



Πανεπιστήμιο Δυτικής Αττικής

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Διπλωματική εργασία

Αναγνώριση συναισθήματος στην αλληλεπίδραση ανθρώπου-ρομπότ με
την χρήση του NAO

Φοιτήτρια: Βαλαγκούτη Ηρώ-Αθηνά

ΑΜ: 161082

Επιβλέποντες

Καθηγήτρια Σγουροπούλου Κλειώ

Δρ. Τρούσσας Χρήστος

Αθήνα, Αιγάλεω

Ιούλιος 2021



University of West Attica

Faculty of Engineering

Department of Informatics and Computer Engineering

Diploma Thesis

Emotion recognition for human-robot interaction with NAO robot

Student: Valagkouti Iro – Athina

Registration Number: 161082

Supervisors

Prof. Sgouropoulou Cleo

Dr. Troussas Christos

Athens, Egaleo

July 2021

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Καθηγήτρια Σγουροπούλου Κλειώ	Δρ. Τρούσσας Χρήστος	Δρ. Φειδάκης Μιχαήλ
----------------------------------	----------------------	---------------------

Copyright ©

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ Βαλαγκούτη Ηρώ – Αθηνά, Ιούλιος, 2021

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Βαλαγκούτη Ηρώ-Αθηνά του Αντωνίου, με αριθμό μητρώου 161082, φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

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

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους συνέβαλλαν και βοήθησαν με οποιονδήποτε τρόπο στην εκπόνηση αυτής της διπλωματικής εργασίας.

Πρέπει να ευχαριστήσω θερμά την καθηγήτρια κα. Σγουροπούλου Κλειώ και τον Δρ. Τρούσσα Χρήστο για την επίβλεψη αυτής της εργασίας, για την καθοδήγησή τους, την υπομονή και το χρόνο που διέθεσαν. Παράλληλα θα ήθελα να ευχαριστήσω τον Δρ. Φειδάκη Μιχαήλ, τόσο για τον χρόνο και τις συμβουλές που διέθεσε σχετικά με τον χειρισμό του ΝΑΟ αλλά και για την ευκαιρία που είχα να χρησιμοποιήσω το ρομπότ στο εργαστήριο CoNSert, παρ' όλες τις δύσκολες συνθήκες εξαιτίας των περιοριστικών μέτρων του Covid-19.

Ακόμη, θα ήθελα να ευχαριστήσω το εργαστήριο CoNSert και όλα τα μέλη του που συμμετείχαν σε διαδικασίες δοκιμής, καθώς και για το υλικό που πρόσφεραν και τον χρόνο τους.

Τέλος, θα ήθελα να ευχαριστήσω βαθύτατα την οικογένειά μου, τους φίλους μου και τους συναδέλφους μου, που καθ' όλη την διάρκεια της εκπόνησης της εργασίας έδειξαν κατανόηση και πρόσφεραν την υποστήριξή τους.

ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία μελετήθηκε η δυνατότητα σύνδεσης του ανθρωποειδούς ρομπότ NAO με ήδη υπάρχων νευρωνικό δίκτυο. Ο τελικός στόχος ήταν η αναγνώριση των έξι βασικών συναισθημάτων σε ανθρώπους: χαρά, φόβος, λύπη, έκπληξη, αηδία και θυμός, όπως επίσης και η ουδετερότητα, με σκοπό να χρησιμοποιηθεί σε όλων των ειδών εφαρμογές, για άτομα που χρήζουν βοήθεια, για μαθητές ή για ψυχαγωγία. Εκτός από την χρήση του NAO, η μεθοδολογία που ακολουθήθηκε χρησιμοποίησε τεχνολογίες όπως sockets, νευρωνικά δίκτυα και τα διαθέσιμα modules του NAO, προκειμένου να κατασκευαστεί μία αρχιτεκτονική τύπου client-server που θα επιτρέψει αποστολή δεδομένων από και προς το ρομπότ. Το τελικό πρόγραμμα που κατασκευάστηκε περιλαμβάνει συζήτηση με τον άνθρωπο, λήψη της φωτογραφίας του και ανάλυση του συναισθήματός του με χρήση του DeepFace API. Τέλος, για την λήψη δεδομένων υπό κανονικές συνθήκες και μετέπειτα αξιολόγηση της αποτελεσματικότητας της εργασίας, πραγματοποιήθηκε παράδειγμα εφαρμογής του λογισμικού σε ένα φεστιβάλ. Συγκεκριμένα, επιχειρήθηκε η αναγνώριση των συναισθημάτων των παρευρισκόμενων μετά από την αλληλεπίδρασή τους με το ρομπότ με ικανοποιητικά αποτελέσματα.

Λέξεις-κλειδιά: ρομπότ NAO, νευρωνικά δίκτυα, αναγνώριση συναισθήματος, DeepFace, sockets

ABSTRACT

In the present diploma thesis, the possibility of connecting the humanoid robot NAO with an already existing neural network was studied. The ultimate goal was to identify the six basic emotions in people: joy, fear, sadness, surprise, disgust and anger, as well as neutrality, to be used in all kinds of applications, for people in need of medical care, for students or for entertainment. In addition to using NAO, the methodology used technologies such as sockets, neural networks and available NAO modules to build a client-server architecture that would allow data to be sent to and from the robot. The final program that was created consists of a discussion with a human, taking his photo and analyzing his emotion using the DeepFace API. Finally, in order to obtain data under real-life conditions and subsequently evaluate the effectiveness of the project, an example of software application was performed at a festival. Specifically, an attempt was made to identify the emotions of the attendees after their interaction with the robot. The results were satisfactory.

Keywords: NAO robot, neural networks, emotion recognition, DeepFace, sockets

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	6
ABSTRACT.....	7
ΠΕΡΙΕΧΟΜΕΝΑ.....	8
Κατάλογος Εικόνων.....	9
Ευρετήριο Ξενόγλωσσων Όρων.....	10
ΕΙΣΑΓΩΓΗ.....	13
1. ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ.....	15
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....	18
2.1 Τεχνητή νοημοσύνη - Μηχανική μάθηση.....	18
2.1.1 Βαθιά Μάθηση (Deep Learning).....	18
2.1.2 Τεχνητός Νευρώνας (Artificial Neuron).....	19
2.1.3 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks).....	20
2.1.4 Συνελκτικά νευρωνικά δίκτυα (Convolution Neural Networks).....	21
2.2 Τεχνολογίες και περιβάλλοντα ανάπτυξης.....	24
2.3 Ανθρωποειδές ΝΑΟ.....	28
3. ΣΧΕΔΙΑΣΜΟΣ & ΥΛΟΠΟΙΗΣΗ.....	32
3.1 Αρχιτεκτονική.....	32
3.2 Υλοποίηση.....	34
3.3 Παράδειγμα λογισμικού.....	46
4. ΑΞΙΟΛΟΓΗΣΗ.....	48
5. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	50
5.1 Επίλογος.....	50
5.2 Περιορισμοί.....	51
5.3 Μελλοντικές επεκτάσεις.....	52
ΠΑΡΑΠΟΜΠΕΣ.....	55

Κατάλογος Εικόνων

Εικόνα 1: Βιολογικός νευρώνας [1].....	19
Εικόνα 2: Βασικό μοντέλο τεχνητού νευρώνα	20
Εικόνα 3: Τεχνητό Νευρωνικό Δίκτυο πρόσθιας τροφοδότησης [1]	21
Εικόνα 4: Συνέλιξη μίας 8x8 εικόνας με ένα 3x3 φίλτρο [2]	23
Εικόνα 5: Εξαγωγή χαρακτηριστικών με συνέλιξη [5]	23
Εικόνα 6: Max pooling [2].....	24
Εικόνα 7: Το περιβάλλον ανάπτυξης εφαρμογών Choregraphe	27
Εικόνα 8: NAO.....	28
Εικόνα 9: Διαστάσεις του NAO [7]	28
Εικόνα 10: Η διαδικασία NAOqi [11].....	29
Εικόνα 11: Θέσεις των καμερών στο NAO [13].....	30
Εικόνα 12: Διάγραμμα ροής συστήματος	33
Εικόνα 13: Διάγραμμα ροής εισερχομένων δεδομένων.....	33
Εικόνα 14: Διάγραμμα ροής εξερχόμενων δεδομένων	33
Εικόνα 15: Δομή νευρωνικού δικτύου DeepFace	34
Εικόνα 16: Κατασκευή του νευρωνικού δικτύου [16]	34
Εικόνα 17: Πρώτο επίπεδο νευρωνικού δικτύου [16]	35
Εικόνα 18: Δεύτερο επίπεδο νευρωνικού δικτύου [16]	35
Εικόνα 19: Τρίτο επίπεδο νευρωνικού δικτύου [16]	36
Εικόνα 20: Μετατροπή σε μονοδιάστατο πίνακα [16]	36
Εικόνα 21: Πλήρης σύνδεση του νευρωνικού δικτύου [16]	36
Εικόνα 22: Classification layer [16].....	36
Εικόνα 23: Ένα Python box στο Choregraphe	37
Εικόνα 24: Τα τέσσερα βασικά κουτιά στο Choregraphe	37
Εικόνα 25: Είσοδοι/έξοδοι κουτιού. Το μαύρο αντιστοιχεί στο bang, το κίτρινο στο number, το γαλάζιο στο string και το γκρι στο dynamic	38
Εικόνα 26: Δομή αρχείου συμπεριφοράς στο Choregraphe	38
Εικόνα 27: Θέμα (topic) και γλώσσα (language).....	39
Εικόνα 28: User rule	39
Εικόνα 29: User rules που βασίζονται στο συναίσθημα.....	40
Εικόνα 30: Σύνδεση με τον server	40
Εικόνα 31: ALVideoDevice service.....	41
Εικόνα 32: Λήψη της εικόνας.....	41
Εικόνα 33: Αποστολή εικόνας και λήψη απάντησης	42
Εικόνα 34: Χρήση μεταβλητών του κουτιού για την προώθηση του αποτελέσματος	42
Εικόνα 35: Δημιουργία socket.....	43
Εικόνα 36: Σύνδεση με τον client	43
Εικόνα 37: Λήψη δεδομένων	44
Εικόνα 38: Ανακατασκευή εικόνας	44
Εικόνα 39: Αναγνώριση συναισθήματος	45
Εικόνα 40: Ανάλυση συναισθήματος στο DeepFace	45
Εικόνα 41: Αποστολή δεδομένων στο NAO	45
Εικόνα 42α) Χαρά	46
Εικόνα 42β) Αποτέλεσμα χαράς από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα	46

Εικόνα 43α) Ουδετερότητα.....	46
Εικόνα 43β) Αποτέλεσμα ουδετερότητας από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα	46
Εικόνα 44α) Λύπη.....	47
Εικόνα 44β) Αποτέλεσμα λύπης από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα	47
Εικόνα 45α) Θυμός.....	47
Εικόνα 45β) Αποτέλεσμα θυμού από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα	47

Ευρετήριο Ξενόγλωσσων Όρων

Software modules: δομοστοιχεία λογισμικού

Documentation: τεκμηρίωση

Module: ενότητα

Deep Learning: Βαθιά Μάθηση

Supervised Learning: μάθηση με επίβλεψη

Unsupervised Learning: μάθηση χωρίς επίβλεψη

Layers: επίπεδα

Artificial Neuron: Τεχνητός Νευρώνας

Sum: αθροιστής

Activation function: συνάρτηση ενεργοποίησης

Weights: βάρη

Step function: βηματική συνάρτηση

Sign function: συνάρτηση πρόσημου

Logistics function: λογιστική συνάρτηση

Summation function: συνάρτηση αθροίσματος

Threshold value: τιμή ενεργοποίησης

Artificial Neural Networks: Τεχνητά Νευρωνικά Δίκτυα

Hidden layers: κρυφά επίπεδα

Output layer: επίπεδο εξόδου

Fully connected: πλήρως συνδεδεμένοι

Partially connected: μερικώς συνδεδεμένοι

Feedforward: δίκτυο πρόσθιας τροφοδότησης

Backpropagation: οπισθοδιάδοση

Convolution Neural Networks: Συνελκτικά νευρωνικά δίκτυα

Patterns: χαρακτηριστικά

Convolutional layers: επίπεδα συνέλιξης

Pixels: εικονοστοιχεία

Pooling: επίπεδα συγκέντρωσης

Data flow graphs: γραφήματα ροής δεδομένων

Tensors: τανυστές

Application programming interface, API: διεπαφή προγραμματισμού εφαρμογών

Framework: πλαίσιο λογισμικού

Client: πελάτης

Server: εξυπηρετητής

Socket: υποδοχή

Handshake: διαδικασία χειραψίας

Boxes: κουτιά

User rules: κανόνες χρήστη

Proposal rules: προτάσεις

Topics: θέματα

Python boxes: κουτιά python

Basic emotions: βασικές συγκινήσεις

Strides: βηματισμός

Feature map: χαρακτηριστικά της εικόνας

Annotations: σχόλια

Classification layer: επίπεδο κατηγοριοποίησης

Diagram: διάγραμμα

Timeline: χρονοδιάγραμμα

Dialog: διάλογος

Bang: εκπυρσοκρότηση

Number: αριθμός

String: συμβολοσειρά

Dynamic: δυναμικός

Language: γλώσσα

Event: γεγονός

Subrules: υπο-κανόνες

Happy: χαρά

Sad: λύπη

Neutral: ουδετερότητα

Angry: θυμός

Fear: φόβος

Disgust: αηδία

Surprise: έκπληξη

Subscribe: εγγραφή

Logger: αρχεία καταγραφής συμβάντων

Predict: προβλέψεις

Bandwidth: εύρος ζώνης

Datasets: δεδομένα

Adaptive-E learning systems: προσαρμοστικά συστήματα ηλεκτρονικής μάθησης

ΕΙΣΑΓΩΓΗ

Σκοπός αυτής της εργασίας είναι να ερευνήσει την δυνατότητα ανάλυσης εκφράσεων του προσώπου κατά την διάρκεια αλληλεπίδρασης ενός ανθρώπου με ένα ανθρωποειδές ρομπότ. Η δυνατότητα αυτή είναι πολύ σημαντική καθώς μπορεί να εφαρμοστεί σε πολλούς τομείς, όπως η ιατρική, για ανάλυση της ψυχικής κατάστασης ενός ασθενή, η εκπαίδευση, για χρήση ως εργαλείο κατά την διάρκεια της διδασκαλίας ή ακόμα και η ψυχαγωγία, ως μέσο αντίληψης των συναισθημάτων του κοινού. Η ανθρωπόμορφη και φιλική όψη του το καθιστά ικανό να χρησιμοποιηθεί σε κάθε ηλικιακή ομάδα.

Συγκεκριμένα, όπως φαίνεται από άλλα δημοσιεύματα (Towards a Reference Model to Ensure the Quality of Massive Open Online Courses and E-Learning, Troussas C. et al.) [22], (Dynamic Detection of Learning Modalities Using Fuzzy Logic in Students' Interaction Activities, Troussas C. et al.) [23], (Fuzzy Logic for Refining the Evaluation of Learners' Performance in Online Engineering Education, Krouska A.) [25], στον τομέα της εκπαίδευσης μπορεί να χρησιμοποιηθεί τόσο ως σύστημα αξιολόγησης των μαθητών (Intelligent and adaptive tutoring through a social network for higher education, Troussas C. et al.) [21] όσο και για την οργάνωσή τους σε ομάδες, αλλά και για την προσωπική οργάνωση του μαθήματος ανάλογα με τις ξεχωριστές ανάγκες που παρουσιάζει ο κάθε μαθητής, καθώς τα συστήματα αυτά βασίζονται και στην ψυχολογική ανάλυση του μαθητή. Δεν είναι αναγκαίο όμως να περιοριστεί σε εφαρμογές για σχολεία καθώς μπορεί να χρησιμοποιηθεί και για αξιολόγηση σε άλλους τομείς, όπως η εκπαίδευση των πυροσβεστών (Measuring User Experience, Usability and Interactivity of a Personalized Mobile Augmented Reality Training System, Papakostas C.) [24] ή στην μηχανική (User acceptance of augmented reality welding simulator in engineering training,], Papakostas, C. et al) [26]

Η εργασία υλοποιήθηκε με χρήση του ρομπότ NAO, το οποίο αν και υποστηρίζει ήδη δικό του μοντέλο ανάλυσης συναισθημάτων δεν χρησιμοποιήθηκε, καθώς κρίθηκε σκόπιμο να γίνει χρήση άλλου μοντέλου ώστε να ερευνηθούν οι δυνατότητες σύνδεσής του με δομοστοιχεία λογισμικού (software modules) σχεδιασμένα από τρίτους. Ένας ακόμα παράγοντας για την απόφαση αυτή ήταν ότι το πρόγραμμα

προσομοίωσης δεν υποστηρίζει την ήδη υπάρχουσα ανάλυση συναισθήματος του ρομπότ. Για την υλοποίηση χρησιμοποιήθηκαν sockets, βιβλιοθήκες όπως οι TensorFlow, Keras και CUDA για την χρήση της κάρτας γραφικών κατά την διάρκεια των υπολογισμών και την υποστήριξη του νευρωνικού δικτύου αλλά και του DeepFace για την χρήση του ίδιου του νευρωνικού δικτύου και των έτοιμων εκπαιδευμένων μοντέλων για την αναγνώριση συναισθήματος. Από το ρομπότ NAO χρησιμοποιήθηκαν modules όπως το ALVideoDevice για την λήψη φωτογραφιών και το ALDialog για την επικοινωνία με τον χρήστη. Η εφαρμογή δοκιμάστηκε στο EcoFest 2021 που έλαβε χώρα στην Αθήνα, ζητώντας από τους παρευρισκόμενους να κάνουν μία αξιολόγηση κάνοντας εκφράσεις προσώπου. Το μεγαλύτερο ποσοστό αναγνωρίστηκε με επιτυχία.

Η εργασία χωρίζεται σε πέντε κεφάλαια. Το πρώτο αφορά την ανασκόπηση βιβλιογραφίας όπου παρουσιάζονται αναλυτικά οι βιβλιογραφικές πηγές της εργασίας και αξιολογούνται, το δεύτερο αφορά το θεωρητικό υπόβαθρο όπου αναφέρονται βασικές έννοιες και τεχνολογίες που χρησιμοποιήθηκαν, το τρίτο τον σχεδιασμό και την υλοποίηση όπου παρουσιάζεται η δομή και ο κώδικας της εργασίας, το τέταρτο την αξιολόγηση και το πέμπτο μελλοντικές επεκτάσεις και συμπεράσματα.

1. ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ

Στο παρόν κεφάλαιο παρουσιάζεται η βιβλιογραφική ανασκόπηση της εργασίας.

Από αναζήτηση που πραγματοποιήθηκε σε μηχανές αναζήτησης, προέκυψαν 23 πιθανές πηγές που περιλαμβάνουν έρευνες, άλλες διπλωματικές εργασίες, άρθρα, διαφάνειες και βίντεο οι οποίες επιλέχθηκαν βάση του τίτλου, από ανάγνωση της περίληψης ή από την συνάφεια που παρουσίασαν λέξεις-κλειδιά. Οι 5 απορρίφθηκαν λόγω μη επαρκής συνάφειας ή αδυναμία πρόσβασης στο πλήρες κείμενο. Από τις υπόλοιπες 16, οι 8 χρησιμοποιήθηκαν για άντληση γενικών γνώσεων και δεν αναφέρονται στην βιβλιογραφία. Συνολικά, στην βιβλιογραφία παρουσιάζονται μία διπλωματική εργασία στην οποία βασίστηκε κυρίως και η παρούσα εργασία, τέσσερα άρθρα που παρουσιάζουν την χρησιμοποιούμενη τεχνολογία και η τεκμηρίωση (documentation) του NAO από την SoftBank Robotics. Για την συγγραφή της διπλωματικής εργασίας λήφθηκαν υπόψη και τέσσερα βιβλία, από τα οποία τα τρία κρίθηκαν ως επιλέξιμα από το περιεχόμενό τους για άντληση γενικών πληροφοριών, τα οποία αναφέρονται και στην βιβλιογραφία. Ακόμη, γίνονται και δύο αναφορές και σε διαφάνειες. Η μία αφορά ένα διδασκόμενο γνωστικό αντικείμενο του πανεπιστημίου, ενώ η άλλη διαφάνειες που παρέχονται από την κατασκευάστρια εταιρεία και συγγραφέας τους είναι προσωπικό της εταιρείας.

Ένα μεγάλο κομμάτι της βιβλιογραφίας αφορά δημοσιεύματα που πραγματεύονται γενικά θέματα και υλοποιήσεις που προκύπτουν από την χρήση των νευρωνικών δικτύων και του τομέα της τεχνητής νοημοσύνης με εφαρμογές κυρίως στην εκπαίδευση. Αυτά χρησιμοποιήθηκαν ως κατευθυντήριες ιδέες σχετικά με τις εφαρμογές του NAO στην εκπαίδευση.

Το γενικότερο συμπέρασμα που προκύπτει από την μελέτη της βιβλιογραφίας είναι ότι αν και οι τεχνολογίες και τα περιβάλλοντα ανάπτυξης που χρησιμοποιούνται απαντώνται στις περισσότερες περιπτώσεις (Keras και Tensorflow), τα νευρωνικά δίκτυα χρησιμοποιούνται για πληθώρα εφαρμογών και υπάρχουν πολλά μοντέλα και τεχνικές που μπορούν να ακολουθηθούν. Η ανίχνευση συναισθημάτων είναι αρκετά δημοφιλής τομέας και τα θέματα των ερευνών που αφορούν το ρομπότ καλύπτουν ένα ευρύ πεδίο, όπως την ανίχνευση των συναισθημάτων του χρήστη για παροχή

βοήθειας, αλλά και την εκμάθηση των συναισθημάτων σε παιδιά με αυτισμό, όπου το ίδιο το ρομπότ καλείται να μιμηθεί ανθρώπινα συναισθήματα. Και τα δύο είδη ερευνών αποτέλεσαν πηγή έμπνευσης για την παρούσα εργασία.

Κυριότερη πηγή πληροφοριών για το θεωρητικό υπόβαθρο αποτέλεσαν δύο από τα μαθήματα του πενταετούς κύκλου σπουδών του τμήματος του πανεπιστημίου, με τίτλους [Τεχνητή Νοημοσύνη \(ICE-5004\)](#) και [Όραση Υπολογιστών \(ICE-8111\)](#). Το βιβλίο που αναφέρεται στην βιβλιογραφία με τίτλο «[Τεχνητή νοημοσύνη](#)» έχει ως συγγραφέα διδακτικό ερευνητικό προσωπικό του τμήματος, ενώ τα υπόλοιπα που χρησιμοποιήθηκαν, πλην του «[Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python](#)» αποτελούσαν προτεινόμενα συγγράμματα για τα δύο μαθήματα. Οι διαφάνειες του μαθήματος της Όρασης υπολογιστών αναφέρονται επίσης ως πηγές στην βιβλιογραφία, καθώς χρησιμοποιήθηκαν τόσο για άντληση πληροφοριών όσο και για τις εικόνες τους. Τα βιβλία αυτά και οι διαφάνειες των μαθημάτων κρίθηκαν σκόπιμο να χρησιμοποιηθούν ως πηγές στην εργασία καθώς προσφέρουν μία κατανοητή και πλήρης παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν, σε ικανοποιητικό βαθμό που εξυπηρετούσε τον σκοπό της εργασίας.

Η τεκμηρίωση (documentation) του NAO αποτελεί το εγχειρίδιο χρήσης του ρομπότ και αναφέρεται τόσο σε γενικά όσο και σε ειδικά θέματα. Οι γενικές πληροφορίες που παρέχονται από την κατασκευάστρια εταιρεία αφορούν βασικές λειτουργίες και δυνατότητες του ρομπότ ενώ οι ειδικές είναι χωρισμένες σε κατηγορίες ανάλογα με την ενότητα (module) στην οποία ανήκουν. Περιλαμβάνει επεξήγηση των διαθέσιμων συναρτήσεων και άλλα τεχνικά χαρακτηριστικά ή πληροφορίες, μερικές φορές ακόμα και παραδείγματα τα οποία μπορεί να κατεβάσει ο ενδιαφερόμενος και να τα χρησιμοποιήσει ή να τα τροποποιήσει. Στην βιβλιογραφία της εργασίας αναφέρονται κυρίως τα modules που χρησιμοποιήθηκαν αλλά και ένα σετ διαφανειών που παρέχεται από την εταιρεία συνοδευτικά με ένα βίντεο για την επεξήγηση λειτουργιών όπως το qiChat και η διασύνδεση της γλώσσας Python με το περιβάλλον ανάπτυξης προγραμμάτων του ρομπότ, Choregraphe. Η αναφορά της τεκμηρίωσης στην βιβλιογραφία γίνεται διότι αποτέλεσε την σημαντικότερη πηγή πληροφοριών σχετικά με το ρομπότ, τις λειτουργίες του και πως αυτές μπορούν να εξατομικευθούν ώστε να επιτευχθεί το τελικό αποτέλεσμα της εργασίας.

Η διπλωματική με τίτλο “Facial Emotional Recognition With Deep Learning On Pepper Robot” του [Hu Mengge](#) που αναφέρεται στην βιβλιογραφία, παρουσιάζει αρκετές ομοιότητες ως προς το θέμα της παρούσας. Ο συγγραφέας πραγματοποιεί ανάλυση συναισθημάτων με την βοήθεια του ρομπότ PEPPER, παίρνοντας 25 φωτογραφίες από τον χρήστη, κατασκευάζοντας και εκπαιδεύοντας ένα συνελκτικό νευρωνικό δίκτυο σε τοπικό υπολογιστή με δεδομένα από το Kaggle. Στην συνέχεια κατασκευάζει ένα αρχείο με τα χαρακτηριστικά το οποίο και χρησιμοποιεί για να αξιολογήσει τις εικόνες από την PEPPER. Τέλος, παρουσιάζει ένα διάγραμμα με τις τιμές των αποτελεσμάτων, δηλαδή κατά πόσο ο χρήστης είναι χαρούμενος, νευριασμένος, φοβισμένος και άλλα. [4]

Η απόφαση να συμπεριληφθεί αυτή η εργασία κρίθηκε σκόπιμη επειδή αποτέλεσε σημαντική πηγή έμπνευσης, χρησιμοποιεί ίδιες ή παρόμοιες τεχνολογίες, η θεματολογία της προσεγγίζει σε πολύ κοντινό βαθμό το θέμα της παρούσας και τμήματα της υλοποίησής της παρουσιάζουν σημαντικές ομοιότητες με την τρέχουσα εργασία.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Στο παρόν κεφάλαιο παρουσιάζονται και αναλύονται κάποιες βασικές έννοιες, τεχνολογίες και περιβάλλοντα ανάπτυξης που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας ως προς την λειτουργία τους, καθώς και το ανθρωποειδές ρομπότ ΝΑΟ.

2.1 Τεχνητή νοημοσύνη - Μηχανική μάθηση

«Η Τεχνητή νοημοσύνη είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με την νοημοσύνη στην ανθρώπινη συμπεριφορά» (Barr και Feigenbaum). Η Τεχνητή νοημοσύνη ασχολείται με την κατασκευή προγραμμάτων που είναι ικανά να μιμηθούν τις γνωστικές ικανότητες του ανθρώπου και καταλήγουν στο να επιλύουν προβλήματα όπως για παράδειγμα μία παρτίδα σκάκι, η κατανόηση φυσικής γλώσσας [3] και άλλα πιο σύνθετα όπως η επιλογή κατάλληλων αντιπάλων σε παιχνίδια (Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments, Krouska A. et al.) [28], η πρόβλεψη απόδοσης ομάδων (A deep learning classification framework for early prediction of team-based academic performance, F. Giannakas et al.) [29] ή η σύσταση συνεργατικών δραστηριοτήτων (Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM, Troussas C. et al.) [30].

2.1.1 Βαθιά Μάθηση (Deep Learning)

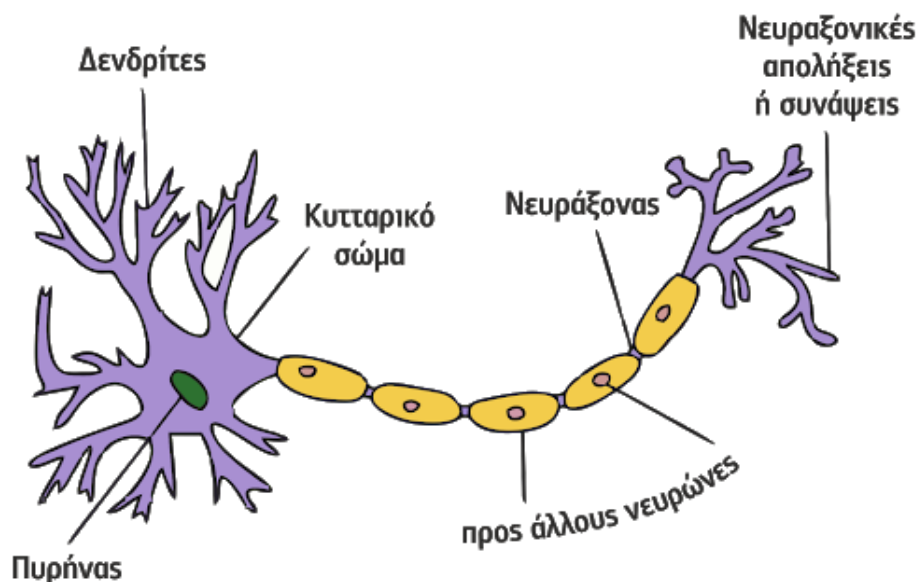
Η Μηχανική Μάθηση αποτελεί μία μέθοδο προσέγγισης της Τεχνητής Νοημοσύνης καθώς αποσκοπεί στην δημιουργία μοντέλων ή προτύπων από ένα σύνολο δεδομένων με στόχο «την απόκτηση νέας γνώσης ή νέων τρόπων οργάνωσης της ήδη υπάρχουσας» (Carbonell 1987), μιμούμενη την ανθρώπινη ικανότητα της επαγωγής και του συσχετισμού. Η Βαθιά Μάθηση είναι μία τεχνική προσέγγισης της Μηχανικής Μάθησης. Χρησιμοποιείται ευρέως σε πολλούς τομείς, με έμφαση κυρίως στην αναγνώριση προτύπων. Υποστηρίζονται πολλές μορφές εκπαίδευσης, με τις δύο κυριότερες να είναι η μάθηση με επίβλεψη (Supervised Learning) και η μάθηση χωρίς επίβλεψη (Unsupervised Learning). Η μάθηση με επίβλεψη επιτρέπει την κατασκευή ενός μοντέλου από ένα σύνολο περιπτώσεων ή στιγμιότυπων, το οποίο ονομάζεται

σύνολο δεδομένων εκπαίδευσης και περιγράφει το συγκεκριμένο μοντέλο με σωστές τιμές, έτσι ώστε το σύστημα να είναι ικανό να μάθει έννοιες μέσα από το μοντέλο αυτό που μπορεί να χρησιμοποιήσει αργότερα, για νέα δεδομένα. Η μάθηση χωρίς επίβλεψη δεν απαιτεί σωστές τιμές που σχετίζονται με τα δεδομένα εισόδου, αλλά αφήνεται στο σύστημα η ταξινόμηση και ο συσχετισμός των δεδομένων. [3]

Το κύριο χαρακτηριστικό της βαθιάς μάθησης είναι η χρήση πολλαπλών επιπέδων (layers), τα οποία επιτρέπουν - μέσα από στάδια - την εξαγωγή χαρακτηριστικών υψηλού επιπέδου. Τα χαρακτηριστικά υψηλού επιπέδου αφορούν έννοιες αντιληπτές από τους ανθρώπους.

2.1.2 Τεχνητός Νευρώνας (Artificial Neuron)

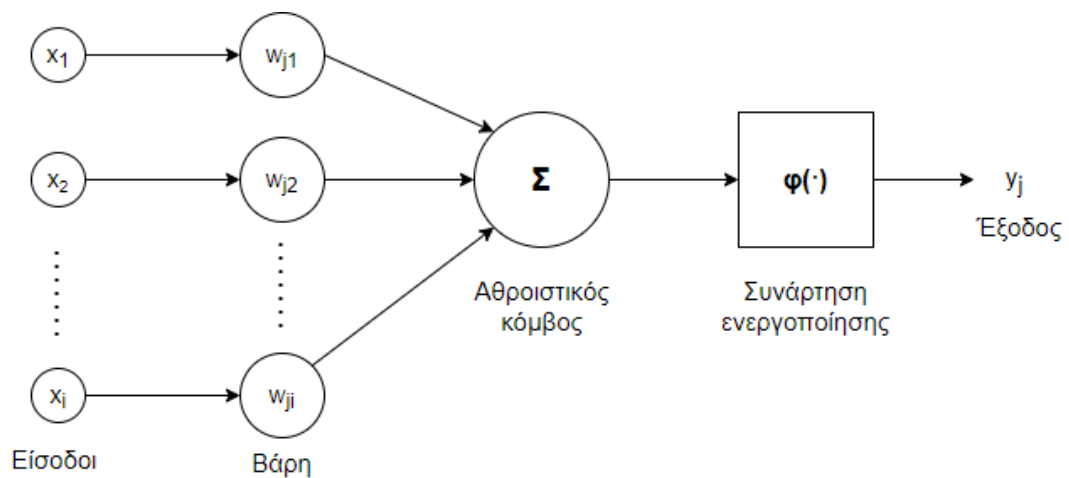
Ένας Τεχνητός Νευρώνας αντιστοιχεί στο υπολογιστικό μοντέλο του βιολογικού νευρώνα. Ο βιολογικός νευρώνας αποτελείται από το σώμα που περιλαμβάνει τον πυρήνα, τον νευράξονα που χρησιμεύει ως έξοδος και σύνδεση με τους άλλους νευρώνες και τους δενδρίτες, (διακλαδιζόμενες νευρικές αποφυάδες). Στις άκρες κάθε δενδρίτη βρίσκονται οι συνάψεις (απειροελάχιστες σχισμές) που λαμβάνουν σήματα από γειτονικούς νευρώνες.



Εικόνα 1: Βιολογικός νευρώνας [1]

Ο τεχνητός νευρώνας χωρίζεται σε δύο μέρη. Το πρώτο καλείται αθροιστής (sum) και το δεύτερο συνάρτηση ενεργοποίησης (activation function). Ένας τεχνητός νευρώνας

δέχεται σήματα εισόδου, τα οποία και αθροίζει, δημιουργώντας μία έξοδο που θα μεταφέρει τις πληροφορίες στους άλλους νευρώνες. Η έξοδος είναι συνδεδεμένη με άλλους νευρώνες όπως και στον βιολογικό, με συνάψεις. Τα σήματα των εισόδων επηρεάζονται από κάποια εξωτερικά ερεθίσματα, τα οποία ονομάζονται βάρη (weights). Τα βάρη είναι τιμές θετικές ή αρνητικές και προστίθενται στις εισόδους. Η συνάρτηση ενεργοποίησης έχει συνήθως την μορφή της βηματικής συνάρτησης (step function), της συνάρτησης πρόσημου (sign function) ή της λογιστικής συνάρτησης (logistics function). [3]



Εικόνα 2: Βασικό μοντέλο τεχνητού νευρώνα

Σύμφωνα με το παραπάνω μοντέλο, η συνάρτηση αθροίσματος (summation function) με x_i τις εισόδους και w_i τα βάρη έχει την παρακάτω μορφή:

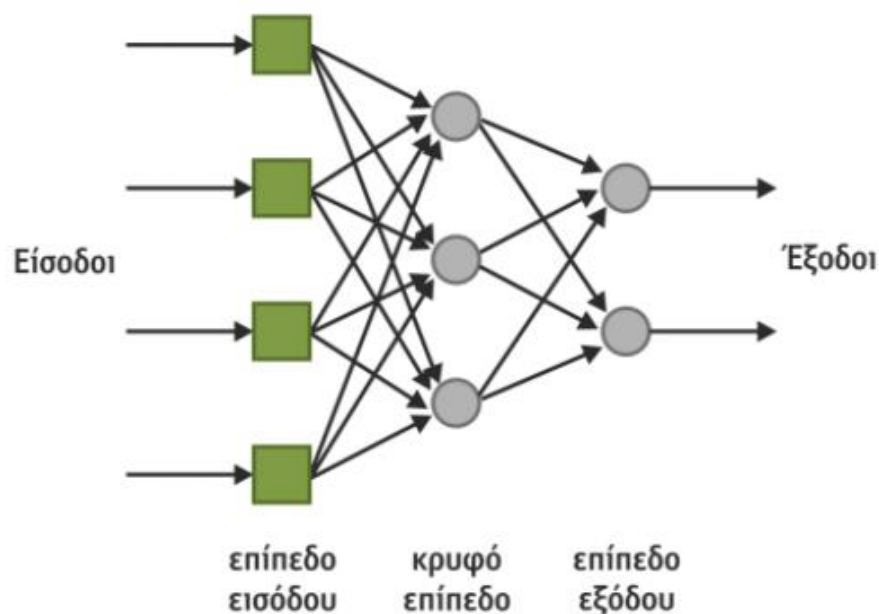
$$\Sigma = \sum_i^n x_i w_i$$

Ένας τεχνητός νευρώνας επιστρέφει αποτέλεσμα μόνο όταν η τιμή της συνάρτησης αθροίσματος είναι μεγαλύτερη από ένα ορισμένο κατώφλι ή αλλιώς, τιμή ενεργοποίησης (threshold value).

2.1.3 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)

Τα Τεχνητά Νευρωνικά Δίκτυα αποτελούνται από διασυνδεδεμένες μονάδες τεχνητών νευρώνων που επεξεργάζονται δεδομένα. Η οργάνωση των τεχνητών νευρώνων γίνεται σε επίπεδα. Το πρώτο επίπεδο χρησιμοποιείται για την εισαγωγή των δεδομένων. Προαιρετικά, μπορεί να ακολουθούν ένα ή περισσότερα κρυφά

επίπεδα (hidden layers). Το τελευταίο επίπεδο ονομάζεται επίπεδο εξόδου (output layer). Οι νευρώνες μπορούν να είναι συνδεδεμένοι με όλους τους υπόλοιπους νευρώνες. Σε τέτοιες περιπτώσεις, θεωρούνται πλήρως συνδεδεμένοι (fully connected). Σε κάθε άλλη περίπτωση, θεωρούνται μερικώς συνδεδεμένοι (partially connected). Δεν είναι υποχρεωτικό η ροή της πληροφορίας να τροφοδοτεί κάθε φορά μόνο μονάδες επόμενου επιπέδου -δίκτυο πρόσθιας τροφοδότησης (feedforward) - καθώς υπάρχει και η οπισθοδιάδοση (backpropagation). Σε τέτοια δίκτυα, οι μονάδες μπορούν να τροφοδοτούν τόσο προηγούμενα επίπεδα όσο και ίδια. [1]



Εικόνα 3: Τεχνητό Νευρωνικό Δίκτυο πρόσθιας τροφοδότησης [1]

2.1.4 Συνελικτικά νευρωνικά δίκτυα (Convolution Neural Networks)

Τα συνελικτικά νευρωνικά δίκτυα μοιάζουν με τα απλά νευρωνικά δίκτυα και ακολουθούν την μορφή δικτύου πρόσθιας τροφοδότησης. Οι νευρώνες είναι πλήρως συνδεδεμένοι μεταξύ τους. Είναι πολύ χρήσιμα στον τομέα της αναγνώρισης και κατηγοριοποίησης εικόνων, καθώς μπορούν να ανιχνεύουν ορισμένα χαρακτηριστικά (patterns). Τα συνελικτικά νευρωνικά δίκτυα διαφοροποιούνται από τα απλά νευρωνικά δίκτυα επειδή εκτός των άλλων επιπέδων, περιέχουν κρυμμένα επίπεδα, τα οποία ονομάζονται επίπεδα συνέλιξης (convolutional layers). Τα επίπεδα συνέλιξης λειτουργούν όπως και τα άλλα επίπεδα, με την διαφορά ότι η συνάρτηση

ενεργοποίησης που εκτελούν είναι μία συνέλιξη. Τα μοτίβα που ανιχνεύονται μπορούν να είναι γωνίες ή ακμές ή ακόμα και ολόκληρα αντικείμενα, σε πιο περίπλοκα συνελικτικά νευρωνικά δίκτυα.

Η συνέλιξη είναι μία μαθηματική πράξη που παράγει μία συνάρτηση η οποία περιγράφει την κοινή επικαλυπτόμενη περιοχή δύο συναρτήσεων f και g , αφού η μία από αυτές έχει αντιστραφεί και μετατοπιστεί.

Ο γενικός τύπος είναι:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x - y)dy$$

1. Γενικός τύπος συνέλιξης [2]

Όπου $(f * g)$ η νέα συνάρτηση, το $f(y)$ το φίλτρο και $g(x - y)$ η αντεστραμμένη συνάρτηση g .

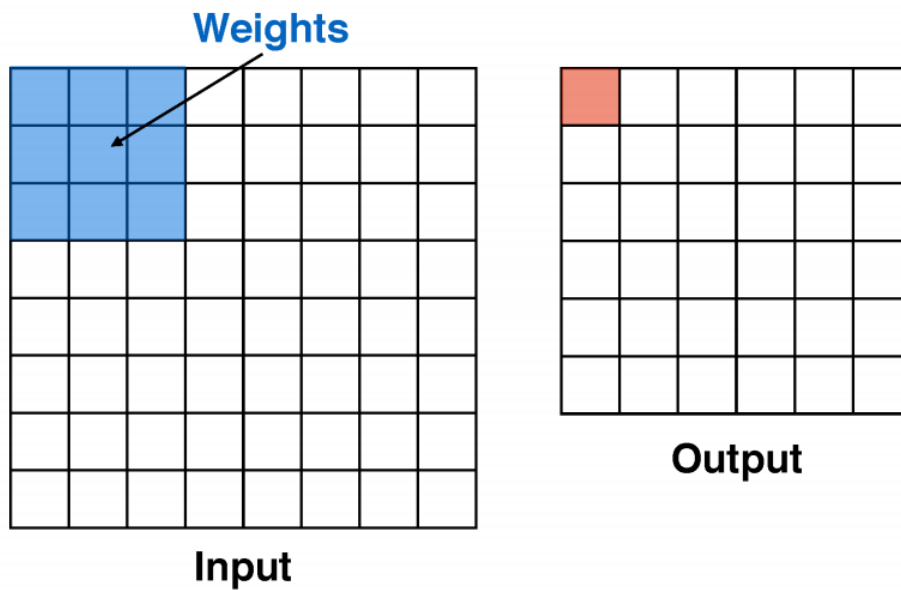
Ο διακριτός τύπος είναι:

$$(f * g)(x, y) = \sum_{i, j=-\infty}^{\infty} f(i, j)I(x - i, y - i)$$

2. Διακριτός τύπος συνέλιξης [2]

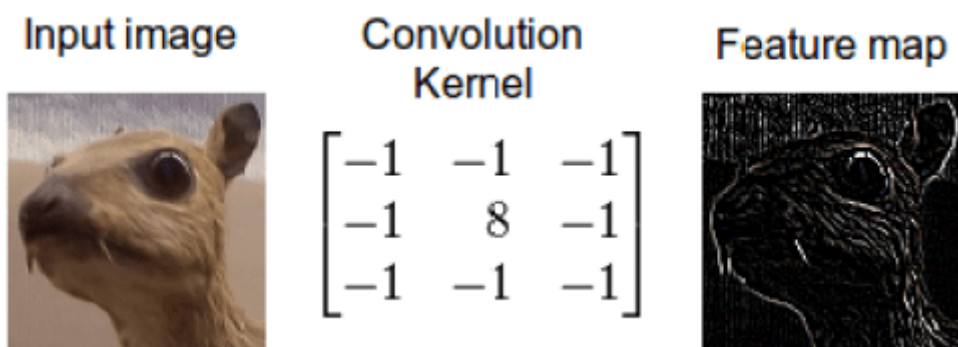
Όπου $(f * g)$ η τελική εικόνα, το $f(i, j)$ το φίλτρο και $I(x - i, y - i)$ η αρχική εικόνα με την αντιστροφή.

Το φίλτρο στα συνελικτικά νευρωνικά δίκτυα είναι ένας πίνακας, συνήθως μικρός σε μέγεθος, που περιέχει τυχαίες τιμές. Ο πίνακας αυτός θα ολισθήσει πάνω από κάθε ομάδα εικονοστοιχείων (pixels) της αρχικής εικόνας που ταιριάζουν στο μέγεθός του, μέχρι να έχει ολισθήσει πάνω από όλες τις πιθανές ομάδες. Η διαδικασία της ολίσθησης γίνεται με την συνέλιξη.



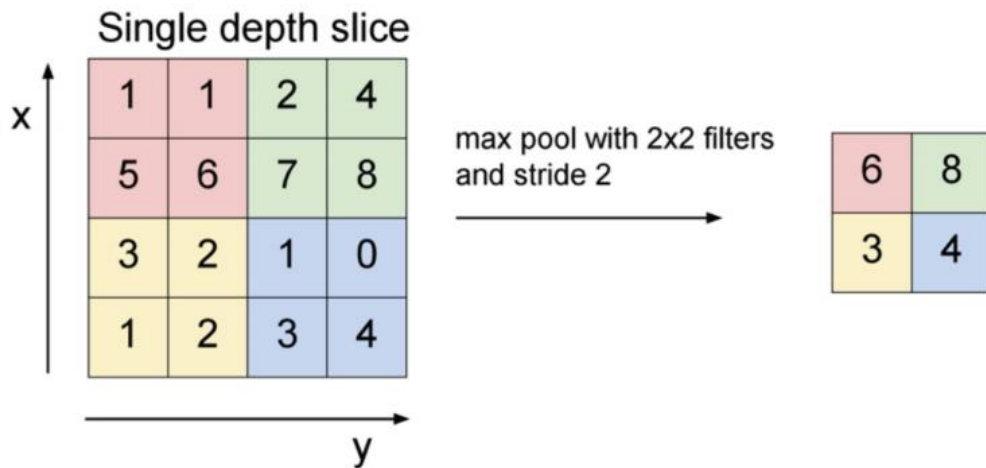
Εικόνα 4: Συνέλιξη μίας 8x8 εικόνας με ένα 3x3 φίλτρο [2]

Με την χρήση πολλαπλών φίλτρων εξάγονται χαρακτηριστικά των εικόνων όπως οι ακμές. Ένα πολύπλοκο δίκτυο επιτρέπει την εξαγωγή πιο εκλεπτυσμένων χαρακτηριστικών, ακόμα και την ανίχνευση ολόκληρων αντικειμένων.



Εικόνα 5: Εξαγωγή χαρακτηριστικών με συνέλιξη [5]

Εκτός από τα επίπεδα συνέλιξης, τα συνελκτικά νευρωνικά δίκτυα περιλαμβάνουν και επίπεδα συγκέντρωσης (pooling). Χρησιμοποιούνται ως μια διαδικασία υποδειγματοληψίας, ώστε να μειωθεί το μέγεθος της εικόνας, διατηρώντας όμως τα σημαντικότερα χαρακτηριστικά της. Υπάρχουν πολλές τεχνικές, με τις δύο κυριότερες να είναι το average pooling και το max pooling.



Εικόνα 6: Max pooling [2]

2.2 Τεχνολογίες και περιβάλλοντα ανάπτυξης

TensorFlow

Το TensorFlow είναι μία βιβλιοθήκη που χρησιμοποιείται ευρέως στην μηχανική μάθηση και τα νευρωνικά δίκτυα. Η χρήση αυτής της βιβλιοθήκης επιτρέπει την γρήγορη υλοποίηση μαθηματικών τύπων και αλγόριθμων που χρησιμοποιούνται στον τομέα των συνελκτικών νευρωνικών δικτύων, με αποτέλεσμα τον σχεδιασμό, την κατασκευή και την εκπαίδευση μοντέλων βαθιάς μάθησης. Οι υπολογισμοί γίνονται με γραφήματα ροής δεδομένων (data flow graphs). Τα γραφήματα αυτά αποτελούνται από κόμβους και ακμές, όπου οι κόμβοι είναι μαθηματικές λειτουργίες ενώ οι ακμές είναι τα δεδομένα. Τα δεδομένα είναι συνήθως τανυστές (tensors).

Keras

Το Keras είναι μία υψηλού επιπέδου διεπαφή προγραμματισμού εφαρμογών (application programming interface, API), που μπορεί να «τρέχει» πάνω στο TensorFlow. Είναι φιλικό προς τον χρήστη και εξυπηρετεί στην γρήγορη κατασκευή νευρωνικών δικτύων χωρίς να χρειάζεται ιδιαίτερη γνώση στο μαθηματικό υπόβαθρο, τις τεχνικές και την βελτιστοποίηση. [10]

CUDA

Το CUDA είναι μία πλατφόρμα παράλληλου προγραμματισμού και προγραμματιστικό μοντέλο που καθιστά εύκολη την χρήση της κάρτας γραφικών για

υπολογισμούς. Ο προγραμματιστής δεν χρειάζεται να γνωρίζει κάποια ειδική γλώσσα προγραμματισμού, καθώς το CUDA υποστηρίζει ένα μεγάλο σύνολο ήδη υπαρχόντων γλωσσών και επεκτείνει το λεξιλόγιό τους με μερικές βασικές εντολές. Οι νέες αυτές εντολές κατευθύνουν τον μεταγλωττιστή στο τμήμα της εφαρμογής που αντιστοιχεί στην κάρτα γραφικών, δίνοντας έτσι την δυνατότητα του παράλληλου προγραμματισμού. [8]

DeepFace

Το DeepFace είναι ένα πλαίσιο λογισμικού (framework) για Python που εστιάζει στην αναγνώριση προσώπου και των χαρακτηριστικών του, όπως η ηλικία, το φύλο, τα συναισθήματα (θυμός, φόβος, ουδετερότητα, λύπη, αηδία, χαρά και έκπληξη) και η φυλή. Βασίζεται κυρίως στις βιβλιοθήκες Keras και Tensorflow. [9]

Pickle

Η βιβλιοθήκη Pickle εφαρμόζει πρωτόκολλα σειριοποίησης, τα οποία παρέχουν την δυνατότητα αποθήκευσης και μεταφόρτωσης αντικειμένων χρησιμοποιώντας δυαδική μορφή. Τα αντικείμενα μπορεί να περιλαμβάνουν αναφορές σε συναρτήσεις, κλάσεις ή ακόμα και αντικείμενα κλάσεων.

OpenCV-2

Το OpenCV είναι μία βιβλιοθήκη ανοικτού κώδικα που χρησιμοποιείται κυρίως στην μηχανική μάθηση και την όραση υπολογιστών. Παρέχει υποδομές και επιταχύνει την αναγνώριση εικόνων και αντικειμένων σε πραγματικό χρόνο. Οι αλγόριθμοι που περιλαμβάνονται σε αυτήν την βιβλιοθήκη επιτρέπουν – μεταξύ άλλων – την αναγνώριση προσώπου και αντικειμένων, αφαίρεση κόκκινων ματιών, παρακολούθηση της κίνησης του ματιού και γενικότερα παρακολούθηση κίνησης αντικειμένων.

Sockets

Το μεγαλύτερο μέρος των εφαρμογών που απαιτούν επικοινωνία ανάμεσα στις διεργασίες τους αποστέλλουν μηνύματα η μία στην άλλη. Οι διεργασίες αναλαμβάνουν τον ρόλο του πελάτη (client) και του εξυπηρετητή (server). Ο client αναλαμβάνει να στέλνει τα μηνύματα ενώ ο server τα επεξεργάζεται και στέλνει

απαντήσεις. Τα μηνύματα συνήθως είναι ρεύματα από bytes. Για να σταλεί το μήνυμα και να περάσει από τις διεργασίες απαιτείται η ύπαρξη δικτύου. Κάθε εφαρμογή δέχεται και στέλνει μηνύματα μέσω μίας διεπαφής λογισμικού που καλείται υποδοχή (socket). Μία διεργασία που θέλει να στείλει μήνυμα σε άλλη διεργασία στέλνει το μήνυμά της μέσω του socket. Απαραίτητη προϋπόθεση είναι να υπάρχει η κατάλληλη υποδομή ώστε το μήνυμα να φτάσει μέσω του δικτύου στην άλλη εφαρμογή η οποία και το παραλαμβάνει πάλι από το socket.

Υπάρχουν δύο πρωτόκολλα μεταφοράς μεταξύ sockets. Το TCP και το UDP. Στην παρούσα εργασία έχει επιλεγθεί το πρωτόκολλο TCP.

Το TCP είναι μία υπηρεσία αξιόπιστης μεταφοράς δεδομένων και έχει ως κύριο χαρακτηριστικό την ανταλλαγή πληροφοριών μεταξύ πελάτη και εξυπηρετητή πριν ξεκινήσει η κανονική αποστολή μηνυμάτων, προκειμένου να διασφαλιστεί ότι και οι δύο πλευρές είναι διαθέσιμες για την ανταλλαγή πακέτων (διαδικασία χειραψίας – handshake). Επομένως, είναι βέβαιο πως στέλνονται όλα τα δεδομένα, χωρίς σφάλματα και με την σωστή σειρά.

Το UDP είναι απλούστερο και ελαφρύτερο από το TCP, καθώς δεν παρατηρείται η διαδικασία χειραψίας. Αυτό όμως το καθιστά αναξιόπιστο ως υπηρεσία μεταφοράς δεδομένων με αποτέλεσμα να μην υπάρχει εγγύηση ότι τα μηνύματα θα παραδοθούν ή ότι θα είναι με την σωστή σειρά.

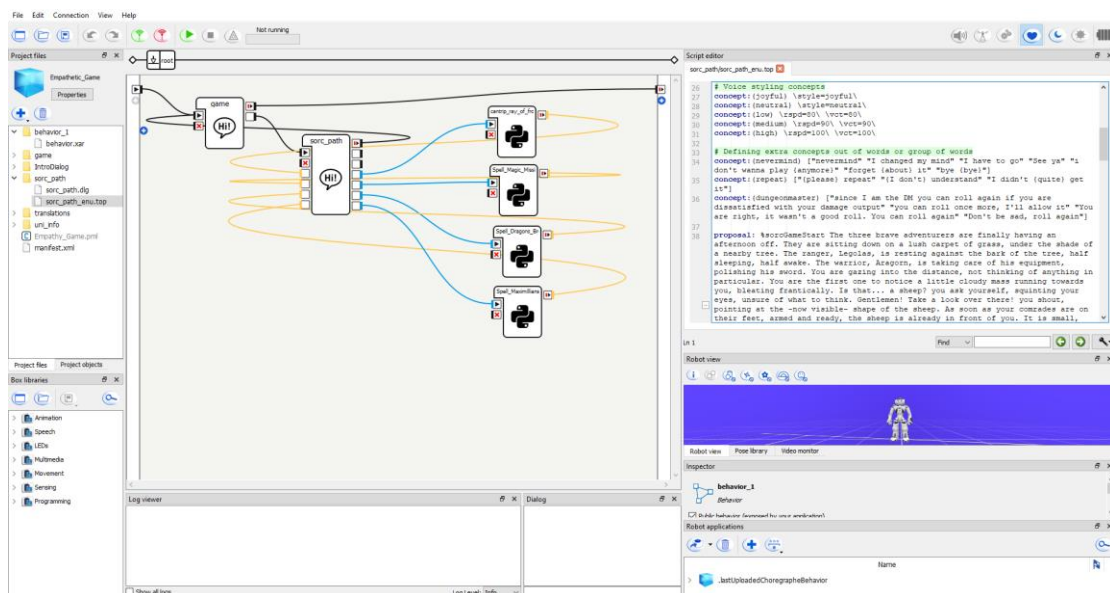
Η ενότητα (module) socket της Python προσφέρει τις απαραίτητες συναρτήσεις για να πραγματοποιηθεί η επικοινωνία. Τυπικά, η μέθοδος υλοποίησης των sockets έχει συγκεκριμένη μορφή. Και οι δύο εφαρμογές δημιουργούν ένα αντικείμενο τύπου socket, ο διακομιστής περιμένει να δεχθεί συνδέσεις και ο πελάτης προσπαθεί να συνδεθεί. Αν η σύνδεση γίνει αποδεκτή, τότε ο διακομιστής δημιουργεί ένα νέο socket ενώ το αρχικό παραμένει ανοιχτό περιμένοντας νέες συνδέσεις. Το συνδεδεμένο socket μπορεί πλέον να χρησιμοποιηθεί για ανταλλαγή δεδομένων ανάμεσα στις διεργασίες. [6]

NumPy

Η βιβλιοθήκη NumPy (Numerical Python) επεκτείνει τις ήδη υπάρχουσες δυνατότητες της Python όσον αφορά τις μαθηματικές πράξεις. Υποστηρίζει μεταξύ άλλων, πράξεις μεταξύ πινάκων, συναρτήσεις γραμμικής άλγεβρας και μετασχηματισμό Fourier.

Choregraphe

Το Choregraphe είναι μία πλατφόρμα ανάπτυξης λογισμικού για τα ρομπότ της SoftBank Robotics. Μέσα από το Choregraphe είναι δυνατή η κατασκευή κινήσεων, συμπεριφορών και διαλόγων, η δοκιμή τους σε προσομοιωτή ή σε κανονικό ρομπότ αλλά και η συγγραφή κώδικα python για εμπλουτισμό των συμπεριφορών. Με το Choregraphe δεν είναι απαραίτητη η γνώση κάποιας γλώσσας προγραμματισμού καθώς υπάρχουν έτοιμα κατασκευασμένα κουτιά (boxes) που περιέχουν τόσο βασικές όσο και πιο σύνθετες λειτουργίες, όπως ο χορός ή η αποστολή e-mails, τα οποία ο χρήστης μπορεί να χρησιμοποιήσει τις συμπεριφορές που κατασκευάζει τοποθετώντας τα απλά στην σειρά της αρεσκείας του και συνδέοντάς τα μεταξύ τους. Τα κουτιά αυτά μπορούν να εξατομικευθούν ανάλογα με τις ανάγκες του χρήστη. Μερικές συνηθισμένες επιλογές που προσφέρονται είναι η επιπλέον παραμετροποίηση ή η προσθήκη μεταβλητών.



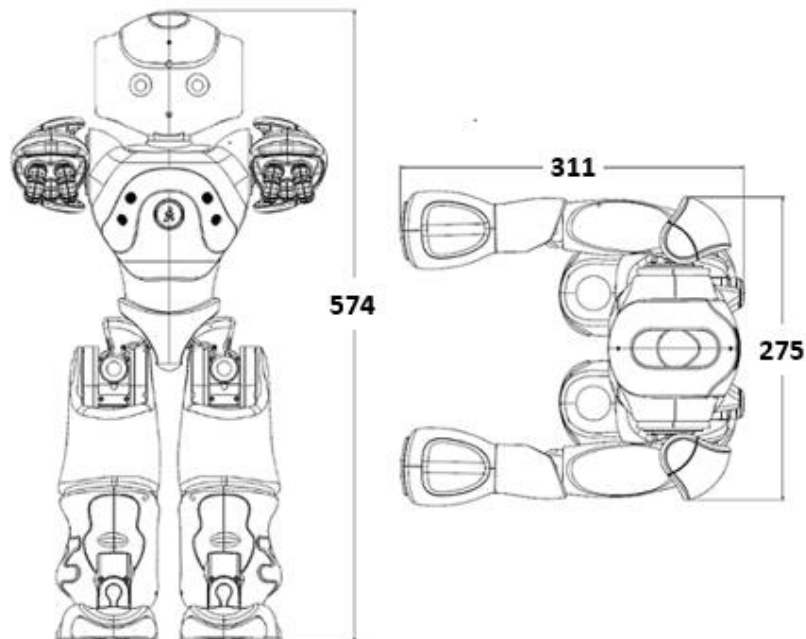
Εικόνα 7: Το περιβάλλον ανάπτυξης εφαρμογών Choregraphe

2.3 Ανθρωποειδές NAO

Το NAO είναι ένα ανθρωποειδές ρομπότ που παρουσιάζει αυτονομία στις κινήσεις του και κατασκευάζεται από την SoftBank Robotics. Χρησιμοποιείται σε τομείς όπως η έρευνα και η εκπαίδευση, αλλά και στην υγεία για να ενημερώνει και να ψυχαγωγεί τους επισκέπτες.



Εικόνα 8: NAO

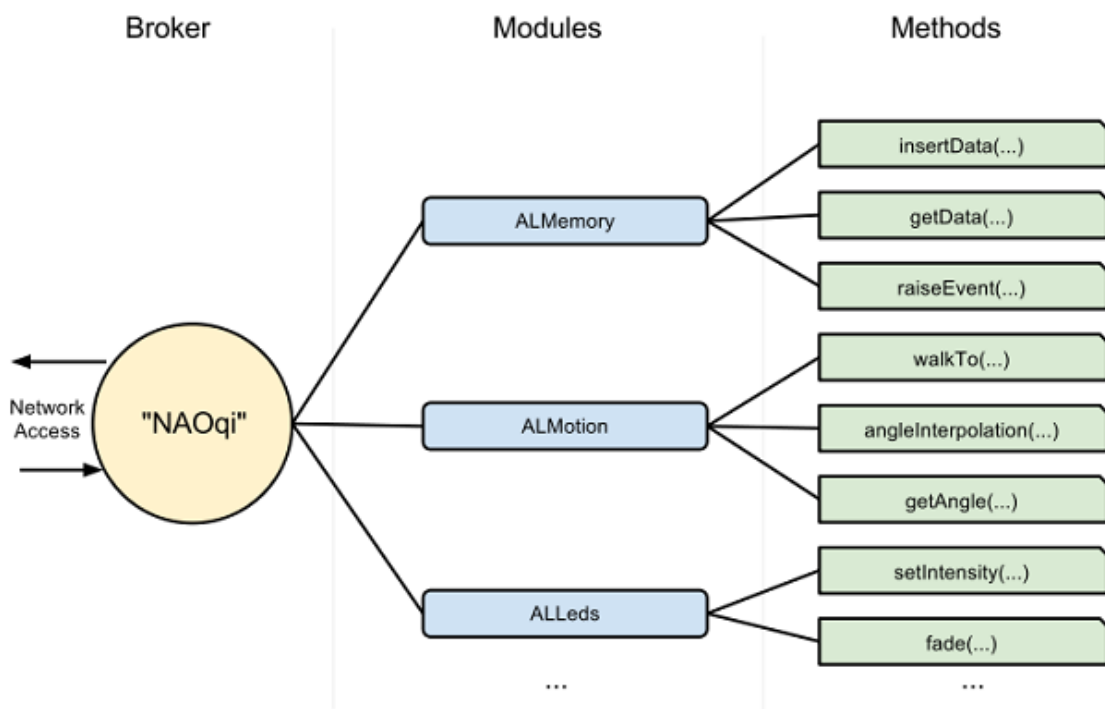


Εικόνα 9: Διαστάσεις του NAO [7]

NAOqi API

Το NAOqi είναι το λειτουργικό σύστημα του ρομπότ. Χρησιμοποιείται τόσο στα NAO όσο και στα PEPPER μοντέλα και είναι διανομή Linux βασισμένο σε Gentoo. Παρέχει προγράμματα και βιβλιοθήκες που είναι απαραίτητα για το ρομπότ.

Το NAOqi Framework περιέχει ενότητες (modules) όπως η κίνηση, η όραση, η ακοή και η αντίληψη. Ακόμη, προσφέρεται και η δυνατότητα να κατασκευάσει κάποιος το δικό του module, προσθέτοντας έτσι λειτουργικότητα ή τροποποιώντας την ήδη υπάρχουσα. Με το NAOqi Framework είναι δυνατός ο προγραμματισμός του ρομπότ για όλων των ειδών τις εργασίες και τις μελέτες, χωρίς να υπάρχει περιορισμός στις πλατφόρμες και στα λογισμικά που χρησιμοποιούνται από τους προγραμματιστές. Το NAOqi Framework έχει την ικανότητα να γνωρίζει ποιες συναρτήσεις είναι διαθέσιμες και που (introspection). Επίσης, έχει δύο API, ένα σε C++ και ένα σε Python. [11]



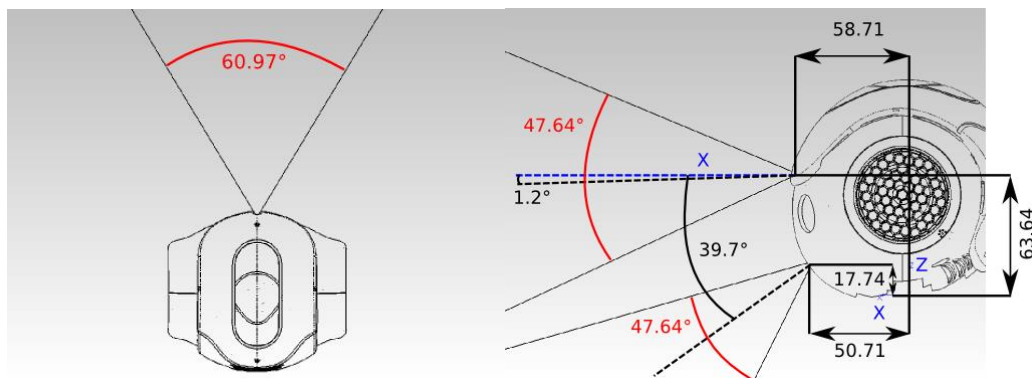
Εικόνα 10: Η διαδικασία NAOqi [11]

ALProxy

Το ALProxy module επιτρέπει κατασκευές αντικειμένων τύπου proxy προς κάποιο module. Ένα proxy αντικείμενο συμπεριφέρεται ακριβώς όπως το module το οποίο εκπροσωπεί. Μπορεί να χρησιμοποιηθεί και για χειρισμό κάποιου module στο NAO, χωρίς ο κώδικας να επιβάλλεται να είναι ανεπτυγμένος στο Choregraphe, το λογισμικό ανάπτυξης εφαρμογών για τα NAO και PEPPER, καθώς επιτρέπει την σύνδεση με το ρομπότ μέσω IP. [12]

ALVideoDevice

Το ALVideoDevice module παρέχει και προωθεί εικόνες είτε από το NAO είτε από τον simulator σε άλλα modules που μπορούν να διαχειριστούν εικόνες, όπως το ALFaceDetection και το ALVisionRecognition. Το ALVideoDevice γνωρίζει τα modules που διαχειρίζονται εικόνες και τις απαιτήσεις τους οπότε μπορεί να θέσει τις απαραίτητες τιμές ώστε να εξυπηρετήσει τα modules αποτελεσματικά. Το NAO είναι εξοπλισμένο με δύο 2D κάμερες, ίδιες μεταξύ τους, που βρίσκονται στο μέτωπο. Μπορούν να παρέχουν ανάλυση ύψους 1280x960 σε 30 καρέ ανά δευτερόλεπτο. [13]



Εικόνα 11: Θέσεις των καμερών στο NAO [13]

ALDialog

Το ALDialog module επιτρέπει ικανότητες ομιλίας στο ρομπότ, χρησιμοποιώντας μια λίστα με κανόνες γραμμένους και κατηγοριοποιημένους με τον κατάλληλο τρόπο. Οι κανόνες χωρίζονται σε δύο κατηγορίες: τους κανόνες χρήστη (user rules) και τις προτάσεις (proposal rules). Τα user rules συσχετίζουν μία συγκεκριμένη είσοδο που

θα δώσει ο χρήστης - για παράδειγμα μία λέξη ή πρόταση - με μία συγκεκριμένη απάντηση από το ρομπότ. Τα proposal rules ξεκινούν μία απάντηση του ρομπότ χωρίς να χρειάζονται είσοδο από τον χρήστη. Η κατηγοριοποίηση αφορά ξεχωριστά θέματα (topics), τα οποία μπορούν να δημιουργηθούν με οποιονδήποτε επεξεργαστή κειμένου ή μέσα από το Choregraphe, με την χρήση των qiChat boxes. [15]

3. ΣΧΕΔΙΑΣΜΟΣ & ΥΛΟΠΟΙΗΣΗ

Στο παρόν κεφάλαιο παρουσιάζεται η αρχιτεκτονική της εργασίας, η ροή των δεδομένων και πως αυτά χρησιμοποιούνται, η δομή του νευρωνικού δικτύου και ο κώδικάς του, τα modules που χρησιμοποιήθηκαν καθώς και παραδείγματα λογισμικού.

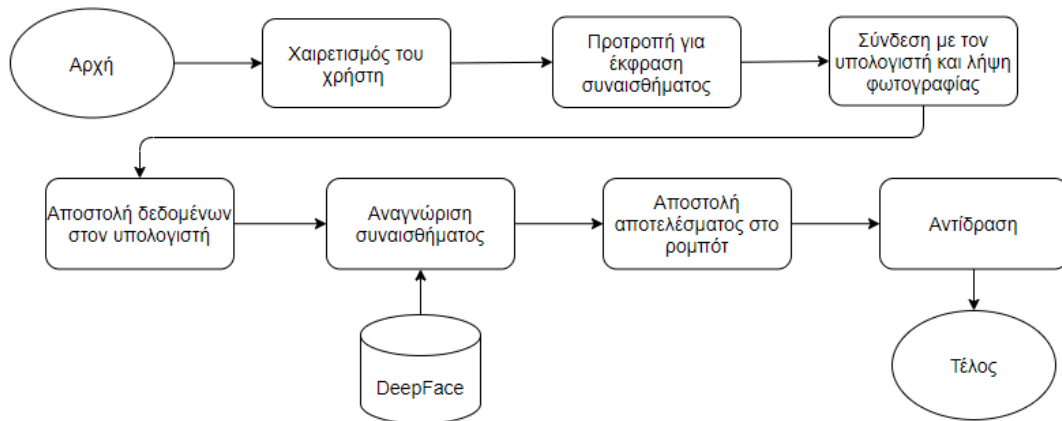
3.1 Αρχιτεκτονική

Λειτουργικότητα

Η λειτουργικότητα της εργασίας παρουσιάζεται ως εξής:

- Το ρομπότ χαιρετάει τον χρήστη και του ζητάει να κάνει μία έκφραση με το πρόσωπό του.
- Το ρομπότ αναμένει μερικά δευτερόλεπτα ώστε ο χρήστης να πάρει την έκφραση.
- Το ρομπότ συνδέεται με τον υπολογιστή και λαμβάνει μία φωτογραφία του χρήστη την οποία θα στείλει στον υπολογιστή.
- Η φωτογραφία στέλνεται στον υπολογιστή. Αποθηκεύεται σε μορφή «.jpg»
- Με την χρήση του συνελκτικού νευρωνικού δικτύου και ενός ήδη εκπαιδευμένου μοντέλου, γίνεται η αποτίμηση της έκφρασης του προσώπου.
- Το αποτέλεσμα στέλνεται πίσω στο ρομπότ.
- Το ρομπότ συνάγει το τελικό αποτέλεσμα και αντιδρά.

Για την μεταβίβαση των δεδομένων από το ρομπότ στον υπολογιστή και το αντίθετο, χρησιμοποιήθηκαν sockets. Ακολουθείται η δομή client-server όπου το NAO λειτουργεί ως client και ο υπολογιστής ως server. Ο κώδικας του client αναπτύχθηκε μέσα στο Choregraphe με χρήση των κουτιών rython (rython boxes) που περιλαμβάνονται στην σουίτα ανάπτυξης λογισμικού. Η εικόνα αποθηκεύεται στον υπολογιστή σε μορφή «.jpg». Η ανακατασκευή των αρχείων στον υπολογιστή μετά την αποστολή τους από το ρομπότ γίνεται με την χρήση του module pickle. Η αποτίμηση των συναισθημάτων του προσώπου γίνεται με το νευρωνικό δίκτυο και το εκπαιδευμένο μοντέλο του [DeepFace](#). Για την παραγωγή αντιδράσεων χρησιμοποιήθηκε διάλογος μέσω qiChat.

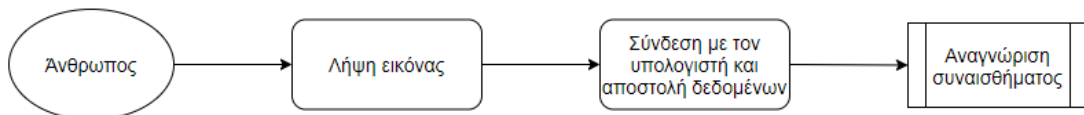


Εικόνα 12: Διάγραμμα ροής συστήματος

Ροή δεδομένων

Εισερχόμενα δεδομένα

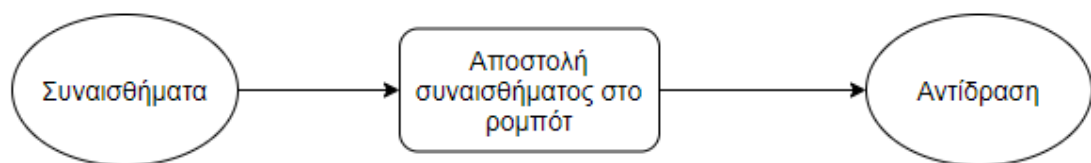
Για τα εισερχόμενα δεδομένα, όταν πραγματοποιηθεί ο χαιρετισμός του ρομπότ προς τον άνθρωπο και περάσουν μερικά δευτερόλεπτα, λαμβάνεται η φωτογραφία του, δημιουργείται η τοπική σύνδεση με τον υπολογιστή και τα αρχεία στέλνονται και αποθηκεύονται εκεί (η εικόνα με μορφή «.jpg»).



Εικόνα 13: Διάγραμμα ροής εισερχόμενων δεδομένων

Εξερχόμενα δεδομένα

Αφού γίνει η αναγνώριση του συναισθήματος από την εικόνα, το αποτέλεσμα στέλνεται στο ρομπότ το οποίο αποφασίζει την ανάλογη αντίδραση.



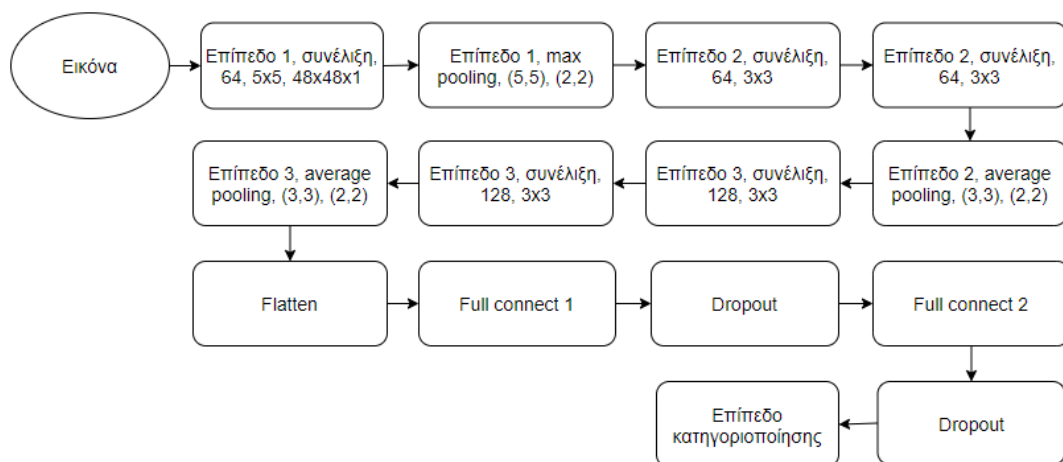
Εικόνα 14: Διάγραμμα ροής εξερχόμενων δεδομένων

3.2 Υλοποίηση

Δομή και κώδικας νευρωνικού δικτύου

Στην παρούσα εργασία, για την ανάλυση των συναισθημάτων του προσώπου χρησιμοποιούνται έτοιμα εκπαιδευμένα μοντέλα, τα οποία παρέχονται από το DeepFace. Συγκεκριμένα, έχουν κατασκευαστεί προσαρμοσμένα μοντέλα, έχουν υπολογιστεί τα βάρη τους και έχουν εκπαιδευτεί, παράγοντας «.h5» αρχεία που χρησιμοποιούνται για να διευκρινιστεί το φύλο, η ηλικία και τα συναισθήματα. Για τους σκοπούς της εργασίας χρησιμοποιείται το αρχείο με όνομα «facial_expression_model_weights.h5» που πραγματοποιεί ανάλυση συναισθημάτων. Τα συναισθήματα που ανιχνεύονται είναι θυμός, φόβος, ουδετερότητα, θλίψη, αηδία, χαρά και έκπληξη, τα οποία είναι σύμφωνα με τις 6 βασικές συγκινήσεις (basic emotions) του μοντέλου FACS του Ekman (1978).

Η αρχιτεκτονική του συνελκτικού νευρωνικού δικτύου του DeepFace περιλαμβάνει τρία επίπεδα συνέλιξης και έχει την παρακάτω μορφή:



Εικόνα 15: Δομή νευρωνικού δικτύου DeepFace

Το δίκτυο έχει αναπτυχθεί σε Keras και Tensorflow. Αρχικά προσδιορίζεται ο αριθμός των συναισθημάτων και γίνεται αρχικοποίηση του δικτύου.

```
num_classes = 7

model = Sequential()
```

Εικόνα 16: Κατασκευή του νευρωνικού δικτύου [16]

Έπειτα, κατασκευάζεται το πρώτο 2D επίπεδο συνέλιξης που περιλαμβάνει 64 φίλτρα και κάθε πυρήνας έχει διαστάσεις 5x5. Ο αριθμός των φίλτρων υποδηλώνει πόσοι πυρήνες θα πραγματοποιήσουν την πράξη της συνέλιξης με την εικόνα. Συνήθως, τα αρχικά επίπεδα των συνελκτικών νευρωνικών δικτύων έχουν μικρότερο αριθμό φίλτρων σε σχέση με τα επίπεδα που βρίσκονται βαθύτερα στην αρχιτεκτονική του δικτύου. Το προκαθορισμένο “relu” (rectified linear unit) χρησιμοποιείται ως συνάρτηση ενεργοποίησης. Γίνεται επίσης και διαμόρφωση της εικόνας ώστε να έχει τις επιθυμητές διαστάσεις, 48x48x1. Ακολουθεί το πρώτο 2D max pooling layer με μέγεθος 5x5 και βηματισμό (strides) 2x2. Σε αυτήν την περίπτωση, το max pooling layer πραγματοποιεί υποδειγματοληψία στην εικόνα, κατασκευάζοντας ένα «παράθυρο» διαστάσεων 5x5 το οποίο λαμβάνει κάθε φορά την μεγαλύτερη τιμή που περιέχεται στα χαρακτηριστικά της εικόνας (feature map) και ολισθαίνει κάθε φορά κατά 2x2, μέχρι να διατρέξει όλη την εικόνα και να έχει εξάγει τις μέγιστες τιμές.

```
#1st convolution layer
model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))
```

Εικόνα 17: Πρώτο επίπεδο νευρωνικού δικτύου [16]

Το δεύτερο επίπεδο του νευρωνικού δικτύου απαρτίζεται από δύο επίπεδα 2D συνέλιξης, όμοια μεταξύ τους, με 64 φίλτρα, διαστάσεις πυρήνων 3x3 και συνάρτηση ενεργοποίησης την “relu”. Το pooling layer περιλαμβάνει 2D average pooling, το οποίο πραγματοποιεί υποδειγματοληψία στην εικόνα όπως και το προηγούμενο, μόνο που στην συγκεκριμένη περίπτωση το παράθυρο έχει διαστάσεις 3x3 και διατηρείται ο μέσος όρος του feature map αντί για την μεγαλύτερη τιμή.

```
#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))
```

Εικόνα 18: Δεύτερο επίπεδο νευρωνικού δικτύου [16]

Το τρίτο επίπεδο είναι πανομοιότυπο με το δεύτερο. Η μόνη αλλαγή παρατηρείται στον αριθμό των φίλτρων, που ανέρχεται στα 128.

```
#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))
```

Εικόνα 19: Τρίτο επίπεδο νευρωνικού δικτύου [16]

Με το flatten, τα δεδομένα του δικτύου μετατρέπονται σε μονοδιάστατο πίνακα ώστε να μπορούν να μεταφερθούν στο επόμενο επίπεδο του δικτύου. Είναι απαραίτητο να γίνει ώστε τα δεδομένα να μπορούν να εισέλθουν στο πλήρως συνδεδεμένο δίκτυο.

```
model.add(Flatten())
```

Εικόνα 20: Μετατροπή σε μονοδιάστατο πίνακα [16]

Τα δύο πλήρη συνδεδεμένα επίπεδα. Η συνάρτηση Dense δείχνει πως η σύνδεση των νευρώνων είναι πυκνή, δηλαδή όλοι οι νευρώνες του επιπέδου λαμβάνουν είσοδο από όλους τους νευρώνες του προηγούμενου επιπέδου. Ο αριθμός 1024 αντιστοιχεί στον αριθμό των νευρώνων που υπάρχουν στο επίπεδο. Χρησιμοποιείται πάλι η συνάρτηση ενεργοποίησης “relu”, ενώ το dropout αφαιρεί τυχαία ένα ποσοστό των νευρώνων, εδώ 0.2, κάθε φορά.

```
#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
```

Εικόνα 21: Πλήρης σύνδεση του νευρωνικού δικτύου [16]

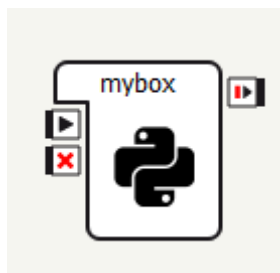
Τέλος, το επίπεδο κατηγοριοποίησης (classification layer) περιέχει τόσους νευρώνες όσες και η κατηγορίες των συναισθημάτων και ως συνάρτηση ενεργοποίησης ορίζεται η “softmax”, που είναι μία γενίκευση της λογιστικής συνάρτησης.

```
model.add(Dense(num_classes, activation='softmax'))
```

Εικόνα 22: Classification layer [16]

Κουτιά (boxes) στο Choregraphe

Ένα κουτί στο Choregraphe αποτελεί το βασικό δομικό στοιχείο προκειμένου ν' αποκτήσει το ρομπότ λειτουργικότητα. Είναι γραμμένα σε Python, με δυνατότητες εξατομίκευσης και βρίσκονται είτε έτοιμα (από την βιβλιοθήκη του Choregraphe) είτε μπορεί να κατασκευάσει δικά του ο χρήστης ώστε να καλύψει τις ανάγκες του.



Εικόνα 23: Ένα Python box στο Choregraphe

Υπάρχουν τέσσερα βασικά είδη κουτιών στο Choregraphe. Στην εικόνα 24, από αριστερά προς τα δεξιά υπάρχουν: Κουτιά διαγράμματος (Diagram), χρονοδιαγράμματος (timeline), κώδικα Python και διαλόγου (dialog). Ένα κουτί διαγράμματος περιέχει συνήθως άλλα κουτιά που σχετίζονται μεταξύ τους. Ένα τέτοιο κουτί είναι και όλη η αρχική συμπεριφορά. Τα κουτιά χρονοδιαγράμματος χρησιμοποιούνται για την κατασκευή κινήσεων. Τα κουτιά κώδικα χρησιμοποιούνται για την συγγραφή κώδικα ενώ τα κουτιά διαλόγου περιέχουν σενάρια διαλόγων.



Εικόνα 24: Τα τέσσερα βασικά κουτιά στο Choregraphe

Ένα κουτί αποτελείται από εισόδους και εξόδους. Οι εισοδοί εντοπίζονται στα αριστερά του κουτιού ενώ οι έξοδοι στα δεξιά. Οι εισοδοί χρησιμοποιούνται για την ενεργοποίηση του κουτιού και μπορούν να λάβουν δεδομένα από άλλα κουτιά ενώ οι έξοδοι συνεχίζουν την ενεργοποίηση άλλων κουτιών και μεταφέρουν δεδομένα σε άλλα κουτιά. Μερικά κουτιά μπορούν να έχουν και παραμέτρους. Τα κουτιά

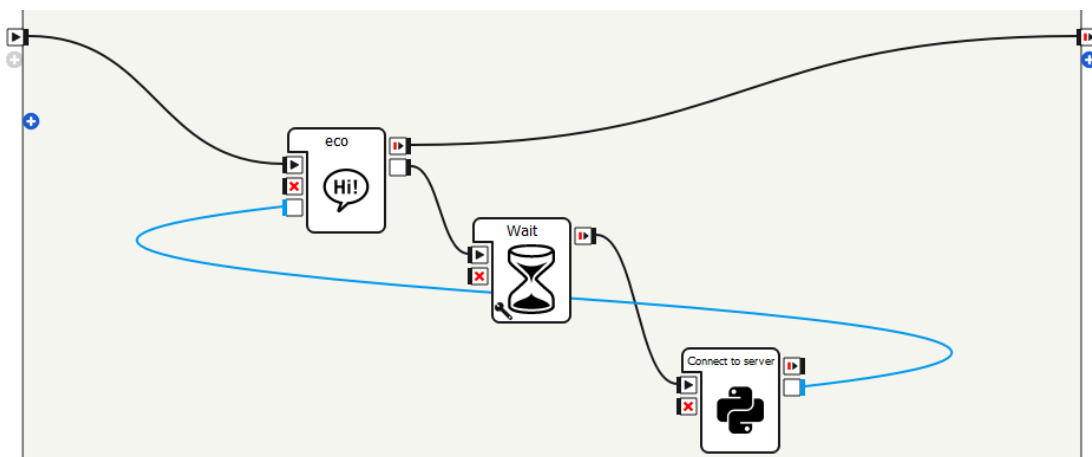
επικοινωνούν μεταξύ τους είτε απλά μέσω των γραμμών που τα συνδέουν είτε παραδίδοντας και πληροφορίες το ένα στο άλλο. Οι τύποι των δεδομένων που υπάρχουν για εισόδους/εξόδους είναι τέσσερις: η εκπυροσκόρτηση (bang) που δεν δέχεται ή μεταφέρει δεδομένα, ο αριθμός (number) που χρησιμοποιείται για μεταφορά αριθμών, η συμβολοσειρά (string) που μεταφέρει συμβολοσειρές και ο δυναμικός (dynamic) που μπορεί να μεταφέρει όλους τους διαθέσιμους τύπους. Κάθε ένας τύπος αντιπροσωπεύεται και από διαφορετικό χρώμα στα κουτιά. Όλες οι εισοδοί/έξοδοι μπορούν να εξατομικευθούν. [14]



Εικόνα 25: Είσοδοι/έξοδοι κουτιού. Το μαύρο αντιστοιχεί στο bang, το κίτρινο στο number, το γαλάζιο στο string και το γκρι στο dynamic

Δομή συμπεριφοράς στο Choregraphe

Το αρχείο συμπεριφοράς που αναπτύχθηκε στο Choregraphe έχει την παρακάτω μορφή:



Εικόνα 26: Δομή αρχείου συμπεριφοράς στο Choregraphe

Αποτελείται από τρία κουτιά. Το πρώτο αφορά ένα κουτί διαλόγου, το δεύτερο ένα προκατασκευασμένο κουτί που έχει ως στόχο την δημιουργία μίας μικρής αναμονής και το τρίτο είναι ένα κουτί για συγγραφή κώδικα ρυθον. Το κουτί του διαλόγου και το κουτί με τον κώδικα ρυθον έχουν υποστεί τροποποιήσεις στις εισόδους και

εξόδους τους με την προσθήκη μεταβλητών που χρησιμοποιούνται για μεταφορά πληροφοριών ή για εκκίνηση άλλων κουτιών.

NAO QiChat

Σ' ένα κουτί διαλόγου στο Choregraphe αρχικά συναντάται το όνομα του θέματος και η γλώσσα. Το όνομα του θέματος συμπίπτει με το όνομα του κουτιού. Το grg είναι το κωδικό όνομα της γλώσσας Ελληνικά στο qiChat.

```
topic: ~eco()  
language: grg
```

Εικόνα 27: Θέμα (topic) και γλώσσα (language)

Στην συνέχεια, με ένα user rule ξεκινάει ο διάλογος. Τα user rules ορίζονται με το u: (input) όπου το input αντιστοιχεί στην είσοδο που θα δώσει ο χρήστης. Εδώ, είναι ένα γεγονός (event), που ενεργοποιείται από την μεταβλητή onStart που βρίσκεται στις εισόδους του κουτιού. Η μεταβλητή αυτή ενεργοποιείται όταν ξεκινήσει να εκτελείται η συμπεριφορά στο ρομπότ. Επομένως, το ρομπότ θα χαιρετήσει τον χρήστη χωρίς να περιμένει κάποιο ερέθισμα από τον χρήστη. Μόλις ολοκληρώσει την πρότασή του, το ρομπότ θα ενεργοποιήσει την μεταβλητή output που βρίσκεται στις εξόδους του κουτιού. Η μεταβλητή αυτή με την σειρά της θα ενεργοποιήσει το κουτί αναμονής, χωρίς να χρειαστεί να σταματήσει το κουτί του διαλόγου.

```
u:(e:onStart) Γεια σας! Αν σας άρεσε το παιχνίδι με το ναο κάντε μία αξιολόγηση!  
Μπορείτε να κάνετε μία γκριμάτσα. Για παράδειγμα, αν σας άρεσε χαμογελάστε. $output=1
```

Εικόνα 28: User rule

Αφού εκτελεστεί ο κώδικας της ανάλυσης συναισθήματος, το αποτέλεσμα επιστρέφεται στο κουτί διαλόγου μέσω της μεταβλητής εισόδου user_emotion. Εφ' όσον η λειτουργία του κουτιού διαλόγου δεν σταμάτησε, το ρομπότ περιμένει την αντίδραση του χρήστη για να συνεχίσει με την ανάλογη απάντηση. Αυτό επιτυγχάνεται με την δημιουργία υπο-κανόνων (subrules). Στο qiChat συμβολίζονται με u1, u2, u3 και λοιπά, ανάλογα με το πόσο θέλει ο προγραμματιστής να εμβαθύνει σε κάποιο θέμα. Λειτουργούν όπως και οι κανονικοί user rules, με την διαφορά ότι οι συγκεκριμένοι διάλογοι ενεργοποιούνται μόνο εάν ο κύριος κανόνας έχει ενεργοποιηθεί.

Εδώ, ο κύριος κανόνας έχει ενεργοποιηθεί από την αρχή της συμπεριφοράς και παραμένει ενεργοποιημένος καθώς το κουτί δεν έχει απενεργοποιηθεί κατά την διάρκεια, οπότε περιμένει να ενεργοποιηθεί η μεταβλητή `user_emotion` (`e:user_emotion`) και ταυτόχρονα να έχει λάβει και μία από τις ακόλουθες τιμές: χαρά (`happy`), λύπη (`sad`), ουδετερότητα (`neutral`), θυμό (`angry`), φόβο (`fear`), αηδία (`disgust`) ή έκπληξη (`surprise`). Μόλις τελειώσει το ρομπότ με την ανάλογη απάντηση, ενεργοποιεί την μεταβλητή `onStopped` η οποία τερματίζει και την λειτουργία της συμπεριφοράς.

```
u1:("e:user_emotion $user_emotion=="happy") Χαίρομαι που σας άρεσε! $onStopped=1
u1:("e:user_emotion $user_emotion=="sad") Λυπάμαι, την επόμενη φορά θα προσπαθήσω
περισσότερο. $onStopped=1
u1:("e:user_emotion $user_emotion=="neutral") Ευχαριστώ πολύ! $onStopped=1
u1:("e:user_emotion $user_emotion=="angry") Λυπάμαι που σας δυσαρέστησα τόσο.
$onStopped=1
u1:("e:user_emotion $user_emotion=="fear") Δεν χρειάζεται να φοβάστε.
$onStopped=1
u1:("e:user_emotion $user_emotion=="disgust") Και τα ρομπότ έχουν αισθήματα.
$onStopped=1
u1:("e:user_emotion $user_emotion=="surprise") Διακρίνω μία έκπληξη! $onStopped=1
```

Εικόνα 29: User rules που βασίζονται στο συναίσθημα

Η διαδικασία της ανάλυσης συναίσθηματος μπορεί να συνεχίσει να τρέχει επαναληπτικά και να υπάρχει περαιτέρω αλληλεπίδραση με το ρομπότ με την προσθήκη επιπλέον κανόνων.

Κώδικας sockets

Client

Αρχικά, με την συνάρτηση `socket.socket()` δημιουργείται ένα αντικείμενο τύπου `socket`. Οι σταθερές `AF_INET` και `SOCK_STREAM` που χρησιμοποιούνται σχετίζονται με τα πρωτόκολλα, τις διευθύνσεις και του τύπους των sockets. Ο client καλεί την συνάρτηση `connect()` ώστε να συνδεθεί με τον server.

```
HOST = '127.0.0.1' # The server's hostname or IP address
PORT = 6666      # The port used by the server

#Establish connection
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
```

Εικόνα 30: Σύνδεση με τον server

Μετά την σύνδεση, προκειμένου να υπάρξει πρόσβαση στις κάμερες του ρομπότ, χρειάζεται το ALVideoDevice proxy. Για να γίνει η «εγγραφή» (subscribe) στο proxy, ορίζεται η ανάλυση, ο χρωματικός χώρος και ο ελάχιστος αριθμός των frames.

```
#Call ALVideoDevice service
session = self.session()
camSession = session.service("ALVideoDevice")
resolution = 2 #VGA
colorSpace = 11 #RGB
fps = 5

nameId = camSession.subscribeCamera("cameratest",
0, resolution, colorSpace, fps)
```

Εικόνα 31: ALVideoDevice service

Με την getImageRemote() λαμβάνεται η τελευταία εικόνα από την κάμερα και πραγματοποιείται η επεξεργασία της (με μετασχηματισμούς) προκειμένου να λάβει την μορφή που χρειάζεται. Η συνάρτηση releaseImage() ελευθερώνει τον buffer της εικόνας από την getImageRemote() και τέλος, γίνεται απεγγραφή από το service.

```
#Get camera image
naoImage = camSession.getImageRemote(nameId)
camSession.releaseImage(nameId)
camSession.unsubscribe(nameId)
```

Εικόνα 32: Λήψη της εικόνας

Αφού ολοκληρωθούν τα παραπάνω, ξεκινάει η διαδικασία της αποστολής των δεδομένων. Πρώτα στέλνεται το μέγεθος των δεδομένων, περισσότερο για δοκιμή. Έπειτα, στέλνονται οι πληροφορίες τις εικόνας. Σύμφωνα με την τεκμηρίωση του NAO, η εικόνα αποθηκεύεται με μία μορφή container, όπου η 6^η θέση αντιστοιχεί σ' ένα δυαδικό πίνακα που περιέχει την πληροφορία της εικόνας. Προκειμένου ο server να γνωρίζει πότε έχουν αποσταλεί όλες οι πληροφορίες, στο τέλος του πίνακα προτίθεται η συμβολοσειρά «[END OF IMAGE]», που έχει οριστεί ως συμβολοσειρά τερματισμού. Η πληροφορία αυτή σειριοποιείται με την συνάρτηση dumps() του module Pickle. Αφού σταλούν όλα τα bytes με την συνάρτηση send(), το NAO αναμένει για την αποστολή της απάντησης από τον server. Τέλος, τυπώνει το αποτέλεσμα για λόγους εντοπισμού σφαλμάτων.

```

try:
    #Send buffer size
    size = sys.getsizeof(pickle.dumps(naoImage[6]))
    s.send(str(size))

    #Send image to server
    s.send(pickle.dumps(naoImage[6]) + b'[END OF IMAGE]')

    #Get emotion response from server
    emotion=b""
    emotion = s.recv(1024)
    self.logger.info(emotion)

```

Εικόνα 33: Αποστολή εικόνας και λήψη απάντησης

Πριν την ολοκλήρωση του προγράμματος, γίνεται χρήση της μεταβλητής `emotion` που έχει προστεθεί στο κουτί ώστε το αποτέλεσμα να μεταφερθεί στο κουτί διαλόγου.

```

finally:
    s.close()
    self.output_emotion(emotion)
    self.onStopped()
pass

```

Εικόνα 34: Χρήση μεταβλητών του κουτιού για την προώθηση του αποτελέσματος

Server

Αρχικά, με την συνάρτηση `socket.socket()` δημιουργείται ένα αντικείμενο τύπου `socket`. Οι σταθερές `AF_INET` και `SOCK_STREAM` που χρησιμοποιούνται σχετίζονται με τα πρωτόκολλα, τις διευθύνσεις και του τύπους των sockets. Η συνάρτηση `setsockopt()` προβλέπει την αποφυγή του σφάλματος «address already in use», που μπορεί να εμφανιστεί εάν ο server ξεκινήσει σε port που είναι ήδη κατειλημμένη από άλλη διαδικασία. Η σταθερά `SOL_SOCKET` ορίζει το επίπεδο (εδώ το socket level) στο οποίο θα οριστούν κάποιες επιλογές, όπως το `SO_REUSEADDR` το οποίο επιτρέπει την επαναχρησιμοποίηση τοπικών διευθύνσεων στο socket. Η `bind()` συσχετίζει το socket με μία συγκεκριμένη διεπαφή δικτύου και θύρα. Η `listen()` επιτρέπει στον server να δέχεται συνδέσεις. Ο αριθμός (εδώ 10), δηλώνει τον αριθμό των μη αποδεκτών συνδέσεων που θα επιτραπεί από το σύστημα πριν την άρνηση νέων συνδέσεων.

```

HOST = '127.0.0.1' # Standard loopback interface address (localhost). Change with current address
PORT = 6666      # Port to listen on (non-privileged ports are > 1023)

#Create socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print("Socket created")
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

s.bind((HOST, PORT))
print("Socket bind complete")
s.listen(10)
print("Socket now listening")

```

Εικόνα 35: Δημιουργία socket

Μετά την δημιουργία του socket ξεκινάει η διαδικασία της «σύνδεσης» με τον client, όπου ο server αναμένει για εισερχόμενες συνδέσεις. Όταν ο client συνδεθεί, δημιουργείται ένα νέο αντικείμενο τύπου socket που αντιπροσωπεύει την συγκεκριμένη σύνδεση. Το νέο αυτό socket είναι διαφορετικό από το αρχικό που χρησιμοποιείται για την υποδοχή νέων συνδέσεων.

```

while True:
    #Establish connection with client
    conn, addr = s.accept()
    print('Got connection from', addr)

```

Εικόνα 36: Σύνδεση με τον client

Μετά την σύνδεση, ξεκινάει η προσπάθεια λήψης των δεδομένων. Αρχικά, στέλνεται το μέγεθος των δεδομένων που πρόκειται να σταλούν. Αν και το μέγεθος δεν χρησιμοποιείται στο πρόγραμμα, δείχνει πως η επικοινωνία μεταξύ client – server έχει ξεκινήσει και πως δεν υπάρχει κάποιο πρόβλημα. Η λήψη των δεδομένων γίνεται με την συνάρτηση recv(). Η παράμετρος της recv() ορίζει το μέγιστο μέγεθος (σε bytes) των δεδομένων που θα παραληφθούν με την μία. Μετά το μέγεθος, στέλνεται η πληροφορία της εικόνας. Η αποστολή – λήψη εκτελείται επαναληπτικά, μέχρι να ληφθεί η συμβολοσειρά «[END OF IMAGE]» η οποία έχει οριστεί να υποδηλώνει το τέλος της εικόνας. Το παραπάνω επιτυγχάνεται με την συνάρτηση endswith(), η οποία επιστρέφει True εάν συναντήσει στο τέλος των δεδομένων που στέλνονται κάθε φορά την συγκεκριμένη συμβολοσειρά. Τα δεδομένα της εικόνας προσαρτώνται κάθε φορά σε έναν πίνακα ο οποίος στο τέλος περιέχει όλες τις πληροφορίες που έχουν σταλεί, εκτός της συμβολοσειράς τερματισμού. Αυτό επιτυγχάνεται με την χρήση της συνάρτησης replace(), η οποία αντικαθιστά την

συμβολοσειρά τερματισμού με το κενό (όχι τον κενό χαρακτήρα), έτσι ώστε να μείνουν μόνο τα δεδομένα της εικόνας.

```
try:
    final_data = b"" #Will be used to append the image data

    #Get buffer size
    size = conn.recv(1024)
    str_size = str(size, encoding = 'utf-8')
    int_size = int(size)
    print(int_size)

    #Get image data
    while True:
        try:
            img_data = conn.recv(4096)
        except:
            break
        binary_array = bytes(img_data)
        if(img_data.endswith(b'[END OF IMAGE]')):
            final_data += binary_array.replace(b'[END OF IMAGE]', b'')
            break
        else:
            final_data += binary_array
```

Εικόνα 37: Λήψη δεδομένων

Με την επιτυχή λήψη των δεδομένων και την κατασκευή του πίνακα που περιέχει τις πληροφορίες της εικόνας, είναι πλέον δυνατή η ανακατασκευή της. Εφ' όσον τα δεδομένα έχουν σειριοποιηθεί με το module Pickle, για την ανάγνωσή τους απαιτείται η χρήση της συνάρτησης loads(), με την οποία διαβάζονται τα δεδομένα και επιστρέφεται το ανακατασκευασμένο αντικείμενο. Η συνάρτηση frombytes() από το module PIL δημιουργεί ένα αντίγραφο εικόνας από δεδομένα σε buffer. Στην απλή μορφή της δέχεται τρεις παραμέτρους. Η πρώτη αφορά τον τύπο της εικόνας, η δεύτερη το μέγεθος και η τρίτη είναι ο buffer. Η εικόνα έπειτα αποθηκεύεται.

```
#Save reconstructed image
img = Image.frombytes('RGB', (640, 480), bytes(pickle.loads(final_data)))
img.save('image.png')
print("Got image")
```

Εικόνα 38: Ανακατασκευή εικόνας

Στην συνέχεια καλείται η συνάρτηση `emotion_reco()` που θα καλέσει το DeepFace για την ανάλυση του συναισθήματος. Η συνάρτηση επιστρέφει το κυρίαρχο συναίσθημα που ανιχνεύθηκε.

```
#Get emotion
detected_emotion = tf_emotion.emotion_reco()
print("TF returned: " + detected_emotion)
```

Εικόνα 39: Αναγνώριση συναισθήματος

Στο DeepFace, η ανάλυση της εικόνας γίνεται με την χρήση της συνάρτησης `analyze()`, η οποία δέχεται ως ορίσματα μία εικόνα και ένα «action» το οποίο αντιστοιχεί στην λειτουργία που θα εκτελέσει η ανάλυση (εδώ έχει επιλεγεί το συναίσθημα). Αν η εικόνα δεν περιέχει πρόσωπο, το DeepFace επιστρέφει κενή συμβολοσειρά και ενημερώνει τον χρήστη ότι δεν μπόρεσε να ανιχνεύσει κάποιο πρόσωπο. Διαφορετικά, επιστρέφονται τα έξι συναισθήματα με το ποσοστό αναγνώρισής τους. Έπειτα, επιλέγεται το κυρίαρχο συναίσθημα.

```
def emotion_reco():
    obj = DeepFace.analyze(img_path = "image.png", actions = ['emotion']) #analyze emotion from image
    print("ran tf") #sanity check
    print(obj) #sanity check
    print("TF detected: " + obj["dominant_emotion"]) #sanity check
    return obj["dominant_emotion"] #return detected emotion
```

Εικόνα 40: Ανάλυση συναισθήματος στο DeepFace

Μετά την ανάλυση του συναισθήματος το αποτέλεσμα στέλνεται στο NAO με την συνάρτηση `send()`. Η συνέχεια του προγράμματος περιλαμβάνει διαχείριση λαθών και το κλείσιμο της σύνδεσης και του socket όταν αυτό δεν χρειάζεται πλέον.

```
#Send data to NAO
conn.send(bytes(detected_emotion, encoding='utf8'))
```

Εικόνα 41: Αποστολή δεδομένων στο NAO

Εμφάνιση και διαχείριση αποτελεσμάτων

Η εικόνα παραμένει αποθηκευμένη στον υπολογιστή μέχρι να διαγραφεί από την επόμενη. Αν και στην πραγματικότητα το ρομπότ δεν δηλώνει ρητά το αποτέλεσμα που έλαβε, η αντίδρασή του εξαρτάται κάθε φορά από την απάντηση του DeepFace.

Όπως αναφέρθηκε παραπάνω όμως, το κυρίαρχο συναίσθημα είναι ορατό τόσο στην κονσόλα του server όσο και στα αρχεία καταγραφής συμβάντων (logger) του ρομπότ, για λόγους αποσφαλμάτωσης.

3.3 Παράδειγμα λογισμικού

Παρακάτω παρατίθενται μερικές ενδεικτικές εικόνες από πειραματική δοκιμή της εργασίας μαζί με τα αποτελέσματά τους. Η δοκιμή έγινε με τον simulator.

Εικόνα 41α) Χαρά



```
{'emotion': {'angry': 1.7636442691647889e-06, 'disgust': 2.148029972305877e-09, 'fear': 5.50025702238699e-07, 'happy': 99.5018779934572, 'sad': 1.832320573460457e-05, 'surprise': 2.3581258352677513e-07, 'neutral': 0.4981016105772025}, 'dominant_emotion': 'happy'}  
TF detected: happy
```

Εικόνα 42β) Αποτέλεσμα χαράς από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα

Εικόνα 43α) Ουδετερότητα



```
{'emotion': {'angry': 0.7036586292088032, 'disgust': 0.003473284959909506, 'fear': 0.27729603461921215, 'happy': 6.050382182002068, 'sad': 1.095475908368826, 'surprise': 0.027734163450077176, 'neutral': 91.84198379516602}, 'dominant_emotion': 'neutral'}  
TF detected: neutral
```

Εικόνα 42β) Αποτέλεσμα ουδετερότητας από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα

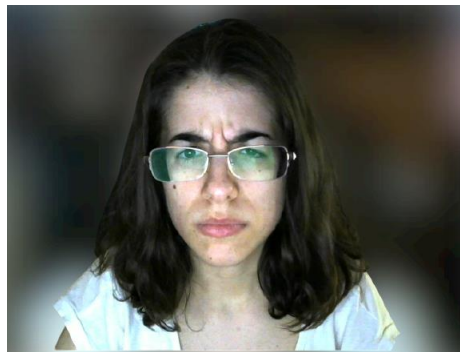
Εικόνα 43α) Λύπη



```
{'emotion': {'angry': 0.4241608548909426, 'disgust': 0.16451303381472826, 'fear': 14.973536133766174, 'happy': 1.0028486140072346, 'sad': 74.38947558403015, 'surprise': 0.11628206120803952, 'neutral': 8.929184079170227}, 'dominant_emotion': 'sad'}  
TF detected: sad
```

Εικόνα 44β) Αποτέλεσμα λύπης από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα

Εικόνα 44α) Θυμός



```
{'emotion': {'angry': 99.99325275421143, 'disgust': 0.0016782671082182787, 'fear': 0.0003816972821368836, 'happy': 2.6130603215067383e-11, 'sad': 0.003346320954733528, 'surprise': 4.7401318981066254e-11, 'neutral': 0.0013427776138996705}, 'dominant_emotion': 'angry'}  
TF detected: angry
```

Εικόνα 45β) Αποτέλεσμα θυμού από το DeepFace. Το DeepFace αξιολογεί την εικόνα με βάση τα έξι βασικά συναισθήματα (χαρά, λύπη, θυμός, φόβος, αηδία, έκπληξη) και την ουδετερότητα και επιστρέφει το κυριότερο συναίσθημα

4. ΑΞΙΟΛΟΓΗΣΗ

Η εφαρμογή δοκιμάστηκε στο EcoFest 2021 που πραγματοποιήθηκε στην Αθήνα. Στην εκδήλωση, ζητήθηκε από τους παρευρισκόμενους να κάνουν μία έκφραση με το πρόσωπό τους μπροστά από την κάμερα του NAO, να το αφήσουν να κάνει την αξιολόγησή του και έπειτα να δηλώσουν ποιο από τα επτά συναισθήματα (χαρά, λύπη, ουδετερότητα, θυμό, φόβο, αηδία ή έκπληξη) ήθελαν να αποτυπώσουν. Οι συμμετέχοντες, είκοσι πέντε στον αριθμό, επιλέχθηκαν τυχαία και περιλάμβαναν άντρες, γυναίκες και παιδιά όλων των ηλικιών. Οι περισσότεροι έλαβαν γνώση του αποτελέσματος και συζητήθηκε μαζί τους η φωτογραφία τους όταν το επιστρεφόμενο συναίσθημα δεν ήταν το επιθυμητό. Αποδεκτές απαντήσεις θεωρήθηκαν οι αρχικά σωστές και οι σχετικές με την έκφραση, ακόμα και αν δεν ήταν το αρχικό συναίσθημα που επιθυμούσε ο χρήστης.

Ο server υποστηρίχθηκε σε laptop με τα παρακάτω χαρακτηριστικά:

CPU Intel Core i7-1065G7 @ 1.30GHz

RAM 16 GB

GPU NVIDIA GeForce MX230

OS Windows 10 Home 64 bit, 20H2 version, 19042.985 build

Η αξιολόγηση του συστήματος έγινε ως προς δύο κύρια χαρακτηριστικά: το ποσοστό επιτυχίας αναγνώρισης συναισθήματος και την ταχύτητα απόκρισης/αναγνώρισης.

Η εφαρμογή είχε υψηλά ποσοστά επιτυχίας. Από τους είκοσι πέντε ερωτηθέντες, πέντε απαντήσεις θεωρήθηκαν λανθασμένες. Παρατηρήθηκε ότι ο φωτισμός τόσο του χώρου όσο και του προσώπου ήταν πολύ σημαντικός παράγοντας για τα αποτελέσματα καθώς πολλές φορές η διαδικασία επαναλήφθηκε λόγω κακού ή υπερβολικού φωτισμού και απέφερε καλύτερα αποτελέσματα.

Ένα μεγάλο μέρος των συμμετεχόντων δήλωνε αρχικά πως ήθελε να εκφράσει ένα συγκεκριμένο συναίσθημα και πολλές φορές το ρομπότ τους απαντούσε πως ανίχνευε κάποιο άλλο. Κατόπιν εξέτασης του στιγμιότυπου λήψης από τον συμμετέχοντα, το συμπέρασμα του ρομπότ ήταν δικαιολογημένο, καθώς πολλές φορές ο συμμετέχοντας δεν είχε την σωστή έκφραση, δηλαδή αντί για λύπη έδειχνε

χαρά ή η έκφρασή του μπορούσε να παρερμηνευθεί. Για παράδειγμα, ο φόβος και η αηδία ή ο φόβος και η έκπληξη συγχέονταν αρκετά και πολλές φορές δεν ήταν ξεκάθαρο ούτε και για το ανθρώπινο μάτι. Ακόμη, σημειώθηκε πολλές φορές ότι ο συμμετέχων δεν διατηρούσε την έκφραση καθ' όλη την διάρκεια της λήψης με αποτέλεσμα το επιστρεφόμενο συναίσθημα να ταιριάζει με την λήψη αλλά όχι με τις αρχικές διαθέσεις του προσώπου.

Ως προς την ταχύτητα, η μετάδοση των δεδομένων από και προς το ρομπότ δεν παρουσιάζει κάποιο πρόβλημα. Η ανάλυση της εικόνας πραγματοποιείται αρκετά γρήγορα ώστε να μην παρουσιάζεται πρόβλημα στην αλληλεπίδραση του ανθρώπου με το ρομπότ κατά την διάρκεια της συζήτησης, με εξαίρεση την πρώτη φορά που εκτελείται το πρόγραμμα, καθώς πρέπει να πραγματοποιηθούν ορισμένες λειτουργίες πριν ξεκινήσει η διαδικασία αναγνώρισης. Σε γενικές γραμμές, ο χρόνος αναμονής είναι κάτω των 10 δευτερολέπτων για μία ολοκληρωμένη συζήτηση. Η ταχύτητα όμως εξαρτάται και από τα χαρακτηριστικά του υπολογιστή στον οποίο εκτελείται ο server.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Επίλογος

Στην παρούσα εργασία, η κατανόηση γνώσεων που αφορούν δομές και αρχές σχεδίασης νευρωνικών δικτύων καθώς και συνελκτικά νευρωνικά δίκτυα είναι πολύ βασικό ακόμα και αν δεν απαιτείται η σχεδίαση ενός τέτοιου δικτύου, καθώς βοηθάει στο να γίνουν κατανοητές έννοιες και παράγοντες που μπορούν να επηρεάσουν την ακρίβεια των δεδομένων ή ακόμα και να δικαιολογήσουν τα αποτελέσματα. Μερικές τέτοιες έννοιες είναι η συνάρτηση ενεργοποίησης, τα φίλτρα, η σύνδεση μεταξύ των νευρώνων και άλλα. Στην διπλωματική εργασία, παρουσιάζονται όλες οι χρησιμοποιούμενες τεχνολογίες και περιβάλλοντα ανάπτυξης, όπως η μηχανική μάθηση και το ρομπότ NAO, καθώς και όλη η λειτουργικότητα μέσα από διαγράμματα ροής. Επίσης, παρατίθενται και όλα τα απαραίτητα τμήματα κώδικα που αναπτύχθηκαν κατά την διάρκεια εκπόνησης της εργασίας.

Σχετικά με το ρομπότ NAO, δείχνει πολλά υποσχόμενο ως προς τον τομέα της ψυχαγωγίας και της εκπαίδευσης - ιδιαίτερα στα μικρά παιδιά – όπως παρατηρήθηκε στο φεστιβάλ. Είτε λόγω της ανθρωπόμορφης εμφάνισής του είτε λόγω του μεγάλου εύρους δυνατοτήτων που παρουσιάζει ως προς την συνομιλία με τον άνθρωπο και όχι μόνο, είναι μία καλή επιλογή για την δοκιμή της ανάλυσης συναισθημάτων, καθώς οι άνθρωποι έδειξαν μεγάλη προθυμία να συνεργαστούν μαζί του. Το DeepFace χρησιμοποιήθηκε γιατί εμφάνισε μεγάλα ποσοστά επιτυχίας όταν δοκιμάστηκε μόνο τους ως API και είχε ήδη έτοιμα εκπαιδευμένα μοντέλα, γεγονός που ανταποκρινόταν στις απαιτήσεις της εργασίας. Με την ανάπτυξη της εργασίας αυτής και την ολοκλήρωση μία επιτυχημένης αρχιτεκτονικής client-server, μπορούν να χρησιμοποιηθούν και άλλα μοντέλα ή έτοιμα frameworks/APIs ή ακόμα και να κατασκευαστούν νέα, που θα εστιάζουν σε συγκεκριμένες απαιτήσεις και θα επιφέρουν καλύτερα αποτελέσματα ή θα αναλύουν επιπλέον συναισθήματα.

Στην εργασία, ένας άνθρωπος μπορεί να συνομιλήσει με το NAO και να το αφήσει να πραγματοποιήσει την λήψη μίας φωτογραφίας του την οποία και θα στείλει σε κοντινό υπολογιστή. Η εικόνα αυτή θα χρησιμοποιηθεί ως είσοδος σε συνελκτικό νευρωνικό δίκτυο που έχει ήδη εκπαιδευτεί ώστε να γίνουν προβλέψεις (predict)

σχετικά με το συναίσθημα που απεικονίζεται. Τα συναισθήματα είναι επτά: χαρά, φόβος, λύπη, αηδία, έκπληξη, θυμός και ουδετερότητα.

5.2 Περιορισμοί

Καθ' όλη τη διάρκεια της εκπόνησης της εργασίας παρατηρήθηκαν κάποιοι περιορισμοί και επιφυλάξεις που καταγράφονται παρακάτω.

Η εργασία αναπτύχθηκε κατά κύριο λόγο στο πρόγραμμα προσομοίωσης που διατίθεται μαζί με το Choregraphe. Αν και δεν παρουσιάστηκε κάποιο πρόβλημα στην προσομοίωση κατά την μεταφορά δεδομένων, στο πραγματικό ρομπότ υπήρξαν δυσκολίες. Οι συναρτήσεις `send()` και `sendall()` δεν έστελναν όλα τα δεδομένα της εικόνας στον υπολογιστή, γι' αυτό και αποφασίστηκε η προσέγγιση με την συμβολοσειρά τερματισμού «[END OF IMAGE]».

Κατά την πρώτη φορά εκτέλεσης του προγράμματος, η αναγνώριση του συναισθήματος διαρκεί λίγο περισσότερο απ' ότι συνήθως καθώς το πρόγραμμα πρέπει να τρέξει τις απαραίτητες διαδικασίες πριν ξεκινήσει την αναγνώριση. Επομένως, είναι σημαντικό να υπάρχει κατάλληλο σύστημα που να μπορεί να υποστηρίξει την αρχιτεκτονική με ευκολία ώστε να μειωθεί τόσο ο χρόνος πρώτης εκτέλεσης όσο και ο γενικός χρόνος αναγνώρισης συναισθημάτων.

Εκτός από μία κακή γωνία λήψης όπου δεν φαίνονται πλήρως τα χαρακτηριστικά του προσώπου, ένας από κυριότερους παράγοντες που μπορεί να αλλοιώσει τα αποτελέσματα είναι το φως. Ιδανικά, το πρόσωπο πρέπει να φωτίζεται αρκετά από όλες τις πλευρές καθώς είναι επιθυμητό να μην δημιουργούνται σκιές, χωρίς όμως υπερβολές. Ακόμη, το φως του ήλιου είναι πολύ δυνατό για την κάμερα του NAO και δεν θα πρέπει να πέφτει κατευθείαν επάνω, καθώς γίνεται αδύνατη η εξαγωγή των χαρακτηριστικών από την εικόνα. Ακόμη, είναι πολύ σημαντικό να υπάρχει μόνο ένα άτομο την φορά μέσα στο καρέ, καθώς το νευρωνικό δίκτυο δεν μπορεί να κάνει ταυτόχρονη ανάλυση.

Στις εργαστηριακές δοκιμές καθώς και στον προσομοιωτή το ρομπότ ανταποκρίνονταν αμέσως, τόσο κατά την φόρτωση του προγράμματος όσο και κατά την εκτέλεση. Στην δοκιμή της εφαρμογής στο EcoFest, παρατηρήθηκε η αντίθετη

συμπεριφορά. Επειδή δεν υπάρχει αιτιολόγηση για αυτό το συμβάν, μπορεί μερικές φορές το ρομπότ να θεωρηθεί αναξιόπιστο.

5.3 Μελλοντικές επεκτάσεις

Μερικές μελλοντικές επεκτάσεις που μπορούν να πραγματοποιηθούν είναι:

- Φόρτωση και εκτέλεση του κώδικα αναγνώρισης συναισθημάτων σε Jetson Nano

Το Jetson Nano σχεδιάζεται και κατασκευάζεται από την NVIDIA και είναι ένας μικρός υπολογιστής. Είναι ειδικά σχεδιασμένο για να καλύπτει τις ανάγκες προγραμματιστών και ερευνητών καθώς οι προδιαγραφές του επιτρέπουν την ταυτόχρονη εκτέλεση πολλών νευρωνικών δικτύων, γεγονός που το καθιστά ιδανικό για εφαρμογές βαθιάς μάθησης ή όρασης υπολογιστών. Το μικρό του μέγεθος προσφέρει εύκολη μεταφορά και εκτέλεση των προγραμμάτων κοντά στην τοποθεσία που απαιτείται (edge computing) με αποτέλεσμα να μειώνεται ο χρόνος απόκρισης και να γίνεται εξοικονόμηση του εύρους ζώνης (bandwidth). Στην προκειμένη περίπτωση, δεν θα χρειαζόταν εξωτερικός υπολογιστής κοντά στο NAO για να εκτελεστεί η αναγνώριση των συναισθημάτων, καθώς το Jetson Nano θα μπορούσε να προσαρτηθεί επάνω στο ίδιο το ρομπότ, γεγονός που προσφέρει ελευθερία κινήσεων στο NAO και προοπτικές για χρήση του σε όλους τους τομείς που έχουν αναφερθεί, χωρίς να χρειάζεται η ανάπτυξη του κατάλληλου περιβάλλοντος κάθε φορά.

- Ελληνικά δεδομένα (datasets) για εκπαίδευση του νευρωνικού δικτύου στην αναγνώριση συναισθημάτων μέσω ομιλίας

Ο συνδυασμός εικόνας και ήχου μπορεί να οδηγήσει σε αποτελέσματα με μεγαλύτερη ακρίβεια, όταν για παράδειγμα η έκφραση του προσώπου του χρήστη είναι διφορούμενη. Η χρήση αγγλικών dataset συνεπάγεται ότι ο χρήστης πρέπει να μιλάει αγγλικά, γεγονός που κάνει την αναγνώριση συναισθημάτων μέσω φωνής να υστερεί, καθώς δεν είναι η μητρική του γλώσσα. Αυτό μπορεί να επεκταθεί και για άλλες γλώσσες.

- Κατασκευή και χρήση άλλων μοντέλων

Τα μοντέλα που χρησιμοποιούνται επιταχύνουν την διαδικασία καθώς είναι έτοιμα και δεν χρειάζονται εκπαίδευση. Η ανάπτυξη και χρήση νέων μοντέλων ειδικά κατασκευασμένων για την περίπτωση θα βελτιώσει την απόδοση και τα αποτελέσματα, καθώς θα μπορεί να γίνει εστίαση σε χαρακτηριστικά συγκεκριμένα για την εργασία, χωρίς να λαμβάνονται υπόψη τυχόν «σχόλια» (annotations) από τα ήδη εκπαιδευμένα που επιβραδύνουν τους υπολογισμούς. Αυτό σημαίνει ότι θα είναι καταλληλότερο για χρήση σε γενικότερο διάλογο με ανθρώπους, όπου χρειάζεται ταχύτητα στις απαντήσεις ή σε καταστάσεις υπό πίεση χρόνου, όταν για παράδειγμα κάποιος ασθενής αδυνατεί να επικοινωνήσει και χρήζει βοήθειας. Θα μπορούσε με κάποια συγκεκριμένη έκφραση του προσώπου να ειδοποιείται άμεσα το ιατρικό προσωπικό. Ακόμη, θα μπορούσε να χρησιμοποιηθεί στην εκπαίδευση για διαμόρφωση γνωστικών καταστάσεων με φιλτράρισμα πληροφοριών καθώς θα μπορούσαν να χρησιμοποιηθούν συγκεκριμένα annotations που να εξυπηρετούν το σύστημα. (Automated reasoning of learners' cognitive states using classification analysis, Christos Troussas et al.) [19] (Redesigning teaching strategies through an information filtering system, Christos Troussas et al.) [20].

- Χρήση πολλαπλών μοντέλων

Η χρήση πολλαπλών μοντέλων που υιοθετούν διαφορετικές τεχνικές (Bootstrap aggregating, boosting και άλλα) και κατά συνέπεια η χρήση του μέσου όρου των εξόδων τους μπορεί να συμβάλλει στην λήψη καλύτερων αποτελεσμάτων. Η ανακρίβεια στα αποτελέσματα μπορεί να δημιουργήσει προβλήματα στην συνομιλία με το λιγότερο να είναι μία απλή σύγχυση ή ο χρήστης να χάσει το ενδιαφέρον του, αλλά ταυτόχρονα καθιστά το ρομπότ ακατάλληλο για χρήση σε πιο ευαίσθητες διαδικασίες, όπως επισκέψεις σε νοσοκομεία ή χρήση από την αστυνομία για ανάλυση ενός υπόπτου υπό πίεση. Ακόμη, η ακρίβεια στα αποτελέσματα μπορεί να εξυπηρετήσει προσαρμοστικά συστήματα ηλεκτρονικής μάθησης (adaptive-E learning systems) [18] ή ως γενικό εργαλείο στην εκπαίδευση (Usability and Educational Affordance of Web 2.0 tools from Teachers' Perspectives, Akrivi Krouska et al.) [27], καθώς το NAO θα μπορεί να δημιουργήσει το κάθε προφίλ του χρήστη ξεχωριστά και με ακρίβεια.

- SpeechRecognition 3.8.1 & Chatter Bot API

Μεγάλη βελτίωση στην εργασία μπορεί να προσφέρει η προσθήκη της βιβλιοθήκης SpeechRecognition. Όπως αναφέρεται και από το όνομά της, η βιβλιοθήκη πραγματοποιεί αναγνώριση φωνής χρησιμοποιώντας άλλες υπηρεσίες, όπως η [Google speech recognition](#), [IBM speech to text](#), [Wit.ai](#) και άλλες. Η αναγνώριση αυτή επιτυγχάνεται ανεβάζοντας αρχεία ήχου στις υπηρεσίες. Το ChatterBot API δίνει την δυνατότητα στον χρήστη να έχει πρόσβαση σε ChatterBots, τα οποία μπορούν να απαντήσουν σε ερωτήσεις του χρήστη χωρίς να είναι προσχεδιασμένος ο διάλογος. Τα παραπάνω μπορούν να αξιοποιηθούν ώστε να υπάρχει ολοκληρωμένη συζήτηση μεταξύ χρήστη και ρομπότ, χωρίς να υπάρχει κάποιου είδους σενάριο ή συγκεκριμένες απαντήσεις από το ρομπότ, προβλέποντας έτσι και σενάρια χρήσης σε σχολεία, όπου οι απαντήσεις των παιδιών μπορεί να είναι απρόβλεπτες. Για παράδειγμα, στο «A Personalized Brain-Based Quiz Game for Improving Students' Cognitive Functions» (Akrivi Krouska et al.) [17], η χρήση του NAO με το speech recognition θα μπορούσε να βοηθήσει στην ανάλυση της συναισθηματικής κατάστασης του χρήστη και να διαπιστώσει το επίπεδο της γνώσης του και τους στόχους του, κάνοντας την διαδικασία ευχάριστη και προσφιλή για τον χρήστη. Συνδυάζοντας την αναγνώριση συναισθήματος, οι απαντήσεις και η συμπεριφορά του NAO μπορούν να ανταποκρίνονται στην συναισθηματική κατάσταση του χρήστη. Ένα ρομπότ που μπορεί ν' ανταποκρίνεται στην συναισθηματική κατάσταση του χρήστη μπορεί να χρησιμοποιηθεί σε παρηγορητική φροντίδα ασθενών με καρκίνο ή άλλων ασθενειών. Συγκεκριμένα, μπορεί να δείξει εμπάθεια ή να είναι πιο προσφιές εξαιτίας της μορφής του, το οποίο συνεπάγεται ότι οι ασθενείς θα ανοίγονται ευκολότερα. Επίσης, θα μπορεί να επισκέπτεται τους ασθενείς οποιαδήποτε ώρα χωρίς να επιβαρύνεται το ιατρικό προσωπικό, το οποίο μπορεί και να μην επαρκεί και ταυτόχρονα απαλλάσσει από μία ψυχοφθόρα διαδικασία που αρκετοί άνθρωποι μπορεί να μην αντέχουν.

ΠΑΡΑΠΟΜΠΕΣ

- [1] Γεωργούλη, Α., 2015. Τεχνητή νοημοσύνη. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3381>
- [2] Διαφάνειες μαθήματος [όρασης υπολογιστών](#) (ICE-8111) κεφάλαια 2,3 και 8. Βασισμένες στο μάθημα 16-385 Computer Vision, Carnegie Mellon Spring 2020
- [3] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου. Τεχνητή Νοημοσύνη - Γ' Έκδοση, ISBN: 978-960-8396-64-7 σελ. 336, 373-378
- [4] "Facial Emotional Recognition With Deep Learning On Pepper Robot" Hu, Mengge. Διαθέσιμο στο: <http://urn.fi/URN:NBN:fi:amk-2019052611938>
- [5] T. Dettmers, "Understanding Convolution in Deep Learning," 26 03 2015. Διαθέσιμο στο: <http://timdettmers.com/2015/03/26/convolution-deep-learning/>
- [6] Μαγκούτης, Κ., Νικολάου, Χ., 2015. *Εισαγωγή στον αντικειμενοστροφή προγραμματισμό με Python*. Σελ. 254 [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/1708>
- [7] "SOFTBANK ROBOTICS DOCUMENTATION," SOFTBANK ROBOTICS. Διαθέσιμο στο: http://doc.aldebaran.com/2-1/home_nao.html
- [8] "What is CUDA?", NVIDIA. Διαθέσιμο στο: <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/>
- [9] "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", Yaniv Taigman, Ming Yang, Marc' Aurelio Ranzato, Lior Wolf. Διαθέσιμο στο: <https://paperswithcode.com/paper/deepface-closing-the-gap-to-human-level-1>
- [10] "The what's what of Keras and TensorFlow", Reetiesh Chandra Διαθέσιμο στο <https://www.upgrad.com/blog/the-whats-what-of-keras-and-tensorflow/>
- [11] "NAOqi Framework". Διαθέσιμο στο: <http://doc.aldebaran.com/1-14/dev/naoqi/index.html>

- [12] “NAOqi Python API”. Διαθέσιμο στο: <http://doc.aldebaran.com/2-4/ref/python-api.html>
- [13] “ALVideoDevice”. Διαθέσιμο στο: <http://doc.aldebaran.com/2-4/naoqi/vision/alvideodevice.html#alvideodevice>
- [14] “Learn Choregraphe – Python in Choregraphe: Special Topics”
<https://docs.google.com/presentation/d/1tcSdyp1q0CVJGNbuzqgjpC6Oco8uQ6sCplgLBWqJWdo/edit?pli=1#slide=id.p1>
- [15] “ALDialog”. <https://developer.softbankrobotics.com/nao6/naoqi-developer-guide/naoqi-apis/naoqi-interaction-engines/aldialog#aldialog>
- [16] DeepFace, κατασκευή νευρωνικού δικτύου.
<https://github.com/serengil/deepface/blob/master/deepface/extendedmodels/Emotion.py>
- [17] Akrivi Krouska, Christos Troussas, Cleo Sgouropoulou: A Personalized Brain-Based Quiz Game for Improving Students' Cognitive Functions. BFAL 2020: 102-106
- [18] Christos Troussas, Cleo Sgouropoulou: Innovative Trends in Personalized Software Engineering and Information Systems - The Case of Intelligent and Adaptive E-learning Systems. Frontiers in Artificial Intelligence and Applications 324, IOS Press 2020, ISBN 978-1-64368-096-5, pp. 1-96
- [19] Christos Troussas, Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou, Ioannis Voyiatzis: Automated reasoning of learners' cognitive states using classification analysis. PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 103–106.
<https://doi.org/10.1145/3437120.3437285>
- [20] Christos Troussas, Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou, Ioannis Voyiatzis: Redesigning teaching strategies through an information filtering system. PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 111-114.
<https://doi.org/10.1145/3437120.3437287>
- [21] C. Troussas, A. Krouska, E. Alepis & M. Virvou (2021) Intelligent and adaptive tutoring through a social network for higher education, New Review of Hypermedia and Multimedia, DOI: 10.1080/13614568.2021.1908436

- [22] Troussas C., Krouska A., Sgouropoulou C. (2020) Towards a Reference Model to Ensure the Quality of Massive Open Online Courses and E-Learning. In: Frasson C., Bamidis P., Vlamos P. (eds) Brain Function Assessment in Learning. BFAL 2020. Lecture Notes in Computer Science, vol 12462. Springer, Cham. https://doi.org/10.1007/978-3-030-60735-7_18
- [23] Troussas C., Krouska A., Sgouropoulou C. (2020) Dynamic Detection of Learning Modalities Using Fuzzy Logic in Students' Interaction Activities. In: Kumar V., Troussas C. (eds) Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science, vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030-49663-0_24
- [24] Papakostas C., Troussas C., Krouska A., Sgouropoulou C.. Measuring User Experience, Usability and Interactivity of a Personalized Mobile Augmented Reality Training System. Sensors. 2021; 21(11):3888. <https://doi.org/10.3390/s21113888>
- [25] Krouska A., Troussas C. and Sgouropoulou C. 2019. Fuzzy Logic for Refining the Evaluation of Learners' Performance in Online Engineering Education. European Journal of Engineering and Technology Research. 4, 6 (Jun. 2019), 50-56. DOI: <https://doi.org/10.24018/ejers.2019.4.6.1369>
- [26] Papakostas, C., Troussas, C., Krouska, A. & Sgouropoulou, C. User acceptance of augmented reality welding simulator in engineering training. Educ Inf Technol (2021). <https://doi.org/10.1007/s10639-020-10418-7>
- [27] Akrivi Krouska, Christos Troussas, Cleo Sgouropoulou: Usability and Educational Affordance of Web 2.0 tools from Teachers' Perspectives. PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 107-110. <https://doi.org/10.1145/3437120.3437286>
- [28] Krouska A., Troussas C., Sgouropoulou C. (2020) Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments. In: Kumar V., Troussas C. (eds) Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science, vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030-49663-0_23

- [29] F. Giannakas, C. Troussas, I. Voyiatzis, C. Sgouropoulou, A deep learning classification framework for early prediction of team-based academic performance, *Applied Soft Computing*, Volume 106, 2021, 107355. <https://doi.org/10.1016/j.asoc.2021.107355>.
- [30] Christos Troussas, Filippos Giannakas, Cleo Sgouropoulou & Ioannis Voyiatzis (2020) Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM, *Interactive Learning Environments*, DOI: [10.1080/10494820.2020.1761835](https://doi.org/10.1080/10494820.2020.1761835)