```

Εικόνα 6.42: Μέθοδος detect

**Διασύνδεση DetectorListener (Γραμμές 116-124):**

Αυτή η διασύνδεση ορίζει δύο μεθόδους που πρέπει να υλοποιήσει ένα αντικείμενο που θέλει να λαμβάνει ενημερώσεις για σφάλματα και αποτελέσματα ανίχνευσης:

**onError(error: String):**

- Καλείται όταν προκύψει σφάλμα κατά την αρχικοποίηση ή τη χρήση του ανιχνευτή.
- Η παράμετρος error περιέχει το μήνυμα σφάλματος.

**onResults(results: MutableList<Detection>?, inferenceTime: Long, imageHeight: Int, imageWidth: Int):**

- Καλείται όταν ολοκληρωθεί η ανίχνευση αντικειμένων σε μια εικόνα.
- Η παράμετρος results περιέχει τη λίστα αποτελεσμάτων ανίχνευσης (μπορεί να είναι null).
- Η παράμετρος inferenceTime περιέχει τον χρόνο εκτέλεσης της ανίχνευσης.
- Η παράμετρος imageHeight περιέχει το ύψος της εικόνας.
- Η παράμετρος imageWidth περιέχει το πλάτος της εικόνας.

```

116  ↓  interface DetectorListener {
117  ↓      fun onError(error: String)
118  ↓      fun onResults(
119          results: MutableList<Detection>?,
120          inferenceTime: Long,
121          imageHeight: Int,
122          imageWidth: Int
123      )
124  ↓  }

```

Εικόνα 6.43: Διασύνδεση DetectorListener

**Companion Object(γραμμές ):**

Η κλάση ObjectDetectorHelper διαθέτει επίσης ένα companion object που ορίζει σταθερές για χρήση σε όλο τον κώδικα της κλάσης:

- DELEGATE\_CPU (σταθερά τιμή: 0): Αναπαριστά τη χρήση της CPU για την επιλογή Delegate
- DELEGATE\_GPU (σταθερά τιμή: 1): Αναπαριστά τη χρήση της GPU για την επιλογή Delegate (εφόσον υποστηρίζεται από τη συσκευή).
- DELEGATE\_NNAPI (σταθερά τιμή: 2): Αναπαριστά τη χρήση του NNAPI (Neural Network API) για την επιλογή Delegate (εφόσον υποστηρίζεται από τη συσκευή).

- MODEL\_MOBILENETV1 (σταθερά τιμή: 0): Ορίζει ότι το αρχείο μοντέλου ανίχνευσης αντικειμένων "mobilenetv1.tflite" θα είναι ίσο με 0.
- MODEL\_EFFICIENTDET0 (σταθερά τιμή: 1): Ορίζει ότι αρχείο μοντέλου ανίχνευσης αντικειμένων "efficientdet-lite0.tflite" θα είναι ίσο με 1.
- MODEL\_EFFICIENTDET1 (σταθερά τιμή: 2): Ορίζει ότι αρχείο μοντέλου ανίχνευσης αντικειμένων "efficientdet-lite1.tflite" θα είναι ίσο με 2.
- MODEL\_EFFICIENTDET2 (σταθερά τιμή: 3): Ορίζει ότι αρχείο μοντέλου ανίχνευσης αντικειμένων "efficientdet-lite2.tflite" θα είναι ίσο με 3.

```

126 companion object {
127     const val DELEGATE_CPU = 0
128     const val DELEGATE_GPU = 1
129     const val DELEGATE_NNAPI = 2
130     const val MODEL_MOBILENETV1 = 0
131     const val MODEL_EFFICIENTDET0 = 1
132     const val MODEL_EFFICIENTDET1 = 2
133     const val MODEL_EFFICIENTDET2 = 3
134 }
135 }

```

Εικόνα 6.44: Companion Object

### 6.3.6 Περιγραφή Κλάσης “OverlayView”

Η κλάση OverlayView είναι μια προσαρμοσμένη κλάση τύπου View. Η κύρια λειτουργία της είναι να σχεδιάζει πλαίσια οριοθέτησης (bounding boxes) και επικαλύψεις κειμένου πάνω σε εικόνες που εμφανίζονται στην οθόνη. Επιπλέον, διαθέτει λειτουργίες για την επεξεργασία των αγγιγμάτων του χρήστη και την παροχή απτικής και ηχητικής ανατροφοδότησης.

**Δήλωση Πακέτου, Imports, Κλάσης, Μεταβλητών (γραμμές 1-39):** Δήλωση του πακέτου στο οποίο ανήκει η κλάση, δήλωση των απαραίτητων βιβλιοθηκών που θα χρησιμοποιηθούν, δήλωση κλάσης και δήλωση των παρακάτω μεταβλητών.

**results:** Μια λίστα αντικειμένων τύπου Detection, η οποία αποθηκεύει τα αποτελέσματα του αλγορίθμου ανίχνευσης αντικειμένων.

**boxPaint:** Ένα αντικείμενο τύπου Paint που χρησιμοποιείται για να σχεδιάσουμε τα πλαίσια οριοθέτησης (bounding boxes) γύρω από τα ανιχνευμένα αντικείμενα.

**textBackgroundPaint:** Ένα αντικείμενο τύπου Paint που χρησιμοποιείται για να σχεδιάσουμε το φόντο πίσω από το κείμενο που εμφανίζεται δίπλα στα ανιχνευμένα αντικείμενα.

**textPaint:** Ένα αντικείμενο τύπου Paint που χρησιμοποιείται για να σχεδιάσουμε το ίδιο το κείμενο που εμφανίζεται δίπλα στα ανιχνευμένα αντικείμενα.

**scaleFactor:** Μια μεταβλητή τύπου float που χρησιμοποιείται για να κλιμακώσουμε τα πλαίσια οριοθέτησης και το κείμενο ανάλογα με το μέγεθος της εικόνας.

**bounds:** Ένα αντικείμενο τύπου Rect που χρησιμοποιείται για να αποθηκεύσουμε τις διαστάσεις του πλαισίου οριοθέτησης του κειμένου.

**isTrackingFinger:** Μια μεταβλητή τύπου boolean που υποδεικνύει εάν ο χρήστης αγγίζει την οθόνη.

**initialTouchX:** Μια μεταβλητή τύπου float που αποθηκεύει την αρχική συντεταγμένη X του αγγίγματος του χρήστη.

**initialTouchY:** Μια μεταβλητή τύπου float που αποθηκεύει την αρχική συντεταγμένη Y του αγγίγματος του χρήστη.

**textToSpeech:** Ένα αντικείμενο τύπου TextToSpeech που χρησιμοποιείται για την προφορική περιγραφή των ανιχνευμένων αντικειμένων.

**HaveSpoken:** Μια μεταβλητή τύπου boolean που παρακολουθεί εάν έχει ήδη γίνει προφορική περιγραφή για το τρέχον ανιχνευμένο αντικείμενο.

**previousDetectedClass:** Μια μεταβλητή τύπου String που αποθηκεύει την ονομασία της κατηγορίας του τελευταία ανιχνευμένου αντικειμένου.

**vibrationPattern:** Ένας πίνακας τύπου long που ορίζει τον τρόπο δόνησης της συσκευής.

**vibrationEffect:** Ένα αντικείμενο τύπου VibrationEffect που δημιουργείται από το vibrationPattern.

```

1   package org.tensorflow.lite.examples.objectdetection
2
3   import android.content.Context
4   import android.graphics.Canvas
5   import android.graphics.Color
6   import android.graphics.Paint
7   import android.graphics.Rect
8   import android.graphics.RectF
9   import android.os.Build
10  import android.os.VibrationEffect
11  import android.os.Vibrator
12  import android.os.VibratorManager
13  import android.speech.tts.TextToSpeech
14  import android.util.AttributeSet
15  import android.view.View
16  import androidx.core.content.ContextCompat
17  import java.util.LinkedList
18  import kotlin.math.max
19  import org.tensorflow.lite.task.vision.detector.Detection
20  import android.view.MotionEvent
21  import android.widget.Toast
22  import android.util.Log
23
24  class OverlayView(context: Context?, attrs: AttributeSet?) : View(context, attrs) {

```

Εικόνα 6.45: Imports

```

26     private var results: List<Detection> = LinkedList<Detection>()
27     private var boxPaint = Paint()
28     private var textBackgroundPaint = Paint()
29     private var textPaint = Paint()
30     private var scaleFactor: Float = 1f
31     private var bounds = Rect()
32     private var isTrackingFinger = false
33     private var initialTouchX: Float = 0.0f
34     private var initialTouchY: Float = 0.0f
35     private var textToSpeech: TextToSpeech? = null
36     private var HaveSpoken = false
37     var previousDetectedClass: String? = null
38     val vibrationPattern = LongArrayOf(0, 10)
39     val vibrationEffect = VibrationEffect.createWaveform(vibrationPattern, repeat: -1)

```

Εικόνα 6.46: Μεταβλητές

**Μέθοδος clear (γραμμές 47-54):** Η μέθοδος clear επαναφέρει τα πεδία boxPaint, textBackgroundPaint, και textPaint στις προεπιλεγμένες ρυθμίσεις τους και καλεί την invalidate για να ανανεώσει την προβολή.

```

47     fun clear() {
48         textPaint.reset()
49         textBackgroundPaint.reset()
50         boxPaint.reset()
51         invalidate()
52         initPaints()
53     }

```

Εικόνα 6.47: Μέθοδος clear

**Μέθοδος initPaints (γραμμές 55-67):** Η μέθοδος initPaints ορίζει τις ρυθμίσεις των πεδίων boxPaint, textBackgroundPaint, και textPaint.

- **boxPaint:** Ορίζεται το χρώμα, το πάχος, και το στυλ γραμμής για το πλαίσιο οριοθέτησης.
- **textBackgroundPaint:** Ορίζεται το χρώμα και το στυλ γραμμής για το φόντο του κειμένου.
- **textPaint:** Ορίζεται το χρώμα, το μέγεθος, και το στυλ γραμμής για το ίδιο το κείμενο.

```

55     private fun initPaints() {
56         textBackgroundPaint.color = Color.BLACK
57         textBackgroundPaint.style = Paint.Style.FILL
58         textBackgroundPaint.textSize = 50f
59
60         textPaint.color = Color.WHITE
61         textPaint.style = Paint.Style.FILL
62         textPaint.textSize = 50f
63
64         boxPaint.color = ContextCompat.getColor(context!!, R.color.bounding_box_color)
65         boxPaint.strokeWidth = 8f
66         boxPaint.style = Paint.Style.STROKE
67     }

```

Εικόνα 6.48: Μέθοδος initPaints

**Μέθοδος draw (γραμμές 69-99):** Η μέθοδος draw σχεδιάζει τα ανιχνευμένα αντικείμενα και τις περιγραφές τους στην οθόνη.

#### Σχεδίαση πλαισίων οριοθέτησης:

- Για κάθε ανιχνευμένο αντικείμενο :
  - Λαμβάνονται οι συντεταγμένες του πλαισίου οριοθέτησης (γραμμές 47-48)
  - Κλιμακώνονται οι συντεταγμένες σύμφωνα με τον παράγοντα scaleFactor.
  - Σχεδιάζεται το πλαίσιο οριοθέτησης στην οθόνη

#### Σχεδίαση κειμένου:

- Για κάθε ανιχνευμένο αντικείμενο:
  - Λαμβάνεται η κατηγορία και η βαθμολογία του αντικειμένου
  - Δημιουργείται η περιγραφή
  - Υπολογίζονται οι διαστάσεις του πλαισίου κειμένου
  - Σχεδιάζεται το κείμενο

```

69     override fun draw(canvas: Canvas) {
70         super.draw(canvas)
71
72         for (result in results) {
73             val boundingBox = result.boundingBox
74
75             val top = boundingBox.top * scaleFactor
76             val bottom = boundingBox.bottom * scaleFactor
77             val left = boundingBox.left * scaleFactor
78             val right = boundingBox.right * scaleFactor
79
80             // Draw bounding box around detected objects
81             val drawableRect = RectF(left, top, right, bottom)
82             canvas.drawRect(drawableRect, boxPaint)
83
84             // Create text to display alongside detected objects
85             val drawableText =
86                 result.categories[0].label + " " +
87                 String.format("%.2f", result.categories[0].score)
88

```

Εικόνα 6.49: Μέθοδος draw

```
89     // Draw rect behind display text
90     textBackgroundPaint.getTextBounds(drawableText, start:0, drawableText.length, bounds)
91     val textWidth = bounds.width()
92     val textHeight = bounds.height()
93     canvas.drawRect(left, top, right:left + textWidth + Companion.BOUNDING_RECT_TEXT_PADDING,
94                     bottom:top + textHeight + Companion.BOUNDING_RECT_TEXT_PADDING, textBackgroundPaint)
95
96     // Draw text for detected object
97     canvas.drawText(drawableText, left, y:top + bounds.height(), textPaint)
98 }
99 }
```

Εικόνα 6.50: Μέθοδος draw

### Μέθοδος setResult (γραμμές 101-110):

#### Ενημέρωση αποτελεσμάτων:

- Λαμβάνει η λίστα ανιχνευμένων αντικειμένων.
- Αποθηκεύει τα αποτελέσματα στη λίστα results

#### Υπολογισμός παράγοντα κλιμάκωσης:

- Λαμβάνει το ύψος και το πλάτος της εικόνας
- Υπολογίζει τον παράγοντα κλιμάκωσης
- Αποθηκεύει τον παράγοντα κλιμάκωσης στο scaleFactor

```
101     fun setResult(
102         detectionResults: MutableList<Detection>,
103         imageHeight: Int,
104         imageWidth: Int,
105     ) {
106         results = detectionResults
107
108
109         scaleFactor = max(a.width * 1f / imageWidth, b.height * 1f / imageHeight)
110     }
```

Εικόνα 6.51: Μέθοδος setResult

**Μέθοδος onTouchEvent (γραμμές 117-178):** Η μέθοδος onTouchEvent χειρίζεται τα αγγίγματα στην οθόνη και αναπτύχθηκε από τον συγγραφέα.

#### Παρακολούθηση αγγίγματος:

- Ανιχνεύει την κατάσταση του αγγίγματος
- Ελέγχει εάν ο χρήστης αγγίζει ένα ανιχνευμένο αντικείμενο
  - Εάν ναι, ενεργοποιεί την δόνηση
  - Ελέγχει εάν η ονομασία της κατηγορίας έχει αλλάξει
    - Εάν ναι, πραγματοποιεί προφορική περιγραφή
    - Ενημερώνει την μεταβλητή HaveSpoken
- Ενημερώνει την μεταβλητή isTrackingFinger

```

117 override fun onTouchEvent(event: MotionEvent): Boolean {
118     when (event.action) {
119         MotionEvent.ACTION_DOWN -> {
120             isTrackingFinger = true
121             initialTouchX = event.x
122             initialTouchY = event.y
123         }
124
125
126
127         MotionEvent.ACTION_MOVE -> {
128             if (isTrackingFinger) {
129                 val touchX = event.x
130                 val touchY = event.y

```

Εικόνα 6.52: Μέθοδος onTouchEvent

```

132     var objectDetectedInsideBox = false
133     for (detection in results) {
134         val (scaledLeft, scaledTop, scaledRight, scaledBottom) =
135             calculateScaledBoundingBox(detection.boundingBox)
136         if (isTouchWithinBoundingBox(
137             scaledLeft,
138             scaledTop,
139             scaledRight,
140             scaledBottom,
141             touchX,
142             touchY
143         )) {
144             objectDetectedInsideBox = true
145             val currentDetectedClass = detection.categories[0].label
146
147             triggerVibration()
148         }

```

Εικόνα 6.53: Μέθοδος onTouchEvent

```

151         if (previousDetectedClass != currentDetectedClass && !HaveSpoken) {
152             speakText(currentDetectedClass)
153             HaveSpoken = true
154
155             triggerVibration()
156
157         }
158
159         triggerVibration()
160         break
161     }
162 }
163
164 // Reset HaveSpoken if finger moves outside all bounding boxes
165 if (!objectDetectedInsideBox) {
166     HaveSpoken = false
167 }
168 }
169 }

```

Εικόνα 6.54: Μέθοδος onTouchEvent

```

171         MotionEvent.ACTION_UP, MotionEvent.ACTION_CANCEL -> {
172             isTrackingFinger = false
173             HaveSpoken = false
174             previousDetectedClass = null
175         }
176     }
177     return true
178 }

```

Εικόνα 6.55: Μέθοδος onTouchEvent

**Μέθοδος calculateScaledBoundingBox(γραμμές 181-187):** Η μέθοδος calculateScaledBoundingBox κλιμακώνει τις συντεταγμένες του πλαισίου οριοθέτησης σύμφωνα με τον παράγοντα scaleFactor.

```

181     private fun calculateScaledBoundingBox(boundingBox: RectF): List<Float> {
182         val top = boundingBox.top * scaleFactor
183         val bottom = boundingBox.bottom * scaleFactor
184         val left = boundingBox.left * scaleFactor
185         val right = boundingBox.right * scaleFactor
186         return listOf(left, top, right, bottom)
187     }

```

Εικόνα 6.56: Μέθοδος calculateScaledBoundingBox

**Μέθοδος isTouchWithinBoundingBox (γραμμές 189-198):** Η μέθοδος isTouchWithinBoundingBox γράφτηκε από τον συγγραφέα, έτσι ώστε να ελέγχει εάν το άγγιγμα του χρήστη βρίσκεται μέσα σε ένα πλαίσιο οριοθέτησης.

```

189     private fun isTouchWithinBoundingBox(
190         left: Float,
191         top: Float,
192         right: Float,
193         bottom: Float,
194         touchX: Float,
195         touchY: Float
196     ): Boolean {
197         return touchX >= left && touchX <= right && touchY >= top && touchY <= bottom
198     }

```

Εικόνα 6.57: Μέθοδος isTouchWithinBoundingBox

**Μέθοδος triggerVibration(γραμμές 200-211):**

Η μέθοδος triggerVibration γράφτηκε από τον συγγραφέα και είναι υπεύθυνη για την απτική ανατροφοδότηση μέσω των δονήσεων.

**Ελεγχος έκδοσης Android:**

- Χρησιμοποιεί Build.VERSION.SDK\_INT για να ελέγξει την έκδοση Android

**Android 31+:**

- Χρησιμοποιεί VibratorManager και VibrationEffect για προηγμένη δόνηση

**Android 26 και παλαιότερα:**

- Χρησιμοποιεί Vibrator για βασική δόνηση

```

200     private fun triggerVibration() {
201
202         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
203             val vibratorManager =
204                 context.getSystemService(Context.VIBRATOR_MANAGER_SERVICE) as VibratorManager
205             val vibrator = vibratorManager.getDefaultVibrator()
206             vibrator.vibrate(vibrationEffect)
207         } else {
208             val vibrator = context.getSystemService(Context.VIBRATOR_SERVICE) as Vibrator
209             vibrator.vibrate(vibrationEffect)
210         }
211     }

```

Εικόνα 6.58: Μέθοδος triggerVibration

**Μέθοδος `initTextToSpeech`(γραμμές 213-223):** Η μέθοδος `initTextToSpeech` αρχικοποιεί το αντικείμενο `TextToSpeech`.

- Δημιουργεί ένα αντικείμενο `TextToSpeech`
- Ορίζει ένα listener για να ελέγξει εάν η αρχικοποίηση ήταν επιτυχής

```

213 private fun initTextToSpeech() {
214     textToSpeech = TextToSpeech(context!!, object : TextToSpeech.OnInitListener {
215         override fun onInit(status: Int) {
216             if (status == TextToSpeech.SUCCESS) {
217                 } else {
218                     Toast.makeText(context, "Text-to-Speech failed to initialize",
219                         Toast.LENGTH_SHORT).show()
220                 }
221             }
222         })
223     }

```

Εικόνα 6.59: Μέθοδος `initTextToSpeech`

**Μέθοδος `onDetachedFromWindow` (γραμμές 225-228):** Η μέθοδος `onDetachedFromWindow` καλείται όταν η `OverlayView` αποσυνδέεται από το παράθυρο.

- Ελευθερώνει τους πόρους του `TextToSpeech` (γραμμή 261).

```

225 override fun onDetachedFromWindow() {
226     super.onDetachedFromWindow()
227     textToSpeech?.shutdown()
228 }

```

Εικόνα 6.60: Μέθοδος `onDetachedFromWindow`

**Μέθοδος `speakText`(γραμμές 230-249):** Η μέθοδος `speakText` πραγματοποιεί προφορική περιγραφή του αντικειμένου που έχει ανιχνευθεί και αναπτύχθηκε από τον συγγραφέα της παρούσης πτυχικής.

#### Λήψη μετάφρασης:

- +Η λήψη μετάφρασης γίνεται από την κλάση “`coco_translations`”, από τον `hashmap translations`.

#### Έλεγχος μετάφρασης:

- Εάν η μετάφραση βρεθεί
  - Πραγματοποιεί προφορική περιγραφή στα ελληνικά
- Εάν η μετάφραση δεν βρεθεί
  - Προαιρετικά, πραγματοποιεί προφορική περιγραφή στα αγγλικά
  - Καταγράφει μήνυμα σφάλματος

```

230 private fun speakText(text: String) {
231     val translations = coco_translations.CocoTranslations.translations
232
233
234     val key = text.toLowerCase()
235
236     if (translations.containsKey(key)) {
237         val greekTranslation = translations[key]!!
238         if (textToSpeech?.speak(greekTranslation, TextToSpeech.QUEUE_FLUSH, params: null) != TextToSpeech.ERROR) {
239
240         } else {
241
242             Toast.makeText(context, text="Text-to-Speech failed to speak", Toast.LENGTH_SHORT).show()
243         }
244     } else {
245
246         Log.w(tag="OverlayView", msg="No Greek translation found for: $text")
247     }
248 }
249 }

```

Εικόνα 6.61: Μέθοδος speakText

### 6.3.7 Περιγραφή Script “activity\_main.xml”

Το script “activity\_main.xml” σχεδιάζει την διάταξη της οθόνης με το κύριο fragment της κάμερας.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.coordinatorlayout.widget.CoordinatorLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     android:background="@android:color/transparent"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent">
9
10     <RelativeLayout
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:orientation="vertical">
14
15         <androidx.fragment.app.FragmentContainerView
16             android:id="@+id/fragment_container"
17             android:name="androidx.navigation.fragment.NavHostFragment"
18             android:layout_width="match_parent"
19             android:layout_height="match_parent"
20             android:background="@android:color/transparent"
21             android:keepScreenOn="true"
22             app:defaultNavHost="true"
23             app:navGraph="@navigation/nav_graph"
24             android:layout_marginTop="?android:attr/actionBarSize"
25             tools:context=".MainActivity"/>
26     </RelativeLayout>
27 </androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Εικόνα 6.62: Activity\_main.xml

### 6.3.8 Περιγραφή Script “fragment\_camera.xml”

Το συγκεκριμένο script δημιουργεί μια οθόνη κάμερας στην εφαρμογή. Η οθόνη εμφανίζει την προεπισκόπηση της κάμερας και χρησιμοποιεί ένα προσαρμοσμένο στοιχείο (OverlayView) για να σχεδιάσει οπτικά στοιχεία πάνω στην προεπισκόπηση που υποδεικνύουν τα ανιχνευμένα αντικείμενα. Επιπλέον, περιλαμβάνει ένα πίνακα ρυθμίσεων στο κάτω μέρος της οθόνης σχετικά με την ανίχνευση αντικειμένων.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      android:id="@+id/camera_container"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8      <androidx.camera.view.PreviewView
9          android:id="@+id/view_finder"
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         app:scaleType="fillStart" />
13
14         <org.tensorflow.lite.examples.objectdetection.OverlayView
15             android:id="@+id/overlay"
16             android:layout_height="match_parent"
17             android:layout_width="match_parent" />
18
19         <include
20             android:id="@+id/bottom_sheet_layout"
21             layout="@layout/info_bottom_sheet" />
22     </androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Εικόνα 6.63: fragment\_camera.xml

### 6.3.9 Περιγραφή Script “info\_bottom\_sheet.xml”

Το παρον xml αρχείο δημιουργεί τον συρόμενο πίνακα ρυθμίσεων στο κάτω μέρος της οθόνης. Ο πίνακας επιτρέπει στο χρήστη να προσαρμόσει διάφορες ρυθμίσεις σχετικά με την επεξεργασία και τα αποτελέσματα της ανίχνευσης αντικειμένων, καθώς και να επιλέξει delegate και μοντέλο για την εφαρμογή.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.core.widget.NestedScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/bottom_sheet_layout"
6      android:layout_width="match_parent"
7      android:layout_gravity="center_horizontal"
8      android:background="@color/bottom_sheet_background"
9      android:gravity="center_horizontal"
10     android:orientation="vertical"
11     android:padding="@dimen/bottom_sheet_padding"
12     android:layout_height="wrap_content"
13     app:behavior_hideable="false"
14     app:behavior_peekHeight="@dimen/bottom_sheet_peek_height"
15     app:layout_behavior="com.google.android.material.bottomsheet.BottomSheetBehavior">
16

```

Εικόνα 6.64: Info\_bottom\_sheet.xml

```

17     <androidx.appcompat.widget.LinearLayoutCompat
18         android:orientation="vertical"
19         android:layout_width="match_parent"
20         android:layout_height="match_parent">
21
22         <!-- Chevron indicating that the bottom sheet is expandable -->
23         <RelativeLayout
24             android:layout_height="wrap_content"
25             android:layout_width="match_parent">
26             <ImageView
27                 android:layout_width="wrap_content"
28                 android:layout_height="wrap_content"
29                 android:layout_centerHorizontal="true"
30                 android:layout_centerVertical="true"
31                 android:src="@drawable/icn_chevron_up"
32                 android:contentDescription="@string/alt_bottom_sheet_chevron" />
33         </RelativeLayout>

```

Εικόνα 6.65: Info\_bottom\_sheet.xml

```

35     <!-- Inference time row -->
36     <androidx.appcompat.widget.LinearLayoutCompat
37         android:layout_width="match_parent"
38         android:layout_height="wrap_content"
39         android:layout_marginTop="@dimen/bottom_sheet_default_row_margin"
40         android:orientation="horizontal">
41         <TextView
42             android:id="@+id/inference_time_label"
43             android:text="@string/label_inference_time"
44             android:textColor="@color/bottom_sheet_text_color"
45             android:layout_width="wrap_content"
46             android:layout_height="wrap_content"
47             android:layout_centerVertical="true"
48             android:textSize="@dimen/bottom_sheet_text_size" />
49         <TextView
50             android:id="@+id/inference_time_val"
51             android:text="0ms"
52             android:layout_width="match_parent"
53             android:layout_height="wrap_content"
54             android:gravity="end"
55             android:layout_centerVertical="true"
56             android:textColor="@color/bottom_sheet_text_color"
57             android:textSize="@dimen/bottom_sheet_text_size" />
58     </androidx.appcompat.widget.LinearLayoutCompat>

```

Εικόνα 6.66: Info\_bottom\_sheet.xml

```

60     <!-- ML confidence threshold adjustment row -->
61     <RelativeLayout
62         android:layout_width="match_parent"
63         android:layout_height="wrap_content"
64         android:layout_marginTop="@dimen/bottom_sheet_default_row_margin"
65         android:gravity="center"
66         android:orientation="horizontal">
67
68         <TextView
69             android:layout_width="wrap_content"
70             android:layout_height="wrap_content"
71             android:text="@string/label_confidence_threshold"
72             android:layout_centerVertical="true"
73             android:textSize="@dimen/bottom_sheet_text_size"
74             android:textColor="@color/bottom_sheet_text_color" />
75         <LinearLayout
76             android:layout_width="wrap_content"
77             android:layout_height="wrap_content"
78             android:layout_alignParentRight="true"
79             android:gravity="center_vertical"
80             android:orientation="horizontal"
81         >

```

Εικόνα 6.67: Info\_bottom\_sheet.xml

```

83     <androidx.appcompat.widget.AppCompatImageButton
84         android:id="@+id/threshold_minus"
85         android:layout_width="@dimen/bottom_sheet_control_btn_size"
86         android:layout_height="@dimen/bottom_sheet_control_btn_size"
87         android:contentDescription="@string/alt_bottom_sheet_threshold_button_minus"
88         android:src="@drawable/ic_minus" />
89
90     <TextView
91         android:id="@+id/threshold_value"
92         android:layout_width="wrap_content"
93         android:layout_height="wrap_content"
94         android:layout_marginLeft="@dimen/bottom_sheet_control_text_side_margin"
95         android:layout_marginRight="@dimen/bottom_sheet_control_text_side_margin"
96         android:gravity="center"
97         android:minEms="@integer/bottom_sheet_control_text_min_ems"
98         android:text="0.50"
99         android:textColor="@color/bottom_sheet_text_color"
100        android:textSize="@dimen/bottom_sheet_text_size" />

```

Εικόνα 6.68: Info\_bottom\_sheet.xml

```

101
102     <androidx.appcompat.widget.AppCompatImageButton
103         android:id="@+id/threshold_plus"
104         android:layout_width="@dimen/bottom_sheet_control_btn_size"
105         android:layout_height="@dimen/bottom_sheet_control_btn_size"
106         android:contentDescription="@string/alt_bottom_sheet_threshold_button_plus"
107         android:src="@drawable/ic_plus" />
108     </LinearLayout>
109 </RelativeLayout>
110

```

Εικόνα 6.69: Info\_bottom\_sheet.xml

```

111 <!-- ML max results adjustment row -->
112 <RelativeLayout
113     android:layout_width="match_parent"
114     android:layout_height="wrap_content"
115     android:layout_marginTop="@dimen/bottom_sheet_default_row_margin"
116     android:gravity="center"
117     android:orientation="horizontal">
118
119     <TextView
120         android:layout_width="wrap_content"
121         android:layout_height="wrap_content"
122         android:text="@string/label_max_results"
123         android:textSize="@dimen/bottom_sheet_text_size"
124         android:layout_centerVertical="true"
125         android:textColor="@color/bottom_sheet_text_color" />
126

```

Εικόνα 6.70: Info\_bottom\_sheet.xml

```

127 <LinearLayout
128     android:layout_width="wrap_content"
129     android:layout_height="wrap_content"
130     android:layout_alignParentRight="true"
131     android:gravity="center"
132     android:orientation="horizontal">
133     <androidx.appcompat.widget.AppCompatImageButton
134         android:id="@+id/max_results_minus"
135         android:layout_width="@dimen/bottom_sheet_control_btn_size"
136         android:layout_height="@dimen/bottom_sheet_control_btn_size"
137         android:contentDescription="@string/alt_bottom_sheet_max_results_button_minus"
138         android:src="@drawable/ic_minus" />

```

Εικόνα 6.71: Info\_bottom\_sheet.xml

```

139 <TextView
140     android:id="@+id/max_results_value"
141     android:layout_width="wrap_content"
142     android:layout_height="wrap_content"
143     android:minEms="@integer/bottom_sheet_control_text_min_ems"
144     android:gravity="center"
145     android:layout_marginLeft="@dimen/bottom_sheet_control_text_side_margin"
146     android:layout_marginRight="@dimen/bottom_sheet_control_text_side_margin"
147     android:text="1"
148     android:textColor="@color/bottom_sheet_text_color"
149     android:textSize="@dimen/bottom_sheet_text_size" />
150 <androidx.appcompat.widget.AppCompatImageButton
151     android:id="@+id/max_results_plus"
152     android:layout_width="@dimen/bottom_sheet_control_btn_size"
153     android:layout_height="@dimen/bottom_sheet_control_btn_size"
154     android:contentDescription="@string/alt_bottom_sheet_max_results_button_plus"
155     android:src="@drawable/ic_plus" />
156 </LinearLayout>

```

Εικόνα 6.72: Info\_bottom\_sheet.xml

```

157 </RelativeLayout>
158
159 <!-- Number of threads adjustment row -->
160 <RelativeLayout
161     android:layout_width="match_parent"
162     android:layout_height="wrap_content"
163     android:layout_marginTop="@dimen/bottom_sheet_default_row_margin"
164     android:gravity="center"
165     android:orientation="horizontal">
166
167     <TextView
168         android:layout_width="wrap_content"
169         android:layout_height="wrap_content"
170         android:text="@string/label_threads"
171         android:layout_centerVertical="true"
172         android:textSize="@dimen/bottom_sheet_text_size"
173         android:textColor="@color/bottom_sheet_text_color" />
174 </LinearLayout>

```

Εικόνα 6.73: Info\_bottom\_sheet.xml

```

175         android:layout_width="wrap_content"
176         android:layout_height="wrap_content"
177         android:layout_alignParentRight="true"
178         android:gravity="center_vertical"
179         android:orientation="horizontal"
180     >
181
182     <androidx.appcompat.widget.AppCompatImageButton
183         android:id="@+id/threads_minus"
184         android:layout_width="@dimen/bottom_sheet_control_btn_size"
185         android:layout_height="@dimen/bottom_sheet_control_btn_size"
186         android:contentDescription="@string/alt_bottom_sheet_thread_button_minus"
187         android:src="@drawable/ic_minus" />
188
189     <TextView
190         android:id="@+id/threads_value"
191         android:layout_width="wrap_content"
192         android:layout_height="wrap_content"

```

Εικόνα 6.74: Info\_bottom\_sheet.xml

```

193         android:layout_marginLeft="@dimen/bottom_sheet_control_text_side_margin"
194         android:layout_marginRight="@dimen/bottom_sheet_control_text_side_margin"
195         android:gravity="center"
196         android:minEms="@integer/bottom_sheet_control_text_min_ems"
197         android:text="2"
198         android:textColor="@color/bottom_sheet_text_color"
199         android:textSize="@dimen/bottom_sheet_text_size" />
200
201     <androidx.appcompat.widget.AppCompatImageButton
202         android:id="@+id/threads_plus"
203         android:layout_width="@dimen/bottom_sheet_control_btn_size"
204         android:layout_height="@dimen/bottom_sheet_control_btn_size"
205         android:contentDescription="@string/alt_bottom_sheet_thread_button_plus"
206         android:src="@drawable/ic_plus" />
207 </LinearLayout>
208 </RelativeLayout>

```

Εικόνα 6.75: Info\_bottom\_sheet.xml

```

210 <!-- Delegate selection row -->
211 <RelativeLayout
212     android:layout_width="match_parent"
213     android:layout_height="wrap_content"
214     android:layout_marginTop="@dimen/bottom_sheet_default_row_margin">
215
216     <TextView
217         android:text="@string/label_delegate"
218         android:layout_width="wrap_content"
219         android:layout_height="wrap_content"
220         android:textSize="@dimen/bottom_sheet_text_size"
221         android:textColor="@color/bottom_sheet_text_color" />
222
223     <androidx.appcompat.widget.AppCompatSpinner
224         android:id="@+id/spinner_delegate"
225         android:layout_width="wrap_content"
226         android:layout_height="wrap_content"
227         android:minWidth="@dimen/bottom_sheet_spinner_delegate_min_width"

```

Εικόνα 6.76: Info\_bottom\_sheet.xml

```

228         android:spinnerMode="dropdown"
229         android:theme="@style/BottomSheetSpinnerItemStyle"
230         android:layout_alignParentRight="true"
231         android:entries="@array/delegate_spinner_titles"/>
232
233     </RelativeLayout>

```

Εικόνα 6.77: Info\_bottom\_sheet.xml

```

235     <!-- Model selection row -->
236     <RelativeLayout
237         android:layout_width="match_parent"
238         android:layout_height="wrap_content"
239         android:layout_marginTop="@dimen/bottom_sheet_default_row_margin">
240
241         <TextView
242             android:text="@string/label_models"
243             android:layout_width="wrap_content"
244             android:layout_height="wrap_content"
245             android:textSize="@dimen/bottom_sheet_text_size"
246             android:textColor="@color/bottom_sheet_text_color" />

```

Εικόνα 6.78: Info\_bottom\_sheet.xml

```

248         <androidx.appcompat.widget.AppCompatSpinner
249             android:id="@+id/spinner_model"
250             android:layout_width="wrap_content"
251             android:layout_height="wrap_content"
252             android:minWidth="@dimen/bottom_sheet_spinner_model_min_width"
253             android:spinnerMode="dropdown"
254             android:theme="@style/BottomSheetSpinnerItemStyle"
255             android:layout_alignParentRight="true"
256             android:entries="@array/models_spinner_titles"/>
257
258     </RelativeLayout>
259 </androidx.appcompat.widget.LinearLayoutCompat>
260 </androidx.core.widget.NestedScrollView>

```

Εικόνα 6.79: Info\_bottom\_sheet.xml

## 6.4 Επίλογος

Στο παρόν κεφάλαιο έγινε ανάλυση της διεπαφής και του κώδικα της εφαρμογής. Είδαμε τους τρόπους με τους οποίους η εφαρμογή μπορεί να βοηθήσει συνανθρώπους μας που αντιμετωπίζουν προβλήματα όρασης, δηλαδή την φωνητική περιγραφή των αντικειμένων καθώς και την απτική ανατροφοδότηση. Στην συνέχεια παρουσιάστηκε διεξοδικά ο κώδικας της εφαρμογής

## Κεφάλαιο 7ο: Αξιολόγηση

### 7.1 Εισαγωγή

Στο παρόν κεφάλαιο θα δούμε το πως διεξήχθη η αξιολόγηση της εφαρμογής με βάση ένα ερωτηματολόγιο και κάποια στατιστικά στοιχεία τα οποία εξάχθηκαν από ένα δείγμα ανθρώπων, οι οποίοι μετά από την δοκιμή της εφαρμογής κλήθηκαν να απαντήσουν στις ερωτήσεις.

### 7.2 Ερωτηματολόγιο

Οι ερωτήσεις αφορούν την εμπειρία χρήσης της εφαρμογής και είναι κομμάτι από ένα ερωτηματολόγιο για εφαρμογές σε κινητά τηλέφωνα [18], το οποίο χωρίζεται στις παρακάτω ενότητες.

- Δημογραφικά Στοιχεία
- Χρησιμότητα
- Ευκολία Χρήσης
- Ευκολία Μάθησης
- Ικανοποίηση
- Μελλοντική Βελτίωση

#### 7.2.1 Δημογραφικά Στοιχεία

##### 1. Ηλικία

- Κάτω των 18
- 18-30
- 30-50
- 50+

##### 2. Φύλλο

- Άνδρας
- Γυναίκα

#### 7.2.2 Χρησιμότητα

Πόσο χρήσιμη θεωρείτε την εφαρμογή για την καθημερινή σας ζωή; (1 = Πολύ Άχρηστη, 5 = Πολύ Χρήσιμη)

- 1
- 2
- 3
- 4
- 5

Κάνει όλα όσα περίμενα να κάνει η εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**Ανταποκρίνεται στις ανάγκες μου η εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)**

- 1
- 2
- 3
- 4
- 5

### **7.2.3 Ευκολία Χρήσης**

**Η εφαρμογή είναι εύχρηστη και κατανοητή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)**

- 1
- 2
- 3
- 4
- 5

**Η εφαρμογή είναι φιλική προς τον χρήστη; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)**

- 1
- 2
- 3
- 4
- 5

**Ήταν εύκολο να βρείτε και να χρησιμοποιήσετε τις λειτουργίες της εφαρμογής; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)**

- 1
- 2
- 3
- 4
- 5

**Η εφαρμογή είναι εύκολη στην πλοήγηση; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)**

- 1
- 2
- 3
- 4
- 5

**Η έκδοση της εφαρμογής, η οποία είχε την λειτουργία δόνησης μου ήταν ποιο εύκολη στην χρήση;**

- Ναι
- Όχι

### 7.2.4 Ευκολία Μάθησης

Πόσο εύκολο ήταν να μάθετε να χρησιμοποιείτε την εφαρμογή; (1 = Πολύ Δύσκολο, 5 = Πολύ Εύκολο)

- 1
- 2
- 3
- 4
- 5

Πόσο χρόνο χρειαστήκατε για να μάθετε τα βασικά της εφαρμογής; (1 = Πολύ Χρόνο, 5 = Πολύ Λίγο Χρόνο)

- 1
- 2
- 3
- 4
- 5

Θυμάμαι εύκολα πώς να χρησιμοποιώ την εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

### 7.2.5 Ικανοποίηση

Πόσο ικανοποιημένοι είστε από την συνολική εμπειρία χρήσης της εφαρμογής; (1 = Πολύ Δυσανεστημένος, 5 = Πολύ Ικανοποιημένος)

- 1
- 2
- 3
- 4
- 5

Το μοτίβο δόνησης μου ήταν ευχάριστο; (1 = Πολύ Δυσανεστημένος, 5 = Πολύ Ικανοποιημένος)

- 1
- 2
- 3
- 4
- 5

Ποια έκδοση μου ήταν ποιο ευχάριστη, αυτή με ή χωρίς την δόνηση;

- Με την δόνηση
- Χωρίς την δόνηση

Ποια έκδοση θεωρείτε καλύτερη αυτή με ή χωρίς την δόνηση;

- Με την δόνηση
- Χωρίς την δόνηση

### 7.2.6 Μελλοντική Βελτίωση

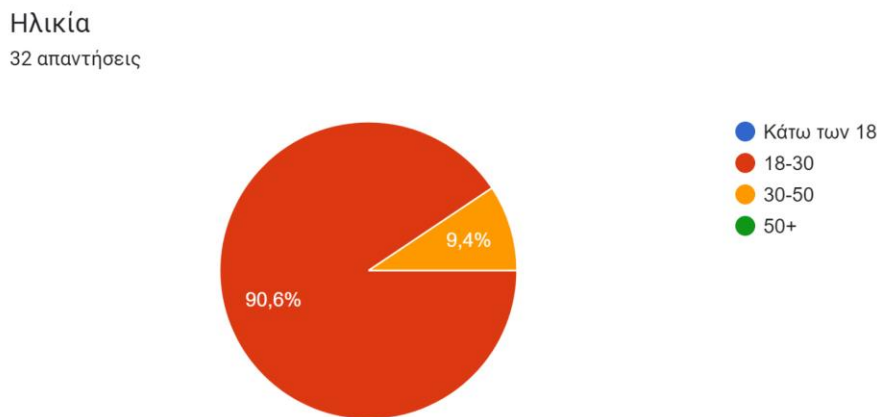
Πώς θα μπορούσαμε να βελτιώσουμε την εφαρμογή μελλοντικά;

### 7.3 Στατιστικά Στοιχεία

Στην παρούσα υποενότητα θα δούμε ορισμένα στατιστικά στοιχεία τα οποία αντλήσαμε από την αξιολόγηση, με σκοπό να δούμε την εμπειρία χρήσης της εφαρμογής και τι μπορούμε να βελτιώσουμε μελλοντικά.

#### 7.3.1 Δημογραφικά Στοιχεία

Για την αξιολόγηση της εφαρμογής ερωτήθηκαν 32 άτομα εκ των οποίων το 90% περίπου ήταν 18-30 και το 9% ήταν 30-50 ετών όπως φαίνεται και στο διάγραμμα 6.1:

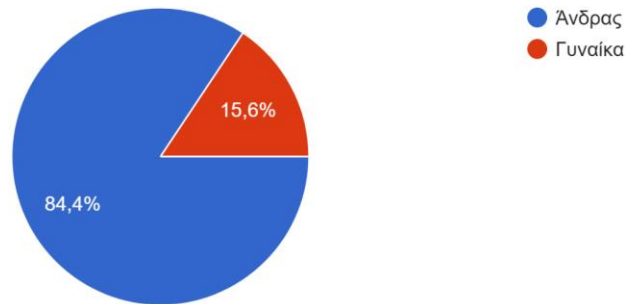


Διάγραμμα 7.1: Διάγραμμα με το ποσοστό ηλικίας.

Στην συνέχεια οι συμμετέχοντες ερωτήθηκαν για το φύλλο, με το 84% περίπου του δείγματος να αποτελείται από άντρες και το 15% περίπου να αποτελείται από γυναίκες σύμφωνα με το διάγραμμα 6.2 παρακάτω:

Φύλλο

32 απαντήσεις



Διάγραμμα 7.2: Διάγραμμα με το ποσοστό φύλλου.

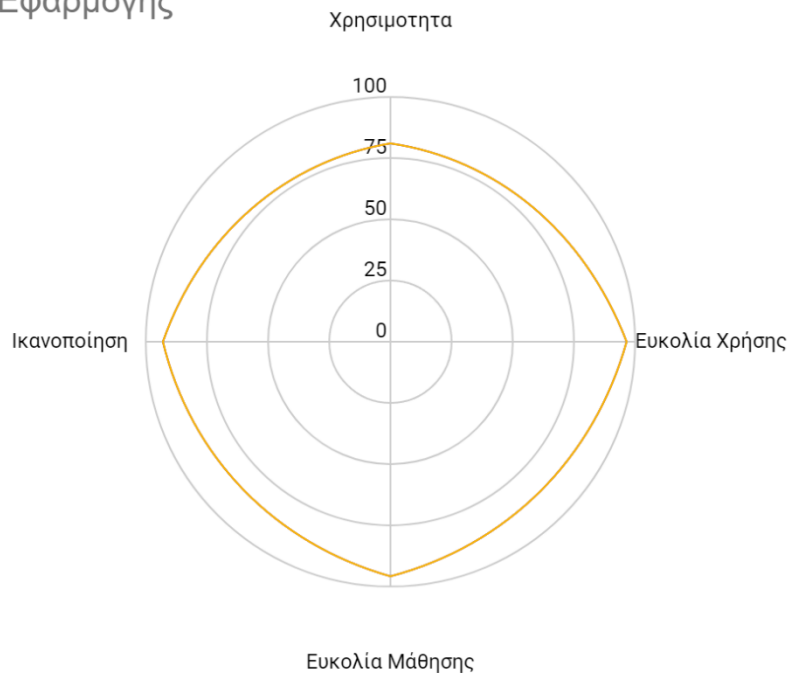
### 7.3.2 Αραχνοειδές Διάγραμμα

Για τις 4 επόμενες κατηγορίες δημιουργήθηκε ένα διάγραμμα τύπου αράχνης, το οποίο αποτελείται από τον μέσο όρο των απαντήσεων των 4 επόμενων κατηγοριών οι οποίες είναι οι εξής:

- Χρησιμότητα
- Ευκολία Χρήσης
- Ευκολία Μάθησης
- Ικανοποίηση

Οι συγκεκριμένες ερωτήσεις ήταν πολλαπλών επιλογών με χαμηλότερη βαθμολογία το 1 και υψηλότερη βαθμολογία το 5, έτσι έχουμε τον τελικό μέσο όρο κάθε κατηγορίας όπως φαίνεται στο παρακάτω διάγραμμα 6.3:

## Αξιολόγηση Εφαρμογής



Διάγραμμα 7.3: Αραχνοειδές διάγραμμα με τις τέσσερις κατηγορίες ερωτήσεων της αξιολόγησης της εφαρμογής.

### 7.3.3 Συγκεντρωτικός Πίνακας Αξιολόγησης Ερωτηματολογίου

Με βάση τα στοιχεία, τα οποία συλλέξαμε από την αξιολόγηση της εφαρμογής προκύπτει ο παρακάτω συγκεντρωτικός πίνακας 7.1

Πίνακας 7.1: Συγκεντρωτικός πίνακας απαντήσεων ερωτηματολογίου

Τιμές	Φύλλο		Γενικό σύνολο
	Άνδρας	Γυναίκα	
Πόσο χρήσιμη θεωρείτε την εφαρμογή για την καθημερινή σας ζωή; (1 = Πολύ Άχρηστη, 5 = Πολύ Χρήσιμη)	4,33	4,00	4,28
Κάνει όλα όσα περίμενα να κάνει η εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,38	4,20	4,35
Ανταποκρίνεται στις ανάγκες μου η εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,15	2,00	3,81
Η εφαρμογή είναι εύχρηστη και κατανοητή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,78	4,60	4,75
Η εφαρμογή είναι φιλική προς των χρήστη; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,85	4,60	4,81
Ήταν εύκολο να βρείτε και να χρησιμοποιήσετε τις λειτουργίες της εφαρμογής; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,85	4,60	4,81
Η εφαρμογή είναι εύκολη στην πλοήγηση; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,93	4,60	4,88
Πόσο εύκολο ήταν να μάθετε να χρησιμοποιείτε την εφαρμογή; (1 = Πολύ Δύσκολο, 5 = Πολύ Εύκολο)	4,70	4,80	4,72

Θυμάμαι εύκολα πώς να χρησιμοποιώ την εφαρμογή; (1 = Διαφωνώ Απόλυτα, 5 = Συμφωνώ Απόλυτα)	4,89	5,00	4,91
Πόσο ικανοποιημένοι είστε από την συνολική εμπειρία χρήσης της εφαρμογής; (1 = Πολύ Δυσανεστημένος, 5 = Πολύ Ικανοποιημένος)	4,78	4,20	4,69
Το μοτίβο δόνησης μου ήταν ευχάριστο; (1 = Πολύ Δυσανεστημένος, 5 = Πολύ Ικανοποιημένος)	4,59	4,80	4,63

Παρόλο που το δείγμα μας είναι άνισο, λόγω του μεγαλύτερου πλήθους ανδρών μπορούμε με βάση τον πίνακα 7.1 να βγάλουμε κάποια συμπεράσματα. Βλέπουμε ότι σε γενικές γραμμές το ανδρικό φύλλο έμεινε περισσότερο ευχαριστημένο όσον αφορά την γενική εμπειρία της εφαρμογής, σε σχέση με το γυναικείο. Παρόλα αυτά από ότι μας έδειξε ο πίνακας το μοτίβο δόνησης ήταν πιο αρεστό στο γυναικείο φύλλο.

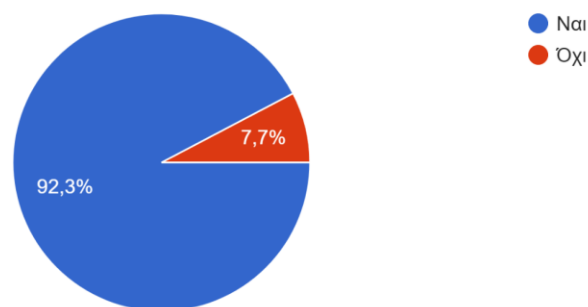
### 7.3.4 Μελλοντική Βελτίωση

Στην ερώτηση για μελλοντική βελτίωση είχαμε μία μόνο απάντηση, οι οποία ήταν η μελλοντική επέκταση της λίστας των ανιχνεύσιμων αντικειμένων από το μοντέλο αναγνώρισης και κατ' επέκταση η περαιτέρω εκπαίδευση του.

## 7.4 Έρευνα χρησιμότητας της Απτικής Αλληλεπίδρασης

Πέραν της αξιολόγησης της εφαρμογής έγινε και μία έρευνα σε 13 άτομα για το κατά πόσο χρήσιμη είναι η απτική ανατροφοδότηση την οποία προσθέσαμε στην εφαρμογή. Τα αποτελέσματα της έρευνας έδειξαν ότι η εφαρμογή είναι πιο εύκολη στην χρήση με την απτική ανατροφοδότηση, με συντριπτική πλειοψηφία 92% να θεωρεί την χρήση της εφαρμογής ευκολότερη όπως δείχνει και το διάγραμμα 6.4 παρακάτω:

Η έκδοση της εφαρμογής, η οποία είχε την λειτουργία δόνησης μου ήταν ποιο εύκολη στην χρήση;  
13 απαντήσεις

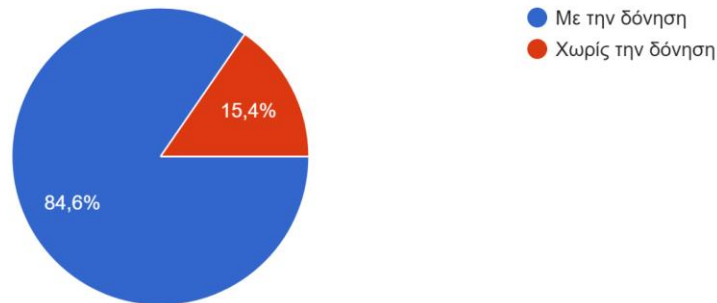


Διάγραμμα 7.4: Διάγραμμα το οποίο δείχνει την προτίμηση των ερωτηθέντων εάν η απτική ανατροφοδότηση μέσω της δόνησης κάνει την εφαρμογή πιο εύκολη στην χρήση.

## Κεφάλαιο 7

Στην συνέχεια οι συμμετέχοντες ερωτήθηκαν εάν η απτική ανατροφοδότηση τους ήταν ευχάριστη, πάλι η συντριπτική πλειοψηφία απάντησε θετικά όπως φαίνεται και στο διάγραμμα:

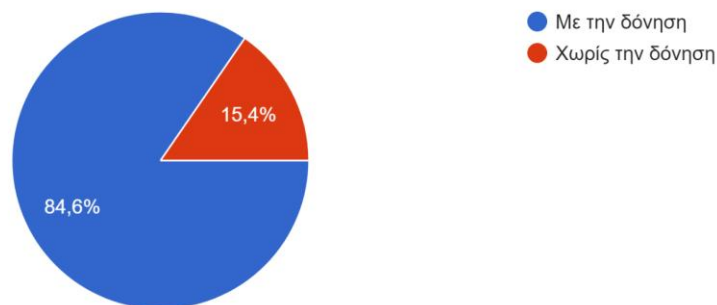
Ποια έκδοση μου ήταν ποιο ευχάριστη, αυτή με ή χωρίς την δόνηση;  
13 απαντήσεις



Διάγραμμα 7.5: Διάγραμμα το οποίο δείχνει εάν η απτική ανατροφοδότηση ήταν ευχάριστη.

Τέλος οι συμμετέχοντες ερωτήθηκαν για το αν θεωρούν την εφαρμογή καλύτερη με ή χωρίς απτική ανατροφοδότηση και η πλειοψηφία απάντησε ότι η εφαρμογή είναι καλύτερη με απτική ανατροφοδότηση όπως φαίνεται στο παρακάτω διάγραμμα:

Ποια έκδοση θεωρείτε καλύτερη αυτή με ή χωρίς την δόνηση  
13 απαντήσεις



Διάγραμμα 7.6: Διάγραμμα το οποίο δείχνει την προτίμηση των ερωτηθέντων εάν κάνει καλύτερη την εφαρμογή η απτική ανατροφοδότηση.

### 7.5 Επίλογος

Στο παρόν κεφάλαιο παρατέθηκε η αξιολόγηση της εφαρμογής με 18 ερωτήσεις σε ένα δείγμα 32 ατόμων, και διεξήχθη μία έρευνα για την χρησιμότητα της απτικής αλληλεπίδρασης μέσω της δόνησης σε ένα δείγμα 13 ατόμων, της οποίας έρευνας τα αποτελέσματα έδειξαν ότι η απτική ανατροφοδότηση κάνει την εφαρμογή ευκολότερη στην χρήση με την συντριπτική πλειοψηφία να απαντάει θετικά σε αυτή.

## Κεφάλαιο 8ο: Συμπεράσματα

### 8.1 Συμπεράσματα

Στην εποχή μας η εξέλιξη της τεχνολογίας καλπάζει με άφταστους ρυθμούς, κάθε μέρα δημιουργούνται καινοτόμες τεχνολογίες και οι απαρχαιωμένες τεχνολογίες αργά ή γρήγορα ξεπερνιούνται. Πλέον η καθημερινότητα μας ολοένα και ψηφιοποιείται. Για αυτόν λοιπόν τον λόγο και στην παρούσα εργασία είδαμε την γέννηση ιστορικά της μεγάλης αυτής καινοτομίας της τεχνητής νοημοσύνης. Μελετήσαμε και αναλύσαμε τον κλάδο της τεχνητής νοημοσύνης, ο οποίος απασχολεί σε μεγάλο βαθμό την επιστημονική κοινότητα και ονομάζεται μηχανική μάθηση. Επιπλέον εξερευνήσαμε και την βαθιά μάθηση, η οποία είναι υποπεδίο της μηχανικής μάθησης. Έπειτα κάναμε μια βουτιά στον κόσμο των αισθήσεων και εξερευνήσαμε την αίσθηση της αφής, βλέποντας τους τρόπους με τους οποίους μπορούμε να την αξιοποιήσουμε σε διάφορες εφαρμογές, όπως επίσης είδαμε και μερικούς τομείς στους οποίους αξιοποιείται. Ακόμα είδαμε πως με την αίσθηση της αφής μπορούμε να φτιάξουμε εφαρμογές και όχι μόνο, με σκοπό να κάνουμε καλύτερη την ζωή των συνανθρώπων μας οι οποίοι αντιμετωπίζουν οπτικές δυσκολίες.

Επιπρόσθετα εξερευνήσαμε τον τομέα της υποβοηθητικής τεχνολογίας ο οποίος έχει να προσφέρει πάρα πολλά σε αυτά τα άτομα βελτιώνοντας την ζωή τους κατά ένα ποσοστό και κάνοντας την ένταξη τους στην κοινωνία πολύ πιο εύκολη, μειώνοντας τον στιγματισμό που δέχονται αυτά τα άτομα λόγω του συμβατικού και αδιάκριτου άσπρου μαστουριού. Νέες καινοτομίες στον τομέα αυτόν φέρνουν, νέες ελπίδες στην βελτίωση του βιοτικού τους επιπέδου. Βλέπουμε ότι με την απτική αλληλεπίδραση και την τεχνητή νοημοσύνη μπορούν αυτά τα άτομα να ανεξαρτητοποιηθούν ακόμα περισσότερο και να έχουν μεγαλύτερη αυτονομία.

Ο συνδυασμός τεχνολογιών όπως αυτοί που προαναφέραμε έχουν να προσφέρουν πλήθος νέων δυνατοτήτων. Είδαμε διάφορες εφαρμογές όπως το *be my eyes*, οι οποίες ήδη έχουν βοηθήσει αυτά τα άτομα σε μεγάλο βαθμό, όπως επίσης τον ίδιο στόχο έχει και η παρούσα εργασία, δηλαδή την ανάπτυξη μιας εφαρμογής ανίχνευσης αντικειμένων, η οποία συνδυάζει την καινοτομία της τεχνητής νοημοσύνης με την πρωτοπόρο τεχνολογία της απτικής ανατροφοδότησης μέσω της δόνησης και της φωνητικής περιγραφής των ανιχνευμένων αντικειμένων, δημιουργώντας καθιστώντας τελικά την εφαρμογή όσο πιο φιλική γίνεται προς τους συνανθρώπους μας οι οποίοι αντιμετωπίζουν οπτικές δυσκολίες. Μετά την ολοκλήρωση της, η εφαρμογή πέρασε από μία αξιολόγηση, καθώς επίσης διεξήχθη και μία έρευνα με θέμα την χρησιμότητα της απτικής ανατροφοδότησης. Τα αποτελέσματα της αξιολόγησης έδειξαν ότι οι χρήστε έμειναν ευχαριστημένοι από την εμπειρία χρήσης της εφαρμογής. Τέλος τα αποτελέσματα της έρευνας για την χρησιμότητα της απτικής ανατροφοδότησης έδειξαν ότι είναι πολύ χρήσιμη καθώς η συντριπτική πλειοψηφία έδωσε θετική απάντηση. Εν κατακλείδι η αξιοποίηση της τεχνολογίας με ηθικό και υπεύθυνο τρόπο αποτελεί βασικό στόχο για τη δημιουργία ενός πιο δίκαιου και συμπεριληπτικού κόσμου για όλες τις ομάδες ανθρώπων ανεξαρτήτως διακρίσεων.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. Roser, «Our World in Data,» Global Change Data Lab, 2022. [Ηλεκτρονικό]. Available: <https://ourworldindata.org/brief-history-of-ai#article-citation>. [Πρόσβαση May 2024].
- [2] H. P. C. S. E. Sheikh, «Artificial Intelligence: Definition and Background. In: Mission AI. Research for Policy.,» σε *Artificial Intelligence: Definition and Background*, Springer, Cham, 2023, p. 15–41.
- [3] G. R. Jovel Juan, «Frontiers in Medicine,» 2021. [Ηλεκτρονικό]. Available: <https://www.frontiersin.org/articles/10.3389/fmed.2021.771607>. [Πρόσβαση 2024].
- [4] C. Z. P. & H. K. Janiesch, «Machine learning and deep learning,» *Electron Markets* 31, pp. 685-695, 2021.
- [5] S. A.-H. Asharul Islam Khan, «Machine Learning in Computer Vision,» *Procedia Computer Science*, pp. 1444-1451, 2020.
- [6] TensorFlow, «GitHub,» [Ηλεκτρονικό]. Available: [https://github.com/tensorflow/examples/tree/master/lite/examples/object\\_detection/android](https://github.com/tensorflow/examples/tree/master/lite/examples/object_detection/android). [Πρόσβαση 2024].
- [7] M. E.-S. D. Sinha, «"Thin MobileNet: An Enhanced MobileNet Architecture,"» New York, 2019.
- [8] T. S. M. Sreelakshmi, «Haptic Technology: A comprehensive review on its applications and future prospects,» *Materials Today: Proceedings*, pp. 4182-4187, 2017.
- [9] J. Corrigan, «ResearchGate,» 2015. [Ηλεκτρονικό]. Available: [https://www.researchgate.net/publication/280777071\\_Haptic\\_Designing\\_Assistive\\_Technology\\_for\\_People\\_who\\_are\\_Blind](https://www.researchgate.net/publication/280777071_Haptic_Designing_Assistive_Technology_for_People_who_are_Blind). [Πρόσβαση 2024].
- [10] A. R. J. A. G. C. a. K. C. See, «Touch, Texture and Haptic Feedback: A Review on How We Feel the World around Us,» 2022.
- [11] J. L. D. O. H. B. Lilit Hakobyan, «Mobile assistive technologies for the visually impaired,» σε *Survey of Ophthalmology*, 2013, pp. 513-528.
- [12] P. & G. M. & D. R. & G. A. & G. T. Kaur, «ResearchGate,» 2017. [Ηλεκτρονικό]. Available: [https://www.researchgate.net/publication/315830657\\_Be\\_My\\_Eyes\\_Android\\_App\\_for\\_visually\\_impaired\\_people](https://www.researchgate.net/publication/315830657_Be_My_Eyes_Android_App_for_visually_impaired_people). [Πρόσβαση 2024].
- [13] BeMyEyes, «Be my eyes,» 2015. [Ηλεκτρονικό]. Available: <https://www.bemyeyes.com/blog/announcing-be-my-ai>. [Πρόσβαση 2024].

- [14] T. A. R. M. P. Y. & B. V. Qureshi, «International Research Journal of Engineering and Technology,» 2021. [Ηλεκτρονικό]. Available: <https://www.irjet.net/archives/V8/i3/IRJET-V8I3554.pdf>. [Πρόσβαση 2024].
- [15] Y.-Y. L. C.-H. C. a. I.-F. W. Hsuan-Eng Chen, «ACM Digital Library,» 2015. [Ηλεκτρονικό]. Available: <https://doi.org/10.1145/2702613.2726953>. [Πρόσβαση 2024].
- [16] J. R. K. & Z. Z. Sun, «arXiv,» 2019. [Ηλεκτρονικό]. Available: <https://arxiv.org/abs/1909.05994>. [Πρόσβαση 2024].
- [17] T. e. a. Lin, «Microsoft COCO: Common Objects in Context,» σε *Lecture Notes in Computer Science, vol 8693.*, Springer, Cham, 2014, p. 740–755.
- [18] A. Lund, «ResearchGate,» 2001. [Ηλεκτρονικό]. Available: [https://www.researchgate.net/profile/Arnold-Lund/publication/230786746\\_Measuring\\_Usability\\_with\\_the\\_USE\\_Questionnaire/links/56e5a90e08ae98445c21561c/Measuring-Usability-with-the-USE-Questionnaire.pdf](https://www.researchgate.net/profile/Arnold-Lund/publication/230786746_Measuring_Usability_with_the_USE_Questionnaire/links/56e5a90e08ae98445c21561c/Measuring-Usability-with-the-USE-Questionnaire.pdf). [Πρόσβαση 2024].