



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΠΡΟΠΤΥΧΙΑΚΟ
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΚΕΝΤΡΟ ΕΡΕΥΝΑΣ ΓΙΑ
ΤΗΝ ΗΛΕΚΤΡΟΝΙΚΗ
ΔΙΑΚΥΒΕΡΝΗΣΗ



«ARGUS»: ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΣΩΠΩΝ ΓΙΑ ΤΗΝ ΕΛΛΗΝΙΚΗ ΑΣΤΥΝΟΜΙΑ

Δημήτριος Μάτσας
[321/2018129, matsasjim@gmail.com]

Επιβλέπων Καθηγητής: Γιάννης Χαραλαμπίδης

ΙΟΥΝΙΟΣ 2024
www.aegean.icsd.gr/is-lab

Η διπλωματική εργασία παρουσιάστηκε ενώπιον του Διδακτικού Προσωπικού του
Πανεπιστημίου Αιγαίου
Σε Μερική Εκπλήρωση των απαιτήσεων για την απόκτηση του διπλώματος του «Μηχανικού
Πληροφοριακών και Επικοινωνιακών Συστημάτων»

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ
ΕΠΙΚΥΡΩΝΕΙ ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ
ΜΑΤΣΑ ΔΗΜΗΤΡΙΟΥ

Γ. Χαραλαμπίδης, Επιβλέπων

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

Ε. Λουκής, Μέλος

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

Χ. Αλεξόπουλος, Μέλος

Επ. Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

© 2024

Με επιφύλαξη παντός δικαιώματος.

ΔΗΜΗΤΡΙΟΣ ΜΑΤΣΑΣ

Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Ευχαριστίες

Στον κ. Ιωάννη Χαραλαμπίδη για την μοναδική ευκαιρία που μου πρόσφερε για την ανάληψη του έργου αυτού.

Στους γονείς μου, τον Σταύρο και τον Βασίλη για την στήριξη τους όλα αυτά τα χρόνια.

Στους φίλους και συναδέλφους που βάδισαν μαζί μου στο ταξίδι αυτό.

Περίληψη

Η διπλωματική εργασία αφορά την ανάπτυξη ενός πρωτοποριακού λογισμικού υπολογιστικής όρασης για την Ελληνική αστυνομία. Το «Argus» είναι ένα σύστημα αναγνώρισης και εντοπισμού προσώπων καθώς και ανίχνευσης κίνησης, ανεπτυγμένο με τη βοήθεια συνελκτικών νευρωνικών δικτύων και αλγορίθμων. Σκοπός αυτής της εφαρμογής είναι η διευκόλυνση των ερευνών της αστυνομίας, χρησιμοποιώντας εργαλεία τελευταίας τεχνολογίας στην υπηρεσία του πολίτη. Στην εργασία που ακολουθεί, γίνεται μια αναφορά στις εισαγωγικές έννοιες που αφορούν τις τεχνολογίες που χρησιμοποιήθηκαν, δίνοντας ιδιαίτερη βάση στην έννοια του νευρωνικού δικτύου και στις εφαρμογές της υπολογιστικής όρασης. Σημαντικό κομμάτι αποτελεί επίσης η μελέτη παρόμοιων εφαρμογών της αγοράς που βασίζονται σε αλγόριθμους υπολογιστικής όρασης. Επιπλέον, παρουσιάζονται εκτενώς τα εργαλεία ανάπτυξης της εφαρμογής καθώς και η μεθοδολογία. Τέλος, η εργασία αυτή περιλαμβάνει και έναν οδηγό χρήσης, παρουσιάζοντας αναλυτικά τις λειτουργίες της εφαρμογής.

Λέξεις – Κλειδιά: υπολογιστική όραση, νευρωνικό δίκτυο, λογισμικό, αναγνώριση προσώπων

Abstract

«Argus»: A Facial Recognition System for the Hellenic Police

This thesis concerns the development of an innovative computer vision software for the Hellenic Police. "Argus" is a system for facial recognition and motion detection, developed with the help of convolutional neural networks and algorithms. The purpose of this application is to facilitate police investigations using state-of-the-art tools in the service of citizens. In this paper, introductory concepts related to the technologies used are discussed, with particular emphasis on neural networks and computer vision applications. An important part of the thesis is the study of similar market applications based on computer vision algorithms. Additionally, the application development tools and methodology are presented in detail. Finally, this paper includes a user guide, presenting the functions of the application in detail.

Keywords: computer vision, neural network, software, facial recognition

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	6
1. ΕΙΣΑΓΩΓΗ	11
1.1 Πλαίσιο	11
1.2 Σκοπός της Διπλωματικής Εργασίας.....	11
1.3 Αντικείμενο της Διπλωματικής Εργασίας	12
1.4 Δομή.....	12
2. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ	14
2.1 Νευρωνικά Δίκτυα	14
2.1.1 Ορισμός.....	14
2.1.2 Τελευταίες Εξελίξεις.....	14
2.1.3 Χρησιμότητα	15
2.2 Μηχανική Μάθηση	16
2.2.1 Ορισμός.....	16
2.2.2 Τελευταίες Εξελίξεις.....	17
2.2.3 Χρησιμότητα	18
2.3 Υπολογιστική Όραση	19
2.3.1 Ορισμός.....	19
2.3.2 Τελευταίες Εξελίξεις.....	19
2.3.3 Χρησιμότητα	20
2.4 Ανίχνευση Κίνησης.....	21
2.5 Ανίχνευση Προσώπων.....	22
2.6 Αναγνώριση Προσώπων.....	23
3. ΑΝΑΛΥΣΗ ΠΕΔΙΟΥ	25
3.1 Ανάλυση Κυριότερων Εφαρμογών	25
3.1.1 Clearview AI	25
3.1.2 Vigilant FaceSearch	26
3.1.3 Veritone IDentify.....	27
3.2 Συμπεράσματα Ανάλυσης	29
4. ΕΡΓΑΛΕΙΑ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ ΑΝΑΠΤΥΞΗΣ	30
4.1 Εργαλεία Ανάπτυξης.....	30
4.1.2 Qt Development Framework	30
4.1.2.1 Γενικές Πληροφορίες	30
4.1.2.2 Πως Χρησιμοποιήθηκε.....	30
4.1.2.3 Ο Μηχανισμός Slots and Signals.....	31
4.1.2.4 Κλάσεις και Βιβλιοθήκες που Χρησιμοποιήθηκαν	32
4.1.2.4.1 QAudioOutput	32
4.1.2.4.2 QWidget	32

4.1.2.4.3 QMainWindow	33
4.1.2.4.4 QVideoWidget.....	33
4.1.2.4.5 QMediaPlayer	33
4.1.2.4.6 QMessageBox.....	34
4.1.2.4.7 QMediaMetaData.....	34
4.1.2.4.8 QSettings	35
4.1.2.4.9 QFileDialog	35
4.1.2.4.10 QObject	36
4.1.2.4.11 QToolBar.....	36
4.1.2.4.12 QProgressBar.....	36
4.1.2.4.13 QPainter.....	36
4.1.2.4.14 QDir.....	37
4.1.2.4.15 QFile.....	37
4.1.2.4.16 QStringList	37
4.1.2.4.17 QTextStream.....	38
4.1.2.4.18 QToolButton	38
4.1.2.4.19 QPushButton.....	38
4.1.2.4.20 QSlider.....	38
4.1.2.4.21 QListView	39
4.1.2.4.22 QModelIndex	39
4.1.2.4.23 QResizeEvent.....	40
4.1.2.4.24 QTextEdit	40
4.1.2.4.25 QGraphicsPixmapItem	40
4.1.2.4.26 QHBoxLayout.....	40
4.1.2.4.27 QVBoxLayout.....	41
4.1.2.4.28 QLabel.....	41
4.1.2.4.29 QDate	41
4.1.2.4.30 QDateEdit.....	41
4.1.2.4.31 QIntValidator.....	41
4.1.2.4.32 QLineEdit	42
4.1.2.4.33 QComboBox	42
4.1.2.4.34 QTableWidget.....	42
4.1.2.4.35 QFrame	43
4.1.2.4.36 QPixmap.....	43
4.1.2.4.37 QFileInfo	43
4.1.2.4.38 QTabWidget	43
4.1.2.4.39 QCheckBox.....	44
4.1.2.4.40 QModelIndexList.....	44
4.1.2.4.41 QSpinBox	45

4.1.2.4.42 QThread.....	45
4.1.2.4.43 QValidator	46
4.1.2.4.44 QChar	47
4.1.2.4.45 QGraphicsView.....	47
4.1.2.4.46 QGraphicsScene.....	47
4.1.2.4.47 QDebug	47
4.1.2.4.48 QGroupBox.....	47
4.1.3 Γλώσσα Προγραμματισμού C++	47
4.1.3.1 Γιατί επιλέχθηκε η C++;.....	47
4.1.3.1.1 Σύγκριση μεταξύ C++ και Python	48
4.1.3.2 Γενικές Πληροφορίες	49
4.1.3.3 Microsoft Visual C++.....	50
4.1.3.4 Βιβλιοθήκες που Χρησιμοποιήθηκαν	50
4.1.3.4.1 vector.....	50
4.1.3.4.2 iostream	51
4.1.3.4.3 fstream.....	51
4.1.3.4.4 string.....	51
4.1.3.4.5 algorithm	52
4.1.4 Open Computer Vision Library	52
4.1.4.1 Γενικές Πληροφορίες	52
4.1.4.2 Πως Χρησιμοποιήθηκε.....	53
4.1.4.3 Αρχεία Μοντέλων Μηχανικής Μάθησης	53
4.1.4.4 Βιβλιοθήκες που Χρησιμοποιήθηκαν	54
4.1.4.4.1 dnn.....	54
4.1.4.4.2 core.....	55
4.1.4.4.3 FaceDetectorYN	56
4.1.4.4.4 FaceRecognizerSF	57
4.2 Πλατφόρμες Ανάπτυξης.....	58
4.2.1 Visual Studio 2022.....	58
4.2.1.1 Γενικές Πληροφορίες	59
4.2.1.2 Γιατί το Visual Studio;	59
4.2.1.3 Οδηγίες Εγκατάστασης Βιβλιοθηκών	60
4.2.2 GitHub.....	61
4.2.2.1 Γενικές Πληροφορίες	61
4.2.2.2 Χρήση του GitHub	61
5. ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ.....	62
5.1 Τρέχουσα Κατάσταση	62
5.2 Περιπτώσεις Χρήσης.....	62
5.2.1 Μελέτη Περίπτωσης Χρήσης: «Ανίχνευση Κίνησης».....	62

5.2.1.1 Περιγραφή	62
5.2.1.2 Διάγραμμα Περίπτωσης Χρήσης	63
5.2.2 Μελέτη Περίπτωσης Χρήσης: «Εισαγωγή Δεδομένων Υπόπτου στη Βάση»	64
5.2.2.1 Περιγραφή	64
5.2.2.2 Διάγραμμα Περίπτωσης Χρήσης	64
5.3 Απαιτήσεις Συστήματος	65
5.3.1 Λειτουργικές Απαιτήσεις	65
5.3.2 Μη Λειτουργικές Απαιτήσεις	66
6. ΜΕΘΟΔΟΛΟΓΙΑ.....	68
6.1 Αρχιτεκτονική Συστήματος	68
6.2 Μεθοδολογία Ανάπτυξης	69
6.3 Βήματα Ανάπτυξης	70
7. ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	72
7.1 Εισαγωγικές Πληροφορίες	72
7.2 Κεντρικό Παράθυρο	72
7.2.1 Video Player	72
7.2.2 Εργαλείο Προεπισκόπησης Εικόνων	73
7.2.3 Ημερολόγιο Γεγονότων.....	74
7.2.4 Γραμμή Εργαλείων	74
7.2.4.1 Προβολή Μεταδεδομένων.....	75
7.2.4.2 Αφαίρεση Φορτωμένου Αρχείου	75
7.2.4.3 Εκκαθάριση Ημερολογίου Γεγονότων	75
7.2.4.4 Εντοπισμός Κίνησης	75
7.2.4.5 Εντοπισμός Προσώπων	77
7.2.4.6 Αναγνώριση Προσώπων.....	78
7.3 Μενού: «Αρχείο»	79
7.3.1 Άνοιγμα.....	79
7.3.2 Έξοδος.....	80
7.4 Μενού: «Εργαλεία».....	80
7.4.1 Κατάλογος Ατόμων	80
7.4.2 Δεδομένα Φωτογραφιών	81
7.5 Μενού: «Ρυθμίσεις»	83
7.5.1 Ρυθμίσεις Νευρωνικού Δικτύου	83
7.5.1.1 Αρχείο Ονομάτων	84
7.5.1.2 Αρχείο Βαρών.....	84
7.5.1.3 Αρχείο Παραμετροποιήσεων	85
7.5.1.4 Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων	85
7.5.1.5 Ρυθμίσεις Μοντέλου Αναγνώρισης Προσώπων	85

7.5.2 Ρυθμίσεις Αποθήκευσης.....	86
7.6 Μενού: «Βοήθεια»	87
7.6.1 Σχετικά	87
7.6.1.1 Σχετικά με το OpenCV	87
7.6.1.2 Σχετικά με το Argus	87
7.6.2 Επικοινωνία	87
7.7 Οδηγίες Εγκατάστασης	88
8. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ.....	89
8.1 Συμπεράσματα	89
8.1.1 Σχετικά με τα εργαλεία	89
8.1.2 Σχετικά με το σύστημα	89
8.2 Αδυναμίες – Όρια του Συστήματος.....	90
8.3 Επόμενα Βήματα.....	91
ΠΑΡΑΡΤΗΜΑΤΑ.....	92
A. Ενδεικτικά Παραδείγματα Πηγαίου Κώδικα	92
A.1 Συνάρτηση Εντοπισμού Αντικειμένων	92
A.2 Κώδικας Δημιουργίας Παραθύρου Προειδοποίησης.....	92
A.3 Συνάρτηση Προσθήκης Ατόμου στην Βάση	93
B. Πίνακας Εικόνων.....	94
Γ. Βιβλιογραφία.....	96

1. ΕΙΣΑΓΩΓΗ

1.1 Πλαίσιο

Το συγκεκριμένο έργο αναπτύχθηκε στα πλαίσια των αναγκών που δημιουργήθηκαν από τις χρονοβόρες διαδικασίες ανάλυσης των στοιχείων μέσα από βίντεο τεράστιας διάρκειας. Ο παραδοσιακός τρόπος χειροκίνητης ανάλυσης απαιτεί την διάθεση ανθρώπινων πόρων για πολλές ώρες πράγμα που καθιστά την διαδικασία αυτή εξαιρετικά κοπιαστική. Τα στοιχεία που προκύπτουν από αυτές τις αναλύσεις είναι βασικά για την εξιχνίαση εγκλημάτων, επομένως πρόκειται για ένα πρόβλημα υψίστης σημασίας το οποίο λύνοντας το μπορεί να βοηθήσει άμεσα τους πολίτες.

Με την ραγδαία ανάπτυξη της τεχνητής νοημοσύνης τα τελευταία χρόνια έχει δοθεί έμφαση και στον κλάδο της υπολογιστικής όρασης, στην προσπάθεια δηλαδή να δώσουν στους υπολογιστές ικανότητες παρόμοιες με την ανθρώπινη όραση. Έτσι λοιπόν βρέθηκε η λύση στο συγκεκριμένο πρόβλημα, να αυτοματοποιήσουμε την χρονοβόρα αυτή διαδικασία προβολής των στοιχείων και σύγκρισης των αποτελεσμάτων. Χρησιμοποιώντας αλγόριθμους και διαδικασίες που μπορούν να αποκτήσουν, να αναλύσουν και να επεξεργαστούν τα στοιχεία που προκύπτουν από τα βίντεο άμεσα και με μεγάλη ακρίβεια.

1.2 Σκοπός της Διπλωματικής Εργασίας

Ας δούμε λοιπόν πιο αναλυτικά τον σκοπό της εργασίας αυτής. Έχοντας υπόψιν τις ήδη υπάρχουσες τεχνολογίες καταγραφής βίντεο με χρήση καμερών ασφάλειας οι οποίες είναι αρκετά εκσυγχρονισμένες, το κομμάτι της ανάλυσης και προβολής γίνεται μέχρι στιγμής με «παραδοσιακούς» τρόπους που απαιτούν διαθέσιμο ανθρώπινο δυναμικό για αρκετές ώρες. Με αυτόν τον τρόπο βέβαια υπάρχουν πιθανότητες να ξεφύγει κάποιο στοιχείο ή πρόσωπο το οποίο μπορεί να είναι καθοριστικό για την επίλυση μιας υπόθεσης. Έτσι με την βοήθεια της υπολογιστικής όρασης καθώς και των νευρωνικών δικτύων (τομείς που υπάγονται στην ευρύτερη έννοια της τεχνητής νοημοσύνης) μπορούμε να πετύχουμε τους εξής πολύ σημαντικούς στόχους.

- **Εντοπισμός σημείων κίνησης σε βίντεο μεγάλης διάρκειας.**

Πιο συγκεκριμένα δημιουργείται η απαίτηση σε νυχτερινές λήψεις όπου δεν συμβαίνουν πολλά στο οπτικό πεδίο της κάμερας να έχουμε ένα εργαλείο αυτόματου scanning για σημεία ενδιαφέροντος. Σε βίντεο μεγάλης διάρκειας συγκεκριμένα ο εντοπισμός κίνησης μας παραπέμπει απευθείας στα σημεία ενδιαφέροντος όπου συμβαίνει κάτι. Έτσι εξαλείφεται η ανάγκη προβολής ολόκληρου του βίντεο από ανθρώπους είτε το σημαντικό γεγονός βρίσκεται στην αρχή είτε στα τελευταία δευτερόλεπτα.

- **Εντοπισμός προσώπων σε βίντεο.**

Η συγκεκριμένη λειτουργία είναι χρήσιμη σε περιπτώσεις όπου θέλουμε να δούμε ακριβώς τα σημεία που εντοπίζονται πρόσωπα στο βίντεο. Με την βοήθεια ειδικά εκπαιδευμένων νευρωνικών δικτύων μπορούμε να εντοπίσουμε με μεγάλη ακρίβεια τα σημεία της εικόνας όπου εμφανίζεται κάποιο πρόσωπο και να εξάγουμε τα αποτελέσματα.

- **Αναγνώριση προσώπων.**

Η τελευταία και η πιο εντυπωσιακή λειτουργία που θέλουμε να πετύχουμε αφορά την ταυτοποίηση των προσώπων που εντοπίζονται από την προαναφερθείσα λειτουργία. Με την

βοήθεια τοπικής βάσης δεδομένων φωτογραφιών και αλγορίθμων σύγκρισης τελευταίας τεχνολογίας η ταυτοποίηση μπορεί να γίνει γρήγορα αλλά και με μεγάλη ακρίβεια.

Σκοπός λοιπόν είναι η διευκόλυνση των ερευνών της αστυνομίας με την βοήθεια εργαλείων τελευταίας τεχνολογίας στην υπηρεσία του πολίτη.

1.3 Αντικείμενο της Διπλωματικής Εργασίας

Το αντικείμενο της εργασίας αυτής αφορά την δημιουργία μιας πρωτοποριακής πλατφόρμας ανάλυσης βίντεο και εξαγωγής δεδομένων. Μέσω των δυνατοτήτων που προσφέρει η υπολογιστική όραση είναι εφικτό να αναλυθούν βίντεο με ακρίβεια που συναγωνίζεται την ανθρώπινη. Έτσι με βάση αυτό αναπτύχθηκε μια πλήρης και λειτουργική πλατφόρμα για περιβάλλοντα Windows η οποία περιλαμβάνει:

- Εύκολη στη χρήση και απλή διεπαφή χρήστη
- Χρήση Ελληνικής γλώσσας
- Εξαγωγή αποτελεσμάτων
- Ανίχνευση κίνησης σε βίντεο
- Εντοπισμός προσώπων
- Αναγνώριση προσώπων
- Προβολή αποτελεσμάτων

Κατά την υλοποίηση του προγράμματος δόθηκε επίσης προσοχή στην ασφάλεια των δεδομένων τα οποία εξαιτίας της φύσης τους είναι ευαίσθητα (φωτογραφίες, ημερομηνίες γέννησης, ονοματεπώνυμα κ.λπ.). Έτσι η συγκεκριμένη εφαρμογή λειτουργεί χωρίς να έχει επαφή είτε με τοπικά δίκτυα είτε με το διαδίκτυο. Κατ' επέκταση η βάση δεδομένων που χρησιμοποιείται βρίσκεται σε τοπικό επίπεδο και ο χρήστης μπορεί να την διαχειριστεί εύκολα μέσω της ίδιας της πλατφόρμας.



Εικόνα 1: Το λογότυπο της εφαρμογής

1.4 Δομή

Παρακάτω παρουσιάζονται περιληπτικά τα κεφάλαια της εργασίας.

Στο κεφάλαιο 2 γίνεται αναφορά στις εισαγωγικές έννοιες οι οποίες είναι απαραίτητες για την κατανόηση των λειτουργιών της εφαρμογής. Αναφέρονται κυρίως σε πεδία της τεχνητής νοημοσύνης όπως η υπολογιστική όραση, η μηχανική μάθηση και τα νευρωνικά δίκτυα.

Στο κεφάλαιο 3 γίνεται μελέτη των κυριότερων εφαρμογών παρόμοιας λειτουργίας και παρουσίαση των συμπερασμάτων της μελέτης αυτής.

Στο κεφάλαιο 4 παρουσιάζονται τα εργαλεία και οι πλατφόρμες οι οποίες χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Πιο συγκεκριμένα γίνεται αναφορά στις βιβλιοθήκες που χρησιμοποιήθηκαν από το πλαίσιο ανάπτυξης Qt καθώς και οι βιβλιοθήκες από την βασική έκδοση της C++. Επιπλέον γίνεται αναφορά στις βιβλιοθήκες υπολογιστικής όρασης που προσφέρει η OpenCV (Open Source Computer Vision Library) τις οποίες χρησιμοποιεί η εφαρμογή. Τέλος, αναλύεται ο λόγος επιλογής της συγκεκριμένης γλώσσας προγραμματισμού καθώς και μια σύντομη παρουσίαση του Visual Studio 2022.

Στο κεφάλαιο 5 έχουμε την ανάλυση περιπτώσεων χρήσης για 2 βασικές λειτουργίες της εφαρμογής καθώς και την παρουσίαση των λειτουργικών και μη λειτουργικών απαιτήσεων. Επιπλέον περιλαμβάνονται διαγράμματα (use case diagrams) για τις περιπτώσεις χρήσης που περιγράφονται.

Στο κεφάλαιο 6 περιγράφεται η μέθοδος ανάπτυξης της πλατφόρμας καθώς και τα βήματα μέσω των οποίων έγινε ο σχεδιασμός και η ανάπτυξη κάθε ξεχωριστής λειτουργίας που περιλαμβάνεται σε αυτή.

Στο κεφάλαιο 7 απεικονίζονται όλες οι λειτουργίες της πλατφόρμας (μενού, γραμμές εργαλείων, παράθυρα, dialogs). Επιπλέον γίνεται περιγραφή των λειτουργιών καθώς και το πως συνδέονται μεταξύ τους. Δίνεται έμφαση στις διεργασίες που τρέχουν στο παρασκήνιο έτσι ώστε να προβάλλονται κατάλληλα τα αποτελέσματα και να γίνεται σωστή εξαγωγή των δεδομένων.

Στο κεφάλαιο 8 περιλαμβάνονται τα συμπεράσματα επί της διπλωματικής εργασίας σχετικά με τα εργαλεία που χρησιμοποιήθηκαν, την δυσκολία χρήσης και το πως έφτασαν να συνδεθούν μεταξύ τους χωρίς προβλήματα. Επίσης, παρουσιάζονται τα συμπεράσματα και οι παρατηρήσεις σχετικά με την πλατφόρμα. Τέλος γίνεται μια ανασκόπηση των απαιτήσεων της εφαρμογής για να καθοριστούν ίσως μελλοντικές ιδέες και βήματα για την περαιτέρω ανάπτυξη της.

Στα παραρτήματα της εργασίας αναγράφεται η βιβλιογραφία της εργασίας καθώς και μερικά ενδεικτικά παραδείγματα πηγαίου κώδικα. Όπως επίσης, οδηγίες εγκατάστασης της πλατφόρμας αλλά και ο πίνακας των εικόνων που χρησιμοποιήθηκαν στην εργασία αυτή.

2. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

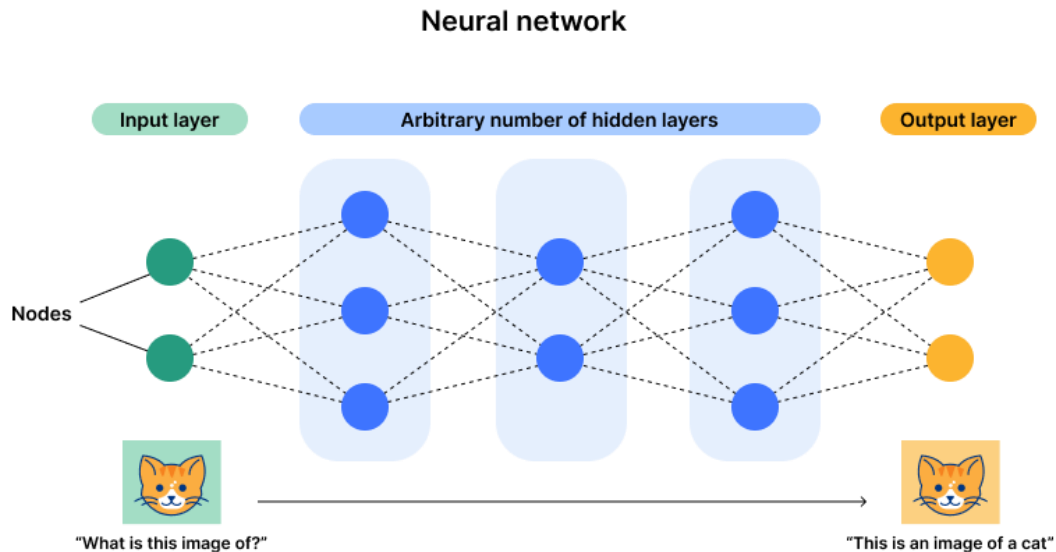
2.1 Νευρωνικά Δίκτυα

2.1.1 Ορισμός

Στη μηχανική μάθηση, ένα νευρωνικό δίκτυο είναι ένα μοντέλο εμπνευσμένο από τη δομή και τη λειτουργία των βιολογικών νευρωνικών δικτύων στους εγκεφάλους των ζώων.

Ένα νευρωνικό δίκτυο αποτελείται από συνδεδεμένες μονάδες ή κόμβους που ονομάζονται τεχνητοί νευρώνες, οι οποίοι μοντελοποιούν στην λειτουργία τους νευρώνες ενός εγκεφάλου. Οι κόμβοι αυτοί συνδέονται με ακμές, οι οποίες με την σειρά τους μοντελοποιούν τις συνάψεις ενός εγκεφάλου. Κάθε τεχνητός νευρώνας λαμβάνει σήματα από συνδεδεμένους νευρώνες, τα επεξεργάζεται και στη συνέχεια στέλνει ένα σήμα σε άλλους συνδεδεμένους νευρώνες. Το "σήμα" είναι ένας πραγματικός αριθμός και η έξοδος κάθε νευρώνα υπολογίζεται από κάποια μη γραμμική συνάρτηση του αθροίσματος των εισόδων του, που ονομάζεται συνάρτηση ενεργοποίησης. Η ισχύς του σήματος σε κάθε σύνδεση καθορίζεται από ένα βάρος, το οποίο προσαρμόζεται κατά τη διάρκεια της διαδικασίας μάθησης.

Συνήθως, οι νευρώνες συγκεντρώνονται σε επίπεδα. Διαφορετικά στρώματα μπορούν να εκτελούν διαφορετικούς μετασχηματισμούς στις εισόδους τους. Τα σήματα ταξιδεύουν από το πρώτο στρώμα (το στρώμα εισόδου) στο τελευταίο στρώμα (το στρώμα εξόδου), περνώντας ενδεχομένως από πολλαπλά ενδιάμεσα στρώματα (κρυφά στρώματα). Ένα δίκτυο ονομάζεται συνήθως βαθύ νευρωνικό δίκτυο αν έχει τουλάχιστον 2 κρυφά στρώματα.



Εικόνα 2: Απλή μορφή ενός νευρωνικού δικτύου.

2.1.2 Τελευταίες Εξελίξεις

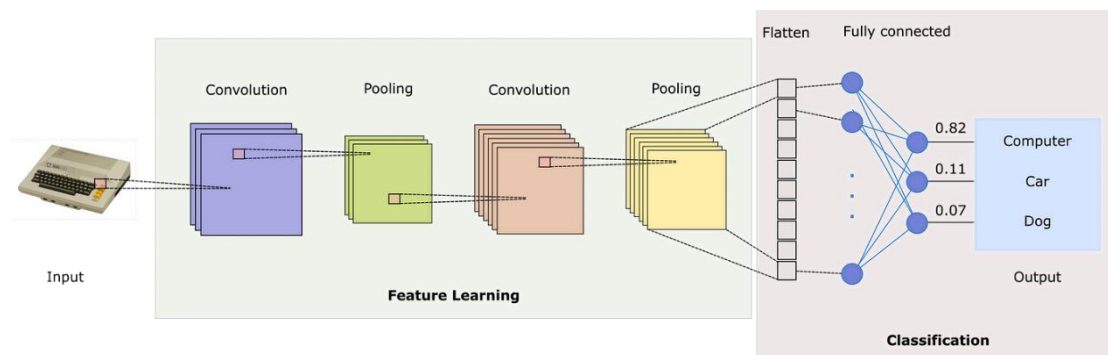
Η μετάβαση από τα συμβατικά νευρωνικά δίκτυα στα νευρωνικά δίκτυα συνέλιξης (Convolutional Neural Network ή αλλιώς CNN) αποτελεί μια σημαντική αναβάθμιση, ιδίως στο πλαίσιο της ανίχνευσης και αναγνώρισης εικόνων. Τα παραδοσιακά νευρωνικά δίκτυα τα οποία χαρακτηρίζονται από πλήρως συνδεδεμένες αρχιτεκτονικές ήταν αποτελεσματικά για

διεργασίες πάνω σε δομημένους τύπους δεδομένων (πίνακες κ.λπ.), αλλά αντιμετώπισαν σημαντικές προκλήσεις όταν εφαρμόστηκαν απευθείας στην ανάλυση εικόνας λόγω της υψηλής διαστατικότητας (κανάλια χρώματος) και της χωρικής πολυπλοκότητας των δεδομένων μιας εικόνας.

Έτσι η ανάπτυξη των CNN υποκινήθηκε από την ανάγκη αποτελεσματικής επεξεργασίας και εξαγωγής ουσιαστικών χαρακτηριστικών από εικόνες, διατηρώντας παράλληλα της χωρικές σχέσεις. Τα CNN εισήγαγαν την έννοια των επιπέδων συνελίξεων (convolutional layers) όπου μικρά φίλτρα ή πυρήνες (kernels) εφαρμόζονται συστηματικά πάνω στην εικόνα εισόδου. Αυτή η προσέγγιση επιτρέπει στο δίκτυο να «λαμβάνει» τοπικά χαρακτηριστικά (π.χ. ακμές, υφές) μέσω μιας διαδικασίας που μοιάζει με χωρικό φιλτράρισμα, μειώνοντας έτσι τον υπολογιστικό φόρτο που συνδέεται με την επεξεργασία κάθε εικονοστοιχείου ξεχωριστά.

Η εξέλιξη των αρχιτεκτονικών CNN περιλαμβάνει αρκετές καινοτομίες. Οι ερευνητές διερεύνησαν τη στοιβάξη πολλαπλών στρωμάτων συνελκτικού τύπου ακολουθούμενων από μη γραμμικές συναρτήσεις ενεργοποίησης (π.χ. ReLU) για την καταγραφή ιεραρχικών αναπαραστάσεων οπτικών χαρακτηριστικών. Επιπλέον, η εισαγωγή στρωμάτων συγκέντρωσης (pooling layers) διευκόλυνε τη χωρική δειγματοληψία, ενισχύοντας περαιτέρω την ικανότητα του δικτύου να εξάγει αναλλοίωτα χαρακτηριστικά από διαφορετικές περιοχές μιας εικόνας.

Η σημασία των CNN έγκειται στην ικανότητά τους να αντιμετωπίζουν αποτελεσματικά τις μοναδικές προκλήσεις που θέτουν τα δεδομένα εικόνας, οδηγώντας σε σημαντικές βελτιώσεις σε εργασίες όπως η αναγνώριση αντικειμένων, η κατανόηση ενός χώρου και η κατάτμηση εικόνων. Οι μηχανισμοί εξαγωγής ιεραρχικών και τοπικών χαρακτηριστικών τους έχουν γίνει θεμελιώδεις στις σύγχρονες εφαρμογές υπολογιστικής όρασης, από συστήματα αναγνώρισης προσώπου έως προηγμένες τεχνολογίες αυτόνομης οδήγησης.



Εικόνα 3: Η δομή ενός συνελκτικού νευρωνικού δικτύου.

2.1.3 Χρησιμότητα

Λόγω της ικανότητάς τους να αναπαράγουν και να μοντελοποιούν μη γραμμικές διαδικασίες, τα νευρωνικά δίκτυα έχουν βρει εφαρμογές σε πολλούς κλάδους. Αυτές περιλαμβάνουν:

- Επεξεργασία δεδομένων.
- Αναγνώριση μοτίβων (συμπεριλαμβανομένων των συστημάτων ραντάρ, της αναγνώρισης προσώπων, της ταξινόμησης σημάτων, της αναγνώρισης αντικειμένων και της λήψης διαδοχικών αποφάσεων).
- Αναγνώριση ακολουθιών, χειρονομιών, ομιλίας, χειρόγραφων και τυπωμένων κειμένων.
- Ανάλυση δεδομένων από αισθητήρες αλλά και ανάλυση εικόνας.
- Ρομποτική.

- Εξόρυξη δεδομένων, συμπεριλαμβανομένης και της εξόρυξης γνώσης από βάσεις δεδομένων (data mining).

Επιπλέον τα CNN έχουν χρησιμοποιηθεί για την ανάλυση αξιοπιστίας των υποδομών που υπόκεινται σε φυσικές καταστροφές και για την πρόβλεψη των καθιζήσεων των θεμελίων. Τέλος έχουν χρησιμοποιηθεί στην ασφάλεια του κυβερνοχώρου, με στόχο τη διάκριση μεταξύ νόμιμων και κακόβουλων δραστηριοτήτων.

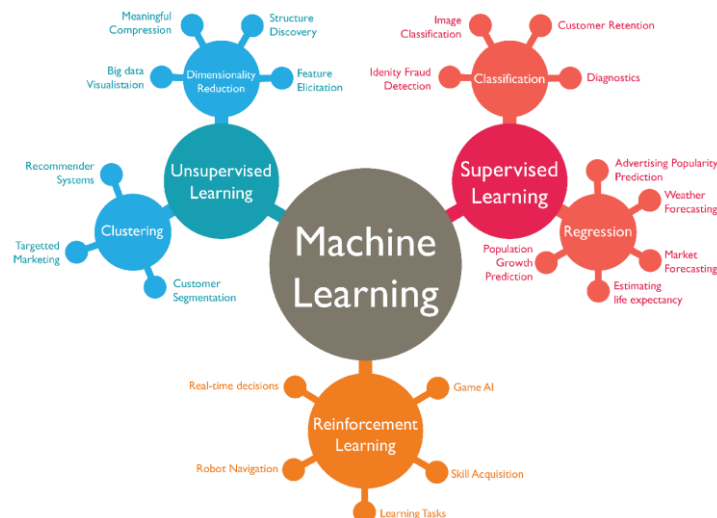
2.2 Μηχανική Μάθηση

2.2.1 Ορισμός

Η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης που επικεντρώνεται στην ανάπτυξη αλγορίθμων και στατιστικών μοντέλων που επιτρέπουν στους υπολογιστές να μαθαίνουν και να λαμβάνουν αποφάσεις ή προβλέψεις βάσει δεδομένων, χωρίς να είναι ρητά προγραμματισμένοι για κάθε εργασία. Στόχος της μηχανικής μάθησης είναι η δημιουργία συστημάτων που μπορούν να μαθαίνουν και να βελτιώνονται αυτόματα με την πάροδο του χρόνου.

Στην ουσία, οι αλγόριθμοι μηχανικής μάθησης ανιχνεύουν μοτίβα και σχέσεις μέσα σε μεγάλα σύνολα δεδομένων για να κάνουν προβλέψεις ή να πάρουν αποφάσεις. Αυτή η διαδικασία περιλαμβάνει την εκπαίδευση του αλγορίθμου σε ιστορικά δεδομένα, επιτρέποντάς του να εντοπίζει μοτίβα και να κάνει γενικεύσεις που μπορούν στη συνέχεια να εφαρμοστούν σε νέα, αθέατα δεδομένα. Η βασική αρχή που διέπει τη μηχανική μάθηση είναι η ικανότητα να μαθαίνει από παραδείγματα και να προσαρμόζεται χωρίς ανθρώπινη παρέμβαση.

Οι τεχνικές μηχανικής μάθησης μπορούν σε γενικές γραμμές να κατηγοριοποιηθούν σε μάθηση με επίβλεψη, μάθηση χωρίς επίβλεψη και ενισχυτική μάθηση. Η μάθηση με επίβλεψη περιλαμβάνει την εκπαίδευση ενός μοντέλου σε επισημασμένα δεδομένα, όπου ο αλγόριθμος μαθαίνει να αντιστοιχίζει τα δεδομένα εισόδου σε αντίστοιχες ετικέτες εξόδου. Η μάθηση χωρίς επίβλεψη, από την άλλη πλευρά, ασχολείται με μη επισημασμένα δεδομένα και επικεντρώνεται στην ανακάλυψη προτύπων ή δομών μέσα στα δεδομένα. Η ενισχυτική μάθηση είναι ένα παράδειγμα όπου ένας πράκτορας μαθαίνει να λαμβάνει διαδοχικές αποφάσεις αλληλοεπιδρώντας με ένα περιβάλλον και λαμβάνοντας ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών.



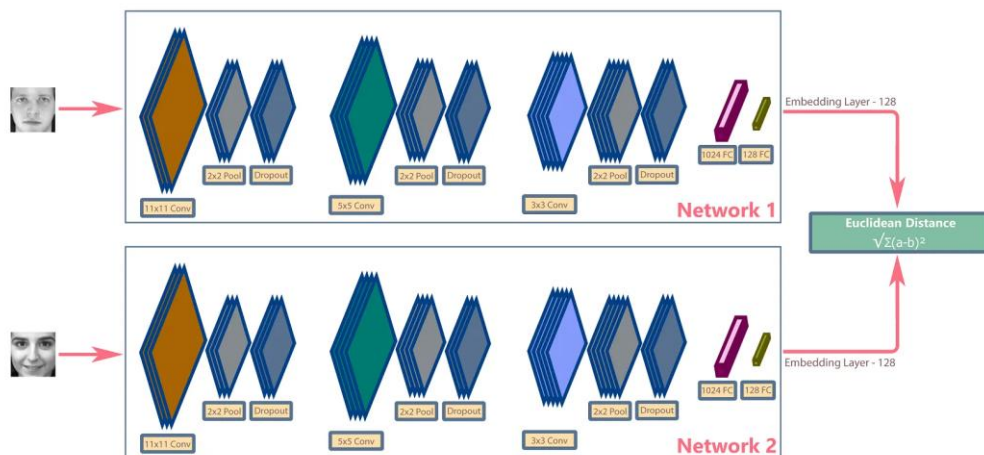
Εικόνα 4: Τεχνικές μηχανικής μάθησης.

2.2.2 Τελευταίες Εξελίξεις

Στο κομμάτι της αναγνώρισης προσώπων η μηχανική μάθηση έχει φέρει τεράστιες εξελίξεις, επιτρέποντας στους υπολογιστές να αναγνωρίζουν μοτίβα και να κάνουν προβλέψεις από δεδομένα. Όπως και στην περίπτωση των νευρωνικών δικτύων οι παραδοσιακές προσεγγίσεις δεν έφταναν ώστε να γίνονται αποτελεσματικά διαδικασίες όπως η αναγνώριση προσώπων ή η επαλήθευση υπογραφών λόγω της διαφοροποιημένης φύσης τους. Ειδικότερα, ένα πολύ αντιπροσωπευτικό παράδειγμα αποτελεί η αναγνώριση του ίδιου προσώπου σε διαφορετικές συνθήκες φωτισμού ή γωνίας λήψης της εικόνας που περιέχει το πρόσωπο. Ενώ ο ανθρώπινος εγκέφαλος κατανοεί ότι πρόκειται για το ίδιο άτομο αναλύοντας αυτόματα τα χαρακτηριστικά του προσώπου χωρίς να έχει σημασία ο φωτισμός ή η γωνία λήψης, η ομοιότητα αυτή γίνεται δύσκολα αντιληπτή από παλαιότερες μεθόδους. Όπως καταλαβαίνουμε τα νευρωνικά δίκτυα αποτελούν αναπόσπαστο κομμάτι της μηχανικής μάθησης.

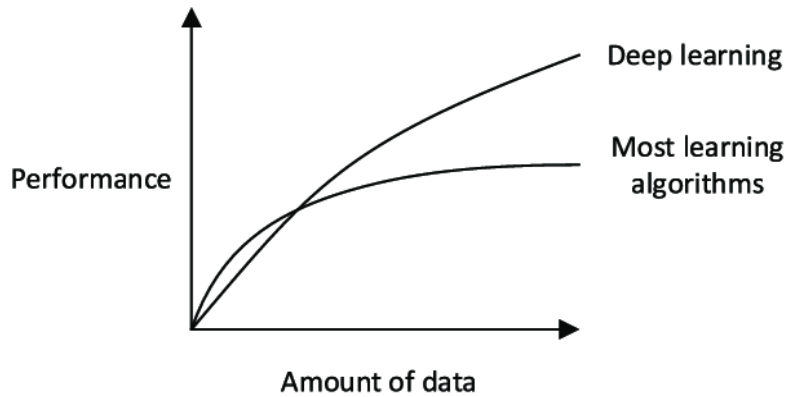
Την λύση έρχεται να δώσει ένα υποσύνολο της μηχανικής μάθησης με τον όρο deep learning (βαθιά μάθηση) και συγκεκριμένα μέσω των σιαμαίων δικτύων (Siamese networks) προσφέροντας την δυνατότητα εκμάθησης χρησιμοποιώντας πιο σύνθετα δεδομένα. Τα σιαμαία δίκτυα αποτελούν μια σημαντική ανακάλυψη για να μπορέσουμε να μάθουμε σε μια μηχανή την ομοιότητα όπως την αντιλαμβάνεται ο ανθρώπινος εγκέφαλος. Χρησιμοποιώντας δίδυμα δίκτυα με κοινά βάρη οι αρχιτεκτονικές τύπου Siamese υπερέχουν στην αναγνώριση ουσιαστικών σχέσεων μεταξύ των εισόδων τους. Αυτή η ικανότητα έχει δώσει τη δυνατότητα στα μοντέλα να εκτελούν εργασίες όπως η επαλήθευση και η ταυτοποίηση προσώπου με πρωτοφανή ακρίβεια, ακόμη και κάτω από δύσκολες συνθήκες, όπως οι μεταβολές στο φωτισμό, τη στάση ή την έκφραση.

Ένα παράδειγμα χρήσης σιαμαίων δικτύων είναι για την σύγκριση μεταξύ της ευκλείδειας απόστασης των αποτελεσμάτων που παράγονται όταν τα δεδομένα μιας εικόνας επεξεργάζονται από τα κρυφά στρώματα ενός νευρωνικού δικτύου. Δύο διαφορετικές εικόνες εισάγονται σε ξεχωριστά δίκτυα με παρόμοια βάρη και ανάλογα με το πόσο «κοντά» είναι τα αποτελέσματα το δίκτυο κρίνει εάν πρόκειται για το ίδιο άτομο ή όχι.



Εικόνα 5: Αναπαράσταση σιαμαίων δικτύων. (Πηγή)

Επιπλέον η βαθιά μάθηση μπορεί να λειτουργήσει πιο αποτελεσματικά σε μεγάλο όγκο δεδομένων πόσω μάλλον όταν πρόκειται για δεδομένα εικόνες που είναι από την φύση τους και πολύπλοκα. Παρακάτω φαίνεται διαγραμματικά η αποτελεσματικότητα μεταξύ της βαθιάς μάθησης και των υπόλοιπων μεθόδων μηχανικής μάθησης.



Εικόνα 6: Σύγκριση αλγορίθμων εκμάθησης.

2.2.3 Χρησιμότητα

Η χρήση αλγορίθμων μηχανικής μάθησης έχει επεκταθεί σε πολλούς κλάδους. Μερικά παραδείγματα χρήσεων αποτελούν:

- Αναγνώριση εικόνας.

Μια από τις πιο αξιοσημείωτες εφαρμογές μηχανικής μάθησης είναι η αναγνώριση εικόνας, η οποία είναι μια μέθοδος καταγραφής και ανίχνευσης ενός αντικειμένου ή χαρακτηριστικού σε μια ψηφιακή εικόνα. Επιπλέον, η τεχνική αυτή χρησιμοποιείται για περαιτέρω ανάλυση, όπως η αναγνώριση μοτίβων, η ανίχνευση προσώπου και η αναγνώριση προσώπου.

- Αναγνώριση ομιλίας.

Αλγόριθμοι μηχανικής μάθησης μπορούν να «καταλάβουν» τις λέξεις που εκφωνούνται από έναν άνθρωπο χρησιμοποιώντας μια σειρά αριθμών που αναπαριστούν το σήμα ομιλίας. Δημοφιλείς εφαρμογές που χρησιμοποιούν την αναγνώριση ομιλίας περιλαμβάνουν την Alexa της Amazon, την Siri της Apple και τους χάρτες της Google.

- Πρόβλεψη κυκλοφοριακής κατάστασης.

Στην συγκεκριμένη περίπτωση η εφαρμογή των χαρτών της Google συλλέγει πληροφορίες σχετικά με την κυκλοφοριακή κατάσταση της περιοχής όταν ενεργοποιούμε την τοποθεσία μας για να δημιουργήσει προβλέψεις σχετικά με την επερχόμενη κυκλοφορία και να προσδιορίσει την ταχύτερη διαδρομή προς τον προορισμό μας.

- Συστάσεις προϊόντων ηλεκτρονικού εμπορίου.

Ένα από τα σημαντικότερα στοιχεία συνήθως κάθε ιστότοπου ηλεκτρονικού εμπορίου είναι οι συστάσεις προϊόντων, οι οποίες περιλαμβάνουν την εξελεγμένη χρήση αλγορίθμων μηχανικής μάθησης. Οι ιστότοποι παρακολουθούν τη συμπεριφορά των πελατών με βάση τις προηγούμενες αγορές, τις συνήθειες περιήγησης, το ιστορικό καλαθιού και στη συνέχεια προτείνουν προϊόντα.

- Φιλτράρισμα ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου.

Μια από τις πιο δημοφιλείς εφαρμογές της μηχανικής μάθησης που όλοι γνωρίζουμε είναι το φιλτράρισμα των ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου. Οι πάροχοι υπηρεσιών ηλεκτρονικού ταχυδρομείου κατασκευάζουν εφαρμογές με φίλτρα spam που

χρησιμοποιούν αλγορίθμους μηχανικής μάθησης για να ταξινομήσουν ένα εισερχόμενο μήνυμα ως spam και να το κατευθύνουν στον κατάλληλο φάκελο.

- Προσωπικοί βοηθοί.

Οι προσωπικοί βοηθοί βοηθούν τους ανθρώπους να έχουν πρόσβαση σε σχετικές πληροφορίες μέσω κειμένου ή φωνής. Όταν τίθεται ένα ερώτημα στο σύστημα, ο προσωπικός βοηθός συλλέγει πληροφορίες αναζητώντας τις ή ανακαλώντας παρόμοιες ερωτήσεις που το άτομο έχει θέσει στο παρελθόν. Ορισμένες δημοφιλείς τεχνικές μηχανικής μάθησης που εμπλέκονται στους εικονικούς αυτούς βοηθούς περιλαμβάνουν την αναγνώριση ομιλίας, τη μετατροπή ομιλίας σε κείμενο, την επεξεργασία φυσικής γλώσσας και τη μετατροπή κειμένου σε ομιλία.

2.3 Υπολογιστική Όραση

2.3.1 Ορισμός

Ο τομέας της υπολογιστικής όρασης περιλαμβάνει μεθόδους απόκτησης, επεξεργασίας, ανάλυσης και κατανόησης ψηφιακών εικόνων, καθώς και εξαγωγής δεδομένων υψηλής διάστασης από τον πραγματικό κόσμο με σκοπό την παραγωγή αριθμητικών ή συμβολικών πληροφοριών, π.χ. σε μορφή αποφάσεων. «Κατανόηση» σε αυτό το πλαίσιο σημαίνει μετατροπή των οπτικών μέσων σε περιγραφές που οδηγούν σε συμπεράσματα. Αυτή η «κατανόηση» της εικόνας μπορεί να θεωρηθεί ως μια ανάμειξη των συμβολικών πληροφοριών από τα δεδομένα της με τη χρήση μοντέλων που κατασκευάζονται με τη βοήθεια της γεωμετρίας, της φυσικής, της στατιστικής και της θεωρίας μάθησης.

Ο πρωταρχικός στόχος της υπολογιστικής όρασης είναι να καταστήσει τις μηχανές ικανές να επεξεργάζονται και να αναλύουν οπτικά δεδομένα, συνήθως με τη μορφή εικόνων ή βίντεο, εξάγοντας σημαντικές πληροφορίες από αυτές τις εισροές. Η διαδικασία αυτή περιλαμβάνει έναν συνδυασμό απόκτησης εικόνων, προεπεξεργασίας, εξαγωγής χαρακτηριστικών και τεχνικών μηχανικής μάθησης. Αξιοποιώντας τη δύναμη των μοντέλων βαθιάς μάθησης, όπως τα νευρωνικά δίκτυα συνελίξεων (CNN), τα συστήματα υπολογιστικής όρασης έχουν επιτύχει πρωτοφανή ακρίβεια και αποτελεσματικότητα σε διεργασίες όπως η ανίχνευση αντικειμένων, η ταξινόμηση εικόνων και η σημασιολογική κατάτμηση (διαχωρισμός των πληροφοριών που εξάγονται).

2.3.2 Τελευταίες Εξελίξεις

Τα τελευταία χρόνια, σε συνδυασμό με την μηχανική μάθηση, ο τομέας της υπολογιστικής όρασης έχει ανοίξει τον δρόμο για την δημιουργία πληθώρας νέων τεχνολογιών που βασίζονται στην ανάλυση εικόνων. Οι αλγόριθμοι μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και οι μέθοδοι βαθιάς μάθησης, έχουν επιτρέψει στους υπολογιστές να αναλύουν και να κατανοούν την περιεχόμενη πληροφορία σε εικόνες με ανεπανάληπτη ακρίβεια και αποτελεσματικότητα. Αυτή η εξέλιξη έχει ανοίξει νέους ορίζοντες σε πεδία όπως η αυτόνομη οδήγηση, η ιατρική και η αυτόματη αναγνώριση μοτίβων (pattern recognition).

Η εφαρμογή της μηχανικής μάθησης στην υπολογιστική όραση έχει δώσει τη δυνατότητα στις μηχανές να αναγνωρίζουν αντικείμενα, πρόσωπα και συμπεριφορές με εντυπωσιακή επιτυχία. Αυτό έχει οδηγήσει σε εφαρμογές που μέχρι τώρα ήταν σενάρια επιστημονικής φαντασίας, όπως οι έξυπνες πόλεις και οι ρομποτικές λύσεις. Παράλληλα, αυτή η τεχνολογία έχει αναδείξει ηθικά καθώς και νομικά ζητήματα που πρέπει να εξεταστούν προσεκτικά, συμπεριλαμβανομένων των θεμάτων προστασίας της ιδιωτικής ζωής και της αναγνώρισης προσώπων.

2.3.3 Χρησιμότητα

Στην υπολογιστική όραση υπάρχουν δύο βασικές κατηγορίες διεργασιών, η αναγνώριση και η ανάλυση κίνησης.

- Αναγνώριση

Ο σκοπός ενός συστήματος που χρησιμοποιεί τεχνολογίες υπολογιστικής όρασης είναι ο προσδιορισμός του κατά πόσον τα δεδομένα μιας εικόνας περιέχουν ή όχι κάποιο συγκεκριμένο αντικείμενο, χαρακτηριστικό ή δραστηριότητα. Υπάρχουν τρεις πολύ κοντινές κατηγορίες λύσεων που πετυχαίνουν τον εξής σκοπό:

1. Αναγνώριση αντικειμένων: Ένα ή περισσότερα προκαθορισμένα αντικείμενα ή κλάσεις αντικειμένων μπορούν να αναγνωριστούν, συνήθως μαζί με τις θέσεις τους σε μια εικόνα (δισδιάστατος χώρος) ή τις στάσεις τους σε μια σκηνή (τρισδιάστατος χώρος). Η συγκεκριμένη κατηγορία έχει διαφορά από την ανίχνευση με την έννοια ότι η αναγνώριση γίνεται σε αντικείμενα του ίδιου τύπου. Ενώ ο ανθρώπινος εγκέφαλος αναγνωρίζει πως μια κούπα παραμένει κούπα όταν είναι σε διαφορετικά χρώματα, μέγεθος, περιστροφή, η ακόμα και όταν φαίνεται μόνο το χερούλι της σε μια εικόνα, οι υπολογιστές χρειάζονται ειδικές στρατηγικές για την επίτευξη του σκοπού αυτού.
2. Ταυτοποίηση: Ταυτοποιείται μια μεμονωμένη περίπτωση ενός αντικειμένου. Παραδείγματα περιλαμβάνουν την αναγνώριση ενός προσώπου ή του δακτυλικού αποτυπώματος ενός συγκεκριμένου ατόμου, την αναγνώριση χειρόγραφων ψηφίων ή την αναγνώριση ενός συγκεκριμένου οχήματος.
3. Ανίχνευση: Τα δεδομένα εικόνας σαρώνονται για συγκεκριμένα αντικείμενα μαζί με τις θέσεις τους. Παραδείγματα περιλαμβάνουν την ανίχνευση ενός εμποδίου στο οπτικό πεδίο του αυτοκινήτου και πιθανών μη φυσιολογικών κυττάρων ή ιστών σε ιατρικές εικόνες ή την ανίχνευση ενός οχήματος σε ένα αυτόματο σύστημα διοδίων.

Επί του παρόντος, οι καλύτεροι αλγόριθμοι για τέτοιου είδους εργασίες βασίζονται σε συνελκτικά νευρωνικά δίκτυα.

- Ανάλυση Κίνησης

Αρκετές διεργασίες σχετίζονται με την ανάλυση της κίνησης, όπου μια ακολουθία εικόνων επεξεργάζεται ώστε να γίνει μια εκτίμηση της ταχύτητας είτε σε κάθε σημείο της εικόνας είτε στην τρισδιάστατη σκηνή ή ακόμη και της κάμερας που παράγει τις εικόνες. Παραδείγματα τέτοιων διεργασιών είναι:

1. Παρακολούθηση: Παρακολούθηση των κινήσεων ενός (συνήθως) μικρότερου συνόλου σημείων ενδιαφέροντος ή αντικειμένων (π.χ. οχημάτων, αντικειμένων, ανθρώπων ή άλλων οργανισμών) στην ακολουθία εικόνων.
2. Οπτική ροή: Προσδιορισμός, για κάθε σημείο της εικόνας, του τρόπου με τον οποίο το σημείο αυτό κινείται σε σχέση με το επίπεδο της εικόνας, δηλαδή της φαινομενικής κίνησής του. Η κίνηση αυτή είναι αποτέλεσμα τόσο του τρόπου με τον οποίο κινείται το αντίστοιχο τρισδιάστατο σημείο στη σκηνή όσο και του τρόπου με τον οποίο κινείται η κάμερα σε σχέση με τη σκηνή.



Εικόνα 7: Παράδειγμα εικόνας μετά την χρήση αλγορίθμων υπολογιστικής όρασης. (Πηγή)

2.4 Ανίχνευση Κίνησης

Η ανίχνευση κίνησης σε βίντεο είναι μια θεμελιώδης διεργασία στον τομέα της υπολογιστικής όρασης και την ανάλυση βίντεο, η οποία επιτρέπει την αυτοματοποιημένη αναγνώριση και παρακολούθηση κινούμενων αντικειμένων σε μια ακολουθία καρέ. Η ικανότητα ανίχνευσης της κίνησης εξυπηρετεί ένα ευρύ φάσμα εφαρμογών, από συστήματα επιτήρησης και ασφάλειας έως ανάλυση αθλητικών αγώνων και επεξεργασία βίντεο. Με την ανίχνευση και την ανάλυση της κίνησης, τα συστήματα μπορούν να ενεργοποιούν αυτόνομα γεγονότα, όπως η έναρξη εγγραφής με βάση την κίνηση, ο εντοπισμός ανωμαλιών σε υλικό παρακολούθησης ή η δημιουργία δυναμικών οπτικών εφέ σε εφαρμογές πολυμέσων.

Για την ανίχνευση κίνησης σε βίντεο χρησιμοποιούνται διάφορες μεθοδολογίες και αλγόριθμοι, ο καθένας με τα δικά του πλεονεκτήματα ανάλογα με τις απαιτήσεις της κάθε εφαρμογής:

- Μέθοδος διαφοράς εικονοστοιχείων

Η προσέγγιση αυτή περιλαμβάνει την αφαίρεση διαδοχικών καρέ για τον εντοπισμό περιοχών όπου συμβαίνουν σημαντικές αλλαγές εικονοστοιχείων, που υποδηλώνουν κίνηση. Για να το πετύχουμε αυτό πρέπει να μετατρέψουμε την εικόνα από το RGB κανάλι στο grayscale κανάλι. Έτσι αντί να υπολογίζουμε την διαφορά σε πίνακα 3 διαστάσεων με διαφορετικές τιμές βάζουμε τις ίδιες σε όλες τις διαστάσεις. Πλέον έχουμε τιμές που αντιπροσωπεύουν την φωτεινότητα (λευκό έως μαύρο) κάνοντας την διαφορά πιο εμφανή αλλά και τους υπολογισμούς να θέλουν λιγότερη επεξεργαστική ισχύ.

- Αλγόριθμοι οπτικής ροής

Οι αλγόριθμοι οπτικής ροής υπολογίζουν την κίνηση των εικονοστοιχείων μεταξύ των καρέ, παρέχοντας πληροφορίες για την κατεύθυνση και το μέγεθος της κίνησης. Αυτό επιτυγχάνεται με την μέθοδο των Lucas-Kanade, υποθέτουμε δηλαδή ότι γύρω από ένα σημείο στο βίντεο τα γειτονικά εικονοστοιχεία σε ένα πλέγμα 3 επί 3 θα έχουν παρόμοια κατεύθυνση κίνησης καθώς και την ίδιο αριθμό φωτεινότητας.

- Αφαίρεση φόντου

Δημιουργώντας ένα μοντέλο του στατικού φόντου σε μια σκηνή, η μέθοδος αυτή εντοπίζει αντικείμενα του προσκηνίου που αποκλίνουν από το φόντο με την πάροδο του χρόνου. Μια από τις καλύτερες μεθόδους αφαίρεσης φόντου χρησιμοποιεί Gaussian κατανομή. Χρησιμοποιεί μια μέθοδο για τη μοντελοποίηση κάθε εικονοστοιχείου του φόντου με ένα μείγμα K Gaussian κατανομών ($K = 3$ έως 5). Τα βάρη του μείγματος αντιπροσωπεύουν τις χρονικές αναλογίες που τα χρώματα αυτά παραμένουν στη σκηνή. Τα πιθανά χρώματα φόντου είναι αυτά που παραμένουν περισσότερο στο βίντεο και είναι πιο στατικά.

- Ανάλυση περιοχών (blob analysis)

Τα κινούμενα αντικείμενα τμηματοποιούνται σε διακριτές περιοχές (blobs) εντός των καρέ, διευκολύνοντας την παρακολούθηση και την ανάλυση των αντικειμένων. Αυτή η μέθοδος μπορεί να συνδυαστεί με την χρήση νευρωνικών δικτύων τα οποία έχουν εκπαιδευτεί να εντοπίζουν τα αντικείμενα στην εικόνα.

Στην εφαρμογή που αναπτύχθηκε έγινε χρήση της τελευταίας μεθόδου. Κυρίως γιατί οι προηγούμενοι τρόποι εντόπιζαν σημεία κίνησης τα οποία δεν ήταν απαραίτητα. Βασικό παράδειγμα είναι η χρήση ώρας επάνω στο βίντεο (timestamp), η εναλλαγή των δευτερολέπτων εντοπιζόταν ως κίνηση με αποτέλεσμα τα σημεία ενδιαφέροντος να χάνονται ανάμεσα σε λανθασμένους εντοπισμούς.

2.5 Ανίχνευση Προσώπων

Η ανίχνευση προσώπων σε βίντεο είναι άλλη μια βασική διεργασία στο πλαίσιο της υπολογιστικής όρασης. Χρησιμοποιείται σε διάφορες εφαρμογές όπως η επιτήρηση, η βιομετρία, η ανάλυση βίντεο και η αλληλεπίδραση ανθρώπου-υπολογιστή. Η ικανότητα αυτόματου εντοπισμού και αναγνώρισης ανθρώπινων προσώπων σε ακολουθίες βίντεο είναι απαραίτητη για τη δημιουργία ευφυών συστημάτων που μπορούν να ερμηνεύουν και να ανταποκρίνονται στις οπτικές πληροφορίες. Με τον εντοπισμό προσώπων μέσα σε καρέ βίντεο, τα συστήματα μπορούν να εξάγουν σημαντικές πληροφορίες και να διευκολύνουν μεταγενέστερες εργασίες όπως η αναγνώριση προσώπου που θα δούμε παρακάτω. Όπως και με την ανίχνευση κίνησης υπάρχουν διάφορες μέθοδοι για την επίτευξη αυτής της λειτουργίας, οι οποίες χρησιμοποιούνται ανάλογα με τις απαιτήσεις τις εκάστοτε εφαρμογής:

- Haar Cascades

Αυτή η κλασική μέθοδος χρησιμοποιεί μια αλληλουχία απλών ταξινομητών που εκπαιδεύονται για τον εντοπισμό χαρακτηριστικών που μοιάζουν με πρόσωπα, γεγονός που την καθιστά υπολογιστικά αποδοτική για εφαρμογές πραγματικού χρόνου.

- Histogram of Oriented Gradients (HOG)

Η μέθοδος HOG λαμβάνει τις τοπικές πληροφορίες των ακμών (περιοχές όπου υπάρχει απότομη αλλαγή χρώματος) για την αναπαράσταση χαρακτηριστικών προσώπου, οι οποίες στη συνέχεια χρησιμοποιούνται σε συνδυασμό με ταξινομητές μηχανικής μάθησης για την ανίχνευση προσώπων.

- Συνελκτικά Νευρωνικά Δίκτυα (CNN)

Αρχιτεκτονικές CNN που βασίζονται στη βαθιά μάθηση, όπως το Single Shot Multibox Detector (SSD), έχουν επιδείξει εξαιρετικές επιδόσεις σε εργασίες ανίχνευσης προσώπου,

μαθαίνοντας ιεραρχικές αναπαραστάσεις των χαρακτηριστικών του προσώπου. Η συγκεκριμένη μέθοδος χρησιμοποιείται στην εφαρμογή μιας γιατί παρέχει μεγάλη ακρίβεια σε συνδυασμό με μικρή επεξεργαστική ισχύ.

2.6 Αναγνώριση Προσώπων

Η αναγνώριση προσώπου, είναι ένας εξελιγμένος τομέας μελέτης που επικεντρώνεται στην αυτόματη αναγνώριση και επαλήθευση ατόμων με βάση τα χαρακτηριστικά του προσώπου τους. Παίζει καθοριστικό ρόλο σε διάφορες εφαρμογές, όπως συστήματα ασφαλείας, έλεγχος πρόσβασης, συστήματα επιβολής του νόμου και εξατομικευμένες εμπειρίες χρηστών. Στόχος των συστημάτων αναγνώρισης προσώπου είναι η ακριβής αντιστοίχιση των προσώπων που ανιχνεύονται σε εικόνες ή βίντεο με μια βάση δεδομένων γνωστών προσώπων, επιτρέποντας την αξιόπιστη ταυτοποίηση και πιστοποίηση. Σε αντίθεση με τις παραδοσιακές μεθόδους ταυτοποίησης, όπως οι κωδικοί πρόσβασης ή οι κωδικοί PIN, η αναγνώριση προσώπου προσφέρει έναν εξαιρετικά αξιόπιστο τρόπο επιβεβαίωσης της ταυτότητας, γεγονός που την καθιστά ιδιαίτερα πολύτιμη για εφαρμογές που απαιτούν απρόσκοπτη και ασφαλή επαλήθευση του χρήστη.

Για την επιτυχή αναγνώριση προσώπων έχουν αναπτυχθεί διάφορες πλατφόρμες και μεθοδολογίες αξιοποιώντας τις εξελίξεις στην υπολογιστική όραση, την αναγνώριση προτύπων και την μηχανική μάθηση. Παραδείγματα μεθόδων περιλαμβάνουν:

- Eigenfaces (Ιδιοπρόσωπα)

Ένα σύνολο ιδιοπρόσωπων μπορεί να δημιουργηθεί με την εκτέλεση μιας μαθηματικής διαδικασίας που ονομάζεται ανάλυση κύριων συνιστωσών (PCA) σε ένα μεγάλο σύνολο εικόνων που απεικονίζουν διαφορετικά ανθρώπινα πρόσωπα. Ανεπίσημα, τα ιδιοπρόσωπα μπορούν να θεωρηθούν ένα σύνολο «τυποποιημένων συστατικών προσώπου», που προέρχονται από τη στατιστική ανάλυση πολλών εικόνων προσώπων. Κάθε ανθρώπινο πρόσωπο μπορεί να θεωρηθεί ότι είναι ένας συνδυασμός αυτών των τυποποιημένων προσώπων. Για παράδειγμα, το πρόσωπο κάποιου μπορεί να αποτελείται από το μέσο πρόσωπο συν 10% από το eigenface 1, 55% από το eigenface 2 και 3% από το eigenface 3. Είναι αξιοσημείωτο ότι δεν χρειάζονται πολλά ιδιοπρόσωπα συνδυασμένα μεταξύ τους για να επιτευχθεί μια δίκαιη προσέγγιση των περισσότερων προσώπων. Επίσης, επειδή το πρόσωπο ενός ατόμου δεν καταγράφεται από μια ψηφιακή φωτογραφία, αλλά αντίθετα ως μια απλή λίστα τιμών (μια τιμή για κάθε ιδιοπρόσωπο στη βάση δεδομένων που χρησιμοποιήθηκε), καταλαμβάνεται πολύ λιγότερος χώρος για το πρόσωπο κάθε ατόμου.

- Fisherfaces

Η μέθοδος Fisherfaces, επίσης γνωστή ως Γραμμική Διακριτική Ανάλυση του Fisher (FLDA) για την αναγνώριση προσώπων, είναι μια τεχνική που χρησιμοποιείται για την εξαγωγή διακριτικών χαρακτηριστικών από εικόνες προσώπου ώστε να είναι δυνατή η αποτελεσματική ταξινόμηση και αναγνώριση προσώπων. Ο πρωταρχικός στόχος της Fisherfaces είναι η προβολή δεδομένων εικόνας προσώπου υψηλής διάστασης σε έναν υποχώρο χαμηλότερης διάστασης, με σκοπό τον διαχωρισμό των χαρακτηριστικών (μάτια, στόμα κ.λπ.). Η μέθοδος αυτή μπορεί να χειριστεί αποτελεσματικά τις διακυμάνσεις του φωτισμού, της πόζας και της έκφρασης του προσώπου, καθιστώντας την ανθεκτική για την αναγνώριση προσώπων σε συνθήκες πραγματικού χρόνου.

- Συνελικτικά Νευρωνικά Δίκτυα (CNNs)

Αποτελεί μια σύγχρονη προσέγγιση που χρησιμοποιεί τεχνικές βαθιάς μάθησης για την αυτόματη εκμάθηση διακριτικών χαρακτηριστικών απευθείας από ακατέργαστες εικόνες

προσώπου. Τα CNN μαθαίνουν αυτόματα ιεραρχικές αναπαραστάσεις των χαρακτηριστικών του προσώπου απευθείας από τις τιμές των εικονοστοιχείων. Τα κατώτερα στρώματα των CNN μαθαίνουν βασικά χαρακτηριστικά όπως οι ακμές και οι υφές, ενώ τα ανώτερα στρώματα μαθαίνουν πιο σύνθετα και αφηρημένα χαρακτηριστικά που αφορούν τα πρόσωπα, όπως τα μάτια, η μύτη και το στόμα.

Αφού εκπαιδευτεί, ένα CNN μπορεί να χρησιμοποιηθεί για την αναγνώριση προσώπων τροφοδοτώντας μια νέα εικόνα προσώπου στο δίκτυο. Το CNN επεξεργάζεται την εικόνα εισόδου μέσω των επιπέδων του, εξάγοντας διακριτικά χαρακτηριστικά σε κάθε στάδιο. Η έξοδος του CNN μπορεί να είναι μια κατανομή πιθανοτήτων σε διάφορες κλάσεις (ατομικές ταυτότητες) ή να παρέχει απευθείας διανύσματα χαρακτηριστικών (feature vectors) που αντιπροσωπεύουν το πρόσωπο εισόδου. Κατά την εξαγωγή συμπερασμάτων, το CNN συγκρίνει τα εξαγόμενα χαρακτηριστικά του προσώπου εισόδου με εκείνα γνωστών προσώπων σε μια βάση δεδομένων χρησιμοποιώντας μετρικές ομοιότητας (π.χ. ευκλείδεια απόσταση, ομοιότητα συνημίτονου). Για να προσδιορίσουμε αν υπάρχει αντιστοιχία ή όχι χρησιμοποιούμε ένα συγκεκριμένο threshold ανάλογα με την μετρική ομοιότητας που έχουμε επιλέξει.

Λόγω της αποτελεσματικότητας που παρέχουν τα CNN αλλά και της ικανότητας να μαθαίνουν αυτόματα πολύπλοκα μοτίβα και παραλλαγές στην εμφάνιση του προσώπου έχουν επιλεγεί σαν μέθοδος αναγνώρισης για την εργασία αυτή.

3. ΑΝΑΛΥΣΗ ΠΕΔΙΟΥ

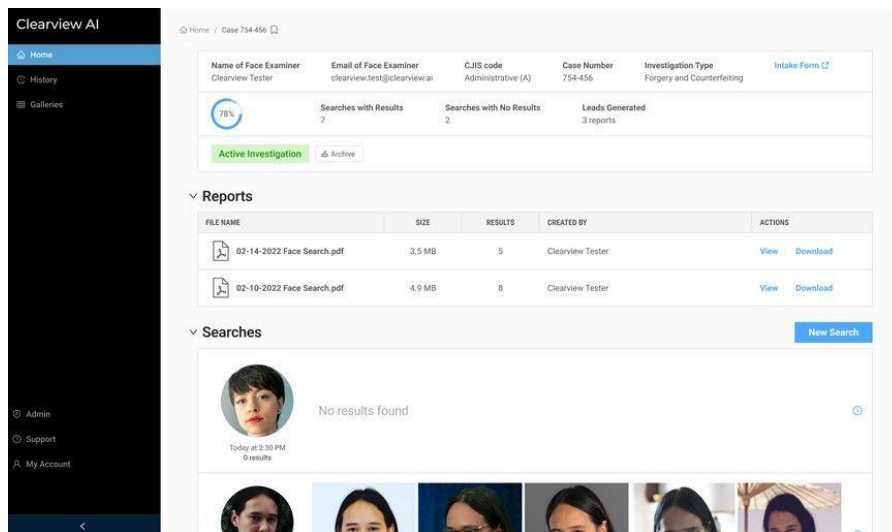
3.1 Ανάλυση Κυριότερων Εφαρμογών

Σε αυτό το κεφάλαιο γίνεται μια state-of-the-art ανάλυση μερικών εφαρμογών και εργαλείων που προσφέρουν υπηρεσίες αναγνώρισης προσώπου, είτε έχουν εμπορικό χαρακτήρα είτε προορίζονται καθαρά για κρατικούς οργανισμούς.

3.1.1 Clearview AI

Η Clearview AI είναι μια ιδιωτική εταιρεία με έδρα τις ΗΠΑ. Έχει αναπτύξει μια επαναστατική, διαδικτυακή πλατφόρμα πληροφοριών για τις κυβερνητικές υπηρεσίες, η οποία χρησιμοποιείται ως εργαλείο για έρευνες που διεξάγουν τα σώματα ασφαλείας. Η πλατφόρμα, η οποία βασίζεται στην τεχνολογία αναγνώρισης προσώπου, περιλαμβάνει τη μεγαλύτερη γνωστή βάση δεδομένων με 40+ δισεκατομμύρια εικόνες προσώπου που προέρχονται από δημόσιες διαδικτυακές πηγές, όπως ειδησεογραφικά μέσα ενημέρωσης, ιστότοπους με φωτογραφίες συλλήψεων, δημόσια μέσα κοινωνικής δικτύωσης και άλλες ανοικτές πηγές.

Αξιωματικοί των αρχών επιβολής του νόμου δήλωσαν ότι η αναγνώριση προσώπου της Clearview είναι πολύ ανώτερη στην αναγνώριση δραστών από οποιαδήποτε γωνία από ό,τι η τεχνολογία που χρησιμοποιούνταν προηγουμένως. Με την Clearview AI, οι αρχές μπορούν να ανεβάσουν μια εικόνα του προσώπου ενός υπόπτου και να την αντιπαραβάλουν με τη βάση δεδομένων τους. Το λογισμικό παρέχει στη συνέχεια συνδέσμους για το πού μπορεί να βρεθεί η «ταύτιση» στο διαδίκτυο.



Εικόνα 8: Η διεπαφή χρήστη της Clearview AI. (Πηγή)

Εξαιτίας της ιδιαίτερης φύσης της εφαρμογής στην οποία πλέον έχουν πρόσβαση μόνο κυβερνητικές οργανώσεις, είναι αδύνατο δυστυχώς να παρέχουμε μια πλήρη παρουσίαση των ικανοτήτων της. Η παραπάνω εικόνα δημοσιεύτηκε από τον ιστότοπο της Clearview το 2022 για την παρουσίαση της νέας αναβάθμισης της πλατφόρμας. Παρά τις λίγες πληροφορίες μπορούμε από την συγκεκριμένη εικόνα να δούμε πως λειτουργεί το κεντρικό παράθυρο της εφαρμογής.

Στο επάνω μέρος βλέπουμε την χρήση αριθμού υπόθεσης, επομένως για κάθε υπόθεση μπορούν να οριστούν διαφορετικοί παράμετροι στον πίνακα. Παρατηρούμε ότι υπάρχουν πληροφορίες όπως το όνομα του χρήστη, ο τύπος της έρευνας και ο αριθμός των

αποτελεσμάτων των αναζητήσεων. Αμέσως από κάτω βλέπουμε να είναι υπογραμμισμένος με πράσινο χρώμα ο όρος «Active Investigation» με ένα κουμπί «Archive» από δίπλα. Δίνεται ίσως η δυνατότητα στον χρήστη να στέλνει μια υπόθεση στο αρχείο αφού ολοκληρωθούν οι έρευνες.

Στο επόμενο κομμάτι βλέπουμε τον πίνακα με τις αναφορές (Reports) τα οποία αρχεία δημιουργούνται σε μορφή PDF. Ο πίνακας αναφέρει το μέγεθος του αρχείου, τον αριθμό των αποτελεσμάτων, από ποιόν δημιουργήθηκε καθώς και τις ενέργειες προβολής (View) και λήψης (Download).

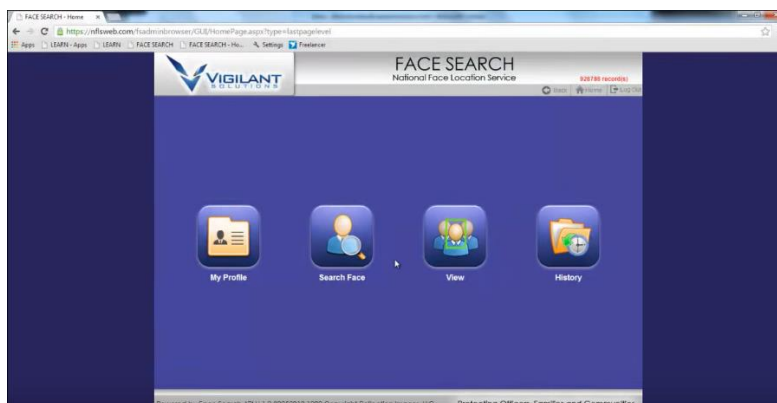
Στο κάτω μέρος του παραθύρου έχουμε τον πίνακα με τις αναζητήσεις (Searches) όπου παρουσιάζεται η φωτογραφία με την πιο πρόσφατη αναζήτηση που έγινε και τα αποτελέσματα της μαζί με το κουμπί για να ξεκινήσει μια νέα αναζήτηση στο επάνω δεξιό μέρος του πίνακα.

Ο τρόπος με τον οποίο η εφαρμογή της Clearview εκτελεί τις υπηρεσίες αναγνώρισης προσώπου είναι:

- Μεταφόρτωση εικόνας ως αντικείμενο διερεύνησης.
- Η εικόνα υφίσταται προεπεξεργασία, η οποία περιλαμβάνει ευθυγράμμιση και περικοπή. Στη συνέχεια, η προ επεξεργασμένη εικόνα μετατρέπεται σε αριθμητικό διάνυσμα εξάγοντας τα χαρακτηριστικά της. Αυτό το επιτυγχάνεται χρησιμοποιώντας νευρωνικά δίκτυα, τα οποία αποδίδουν σημαντικά καλύτερα σε σύγκριση με τις συμβατικές μεθόδους.
- Τέλος, το διάνυσμα χαρακτηριστικών της εικόνας που διερευνάται συγκρίνεται με τα σχετικά τμήματα μιας βάσης δεδομένων και η βάση δεδομένων επιστρέφει ένα σύνολο πιθανών αντιστοιχιών, εάν υπάρχει, ταξινομημένων με βάση την ομοιότητά τους με το πρόσωπο εισόδου. Με βάση τα κατώτατα όρια ομοιότητας τα αποτελέσματα μπορούν στη συνέχεια να χρησιμοποιηθούν για να βοηθήσουν στη διαδικασία ταυτοποίησης ενός ατόμου ή επαλήθευσης της ταυτότητάς του.

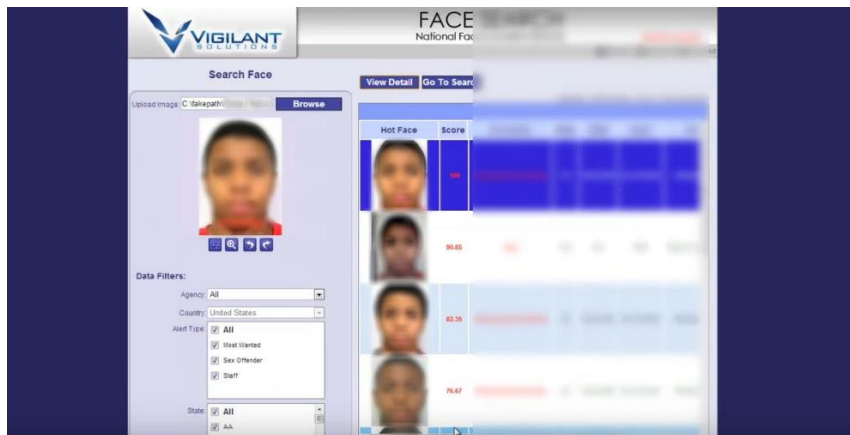
3.1.2 Vigilant FaceSearch

Η FaceSearch είναι ακόμη μια εφαρμογή αναγνώρισης προσώπου από την Vigilant σε συνεργασία με την Motorola Solutions και την Avigilon. Το λογισμικό είναι διαθέσιμο μέσω του διαδικτύου, σε κινητές συσκευές και ως εφαρμογή που βασίζεται σε υπολογιστή. Οι αλγόριθμοι αναγνώρισης προσώπου που χρησιμοποιεί έχουν δημιουργηθεί από την ίδια την εταιρεία (in-house).



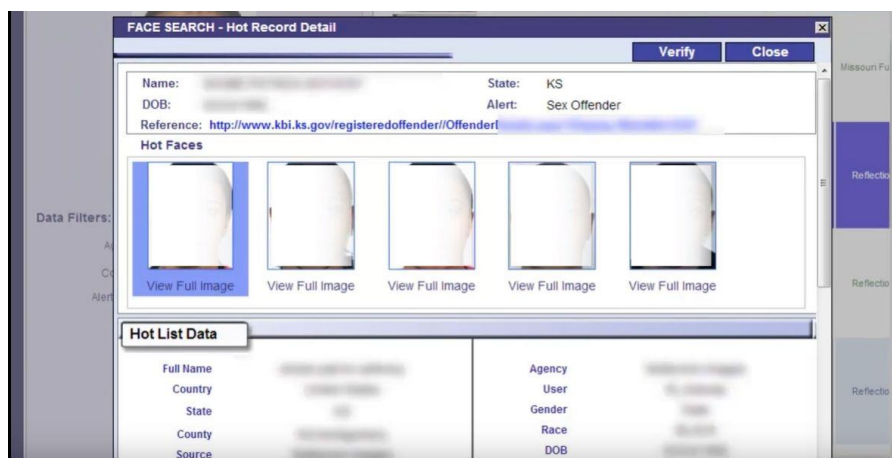
Εικόνα 9: Η αρχική οθόνη της FaceSearch.

Στο αρχικό παράθυρο της web εφαρμογής βλέπουμε ότι υπάρχουν λειτουργίες προβολής της βάσης, αναγνώρισης προσώπου αλλά και προβολής ιστορικού αναζητήσεων.



Εικόνα 10: Η λειτουργία αναγνώρισης προσώπου.

Στο παράθυρο αυτό ο χρήστης μπορεί να ανεβάσει μια φωτογραφία και να ορίσει συγκεκριμένα φίλτρα για τον ύποπτο όπως τον τύπο εγκλήματος που έχει διαπράξει την πολιτεία στην οποία έγινε το έγκλημα, το φύλο και την φυλή. Τέλος για να ξεκινήσει η αναζήτηση ο χρήστης πρέπει να ορίσει την ποιότητα της εικόνας αλλά και τον αριθμό αποτελεσμάτων που θέλει να εμφανίσει (π.χ. τα δέκα πιο κοντινά αποτελέσματα). Πατώντας το κουμπί αναζήτησης εμφανίζεται στο δεξί μέρος ο πίνακας με τις πιθανές αντιστοιχίες του αρχικού ύποπτου. Διπλά στην φωτογραφία εμφανίζεται επίσης το ποσοστό αντιστοιχίας.



Εικόνα 11: Παράθυρο προβολής πληροφοριών υπόπτου.

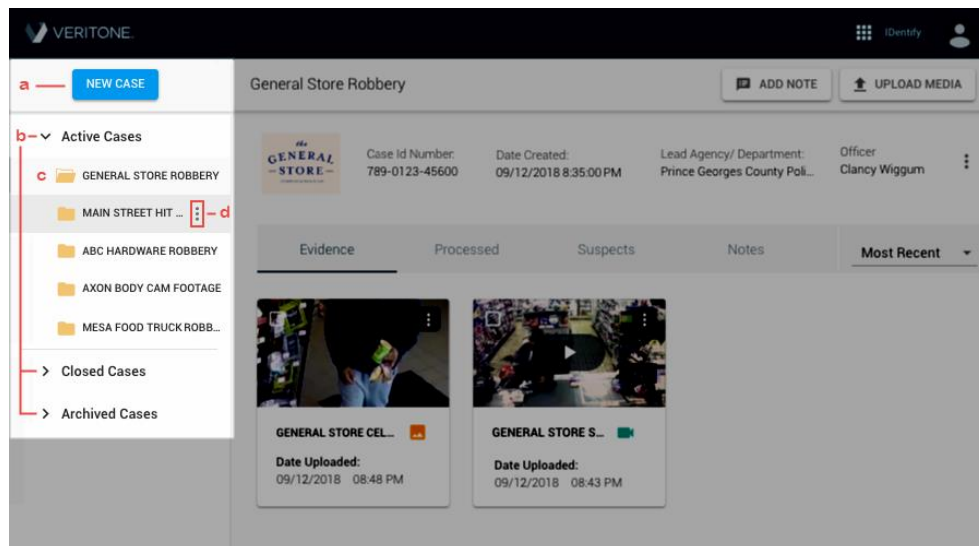
Σε περίπτωση που ο χρήστης επιθυμεί να δει περισσότερες πληροφορίες για τα αποτελέσματα της αναζήτησης του μπορεί να πατήσει επάνω στο όνομα που συνοδεύει την φωτογραφία. Στην Εικόνα 11 μπορούμε να δούμε ολόκληρη την βάση φωτογραφιών για το συγκεκριμένο άτομο καθώς και πληροφορίες που αφορούν την περιοχή, την ημερομηνία γέννησης και τα εξωτερικά χαρακτηριστικά του.

3.1.3 Veritone IDentify

Η εφαρμογή IDentify της Veritone αποτελεί επίσης ένα πανίσχυρο εργαλείο για τις αστυνομικές αρχές. Η IDentify προσφέρει υπηρεσίες αναγνώρισης προσώπου, αρχείου

υπόπτων καθώς και την δυνατότητα δημιουργίας και διαχείρισης μεμονωμένων υποθέσεων. Πιο συγκεκριμένα η IDentify έχει τις εξής δυνατότητες.

- Αυτόματη σύγκριση ατόμων που έχουν καταγραφεί σε βίντεο ή φωτογραφικό υλικό με τη βάση δεδομένων της υπηρεσίας για τους γνωστούς παραβάτες.
- Περιορισμός των πιθανών αντιστοιχιών με βάση βασικά αναγνωριστικά στοιχεία όπως το φύλο, η ηλικία, το ύψος, το χρώμα των μαλλιών, το χρώμα των ματιών και η εθνικότητα που είναι αποθηκευμένα στη βάση δεδομένων κράτησης.
- Δημιουργία εικονικών σειρών πιθανών υπόπτων για περαιτέρω αξιολόγηση και διερεύνηση. Διευκόλυνση της ανταλλαγής καταλόγων υπόπτων και πληροφοριών κράτησης μεταξύ συναδέλφων εντός και μεταξύ υπηρεσιών.
- Συγκέντρωση των πληροφοριών των υπόπτων σε μια ασφαλή διαδικτυακή πλατφόρμα, συμπεριλαμβανομένης της εμφάνισης τους σε αποδεικτικά στοιχεία, των φωτογραφιών κράτησης, της τελευταίας γνωστής διεύθυνσης και άλλων σχετικών στοιχείων.



Εικόνα 12: Το περιβάλλον της εφαρμογής IDentify.

Παρατηρούμε επίσης ότι το περιβάλλον χρήστη της IDentify είναι παρόμοιο με αυτό της Clearview όσον αφορά τον τρόπο που χειρίζεται τις υποθέσεις. Στην Εικόνα 12 βλέπουμε ότι ο χρήστης μπορεί με το κουμπί «New Case» να δημιουργήσει μια νέα υπόθεση και να καταθέσει στοιχεία φωτογραφιών ή βίντεο. Όσον αφορά την ταξινόμηση των υποθέσεων έχουμε τρεις κατηγορίες φακέλων, τις ανοιχτές υποθέσεις, τις ολοκληρωμένες υποθέσεις και τις αρχειοθετημένες.

Στο κύριο μέρος της εφαρμογής, το παράθυρο «Evidence» περιέχει τα στοιχεία της υπόθεσης που έχει ανεβάσει ο χρήστης προς επεξεργασία. Πάνω σε αυτά μπορούν να εκτελεστούν οι αλγόριθμοι αναγνώρισης προσώπου αλλά και η δυνατότητα «Trim and Process» στην οποία ο χρήστης αφαιρεί τα κομμάτια του βίντεο που δεν χρειάζεται. Στο παράθυρο «Processed» εμφανίζονται τα αρχεία στα οποία έχει πραγματοποιηθεί η αναγνώριση προσώπων μαζί με τα αποτελέσματα για τα πρόσωπα που βρέθηκαν στο κάθε αρχείο. Τέλος, στο παράθυρο «Suspects» εμφανίζονται τα άτομα τα οποία έχουν οριστεί ως ύποπτοι για την συγκεκριμένη υπόθεση από την βάση δεδομένων του εκάστοτε αστυνομικού τμήματος.

3.2 Συμπεράσματα Ανάλυσης

Μέσα από την ανάλυση αυτή παρατηρούμε ότι και οι τρεις εφαρμογές ενώ διαφέρουν στην λεπτομέρεια, προσφέρουν παρόμοιες βασικές λειτουργίες με κυριότερη την αναγνώριση προσώπων. Ενώ η IDentify και η Clearview λειτουργούν με βάση τις υποθέσεις, η FaceSearch μπορεί να πραγματοποιήσει αυτόνομες αναζητήσεις και να συγκρίνει πρόσωπα με τις βάσεις δεδομένων του κάθε αστυνομικού τμήματος. Η πιο σημαντική διαφορά βέβαια μεταξύ των τριών εφαρμογών είναι το γεγονός πως μόνο η IDentify μπορεί να πραγματοποιήσει αναγνώριση προσώπων επάνω σε βίντεο. Από την άλλη μεριά, στο κομμάτι του μεγέθους των βάσεων δεδομένων φωτογραφιών που διαθέτει η κάθε εφαρμογή, η Clearview υπερέχει από τις υπόλοιπες αφού έχει στην κατοχή της τουλάχιστον 40 δισεκατομμύρια φωτογραφίες. Αξίζει να σημειωθεί επίσης ότι η Clearview δεν προσφέρει την δυνατότητα δημιουργίας βάσης δεδομένων με συγκεκριμένες φωτογραφίες υπόπτων που έχουν οι άλλες 2 εφαρμογές.

Όσον αφορά την διεπαφή χρήστη, η Clearview και η IDentify φαίνεται ότι διαθέτουν πιο σύγχρονο και μινιμαλιστικό σχεδιασμό με σκοπό την διευκόλυνση του χρήστη. Οι κεντρικές οθόνες έχουν απλό σχεδιασμό και ολοκληρώνουν οποιαδήποτε λειτουργία σε πολύ λίγα βήματα.

4. ΕΡΓΑΛΕΙΑ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ ΑΝΑΠΤΥΞΗΣ

4.1 Εργαλεία Ανάπτυξης

4.1.2 Qt Development Framework

4.1.2.1 Γενικές Πληροφορίες

Το Qt είναι ένα πλαίσιο ανάπτυξης εφαρμογών για την δημιουργία γραφικών διεπαφών χρήστη καθώς και εφαρμογών που μπορούν να εκτελεστούν σε διάφορες πλατφόρμες λογισμικού και υλικού, όπως Windows, Linux, macOS και Android. Το βασικότερο προτέρημα του πλαισίου αυτού είναι η ικανότητα των εφαρμογών που αναπτύσσονται να εκτελούνται σε διαφορετικές πλατφόρμες χωρίς μεγάλες αλλαγές στον πηγαίο κώδικα.

Η ανάπτυξη του πλαισίου ξεκίνησε το 1991 με την πρώτη επίσημη έκδοση να είναι διαθέσιμη το 1995. Σήμερα το λογισμικό αναπτύσσεται από την QT Company και βρίσκεται στην έκδοση 6.7. Το Qt είναι διαθέσιμο τόσο υπό εμπορικές άδειες όσο και υπό άδειες ανοικτού κώδικα.

Ο πηγαίος κώδικας του Qt έχει γραφτεί σε C++ και υποστηρίζει τους μεταγλωττιστές GCC και Clang C++. Επιπλέον, μπορεί να ενσωματωθεί ως πρόσθετο στο Visual Studio 2022 ώστε να συνδυαστούν οι δυνατότητες λεπτομερούς debugging που προσφέρει η πλατφόρμα με την γρήγορη ανάπτυξη γραφικών διεπαφών χρήστη. Οι βιβλιοθήκες που διαθέτει μπορούν να χρησιμοποιηθούν με Python ή C++.



Εικόνα 13: Το λογότυπο του Qt Development Framework.

4.1.2.2 Πως Χρησιμοποιήθηκε

Κατά τη διάρκεια της μελέτης των λειτουργικών απαιτήσεων του προγράμματος, προέκυψε η ανάγκη για την ανάπτυξη ενός γραφικού περιβάλλοντος χρήστη. Αρχικά, καταβλήθηκαν προσπάθειες να χρησιμοποιηθεί το Win32 API, το οποίο, αν και έχει χρησιμοποιηθεί εκτενώς από τη Microsoft για την ανάπτυξη native εφαρμογών στα Windows, είναι εξαιρετικά πολύπλοκο και δύσχρηστο. Η πολυπλοκότητα και η εκτενής φύση των εργαλείων που έπρεπε να αναπτυχθούν για το Argus καθιστούσαν αυτή τη λύση μη ρεαλιστική.

Ως επόμενη επιλογή για τη δημιουργία γραφικών διεπαφών σε C++, εξετάστηκε το πλαίσιο wxWidgets. Το συγκεκριμένο πλαίσιο, αν και πιο προσιτό στη χρήση από το Win32 API, παρουσιάζει περιορισμούς όσον αφορά την επεκτασιμότητα και την υποστήριξη της αντικειμενοστραφούς προγραμματιστικής λογικής. Επιπλέον, το wxWidgets, λόγω της παλαιότητας της αρχιτεκτονικής του, δεν πληροί τις σύγχρονες απαιτήσεις που τέθηκαν για το έργο Argus, ιδίως σε ό,τι αφορά τη συντήρηση και την επεκτασιμότητα της εφαρμογής.

Η ανάλυση των διαθέσιμων επιλογών οδήγησε στην απόφαση να αναζητηθούν εναλλακτικά πλαίσια που να προσφέρουν πιο σύγχρονες δυνατότητες και ευελιξία. Συγκεκριμένα, εξετάστηκαν πλαίσια όπως το Qt, το οποίο είναι γνωστό για την πλούσια υποστήριξή του σε αντικειμενοστραφή προγραμματισμό και την ευκολία του στη δημιουργία σύνθετων και

επεκτάσιμων γραφικών διεπαφών. Το Qt προσφέρει επίσης εκτενή τεκμηρίωση και μια ενεργή κοινότητα, γεγονός που διευκολύνει την ανάπτυξη και την επίλυση προβλημάτων. Επιπλέον το Qt παρέχει την δυνατότητα ενσωμάτωσης του πλαισίου με το Visual Studio χρησιμοποιώντας πρόσθετο Qt VS Tools.

Το Qt VS Tools για το Visual Studio 2022 αποτελεί μια ολοκληρωμένη λύση που συνδυάζει το σύγχρονο πλαίσιο Qt με τις προηγμένες δυνατότητες που προσφέρει το Visual Studio. Αυτή η ενσωμάτωση επιτρέπει την πλήρη αξιοποίηση των ισχυρών βιβλιοθηκών και των εργαλείων που παρέχει το Qt για την ανάπτυξη γραφικών διεπαφών, ενώ παράλληλα επωφελείται από το ευέλικτο και πλούσιο σε χαρακτηριστικά περιβάλλον ανάπτυξης του Visual Studio.

Ολόκληρη η γραφική διεπαφή έχει σχεδιαστεί από την αρχή με την βοήθεια του Qt με πρωταρχικό στόχο την συνολική εμπειρία χρήστη. Έχει δοθεί έμφαση στην λεπτομέρεια, την αποδοτικότητα και την ευχρηστία με σχεδιαστικές επιρροές από παλαιότερες και διαδεδομένες εφαρμογές των Windows.

4.1.2.3 Ο Μηχανισμός Slots and Signals

Το σύστημα slots και signals είναι ένας από τους πιο χαρακτηριστικούς μηχανισμούς του Qt framework, που επιτρέπει την επικοινωνία μεταξύ αντικειμένων σε μια εφαρμογή με γραφικό περιβάλλον χρήστη. Αποτελεί μια μορφή παρατηρητή (observer pattern) που διευκολύνει την ενδοσυνεννόηση αντικειμένων με τρόπο που είναι ασφαλής για νήματα και εύκολος στη χρήση. Ο συγκεκριμένος μηχανισμός χρησιμοποιείται σε όλη την έκταση του προγράμματος όπου έχουμε αλληλεπίδραση μεταξύ χρήστη και προγράμματος ή προγράμματος με τον εαυτό του. Για παράδειγμα, το πάτημα ενός κουμπιού θεωρείται αλληλεπίδραση με τον χρήστη ενώ η ενεργοποίηση ενός κουμπιού μόλις μια διεργασία ολοκληρώνεται γίνεται με τον εαυτό του.

- **Signals:** Τα signals είναι γεγονότα που εκπέμπονται (emit) από ένα αντικείμενο όταν συμβαίνει μια συγκεκριμένη ενέργεια. Για παράδειγμα, όταν ο χρήστης κάνει κλικ σε ένα κουμπί, το κουμπί μπορεί να εκπέμψει ένα signal.
- **Slots:** Τα slots είναι μέθοδοι που μπορούν να συνδεθούν με signals. Όταν ένα signal εκπέμπεται, τα συνδεδεμένα slots εκτελούνται.

Για να χρησιμοποιήσουμε signals and slots πρέπει να κληρονομήσουμε από την κλάση QObject και να συμπεριλάβουμε την μακροεντολή Q_OBJECT στην κλάση μας.

```

21 namespace Ui { class CatalogDialog; }
22
23 class CatalogDialog : public QDialog
24 {
25     Q_OBJECT
26
27 public:
28     CatalogDialog(QWidget* parent = nullptr);
29     ~CatalogDialog();
30
31 private slots:
32     void buttonAddClicked();
33     bool isFolderEmpty();

```

Εικόνα 14: Παράδειγμα κλάσης που χρησιμοποιεί slots and signals.

Η σύνδεση ενός signal με ένα slot γίνεται με την μέθοδο `connect()`. Αυτή η σύνδεση μπορεί να πραγματοποιηθεί είτε κατά την δημιουργία των αντικειμένων είτε δυναμικά κατά την εκτέλεση. Για παράδειγμα μπορούμε να συνδέσουμε το signal που εκπέμπει το πάτημα ενός

κουμπιού όταν εκτελείται η συνάρτηση `clicked()` με το slot `buttonAddClicked()` για να προσδιορίσουμε τι θα γίνει όταν πατηθεί το συγκεκριμένο κουμπί.

Στο παράδειγμα της Εικόνας 14 η `QDialog` είναι μια υποκλάση της `QWidget`, η οποία με τη σειρά της είναι υποκλάση της `QObject`. Έτσι, η `QDialog` κληρονομεί όλα τα χαρακτηριστικά της `QObject`, συμπεριλαμβανομένης της λειτουργικότητας των `signals` και `slots`. Αυτό επιτρέπει την εύκολη δημιουργία παραθύρων διαλόγου που μπορούν να επικοινωνούν με άλλα αντικείμενα μέσω `signals` και `slots`. Επιπλέον για να μπορέσουμε να χρησιμοποιήσουμε μια συνάρτηση ως slot πρέπει να εισάγουμε την μακροεντολή «slots» δίπλα στον προσδιορισμό πρόσβασης «private».

4.1.2.4 Κλάσεις και Βιβλιοθήκες που Χρησιμοποιήθηκαν

Παρακάτω παρουσιάζονται όλες οι βιβλιοθήκες και οι κλάσεις του Qt που χρησιμοποιήθηκαν για την ανάπτυξη των λειτουργιών της πλατφόρμας Argus. (Σημείωση: Όλες οι αναφορές σε κώδικα, συναρτήσεις, μεταβλητές και κλάσεις που ανήκουν στην εφαρμογή Argus τονίζονται με συγκεκριμένη γραμματοσειρά.)

4.1.2.4.1 QAudioOutput

Η `QAudioOutput` είναι μια κλάση που ανήκει στην βιβλιοθήκη Qt Multimedia, η οποία παρέχει λειτουργίες για την αναπαραγωγή ήχου. Η βιβλιοθήκη Qt Multimedia περιλαμβάνει εργαλεία που αφορούν τα πολυμέσα, όπως ήχο και βίντεο, και η `QAudioOutput` συγκεκριμένα επιτρέπει την αναπαραγωγή ηχητικών δεδομένων (συνήθως μέσω ηχείων ή ακουστικών). Στην περίπτωση μας εκτός από την αναπαραγωγή ηχητικών δεδομένων χρησιμοποιείται και για τον έλεγχο της έντασης.

```

122     audioOutput = new QAudioOutput(this);
123     videoPlayer->setAudioOutput(audioOutput);
124
125     labelVolume = ui->labelVolume;
126     sliderVolume = ui->sliderVolume;
127     sliderVolume->setRange(0, 100);
128     sliderVolume->setValue(audioOutput->volume());
129     connect(sliderVolume, &QSlider::valueChanged, audioOutput, &QAudioOutput::setVolume);
130     connect(sliderVolume, &QSlider::valueChanged, [&](int volume)
131     {
132         QString currentVolume = QString::number(volume) + "%";
133         labelVolume->setText(currentVolume);
134     });

```

Εικόνα 15: Χρήση της `QAudioOutput`.

Στην Εικόνα 15, γραμμή 123 χρησιμοποιούμε το `audioOutput` για να θέσουμε την πηγή ήχου στο `videoPlayer`. Επιπλέον, στις γραμμές 128 έως 130, χρησιμοποιούμε ένα `signal` για να συνδέσουμε έναν `slider` ελέγχου της έντασης με την αλλαγή της έντασης του ήχου. Αυτή η σύνδεση επιτρέπει στο χρήστη να ρυθμίζει την ένταση του ήχου του βίντεο χρησιμοποιώντας τον `slider`.

4.1.2.4.2 QWidget

Η `QWidget` είναι μια θεμελιώδης κλάση στο Qt Framework που αντιπροσωπεύει ένα στοιχείο γραφικού περιβάλλοντος χρήστη (GUI). Όλα τα οπτικά στοιχεία του Qt, όπως παράθυρα, κουμπιά, πεδία κειμένου και άλλα, κληρονομούν από την `QWidget`. Είναι η βάση για όλα τα οπτικά αντικείμενα και παρέχει τις βασικές λειτουργίες για την εμφάνιση και τη διαχείριση γραφικών στοιχείων.

Στην εφαρμογή χρησιμοποιείται κυρίως μέσω της υποκλάσης QDialog στην οποία θα αναφερθούμε παρακάτω. Επιπλέον μπορεί να χρησιμοποιηθεί και ως template κλάση για την δημιουργία οποιουδήποτε προσαρμοσμένου παραθύρου.

4.1.2.4.3 QMainWindow

Η QMainWindow ανήκει στις θεμελιώδεις κλάσεις του framework που παρέχει τη βασική δομή για τη δημιουργία παραθύρων (main windows) με ένα τυπικό γραφικό περιβάλλον χρήστη. Πρόκειται για μια πιο εξειδικευμένη μορφή της QWidget, η οποία προσφέρει επιπλέον λειτουργίες και προκαθορισμένες περιοχές για τη δημιουργία πολύπλοκων παραθύρων εφαρμογών. Δημιουργώντας ένα παράθυρο με την QMainWindow έχουμε τις εξής προσθήκες:

- **Κεντρικό Widget (Central Widget):** Η QMainWindow διαθέτει μια κεντρική περιοχή όπου μπορεί να τοποθετηθεί ένα widget, το οποίο καταλαμβάνει το μεγαλύτερο μέρος του παραθύρου.
- **Μπάρα Μενού (Menu Bar):** Περιλαμβάνει μια μπάρα μενού στο πάνω μέρος του παραθύρου, επιτρέποντας την προσθήκη και τη διαχείριση μενού και υπομενού.
- **Μπάρα Εργαλείων (Tool Bar):** Παρέχει μία ή περισσότερες μπάρες εργαλείων που μπορούν να τοποθετηθούν σε διάφορες θέσεις γύρω από την κεντρική περιοχή.
- **Μπάρα Κατάστασης (Status Bar):** Περιλαμβάνει μια μπάρα κατάστασης στο κάτω μέρος του παραθύρου, χρήσιμη για την εμφάνιση μηνυμάτων και πληροφοριών κατάστασης.

Η QMainWindow χρησιμοποιείται στον constructor της κλάσης mainwindow της εφαρμογής η οποία είναι υπεύθυνη για την δημιουργία του κύριου γραφικού περιβάλλοντος.

```

4      ~MainWindow::MainWindow(QWidget *parent)
5      : QMainWindow(parent)
6      , ui(new Ui::MainWindow)
7      {

```

Εικόνα 16: Χρήση της QMainWindow.

4.1.2.4.4 QVideoWidget

Η QVideoWidget ανήκει στην βιβλιοθήκη Qt Multimedia η οποία παρέχει ένα widget για την προβολή βίντεο. Είναι σχεδιασμένη να χρησιμοποιείται με την QMediaPlayer ή άλλες κλάσεις του Qt Multimedia για την αναπαραγωγή και προβολή βίντεο σε εφαρμογές Qt. Στην συγκεκριμένη περίπτωση ρυθμίζουμε το μέρος όπου θα προβάλλεται το βίντεο στην εφαρμογή συνδέοντας το widget με το αντικείμενο που χειρίζεται την αναπαραγωγή των πολυμέσων.

```

116     videoPlayer = new QMediaPlayer(this);
117     videoWidget = new QVideoWidget(this);
118     videoPlayer->setVideoOutput(videoWidget);
119     videoFrame = ui->videoFrame;
120     videoFrame->addWidget(videoWidget);

```

Εικόνα 17: Χρήση της QVideoWidget.

4.1.2.4.5 QMediaPlayer

Όπως και η QVideoWidget έτσι και η QMediaPlayer ανήκει στην βιβλιοθήκη Qt Multimedia. Είναι μια κλάση που παρέχει λειτουργικότητα για την αναπαραγωγή πολυμέσων ήχου και βίντεο. Πρόκειται για μια ισχυρή και ευέλικτη κλάση που υποστηρίζει διάφορες μορφές αρχείων και πηγές δεδομένων. Επιπλέον παρέχει λειτουργίες για την εκκίνηση, παύση, διακοπή

και επαναφορά της αναπαραγωγής. Μέσω του μηχανισμού slots και signals μπορεί κάποιος να ελέγξει την κατάσταση που βρίσκεται ανά πάσα στιγμή το βίντεο ώστε να εκτελεστούν ταυτόχρονα και άλλες λειτουργίες. Στον κώδικα της Εικόνας 18 που ακολουθεί χρησιμοποιούμε την μακροεντολή `LoadedMedia` για να δούμε πότε έχει φορτωθεί το βίντεο στην εφαρμογή ώστε να ενεργοποιήσουμε ορισμένες λειτουργίες και κουμπιά. Στην γραμμή 248 με την εντολή `emit onVideoLoaded(true)` εκπέμπεται το σήμα το οποίο ενεργοποιεί τις λειτουργίες που απαιτούν να έχει φορτωθεί κάποιο βίντεο.

```

236 //Catch the LoadedMedia status so we know the video is ready to be played.
237 connect(videoPlayer, &QMediaPlayer::mediaStatusChanged, [&](QMediaPlayer::MediaStatus status)
238 {
239     if (status == QMediaPlayer::LoadedMedia && !videoLoaded)
240     {
241         mediaPlayer->setPlaybackRate(playbackRate);
242         actionMetadata->setDisabled(0);
243         setVideoDuration();
244         output = getLogTime() + " Έτοιμο.";
245         textLogger->append(output);
246         videoLoaded = true;
247         //Emit the signal indicating the change in the video loaded state.
248         emit onVideoLoaded(true);
249     }

```

Εικόνα 18: Χρήση μακροεντολών από την κλάση `QMediaPlayer`.

4.1.2.4.6 QMessageBox

Η `QMessageBox` παρέχει έναν απλό τρόπο εμφάνισης προσαρμοσμένων μηνυμάτων προειδοποίησης, βοήθειας, πληροφοριών ή σφάλματος. Πρόκειται για ένα τυποποιημένο παράθυρο το οποίο επιτρέπει στον προγραμματιστή να ενημερώνει τον χρήστη ή να ζητάει επιβεβαίωση για κάποια ενέργεια. Ένα παράθυρο μηνύματος υποστηρίζει επίσης διάφορους τύπους κουμπιών όπως «OK», «Ακύρωση», «Εφαρμογή» και «Αποθήκευση», τα οποία μπορούν να προσαρμοστούν ανάλογα με τις ανάγκες της εφαρμογής. Τέλος, διαθέτει στατικές μεθόδους για την εμφάνιση των πιο συνηθισμένων τύπων μηνυμάτων με μια μόνο κλήση, κάνοντας τη χρήση της πολύ απλή. Στην Εικόνα 19 βλέπουμε πως «χτίζεται» ένα τέτοιο παράθυρο για να ενημερώσουμε τον χρήστη ότι η ενέργεια του ήταν επιτυχής.

```

291 void AddPersonDialog::createSuccessDialog() const
292 {
293     QMessageBox successBox;
294     successBox.setWindowTitle("Επιτυχία.");
295     QIcon windowIcon(":/argus/res/app_icons/plus.png");
296     successBox.setWindowIcon(windowIcon);
297     successBox.setText("Η καταχώρηση έγινε με αναγνωριστικό: " + id);
298     QPixmap successIcon(":/argus/res/dialog_icons/OK_DIALOG.png");
299     successBox.setIconPixmap(successIcon);
300     successBox.addButton(QMessageBox::Ok);
301     successBox.exec();
302 }

```

Εικόνα 19: Δημιουργία προσαρμοσμένου παραθύρου `QMessageBox`.

4.1.2.4.7 QMediaMetaData

Η συγκεκριμένη κλάση ανήκει στην βιβλιοθήκη Qt Multimedia η οποία παρέχει μια δομή για την αποθήκευση και την πρόσβαση σε μεταδεδομένα πολυμέσων. Τα μεταδεδομένα αυτά περιλαμβάνουν πληροφορίες όπως η ανάλυση, ο τίτλος, το μέγεθος του αρχείου και ο ρυθμός ανανέωσης των καρτέ ανά δευτερόλεπτο (framerate per second). Μια σημαντική λειτουργία είναι ότι χρησιμοποιεί προκαθορισμένα κλειδιά για κάθε πληροφορία ώστε η πρόσβαση στα στοιχεία αυτά να είναι ευκολότερη. Στην εφαρμογή, χρησιμοποιούμε την `QMediaMetaData` για να δείξουμε στον χρήστη πληροφορίες που είναι απαραίτητες για την παραμετροποίηση των λειτουργιών ανίχνευσης κίνησης και εντοπισμού προσώπων. Στην Εικόνα 20 παίρνουμε τις πληροφορίες για τον ρυθμό ανανέωσης των καρτέ.

```

664 void MainWindow::getMetadata()
665 {
666     textLogger->append("--Μεταδεδομένα--");
667     QMediaMetaData mData = mediaPlayer->metaData();
668     // Extract and display all metadata
669     QList<QMediaMetaData::Key> keys = mData.keys();
670     for (const QMediaMetaData::Key &key : keys)
671     {
672         QVariant value = mData.value(key);
673         if (key == QMediaMetaData::VideoFrameRate)
674         {
675             if (value.isValid())
676             {

```

Εικόνα 20: Χρήση της QMediaMetaData.

4.1.2.4.8 QSettings

Η QSettings προσφέρει έναν εύκολο και φορητό τρόπο για την αποθήκευση και την ανάκτηση των ρυθμίσεων της εφαρμογής. Οι ρυθμίσεις μπορεί να περιλαμβάνουν προτιμήσεις χρήστη, παραμέτρους διαμόρφωσης και άλλες πληροφορίες που πρέπει να αποθηκεύονται μεταξύ των εκτελέσεων της εφαρμογής. Με την QSettings στην εφαρμογή δημιουργείται ένα αρχείο παραμετροποιήσεων το οποίο περιέχει όλες τις τοποθεσίες αποθήκευσης δεδομένων καθώς και τις τοποθεσίες των αρχείων που χρειάζονται για τους αλγόριθμους αναγνώρισης προσώπων και εντοπισμού κίνησης. Χρησιμοποιεί ιεραρχική δομή για την οργάνωση των ρυθμίσεων, επιτρέποντας την εύκολη ομαδοποίηση σχετικών δεδομένων. Στην Εικόνα 21 βλέπουμε την συνάρτηση η οποία φορτώνει τις απαραίτητες ρυθμίσεις από το αρχείο όταν ανοίγει η εφαρμογή.

```

191 void MainWindow::loadConfiguration()
192 {
193     QSettings settings("config.ini", QSettings::IniFormat);
194     if (settings.contains("DNN_Configuration/Names_File"))
195     {
196         namesFilePath = settings.value("DNN_Configuration/Names_File").toString();
197     }
198     if (settings.contains("DNN_Configuration/Weights_File"))
199     {
200         weightsFilePath = settings.value("DNN_Configuration/Weights_File").toString();
201     }
202     if (settings.contains("DNN_Configuration/Config_File"))
203     {
204         configFilePath = settings.value("DNN_Configuration/Config_File").toString();
205     }

```

Εικόνα 21: Φόρτωση ρυθμίσεων με την χρήση της QSettings.

4.1.2.4.9 QFileDialog

Η QFileDialog παρέχει ένα τυπικό παράθυρο διαλόγου για την επιλογή αρχείων ή φακέλων. Χρησιμοποιείται για να επιτρέπει στους χρήστες να επιλέγουν αρχεία για άνοιγμα, αποθήκευση ή να επιλέγουν φακέλους ως τοποθεσίες αποθήκευσης. Υποστηρίζει διάφορες λειτουργίες και μπορεί να διαμορφωθεί ανάλογα με τις ανάγκες της εφαρμογής. Επιπλέον δίνεται η δυνατότητα ρύθμισης φίλτρων για την εμφάνιση μόνο συγκεκριμένων τύπων αρχείων.

```

86 void CaffeConfigDialog::openYunetFileClicked()
87 {
88     yunetFilePath = QFileDialog::getOpenFileName(this, tr("Επιλογή Αρχείου..."), "", tr("Αρχεία Μοντέλου (*.onnx)"));
89     yunetFLineEdit->setText(yunetFilePath);
90
91     if (yunetFilePath != originalYunetFilePath)
92     {
93         saveButton->setEnabled(true);
94     }
95 }

```

Εικόνα 22: Άνοιγμα διαλόγου επιλογής αρχείου με την QFileDialog.

Στην Εικόνα 22, χρησιμοποιούμε την QFileDialog για να επιλέξουμε το αρχείο που χρειάζεται ο αλγόριθμος εντοπισμού προσώπων.

4.1.2.4.10 QObject

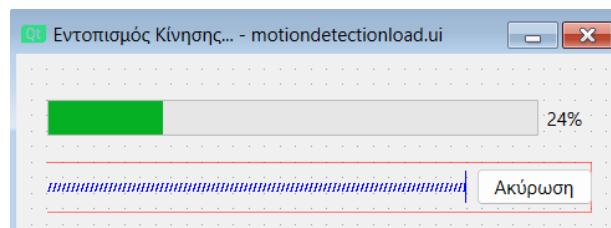
Η QObject είναι η θεμελιώδης κλάση του Qt που παρέχει τη βασική λειτουργικότητα για όλα τα αντικείμενα στο πλαίσιο Qt. Αποτελεί τη βάση για πολλές άλλες κλάσεις και προσφέρει σημαντικά χαρακτηριστικά όπως το μηχανισμό signals και slots, τη διαχείριση μνήμης μέσω ιδιοκτησίας (parent-child hierarchy), και τη δυνατότητα εκτέλεσης γεγονότων (event handling). Πολλές κλάσεις στην εφαρμογή Argus κληρονομούν από την QObject.

4.1.2.4.11 QToolBar

Η κλάση QToolBar είναι μέρος της βιβλιοθήκης Qt Widgets και χρησιμοποιείται για τη δημιουργία γραμμών εργαλείων σε γραφικές εφαρμογές. Μια γραμμή εργαλείων περιέχει ένα σύνολο από κουμπιά και άλλους διαδραστικούς ελέγχους, όπως μενού και πεδία εισαγωγής, που παρέχουν γρήγορη πρόσβαση στις λειτουργίες μιας εφαρμογής. Επιπλέον, παρέχει μεθόδους για την προσθήκη QAction αντικειμένων, τα οποία αντιπροσωπεύουν ενέργειες που μπορούν να εκτελεστούν, π.χ. κουμπιά με εικονίδια και κείμενο. Ένα παράδειγμα χρήσης μιας τέτοιας γραμμής εργαλείων βρίσκεται στο κεντρικό παράθυρο της εφαρμογής όπου υπάρχουν διάφορες λειτουργίες με ειδικά διαμορφωμένα εικονίδια. Στην εφαρμογή δεν υπάρχει παράδειγμα κώδικα για την γραμμή εργαλείων αφού παραμετροποιείται εξ' ολοκλήρου από τον Qt Designer που είναι ενσωματωμένος στο Visual Studio.

4.1.2.4.12 QProgressBar

Η κλάση QProgressBar στο Qt είναι ένα widget που χρησιμοποιείται για να δείχνει την πρόοδο μιας διαδικασίας. Εμφανίζεται ως μια οριζόντια μπάρα που γεμίζει καθώς η διαδικασία προχωράει. Είναι ιδανική για περιπτώσεις όπου θέλουμε να δώσουμε οπτική ανατροφοδότηση στον χρήστη σχετικά με την πρόοδο μιας μακρόχρονης λειτουργίας, όπως η φόρτωση αρχείων, η εγκατάσταση λογισμικού, η λήψη δεδομένων, κλπ. Στην εφαρμογή, βρίσκεται σε σημεία όπου ο χρήστης χρειάζεται να περιμένει μέχρι να ολοκληρωθεί μια διεργασία. Επιπλέον, υπάρχει η δυνατότητα προβολής μηνυμάτων προόδου ώστε να γνωρίζει ο χρήστης ανά πάσα στιγμή τι εκτελείται εκείνη την στιγμή.



Εικόνα 23: Προεπισκόπηση διαλόγου που περιέχει αντικείμενο QProgressBar.

4.1.2.4.13 QPainter

Η κλάση QPainter χρησιμοποιείται για τη σχεδίαση 2D γραφικών σε widgets και άλλες επιφάνειες σχεδίασης. Παρέχει μια μεγάλη ποικιλία από μεθόδους για σχεδίαση σχημάτων, κειμένων, εικόνων και άλλων γραφικών αντικειμένων. Στην εφαρμογή χρησιμοποιείται ως βοηθητική κλάση για τις λειτουργίες μεγέθυνσης και σμίκρυνσης στο παράθυρο προεπισκόπησης εικόνας.

```
749 QMainWindow::resizeEvent(event);
750 QRectF bounds = imagegraphicsScene->itemsBoundingRect();
751 imagegraphicsView->fitInView(bounds, Qt::KeepAspectRatio);
```

Εικόνα 24: Χρήση της QPainter.

4.1.2.4.14 QDir

Η κλάση QDir χρησιμοποιείται για την αλληλεπίδραση με το σύστημα αρχείων. Παρέχει μεθόδους για την πλοήγηση στους καταλόγους (directories), τη δημιουργία και διαγραφή καταλόγων, την καταχώρηση αρχείων και καταλόγων, καθώς και άλλες λειτουργίες σχετικές με το σύστημα αρχείων. Υποστηρίζει φιλτράρισμα των καταχωρήσεων με βάση κριτήρια όπως το αν είναι αρχεία ή κατάλογοι, καθώς και τη δυνατότητα ταξινόμησης με διάφορους τρόπους (π.χ., κατά όνομα, μέγεθος, ημερομηνία τροποποίησης). Στην εφαρμογή οι μέθοδοι της χρησιμοποιούνται για την διαχείριση των φακέλων αποθήκευσης των αποτελεσμάτων, αλλά και της βάσης δεδομένων των υπόπτων ώστε να γίνονται σωστά οι προσθήκες και οι διαγραφές των ατόμων στη βάση. Στον κώδικα της Εικόνας 25 ελέγχουμε με την βοήθεια της QDir αν ο φάκελος καταλόγου των υπόπτων είναι κενός.

```
45 bool CatalogService::isCatalogFolderEmpty()
46 {
47     QDir directory(catalogFolderPath);
48     QStringList entries = directory.entryList(QDir::NoDotAndDotDot | QDir::AllEntries);
49     return entries.isEmpty();
50 }
```

Εικόνα 25: Παράδειγμα χρήσης της QDir.

4.1.2.4.15 QFile

Η κλάση QFile χρησιμοποιείται για τη διαχείριση αρχείων. Παρέχει μεθόδους για την ανάγνωση, την εγγραφή και τον χειρισμό αρχείων στο σύστημα. Επιπλέον, διαθέτει μεθόδους για τον έλεγχο της ύπαρξης αρχείου (exists()), το μέγεθος του αρχείου (size()), την αφαίρεση (remove()), τη μετονομασία (rename()) και την αντιγραφή αρχείων (copy()). Στην εφαρμογή υπάρχει σε περιπτώσεις που απαιτείται οποιαδήποτε αλληλεπίδραση με αρχεία.

```
60 bool CatalogService::personExists(QString id)
61 {
62     QFile file(catalogFolderPath + "/catalog.txt");
63     if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
64     {
65         qDebug() << "Σφάλμα ανάγνωσης του αρχείου:" << file.errorString();
66         return false;
67     }
68 }
```

Εικόνα 26: Άνοιγμα αρχείου με την QFile.

4.1.2.4.16 QStringList

Η QStringList είναι μια κλάση που χρησιμοποιείται για την διαχείριση λιστών από QString αντικείμενα. Είναι ουσιαστικά, μια εξειδίκευση της κλάσης QList<QString> και παρέχει επιπλέον μεθόδους και λειτουργικότητα ειδικά για τη διαχείριση και τον χειρισμό λιστών από συμβολοσειρές. Στην εφαρμογή την συναντάμε στα κομμάτια του κώδικα που είναι υπεύθυνα για τον έλεγχο της βάσης δεδομένων, ή για την επιστροφή στοιχείων από την βάση τα οποία εμφανίζονται στην οθόνη.

```
168 std::string line;
169 while (std::getline(file, line))
170 {
171     QStringList tokens = QString::fromStdString(line).split(' ');
172     entries.push_back(tokens);
173 }
174 file.close();
175
176 return entries;
177 }
```

Εικόνα 27: Παράδειγμα χρήσης της QStringList σε ανάγνωση αρχείου.

4.1.2.4.17 QTextStream

Η QTextStream χρησιμοποιείται για την ανάγνωση και εγγραφή κειμένου σε ροές δεδομένων (streams). Επιτρέπει την εύκολη διαχείριση κειμένου μέσω ροών από και προς αρχεία, συμβολοσειρές, ή άλλες πηγές δεδομένων. Στην Εικόνα 28 ανοίγουμε ένα αρχείο σε λειτουργία εγγραφής κειμένου για την καταγραφή των στοιχείων ενός ατόμου. Για να πραγματοποιήσουμε εγγραφές κειμένου σε ένα αρχείο πρέπει πρώτα να δημιουργήσουμε ένα αντικείμενο QTextStream το οποίο συνδέεται με αυτό. Έτσι, μας δίνεται η δυνατότητα καταγραφής χρησιμοποιώντας τον τελεστή << μετά το αντικείμενο που ορίσαμε. Μόλις ολοκληρωθεί το κείμενο, κλείνουμε το αρχείο με τη μέθοδο close().

```

114 // Write the information to the ID.txt file.
115 QString fileName = folderName + "/" + id + ".txt";
116 QFile personFile(fileName);
117 if (!personFile.open(QIODevice::WriteOnly | QIODevice::Text))
118 {
119     return 1;
120 }
121 QTextStream personStream(&personFile);
122 personStream << name << "\n";
123 personStream << surname << "\n";
124 personStream << id << "\n";
125 personStream << dob.toString("yyyy-MM-dd") << "\n";
126 personStream << latinName << "\n";
127 personStream << latinSurname << "\n";
128 personStream << "No\n";
129 personStream << "0\n";
130 personStream << profilePicturePath;
131 personFile.close();

```

Εικόνα 28: Παράδειγμα χρήσης της QTextStream.

4.1.2.4.18 QToolButton

Η κλάση QToolButton στο Qt είναι μια εξειδικευμένη έκδοση του QAbstractButton που χρησιμοποιείται για την παροχή κουμπιών εργαλείων (tool buttons) σε γραφικές διεπαφές χρήστη. Τα κουμπιά αυτά χρησιμοποιούνται συνήθως σε γραμμές εργαλείων (toolbars) για την εκτέλεση συγκεκριμένων εντολών. Τα QToolButton μπορούν να εμφανίζουν εικόνες, κείμενο ή και τα δύο, και παρέχουν μεθόδους για την προσαρμογή της εμφάνισης και της συμπεριφοράς τους. Η παραμετροποίηση των κουμπιών μπορεί να γίνει και από τον Qt Designer χωρίς την χρήση κώδικα στον constructor της κλάσης. Στην Εικόνα 29 έχουμε τη σύνδεση ενός κουμπιού με μια μέθοδο.

```

160 buttonOpenFolder = ui->buttonOpenFolder;
161 connect(buttonOpenFolder, &QToolButton::clicked, this, &MainWindow::buttonOpenFolderClicked);

```

Εικόνα 29: Σύνδεση ενός QToolButton με μια μέθοδο.

4.1.2.4.19 QPushButton

Η QPushButton είναι μια κλάση που αντιπροσωπεύει ένα τυπικό κουμπί το οποίο μπορεί να χρησιμοποιηθεί σε γραφικές διεπαφές χρήστη για την πραγματοποίηση ενεργειών όταν ο χρήστης το πατήσει. Τα QPushButton χρησιμοποιούνται ευρέως σε φόρμες, διαλόγους και άλλα είδη παραθύρων για την εκκίνηση διαδικασιών, την υποβολή δεδομένων, και άλλες σχετικές λειτουργίες.

4.1.2.4.20 QSlider

Η QSlider είναι ένα γραφικό στοιχείο (widget) που επιτρέπει στους χρήστες να επιλέξουν μια τιμή από ένα καθορισμένο εύρος τιμών, σύροντας έναν δείκτη κατά μήκος μιας γραμμής. Είναι ιδιαίτερα χρήσιμη για την είσοδο τιμών όπως η ένταση ήχου, η φωτεινότητα όπως επίσης και

για την προβολή προόδου μιας διαδικασίας και άλλα παρόμοια δεδομένα τα οποία μπορούν να εκφραστούν με μια συνεχή ή διακριτή κλίμακα. Στην εφαρμογή χρησιμοποιούμε sliders για τον έλεγχο της έντασης στην αναπαραγωγή βίντεο, στην μπάρα διάρκειας του βίντεο με την οποία ο χρήστης μπορεί να επιλέξει μια συγκεκριμένη χρονική στιγμή καθώς και για την επιλογή τιμών στην παραμετροποίηση του αλγορίθμου ανίχνευσης κίνησης.



Εικόνα 30: Χρήση QSlider στο video player της εφαρμογής.

4.1.2.4.21 QListView

Η QListView χρησιμοποιείται για την εμφάνιση δεδομένων σε μορφή λίστας. Είναι μέρος της model-view αρχιτεκτονικής του Qt, η οποία διαχωρίζει τον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα από τον τρόπο που παρουσιάζονται στον χρήστη. Η QListView είναι ιδιαίτερα ευέλικτη και μπορεί να χρησιμοποιηθεί για την εμφάνιση λιστών αντικειμένων με διάφορες μορφές διάταξης. Παράδειγμα χρήσης αποτελεί το widget προβολής αρχείων στο κεντρικό παράθυρο της εφαρμογής το οποίο εμφανίζει μια λίστα με τις εικόνες που υπάρχουν στον φάκελο που έχουμε επιλέξει.



Εικόνα 31: Widget προβολής λίστας αρχείων.

4.1.2.4.22 QModelIndex

Η κλάση QModelIndex είναι ένα θεμελιώδες στοιχείο στην αρχιτεκτονική model-view. Αντιπροσωπεύει ένα δείκτη σε δεδομένα που βρίσκονται σε ένα μοντέλο δεδομένων (QAbstractItemModel). Οι QModelIndex δείκτες χρησιμοποιούνται για την πρόσβαση σε συγκεκριμένα στοιχεία δεδομένων και για την επικοινωνία μεταξύ των μοντέλων και της προβολής τους. Στο widget της λίστας στην [Εικόνα 31](#) για να συνδέσουμε την ενέργεια του διπλού κλικ επάνω σε ένα στοιχείο με το άνοιγμα της εικόνας στο παράθυρο προεπισκόπησης καλούμε ένα αντικείμενο της κλάσης QModelIndex. Συγκεκριμένα, βρίσκουμε ακριβώς ποιο στοιχείο πατήθηκε και λαμβάνουμε τις πληροφορίες τοποθεσίας του αρχείου (file path) για να εμφανιστεί στην οθόνη.

```

355 void MainWindow::onGalleryFileDoubleClicked(const QModelIndex &index)
356 {
357     if(toggleViewButtonState == true)
358     {
359         lastImagePath = galleryModel.fileInfo(index).absoluteFilePath();
360         loadImage(lastImagePath);
361     }
362     else if(toggleViewButtonState == false)
363     {
364         lastImagePath = galleryModel.fileInfo(index).absoluteFilePath();
365         return;
366     }
367 }
368
369

```

Εικόνα 32: Χρήση της QModelIndex.

4.1.2.4.23 QResizeEvent

Η `QResizeEvent` χρησιμοποιείται για την περιγραφή συμβάντων αλλαγής μεγέθους (resize events) σε γραφικά στοιχεία (widgets). Για παράδειγμα όταν ένας χρήστης αλλάζει μέγεθος σε μια εικόνα, δημιουργείται ένα `QResizeEvent` και αποστέλλεται στο γραφικό στοιχείο για να ενημερωθεί για την αλλαγή, επιτρέποντας του να προσαρμόζεται δυναμικά. Αυτό το συμβάν περιέχει πληροφορίες για το νέο και το παλιό μέγεθος του γραφικού στοιχείου. Στην εφαρμογή εμφανίζεται στην συνάρτηση `resizeEvent(QResizeEvent* event)` η οποία είναι υπεύθυνη για την επαναφορά της εικόνας στην αρχική της θέση στο παράθυρο προεπισκόπησης εικόνων.

```
747 void MainWindow::resizeEvent(QResizeEvent* event)
748 {
749     QMainWindow::resizeEvent(event);
750     QRectF bounds = imagegraphicsScene->itemsBoundingRect();
751     imagegraphicsView->fitInView(bounds, Qt::KeepAspectRatio);
752 }
```

Εικόνα 33: Επαναφορά εικόνας με την `QResizeEvent`.

4.1.2.4.24 QTextEdit

Η κλάση `QTextEdit` είναι ένα γραφικό στοιχείο (widget) που επιτρέπει την επεξεργασία και την εμφάνιση πλούσιου κειμένου (rich text). Υποστηρίζει λειτουργίες μορφοποίησης κειμένου, όπως γραμματοσειρά, στυλ, χρώμα και ευθυγράμμιση. Η `QTextEdit` μπορεί να χρησιμοποιηθεί για τη δημιουργία επεξεργαστών κειμένου, σημειώσεων και άλλων εφαρμογών που απαιτούν δυνατότητες επεξεργασίας κειμένου. Στην δική μας περίπτωση συναντάται ως γραφικό στοιχείο προβολής κειμένου για το ημερολόγιο εκτέλεσης του κεντρικού παραθύρου (με την μέθοδο `setReadOnly(true)`) καθώς και για εισαγωγή κειμένου σε φόρμες.



Εικόνα 34: Το ημερολόγιο εκτέλεσης.

4.1.2.4.25 QGraphicsPixmapItem

Η `QGraphicsPixmapItem` είναι μια εξειδικευμένη κλάση για την εμφάνιση εικόνων σε σκηνές γραφικών (graphics scenes). Αποτελεί μέρος του πλαισίου Graphics View Framework του Qt, το οποίο χρησιμοποιείται για την κατασκευή και την παρουσίαση σύνθετων γραφικών σκηνών. Παρέχει μεθόδους για την απεικόνιση εικόνων, όπως bitmaps ή pixmap, μέσα σε μια σκηνή γραφικών ([QGraphicsScene](#)). Υποστηρίζει λειτουργίες τοποθέτησης, περιστροφής, κλιμάκωσης (scaling) και άλλων μετασχηματισμών, επιτρέποντας την προσαρμογή της εμφάνισης της εικόνας στη σκηνή. Εξαιτίας αυτών των λειτουργιών μπορούμε να έχουμε μεγέθυνση και σμίκρυνση της εικόνας στο παράθυρο προεπισκόπησης.

4.1.2.4.26 QHBoxLayout

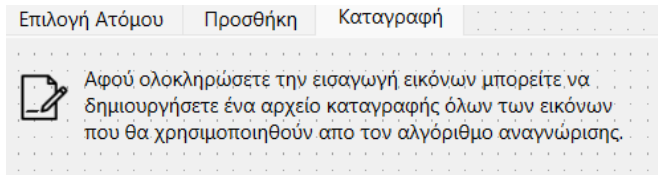
Η `QHBoxLayout` είναι μια διάταξη (layout) που οργανώνει τα γραφικά στοιχεία (widgets) οριζόντια, τοποθετώντας τα το ένα δίπλα στο άλλο σε μια γραμμή. Ανήκει στο πλαίσιο διατάξεων του Qt και χρησιμοποιείται για τη δημιουργία και διαχείριση των θέσεων και των μεγεθών των γραφικών στοιχείων μέσα σε ένα πλαίσιο (container), όπως ένα [QWidget](#).

4.1.2.4.27 QVBoxLayout

Ομοίως με την [QHBoxLayout](#) πρόκειται για μια διάταξη (layout) η οποία οργανώνει τα γραφικά στοιχεία το ένα κάτω από το άλλο σε μια στήλη.

4.1.2.4.28 QLabel

Η QLabel είναι ένα γραφικό στοιχείο που χρησιμοποιείται για την εμφάνιση κειμένου ή εικόνων σε ένα παράθυρο. Στην εφαρμογή βρίσκεται στα σημεία όπου εμφανίζεται κείμενο σε μορφή ετικέτας για την παρουσίαση κάποιας πληροφορίας στον χρήστη.



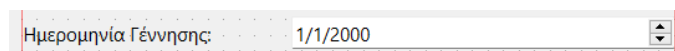
Εικόνα 35: Η εικόνα και το κείμενο είναι γραφικά στοιχεία QLabel.

4.1.2.4.29 QDate

Η QDate είναι μια κλάση που χρησιμοποιείται για την αναπαράσταση και διαχείριση ημερομηνιών. Παρέχει μια σειρά από λειτουργίες για τον χειρισμό ημερομηνιών, όπως η δημιουργία, η σύγκριση, η προσθήκη ή αφαίρεση ημερών, μηνών ή ετών, καθώς και ο έλεγχος εγκυρότητας των ημερομηνιών. Ο τύπος δεδομένων των ημερομηνιών γέννησης στη βάση δεδομένων είναι QDate.

4.1.2.4.30 QDateEdit

Η QDateEdit είναι ένα γραφικό στοιχείο που επιτρέπει στους χρήστες να εισάγουν και να επεξεργάζονται ημερομηνίες μέσω ενός πεδίου εισαγωγής και ενός ημερολογίου. Είναι μια εξειδικευμένη έκδοση της κλάσης QDateTimeEdit που επικεντρώνεται αποκλειστικά στη διαχείριση ημερομηνιών. Επιπλέον, επικυρώνει αυτόματα τις ημερομηνίες που εισάγονται από τον χρήστη για να εξασφαλίσει ότι είναι έγκυρες. Μέσω της QDateEdit μπορούμε να πάρουμε απευθείας δεδομένα τύπου [QDate](#) για τις ημερομηνίες γέννησης.



Εικόνα 36: Ένα γραφικό στοιχείο QDateEdit.

4.1.2.4.31 QIntValidator

Η QIntValidator περιορίζει την είσοδο κειμένου σε ένα πεδίο εισαγωγής (όπως [QLineEdit](#)) ώστε να δέχεται μόνο ακέραιους αριθμούς. Είναι μια υποκλάση της QValidator και παρέχει μεθόδους για τον καθορισμό ενός αποδεκτού εύρους τιμών για τους ακέραιους αριθμούς που μπορούν να εισαχθούν. Με την συνάρτηση setValidator(new QIntValidator) θέτουμε σε ένα γραφικό στοιχείο QLineEdit τον περιορισμό αυτόν.

Σημαντικό προτέρημα της συγκεκριμένης λειτουργίας είναι ότι μπορεί να κάνει δυναμικό έλεγχο της εισόδου στο QLineEdit. Αυτό γίνεται συνδέοντας το signal QLineEdit::textChanged με μια απλή συνάρτηση ελέγχου του κειμένου. Στην Εικόνα 36 παρουσιάζεται η συνάρτηση ελέγχου της εισόδου για το ID ενός ατόμου πριν εισαχθεί στην βάση δεδομένων.

```

193     else if (targetLineEdit == idEdit)
194     {
195         int num = text.toInt();
196         if (text.at(0) == 0)
197         {
198             idErrorLabel->setStyleSheet("QLabel { color : #ff00ff ; }");
199             idErrorLabel->setText("Το αναγνωριστικό δεν μπορεί να έχει το '0' πρώτα.");
200             idValid = false;
201         }
202         if (num < 1000 || num > 9999)
203         {
204             idErrorLabel->setStyleSheet("QLabel { color : #ff00ff ; }");
205             idErrorLabel->setText("Το αναγνωριστικό πρέπει να είναι από 1000 έως 9999.");
206             idValid = false;
207         }
208         if (num >= 1000 && num <= 9999)
209         {
210             idErrorLabel->setStyleSheet("QLabel { color : #000000 ; }");
211             idErrorLabel->setText("Το αναγνωριστικό είναι έγκυρο.");
212             id = text;
213             idValid = true;
214         }

```

Εικόνα 37: Συνάρτηση ελέγχου του ID.

4.1.2.4.32 QLineEdit

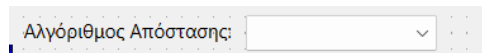
Η QLineEdit παρέχει ένα βασικό γραφικό στοιχείο που επιτρέπει στους χρήστες να εισάγουν και να επεξεργάζονται μία μόνο γραμμή κειμένου. Στην εφαρμογή χρησιμοποιείται σε φόρμες και σε σημεία όπου είναι απαραίτητη η εισαγωγή δεδομένων από τον χρήστη (παραμετροποιήσεις αλγορίθμων).



Εικόνα 38: Ένα γραφικό στοιχείο QLineEdit.

4.1.2.4.33 QComboBox

Η QComboBox είναι ένα γραφικό στοιχείο το οποίο παρέχει μια dropdown λίστα από προκαθορισμένες επιλογές. Με την συνάρτηση addItem("Επιλογή 1") μπορούμε να εισάγουμε τις επιλογές στο combo box.

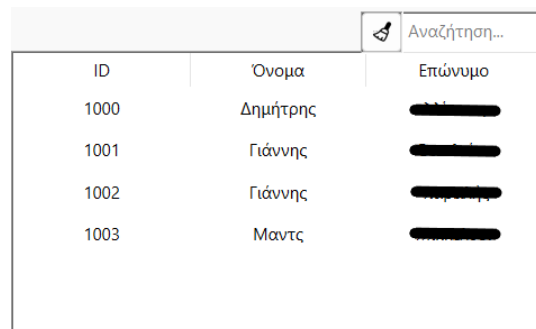


Εικόνα 39: Ένα γραφικό στοιχείο QComboBox.

4.1.2.4.34 QTableWidget

Η κλάση QTableWidget είναι ένα γραφικό στοιχείο πίνακα (table) για την εμφάνιση και επεξεργασία δεδομένων σε μορφή πλέγματος (grid). Είναι υποκλάση της QTableView και παρέχει λειτουργικότητα για την εύκολη διαχείριση και αλληλεπίδραση με τα δεδομένα του πίνακα χωρίς την ανάγκη χρήσης ενός ξεχωριστού μοντέλου δεδομένων, όπως απαιτεί η QTableView. Αποτελείται από ένα πλέγμα με σταθερές γραμμές και στήλες όπου κάθε κελί μπορεί να περιέχει κείμενο, εικόνες, κουμπιά και άλλα γραφικά στοιχεία. Είναι ιδανικό για την οπτικοποίηση της βάσης δεδομένων των ατόμων καθώς και για την παρουσίαση των αποτελεσμάτων του αλγορίθμου αναγνώρισης προσώπων στον χρήστη.

Εκτός από την οπτικοποίηση δεδομένων η κλάση αυτή διαθέτει και signals τα οποία εκπέμπονται όταν ο χρήστης πατάει σε μια συγκεκριμένη γραμμή του πίνακα. Χρησιμοποιώντας αυτή την λειτουργικότητα μπορούμε να εμφανίσουμε πληροφορίες ή να ενεργοποιήσουμε κουμπιά μόλις γίνεται κάποιο κλικ.



ID	Όνομα	Επώνυμο
1000	Δημήτρης	██████████
1001	Γιάννης	██████████
1002	Γιάννης	██████████
1003	Μαντς	██████████

Εικόνα 40: Ένα γραφικό στοιχείο QTableWidgetItem.

4.1.2.4.35 QFrame

Η κλάση QFrame διαχειρίζεται τα πλαίσια (frames) γύρω από άλλα γραφικά στοιχεία ή περιεχόμενο. Πιο συγκεκριμένα χρησιμεύει ως βάση για την κατασκευή πιο σύνθετων γραφικών στοιχείων και μπορεί να χρησιμοποιηθεί για να διαχωρίσει οπτικά, τα τμήματα μιας διεπαφής χρήστη.

4.1.2.4.36 QPixmap

Η QPixmap είναι μια κλάση διαχείρισης γραφικών και εικόνων η οποία είναι βελτιστοποιημένη για την αποθήκευση και την εμφάνιση εικόνων στις συσκευές εξόδου. Μπορεί να φορτώσει, να αποθηκεύσει και να χειριστεί εικόνες από διάφορες μορφές αρχείων, όπως PNG, JPEG, BMP, κ.λπ. Τέλος, σε αντίθεση με την QImage, η οποία είναι πιο κατάλληλη για επεξεργασία εικόνων σε χαμηλό επίπεδο, η QPixmap παρέχει αποδοτική διαχείριση μνήμης σε γραφικά περιβάλλοντα. Οποιαδήποτε εικόνα εμφανίζεται στο γραφικό περιβάλλον του Argus είναι αντικείμενο της κλάσης αυτής.

4.1.2.4.37 QFileInfo

Μια κλάση που παρέχει πληροφορίες σχετικά με τα μεταδεδομένα των αρχείων. Έχει παρόμοιες λειτουργίες με την [QMediaMetaData](#) με την διαφορά ότι επικεντρώνεται στο κομμάτι των καταλόγων και των αρχείων. Χρησιμοποιείται για την ανάκτηση και τη διαχείριση μεταδεδομένων, όπως το μέγεθος, τις ημερομηνίες δημιουργίας και τροποποίησης, την ύπαρξη του καταλόγου (ή αρχείου) και τα δικαιώματα πρόσβασης. Στην εφαρμογή μας είναι υπεύθυνη για την εύρεση της τοποθεσίας των αρχείων και για την σωστή κατανομή τους στην τοπική βάση δεδομένων σε συνεργασία με την [QFile](#) και [QDir](#).

Στον κώδικα της Εικόνας 40 χρησιμοποιούμε τη μέθοδο fileName() για να εξαγάγουμε το μονοπάτι του αρχείου εικόνας προφίλ και να το συνδέσουμε με το όνομα του φακέλου ώστε να παράγουμε το μονοπάτι προορισμού.

```

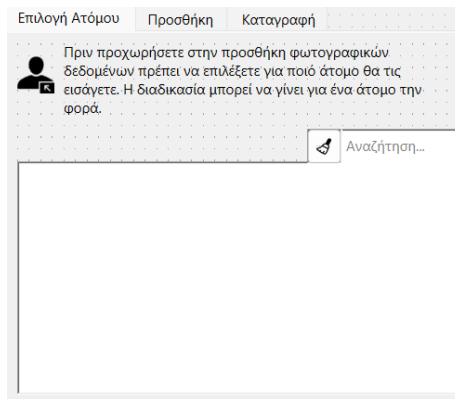
122 // Copy the profile picture to the appropriate folder.
123 QString destinationPath = folderName + "/" + QFileInfo(profilePicturePath).fileName();
124 if (QFile::copy(profilePicturePath, destinationPath))
125 {
126     qDebug() << "Profile picture copied successfully to:" << destinationPath;
127 }
128 else
129 {
130     qDebug() << "Failed to copy profile picture to:" << destinationPath;
131 }

```

Εικόνα 41: Παράδειγμα χρήσης της QFileInfo.

4.1.2.4.38 QTabWidget

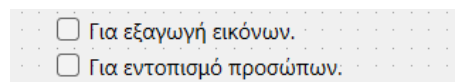
Η κλάση QTabWidget ένα container widget που χρησιμοποιείται για τη διαχείριση πολλαπλών σελίδων περιεχομένου. Κάθε σελίδα παρουσιάζεται ως ένα tab και οι χρήστες μπορούν να πλοηγηθούν μεταξύ των διαφορετικών σελίδων επιλέγοντας το αντίστοιχο tab. Το QTabWidget είναι ιδιαίτερα χρήσιμο για την οργάνωση περιεχομένου σε περιορισμένο χώρο, προσφέροντας παράλληλα μια ευχάριστη εμπειρία χρήστη. Στην εφαρμογή εμφανίζεται σε παράθυρο διαλόγου που έχουν πολλαπλές λειτουργίες και είναι αναγκαίος ο διαχωρισμός μεταξύ αυτών. Για παράδειγμα, στο παράθυρο διαλόγου που είναι υπεύθυνο για το ανέβασμα εικόνων στη βάση δεδομένων οι λειτουργίες χωρίζονται με βάση την σειρά που πρέπει να γίνουν.



Εικόνα 42: Ένα γραφικό στοιχείο QTabWidget.

4.1.2.4.39 QCheckBox

Η QCheckBox παρέχει ένα γραφικό στοιχείο επιλογής το οποίο έχει δύο καταστάσεις (επιλεγμένο ή μη επιλεγμένο) μαζί με μια ετικέτα. Χρησιμοποιείται κυρίως την παρουσίαση επιλογών στον χρήστη που μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν ανεξάρτητα. Πιο συγκεκριμένα, QCheckBox συναντάμε στα σημεία όπου ο χρήστης καλείται να επιλέξει τον τρόπο αποθήκευσης δεδομένων στις λειτουργίες εξαγωγής κίνησης, εντοπισμού προσώπων και αναγνώρισης προσώπων.



Εικόνα 43: Γραφικά στοιχεία QCheckBox.

4.1.2.4.40 QModelIndexList

Η συγκεκριμένη κλάση είναι ένας τύπος δεδομένων που χρησιμοποιείται για την αναπαράσταση λίστας από αντικείμενα QModelIndex. Η QModelIndexList βρίσκεται σε πολύ περιορισμένη χρήση στην εφαρμογή, συγκεκριμένα σε σημεία όπου θέλουμε να πάρουμε συγκεκριμένα indexes ενός πίνακα. Παραδείγματος χάριν, όταν επιλέγουμε να διαβάσουμε ένα αρχείο εικόνας ενός ατόμου από τον πίνακα ο οποίος προβάλλει μια λίστα με τα περιεχόμενα της βάσης φωτογραφικού υλικού.

```
void DatasetDialog::onDeleteButtonClicked()
{
    QModelIndexList selectedIndexes = datasetDisplayTableWidget->selectionModel()->selectedIndexes();
    int row = selectedIndexes.first().row();
    QTableWidgetItem* selectedItem = datasetDisplayTableWidget->item(row, 0);
}
```

Εικόνα 44: Χρήση της QModelIndexList για την επιλογή συγκεκριμένης γραμμής του QTableWidgetItem.

4.1.2.4.41 QSpinBox

Η QSpinBox είναι ένα γραφικό στοιχείο που επιτρέπει στον χρήστη να επιλέξει μια αριθμητική τιμή από ένα εύρος τιμών χρησιμοποιώντας είτε βέλη αύξησης/μείωσης είτε πληκτρολογώντας την τιμή απευθείας. Το QSpinBox είναι ιδανικό για τη λήψη αριθμητικών εισόδων από τον χρήστη με ένα καθορισμένο εύρος και βήμα. Επιπλέον, η δυνατότητα εκπομπής σημάτων όταν η τιμή αλλάζει, επιτρέπει την εύκολη αλληλεπίδραση με άλλες συναρτήσεις και widgets της εφαρμογής.



Εικόνα 45: Ένα γραφικό στοιχείο QSpinBox.

4.1.2.4.42 QThread

Η κλάση QThread παρέχει έναν μηχανισμό για την εκτέλεση κώδικα σε ένα ξεχωριστό νήματα (threads). Τα threads επιτρέπουν την εκτέλεση πολλαπλών εργασιών ταυτόχρονα, εκμεταλλευόμενα τις δυνατότητες των σύγχρονων πολυπύρηνων επεξεργαστών για να βελτιώσουν την απόδοση και την ανταπόκριση των εφαρμογών. Το πρόβλημα που οδήγησε στην χρήση της QThread αφορούσε την αδυναμία ανταπόκρισης της διεπαφής χρήστη όταν το σύστημα εκτελούσε απαιτητικές διεργασίες. Πιο συγκεκριμένα, σε περίπτωση που ένας χρήστης επέλεγε να εκτελέσει την λειτουργία εντοπισμού κίνησης, η διεπαφή χρήστη και κατά συνέπεια το παράθυρο που ενημέρωνε τον χρήστη για την πρόοδο (βλ. [QProgressBar](#)) σταματούσε να ανταποκρίνεται (freezing). Αυτό συνέβαινε εξ' αιτίας της πολυπλοκότητας των διεργασιών που «έτρεχαν» πάνω στο ίδιο νήμα το οποίο είχε ως αποτέλεσμα να σταματάει οποιαδήποτε αλλαγή στο UI ώστε να ολοκληρωθεί πρώτα ο εντοπισμός κίνησης. Έτσι, για να αντιμετωπίσουμε το πρόβλημα αυτό έγινε χρήση των δυνατοτήτων της QThread ώστε να γίνεται παράλληλη εκτέλεση των διεργασιών που είναι υπεύθυνες για τον χειρισμό της διεπαφής και των μεθόδων που πραγματοποιούσαν τον εντοπισμό κίνησης.

Μερικά βασικά χαρακτηριστικά της QThread είναι:

- Επιτρέπει τη δημιουργία, την εκκίνηση, τη διακοπή και την παρακολούθηση της κατάστασης των νημάτων.
- Μπορεί να χρησιμοποιηθεί για την εκτέλεση χρονοβόρων λειτουργιών στο παρασκήνιο, χωρίς να εμποδίζεται το κύριο νήμα της εφαρμογής.
- Υποστηρίζει πλήρως το σύστημα slots and signals, επιτρέποντας την επικοινωνία μεταξύ νημάτων και αντικειμένων.
- Κάθε QThread μπορεί να έχει το δικό του event loop, το οποίο επιτρέπει την εκτέλεση συμβάντων μόνο μέσα στο πλαίσιο του νήματος (ξεχωριστά από το κύριο νήμα).
- Παρέχει μηχανισμούς για την ασφαλή πρόσβαση σε κοινόχρηστους πόρους μεταξύ νημάτων.

Οι βασικές μέθοδοι που χρησιμοποιούνται στην εφαρμογή και ανήκουν στην QThread είναι:

- **start():** Ξεκινά την εκτέλεση του νήματος.
- **run():** Περιέχει τον κώδικα που θα εκτελεστεί στο νέο νήμα. Μπορεί να παρακαμφθεί σε κλάσεις που κληρονομούν από την QThread.
- **quit():** Σταματά το event loop του νήματος.
- **wait():** Περιμένει να τερματιστεί το νήμα.
- **isRunning():** Επιστρέφει true αν το νήμα είναι σε λειτουργία.
- **isFinished():** Επιστρέφει true αν το νήμα έχει τερματιστεί.

Όπως θα δούμε και στην Εικόνα 45 για να ξεκινήσουμε μια διεργασία μέσω ενός καινούργιου νήματος πρέπει να περάσουμε όλες τις παραμέτρους που έχει επιλέξει ο χρήστης μέσω της εφαρμογής αφού έχουμε δημιουργήσει ένα ξεχωριστό αντικείμενο για τον χειρισμό του thread.

```
94 void FaceDetectionDialog::startButtonClicked()
95 {
96     FaceDetectionDialog::setParams();
97     FaceDetection init = new FaceDetection();
98     init.start(this, loadedVideoPath, saveToVideoFlag, saveToImageFlag, saveToTxtFlag,
99             scaleFactor, confidenceThreshold, yConfidenceThreshold, nmsThreshold, detectionCount, selectedModel);
100 }
```

Εικόνα 46: Δημιουργία αντικειμένου για το setup του νήματος.

Έπειτα πρέπει να συνδέσουμε όλα τα slots με τα κατάλληλα signals για να καταστήσουμε την επικοινωνία μεταξύ του κύριου νήματος και του νέου. Στο αρχείο της κλάσης που είναι υπεύθυνο για την κλήση του thread κάνουμε τις κατάλληλες συνδέσεις, εισάγουμε τις παραμέτρους και χρησιμοποιούμε την start() για την εκκίνηση.

```
23 void FaceDetection::start(FaceDetectionDialog* initDialog, QString videoPath,
24 bool saveToVideoFlag, bool saveToImageFlag, bool saveToTxtFlag, double scaleFactor,
25 double confidenceThreshold, double yConfidenceThreshold, double nmsThreshold, int detectionCount,
26 QString selectedModel)
27 {
28     FaceDetectionThread* thread = new FaceDetectionThread(videoPath,
29         saveToVideoFlag, saveToImageFlag, saveToTxtFlag, scaleFactor, confidenceThreshold, yConfidenceThreshold,
30         nmsThreshold, detectionCount, selectedModel);
31     FaceDetection::initializeLoadingDialog();
32     connect(thread, &FaceDetectionThread::started, initDialog, &FaceDetectionDialog::disableButtons);
33     connect(loadDialog, &MotionDetectionLoad::cancelClicked, thread, &FaceDetectionThread::threadQuit);
34     connect(thread, &FaceDetectionThread::loadingProgress, loadDialog, &MotionDetectionLoad::updateProgress);
35     connect(thread, &FaceDetectionThread::currentStatus, loadDialog, &MotionDetectionLoad::updateStatus);
36     connect(thread, &FaceDetectionThread::resetProgress, loadDialog, &MotionDetectionLoad::resetProgress);
37     connect(thread, &FaceDetectionThread::threadQuitRequested, loadDialog, &MotionDetectionLoad::onCompletion);
38     connect(thread, &FaceDetectionThread::threadQuitRequested, initDialog, &FaceDetectionDialog::enableButtons);
39     connect(thread, &FaceDetectionThread::finished, thread, &FaceDetectionThread::deleteLater);
40     thread->start();
41 }
```

Εικόνα 47: Δημιουργία αντικειμένου και εκκίνηση του νήματος.

Τέλος χρειαζόμαστε μια ακόμη κλάση η οποία κληρονομεί την QThread και στην οποία χρησιμοποιούμε την run() για να εκτελέσουμε τον κώδικα παράλληλα με το κύριο νήμα.

```
22 void FaceDetectionThread::run()
23 {
24     returnCode = FaceDetectionThread::detectFacesDNN(videoPath,
25         saveToVideoFlag,
26         saveToImageFlag,
27         saveToTxtFlag,
28         selectedModel);
29
30     if (returnCode == 0)
31     {
32         emit threadQuitRequested();
33         quit();
34     }
35 }
```

Εικόνα 48: Η συνάρτηση run() εκτελεί το event loop του νήματος.

4.1.2.4.43 QValidator

Η κλάση QValidator είναι μια abstract κλάση που παρέχει ένα interface για τον έλεγχο της εγκυρότητας των δεδομένων εισόδου. Χρησιμοποιείται κυρίως με widgets εισόδου όπως τα [QLineEdit](#), [QSpinBox](#), και άλλα για να διασφαλίσει ότι οι τιμές που εισάγει ο χρήστης είναι έγκυρες σύμφωνα με καθορισμένους κανόνες. Στην εφαρμογή δεν χρησιμοποιούμε την ίδια την QValidator ως έχει αλλά σαν επέκταση για την δημιουργία ελέγχων εγκυρότητας ελληνικών και λατινικών χαρακτήρων στο παράθυρο προσθήκης ατόμων στη βάση δεδομένων.

4.1.2.4.44 QChar

Η κλάση QChar στο Qt είναι μια κλάση που αντιπροσωπεύει έναν μεμονωμένο χαρακτήρα Unicode. Χρησιμοποιείται για την αποθήκευση και τη διαχείριση χαρακτήρων και παρέχει ένα σύνολο μεθόδων για τον χειρισμό και την επεξεργασία χαρακτήρων. Η QChar μπορεί να περιέχει οποιοδήποτε χαρακτήρα Unicode και προσφέρει λειτουργίες για τη σύγκριση, τον μετασχηματισμό και τον έλεγχο της κατηγορίας των χαρακτήρων.

```
19 sortComboBox = ui->sortComboBox;
20 sortComboBox->addItem("ID " + QString(QChar(0x2191)) + " (Αύξουσα)");
21 sortComboBox->addItem("ID " + QString(QChar(0x2193)) + " (Φθίνουσα)");
```

Εικόνα 49: Χρήση της QChar για την εμφάνιση ειδικών Unicode χαρακτήρων.

4.1.2.4.45 QGraphicsView

Η κλάση QGraphicsView είναι μέρος του γραφικού πλαισίου QGraphics View Framework και χρησιμοποιείται για την προβολή και την αλληλεπίδραση με στοιχεία γραφικών που είναι τοποθετημένα σε μια [QGraphicsScene](#). Η QGraphicsView παρέχει έναν widget-based τρόπο ώστε ο χρήστης να μπορεί να αλληλοεπιδρά με το παράθυρο προεπισκόπησης εικόνων.

4.1.2.4.46 QGraphicsScene

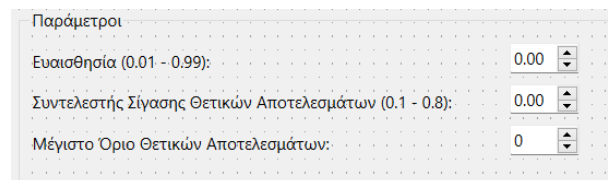
Η κλάση QGraphicsScene είναι το κύριο δοχείο για τη διαχείριση και την απόδοση γραφικών αντικειμένων σε μια σκηνή. Είναι και αυτή μέρος του γραφικού πλαισίου QGraphics View Framework και χρησιμοποιείται σε συνδυασμό με την κλάση [QGraphicsView](#) για να παρουσιάζει τα γραφικά αντικείμενα και να διαχειρίζεται την αλληλεπίδραση με αυτά.

4.1.2.4.47 QDebug

Η QDebug είναι ένα εργαλείο για τη δημιουργία μηνυμάτων καταγραφής (logging messages). Παρέχει έναν εύκολο τρόπο για την εκτύπωση μηνυμάτων στο τερματικό που βοηθούν τους προγραμματιστές κατά τη διάρκεια του debugging και της ανάπτυξης της εφαρμογής. Χρησιμοποιώντας τον τελεστή «<<» και την μέθοδο qDebug() μπορούμε να εκτυπώσουμε μηνύματα στο τερματικό του Visual Studio.

4.1.2.4.48 QGroupBox

Η QGroupBox προσφέρει ένα γραφικό στοιχείο πλαισίου με μια ετικέτα συνήθως στο πάνω αριστερό σημείο. Είναι ένας απλός τρόπος διαχωρισμού των γραφικών στοιχείων σε ένα παράθυρο διαλόγου για την δημιουργία μιας καθαρής και οργανωμένης διάταξης.



Εικόνα 50: Ένα πλαίσιο QGroupBox με 3 στοιχεία.

4.1.3 Γλώσσα Προγραμματισμού C++

4.1.3.1 Γιατί επιλέχθηκε η C++;

Ο λόγος επιλογής της C++ βασίστηκε κυρίως σε τέσσερις παράγοντες, την διαθεσιμότητα των εργαλείων, την ευκολία χρήσης, την απόδοση και την οργάνωση του κώδικα.

- Διαθεσιμότητα εργαλείων

Λόγω της διαδεδομένης φύσης της συγκεκριμένης γλώσσας προγραμματισμού, υπάρχει μεγάλη ποικιλία στη γκάμα των διαθέσιμων εργαλείων. Πανίσχυρες πλατφόρμες όπως το περιβάλλον ανάπτυξης Qt και οι ικανότητες debugging που προσφέρει το Visual Studio καθιστούν αυτή την επιλογή ακόμα πιο ελκυστική. Επιπλέον υπάρχει πληθώρα βιβλιοθηκών και frameworks που διευκολύνουν την ανάπτυξη και την ενσωμάτωση πολύπλοκων λειτουργιών.

- Ευκολία χρήσης

Παρόλο που η C++ μπορεί να φαίνεται δύσκολη σε νέους προγραμματιστές, η πολυπλοκότητα της ανταμείβει τους πιο έμπειρους χρήστες. Προσφέρει μεγάλη ευελιξία και έλεγχο, η σύνταξη της είναι σαφής και επιτρέπει τόσο την υψηλού επιπέδου προγραμματιστική προσέγγιση όσο και την ικανότητα διαχείρισης της μνήμης. Στην συγκεκριμένη περίπτωση αν και η Python επισκιάζει σε ευκολία την C++, το μέγεθος της εφαρμογής και η πολυπλοκότητα καθιστούν την χρήση της μη ιδανική.

- Απόδοση

Η C++ είναι γνωστή για την υψηλή της απόδοση και την ικανότητα να εκμεταλλεύεται πλήρως τους πόρους του υλικού. Με την δυνατότητα χρήσης των δεικτών (pointers) και της άμεσης διαχείρισης μνήμης, μπορεί να επιτύχει ταχύτητα και αποτελεσματικότητα που δύσκολα βρίσκει κανείς σε άλλες γλώσσες προγραμματισμού. Η απόδοση αυτή είναι κρίσιμη για εφαρμογές που περιέχουν πολύπλοκες μεθόδους και επεξεργάζονται δεδομένα μεγάλου όγκου.

- Οργάνωση του κώδικα

Η C++ υποστηρίζει διάφορα προγραμματιστικά πρότυπα και μεθοδολογίες, συμπεριλαμβανομένης της αντικειμενοστραφούς προγραμματιστικής προσέγγισης, που επιτρέπει την καθαρή και καλά δομημένη οργάνωση του κώδικα. Οι δυνατότητες που προσφέρει για την χρήση templates και προτύπων, κάνουν την γλώσσα εξαιρετικά ευέλικτη και επεκτάσιμη, διασφαλίζοντας τη δυνατότητα επαναχρησιμοποίησης του κώδικα και την εύκολη συντήρησή του.

Εκτός από τους παραπάνω λόγους, αξίζει να σημειωθεί ότι διαθέτει μια από τις μεγαλύτερες και πιο ενεργές κοινότητες προγραμματιστών στον κόσμο. Αυτό σημαίνει ότι υπάρχει πληθώρα διαθέσιμων παραδειγμάτων κώδικα, λύσεων σε κοινά προβλήματα και συνεχούς υποστήριξης από άλλους χρήστες.

4.1.3.1.1 Σύγκριση μεταξύ C++ και Python

Κατά την διάρκεια της μελέτης των απαιτήσεων της εφαρμογής, τέθηκε το ζήτημα της επιλογής της γλώσσας ανάπτυξης της εφαρμογής. Τα εργαλεία ανάπτυξης ήταν διαθέσιμα και στις δύο γλώσσες οπότε έπρεπε να λάβουμε υπόψιν τα πλεονεκτήματα και τα μειονεκτήματα της κάθε επιλογής.

Πλεονεκτήματα της C++:

1. Υψηλή απόδοση: Η C++ είναι γνωστή για την ταχύτητά της και την αποτελεσματική διαχείριση πόρων. Η δυνατότητα άμεσης διαχείρισης μνήμης και η χρήση pointers επιτρέπουν τη δημιουργία εξαιρετικά αποδοτικών εφαρμογών.
2. Χαμηλού επιπέδου προγραμματισμός: Προσφέρει δυνατότητες χαμηλού επιπέδου διαχείρισης μνήμης, κάτι που είναι χρήσιμο για συστήματα με περιορισμένους πόρους ή απαιτητικές εφαρμογές.
3. Βιομηχανικό πρότυπο: Χρησιμοποιείται ευρέως σε βιομηχανικές εφαρμογές, παιχνίδια, λειτουργικά συστήματα, και real-time συστήματα.
4. Μεγάλη κοινότητα και πλούσιο οικοσύστημα εργαλείων: Υπάρχουν πολλές βιβλιοθήκες, frameworks και εργαλεία, όπως Qt, Boost, και Visual Studio, που διευκολύνουν την ανάπτυξη.

Μειονεκτήματα της C++:

1. Πολυπλοκότητα μάθησης: Η C++ μπορεί να είναι δύσκολη για νέους προγραμματιστές λόγω της πολυπλοκότητάς της και της ανάγκης για διαχείριση μνήμης.
2. Μακρύτερος χρόνος ανάπτυξης: Ο κώδικας σε C++ μπορεί να είναι πιο πολύπλοκος και να απαιτεί περισσότερο χρόνο για να γραφτεί και να συντηρηθεί.

Πλεονεκτήματα της Python:

1. Ευκολία μάθησης και χρήσης: Η Python έχει απλή και κατανοητή σύνταξη, κάνοντάς την ιδανική για νέους προγραμματιστές και γρήγορη ανάπτυξη εφαρμογών.
2. Γρήγορη ανάπτυξη: Η δυναμική φύση της Python επιτρέπει τη γρήγορη ανάπτυξη και την ευκολία στη συντήρηση του κώδικα.
3. Πλούσια βιβλιοθήκη και κοινότητα: Διαθέτει πληθώρα βιβλιοθηκών (π.χ. NumPy, Pandas, TensorFlow) και μεγάλη κοινότητα υποστήριξης.
4. Διαλειτουργικότητα: Μπορεί να ενσωματωθεί εύκολα με άλλες γλώσσες και συστήματα.

Μειονεκτήματα της Python:

1. Απόδοση: Η Python είναι συνήθως πιο αργή από τη C++ λόγω της interpreted φύσης της και της διαχείρισης μνήμης από τον garbage collector.
2. Χρήση πόρων: Μπορεί να απαιτεί περισσότερους πόρους συστήματος, κάνοντάς την λιγότερο κατάλληλη για εφαρμογές real-time ή με περιορισμένους πόρους.
3. Τυποποίηση: Η έλλειψη αυστηρής τυποποίησης μπορεί να οδηγήσει σε σφάλματα που ανακαλύπτονται μόνο κατά την εκτέλεση.

Συμπερασματικά, παρόλο που η Python είναι εξαιρετική για γρήγορη ανάπτυξη και εύκολη συντήρηση, η C++ παραμένει η προτιμώμενη επιλογή για projects που απαιτούν υψηλή απόδοση και λεπτομερή έλεγχο των πόρων του συστήματος.

4.1.3.2 Γενικές Πληροφορίες

Η γλώσσα προγραμματισμού C++ είναι ένα ισχυρό εργαλείο που δημιουργήθηκε από τον Bjarne Stroustrup και έκανε την πρώτη της εμφάνιση το 1983 ως "C with Classes". Το 1985 κυκλοφόρησε η πρώτη επίσημη έκδοση της C++, η οποία εισήγαγε βασικές δυνατότητες όπως οι κλάσεις, η κληρονομικότητα και η δυναμική διαχείριση μνήμης. Ο αρχικός στόχος του Stroustrup ήταν να προσθέσει δυνατότητες αντικειμενοστραφούς προγραμματισμού στη γλώσσα C, διατηρώντας ταυτόχρονα την αποδοτικότητα και την ευελιξία της.

Η εξέλιξη της γλώσσας συνεχίστηκε με την πρώτη τυποποιημένη έκδοση της C++ το 1998 από τον οργανισμό ISO (ISO/IEC 14882:1998), γνωστή ως C++98. Αυτή η έκδοση παρείχε μια σταθερή βάση για την ανάπτυξη της γλώσσας και τη χρήση της σε βιομηχανικό επίπεδο. Το 2003, η C++03 έκανε μικρές διορθώσεις και βελτιώσεις στην C++98. Μια σημαντική αναθεώρηση ήρθε το 2011 με την κυκλοφορία της C++11, η οποία εισήγαγε πολλές νέες δυνατότητες όπως το `auto`, τους έξυπνους δείκτες (smart pointers) και βελτιώσεις στη βιβλιοθήκη STL (Standard Template Library). Οι αναθεωρήσεις συνεχίστηκαν με την C++14 και την C++17, που κυκλοφόρησαν το 2014 και το 2017 αντίστοιχα, προσφέροντας νέες δυνατότητες και βελτιώσεις στην απόδοση. Η C++20, που κυκλοφόρησε το 2020, προσέθεσε χαρακτηριστικά όπως τα `concepts`, `ranges` και `coroutines`.



Εικόνα 51: Το λογότυπο της C++.

4.1.3.3 Microsoft Visual C++

Η εφαρμογή Argus αναπτύχθηκε με στόχο την απόδοση και την σωστή λειτουργία σε συστήματα που χρησιμοποιούν Windows. Έτσι, έπρεπε να γίνει επιλογή κατάλληλου compiler για καλύτερη απόδοση στο συγκεκριμένο περιβάλλον. Οι δύο βασικότεροι compilers για την C++, ο GNU GCC και ο MSVC προσφέρουν διαφορετικά πλεονεκτήματα και μειονεκτήματα. Το βασικότερο προτέρημα του GCC είναι ότι υποστηρίζει πολλαπλά λειτουργικά συστήματα οπότε είναι ευρέως χρησιμοποιούμενος σε περιβάλλοντα Unix και Linux, επομένως ταιριάζει περισσότερο σε projects ανοικτού κώδικα.

Από την άλλη μεριά ο MSVC, είναι ο μεταγλωττιστής της Microsoft για τη γλώσσα C και C++. Αναπτύσσεται και συντηρείται από τη Microsoft και ενσωματώνεται πλήρως στο Visual Studio. Παρέχει μια ολοκληρωμένη λύση για την ανάπτυξη λογισμικού, προσφέροντας εργαλεία για σύνταξη, μεταγλώττιση, εντοπισμό σφαλμάτων (debugging) και ανάλυση απόδοσης.

Στη φάση της αναζήτησης εργαλείων ανάπτυξης για την εφαρμογή τέθηκε το ζήτημα ότι όλα τα εργαλεία έπρεπε να είναι compiled με τον ίδιο compiler αλλιώς, υπήρχε μεγάλη πιθανότητα ασυμβατότητας μεταξύ των βιβλιοθηκών. Έτσι επιλέχθηκε ο MSVC γιατί ήταν ευκολότερη η ενσωμάτωση των βιβλιοθηκών στο Visual Studio αλλά και πιο σίγουρη η διαδικασία «χτισίματος» (build) των βιβλιοθηκών υπολογιστικής όρασης από τον πηγαίο κώδικα.

4.1.3.4 Βιβλιοθήκες που Χρησιμοποιήθηκαν

4.1.3.4.1 vector

Η βιβλιοθήκη vector στη C++ είναι μέρος της Standard Template Library (STL) και παρέχει μια δυναμική δομή δεδομένων που μοιάζει με πίνακα, αλλά με τη δυνατότητα αλλαγής μεγέθους κατά την εκτέλεση. Ο τύπος `std::vector` είναι μια template κλάση, που σημαίνει ότι μπορεί να κρατήσει οποιονδήποτε τύπο δεδομένων. Εξαιτίας αυτής της ιδιότητας να αποθηκεύει οποιονδήποτε τύπο δεδομένων είναι κατάλληλη για πιο σύνθετα δεδομένα από τις βιβλιοθήκες της OpenCV όπως αντικείμενα τύπου `Mat` και `Rect`.

Στην Εικόνα 51 έχουμε ένα παράδειγμα χρήσης μεταβλητών τύπου vector στη μέθοδο detectFacesYunet(), μέσω της οποίας πραγματοποιούμε εντοπισμό προσώπων επάνω σε αρχεία βίντεο. Χρησιμοποιούμε μια μεταβλητή vector με αντικείμενα Mat (γραμμή 222) για την αποθήκευση των στιγμιότυπων που περιέχουν κάποιο πρόσωπο και μια για μεταβλητές τύπου double (γραμμή 223) για την αποθήκευση των χρονικών στιγμών που γίνεται ο εντοπισμός.

```
215 void FaceDetectionThread::detectFacesYunet()
216 {
217     emit currentStatus("Εντοπισμός προσώπων...");
218     int processedFrames = 0;
219     std::string videoPathString = videoPath.toString();
220     cv::VideoCapture videoCapture(videoPathString);
221     int totalFrames = videoCapture.get(cv::CAP_PROP_FRAME_COUNT);
222     std::vector<cv::Mat> framesWithFaces;
223     std::vector<double> timestampVector;
```

Εικόνα 52: Χρήση μεταβλητών τύπου std::vector.

4.1.3.4.2 iostream

Η βιβλιοθήκη iostream στη C++ είναι μια από τις θεμελιώδεις βιβλιοθήκες του C++ Standard Library και παρέχει τις βασικές λειτουργίες εισόδου και εξόδου για προγράμματα C++. Η βιβλιοθήκη αυτή περιλαμβάνει τις κλάσεις και τα αντικείμενα που επιτρέπουν τη διαχείριση ροών εισόδου και εξόδου, όπως την ανάγνωση από το πληκτρολόγιο και την εγγραφή στην οθόνη ή σε αρχεία. Βασικά αντικείμενα που χρησιμοποιούνται από την iostream είναι τα std::cerr για error handling και std::cout για debugging σε ορισμένες περιπτώσεις.

```
188 // Split the line by space.
189 if (!(iss >> path >> id))
190 {
191     std::cerr << "Invalid line format: " << line << std::endl;
192     continue;
193 }
```

Εικόνα 53: Χρήση του std::cerr σε περίπτωση που η γραμμή δεν έχει την σωστή μορφή.

4.1.3.4.3 fstream

Η βιβλιοθήκη fstream στη C++ είναι μέρος της Standard Library και παρέχει εργαλεία για τη διαχείριση αρχείων (file streams). Περιλαμβάνει κλάσεις και λειτουργίες που επιτρέπουν την ανάγνωση (std::ifstream ή αλλιώς input file stream) και την εγγραφή (std::ofstream ή αλλιώς output file stream) δεδομένων σε αρχεία, καθώς και τον συνδυασμό αυτών των λειτουργιών. Στην εφαρμογή χρησιμοποιείται ως εναλλακτική της QTextStream σε περιπτώσεις που απλά χρειαζόμαστε μια γρήγορη μέθοδο ανάγνωσης αρχείων.

```
156 std::vector<QStringList> CatalogService::getAllEntries() const
157 {
158     std::vector<QStringList> entries;
159
160     // Open the catalog.txt file.
161     std::ifstream file(catalogFolderPath.toString() + "/catalog.txt");
162     if (!file.is_open())
163     {
164         // Return an empty vector.
165         return entries;
166     }
```

Εικόνα 54: Χρήση της std::ifstream για ανάγνωση αρχείων.

4.1.3.4.4 string

Η βιβλιοθήκη string στη C++ είναι μέρος της Standard Library και παρέχει την κλάση std::string, η οποία επιτρέπει τη διαχείριση και τη λειτουργία των αλφαριθμητικών (strings). Η

κλάση αυτή αντικαθιστά τις κλασικές C-style strings (πίνακες χαρακτήρων) και παρέχει πλούσιες δυνατότητες και εύκολη χρήση. Η χρήση της `std::string` είναι αναγκαία εξαιτίας των μεθόδων των βιβλιοθηκών της OpenCV οι οποίες δεν μπορούν να χρησιμοποιήσουν την κλάση `QString` στα ορίσματα που χρειάζονται strings.

4.1.3.4.5 algorithm

Η `algorithm` δεν είναι πραγματική κλάση αλλά ένα σύνολο από μεθόδους που παρέχονται μέσω της κεφαλίδας `<algorithm>`. Η κεφαλίδα αυτή αποτελεί μέρος της Standard Template Library (STL) και περιλαμβάνει μια συλλογή από αλγόριθμους που μπορούν να χρησιμοποιηθούν για την εκτέλεση κοινών λειτουργιών επεξεργασίας δεδομένων. Οι αλγόριθμοι αυτοί είναι σχεδιασμένοι για να λειτουργούν με οποιοδήποτε είδος container που υποστηρίζει επαναλήψεις (iterators).

Η `algorithm` ήταν ιδιαίτερα χρήσιμη στις διαδικασίες διαχείρισης καταλόγου που περιλαμβάνονται στα αρχεία `catalogservice.cpp` και `catalogservice.h`. Η κλάση `CatalogService` επιτρέπει (με την δημιουργία ενός αντικειμένου) την διαχείριση της βάσης δεδομένων του καταλόγου ο οποίος περιέχει όλες τις φωτογραφίες σύγκρισης και τα προφίλ των υπόπτων.

Οι συναρτήσεις που χρησιμοποιούνται είναι:

- `find_if`: Βρίσκει το πρώτο στοιχείο σε μια περιοχή που ικανοποιεί ένα δεδομένο κριτήριο. Το κριτήριο αυτό παρέχεται ως μια λογική συνάρτηση ή ένα λειτουργικό αντικείμενο (functor) που επιστρέφει `true` για το στοιχείο που ψάχνουμε και `false` για τα υπόλοιπα. Για να λειτουργήσει σωστά πρέπει να εισάγουμε το πρώτο στοιχείο ενός vector με τη μέθοδο `begin` και το τελευταίο με την μέθοδο `end`. Το τελευταίο όρισμα είναι η λογική συνάρτηση η οποία ψάχνει το `id` σε μια λίστα από strings.

```
238 auto iterator = std::find_if(allEntries.begin(), allEntries.end(), [id](const QStringList& entry)
239 {
240     return !entry.isEmpty() && entry[0] == id;
241 });
```

Εικόνα 55: Η `find_if`.

- `swap`: Ανταλλάσσει τις τιμές δύο αντικειμένων

4.1.4 Open Computer Vision Library

4.1.4.1 Γενικές Πληροφορίες

Η OpenCV (Open Source Computer Vision Library) είναι μία βιβλιοθήκη ανοιχτού κώδικα που αναπτύχθηκε αρχικά από την Intel το 1999 και έκτοτε έχει συνεισφέρει καθοριστικά στην πρόοδο της υπολογιστικής όρασης. Η OpenCV προσφέρει εργαλεία για την ανάλυση και την επεξεργασία εικόνων και βίντεο, καθιστώντας την έναν από τους πιο δημοφιλείς πόρους για την ανάπτυξη εφαρμογών υπολογιστικής όρασης.

Αρχικά, η OpenCV αναπτύχθηκε με στόχο να παρέχει μια κοινή υποδομή για εφαρμογές όρασης και να επιταχύνει τη χρήση της μηχανικής μάθησης σε εμπορικά προϊόντα. Η βιβλιοθήκη σχεδιάστηκε να είναι εξαιρετικά αποδοτική και ευέλικτη, γραμμένη σε C και C++, ενώ ταυτόχρονα παρέχει τρόπους χρήσης της σε Python, Java και άλλες γλώσσες προγραμματισμού, καθιστώντας την προσιτή σε ένα ευρύ φάσμα προγραμματιστών και ερευνητών.

Η OpenCV περιλαμβάνει περισσότερες από 2500 βελτιστοποιημένες αλγοριθμικές υλοποιήσεις που καλύπτουν ένα ευρύ φάσμα εφαρμογών, από την αναγνώριση προσώπων και την παρακολούθηση αντικειμένων μέχρι την επεξεργασία εικόνων και τη δημιουργία τρισδιάστατων αναπαραστάσεων. Οι δυνατότητες της βιβλιοθήκης επεκτείνονται συνεχώς, με τη συμβολή μιας μεγάλης κοινότητας προγραμματιστών και ερευνητών που προσθέτουν νέες λειτουργίες και βελτιώσεις.

Εκτός από την ευρεία χρήση της στην έρευνα και την ανάπτυξη, η OpenCV έχει υιοθετηθεί ευρέως στη βιομηχανία για την ανάπτυξη εφαρμογών σε τομείς όπως η ρομποτική, η ιατρική διάγνωση, η ανάλυση αθλητικών επιδόσεων και των συστημάτων ασφαλείας. Η χρήση της OpenCV σε αυτές τις εφαρμογές τονίζει τη σημασία της ως εργαλείο για την ανάπτυξη καινοτόμων και αποτελεσματικών λύσεων που βασίζονται στην υπολογιστική όραση.

4.1.4.2 Πως Χρησιμοποιήθηκε

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο οι αλγόριθμοι της OpenCV προσφέρουν γρήγορες και αποδοτικές λύσεις στον τομέα της υπολογιστικής όρασης. Όλες οι βιβλιοθήκες που χρησιμοποιεί η εφαρμογή είναι compiled με τον MSVC για περιβάλλοντα Windows x64, επομένως ο κώδικας μπορούσε να ενσωματωθεί απευθείας.

Όλες οι λειτουργίες της εφαρμογής που απαιτούν αλγορίθμους υπολογιστικής όρασης γίνονται μέσω της OpenCV με απλά includes των κλάσεων που ήταν απαραίτητες. Ακόμη, γίνεται χρήση των βασικών δομών δεδομένων που προσφέρει, όπως είναι η Mat (μεταβλητή αποθήκευσης πληροφοριών εικόνας) αλλά και η Rect (μεταβλητή αποθήκευσης συντεταγμένων ορθογωνίου).

Μαζί με τις βασικές βιβλιοθήκες, η OpenCV παρέχει επίσης πανίσχυρες ερευνητικές βιβλιοθήκες τελευταίας τεχνολογίας όπως η FaceDetectorYN και η FaceRecognizerSF ως πρόσθετο υλικό.

4.1.4.3 Αρχεία Μοντέλων Μηχανικής Μάθησης

Τα αρχεία μοντέλων μηχανικής μάθησης χρησιμοποιούνται για την αποθήκευση και την ανταλλαγή εκπαιδευμένων μοντέλων και των παραμέτρων τους. Κάθε τύπος αρχείου έχει το δικό του σκοπό και προέρχεται από διάφορα πλαίσια μηχανικής μάθησης. Στην εφαρμογή γίνεται χρήση των μοντέλων Caffe, Darknet YOLO, YuNet και SFace τα οποία παρέχουν ξεχωριστές λειτουργίες. Μέσω της βιβλιοθήκης dnn που προσφέρει η OpenCV μπορούμε να φορτώσουμε της ρυθμίσεις και το μοντέλο σε ένα καινούργιο αντικείμενο νευρωνικού δικτύου. Παρακάτω γίνεται μια αναφορά στα είδη αρχείων που χρειάζεται η εφαρμογή.

- .onnx (Open Neural Network Exchange)

Το ONNX είναι ένα ανοιχτό πρότυπο για την αναπαράσταση μοντέλων μηχανικής μάθησης. Δημιουργήθηκε από συνεργασία εταιρειών όπως η Microsoft και το Facebook. Σκοπός του είναι να επιτρέπει τη μεταφορά μοντέλων μεταξύ διαφορετικών εργαλείων μηχανικής μάθησης, όπως το PyTorch, το TensorFlow, το Caffe2 κ.λπ. Τα αρχεία .onnx αποθηκεύουν τη δομή του μοντέλου και τις εκπαιδευμένες παραμέτρους του.

- .caffemodel

Το αρχείο .caffemodel χρησιμοποιείται από το Caffe για την αποθήκευση των εκπαιδευμένων βαρών ενός μοντέλου. Συνεργάζεται με τα αρχεία .prototxt για να φορτώσει τόσο τη δομή όσο και τα βάρη του μοντέλου κατά τη διάρκεια της εκτέλεσης.

- .prototxt

Το αρχείο .prototxt χρησιμοποιείται στο πλαίσιο του Caffè για την αναπαράσταση της αρχιτεκτονικής ενός νευρωνικού δικτύου. Περιγράφει τη δομή του δικτύου, τα επίπεδα, τις παραμέτρους κάθε επιπέδου και τον τρόπο με τον οποίο αυτά συνδέονται μεταξύ τους. Δεν περιέχει τα βάρη του εκπαιδευμένου μοντέλου παρά μόνο τη διαμόρφωση του δικτύου.

- .cfg (Configuration)

Τα αρχεία .cfg χρησιμοποιούνται συχνά στο πλαίσιο του Darknet, το οποίο είναι το βασικό πλαίσιο πίσω από το YOLO (You Only Look Once). Αυτά τα αρχεία περιέχουν την αρχιτεκτονική του νευρωνικού δικτύου και τις παραμέτρους που χρησιμοποιούνται κατά την εκπαίδευση και την εκτέλεση του μοντέλου.

- .weights

Το αρχείο .weights αποθηκεύει τα εκπαιδευμένα βάρη ενός μοντέλου, συνήθως σε δυαδική μορφή. Στο πλαίσιο του Darknet και του YOLO, τα αρχεία .weights περιέχουν τις παραμέτρους του εκπαιδευμένου μοντέλου που χρησιμοποιούνται για την πρόβλεψη.

- .names

Τα αρχεία .names χρησιμοποιούνται επίσης στο πλαίσιο του Darknet. Περιέχουν τις ετικέτες κατηγοριών που αντιστοιχούν στις προβλέψεις του μοντέλου. Για παράδειγμα, σε ένα μοντέλο YOLO που ανιχνεύει αντικείμενα σε εικόνες, το αρχείο .names θα περιέχει τις ονομασίες των αντικειμένων (π.χ. «dog», «cat», «car», «person»).

4.1.4.4 Βιβλιοθήκες που Χρησιμοποιήθηκαν

4.1.4.4.1 dnn

Η βιβλιοθήκη dnn της OpenCV προσφέρει υποστήριξη για την φόρτωση και την χρήση μοντέλων μηχανικής μάθησης. Η dnn παρέχει τις βασικές μεθόδους φόρτωσης των αρχείων που αναλύθηκαν στο προηγούμενο κεφάλαιο οι οποίες παίρνουν σαν ορίσματα τις τοποθεσίες τους.

Η πρώτη συνάρτηση που χρησιμοποιήθηκε στην εφαρμογή από την βιβλιοθήκη dnn είναι η `cv::dnn::readNetFromDarknet()` για την λειτουργία εντοπισμού κίνησης, η οποία έχει ως ορίσματα το αρχείο .cfg (configuration) και το αρχείο βαρών (.weights). Αφού δημιουργήσουμε το αντικείμενο νευρωνικού δικτύου μπορούμε να θέσουμε και κάποιες επιπλέον παραμέτρους προεπεξεργασίας (pre-processing) που αφορούν τα μεγέθη της εικόνας και των σημείων εντοπισμού. Στην Εικόνα 55 βλέπουμε την συνάρτηση `initializeDNN()` όπου φαίνεται η παραπάνω διαδικασία.

```
60 void MotionDetectionThread::initializeDNN(const std::string& weightsFilePath, const std::string& configFilePath)
61 {
62     DNN = cv::dnn::readNetFromDarknet(configFilePath, weightsFilePath);
63     detectionModel = cv::dnn::DetectionModel(DNN);
64     detectionModel.setInputParams(1 / 255.0, cv::Size(416, 416), cv::Scalar(), true);
65 }
```

Εικόνα 56: Δημιουργία ενός νευρωνικού δικτύου εντοπισμού κίνησης.

Στην περίπτωση του Darknet, η χρήση ενός αρχείου ονομάτων (.names) είναι απαραίτητη ώστε ο αλγόριθμος να εντοπίσει τις κατηγορίες αντικειμένων. Για να φιλτράρουμε τις κατηγορίες που επιθυμούμε, αρχικά διαβάζουμε το αρχείο ονομάτων και περνάμε τις επιλεγμένες κατηγορίες σε ένα vector. Αυτό το φίλτρο μπορεί στη συνέχεια να χρησιμοποιηθεί ως όρισμα στη συνάρτηση `detect()` της βιβλιοθήκης.

Όσον αφορά την χρήση της `detect()` για τον εντοπισμό κίνησης, πρέπει αφού εντοπίσουμε τις κατηγορίες που έχουμε ορίσει προηγουμένως να χρησιμοποιήσουμε τις συντεταγμένες των ορθογωνίων γύρω από τα `objects` που παράγει ο αλγόριθμος για να γίνει οπτικοποίηση των αποτελεσμάτων. Οι βιβλιοθήκες της OpenCV μας δίνουν την δυνατότητα να «ζωγραφίζουμε» απευθείας επάνω στα στιγμιότυπα του βίντεο, τονίζοντας έτσι τα σημεία ενδιαφέροντος. Στην Εικόνα 56 φαίνεται η συνάρτηση `detectObjects()` η οποία πραγματοποιεί τον εντοπισμό σε κάθε εικόνα και ζωγραφίζει ένα ορθογώνιο γύρω από τα σημεία όπου έχει βρεθεί ένα αντικείμενο. Επιπλέον, η συνάρτηση `cv::putText` μας επιτρέπει να γράψουμε σε ποια κατηγορία ανήκει το αντικείμενο που βρέθηκε.

```

67 void MotionDetectionThread::detectObjects(cv::Mat& frame, std::vector<int>& classIds,
68 std::vector<float>& scores, std::vector<cv::Rect>& boxes, const std::vector<std::string>& classes, double
69 {
70     detectionModel.detect(frame, classIds, scores, boxes, sensitivity, 0.4);
71
72     for (int i = 0; i < classIds.size(); i++)
73     {
74         cv::rectangle(frame, boxes[i], cv::Scalar(0, 255, 0), 2);
75         char text[100];
76         sprintf(text, "%s: %.2f", classes[classIds[i]].c_str(), scores[i]);
77         cv::putText(frame, text, cv::Point(boxes[i].x, boxes[i].y - 5), cv::FONT_HERSHEY_SIMPLEX, 1,
78             cv::Scalar(0, 255, 0), 2);
79         // If timestamp flag is enabled write the detected classes to a vector of strings.
80         if (timestampFlag == true)
81         {
82             detectedClasses.push_back(classes[classIds[i]].c_str());
83         }
84     }
85 }

```

Εικόνα 57: Η συνάρτηση `detectObjects()`.

Με τον ίδιο τρόπο, γίνεται χρήση της συνάρτησης `cv::dnn::readNetFromCaffe()` για τον εντοπισμό προσώπων σε βίντεο η οποία έχει ως ορίσματα τα αρχεία `.prototxt` και `.caffemodel`. Η βασική διαφορά σε σχέση με το προηγούμενο νευρωνικό δίκτυο είναι ο τρόπος με τον οποίο δίνουμε την είσοδο στο δίκτυο και λαμβάνουμε τα αποτελέσματα από αυτό.

```

135 std::vector<cv::Rect> FaceDetectionThread::detectFaceRectanglesCaffe(const cv::Mat& frame)
136 {
137     meanValues = { 104., 177.0, 123.0 };
138     // Transforming the input image into a data blob.
139     // The function cv::dnn::blobFromImage takes care of rescaling the image to
140     // the correct input size for the network. It also subtracts the mean value
141     // in each color channel.
142     cv::Mat inputBlob = cv::dnn::blobFromImage(frame, scaleFactor,
143         cv::Size(videoWidth, videoHeight), meanValues, false, false);
144
145     // Forwarding the data through the network. Result is saved in the
146     // variable detectionMatrix.
147     DNN.setInput(inputBlob, "data");
148     cv::Mat detection = DNN.forward("detection_out");
149     cv::Mat detectionMatrix(detection.size[2], detection.size[3], CV_32F, detection.ptr<float>());
150
151     // Iterating through the rows of the matrix.
152     // Each row contains one detection. While iterating, check if the confidence
153     // value exceeds the threshold. If so, construct a cv::Rect and save it in
154     // the result vector faces.

```

Εικόνα 58: Η συνάρτηση `detectFaceRectanglesCaffe()`.

Αρχικά, γίνεται ο μετασχηματισμός της εικόνας σε blob δεδομένων. Τα blob είναι περιοχές της εικόνας όπου ιδιότητες όπως η ένταση ή το χρώμα είναι περίπου σταθερές. Η συνάρτηση `cv::dnn::blobFromImage()` φροντίζει για την αλλαγή της κλίμακας της εικόνας στο σωστό μέγεθος εισόδου για το δίκτυο. Επιπλέον, αφαιρεί τη μέση τιμή για κάθε κανάλι χρώματος. Για να λάβουμε τα αποτελέσματα, προωθούμε τα δεδομένα μέσα στο νευρωνικό δίκτυο και αποθηκεύουμε την έξοδο στην μεταβλητή `detectionMatrix`. Τέλος, διαβάζουμε όλες τις γραμμές του πίνακα `detectionMatrix` όπου κάθε γραμμή αντιστοιχεί σε έναν εντοπισμό. Σε περίπτωση που ένας εντοπισμός περάσει το κατώφλι εμπιστοσύνης που έχει ορίσει ο χρήστης τότε και μόνο τότε αποθηκεύεται σε μια μεταβλητή τύπου `Rect` για περαιτέρω επεξεργασία.

4.1.4.4.2 core

Η βιβλιοθήκη `core` της OpenCV αποτελεί τη θεμελιώδη υποδομή της πλατφόρμας και περιλαμβάνει βασικές λειτουργίες και δομές δεδομένων που χρησιμοποιούνται στις υπόλοιπες βιβλιοθήκες. Παρέχει εργαλεία για βασικές μαθηματικές λειτουργίες όπως αθροίσεις, αφαιρέσεις, πολλαπλασιασμούς και διαιρέσεις. Προσφέρει πιο σύνθετες μαθηματικές λειτουργίες όπως συναρτήσεις πινάκων, ιδιοτιμές, μετασχηματισμούς Fourier κ.ά. Επιπλέον, η βιβλιοθήκη περιλαμβάνει βασικές δομές δεδομένων όπως το `cv::Mat` για την αναπαράσταση

εικόνων και πινάκων, καθώς και σημεία (cv::Point), γραμμές (cv::Line), και ορθογώνια (cv::Rect). Παρέχει λειτουργίες ανάγνωσης και εγγραφής εικόνων και βίντεο σε διάφορες μορφές μέσω των συναρτήσεων imread, imwrite και VideoWriter, και υποστηρίζει επίσης τη διαχείριση βίντεο με τον κωδικοποιητή cv::VideoWriter::fourcc. Ο κωδικοποιητής δίνει στον προγραμματιστή την δυνατότητα επιλογών όπως, ο αριθμός των καρτέ ανάλυσης δευτερόλεπτο αλλά και συγκεκριμένων κωδικοποιήσεων και μεθόδων συμπίεσης των βίντεο. Στην εφαρμογή έχουμε κωδικοποίηση H. 264 ή οποία χρησιμοποιεί προηγμένες τεχνικές συμπίεσης για να παρέχει βίντεο υψηλής ποιότητας σε χαμηλούς ρυθμούς bit.

Η βιβλιοθήκη core προσφέρει επίσης λειτουργίες για την προεπεξεργασία εικόνων, όπως αλλαγή μεγέθους, αποκοπή, φιλτράρισμα και μετατροπές χρώματος, καθώς και τη λήψη πλαισίων από κάμερες και αρχεία βίντεο. Συμπερασματικά, η βιβλιοθήκη core είναι η καρδιά της υποδομής του OpenCV, πάνω στην οποία χτίζονται πιο εξειδικευμένες βιβλιοθήκες και λειτουργίες, καθιστώντας την ουσιαστική για την ανάπτυξη εφαρμογών υπολογιστικής όρασης και επεξεργασίας εικόνων.

4.1.4.4.3 FaceDetectorYN

Η FaceDetectorYN είναι μια πρόσθετη κλάση – μοντέλο τελευταίας τεχνολογίας η οποία αναπτύχθηκε το 2022 από τους ερευνητές Wei Wu, Hanyang Peng και Shiqi Yu. Ο αλγόριθμος εντοπισμού προσώπου που παρέχει η συγκεκριμένη κλάση υπερέρχει σε ταχύτητα και ακρίβεια σε σύγκριση με μοντέλα παρόμοιας λειτουργίας. Αυτό επιτυγχάνεται εξαιτίας του μικρού συνολικού μεγέθους του δικτύου και της αποτελεσματικής μεθόδου εξαγωγής χαρακτηριστικών που κατέχει.^[1] Το συγκεκριμένο μοντέλο διαθέτει τις δικές του μεθόδους δημιουργίας ενός νευρωνικού δικτύου όπως φαίνεται στην Εικόνα 58.

```

112  ~int FaceDetectionThread::initializeDNNYunet()
113  {
114      cv::Size inputSize(videoWidth, videoHeight);
115      yunetDNN = cv::FaceDetectorYN::create(yunetModelPath,
116      "",
117      inputSize,
118      yConfidenceThreshold,
119      nmsThreshold,
120      detectionCount,
121      cv::dnn::Backend::DNN_BACKEND_OPENCV,
122      cv::dnn::DNN_TARGET_CPU);
123      return 0;
124  }

```

Εικόνα 59: Δημιουργία ενός δικτύου YuNet.

Το νευρωνικό δίκτυο λαμβάνει περισσότερα ορίσματα από αυτά που αναλύθηκαν στα προηγούμενα κεφάλαια. Ας δούμε μερικές πληροφορίες για το κάθε ένα από αυτά.

- yunetModelPath: Η τοποθεσία του αρχείου .onnx στο οποίο βρίσκεται η δομή του μοντέλου και οι εκπαιδευμένες παράμετροι του.
- Configuration: Στην συγκεκριμένη περίπτωση, το πεδίο "Configuration" δεν είναι απαραίτητο όταν χρησιμοποιούνται αρχεία .onnx. Γενικά, το πεδίο αυτό είναι απαραίτητο για τη διασφάλιση της συμβατότητας όταν χρησιμοποιούνται άλλοι τύποι αρχείων.
- inputSize: Η ανάλυση των εικόνων που παρέχει το αρχείο βίντεο π.χ. 1920x1080p.
- yConfidenceThreshold: Το ποσοστό εμπιστοσύνης που θέτει ο χρήστης. Όταν ένας επιτυχής εντοπισμός ξεπερνάει το κατώφλι τότε περνάει στον πίνακα με τα θετικά αποτελέσματα.
- nmsThreshold: Είναι ο συντελεστής σίγασης των θετικών αποτελεσμάτων του αλγορίθμου. Ενεργοποιείται όταν οι εντοπισμοί ξεπερνάνε το κατώφλι που θέτει ο χρήστης στη μεταβλητή detectionCount.

- **detectionCount**: Ο επιτρεπόμενος αριθμός των θετικών αποτελεσμάτων πριν ενεργοποιηθεί ο συντελεστής σίγασης.
- **Backend Target**: Το σύστημα πάνω στο οποίο θα δημιουργηθεί το νευρωνικό δίκτυο. Επιλέγουμε από default το OpenCV για λόγους ευκολίας.
- **Device Target**: Επιλέγουμε τον επεξεργαστή (CPU) λόγω συμβατότητας αφού μπορεί να μην κατέχουν όλοι οι υπολογιστές κάρτες γραφικών της Nvidia για να γίνεται χρήση του CUDA.

Αφού δημιουργηθεί το αντικείμενο δικτύου YUNet, η κλάση παρέχει την μέθοδο `detect()` η οποία έχει ορίσματα την αρχική εικόνα (`cv::Mat`) και μια κενή μεταβλητή `cv::Mat` στην οποία αποθηκεύονται οι τοποθεσίες των προσώπων που εντοπίστηκαν επιτυχώς.

```
185 cv::Mat FaceDetectionThread::detectFaceRectanglesYunet(const cv::Mat frame)
186 {
187     cv::Mat res;
188     yunetDNN->detect(frame, res);
189     return res;
190 }
```

Εικόνα 60: Χρήση της μεθόδου `detect()`.

Όμως, η `FaceDetectorYN` δεν συμπεριλαμβάνει μεθόδους οπτικοποίησης των δεδομένων, επομένως πρέπει να δημιουργήσουμε μια custom μέθοδο για τον σκοπό αυτό. Στην Εικόνα 60 βλέπουμε την μέθοδο `visualizeDetectionsYunet()` η οποία λαμβάνει την αρχική εικόνα και την μεταβλητή `cv::Mat` που περιέχει τις συντεταγμένες των προσώπων. Στην συνέχεια, διαβάζουμε τις συντεταγμένες από την μεταβλητή `frameWithFaces` και ζωγραφίζουμε με την μέθοδο `cv::rectangle()` τα ορθογώνια γύρω από τα πρόσωπα.

```
192 cv::Mat FaceDetectionThread::visualizeDetectionsYunet(const cv::Mat& frame,
193                                                       const cv::Mat& frameWithFaces,
194                                                       int frameNumber)
195 {
196     static cv::Scalar boxColor{ 0, 255, 0 };
197     static cv::Scalar textColor{ 0, 255, 0 };
198     auto outputImage = frame.clone();
199     int frameCounter = 0;
200
201     for (int i = 0; i < frameWithFaces.rows; ++i)
202     {
203         // Draw bounding boxes.
204         int x1 = static_cast<int>(frameWithFaces.at<float>(i, 0));
205         int y1 = static_cast<int>(frameWithFaces.at<float>(i, 1));
206         int w = static_cast<int>(frameWithFaces.at<float>(i, 2));
207         int h = static_cast<int>(frameWithFaces.at<float>(i, 3));
208         cv::rectangle(outputImage, cv::Rect(x1, y1, w, h), boxColor, 2);
209     }
```

Εικόνα 61: Η μέθοδος `visualizeDetectionsYunet()`.

4.1.4.4.4 FaceRecognizerSF

Η `FaceRecognizerSF` αποτελεί επίσης μια πρόσθετη κλάση – μοντέλο του OpenCV η οποία ειδικεύεται στην αναγνώριση προσώπων. Αναπτύχθηκε από την Yaoyao Zhong, στο πανεπιστήμιο του Πεκίνο το 2021^[2] και αποτελεί ένα πανίσχυρο εργαλείο για την ταυτοποίηση προσώπων. Η χρήση της κλάσης δεν διαφέρει πολύ από την `FaceDetectorYN` στη δημιουργία αντικειμένων δικτύου αλλά είναι απαραίτητη η δημιουργία ειδικών συναρτήσεων για την οπτικοποίηση των δεδομένων και την αποφυγή διπλότυπων ταυτοποιήσεων.

Στην Εικόνα 61 βλέπουμε την συνάρτηση αρχικοποίησης του νευρωνικού δικτύου. Στην `initializeSFaceDNN()` θέτουμε στην μεταβλητή `sfaceDNN` ένα καινούργιο δίκτυο με την μέθοδο `cv::FaceRecognizerSF::create()` την οποία μας παρέχει η κλάση.

Τα ορίσματα της `create()` είναι:

- **modelPathString**: Η τοποθεσία του αρχείου `.onnx` στο οποίο βρίσκεται η δομή του μοντέλου και οι εκπαιδευμένες παράμετροι του.

- Configuration: Στην συγκεκριμένη περίπτωση, το πεδίο "Configuration" δεν είναι απαραίτητο όταν χρησιμοποιούνται αρχεία .onnx. Γενικά, το πεδίο αυτό είναι απαραίτητο για τη διασφάλιση της συμβατότητας όταν χρησιμοποιούνται άλλοι τύποι αρχείων.
- Backend Target: Το σύστημα πάνω στο οποίο θα δημιουργηθεί το νευρωνικό δίκτυο. Επιλέγουμε από default το OpenCV για λόγους ευκολίας.
- Device Target: Επιλέγουμε τον επεξεργαστή (CPU) λόγω συμβατότητας αφού μπορεί να μην κατέχουν όλοι οι υπολογιστές κάρτες γραφικών της Nvidia για να γίνεται χρήση του CUDA.

```

56 int FaceRecognitionThread::initializeFaceDNN()
57 {
58     std::string modelPathString = modelPath.toString();
59     sfaceDNN = cv::FaceRecognizerSF::create(modelPathString,
60                                             "",
61                                             cv::dnn::Backend::DNN_BACKEND_OPENCV,
62                                             cv::dnn::DNN_TARGET_CPU);
63     return 0;
64 }

```

Εικόνα 62: Η συνάρτηση initializeFaceDNN().

Το επόμενο συστατικό που χρειάζεται η λειτουργία αναγνώρισης είναι μια μέθοδος εξαγωγής των χαρακτηριστικών ενός προσώπου. Μέσω της συνάρτησης `imageInference()` (Εικόνα 62) εισάγουμε την αρχική εικόνα ενός προσώπου και εξάγουμε τα χαρακτηριστικά του με την μέθοδο `feature()` αποθηκεύοντας τα σε μια μεταβλητή τύπου `cv::Mat`. Εννοείται βέβαια πως η διαδικασία εξαγωγής γίνεται 1 φορά για όλα τα γνωστά πρόσωπα της βάσης δεδομένων και στη συνέχεια μέσω ενός βρόγχου εξάγονται και τα χαρακτηριστικά των αγνώστων.

```

66 cv::Mat FaceRecognitionThread::imageInference(cv::Mat inputImage)
67 {
68     cv::Mat features;
69     sfaceDNN->feature(inputImage, features);
70     features = features.clone();
71     return features;
72 }

```

Εικόνα 63: Η συνάρτηση imageInference().

Στην συνέχεια ακολουθεί το στάδιο της ταυτοποίησης των δύο προσώπων. Αφού εξάγουμε τα χαρακτηριστικά από τις φωτογραφίες τα συγκρίνουμε μέσω της συνάρτησης `matchImages()`. Ανάλογα με την μέθοδο σύγκρισης που έχουμε επιλέξει και μαζί με τα χαρακτηριστικά, χρησιμοποιούμε την συνάρτηση `match()` για να δούμε εάν τα πρόσωπα ταιριάζουν. Στο παράδειγμα της Εικόνας 63 εάν το αποτέλεσμα της μεθόδου των συνημίτονων είναι μεγαλύτερο ή ίσο με την τιμή του κατωφλίου τότε έχουμε επιτυχή ταυτοποίηση μεταξύ των προσώπων.

```

74 int FaceRecognitionThread::matchImages(cv::Mat unknownFeatures, cv::Mat referenceFeatures)
75 {
76     if (distanceCalculationMode == "Συνημίτονου")
77     {
78         double cosineScore = sfaceDNN->match(unknownFeatures,
79                                             referenceFeatures,
80                                             cv::FaceRecognizerSF::DisType::FR_COSINE);
81         if (cosineScore >= thresholdCosine)
82         {
83             return 0;
84         }
85         else
86         {
87             return 1;
88         }
89     }

```

Εικόνα 64: Η συνάρτηση matchImages().

4.2 Πλατφόρμες Ανάπτυξης

4.2.1 Visual Studio 2022

4.2.1.1 Γενικές Πληροφορίες

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (IDE) που αναπτύχθηκε από τη Microsoft. Κυκλοφόρησε για πρώτη φορά το 1997 και έχει εξελιχθεί σε ένα από τα πιο δημοφιλή εργαλεία για την ανάπτυξη εφαρμογών. Το Visual Studio υποστηρίζει πολλαπλές γλώσσες προγραμματισμού, συμπεριλαμβανομένων των C++, C#, VB.NET, Python, JavaScript και άλλων. Παρέχει ισχυρά εργαλεία για τον έλεγχο του κώδικα, τη διόρθωση σφαλμάτων, τη διαχείριση έργων και τη συνεργασία μεταξύ ομάδων προγραμματιστών. Το IDE αυτό περιλαμβάνει επίσης ενσωματωμένα εργαλεία για τη δημιουργία, την ανάπτυξη και τη συντήρηση διαδικτυακών εφαρμογών, εφαρμογών κινητής τηλεφωνίας και παιχνιδιών.

Η πιο πρόσφατη έκδοση του Microsoft Visual Studio, το Visual Studio 2022, φέρνει μια σειρά από σημαντικές αναβαθμίσεις και νέες δυνατότητες που αποσκοπούν στη βελτίωση της παραγωγικότητας των προγραμματιστών και την υποστήριξη σύγχρονων τεχνολογιών ανάπτυξης λογισμικού. Πρωτίστως, το Visual Studio 2022 είναι η πρώτη έκδοση που είναι διαθέσιμη σε 64-bit, επιτρέποντας τη χρήση μεγαλύτερης μνήμης και τη διαχείριση μεγαλύτερων και πιο πολύπλοκων έργων με αυξημένη ταχύτητα και απόδοση. Το νέο περιβάλλον εργασίας είναι πιο ευέλικτο και διαμορφώσιμο, προσφέροντας βελτιωμένη εμπειρία χρήστη. Επιπλέον, οι νέες δυνατότητες debugging περιλαμβάνουν το Hot Reload, που επιτρέπει την αλλαγή του κώδικα και την άμεση εφαρμογή των αλλαγών χωρίς να χρειάζεται επανεκκίνηση της εφαρμογής.



Εικόνα 65: Το λογότυπο του Visual Studio 2022.

4.2.1.2 Γιατί το Visual Studio;

Το Microsoft Visual Studio αποτελεί ιδανικό εργαλείο για την ανάπτυξη έργων μεγάλης έκτασης σε C++ λόγω των εκτενών δυνατοτήτων και εργαλείων που προσφέρει. Πρώτον, το Visual Studio υποστηρίζει την ανάπτυξη 64-bit εφαρμογών, επιτρέποντας τη διαχείριση τους με αυξημένη απόδοση και χωρίς περιορισμούς στη μνήμη. Δεύτερον, η ισχυρή ενσωματωμένη υποστήριξη για C++ περιλαμβάνει εργαλεία αυτόματης ολοκλήρωσης κώδικα, όπως το IntelliSense, που βελτιώνει την παραγωγικότητα και μειώνει τα σφάλματα. Επιπλέον, οι προηγμένες δυνατότητες debugging, όπως το Live Unit Testing και το Hot Reload, επιτρέπουν τον άμεσο εντοπισμό και τη διόρθωση σφαλμάτων, μειώνοντας σημαντικά τον χρόνο ανάπτυξης.

Το Visual Studio προσφέρει επίσης πλήρη ενσωμάτωση με εργαλεία διαχείρισης έκδοσης κώδικα όπως το Git, διευκολύνοντας το versioning του κώδικα αλλά και παρέχοντας μια ολοκληρωμένη πλατφόρμα με δυνατότητες backup. Η ευρεία υποστήριξη για σύγχρονα πρότυπα της C++ (όπως το C++20) και η διαρκής ενημέρωση της πλατφόρμας εξασφαλίζουν ότι οι προγραμματιστές μπορούν να εκμεταλλευτούν τις τελευταίες βελτιώσεις της γλώσσας και των βιβλιοθηκών της. Εν κατακλείδι, το Visual Studio συνδυάζει ισχυρά εργαλεία ανάπτυξης, προηγμένες δυνατότητες debugging και άριστη ενσωμάτωση με το Git, καθιστώντας το ιδανικό για την ανάπτυξη μεγάλων και πολύπλοκων έργων σε C++.

4.2.1.3 Οδηγίες Εγκατάστασης Βιβλιοθηκών

Η εγκατάσταση βιβλιοθηκών σε ένα solution στο Visual Studio 2022 είναι μια διαδικασία που αφορά την προσθήκη των απαραίτητων αρχείων βιβλιοθήκης στην εφαρμογή και την ρύθμιση των παραμέτρων χρήσης τους. Παρακάτω ακολουθεί η διαδικασία εγκατάστασης του OpenCV στο Visual Studio. (Οι οδηγίες μπορούν να εφαρμοστούν σε γενικό βαθμό.)

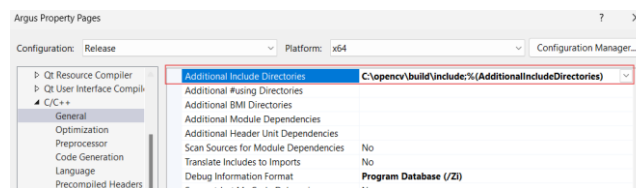
1. «Χτίσιμο» της βιβλιοθήκης.

Για να ενσωματώσουμε τις βιβλιοθήκες στο Visual Studio και να τις χρησιμοποιούμε χωρίς προβλήματα, πρέπει πρώτα να διασφαλίσουμε ότι το build γίνεται στη σωστή έκδοση. Το Visual Studio χρησιμοποιεί την τελευταία έκδοση του MSVC, επομένως είναι σημαντικό να επιλέξουμε τον κατάλληλο compiler. Χρησιμοποιώντας το εργαλείο CMake, μπορούμε να κάνουμε compile τις βιβλιοθήκες του OpenCV με την έκδοση του MSVC που έχουμε εγκατεστημένη στον υπολογιστή μας. Αυτή η διαδικασία παράγει δύο φακέλους: τον φάκελο include και τον φάκελο lib.

Ο φάκελος include περιέχει πληροφορίες σχετικά με τους ορισμούς των συναρτήσεων, τα ονόματά τους και τα ορίσματα που χρησιμοποιούν. Αντίθετα, ο φάκελος lib περιέχει τα αρχεία με τον προ-μεταγλωττισμένο κώδικα των συναρτήσεων, έτσι ώστε να μην χρειάζεται να γίνεται compile κάθε φορά, μειώνοντας τον χρόνο που απαιτείται για τη μεταγλώττιση της εφαρμογής μας. Αυτή η προσέγγιση εξασφαλίζει τη σωστή και αποδοτική χρήση των βιβλιοθηκών στο Visual Studio.

2. C/C++ Include Directories

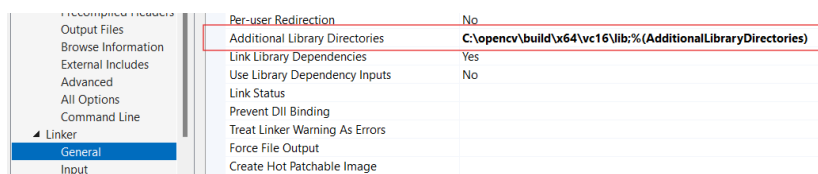
Στη συνέχεια πρέπει να συμπεριλάβουμε την τοποθεσία των ορισμών των συναρτήσεων που θέλουμε να χρησιμοποιήσουμε στην εφαρμογή μας. Πηγαίνοντας στο παράθυρο Property Pages > C/C++ > General εισάγουμε την τοποθεσία των φακέλων include στο πεδίο Additional Include Directories και πατάμε Apply.



Εικόνα 66: C/C++ Include Directories.

3. Linker General Options

Τέλος εισάγουμε την τοποθεσία του φακέλου που περιέχει τον pre-compiled κώδικα με τα libraries. Πηγαίνουμε στο παράθυρο Property Pages > Linker > General και συμπληρώνουμε στο πεδίο Additional Library Directories το path του φακέλου και πατάμε Apply.



Εικόνα 67: Linker General Options.

4.2.2 GitHub

4.2.2.1 Γενικές Πληροφορίες

Το GitHub, ιδρυμένο το 2008 από τους Tom Preston-Werner, Chris Wanstrath, PJ Hyett, και Scott Chacon, είναι μια πλατφόρμα φιλοξενίας κώδικα και συνεργασίας προγραμματιστών που βασίζεται στο σύστημα ελέγχου έκδοσης Git. Λειτουργώντας ως κοινωνικό δίκτυο για προγραμματιστές, το GitHub επιτρέπει την αποθήκευση, τη διαχείριση και την κοινή χρήση κώδικα, υποστηρίζοντας παράλληλα την παρακολούθηση αλλαγών στον κώδικα μέσω ενός ιστορικού εκδόσεων.

Η πλατφόρμα προσφέρει ένα ευρύ φάσμα εργαλείων για την υποστήριξη της ανάπτυξης λογισμικού, όπως συστήματα διαχείρισης έργων, λειτουργίες ανάλυσης κώδικα και ενσωμάτωση με άλλα εργαλεία συνεχούς ολοκλήρωσης και παράδοσης (CI/CD). Το 2018, η Microsoft εξαγόρασε το GitHub, ενισχύοντας περαιτέρω την υποδομή και τις δυνατότητές του. Με εκατομμύρια χρήστες παγκοσμίως, το GitHub έχει καθιερωθεί ως πρότυπο στην κοινότητα ανάπτυξης λογισμικού, προωθώντας την ανοιχτή συνεργασία και την καινοτομία μέσω της φιλοξενίας και της συντήρησης ενός τεράστιου αριθμού ανοικτού κώδικα και ιδιωτικών έργων.

4.2.2.2 Χρήση του GitHub

Στην περίπτωση της εφαρμογής Argus, το GitHub χρησιμοποιήθηκε ως εργαλείο ελέγχου έκδοσης (versioning) για τη διαχείριση και την αποθήκευση των λειτουργικών εκδόσεων της εφαρμογής. Μετά την ολοκλήρωση μιας νέας λειτουργίας ή ενός σημαντικού τμήματος κώδικα, γινόταν συγχώνευση (merge) με την τρέχουσα έκδοση στο GitHub για να διασφαλιστεί ότι η συγκεκριμένη έκδοση ήταν σταθερή και δεν περιείχε σημαντικά σφάλματα. Αυτή η διαδικασία διευκόλυνε επίσης την επιστροφή (rollback) σε μια προηγούμενη λειτουργική έκδοση σε περίπτωση που παρουσιαζόταν πρόβλημα με τον νέο κώδικα.

Επιπλέον, το αποθετήριο GitHub δημιουργήθηκε για την παρουσίαση του πηγαίου κώδικα, υποστηρίζοντας τους σκοπούς της διπλωματικής εργασίας. Αυτό, επιτρέπει την εύκολη πρόσβαση και αξιολόγηση του κώδικα, καθώς και την τεκμηρίωση και διατήρηση του έργου.

5. ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

5.1 Τρέχουσα Κατάσταση

Όπως αναφέρθηκε στο εισαγωγικό κεφάλαιο της εργασίας αυτής, το κύριο πρόβλημα αποτελεί η χειροκίνητη προβολή των στοιχείων βίντεο από οποιαδήποτε πηγή. Η μεγάλη διάρκεια των συγκεκριμένων βίντεο δεσμεύει ουσιαστικά το διαθέσιμο εργατικό δυναμικό. Σε περίπτωση που κάποιος δεν γνωρίζει την χρονική στιγμή ενός συμβάντος πρέπει να δει ολόκληρο το βίντεο είτε το συμβάν βρίσκεται στα πρώτα πέντε λεπτά είτε στο τέλος. Εκτός από την ανίχνευση αντικειμένων πολύ σημαντικό κομμάτι μιας έρευνας αποτελεί και η αναγνώριση προσώπων. Σε περίπτωση όπου και αυτή η διαδικασία γίνεται χειροκίνητα, ένας υπάλληλος θα πρέπει να εντοπίσει μόνος του τα σημεία όπου εμφανίζεται κάποιο πρόσωπο στο βίντεο, να εξάγει την εικόνα αυτή και να την συγκρίνει με τους πιθανούς υπόπτους. Αυτοματοποιώντας τις διαδικασίες αυτές μέσω της εφαρμογής μπορούμε να μειώσουμε τις ώρες όπου ένας υπάλληλος είναι δεσμευμένος.

Για να επιτευχθεί ο συγκεκριμένος στόχος θα πρέπει να χωρίσουμε το πρόβλημα μας σε τρία στάδια. Το πρώτο στάδιο αφορά την εξαγωγή των εντοπισμένων προσώπων από το βίντεο ώστε να υποστούν περαιτέρω επεξεργασία (cropping και αποθήκευση στον υπολογιστή). Το δεύτερο στάδιο απαιτεί την δημιουργία μιας βάσης δεδομένων γνωστών προσώπων ώστε να υπάρχει το απαραίτητο υλικό για να μπορέσουν να πραγματοποιηθούν οι συγκρίσεις. Το τρίτο και τελευταίο στάδιο αφορά την σύγκριση μεταξύ των εικόνων που έχουν εξαχθεί από το βίντεο και των εικόνων όπου βρίσκονται στην βάση δεδομένων.

5.2 Περιπτώσεις Χρήσης

5.2.1 Μελέτη Περίπτωσης Χρήσης: «Ανίχνευση Κίνησης»

5.2.1.1 Περιγραφή

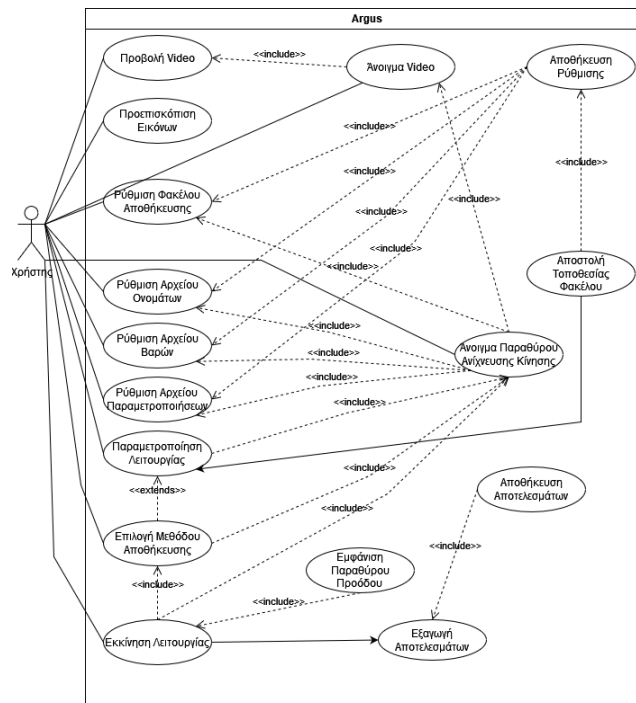
Με την εκκίνηση της εφαρμογής, ο χρήστης βρίσκεται στο κεντρικό παράθυρο το οποίο περιλαμβάνει τον video player και το εργαλείο προεπισκόπησης εικόνας. Ανοίγοντας για πρώτη φορά την εφαρμογή απαιτείται από τον χρήστη να πραγματοποιήσει ορισμένες βασικές ρυθμίσεις για την ορθή λειτουργία της. Σε πρώτη φάση πρέπει να ορίσει την τοποθεσία αποθήκευσης των αποτελεσμάτων που εξάγει η κάθε λειτουργία συμπεριλαμβανομένης και της ανίχνευσης κίνησης. Πατώντας στην επιλογή «Ρυθμίσεις» στο κεντρικό μενού της εφαρμογής εμφανίζεται το δεύτερο υπό-μενού το οποίο περιέχει την επιλογή «Ρυθμίσεις Αποθήκευσης». Ο χρήστης ανοίγει το συγκεκριμένο παράθυρο ώστε να ορίσει την τοποθεσία ενός κενού φακέλου πατώντας στο κουμπί «Επιλογή Φακέλου» στο κουτί που αναγράφεται η ετικέτα «Ρυθμίσεις Αποθήκευσης Αποτελεσμάτων». Μόλις ολοκληρωθεί η διαδικασία, εμφανίζεται στο πεδίο δίπλα από το κουμπί η επιλεγμένη τοποθεσία και ο χρήστης πατάει την επιλογή «Αποθήκευση» ώστε να γραφτεί η ρύθμιση στο σχετικό αρχείο. Στη συνέχεια απαιτείται η ρύθμιση του νευρωνικού δικτύου που είναι υπεύθυνο για την ανίχνευση της κίνησης. Σε περίπτωση που ο χρήστης δεν έχει ολοκληρώσει τις σχετικές ρυθμίσεις, ειδοποιείται με ένα ειδικό παράθυρο σφάλματος το οποίο αποτρέπει τον χρήστη από το να προχωρήσει (το ίδιο ισχύει και για την τοποθεσία αποθήκευσης). Για να ρυθμιστεί το νευρωνικό δίκτυο ο χρήστης μεταβαίνει στις τρεις επιλογές του τρίτου υπό-μενού κάτω από την επιλογή «Ρυθμίσεις Νευρωνικού Δικτύου». Στο παράθυρο «Αρχείο Ονομάτων» ο χρήστης επιλέγει πατώντας το κουμπί με το εικονίδιο φακέλου ένα αρχείο με extension .names και να πατήσει «Αποθήκευση». Στη συνέχεια στο παράθυρο «Αρχείο Παραμετροποιήσεων» με τον ίδιο τρόπο επιλέγει ένα αρχείο με extension .cfg. Ομοίως για το παράθυρο «Αρχείο Βαρών» ένα αρχείο με extension .weights.

Για να ξεκινήσει μια διεργασία ανίχνευσης κίνησης πρέπει να φορτώσει αρχικά ένα αρχείο βίντεο στην εφαρμογή. Πηγαίνοντας τον κέρσορα στο μενού της εφαρμογής και πατώντας το υπό-μενού «Αρχείο» ανοίγει το dropdown μενού με τις επιλογές «Άνοιγμα» και «Εξοδος». Πατώντας επάνω στην επιλογή «Άνοιγμα», ο χρήστης ανοίγει το παράθυρο επιλογής αρχείων το οποίο είναι ρυθμισμένο με τρόπο που εμφανίζει μόνο τα αρχεία βίντεο στον υπολογιστή του. Έτσι ο χρήστης επιλέγει το επιθυμητό αρχείο βίντεο και πατάει το κουμπί «Άνοιγμα» στο κάτω μέρος του παραθύρου. Πλέον το βίντεο είναι φορτωμένο στην εφαρμογή οπότε ο χρήστης προχωράει στο επόμενο βήμα.

Μόλις πατήσει την επιλογή «Ανίχνευση Κίνησης» στην γραμμή εργαλείων ανοίγει το παράθυρο «Παραμετροποίηση Ανίχνευσης Κίνησης» το οποίο διαθέτει διάφορες ρυθμίσεις για την λειτουργία αυτή. Σε πρώτη φάση μπορεί να επιλέξει τον αριθμό παράλειψης καρέ ο οποίος μειώνει τον χρόνο εκτέλεσης αφού δεν επεξεργάζεται ολόκληρο το αρχείο. Στην συνέχεια ρυθμίζει την ευαισθησία ανίχνευσης, το ποσοστό δηλαδή εγκυρότητας των αποτελεσμάτων. Τέλος, ο χρήστης επιλέγει να γίνει εξαγωγή των χρονικών στιγμών κίνησης σε αρχείο κειμένου καθώς και την αποθήκευση των στιγμών που εντοπίζεται η κίνηση σε ξεχωριστό βίντεο. Αξίζει να σημειωθεί ότι το κουμπί εκκίνησης της λειτουργίας ενεργοποιείται μόνο αν ο χρήστης έχει επιλέξει τουλάχιστον μια επιλογή αποθήκευσης. Πατώντας το κουμπί «Εκκίνηση» εμφανίζεται ένα παράθυρο με μια μπάρα προόδου το οποίο ενημερώνει τον χρήστη για την κατάσταση της διαδικασίας. Μετά την ολοκλήρωση της διαδικασίας ο χρήστης μπορεί να δει τα αποτελέσματα στον φάκελο που είχε ορίσει στην αρχή.

5.2.1.2 Διάγραμμα Περίπτωσης Χρήσης

Τα παρακάτω διαγράμματα δημιουργήθηκαν μέσω της πλατφόρμας draw.io η οποία είναι διαθέσιμη και δωρεάν προς χρήση στον σύνδεσμο <https://app.diagrams.net/>.



Εικόνα 68: Διάγραμμα περίπτωσης χρήσης «Ανίχνευσης Κίνησης».

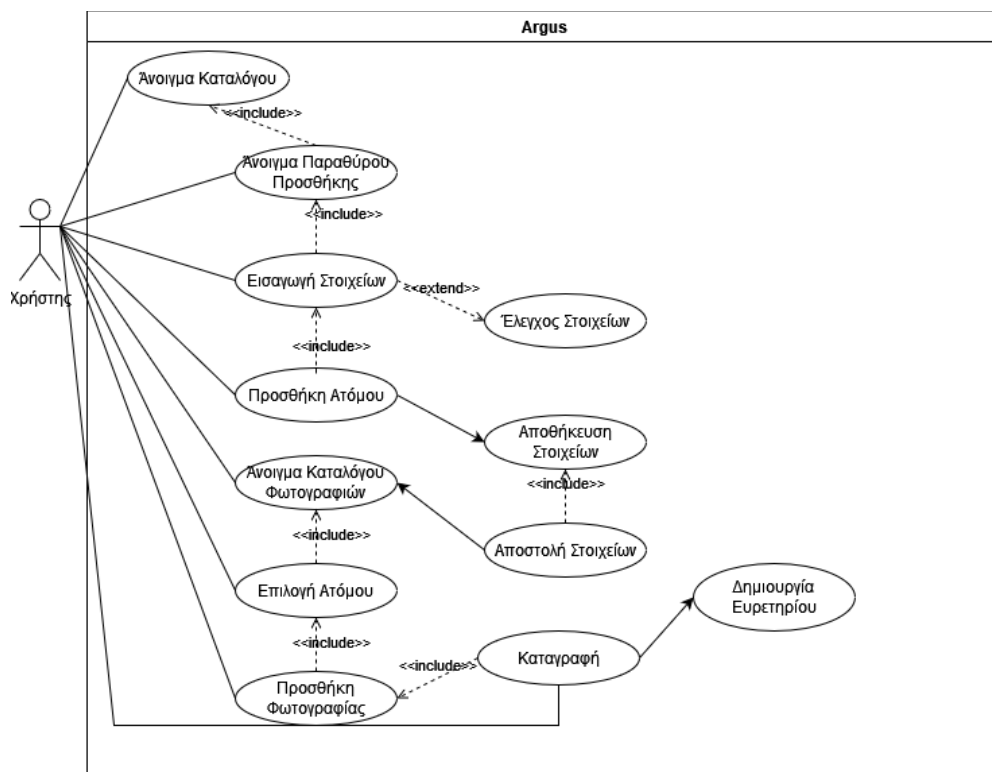
5.2.2 Μελέτη Περίπτωσης Χρήσης: «Εισαγωγή Δεδομένων Υπόπτου στη Βάση»

5.2.2.1 Περιγραφή

Με την είσοδο στην εφαρμογή, ο χρήστης βρίσκεται στο κεντρικό παράθυρο που περιλαμβάνει τον video player και το εργαλείο προεπισκόπησης εικόνας. Για την ταυτοποίηση ατόμων, ο χρήστης εισάγει ένα νέο άτομο στη βάση δεδομένων των υπόπτων. Επιλέγοντας «Εργαλεία» και στη συνέχεια «Κατάλογος Ατόμων», ανοίγει το παράθυρο καταλόγου, το οποίο αρχικά είναι κενό. Για να προσθέσει ένα νέο άτομο, κάνει κλικ στο κουμπί «Προσθήκη», που ανοίγει ένα νέο παράθυρο με μια φόρμα εισαγωγής πληροφοριών όπως όνομα, επώνυμο, μοναδικό αριθμό ID και φωτογραφία συγκεκριμένων διαστάσεων. Μετά την εισαγωγή των δεδομένων, αποθηκεύει την εγγραφή και ενημερώνεται με μήνυμα επιτυχίας. Στο παράθυρο «Κατάλογος Ατόμων», πατώντας το κουμπί ανανέωσης, ενημερώνεται η λίστα για να εμφανιστεί η νέα προσθήκη.

Στη συνέχεια, ο χρήστης προσθέτει τις φωτογραφίες που θα χρησιμοποιήσει ο αλγόριθμος για την ταυτοποίηση των άγνωστων ατόμων. Μεταβαίνει στο παράθυρο «Δεδομένα Φωτογραφιών» από το μενού «Εργαλεία». Εκεί, επιλέγει τον ύποπτο που έχει εισαχθεί από το προηγούμενο παράθυρο, ο οποίος εμφανίζεται στη λίστα. Επιλέγοντας το άτομο, ενεργοποιείται η καρτέλα «Προσθήκη», όπου εμφανίζονται το αναγνωριστικό του ατόμου και ένας πίνακας με τις διαθέσιμες φωτογραφίες στη βάση δεδομένων. Κάνοντας κλικ στο κουμπί «Προσθήκη», ανοίγει το παράθυρο επιλογής αρχείων για να επιλέξει την εικόνα που θέλει να προσθέσει. Τέλος, για να χρησιμοποιηθεί η φωτογραφία από τον αλγόριθμο ταυτοποίησης, ο χρήστης μεταβαίνει στην καρτέλα «Καταγραφή», ελέγχει την εγκυρότητα της τοποθεσίας καταλόγου και πατάει το κουμπί «Καταγραφή». Η καταγραφή δημιουργεί ένα αρχείο σε μορφή ευρετηρίου για τον αλγόριθμο ταυτοποίησης.

5.2.2.2 Διάγραμμα Περίπτωσης Χρήσης



Εικόνα 69: Διάγραμμα περίπτωσης χρήσης «Εισαγωγή Δεδομένων Υπόπτου στη Βάση».

5.3 Απαιτήσεις Συστήματος

5.3.1 Λειτουργικές Απαιτήσεις

1. Η εφαρμογή θα πρέπει να υποστηρίζει την αναπαραγωγή βίντεο.
2. Η εφαρμογή θα πρέπει να διαθέτει κατάλληλο έλεγχο για την αναπαραγωγή των αρχείων βίντεο.
3. Η εφαρμογή θα πρέπει να διαθέτει πλήρως λειτουργική διεπαφή χρήστη.
4. Η εφαρμογή θα πρέπει να διαθέτει λειτουργίες εμφάνισης εικόνων.
5. Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο μήνυμα για επιτυχείς ενέργειες.
6. Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο μήνυμα σε περίπτωση σφάλματος.
7. Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη με κατάλληλο μήνυμα και να ζητάει την συγκατάθεση του πριν προχωρήσει σε σημαντικές ενέργειες.
8. Η εφαρμογή θα πρέπει να διατηρεί ελέγχους για να αποτρέπει τον χρήστη να προβεί σε μη επιτρεπτές ενέργειες.
9. Η εφαρμογή θα πρέπει να ενεργοποιεί και να απενεργοποιεί λειτουργίες ανάλογα με την κατάσταση που βρίσκεται.
10. Η εφαρμογή θα πρέπει να διατηρεί φακέλους για την αποθήκευση των δεδομένων.
11. Η εφαρμογή θα πρέπει να δημιουργεί κατάλληλο αρχείο για την αποθήκευση ρυθμίσεων.
12. Η εφαρμογή θα πρέπει να έχει την δυνατότητα να διαβάσει το αρχείο ρυθμίσεων κατά την εκκίνηση της.
13. Η εφαρμογή θα πρέπει να αποθηκεύει οποιαδήποτε αλλαγή στις ρυθμίσεις της ανά πάσα στιγμή.
14. Η εφαρμογή θα πρέπει να εφαρμόζει τις ρυθμίσεις από το αρχείο ρυθμίσεων κατά την εκκίνηση.
15. Η εφαρμογή θα πρέπει να διατηρεί τοπική βάση δεδομένων στον υπολογιστή.
16. Η εφαρμογή θα πρέπει να επιτρέπει την επεξεργασία μόνο ενός βίντεο την φορά.
17. Η εφαρμογή θα πρέπει να διαθέτει κατάλληλα εικονίδια για τα κουμπιά εργαλείων.
18. Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη για την πρόοδο των λειτουργιών μέσω της διεπαφής.
19. Η εφαρμογή θα πρέπει να διαθέτει ένα κεντρικό μενού για τον διαχωρισμό των λειτουργιών της.
20. Το μενού θα πρέπει να διαχωρίζεται σε περαιτέρω υπό-μενού για τον καλύτερο διαχωρισμό των λειτουργιών.
21. Τα κουμπιά θα πρέπει να ενημερώνουν τον χρήστη για την λειτουργία τους μέσω ενός σύντομου κειμένου.
22. Το κείμενο πληροφοριών για τα κουμπιά πρέπει να εμφανίζεται μόλις ο κέρσορας βρίσκεται επάνω από το κουμπί.
23. Η εφαρμογή θα πρέπει να παρέχει λειτουργίες καταλόγου ατόμων.
24. Η εφαρμογή θα πρέπει να εμφανίζει όλα τα άτομα που έχουν προστεθεί στην βάση δεδομένων.
25. Η εφαρμογή θα πρέπει να δημιουργεί ένα αρχείο καταλόγου φωτογραφικών δεδομένων.
26. Η εφαρμογή θα πρέπει να δημιουργεί ένα αρχείο καταλόγου με τα μοναδικά αναγνωριστικά (ID) όλων των ατόμων που βρίσκονται στην βάση δεδομένων.
27. Όλα τα sliders, και spinboxes θα πρέπει να είναι οριοθετημένα και να μην επιτρέπουν όλες τις τιμές.
28. Η εφαρμογή θα πρέπει να εμφανίζει όλες τις πληροφορίες για τα άτομα που έχουν προστεθεί.
29. Η εφαρμογή θα πρέπει να εμφανίζει την εικόνα προφίλ των ατόμων.

30. Η εφαρμογή θα πρέπει να ελέγχει τα πεδία της φόρμας στο παράθυρο προσθήκης για μη επιτρεπόμενα αλφαριθμητικά σύμβολα.
31. Η εφαρμογή θα πρέπει να ενημερώνει τον χρήστη δυναμικά κάτω από τα πεδία της φόρμας για τυχόν σφάλματα.
32. Η εφαρμογή θα πρέπει να διαθέτει επιλογές παραμετροποίησης σε όλες τις λειτουργίες της.
33. Η εφαρμογή θα πρέπει να εξάγει τα αποτελέσματα σε μορφή που μπορούν να προβληθούν από τον χρήστη.
34. Η εφαρμογή θα πρέπει να ελέγχει το μέγεθος της φωτογραφίας προφίλ που εισάγει ο χρήστης.
35. Η εφαρμογή θα πρέπει να δημιουργεί έναν καινούργιο υποφάκελο ατόμου μέσα στον κεντρικό φάκελο του καταλόγου.
36. Ο υποφάκελος του ατόμου θα πρέπει να έχει ως όνομα το μοναδικό αναγνωριστικό του ατόμου.
37. Ο υποφάκελος του ατόμου θα πρέπει να περιέχει ένα αρχείο κειμένου με τις πληροφορίες για αυτό, την φωτογραφία προφίλ και έναν φάκελο με το όνομα dataset για τα δεδομένα φωτογραφιών.
38. Κάθε άτομο που βρίσκεται στην βάση δεδομένων θα πρέπει να έχει όνομα, επώνυμο, όνομα με λατινικούς χαρακτήρες, επώνυμο με λατινικούς χαρακτήρες, ημερομηνία γέννησης και ένα μοναδικό αριθμητικό χαρακτηριστικό (ID).
39. Εάν διαγραφεί ένα άτομο το αριθμητικό χαρακτηριστικό του γίνεται διαθέσιμο προς χρήση ξανά.
40. Ο χρήστης θα πρέπει να έχει την δυνατότητα επιλογής τοπικών αρχείων από τον υπολογιστή του.
41. Ο χρήστης θα πρέπει να έχει την δυνατότητα προβολής του ιστορικού ενεργειών του.
42. Ο χρήστης θα πρέπει να έχει την δυνατότητα εκκαθάρισης του ιστορικού ενεργειών του.
43. Ο χρήστης θα πρέπει να έχει την δυνατότητα διαγραφής αρχείων μέσα από την εφαρμογή
44. Ο χρήστης θα πρέπει να έχει την δυνατότητα να προβάλλει τις πληροφορίες σχετικά με τα μεταδεδομένα ενός βίντεο.
45. Ο χρήστης θα πρέπει να έχει την δυνατότητα ανοίγει ένα βίντεο μέσω της εφαρμογής για επεξεργασία.
46. Ο χρήστης θα πρέπει να έχει την δυνατότητα να αφαιρεί ένα βίντεο μέσω της εφαρμογής.
47. Ο χρήστης θα πρέπει να έχει την δυνατότητα προσθήκης ενός ατόμου στον κατάλογο.
48. Ο χρήστης θα πρέπει να έχει την δυνατότητα διαγραφής ενός ατόμου από τον κατάλογο.
49. Ο χρήστης θα πρέπει να έχει την δυνατότητα αναζήτησης ενός ατόμου στον κατάλογο.
50. Ο χρήστης θα πρέπει να έχει την δυνατότητα να ταξινομεί τον κατάλογο αλφαβητικά και με βάση το μοναδικό αναγνωριστικό κάθε ατόμου.
51. Ο χρήστης θα πρέπει να έχει την δυνατότητα να ανανεώνει τον πίνακα με τα άτομα που βρίσκονται στον κατάλογο για τυχόν αλλαγές ή προσθήκες.
52. Ο χρήστης θα πρέπει να έχει την δυνατότητα να αλλάζει τον φάκελο αποθήκευσης της βάσης δεδομένων ανά πάσα στιγμή.
53. Ο χρήστης θα πρέπει να έχει την δυνατότητα να αλλάζει τον φάκελο αποθήκευσης των αποτελεσμάτων ανά πάσα στιγμή και από το μενού των ρυθμίσεων και από τα παράθυρα της κάθε λειτουργίας.

5.3.2 Μη Λειτουργικές Απαιτήσεις

1. Η εφαρμογή θα πρέπει να έχει κατέχει την δυνατότητα να εκτελεί τις λειτουργίες σε λογικά χρονικά πλαίσια.
2. Η βάση δεδομένων της εφαρμογής πρέπει να έχει την δυνατότητα επέκτασης.

3. Η διεπαφή χρήστη πρέπει να είναι φιλική και εύκολη στη χρήση, με μέγιστο χρόνο εκμάθησης 1 ώρα για νέους χρήστες.
4. Πρέπει να παρέχεται εκτενής τεκμηρίωση και οδηγίες χρήσης.
5. Πρέπει να είναι συμβατή με λειτουργικά συστήματα Windows.
6. Ο κώδικας της εφαρμογής πρέπει να είναι καλά τεκμηριωμένος και να ακολουθεί τα πρότυπα της C++.
7. Πρέπει να υπάρχει δυνατότητα για εύκολες αναβαθμίσεις και ενημερώσεις της εφαρμογής.
8. Η εφαρμογή θα πρέπει να είναι στην Ελληνική γλώσσα.
9. Η εφαρμογή θα πρέπει να τρέχει τοπικά σε έναν υπολογιστή.
10. Η εφαρμογή θα πρέπει να χρησιμοποιεί νήματα (threads) για την ορθή λειτουργία της.
11. Τα αριθμητικά χαρακτηριστικά (ID) κυμαίνονται από το 1000 έως το 9999.
12. Κάθε αριθμητικό χαρακτηριστικό πρέπει να είναι μοναδικό.
13. Το επιτρεπόμενο μέγεθος φωτογραφίας προφίλ θα πρέπει να είναι 256 ύψος επί 256 πλάτος.
14. Το κεντρικό παράθυρο της εφαρμογής θα πρέπει να εμφανίζεται maximized στην επιφάνεια εργασίας του χρήστη.
15. Τα φωτογραφικά δεδομένα θα πρέπει να είναι μεγέθους από 64 ύψος επί 64 πλάτος έως και 512 ύψος επί 512 πλάτος.

6. ΜΕΘΟΔΟΛΟΓΙΑ

6.1 Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική του συστήματος περιλαμβάνει τρία κύρια συστατικά στοιχεία: τη Γραφική Διεπαφή Χρήστη (UI), το Κύριο Σύστημα και τη Βάση Δεδομένων. Ο πρωταρχικός στόχος είναι να διασφαλιστεί ότι ο χρήστης μπορεί να αλληλοεπιδράσει αποκλειστικά μέσω της γραφικής διεπαφής, χωρίς να απαιτείται η έξοδος από την εφαρμογή για αλλαγή ρυθμίσεων ή εκτέλεση λειτουργιών. Το back-end της εφαρμογής λειτουργεί ως ο συνδεδετικός κρίκος μεταξύ της βάσης δεδομένων, των δυνατοτήτων υπολογιστικής όρασης και της γραφικής διεπαφής. Ακολουθεί μια λεπτομερής περιγραφή των συστατικών του συστήματος.

- Γραφική Διεπαφή Χρήστη

Η Γραφική Διεπαφή Χρήστη (UI) είναι το τμήμα μέσω του οποίου ο χρήστης αλληλοεπιδρά με το κύριο σύστημα της εφαρμογής για την εκτέλεση οποιασδήποτε λειτουργίας. Μέσω του UI και των προσφερόμενων στοιχείων (κουμπιά, μενού, μπάρες εργαλείων), ο χρήστης μπορεί να στείλει τα κατάλληλα σήματα με τη βοήθεια του μηχανισμού slots and signals, ώστε να εκτελεστούν οι επιθυμητές λειτουργίες. Αντίστοιχα, μέσω των σημάτων που στέλνει το σύστημα στη γραφική διεπαφή, ο χρήστης μπορεί να προβάλλει τις πληροφορίες και τα αποτελέσματα που εξάγει το σύστημα.

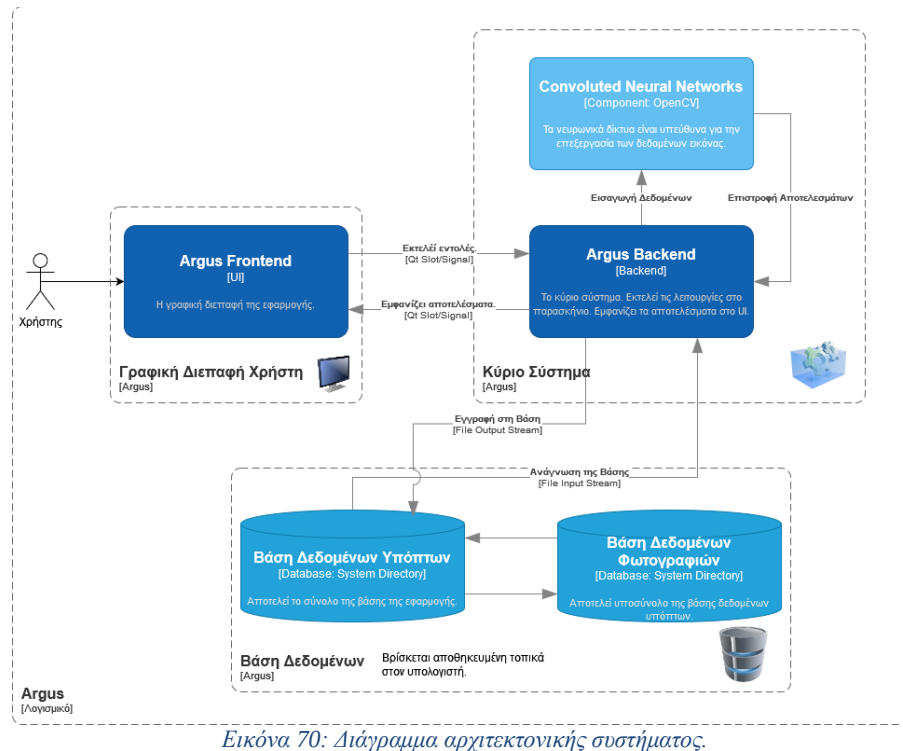
- Κύριο Σύστημα

Το κύριο σύστημα αποτελεί τον πυρήνα της εφαρμογής. Ο σκοπός του είναι να εκτελεί τις διεργασίες που του αναθέτει ο χρήστης μέσω της γραφικής διεπαφής. Το σύστημα έχει άμεση επικοινωνία με τη βάση δεδομένων, στην οποία μπορεί να γράψει, να διαβάσει, να επεξεργαστεί και να διαγράψει δεδομένα. Για παράδειγμα, όταν ο χρήστης ζητά να προβάλλει τον κατάλογο των υπόπτων, το σύστημα διαβάζει τα δεδομένα από τη βάση και τα προβάλλει στη γραφική διεπαφή μέσω του μηχανισμού slots and signals. Επιπλέον, το κύριο σύστημα είναι υπεύθυνο για την αποστολή δεδομένων εικόνας στα κατάλληλα νευρωνικά δίκτυα για επεξεργασία, την απλοποίηση των αποτελεσμάτων και την παρουσίασή τους σε φιλική μορφή προς τον χρήστη. Τέλος, το σύστημα αναλαμβάνει την εξαγωγή των αρχείων και την αποθήκευσή τους στον υπολογιστή.

- Βάση Δεδομένων

Η βάση δεδομένων καθορίζεται από τον χρήστη μέσω της εφαρμογής. Δημιουργείται ένας κεντρικός φάκελος στον υπολογιστή, εντός του οποίου η εφαρμογή αρχικοποιεί τη βάση με τη μορφή directories. Στο κεντρικό directory υπάρχει το ευρετήριο των ατόμων με τη μορφή ενός καταλόγου με IDs. Για κάθε νέο άτομο στη βάση δημιουργείται ένας νέος φάκελος με τα στοιχεία του, την εικόνα προφίλ και έναν φάκελο με τις φωτογραφίες που συγκρίνονται μέσω του αλγορίθμου αναγνώρισης προσώπων. Η βάση δεδομένων των φωτογραφιών αποτελεί ξεχωριστό υποσύνολο της βάσης, το οποίο συγκεντρώνεται σε ένα αρχείο-ευρετήριο για τη διευκόλυνση του συστήματος στην καταμέτρηση των υπάρχοντων φωτογραφιών.

Η αρχιτεκτονική αυτή διασφαλίζει την αδιάλειπτη και αποτελεσματική αλληλεπίδραση του χρήστη με το σύστημα, επιτυγχάνοντας παράλληλα την αξιόπιστη διαχείριση και επεξεργασία δεδομένων.



Εικόνα 70: Διάγραμμα αρχιτεκτονικής συστήματος.

6.2 Μεθοδολογία Ανάπτυξης

Η ανάπτυξη της εφαρμογής Argus έγινε με βάση την Agile μεθοδολογία ανάπτυξης. Η Agile μεθοδολογία ανάπτυξης λογισμικού είναι μια προσέγγιση που εστιάζει στην επαναληπτική ανάπτυξη και την προσαρμοστικότητα. Αντί να ακολουθεί ένα αυστηρό, γραμμικό μονοπάτι όπως το παραδοσιακό μοντέλο Waterfall, η Agile χωρίζει το έργο σε μικρότερα, διαχειρίσιμα κομμάτια που ονομάζονται sprints. Πιο συγκεκριμένα, είναι ένα σύνολο αρχών και πρακτικών για τη διαχείριση και την ανάπτυξη λογισμικού που τονίζει τη συνεργασία, την ευελιξία και τη συνεχή βελτίωση. Βασίζεται στις αξίες και τις αρχές που περιγράφονται στο [Agile Manifesto](#):

1. Άνθρωποι και αλληλεπιδράσεις πάνω από διαδικασίες και εργαλεία.
2. Λειτουργικό λογισμικό πάνω από εκτενή τεκμηρίωση.
3. Συνεργασία με τον πελάτη πάνω από τις συμβατικές διαπραγματεύσεις.
4. Ανταπόκριση στην αλλαγή πάνω από την τήρηση ενός προδιαγεγραμμένου σχεδίου.

Η Agile είναι συνήθως σχεδιασμένη για ομάδες προγραμματιστών, καθώς περιλαμβάνει πολλές πρακτικές που στηρίζονται στη συνεργασία και την αλληλεπίδραση μεταξύ των μελών της ομάδας. Ωστόσο, ορισμένες αρχές και πρακτικές της Agile μπορούν να εφαρμοστούν και από μεμονωμένους προγραμματιστές για να βελτιώσουν την αποτελεσματικότητα και την ποιότητα του λογισμικού τους. Όπως θα δούμε στη συνέχεια, τα στάδια ανάπτυξης της εφαρμογής βασίστηκαν επάνω στην μεθοδολογία Agile, προσαρμοσμένη κατάλληλα για ένα άτομο.

1. Μελέτη και καταγραφή των απαραίτητων λειτουργικών απαιτήσεων.

Το πρώτο στάδιο περιλαμβάνει την κατανόηση και τη σαφή καταγραφή του τι πρέπει να κάνει η εφαρμογή. Οι λειτουργικές απαιτήσεις ορίζουν τις βασικές λειτουργίες και τις δυνατότητες της εφαρμογής.

2. Σχεδίαση ιδεών για την γραφική διεπαφή.

Η γραφική διεπαφή χρήστη (UI) είναι κρίσιμη για την εμπειρία του χρήστη (UX). Σε αυτό το βήμα, γίνεται ο σχεδιασμός των παραθύρων που πλαισιώνουν μια λειτουργία. Δημιουργία αρχικών σκίτσων (wireframes) και πρωτοτύπων (mockups) για τις κύριες οθόνες της εφαρμογής.

3. Επαναληπτική ανάπτυξη πηγαίου κώδικα.

Η ανάπτυξη του πηγαίου κώδικα γίνεται σε επαναλήψεις, κάθε μία από τις οποίες προσθέτει συγκεκριμένες λειτουργίες στην εφαρμογή. Διαχωρισμός της ανάπτυξης σε μικρούς κύκλους εργασίας, συνήθως διάρκειας δύο έως τεσσάρων εβδομάδων (sprints). Εστίαση σε συγκεκριμένες λειτουργίες που έχουν προτεραιότητα, γράφοντας τον κώδικα και διασφαλίζοντας ότι πληροί τις απαιτήσεις.

4. Συνεχής ενσωμάτωση λειτουργικού κώδικα.

Η συνεχής ενσωμάτωση (Continuous Integration - CI) διασφαλίζει ότι όλες οι αλλαγές στον κώδικα συνδυάζονται ομαλά και λειτουργούν μαζί χωρίς προβλήματα. Γίνεται χρήση ενός συστήματος ελέγχου έκδοσης (όπως το Git) για τη διαχείριση του κώδικα και τη συνεχή ενσωμάτωση των αλλαγών.

5. Δοκιμές και debugging.

Το στάδιο των δοκιμών είναι κρίσιμο για την εξασφάλιση της ποιότητας και της αξιοπιστίας της εφαρμογής. Γίνεται έλεγχος των μεμονωμένων μονάδων κώδικα για να διασφαλιστεί ότι λειτουργούν σωστά (unit testing). Δοκιμή της αλληλεπίδρασης μεταξύ των διαφόρων μονάδων κώδικα (integration testing). Δοκιμή του ολοκληρωμένου συστήματος για να διασφαλιστεί ότι πληροί τις προδιαγραφές (system testing).

6. Ανασκόπηση και βελτίωση.

Η ανασκόπηση και η συνεχής βελτίωση είναι ζωτικής σημασίας για τη σωστή ανάπτυξη της εφαρμογής. Στο τελικό στάδιο, μετά από κάθε sprint, γίνεται μια ανασκόπηση ώστε να προσδιοριστεί τι πήγε καλά και τι πρέπει να βελτιωθεί.

6.3 Βήματα Ανάπτυξης

Η ανάπτυξη των λειτουργιών της εφαρμογής έγινε με την εξής σειρά:

Κεντρικό Παράθυρο: Αρχικά, πραγματοποιήθηκε ο σχεδιασμός του κεντρικού παραθύρου της εφαρμογής. Για τον σχεδιασμό της γραφικής διεπαφής, διεξήχθη μελέτη των προτύπων σχεδιασμού εφαρμογών της Microsoft. Στη συνέχεια, αναπτύχθηκαν οι λειτουργίες αναπαραγωγής βίντεο, προεπισκόπησης εικόνων και ανοίγματος αρχείων. Επιπλέον, προστέθηκε το ημερολόγιο εκτέλεσης κάτω από τον video player. Τέλος, μετά από μια αρχική μελέτη των λειτουργικών απαιτήσεων, προστέθηκε η γραμμή εργαλείων και το κεντρικό μενού.

Λειτουργία Ανίχνευσης Κίνησης: Σε αυτή τη φάση ανάπτυξης, ξεκίνησε η μελέτη των βιβλιοθηκών υπολογιστικής όρασης. Η σύνδεση με το υπόλοιπο πρόγραμμα πραγματοποιήθηκε σταδιακά, αρχικά υλοποιώντας απλές συναρτήσεις ανάγνωσης και επεξεργασίας εικόνων με τη βοήθεια του OpenCV και στη συνέχεια πιο πολύπλοκες συναρτήσεις που αφορούσαν τον εντοπισμό αντικειμένων. Ο εντοπισμός σχεδιάστηκε αρχικά

για να λειτουργεί σε εικόνες, πριν μετατραπεί ώστε να λειτουργεί και σε αρχεία βίντεο. Επιπλέον, σχεδιάστηκαν οι γραφικές διεπαφές για την εισαγωγή των αρχείων που απαιτεί το νευρωνικό δίκτυο και το παράθυρο ρυθμίσεων των φακέλων αποθήκευσης.

Λειτουργία Εντοπισμού Προσώπων: Η ανάπτυξη αυτής της λειτουργίας πραγματοποιήθηκε παράλληλα με την αναδιαμόρφωση όλων των παραθύρων της εφαρμογής. Αλλάζοντας τα εικονίδια σε όλα τα κουμπιά ώστε να διατηρείται ένα συνεκτικό στυλ, η γραφική διεπαφή αναβαθμίστηκε οπτικά. Όσον αφορά τη λειτουργία εντοπισμού προσώπου, η ανάπτυξή της ήταν ευκολότερη και ολοκληρώθηκε ταχύτερα σε σύγκριση με την ανίχνευση κίνησης. Πιο συγκεκριμένα, λόγω της προηγηθείσας μελέτης και της ενσωμάτωσης των βιβλιοθηκών στο Visual Studio, η ανάπτυξη ήταν πιο ομαλή. Επιπλέον, η σχεδίαση της γραφικής διεπαφής έγινε με τη χρήση προτύπων από προηγούμενα παράθυρα που είχαν αναπτυχθεί.

Βάση Δεδομένων Υπόπτων: Όπως αναφέρθηκε στο Κεφάλαιο 6.1, σε κάθε νέο sprint ανάπτυξης έπρεπε να πραγματοποιείται μελέτη των λειτουργικών απαιτήσεων της εφαρμογής. Στην συγκεκριμένη φάση, πριν αναπτυχθεί η λειτουργία αναγνώρισης προσώπων, έπρεπε να υλοποιηθεί η βάση δεδομένων μέσω της οποίας θα λειτουργούσε. Αρχικά, δημιουργήθηκαν τα σχεδιαγράμματα της βάσης δεδομένων, καθορίζοντας πώς θα λειτουργούσε τοπικά στον υπολογιστή. Έτσι, αναπτύχθηκε το κατάλληλο σύστημα αρχείων, το οποίο ανανεώνεται αυτόματα με κάθε νέα προσθήκη ή διαγραφή, χωρίς την ανάγκη παρέμβασης του χρήστη. Στη συνέχεια, σχεδιάστηκε το παράθυρο προβολής των δεδομένων της βάσης, το οποίο περιλαμβάνει δυνατότητες αναζήτησης, διαγραφής και προσθήκης δεδομένων, με την υποστήριξη κατάλληλου παραθύρου φόρμας.

Βάση Δεδομένων Φωτογραφιών: Αφού ολοκληρώθηκε η ανάπτυξη της βάσης δεδομένων υπόπτων, ενσωματώθηκε η βάση δεδομένων φωτογραφιών σε αυτήν. Μέσα σε κάθε φάκελο υπόπτου δημιουργήθηκε ένας ξεχωριστός φάκελος φωτογραφιών, στον οποίο έχουμε πρόσβαση μέσω του παραθύρου "Δεδομένα Φωτογραφιών". Τέλος, υλοποιήθηκε η δυνατότητα δημιουργίας καταλόγου φωτογραφιών, ώστε να διευκολυνθεί η χρήση τους από τον αλγόριθμο αναγνώρισης προσώπων.

Λειτουργία Αναγνώρισης Προσώπων: Μετά την υλοποίηση των δύο βάσεων δεδομένων, ξεκίνησε η ανάπτυξη της λειτουργίας αναγνώρισης προσώπων. Η ανάπτυξη αυτής της λειτουργίας ολοκληρώθηκε σε λιγότερο χρόνο σε σχέση με τις υπόλοιπες, καθώς σχεδιάστηκε με βάση τον κώδικα της λειτουργίας εντοπισμού, που είχε πολλές ομοιότητες.

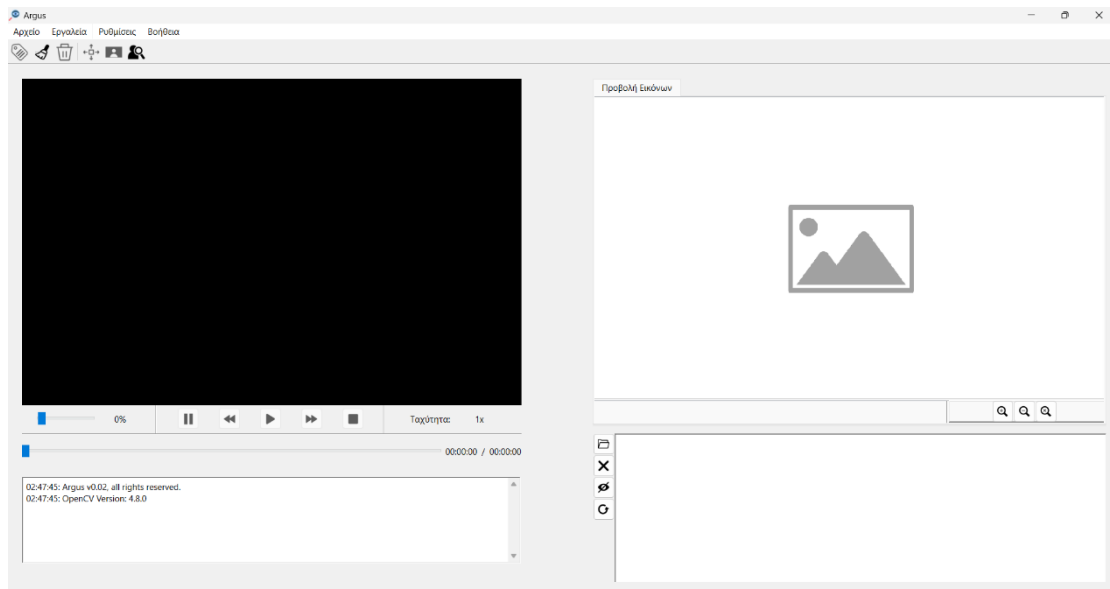
7. ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

7.1 Εισαγωγικές Πληροφορίες

Στα κεφάλαια που ακολουθούν παρουσιάζονται με λεπτομέρεια οι λειτουργίες και τα παράθυρα της εφαρμογής. Η ανάπτυξη τους έγινε με την σειρά που αναφέρθηκε στο Κεφάλαιο 6.2.

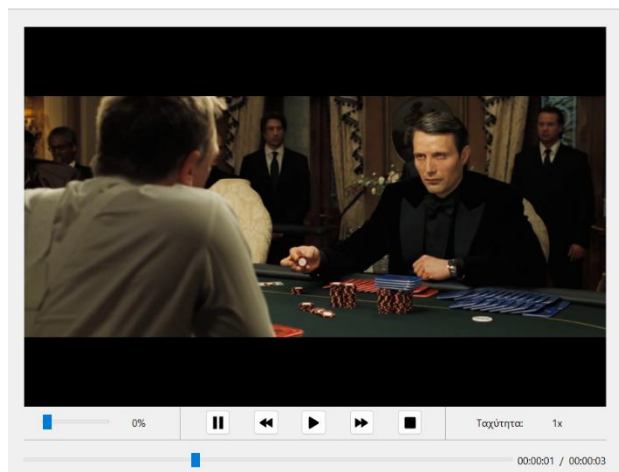
7.2 Κεντρικό Παράθυρο

Το κεντρικό παράθυρο αποτελεί την βάση πάνω στην οποία χτίστηκαν όλες οι λειτουργίες. Με το άνοιγμα του προγράμματος βρισκόμαστε στο συγκεκριμένο παράθυρο το οποίο παρέχει κάποιες βασικά εργαλεία προβολής πολυμέσων. Στα κεφάλαια που ακολουθούν θα δούμε αναλυτικά το κάθε κομμάτι του παραθύρου και πως αυτό αλληλοεπιδρά με τον χρήστη αλλά και με την ίδια την εφαρμογή.



Εικόνα 70: Άποψη του κεντρικού παραθύρου.

7.2.1 Video Player

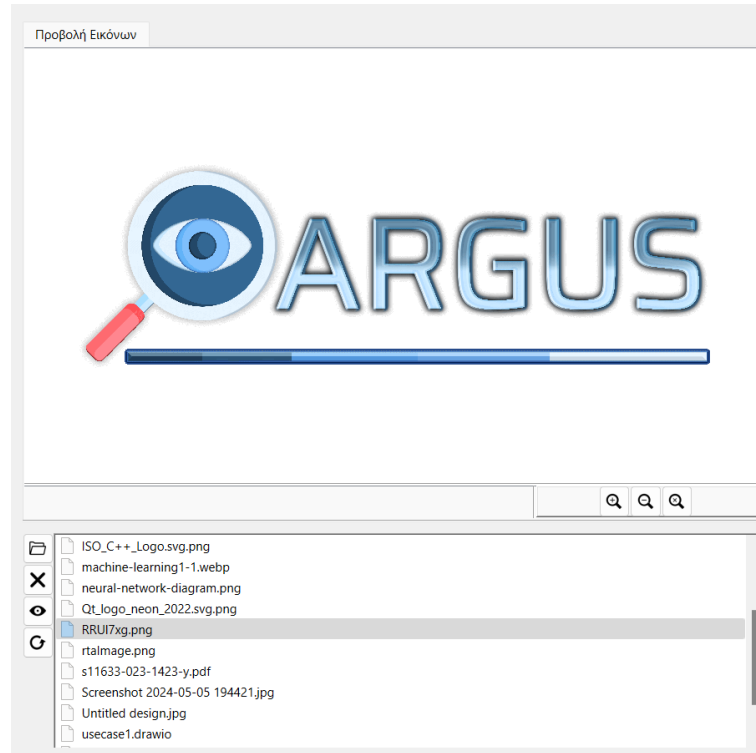


Εικόνα 71: Video player.

Αποτελεί ένα απλό εργαλείο αναπαραγωγής αρχείων βίντεο. Υποστηρίζει αρχεία τύπου .avi, .mpg και .mp4. Ας δούμε πιο αναλυτικά το κάθε γραφικό στοιχείο.

- **Έλεγχος Έντασης:** Η πρώτη μπάρα στην πάνω αριστερή μεριά αποτελεί το slider ελέγχου της έντασης. Ο χρήστης μπορεί να το σύρει με το ποντίκι για να ελέγξει το πόσο δυνατά ή σιγά θα ακούγεται ο ήχος. Δίπλα από την μπάρα αναγράφεται το ποσοστό της έντασης το οποίο αλλάζει δυναμικά όσο ο χρήστης αλλάζει την ένταση.
- **Κουμπί Παύσης:** Το πρώτο κουμπί απο την αριστερή μεριά χρησιμοποιείται για να σταματήσει ο χρήστης την αναπαραγωγή προσωρινά χωρίς όμως να επαναφέρει το βίντεο στην αρχή.
- **Κουμπί Μείωσης Ταχύτητας Αναπαραγωγής:** Το δεύτερο σε σειρά κουμπί μειώνει την ταχύτητα αναπαραγωγής του βίντεο με βήμα 0.25 ($1x > 0.75x > 0.5x > 0.25x$).
- **Κουμπί Αναπαραγωγής:** Ξεκινάει την αναπαραγωγή του βίντεο.
- **Κουμπί Αύξησης Ταχύτητας Αναπαραγωγής:** Αυξάνει την ταχύτητα αναπαραγωγής του βίντεο με βήμα 0.25 ($1x > 1.25x > 1.5x > 2x$).
- **Κουμπί Διακοπής:** Διακόπτει την αναπαραγωγή του βίντεο. Σε περίπτωση που ο χρήστης πατήσει ξανά το κουμπί αναπαραγωγής το βίντεο ξεκινάει από την αρχή.
- **Ένδειξη Ταχύτητας Αναπαραγωγής:** Στην πάνω δεξιά μεριά αναγράφεται η ταχύτητα αναπαραγωγής. Αλλάζει δυναμικά μόλις ο χρήστης αλλάξει την ταχύτητα.
- **Μπάρα Αναζήτησης (Seek Bar):** Επιτρέπει στον χρήστη να μεταβεί σε οποιοδήποτε σημείο του βίντεο επιθυμεί κάνοντας κλικ και σύροντας την δεξιά ή αριστερά.
- **Ένδειξη Τρέχοντος Χρονικού Σημείου:** Δίπλα στην μπάρα αναζήτησης αναγράφονται δύο χρόνοι. Στην αριστερή μεριά βλέπουμε την χρονική στιγμή που βρίσκεται το βίντεο ενώ από τη δεξιά μεριά την συνολική διάρκεια.

7.2.2 Εργαλείο Προεπισκόπησης Εικόνων



Εικόνα 72: Εργαλείο προεπισκόπησης εικόνων.

Αποτελεί ένα απλό εργαλείο προβολής εικόνων με ενσωματωμένες δυνατότητες περιήγησης και διαγραφής αρχείων. Οι λειτουργίες που περιλαμβάνει είναι:

- **Παράθυρο Προβολής:** Είναι υπεύθυνο για το rendering των εικόνων. Εμφανίζει τις φωτογραφίες και επιτρέπει την μετακίνηση τους κάνοντας κλικ και σύροντας την στο επιθυμητό σημείο.
- **Κουμπί Μεγέθυνσης:** Βρίσκεται στο κάτω δεξιό μέρος του παραθύρου προβολής. Το πρώτο από τα τρία κουμπιά μεγεθύνει την εικόνα που εμφανίζεται στο παράθυρο.
- **Κουμπί Σμίκρυνσης:** Το δεύτερο σε σειρά κουμπί μικραίνει την εικόνα στο παράθυρο προβολής.
- **Κουμπί Επαναφοράς:** Το τελευταίο κουμπί επαναφέρει την εικόνα στο αρχικό επίπεδο μεγέθυνσης.
- **Παράθυρο Περιήγησης Αρχείων:** Εμφανίζει την λίστα των αρχείων από τον φάκελο που έχει επιλέξει να ανοίξει ο χρήστης. Κάνοντας διπλό κλικ επάνω σε ένα αρχείο αυτό εμφανίζεται στο παράθυρο προβολής.
- **Κουμπί Επιλογής Φακέλου:** Βρίσκεται στην αριστερή μεριά του παραθύρου περιήγησης ξεκινώντας από επάνω. Κάνοντας κλικ ανοίγει το παράθυρο επιλογής φακέλου μέσα από το οποίο επιλέγει ο χρήστης ποιον θα εμφανίσει στο παράθυρο περιήγησης.
- **Κουμπί Διαγραφής Φωτογραφίας:** Διαγράφει το επιλεγμένο στοιχείο από τον φάκελο και κατ' επέκταση από τον υπολογιστή.
- **Κουμπί Ενεργοποίησης/Απενεργοποίησης Προβολής:** Λειτουργεί σαν διακόπτης για την εμφάνιση της φωτογραφίας στο παράθυρο προβολής. Σε περίπτωση που απενεργοποιηθεί η προβολή, εμφανίζεται στην θέση της φωτογραφίας μια default εικόνα.
- **Κουμπί Ανανέωσης:** Ανανεώνει την προβολή του επιλεγμένου φακέλου για να αντικατοπτρίζει τις αλλαγές σε περίπτωση διαγραφής ενός αρχείου η προσθήκης.

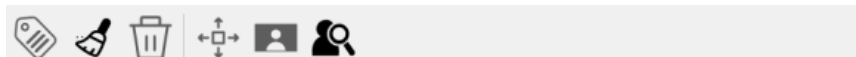
7.2.3 Ημερολόγιο Γεγονότων



Εικόνα 73: Ημερολόγιο γεγονότων.

Παρέχει έναν τρόπο παρακολούθησης των συμβάντων της εφαρμογής. Χρησιμοποιείται για την ενημέρωση του χρήστη σχετικά με τις εκδόσεις των βιβλιοθηκών που ξεκινάνε να εκτελούνται κατά την εκκίνηση της εφαρμογής. Επιπλέον, οι πληροφορίες μεταδεδωμένων του βίντεο εκτυπώνονται εδώ. Μαζί με το μήνυμα εμφανίζεται επίσης η χρονική στιγμή που στάλθηκε σύμφωνα με την ώρα του υπολογιστή.

7.2.4 Γραμμή Εργαλείων



Εικόνα 74: Γραμμή εργαλείων.

Βρίσκεται κάτω από το κεντρικό μενού στο επάνω μέρος της εφαρμογής. Χρησιμοποιείται για την διάταξη των λειτουργιών οι οποίες πρέπει να είναι άμεσα προσβάσιμες από τον χρήστη. Η χρήση γραμμής εργαλείων κάνει το περιβάλλον πιο φιλικό προς τον χρήστη και αυξάνει την ταχύτητα εργασίας. Στα κεφάλαια παρακάτω θα εξετάσουμε τις λειτουργίες με την σειρά που εμφανίζονται.

7.2.4.1 Προβολή Μεταδεδομένων



Εικόνα 75: Εμφάνιση μεταδεδομένων στο ημερολόγιο γεγονότων.

Πατώντας το κουμπί με το σύμβολο της ετικέτας ο χρήστης μπορεί να προβάλλει πληροφορίες για το βίντεο που έχει φορτώσει στην εφαρμογή. Οι πληροφορίες που εμφανίζονται είναι χρήσιμες για να μπορέσουν να παραμετροποιήσουν σωστά τις λειτουργίες ανίχνευσης κίνησης κ.λπ. Πιο συγκεκριμένα ο ρυθμός ανανέωσης καρτέ χρειάζεται για την ρύθμιση παράλειψης καρτέ που επιταχύνει την επεξεργασία των βίντεο.

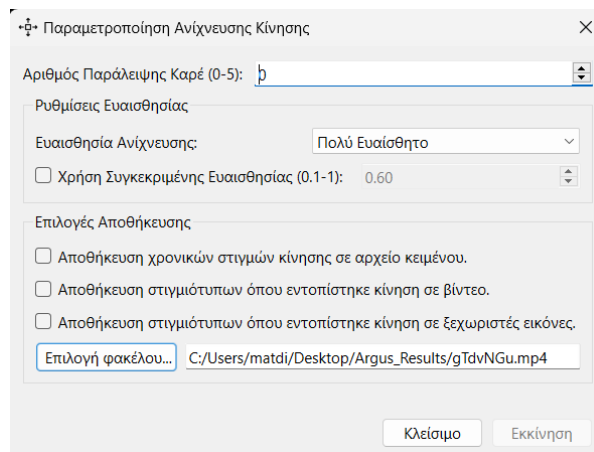
7.2.4.2 Αφαίρεση Φορτωμένου Αρχείου

Πατώντας το κουμπί με το σύμβολο ενός κάδου απορριμμάτων ο χρήστης μπορεί να αποφορτώσει ένα βίντεο από την εφαρμογή για να επιλέξει κάποιο άλλο. Αποφορτώνοντας ένα βίντεο οι σχετικές λειτουργίες που απαιτούν να υπάρχει επιλεγμένο ένα αρχείο απενεργοποιούνται αυτόματα.

7.2.4.3 Εκκαθάριση Ημερολογίου Γεγονότων

Πατώντας το κουμπί με το σύμβολο της σκούπας ο χρήστης μπορεί να καθαρίσει το ημερολόγιο γεγονότων διαγράφοντας το κείμενο που έχει μέσα.

7.2.4.4 Εντοπισμός Κίνησης



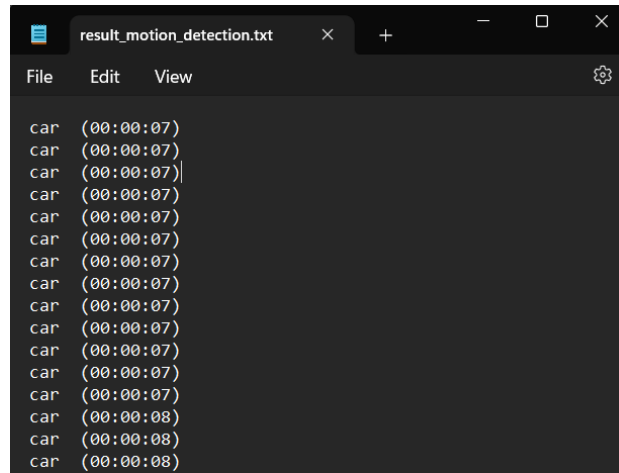
Εικόνα 76: Το παράθυρο ανίχνευσης κίνησης.

Το συγκεκριμένο παράθυρο εμφανίζεται μόλις ο χρήστης φορτώσει ένα βίντεο και πατήσει το κουμπί με το σύμβολο του τετραγώνου με βέλη. Ας δούμε αναλυτικά όλες τις επιλογές που έχει ο χρήστης πριν ξεκινήσει την διαδικασία.

- **Αριθμός Παράλειψης Καρτέ:** Δίνει στον αλγόριθμο την δυνατότητα παραλείψει καρτέ κατά την διάρκεια της επεξεργασίας. Για παράδειγμα εισάγοντας τον αριθμό 5 θα γίνεται ανάλυση της κίνησης ανά 5 εικόνες. Αυτό μειώνει τον συνολικό φόρτο

εργασίας αυξάνοντας την ταχύτητα επεξεργασίας αλλά ενδέχεται να παραλείψει κάποιο σημαντικό στοιχείο μειώνοντας την ακρίβεια.

- **Ευαισθησία Ανίχνευσης:** Το ποσοστό ευαισθησίας του αλγορίθμου. Ανάλογα με την τιμή που εισάγουμε αυξάνεται το κατώφλι των θετικών αποτελεσμάτων. Παράδειγμα, για την τιμή 0.8, όλα τα αποτελέσματα με ποσοστό αυτοπεποίθησης κάτω από αυτό δεν θα ληφθούν υπόψη στην εξαγωγή αποτελεσμάτων. Εκτός από την επιλογή συγκεκριμένης ευαισθησίας, υπάρχουν και κάποιες προκαθορισμένες τιμές που μπορεί να επιλέξει ο χρήστης.
- **Αποθήκευση σε Αρχείο Κειμένου:** Με την ενεργοποίηση της επιλογής αυτής δημιουργείται ένα αρχείο κειμένου το οποίο περιέχει τις χρονικές στιγμές και τα αντικείμενα που βρέθηκαν στο βίντεο.



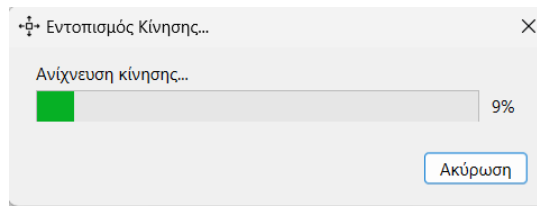
Εικόνα 77: Αρχείο κειμένου αποτελεσμάτων.

- **Αποθήκευση σε Βίντεο:** Με την ενεργοποίηση της επιλογής αυτής δημιουργείται ένα αρχείο βίντεο μόνο με τις χρονικές στιγμές τις οποίες ο αλγόριθμος εντόπισε κίνηση με ποσοστό αυτοπεποίθησης μεγαλύτερο από το κατώφλι.
- **Αποθήκευση σε Εικόνες:** Με την ενεργοποίηση της επιλογής αυτής τα στιγμιότυπα αποθηκεύονται σε μορφή εικόνας (αρχείο .jpg).
- **Επιλογή Φακέλου:** Παρόλο που η εφαρμογή ορίζει αυτόματα τον φάκελο αποτελεσμάτων για το φορτωμένο βίντεο, ο χρήστης μπορεί να επιλέξει οποιονδήποτε φάκελο πατώντας το σχετικό κουμπί.



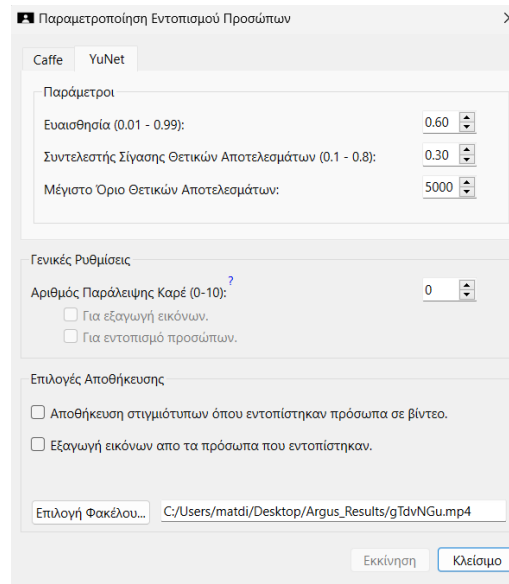
Εικόνα 78: Στιγμιότυπο φωτογραφικού αποτελέσματος.

Επιπλέον ο χρήστης ενημερώνεται για την πρόοδο της διαδικασίας μετά την εκκίνηση με το παράθυρο προόδου. Αυτό περιλαμβάνει το κουμπί «Ακύρωση» για να σταματήσει η διαδικασία καθώς και μια ετικέτα η οποία ενημερώνει τον χρήστη για την τρέχουσα διαδικασία που εκτελείται.



Εικόνα 79: Παράθυρο ενημέρωσης προόδου.

7.2.4.5 Εντοπισμός Προσώπων

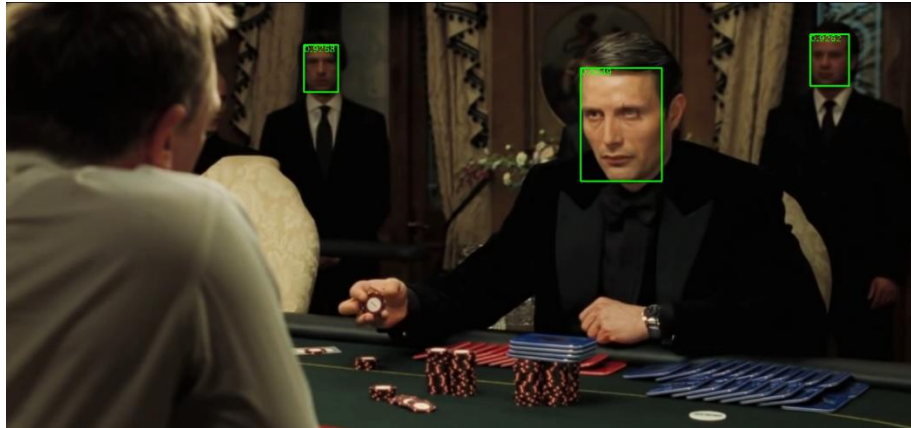


Εικόνα 80: Παράθυρο λειτουργίας εντοπισμού προσώπων.

Το συγκεκριμένο παράθυρο εμφανίζεται μόλις ο χρήστης φορτώσει ένα βίντεο και πατήσει το κουμπί με το σύμβολο ενός ατόμου μέσα σε ένα σκούρο πλαίσιο. Ας δούμε τις επιλογές που έχει ο χρήστης πριν ξεκινήσει την διαδικασία.

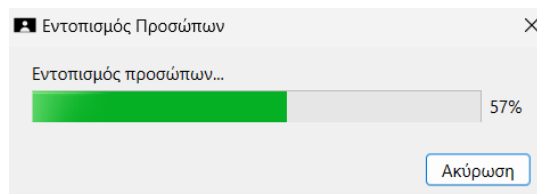
- **Ευαισθησία:** Πρόκειται για το ποσοστό εμπιστοσύνης του αλγορίθμου. Η πιθανότητα δηλαδή που πιστεύει ο αλγόριθμος ότι αυτό που ανίχνευσε είναι όντως κάποιο πρόσωπο. Για παράδειγμα βάζοντας το ποσοστό στο 90% ανιχνεύουμε περιπτώσεις που είναι σχεδόν σίγουρο ότι πρόκειται για κάποιο πρόσωπο.
- **Συντελεστής Σίγασης Θετικών Αποτελεσμάτων:** Είναι ο συντελεστής σίγασης των θετικών αποτελεσμάτων του αλγορίθμου. Ενεργοποιείται όταν οι εντοπισμοί ξεπερνάνε το κατώφλι που θέτει ο χρήστης στο μέγιστο όριο θετικών αποτελεσμάτων.
- **Μέγιστο Όριο Θετικών Αποτελεσμάτων:** Ορίζει τον επιτρεπόμενο αριθμός των θετικών αποτελεσμάτων πριν ενεργοποιηθεί ο συντελεστής σίγασης.
- **Αριθμός Παράλειψης Καρέ:** Κατέχει την ίδια λειτουργία με το προηγούμενο παράθυρο. Για αυτή την ρύθμιση έχουμε δύο επιλογές. Η επιλογή εξαγωγής εικόνων παραλείπει καρέ μόνο στην εξαγωγή αποτελεσμάτων και σε περίπτωση που ενεργοποιήσουμε την επιλογή αποθήκευσης εικόνων από τα πρόσωπα. Η επιλογή παράλειψης καρέ στον εντοπισμό αυξάνει ελάχιστα την ταχύτητα και έχει αντίκτυπο και στις δύο επιλογές αποθήκευσης.
- **Αποθήκευση Εντοπισμών σε Βίντεο:** Αποθηκεύει μόνο τα σημεία όπου έχουμε εντοπισμό προσώπων σε μορφή βίντεο.
- **Εξαγωγή Προσώπων:** Εξάγει τα αποτελέσματα εντοπισμού σε ξεχωριστές φωτογραφίες. Σε περίπτωση που ο χρήστης θελήσει να τρέξει τον αλγόριθμο αναγνώρισης για κάποιο βίντεο πρέπει οπωσδήποτε να ενεργοποιήσει αυτή τη λειτουργία.

- **Επιλογή Φακέλου:** Η εφαρμογή θέτει αυτόματα τον φάκελο αποθήκευσης των αποτελεσμάτων και δίνει την δυνατότητα στον χρήστη να επιλέξει αυτός.



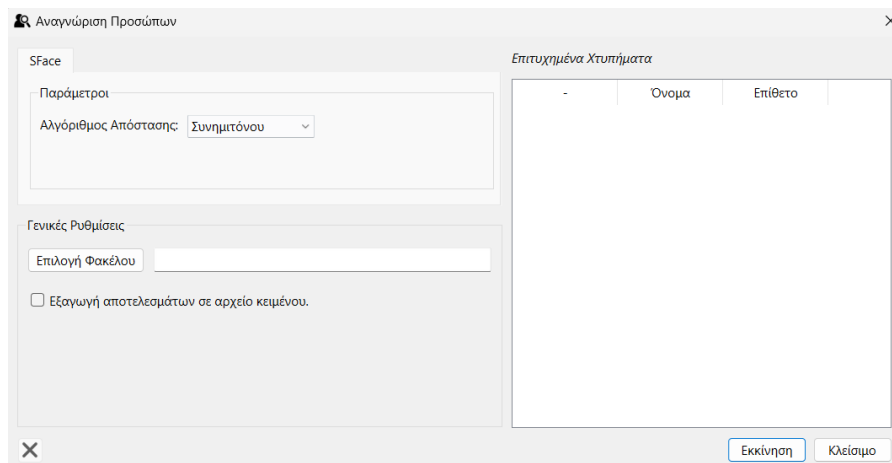
Εικόνα 81: Στιγμιότυπο από την εξαγωγή βίντεο. (Πηγή: Casino Royale 2006)

Στην Εικόνα 79 βλέπουμε ότι ο αλγόριθμος υπογραμμίζει τα πρόσωπα και προβάλλει το ποσοστό εμπιστοσύνης. Επιπλέον, η συγκεκριμένη λειτουργία διαθέτει επίσης την δυνατότητα ενημέρωσης προόδου.



Εικόνα 82: Παράθυρο ενημέρωσης προόδου (εντοπισμός προσώπων).

7.2.4.6 Αναγνώριση Προσώπων

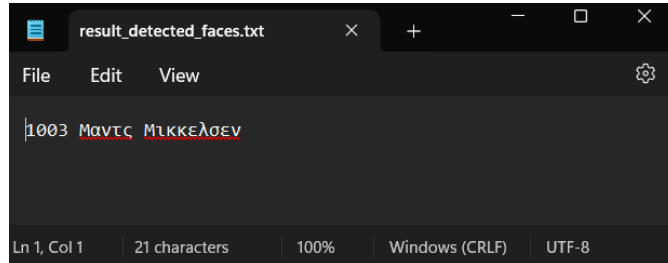


Εικόνα 83: Το παράθυρο αναγνώρισης προσώπων.

Παρέχει την δυνατότητα αναγνώρισης προσώπων με βάση τα άτομα που βρίσκονται στην βάση δεδομένων των υπόπτων. Οι άγνωστες φωτογραφίες συγκρίνονται με ολόκληρη την βάση δεδομένων φωτογραφιών και τα αποτελέσματα προβάλλονται απευθείας σε αυτό το παράθυρο.

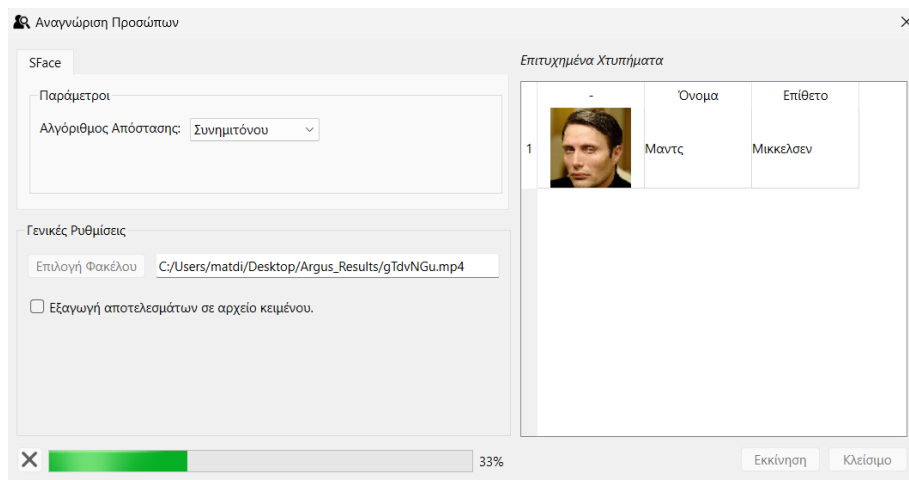
- **Αλγόριθμος Απόστασης:** Ο χρήστης μπορεί να επιλέξει τον τρόπο με τον οποίο συγκρίνονται οι 2 φωτογραφίες. Ο υπολογισμός της ευκλείδειας απόστασης είναι ελαφρώς ταχύτερος.

- **Επιλογή Φακέλου:** Ο χρήστης πρέπει να επιλέξει τον φάκελο στον οποίο περιέχονται οι φωτογραφίες που θα συγκρίνει ο αλγόριθμος με την βάση δεδομένων μας.
- **Εξαγωγή Αποτελεσμάτων σε Αρχείο Κειμένου:** Τα αποτελέσματα αναγνώρισης αποθηκεύονται σε μορφή κειμένου. Αναγράφεται το μοναδικό αναγνωριστικό και το ονοματεπώνυμο.



Εικόνα 84: Αποθήκευση αναγνώρισεων σε αρχείο κειμένου.

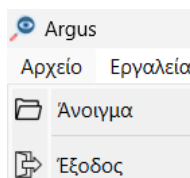
- **Κουμπί Διακοπής:** Στην κάτω αριστερή γωνία βρίσκεται το κουμπί διακοπής με το οποίο ο χρήστης μπορεί να διακόψει την λειτουργία εάν το επιθυμεί.
- **Πίνακας Επιτυχημένων Χτυπημάτων:** Σε αυτόν τον πίνακα εμφανίζονται δυναμικά τα αποτελέσματα της αναγνώρισης. Ο πίνακας δείχνει την φωτογραφία προφίλ του ατόμου που εντοπίστηκε μαζί με το ονοματεπώνυμο του.



Εικόνα 85: Παράδειγμα εμφάνισης επιτυχημένης αναγνώρισης.

7.3 Μενού: «Αρχείο»

Περιέχει βασικές ενέργειες για την λειτουργία της εφαρμογής.



Εικόνα 86: Το μενού «Αρχείο».

7.3.1 Άνοιγμα

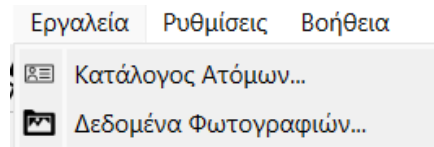
Μέσω αυτής της επιλογής ο χρήστης μπορεί να φορτώσει ένα βίντεο στην εφαρμογή για προβολή ή επεξεργασία.

7.3.2 Έξοδος

Δίνει στον χρήστη την δυνατότητα εξόδου από την εφαρμογή. Έχει την ίδια λειτουργία με το κουμπί «X» στο επάνω δεξιό μέρος του παραθύρου.

7.4 Μενού: «Εργαλεία»

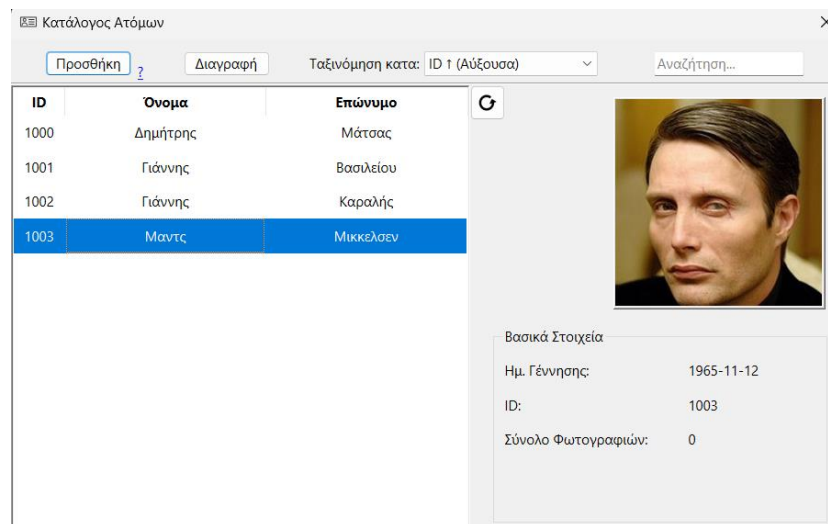
Το μενού «Εργαλεία» περιέχει λειτουργίες για την πρόσβαση στην βάση δεδομένων της εφαρμογής. Διαθέτει τα παράθυρα που είναι πιο πολύπλοκα σε λειτουργία και δεν μπορούν να μπουν στην γραμμή εργαλείων.



Εικόνα 87: Το μενού «Εργαλεία».

7.4.1 Κατάλογος Ατόμων

Ο κατάλογος ατόμων αποτελεί την γραφική διεπαφή διαχείρισης των υπόπτων της βάσης. Διαθέτει λειτουργίες διαγραφής, προσθήκης και αναζήτησης.



Εικόνα 88: Ο κατάλογος ατόμων.

- **Πίνακας Ατόμων:** Προβάλλει μια λίστα με τις καταχωρήσεις των ατόμων στη βάση δεδομένων. Κάνοντας κλικ επάνω σε ένα άτομο εμφανίζονται οι πληροφορίες μαζί με την εικόνα προφίλ στο δεξί μέρος του παραθύρου.
- **Κουμπί Ανανέωσης:** Βρίσκεται δίπλα στον πίνακα. Με το κουμπί αυτό ο χρήστης μπορεί να ανανεώσει τον πίνακα σε περίπτωση κάποιας διαγραφής ή προσθήκης για να εμφανιστούν οι αλλαγές.
- **Πεδίο Αναζήτησης:** Ο χρήστης μπορεί να πληκτρολογήσει οποιοδήποτε από τα τρία πεδία επιθυμεί για να αναζητήσει. Τα αποτελέσματα της αναζήτησης εμφανίζονται δυναμικά στον πίνακα όσο πληκτρολογεί ο χρήστης.
- **Επιλογές Ταξινόμησης:** Ο χρήστης μπορεί να ταξινομήσει τον πίνακα με βάση το ID (αύξουσα ή φθίνουσα σειρά) ή αλφαβητικά.
- **Κουμπί Διαγραφής:** Διαγράφει το επιλεγμένο άτομο από την βάση δεδομένων. Η συγκεκριμένη επιλογή διαγράφει επίσης και όλα τα φωτογραφικά δεδομένα για το συγκεκριμένο άτομο.

- **Κουμπί Προσθήκης:** Ανοίγει το παράθυρο προσθήκης ενός νέου ατόμου στην βάση.

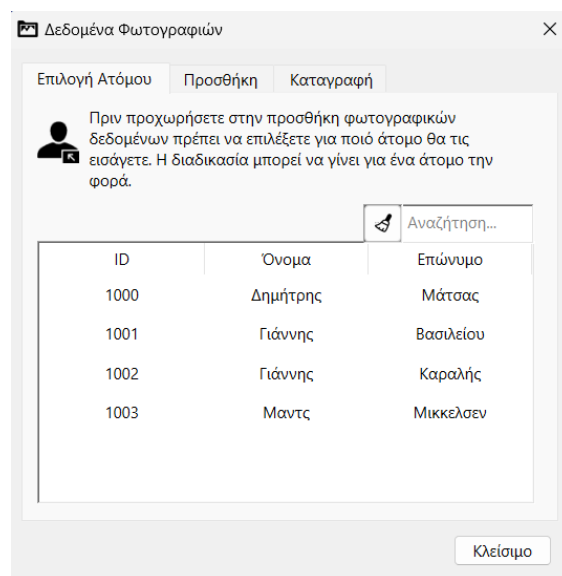
Εικόνα 89: Το παράθυρο προσθήκης ατόμων.

Το παράθυρο προσθήκης περιέχει μια φόρμα την οποία ο χρήστης πρέπει να συμπληρώσει με τα κατάλληλα στοιχεία για να εισάγει ένα καινούργιο άτομο στην βάση δεδομένων. Ας δούμε ξεχωριστά τα δεδομένα που απαιτεί το κάθε πεδίο:

- **Όνομα:** Το όνομα του ατόμου με ελληνικούς χαρακτήρες. Η χρήση αριθμών ή λατινικών χαρακτήρων δεν επιτρέπεται. Σε περίπτωση που το όνομα είναι μικρότερο ή ίσο με δύο χαρακτήρες εμφανίζεται σχετικό μήνυμα λάθους κάτω από το πεδίο.
- **Επώνυμο:** Το επώνυμο του ατόμου με ελληνικούς χαρακτήρες. Διαθέτει ακριβώς τα ίδια χαρακτηριστικά με το πεδίο ονόματος.
- **Μοναδικό Αναγνωριστικό (ID):** Δέχεται τιμές από 1000 έως 9999. Σε περίπτωση που ο χρήστης πληκτρολογήσει αριθμό εκτός ορίων θα εμφανιστεί σχετικό μήνυμα λάθους. Εάν ο χρήστης επιλέξει έναν αριθμό που υπάρχει ήδη στο σύστημα θα ενημερωθεί με κατάλληλο μήνυμα σφάλματος μόλις πατήσει το κουμπί «Αποθήκευση».
- **Ημερομηνία Γέννησης:** Ο χρήστης πρέπει να εισάγει την ημερομηνία γεννήσεως του ατόμου.
- **Όνομα (Λατινικά):** Το όνομα του ατόμου με λατινικούς χαρακτήρες. Δεν επιτρέπονται αριθμοί ή ελληνικοί χαρακτήρες. Έχει ίδιους ελέγχους με το πεδίο ονόματος.
- **Επίθετο (Λατινικά):** Το επώνυμο του ατόμου με λατινικούς χαρακτήρες.
- **Προσθήκη Φωτογραφίας:** Πατώντας το κουμπί με το εικονίδιο ενός φακέλου ο χρήστης πρέπει να επιλέξει μια φωτογραφία προφίλ μεγέθους 256 πλάτος επί 256 ύψος. Σε περίπτωση που ο χρήστης εισάγει μια μη έγκυρη εικόνα, εμφανίζεται σχετικό μήνυμα λάθους.
- **Κουμπί Αποθήκευσης:** Ενεργοποιείται μόνο σε περίπτωση που όλα τα στοιχεία είναι έγκυρα και τα αποθηκεύει. Κατά την ολοκλήρωση της αποθήκευσης εμφανίζει σχετικό μήνυμα επιτυχίας.

7.4.2 Δεδομένα Φωτογραφιών

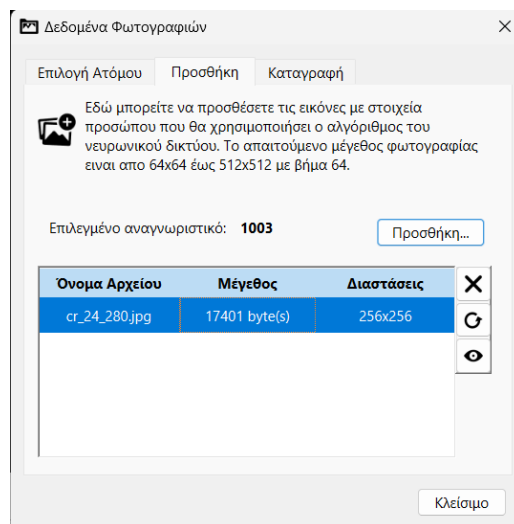
Το παράθυρο δεδομένων φωτογραφιών διαθέτει λειτουργίες διαχείρισης της βάσης δεδομένων φωτογραφιών για τον κάθε ύποπτο που βρίσκεται στην βάση. Ο κάθε ύποπτος διαθέτει έναν ατομικό φάκελο με την ονομασία «dataset» στον οποίο αποθηκεύονται αυτές οι φωτογραφίες. Το παράθυρο χωρίζεται σε τρεις καρτέλες η κάθε μια με ξεχωριστές λειτουργίες.



Εικόνα 90: Η καρτέλα «Επιλογή Ατόμου».

Η καρτέλα «Επιλογή Ατόμου» αποτελεί την πρώτη φάση της διαδικασίας προσθήκης φωτογραφικών δεδομένων και διαθέτει τις εξής λειτουργίες:

- **Πεδίο Αναζήτησης:** Ο χρήστης μπορεί να αναζητήσει με βάση το όνομα, το επώνυμο ή το ID. Ο πίνακας ενημερώνεται δυναμικά όσο πληκτρολογεί ο χρήστης και εμφανίζει τα κατάλληλα αποτελέσματα.
- **Πίνακας Ατόμων:** Εμφανίζει τα περιεχόμενα της βάσης. Κάνοντας κλικ σε μια γραμμή ενεργοποιείται η καρτέλα «Προσθήκη».
- **Κουμπί Εκκαθάρισης:** Σε περίπτωση που ο χρήστης επιθυμεί να επιλέξει άλλο άτομο, μπορεί να επαναφέρει τον πίνακα στην αρχική του κατάσταση πατώντας το κουμπί με το εικονίδιο της σκούπας.



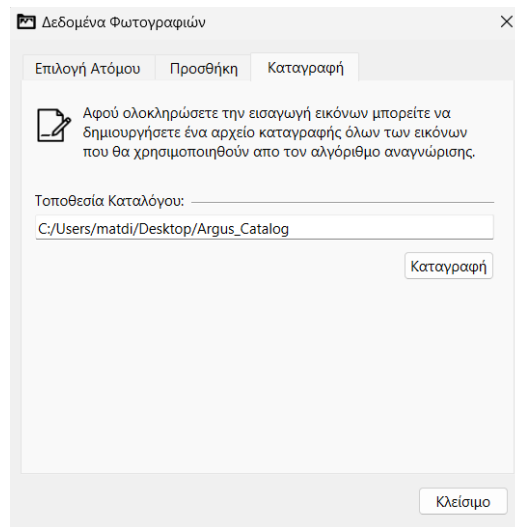
Εικόνα 91: Η καρτέλα «Προσθήκη».

Η καρτέλα «Προσθήκη» ενεργοποιείται μόλις ο χρήστης επιλέξει ένα άτομο από την καρτέλα «Επιλογή Ατόμου». Στην καρτέλα εμφανίζεται το επιλεγμένο ID και δίνει την δυνατότητα στον χρήστη να προσθέσει φωτογραφίες στον φάκελο «dataset» του επιλεγμένου ατόμου.

- **Πίνακας Εικόνων:** Εδώ εμφανίζονται όλες οι εικόνες που βρίσκονται στον φάκελο «dataset» με το όνομα του αρχείου, το μέγεθος και τις διαστάσεις τους.

- **Κουμπί Προσθήκης:** Ο χρήστης μπορεί να επιλέξει μια εικόνα από τον υπολογιστή του για να προσθέσει στον φάκελο του συγκεκριμένου ατόμου. Το επιτρεπόμενο μέγεθος για τις εικόνες είναι από 64x64 έως 512x512.
- **Κουμπί Διαγραφής:** Βρίσκεται δίπλα από τον πίνακα στην δεξιά μεριά (εικονίδιο X). Το συγκεκριμένο κουμπί διαγράφει την επιλεγμένη φωτογραφία από την βάση.
- **Κουμπί Ανανέωσης:** Ανανεώνει τον πίνακα σε περίπτωση προσθήκης ή διαγραφής μιας φωτογραφίας.
- **Κουμπί Προεπισκόπησης:** Ανοίγει ένα παράθυρο το οποίο προβάλλει την επιλεγμένη εικόνα.

Μόλις ο χρήστης ολοκληρώσει την διαδικασία προσθήκης, πρέπει να μεταβεί στην καρτέλα «Καταγραφή» για να καταγράψει τις νέες εικόνες στο αρχείο εικόνων.

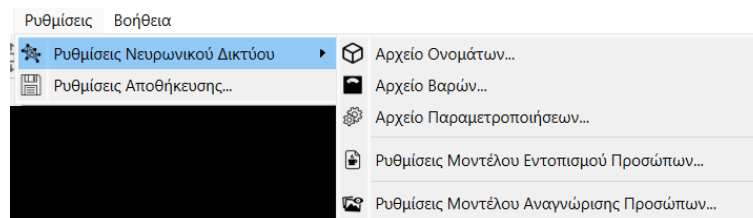


Εικόνα 92: Η καρτέλα «Καταγραφή».

- **Τοποθεσία Καταλόγου:** Συμπληρώνεται αυτόματα από το σύστημα και δεν μπορεί να αλλάξει από εδώ. Ο χρήστης πρέπει να μεταβεί στο παράθυρο «Ρυθμίσεις Αποθήκευσης» για να παραμετροποιήσει την τοποθεσία αποθήκευσης καταλόγου.
- **Κουμπί Καταγραφής:** Δημιουργεί ένα index αρχείο με όλες τις διαδρομές των εικόνων που βρίσκονται στην βάση δεδομένων. Το αρχείο αυτό είναι απαραίτητο για την λειτουργία του αλγορίθμου αναγνώρισης προσώπων.

7.5 Μενού: «Ρυθμίσεις»

Το συγκεκριμένου μενού περιέχει όλα τα παράθυρα ρυθμίσεων της εφαρμογής.



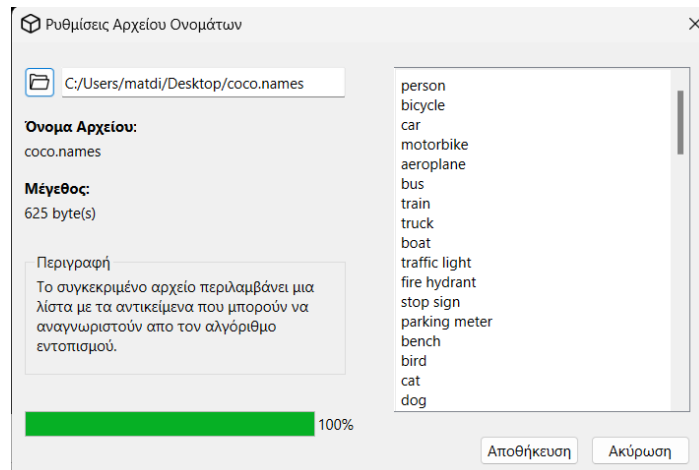
Εικόνα 93: Το μενού «Ρυθμίσεις».

7.5.1 Ρυθμίσεις Νευρωνικού Δικτύου

Το υπομενού αυτό περιέχει τις ρυθμίσεις που αφορούν τα νευρωνικά δίκτυα. Οι επιλογές «Αρχείο Ονομάτων», «Αρχείο Βαρών» και «Αρχείο Παραμετροποιήσεων» αφορούν το νευρωνικό δίκτυο εντοπισμού κίνησης.

7.5.1.1 Αρχείο Ονομάτων

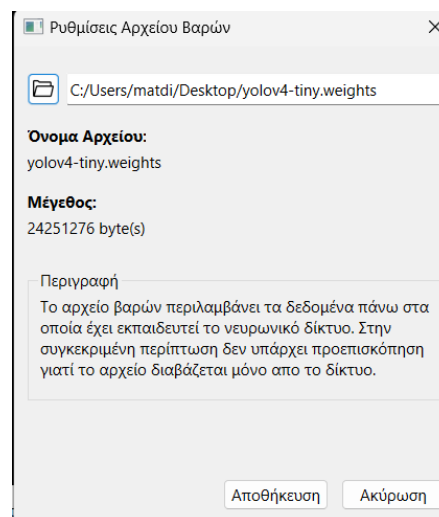
Στο συγκεκριμένο παράθυρο ο χρήστης πρέπει να επιλέξει ένα αρχείο ονομάτων πατώντας το κουμπί με το σύμβολο του φακέλου πάνω αριστερά. Με το άνοιγμα του αρχείου προβάλλονται στο παράθυρο οι πληροφορίες του ονόματος, το μέγεθος καθώς και τα περιεχόμενα του αρχείου.



Εικόνα 94: Το παράθυρο «Ρυθμίσεις Αρχείου Ονομάτων».

7.5.1.2 Αρχείο Βαρών

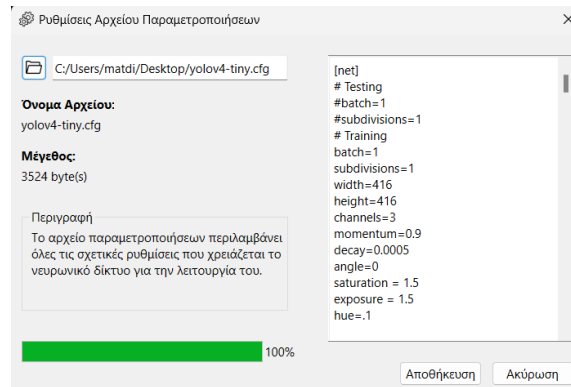
Σε αυτό το παράθυρο ο χρήστης πρέπει να επιλέξει ένα αρχείο βαρών πατώντας το κουμπί με το σύμβολο του φακέλου επάνω αριστερά. Με το άνοιγμα του αρχείου προβάλλονται στο παράθυρο οι πληροφορίες του ονόματος και το μέγεθος.



Εικόνα 95: Το παράθυρο «Ρυθμίσεις Αρχείου Βαρών».

7.5.1.3 Αρχείο Παραμετροποιήσεων

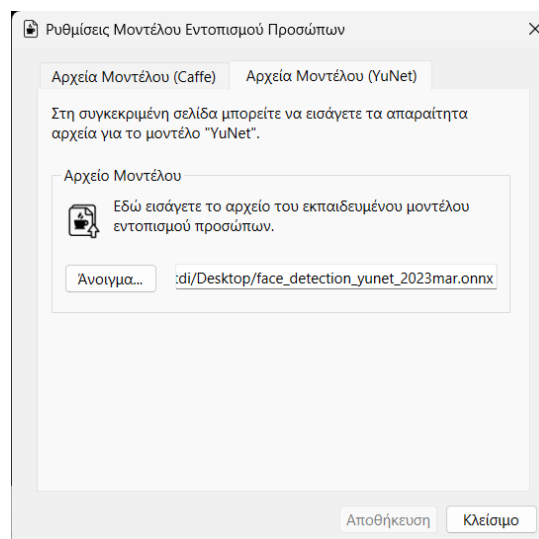
Στο συγκεκριμένο παράθυρο ο χρήστης πρέπει να επιλέξει ένα αρχείο παραμετροποιήσεων πατώντας το κουμπί με το σύμβολο του φακέλου πάνω αριστερά. Με το άνοιγμα του αρχείου προβάλλονται στο παράθυρο οι πληροφορίες του ονόματος, το μέγεθος καθώς και τα περιεχόμενα του αρχείου.



Εικόνα 96: Το παράθυρο «Ρυθμίσεις Αρχείου Παραμετροποιήσεων».

7.5.1.4 Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων

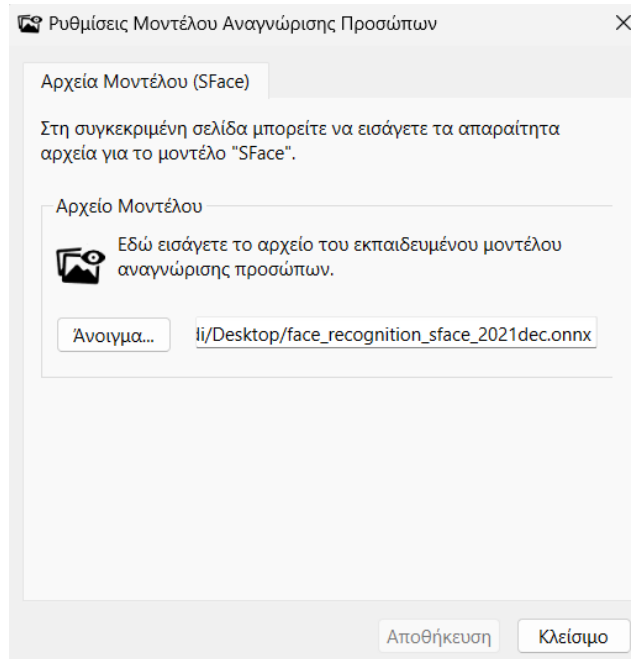
Σε αυτό το παράθυρο ρυθμίζεται η τοποθεσία του αρχείου που χρησιμοποιεί το μοντέλο YUNet για τον εντοπισμό προσώπων. Πατώντας το κουμπί «Άνοιγμα» ο χρήστης μπορεί να το επιλέξει από τον υπολογιστή του.



Εικόνα 97: Το παράθυρο «Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων».

7.5.1.5 Ρυθμίσεις Μοντέλου Αναγνώρισης Προσώπων

Σε αυτό το παράθυρο ρυθμίζεται η τοποθεσία του αρχείου που χρησιμοποιεί το μοντέλο SFace για την αναγνώριση προσώπων. Πατώντας το κουμπί «Άνοιγμα» ο χρήστης μπορεί να το επιλέξει από τον υπολογιστή του.

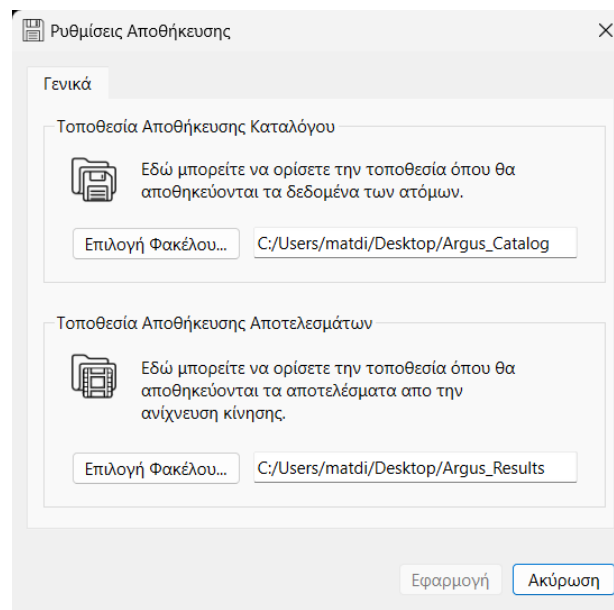


Εικόνα 98: Το παράθυρο «Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων».

7.5.2 Ρυθμίσεις Αποθήκευσης

Εδώ ο χρήστης μπορεί να ρυθμίσει τους φακέλους αποθήκευσης που θα χρησιμοποιεί η εφαρμογή.

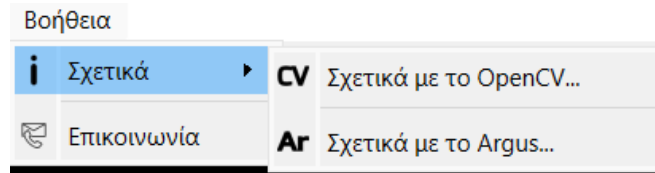
- **Τοποθεσία Αποθήκευσης Καταλόγου:** Εδώ ρυθμίζεται ο φάκελος αποθήκευσης του καταλόγου ατόμων και της βάσης δεδομένων φωτογραφιών.
- **Τοποθεσία Αποθήκευσης Αποτελεσμάτων:** Ο χρήστης μπορεί να επιλέξει τον φάκελο στον οποίο θα αποθηκεύονται τα αποτελέσματα που εξάγουν οι αλγόριθμοι της εφαρμογής.



Εικόνα 99: Το παράθυρο «Ρυθμίσεις Αποθήκευσης».

7.6 Μενού: «Βοήθεια»

Το μενού «Βοήθεια» περιλαμβάνει τα στοιχεία επικοινωνίας του δημιουργού για την αποστολή ιδεών και προβλημάτων. Επιπλέον, εδώ βρίσκονται τα παράθυρα πληροφοριών των εργαλείων που χρησιμοποιήθηκαν.

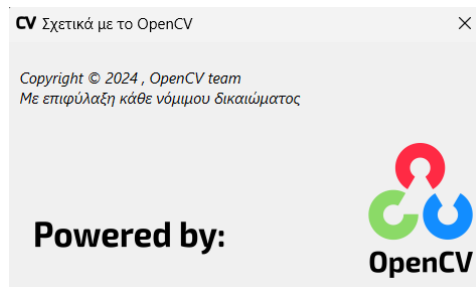


Εικόνα 100: Το μενού «Βοήθεια».

7.6.1 Σχετικά

Εδώ περιλαμβάνονται τα παράθυρα σχετικά με τα εργαλεία και τους δημιουργούς της εφαρμογής.

7.6.1.1 Σχετικά με το OpenCV



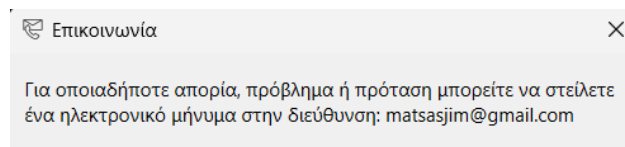
Εικόνα 101: Το παράθυρο «Σχετικά με το OpenCV».

7.6.1.2 Σχετικά με το Argus



Εικόνα 102: Το παράθυρο «Σχετικά με το Argus».

7.6.2 Επικοινωνία



Εικόνα 103: Το παράθυρο «Επικοινωνία».

7.7 Οδηγίες Εγκατάστασης

Για να εγκαταστήσετε την εφαρμογή στον υπολογιστή σας ακολουθείτε τα εξής βήματα:

1. Κατεβάζετε τον φάκελο Argus Application από τον σύνδεσμο:
https://drive.google.com/drive/folders/1nbGT51o5tl1f84ENhC3rhGtZfprQOCL4?usp=drive_link
2. Κατεβάζετε τον φάκελο DNN Files από τον σύνδεσμο:
https://drive.google.com/drive/folders/1I5Dm7FVd4tGBznKM-czqQ8kTmudg9fR?usp=drive_link
3. Αποσυμπιέστε τα αρχεία στην επιφάνεια εργασίας σε ξεχωριστούς φακέλους.
4. Για να ξεκινήσει η εφαρμογή τρέχετε το αρχείο Argus.exe **ΧΩΡΙΣ** να το μετακινήσετε εκτός φακέλου. (Μπορείτε να δημιουργήσετε shortcut στο desktop).
5. Για τις οδηγίες χρήσης αναφερθείτε στο Κεφάλαιο 7 της εργασίας.

8. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ

8.1 Συμπεράσματα

8.1.1 Σχετικά με τα εργαλεία

Αρχικά, επιλέχθηκε η γλώσσα C++ για την ανάπτυξη της εφαρμογής. Μετά από την αρχική ανάλυση των λειτουργικών απαιτήσεων, διαπιστώθηκε ότι πρόκειται για μία εφαρμογή μεγάλης έκτασης, όπου γλώσσες όπως η Python δεν θα μπορούσαν να ανταπεξέλθουν με την ίδια αποτελεσματικότητα στις απαιτήσεις επεξεργασίας. Η αντικειμενοστρέφεια που προσφέρει η C++ από τη φύση της, καθώς και η σύνταξή της, συμβάλλουν στην καλύτερη οργάνωση του κώδικα. Ένας ακόμα σημαντικός λόγος επιλογής της γλώσσας αυτής είναι η εκτενής γνώση που έχω αποκτήσει κατά τη διάρκεια των προπτυχιακών μου σπουδών. Παρόλο που η C++ θεωρείται σχετικά δύσκολη στην εκμάθηση, τα οφέλη που προσφέρει ανταμείβουν τους προγραμματιστές που την χρησιμοποιούν.

Επιπλέον, η C++ διαθέτει μεγαλύτερη γκάμα εργαλείων ανάπτυξης μεγάλων εφαρμογών, όπως το πλαίσιο ανάπτυξης Qt και οι αλγόριθμοι υπολογιστικής όρασης της OpenCV, που έχουν γραφτεί και τα δύο σε C++, οπότε ενσωματώθηκαν άριστα στην εφαρμογή. Όλα αυτά συνδέθηκαν μέσω της πλατφόρμας ανάπτυξης εφαρμογών της Microsoft, Visual Studio 2022, η οποία, χάρη στις δυνατότητες debugging και IntelliSense που προσφέρει, αύξησε σημαντικά την ταχύτητα ανάπτυξης της εφαρμογής. Το λεπτομερές debugging με βοήθησε να επιλύω γρήγορα πιθανά bugs που εμφανίζονταν μετά από κάθε sprint ανάπτυξης, ενώ το IntelliSense με ενημέρωνε σχετικά με τις μεθόδους της κάθε βιβλιοθήκης.

Παρόλο που το Qt διαθέτει το δικό του περιβάλλον ανάπτυξης εφαρμογών, το Qt Creator, οι βιβλιοθήκες υπολογιστικής όρασης δεν μπορούσαν να ενσωματωθούν σωστά. Έτσι, μετά από αναζήτηση πιθανών λύσεων, βρέθηκε το extension Qt Tools για το Visual Studio 2022, το οποίο ενσωματώνει όλες τις βιβλιοθήκες που προσφέρει το Qt μαζί με το εργαλείο σχεδίασης παραθύρων (Qt Designer) στο Visual Studio. Μετά από ενάμιση μήνα εργασίας, έγινε η μεταφορά της εφαρμογής στο περιβάλλον του Visual Studio, όπου και παρέμεινε μέχρι την ολοκλήρωση της ανάπτυξης της.

Η ενσωμάτωση της OpenCV στο Visual Studio ήταν εξίσου εύκολη και λειτουργική. Χρειάζοταν ιδιαίτερη προσοχή κατά το compile του πηγαίου κώδικα, το οποίο έπρεπε να γίνει μέσω του προγράμματος CMake. Συγκεκριμένα, έπρεπε να επιλέξω τον κατάλληλο compiler ώστε να είναι συμβατός με την έκδοση του Visual Studio που θα χρησιμοποιούσα. Ένα από τα θετικά των εργαλείων αυτών είναι ότι διαθέτουν οργανωμένες κοινότητες πολλών χρόνων. Οι κοινότητες αυτές παρείχαν απαντήσεις για πολλά σχεδιαστικά προβλήματα που αντιμετώπισα κατά τη διάρκεια της ανάπτυξης.

Όσον αφορά την OpenCV, η χρήση της ξεκίνησε σταδιακά. Αρχικά, έγινε απλή ανάγνωση και επεξεργασία των βίντεο και στη συνέχεια άρχισε η χρήση των νευρωνικών δικτύων. Για να μπορέσω να αξιοποιήσω κατάλληλα τις δυνατότητές τους, χρειάστηκε να πραγματοποιήσω εκτεταμένες έρευνες και να μελετήσω παραδείγματα σχετικά με τη χρήση τους και τον κώδικα που απαιτούσαν. Στον τομέα της αναγνώρισης και του εντοπισμού προσώπων, την λύση παρείχαν κάποιοι πρόσθετοι αλγόριθμοι της OpenCV. Αν και τα παραδείγματα κώδικα ήταν σε Python, η μετατροπή τους σε C++ ήταν γρήγορη και αποτελεσματική.

8.1.2 Σχετικά με το σύστημα

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε σε διαδοχικά στάδια, ακολουθώντας τη μέθοδο Agile, η οποία διαχωρίζει την ανάπτυξη κάθε λειτουργίας σε sprints. Αρχικά, αναπτύχθηκε ο

πυρήνας της εφαρμογής, δηλαδή η κεντρική γραφική διεπαφή, πάνω στην οποία θα βασίζονταν οι υπόλοιπες λειτουργίες, όπως η προβολή βίντεο και εικόνων. Η απόφαση να ξεκινήσει η ανάπτυξη με αυτόν τον τρόπο είχε ως στόχο την εξοικείωσή μου με τα εργαλεία του Qt και την κατανόηση της διαδικασίας δημιουργίας μιας απλής και λειτουργικής διεπαφής χρήστη. Μια από τις βασικές απαιτήσεις ήταν η δημιουργία μιας διεπαφής που θα ήταν εύκολη στην εκμάθηση ακόμα και από χρήστες χωρίς ιδιαίτερες γνώσεις υπολογιστών. Σε αυτό συνέβαλε και η επιλογή της ελληνικής γλώσσας, προκειμένου το λογισμικό να είναι εύχρηστο για τους υπαλλήλους της εκάστοτε αστυνομικής υπηρεσίας.

Το δεύτερο στάδιο ανάπτυξης αφορούσε τη μελέτη των βιβλιοθηκών της OpenCV και την ανάπτυξη της ανίχνευσης κίνησης, το οποίο αποδείχθηκε το πιο δύσκολο κομμάτι. Η δυσκολία εντοπίστηκε κυρίως στην εύρεση των κατάλληλων μεθόδων και συναρτήσεων για την ανίχνευση κίνησης με τρόπο που δεν επιβάρυνε υπερβολικά το σύστημα. Επιπλέον, η εκμάθηση μιας βιβλιοθήκης υπολογιστικής όρασης απαιτεί χρόνο για την κατανόηση των λειτουργιών και την ενσωμάτωσή τους στο υπόλοιπο πρόγραμμα. Ένα άλλο πρόβλημα που αντιμετωπίστηκε ήταν το πάγωμα της γραφικής διεπαφής κατά τη διάρκεια πολύπλοκων διεργασιών. Η λύση αυτού του προβλήματος με τη χρήση threads βελτίωσε όχι μόνο τη λειτουργικότητα και την αισθητική του UI, αλλά και τη δυνατότητα της εφαρμογής να παρέχει άμεση ανατροφοδότηση στον χρήστη.

Τα επόμενα στάδια ανάπτυξης, που αφορούσαν τη βάση δεδομένων και τις λειτουργίες αναγνώρισης προσώπων, ήταν πιο σύντομα λόγω της επαναχρησιμοποίησης κώδικα που είχα προνοήσει να γράψω. Οι δυνατότητες της εφαρμογής αναπτύχθηκαν πλήρως σύμφωνα με τις λειτουργικές απαιτήσεις που καθορίστηκαν αρχικά. Οι λειτουργίες ανίχνευσης κίνησης, εντοπισμού και αναγνώρισης προσώπων λειτουργούν αξιόπιστα και παράγουν τα αναμενόμενα αποτελέσματα, είτε εντός της εφαρμογής είτε σε μορφή αρχείων.

Η δυνατότητα διαχείρισης πολλών βίντεο ταυτόχρονα, όπως είχε προταθεί αρχικά, αποδείχθηκε μη εφικτή λόγω των υψηλών υπολογιστικών πόρων που απαιτεί. Επιπλέον, οι λειτουργίες της εφαρμογής διαχωρίστηκαν με τέτοιο τρόπο ώστε να είναι χρήσιμες ανεξάρτητα η μία από την άλλη. Λαμβάνοντας υπόψη την ευαίσθητη φύση των δεδομένων που επεξεργάζεται η εφαρμογή, η οποία αφορά εγκληματολογικά στοιχεία, δεν δόθηκε καμία πρόσβαση σε υπηρεσίες διαδικτύου για την αποφυγή προβλημάτων ασφαλείας. Η βάση δεδομένων αποθηκεύεται τοπικά στον υπολογιστή, χωρίς τη χρήση SQL βάσεων που τρέχουν σε τοπικά δίκτυα ή στο διαδίκτυο.

8.2 Αδυναμίες – Όρια του Συστήματος

Το σύστημα αντιμετωπίζει ορισμένες αδυναμίες που επηρεάζουν την απόδοσή του. Ένα από τα κύρια προβλήματα είναι η αδυναμία επεξεργασίας πολλαπλών βίντεο ταυτόχρονα, λόγω των υψηλών υπολογιστικών πόρων που απαιτούνται. Ο χρόνος επεξεργασίας των βίντεο εξαρτάται από πολλούς παράγοντες, όπως η ισχύς του επεξεργαστή του υπολογιστή, η ανάλυση του βίντεο, τα καρέ ανά δευτερόλεπτο, η μνήμη κ.α. Μείωση του χρόνου επεξεργασίας μπορεί να επιτευχθεί με τη μείωση της ανάλυσης του βίντεο και των καρέ, διότι ο αλγόριθμος θα επεξεργάζεται λιγότερα εικονοστοιχεία, επιταχύνοντας έτσι την εξαγωγή των αποτελεσμάτων.

Παρά τη δύναμη και την ακρίβεια των νευρωνικών δικτύων, υπάρχει πάντοτε η πιθανότητα λάθους ή αδυναμίας εντοπισμού. Αυτό συμβαίνει κυρίως σε περιπτώσεις όπου ένας ύποπτος εμφανίζεται σε γωνία όπου δεν διακρίνεται καθαρά το πρόσωπό του, όπως όταν φαίνεται το πίσω μέρος του κεφαλιού. Επιπλέον, ο ανεπαρκής φωτισμός της εικόνας μπορεί να εμποδίσει το σύστημα να διακρίνει τα χαρακτηριστικά του προσώπου, αφού δεν υπάρχουν διακριτές διαφορές μεταξύ της σκηνής (background) και του ατόμου όπου εμφανίζεται στην κάμερα.

Η ανάλυση στην οποία αποθηκεύονται τα πρόσωπα των φωτογραφιών παίζει επίσης σημαντικό ρόλο. Όσο μικρότερη είναι η ανάλυση, τόσο λιγότερα εικονοστοιχεία έχει στη διάθεσή του ο αλγόριθμος για επεξεργασία, με αποτέλεσμα να μην μπορεί να εντοπίσει με ακρίβεια το πρόσωπο στην εικόνα. Σε τέτοιες περιπτώσεις, ο χρήστης πρέπει να προβεί σε περαιτέρω προεπεξεργασία της εικόνας, εφαρμόζοντας τεχνικές τεχνητής αύξησης της ανάλυσης (upscaling), ώστε να βελτιωθεί η ποιότητα της εικόνας και να μπορεί να συγκριθεί σωστά με τη βάση δεδομένων.

Η αντιμετώπιση αυτών των αδυναμιών απαιτεί προσεκτικό σχεδιασμό και ενδεχομένως την ενσωμάτωση πρόσθετων τεχνολογιών και τεχνικών για τη βελτιστοποίηση της απόδοσης και της ακρίβειας του συστήματος.

8.3 Επόμενα Βήματα

Στο συγκεκριμένο στάδιο η εφαρμογή μπορεί να θεωρηθεί μια πλήρως λειτουργική πλατφόρμα η οποία μπορεί να χρησιμοποιηθεί από τις αστυνομικές υπηρεσίες για την διευκόλυνση των υπαλλήλων στην επεξεργασία των στοιχείων. Ωστόσο, θα μπορούσαν να πραγματοποιηθούν περαιτέρω επεκτάσεις ώστε να μετατραπεί σε μια πλήρη σουίτα ελέγχου αστυνομικών υποθέσεων. Όπως είδαμε στην ανάλυση πεδίου σε προηγούμενο κεφάλαιο η εφαρμογή IDentify χρησιμοποιεί ένα πρωτοποριακό σύστημα διαχείρισης των στοιχείων με βάση τις υποθέσεις και την κατηγοριοποίηση τους σε ανοιχτές, κλειστές και αρχειοθετημένες. Θα μπορούσα λοιπόν να εφαρμόσω μια παρόμοια μέθοδο διαχείρισης των στοιχείων κάνοντας την εφαρμογή να λειτουργεί με βάση μια ανοιχτή υπόθεση. Πιο συγκεκριμένα, ο χρήστης θα μπορεί να επιλέγει μια ανοιχτή υπόθεση η οποία κατέχει τον δικό της φάκελο με στοιχεία και αποτελέσματα. Επιπλέον, μπορεί να βρίσκεται σε άμεση σύνδεση με την βάση δεδομένων των υπόπτων ώστε να ειδοποιεί την αστυνομική διεύθυνση για πιθανόν νέες ταυτοποιήσεις στα διαθέσιμα στοιχεία. Έτσι συνδυάζουμε τα θετικά και των δύο εφαρμογών σε ένα πανίσχυρο νέο σύστημα. Ακόμα πιο φιλόδοξη ιδέα αποτελεί η χρήση της εφαρμογής από πολλά αστυνομικά τμήματα χρησιμοποιώντας μια κοινή βάση δεδομένων προσβάσιμη από όλα τα τμήματα μέσω του δικτύου της αστυνομίας. Επίσης, θα ήταν ιδανική η δοκιμή της εφαρμογής σε συνθήκες πραγματικού χρόνου για την εύρεση σφαλμάτων, bugs και σχεδιαστικών λαθών που μπορεί να ξέφυγαν κατά την διάρκεια των δοκιμών που έγιναν στην φάση ανάπτυξης.

ΠΑΡΑΡΤΗΜΑΤΑ

A. Ενδεικτικά Παραδείγματα Πηγαίου Κώδικα

A.1 Συνάρτηση Εντοπισμού Αντικειμένων

```

1. void MotionDetectionThread::detectObjects(cv::Mat& frame, std::vector<int>&
classIds,
2.     std::vector<float>& scores, std::vector<cv::Rect>& boxes, const
std::vector<std::string>& classes, double sensitivity)
3. {
4.     detectionModel.detect(frame, classIds, scores, boxes, sensitivity, 0.4);
5.
6.     for (int i = 0; i < classIds.size(); i++)
7.     {
8.         cv::rectangle(frame, boxes[i], cv::Scalar(0, 255, 0), 2);
9.         char text[100];
10.        snprintf(text, sizeof(text), "%s: %.2f", classes[classIds[i]].c_str(),
scores[i]);
11.        cv::putText(frame, text, cv::Point(boxes[i].x, boxes[i].y - 5),
cv::FONT_HERSHEY_SIMPLEX, 1,
12.            cv::Scalar(0, 255, 0), 2);
13.        // If timestamp flag is enabled write the detected classes to a vector of
strings.
14.        if (timestampFlag == true)
15.        {
16.            detectedClasses.push_back(classes[classIds[i]].c_str());
17.        }
18.    }
19. }
20.

```

A.2 Κώδικας Δημιουργίας Παραθύρου Προειδοποίησης

```

1. int CatalogDialog::createWarningDialog()
2. {
3.     QMessageBox warningBox;
4.     warningBox.setWindowTitle("Επιβεβαίωση Διαγραφής");
5.     warningBox.setText("Όλα τα δεδομένα για το συγκεκριμένο άτομο θα διαγραφούν.
Συνέχεια;");
6.     QPushButton* okButton = warningBox.addButton(tr("OK"), QMessageBox::AcceptRole);
7.     QPushButton* cancelButton = warningBox.addButton(tr("Ακύρωση"),
QMessageBox::RejectRole);
8.     warningBox.setDefaultButton(okButton);
9.     warningBox.setIcon(QMessageBox::Warning);
10.    QIcon winIcon = QIcon("C:/Users/matdi/Documents/Argus/res/icons/id.png");
11.    warningBox.setWindowIcon(winIcon);
12.    warningBox.exec();
13.
14.    if (warningBox.clickedButton() == okButton)
15.    {
16.        return 0;
17.    }
18.    else if (warningBox.clickedButton() == cancelButton)
19.    {
20.        return 1;
21.    }
22.
23.    return 1;
24. }
25.

```

Α.3 Συνάρτηση Προσθήκης Ατόμου στην Βάση

```

1. int CatalogService::addPerson(QString name, QString surname, QString id, QDate dob,
   QString latinName, QString latinSurname, QString profilePicturePath) const
2. {
3.     // Append ID, name, and surname to id.txt.
4.     QFile catalogFile(catalogFolderPath + "/catalog.txt");
5.     if (!catalogFile.open(QIODevice::WriteOnly | QIODevice::Append |
   QIODevice::Text))
6.     {
7.         return 1;
8.     }
9.     QTextStream stream(&catalogFile);
10.    stream << id << " " << name << " " << surname << "\n";
11.    catalogFile.close();
12.
13.    // Create folder named with ID and add a file with ID.txt.
14.    QString folderName = catalogFolderPath + "/" + id;
15.    QDir dir(folderName);
16.    if (!dir.exists())
17.    {
18.        if (!dir.mkpath(folderName))
19.        {
20.            qDebug() << "Αποτυχία δημιουργίας φακέλου:" << folderName;
21.            return 1;
22.        }
23.    }
24.
25.    // Create empty folder named "dataset" in the person folder.
26.    QString datasetFolder = folderName + "/dataset";
27.    QDir datasetDir(datasetFolder);
28.    if (!datasetDir.exists())
29.    {
30.        datasetDir.mkpath(".");
31.    }
32.
33.    // Copy the profile picture to the appropriate folder.
34.    QString destinationPath = folderName + "/" +
   QFileInfo(profilePicturePath).fileName();
35.    if (QFile::copy(profilePicturePath, destinationPath))
36.    {
37.        qDebug() << "Profile picture copied successfully to:" << destinationPath;
38.    }
39.    else
40.    {
41.        qDebug() << "Failed to copy profile picture to:" << destinationPath;
42.    }
43.
44.    // Write the information to the ID.txt file.
45.    QString fileName = folderName + "/" + id + ".txt";
46.    QFile personFile(fileName);
47.    if (!personFile.open(QIODevice::WriteOnly | QIODevice::Text))
48.    {
49.        return 1;
50.    }
51.    QTextStream personStream(&personFile);
52.    personStream << name << "\n";
53.    personStream << surname << "\n";
54.    personStream << id << "\n";
55.    personStream << dob.toString("yyyy-MM-dd") << "\n";
56.    personStream << latinName << "\n";
57.    personStream << latinSurname << "\n";
58.    personStream << "No\n";
59.    personStream << "0\n";
60.    personStream << destinationPath;
61.    personFile.close();
62.
63.    CatalogService::sortCatalog(false, "0");

```

```
64.     return 0;  
65. }  
66.
```

B. Πίνακας Εικόνων

- Εικόνα 1: Το λογότυπο της εφαρμογής. Σελίδα 12
- Εικόνα 2: Απλή μορφή ενός νευρωνικού δικτύου. Σελίδα 14
- Εικόνα 3: Η δομή ενός συνελκτικού νευρωνικού δικτύου. Σελίδα 15
- Εικόνα 4: Τεχνικές μηχανικής μάθησης. Σελίδα 16
- Εικόνα 5: Αναπαράσταση σιαμαίων δικτύων. Σελίδα 17
- Εικόνα 6: Σύγκριση αλγορίθμων εκμάθησης. Σελίδα 18
- Εικόνα 7: Παράδειγμα εικόνας μετά την χρήση αλγορίθμων υπολογιστικής όρασης. Σελίδα 21
- Εικόνα 8: Η διεπαφή χρήστη της Clearview AI. Σελίδα 25
- Εικόνα 9: Η αρχική οθόνη της FaceSearch. Σελίδα 26
- Εικόνα 10: Η λειτουργία αναγνώρισης προσώπου. Σελίδα 27
- Εικόνα 11: Παράθυρο προβολής πληροφοριών υπόπτου. Σελίδα 27
- Εικόνα 12: Το περιβάλλον της εφαρμογής IDentify. Σελίδα 28
- Εικόνα 13: Το λογότυπο του Qt Development Framework. Σελίδα 30
- Εικόνα 14: Παράδειγμα κλάσης που χρησιμοποιεί slots and signals. Σελίδα 31
- Εικόνα 15: Χρήση της QAudioOutput. Σελίδα 32
- Εικόνα 16: Χρήση της QMainWindow. Σελίδα 33
- Εικόνα 17: Χρήση της QVideoWidget. Σελίδα 33
- Εικόνα 18: Χρήση μακροεντολών από την κλάση QMediaPlayer. Σελίδα 34
- Εικόνα 19: Δημιουργία προσαρμοσμένου παραθύρου QMessageBox. Σελίδα 34
- Εικόνα 20: Χρήση της QMediaMetaData. Σελίδα 35
- Εικόνα 21: Φόρτωση ρυθμίσεων με την χρήση της QSettings. Σελίδα 35
- Εικόνα 22: Άνοιγμα διαλόγου επιλογής αρχείου με την QFileDialog. Σελίδα 35
- Εικόνα 23: Προεπισκόπηση διαλόγου που περιέχει αντικείμενο QProgressBar. Σελίδα 36
- Εικόνα 24: Χρήση της QPainter. Σελίδα 36
- Εικόνα 25: Παράδειγμα χρήσης της QDir. Σελίδα 37
- Εικόνα 26: Άνοιγμα αρχείου με την QFile. Σελίδα 37
- Εικόνα 27: Παράδειγμα χρήσης της QStringList σε ανάγνωση αρχείου. Σελίδα 37
- Εικόνα 28: Παράδειγμα χρήσης της QTextStream. Σελίδα 38
- Εικόνα 29: Σύνδεση ενός QPushButton με μια μέθοδο. Σελίδα 38
- Εικόνα 30: Χρήση QSlider στο video player της εφαρμογής. Σελίδα 39
- Εικόνα 31: Widget προβολής λίστας αρχείων. Σελίδα 39
- Εικόνα 32: Χρήση της QModelIndex. Σελίδα 39
- Εικόνα 33: Επαναφορά εικόνας με την QResizeEvent. Σελίδα 40
- Εικόνα 34: Το ημερολόγιο εκτέλεσης. Σελίδα 40
- Εικόνα 35: Η εικόνα και το κείμενο είναι γραφικά στοιχεία QLabel. Σελίδα 41
- Εικόνα 36: Ένα γραφικό στοιχείο QDateEdit. Σελίδα 41
- Εικόνα 37: Συνάρτηση ελέγχου του ID. Σελίδα 42
- Εικόνα 38: Ένα γραφικό στοιχείο QLineEdit. Σελίδα 42
- Εικόνα 39: Ένα γραφικό στοιχείο QComboBox. Σελίδα 42
- Εικόνα 40: Ένα γραφικό στοιχείο QTableWidget. Σελίδα 43
- Εικόνα 41: Παράδειγμα χρήσης της QFileInfo. Σελίδα 43
- Εικόνα 42: Ένα γραφικό στοιχείο QTabWidget. Σελίδα 44
- Εικόνα 43: Γραφικά στοιχεία QCheckBox. Σελίδα 44
- Εικόνα 44: Χρήση της QModelIndexList για την επιλογή συγκεκριμένης γραμμής του QTableWidget. Σελίδα 44
- Εικόνα 45: Ένα γραφικό στοιχείο QSpinBox. Σελίδα 45
- Εικόνα 46: Δημιουργία αντικειμένου για το setup του νήματος. Σελίδα 46

- Εικόνα 47: Δημιουργία αντικειμένου και εκκίνηση του νήματος. Σελίδα 46
- Εικόνα 48: Η συνάρτηση `run()` εκτελεί το `event loop` του νήματος. Σελίδα 46
- Εικόνα 49: Χρήση της `QChar` για την εμφάνιση ειδικών `Unicode` χαρακτήρων. Σελίδα 47
- Εικόνα 50: Ένα πλαίσιο `QGroupBox` με 3 στοιχεία. Σελίδα 47
- Εικόνα 51: Το λογότυπο της `C++`. Σελίδα 50
- Εικόνα 52: Χρήση μεταβλητών τύπου `std::vector`. Σελίδα 51
- Εικόνα 53: Χρήση του `std::cerr` σε περίπτωση που η γραμμή δεν έχει την σωστή μορφή. Σελίδα 51
- Εικόνα 54: Χρήση της `std::ifstream` για ανάγνωση αρχείων. Σελίδα 51
- Εικόνα 55: Η `find_if`. Σελίδα 52
- Εικόνα 56: Δημιουργία ενός νευρωνικού δικτύου εντοπισμού κίνησης. Σελίδα 54
- Εικόνα 57: Η συνάρτηση `detectObjects()`. Σελίδα 55
- Εικόνα 58: Η συνάρτηση `detectFaceRectanglesCaffe()`. Σελίδα 55
- Εικόνα 59: Δημιουργία ενός δικτύου `YuNet`. Σελίδα 56
- Εικόνα 60: Χρήση της μεθόδου `detect()`. Σελίδα 57
- Εικόνα 61: Η μέθοδος `visualizeDetectionsYunet()`. Σελίδα 57
- Εικόνα 62: Η συνάρτηση `initializeSFaceDNN()`. Σελίδα 58
- Εικόνα 63: Η συνάρτηση `imageInference()`. Σελίδα 58
- Εικόνα 64: Η συνάρτηση `matchImages()`. Σελίδα 58
- Εικόνα 65: Το λογότυπο του `Visual Studio 2022`. Σελίδα 59
- Εικόνα 66: `C/C++ Include Directories`. Σελίδα 60
- Εικόνα 67: `Linker General Options`. Σελίδα 60
- Εικόνα 68: Διάγραμμα περίπτωσης χρήσης «Ανίχνευσης Κίνησης». Σελίδα 63
- Εικόνα 69: Διάγραμμα περίπτωσης χρήσης «Εισαγωγή Δεδομένων Υπόπτου στη Βάση». Σελίδα 64
- Εικόνα: Διάγραμμα αρχιτεκτονικής συστήματος. Σελίδα 69
- Εικόνα 70: Αποψη του κεντρικού παραθύρου. Σελίδα 71
- Εικόνα 71: `Video player`. Σελίδα 71
- Εικόνα 72: Εργαλείο προεπισκόπησης εικόνων. Σελίδα 72
- Εικόνα 73: Ημερολόγιο γεγονότων. Σελίδα 73
- Εικόνα 74: Γραμμή εργαλείων. Σελίδα 73
- Εικόνα 75: Εμφάνιση μεταδεδωμένων στο ημερολόγιο γεγονότων. Σελίδα 74
- Εικόνα 76: Το παράθυρο ανίχνευσης κίνησης. Σελίδα 74
- Εικόνα 77: Αρχείο κειμένου αποτελεσμάτων. Σελίδα 75
- Εικόνα 78: Στιγμιότυπο φωτογραφικού αποτελέσματος. Σελίδα 75
- Εικόνα 79: Παράθυρο ενημέρωσης προόδου. Σελίδα 76
- Εικόνα 80: Παράθυρο λειτουργίας εντοπισμού προσώπων. Σελίδα 76
- Εικόνα 81: Στιγμιότυπο από την εξαγωγή βίντεο. Σελίδα 77
- Εικόνα 82: Παράθυρο ενημέρωσης προόδου (εντοπισμός προσώπων). Σελίδα 77
- Εικόνα 83: Το παράθυρο αναγνώρισης προσώπων. Σελίδα 77
- Εικόνα 84: Αποθήκευση αναγνώρισεων σε αρχείο κειμένου. Σελίδα 78
- Εικόνα 85: Παράδειγμα εμφάνισης επιτυχημένης αναγνώρισης. Σελίδα 78
- Εικόνα 86: Το μενού «Αρχείο». Σελίδα 78
- Εικόνα 87: Το μενού «Εργαλεία». Σελίδα 79
- Εικόνα 88: Ο κατάλογος ατόμων. Σελίδα 79
- Εικόνα 89: Το παράθυρο προσθήκης ατόμων. Σελίδα 80
- Εικόνα 90: Η καρτέλα «Επιλογή Ατόμου». Σελίδα 81
- Εικόνα 91: Η καρτέλα «Προσθήκη». Σελίδα 81
- Εικόνα 92: Η καρτέλα «Καταγραφή». Σελίδα 82
- Εικόνα 93: Το μενού «Ρυθμίσεις». Σελίδα 82
- Εικόνα 94: Το παράθυρο «Ρυθμίσεις Αρχείου Ονομάτων». Σελίδα 83
- Εικόνα 95: Το παράθυρο «Ρυθμίσεις Αρχείου Βαρών». Σελίδα 83
- Εικόνα 96: Το παράθυρο «Ρυθμίσεις Αρχείου Παραμετροποιήσεων». Σελίδα 84
- Εικόνα 97: Το παράθυρο «Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων». Σελίδα 84

Εικόνα 98: Το παράθυρο «Ρυθμίσεις Μοντέλου Εντοπισμού Προσώπων». Σελίδα 85

Εικόνα 99: Το παράθυρο «Ρυθμίσεις Αποθήκευσης». Σελίδα 85

Εικόνα 100: Το μενού «Βοήθεια». Σελίδα 86

Εικόνα 101: Το παράθυρο «Σχετικά με το OpenCV». Σελίδα 86

Εικόνα 102: Το παράθυρο «Σχετικά με το Argus». Σελίδα 86

Εικόνα 103: Το παράθυρο «Επικοινωνία». Σελίδα 86

Γ. Βιβλιογραφία

MIT – Explained: Neural Networks

<https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>

Wikipedia – Neural Network (Machine Learning)

[https://en.wikipedia.org/wiki/Neural_network_\(machine_learning\)#](https://en.wikipedia.org/wiki/Neural_network_(machine_learning)#)

IBM – What is Machine Learning (ML)?

<https://www.ibm.com/topics/machine-learning>

Google – Google Maps 101: How AI helps predict traffic and determine routes.

<https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/>

Wikipedia – Computer Vision

https://en.wikipedia.org/wiki/Computer_vision

Wikipedia – Facial Recognition System

https://en.wikipedia.org/wiki/Facial_recognition_system

OpenCV – Face Recognition with OpenCV

https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html

Clearview AI – Overview

<https://www.clearview.ai/overview>

Jeffrey Jay Blatt – Some Observations on the Clearview AI Facial Recognition System from Someone Who Has Actually Used It...

<https://www.linkedin.com/pulse/some-observations-clearview-ai-facial-recognition-system-blatt/>

Clearview AI – How we store and search 30 billion faces.

<https://www.clearview.ai/post/how-we-store-and-search-30-billion-faces>

Vigilant Solutions – Facial Recognition

<https://www.ra-comm.com/vigilant-solutions/facial-recognition.htm>

Veritone – IDentify: An Overview

https://support.veritone.com/s/article/000001551?language=en_US

Qt – Qt Documentation Pages

<https://doc.qt.io/>

Medium - Building a face detector with OpenCV in C++.

<https://medium.com/analytics-vidhya/building-a-face-detector-with-opencv-in-c-8814cd374ea1>

OpenCV – OpenCV Documentation Pages

<https://docs.opencv.org/4.x/>

Wikipedia – Agile software development.

https://en.wikipedia.org/wiki/Agile_software_development

Github – YuNet face detection repository.

https://github.com/opencv/opencv_zoo/tree/main/models/face_detection_yunet

Github – SFace face recognition repository.

https://github.com/opencv/opencv_zoo/tree/main/models/face_recognition_sface

DevDocs – C++ Documentation Pages

<https://devdocs.io/cpp/>

AutomaticAddison – How to add an external C++ library to your project.

<https://automaticaddison.com/how-to-add-an-external-c-library-to-your-project/>

Qt – Threads

<https://doc.qt.io/qt-6/threads.html>

Microsoft – Visual Studio 2022

<https://visualstudio.microsoft.com/vs/>

Github – Argus Repository

<https://github.com/Felfired/Argus>

The Noun Project – Icons

<https://thenounproject.com/>

[1] Wu, W., Peng, H. & Yu, S. YuNet: A Tiny Millisecond-level Face Detector. Mach. Intell. Res. 20, 656–665 (2023). <https://doi.org/10.1007/s11633-023-1423-y>

[2] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li and D. Wen, "SFace: Sigmoid-Constrained Hypersphere Loss for Robust Face Recognition," in IEEE Transactions on Image Processing, vol. 30, pp. 2587-2598, 2021, doi: 10.1109/TIP.2020.3048632.

Bishop CM (17 August 2006). Pattern Recognition and Machine Learning. New York: Springer. ISBN 978-0-387-31073-2

Venkatesan, Ragav; Li, Baoxin (2017-10-23). Convolutional Neural Networks in Visual Computing: A Concise Guide. CRC Press. ISBN 978-1-351-65032-8.