|  | Εθνικό Μετσόβιο Πολυτεχνείο  Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ |
| --- | --- |

**ΑΝΑΛΥΣΗ ΣΥΜΜΟΡΦΩΣΗΣ ΣΕ ΚΙΝΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των  
ΚΑΝΔΥΛΑΚΗ Η. ΣΤΥΛΙΑΝΟΥ  
ΟΡΦΑΝΟΠΟΥΛΟΥ Β. ΧΡΗΣΤΟΥ

**Επιβλέπων :** Τσανάκας Παναγιώτης  
 Καθηγητής ΕΜΠ

**Συνεπιβλέπων:** Σιόλας Γεώργιος

μέλος ΕΔΙΠ

Αθήνα, Ιούνιος 2022

|  | Εθνικό Μετσόβιο Πολυτεχνείο  Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ |
| --- | --- |

**ΑΝΑΛΥΣΗ ΣΥΜΜΟΡΦΩΣΗΣ ΣΕ ΚΙΝΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των  
ΚΑΝΔΥΛΑΚΗ Η. ΣΤΥΛΙΑΝΟΥ  
ΟΡΦΑΝΟΠΟΥΛΟΥ Β. ΧΡΗΣΤΟΥ

**Επιβλέπων :** Τσανάκας Παναγιώτης  
 Καθηγητής ΕΜΠ

**Συνεπιβλέπων:** Σιόλας Γεώργιος

μέλος ΕΔΙΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την \*\* Ιουνίου 2021.

......................................

Τσανάκας Παναγιώτης  
Καθηγητής Ε.Μ.Π.......................................

Σταφυλοπάτης   
Ανδρέας-Παναγιώτης  
Καθηγητής Ε.Μ.Π.......................................

Κουτσουρής   
Δημήτριος-Διονύσιος  
Καθηγητής Ε.Μ.Π

|  | Εθνικό Μετσόβιο Πολυτεχνείο  Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ |
| --- | --- |



# Περίληψη

Στην εποχή μας, οι εφαρμογές (apps) αποτελούν αναπόσπαστο κομμάτι της καθημερινότητάς μας. Από τα μέσα κοινωνικής δικτύωσης (social media) έως τις εφαρμογές αυτοβελτίωσης, η χρήση τους είναι διαδεδομένη. Η ραγδαία εξέλιξη τόσο του υλισμικού (hardware) όσο και του λογισμικού (software) επιτρέπει πλέον σε απαιτητικές εφαρμογές να «τρέχουν» και στα προσωπικά μας έξυπνα τηλέφωνα (smartphones).

Ιδιαίτερα με την εμφάνιση του κορωνοϊού, το ενδιαφέρον για ιατρικές εφαρμογές αυξήθηκε, τόσο από τους επαγγελματίες γιατρούς όσο και από τους ασθενείς. Η εξ αποστάσεως επικοινωνία γίνεται πλέον αυτοσκοπός.

//Τα δικά μας

Λέξεις Κλειδιά

Ιατρικές Εφαρμογές, Μηχανική Μάθηση, Νευρωνικά Δίκτυα,

Πολυστρωματικά Δίκτυα Perceptron, Όραση Υπολογιστών, Android

# Abstract

Keywords

Machine Learning, Neural Networks, Multilayer Neural Networks, Computer Vision, Android

# Ευχαριστίες

# Περιεχόμενα

[**ΠερIληψη i**](#_heading=h.tyjcwt)

[**Abstract iii**](#_heading=h.3dy6vkm)

[**ΕυχαριστIες v**](#_heading=h.1t3h5sf)

[**ΠεριεXOμενα vii**](#_heading=h.4d34og8)

**1** [**ΕΙΣΑΓΩΓΗ 1**](#_heading=h.2s8eyo1)

**1.1** Σύγχρονη Κοινωνία και Ιατρικές Εφαρμογές **1**

**1.2** Αντικείμενο της εργασίας **2**

**1.3** Δομή της εργασίας **2**

**2 m-health** [**ΕΦΑΡΜΟΓΕΣ 3**](#_heading=h.3rdcrjn)

**2.1** Εισαγωγή **3**

**2.2** m-Health **3**

**2.3** Android - iOS **3**

[**3 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**](#_heading=h.3rdcrjn)

**3.1** [Εισαγωγή](#_heading=h.26in1rg)

**3.2** [Ιστορική Αναδρομή](#_heading=h.26in1rg)

**3.3** [Μηχανική Μάθηση](#_heading=h.26in1rg)

**3.4** Συνελικτικά Νευρωνικά Δίκτυα

**3.5**  Μοντέλα Εκτίμησης στάσης σώματος

[**4** Η εφαρμογη **physAi**](#_heading=h.3rdcrjn)

[**5 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ**](#_heading=h.3rdcrjn)

(used Tools and Technology)

[**6 ΑΠΟΤΕΛΕΣΜΑΤΑ**](#_heading=h.3rdcrjn)

**6.1**[Πειράματα](#_heading=h.1ksv4uv)

[**7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ**](#_heading=h.3rdcrjn)

**7.1** [Συμπεράσματα](#_heading=h.1ksv4uv)

**7.2**[Μελλοντικές Προεκτάσεις](#_heading=h.1ksv4uv)

Βιβλιογραφία

Παραπομπές

Ευρετήριο διαγραμμάτων

Ευρετήριο πινάκων

# Κεφάλαιο 1

Εισαγωγή

* 1. Εξελίξεις στις εξατομικευμένες ιατρικές εφαρμογές

Η πρόσφατη πανδημία του covid-19, κατέδειξε εμφανώς τις συνέπειες που επιφέρει η παραμέληση της ιατρικής παρακολούθησης των ασθενών, αλλά και η αναβολή τακτικών ιατρικών ελέγχων στο γενικό πληθυσμό. Λόγω των συνθηκών εγκλεισμού πλήθος ανθρώπων δεν ήταν σε θέση να λάβει την ιατρική περίθαλψη που χρειαζόταν[https://www.healthyagingpoll.org/reports-more/poll-extras/pandemic-disruptions-mean-many-older-adults-still-havent-gotten-needed]. Οι συνθήκες αυτές βέβαια αποτέλεσαν με τη σειρά τους εφαλτήριο για την εμφάνιση μιας ποικιλίας εφαρμογών υγείας, που αφορούν κυρίως την υποστήριξη της κατ’ οίκον ιατρικής και που στοχεύουν στη γενικότερη βελτίωση της σωματικής και ψυχολογικής υγείας νοσούντων και μη.

Βέβαια το ενδιαφέρον για εφαρμογές υγείας δεν είναι κάτι πρωτόγνωρο, αφού συγκεκριμένοι λόγοι συνέβαλαν καθοριστικά στην εξάπλωσή τους τις τελευταίες δεκαετίες. Παράγοντες κοινωνικοί, όπως οι έντονες δημογραφικές αλλαγές που παρατηρούνται, οικονομικοί, όπως το αυξημένο κόστος ιατροφαρμακευτικών υπηρεσιών και η παγκοσμιοποίηση των αγορών και τέλος πολιτικοί, λόγω της τάσης διεύρυνσης του κράτος πρόνοιας και των αντίστοιχων πιέσεων για περισσότερες κοινωνικές παροχές.

Είναι επιπλέον σχεδόν αυτονόητο ότι πάντοτε υπήρχε ανάγκη για συνέχιση των ιατρικών διαδικασιών και πέραν των νοσοκομειακών και λοιπών υγειονομικών εγκαταστάσεων, ιδίως κατά το στάδιο της αποκατάστασης, αλλά και δευτερευόντως κατά τη θεραπεία ενός προβλήματος. Ανάλογα με τη φύση της ασθένειας ο ασθενής καλείται να ακολουθήσει συγκεκριμένη διατροφή, να τηρεί ορισμένο πρόγραμμα ύπνου, να εκτελεί διάφορες ασκήσεις και να φροντίζει συνολικά την υγεία του. Αδιαμφισβήτητα όσο μεγαλύτερη είναι η συμμόρφωση του θεραπευομένου στις οδηγίες του θεραπευτή τόσο καλύτερη θα είναι και η θεραπεία. Ωστόσο σύμφωνα με πρόσφατες έρευνες [], μεγάλο ποσοστό ασθενών δεν ακολουθεί πιστά τις ιατρικές οδηγίες των επαγγελματιών. Αυτό συνεπώς δημιουργεί την ανάγκη για διαρκή και στενότερη ιατρική εποπτεία της προόδου του ασθενούς.

Σε προηγούμενες περιόδους η ιατρική επιτήρηση μπορούσε να επιτευχθεί σχεδόν αποκλειστικά μέσω της φυσικής παρουσίας θεραπευομένου και θεραπευτή. Ωστόσο λόγω του αυξανόμενου πληθυσμού με σωματικές παθήσεις, τις μεγαλύτερες αποστάσεις από δομές φροντίδας, αλλά και ενίοτε των υγειονομικών περιορισμών καθίσταται δύσκολη η μετακίνησή των ασθενών[https://pubmed.ncbi.nlm.nih.gov/18728716/]. Επομένως, κρίνεται απαραίτητη η αξιοποίηση των ψηφιακών τεχνολογικών δυνατοτήτων ως μέσο εποπτείας και ενίσχυσης του τρέχοντος συστήματος υγείας.

Καθοριστικός επιπλέον παράγοντας στη δημιουργία νέων ιατρικών εφαρμογών αποτελεί η τεράστια τεχνολογική εξέλιξη που συντελείται από την εμφάνιση του Διαδικτύου και έπειτα. Η πρόοδος στις ασύρματες επικοινωνίες και η συνεπακόλουθη εμφάνιση του Internet of Things συνέβαλε καθοριστικά στην διεύρυνση της ηλεκτρονικής υγείας. Η σταδιακή εξάπλωση του 5G, με τις αναβαθμισμένες ταχύτητες σύνδεσης, θα επιτρέψει ακόμα και στους πιο απομακρυσμένους οικισμούς να συνδέονται αποδοτικά και οικονομικά με τον απαιτούμενο ιατρικό πάροχο μέσω απλών εφαρμογών. Οι εφαρμογές αυτές εγκαθίστανται εύκολα σε έξυπνα κινητά (smartphones), ρολόγια (smartwatches) και πλήθος τερματικών, ενώ πλέον το μεγαλύτερο μέρος του πληθυσμού διαθέτει τις απαιτούμενες ψηφιακές δεξιότητες, γεγονός στο οποίο συνέβαλε και η υγειονομική κρίση.

Παράλληλα τη τελευταία δεκαετία σημειώνονται ραγδαίες εξελίξεις στον τομέα των νευρωνικών δικτύων. Εταιρίες κολοσσοί της τεχνολογικής βιομηχανίας, ερευνητές και Πανεπιστήμια έχουν δημιουργήσει μια πληθώρα εργαλείων και μοντέλων που καθιστούν πλέον εφικτή την οπτική αναγνώριση και κατηγοριοποίηση, σε πραγματικό χρόνο ακόμα και σε φορητές συσκευές με σχετικά περιορισμένη επεξεργαστική ισχύ.

Ενώ λοιπόν μέχρι πρότινος η εποπτεία κινητικών ασκήσεων φυσικοθεραπευτικού χαρακτήρα μπορούσε να πραγματοποιηθεί μόνο από κάποιον ειδικό, οι σύγχρονες εξελίξεις συνέβαλαν στη δημιουργία ψηφιακών εργαλείων αυτόματου εποπτείας. Ενώ μέχρι σήμερα ο κατ’οίκον έλεγχος των φυσικοθεραπειών ήταν ανέφικτος χωρίς την παρουσία ειδικού, πλέον δημιουργούνται νέες δυνατότητες για εξατομικευμένο ασύγχρονο έλεγχο. Σκοπός έτσι της παρούσας εργασίας είναι να καταδείξει ως εφικτή τη χρήση φορητών συσκευών για την αξιολόγηση κινητικών ασκήσεων χωρίς την ανάγκη άμεσης εποπτείας από ειδικό, καλύπτοντας έτσι ένα σημαντικό κενό στην κατ’ οίκων φυσικοθεραπεία που άφησαν οι υγειονομικοί περιορισμοί λόγω της πανδημίας.

* 1. Φυσικοθεραπεία τον καιρό της πανδημίας

1.2.1 Σε τι ωφελεί η γενικά φυσικοθεραπεία

Η φυσικοθεραπεία είναι ιδιαιτέρως ωφέλιμη ιατρική επιστήμη που ασχολείται με τους μυοσκελετικούς πόνους και με τη δυσκολία κίνησης του σώματος. Περιλαμβάνει ένα πλήθος εφαρμογών όπως:

* στην πρόληψη, αξιολόγηση, θεραπεία και αποκατάσταση καθώς και στην βελτίωση της απόδοσης του σωματικά δραστήριου ατόμου.
* στους ασθενείς με προβλήματα του αναπνευστικού συστήματος
* σε ασθενείς μετά από αγγειακό εγκεφαλικό επεισόδιο
* στην καρδιαγγειακή αποκατάσταση
* στην αντιμετώπιση του λεμφοιδήματος
* στην πρόληψη και η θεραπεία των λοιμώξεων του αναπνευστικού συστήματος και η κινητοποίηση του ασθενούς που βρίσκεται σε ΜΕΘ
* στην αντιμετώπιση της οστεοαρθρίτιδας
* στους ασθενείς με τη νόσο του πάρκινσον
* στην πρόληψη των πτώσεων σε άτομα τρίτης ηλικίας
* στην ρευματοειδή αρθρίτιδα

1.2.2 Η εξ αποστάσεως φυσικοθεραπεία

Αδιαμφισβήτητα η περίοδος της έξαρσης του κορωνοϊού δημιούργησε πολλά προβλήματα σε όλες τις ανθρώπινες δραστηριότητες, και ιδίως στον τομέα της υγείας, όπου σε πολλές περιπτώσεις δεν ήταν εφικτή η δια ζώσης θεραπευτική προσέγγιση. Ειδικά στο κομμάτι της φυσικοθεραπείας, που μπορεί ενδεχομένως να θεωρηθεί δευτερεύουσα ιατρική διαδικασία (αν και η ανάγκη για αναπνευστική αποκατάσταση ασθενών με covid-19 ηταν ιδαίτερα αυξημένη [<https://psf.org.gr/psf-news-6427.htm>]), τα υγειονομικά μέτρα δεν επέτρεπαν πολλές φορές τη φυσική παρουσία και επομένως η χρήση εξ αποστάσεως εναλλακτικών ήταν μονόδρομος.

Σύμφωνα με μελέτη που πραγματοποιήθηκε στην Ελβετία για τις φυσικοθεραπείες την περίοδο της πανδημίας [<https://archivesphysiotherapy.biomedcentral.com/articles/10.1186/s40945-021-00112-3>] το μεγαλύτερο μέρος των φυσικοθεραπευτών συνέχισε τις δραστηριότητές του χρησιμοποιώντας κυρίως εργαλεία τηλεδιασκέψεων ή/και απλού τηλεφώνου ενώ μικρό τμήμα χρησιμοποιήσε ειδικές εφαρμογές. Αυτό φυσικά είχε ως αποτέλεσμα ότι μόνο το 30% των φυσικοθεραπευτών δήλωσε ότι ασχολήθηκε με την εποπτεία ασκήσεων, με τους περισσότερους να επικεντρώνονται στην εκμάθηση ασκήσεων, ή στη συνέχιση θεραπειών και αποκατάστασης. Παρά ταύτα, τα συμπεράσματα στα οποία καταλήγει η ερεύνα είναι ότι παρά την ευρεία διαθεσιμότητα των ψηφιακών τεχνολογιών για τη διασύνδεση ασθενών με θεραπευτές, η εξ αποστάσεως φυσικοποθεραπεία δεν αποδείχθηκε τόσο ευεργετική. Πιο συγκεκριμένα η έρευνα καταδεικνύει την έλλειψη αντίστοιχης ψηφιακής εκπαίδευσης από μεριάς φυσικοποθεραπευτών, αλλά και την απουσία κατάλληλων εφαρμογών ως βασικούς παράγοντες για τη μειωμένη απόδοση.

Παρόμοια ή και ενδεχομένως πιο απαισιόδοξα συμπεράσματα θα αναμέναμε και στη χώρα μας, στην οποία η εξοικείωση με και η διαθεσιμότητα των ψηφιακών μέσων δεν είναι τόσο μεγάλη σε μεγαλύτερης ηλικίας άτομα. Παράλληλα λόγω της γεωγραφικής ιδιαιτερότητας της Ελλάδας, με τα πολλά ακριτικά νησιά και τα δυσπρόσιτα ορεινά χωριά, η ανάπτυξη της εξ αποστάσεως φυσικοθεραπείας βρίσκει επιπλέον εφαρμογές, πέραν των συνθηκών της πανδημίας.

Συμπερασματικά η πανδημία δημιούργησε πολλές προκλήσεις στον τομέα των φυσικοθεραπειών, δημιουργώντας μία αδήριτη ανάγκη για την ανάπτυξη εξειδικευμένων εφαρμογών που θα μπορούν να βοηθήσουν το έργο της εποπτείας και την καλύτερη συμμετοχή όλων των εμπλεκόμενων μερών.

* 1. Αντικείμενο της εργασίας

Λαμβάνοντας υπόψιν όλες τις σύγχρονες τεχνολογικές εξελίξεις που αναφέρθηκαν προηγουμένως, τις νέες τάσεις στο χώρο των ιατρικών εφαρμογών αλλά και αναλογιζόμενοι τα ψυχοσωματικά προβλήματα που προέκυψαν λόγω των περιοριστικών μέτρων για το περιορισμό της πανδημίας, θελήσαμε να υλοποιήσουμε μία νέα εφαρμογή που θα επιτρέπει την ψηφιακή ασύγχρονη εποπτεία και αξιολόγηση κινητικών ασκήσεων.

Η εκπονηθείσα εργασία αποτελεί μελέτη και υλοποίηση της εφαρμογής υγείας **PhysAI** *(Physiotherapy Artificial Intelligence),* η οποία με χρήση αλγορίθμων μηχανικής μάθησης θα μπορεί να αναγνωρίζει και να αξιολογεί τις κινητικές ασκήσεις που εκτελεί ο ασθενής. Η εφαρμογή, που εγκαθίσταται σε έξυπνα κινητά τηλέφωνα (android, iOS) και η web έκδοσή της παρέχουν στο φυσικοθεραπευτή ένα επιπλέον εργαλείο για τον αυτόματο έλεγχο της σωματικής άσκησης που έχει τεθεί στον ασθενή. Έτσι ο θεράπων μπορεί να παρακολουθεί κατά πόσο συμμορφώνεται ο ασθενής με τις οδηγίες που έχουν δοθεί. Ο ασθενής από την άλλη, καθώς εκτελεί οποιαδήποτε άσκηση ενημερώνεται σε ζωντανό χρόνο για το πόσες επαναλήψεις έχει εκτελέσει, και με τι ποσοστό επιτυχίας.

Πιο συγκεκριμένα οι απαιτήσεις που θέσαμε κατά την έναρξη της υλοποίησης της εφαρμογής ήταν οι εξής:

* Η εφαρμογή να τρέχει αποδοτικά σε φορητές συσκευές (Android, iOS) που διαθέτει ο μέσος χρήστης.
* Να είναι φιλική προς το χρήστη, χωρίς να απαιτεί από αυτόν ειδικές ψηφιακές δεξιότητες.
* Να αξιολογεί τις κινητικές ασκήσεις σε ζωντανό χρόνο - σύγχρονα.
* Να σέβεται τα προσωπικά δεδομένα του χρήστη.
* Να μπορεί να αναγνωρίζει αυτόματα και να αξιολογεί ορθά τις κινήσεις που ο χρήστης εκτελεί, καθώς και τον αριθμό των επαναλήψεων.

Παράλληλα, δευτερεύων στόχος κατά την εκπόνηση της διπλωματικής εργασίας ήταν η χρήση και αξιολόγηση της απόδοσης σύγχρονων τεχνικών στην ανάπτυξη λογισμικού που χρησιμοποιούνται σε επαγγελματικά περιβάλλοντα. Καθότι η εφαρμογή που υλοποιήσαμε έχει αρκετά χαρακτηριστικά εμπορικού λογισμικού, χρησιμοποιήσαμε σε όλα τα στάδια υλοποίησης της (ανάλυση απαιτήσεων, υλοποίηση, έλεγχος, τεκμηρίωση) agile μεθοδολογία, ότι συναντάται και σε ένα σύγχρονο επαγγελματικό περιβάλλον.

Οι παραπάνω απαιτήσεις και μεθοδολογίες, αλλά και τα εργαλεία λογισμικού που χρησιμοποιήθηκαν θα αναλυθούν διεξοδικότερα στις αντίστοιχες ενότητες της εργασίας.

* 1. Δομή της εργασίας

Το **Κεφάλαιο 1** εισάγει τον αναγνώστη στο περιεχόμενο και το σκοπό της τρέχουσας διπλωματικής.

Τα Κεφάλαια 2 και 3 περιέχουν θεμελιώδεις έννοιες-ορισμούς των τεχνολογιών που σχετίζονται με το περιεχόμενο της παρούσας διπλωματικής εργασίας.

Στο **Κεφάλαιο 2** παρατίθενται η έννοια του m-health και βασικά στοιχεία των λογισμικών στα σύγχρονα κινητά τηλέφωνα, τα εργαλεία που χρησιμοποιούνται για την ανάπτυξή τους, η αρχιτεκτονική υλοποίησή τους και οι περιορισμοί τους.

Στο **Κεφάλαιο 3** εισάγονται οι θεμελιώδεις αρχές της μηχανικής μάθησης και των νευρωνικών δικτύων, αφιερώνοντας μεγάλο μέρος στα συνελικτικά νευρωνικά δίκτυα (convolutional neural nets), αφού αποτελούν τον κορμό των περισσότερων εφαρμογών υπολογιστικής όρασης (computer vision). Επιπλέον, αναλύονται ενδελεχώς οι αρχές λειτουργίας και η δομή τους και οι γνωστότεροι αλγόριθμοι εκτίμησης στάσης σώματος.

Στο **Κεφάλαιο 4** παρουσιάζεται διεξοδικά η εφαρμογή, η στρατηγική που ακολουθήθηκε για την υλοποίησή της, τα χαρακτηριστικά, οι δυνατότητες και η αρχιτεκτονική της.

Στο **Κεφάλαιο 5** παρουσιάζεται η agile μεθοδολογία που ακολουθήθηκε κατά την υλοποίηση της εφαρμογής και αξιολογούνται τα διάφορα εργαλεία συγγραφής λογισμικού και διαχείρισης έργου που χρησιμοποιήθηκαν.

Τέλος, τα κεφάλαια 6, 7, 8 απαρτίζουν το πειραματικό μέρος της παρούσας διπλωματικής εργασίας.

Στο **Κεφάλαιο 6** αποτυπώνεται η πειραματική διαδικασία που ακολουθήθηκε, τα σύνολα δεδομένων που χρησιμοποιήθηκαν, η τελική επίδοση του συστήματος.

Στο **Κεφάλαιο 7** παρατίθενται τα αποτελέσματα των πειραμάτων σε γραφικές αναπαραστάσεις.

Στο **Κεφάλαιο 8** γίνεται μελέτη και ερμηνεία των πειραματικών αποτελεσμάτων και τα βασικά συμπεράσματα που αποκομίστηκαν, καθώς και οι πιθανές προεκτάσεις της εφαρμογής.

# Κεφάλαιο 2

m-Health Εφαρμογές

2.1 Εισαγωγή - ehealth

Στην εποχή της πληροφορίας οι εφαρμογές (applications) αποτελούν χρήσιμα εργαλεία που διευκολύνουν και βελτιώνουν το τρόπο ζωής μας. Η αξιοποίησή των εφαρμογών αυτών για συλλογή ιατρικών δεδομένων που βοηθούν στη λήψη ιατρικών αποφάσεων από τους παρόχους υγείας εντάσσεται στο ευρύτερο Health information technologies (eHealth).   
Ωστόσο, το e-health δεν περιορίζεται αποκλειστικά σε ιατρικές τεχνολογίες, αλλά αποτελεί τη τομή υπηρεσιών δημόσιας υγείας και παροχής ιδιωτικής φροντίδας.

Το e-health ανάλογα με τις εφαρμογές του μπορεί να κατηγοριοποιηθεί σε 6 μεγάλες κατηγορίες[https://www.routledge.com/Introduction-to-Smart-eHealth-and-eCare-Technologies-1st-Edition/Merilampi-Sirkka-Iniewski/p/book/9781315368818]:

* Τεχνολογίες υποστήριξης: είναι εργαλεία, εξοπλισμός και συσκευές που βοηθούν άτομα με σωματικές βλάβες να ανταποκριθούν στις ανάγκες της καθημερινότητας.  
  Η τηλεϊατρική και η υγεία μέσω κινητών συσκευών (m-health) αποτελούν κύρια παραδείγματα.
* Τεχνολογίες ασφάλειας: πρόκειται για εργαλεία ασφαλείας δικτυωμένα με υπηρεσίες επιτήρησης όπως έξυπνα δάπεδα και βραχιόλια ασφαλείας, αισθητήρες συνδεδεμένοι με απομακρυσμένους διακομιστές που επεξεργάζονται τα δεδομένα. Πολύ γνωστή είναι η εφαρμογή τους για ανίχνευση πτώσης (fall detection).
* Κοινωνικές τεχνολογίες: χαρακτηριστικό παράδειγμα κοινωνικής τεχνολογίας είναι η τηλεόραση υγείας που παρέχει ομιλία και οπτική σύνδεση μέσω ευρυζωνικού δικτύου.
* Τεχνολογίες άμεσης υγείας: πρόκειται για συστήματα αυτοελέγχου και αυτοεξυπηρέτησης, όπως πιεσόμετρα, πύλες (portals) ηλεκτρονικής υγείας και άλλες διαδικτυακές υπηρεσίες που επιτρέπουν άμεση επαφή με τους παρόχους υγείας.
* Νοσοκομειακή τεχνολογία: πρόκειται για εργαλεία που προορίζονται για συλλογή δεδομένων από ασθενείς μέσω έξυπνων συσκευών, συστήματα διαχείρισης φαρμάκων και άμεσης παρακολούθησης της υγείας τους. Παρέχουν χρήσιμες ιατρικές πληροφορίες όπως φυσική κατάσταση, απόκριση στις συμβουλές ιατρού και ημερήσια χαρτογράφηση του ασθενούς.
* Τεχνολογίες διαχείρισης πληροφοριών: πρόκειται για συστήματα αποθήκευσης του ιατρικού ιστορικού των ασθενών (EHR), ηλεκτρονική συνταγογράφιση. eDespensing πληροφοριακά συστήματα φαρμακείων.

Στην Ελλάδα τα τελευταία χρόνια δόθηκε μεγάλη έμφαση στη δημιουργία ενός ολοκληρωμένου ψηφιοποιημένου συστήματος υγείας. Τα δίκτυα πληροφοριών υγείας, το ηλεκτρονικό μητρώο υγείας, η ηλεκτρονική κάρτα υγείας, οι ηλεκτρονικές συνταγές φαρμάκων, η τηλε-ιατρική αποτελούν μερικές από τις συνιστώσες του [https://www.researchgate.net/publication/349166767\_E-Health\_Applications\_for\_Smart\_and\_Pervasive\_Healthcare\_in\_Greece\_What\_Can\_We\_Expect].

Γενικότερα, αυτή τη περίοδο υπάρχει έντονη συνεργατικότητα παγκοσμίως σε κλάδους και τομείς υγείας. Αυτή η συνεργασία έχει προκαλέσει μεγάλο ερευνητικό ενδιαφέρον λόγω των πιθανών κοινωνικο-οικονομιών επιπτώσεων[]. Η προοπτική επέκτασης του e-health έχει επίπτωση στη τεχνολογία, πολιτική, οικονομία, διεθνή κοινότητα, στην έρευνα και στους επενδυτές. Επιπλέον, οι τεχνολογίες e-health δεν περιορίζονται μόνο στα ψηφιακά συστήματα, αλλά περιλαμβάνουν ρομποτικές τεχνολογίες και αυτοματοποιημένα συστήματα για πρόβλεψη ασθενειών και προηγμένα συστήματα ανάλυσης ιατρικών δεδομένων για εύρεση συσχετίσεων μεταξύ ασθενειών.

Ενώ οι παραδοσιακές τεχνολογικές μέθοδοι που προαναφέρθηκαν συμβάλλουν ουσιαστικά στην επέκταση του ehealth, υπάρχει άλλο ένα εργαλείο που μπορεί να αυτοματοποιήσει και να προσδώσει «ευφυΐα» στις μεθόδους που ήδη χρησιμοποιούνται. Το εργαλείο αυτό είναι η Τεχνητή Νοημοσύνη και ειδικότερα τα μοντέλα μηχανικής μάθησης [https://pubmed.ncbi.nlm.nih.gov/33836752/]. Σκοπός των μοντέλων αυτών είναι η σωστή ερμηνεία και χρήση των δεδομένων του ασθενή-χρήστη για εξαγωγή λογικών συμπερασμάτων. Τα μοντέλα αυτά, μετατρέπουν τις εξαρτήσεις μεταξύ των δεδομένων σε αλγοριθμικές εκφράσεις και στη συνέχεια το σύστημα μηχανικής μάθησης εκπαιδεύεται στο να μπορεί να αντιστοιχίζει δεδομένα με παρόμοια χαρακτηριστικά σε διακριτά σύνολα. Αυτή η δυνατότητα καλείται γενίκευση. Αυτή η «γνώση» εξάγεται από την ανάλυση των αντίστοιχων δεδομένων και συσσωρεύεται κατά τη διάρκεια μιας μαθησιακής διαδικασίας.

Στο εγγύς μέλλον, ο τεράστιος όγκος συσσωρευμένων πληροφοριών (big data) θα απαιτήσει την άμεση προσαρμογή των οργανισμών υγείας στις νέες τεχνολογίες. Ο κύριος στόχος λοιπόν της ηλεκτρονικής υγείας (ehealth) που υποστηρίζεται από την τεχνητή νοημοσύνη είναι ένα σύστημα υγειονομικής περίθαλψης ενισχυμένο με μια σειρά στρατηγικών που βασίζονται στην εξαγωγή γνώσης (knowledge distillation) από το μεγάλο όγκο δεδομένων που δημιουργεί το σύστημα από τους ασθενείς.

Τα δεδομένα που συλλέγονται από ορισμένες φορητές συσκευές (smartwatches, smartphones κτλ), μπορούν εύκολα να υποβληθούν σε επεξεργασία για να αποσταλούν και να ενσωματωθούν σε υψηλότερο επίπεδο, στο ιατρικό ιστορικό του ασθενούς (EHR Electronic Health Record), σε διακομιστές ή στο υπολογιστικό νέφος του νοσοκομείου αναφοράς. Ακόμα, το σύνολο των δεδομένων που σχετίζονται με την ηλεκτρονική συνταγογράφηση, επιτρέπει στους επαγγελματίες να έχουν προνομιακή πληροφόρηση και γνώση για την κατάσταση και την εξέλιξη του ασθενούς. Βεβαίως, αν και ο τεράστιος όγκος δεδομένων μπορεί πράγματι να συνεισφέρει στην εξαγωγή πολλών συμπερασμάτων, θα πρέπει πάντοτε να δίνεται ειδική μέριμνα στην ασφαλή αποθήκευση και διαχείριση των ιατρικών δεδομένων με σεβασμό στο ιατρικό απόρρητο και τα προσωπικά δεδομένα των ασθενών [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820452/].

Όπως προαναφέρθηκε, ο τύπος του αλγορίθμου εκμάθησης που θα χρησιμοποιηθεί εξαρτάται σε μεγάλο βαθμό από τη δυσκολία του προβλήματος και τις οριακές συνθήκες που πρέπει να τηρούνται. Αυτοί οι περιορισμοί καθορίζουν, μεταξύ των άλλων, την απαραίτητη υπολογιστική ικανότητα, τον απαραίτητο αριθμό δεδομένων και παραμέτρων του μοντέλου.

Για παράδειγμα, αν ο αλγόριθμος πρέπει να εκτελεστεί σε επίπεδο wearable αισθητήρα η μέθοδος πρέπει να επιλεγεί προσεκτικά και να εξεταστεί η ανάγκη για μειωμένες απαιτήσεις μνήμης και υπολογιστικής δύναμης. Εντούτοις, σήμερα το λογισμικό των συσκευών έχει τη δυνατότητα να εκμεταλλεύεται στο έπακρο τις δυνατότητες του υλισμικού, όπως για παράδειγμα συμβαίνει με την αξιοποίηση της κάρτας γραφικών (GPU) στην εκτέλεση αλγορίθμων νευρωνικών δικτύων.

Σε κάθε περίπτωση τα πλεονεκτήματα χρήσης τεχνητής νοημοσύνης στην υγεία υπερτερούν σημαντικά, σε σχέση με το κόστος και την πολυπλοκότητα που εισάγουν στη θεραπευτική διαδικασία. Τα οφέλη αυτά αφορούν τόσο τον ασθενή όσο και τον πάροχο.

Ο ασθενής λαμβάνει εξατομικευμένη ιατρική θεραπεία, η οποία βασίζεται στο ιατρικό ιστορικό του ίδιου, τη γενετική κληρονομικότητα, τις προηγούμενες ασθένειες, τη διατροφή, ψυχολογικούς παράγοντες, τα επιθυμητά αποτελέσματα που έχει επιδείξει κτλ. Το σημαντικότερο όμως, είναι ότι πλέον ο ασθενής μπορεί να συμμετέχει ενεργά στις ιατρικές αποφάσεις που αφορούν την υγεία του, αφού μπορεί να αντλήσει πληροφορία από το σύστημα και να παρατηρεί άμεσα και διαδραστικά την πρόοδό του. Τέλος, μειώνεται το κόστος της ιατρικής διαδικασίας, αφού μειώνονται οι συχνές μετακινήσεις.

Από την άλλη η χρήση της τεχνητής νοημοσύνης από τη μεριά του ιατρικού παρόχου συνεισφέρει στην ταχύτερη και ακριβέστερη διάγνωση ασθενειών, στην καλύτερη πρόγνωση της υγείας, και σε βελτιωμένες, πιο στοχευμένες, ιατρικές αποφάσεις με μειωμένη πιθανότητα σφάλματος [https://www.dbooks.org/artificial-intelligence-in-medical-imaging-3319948784/read/]. Η τεχνητή επιπλέον νοημοσύνη μπορεί να γεφυρώσει την απόσταση του ασθενούς με τον ιατρού του, αφού επιτρέπει σε πολλά εργαλεία τη συνέχιση της ιατρικής διαδικασίας χωρίς την ανάγκη της φυσικής παρουσίας. Ο γιατρός δηλαδή μπορεί ανά πάσα στιγμή να δέχεται πληροφορίες για την υγεία του ασθενούς σε πραγματικό χρόνο ώστε να μπορεί να επέμβει.

2.2 m-health

Μία από τις δυνατότητες του e-health είναι η αξιοποίηση κινητών συσκευών στο τομέα της υγείας. Τα σύγχρονα έξυπνα τηλέφωνα (smartphones) έχουν αποδειχτεί έμπιστος σύμμαχος, με τους ενεργούς χρήστες να αυξάνονται ταχύτατα [https://www.bankmycell.com/blog/how-many-phones-are-in-the-world], ιδίως στις ανεπτυγμένες χώρες αλλά πλέον σταδιακά και στις αναπτυσσόμενες. Το κινητό τηλέφωνο αποτελεί αναπόσπαστο κομμάτι της καθημερινότητάς μας. Επομένως, οποιαδήποτε εφαρμογή που σχετίζεται με την υγεία έχει μεγάλες πιθανότητες αποδοχής από το χρήστη. Μέσω GPS, SMS και τηλεδιασκέψεων με τεχνολογίες WLAN/GPRS/5G είναι εφικτός ο εντοπισμός του ασθενούς που χρειάζεται άμεση υποστήριξη. Επιπλέον, ο ασθενής μπορεί να κρίνει ο ίδιος τη φυσική του κατάσταση λαμβάνοντας σε ζωντανό χρόνο (real time) βιο-σήματα και έτσι να αποφασίσει αν χρειάζεται επίσκεψη στον ιατρό. Οι χρόνιες ασθένειες, για παράδειγμα, αποτελούν τις πιο συχνές αιτίες θανάτου παγκοσμίως []. Το παραδοσιακό σύστημα περιστασιακής περίθαλψης σε κλινική αποδείχτηκε ότι δεν είναι απαρκές. Ο λόγος, η αδυναμία συμμόρφωσης με τις οδηγίες παρόχου σε μακροπρόθεσμο επίπεδο. Ιδίως σε δυσπρόσιτες περιοχές, ή σε περιόδους υγειονομικών περιορισμών η τακτική παρακολούθηση των ασθενών δυσχεραίνεται. Μια έξυπνη εφαρμογή ολοκληρωμένης ιατρικής παρακολούθησης μπορεί να δώσει λύση σε αυτό το πρόβλημα, ενθαρρύνοντας έναν ασθενή να συμμετάσχει ενεργά χωρίς ενδοιασμούς. Σύμφωνα με το Παγκόσμιο Οργανισμό Υγείας, η αυξημένη συμμόρφωση στην ιατρική αγωγή μπορεί να βοηθήσει τους ασθενείς να ζήσουν καλύτερα και περισσότερο [].

Η χρήση μάλιστα των m-health τεχνολογιών μπορεί να αποδειχθεί ιδιαίτερα χρήσιμη σε ευαισθητες ομάδες ασθενών, όπως για παράδειγμα στα καρκινοπαθή παιδιά. Παιχνιδοποιήμένες έξυπνες εφαρμογές (gamified apps), όπως για παράδειγμα το The Pain Squad, επιτρέπουν την καθημερινή παρακολούθηση των επιπέδων πόνου που αισθάνεται ένας νεαρός ασθενής κατά τη διάρκεια θεραπείας καρκίνου, αποφεύγοντας τους συμβατικούς τρόπους καταγραφής που τα αποθαρρύνουν λόγω της συνήθως βεβαρυμένης ψυχικής τους κατάστασης [<https://www.iasp-pain.org/publications/relief-news/article/pain-squad-app/>].

Η πιο διαδεδομένη χρήση τους βέβαια, είναι η λήψη αυτοματοποιημένων μηνυμάτων. Στην Ελλάδα αποστέλλονται μηνύματα για υπενθύμιση χορήγησης φαρμάκων και ραντεβού, για έγκαιρο εμβολιασμό, για υπενθύμιση προληπτικών ελέγχων, για έκτακτες ανάγκες από τη πολιτική προστασία κτλ.

Γενικότερα, Μ-health λογίζεται ως «η χρήση ασύρματων κινητών τερματικών, ιατρικών αισθητήρων και τεχνολογιών επικοινωνίας για παροχή ιατροφαρμακευτικής φροντίδας».   
  
Οι σημαντικότερες λειτουργίες κινητών τεχνολογιών είναι οι εξής:

* επιτήρηση περιβαλλοντικών συνθηκών
* επιτήρηση επιδημιολογικών φαινομένων
* παρακολούθηση εξέλιξης της νόσου
* υποστήριξη της θεραπευτικής διαδικασίας
* αλληλεπίδραση μεταξύ ασθενούς και παρόχου υγείας (ιατρού, νοσοκομείου)
* προώθηση ενός υγιεινού τρόπου ζωής
* αντιμετώπιση των εξαρτήσεων

Στην εικόνα[] φαίνονται οι χρήσεις (use cases) των κινητών συσκευών από τους παρόχους υγείας:



Oι περισσότερες λειτουργίες μπορούν να ενσωματωθούν σε έξυπνα κινητά, λόγω της επεξεργαστικής ισχύς που διαθέτουν, η οποία πολλαπλασιάστηκε κατά την τελευταία δεκαετία.

Η υιοθέτηση υγιεινού τρόπου ζωής συνεπάγεται και ψυχοσωματική υγεία. Η φυσική δραστηριότητα (Physical Activity) αποτελεί βασικό εργαλείο για την επίτευξή της.   
Ο Παγκόσμιος οργανισμός υγείας (WHO) προτείνει 150 λεπτά φυσικής άσκησης κάθε εβδομάδα σε άτομα μεγαλύτερης ηλικίας [https://www.researchgate.net/publication/341485785\_A\_Systematic\_Review\_on\_the\_use\_of\_mHealth\_to\_increase\_physical\_activity\_in\_older\_people]. Ωστόσο, οι περισσότεροι το αμελούν λόγω έλλειψης ενδιαφέροντος, έλλειψης χρόνου, φοβίες τραυματισμού. Αυτό μπορεί να προκαλέσει σοβαρά σωματικά και νοητικά προβλήματα, κατάθλιψη,αναπηρία ακόμα και χρόνιες παθήσεις [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4241367/]. Επομένως, τα έξυπνα τηλέφωνα (smartphones) και άλλες έξυπνες φορητές συσκευές (wearables), όπως activity trackers και smartwatches[\*], πρέπει να παίξουν καθοριστικό ρόλο στη βελτίωση της φυσικής υγείας.

2.3 Android-iOS

2.3.1 Εισαγωγή:

Η ταχεία ανάπτυξη της βιομηχανίας κινητών τηλεφώνων είναι θα ήταν αδύνατη χωρίς την ταυτόχρονη ανάπτυξη προηγμένων λειτουργικών συστημάτων ειδικά σχεδιασμένων για να εκμεταλλεύονται στο έπακρο το υλισμικό των συσκευών.

Η εύκολη χρήση και εγκατάσταση εφαρμογών σε κινητές συσκευές οφείλεται κατά κύριο βαθμό στο λογισμικό που χρησιμοποιούν. Όπως ακριβώς και το λογισμικό υπολογιστή, το λογισμικό των κινητών καθορίζει τις ιδιότητες και λειτουργικές δυνατότητες της συσκευής.

Ενώ έχουν αναπτυχθεί πολλά λειτουργικά συστήματα για κινητά τηλέφωνα, αυτά που έχουν επικρατήσει πλέον στις περισσότερες συσκευές είναι το Android της Google και το iOS της Apple. Η επικράτηση των δύο συγκεκριμένων λειτουργικών συστημάτων, και η δημιουργία ενιαίων διεπαφών σε όλες τις διαθέσιμες φορητές συσκευές διευκόλυνε επιπλέον την ανάπτυξη λογισμικού.

2.3.2 Κινητή υπολογιστική και smartphone

To έξυπνο κινητό είναι ένα κινητό τηλέφωνο που ενσωματώνει δυνατότητες ενός υπολογιστή, δηλαδή διαθέτει λειτουργικό σύστημα στο οποίο μπορούν να τρέξουν διάφορες εφαρμογές χρήστη, έχει πιο προηγμένη υπολογιστική ικανότητα και περισσότερες διασυνδέσεις σε σύγκριση με ένα απλό κινητό. Ιστορικά, σε σύγκριση με τους κλασικούς υπολογιστές (laptop, desktop) τα smartphones είχαν μικρότερο μέγεθος οθόνης, μικρότερη χωρητικότητα μνήμης και περιορισμένες δυνατότητες και λειτουργικότητες.[silberchatz] Πρόσφατα ωστόσο τα χαρακτηριστικά και η υπολογιστική ισχύς των φορητών συσκευών, συμπεριλαμβανομένων και των ταμπλέτων (tablets), έχουν προχωρήσει πολύ, ώστε η διαφορές λειτουργικότητας ανάμεσα σε ένα κινητό και έναν υπολογιστή είναι πλέον δυσδιάκριτες. Μάλιστα για ορισμένες λειτουργικότητες των smartphones, όπως για παράδειγμα οι εφαρμογές επαυξημένης πραγματικότητας (AR), οι οποίες διαστρωματώνουν πληροφορίες του τρέχοντος περιβάλλοντος πάνω σε μία οθόνη, είναι δύσκολο να φανταστούμε πώς θα μπορούσαν να αναπτυχθούν ισοδύναμες εφαρμογές σε κλασικούς υπολογιστές που δε διαθέτουν τη δυνατότητα εύκολης μετακίνησης στο χώρο.  
Παραδείγματα λειτουργικών δυνατοτήτων των έξυπνων κινητών:

* Ενσωματωμένη ψηφιακή κάμερα
* GPS navigation
* Web browsers
* οθόνη αφής
* Wi-Fi
* AR/VR (επαυξημένη πραγματικότητα/εικονική πραγματικότητα)

Ένα από τα μεγαλύτερα μειονεκτήματα όμως των κινητών τηλεφώνων είναι η ανάγκη για περιορισμένη κατανάλωση ισχύος, λόγω της χρήσης μπαταρίας. Ωστόσο και σε αυτό το τομέα οι εξελίξεις τόσο στη τεχνολογία συσσωρευτών όσο και στην αποδοτικότερη διαχείριση των πόρων από μεριάς υλισμικού επιτρέπουν πλέον υψηλές αποδόσεις.   
Σε αυτή την κατεύθυνση βέβαια συνεισφέρει και η παρουσία πολυπήρηνων επεξεργαστών, βραδύτερων μεν σε συχνότητα, αλλά περισσότερων σε σχέση με τους κλασικούς υπολογιστές.

2.3.3 Λογισμικά για Smartphones

Υπάρχουν αρκετά λειτουργικά συστήματα για κινητές συσκευές. Τα σημαντικότερα είναι τα εξής:

* Android από τη Google
* iOS από την Apple
* Symbian OS από τη Nokia
* Blackberry OS από τη RIM
* Windows Phone από τη Microsoft
* MIUI από τη Xiaomi
* Bada από τη Samsung
* Harmony OS της Huawei
* Επιπλέον open-source βαριάντες του Linux

Τέτοιου είδους λογισμικά μπορούν να εγκατασταθούν σε διάφορα μοντέλα κινητών τηλεφώνων και αναβαθμίζονται τακτικά με βελτιωμένες εκδόσεις. Αξίζει να σημειωθεί ότι

το Android και το iOS, τα οποία έχουν επικρατήσει στην αγορά, παρέχουν αντίστοιχα και πλατφόρμα ανάπτυξης λογισμικού, γιατί ο ανταγωνισμός μεταξύ τους είναι μεγάλος. Επιπλέον το MIUI και το Harmony OS έχουν αρκετή ομοιότητα με το Android, οπότε υπάρχει μεγάλη συμβατότητα στις εφαρμογές χρήστη, μεταξύ των τριών αυτών λειτουργικών. Η δομή των λειτουργικών συστημάτων που ακολουθούν είναι υβριδική.

2.3.4 Android

Η εταιρία Android, Inc ιδρύθηκε από τον Andy Rubin το 2003. Δύο χρόνια αργότερα, η Google την εξαγόρασε και στη συνέχεια η ομάδα με επικεφαλής τον Ρούμπιν ανέπτυξε πλατφόρμα κινητών συσκευών βασισμένη σε πυρήνα Linux.  
Το 2007 η Open Handset Alliance, επιχειρηματική συμμαχία πολλών εταιρειών στις οποίες περιλαμβάνεται η Google,η HTC, η Intel, η LG και άλλες 76 εταιρείες ενώθηκαν παρουσιάζοντας το πρώτο τους προϊόν, το Android, μια πλατφόρμα φορητών συσκευών που βασίστηκε στην έκδοση 2.6 του πυρήνα Linux. Το Linux χρησιμοποιείται κυρίως για υποστήριξη διεργασιών, μνήμης και οδηγών συσκευών για υλικό και έχει επεκταθεί για να περιλαμβάνει και διαχείριση ισχύος.Η πρώτη εμπορικά διαθέσιμη συσκευή που χρησιμοποίησε το λειτουργικό σύστημα Android ήταν το HTC Dream, που κυκλοφόρησε το 2008.

Στο παρακάτω διάγραμμα αποτυπώνεται η δομή του λειτουργικού συστήματος Android [https://developer.android.com/guide/platform].



Αποτελείται λοιπόν από διάφορα αφαιρετικά επίπεδα (layers):

* Applications

Σε αυτό το επίπεδο εγκαθίστανται default εφαρμογές όπως κάμερα, επαφές, email, αποστολή SMS, ημερολόγιο, μηχανή αναζήτησης κτλ, αλλά και εξωτερικές εφαρμογές που μπορεί να κατεβάσει ο χρήστης από το Play Store της Google. Οι περισσότερες εφαρμογές υλοποιούνται σε Java, αλλά υπάρχουν frameworks και για γλώσσες όπως η Kotlin, C#, Javascript.

* Application Framework - Επίπεδο διασύνδεσης

Περιλαμβάνει διάφορες κλάσεις (classes), οι οποίες είναι απαραίτητες για τη δημιουργία Android εφαρμογών. Επίσης, περιλαμβάνει πλήθος από διασυνδέσεις λογισμικού API σε Java παρέχοντας πρόσβαση στο developer στις πηγαίες βιβλιοθήκες.

* Native Libraries

Πρόκειται για εγγενείς βιβλιοθήκες υλοποιημένες σε C/C++ που χρησιμοποιούνται από διάφορες συνιστώσες και υπηρεσίες, όπως τα ART και HAL. Οι βιβλιοθήκες είναι προσβάσιμες από τους developers μέσω του Java API framework. To σύνολο αυτό περιέχει πλαίσια εργασίας για ανάπτυξη προγραμμάτων πλοήγησης (webkit), υποστήριξη βάσεων δεδομένων (SQLite) και πολυμέσα. Η βιβλιοθήκη Libc είναι παρόμοια με την πρότυπη βιβλιοθήκη C, αλλά είναι πολύ μικρότερη και έχει σχεδιαστεί για τις βραδύτερες CPU, που χαρακτηρίζουν τις κινητές συσκευές.

* Android Runtime  
  Αποτελεί μια από τις πιο σημαντικές συνιστώσες του λογισμικού.   
  Περιλαμβάνει τις core βιβλιοθήκες και την εικονική μηχανή Dalvik (πλέον ART). Οι πηγαίες βιβλιοθήκες (core libraries) είναι wrappers υλοποιημένα σε Java για τις εγγενείς C/C++ βιβλιοθήκες που προσφέρουν την επικοινωνία προγραμματιστών και πυρήνα. Κάθε εφαρμογή του υψηλότερου επιπέδου τρέχει ως αυτόνομη διεργασία, με δικό της στιγμιότυπο στην εικονική μηχανή Dalvik και μεταγλωττίζεται από αυτήν.
* Hardware Abstraction Layer

Προσφέρει διεπαφές (interfaces) ώστε το Application Framework να μπορεί να εκμεταλλευτεί τις δυνατότητες του υλισμικού της συσκευής. Το HAL συντίθεται από διάφορα modules βιβλιοθηκών, καθένα από τα οποία υλοποιεί μια διεπαφή για συγκεκριμένο υλικό της συσκευής (κάμερα,Bluetooth). Ουσιαστικά, κάθε φορά που το στρώμα Java API Framework «απαιτεί» πρόσβαση σε υλικό της συσκευής, φορτώνεται στη μνήμη το αντίστοιχο module.

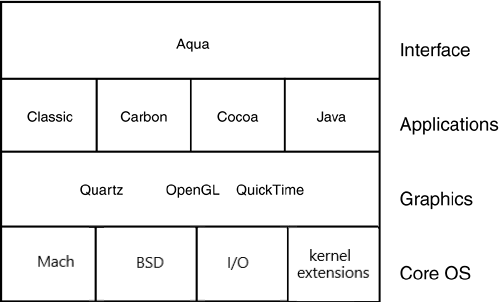
* Kernel

Η βάση του Android είναι ο πυρήνας Linux υλοποιημένος σε C. Για παράδειγμα, το Android Runtime βασίζεται στο πυρήνα για τη διαχείριση των διεργασιών και της μνήμης. Επίσης, επιτρέπει στους κατασκευαστές να δημιουργήσουν drivers ευκολότερα και σχετίζεται άμεσα με την ασφάλεια της συσκευής.

Πριν προχωρήσουμε στο λειτουργικό σύστημα της Apple για κινητά τηλέφωνα (iOS) κρίνουμε απαραίτητο να παρουσιάσουμε συνοπτικά και το αρχικό λειτουργικό σύστημα για υπολογιστές Mac OS X, αφού σε αυτό βασίζεται σε μεγάλο βαθμό το iOS.

2.3.5 Mac OS X

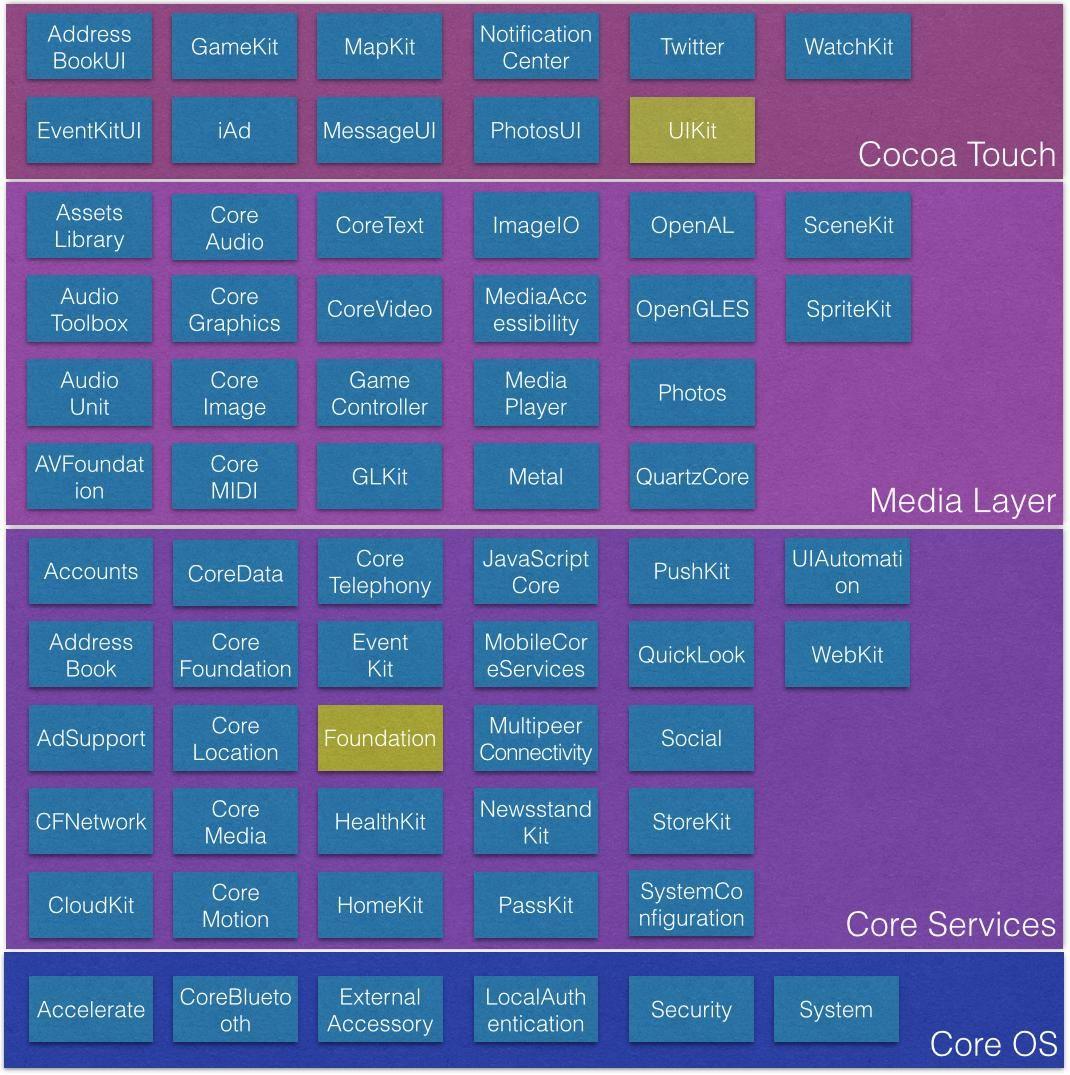
Το λειτουργικό σύστημα Apple MAc OS X χρησιμοποιεί μια υβριδική δομή. Όπως φαίνεται στην εικόνα[] αποτελείται από επίπεδα. Τα ανώτερα επίπεδα περιλαμβάνουν τη διεπαφή χρήστη Aqua και ένα σύνολο περιβαλλόντων και υπηρεσιών εφαρμογής. Είναι αξιοπρόσεκτο ότι το περιβάλλον Cocoa καθορίζει ένα API για τη γλώσσα προγραμματισμού Objective-C, η οποία χρησιμοποιείται στη συγγραφή εφαρμογών Mac OS X. Κάτω από αυτά τα επίπεδα βρίσκεται το περιβάλλον πυρήνα, το οποίο αποτελείται κυρίως από το μικροπυρήνα Mach και το πυρήνα BSD Linux. Το Mach παρέχει διαχείριση μνήμης, υποστήριξη για κλήσεις απομακρυσμένων διαδικασιών (IPC), που περιλαμβάνουν το πέρασμα μηνυμάτων και του χρονοπρογραμματισμού νημάτων (threads). Το συστατικό BSD παρέχει μια διεπαφή γραμμής εντολών του BSD, υποστήριξη για δικτύωση και για συστήματα αρχείων και για υλοποίηση των POSIX API, συμπεριλαμβανομένων και των Pthreads. Επιπλέον, στο Mach και στο BSD, το περιβάλλον πυρήνα παρέχει ένα κιτ εισόδου-εξόδου για ανάπτυξη οδηγών συσκευών.



2.3.6 iOS

To iOS είναι το λογισμικό της Apple για κινητές συσκευές. Αρχικά προοριζόταν για κινητά iPhone, στη συνέχεια επεκτάθηκε και στα υπόλοιπα προϊόντα της εταιρίας, όπως iPods, iPads, smartwatches, smart TV’s. Αποτελεί παράγωγο του λογισμικού για υπολογιστές   
Mac OS X, δεν είναι open-source και είναι σχεδιασμένο αποκλειστικά για προϊόντα Apple.

Στο παρακάτω διάγραμμα αποτυπώνεται η δομή του λειτουργικού συστήματος iOS  
[https://www.programmersought.com/article/42263842016/].



Ομοίως με το Android,αποτελείται από διάφορα αφαιρετικά επίπεδα [<https://www.amazon.com/Advanced-iOS-App-Architecture-Second/dp/194287894X>]:

* Cocoa touch

Είναι API για Objective-C, που παρέχει αρκετά πλαίσια εργασίας, ανάπτυξης εφαρμογών, που εκτελούνται σε συσκευές iOS. Η θεμελιώδης διαφορά ανάμεσα στο Cocoa, που αναφέρθηκε προηγουμένως και στο Cocoa Touch είναι ότι το δεύτερο παρέχει στήριξη για χαρακτηριστικά υλικού, τα οποία είναι μοναδικά σε κινητές συσκευές, όπως είναι οι οθόνες αφής. Αποτελεί ουσιαστικά το interface μεταξύ λογισμικού και χρήστη και περιλαμβάνει διάφορα APIs προς δημιουργία εφαρμογών από προγραμματιστές.

* Media Layer

Πρόκειται για frameworks υπεύθυνα για διαχείριση των γραφικών και του ήχου.  
Περιλαμβάνει τις απαραίτητες βιβλιοθήκες εικόνας και ήχου (OpenGL, Core Graphics, OpenAL κτλ)

* Core Services

Το επίπεδο αυτό παρέχει μια ποικιλία χαρακτηριστικών, που περιλαμβάνουν υποστήριξη για υπολογιστική νέφους και για βάσεις δεδομένων. Περιέχει απαραίτητα frameworks για τη συντήρηση, επιδιόρθωση και βελτίωση της απόδοσης του συστήματος. Είναι υλοποιημένα σε C και Objective C. Σκοπός η ομαλή λειτουργία βασικών υπηρεσιών του συστήματος, όπως οι επαφές, δικτυακές ιδιότητες, διαχείριση δεδομένων, ημερολόγια, GPS κτλ

* Core OS

Ο πυρήνας του λειτουργικού συστήματος, περιλαμβάνει βοηθητικά frameworks διαχείρισης των νημάτων του συστήματος, της μνήμης, sockets, IO, DNS, bluetooth και ρουτίνες για την ασφάλεια του.

Συγκριση????

Ενώ το iOS σχεδιάστηκε για να εκτελείται αποκλειστικά σε κινητές συσκευές της Apple και ο πηγαίος κώδικας είναι κλειστός, το Android εκτελείται σε μια ποικιλία πλατφορμών για κινητές συσκευές και είναι ανοικτού κώδικα, κάτι που εξηγεί τη ταχεία δημοτικότητά του.

To android, όντας λογισμικό ανοιχτού κώδικα, επεκτάθηκε και προσαρμόστηκε στα χαρακτηριστικά των μεγαλύτερων εταιριών κατασκευής κινητών, αφού αποτελεί οικονομικότερη αλλά ατόφια επιλογή. Αντίθετα, το iOS, επικεντρώθηκε αποκλειστικά στις συσκευές Apple, πετυχαίνοντας μέγιστη απόδοση και απόκριση σε αυτές.

# Κεφάλαιο 3

Μηχανική Μάθηση και Νευρωνικά Δίκτυα

3.1 Εισαγωγή

Η μηχανική μάθηση (machine learning) αποτελεί έναν από τους παλαιότερους κλάδους της Τεχνητής Νοημοσύνης και εστιάζει στη χρήση δεδομένων και αλγορίθμων με στόχο τα συστήματα να μαθαίνουν αυτόνομα από την εμπειρία. Η έννοια της μηχανικής μάθησης σε ένα γνωστικό σύστημα (cognitive system), όπως γίνεται αντιληπτή καθημερινά, συνδέεται με δύο βασικές ικανότητες:

* την παραγωγή επιπλέον γνώσης (γενίκευση) κατά την αλληλεπίδραση με το περιβάλλον στο οποίο δραστηριοποιείται
* τη βελτίωση μέσω επανάληψης του τρόπου εκτέλεσης μίας συγκεκριμένης ενέργειας

Στην παραγωγή, οι αλγόριθμοι μηχανικής μάθησης εφαρμόζονται αποδοτικά σε διάφορους τομείς. Στο κλάδο της Υγείας για παράδειγμα, χρησιμοποιούνται για πρόληψη, κατηγοριοποίηση και διάγνωση ασθενειών. Στο κλάδο της Όρασης Υπολογιστών, για αναγνώριση δραστηριότητας, εκτίμηση της στάσης του σώματος, ποιοτική αξιολόγηση προϊόντων, και κυρίως για αναγνώριση προσώπων και αντικειμένων. Στο κλάδο της Επεξεργασίας Φυσικής Γλώσσας, για την κατηγοριοποίηση είδους κειμένου, μετάφραση κειμένου, σε chatbots, κατηγοριοποίηση συναισθημάτων, μετατροπή φωνής σε κείμενο και ανάκτηση πληροφοριών. Επίσης, στο κλάδο της Οικονομίας, ιδίως στο χρηματιστήριο για τη πρόβλεψη τιμών μετοχής, για ανίχνευση απάτης, καθώς και ανάλυση κινδύνου.

3.2 Ιστορική Αναδρομή

Ο όρος «μηχανικής μάθηση» διατυπώθηκε από τον Arthur Samuel το 1959, επιστήμονα και μηχανικό της IBM, ο οποίος σχεδίασε πρόγραμμα για το γνωστό game checkers με την ικανότητα να διακρίνει τη βέλτιστη στρατηγική. Ο μαθηματικός ορισμός ήρθε από τον Tom Mitchell το 1997 ως:

«Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία Ε ως προς κάποια κλάση εργασιών Τ και μέτρο απόδοσης Ρ, αν η απόδοσή του σε εργασίες από το Τ, όπως μετριέται από το Ρ, βελτιώνεται μέσω της εμπειρίας Ε.»

Σημείο εκκίνησης τόσο για τη Μηχανική Μάθηση όσο και για το ευρύτερο πεδίο της Τεχνητής Νοημοσύνης αποτελεί η σύλληψη της ιδέας του τεχνητού νευρώνα (perceptron) από τους McCulloch και Pitts το 1943 []. Η απλότητά του μοντέλου ωστόσο έκανε αδύνατο τον υπολογισμό πιο πεπλεγμένων συναρτήσεων, έως τη παρέμβαση του Donald Hebb το 1949, ο οποίος πρότεινε ένα απλό κανόνα μεταβολής της δύναμης σύνδεσης μεταξύ των νευρώνων έτσι ώστε να επιτυγχάνεται η μάθηση. Το 1957 συνδυάζοντας τις ιδέες των McCulloch-Pitts και το κανόνα του Hebbs, υλοποιεί το perceptron με χρήση βαρών στις εισόδους του [].

Στη δεύτερο μισό του 20ου αιώνα ακολούθησαν επιπλέον καθοριστικές ανακαλύψεις όπως ο ταξινομητής κοντινότερων γειτόνων το 1967, ο αλγόριθμος backpropagation το 1974, τα δίκτυα Hopfield το 1982, τα πολυστρωματικά νευρωνικά δίκτυα (multilayer perceptrons) το 1986, τα δίκτυα LSTM το 1997 και πολλά άλλα. Αξίζει να σημειωθεί ότι η τεχνική πρόοδος στο hardware τις τελευταίες δεκαετίες, βοήθησε ουσιαστικά στην εφαρμογή ML μοντέλων στη πράξη, με πιο απτό παράδειγμα τα βαθιά συνελικτικά δίκτυα (deep learning neural nets).

3.3 Μηχανική Μάθηση

Σύμφωνα με τη βιβλιογραφία, ο τομέας της μηχανικής μάθησης αποτελείται από τρεις κατηγορίες τεχνικών εκπαίδευσης, οι οποίες προσομοιάζουν σε μεγάλο βαθμό και τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος. Πρόκειται για τους εξής[Τεχνητή Νοημοσύνη ΑΠΘ]:

* **Επιβλεπόμενη Μάθηση (Supervised Learning)**: στον οποίο ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο.  
  Χρησιμεύει κυρίως σε προβλήματα ταξινόμησης (classification), πρόβλεψης (predictions) και διερμηνείας (interpretation).  
  Γνωστοί αλγόριθμοι: k-Nearest Neighbours, Naïve-Bayes, SVMs, SOMs regression, decision trees
* **Μη επιβλεπόμενη Μάθηση (Unsupervised Learning)**: στον οποίο ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους.  
  Χρησιμεύει σε προβλήματα ομαδοποίησης (clustering), ανάλυσης συσχετισμών (association mining) και εύρεσης ανωμαλιών (anomaly detection).   
  Γνωστοί αλγόριθμοι: k-Means, hidden Markov models, Gaussian models, hierarchical, Kohonen’s self-organizing maps
* **Ενισχυτική μάθηση (Reinforced Learning)**: στον οποίο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον.  
  Χρησιμεύει κυρίως σε προβλήματα σχεδιασμού (Planning) στη ρομποτική.

Γνωστοί αλγόριθμοι: Q learning

Αξίζει να επισημανθεί ότι τα πολυστρωματικά νευρωνικά δίκτυα (multilayer neural networks) εφαρμόζονται επί το πλείστον σε προβλήματα επιβλεπόμενης μάθησης, όπως παρουσιάζεται εν συνεχεία. Προκειμένου να παρουσιαστεί η αρχιτεκτονική και οι ιδιότητες των συνελικτικών νευρωνικών δικτύων κρίνεται απαραίτητο να διατυπωθούν βασικοί ορισμοί της επιστήμης των

3.3 Νευρωνικά Δίκτυα

3.3.1 Το Perceptron

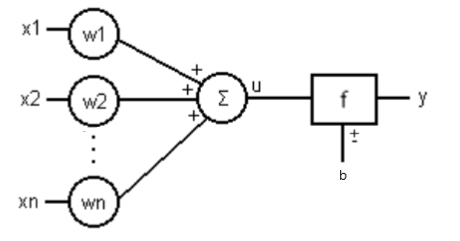
Σημείο εκκίνησης για την επιστήμη των νευρωνικών δικτύων είναι το perceptron του Rossenblatt [https://www.semanticscholar.org/paper/The-perceptron%3A-a-probabilistic-model-for-storage-Rosenblatt/5d11aad09f65431b5d3cb1d85328743c9e53ba96], το οποίο βασίστηκε στο μοντέλο νευρώνα των McCulloch-Pitts.   
Αυτό το νευρωνικό μοντέλο λαμβάνει ως είσοδο διάφορες αριθμητικές τιμές (πληροφορία). Οι τιμές εισόδου θα πολλαπλασιαστούν με αντίστοιχους συντελεστές, οι οποίοι αποκαλούνται βάρη. Έτσι, ο κόμβος υπολογίζει το εσωτερικό γινόμενο μεταξύ των διανυσμάτων εισόδου - βαρών και προστίθεται μια σταθερά b, η οποία ονομάζεται κατώφλι (threshold). Η αριθμητική τιμή της εξόδου του κόμβου άθροισης «διέρχεται» μέσω μιας συνάρτησης ενεργοποίησης, η οποία κανονικοποιεί τη τιμή συνήθως στο διάστημα [0,1].

Οι εξισώσεις που περιγράφουν το perceptron είναι οι εξείς:

u = Σwi\*xi +- b   
y = f( u - b ) , b>0

input = (x1, x2, .. , xm)

weights = ( w1,w2…. wm)



Η δομή της εικόνας[] είναι ένα Perceprton. Αν αυτό αποτελεί κομμάτι ενός ευρύτερου συνόλου μέσα σε νευρωνικό δίκτυο τότε αποκαλείται νευρώνας (neurons).

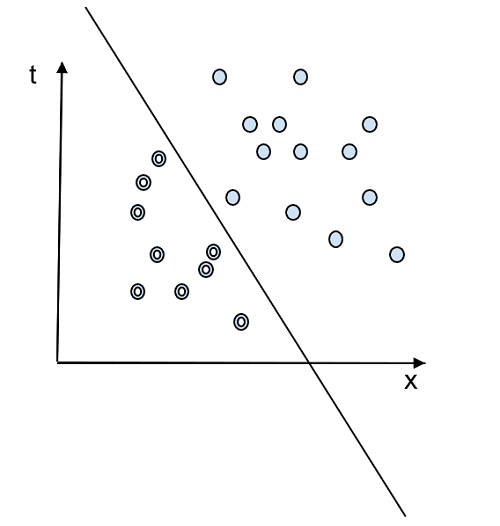
3.3.2 To perceptron ως ταξινομητής

Είναι εμφανές ότι η έξοδος του perceptron στο χώρο m διαστάσεων αποτελεί υπερεπίπεδο το οποίο ορίζεται από τις εξίσωση:

Σwi\*xi + b = 0 (\*)

Επίσης, ανάλογα με το πρόσημο της εξίσωσης (\*) προκύπτουν διακριτές τιμές στην έξοδο:

y = f( Σwi\*xi - b ) = { 1 αν >0 και 0 αν <0 }

Αυτή η παρατήρηση είναι θεμελιώδους σημασίας γιατί αποδεικνύεται μαθηματικά[θεωρημα συγκλισης] ότι η εξίσωση (\*) λειτουργεί ως όριο απόφασης, δηλαδή ταξινομητής. Πράγματι, στη περίπτωση διανύσματος εισόδου δύο μεταβλητών (x1,x2) όπως φαίνεται στο σχήμα[] το υπερεπίπεδο σε αυτή τη περίπτωση, η ευθεία, ταξινομεί σε δύο σύνολα (κλάσεις) τα δεδομένα εκατέρωθεν της.

Επομένως η ταξινόμηση έγκειται στη εύρεση των κατάλληλων βαρών του υπερεπιπέδου με τη προϋπόθεση οι κλάσεις των δεδομένων να είναι γραμμικά διαχωρίσιμες. τους. Αυτός ο περιορισμός, που είναι έχει ως αντιπαράδειγμα το γνωστό πρόβλημα XOR [] έχρηζε αντιμετώπισης. Η λύση δόθηκε από τα πολυεπίπεδα perceptron (Multi-layer perceprton).

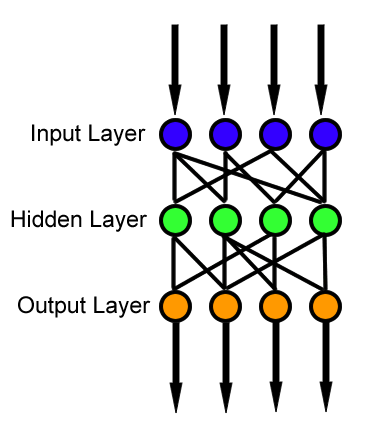
3.3.3 Πολυεπίπεδο perceptron

||

Νευρωνικό δίκτυο προσοτροφοδοτησης (feedforward neural net):   
Νευρωνικό δίκτυο στο οποίο η πληροφορία διαδίδεται από την είσοδο προς την έξοδο. Δεν εμπεριέχονται βρόγχοι ανάδρασης ( feedback loops).

||

Στην εικόνα[] φαίνεται ένα δίκτυο προώθησης.



||

Πολυστρωματικό δίκτυο perceptron (MLP):

Ορίζεται το νευρωνικό που αποτελείται από τουλάχιστον 3 επίπεδα:

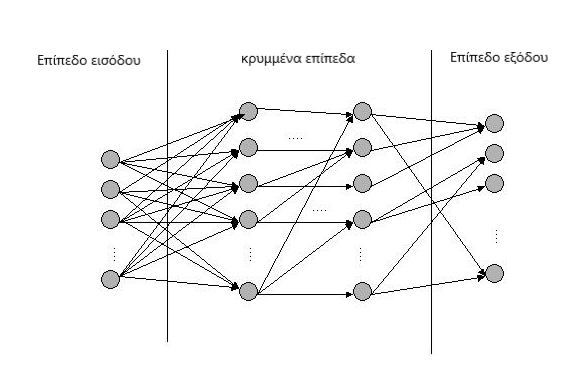
* 1 εισόδου
* 1 ή περισσότερα κρυμμένα
* 1 εξόδου

καθένα από τα οποία περιέχει πεπερασμένο αριθμό από perceptron και έχει τις εξής ιδιότητες:

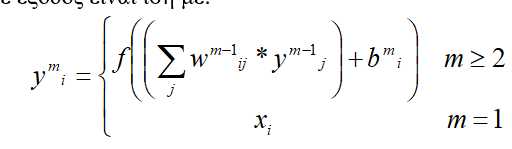
* ο νευρώνας του τρέχοντος επιπέδου συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου
* είναι δίκτυο προσοτροφοδοτησης

||

Παράδειγμα πολυστρωματικού δικτύου φαίνεται στη εικόνα[].



Έστω βάρη *wmij* , όπου *wmij*είναι το βάρος που συνδέει το *i* νευρώνα του (m - 1)-οστού επιπέδου με το *j* νευρώνα του *m-οστού* επιπέδου. Ο νευρώνας όπως προαναφέρθηκε είναι ένα perceprton. Η έξοδος ενός νευρώνα εφαρμόζεται ως είσοδος στους νευρώνες των επόμενων επιπέδων με τους οποίους συνδέεται μέχρι το επίπεδο εξόδου. Πιο αναλυτικά αν συμβολίσουμε με με *bmi* το κατώφλι του τρέχοντος νευρώνα και με *ymi* την έξοδό του, τότε η κάθε έξοδος είναι ίση με:



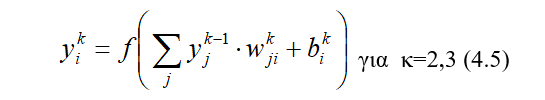
όπου *f* η συνάρτηση ενεργοποίησης και *xi* οι είσοδοι. Παρατηρούμε ότι για το πρώτο επίπεδο, δηλαδή για m=1 η έξοδος του κάθε νευρώνα είναι ίδια με την είσοδο.

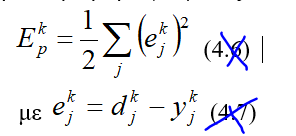
3.3.4 Αλγόριθμος οπισθοδιάδοσης

Το χαρακτηριστικό των πολυστρωματικών δικτύων είναι η δυνατότητά τους να «μαθαίνουν». Δηλαδή να προσαρμόζουν τις τιμές των βαρών τους ώστε να προκύπτουν οι επιθυμητές έξοδοι με βάση κάποιο κριτήριο. Προφανώς το κριτήριο είναι η ελαχιστοποίηση κάποιας μετρικής σφάλματος. Τα περισσότερα νευρωνικά εκπαιδεύονται με επίβλεψη. Ο πιο γνωστός αλγόριθμος εκπαίδευσης βαρών με επίβλεψη είναι η οπισθοδιάδοση (backpropagation).

Για την ερμηνεία του αλγορίθμου οι παρακάτω μαθηματικές εξισώσεις:

Έστω *yki* είναι η έξοδος του k-οστού επιπέδου, *δki* το σήμα σφάλματος του *i-οστού* νευρώνα του *k-οστού* επιπέδου και *wkji* το βάρος που συνδέει τον *j-οστό* νευρώνα του (*k-1)-οστού* επιπέδου με τον *i-οστό* του *k-οστού επιπέδου*. Τότε η έξοδος σε κάθε επίπεδο για κάθε νευρώνα είναι:





όπου *dkj* είναι η επιθυμητή έξοδος για την έξοδο *j* του επιπέδου *k* .

Στόχος είναι η ελαχιστοποίηση της συνάρτησης σφάλματος E συναρτήσει των βαρών του νευρωνικού. Αυτό είναι ισοδύναμο με εύρεση των τοπικών-ολικών ελαχίστων μιας συνάρτησης πολλών μεταβλητών. Υπάρχουν αρκετοί γνωστοί αλγόριθμοι όπως η μέθοδος Newton και επεκτάσεις της, η μέθοδος μη-γραμμικών ελαχίστων τετραγώνων και η μέθοδος Gradient descent. Στη προκειμένη περίπτωση θα χρήση του τελευταίου.

Με βάσει τον κανόνα κατάβασης παραγώγου (Gradient Descent) η παρακάτω σχέση δίνει το τρόπο αλλαγής των βαρών προκειμένου να μειωθεί το σφάλμα του δικτύου:

//προσθήκη εξισώσεων

3.4 Συνελικτικά Νευρωνικά Δίκτυα  
3.4.1 Εισαγωγικά

Το τεράστιο ενδιαφέρον για μοντέλα βαθιάς μάθησης (deep learning) και η πρόοδος στις κάρτες γραφικών (GPUs) τις τελευταίες δύο δεκαετίες οδήγησαν στην επικράτηση των συνελικτικών νευρωνικών δικτύων (CNNs). Πρόκειται για ειδική κατηγορία πολυστρωματικών δικτύων η αρχιτεκτονική των οποίων αποτελεί τη κυρίαρχη λύση, κυρίως σε προβλήματα όρασης υπολογιστών (computer vision).

Τα CNNs χρησιμοποιούνται κυρίως για [https://ijcsit.com/docs/Volume%207/vol7issue5/ijcsit20160705014.pdf]:

* ανάλυση εγγράφων (Document analysis) με οπτική αναγνώριση χαρακτήρων (OCR) σε εικόνες
* κατηγοριοποίηση αντικειμένων (Scene labelling) που εμπεριέχονται σε εικόνες
* αναγνώριση προσώπου (Face recognition)
* κατηγοριοποίηση εικόνων (Image classification)
* αναγνώριση κινήσεων (Action recognition) σε εικόνες, βίντεο
* εκτίμηση ανθρώπινης στάσης (Human pose estimation) σε εικόνες, βίντεο

Το CNN πρωτοεμφανίστηκε σε εργασία τη δεκαετία του 80 από τον LeCun, εξέχοντα ερευνητή στο πεδίο της Μηχανικής Μάθησης. Το πρώτο συνελικτικό δίκτυο ήταν το LeNet[], το οποίο είχε εκπαιδευτεί να αναγνωρίζει χειρόγραφα ψηφία και να τα ταξινομεί. Τη τελευταία δεκαετία όμως, η ύπαρξη μεγάλων βάσεων δεδομένων με εικόνες, όπως τα ImageNet, MNIST, CIFAR-10 επανέφερε το ενδιαφέρον της ερευνητικής κοινότητας για τα συνελικτικά. Το 2012 κάνει την εμφάνισή του το μοντέλο AlexNet το οποίο έχει παρόμοια αρχιτεκτονική με το LeNet και έχει εκπαιδευτεί στη κατηγοριοποίηση εικόνων γενικού περιεχομένου. Το σκεπτικό πίσω από την αλγοριθμική σχεδίαση των συνελικτικών είναι δανεισμένο από τα βιολογικά νευρωνικά δίκτυα και ανατρέχει στη πρωτοποριακή εργασία των Hubel και Wiesel (1962) πάνω στους τοπικά ευαίσθητους και επιλεκτικούς ως προς τον προσανατολισμό νευρώνες του οπτικού φλοιού της γάτας. Ένα συνελικτικό είναι perceptron πολλών επιπέδων ειδικά σχεδιασμένο να αναγνωρίζει δισδιάστατα σχήματα με υψηλό βαθμό μη ευαισθησίας στη μετατόπιση, κλιμάκωση, στρέβλωση και άλλες μορφές παραμόρφωσης της εικόνας.

Με ένα απλό παράδειγμα γίνεται ξεκάθαρο ποιο πρόβλημα οδήγησε στη κατασκευή συνελικτικών δικτύων εις βάρος των απλών νευρωνικών.  
Έστω εικόνα εισόδου 24x24x3 (24x24 pixels και 3 κανάλια χρωμάτων).  
Σε ένα πολυστρωματικό νευρωνικό κάθε νευρώνας πρέπει να συνδεθεί με όλα τα pixels της εικόνας. Επομένως τα συνολικά βάρη του νευρώνα θα είναι 1728. Αν όμως η εικόνα είναι μεγαλύτερη, π.χ. 192x192x3 τότε παρατηρούμε ότι τα βάρη θα γίνουν 110592, δραματική αύξηση. Επιπλέον, το δίκτυο θα αποτελούνταν από πολλούς τέτοιους νευρώνες, κάτι που καθιστά την αύξηση βαρών εκθετική. Τα Συνελικτικά Νευρωνικά Δίκτυα αξιοποιούν το γεγονός ότι η είσοδος αποτελείται από εικόνες και περιορίζουν τα βάρη με κατάλληλη αρχιτεκτονική. Ειδικότερα, οι νευρώνες σε ένα Συνελικτικό Δίκτυο είναι διατεταγμένοι σε 3 διαστάσεις: πλάτος, ύψος, βάθος. Οι νευρώνες κάθε επιπέδου όμως θα συνδέονται,   
με μια μικρή περιοχή του προηγούμενου επιπέδου, αντί με όλους τους προηγούμενους νευρώνες, όπως γινόταν στη πλήρως συνδεδεμένη εκδοχή.   
Στην επόμενη παράγραφο παρουσιάζονται πιο εξονυχιστικά τα δομικά στοιχεία ενός τέτοιου δικτύου.

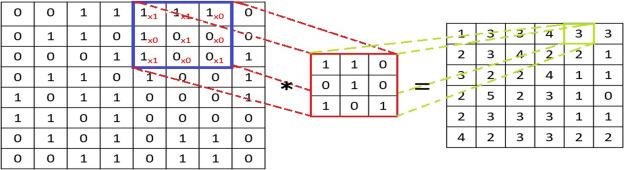
3.4.2 Αρχιτεκτονική

Η αρχιτεκτονική του CNN συντίθενται από τέσσερα είδη επιπέδων:

* Συνελικτικό επίπεδο
* Pooling layer
* Fully connected
* Dropout layer

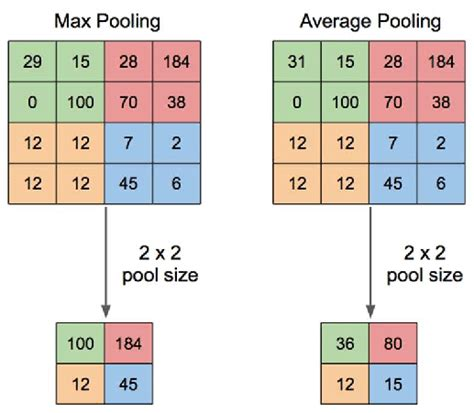
Συνελικτικό επίπεδο

Αποτελεί το δομικό συστατικό του CCN.  
Περιέχει φίλτρα (kernels) που εφαρμόζονται συνελικτικά στην αρχική εικόνα εξάγοντας τα ζητούμενα χαρακτηριστικά. Ειδικότερα, οι πυρήνες είναι μήτρες (matrices) μικρότερων διαστάσεων με συγκεκριμένα βάρη οι οποίες «ολισθαίνουν» κατά μήκος της και κατά πλάτος της μήτρας εισόδου και το εσωτερικό γινόμενο που προκύπτει αποτελούν τα εξαγόμενα χαρακτηριστικά όπως φαίνεται στην εικόνα[].



Ο πολλαπλασιασμός των Pixel της εικόνας με τα βάρη της μήτρας είναι η γνωστή πράξη της διακριτής συνέλιξης. Προκειμένου να μην υπάρχουν επικαλύψεις μεταξύ των περιοχών όπου εφαρμόζεται η συνέλιξη υπάρχουν ειδικές παράμετροι όπως το padding και το stride, που ρυθμίζουν την απόσταση μεταξύ των διαδοχικών ολισθήσεων. Η ίδια διαδικασία εφαρμόζεται και κατά βάθος, αφού οι εικόνες είναι RGB. Σκοπός είναι η εύρεση των βέλτιστων παραμέτρων-βαρών αυτών των φίλτρων, το οποίο επιτυγχάνεται μέσω της εκπαίδευσης του νευρωνικού δικτύου με μέθοδο επίβλεψης.

Pooling layer

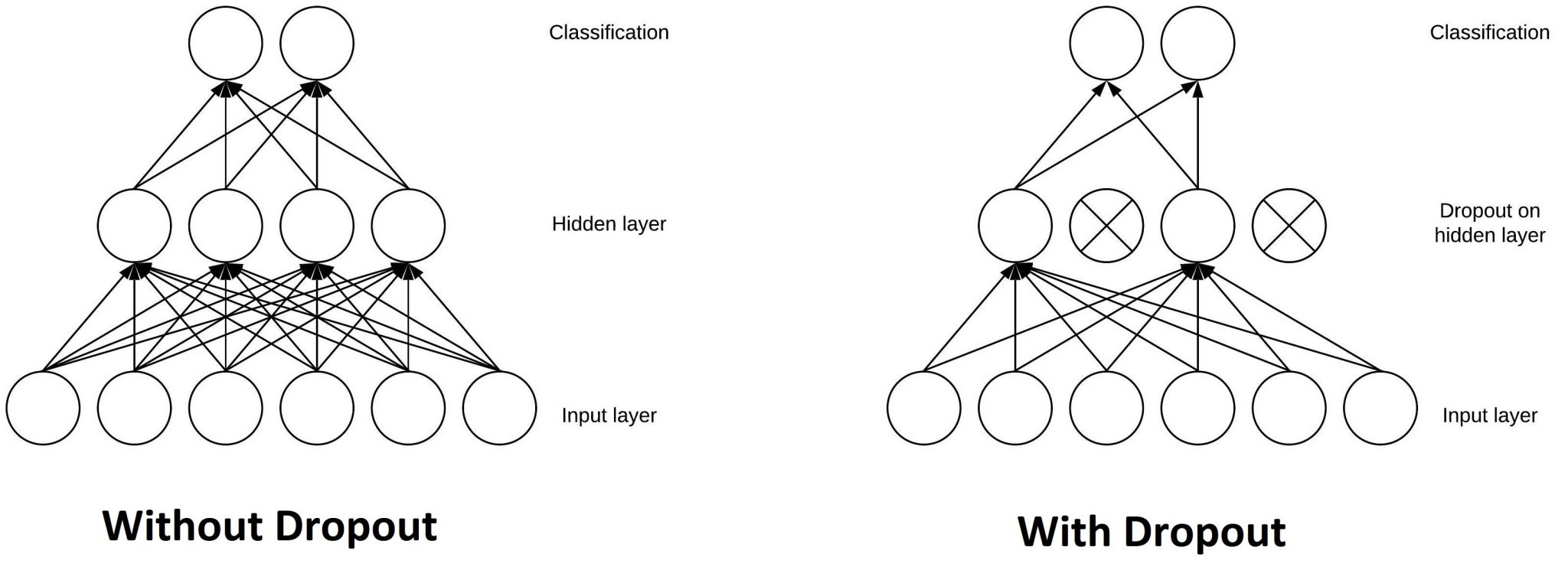
Συνήθως παρεμβάλλονται μεταξύ δύο διαδοχικών συνελικτικών επιπέδων και χρησιμεύουν ώστε να μειωθούν οι παράμετροι του συνελικτικού επιπέδου, χωρίς όμως να χάνονται τα σημαντικότερα χαρακτηριστικά. Οι δημοφιλέστερες συναρτήσεις για αυτή τη διαδικασία είναι η max-pool και avg-pool, οι οποίες «φιλτράρουν» τη μέγιστη και τη μέση τιμή των εικονοστοιχείων αντίστοιχα όπως φαίνεται στην εικόνα[].  
Η διαδικασία αυτή έχει ως στόχο να καταστήσει ακλόνητη την αναπαράσταση σε μικρές μεταβολές στην είσοδο. Δηλαδή αν οι μεταβολές στην είσοδο είναι πολύ μικρές, οι τιμές των περισσοτέρων εξόδων μετά το pooling επίπεδο δεν θα επηρεαστούν. Για παράδειγμα, αν μια εικόνα είναι ελαφρώς παραμορφωμένη, θα διατηρηθούν τα χαρακτηριστικά που τη διέπουν. Αυτή η ιδιότητα είναι σημαντικότατη για μικρές τοπικές αλλαγές ιδιαίτερα αν μας ενδιαφέρει περισσότερο η παρουσία κάποιου χαρακτηριστικού παρά η ακριβής θέση του. 

Πλήρως διασυνδεδεμένο επίπεδο

Πρόκειται για τυπικό feedforward δίκτυο και τοποθετείται πάντα στο τέλος του νευρωνικού. O ρόλος του είναι η ταξινόμηση των εικόνων. Επιστρέφει ένα διάνυσμα μεγέθους Ν, όπου Ν είναι ο αριθμός των κλάσεων στο πρόβλημα ταξινόμησης εικόνων. Κάθε στοιχείο του διανύσματος εκφράζει το ποσοστό πιθανότητας να ανήκει στη συγκεκριμένη κλάση η εικόνα. Ο υπολογισμός των πιθανοτήτων γίνεται εφαρμόζοντας συνάρτηση ενεργοποίησης στην έξοδο του συνήθως log για δυαδική ταξινόμηση και softmax για ταξινόμηση σε περισσότερες κλάσεις. Είναι το δεύτερο πιο χρονικά κοστοβόρο επίπεδο μετά το συνελικτικό.

Dropout

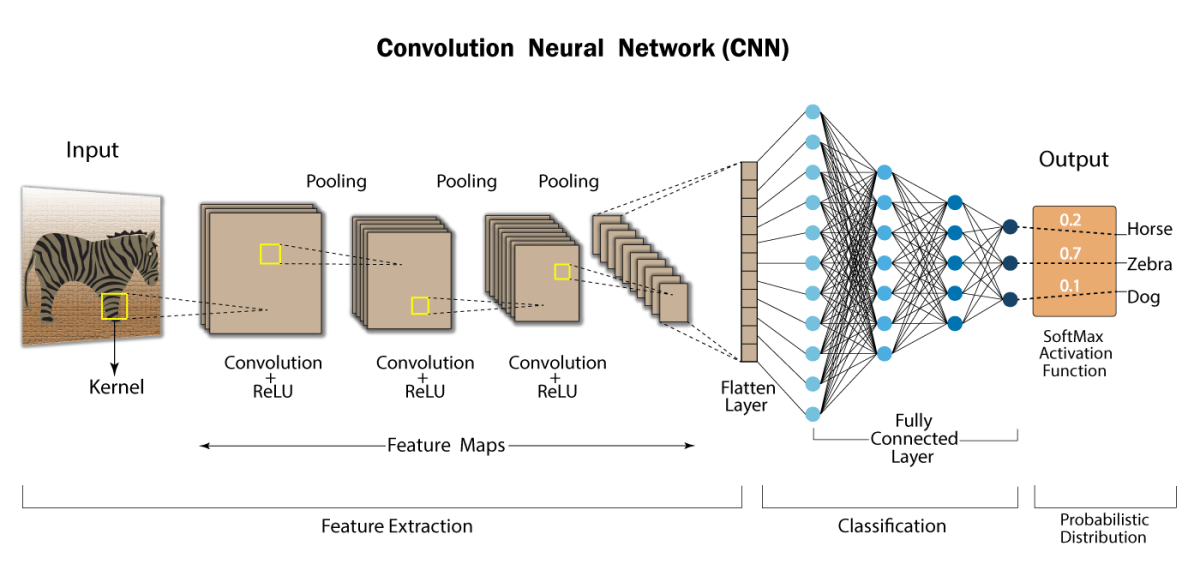
Χρησιμεύουν για να ξεπεραστεί το πρόβλημα της υπερεκπαίδευσης του μοντέλου, που οφείλεται στην υπερβολική απομόνωση χαρακτηριστικών της εικόνας. Το επίπεδο αυτό προσδίδει ποικιλομορφία στα χαρακτηριστικά, αφαιρώντας ένα ποσοστό από τις συνδέσεις τυχαία κατά την εκπαίδευση. Έτσι αποτρέπει το δίκτυο από το να «μαθαίνει» πολύ συγκεκριμένες αντιστοιχίσεις μεταξύ προτύπων εισόδου και εξόδου, και και πλεονασμό πάνω στα εκπαιδεύσιμα βάρη.



Συνοπτικά, η εικόνα[] παρουσιάζει επιμελώς την τυπική αρχιτεκτονική ενός συνελικτικού.

Αποτελείται από δύο βασικά μπλοκ με ξεχωριστές λειτουργίες, Το πρώτο μπλοκ εξάγει συγκεκριμένα χαρακτηριστικά από την εικόνα εισόδου μέσω συνελικτικού φιλτραρίσματος.

Το δεύτερο μπλοκ είναι ένα τετριμμένο πολυστρωματικό perceptron, τα βάρη του οποίου προκύπτουν από το γνωστό αλγόριθμο οπισθοδιάδοσης, όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο.

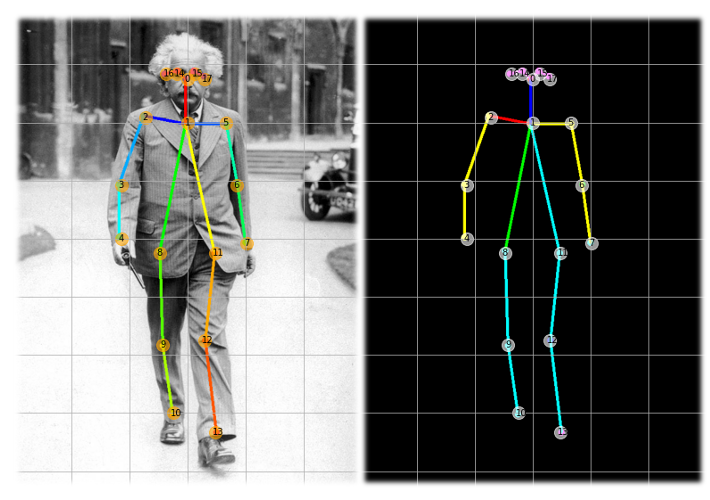


3.5 Μοντέλα Εκτίμησης Στάσης Σώματος

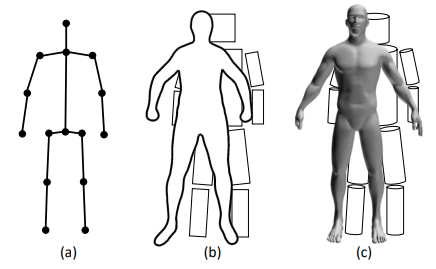
3.5.1 Εκτίμηση ανθρώπινης στάσης σώματος

Η εκτίμηση της στάσης του σώματος (Human Pose Estimation) εδώ και δεκαετίες αποτελεί αντικείμενο εκτενούς μελέτης στο τομέα της υπολογιστικής όρασης (computer vision). Πρόκειται για αναπαράσταση του ανθρώπινου σώματος με αρθρικές συντεταγμένες (key points) από εικόνες ή βίντεο.

Αυτή επιτυγχάνεται με τον εντοπισμό των θέσεων καίριων σημείων (key points) του σώματος, όπως οι αρθρώσεις και το κεφάλι, όπως φαίνεται στην εικόνα\*.



Ανάλογα με τον τρόπο σύζευξης των καίριων σημείων υφίστανται διάφορα μοντέλα απεικόνισης της σκελετικής δομής.   
Τα πιο συνηθισμένα μοντέλα απεικόνισης είναι [https://arxiv.org/pdf/2006.01423.pdf]:  
(a) Κινηματικά ή σκελετοκεντρικά μοντέλα (Kinematic or skeleton-based model). Αποτελούνται από σημεία ή ζεύγη σημείων. Εφαρμόζεται για 2D και 3D εκτιμήσεις.  
(b) Επίπεδα ή καμπυλοειδή μοντέλα για (Planar or contour-based models): αναπαρίστανται τα μέρη του σώματος ως ορθογώνια κουτιά. Εφαρμόζεται μόνο για 2D εκτιμήσεις.  
(c) Βαθμονομημένα ή ογκομετρικά μοντέλα (Volumetric or volume-based models)   
Πρόκειται για αναλυτικό μοντέλο που εφαρμόζεται για 3D εκτιμήσεις.



Υπάρχει ουσιαστική πρόοδος στον εντοπισμό της σκελετικής δομής από 2D εικόνες, με ποσοστά επιτυχούς εντοπισμού γύρω στο 90% για όλα τα σημεία του σώματος. []

Η πρόοδος οφείλεται εν πολλοίς στα συνελικτικά νευρωνικά δίκτυα και σε διαθέσιμες μεγάλες βάσεις δεδομένων, απ’ όπου προέρχονται οι εικόνες για την εκπαίδευσή τους.

Πρόσφατα, ιδιαίτερο ενδιαφέρον υπάρχει στην επέκταση των αλγορίθμων σε 3D εικόνες, με πιο γνωστά μοντέλα VideoPose3D [] , BlazePose [].

Από αλγοριθμική σκοπιά, υπάρχουν δύο βασικές μέθοδοι προσέγγισης του προβλήματος[https://www.scholars.northwestern.edu/en/publications/using-deep-neural-networks-for-inverse-problems-in-imaging-beyond]:

- Πιθανοτικοί αλγόριθμοι γράφων που αποτυπώνουν τις εξαρτήσεις μεταξύ των αρθρώσεων

- Βαθιά συνελικτικά δίκτυα που εκπαιδεύονται να αναγνωρίζουν τις εξαρτήσεις εκπαιδευμένα σε μεγάλο πλήθος δεδομένων

Η πρώτη κατηγορία αλγορίθμων, που είναι η παλαιότερη, βασίζεται σε υλοποιήσεις δενδρικών αλγορίθμων [143,164,174,196] και χρήση Pictorial Structures [https://www.semanticscholar.org/paper/The-Representation-and-Matching-of-Pictorial-Fischler-Elschlager/719da2a0ddd38e78151e1cb2db31703ea8b2e490]. Λειτουργούν ικανοποιητικά για 2D εικόνες όταν οι αρθρώσεις είναι ευδιάκριτες, σε διαφορετική περίπτωση σφάλλουν σημαντικά. Αναπτύχθηκαν διάφορες μέθοδοι αντιμετώπισης της αστοχίας (histogram oriented gaussian (HOG), contours, histograms)[], ωστόσο το κλασσικό μοντέλο στερείται απόδοσης, δυνατότητας συσχέτισης και γενίκευσης.

Έτσι, μετά τη πρωτότυπη παρουσίαση του μοντέλου βαθιάς μάθησης DeepPose [https://arxiv.org/pdf/1312.4659v3.pdf] το 2014, επικράτησαν τα βαθιά συνελικτικά δίκτυα με δύο κυρίαρχες αρχιτεκτονικές εκπαίδευσης:  
- Παλινδρόμισης (regression-based): όπου συγκεκριμένα υπο-στιγμιότυπα της εικόνας εξόδου ανατροφοδοτούνται στην είσοδο ώστε να αυξηθεί η ακρίβεια.   
- Εστιασμού (Detection-based): όπου τα επιμέρους καίρια σημεία φιλτράρονται περεταίρω μέσω θερμικού χάρτη (Heatmap-based) ώστε να μειωθεί το σφάλμα.

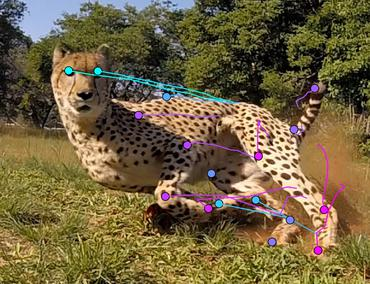
Τα γνωστότερα μοντέλα εξειδικευμένα για 2D Human Pose estimation φαίνονται στο πίνακα[].

| Model | Type of estimations | Variants |
| --- | --- | --- |
| [OpenPose](https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/) | Multi-person | Lightweight |
| [MoveNet](https://storage.googleapis.com/movenet/MoveNet.SinglePose%20Model%20Card.pdf) | Single | Ligthning / Thunder |
| [PoseNet](https://arxiv.org/pdf/1505.07427.pdf) | Single/Multi-person | ResNet50/MobileNetV1 |
| [DensePose](http://densepose.org/) | Single/Multi-person | - |
| [AlphaPose](https://opensourcelibs.com/lib/alphapose) (RMPE) | Single/ Multi-person | - |
| [BlazePose](https://google.github.io/mediapipe/solutions/pose) GHUM | Single | Heavy /Full /Light |
| [DCPose](https://github.com/Pose-Group/DCPose) | Single/Multi-person | - |
| [TransPose](https://arxiv.org/abs/2012.14214) (arch) | Single | - |
| [DeepPose](https://arxiv.org/abs/1312.4659) (arch) | Single | - |
| [HRNet](https://arxiv.org/abs/1908.07919) (arch) | Single | HigherHRNet /  HigherHRNet+ |
| [DeepCut](https://arxiv.org/abs/1511.06645) (arch) | Multi-person | DeeperCut |

Με πράσινο χρώμα είναι εκείνα που διαθέτουν επίσημη υλοποίηση και μπορούν να κληθούν από γνωστά APIs ( keras, googleapis κτλ). Με πορτοκαλί αυτά που βρίσκονται υπό υλοποίηση ή υπάρχουν πολλές διαφορετικές.

3.5.2 Εκτίμηση στάσης σώματος σε ζώα

Αντίστοιχα μοντέλα εκτίμησης στάσης σώματος έχουν αναπτυχθεί και για το ζωικό βασίλειο, με τις εφαρμογές ωστόσο να είναι περιορισμένες σε σχέση με τον άνθρωπο, λόγω της έλλειψης αντίστοιχου ενδιαφέροντος. Ενδεικτικά αξίζει να αναφερθεί το DeepLabCut [<http://www.mackenziemathislab.org/deeplabcut>] το οποίο αποτελεί ένα ανοικτού κώδικα συνελικτικό νευρωνικό δίκτυο (CNN), με δυνατότητα εκτίμησης πόζας διάφορων ζώων, μεταξύ των οποίων κατοικιδίων, ψαριών, τρωκτικών, και ερπετών. Ένα ενδιαφέρον στοιχείο που το διαφοροποιεί από τα προηγούμενα μοντέλα που εφαρμόζονται για την αναγνώριση της ανθρώπινης στάσης σώματος είναι ότι προαιρετικά περιλαμβάνει ένα επιπλέον στρώμα προκειμένου να αναγνωρίζει την ταυτότητα του ζώου. Φυσικά αυτό εισάγει επιπλέον πολυπλοκότητα στο μοντέλο.



# Κεφάλαιο 4

Σχεδίαση Εφαρμογής και Προγραμματιστικά Εργαλεία

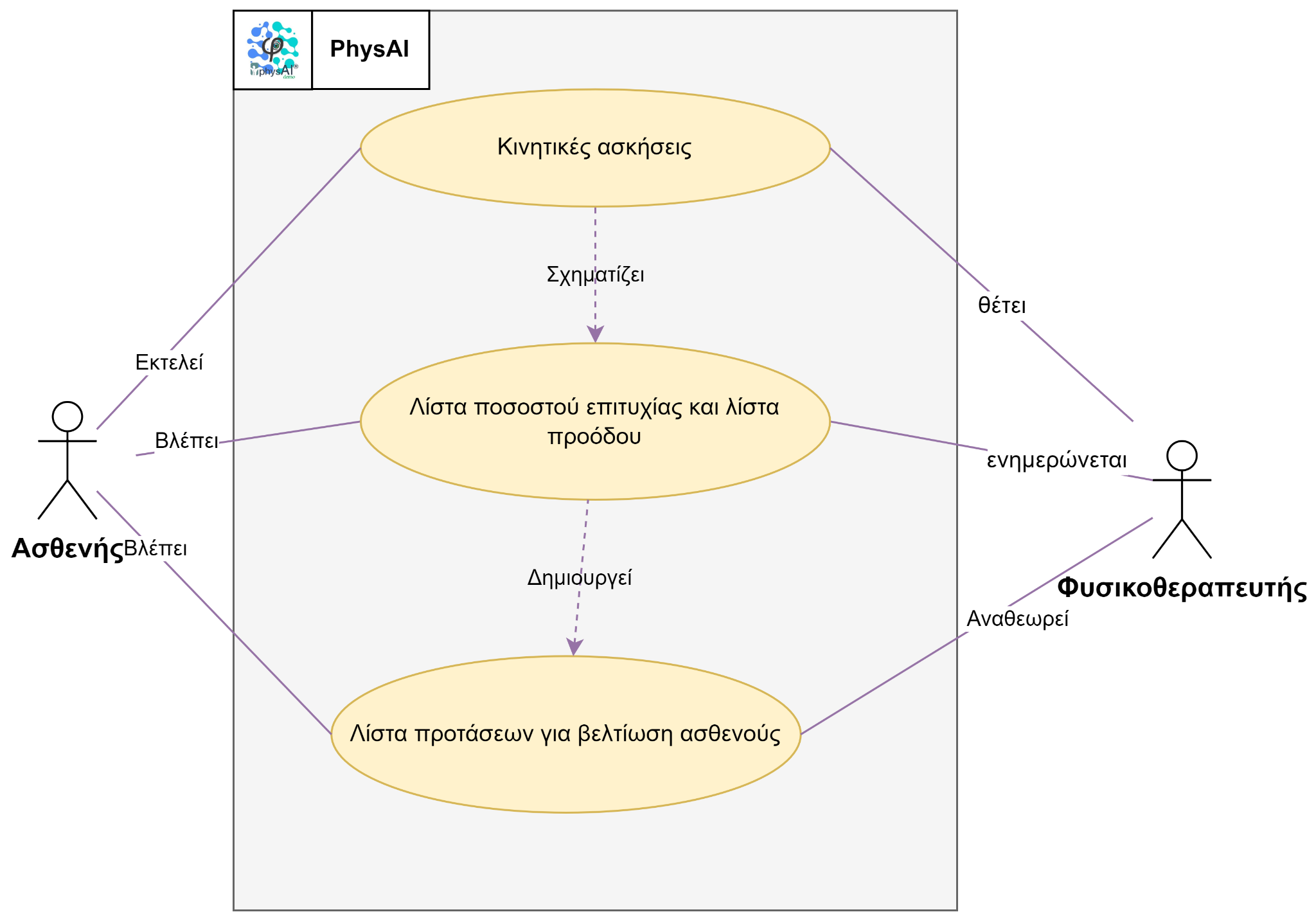
**4.1 Αρχιτεκτονική εφαρμογής**

4.1.1 Γενικές προδιαγραφές εφαρμογής - Διάγραμμα περιπτώσεων χρήσης

Μία από τις πρώτες ενέργειες κατά την έναρξη ενός έργου είναι η ανάλυση των προδιαγραφών και των λειτουργικών απαιτήσεων της εφαρμογής. Φυσικά αυτό συνέβη και στην δική μας περίπτωση κατά την επιλογή του θέματος της διπλωματικής εργασίας, Όπως θα δούμε στο κεφάλαιο 5, για τις ευέλικτες μεθοδολογίες, οι απαιτήσεις αυτές μπορούν να τροποποιηθούν κατά τη διάρκεια της υλοποίησης, και οι σύγχρονοι προγραμματιστές θα πρέπει να είναι σε θέση να αποδέχονται τις αλλαγές, ωστόσο οι γενικές προδιαγραφές παραμένουν κατά κύριο λόγο σταθερές, καθώς περιγράφουν τον βασικό στόχο της εφαρμογής. Στην περίπτωση του PhysAI σκοπός είναι η δημιουργία μιας εφαρμογής για υπολογιστές και κινητά τηλέφωνα, η οποία θα είναι σε θέση να αναγνωρίζει την κινητική άσκηση που εκτελεί ένας χρήστης, τις επαναλήψεις της και να συμπεραίνει πόσο καλά την εκτέλεσε, λαμβάνοντας ως είσοδο την εικόνα από την κάμερα του χρήστη. Η εφαρμογή αυτή θα πρέπει να λειτουργεί αποδοτικά σε κινητά τηλέφωνα, και να παράγει αποτελέσματα σε ζωντανό χρόνο. Προαιρετικά η εφαρμογή θα πρέπει να μπορεί να εντοπίσει τους λόγους για τους οποίους ένας χρήστης δεν εκτέλεσε σωστά μία κινητική άσκηση και να προτείνει λύσεις. Παράλληλα θα πρέπει να χρησιμοποιεί την υπολογιστική νέφους (περισσότερα στο κεφάλαιο 5). Όλα τα παραπάνω θα πρέπει φυσικά να γίνονται με σεβασμό στα προσωπικά δεδομένα και το ιατρικό απόρρητο των χρηστών.

Στην πλήρη έκδοση της εφαρμογής θα αναμέναμε από το λογισμικό να αναγνωρίζει ένα μεγάλο πλήθος από διαφορετικές ασκήσεις, στην περιορισμένη ωστόσο κλίμακα της διπλωματικής εργασίας, η εφαρμογή μας αναγνωρίζει συνολικά 4 κινήσεις/στάσεις σώματος: την όρθια στάση (stand), το κάθισμα (squat), την κάμψη (pushup), και την άρση (pushjerk). Οι 4 αυτές ασκήσεις επιλέχθηκαν με κριτήριο την ευκολία στην εκπαίδευση του νευρωνικού δικτύου, και λόγω της υψηλής διαθεσιμότητας αντίστοιχων δεδομένων στο διαδίκτυο.

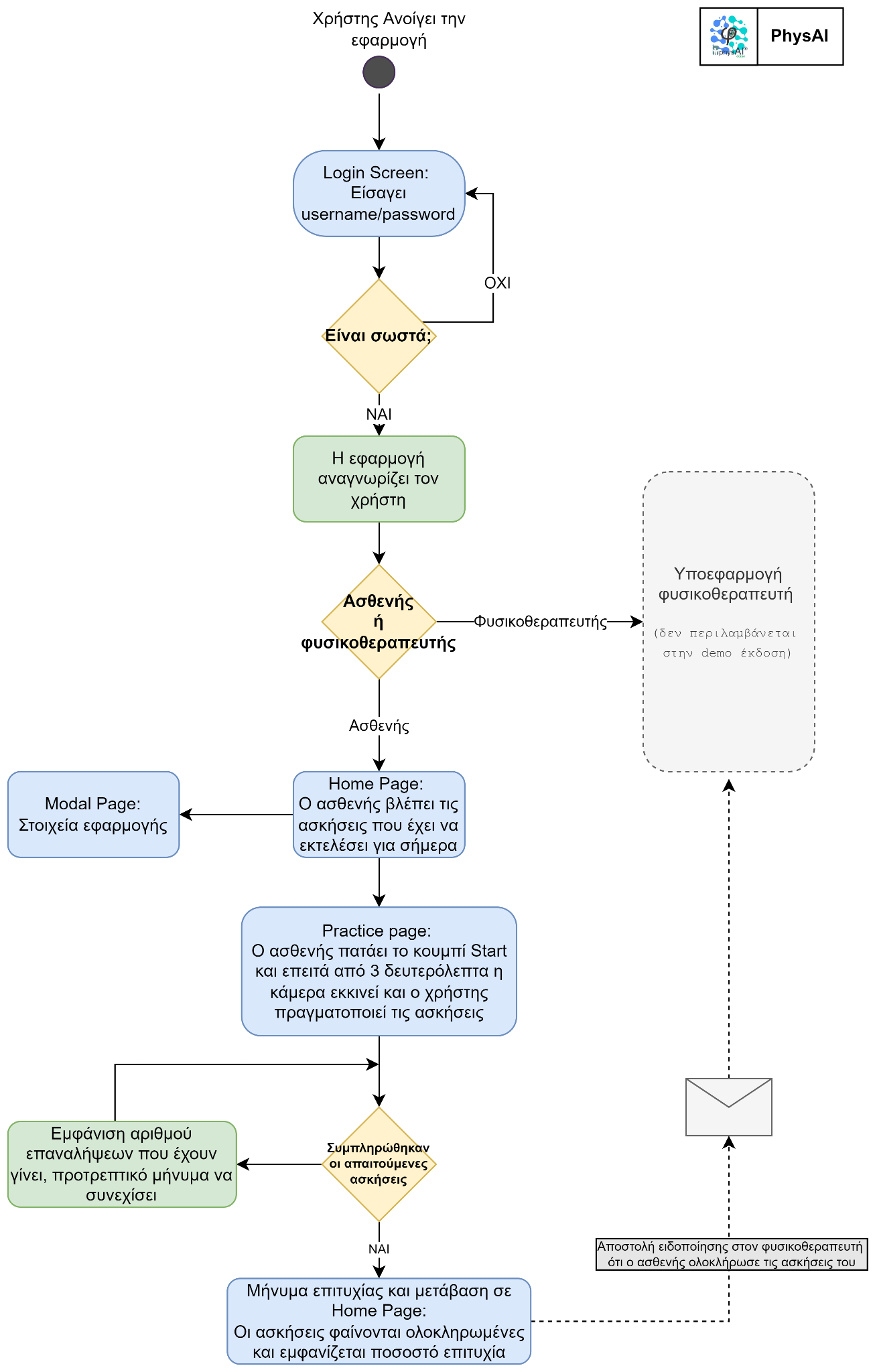
Από την άλλη, στην πλήρη της έκδοση η εφαρμογή διαθέτει βάση δεδομένων στην οποία θα αποθηκεύει τα δεδομένα για τους χρήστες της, τους φυσικοθεραπευτές και τους ασθενείς. Ο ο φυσικοθεραπευτής θα μπορεί να θέτει ασκήσεις στον ασθενή και να λαμβάνει καθημερινές ενημερώσεις για την πρόοδο του. Η αναμενόμενη συμπεριφορά της πλήρους έκδοσης της εφαρμογής περιγράφεται από το παρακάτω διάγραμμα περιπτώσεων χρήσης (use case diagram):



Η δική μας έκδοση υποσημειώνεται ως δοκιμαστική (demo) καθώς εστιάζει κυρίως στην απόδειξη της εφικτότητας (proof of concept) της αναγνώρισης και ανάλυσης κινητικών ασκήσεων από συσκευές smartphone, και στην αντίστοιχη μελέτη απόδοσης.

4.1.2 Διάγραμμα ακολουθίας ενεργειών

Τα διαγράμματα ακολουθίας ενεργειών (activity diagram) αποτελούν μία προσπάθεια για μεγαλύτερη τεχνική ανάλυση των επιμέρους ενεργειών που περιγράφονται στα διαγράμματα περιπτώσεων χρήσης. Παρακάτω παρουσιάζεται το διάγραμμα ακολουθίας για την περίπτωση που ο χρήστης συνδέεται στην εφαρμογή και εκτελεί τις καθημερινές ασκήσεις που του έχει θέση ο φυσικοθεραπευτής.



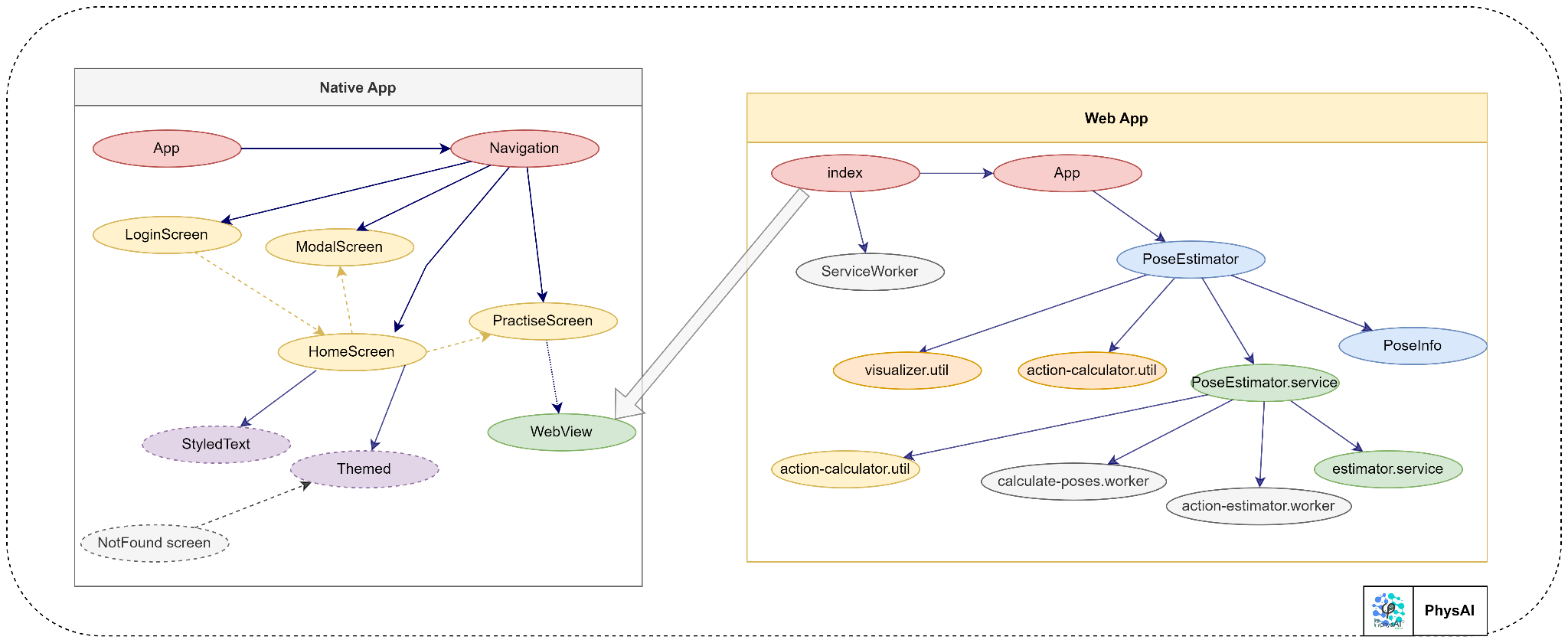
4.1.3 Συστατικά εφαρμογής - Διάγραμμα συστατικών

Δομικοί λίθοι στην σχεδίαση της αρχιτεκτονικής ενός λογισμικού αποτελούν συστατικά (components). Στο πλαίσιο της αντικειμενοστραφούς (object-oriented) τεχνολογίας λογισμικού ένα συστατικό περιέχει ένα σύνολο από μία ή περισσότερες συνεργαζόμενες κλάσει, Κάθε συστατικό συμμετέχει στην επίτευξη ορισμένων στόχων και απαιτήσεων του συστήματος που θα κατασκευαστεί. Μάλιστα πολλές φορές τα συστατικά μίας εφαρμογής οφείλουν να συνεργάζονται με συστατικά/οντότητες που βρίσκονται σε άλλα συστήματα, έξω από τα όρια της εφαρμογής.

Τα components παίζουν ιδιαίτερο ρόλο στην Javascript, την οποία θα μελετήσουμε στη συνέχεια, για το πώς θα επικοινωνούν μεταξύ τους οι διάφορες κλάσεις που επιτελούν τις λειτουργικότητες της εφαρμογές. Μεταξύ των components αναπτύσσονται ιδιαίτερες σχέσεις εξάρτησης (dependency), και κάθε φορά που ένα τμήμα της εφαρμογής θέλει να χρησιμοποιήσει κάποιο component οφείλει να το εισάγει (import) ρητά. Αντίστοιχα στο σημείο που ορίζεται το συγκεκριμένο component θα πρέπει να περιμένουμε μία εντολή εξαγωγής (export).

Για τις εξαρτήσεις από components εξωτερικά της εφαρμογής συνήθως χρησιμοποείται κάποιος διαχειριστής πακέτων (package manager), όπου το πακέτο είναι μία συλλογή από παρεμφερή components. Ένας διαχειριστης πακέτων διευκολύνει τους προγραμματιστές στο διαμοιρασμό, την δημιουργία, την επαναχρησιμοποίηση και το ανέβασμα πακέτων. Οι πιο γνωστοί διαχειριστές πακέτων είναι το npm και το Yarn. Στη δική μας περίπτωση χρησιμοποιήσαμε κυρίως τον δεύτερο, γιατί μπορούσε να αποφύγει καλύτερα κάποια προβλήματα με ασυμβατότητες αρθρωμάτων (modules) που αντιμετωπίσαμε. [<https://waverleysoftware.com/blog/yarn-vs-npm/>]

Τα κύρια components που χρησιμοποιεί η PhysAI, καθώς και οι μεταξύ τους εξαρτήσεις, παρουσιάζονται στο παρακάτω component diagram:

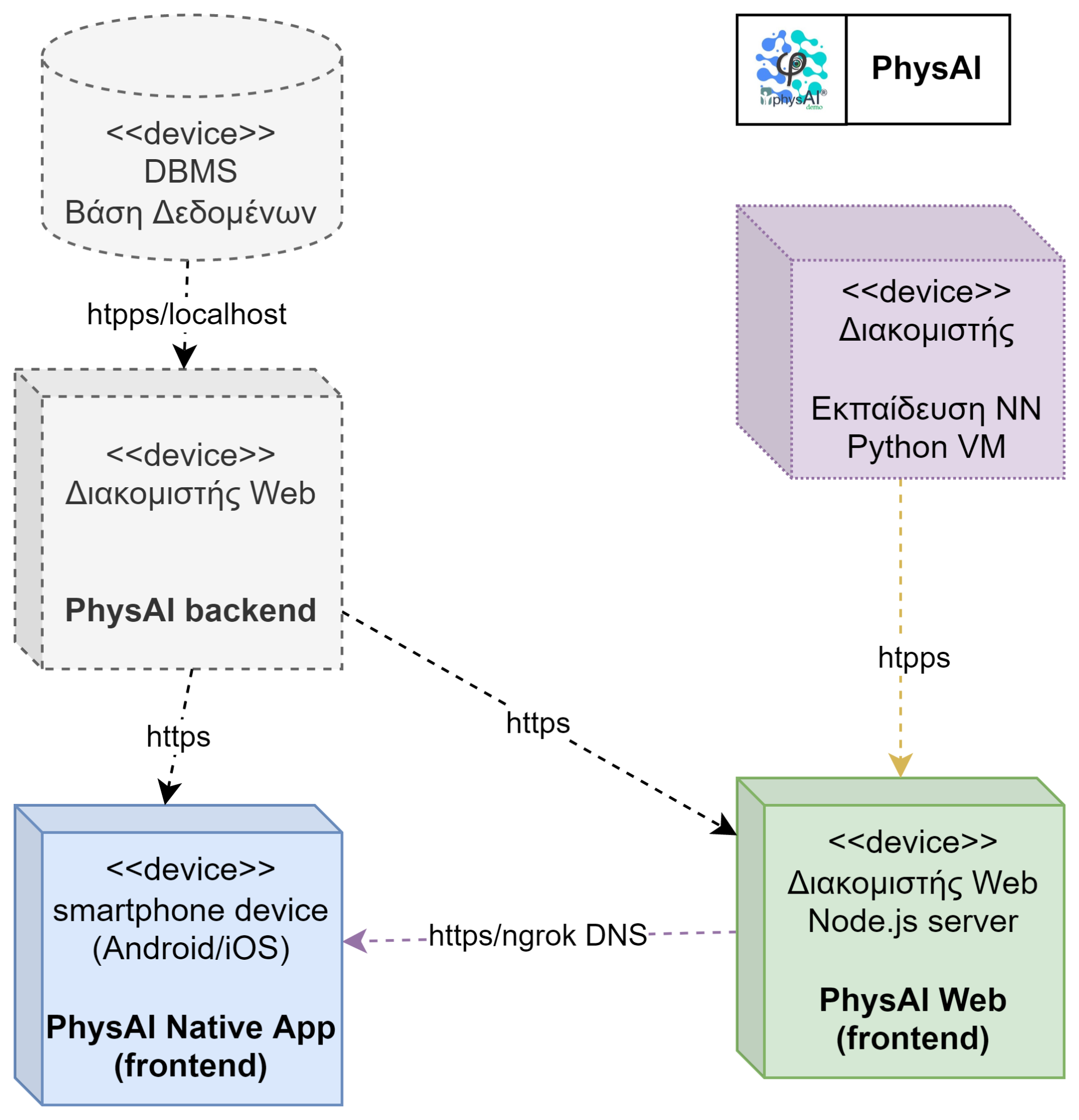


[βιβλίο Pressman, Maxim]

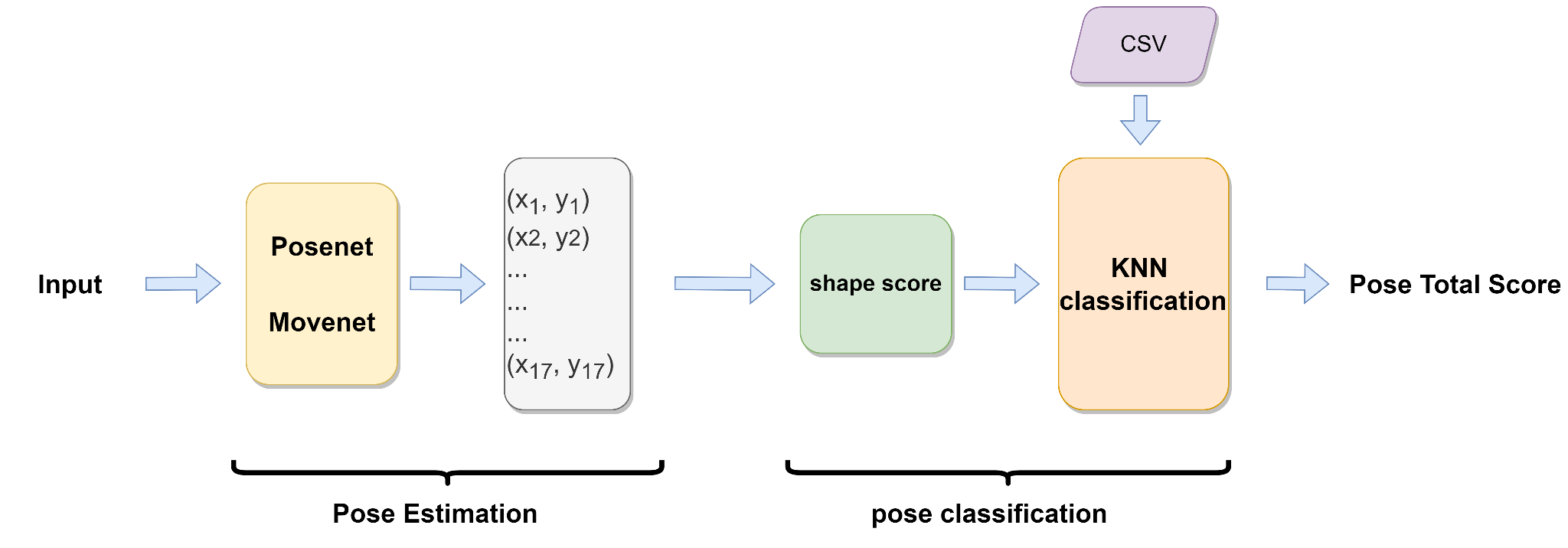
4.1.4 Διάγραμμα Ανάπτυξης

Το διάγραμμα ανάπτυξης (deployment diagram) δίνει έμφαση στη φυσική δομή του συστήματος λογισμικού, δηλαδή στη φυσική κατανομή του μεταξύ πλατφορμών υλικού και περιβαλλόντων εκτέλεσης. Οι διαδρομές επικοινωνίας ανάμεσα στα συστατικά υλικού σχεδιάζονται με γραμμές οι οποίες μπορούν να περιλαμβάνουν σημειώσεις για το πρωτόκολλο επικοινωνίας.

Στη συνέχεια παρουσιάζουμε το διάγραμμα ανάπτυξης για την εφαρμογή PhysAI. Σημειώνουμε με διακεκομμένη διαγράμμιση τα συστατικά εκείνα που αφορούν την πλήρη έκδοση της εφαρμογής, και με συνεχή τη δοκιμαστική. Παράλληλα προσθέσαμε το αυτόνομο τμήμα του PhysAI, το οποίο αφορά την εκπαίδευση των δεδομένων του νευρωνικού δικτύου ταξινόμησης των κινήσεων.



Αξίζει τέλος να σημειώσουμε ότι μεταξύ όλων των components χρησιμοποιούνται συνδέσεις τύπου https, δηλαδή με κρυπτογράφηση. Ο λόγος που γίνεται αυτός είναι προφανής: προστασία προσωπικών δεδομένων.

4.1.5 Λειτουργική δομή

**4.2 Προγραμματιστικά εργαλεία**

4.2.1 Python

4.2.2. Javascript - Typescript

4.2.3 Tensorflow JS

4.2.4 Βελτιστοποίηση

4.2.5 React

4.2.6 React Native

**4.3 Νευρωνικά δίκτυα**

4.2 Προγραμματιστικά Εργαλεία

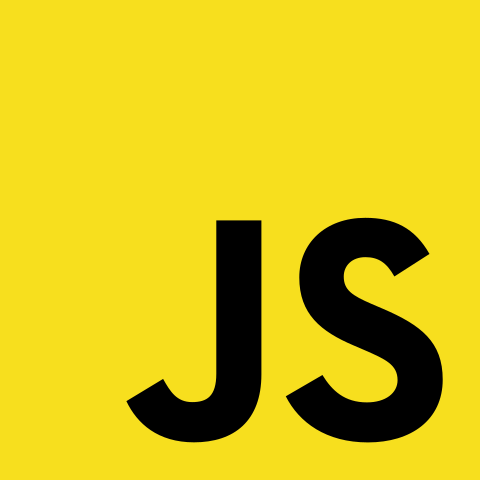
4.2.1 Python

Η ιδέα της Python πρωτοεμφανίζεται το 1989 στην Ολλανδία και συγκεκριμένα στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) από τον Guidovan Rossum. Η Python έχει επιρροές από την γλώσσα προγραμματισμού ABC. Αρχικά δημιουργήθηκε σαν γλώσσα σεναρίων για το κατανεμημένο λειτουργικό σύστημα Amoeba. Σχεδιάστηκε με τέτοιο τρόπο ώστε να επεκτείνεται εύκολα, να παρέχει ενσωματωμένα στοιχεία όπως εντολές και τύπους αντικειμένων και να δίνει την δυνατότητα στους προγραμματιστές να προσθέτουν επιπλέον στοιχεία για την κάλυψη αναγκών των συστημάτων που χρησιμοποιούν. Η πρώτη έκδοση κυκλοφόρησε το 1991 και εξελίσσεται γρήγορα με ανανεωμένες εκδόσεις. Επειδή η γλώσσα είναι συμβατή και με παλαιότερες εκδόσεις, πολλά χαρακτηριστικά της Python 3.0 υιοθετήθηκαν από τις εκδόσεις 2.6 και 2.7. H πιο πρόσφατη έκδοση είναι η 3.10.5. Η Python αποτελεί τη δημοφιλέστερη γλώσσα προγραμματισμού για εφαρμογές σχετικές με επιστήμη δεδομένων και μηχανική μάθηση[Introduction to Machine Learning with Python: A Guide for Data Scientists Sarah Guido**]**

Πρόκειται για μια γλώσσα προγραμματισμού γενικού σκοπού και υψηλού επιπέδου. Η ευκολία εκμάθησης της γλώσσας είναι ο λόγος που επιλέγεται τόσο από αρχάριους προγραμματιστές όσο και από έμπειρους. Η σύνταξη της γλώσσας είναι κομψή και οι τύποι που χρησιμοποιούνται σε αυτήν είναι δυναμικοί. Ο κώδικας ομαδοποιείται σε ενότητες (modules) και πακέτα (packages) και επεξεργάζεται από έναν διερμηνέα (interpreter). H Python είναι κατάλληλη για την ανάπτυξη εφαρμογών απλού εκπαιδευτικού σκοπού μέχρι και ολοκληρωμένες σύνθετες εφαρμογές. Υποστηρίζει πολλές γνωστές προγραμματιστικές προσεγγίσεις όπως η αντικειμενοστρέφεια, ο διαδικαστικός και συναρτησιακός προγραμματισμός, διαθέτοντας δομές δεδομένων υψηλού επιπέδου. Ένας βασικό πλεονέκτημα της Python είναι το πλήθος των βιβλιοθηκών που διαθέτει.Είναι κατάλληλη για επεξεργασία πινακοειδών δομών όπως arrays,tensors, dataframes και παρέχονται ειδικές βιβλιοθήκες τέτοιου σκοπού όπως Numpy, Scipy, Scikit-learn, Pandas. Επιπλέον, παρέχει χρήσιμα εργαλεία για μηχανική μάθηση, όπως είναι η βιβλιοθήκες Tensorflow, PyTorch και το φιλικό προς το χρήστη Keras API για εύκολη αλληλεπίδραση με αυτές.

Επιπλέον, δίνεται η δυνατότητα επέκτασης νέων βιβλιοθηκών σε γλώσσα προγραμματισμού C και C++. Σε σύγκριση με γλώσσες όπως οι C, C++και Java,τα προγράμματα που γράφονται σε Python είναι συμπαγή, ευανάγνωστα και συντηρούνται πιο εύκολα. Υπάρχουν όμως και κάποια μειονεκτήματα. Εφόσον πρόκειται για μια διερμηνευόμενη γλώσσα προγραμματισμού, ο χρόνος εκτέλεσης των προγραμμάτων είναι πιο αργός από άλλες γλώσσες που χρησιμοποιούν μεταγλωττιστές (compilers) αντί για διερμηνείς. Για αυτόν τον λόγο δεν είναι αποδοτική γλώσσα προγραμματισμού για τη δημιουργία λειτουργικών συστημάτων. Αυτό το μειονέκτημα αντισταθμίζεται από το γεγονός ότι ο χρόνος ανάπτυξης μιας εφαρμογής σε γλώσσα Python είναι μικρότερος σε σύγκριση με με C, C++ ή Java.

4.2.2 Javascript - Typescript

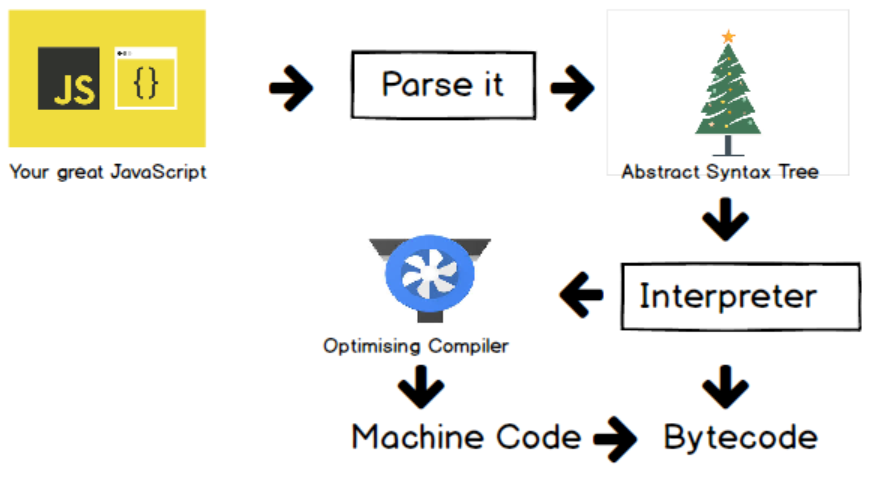


Σύμφωνα με την ετήσια έκθεση της δημοφιλούς πλατφόρμας Stack Overflow, στην οποία προγραμματιστές μπορούν να θέτουν και να απαντούν σε ερωτήματα άλλων χρηστών, η Javascript ψηφίστηκε για το 2021 ως η δημοφιλέστερη γλώσσα, και μάλιστα για 9η συνεχή χρονια. [<https://insights.stackoverflow.com/survey/2021#technology>]. Οι βασικοί λόγοι γι’ αυτή τη δημοφιλία είναι η ευχρηστία της, το πλήθος των διαθέσιμων βιβλιοθηκών, framework και των εργαλείων που υπάρχουν γι’ αυτή καθώς και το γεγονός ότι μπορεί να τρέξει σε πάρα πολλές συσκευές. Παρέχει επομένως τη δυνατότητα για ανάπτυξη σχεδόν κάθε είδους εφαρμογής, αν και οι περισσότεροι την προτιμούν για front-end ανάπτυξη.

Η Javascript πρόκειται για μία διερμηνευτική γλώσσα σεναρίων (script). Ανήκει επιπλέον στην κατηγορία των αντικειμενοστραφών γλωσσών προγραμματισμού, διαθέτει ωστόσο χαρακτηριστικά που παραπέμπουν στον συναρτησιακό προγραμματισμό, δηλαδή υποστηρίζει συναρτήσεις υψηλότερης τάξης. Είναι δυναμική αφού διαθέτει δυναμικό σύστημα τύπων, δηλαδή δεν απαιτεί τον ορισμό του τύπου των μεταβλητών που θα χρησιμοποιηθούν.

Όλοι οι συγχρονοι browsers είναι σε θέση να τρέξουν κάποιον κώδικα Javascript. Όπως όμως ήδη αναφέραμε η Javascript μπορεί να εκτελεστεί και σε άλλες συσκευές, από διακομιστές μέχρι κινητά τηλέφωνα, είναι δηλαδή cross-platform. Για να το επιτύχει αυτό, η Javascript χρησιμοποιεί, όπως και πολλές άλλες cross-platform γλώσσες προγραμματισμού, κάποια εικονική μηχανή, η οποία μεταφράζει τον ενδιάμεσο κώδικα που παρήγαγε ο διερμηνέας σε φυσική γλώσσα μηχανής. Κάθε επομένως σύγχρονος browser και συσκευή περιλαμβάνει και μία τέτοια μηχανή. Γνωστή είναι για παράδειγμα η V8, που ενσωματώνεται στον Chrome, τον Opera και τον Edge, ο Spidermonkey για το Mozilla, ο Nitro για το Safari. [<https://javascript.info/intro>].

Στο παρακάτω σχήμα φαίνονται αδρά τα στάδια από τα οποία περνάει ο κώδικας Javascript από τη γλώσσα υψηλού επιπέδου σε φυσική γλώσσα μηχανής:

[](https://www.digitalocean.com/community/tutorials/js-v8-engine)

Η ύπαρξη των εικονικών μηχανών από τη μεριά της συσκευής του χρήστη είναι πολύ σημαντικό χαρακτηριστικό της Javascript και είναι ένας από τους κύριους λόγους για την επιλογή της συγκεκριμένης γλώσσας για την εφαρμογή μας. Η Javascript εκτελείται στη μεριά του browser, και συνεπώς δεν δημιουργεί επιβάρυνση στο διακομιστή. Αυτό με άλλα λόγια σημαίνει ότι βαριές διαδικασίες, οι οποίες προσθέτουν καθυστερήσεις στο δίκτυο μπορούν να εκτελεστούν απευθείας στις τερματικές συσκευές (edge computing). Ως αποτέλεσμα, εξοικονομείται χρόνος και η συνολική απόκριση του συστήματος είναι καλύτερη.

Στη δική μας περίπτωση η απουσία του edge computing θα καθιστούσε αδύνατη την εκτέλεση της εφαρμογής. Η οπτική αναγνώριση και ταξινόμηση των κινήσεων του χρήστη, που επιτυγχάνεται μέσω της χρήσης νευρωνικού δικτύου, θα ήταν μη αποδοτική σε ένα κεντροποιημένο σύστημα, το οποίο θα απαιτούσε κάθε στιγμιότυπο που συλλέγει ο φακός της κάμερας να αποστέλλεται μέσω διαδικτύου στον διακομιστή. Εκτός των τεράστιων καθυστερήσεων από μεριάς δικτύου, κάτι τέτοιο θα αποτελούσε και σημαντικό κενό ασφαλείας για τα προσωπικά δεδομένα των χρηστών. Ευτυχώς ωστόσο η Javascript διαθέτει πολλές βιβλιοθήκες μηχανικής μάθησης, όπως το Tensorflow.js, που μπορούν να εκτελεστούν στη μεριά του χρήστη.



Ενώ επιλέξαμε τη Javascript για την υλοποίηση της εφαρμογής μας, αυτό είναι εν μέρει ανακρίβεια. Στην πραγματικότητα χρησιμοποιούμε ένα υπερσύνολο της Javascript, την Typescript, η οποία αποτελεί μία πιο αυστηρή συντακτικά εκδοχή της Javascript και προσθέτει προαιρετικούς στατικούς τύπους, όπως κλάσεις και interfaces. [<https://en.wikipedia.org/wiki/TypeScript>] [<https://radixweb.com/blog/typescript-vs-javascript>]. Υπάρχει πλήρης συμβατότητα μεταξύ των δύο γλωσσών. Η υπάρξη στατικών τύπων είναι ιδιαίτερα χρήσιμο χαρακτηριστικό για τη δική μας περίπτωση, καθώς διευκολύνει τη διαδικασία αποσφαλμάτωσης και προλαμβάνει προβλήματα ασυμβατότητας τύπων, τα οποία όπως φάνηκε στην πράξη κατά τη συγγραφή του κώδικα ήταν πολλά. Όπως εξάλλου παρουσιάστηκε προηγουμένως τα συστατικά (components) της εφαρμογής μας ήταν πολυάριθμα, με διάφορες εξαρτήσεις ανάμεσά τους, οπότε ήταν αναμενόμενο να υπάρχουν αρκετές λάθος κλήσεις, οι οποίες ευτυχώς εντοπίστηκαν χάρις στην αυστηρότητα της Typescript.

https://machinelearningknowledge.ai/what-is-tensorflow-js-introduction-for-beginners/

https://javascript.info/intro

4.2.3 Tensorflow js

Το Tensorflow.js, που κυκλοφόρησε από την Google το 2018, είναι η πιο δημοφιλής και ανοιχτού κώδικα βιβλιοθήκη μηχανικής μάθησης σε Javascript. Δίνει τη δυνατότητα στον developer να αναπτύξει, να εκπαιδεύσει και να δοκιμάσει μοντέλα μηχανικής μάθησης είτε τοπικά σε node.js είτε σε browser. Παραδοσιακά, τα μοντέλα βαθιάς μάθησης υλοποιούνταν στη πλευρά του διακομιστή (server-side) κυρίως με βιβλιοθήκες Python. Η σταδιακή όμως ενσωμάτωση τεχνολογιών javascript και κατάλληλων javascript APIs στις μηχανές αναζήτησης για επεξεργασία δεδομένων κειμένου, ήχου και βίντεο αναβάθμισαν τη πλευρά του χρήστη (client-side). Πλέον, μοντέλα βαθιάς μάθησης της βιβλιοθήκης tensorflow μπορούν να «τρέξουν» στη μηχανή αναζήτησης αποδοτικά, η οποία αξιοποιεί πλέον την υπολογιστική μονάδα γραφικών (GPU). Πράγματι, χωρίς τη χρήση της κάρτας γραφικών οι απαιτητικές πράξεις μεταξύ πινάκων σε συνελικτικά δίκτυα δεν θα ήταν εύκολη υπόθεση. Επειδή ακριβώς η κάρτα γραφικών μπορεί να κάνει παράλληλα τις πράξεις κατανέμοντας τους υπολογισμούς σε πολλά νήματα υπερτερεί έναντι της CPU. Το Tensorflow.js καθιστά αυτή τη δυνατότητα εφικτή μέσω του προγράμματος περιήγησης χρησιμοποιώντας το WebGL.

Υπάρχουν τρεις βασικοί τρόποι με τους οποίους ένας προγραμματιστής μπορεί να εκμεταλλευτεί τις δυνατότητες της βιβλιοθήκης:

* για εκπαίδευση νευρωνικών μοντέλων

Το tensorflow.js διαθέτει κατάλληλα APIs παρόμοια με αυτά της βιβλιοθήκης Keras που διευκολύνουν τη αρχικοποίηση και εκπαίδευση των μοντέλων βαθιάς μάθησης στο browser.

* για ενσωμάτωση προ-εκπαιδευμένων μοντέλων

Πρόκειται για φόρτωση και ζωντανή χρήση μοντέλων βαθιάς μάθησης με εκπαιδευμένα βάρη. Μπορεί να είναι έτοιμα μοντέλα όπως τα VGG16, ResNet, DenseNet, MobileNet ή μετατροπή ατομικής υλοποίησης σε άλλη γλώσσα προγραμματισμού (π.χ. Python). Υπάρχει η δυνατότητα μετατροπής του μοντέλου σε tensorflow.js , καθιστώντας το λειτουργικό.

* για μεταφορά μάθησης

Τα περισσότερα μοντέλα που διατίθενται σε βιβλιοθήκες είναι για τετριμμένες διαδικασίες όπως ταξινόμηση εικόνων, αναγνώριση πόζας ή αντικειμένων. Ωστόσο, μπορεί να χρειάζεται ειδική προσαρμογή ενός ήδη υπάρχοντος μοντέλου για απομόνωση ή εντοπισμό συγκεκριμένων χαρακτηριστικών. Αυτό επιτυγχάνεται αφαιρώντας τα τελευταία επίπεδα του δικτύου εκπαιδεύοντας τα με καινούρια δεδομένα. Το tensorflow παρέχει αυτή τη δυνατότητα. H βιβλιοθήκη Tensorflow.js αποτελείται από δύο διαφορετικά πακέτα (packages), το πρώτο είναι ένα ευέλικτο API πυρήνα (Core API) το οποίο διατίθεται αποκλειστικά για δημιουργία νευρωνικών δικτύων και το δεύτερο είναι ένα υψηλού επιπέδου API (layer API) το οποίο παρουσιάζει ομοιότητες με το Keras API.

Πρόσφατα, άλλο ένα χρήσιμο εργαλείο έκανε την εμφάνισή του, το TensorFire[https://tenso.rs/]. Πρόκειται για βιβλιοθήκη που «τρέχει» εκπαιδευμένα νευρωνικά μοντέλα στο browser. Η ειδοποιός διαφορά όμως από το Tensorflow.js είναι ότι δεν χρειάζεται εγκατάσταση ενδιάμεσων APIs για την αλληλεπίδραση με τη GPU και δεν υπάρχουν περιορισμοί ως προς το μοντέλο GPU.

4.2.4 Βελτιστοποίηση επεξεργαστικής ικανότητας

Τις πρώτες δεκαετίες οι περιηγητές διαδικτύου (browser) χρησιμοποιούσαν την ισχύ της κεντρικής υπολογιστικής μονάδας (CPU) για τη φόρτωση και διαχείριση σελίδων, ενώ για την απόδοση 2D/3D περιεχομένου χρειαζόταν να γίνει εγκατάσταση ειδικών επεκτάσεων (π.χ. Flash player, Unity Web player). Αυτό όμως είχε ως συνέπεια κάθε τέτοια επέκταση (plugin) να απαιτεί κατάλληλη υλοποίηση, ώστε να είναι συμβατή με το εκάστοτε λογισμικό (Windows, Linux, MacOS) το οποίο ήταν ασύμφορο. Τα τελευταία όμως χρόνια βιβλιοθήκες και APIs υλοποιημένα σε javascript, όπως το WebGL [https://www.lewuathe.com/custome-webgl-operation-in-tensorflow.js.html], επιτρέπουν την απευθείας αξιοποίηση των πόρων της υπολογιστικής μονάδας γραφικών (GPU) από τον browser. Η εκπαίδευση ενός συνελικτικού νευρωνικού δικτύου αποτελεί χρονοβόρα διαδικασία, με υψηλές απαιτήσεις όσον αφορά την υπολογιστική ισχύ, ειδικά όταν η αρχιτεκτονική αποτελείται από πολλά επίπεδα, καθώς αυξάνεται ο αριθμός των παραμέτρων που πρέπει να υπολογιστούν. Η εκπαίδευση ενός μεγαλύτερου συνελικτικού δικτύου λοιπόν είναι μη πρακτική σε έναν τυπικό υπολογιστή, αξιοποιώντας απλά την υπολογιστική ισχύ της Κεντρικής Μονάδας Επεξεργασίας (CPU). Ευτυχώς, οι υπολογισμοί ενός συνελικτικού νευρωνικού δικτύου είναι πράξεις μεταξύ πινάκων ή τανυστών (tensors) οι οποίες μπορούν να παραλληλοποιηθούν αποδοτικά. Συνεπώς, η παράλληλη υπολογιστική ισχύς των μονάδων επεξεργασίας γραφικών (GPUs ) συντελεί σε ταχύτερη και πιο αποδοτική εκπαίδευση των νευρωνικών δικτύων. Γι’ αυτό το λόγο και τα περισσότερα από τα δημοφιλή frameworks βαθιάς μάθησης υποστηρίζουν επιτάχυνση με χρήση GPU.

Μία άλλη μέθοδος με την οποία οι σύγχρονοι περιηγητές καταφέρνουν να πετύχουν καλύτερες επιδόσεις στην εικόνα που λαμβάνει ο χρήστης από την εφαρμογή, γνωστή και ως user experience (εμπειρία χρήστη), είναι να εκμεταλλεύονται τη κρυφή μνήμη (cache) προκειμένου να μη χρειάζεται να φορτώνουν κάθε φορά την ίδια σελίδα, εάν αυτή δεν έχει υποστεί στο μεταξύ αλλαγές. Στη δική μας περίπτωση, με δεδομένο ότι το νευρωνικό δίκτυο ταξινόμησης των κινήσεων είναι απαιτητικό σε μνήμη, ήταν πολύ σημαντικό να λάβουμε ειδική μέριμνα γι’ αυτό. Έτσι εκμεταλλευτήκαμε μία κατάλληλη βιβλιοθήκη που ορίζει το συστατικό Service Workers, το οποίο επιτρέπει στην εφαρμογή να φορτώνει ταχύτερα, ή ακόμα να λειτουργεί και εκτός σύνδεσης (offline) επειδή χρησιμοποιεί την cache στην εφαρμογή. Βέβαια το αρνητικό της υπόθεσης είναι ότι σε περίπτωση ανανέωσης της σελίδας θα πρέπει κάθε ανοικτή στη σελίδα αυτή καρτέλα να κλείσει, προκειμένου να δούμε την ενημερωμένη έκδοση. Αυτή βέβαια η συνέπεια αφορά κυρίως τους προγραμματιστές κατά τη διάρκεια της κατασκευής του λογισμικού, και όχι τόσο τους χρήστες, κατά τη διάρκεια της λειτουργίας.

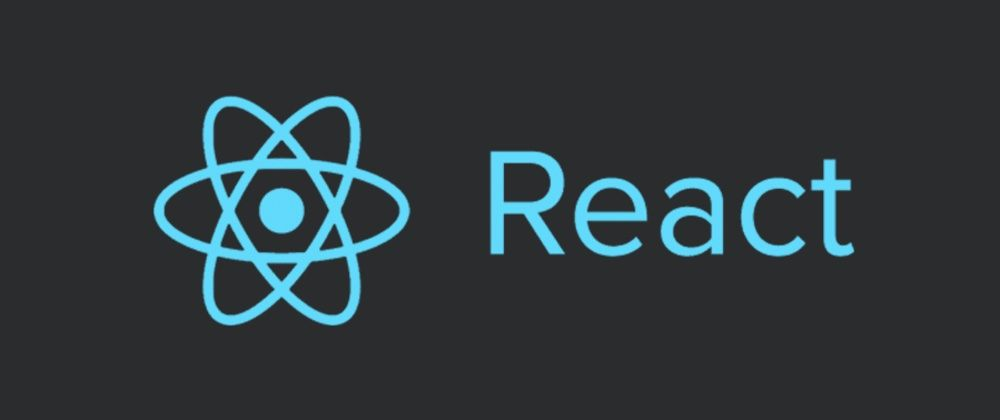
Webgl για tensorflow.js

edgeTPU ειναι hardware, δεν μας νοιαζει

GPU- nVidia CUDA

web workers,

4.2.5 React

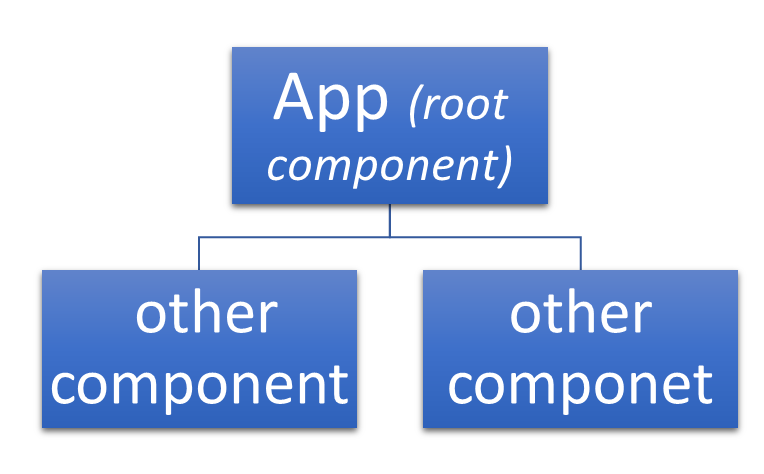


Σύμφωνα πάλι με την ετήσια έκθεση της δημοφιλούς πλατφόρμας Stack Overflow, το React.js ψηφίστηκε για το 2021 ως το δημοφιλέστερο framework για την ανάπτυξη web εφαρμογών, αφήνοντας το jQuery στη δεύτερη θέση. [<https://insights.stackoverflow.com/survey/2021#technology>] Αυτός ο τίτλος ωστόσο είναι κάπως παραπλανητικός για το React, αφού τυπικά δεν είναι ολοκληρωμένο front-end framework, όπως για παράδειγμα είναι τα επίσης δημοφιλή Vue.js και Angular. Το React πρόκειται για μία βιβλιοθήκη javascript για την ανάπτυξη front-end διεπαφών χρήστη στο διαδίκτυο. Ωστόσο μεγάλο μέγεθος της βιβλιοθήκης αυτής, με την προσθήκη κάποιων φαινομενικά αυτόνομων εργαλείων, όπως το react-hook, react-router, καθιστούν το React μία ολοκληρωμένη λύση για την παραγωγή λογισμικού.

Το React δημιουργήθηκε το 2011 από τη Facebook, και γρήγορα απέκτησε μεγάλη απήχηση, λόγω των μεγάλων δυνατοτήτων που παρείχε στους προγραμματιστές. Το React επιτρέπει στους browsers να διαχειρίζονται σύνθετα animations, μεγάλες ποσότητες δεδομένων, και άλλες βαριές υπολογιστικά διαδικασίες, όπως είναι στην περίπτωση μας η εκτέλεση του νευρωνικού δικτύου. Αυτό το επιτυγχάνει γιατί τρέχει στη μεριά του χρήστη ως μία SPA (single page application, εφαρμογή μίας σελίδας). Επίσης μπορεί να συνδυαστεί με άλλες τεχνολογίες στοίβας για να συνδεθεί με έναν διακομιστή μέσω ενός API.

Το React έχει τα εξής κύρια χαρακτηριστικά:

* Είναι δηλωτικό (declarative), σε ό,τι αφορά τη διαχείριση του DOM\*, δηλαδή δε χρειάζεται να ορίσει ο προγραμματιστής επακριβώς τι χρειάζεται το DOM να κάνει, αλλά δηλώνει πώς επιθυμεί να λειτουργεί και το React αναλαμβάνει να εκτελέσει τις απαραίτητες ενέργειες. Αυτό καθιστά το σύστημα πιο εύκολο στην αποσφαλμάτωση, καθώς ο κώδικας γίνεται πιο προβλέψιμος. Παράλληλα το σύστημα γίνεται δυναμικό, δηλαδή ο προγραμματιστής μπορεί να δει απευθείας τις αλλαγές που κάνει στον κώδικα, χωρίς την ανάγκη για νέα μεταγλώττιση. Μάλιστα ακριβώς από αυτή την ιδιότητα παίρνει και το ονομά του, ότι «αντιδρά» σε αλλαγές κατάστασης (reacts to states changes).
* Είναι ανεξάρτητο και επαναχρησιμοποιήσιμο, δηλαδή είναι ανεξάρτητο από την υπόλοιπη τεχνολογική στοίβα και μπορεί να ενσωματώσει νέα χαρακτηριστικά χωρίς να γραφεί εκ νέου ο παλιός κώδικας για το νέο χαρακτηριστικό.
* Είναι component-based, δίνει δηλαδή έμφαση στα συστατικά (components). Κάθε εφαρμογή ουσιαστικά δομείται από συστατικά, τα οποία μπορούν να χρησιμοποιηθούν πολλές φορές μέσα στην εφαρμογή και αναπτύσσουν μεταξύ τους εξαρτήσεις δενδρικής δομής. Το κύριο συστατικό κάθε εφαρμογής στο React είναι το App το οποίο καλεί κάθε άλλο συστατικό. Παρουσιάσαμε τα συστατικά της δικής μας εφαρμογή στο διάγραμμα συστατικών στην αντίστοιχη ενότητα προηγουμένως. Χρειάζεται βέβαια να τονιστεί, ότι η δεντρική δομή τηρείται και στη δική μας περίπτωση, οι διακεκομμένες γραμμές που φαίνεται να την παραβιάζουν στο σχήμα αφορούν εννοιολογικές συσχετίσεις των συστατικών και όχι τυπικές.



Οι λόγοι για τους οποίους μας οδήγησαν να χρησιμοποιήσουμε το React είναι κυρίως οι υψηλές σχεδιαστικές δυνατότητες που παρέχει, και η πρότερη εξοικείωση μας, όπως επίσης ότι θα το χρησιμοποιούσαμε σε συνδυασμό με το React Native για την εφαρμογή για κινητά.

\*DOM: document object model: μία προγραμματιστική διεπαφή η οποία βοηθάει στην απεικόνιση μιας διαδικτυακής ιστοσελίδας HTML σε μία δεντρική δομή αντικειμένων (object-oriented)

[<https://reactjs.org/>]

4.2.6 React Native



Παραδοσιακά για την ανάπτυξη εφαρμογών σε smartphones απαιτούνταν ξεχωριστά εργαλεία ανάλογα με το λειτουργικό σύστημα (Android, iOS κτλ). Κάτι τέτοιο ωστόσο καθυστερούσε πολύ την παραγωγή προγραμμάτων και αύξανε σημαντικά το κόστος. Η σταδιακή αύξηση της επεξεργαστικής ισχύος των κινητών επέτρεψε τη χρήση εικονικών μηχανών και τεχνικών παρα-εικονικοποίησης (paravirtualization), που είναι σε θέση να εξουδετερώνουν σε υψηλό προγραμματιστικό επίπεδο τις διαφορές των φορητών συσκευών προκειμένου να είναι εφικτή η συγγραφή μίας μόνο εφαρμογής που θα είναι συμβατή με όλες. Ένα από τα πρώτα frameworks σε αυτή την κατεύθυνση ήταν το React Native, το οποίο μάλιστα έχει ως σύνθημα τη φράση «Learn once, write anywhere» («μάθε άπαξ, γράψε παντού»).

Το React Native είναι framework που επιτρέπει τη συγγραφή κώδικα που εκμεταλλεύεται τις φυσικές συσκευές των Android και iOS κινητών, ώστε να μπορεί με αποδοτικό τρόπο να αποδίδει τα διάφορα εικονικά στοιχεία (rendering). Αναπτύσσεται και αυτό από τη Facebook, και όπως προδίδει το όνομά του βασίζεται στο React, που ανφέραμε προηγουμένως.

Παρόμοια με το React για το διαδίκτυο, στο React Native οι εφαρμογές γράφονται και εκτελούνται σε Javascript. Η React Native στη συνέχεια γεφυρώνει το χάσμα καλώντας με τις φυσικές συσκευές, κάνοντας κατάλληλες κλήσεις στο API του Android, που είναι γραμμένο σε Java, ή του iOS, που είναι γραμμένο σε Objective-C. Έτσι το React Native μπορεί να παρέχει «native» δυνατότητες στους προγραμματιστές, όπως για παράδειγμα να χρησιμοποιήσουν την κάμερα του κινητού, την τοποθεσία της συσκευής, το σύστημα ειδοποιήσεων, τη δόνηση κ.α.

Ο λόγος που επιλέξαμε επομένως το React Native για την εφαρμογή μας ήταν, αφενός η αυξημένη απόδοση που θα παρέχει στο σύστημα, την οποία τη χρειαζόμαστε για να τρέξει το νευρωνικό δίκτυο, και αφετέρου η άμεσή της συνάφεια με το React μας επιτρέπει με μεγάλη ευκολία να εργαστούμε παράλληλα στις εφαρμογές (Web και native) που αναπτύσσουμε.

[<https://reactnative.dev/>] [<https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>]

4.4 Νευρωνικά δίκτυα στο PhysAI

Σκοπός της εργασίας είναι η δημιουργία αρχιτεκτονικής που μπορεί να επιβεβαιώσει αν ο χρήστης ακολουθεί επαρκώς τις κινητικές οδηγίες του ιατρού. Συνεπώς, απαραίτητο εργαλείο είναι τα συνελικτικά δίκτυα εκτίμησης στάσης σώματος (Human Pose Estimation).

Όπως αποτυπώθηκε στο Κεφάλαιο 3, υπάρχουν πολλά μοντέλα και βιβλιοθήκες νευρωνικών δικτύων για εκτίμηση της στάσης του σώματος. Καθένα από αυτά έχει πλεονεκτήματα και μειονεκτήματα σε αντίστοιχες εφαρμογές και τομείς.

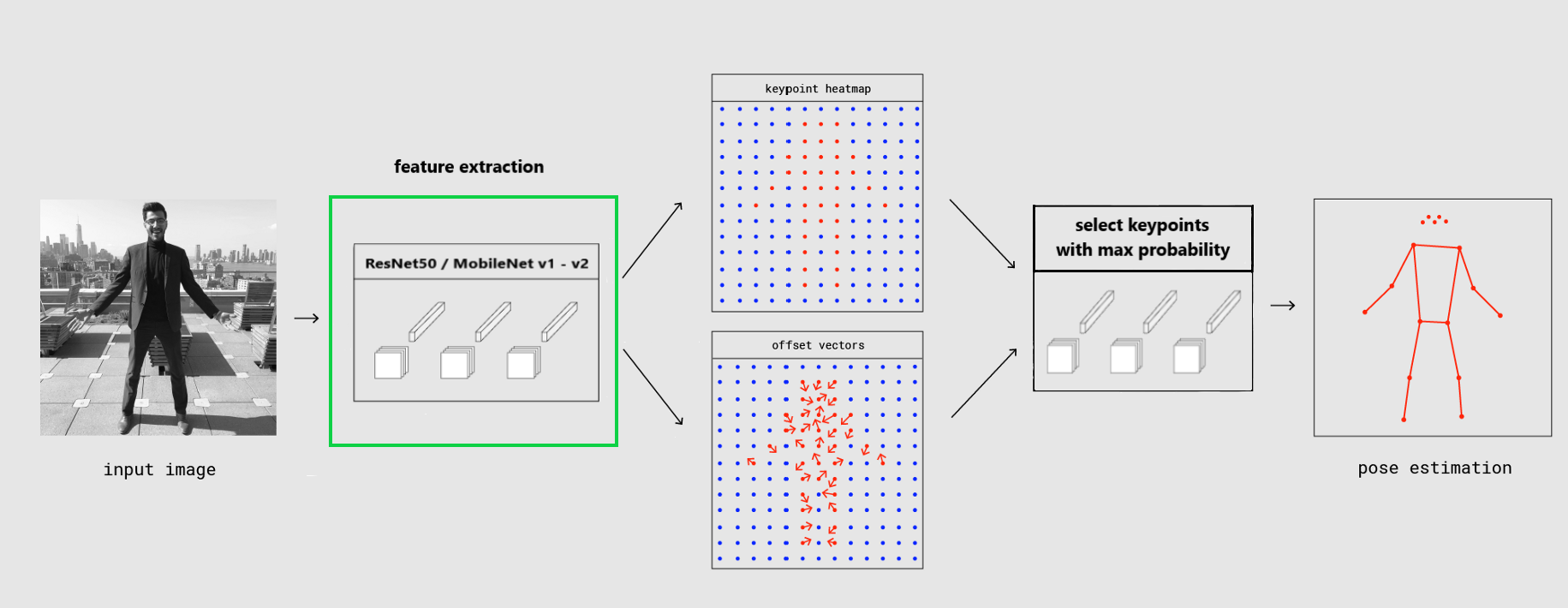
Ύστερα από εκτενή μελέτη καταλήξαμε στην αξιοποίηση δύο προ-εκπαιδευμένων μοντέλων νευρωνικής μάθησης στην εφαρμογή μας. Του posenet και του Movenet.

Ο λόγος που τα επιλέξαμε είναι η συμβατότητά τους με τα λογισμικά κινητών συσκευών (Android και iOS) και παραμετρικές ιδιότητες που τα καθιστούν εύκολα στη χρήση.

Η αρχιτεκτονική των επιλεγμένων μοντέλων αποτελείται από δύο αφαιρετικά επίπεδα (encoder-decoder architecture) όπως φαίνεται στην εικόνα [].

* Στο πρώτο επίπεδο γίνεται εντοπισμός των πιθανών περιοχών όπου μπορεί να υπάρχουν καίρια σημεία. Αυτό επιτυγχάνεται με χρήση συνελικτικού νευρωνικού δικτύου. Η αποτύπωση αυτών των περιοχών γίνεται σε θερμικούς χάρτες (heatmaps) και με τη βοήθεια offset διανυσμάτων (offset vectors) που προσδίδουν πιθανοτική ακρίβεια.
* Στο δεύτερο επίπεδο φιλτράρονται τα pixels της εικόνας στα οποία η πιθανότητα ύπαρξης καίριου σημείου είναι μέγιστη. Αυτό επιτυγχάνεται με επιπλέον φίλτρα.

Τελικά, αποτυπώνονται τα καίρια σημεία και η προβλεπόμενη σκελετική δομή του ατόμου.



Στη συνέχεια θα παρουσιάσουμε τα δύο βασικά μοντέλα: την αρχιτεκτονική, τις ιδιότητές και τις υπερ-παραμέτρους που διαθέτουν. Επιπλέον, θα τα συγκρίνουμε ως προς την απόδοση και τις δυνατότητές τους.

**Posenet (ή PersonLab)** [https://arxiv.org/pdf/1803.08225.pdf ]

Μοντέλο βαθιάς μάθησης ανοιχτού κώδικα που ειδικεύεται στην ανίχνευση της ανθρώπινης στάσης (πόζας) για μεμονωμένους ή περισσότερους ανθρώπους ταυτόχρονα. Έχει εκπαιδευτεί στο σύνολο εικόνων COCO και δημοσιεύτηκε το 2017. Βασίζεται στη βιβλιοθήκη Tensorflow και εντοπίζει σε πραγματικό χρόνο 17 σημεία, τα οποία όταν συζευχθούν, αποτυπώνουν τη τρέχουσα σκελετική αναπαράσταση του σώματος. Παρέχει και ποσοστό ευστοχίας, δηλαδή πόσο κοντά στη πραγματική σκελετική δομή είναι η πρόβλεψή του. Όλες οι πληροφορίες σχετικά με την υλοποίηση και τις παραμέτρους του είναι δημόσια προσβάσιμες μέσω github[https://github.com/google-coral/project-posenet].

Δομή

Είσοδος: RGB εικόνα

Παράμετροι:

Το μοντέλο για μεμονωμένα άτομα λαμβάνει τις εξής παραμέτρους:

* + Παράγοντα κλιμακοσιμότητας εικόνας
  + Οριζόντιος καθρεφτισμός εικόνας
  + Stride εξόδου

Ενώ αυτό για πολλαπλές ανιχνεύσεις:

* Μέγιστος αριθμός στάσεων ως προς ανίχνευση
* Κατώτατο όριο ευστοχίας
* Ακτίνα NMS

Έξοδος:

* Keypoints με συντεταγμένες (x,y)
* Score βεβαιότητας για το εκάστοτε keypoint και συνολικά ανα άτομο

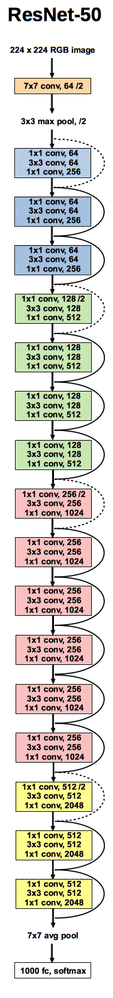
Ο αλγόριθμος ανίχνευσης στάσης σε μεμονωμένο άτομο είναι ταχύτερος και απλούστερος από τον αλγόριθμο ανίχνευσης πολλών ατόμων. Επίσης, η αρχική έκδοση της εφαρμογής προσφέρεται για ατομική χρήση. Επομένως, θα εντρυφήσουμε/εστιάσουμε στις αρχιτεκτονικές για μεμονωμένη ανίχνευση στάσης.

Το μοντέλο Posenet έχει προσέγγιση «από πάνω προς τα κάτω» (top-down) και διαθέτει δύο επιλογές ως προς την αρχιτεκτονική του συνελικτικού δικτύου:   
- ResNet50

- MobileNetV1

Στη συνέχεια θα παρουσιαστούν αναλυτικά τα δύο δίκτυα και θα γίνει σύγκριση μεταξύ τους για το σκοπό της εργασίας.

**ResNet50**



Το ResNet [https://arxiv.org/abs/1512.03385] προέκυψε από την ανάγκη διόρθωσης του γνωστού προβλήματος της «φθίνουσας παραγώγου»\*.

Από τη πρώτη εμφάνιση του συνελικτικού δικτύου AlexNet (2012), το οποίο κέρδισε το διαγωνισμό ταξινόμησης των εικόνων Imagenet (2012) , διαπιστώθηκε ότι αύξηση των συνελικτικών επιπέδων συνεπάγεται και συνολική βελτίωση επίδοσης του δικτύου. Είναι λογικό, αφού περισσότεροι παράμετροι επιτρέπουν εκμάθηση πιο πεπλεγμένων χαρακτηριστικών της εικόνας.

Εντούτοις, υπάρχει ένα πάνω φράγμα στον αριθμό των επιπέδων που μπορούν να προστεθούν, καθώς αν ξεπεραστεί αυτό οι τιμές των παραγώγων φθίνουν γρήγορα έως ότου μηδενιστούν. Συνεπακόλουθα, επέρχεται κορεσμός στην απόδοση του μοντέλου και το σφάλμα αυξάνεται πολύ γρήγορα.

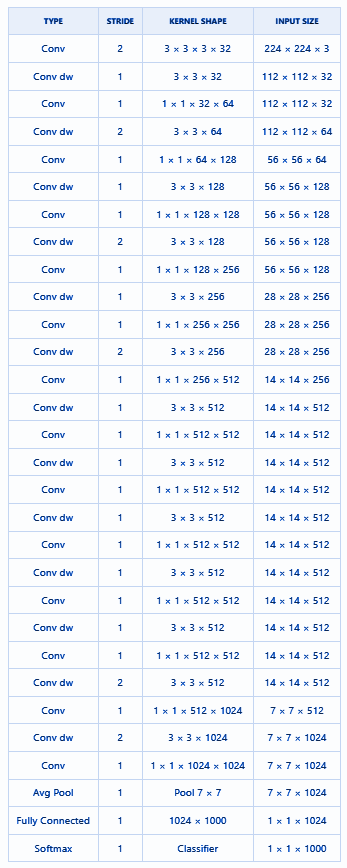
Για να επιλυθεί αυτό το θεμελιώδες πρόβλημα, μία από τις λύσεις που προτάθηκαν είναι τα δίκτυα προπορίας (residual networks), εξου και το όνομα του ResNet. Ένα τέτοιο μοντέλο κέρδισε τον ίδιο διαγωνισμό ταξινόμησης Imagenet το 2015.

Συνοπτικά, προστίθενται στην αρχιτεκτονικη του συνελικτικού δικτύου κάποια μπλοκ, τα οποία μεταφέρουν την είσοδο του τρέχοντος επιπέδου κατευθείαν ως είσοδο σε μετέπειτα επίπεδο, όπως φαίνεται στην εικόνα[]. Αυτό επιτρέπει στη πληροφορία να διατρέχει τα επίπεδα χωρίς να εμποδίζεται η δυνατότητα «μάθησης» του δικτύου.

Το ResNet 50 αποτελείται συνολικά από 50 επίπεδα ( συνελικτικά, FC, πυρήνες) και κάθε block προπορίας προσπερνά 3 επίπεδα. Πετυχαίνει απόδοση 3.8 δισεκατομμυρίων υπολογισμών ανά δευτερόλεπτο (FLOPS). Ο συνολικός αριθμός παραμέτρων του μοντέλου κυμαίνεται στα 26 εκατομμύρια.

Πρόκειται για προεκπαιδευμένο δίκτυο που έχει εκπαιδευτεί σε χιλιάδες εικόνες του συνόλου ImageNet και έχει αποκτήσει τη δυνατότητα να ταξινομεί έως και 1000 διαφορετικά αντικείμενα.   
Ως συνέπεια μπορεί να εξάγει χαρακτηριστικά από πλήθος εικόνων που σε συνδυασμό με θερμικούς χάρτες αποτυπώνουν τα καίρια σημεία του σώματος.

**MobileNet v1**



Το MobileNetV1 [https://arxiv.org/abs/1704.04861] ανήκει στη κατηγορία ειδικά σχεδιασμένων νευρωνικών δικτύων για κινητές συσκευές (MobileNets), εξου και το όνομά του. Η πεπερασμένη υπολογιστική ισχυ μικρών συσκευών (smartphones, tablets) θέτει περιορισμούς στην απόδοση των συνελικτικών δικτύων. Έτσι, το 2017 προτάθηκε η αρχιτεκτονική του MobileNet v1 από τον Andrew G. Howard της ερευνητικής ομάδας στη Google []. Αυτό το μοντέλο άνοιξε το δρόμο σε μια ολόκληρη οικογένεια μοντέλων ειδικευμένα στην Υπολογιστική Όραση. Είναι μικρά, χαμηλής καθυστέρησης και χαμηλής ισχύος παραμετροποιήσιμα μοντέλα που ανταποκρίνονται στους περιορισμούς της εκάστοτε συσκευής. Δίνουν τη δυνατότητα στο προγραμματιστή να ελέγχει μέσω δύο καθολικών υπερ-παραμέτρων το μέγεθος του μοντέλου ανάλογα με τις απαιτήσεις.

Ο συνολικός αριθμός παραμέτρων σε ένα τυπικό MobileNet v1 είναι 4.2 εκατομμύρια, ο οποίος είναι σημαντικά μικρότερος από άλλες αρχιτεκτονικές CNN, συμπεριλαμβανομένου και του ResNet.

Η αρχιτεκτονική, όπως φαίνεται στην εικόνα[] είναι παρόμοια με την αρχιτεκτονική συνελικτικών επιπέδων.

Η ειδοποιός διαφορά με τα κλασικά μοντέλα είναι τα βαθοκεντρικά συνελικτικά δίκτυα (deep-wise conv nets).

Πρόκειται για δίκτυα που εφαρμόζουν συνέλιξη σε ένα μόνο κανάλι από τα 3 μιας RGB εικόνας. Έτσι μειώνονται οι διαστάσεις εξόδου του και το υπολογιστικό κόστος του μοντέλου.

Συνολικά, το MobileNet V1 αποτελείται από 28 συνελικτικά επίπεδα και δύο υπερ-παραμέτρους.

* Η πρώτη παράμετρος είναι ο πολλαπλασιαστής πλάτους (που συμβολίζεται με α) που καθορίζει των αριθμό των καναλιών που θα φιλτράρονται από τα επιμέρους συνελικτικά επίπεδα. Οι τιμές της είναι μεταξύ 0 και 1. Η μείωση του υπολογιστικού κόστους είναι ανάλογη του τετραγώνου του α.
* Η δεύτερη παράμετρος είναι ο αναλυτικός πολλαπλασιαστής (που συμβολίζεται με ρ) που καθορίζει την ανάλυση της εικόνας εισόδου. Για δεδομένη τιμή ρ η ανάλυση της εικόνας εισόδου γίνεται 224\*ρ. Συνεπακόλουθα μειώνεται η είσοδος κάθε συνελικτικού κατα τον ίδιο παράγοντα. Η μείωση του υπολογιστικού κόστους είναι και εδώ ανάλογη του τετραγώνου του ρ.

Η σύγκριση ως προς τα σημαντικότερα σημεία των δύο μοντέλων φαίνεται εδώ:

|  | **ResNet50** | **MobileNetV1** |
| --- | --- | --- |
| μέγεθος (MB) | 22.7 | 4.3 |
| ταχύτητα απόκρισης | αργή | γρήγορη |
| ακρίβεια | καλύτερη | χειρότερη |
| παραμετροι (mill) | 25.6 | 4.24 |
| pixel εικόνας εισόδου | 224 x 224 | |

**MoveNet** [https://storage.googleapis.com/movenet/MoveNet.SinglePose%20Model%20Card.pdf]

Μοντέλο τελευταίας γενιάς που κυκλοφόρησε το 2021 από την ομάδα έρευνας της Google. Ανιχνεύει την ανθρώπινη στάση σε μεμονωμένα άτομα ενώ έκδοση ανίχνευσης στάσης και για περισσότερα άτομα είναι υπό υλοποίηση. Βασίζεται στη βιβλιοθήκη Tensorflow και εντοπίζει σε πραγματικό χρόνο 17 σημεία. Είναι μοντέλο υψηλής ακρίβειας και εξαιρετικά γρήγορο. Είναι προ-εκπαιδευμένο σε δύο σύνολα δεδομένων, σε 28.000 εικόνες του COCO και σε 23.500 εικόνες ενός εσωτερικού συνόλου δεδομένων της Google που ονομάζεται Active. To Active είναι εικόνες από μεμονωμένους ανθρώπους που κάνουν yoga, γυμναστική και χορό.

Δομή

Είσοδος: RGB εικόνα

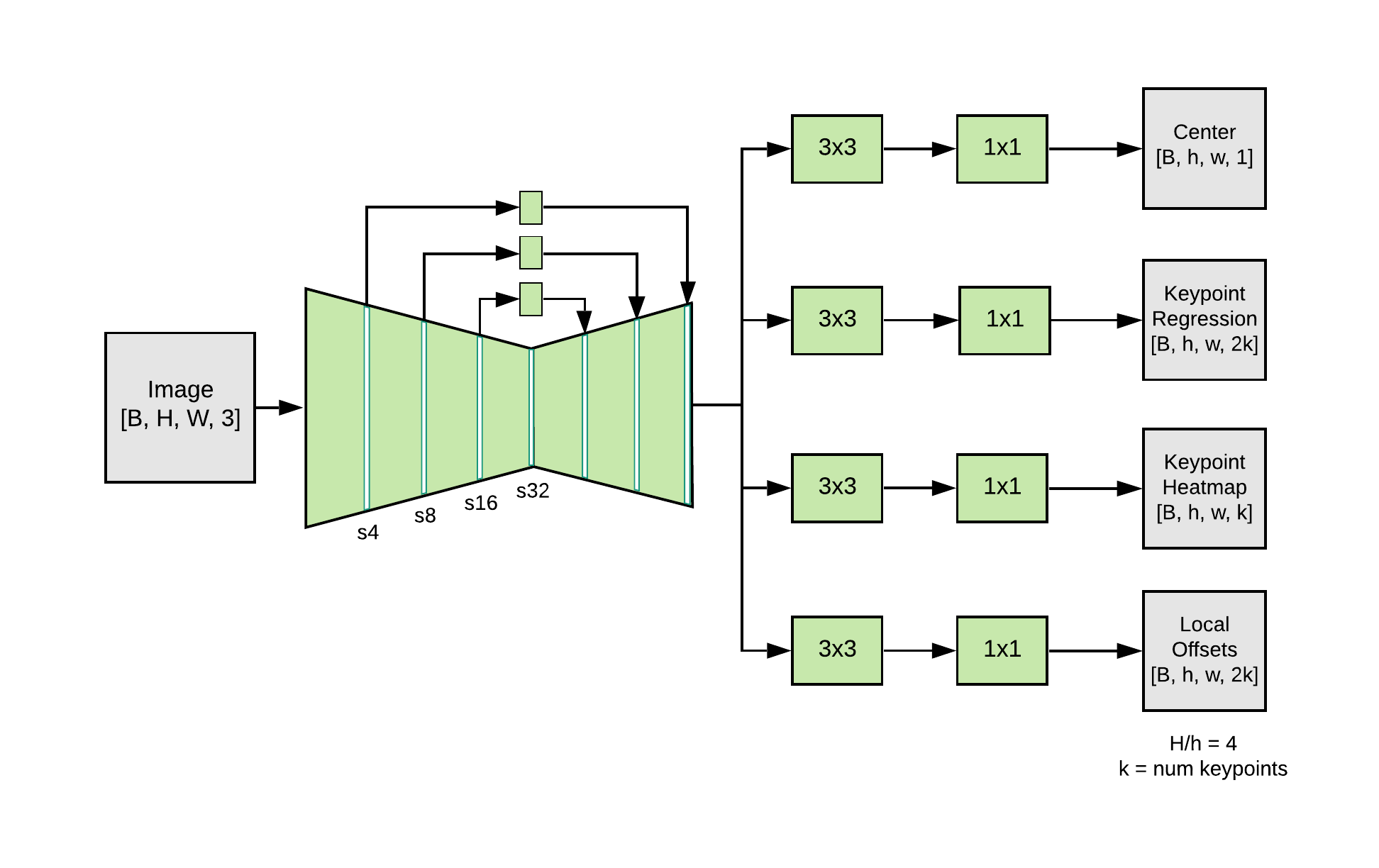
Έξοδος:

* 17 Keypoints με συντεταγμένες (x,y)
* Score βεβαιότητας για το εκάστοτε keypoint του ατόμου

Αρχιτεκτονική  
  
To MoveNet είναι μοντέλο με αρχιτεκτονική «από κάτω προς τα πάνω» (bottom-up) [https://openaccess.thecvf.com/content/CVPR2021/papers/Geng\_Bottom-Up\_Human\_Pose\_Estimation\_via\_Disentangled\_Keypoint\_Regression\_CVPR\_2021\_paper.pdf].  
Αποτελείται από δύο βασικά αφαιρετικά επίπεδα:

* δίκτυο εξαγωγής χαρακτηριστικών: εξάγονται σε υψηλή ανάλυση σημασιολογικά χαρακτηριστικά της εικόνας
* σύνολο από κεφαλές εκτίμησης: νευρωνικό που προβλέπει τις θέσεις των καίριων σημείων

Στη συνέχεια, θα εστιάσουμε στη δομή των αφαιρετικών επιπέδων που αποτελούν το μοντέλο.



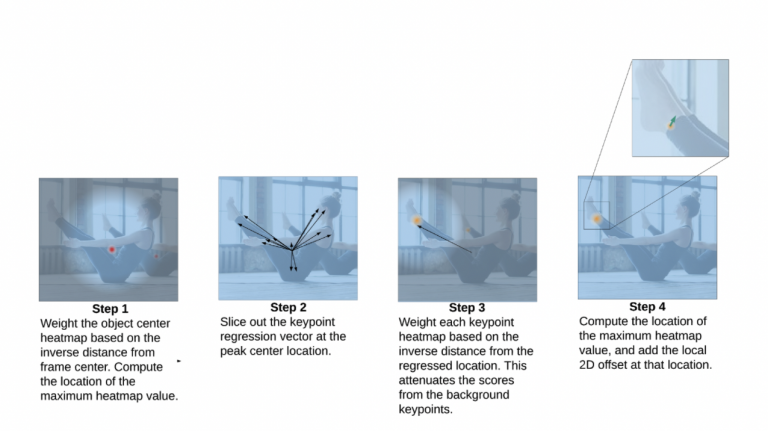
Το δίκτυο εξαγωγής χαρακτηριστικών (feature extractor) αποτελείται από τη βελτιωμένη έκδοση του MobileNet, το MobileNet v2 και από ένα πυραμιδοειδές δίκτυο χαρακτηριστικών (FPN). Στο συνελικτικό δίκτυο μειώνεται η ανάλυση των χαρακτηριστικών της εικόνας όσο προστίθενται επίπεδα. Ο ρόλος του FPN είναι να εξομαλύνει αυτή τη μείωση, ώστε να παραμένει υψηλή η ανάλυση της εικόνας εξόδου.

Στη συνέχεια η πληροφορία εισέρχεται σε κεφαλές πρόβλεψης σημείων (prediction heads), δηλαδή νευρωνικά που προβλέπουν σε ποιες περιοχές της αρχικής εικόνας υπάρχουν καίρια σημεία της στάσης του σώματος με την αντίστοιχη πιθανότητα.

Συγκεκριμένα, τα νευρωνικά αυτά βασίζονται στο μοντέλο CenterNet και πρόκειται για τέσσερα δίκτυα που «τρέχουν» παράλληλα. Είναι τα παρακάτω:

* Θερμικός χάρτης ανίχνευσης ατόμου (person center heatmap):   
  προβλέπει το γεωμετρικό κέντρο βάρους του ατόμου.
* Πεδίο παλινδρόμησης καίριων σημείων (keypoint regression field): προβλέπει ένα σύνολο από καίρια σημεία συναρτήσει του γεωμετρικού κέντρου. Σκοπός να αντιστοιχηθούν τα καίρια σημεία σε ένα συγκεκριμένο άτομο.
* Θερμικός χάρτης καίριων σημείων: προβλέπει τα καίρια σημεία που αποτυπώνουν τη στάση. Κάθε pixel της εικόνας πολλαπλασιάζεται με ένα βάρος(αντιστρόφως ανάλογο της απόστασης από το εκάστοτε καίριο σημείο) ώστε να αποφευχθεί σύγχυση με άλλα αντικείμενα της εικόνας.
* πεδίο offset διανυσμάτων: προβλέπει τη θέση των offset διανυσμάτων που προσδίδουν πιθανοτική ακρίβεια στα καίρια σημεία

Σε όλα τα παραπάνω στάδια γίνεται διαλογή των σημείων με βάση την υποψήφια.



Ενώ καταβλήθηκε ιδιαίτερη προσπάθεια από τους ερευνητές στον αποδοτικό σχεδιασμό στην αρχιτεκτονική του δικτύου, στην επεξεργασίας των εικόνων και στην επιλογή των δεδομένων εκπαίδευσης, ακόμη μεγαλύτερη έμφαση δόθηκε στη ταχύτητα απόκρισης του μοντέλου. Αυτό επιτεύχθηκε με αφαίρεση συνελικτικών επιπέδων από τις κεφαλές και χρήση πολλών επιπλέον βαθοκεντρικών συνελίξεων σε αυτές. Ο κορμός του μοντέλου, το MobileNetV2 δεν υπέστη καμία αλλαγή ώστε το MoveNet να πετυχαίνει υψηλής ποιότητας ανίχνευση.

Το μοντέλο Posenet διαθέτει δύο παραλλαγές:   
- Thunder

- Lightning

Στο πίνακα φαίνονται οι κύριες διαφορές μεταξύ τους:

|  | **Thunder** | **Lightning** |
| --- | --- | --- |
| μέγεθος (MB) | 12.6 25.0 | 4.8 9.4 |
| ταχύτητα απόκρισης (fps) | >50 | >30 |
| ακρίβεια | υψηλή | χειρότερη |
| παραμετροι (mill) | 6.23 | 2.32 |
| pixels εικόνας εισόδου | 256 x 256 | 192 x 192 |
| πολλαπλασιαστής βάθους | 1.75 | 1.5 |

# Κεφάλαιο 5

Διαχείριση έργου & ευέλικτη τεχνολογία λογισμικού

5.1 Το Λογισμικό ως υπηρεσία

5.1.1. Τι είναι το λογισμικό ως υπηρεσία

Ο τρόπος διάθεσης ενός προϊόντος λογισμικού στην αγορά είναι πολλές φορές πιο σημαντικός και από το ίδιο το λογισμικό. Καθώς η παροχή λογισμικού αποτελεί μία από τις πιο ταχέως αναπτυσσόμενες βιομηχανίες τις τελευταίες δεκαετίες, νέες μέθοδοι κατασκευής και διάθεσης λογισμικού έχουν προταθεί ανά καιρούς. Η ανάπτυξη βέβαια αυτή είναι φυσιολογική και ακολουθεί πιστά την παράλληλη εξέλιξη σε τομείς του hardware και των δικτύων.

Έτσι τον πρώτο καιρό που εμφανίστηκαν οι υπολογιστές τα προϊόντα λογισμικού εγκαθιστόντουσαν στα τερματικά των χρηστών και η όποια ανανέωση του λογισμικού μπορούσε να γίνει μόνο χειροκίνητα με ευθύνη του χρήστη. Ως εκ τούτου προβλήματα συμβατότητας ήταν ένα σύνηθες φαινόμενο, πράγμα το οποίο σήμαινε ότι η εταιρία του προΪόντος χρειαζόταν να συντηρεί διάφορες εκδόσεις του. Από την άλλη ο χρήστης ήταν ταυτόχρονα κάτοχος του προϊόντος, δηλαδή λάμβανε μία άδεια χρήσης του λογισμικού.

Ο κύκλος ζωής λοιπόν του λογισμικού ακολουθούσε τη λογική του καταρράκτη (waterfall), στην οποία σχεδιασμός, υλοποίηση, διάθεση και συντήρηση του λογισμικού αποτελούν διαδοχικά, διακριτά και μη επαναλαμβανόμενα στάδια.

Η προσέγγιση αυτή, αν και χρησιμοποιείται περιορισμένα μέχρι και σήμερα, θεωρείται από πολλούς απαρχαιωμένη. Ολό και πιο συχνά πλέον, τα προϊόντα λογισμικού παραδίδονται ως υπηρεσία (SaaS - Software as a service). Στην αρχιτεκτονική SaaS το λογισμικό διατίθεται και διανέμεται μέσω του Διαδικτύου στους προσωπικούς υπολογιστές και στις φορητές συσκευές.   
Τα οφέλη τόσο για τους χρήστες όσο και για τους προγραμματιστές είναι πολλά.

Οι πελάτες αφενός δε χρειάζεται να το εγκαθιστούν και μπορούν να έχουν πρόσβαση από παντού, και από οποιαδήποτε συσκευή. Συνήθως η υπηρεσία διατίθεται σε κλασικούς υπολογιστές μέσω κάποιας διαδικτυακής εφαρμογής (Web App) και σε φορητές συσκευές μέσω κάποιας αποκλειστικής εφαρμογής. Το μοντέλο που συνήθως ακολουθείται στην περίπτωση του SaaS είναι συνδρομητικό, δηλαδή οι χρήστες δεν αγοράζουν μία μόνιμη άδεια χρήσης του λογισμικού, αλλά πληρώνουν μία συνδρομή για να χρησιμοποιούν το λογισμικό για όσο χρόνο επιθυμούν. Σε πολλές περιπτώσεις μάλιστα η υπηρεσία διατίθεται δωρεάν στο τελικό χρήστη, με τον πάροχο να επιλέγει να αποκομίσει με διαφορετικό τρόπο τα έσοδά του, συνήθως μέσω διαφημίσεων, ή προαιρετικών επί πληρωμή παροχών.

Από την άλλη το μοντέλο SaaS προσφέρει πολλά πλεονεκτήματα και στους προγραμματιστές. Επειδή πλέον το λογισμικό διανέμεται μέσω διαδικτύου, και στην περίπτωση των διαδικτυακών εφαρμογών δεν εγκαθίσταται στις συσκευές των χρηστών, τότε δεν υφίσταται η ανάγκη ύπαρξης πολλαπλών εκδόσεων της ίδιας εφαρμογής. Άρα οι προγραμματιστές μπορούν να εργάζονται απρόσκοπτα στη βελτίωση της τρέχουσας έκδοσης, χωρίς να προβληματίζονται για τυχόν ασυμβατότητες. Αυτό φυσικά οδηγεί στην επιτάχυνση της διαδικασίας συγγραφής κώδικα σε σύγκριση με το μοντέλο του καταρράκτη. Επιπλέον, λόγω της διαδικτυακής φύσης των εφαρμογών η ανάγκη για χρονοβόρες αναβαθμίσεις που συνήθως απασχολούν τους χρήστες δεν υφίσταται.

Όλα τα παραπάνω εκτόξευσαν τους ρυθμούς εργασίας στη βιομηχανία λογισμικού, δημιουργώντας παράλληλα έντονο ανταγωνισμό μεταξύ των εταιρειών για την επικράτηση στην αγορά. Εδώ λοιπόν αξίζει να αναφέρουμε και έναν από τα βασικότερα μειονεκτήματα, ή ακριβέστερα κίνδυνο, της SaaS αρχιτεκτονικής: ότι όσο σύντομα μπορεί να αναπτυχθεί μία εφαρμογή, τόσο σύντομα μπορεί να θεωρηθεί ξεπερασμένη, αν δεν εξελίσσεται διαρκώς. Γι’ αυτό μάλιστα λένε χαρακτηριστικά ότι «καινοτομείς ή πεθαίνεις» (innovate or die). Στο σημείο αυτό θα μπορούσαμε να παραθέσουμε μία λίστα από άλλοτε δημοφιλείς υπηρεσίες που εξαλείφθηκαν ακριβώς γιατί δεν κατάφεραν να ακολουθήσουν το ρυθμό εξέλιξης.

Θα παραθέσουμε ωστόσο κάποια σύγχρονα παραδείγματα λογισμικού, προκαλώντας το μελλοντικό αναγνώστη να αναλογιστεί ποια από αυτά επιβίωσαν στο πέρασμα του χρόνου:

* Netflix, Disney+, HBO Max
* Google Workspace
* Office 365
* Slack
* Gmail, yahoo mail, protonmail
* Spotify, soundcloud
* Facebook, Twitter, Youtube, Instagram, TikTok
* Fortnite, Pokemon Go
* Τραπεζικές εφαρμογές e-banking κτλ.

Είναι αξιοσημείωτο ότι οι περισσότερες δημοφιλείς εφαρμογές με τις οποίες αλληλεπιδρά ο μέσος χρήστης ανήκουν στην κατηγορία των SaaS, και μάλιστα πολλές από αυτές διατίθενται δωρεάν.

Στη δική μας περίπτωση κατά την έναρξή της διπλωματικής εργασίας κληθήκαμε να αποφασίσουμε ποια αρχιτεκτονική πρέπει να ακολουθήσουμε. Για όλους τους προαναφερθέντες λόγους επιλέξαμε να υλοποιήσουμε την εφαρμογή PhysAI ως SaaS. Εξάλλου η αναμενόμενη χρήση της εφαρμογής, όπως θα αναλύσουμε και στη συνέχεια, αφορά ασθενείς που εκτελούν φυσικοθεραπείες, και θα χρησιμοποιούν ως επί το πλείστον το κινητό τους τηλέφωνο, στο οποίο όλες σχεδόν οι ανταγωνιστικές εφαρμογές αναπτύσσονται επίσης ως SaaS.

[βιβλίο fox - patterson]

[βιβλίο Sommerville]

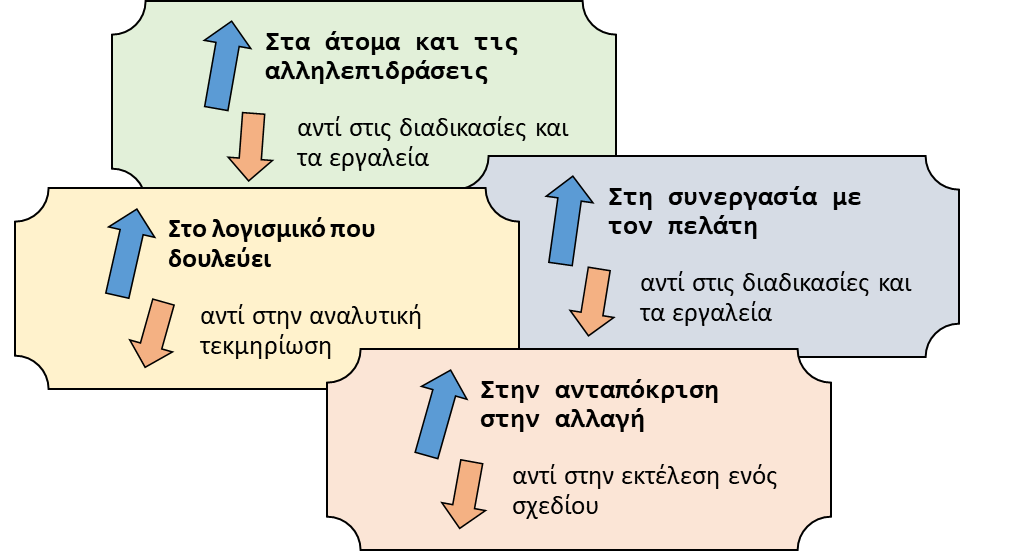
5.1.2 Υπολογιστική νέφους

Όπως ήδη ειπώθηκε το SaaS λογισμικό διανέμεται σχεδόν αποκλειστικά μέσω Διαδικτύου, επομένως η ανάγκη για ύπαρξη αξιόπιστων διακομιστών που θα παρέχουν τις υπηρεσίες ακατάπαυστα ήταν πρόδηλη από πολύ νωρίς. Στα πρώιμα στάδια κάθε μεγάλη εταιρεία λογισμικού βασίστηκε σε ιδιόχρηστους διακομιστές. Γρήγορα ωστόσο η ανάγκη για εύκολη κλιμακωσιμότητα των υπηρεσιών οδήγησε στη δημιουργία μεγάλων αποθηκών για βάσεις δεδομένων (data centers). Οι αποθήκες αυτές μπορούν να περιλαμβάνουν εκατοντάδες χιλιάδες διακομιστές. Η επακόλουθη οικονομία κλίμακας που πετυχαίνουν οι αποθήκες επιτρέπει τη διάθεση της υπολογιστικής νέφους ως υπηρεσία (Infrastructure as a Service IaaS) ανάλογα με την αντίστοιχη ζήτηση (on demand). Αυτές οι υπηρεσίες καλούνται συνήθως με τον όρο νέφος (cloud). Πολύ γνωστές σύγχρονες υπηρεσίες νέφους είναι το Amazon Web Services, το Microsoft Azure, το Google App Engine.

Στη δική μας περίπτωση λόγω της μικρής κλίμακας της εφαρμογής και καθ’ότι δεν υπάρχει η πρόθεση για εμπορική εκμετάλλευση αλλά αποτελεί αντικείμενο έρευνας για την απόδειξη της ορθότητας της ιδέας (proof of concept), δεν απαιτήθηκε η μίσθωση ενός εμπορικού διακομιστή, αλλά προτιμήθηκαν ιδιωτικοί. Για την εξαγωγή της εφαρμογής στο Διαδίκτυο προτιμήθηκε η δωρεάν (για προσωπική χρήση) εφαρμογή ngrok.

5.2 Ευέλικτη τεχνολογία λογισμικού

5.2.1 Εισαγωγή στην agile μεθοδολογία



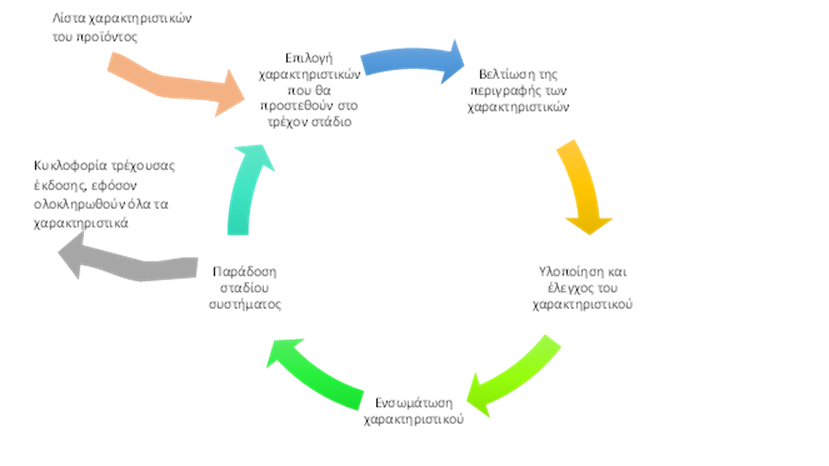
Μέχρι να εδραιωθεί πλήρως το Διαδίκτυο η πιο διαδεδομένη μέθοδος ανάπτυξης λογισμικού ήταν η προσχεδιασμένη, δηλαδή ο εξ αρχής λεπτομερής σχεδιασμός των απαιτήσεων, η ελεγχόμενη σταδιακή ανάπτυξη του λογισμικού με χρήση σχολαστικών διαδικασιών τεκμηρίωσης και ο εξαντλητικός έλεγχος της εφαρμογής προτού κυκλοφορήσει στην αγορά. Η αντίληψη αυτή προέρχοταν κυρίως από την κοινότητα της τεχνολογίας λογισμικού η οποία ασχολούνταν με μεγάλα, μακροχρόνια έργα συστημάτων λογισμικού, που κυρίως απασχολούσαν τους κρατικούς φορείς. Σε έργα όμως που απασχολούν λίγους προγραμματιστές, έχουν περιορισμένους πόρους, αλλά και μικρότερο μέγεθος η εξαντλητική τεκμηρίωση και ο λεπτομερής σχεδιασμός προκαλούν σημαντική επιβάρυνση στην παραγωγική διαδικασία. Αναλώνεται έτσι πολύς χρόνος στην σύνταξη εγγράφων, που πολύ πιθανόν να αποδειχθούν άχρηστα ενώ οι απαιτήσεις και οι συνθήκες με τις οποίες ξεκίνησε ο σχεδιασμός της εφαρμογής μπορεί να έχουν τροποποιηθεί όταν αυτή ολοκληρώθηκε.

Όλα τα παραπάνω αποτελούσαν πρόβλημα για τους προγραμματιστές, οι οποίοι άρχισαν σιγά-σιγά να υιοθετούν πιο ευέλικτες (agile) προσεγγίσεις. Χρονιά σταθμός στην πορεία αυτή αποτελεί το 2001, κατά την οποία δημοσιεύτηκε το Ευέλικτο Μανιφέστο ([Agile Manifesto](https://agilemanifesto.org/)), το οποίο τονίζει ότι δίνει περισσότερη σημασία:

* **Στα άτομα και τις αλληλεπιδράσεις** αντί στις διαδικασίες και τα εργαλεία
* **Στο λογισμικό που δουλεύει** αντί στην αναλυτική τεκμηρίωση
* **Στη συνεργασία με τον πελάτη** αντί στην διαπραγμάτευση του συμβολαίου
* **Στην ανταπόκριση στην αλλαγή** αντί στην εκτέλεση ενός σχεδίου

Αυτό το εναλλακτικό σχέδιο ανάπτυξης στηρίζει την επιτυχία του στην αποδοχή της αλλαγής ως αναπόφευκτο γεγονός, δηλαδή οι προγραμματιστές θα πρέπει να είναι σε θέση να τροποποιήσουν κατά τη διάρκεια της παραγωγής ένα ημιτελές αλλά λειτουργικό πρωτότυπο προκειμένου να μπορούν να ικανοποιήσουν τις μεταβαλλόμενες ανάγκες και προτιμήσεις του πελάτη. Αυτό σημαίνει ότι θα πρέπει συνεχώς να μπορεί να δέχεται ανατροφοδότηση.

Η ευέλικτη προσέγγιση δίνει επιπλέον έμφαση στην ανάπτυξη μέσω δοκιμών   
(test-driven development TDD), με τον κώδικα ελέγχου να γράφεται πριν τη συγγραφή του αντίστοιχου κώδικα. Επιπροσθέτως δίνει ιδιαίτερη σημασία στις ιστορίες χρηστών, δηλαδή στην περιγραφή των απαιτήσεων με τη μορφή παραδειγμάτων χρήσης (use case). Τέλος η ευέλικτη προσέγγιση εστιάζει αρκετά στην ταχύτητα, προσπαθώντας όσο το δυνατόν συντομότερα να παραδίδει μέρος των λειτουργιών της τελικής εφαρμογής. Η παραγωγή δηλαδή χωρίζεται σε μικρά στάδια βαθμιαίας ανάπτυξης, όπου σε κάθε στάδιο ολοκληρώνεται από ένα χαρακτηριστικό όπως φαίνεται στο παρακάτω σχήμα.



Εδώ βέβαια αξίζει να τονίσουμε ότι δεν υπάρχει μόνο ένας Ευέλικτος κύκλος ζωής. Το Ευέλικτο Μανιφέστο θέτει τις αρχές πάνω στις οποίες έχουν δομηθεί διάφορες ευέλικτες μεθοδολογίες. Από αυτές οι πιο γνωστές είναι οι εξής:

* Ακραίος προγραμματισμός (Extreme Programming XP)
* Τεχνική συνάθροισης (Scrum)
* Προσέγγιση Kanban
* Υβριδική προσέγγιση ScrumBan

Κατά την εκπόνηση της διπλωματικής μας υιοθετήσαμε την τελευταία μεθοδολογία τόσο για την ανάπτυξη της εφαρμογής, όσο και για τη συγγραφή του τελικού παραδοτέου. Οι λόγοι που επιλέξαμε τη συγκεριμένη μέθοδο είναι ότι, όπως θα δούμε παρακάτω, συνδυάζει τα πλεονεκτήματα δύο άλλων μεθόδων και διαθέταμε αρκετή εξοικείωση από άλλα έργα που είχαμε αναλάβει στα πλαίσια της Σχολής.

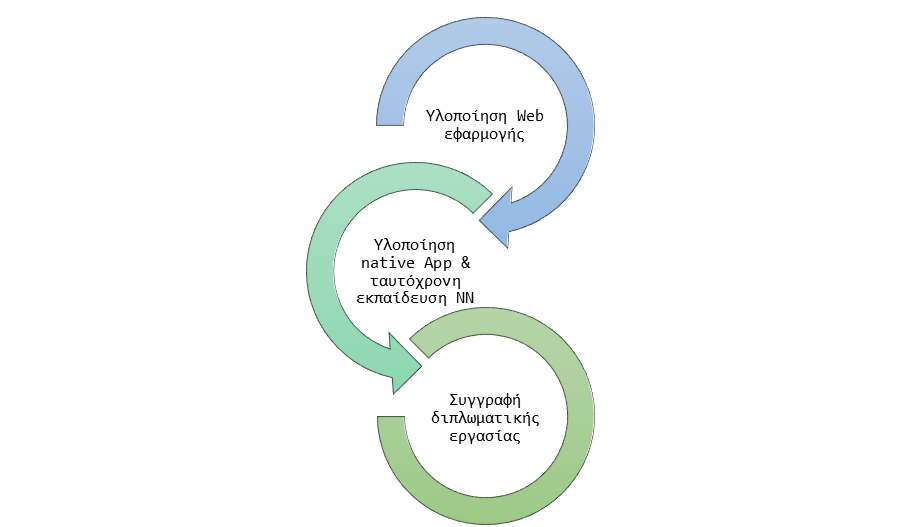
5.2.2 ScrumBan

Η ευέλικτη τεχνική ScrumBan αποτελεί υβριδική προσέγγιση των τεχνικών Scrum και Kanban. Καταρχάς στην τεχνική της συνάθροισης, Scrum, ο βασικός πυλώνας της παραγωγικής διαδικασίας είναι οι αυτοοργανωμένες ομάδες που λαμβάνουν μέρος σε κούρσες (sprints), μικρής χρονικής διάρκειας συνήθως λίγων εβδομάδων. Η θεμελιώδης συνεπώς ιδέα στην οποία βασίζεται η διαδικασία Scrum είναι ότι το λογισμικό θα πρέπει να αναπτύσσεται σταδιακά σε μία σειρά από επαναλαμβανόμενες κούρσες, όπου στο τέλος καθεμίας οι διάφορες ομάδες συγχωνεύονται και σχεδιάσουν την επόμενη κούρσα. Καθοριστικής σημασίας στην επιτυχία της μεθοδολογίας αυτής είναι οι καθημερινές βραχύχρονες συναντήσεις των μελών της ομάδας προκειμένου να συζητήσουν οποιαδήποτε δυσκολίες υπάρχουν, ή τους στόχους της ημέρας. Για την καταγραφή κούρσας χρησιμοποιείται η λίστα εκκρεμοτήτων κούρσας (sprint backlog).

Από την άλλη η τεχνική του Kanban αποτελεί προσαρμογή της λογικής JIT (just-in-time) που ανέπτυξε η Toyota στην αυτοκινητοβιομηχανία της. Έτσι αντιμετωπίζει την ανάπτυξη λογισμικού ως μία γραμμή παραγωγής (pipeline). Εν αντιθέση με άλλες τεχνικές, όπως η scrum, προβλέπει διακριτούς ρόλους στα μέλη της ομάδας που εργάζονται σε ένα έργο, με σκοπό να εξισορροπούν τον αριθμό των μελών ομάδας ώστε να μη δημιουργούνται στενωποί (bottlenecks) που καθυστερούν τη γραμμή παραγωγής. Τα διάφορα τμήματα της γραμμής παραγωγής καταφέρνουν να συντονίζονται χάρις σε έναν πίνακα καρτών (cardboard) που απεικονίζει την κατάσταση όλων των εκκρεμοτήτων στη γραμμή παραγωγής.

Ο υβριδικός κύκλος ScrumBan χρησιμοποιεί τις καθημερινές συναντήσεις και τις κούρσες του Scrum, αλλά αντικαθιστά τη φάση σχεδιασμού, με τον πιο δυναμικό έλεγχο της γραμμής παραγωγής μέσω του cardboard του Kanban.

Στη δική μας περίπτωση, που φυσικά είχαμε διμελή ομάδα, ακολουθώντας το υβριδικό μοντέλο ScrumBan στην αρχή κάθε ημέρας πραγματοποιούσαμε μία τηλεφωνική συνάντηση προκειμένου να καθορίσουμε τους στόχους της τρέχουσας μέρας, ενώ μέσω μίας τηλεδιάσκεψης στο τέλος της ημέρας αντιμετωπίζαμε τυχόν προβλήματα που συναντήσαμε κατά τη διάρκεια αυτής. Το έργο χωρίστηκε σε 3 επιμέρους κούρσες, καθεμία από τις οποίες είχε διάρκεια περίπου 1 μήνα. Πριν την έναρξη της πρώτης κούρσας έγινε ο προσδιορισμός των απαιτήσεων που θα αναλυθούν στη συνέχεια, και ο σχεδιασμός του τρόπου δράσης. Στην πρώτη κούρσα υλοποιήσαμε την πρώτη έκδοση της διαδικτυακής εφαρμογής, στη δεύτερη ασχοληθήκαμε με την εκπαίδευση του νευρωνικού μας δικτύου και τη δημιουργία της εφαρμογής για smartphones, και στην τρίτη ασχοληθήκαμε με τη συγγραφή της διπλωματικής. Καθόλη τη διάρκεια εκπόνησης της εργασίας οι ρόλοι ήταν διακριτοί, και ο τομέας στον οποίο εργαζόταν ο καθένας ήταν ξεχωριστός, αλλά παράλληλα γινόταν αμοιβαίος έλεγχος. Τέλος για το συντονισμό των ενεργειών και τον έλεγχο της προόδου δημιουργήθηκε ένα κοινόχρηστο cardboard στην πλατφόρμα Github.



5.2.3 Εργαλεία διαχείρισης έργου

Στη σύγχρονη βιομηχανία λογισμικού η διαχείριση ενός έργου, και συντονισμός μίας ομάδας προγραμματιστών, όσο μικρής κλίμακας και αν είναι, θα ήταν μάλλον εντελώς αδύνατη χωρίς κάποιο εργαλείο διαχείρισης έργου (project management tool). Ακόμα και αν πρόκειται για ατομική εργασία, οι δυνατότητες για οργάνωση που παρέχουν αυτά τα εργαλεία μπορούν να επιταχύνουν πολύ την παραγωγική διαδικασία, παρά της φαινομενικής πολυπλοκότητας που εισάγουν.

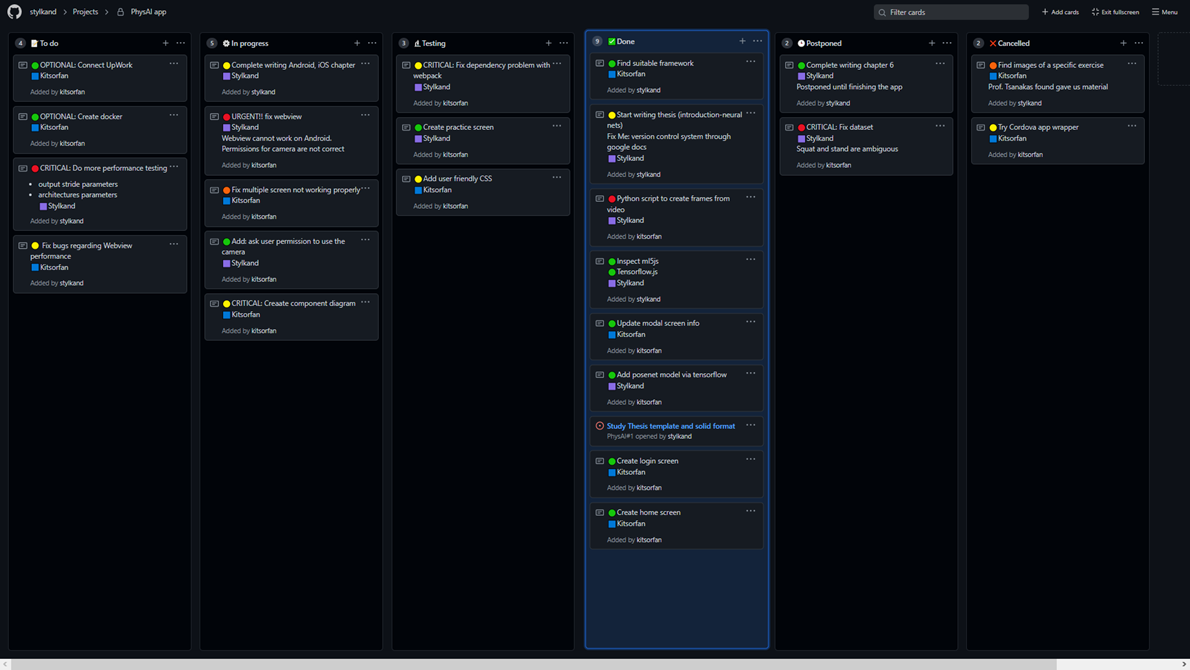
Κατά την υλοποίηση κάποιας εφαρμογής απαιτείται αρκετός σχεδιασμός, ανάθεση και καταμερισμός εργασιών, εποπτεία πόρων, προσδιορισμός των χρονικών ορίων, και έλεγχος ότι όλοι τα τηρούν, καταγραφή του χρόνου που αναλώνεται σε κάποια διαδικασία κτλ. Όλες αυτές οι διαδικασίες χρειάζεται πολλές φορές να ποσοτικοποιηθούν, να διαχωριστούν μεταξύ τους και να ιεραρχηθούν κατάλληλα. Παράλληλα πρέπει να ανατεθούν με τέτοιο τρόπο στα μέλη της ομάδας, ώστε η παραγωγική αλυσίδα να λειτουργεί όσο το δυνατόν απροβλημάτιστα. Σε αυτή την κατεύθυνση η χρήση κάποιο ειδικού ψηφιακού εργαλείου μπορεί να υπάρξει ευεργετική, στην οργάνωση των ενεργειών, στην ενημέρωση όλων των εμπλεκομένων και γενικότερα στην εξοικονόμηση χρόνου.

Παραδείγματα τέτοιων σύγχρονων εργαλείων είναι τα εξής:

* monday.com
* JIRA Service Management
* YouTrack
* teamdeck
* Bitrix24
* airfocus
* Easy Projects
* Zoho Sprints
* Github projects
* Freshservice
* Harverst
* κ.α.

[<https://www.capterra.com/sem-compare/it-project-management-software/>]

Από τα παραπάνω για τη διαχείριση του δικού μας έργου επιλέξαμε το δωρεάν εργαλείο Github projects, το οποίο προστέθηκε πρόσφατα στο οικοσύστημα του Github, και παρέχει τη δυνατότητα δημιουργίας ενός Kanban cardboard με παράλληλη διασύνδεση στον διαμοιραζόμενο κώδικα. Στη δική μας περίπτωση, όπως αναλύθηκε και προηγουμένως, χωρίσαμε την υλοποίηση σε τρία διαδοχικά sprints, για καθένα από τα οποία χρειάστηκε ένα cardboard. Ενδεικτικά παραθέτουμε στιγμιότυπο από το δεύτερο sprint.

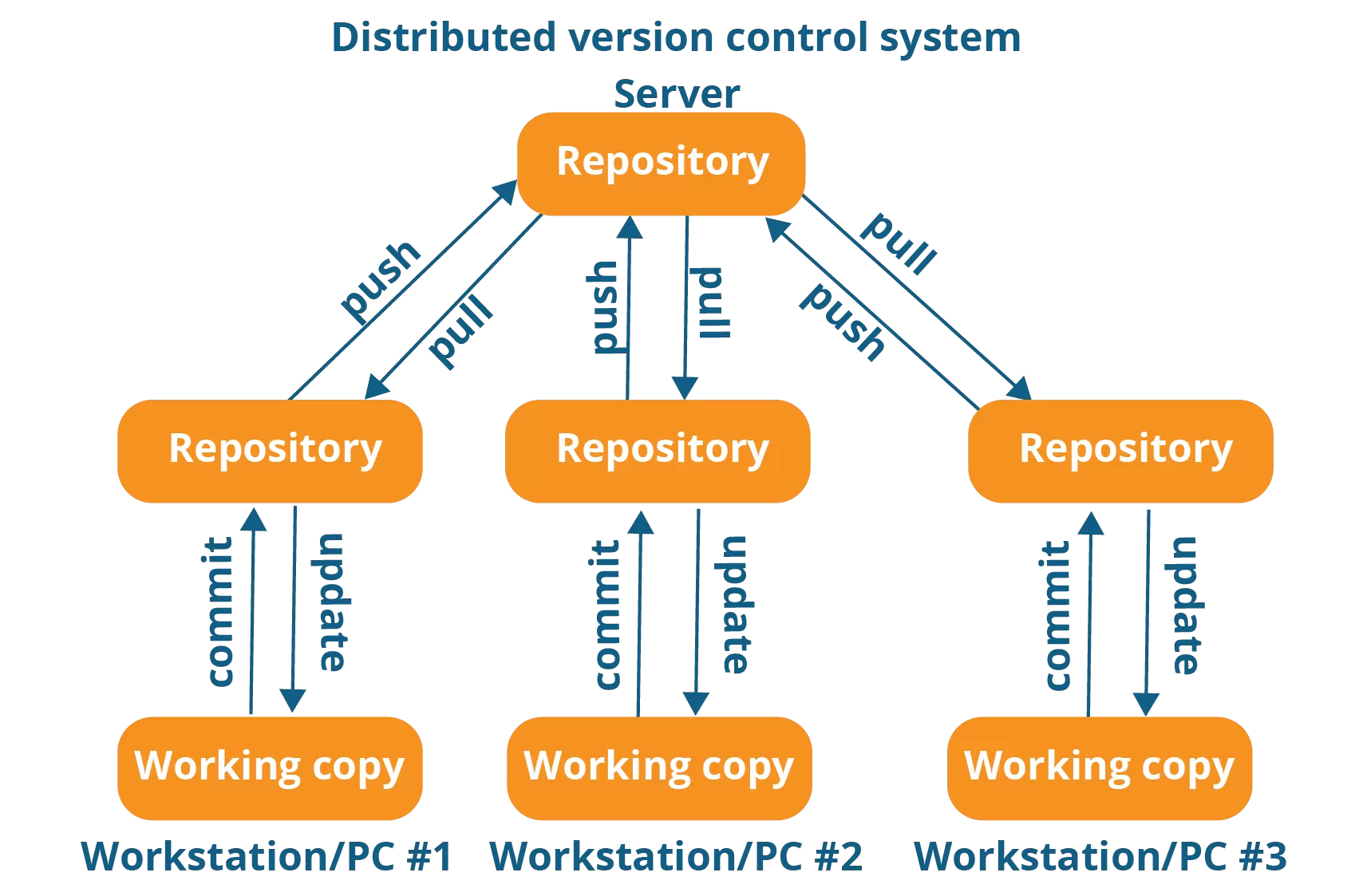


5.3 Συστήματα διαχείρισης κώδικα

Σε κάθε εταιρεία που ασχολείται με την ανάπτυξη εφαρμογών η ανάγκη για τον διαμοιρασμό του πηγαίου κώδικα μεταξύ των προγραμματιστών είναι προφανής. Αυτή η ανάγκη οδήγησε στη δημιουργία συστημάτων για τη διαχείριση και την κοινή χρήση κώδικα. Τα πρώτα τέτοια συστήματα βασίζονταν στη συγκεντρωτική αρχιτεκτονική αποθετηρίου, στην οποία οι προγραμματιστές είναι υποχρεωμένοι να παραδίδουν και να παραλαμβάνουν τα αρχεία (CICO - checking in/checking out). Έτσι αν ένας χρήστης χρήστης παραλάβει ένα αρχείο λαμβάνει και το δικαίωμα της αποκλειστικής του χρήσης μέχρι να το παραδώσει, έτσι ώστε οποιοσδήποτε άλλος προσπαθήσει να επεξεργαστεί το ίδιο αρχείο να μη μπορεί να το κάνει ταυτόχρονα. Αυτή η αποκλειστικότητα στη διαχείριση αρχείων ωστόσο δημιουργεί καθυστερήσεις στην υλοποίηση του λογισμικού, ειδικά όταν πρόκειται για μεγάλα αρχεία, στα οποία διαφορετικοί προγραμματιστές θα μπορούσαν να εργάζονται παράλληλα σε διαφορετικά τμήματά τους. Το κεντροποιημένο λοιπόν σύστημα διαχείρισης αποτελούσε πρόβλημα, και σύντομα αντικαταστάθηκε, στην πλειονότητα των οργανισμών, από κατανεμημένα συστήματα.

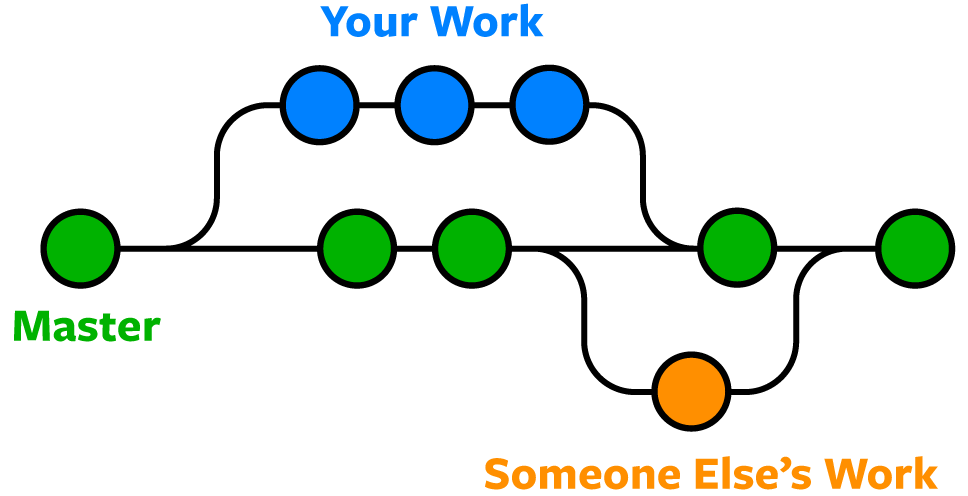
5.3.1 Κατανεμημένο σύστημα διαχείρισης εκδόσεων - Git

Το πρώτο κατανεμημένο σύστημα διαχείρισης εκδόσεων (DVCS - distributed version control system), το Git, δημιουργήθηκε από τον Linus Torvalds με τον σκοπό να βοηθήσει στη διαχείριση και στην υποστήριξη της ανάπτυξης του ανοικτού κώδικα του λειτουργικού συστήματος Linux που επίσης δημιούργησε. Στο Git αντί ο κάθε χρήστης να διατηρεί μόνο τα αντίγραφα του κώδικα τα οποία επεξεργάζεται, έχει στην κατοχή του έναν κλώνο του αποθετηρίου (repository).

[](https://shiftasia.com/column/introduction-to-git/)

Κάθε φορά η τρέχουσα βασική έκδοση του λογισμικού, στην οποία εργάζεται η ομάδα των προγραμματιστών, αποθηκεύεται στον κύριο κλάδο (main branch) του Git αποθετηρίου. Κάθε χρήστης που θέλει να επεξεργαστεί τον κώδικα του λογισμικού λαμβάνει έναν κλώνο του αποθετηρίου, λαμβάνοντας παράλληλα ένα αντίγραφο του κύριου κλάδου τον οποίο μπορούν τοπικά να επεξεργαστούν ανεξάρτητα από κάθε άλλον προγραμματιστή. Κάθε φορά που θέλει να προσθέσει νέες αλλαγές στον αντίστοιχο απομακρυσμένο κλάδο στον οποίο εργάζεται κάνει ένα αίτημα προώθησης (push request), ενώ όταν θέλει ή πρέπει να λάβει κάποια ενημέρωση δέχεται μία αίτηση ώθησης (pull request).

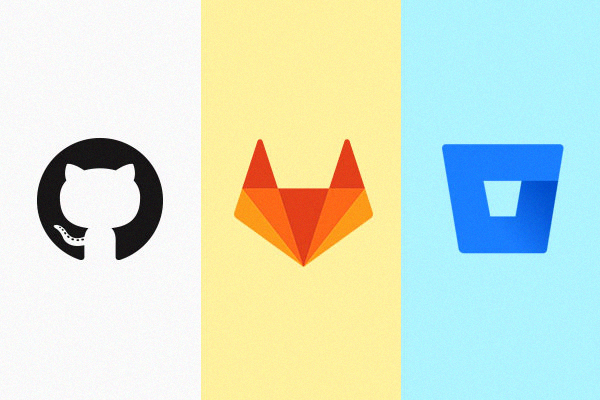
Παράλληλα κάθε χρήστης μπορεί να δημιουργήσει ένα νέο κλάδο αν επιθυμεί να πειραματιστεί σε μία νέα έκδοση. Όταν κάποιος χρήστης κρίνει ότι οι αλλαγές που έχει κάνει μπορούν να ενσωματωθούν στη βασική έκδοση του κύριου κλάδου, δημιουργεί ένα αίτημα συγχώνευσης (merge request) και ο διαχειριστής του έργου μπορεί να το εγκρίνει ή να το απορρίψει.

[](https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it)

Τα πλεονεκτήματα του git σε σχέση με τα κεντροποιημένα συστήματα διαχείρισης είναι ότι παρέχει αυξημένη ανθεκτικότητα σε απώλειες, αφού καθένας εμπλεκόμενος έχει ένα κλώνο του πηγαίου κώδικα, ταχύτητα, αφού κάθε αλλαγή στο αποθετήριο μπορεί να γίνεται τοπικά και οι υπόλοιποι χρήστες να ενημερώνονται μόνο για τις αλλαγές, χωρίς να χρειάζεται να αντιγράφουν όλο το αρχείο. Τέλος προσφέρεται ευελιξία, αφού κάθε προγραμματιστής μπορεί να δουλεύει σε όποιο κομμάτι του έργου επιθυμεί χωρίς το φόβο να επηρεάσει τη δουλειά κάποιου άλλου.

Οι περισσότερες εταιρείες λογισμικού σήμερα χρησιμοποιούν κάποιο σύστημα Git. Τα αποθετήρια τύπου Git ωστόσο βρίσκουν εφαρμογή ακόμη και σε ατομικά έργα, γιατί η διαχείριση εκδόσεων παρέχει τεράστια ευχρηστία επιτρέποντας στον προγραμματιστή να πειραματιστεί σε νέες αλλαγές, μπορώντας πάντοτε να ανατρέξει σε μία προηγούμενη έκδοση.

Υπάρχουν αρκετές πλατφόρμας που φιλοξενούν αποθετήρια Git, μεταξύ των οποίων οι πιο γνωστές είναι οι Github, Gitlab και Bitbucket

[](https://sparkbox.com/foundry/github_vs_gitlab_vs_bitbucket)

5.3.2 Github

Για τη δική μας εφαρμογή δημιουργήσαμε ένα αποθετήριο στο Github, το οποίο από το 2018 είναι θυγατρική εταιρεία της Microsoft. Στο αποθετήριο αυτό δημιουργήσαμε σταδιακά, και ανάλογα με τις τρέχουσες κάθε φορά ανάγκες, τρεις κλάδους για την υλοποίηση της εφαρμογής. Ο λόγος που επιλέξαμε το Github είναι καταρχάς γιατί έχουμε μεγάλη εξοικείωση με αυτό, παρέχεται δωρεάν, επειδή ενσωματώνεται πολύ καλά στον ολοκληρωμένο πρόγραμμα επεξεργασίας πηγαίου κώδικα Visual Studio Code, και τέλος επειδή παρείχε δυνατότητα για παράλληλη δημιουργία ενός Github project το οποίο περιλαμβάνει το Kanban cardboard που αναφέρθηκε προηγουμένως.

Το αποθετηρίο μας στο github, το οποίο περιλαμβάνει τον πηγαίο κώδικα της εφαρμογής και όλα τα σχετικά με τη διπλωματική μας αρχεία βρίσκεται σε αυτόν τον [σύνδεσμο](https://github.com/kitsorfan/physAI_demo_V2).

Κεφάλαιο 6

Αναλυτική παρουσίαση της εφαρμογής physAI

Τί είναι;

Είναι δωρεάν;

Σε ποιους απευθύνεται; (και στους νέους με σωματικά προβλήματα + έρευνες)

Δυνατότητες - Επισκόπηση

Πλεονεκτήματα – Μειονεκτήματα

# Κεφάλαιο 7

Μελέτη απόδοσης νευρωνικών δικτύων

Συγκριση Posenet vs Movenet και οι βαριάντες τους  
Αλλαγή παραμέτρων

# Κεφάλαιο 8

Αποτελέσματα-Προεκτάσεις

8.1 Από το proof of concept στη λειτουργική εφαρμογή

Όπως εξηγήσαμε στο Κεφάλαιο 5 ένα από τα χαρακτηριστικά της ανάπτυξης μίας SaaS εφαρμογής είναι ότι δε σταματάει ποτέ («innovate or die»). Έτσι παρά την πλήρη παρουσίαση της βασικής λειτουργικότητας της εφαρμογής θα ήταν παράλειψη να μην αναφέρουμε τη δυνητική εξέλιξη, γιατί δε θα ήμασταν συνεπείς με τις αρχές διαχείρισης του έργου.

Καταρχάς όπως ήδη εξηγήσαμε η έκδοση της εφαρμογής PhysAI που παρουσιάσαμε είναι δοκιμαστική, περιλαμβάνοντας τη βασική λειτουργικότητα της εφαρμογής, δηλαδή την αναγνώριση και ταξινόμηση της κίνησης που εκτελεί ο χρήστης, καθώς και του αριθμού των επαναλήψεων της. Συνολικά η εφαρμογή μπορεί να εντοπίζει 4 κινήσεις. Η επιδίωξη μας ήταν να αποδείξουμε την εφικτότητα της λύσης, στόχος οποίος επιτεύχθηκε. Η αναμενόμενη συμπεριφορά της εφαρμογής στην πλήρη της έκδοση, στην οποία θα διαθέτει λειτουργική βάση δεδομένων και διακομιστή back-end έχει ήδη αναφερθεί στο κεφάλαιο 5.

Πέραν όμως των όσων απαιτούνται για την πλήρη έκδοση και της χρήσης των αντίστοιχων υπολογιστικών πόρων, υπό τη μορφή κλιμακώσιμης υπολογιστικής νέφους θα πρέπει να δοθεί έμφαση στην ασφάλεια των δεδομένων των χρηστών.

Τέλος σε λειτουργικό επίπεδο η εφαρμογή σε επόμενη της έκδοση θα μπορεί:

* να αναγνωρίζει τους λόγους για τους οποίους μία άσκηση δεν εκτελέστηκε και να προτείνει στο χρήστη και στον εκπαιδευτή λύσεις
* να εκτιμά καλύτερα την απόσταση του χρήστη από την κάμερα και να του προτείνει καλύτερη θέση τοποθέτησης του κινητού του τηλεφώνου αν το κρίνει,
* να χρησιμοποιεί το σύστημα των ειδοποιήσεων για να υπενθυμίζει στο χρήστη να εκτελέσει τις ασκήσεις του
* να μπορεί να δέχεται φωνητικές εντολές για να μπορεί να εξυπηρετήσει καλύτερα τα άτομα με κινητικές δυσκολίες.

8.2 Επέκταση και σε άλλους τομείς δραστηριοτήτων

Η εφαρμογή PhysAI που αναγνωρίζει και αξιολογεί κινητικές ασκήσεις δεν είναι απαραίτητο να περιορίζεται μόνο σε φυσικοθεραπείες. Στην πράξη το νευρωνικό δίκτυο που κατασκευάσαμε, με την κατάλληλη εκπαίδευση είναι σε θέση να αναγνωρίζει πολλών ειδών διαφορετικές κινήσεις, από αθλητικές ασκήσεις μέχρι χορευτικά βήματα. Ένας άλλος τομέας στον οποίο μπορεί να επεκταθεί είναι στην οπτική αναγνώριση κινητικών ασκήσεων σε ζώα, η οποία μπορεί να βρει εφαρμογές στην εκπαίδευση κατοικίδιων και συνοδών ζώων.

8.3 Σχεδιαστικές καινοτομίες

Παράλληλα μία μελλοντική έκδοση της PhysAI μπορεί να ενσωματώσει και άλλες καινοτόμες τεχνολογίες που ακόμα βρίσκονται στα πρώτα τους στάδια.

8.3.1 Συναισθηματική Υπολογιστική (affective computing)

Μία από αυτές είναι η συναισθηματική υπολογιστική (affective computing), δηλαδή η δυνατότητα του συστήματος να αναγνωρίσει και να ερμηνεύσει ανθρώπινα συναισθήματα. Η αναγνώριση των συναισθημάτων των άλλων ανθρώπων είναι μία δύσκολη γενικά διαδικασία, στην οποία ακόμα και οι άνθρωποι, ιδίως οι φοιτητές της Σχολής ΗΜΜΥ, πολλές φορές αποτυγχάνουν. Αυτό συμβαίνει γιατί η αναγνώριση προκύπτει από έμμεσες παρατηρήσεις που αφορούν τη φωνή (πχ. χροιά, ένταση κτλ), τις εκφράσεις του προσώπου, τη στάση του σώματος, τις χειρονομίες, τις λέξεις στην ομιλία που χρησιμοποιεί κτλ. Όλα τα παραπάνω μπορούν να λειτουργήσουν ως είσοδος σε κατάλληλα νευρωνικά δίκτυα που θα προσπαθήσουν να εντοπίσουν τη συναισθηματική κατάσταση του χρήστη.

[[https://apothesis.eap.gr/bitstream/repo/41502/1/114276\_ΡΟΥΣΣΟΥ\_ΑΝΝΑ.pdf](https://apothesis.eap.gr/bitstream/repo/41502/1/114276_%CE%A1%CE%9F%CE%A5%CE%A3%CE%A3%CE%9F%CE%A5_%CE%91%CE%9D%CE%9D%CE%91.pdf)]

Ειδικά στην περίπτωση μας η συναισθηματική υπολογιστική μπορεί να φανεί πολλές εφαρμογές όπως:

* Αναγνώριση έκφρασης πόνου κατά την εκτέλεση κάποιας άσκησης. Η εφαρμογή θα μπορεί να είναι σε θέση να καταλάβει αν κάποιος ασθενής πονάει κατά την τέλεση μιας άσκησης και να αξιολογεί το πόσο σημαντικός αυτός είναι, οπότε αν κρίνεται αναγκαίο να προτείνει την ακύρωση ή την τροποποίηση της.
* Αναγνώριση αίσθηματος πλήξης, ώστε να παροτρύνει κατάλληλα το χρήστη να συνεχίζει την άσκησή του.
* Αναγνώριση κόπωσης, ώστε να μπορεί αντίστοιχα να προτείνει διαλείμματα ανάπαυσης

8.3.2 Παιχνιδοποίηση (Gamification)

8.3.3 Μικτή πραγματικότητα

# Κεφάλαιο 9

Επίλογος

Στον επίλογο να αναφέρουμε ότι ουσιαστικά η διπλωματική μας αποτέλεσμα μία συνόψιση πολλών μαθημάτων από Υ, Λ, (αλληλεπίδραση, νευρωνικά, λειτουργικά, SaaS, Softeng, Τεχνητή Νοημοσύνη, Ανάλυση Πληροφοριακών Συστημάτων)

Επίσης για τον αντίκτυπο της εφαρμογής στην κοινωνία κτλ

Βιβλιογραφία

[] "Pandemic disruptions mean many older adults still haven’t gotten needed care", National Poll on Healthy Aging, 2022

-

[] "Factors affecting therapeutic compliance: A review from the patient’s perspective", Pubmed, 2008

[] "COVID-19 exposes the critical importance of patient rehabilitation", World Health Organization, 2020

[] Rausch A. "Physiotherapists’ use and perceptions of digital remote physiotherapy during COVID-19 lockdown in Switzerland: an online cross-sectional survey - Archives of Physiotherapy", BioMed Central, 2021

[] "Introduction to Smart eHealth and eCare Technologies" Routledge & CRC Press. 2020

[] Sofia Voutsidou, "E-Health Applications for Smart and Pervasive Healthcare in Greece. What Can We Expect?", 2021

-

[] Secinaro S. Calandra D. Secinaro A. Muthurangu V. Biancone P. "The role of Artificial Intelligence in healthcare: A structured literature review", BMC medical informatics and decision making, 2022

[] Kayaalp M. "Patient privacy in the era of Big Data", Balkan medical journal", 2018

[] Erik R. Ranschaert, Sergey Morozov, Paul R. Algra "Artificial Intelligence in Medical Imaging: Opportunities, Applications and Risks" 1st ed. 2019 Edition

[]Turner A. "How many people have smartphones worldwide" BankMyCell, 2022, https://www.bankmycell.com/blog/how-many-phones-are-in-the-world

[] "The pain squad app", International Association for the Study of Pain (IASP), 2021

[] Sohaib Aslam, Sanne van Luenen, Shaista Aslam S. David Van Bodegom Niels Chavannes "A systematic review on the use of mHealth to increase physical activity in older people", 2022

[] Kakria P. Tripathi N. Kitipawang P. "A real-time health monitoring system for remote cardiac patients using smartphone and wearable sensors", International journal of telemedicine and applications, 2015

[] Frank W. Christian K. Matthew J. "Lack of exercise is a major cause of chronic diseases", Comprehensive Physiology, 2022

[] Cacheaux René Berlin J. "Advanced Ios app Architecture", Razeware LLC, 2nd ed. 2020 Edition

[] Rosenblatt, F. "the Perceptron: A probabilistic model for information storage and organization in the brain", Semantic scholar, 1970

[] Aggarwal C. Neural networks and deep learning: A textbook, Springer International Publishing, 1st edition 2018

[] Haykin S. Neural networks and learning machines, Pearson, 2009

[] Vlahavas I. Kefalas P. Bassiliades N. Kokkoras F. Sakellariou I. Artificial Intelligence - 3rd Εdition, , University of Macedonia Press, 2011.

[] S. J. Russell, P. Norvig. Artificial intelligence: a modern approach (3rd ed.). Pearson, 2013.

[] Bishop C. Pattern recognition and machine learning. Boston: Academic Press, 1992.

[] Ashwin B. Maithili B. Pranav G. Rohan C. "Applications of Convolutional Neural Networks", International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2016

[] Lucas A. Iliadis M. Molina R. Katsaggelos A. "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods", Northwestern University publications, 2018

[] Fischler M. Elschlager R. "The Representation and Matching of Pictorial Structures", IEEE Transactions on Computers, Volume: C-22, Issue: 1, 1973

[] Toshev A. Szegedy C. "DeepPose: Human Pose Estimation via Deep Neural Networks", IEEE Conference on Computer Vision and Pattern Recognition, 2014

[] http://www.mackenziemathislab.org/deeplabcut

[]https://waverleysoftware.com/blog/yarn-vs-npm/

[] Müller A. Guido S. "Introduction to machine learning with python a guide for Data scientists", O'Reilly, 2018

[] https://insights.stackoverflow.com/survey/2021#technology

[] https://javascript.info/intro

[] https://en.wikipedia.org/wiki/TypeScript [https://radixweb.com/blog/typescript-vs-javascript]

[] https://insights.stackoverflow.com/survey/2021#technology

[] https://reactjs.org/

[] Zhu T. Chen L. Gidaris S. Tompson J. Murphy K. "PersonLab: Person Pose Estimation andInstance Segmentation with a Bottom-Up,Part-Based, Geometric Embedding ModelGeorge Papandreou", European Conference on Computer Vision,2018

[] https://github.com/google-coral/project-posenet

[] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition, 2016

[] Howard A. Zhu M. Chen B. Kalenichenko D. Wang W. Weyand T. Andreetto M. Adam H. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Google Inc., 2017

[] https://storage.googleapis.com/movenet/MoveNet.SinglePose%20Model%20Card.pdf

[] Armando F. Patterson D. Τεχνολογία ανάπτυξης λογισμικού ως υπηρεσίας: μία ευέλικτη προσέγγιση με χρήση υπολογιστικής νέφους, εκδόσεις Κλειδάριθμος, ελληνική έκδροση 2017

[] Pressman S. Maxim R. Τεχνολογία Λογισμικού: Μια πρακτική Προσέγγιση, εκδόσεις ΤΖΙΟΛΑ, ελληνική έκδοση 2019

[] Sommerville I. Τεχνολογία Προϊόντων Λογισμικού: μια εισαγωγή στη σύγχρονη τεχνολογία λογισμικού, εκδόσεις Κλειδάριθμος, ελληνική έκδοση 2020

[] Silberschatz A. Galvin P. Gagne G. Λειτουργικά Συστήματα, εκδόσεις Γκιούρδας, 2013

[] Ρούσσου Α. "Συναισθηματική Υπολογιστική: Αναγνώριση συναισθήματος από ομιλία με χρήση τεχνικών βαθιάς μάθησης", Ελληνικό Ανοικτό Πανεπιστήμιο, ελληνική έκδοση 2019

Εικόνες