



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΑΣ**

**Τμήμα Ψηφιακών Συστημάτων**

**«Σχεδιασμός και υλοποίηση ρομποτικού συστήματος βασισμένο σε IoT τεχνολογίες, για εφαρμογή σποράς ακριβείας και ελέγχου καλλιέργειας θερμοκηπίου»**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΠΑΠΑΣΤΕΡΓΙΟΥ ΓΕΩΡΓΙΟΣ (ΑΜ: 8117008)**

**Επιβλέπων: Δρ. Απόστολος Ξενάκης**

**ΛΑΡΙΣΑ 2019**



*«Εγώ ο Παπαστεργίου Γεώργιος, δηλώνω υπεύθυνα ότι η παρούσα Πτυχιακή Εργασία με τίτλο «Σχεδιασμός και υλοποίηση ρομποτικού συστήματος βασισμένο σε IoT τεχνολογίες, για εφαρμογή σποράς ακριβείας και ελέγχου καλλιέργειας θερμοκηπίου» είναι δική μου και βεβαιώνω ότι:*

- Σε όσες περιπτώσεις έχω συμβουλευτεί δημοσιευμένη εργασία τρίτων, αυτό επισημαίνεται με σχετική αναφορά στα επίμαχα σημεία.*
- Σε όσες περιπτώσεις μεταφέρω λόγια τρίτων, αυτό επισημαίνεται με σχετική αναφορά στα επίμαχα σημεία. Με εξαίρεση τέτοιες περιπτώσεις, το υπόλοιπο κείμενο της πτυχιακής αποτελεί δική μου δουλειά.*
- Αναφέρω ρητά όλες τις πηγές βοήθειας που χρησιμοποίησα.*
- Σε περιπτώσεις που τμήματα της παρούσας πτυχιακής έγιναν από κοινού με τρίτους, αναφέρω ρητά ποια είναι η δική μου συνεισφορά και ποια των τρίτων.*
- Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος για την επέλευση των νομίμων συνεπειών»*

*< υπογραφή >*

**ΠΑΠΑΣΤΕΡΓΙΟΥ ΓΕΩΡΓΙΟΣ**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

**Τόπος:** .....

**Ημερομηνία:** .....

### **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. ....

2. ....

3. ....

# Περίληψη

Η παρούσα εργασία αφορά τη μελέτη, το σχεδιασμό και την υλοποίηση μιας Internet of Things (IoT) πλατφόρμας η οποία ελέγχει μια ελαφριά τροχήλατη ρομποτική κατασκευή και σε συνεργασία με υπολογιστικό σύστημα, το οποίο λειτουργεί ως έμπειρο σύστημα, διεκπεραιώνουν σπορά ακριβείας και παρακολούθηση καλλιέργειας ενημερώνοντας κατάλληλα τον υπεύθυνο του θερμοκηπίου.

Η ρομποτική κατασκευή διεκπεραιώνει σπορά ακριβείας, παρακολούθηση των περιβαλλοντικών συνθηκών της παραγωγής, επιθεώρηση της κατάστασης της υγείας των φυτών, διάγνωση ασθενειών και παρέχει διορθωτικά μέτρα χωρίς να υποβαθμίζει την ποιότητα του εδάφους λόγω του μικρού της βάρους.

Ο στόχος της ενσωμάτωσης του Έμπειρου Συστήματος που αναπτύχθηκε είναι να φανεί χρήσιμο στον αρχάριο χρήστη παρέχοντάς του γνώσεις τις οποίες μόνο κάποιος εμπειρογνώμονας θα μπορούσε να γνωρίζει. Μπορεί επίσης να χρησιμοποιηθεί και από τους ειδικούς, οι οποίοι έχουν τη δυνατότητα χρησιμοποιώντας το, να παίρνουν καλύτερες αποφάσεις και να παρέχουν χρήσιμες συμβουλές στους καλλιεργητές.

Με τη βοήθεια του προτεινόμενου συστήματος, που μπορεί να θεωρηθεί ότι προσφέρει “χέρια”, “μάτια” και “μυαλό” από απόσταση, ο υπεύθυνος του θερμοκηπίου παρακολουθεί και ελέγχει από τη φορητή του συσκευή την παραγωγή, παίρνει γρήγορα σωστές αποφάσεις που θα τον βοηθήσουν να βελτιώσει και να αναβαθμίσει την ποσότητα και την ποιότητα του παραγόμενου προϊόντος, αξιοποιώντας όλες τις τεχνολογίες αιχμής.



# Ευχαριστίες

Με την περάτωση της παρούσας εργασίας θα ήθελα να ευχαριστήσω θερμά τον κ. Αποστόλη Ξενάκη, Δρ. Ηλεκτρολόγο Μηχανικό και Μηχανικό Υπολογιστών του Πανεπιστημίου Θεσσαλίας για την εμπιστοσύνη που έδειξε στο πρόσωπό μου, για τη δυνατότητα που μου έδωσε να πραγματοποιήσω την παρούσα εργασία και για την απαραίτητη καθοδήγηση και τις επιστημονικές συμβουλές που μου προσέφερε από τη θέση του επιβλέποντα.

Θα ήθελα να ευχαριστήσω επίσης του φίλους μου που ήταν δίπλα τόσο στην δημιουργία της εργασίας μου όσο και σε όλη την διάρκεια των σπουδών μου.

Τέλος οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου που καθ' όλη την διάρκεια των σπουδών μου υπήρξαν αρωγοί των προσπαθειών μου και ήταν δίπλα μου προσφέροντας την πολύτιμη συμπαράστασή τους, δίχως την οποία η εργασία αυτή δεν θα είχε ολοκληρωθεί.

Παπαστεργίου Γεώργιος  
ημερομηνία





# Περιεχόμενα

ΠΕΡΙΛΗΨΗ .....	I
ΕΥΧΑΡΙΣΤΙΕΣ.....	III
ΠΕΡΙΕΧΟΜΕΝΑ.....	V
<b>1 ΕΙΣΑΓΩΓΗ .....</b>	<b>1</b>
<b>2 ΑΝΑΣΚΟΠΗΣΗ ΤΟΥ ΓΝΩΣΤΙΚΟΥ ΠΕΔΙΟΥ .....</b>	<b>3</b>
2.1 AGRICULTURE 4 .....	3
2.1.1 Γεωργία ακριβείας .....	3
2.1.2 Ψηφιακός μετασχηματισμός αγροτικών επιχειρήσεων .....	6
2.1.3 Αγροτικό IoT.....	13
2.2 BIG DATA ΑΓΡΟΤΙΚΟΥ ΤΟΜΕΑ .....	15
2.3 ROBOT ΣΤΗ ΓΕΩΡΓΙΑ.....	21
2.3.1 Ψηφιακές καλλιεργητικές δεξιότητες .....	21
2.3.2 Πράσινη γεωργία με τη ρομποτική.....	26
2.4 ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ .....	28
2.4.1 Χρήση της μηχανικής όρασης .....	28
2.4.2 Τεχνικές επεξεργασίας εικόνας .....	31
2.5 ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ .....	31
2.5.1 Εισαγωγή.....	31
2.5.2 Ορισμός της Τεχνητής Νοημοσύνης.....	32
2.5.3 Διαχωρισμός της τεχνητής από τη φυσική Νοημοσύνη.....	35
2.5.4 Τομείς της Τεχνητής Νοημοσύνης .....	36
2.6 ΈΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ .....	38
2.6.1 Ορισμός .....	38
2.6.2 Δομή ενός Έμπειρου Συστήματος.....	39
2.6.3 Ανάπτυξη Έμπειρων Συστημάτων .....	42
2.6.4 Έμπειρα Συστήματα και συμβατικά προγράμματα .....	49

2.6.5	Πλεονεκτήματα/Μειονεκτήματα του ΕΣ σε σχέση με τον Ανθρωπο-Ειδικό.....	50
2.7	ΔΙΑΓΝΩΣΗ ΑΣΘΕΝΕΙΩΝ ΤΩΝ ΦΥΤΩΝ.....	51
2.8	ΚΟΝΤΙΝΕΣ ΕΡΕΥΝΗΤΙΚΕΣ ΠΡΟΣΠΑΘΕΙΕΣ.....	54
<b>3</b>	<b>ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ.....</b>	<b>69</b>
3.1	SYSTEM MODEL .....	69
3.2	ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ.....	71
3.2.1	Περιγραφή αρχιτεκτονικής.....	71
3.2.2	Διαγράμματα ροής.....	73
3.3	ΤΙ ΠΕΤΥΧΑΙΝΟΥΜΕ ΜΕ ΤΟ ΣΥΣΤΗΜΑ.....	78
3.3.1	Συγκεντρωτικός έλεγχος από μια φορητή συσκευή.....	78
3.3.2	Άμεση ενημέρωση του αγρότη.....	80
3.3.3	Προγραμματισμός συγκομιδής.....	81
3.3.4	Απομακρυσμένη εποπτεία καλλιέργειας.....	82
3.3.5	Αυτοματοποιημένη σπορά ακριβείας.....	82
3.3.6	Δυνατότητα έγκαιρου και έγκυρου εντοπισμού αρχομένων ασθενειών των φυτών.....	84
3.3.7	Παροχή κατάλληλων συμβουλών θεραπείας των ασθενειών ....	84
<b>4</b>	<b>ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ.....</b>	<b>85</b>
4.1	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ANDROID ΕΦΑΡΜΟΓΗΣ “GREEN-OS”.....	85
4.2	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ-ΡΥΘΜΙΣΕΙΣ ΤΗΣ ΜΟΝΑΔΑΣ EV3.....	91
4.3	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΠΛΑΚΕΤΑΣ ESP 8266 ΩΣ REST WEB SERVER.....	93
4.4	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ REST CLIENT ΕΦΑΡΜΟΓΗΣ “URL_REQUEST” .....	107
4.5	ΒΟΗΘΗΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ .....	122
4.5.1	mobile app “IP Webcam”.....	122
4.5.2	FTP Server στα Windows 10.....	125
4.5.3	mobile app “FtpCafe” (FTP client) .....	138
4.6	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ DIAGNOSIS SUPPORT SYSTEM (DISS).....	142
4.6.1	Προγραμματισμός του μοντέλου του ταξινομητή εικόνων με χρήση συνελκτικού νευρωνικού δικτύου (CNN).....	142
4.6.2	Προγραμματισμός της διεπαφής χρήστη για τη διάγνωση των ασθενειών .....	159

<b>5</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ - ΔΥΝΑΤΟΤΗΤΕΣ ΕΠΕΚΤΑΣΗΣ .....</b>	<b>165</b>
5.1	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	165
5.2	ΔΥΝΑΤΟΤΗΤΕΣ ΕΠΕΚΤΑΣΗΣ .....	166
	<b>ΠΗΓΕΣ .....</b>	<b>169</b>
	<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>173</b>
	<b>ΠΑΡΑΡΤΗΜΑ Β.....</b>	<b>175</b>



# 1 Εισαγωγή

Σκοπός της παρούσας πτυχιακής εργασίας είναι να σχεδιασθεί και να δημιουργηθεί πλατφόρμα IoT η οποία θα ελέγχει ρομποτική κατασκευή και σε συνεργασία με υπολογιστικό σύστημα θα μπορούν να διεκπεραιώνουν σπορά ακριβείας και παρακολούθηση καλλιέργειας και θα ενημερώνουν κατάλληλα τον υπεύθυνο του θερμοκηπίου.

Βασικός στόχος είναι να δημιουργηθούν οι βάσεις για τον ψηφιακό μετασχηματισμό των επιχειρήσεων που δραστηριοποιούνται στον αγροτικό τομέα ώστε να επωφεληθούν από την 4η βιομηχανική επανάσταση, να αναβαθμιστεί η παραγόμενη τροφή σε ποσότητα και ποιότητα και να αντιμετωπιστεί ένα από τα μεγαλύτερα προβλήματα του πλανήτη, αυτό της σίτισης του υπερπληθυσμού.

Επιμέρους στόχοι:

- α) να εισάγει την αυτοματοποίηση στον αγροτικό τομέα ώστε να καλύψει το κενό που αφήνει το γερασμένο εργατικό δυναμικό.
- β) να αυξήσει τις αποδόσεις και να μειώσει τις εισροές στην περιορισμένη καλλιεργήσιμη έκταση του πλανήτη.
- γ) να καλύψει την έλλειψη ή την καθυστερημένη άφιξη γεωπόνων μέσω των τεχνολογιών επεξεργασίας εικόνας, τεχνητής νοημοσύνης και έμπειρων συστημάτων οι οποίες θα είναι σε θέση να κάνουν έγκαιρη αλλά και έγκυρη διάγνωση αρχομένων ασθενειών των φυτών.
- δ) να οδηγήσει στην αντικατάσταση των υπαρχουσών βαρέων αγροτικών μηχανών με πιο ελαφρές οι οποίες θα κινούνται με πιο «καθαρά» καύσιμα και θα έχουν μικρότερο περιβαλλοντικό αποτύπωμα.
- ε) να βοηθήσει στην εφαρμογή τεχνικών γεωργίας ακριβείας.
- στ) να άρει την δυσκολία των αγροτών στη χρήση νέων τεχνολογιών παρέχοντας τον έλεγχο της καλλιέργειας μέσα από φιλικές και εύχρηστες εφαρμογές.
- ζ) να αυξήσει την εμπιστοσύνη των αγροτών στις ρομποτικές μηχανές και στα αυτοματοποιημένα έμπειρα συστήματα.

Στο κεφάλαιο 2 γίνεται ανασκόπηση του ευρύτερου γνωστικού πεδίου στο οποίο εντάσσεται η πτυχιακή και αναφέρονται παρόμοιες εργασίες που έχουν προταθεί και υλοποιηθεί καθώς και με ποιο τρόπο προσεγγίστηκαν οι λύσεις τους.

Στο κεφάλαιο 3 γίνεται περιγραφή του προτεινόμενου συστήματος και αποσαφηνίζονται όλες οι τεχνικές λεπτομέρειες που αφορούν τον σχεδιασμό και την λειτουργικότητά του και αναφέρεται τι πετυχαίνουμε με το σύστημα.

Στο κεφάλαιο 4 γίνεται πλήρης αναφορά και περιγραφή των εργαλείων υλοποίησης (λογισμικών) που χρησιμοποιήθηκαν, δηλαδή περιβάλλοντα προγραμματισμού, βιβλιοθήκες, βοηθητικές εφαρμογές και δίνεται και ένας σύντομος οδηγός χρήσης.

Στο κεφάλαιο 5 γίνεται αναφορά στο τι τελικά υλοποιήθηκε, καταγράφονται συμπεράσματα και προτείνονται βελτιώσεις και επεκτάσεις.

## 2 Ανασκόπηση του γνωστικού πεδίου

Στις επόμενες παραγράφους γίνεται μια αναφορά στο agriculture 4 και στα big data του αγροτικού τομέα, εξετάζονται οι τεχνικές επεξεργασίας εικόνας, η τεχνητή νοημοσύνη, τα έμπειρα συστήματα και η εφαρμογή τους στα γεωργικά robot για διάγνωση ασθενειών των φυτών. Στο τέλος του κεφαλαίου παραθέτονται κάποιες κοντινές ερευνητικές προσπάθειες.

### 2.1 Agriculture 4

#### 2.1.1 Γεωργία ακριβείας

Γεωργία ακριβείας είναι να κάνεις τη σωστή ενέργεια τη σωστή στιγμή στο σωστό σημείο του χωραφιού με τη σωστή ποσότητα. Για να γίνει αυτό είναι απαραίτητη η χρήση της επιστήμης και της σύγχρονης τεχνολογίας στη γεωργία. (Κουρούδης, 2017)

Χρησιμοποιούνται διάφοροι όροι για το ίδιο πράγμα. Στο εξωτερικό ξεκίνησε με το όρο precision agriculture αλλά χρησιμοποιείται και το digital farming. Εδώ το λέμε γεωργία ακριβείας ή ψηφιακή γεωργία ή ευφυή γεωργία. (ο. π.)

Η Έξυπνη Γεωργία αντιπροσωπεύει την εφαρμογή των σύγχρονων Τεχνολογιών Πληροφορίας και Επικοινωνιών (ΤΠΕ) στην γεωργία, που οδηγεί σε αυτό που μπορεί να ονομαστεί ως τρίτη Πράσινη Επανάσταση. Μετά τις επαναστάσεις αναπαραγωγής των φυτών και της γενετικής, αυτή η τρίτη Πράσινη Επανάσταση αρχίζει και επιβάλλεται στο γεωργικό κόσμο με βάση τη συνδυασμένη εφαρμογή των λύσεων ΤΠΕ όπως ο εξοπλισμός ακριβείας, το Διαδίκτυο των Πραγμάτων (Internet of Things – IoT), οι αισθητήρες και ενεργοποιητές, τα συστήματα γεω-εντοπισμού, τα Μεγάλα Δεδομένα (Big Data), τα μη επανδρωμένα εναέρια οχήματα (Unmanned Aerial Vehicles – UAV), τα μη επανδρωμένα αεροσκάφη, η ρομποτική, κλπ. Η Έξυπνη Γεωργία έχει πραγματική δυνατότητα να δώσει πιο παραγωγική και βιώσιμη γεωργική παραγωγή, η οποία θα βασίζεται σε μια προσέγγιση πιο ακριβούς και

αποδοτικής χρήσης των πόρων. Ωστόσο, ενώ στις ΗΠΑ το 20-80% των γεωργών χρησιμοποιούν κάποιο είδος Τεχνολογίας Έξυπνης Γεωργίας, στην Ευρώπη μόνο το 0% -24% των γεωργών τις χρησιμοποιεί. Από την πλευρά του γεωργού, η Έξυπνη Γεωργία θα πρέπει να παρέχει στο γεωργό προστιθέμενη αξία με τη μορφή της καλύτερης λήψης αποφάσεων ή της πιο αποτελεσματικής λειτουργίας και διαχείρισης της εκμετάλλευσής του. Με αυτή την έννοια, η Έξυπνη Γεωργία συνδέεται στενά με τρεις διασυνδεδεμένους τομείς τεχνολογιών: (Δίκτυο smart-AKIS, 2016).

- **Πληροφοριακά Συστήματα Διοίκησης:** Προγραμματισμένα συστήματα για τη συλλογή, επεξεργασία, αποθήκευση και διάδοση των δεδομένων με τη μορφή που απαιτείται για την εκτέλεση των εργασιών και λειτουργιών μιας γεωργικής επιχείρησης. (ο. π.)
- **Γεωργία Ακριβείας:** Διαχείριση της χωρικής και χρονικής μεταβολής για τη βελτίωση της οικονομικής απόδοσης σε συνδυασμό με τη μείωση των εισροών και των περιβαλλοντικών επιπτώσεων. Περιλαμβάνει Συστήματα Υποστήριξης Αποφάσεων (Decision Support Systems – DSS) για ολόκληρη την διαχείριση των γεωργικών εκμεταλλεύσεων με στόχο την βελτιστοποίηση των αποδόσεων επί των εισροών με παράλληλη διατήρηση των πόρων, τα οποία χαρακτηρίζονται από την ευρεία χρήση των συστημάτων γεω-εντοπισμού (GPS, GNSS), αεροφωτογραφιών από UAVs και την τελευταία γενιά των υπερχρονικών εικόνων που παρέχονται από τους δορυφόρους Sentinel, που έχουν σαν συνέπεια τη δημιουργία χαρτών χωρικής παραλλακτικότητας διαφόρων μεταβλητών που μπορεί να μετρηθούν (π.χ. απόδοση των καλλιεργειών, χαρακτηριστικά του εδάφους / τοπογραφία, περιεκτικότητα σε οργανική ύλη, επίπεδα υγρασίας, επίπεδα αζώτου, κλπ.). (ο. π.)
- **Γεωργικοί αυτοματισμοί και ρομποτική:** Η διαδικασία της εφαρμογής της ρομποτικής, του αυτόματου ελέγχου και των τεχνικών τεχνητής νοημοσύνης σε όλα τα επίπεδα της γεωργικής παραγωγής, συμπεριλαμβανομένων των farmbots και των farmdrones. (ο. π.)

Οι εφαρμογές Έξυπνης Γεωργίας δεν στοχεύουν μόνο σε μεγάλες, συμβατικής γεωργίας εκμεταλλεύσεις, αλλά θα μπορούσαν επίσης να δράσουν για την τόνωση άλλων κοινών ή αυξανόμενων τάσεων στις γεωργικές εκμεταλλεύσεις, όπως η οι-



κογενειακή γεωργία (μικρές ή συγκρότημα εκμεταλλεύσεων, ειδικές καλλιέργειες ή / και κτηνοτροφία, διατήρηση υψηλής ποιότητας ή ειδικές ποικιλίες, κτλ.) και η βιολογική γεωργία, και να ενισχύσουν την γεωργία ως ένα κλάδο που εμπνέει σεβασμό και διαφάνεια στον ευρωπαϊό καταναλωτή, την κοινωνία και τη συνείδηση της αγοράς. Η Έξυπνη Γεωργία μπορεί επίσης να παρέχει μεγάλα οφέλη σχετικά με το περιβάλλον, για παράδειγμα, μέσω της αποτελεσματικότερης χρήσης του νερού, ή της βελτιστοποίησης των γεωργικών πρακτικών. (ο. π.)

Η ανάγκη για γεωργία ακριβείας είναι υπαρκτή και μεγάλη.

Ο παγκόσμιος πληθυσμός αυξάνεται με πολύ γρήγορους ρυθμούς. Το 2050 θα είμαστε 10 δισεκατομμύρια και όλες οι μελέτες δείχνουν ότι θα χρειαζόμαστε τα διπλάσια αγροτικά προϊόντα από ότι σήμερα. Και μάλιστα με ποιοτικές προδιαγραφές που θα γίνονται όλο και πιο αυστηρές. Επίσης θα πρέπει να παράγονται με βιώσιμες γεωργικές πρακτικές. (Κουρούδης, 2017)

Όμως η καλλιεργήσιμη έκταση είναι συγκεκριμένη: το 3% της επιφάνειας της γης και από αυτήν μόνο το 18% είναι για παραγωγή τροφίμων. (ο. π.)

Λύση σε αυτό μπορεί να δώσει μόνο η γεωργία ακριβείας, που αυξάνει τις στρεμματικές αποδόσεις, μειώνει τις εισροές, ρίχνει το κόστος παραγωγής, ελαττώνει το περιβαλλοντικό αποτύπωμα και βελτιώνει την ποιότητα. (ο. π.)

Πέραν τούτων: Η τεχνολογική εξέλιξη είναι κάτι που έρχεται αναπόφευκτα σε όλους τους τομείς. Το Industry 4.0 είναι ευρέως γνωστό. Αυτό που είναι λιγότερο γνωστό είναι το **Agriculture 4.0**. οι τρεις προηγούμενες επαναστάσεις στον αγροτικό τομέα ήταν η εισαγωγή της εκμηχάνισης, η χρήση ορυκτών λιπασμάτων και η εκβιομηχάνιση των παραγωγικών διαδικασιών. Η τέταρτη αγροτική επανάσταση έρχεται τώρα με τη συνδεσιμότητα το αγροτικό IoT και τη διαχείριση δεδομένων. (ο. π.)

Η γεωργία ακριβείας ενδιαφέρει οποιονδήποτε έχει να κάνει με τον αγροτικό τομέα και όχι μόνο. Πρώτα και κύρια βέβαια τους αγρότες μιας και η γεωργία ακριβείας αυξάνει τις στρεμματικές αποδόσεις, μειώνει το κόστος και βελτιώνει την ποιότητα. Επειδή όμως η αγορά αυτή είναι τεράστια και θα αποτελέσει βασικό τομέα ανάπτυξης τα επόμενα χρόνια, ενδιαφέρει άμεσα και τις μεγάλες εταιρείες. (ο. π.)

Όλοι οι τεχνολογικοί κολοσσοί ασχολούνται εντατικά με αυτήν. Η Microsoft έκανε μια θυγατρική την Farmbeats για να φέρει το Internet of Things στη γεωργία.

Η Monsanto με τη θυγατρική της Climate Corp., θέλει να δημιουργήσει το μεγαλύτερο δίκτυο αισθητήρων δεδομένων στα χωράφια στον κόσμο. Η Bayer, δημιούργησε επίσης ένα νέο μεγάλο τομέα γεωργίας ακριβείας. Το Bloomberg εκτιμά ότι το 2015 επενδύθηκαν 25 δισεκατομμύρια δολάρια στο «precision agriculture», ενώ τα έσοδα από τα γεωργικά ρομπότ θα φτάσουν τα 74,1 δισεκατομμύρια δολάρια παγκοσμίως μέχρι το 2024, σύμφωνα με την Tractica. (ο. π.)

Πέραν τούτων τα δεδομένα από μόνα τους έχουν μεγάλη αξία και για «τρίτους». Η επιτροπή εμπορίου των ΗΠΑ εκτιμά ότι η αξία των δεδομένων αγροτικού τομέα στις ΗΠΑ ανέρχεται στο 1 δις. Δολάρια ετησίως και αγοραστές μπορεί να είναι εταιρείες ανάλυσης, ασφαλιστικές εταιρείες, βιομηχανίες κατασκευής γεωργικών μηχανημάτων κλπ. (ο. π.)

### 2.1.2 Ψηφιακός μετασχηματισμός αγροτικών επιχειρήσεων

Η αντικατάσταση της ανθρώπινης εργασίας με αυτοματοποίηση είναι μια αυξανόμενη τάση σε πολλές βιομηχανίες και η γεωργία δεν αποτελεί εξαίρεση. Οι περισσότερες πτυχές της γεωργίας είναι εξαιρετικά έντονες, με μεγάλο μέρος της εργασίας να αποτελείται από επαναλαμβανόμενα και τυποποιημένα καθήκοντα - μια ιδανική θέση για τη ρομποτική και τον αυτοματισμό.

Βλέπουμε ήδη γεωργικά ρομπότ - ή AgBots - που αρχίζουν να εμφανίζονται σε αγροκτήματα και εκτελούν καθήκοντα που κυμαίνονται από φύτευση και άρδευση έως συγκομιδή και διαλογή. Τελικά, αυτό το νέο κύμα έξυπνου εξοπλισμού θα καταστήσει δυνατή την παραγωγή περισσότερων και υψηλότερης ποιότητας τροφίμων με λιγότερα ανθρώπινα δυναμικά. (Εικόνα 1) (Ομάδα Γεωπόνων της Farmacon, 2018)



Εικόνα 1. χρήση ρομποτικών βραχιόνων σε χωράφι

## Τρακτέρ χωρίς οδηγό

Τα τρακτέρ είναι η καρδιά μιας αγροτικής εκμετάλλευσης τα οποία χρησιμοποιούνται για πολλές διαφορετικές εργασίες, ανάλογα με το είδος της εκμετάλλευσης και τη διαμόρφωση του βοηθητικού εξοπλισμού της. Καθώς προωθούνται οι αυτόνομες τεχνολογίες οδήγησης, τα τρακτέρ αναμένεται να καταστούν μερικές από τις πρώτες μηχανές που πρέπει να μετατραπούν σε αυτόνομες μηχανές. (ο. π.)

Σε πρώιμα στάδια, θα χρειαστεί ακόμα ανθρώπινη προσπάθεια για τη δημιουργία τοπογραφικών χαρτών χωραφιού και ορίων, για τον προγραμματισμό των καλύτερων διαδρομών μέσα στο χωράφι χρησιμοποιώντας λογισμικό προγραμματισμού διαδρομών και για τη λήψη άλλων λειτουργικών συνθηκών. Οι άνθρωποι θα εξακολουθούν να απαιτούνται για τακτική επισκευή και συντήρηση. (ο. π.)

Ωστόσο, οι αυτόνομοι ελκυστήρες θα γίνουν πιο ικανοί και αυτοσυντηρούμενοι με την πάροδο του χρόνου, ειδικά με τη συμπερίληψη πρόσθετων μηχανών και συστημάτων μηχανικής όρασης, GPS πλοήγησης, διασύνδεσης IoT για την απομακρυσμένη παρακολούθηση και λειτουργία και ραντάρ και LiDAR για ανίχνευση και αποφυγή αντικειμένων. Όλες αυτές οι τεχνολογικές εξελίξεις θα μειώσουν σημαντικά την ανάγκη των ανθρώπων να ελέγχουν ενεργά αυτά τα μηχανήματα. (Εικόνα 2) (ο. π.)



Εικόνα 2. αυτόνομοι ρομποτικοί ελκυστήρες

Απώτερος σκοπός είναι αυτοί οι ελκυστήρες να χρησιμοποιούν στο μέλλον «μεγάλα δεδομένα» όπως π.χ. δεδομένα σε πραγματικό χρόνο, καλύτερη χρήση των ιδανικών συνθηκών, ανεξάρτητα από την ανθρώπινη εισροή και ανεξάρτητα από την ώρα της ημέρας. (ο. π.)

### **Σπορά και φύτευση**

Η σπορά ήταν κάποτε μια επίπονη χειρωνακτική διαδικασία. Η σύγχρονη γεωργία βελτιώθηκε σε αυτό με μηχανές σποράς, οι οποίες μπορούν να καλύψουν περισσότερο έδαφος πολύ πιο γρήγορα από τον άνθρωπο. Ωστόσο, αυτές συχνά χρησιμοποιούν μια μέθοδο διασκορπισμού που μπορεί να είναι ανακριβής και σπάταλη όταν οι σπόροι βρίσκονται εκτός της βέλτιστης τοποθεσίας. Η αποτελεσματική σπορά απαιτεί έλεγχο πάνω σε δύο μεταβλητές: φύτευση σπόρων στο σωστό βάθος και φυτά σε κατάλληλη απόσταση ώστε να επιτρέπεται η βέλτιστη ανάπτυξη. (ο. π.)

Ο εξοπλισμός σποράς ακριβείας έχει σχεδιαστεί για να μεγιστοποιεί τις μεταβλητές αυτές κάθε φορά. Ο συνδυασμός δεδομένων γεωμετρίας και αισθητήρων που αναλύουν την ποιότητα του εδάφους, την πυκνότητα, την υγρασία και τα επίπεδα θρεπτικών ουσιών απαιτεί πολλές εικασίες από τη διαδικασία σποράς. (ο. π.)

Οι σπόροι έχουν την καλύτερη πιθανότητα να φυτρώσουν και να αναπτυχθούν και η συνολική καλλιέργεια θα έχει μεγαλύτερη συγκομιδή. (ο. π.)

Καθώς η γεωργία μετακινείται στο μέλλον, οι υπάρχοντες σπαρτικές μηχανές ακριβείας θα έρθουν σε επαφή με αυτόνομους ελκυστήρες και συστήματα με δυνατότητα IoT που θα τροφοδοτούν τις πληροφορίες πίσω στον αγρότη. Ένα ολόκληρο χωράφι θα μπορούσε να φυτευτεί με αυτό τον τρόπο, με μόνο έναν άνθρωπο να παρακολουθεί τη διαδικασία μέσω μιας τροφοδοσίας βίντεο ή ενός πίνακα ψηφιακού ελέγχου σε έναν υπολογιστή ή tablet, ενώ πολλές μηχανές κυλούν στο χωράφι ταυτόχρονα. (Εικόνα 3) (ο. π.)



Εικόνα 3. σπαρτική μηχανή ακριβείας ελεγχόμενη από πλατφόρμα IoT

### **Αυτόματη άρδευση**

Η στάγδην άρδευση είναι ήδη μια επικρατούσα μέθοδος άρδευσης που επιτρέπει στους αγρότες να ελέγχουν πότε και πόσο νερό λαμβάνουν οι καλλιέργειες τους. Συνδυάζοντας αυτά τα συστήματα SDI με ολοένα και πιο εξελιγμένους αισθητήρες με δυνατότητα IoT, ώστε να παρακολουθούν συνεχώς τα επίπεδα υγρασίας και την υγεία των φυτών, οι αγρότες θα μπορούν να παρεμβαίνουν μόνο όταν είναι απαραίτητο, επιτρέποντας έτσι το σύστημα να λειτουργεί αυτόνομα. (ο. π.)

Ενώ τα συστήματα SDI δεν είναι ακριβώς ρομποτικά, θα μπορούσαν να λειτουργήσουν εντελώς αυτόνομα σε ένα έξυπνο αγρόκτημα, βασιζόμενα σε δεδομένα από αισθητήρες που υφίστανται γύρω από τα χωράφια για να εκτελούν άρδευση ανάλογα με τις εκάστοτε αρδευτικές ανάγκες. (ο. π.)

### **Έλεγχος ζιζανίων και συντήρηση καλλιεργειών**

Ο έλεγχος των ζιζανίων και των παρασίτων είναι κρίσιμες πτυχές της συντήρησης των καλλιεργειών και είναι αυτές οι αγροτικές εργασίες που είναι ιδανικές για αυτόνομα ρομπότ. (ο. π.)

Έχει κατασκευαστεί, παραδείγματος χάριν, ένα ρομπότ περίπου στο μέγεθος ενός αυτοκινήτου και μπορεί να πλοηγηθεί αυτόνομα μέσα από ένα χωράφι χρησιμοποιώντας βίντεο, LiDAR και δορυφορικό GPS. Οι προγραμματιστές του χρησιμοποιούν τη μηχανική μάθηση για να διδάξουν σε αυτό να εντοπίζει τα ζιζάνια πριν τα αφαιρέσει. Με την προηγμένη μηχανική μάθηση ή ακόμα και την τεχνητή

νοημοσύνη (ΑΙ) που ενσωματώνεται στο μέλλον, μηχανήματα όπως αυτό θα μπορούσαν να αντικαταστήσουν πλήρως την ανάγκη για ανθρώπους να παρακολουθούν καλλιέργειες ή να κάνουν την παραπάνω διαδικασία χειροκίνητα. (Εικόνα 4) (ο. π.)



Εικόνα 4. ρομπότ που ξεχορταριάζει

Ένα άλλο αντίστοιχο ρομπότ που έχει κατασκευασθεί λειτουργεί λίγο διαφορετικά. Ο καλλιεργητής του ρυμουλκείται πίσω από έναν ελκυστήρα και είναι εφοδιασμένος με συστήματα απεικόνισης που μπορούν να αναγνωρίσουν μια φθορίζουσα χρωστική ουσία την οποία καλύπτουν οι σπόροι όταν φυτεύονται και η οποία μεταφέρεται στα νεαρά φυτά καθώς αυτά φυτρώνουν και αρχίζουν να αναπτύσσονται. Ο καλλιεργητής στη συνέχεια κόβει τα μη λαμπερά ζιζάνια. Οι παραπάνω ρομποτικές μηχανές είναι σχεδιασμένες τόσο για τον έλεγχο των ζιζανίων όσο και για το εντοπισμό εχθρών και ασθενειών, χρησιμοποιώντας τους κατάλληλους κάθε φορά αισθητήρες. (Εικόνα 5) (ο. π.)





Εικόνα 5. ρομποτικές μηχανές για έλεγχο των ζιζανίων

### Συγκομιδή

Υπάρχει ένα πλήθος μηχανών οι οποίες είναι σε θέση να συγκομίζουν μηχανικά τους καρπούς που είναι έτοιμοι για συγκομιδή. (ο. π.)

Η ανάπτυξη τεχνολογίας ικανής για ευαίσθητες εργασίες συγκομιδής, όπως η συλλογή καρπών από δέντρα ή λαχανικά όπως οι ντομάτες, είναι όπου τα υψηλής τεχνολογίας αγροκτήματα θα διαπρέψουν πραγματικά. Οι μηχανικοί εργάζονται για να δημιουργήσουν τα σωστά ρομποτικά εξαρτήματα για αυτά τα εξελιγμένα καθήκοντα, όπως το ρομπότ συλλογής τομάτας της Panasonic, το οποίο ενσωματώνει εκλεπτυσμένες κάμερες και αλγόριθμους για τον προσδιορισμό του χρώματος, του σχήματος και της θέσης της ντομάτας για τον προσδιορισμό της ωριμότητάς της. (Εικόνα 6) (ο. π.)



Εικόνα 6. ρομποτικός βραχίονας για συγκομιδή φράουλας

## Μείωση της εργασίας, αύξηση της παραγωγής και της αποδοτικότητας

Η βασική ιδέα της ενσωμάτωσης της αυτόνομης ρομποτικής στη γεωργία παραμένει ο στόχος της μείωσης της εξάρτησης από τη χειρωνακτική εργασία, ενώ παράλληλα η επίτευξη της αύξησης της παραγωγής, της απόδοσης και της ποιότητας του προϊόντος. (ο. π.)

Ο αγρότης του μέλλοντος θα ξοδέψει το χρόνο του εκτελώντας καθήκοντα όπως η επισκευή μηχανημάτων, η σωστή χρήση ρομποτικών μηχανών, η ανάλυση δεδομένων και ο αποτελεσματικός σχεδιασμός των αγροτικών επιχειρήσεων. (ο. π.)

Όπως σημειώνεται με όλα αυτά τα γεωργικά ρομπότ, η ύπαρξη μιας σωστής βάσης αισθητήρων και IoT ενσωματωμένων στην υποδομή της γεωργικής εκμετάλλευσης είναι απαραίτητη. **Το κλειδί για μια πραγματικά “έξυπνη” εκμετάλλευση βασίζεται στην ικανότητα όλων των μηχανών και των αισθητήρων να επικοινωνούν μεταξύ τους και με τον αγρότη, ακόμη και όταν αυτοί λειτουργούν αυτόνομα.** (Εικόνα 7) (ο. π.)



Εικόνα 7. πλατφόρμα IoT για αγροτική χρήση



### 2.1.3 Αγροτικό IoT

Καινοτόμα και αυτόνομα γεωργικά ρομπότ είναι σίγουρα χρήσιμα μηχανήματα, αλλά αυτό που θα κάνει πραγματικά έξυπνο το μελλοντικό αγρόκτημα, είναι αυτό που φέρνει όλη αυτή την τεχνολογία μαζί: **το Διαδίκτυο των πραγμάτων**. (Ομάδα Γεοπόνων της Farmacon, 2018)

Το IoT έχει γίνει ένα κομμάτι ενός συνόλου όρων για την ιδέα της ύπαρξης ηλεκτρονικών υπολογιστών, μηχανών, εξοπλισμού και συσκευών παντός τύπου, ανταλλαγής δεδομένων και επικοινωνίας με τρόπους που τους επιτρέπουν να λειτουργούν ως το λεγόμενο "έξυπνο σύστημα. Βλέπουμε ήδη τεχνολογίες IoT που χρησιμοποιούνται με πολλούς τρόπους, όπως έξυπνες οικιακές συσκευές και ψηφιακοί βοηθοί, έξυπνα εργοστάσια και έξυπνες ιατρικές συσκευές. (ο. π.)

Η έξυπνη φάρμα θα έχει ενσωματωμένους αισθητήρες σε κάθε στάδιο των καλλιεργητικών διαδικασιών και σε κάθε εξοπλισμό. Οι αισθητήρες που είναι εγκατεστημένοι σε όλα τα χωράφια θα συλλέγουν δεδομένα σχετικά με το **επίπεδο φωτισμού**, τις **συνθήκες του εδάφους**, την **άρδευση**, την **ποιότητα του αέρα** και τον **καιρό**. Αυτά τα δεδομένα θα επιστρέψουν στον αγρότη, ή απευθείας σε εξειδικευμένα γεωργικά ρομπότ στο χωράφι. (Εικόνα 8) (ο. π.)



Εικόνα 8. εφαρμογή για συλλογή δεδομένων από αισθητήρες σε χωράφι

Ομάδες από αυτά τα ρομπότ θα διασχίζουν τα χωράφια και θα εργάζονται αυτόνομα για να ανταποκριθούν στις ανάγκες των καλλιεργειών και θα εκτελούν λει-

τουργίες ζιζανιοκτονίας, άρδευσης, κλαδέματος και συγκομιδής, οι οποίες και θα καθοδηγούνται από τη δική τους συλλογή αισθητήρων, πλοήγησης και δεδομένων των καλλιεργειών. (ο. π.)

Drones θα περιηγούνται στον ουρανό και θα συλλέγουν στοιχεία και δεδομένα όσον αφορά στην υγεία των φυτών και τις εδαφολογικές συνθήκες, ή θα δημιουργούν χάρτες που θα καθοδηγούν τα ρομπότ και θα βοηθούν τους αγρότες να σχεδιάσουν τα επόμενα βήματα για τη γεωργική τους εκμετάλλευση. Όλα αυτά θα συμβάλουν στη δημιουργία μεγαλύτερης καλλιέργειας φυτών και στην αύξηση της διαθεσιμότητας και της ποιότητας των τροφίμων. (ο. π.)

Εκτιμάται πως οι συσκευές IoT που έχουν εγκατασταθεί στη γεωργία θα αυξηθούν από 30 εκατομμύρια το 2015 σε 75 εκατομμύρια μέχρι το 2020. Σύμφωνα με αυτή την τάση, οι συνδεδεμένες εκμεταλλεύσεις αναμένεται να παράγουν έως και 4,1 εκατομμύρια σημεία δεδομένων κάθε μέρα το 2050- μόλις 190.000 το 2014. (Εικόνα 9) (ο. π.)



Εικόνα 9. έλεγχος συνθηκών καλλιέργειας μήλου από εφαρμογή για smartphone

Αυτό το βουνό δεδομένων και άλλες πληροφορίες που παράγονται από την τεχνολογία των γεωργικών εκμεταλλεύσεων και η συνδεσιμότητα που επιτρέπει την κοινή χρήση τους θα αποτελέσουν τη ραχοκοκαλιά της μελλοντικής έξυπνης εκμε-

τάλλευσης. Οι αγρότες θα μπορούν να “βλέπουν” όλες τις πτυχές της λειτουργίας τους – τα φυτά είναι υγιή ή χρειάζονται προσοχή, ένα χωράφι μπορεί να χρειάζεται νερό... και επιτέλους θα λαμβάνονται τεκμηριωμένες αποφάσεις. (ο. π.)

Και αυτή η συζήτηση έχει αγγίξει μόνο την άκρη του παροιμιώδους παγόβουνου με την εστίαση στις φυτικές καλλιέργειες. Υπάρχει σαφώς μια ίση αφετηρία της υιοθέτησης της έξυπνης φάρμας για την κτηνοτροφία αλλά και για οποιαδήποτε πτυχή της γεωργίας. Εάν κάθε αγρόκτημα στη χώρα γίνει ένα έξυπνο αγρόκτημα, η επίτευξη αυτής της αύξησης της παραγωγής τροφίμων κατά 70% είναι μια βεβαιότητα. (ο. π.)

## 2.2 Big data αγροτικού τομέα

Ευρισκόμενοι από τη σκοπιά του καλλιεργητή ως μονάδα, ο **κύριος στόχος** είναι η **κερδοφορία** σε ένα όλο και πιο ανταγωνιστικό περιβάλλον το οποίο απαιτεί όλο και **μεγαλύτερες αποδόσεις, παραγωγές και βιωσιμότητα**. (Ομάδα Γεωπόων της Farmacon, 2018)

Οι αγρότες διαρκώς βρίσκονται μπροστά σε επικείμενες λήψεις αποφάσεων. Τότε λοιπόν είναι που αντιμετωπίζουν διλήμματα πολλά και **διατρέχουν τον άμεσο κίνδυνο της λήψης μη σωστής απόφασης** τελικά. Τέτοια παραδείγματα είναι αποφάσεις σχετιζόμενες με αστάθμητους παράγοντες όπως, το κλίμα, τα καιρικά φαινόμενα, τη διαθεσιμότητα νερού, αποτυχίες εξοπλισμού, το κόστος των γεωργικών εισροών και πολλά άλλα. (ο. π.)

Το πρόβλημα στην ουσία δεν είναι η μη σωστή τελικά απόφαση αλλά το ότι οι αγρότες δεν είναι σε θέση να προβλέψουν, αρά και να δράσουν άμεσα, στην περίπτωση δυσμενών αποφάσεων. (ο. π.)

Η θρέψη από μόνη της είναι ένας πολύπλοκος και ιδιαίτερος γεωργικός τομέας, γεγονός που από μόνο του καθιστά μια ουτοπία την εύρεση της «βέλτιστης θρεπτικής φόρμουλας» για οποιαδήποτε καλλιέργεια. Γι’ αυτόν ακριβώς το λόγο οι περισσότεροι καλλιεργητές εξακολουθούν να βασίζονται σε δοκιμές, εικασίες και εκτιμήσεις. Το μόνο σίγουρο αποτέλεσμα είναι καλλιέργειες που δεν ικανοποιούν

τις δυνατότητες παραγωγής τους και την αυξημένη περιβαλλοντική ρύπανση αλλά και σπατάλη χρημάτων. (ο. π.)

Για την ελαχιστοποίηση αυτού του κινδύνου θα πρέπει ο κάθε παραγωγός πλέον, να μπορεί να λαμβάνει **κατάλληλες και πιο ενημερωμένες αποφάσεις**, κάτι που απαιτεί από τους αγρότες και τους αγροτικούς συμβούλους να βασίζονται περισσότερο στα **ακριβή δεδομένα** και τα **αναλυτικά στοιχεία** τους, παρά στη διαίσθησή τους. (ο. π.)

Και ενώ μέχρι προσφάτως οι γενικές κατευθύνσεις και η διαισθητικές αποφάσεις χαρακτηρίζαν τους περισσότερους, ακόμη και επαγγελματίες παραγωγούς, ήρθε τώρα η σειρά της τεχνολογίας να αναλάβει δράση και να παίζει σημαντικό ρόλο στη λήψη των τελικών αποφάσεων, στηριζόμενη πλέον σε πραγματικά στοιχεία και όχι εικασίες. (Εικόνα 10) (ο. π.)



Εικόνα 10. εφαρμογή παρακολούθησης περιβαλλοντικών συνθηκών χωραφιού

Η "**έξυπνη γεωργία**" περιλαμβάνει μια τεχνολογικά μεσολαβούμενη μελέτη του τρόπου με τον οποίο οι διακυμάνσεις στον αγρό επηρεάζουν την ανάπτυξη των φυτών. (ο. π.)

Τέτοιες τεχνολογίες επιτρέπουν στους αγρότες να αυξάνουν την παραγωγικότητα, να εξοικονομούν πόρους και να ελαχιστοποιούν τις περιβαλλοντικές επιπτώσεις. (ο. π.)

Η δυνατότητα συλλογής ψηφιακών δεδομένων είναι η επανάσταση στη γεωργία. Επιτρέπει τη συλλογή πολύ μεγάλου όγκου δεδομένων, την ενσωμάτωση πηγών δεδομένων, την πρόσβαση στα δεδομένα από οπουδήποτε και την ανάλυση μεγάλων συνόλων δεδομένων για τη λήψη αποφάσεων. (ο. π.)

**Εδώ είναι μερικές από τις βασικές διαδικασίες συλλογής και ανάλυσης δεδομένων:**

► **Δορυφορικές εικόνες** αλλά και **εικόνες από drones** - χρησιμοποιώντας τεχνολογίες τηλεανίχνευσης, όπως οι δορυφορικές εικόνες, παρέχουν δεδομένα που μπορούν να χρησιμοποιηθούν για τη βελτιστοποίηση του αζώτου. (ο. π.)

► Διάφοροι **φασματικοί δείκτες** αναπτύσσονται και αναλύονται. Οι πιο συνηθισμένοι είναι ο NDVI (Κανονικοποιημένος Δείκτης Βλάστησης), ο οποίος εκτιμά την κάλυψη της βλάστησης, τον CCCI (δείκτη περιεχομένου χλωροφύλλης κόμης) και τον CNI (δείκτης αζώτου της κόμης). (ο. π.)

► **Αισθητήρες χωραφίου** - αισθητήρες αζώτου μπορούν να εγκατασταθούν στο χωράφι ή σε μηχανήματα, όπως ελκυστήρες, και να συλλέξουν δεδομένα σχετικά με τα επίπεδα αζώτου. (ο. π.)

► **Μετεωρολογικά δεδομένα** - δεδομένα από χιλιάδες μετεωρολογικούς σταθμούς συλλέγονται και αναλύονται. Η ακριβής πρόγνωση του καιρού είναι σημαντική για πολλές αποφάσεις που λαμβάνουν οι γεωργοί, συμπεριλαμβανομένου του χρονοδιαγράμματος της φύτευσης και των εφαρμογών λιπασμάτων. (ο. π.)

► **Βιβλιοθήκες δεδομένων** - Αρκετοί διεθνείς οργανισμοί έχουν δημοσιεύσει βιβλιοθήκες δεδομένων σχετικά με τα εδάφη και τις ιδιότητές τους, μερικές εταιρείες ψηφιακού τύπου δημιουργούν βιβλιοθήκες δεδομένων για τις απαιτήσεις των θρεπτικών ουσιών, τις αποδόσεις και τα εδάφη με βάση τις αναλύσεις δεδομένων. (ο. π.)

► Οι γεωργικές εισροές - αρκετοί αγρότες χρησιμοποιούν εφαρμογές cloud στις οποίες καταγράφονται τα δεδομένα των χωραφιών (τι εφαρμόστηκε, πότε εφαρμόστηκε κ.τ.λ.) Αυτά τα συγκεντρωτικά στοιχεία από τους γεωργούς δημιουργούν νέες ευκαιρίες για τη διάδοση της γνώσης. (ο. π.)

Οι αναλύσεις των δεδομένων χωραφίου και οι συνεχώς ενημερωμένες βάσεις γνώσεων αποτελούν στην ουσία μια επανάσταση για τον τρόπο λήψης των καθημερινών αποφάσεων από τους παραγωγούς. (ο. π.)

Σήμερα, οι αγρότες μπορούν ήδη να έχουν άμεση πρόσβαση σε εργαλεία λήψης αποφάσεων εφαρμογών cloud, τα οποία μεταφράζουν τεράστια αριθμό δεδομένων και αναλύσεων σε βέλτιστες πρακτικές και πληροφορίες που μπορούν να αποτελέσουν αντικείμενο δράσης. (ο. π.)

Σε αντίθεση με τα περισσότερα άλλα εργαλεία στη γεωργία, οι αλγόριθμοι λογισμικού και οι αναλύσεις δεδομένων δεν απαιτούν εγκατάσταση hardware. Στην ιδανική περίπτωση, η λύση λογισμικού θα πρέπει να είναι εύκολα ενσωματωμένη σε οποιαδήποτε εξωτερική πηγή δεδομένων, όπως δορυφόροι, αισθητήρες, drones, μηχανήματα και ρομπότ. (ο. π.)

Νέες ευκαιρίες προσφέρονται για **βελτιστοποίηση της απόδοσης με τη διαχείριση της εφαρμογής των λιπασμάτων**. Η βελτιστοποίηση της χρήσης λιπασμάτων απαιτεί συνεχή και **πολύπλοκη ανάλυση μεγάλων δυναμικών συνόλων δεδομένων**. Αυτό επιτυγχάνεται με τη δημιουργία λογισμικών τα οποία θα βοηθήσουν τους παραγωγούς να αντιμετωπίσουν σύνθετα προβλήματα των καλλιεργειών, χρησιμοποιώντας μεγάλο αριθμό δεδομένων και γνώσης. (ο. π.)

Με λίγη προσπάθεια, οι αγρότες μπορούν να επωφεληθούν από μια τέτοια ανάλυση και να βελτιώσουν σημαντικά την αποτελεσματικότητά τους και τη λήψη αποφάσεων. (ο. π.)

Για να γίνει περισσότερο κατανοητό, ένα τέτοιου είδους λογισμικό μπορεί να ενσωματώνει εισροές για το κλίμα, το έδαφος, το νερό, τη γενετική, την ενέργεια, τους οικονομικούς πόρους, το ιστορικό του χωραφίου, τις αποδόσεις και άλλα. (ο. π.)



Για παράδειγμα, η **χρήση άζωτου** μπορεί να γίνει πιο αποτελεσματική με την ανάλυση των ιδιοτήτων του εδάφους και του ρυθμού διείσδυσης του νερού μέσα σε αυτό και τη σύνδεσή του με τη δοκιμή εδάφους εντός του χωραφίου, τη χαρτογράφηση του εδάφους και της περιεκτικότητας σε άζωτο, δεδομένα σχετικά με τα ποσοστά πρόσληψης άζωτου από την καλλιέργεια, χάρτες αποδόσεων, θερμοκρασίες, κατακρήμνιση και άλλα δεδομένα. (ο. π.)

Σαφώς όμως δεν είναι μόνο το άζωτο που μας ενδιαφέρει, υπάρχουν **13 απαραίτητα θρεπτικά συστατικά που απαιτούνται για την κατάλληλη ανάπτυξη των φυτών**. Το επίπεδο καθενός από αυτά θα πρέπει να βελτιστοποιηθεί και τα αναλυτικά στοιχεία μπορούν να υποστηρίξουν μια τόσο περίπλοκη ανάλυση. (ο. π.)

Η **ταχεία ανάλυση** των μεγάλων σειρών δεδομένων επιτρέπει επίσης στους αγρότες να αντιδρούν σε γεγονότα σε πραγματικό χρόνο. (ο. π.)

Η **ανάλυση των δορυφορικών εικόνων** μπορεί να εντοπίσει προβλήματα σε συγκεκριμένες περιοχές του χωραφίου, βοηθώντας τον αγρότη να εντοπίσει το πρόβλημα πιο εύκολα από ποτέ άλλοτε. (ο. π.)

Ένα άλλο παράδειγμα για τη χρήση της ανάλυσης μεγάλου αριθμού δεδομένων είναι η **ικανότητα εκτίμησης του δυναμικού απόδοσης** για κάθε περιοχή του χωραφίου. (ο. π.)

**Ανάλυση δεδομένων σημαίνει πρόβλεψη και πρόληψη μιας κατάστασης**

Οι **προγνωστικές αναλύσεις χρησιμοποιούν στατιστικά μοντέλα και αλγόριθμους** για την πρόβλεψη μελλοντικών γεγονότων και συμπεριφορών. Αυτή η δυνατότητα έγινε δυνατή λόγω της συλλογής μεγάλου αριθμού δεδομένων. (ο. π.)

Η ανάλυση ιστορικών δεδομένων, όπως οι αποδόσεις, ο καιρός, οι τάσεις στο έδαφος, οι εισροές λιπασμάτων και άλλα, μαζί με δεδομένα σε πραγματικό χρόνο, δίνουν στον γεωργό ισχυρά εργαλεία για να λαμβάνει τεκμηριωμένες αποφάσεις και να διαχειρίζεται τους κινδύνους. (ο. π.)

## Πρόσβαση στα δεδομένα

Τα αποτελέσματα ερευνών και οι επιστημονικές δημοσιεύσεις δεν είναι άμεσα διαθέσιμα στους αγρότες. Όταν είναι διαθέσιμα, τα δεδομένα συχνά είναι δύσκολο να ερμηνευθούν, είναι μερικά ή δεν σχετίζονται με τις ειδικές συνθήκες του αγρότη. Έτσι, οι αγρότες δεν μπορούν συνήθως να επωφεληθούν και να αποκτήσουν γνώσεις από τέτοια δεδομένα. (ο. π.)

**Σήμερα, οι αγρότες μπορούν να έχουν άμεση πρόσβαση σε εργαλεία λήψης αποφάσεων βασισμένα σε εφαρμογές cloud, τα οποία μεταφράζουν τεράστια ποσά δεδομένων και αναλύσεων σε βέλτιστες πρακτικές και πληροφορίες που μπορούν να αποτελέσουν αντικείμενο επεξεργασίας.** (ο. π.)

Τα δεδομένα και η γνώση γίνονται διαθέσιμα σε όλους, μεγάλους ή μικρούς αγρότες, σε ανεπτυγμένες ή αναπτυσσόμενες χώρες. Ακόμη και οι παραδοσιακοί αγρότες μπορούν να ασκήσουν εξειδικευμένη λίπανση σε κάποιο βαθμό και να βελτιώσουν σημαντικά την παραγωγικότητα και τη βιωσιμότητά τους. (Εικόνα 11) (ο. π.)



Εικόνα 11. εφαρμογή cloud διαθέσιμη στον αγρότη

Υπάρχει μια συνεχής συζήτηση για το «ποιός κατέχει τα δεδομένα». Οι αγρότες ανησυχούν πολύ για αυτή την ερώτηση. Τα δεδομένα που συλλέγονται με τη χρήση αισθητήρων στις εκμεταλλεύσεις τους διατηρούνται σε ένα σύννεφο (cloud),



οι δορυφόροι λαμβάνουν εικόνες των χωραφιών και αναλύουν αυτές τις πληροφορίες. (ο. π.)

Παρόλο που αυτές οι νέες τεχνολογίες υπόσχονται πολλά και θαυμαστά αποτελέσματα, αυτό δεν αρκεί. (ο. π.)

Για την τεχνολογία που υιοθετεί ο γεωργός, η λύση πρέπει να είναι, η ευκολία στη χρήση, να προσθέτει πραγματική αξία στο τελικό προϊόν και να μπορεί εύκολα να ενσωματωθεί με τον υπάρχοντα εξοπλισμό και την καθημερινή ροή εργασίας. (ο. π.)

## **2.3 Robot στη γεωργία**

### **2.3.1 Ψηφιακές καλλιεργητικές δεξιότητες**

Ας ξεχάσουμε λοιπόν τα σκιάχτρα, το μέλλον της γεωργίας βρίσκεται στα χέρια των μηχανημάτων. (Εικόνα 12) (Ομάδα Γεωπόνων της Farmacon, 2017)

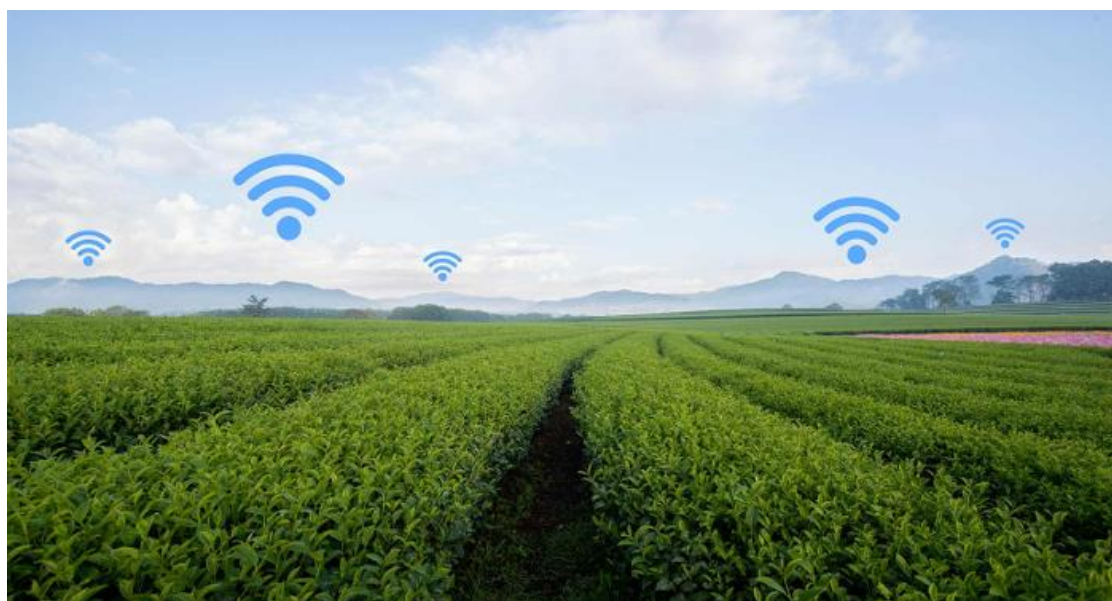


Εικόνα 12. το μέλλον της γεωργίας είναι ψηφιακό

Είναι γεγονός, η ανθρωπότητα αντιμετωπίζει ένα υπαρκτό πρόβλημα. Από τη στιγμή που το είδος της το *Homo sapiens*, κάθε χρόνο αυξάνεται ενώ ο πλανήτης γη παραμένει πεισματικά ο ίδιος σε μέγεθος, αυτομάτως αντιλαμβανόμαστε πως η ίδια έκταση γης θα πρέπει να είναι σε θέση να θρέψει πληθυσμούς ανθρώπων που διαρκώς αυξάνονται. Εάν σε όλα τα παραπάνω δεδομένα συμπεριλάβουμε και την υπερθέρμανση του πλανήτη, την λειψυδρία αλλά και τα απρόβλεπτα και μη φυσιολογικά καιρικά φαινόμενα, ναι, τελικά η ανθρώπινη φυλή έχει σοβαρό πρόβλημα σίτισης. (ο. π.)

Τελικά, ίσως έφτασε η ώρα των μηχανημάτων. Πραγματικά έξυπνα, πραγματικά εντυπωσιακά ρομπότ και αλγόριθμοι μπορούν να βοηθήσουν να ξεκινήσουμε μια νέα Πράσινη Επανάσταση ώστε οι ανθρωπότητα να συνεχίζει να τροφοδοτείται σε έναν ολοένα και πιο θορυβώδη πλανήτη.

Σκεφτείτε δορυφόρους που ανιχνεύουν αυτομάτως την ξηρασία, τρακτέρ που παρακολουθούν από κοντά την καλλιέργεια και καταστρέφουν τα άρρωστα φυτά και μια εφαρμογή που μπορεί να πει σε έναν αγρότη ποιά ασθένεια έχει προσβάλλει την καλλιέργειά του. (Εικόνα 13) (ο. π.)



Εικόνα 13. το χωράφι του μέλλοντος θα είναι συνεχώς "online"

## Ψηφιακές πλέον καλλιεργητικές δεξιότητες

Μία από τις πλέον αποδοτικότερες χρήσεις του υπολογιστή είναι η «εκπαίδευση» του από τους προγραμματιστές, να αναγνωρίζει συγκεκριμένα προβλήματα, παρά να του λένε ρητά τι να κάνει. (ο. π.)

Θα μπορούσαμε για παράδειγμα, να τροφοδοτήσουμε μια εφαρμογή ενός ηλεκτρονικού υπολογιστή με φωτογραφίες από ασθενικά και υγιή φύλλα φυτών, τα οποία έχουν επισημανθεί ως τέτοια. (ο. π.)

Από τις φωτογραφίες αυτές θα μάθει το ίδιο το μηχάνημα να ξεχωρίζει ποιά είναι τα προσβεβλημένα και ποιά τα υγιή φύλλα και στην ουσία πώς είναι αυτά μορφολογικά και έτσι θα μπορεί να καθορίσει την υγεία των νέων φύλλων η εφαρμογή από μόνη της.

Αυτό ακριβώς έκαναν ο βιολόγος David Hughes και ο επιδημιολόγος Marcel Salathé με 14 καλλιέργειες που είχαν προσβληθεί από 26 ασθένειες. (ο. π.)

**Τροφοδότησαν έναν υπολογιστή με περισσότερες από 50.000 εικόνες και το πρόγραμμα από μόνο του μπόρεσε να αναγνωρίσει σωστά το 99,35% των νέων εικόνων που φορτώθηκαν σε αυτό.** (ο. π.)

Και βέβαια, τα προβλήματα των καλλιεργειών δεν είναι μόνο οι προσβολές από παθογόνα και οι επιδρομές εχθρών, όπως τα έντομα ή τα ακάρεα αλλά και προβλήματα θρέψης, φυσιολογικές ανωμαλίες κ.α. (ο. π.)

Σε αυτές τις περιπτώσεις, **η λάθος διάγνωση είναι πολύ συχνή με πολλά δυσάρεστα επακόλουθα για τους καλλιεργητές**, όπως χάσιμο χρημάτων, χάσιμο χρόνου αλλά και επιβάρυνση για το περιβάλλον χωρίς λόγο τις περισσότερες φορές. (ο. π.)

**Στο μέλλον, η Τεχνητή Νοημοσύνη θα μπορούσε να βοηθήσει τους αγρότες να εντοπίσουν γρήγορα και με ακρίβεια το πρόβλημα.** (ο. π.)

Λαμβάνοντας υπόψη όλα τα παραπάνω, επιτέλους οι καλλιεργητές θα μπορούν να εφησυχάσουν αρκετά όσον αφορά τον συνεχή έλεγχο των καλλιεργειών τους για πιθανά προβλήματα. (ο. π.)

Μία εφαρμογή πλέον θα μπορεί να εντοπίσει το πρόβλημα και να προσαρμόσει τη λύση σε αυτό. Βέβαια, η τεχνολογία είναι ένα μεγάλο μέρος της λύσης του προ-

βλήματος αλλά οι τελικές αποφάσεις για τη διαχείριση των προβλημάτων θα πρέπει να λαμβάνονται σε συνεργασία με εμπειρογνώμονες και σε επιτόπια βάση, δηλαδή στον αγρό. (ο. π.)

### **Η επέλαση των ιδιοφυών τρακτέρ**

Ενώ ο αναπτυσσόμενος κόσμος διψάει για γεωργικές γνώσεις, ο ανεπτυγμένος κόσμος πνίγεται σε φυτοφάρμακα και ζιζανιοκτόνα. Στις ΗΠΑ κάθε χρόνο, οι αγρότες χρησιμοποιούν 310 εκατομμύρια λίβρες ζιζανιοκτόνου - μόνο σε αγρούς καλαμποκιού, σόγιας και βαμβακιού. (ο. π.)

Μια εταιρεία που ονομάζεται Blue River Technology μπορεί να έχει τη λύση, τουλάχιστον όσον αφορά στα μαρούλια. Το LettuceBot μοιάζει με ένα τυπικό τρακτέρ, αλλά στην πραγματικότητα πρόκειται για μηχανήμα που συνεχώς μαθαίνει... Η εταιρεία ισχυρίζεται ότι το LettuceBot μπορεί να περάσει μέσα από ένα χωράφι και να φωτογραφίσει 5.000 νεαρά φυτά ανά λεπτό, χρησιμοποιώντας αλγόριθμους και μηχανική όραση για να διαχωρίσει το καλλιεργούμενο φυτό από το ζιζάνιο. (ο. π.)

Σας φαίνεται υπερβολικά και απίστευτα γρήγορο; Ένα τσιπ γραφικών μπορεί να αναγνωρίσει μια εικόνα σε μόλις 0.02 δευτερόλεπτα. (ο. π.)

Με την ακρίβεια των σχεδόν 0.6 εκατοστών, το συγκεκριμένο τρακτέρ μπορεί να εντοπίζει και να ψεκάζει κάθε ζιζάνιο στο χωράφι. (ο. π.)

Από τη στιγμή που τα φυτά μαρουλιού παρακολουθούνται στενά και διαπιστωθεί πως για κάποιο λόγο δεν αναπτύσσονται όπως θα έπρεπε, αυτά αυτόματα θα ψεκασθούν. Επίσης, στην περίπτωση που δύο φυτά είναι πολύ κοντά φυτεμένα το ένα με το άλλο, (στην ουσία με όχι ιδανικές αποστάσεις), το μηχανήμα μπορεί να τα διακρίνει και να αφαιρέσει το ένα από αυτά. (ο. π.)

Αυτό που τελικά μπορούν και κάνουν τα μηχανήματα καλύτερα από το άνθρωπο, δεν είναι τόσο η μηχανική εκμάθηση αλλά η ακρίβεια. Οι μηχανές δεν μπορούν να τρέξουν σαν το άνθρωπο ή να χειριστούν αντικείμενα όπως ακριβώς κάνει ο άνθρωπος, αλλά είναι πολύ συνεπείς και σχολαστικές. (Εικόνα 14) (ο. π.)



Εικόνα 14. τα τρακτέρ γίνονται "ιδιοφυή"

### Η ζωή από ψηλά

Σε τροχιά μεγαλύτερη από 400 μίλια πάνω από το κεφάλι μας, οι δορυφόροι παρέχουν μια εντελώς μαγική επισκόπηση της επιφάνειας της Γης σε ένα πλήθος από εύρη ζωνών πολύ πέρα από το ορατό φάσμα. Όλα αυτά τα στρώματα πληροφοριών είναι δύσκολο να γίνουν αντιληπτά από τον ανθρώπινο νου, αλλά για έναν υπολογιστή που χρησιμοποιεί και εφαρμόζει αλγορίθμους, είναι πραγματικά πολύ εύκολο.

Αυτό θα μπορούσε να είναι εξαιρετικά πολύτιμο για την παρακολούθηση της γεωργίας, ιδίως στις αναπτυσσόμενες χώρες, όπου οι κυβερνήσεις και οι τράπεζες αντιμετωπίζουν έλλειψη δεδομένων κατά τη λήψη αποφάσεων σχετικά με τους γεωργούς στους οποίους χορηγούν δάνεια ή βοήθεια έκτακτης ανάγκης. Κατά τη διάρκεια μιας ξηρασίας στην Ινδία, για παράδειγμα, όχι μόνο οι περιφέρειες θα υποφέρουν σε διαφορετικούς βαθμούς, αλλά σε αυτές τις περιοχές κάποιοι αγρότες θα μπορούσαν να έχουν καλύτερα μέσα προμήθειας νερού. Η ανάλυση δορυφορικών δεδομένων, σε μία μεγάλη κλίμακα μέσω των υπολογιστών, θα μπορούσε να βοηθήσει τα ιδρύματα να διανέμουν χρήματα πιο αποτελεσματικά. (Εικόνα 15) (ο. π.)





Εικόνα 15. επισκόπηση καλλιεργητικής έκτασης από δορυφόρο

Να διαχωρίσει τους αγρότες και τις περιοχές που έχουν πραγματικό και υπαρκτό πρόβλημα και τελικά οι τράπεζες ή οι κυβερνήσεις να μεταφέρουν χρηματική βοήθεια στο σωστό σύνολο ανθρώπων. (ο. π.)

Ενώ ένας ανθρώπινος αναλυτής μπορεί να χειριστεί 10, ίσως 15 μεταβλητές κάθε φορά, οι αλγόριθμοι μπορούν να χειριστούν 2.000 ή περισσότερες.

Αυτή είναι η νέα τάξη πραγμάτων.

Οι καλλιεργητές θα πρέπει να καλλιεργούν πιο έξυπνα.

Υπάρχουν περισσότερα δεδομένα, περισσότερα μηχανήματα και περισσότερη Τεχνητή Νοημοσύνη. (ο. π.)

### 2.3.2 Πράσινη γεωργία με τη ρομποτική

«Το να περιμένετε να δείτε έναν αγρότη και ένα ρομπότ σε ένα χωράφι που θα του κάνει όλη την δουλειά και εκείνος απλά θα χειρίζεται μία οθόνη, δεν είναι ρεαλιστικό, τουλάχιστον για το ορατό μέλλον. Όμως η ύπαρξη ρομποτικών συστημάτων για υποστηρικτικές εργασίες, που μπορεί να είναι και δύσκολες, για αναλύσεις, για μετρήσεις, για την εξασφάλιση ιχνηλασιμότητας στο τελικό προϊόν και βέβαια

για την εξοικονόμηση πολύτιμων φυσικών πόρων, όπως το νερό στις καλλιέργειες, είναι υπαρκτή και ήδη εμπορικά διαθέσιμη» σημειώνει ο Διευθυντής του **Ινστιτούτου Αγρο-τεχνολογίας και Βιο-οικονομίας (ΙΒΟ)** του Εθνικού Κέντρου Έρευνας και Τεχνολογικής Ανάπτυξης (**ΕΚΕΤΑ**). (Διαμαντίδης, 2018)

Το ρομπότ που επιδεικνύει το ΕΚΕΤΑ εφοδιάζεται με **έως και τέσσερις ισχυρούς ηλεκτρικούς κινητήρες για την κίνηση και το σύστημα διεύθυνσής του**. Αυτό του δίνει εξαιρετικές δυνατότητες έλξης και κίνησης, ενώ μπορεί να λειτουργήσει σε δύσκολα περιβάλλοντα και σε όλες τις καιρικές συνθήκες. Ακόμη, μπορεί να παραδοθεί με **τετρακίνηση** για πολύ απαιτητικές εργασίες. (ο. π.)

Ενδεικτικά ορισμένες από τις εργασίες που εκτελεί είναι:

α) Έχει τη δυνατότητα να καταπολεμήσει τον περονόσπορο στα φυτά με τη χρήση **UV-Light (υπεριώδες φως)**, εκτελώντας τη συγκεκριμένη εργασία ακόμη και νυχτερινές ώρες, (ο. π.)

β) Μπορεί με τη χρήση καμερών και ειδικού λογισμικού να **ξεχορταριάζει** με πολύ προσιτό κόστος. Έτσι αποφεύγεται η χρήση φυτοφαρμάκων για τον ίδιο σκοπό με προφανείς θετικές συνέπειες τόσο κόστους, όσο και στο επίπεδο της προστασίας του περιβάλλοντος και (ο. π.)

γ) Παρακολουθεί την εξέλιξη της παραγωγής και μπορεί να «αντιληφθεί» σημάδια αρχόμενων ασθενειών στα φυτά ή έλλειψης απαραίτητων θρεπτικών συστατικών. (ο. π.)

Όλα αυτά τα εκτελεί έχοντας πολύ μικρό βάρος και όγκο, με αποτέλεσμα **να μην επιβαρύνει και να μην υποβαθμίζει το έδαφος**. (ο. π.)

«Δεν υπάρχει, αυτή τη στιγμή, κάπου στον πλανήτη, έστω και ένα αγροτικό ρομπότ (εξαιρουμένων των αποστολών εναέριας επισκόπησης – **drones**), το οποίο να είναι σε θέση να αρχίσει και να ολοκληρώσει μια συγκεκριμένη εργασία σε πλήρη αυτονομία». (ο. π.)

Το βασικό όφελος της ανάπτυξης της ρομποτικής στη σύγχρονη γεωργία, είναι η μεταφορά της **αναπτυσσόμενης τεχνογνωσίας και των τεχνολογιών ρομποτι-**

**κής στην άνθρωπο-κινούμενη γεωργία**, με σειρά πετυχημένων παραδειγμάτων, όπως τα συστήματα πλοήγησης των **σύγχρονων γεωργικών μηχανημάτων** (auto-steering), τα συστήματα τηλεματικής, την μηχανική αναγνώριση ιδιοτήτων του εδάφους και των φυτών (ασθένειες, αναγνώριση και στοχευμένη διαχείριση ζιζανίων), την **αυτόματη χαρτογράφηση καλλιεργειών**. (ο. π.)

Γιατί όμως η χρήση της ρομποτικής στη γεωργία καθιστά τα αγροδιατροφικά προϊόντα πιο «πράσινα»; (ο. π.)

Η ανάπτυξη της αγροτικής παραγωγής προκάλεσε την άνοδο των ενεργειακών εισροών στη γεωργία, (μεγαλύτερα και πιο ισχυρά μηχανήματα) με άμεσο επακόλουθο και παράπλευρα φαινόμενα όπως η συμπίεση του εδάφους, η διάβρωση και η μείωση της οργανικής ύλης σε αυτό, η υπερβολική χρήση εδαφοβελτιωτικών, αγροχημικών και νερού κ.ά. (ο. π.)

Η ρομποτική με την υποστήριξη των τεχνολογιών πληροφορίας και επικοινωνιών, αντικαθιστά βαθμιαία τα μεγάλα ενεργοβόρα γεωργικά οχήματα και μηχανήματα με μικρότερα και περισσότερα ευφυή συστήματα (που δαπανούν και λιγότερη ενέργεια), ενώ γίνεται πράξη και η **έννοια της γεωργίας ακριβείας** με ό,τι θετικό για το περιβάλλον συνεπάγεται η τελευταία. Δηλαδή συρρίκνωση της χρήσης πόρων όπως το νερό, πολύ λιγότερα φυτοφάρμακα και λιπάσματα και μεγαλύτερη ασφάλεια στην τροφική αλυσίδα, καθώς η ταυτότητα του τελικού προϊόντος παρακολουθείται και επικαιροποιείται αδιάκοπα από την πρώτη έως την τελευταία στιγμή. (ο. π.)

## **2.4 Επεξεργασία εικόνας**

### **2.4.1 Χρήση της μηχανικής όρασης**

Στη σύγχρονη γεωργία, έχουν αναπτυχθεί τεράστιες υπολογιστικές μέθοδοι και τεχνικές που βοηθούν τους αγρότες να παρακολουθούν την ορθή ανάπτυξη των καλλιεργειών τους. Στο αρχαίο γεωργικό σύστημα, η παρατήρηση διά γυμνού οφθαλμού από τους αγρότες ή τους ειδικούς είναι η κύρια προσέγγιση που υιοθετείται για την ανίχνευση και τον εντοπισμό των ασθενειών των καλλιεργειών υπό μι-



κροσκοπικές συνθήκες στο εργαστήριο. Ωστόσο, αυτή η προσέγγιση απαιτεί συνεχή παρακολούθηση από εμπειρογνώμονες, η οποία μπορεί να είναι αρκετά δαπανηρή σε μεγάλες καλλιέργειες. Επιπλέον, σε ορισμένες αναπτυσσόμενες χώρες, οι γεωργοί πρέπει να διανύσουν μεγάλες αποστάσεις για να έλθουν σε επαφή με τους εμπειρογνώμονες, γεγονός που καθιστά την επικοινωνία μαζί τους υπερβολικά δαπανηρή και χρονοβόρα. Τα βασικά προβλήματα των καλλιεργειών είναι η ταχεία και ακριβής αναγνώριση και ταξινόμηση των ασθενειών που προκύπτει από την παρατήρηση των εικόνων των μολυσμένων φύλλων. Υπάρχουν δύο βασικά χαρακτηριστικά των μεθόδων ανίχνευσης ασθενικών φύλλων μέσω μηχανικής όρασης που πρέπει να επιτευχθούν: ταχύτητα και ακρίβεια. Η διάγνωση της ασθένειας των διαφορετικών καλλιεργειών γίνεται παραδοσιακά με το χέρι. Έχουν γίνει αρκετές προσπάθειες για την αυτοματοποίηση της διαδικασίας ταυτοποίησης της ασθένειας και την παροχή λύσης. (Subhashini, 2018)

Οι ασθένειες αποτελούν σημαντικούς ανασταλτικούς παράγοντες για την ανάπτυξη των καλλιεργειών στον γεωργικό τομέα, γεγονός που μπορεί να μειώσει σημαντικά την απόδοση των καλλιεργειών και την ποιότητα των προϊόντων. Επί του παρόντος, η διάγνωση των ασθενειών των καλλιεργειών εξαρτάται κυρίως από την ανθρώπινη οπτική αναγνώριση, αλλά αυτό μπορεί να οδηγήσει σε πρόβλημα, διότι μπορεί να γίνει λάθος διάγνωση από τους αγρότες, επειδή κρίνουν συνήθως τα συμπτώματα από τις εμπειρίες τους. Από την άλλη πλευρά, η θεραπεία της ασθένειας μπορεί να διακοπεί επειδή ο τεχνικός ή ο ειδικός δεν θα μπορέσει να πάει στην περιοχή για να διαγνώσει εγκαίρως. Σχετικά με την ανθρώπινη όραση, η τεχνική επεξεργασίας εικόνας μέσω υπολογιστή πλεονεκτεί σε ορισμένα χαρακτηριστικά όπως ταχύτητα, μεγάλος όγκος πληροφοριών και δυνατότητα να διακρίνει τη μικρή ποικιλομορφία που δεν μπορεί να διακριθεί από τα μάτια ενός ανθρώπου, οπότε η τεχνική επεξεργασίας εικόνας μέσω υπολογιστή μπορεί να βοηθήσει τους αγρότες να διακρίνουν τους λόγους και τη σοβαρότητα των ασθενειών των καλλιεργειών, και παίρνει σημαντική θεωρητική και πρακτική σημασία για τη βελτίωση της αυτόματης διαχείρισης της καλλιέργειας. Τα συνηθισμένα συμπτώματα περιλαμβάνουν την ανώμαλη αύξηση των φύλλων, την αλλοίωση των χρωμάτων, την καχεκτική ανάπτυξη, τα συρρικνωμένους και χαλασμένους λοβούς. Παρόλο που οι ασθένειες και τα παράσιτα των εντόμων μπορούν να προκαλέσουν σημαντικές απώλειες απόδοσης ή να οδηγήσουν σε θάνατο τα φυτά, επηρεάζουν άμεσα και την ανθρώπινη υγεία. (Thakre, More, Gajakosh, Yewale, Shamkuwar, 2017)

Η Ινδία είναι γνωστή γεωργική χώρα στην οποία περίπου το 70% του πληθυσμού εξαρτάται από τη γεωργία. Οι γεωργοί έχουν στη διάθεσή τους ένα ευρύ φάσμα για να επιλέξουν την κατάλληλη καλλιέργεια. Ωστόσο, η καλλιέργεια αυτών των φυτών για βέλτιστη απόδοση και ποιότητα παραγωγής είναι ως επί το πλείστον τεχνική. Μπορεί να βελτιωθεί με τη βοήθεια τεχνολογικής υποστήριξης. Η διαχείριση πολυετών φυτών απαιτεί στενό έλεγχο ειδικά για τη διαχείριση ασθενειών που μπορούν να επηρεάσουν σημαντικά την παραγωγή αλλά και τη ζωή μετά τη συγκομιδή. Η επεξεργασία εικόνας είναι η καλύτερη τεχνική που χρησιμοποιείται στις γεωργικές εφαρμογές για τους ακόλουθους λόγους: Προβλέπει την ασθένεια των φυτών από την εικόνα των φυτών. Η διάγνωση της ασθένειας των φυτών περιορίζεται από τις ανθρώπινες οπτικές ικανότητες, επειδή τα περισσότερα από τα πρώτα συμπτώματα είναι μικροσκοπικά. Αυτή η διαδικασία είναι κουραστική και χρονοβόρα. Υπάρχει ανάγκη για σύστημα σχεδιασμού που αναγνωρίζει αυτόματα, ταξινομεί και ανιχνεύει ποσοτικά τα συμπτώματα των ασθενειών των φυτών. (Chopade, Bhagyashri, 2016)

Το πεδίο της ψηφιακής επεξεργασίας εικόνας αναφέρεται στην επεξεργασία ψηφιακών εικόνων μέσω ψηφιακού υπολογιστή. Τα τελευταία χρόνια, διάφοροι ερευνητές στρέφονται προς την ανάπτυξη αποτελεσματικών, αποδοτικών και ισχυρών υπολογιστικών τεχνικών / μεθόδων για την ανάλυση της επεξεργασίας εικόνας. Η όραση είναι η πιο προηγμένη από τις αισθήσεις μας, οπότε δεν προκαλεί έκπληξη το γεγονός ότι οι εικόνες παίζουν τον σημαντικότερο ρόλο στην ανθρώπινη αντίληψη. (Maheswari, 2018)

Η αναγνώριση των ασθενειών των φυτών που γίνεται με γυμνό μάτι συγκριτικά είναι αρκετά αναποτελεσματική, ανακριβής και δύσκολη. Η αυτόματη ανίχνευση των ασθενειών των φυτών είναι πολύ σημαντική για το πεδίο της έρευνας, καθώς μπορεί να αναδείξει τα οφέλη στην παρακολούθηση μεγάλων καλλιεργήσιμων εκτάσεων και έτσι να ανιχνεύσει αυτόματα τα συμπτώματα των ασθενειών όπως εμφανίζονται στα φύλλα των φυτών. Συνεπώς, η αναζήτηση γρήγορης, αυτόματης, χαμηλού κόστους και μεγάλης ακρίβειας μεθόδου για την ανίχνευση αρχομένων ασθενειών των φυτών έχει μεγάλη πρακτική σημασία. Η ανίχνευση και η αναγνώριση ασθενειών των φυτών με μηχανική μάθηση μπορεί να παρέχει εκτεταμένες ενδείξεις για τον εντοπισμό και τη θεραπεία των ασθενειών σε πολύ πρώιμα στάδια. Χωρίς σωστή διάγνωση ασθενειών των φυτών, οι πραγματικές ενέργειες ελέγχου δεν μπορούν να εφαρμοστούν στον κατάλληλο χρόνο. Η επεξεργασία εικόνας

είναι μία από τις ευρέως χρησιμοποιούμενες τεχνικές για την ανίχνευση και ταξινόμηση των ασθενειών των φυτών. (Chaitra, Faiza, Harshitha, Meghana, Rachitha, 2018)

#### **2.4.2 Τεχνικές επεξεργασίας εικόνας**

Η επεξεργασία εικόνας είναι πάντα ένα ενδιαφέρον πεδίο δεδομένου ότι παρέχει ενισχυμένα οπτικά δεδομένα για την ανθρώπινη αντίληψη και την επεξεργασία των δεδομένων εικόνας για μετάδοση και απεικόνιση από την οπτική της μηχανής. Οι ψηφιακές εικόνες τυγχάνουν επεξεργασίας για να δώσουν καλύτερη λύση χρησιμοποιώντας την επεξεργασία εικόνων. Χρησιμοποιούνται τεχνικές όπως η μετατροπή κλίμακας γκρι, η κατάτμηση εικόνων, η ανίχνευση ακμών, η εξαγωγή χαρακτηριστικών, η ταξινόμηση στην επεξεργασία εικόνων. (Sahil, 2018)

## **2.5 Τεχνητή νοημοσύνη**

### **2.5.1 Εισαγωγή**

Η Τεχνητή Νοημοσύνη - TN (Artificial Intelligence -AI) αν και συμπλήρωσε μισό αιώνα ζωής εξακολουθεί να είναι μια από τις πιο μοντέρνες ερευνητικές περιοχές. Τυπικά ξεκίνησε το 1956 στη συνάντηση μερικών επιφανών επιστημόνων, όπως ο John McCarty, Marvin Minsky, Claude Shannon και άλλοι, αν και η έρευνα είχε ξεκινήσει 5 χρόνια πριν. Από την άλλη, η μελέτη της νοημοσύνης είναι ένα από τα πιο παλιά θέματα της ανθρώπινης αναζήτησης. Για περισσότερο από 2000 χρόνια, οι φιλόσοφοι προσπάθησαν να ερμηνεύσουν το μηχανισμό της όρασης, της μάθησης, της απομνημόνευσης, της αντίληψης και του συλλογισμού, όπως ο Αριστοτέλης, ο Ηράκλειτος, ο Descartes κλπ. (Δημόπουλος, 2012)

Η ΤΝ περικλείει ένα πλήθος ερευνητικών πεδίων, από γενικού σκοπού όπως η αντίληψη και η συλλογιστική έως πιο συγκεκριμένων, όπως το παίξιμο σκακιού, η απόδειξη θεωρημάτων, η διάγνωση ασθενειών, κλπ. Συχνά ερευνητές από άλλες επιστημονικές περιοχές καταφεύγουν στην ΤΝ με σκοπό να βρουν εργαλεία για να αυτοματοποιήσουν τα λογικά βήματα που χρησιμοποιούν στην εργασία τους. Όμοια, ερευνητές της ΤΝ εφαρμόζουν τις μεθόδους τους σε διάφορες περιοχές όπου απαιτείται ανθρώπινη ευφυή προσπάθεια.

(ο. π.)

Αντικείμενο της Τεχνητής Νοημοσύνης αποτελεί η προσπάθεια μεταφύτευσης των ικανοτήτων της αντίληψης, του συλλογισμού και της κατανόησης στους ηλεκτρονικούς υπολογιστές. Η διατύπωση αυτή συμφωνεί απόλυτα με τον κλασσικό πλέον ορισμό που δίνει η Elaine Rich (1983), σύμφωνα με τον οποίο: “Τεχνητή Νοημοσύνη είναι η μελέτη του πώς θα καταστήσουμε τους υπολογιστές ικανούς να κάνουν πράγματα, στα οποία οι άνθρωποι, προς το παρόν, τα καταφέρνουν καλύτερα”.

(Κρικέτος, Πάστρας, 1991)

Η ΤΝ χωρίζεται σε δύο κατηγορίες, τη Συμβολική και τη Μη-Συμβολική ΤΝ. Η *Συμβολική Τεχνητή Νοημοσύνη (symbolic AI-Artificial Intelligence)* προσομοιώνει τον τρόπο σκέψης του ανθρώπου, χρησιμοποιώντας ως δομικές μονάδες τα σύμβολα. Ένα σύμβολο μπορεί να αναπαριστά μία έννοια ή μία σχέση ανάμεσα σε έννοιες. Παραδείγματα αυτής της κατηγορίας είναι οι εφαρμογές της ΤΝ που χρησιμοποιούν αναπαράσταση γνώσης με λογική, κανόνες, πλαίσια, κλπ. Η *Μη Συμβολική Τεχνητή Νοημοσύνη (non symbolic AI-Artificial Intelligence)* προσομοιώνει βιολογικές διεργασίες, όπως τη διαδικασία εξέλιξης των ειδών ή τη λειτουργία του εγκεφάλου. Παραδείγματα τέτοιων τεχνικών αποτελούν τα νευρωνικά δίκτυα και οι γενετικοί αλγόριθμοι (Βλαχάβας κ.α., 2002).

## 2.5.2 Ορισμός της Τεχνητής Νοημοσύνης

Η Τεχνητή Νοημοσύνη είναι το γενικό όνομα το οποίο δόθηκε στο πεδίο

της επιστήμης των υπολογιστών αφιερωμένο στην ανάπτυξη των προγραμμάτων τα οποία τείνουν να αντιγράψουν πιστά την ανθρώπινη νοημοσύνη (Fonseca and Navarrese, 2002). Κατά καιρούς έχουν διατυπωθεί διάφοροι ορισμοί της Τεχνητής Νοημοσύνης (ΤΝ), από τους οποίους άλλοι επικεντρώνονται στη διαδικασία σκέψης και συλλογισμού και άλλοι στη συμπεριφορά.

Ένας από τους πρώτους ορισμούς που διατυπώθηκαν από τους Barr και Feigenbaum αναφέρει ότι, *“ΤΝ είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφύων (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά.”* (Βλαχάβας κ.α., 2002)

Υπάρχουν διάφοροι ορισμοί σύμφωνα με τους οποίους στόχος της ΤΝ είναι να φτιάξει συστήματα που:

- Σκέφτονται όπως οι άνθρωποι:

Η προσπάθεια να κατασκευάσουμε υπολογιστές με διανοητική ικανότητα με την πλήρη και κυριολεκτική έννοια του όρου (Haugeland, 1989).

- Συμπεριφέρονται όπως οι άνθρωποι:

Η μελέτη του πώς να κάνουμε τους υπολογιστές να κάνουν πράγματα στα οποία αυτήν τη στιγμή οι άνθρωποι είναι καλύτεροι (Rich and Knight, 1991).

- Σκέφτονται λογικά:

Η μελέτη των υπολογισμών που καθιστούν εφικτή την αντίληψη, τη λογική σκέψη και την αντίδραση (Winston, 1992).

- Αντιδρούν λογικά:

Ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την αυτοματοποίηση της ευφύους συμπεριφοράς (Luger and Stubblefield, 1993).

Ένας γενικός ορισμός που περιλαμβάνει τα περισσότερα στοιχεία από τους ανωτέρω θα μπορούσε να είναι ο εξής:

*“Τεχνητή Νοημοσύνη είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά,, όπως για*

*παράδειγμα η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κλπ. ”*

Άμεση συνέπεια των παραπάνω ορισμών είναι η αποδοχή ότι η ΤΝ είναι ένας συνεχώς εξελισσόμενος τομέας την Επιστήμης των Υπολογιστών που προσπαθεί να κάνει πραγματικότητα, ότι αυτή η επιστήμη δεν έχει καταφέρει μέχρι στιγμής. Κι αυτό είναι μία μεγάλη αλήθεια, αν σκεφτεί κανείς ότι η τεχνολογία του σήμερα δεν έχει προσφέρει στην υπηρεσία του ανθρώπου, τίποτε άλλο εκτός από μηχανές οι οποίες αποθηκεύουν τεράστιες ποσότητες πληροφορίας και έχουν την ικανότητα να τις προσπελάσουν σε ελάχιστο χρόνο. Ωστόσο, οι μηχανές αυτές απέχουν ακόμη από το να μην απαιτούν ειδικές γνώσεις για το χειρισμό τους, να προσαρμόζονται στις ανάγκες του χρήστη, να μαθαίνουν από τα λάθη τους και να επιλύουν πραγματικά, δύσκολα, καθημερινά προβλήματα και όχι μόνον αριθμητικά. (Δημόπουλος, 2012)

Οι μεθοδολογίες που αναπτύχθηκαν για την ΤΝ έχουν αποδώσει καρπούς σε πολλές από τις επιμέρους περιοχές αυτής της επιστήμης, όπως:

- Απόδειξη θεωρημάτων
- Επεξεργασία φυσικής γλώσσας
- Τεχνητή όραση
- Μηχανική μάθηση
- Σχεδιασμός ενεργειών και χρονοπρογραμματισμός
- Αυτόνομα robot
- Έμπειρα Συστήματα και συστήματα γνώσης

Υπάρχουν δύο προσεγγίσεις για την ΤΝ. Η *κλασική ή συμβολική* προσέγγιση, που βασίζεται στην κατανόηση των νοητικών διεργασιών και ασχολείται με τη προσομοίωση της ανθρώπινης νοημοσύνης προσεγγίζοντας την με αλγορίθμους και συστήματα που βασίζονται στη γνώση και η *συνδετική* (connectionist approach) ή *μη-συμβολική* προσέγγιση που βασίζεται στη μίμηση της βιολογικής λειτουργίας του εγκεφάλου προσεγγίζοντας το θέμα με τα λεγόμενα νευρομορφικά ή νευρωνικά δίκτυα. (ο. π.)

### 2.5.3 Διαχωρισμός της τεχνητής από τη φυσική Νοημοσύνη

Η εφαρμογή της Τεχνητής Νοημοσύνης σε μηχανές καθιστούν τις μηχανές αυτές ευφυείς και τις κάνουν να συμπεριφέρονται σχεδόν όπως ο άνθρωπος. Ο διαχωρισμός των μηχανών αυτών ως ευφυείς ή όχι καθορίζεται με τη βοήθεια του κριτηρίου που έθεσε ο Turing (δοκιμασία Turing). (Αποστόλου, 2004).

Ο Alan Turing (1912-1953), ο οποίος θεωρείται ο πατέρας της ΤΝ, εμπνεύστηκε το 1950 μία δοκιμασία η οποία πήρε και το όνομα του (Turing Test - Δοκιμασία Turing), για το χαρακτηρισμό των μηχανών. Αυτό βασίζεται σε μία σειρά από ερωτήσεις που υποβάλει κάποιος ταυτόχρονα σε έναν άνθρωπο και μία μηχανή χωρίς να γνωρίζει εκ των προτέρων ποιος είναι τι. Αν στο τέλος δεν καταφέρει να ξεχωρίσει τον άνθρωπο από τη μηχανή, τότε η μηχανή πετυχαίνει στη δοκιμασία και θεωρείται ευφυής. Αν και η αποτελεσματικότητα της Δοκιμασίας Turing εξαρτάται από πολλούς παράγοντες, θεωρείται μέχρι σήμερα ένα καλό μέτρο σύγκρισης της φυσικής με την τεχνητή νοημοσύνη και πολλοί διαγωνισμοί διοργανώνονται σε ετήσια βάση, χωρίς όμως ιδιαίτερα σοβαρά ή τουλάχιστο χρήσιμα αποτελέσματα. (ο. π.).

Ο προγραμματισμός ενός υπολογιστή για να περάσει τη δοκιμασία Turing απαιτεί τη συμμετοχή αρκετών επιστημονικών περιοχών, όπως της επεξεργασίας φυσικής γλώσσας (natural language processing) για επικοινωνία σε φυσική γλώσσα, της αναπαράστασης γνώσης (knowledge representation) για την αποθήκευση της γνώσης πριν και κατά τη διάρκεια της δοκιμασίας, της αυτοματοποιημένης συλλογιστικής (automated reasoning) για τη χρήση της αποθηκευμένης πληροφορίας και την εξαγωγή συμπερασμάτων, της μηχανικής μάθησης (machine learning) για προσαρμογή σε νέες περιπτώσεις, κλπ. (ο. π.).

Στην αρχική της μορφή, η δοκιμασία Turing δεν προέβλεπε φυσική επαφή ανθρώπου μηχανής. Ωστόσο μια επέκταση της (Πλήρης Δοκιμασία Turing) περιλαμβάνει και την αναγνώριση εικόνων και αντικειμένων που ανταλλάσσονται μέσα από κάποια θυρίδα ώστε να μην υπάρχει οπτική επαφή με το δοκιμαζόμενο για τον έλεγχο των δυνατοτήτων αντίληψής του. Για το σκοπό αυτό απαιτείται η συμμετοχή και άλλων δύο επιστημονικών περιοχών, της μηχανικής όρασης (machine vision) για την αναγνώριση και της ρομποτικής (robotics) για τη μετακίνησή τους

(ο. π.).

## 2.5.4 Τομείς της Τεχνητής Νοημοσύνης

Η Τεχνητή Νοημοσύνη εφαρμόζεται σε όλο και περισσότερους τομείς που σχετίζονται με την τεχνολογία και τις διάφορες επιστήμες. Εφαρμόζεται σε συσκευές καθημερινής χρήσης, την βιομηχανία την άμυνα ακόμη και τα ηλεκτρονικά παιχνίδια. Κατά συνέπεια αναπτύχθηκαν πολλοί επιμέρους κλάδοι της επιστήμης αυτής (Γιαλούρης, 2011).

Μερικοί από τους κλάδους της Τεχνητής Νοημοσύνης είναι:

**Η Επεξεργασία Φυσικής Γλώσσας**, που σκοπός της είναι η επικοινωνία του ανθρώπου και του υπολογιστή σε φυσική γλώσσα. Η επικοινωνία αυτή περιλαμβάνει την κατανόηση από πλευράς υπολογιστή προτάσεων σε φυσική γλώσσα προκειμένου να εκτελέσει κάποιες λειτουργίες όπως για παράδειγμα φωνητική προσπέλαση σε Βάσεις Δεδομένων. Ο τομέας της επεξεργασίας φυσικής γλώσσας περιλαμβάνει ως εφαρμογή και την αυτόματη μετάφραση μεταξύ δύο φυσικών γλωσσών (π.χ. Αγγλικά - Ελληνικά). (ο. π.)

**Τα Έμπειρα Συστήματα**, που αποσκοπούν στην αποθήκευση στον υπολογιστή της εμπειρίας ενός ειδικού σε κάποιο τομέα και στην εξασφάλιση της δυνατότητας του υπολογιστή να δρα πλέον ως ειδικός στον τομέα αυτό. Τα Έμπειρα Συστήματα είναι από τα πλέον χαρακτηριστικά παραδείγματα εφαρμογών της Τεχνητή Νοημοσύνης. (ο. π.)

**Η Αναγνώριση Εικόνας**, η οποία έχει στόχο να δίδει στον υπολογιστή τη δυνατότητα, μέσω καταλλήλων οπτικών συστημάτων, να βλέπει και να αναγνωρίζει αντικείμενα. Η αναγνώριση εικόνας χρησιμοποιείται σε ρομποτικά συστήματα, σε συστήματα ασφαλείας, σε διαγνωστικά συστήματα κλπ. (ο. π.)

**Η Μηχανική Μάθηση ή Μάθηση της Μηχανής**, που έχει σκοπό να δώσει στον υπολογιστή την δυνατότητα να αυξάνει την απόδοση, αυτό βελτιώνοντας τη γνώση του σε ένα ορισμένο τομέα χωρίς να επεμβαίνει ο άνθρωπος. Σε ένα υπολογιστικό σύστημα το οποίο έχει την ι-



κανότητα να μαθαίνει η γνώση του βρίσκεται σε διαρκή μεταβολή, όπως ακριβώς συμβαίνει και στον άνθρωπο. Η μεταβολή της γνώσης μπορεί να γίνει είτε με μετασχηματισμό της νέας γνώσης και αποθήκευσή της σε μία δομή κατάλληλα επεξεργάσιμη από το σύστημα είτε με τον αυτό μετασχηματισμό του ιδίου του συστήματος όπως αυτό συμβαίνει στα Νευρωνικά Δίκτυα. (ο. π.)

Τα **Νευρωνικά Δίκτυα**, μέσω των οποίων γίνεται προσομοίωση της λειτουργίας του ανθρώπινου εγκεφάλου σύμφωνα με τις θεωρίες της σύγχρονης Ψυχιατρικής, Ψυχολογίας και Νευροφυσιολογίας. Τα Νευρωνικά Δίκτυα χρησιμοποιούν τεχνικές και μεθοδολογίες από διάφορους επιστημονικούς χώρους όπως Μαθηματικά, Στατιστική, Φυσική κλπ. Και εξαπλώνονται σε πολλούς τομείς. (ο. π.)

Τα Νευρωνικά Δίκτυα απαρτίζονται από τεχνητούς νευρώνες που αλληλεπιδρούν μέσω συνδέσμων που ονομάζονται συντελεστές βάρους. Θετικά ή αρνητικά βάρη αντιστοιχούν σε συνάψεις που μεταδίδουν προς άλλους νευρώνες ή αναστέλλουν ερεθίσματα από άλλους νευρώνες. (ο. π.)

Η **Ρομποτική**, η οποία εξασφαλίζει τη συνεργασία διαφόρων κλάδων της Τεχνητής Νοημοσύνης, σε συνδυασμό με ηλεκτρομηχανικές διατάξεις, για την εκτέλεση διαφόρων εξειδικευμένων εργασιών. Σύγχρονα ρομπότ χρησιμοποιούνται σε αλυσίδες παραγωγής αυτοκινήτων ή εκτελούν εργασίες σε χώρους επικίνδυνους για τον άνθρωπο. Γενικά ένα ρομπότ αποτελείται από τρεις βασικές συνιστώσες: τους αισθητήρες, τις μονάδες επίδρασης στο περιβάλλον του ρομπότ και την μονάδα ελέγχου. (ο. π.)

Οι **Γενετικοί Αλγόριθμοι** είναι προσανατολισμένοι στην αναζήτηση της βέλτιστης λύσης μέσα από ένα σύνολο αρχικών πιθανών λύσεων ενός προβλήματος. Κάθε τέτοια πιθανή λύση αναπαρίσταται με ένα μία σειρά χαρακτήρων (string) αποτελούμενη από bits ή οποία ονομάζεται και χρωμόσωμα (chromosome) ενώ κάθε χαρακτήρας της σειράς ονομάζεται γονίδιο (gene). Η επιλογή βασίζεται στην βαθμολόγηση κάθε λύσης με την χρήση μιας συνάρτησης καταλληλότητας (fitness function).

Η Συνάρτηση αυτή έχει ως όρισμα ένα χρωμόσωμα και δίδει ως αποτέλεσμα τιμή που δηλώνει την καταλληλότητα του αντιστοίχου χρωμοσώματος. Η διαδικασία ολοκληρώνεται με την μεγιστοποίηση της συνάρτησης. (ο. π.)

**Οι Ευφυείς Πράκτορες (Intelligent Agents)** αποτελούν έναν από τους νεότερους κλάδους της Τεχνητής Νοημοσύνης. Ένας πράκτορας είναι μία οντότητα που έχει αντίληψη του περιβάλλοντος στο οποίο ευρίσκεται και αντιλαμβάνεται αυτό μέσω αισθητήρων ενώ επενεργεί πάνω σε αυτό με ειδικούς μηχανισμούς δράσης (actuators). Έχει τη δυνατότητα να πραγματοποιεί συλλογισμούς και να επιδρά πάνω στο περιβάλλον. Μπορούμε να διακρίνουμε τους λογισμικούς πράκτορες οι οποίοι λειτουργούν σε ένα υπολογιστικό περιβάλλον και τους ρομποτικούς πράκτορες που λειτουργούν σε πραγματικό περιβάλλον. Μία ειδική κατηγορία πρακτόρων είναι αυτή των κινητών πρακτόρων (mobile agents) που χρησιμοποιούνται σε εφαρμογές διαδικτύου. (ο. π.)

## 2.6 Έμπειρα συστήματα

### 2.6.1 Ορισμός

Έχουν δοθεί διάφοροι ορισμοί για τα Έμπειρα Συστήματα. Ένα Έμπειρο Σύστημα (expert systems) θα μπορούσε να οριστεί ως:

- Ένα υπολογιστικό σύστημα ικανό να δώσει συμβουλές σε ένα συγκεκριμένο πεδίο γνώσης εξαιτίας του γεγονότος ότι περιέχει τη γνώση ενός ειδικού στο πεδίο αυτό (Αργιαλάς, 2001).

- Ένα Έμπειρο Σύστημα είναι ένα έξυπνο πρόγραμμα που χρησιμοποιεί γνώση και μηχανισμούς εξαγωγής συμπερασμάτων για να μπορεί να λύνει προβλήματα τα οποία είναι δύσκολα και μπορεί να απαιτούν σημαντική ανθρώπινη εμπειρία για τη λύση τους” (Feigenbaum, 1978).

Όπως παρατηρούμε για ένα Έμπειρο Σύστημα σημαντικότερες είναι οι έννοιες του προβλήματος και της γνώσης. Ένα Έμπειρο Σύστημα σχεδιάζεται για να αντιμετωπίσει ένα αυστηρά καθορισμένο πρόβλημα με βάση τη γνώση που παρέχεται από τους ειδικούς. (Δημόπουλος, 2012)

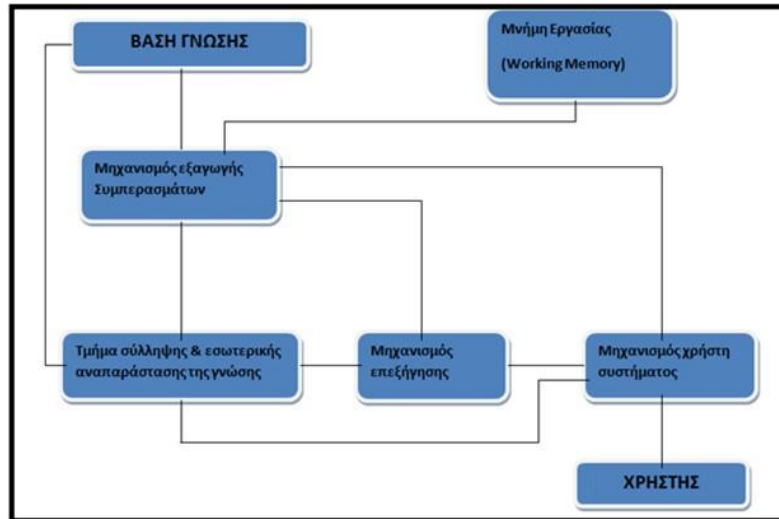
Ένα Έμπειρο Σύστημα αποτελείται από τη διεπαφή με τον χρήστη, την μηχανή εξαγωγής συμπερασμάτων, τη Βάση Γνώσης η οποία περιέχει την γνώση του ειδικού και ένα υποσύστημα παροχής επεξηγήσεων. (ο. π.)

Για την ανάπτυξη ενός Έμπειρου Συστήματος απαιτείται αρχικά η εξαγωγή της γνώσης από τις διάφορες πηγές (ειδικοί, εγχειρίδια, κτλ.), κατόπιν η αναπαράσταση της γνώσης (σχήμα) και τέλος η υλοποίηση του Έμπειρου Συστήματος. (ο. π.)

Για την υλοποίηση ενός Έμπειρου Συστήματος μπορούν να χρησιμοποιηθούν είτε εξειδικευμένες γλώσσες προγραμματισμού όπως η CLIPS είτε τα λεγόμενα κελύφη ή φλοιοί Έμπειρων Συστημάτων. Μια γλώσσα προγραμματισμού προσφέρει μεγαλύτερη ελευθερία στην ανάπτυξη ενός Έμπειρου συστήματος αλλά μέσω της χρήσης ενός κελύφους Έμπειρων Συστημάτων εξοικονομούνται χρόνος και χρήμα. (ο. π.)

### **2.6.2 Δομή ενός Έμπειρου Συστήματος**

Ένα Έμπειρο Σύστημα αποτελείται από τη Βάση γνώσης, τη Μνήμη Εργασίας, το Μηχανισμό Εξαγωγής Συμπερασμάτων, το Τμήμα Σύλληψης και Εσωτερικής Αναπαράστασης της Γνώσης, το Μηχανισμός Επεξήγησης και το Μηχανισμό Χρήστη- Συστήματος (Σχήμα 1).



Σχήμα 1. Αρχιτεκτονική ενός Έμπειρου Συστήματος

Η **Βάση Γνώσης** (knowledge base) περιέχει όλη τη γνώση του συστήματος, όπως την κατέγραψε ο μηχανικός γνώσης με τη βοήθεια του ανθρώπου – ειδικού κατά τη διαδικασία ανάπτυξης του συστήματος γνώσης και αποτελείται από δύο μέρη: (Γιαλούρης, 2011)

Τη στατική η οποία περιέχει διαδικασίες, κανόνες, πλαίσια που περιγράφουν το πρόβλημα και τις γνωσιολογικές διαδικασίες επίλυσής τους (αρχικά δεδομένα). Ο όρος στατική υποδηλώνει ότι αυτό το τμήμα γνώσης δε μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. (ο. π.)

Τη δυναμική η οποία περιέχει τα δεδομένα και πλαίσια που περιγράφουν τη λύση του προβλήματος. (ο. π.)

Η **Μνήμη Εργασίας** είναι το τμήμα του Εμπειρου Συστήματος στο οποίο αποθηκεύονται τα γεγονότα και τα ενδιάμεσα ή τελικά συμπεράσματα. Από τη μνήμη εργασίας ο συμπερασματικός μηχανισμός θα χρησιμοποιήσει τα υπάρχοντα γεγονότα για να αποφασίσει αν μια συνθήκη είναι αληθής. Επίσης θα καταχωρήσει τις απαντήσεις του χρήστη στις ερωτήσεις που το Έμπειρο Σύστημα τον ερωτά. Θα καταχωρίσει τα αποτελέσματα ή το συμπέρασμα που προκύπτει από κάποιον κανόνα και τέλος θα αποθηκεύσει και τα τελικά συμπεράσματα. Στη Μνήμη Εργασίας τηρούνται και άλλα στοιχεία, ανάλογα με το σύστημα, όπως για παράδειγμα η χρονική σειρά με την οποία έχουν εισαχθεί τα γεγονότα. (ο. π.)

Ο **Μηχανισμός Εξαγωγής Συμπερασμάτων** (inference engine) είναι το τμήμα του πυρήνα που είναι υπεύθυνο για το χειρισμό της υπάρχουσας γνώσης και την εξαγωγή συμπερασμάτων από αυτήν. Πρόκειται για έναν μηχανισμό ελέγχου που εφαρμόζει την αξιωματική γνώση στη βάση γνώσης και τα δεδομένα του υπό εξέταση προβλήματος για να φτάσει σε κάποια λύση ή συμπέρασμα. Η μηχανή συμπερασμού εξάγει συμπεράσματα αποφασίζοντας για ποιους από τους κανόνες, τα πρότυπα στο αριστερό τους μέρος ικανοποιούνται από γεγονότα ή από αντικείμενα στη μνήμη εργασίας, καθορίζοντας προτεραιότητες για τους κανόνες αυτούς με βάση τη στρατηγική επίλυσης διενέξεων («conflict resolution strategy») και εκτελώντας τον κανόνα με τη μεγαλύτερη προτεραιότητα (η εκτέλεση του κανόνα καλείται και πυροδότηση, «firing»). (ο. π.)

Ο **Μηχανισμός Επεξήγησης** αποτελεί πλέον αναπόσπαστο κομμάτι ενός ολοκληρωμένου Έμπειρου Συστήματος. Ο μηχανισμός επεξήγησης πρέπει να απαντά σε δύο τουλάχιστον τύπους ερωτήσεων:

- Πώς (how) κατέληξε σε ένα συμπέρασμα. Σε αλληλεπίδραση με το μηχανισμό εξαγωγής συμπερασμάτων, ο μηχανισμός επεξήγησης κρατάει πληροφορίες σχετικά με την αποδεικτική διαδικασία. Ως απάντηση στην ερώτηση παραθέτει τους κανόνες που ενεργοποιήθηκαν σε κάθε κύκλο λειτουργίας και οδήγησαν στη δεδομένη απάντηση.

- Γιατί (why) ζητά κάποια πληροφορία από το χρήστη. Στην πορεία των συλλογισμών για την κατάληξη σε ένα συμπέρασμα, ο μηχανισμός εξαγωγής είναι δυνατό να ζητήσει πληροφορίες από το χρήστη που θα καθορίσουν τη συνέχεια της εκτέλεσης των κανόνων. Για να απαντήσει στο ερώτημα που αφορά το σκοπό των ερωτήσεων, ο μηχανισμός επεξήγησης αναζητά όλους τους κανόνες που έχουν στην υπόθεσή τους τη δεδομένη πληροφορία, και επιστρέφει όλη την αλυσίδα των συλλογισμών που θα εκτελεστεί ως συνέπεια της ενεργοποίησης κανόνων με βάση την απάντηση του χρήστη. (ο. π.)

Το **Τμήμα Σύλληψης και Εσωτερικής Αναπαράστασης της Γνώσης** είναι υπεύθυνο για την εσωτερική αναπαράσταση της γνώσης μέσα

στη Βάση Γνώσης. Η Βάση Γνώσης γράφεται, τις περισσότερες φορές σε μία μορφή κειμένου ή ακόμη και με κάποιο τυποποιημένο τρόπο που έχει σχεδιάσει ο κατασκευαστής. Το τμήμα αυτό ελέγχει την ορθότητα της Βάσης Γνώσης, την μετατρέπει σε μία εσωτερική μορφή κατανοητή για το σύστημα, αλλά ταυτόχρονα η εσωτερική αυτή μορφή έχει τη δυνατότητα να μετατραπεί σε αυτή που είχε αρχικά δοθεί. (ο. π.)

Ο **Ενδιάμεσος Μηχανισμός Χρηστή-Συστήματος** είναι υπεύθυνος για τη διαδικασία επικοινωνίας χρήστη-μηχανής, επικοινωνεί με την παραγωγική μηχανή, το τμήμα επεξήγησης και το τμήμα σύλληψης και αναπαράστασης της γνώσης (ο. π.)

### 2.6.3 Ανάπτυξη Έμπειρων Συστημάτων

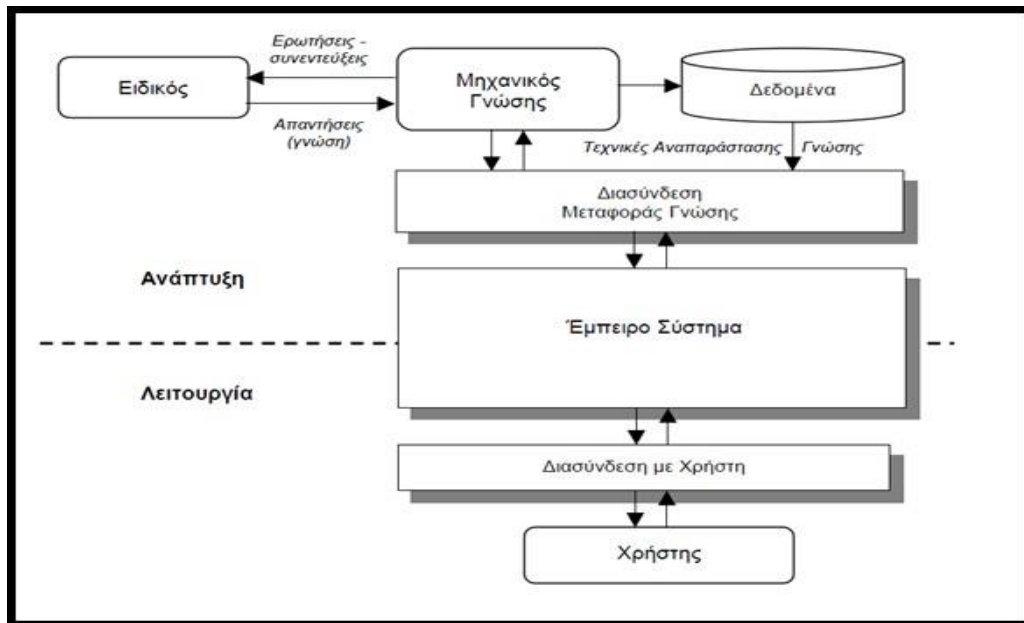
Η ανάπτυξη ενός Έμπειρου Συστήματος απαιτεί τη συνεργασία και τη συνεισφορά τριών ομάδων ή μεμονωμένων ανθρώπων: τους **Μηχανικούς Γνώσης** (Knowledge Engineers), τους **Ειδικούς του Τομέα** (Human Domain Experts), και τους **Χρήστες** (Users) (Σχήμα 2). (Χαντζάρα, 2008)

**Κατηγορίες ειδικών στην ανάπτυξη ενός Έμπειρου Συστήματος**  
Ο **Ειδικός του Τομέα** είναι ένας άνθρωπος με εξειδικευμένη γνώση, εμπειρία ή ικανότητα στον τομέα που αφορά το συγκεκριμένο σύστημα, και του οποίου η γνώση θα αποτελέσει τη Βάση Γνώσης του Έμπειρου Συστήματος. Η επιλογή του σωστού ανθρώπου ειδικού έχει καθοριστική σημασία για την επιτυχία του συστήματος. Βασικό προσόν ενός ειδικού για να χαρακτηριστεί ως κατάλληλος είναι η ποιότητα του γνωστικού του υπόβαθρου και η ικανή εμπειρία του πάνω στο αντικείμενο. Αν και όχι απαραίτητο, είναι συνήθως επιθυμητό ο ειδικός να έχει κάποια εμπειρία και γνώση σχετικά με τις αρχές και τη λειτουργία των Έμπειρων Συστημάτων. (ο. π.)

Ο **Μηχανικός Γνώσης** είναι ένας επιστήμονας της πληροφορικής, ειδικευμένος σε θέματα Τεχνητής Νοημοσύνης και Έμπειρων Συστημά-

των. Είναι ο υπεύθυνος για τη δόμηση και κατασκευή του Έμπειρου Συστήματος. Έχουν διατυπωθεί πολλές διαφορετικές απόψεις για τα προσόντα και τις ευθύνες ενός μηχανικού γνώσης, μεταξύ των οποίων είναι η διαχείριση του έργου ανάπτυξης, ο ορισμός και ανάλυση του προβλήματος, η επιλογή του υλικού και του λογισμικού για την ανάπτυξη του συστήματος, η εκμαίευση της γνώσης από τους ειδικούς και η αναπαράστασή της σε μορφή συμβατή με την τεχνολογία Έμπειρου Συστήματος που έχει επιλεγθεί, η αλληλεπίδραση με τους χρήστες, η επαλήθευση και ο έλεγχος της αξιοπιστίας του συστήματος, η εκπαίδευση των χρηστών, η συντήρηση του συστήματος, και τέλος η παροχή συμβουλών για μελλοντικές επεκτάσεις και τροποποιήσεις του συστήματος. Επιθυμητά χαρακτηριστικά του μηχανικού της γνώσης, είναι η εξοικείωση με τις αρχές των Έμπειρων Συστημάτων και την εμπειρία στην ανάπτυξη τέτοιων συστημάτων, η ικανότητα εφαρμογής τεχνικών εκμαίευσης της γνώσης και κάποιο θεωρητικό υπόβαθρο στον τομέα που αφορά το συγκεκριμένο σύστημα. (ο. π.)

Οι **Χρήστες** είναι οι άνθρωποι που θα χρησιμοποιήσουν το Έμπειρο Σύστημα για να επωφεληθούν από τα αποτελέσματα της εμπειρογνώμοσύνης του. Η επικρατούσα άποψη υποστηρίζει ότι οι χρήστες πρέπει να συμμετέχουν όχι μόνο στη διαδικασία του ελέγχου της ορθότητας της λειτουργίας του συστήματος, αλλά και στο αρχικό στάδιο απόκτησης της γνώσης, ιδιαίτερα στο στάδιο της ανάλυσης του προβλήματος, καθώς διαθέτουν μία ευρύτερη αντίληψη για το πώς το σύστημα μπορεί να τους βοηθήσει. Έχει αποδειχθεί ότι τα συστήματα που έχουν περιλάβει χρήστες στη διαδικασία ανάπτυξης τείνουν να είναι πιο αποδεκτά από τους χρήστες κατά την ολοκλήρωση και παράδοσή τους (Βλαχάβας κ.α., 2002), (Χαντζάρα, 2008)



Σχήμα 2. Ανάπτυξη Έμπειρων Συστημάτων

### Διαδικασία ανάπτυξης ενός Έμπειρου Συστήματος

Εφόσον καθοριστούν οι άνθρωποι που θα συμμετέχουν στην ανάπτυξη του Έμπειρου Συστήματος, ξεκινά ο κύκλος υλοποίησης του συστήματος. (Χαντζάρα, 2008)

### Ανάλυση του προβλήματος

Στο στάδιο της ανάλυσης του προβλήματος προσδιορίζεται η μορφή της επιθυμητής λύσης του προβλήματος. Ένα από τα κυριότερα ζητήματα που πρέπει να απασχολήσουν τους μηχανικούς και τους ειδικούς στο στάδιο της ανάλυσης είναι αν το πρόβλημα είναι κατάλληλο για επίλυση από ένα έμπειρο σύστημα ή ένα συμβατικό πρόγραμμα. Παρόλο που ένα Έμπειρο Σύστημα παρουσιάζει πολλά πλεονεκτήματα, έχει και αρκετούς περιορισμούς στη χρήση του, που δεν το καθιστά πάντα την καλύτερη λύση. Ακόμα, πρέπει να εκτιμηθεί το κόστος της ανάπτυξης ενός Έμπειρου Συστήματος, συμπεριλαμβανομένου του ανθρώπινου δυναμικού, του χρόνου, και των χρημάτων που πρέπει να αφιερωθούν, και κατά πόσο η ανταμοιβή από τη λειτουργία ενός τέτοιου συστήματος, η οποία



μπορεί να μην έρθει με τη μορφή χρημάτων αλλά ίσως αύξησης της αποδοτικότητας, θα καλύψει το κόστος της επένδυσης. (ο. π.)

### ***Απόκτηση της γνώσης***

Απόκτηση Γνώσης είναι η διαδικασία της εξαγωγής, δόμησης και οργάνωσης της γνώσης από πολλαπλές πηγές γνώσεων, συνήθως ανθρώπους ειδικούς, έτσι ώστε η εμπειρογνωμοσύνη που είναι απαραίτητη και ικανή για τη λύση του προβλήματος να μετασχηματιστεί σε μία μορφή αναγνώσιμη από το υπολογιστικό σύστημα. Η γνώση ίσως αποτελεί το πιο βασικό δομικό στοιχείο ενός Έμπειρου Συστήματος, καθώς αποτελεί τη βάση της συλλογιστικής διαδικασίας του συστήματος. (ο. π.)

Το πρώτο στάδιο της απόκτησης της γνώσης είναι η **εκμαίευση της γνώσης (knowledge elicitation)** από τον άνθρωπο ειδικό, μέσω ενός κύκλου συνόδων μεταξύ του μηχανικού γνώσης και του ειδικού του τομέα, σε απόλυτο κλίμα συνεργασίας. (ο. π.)

Ο πιο διαδεδομένος και αποδοτικός τρόπος εκμαίευσης γνώσης είναι η διαδικασία των συνεντεύξεων, με τη μορφή των μη-δομημένων, ημι-δομημένων ή δομημένων συνεντεύξεων. Οι μη-δομημένες ή ελεύθερες συνεντεύξεις αποτελούνται από γενικές ερωτήσεις που υποβάλλονται από το μηχανικό με την ελπίδα της καταγραφής όσο περισσότερων πληροφοριών γίνεται. Ξεκινώντας με πολύ γενικές ερωτήσεις, κατά την πορεία της η συζήτηση εξειδικεύεται, εφόσον ζητείται από τον ειδικό να επεξηγήσει ή να διευκρινίσει κάποια συγκεκριμένα σημεία. Οι ημι-δομημένες συνεντεύξεις περιλαμβάνουν μια σειρά ανοιχτών ερωτήσεων και θεμάτων που πρέπει να καλυφθούν. Οι δομημένες συνεντεύξεις στηρίζονται στη χρήση ενός ερωτηματολογίου με αυστηρά καθορισμένη δομή που περιλαμβάνει συγκεκριμένες ερωτήσεις σχετικές με τα χαρακτηριστικά του προβλήματος. (ο. π.)

Στις κλασικές μεθόδους εκμαίευσης της γνώσης αξίζει να αναφέρουμε τη χρήση των **πλεγμάτων ρεπερτορίων (repertory grids)**. Κατά την εφαρμογή της μεθόδου κάθε στοιχείο της περιοχής κατηγοριοποιείται σύμφωνα με ένα σύνολο από έννοιες ή χαρακτηρισμούς, οι οποίες

εφαρμόζονται σε όλα τα στοιχεία σε κάποιο βαθμό. Κάθε έννοια ή χαρακτηρισμός εκφράζεται σε μια γραμμική, αριθμητική κλίμακα, η οποία είναι η ίδια κάθε φορά, και η οποία χαρακτηρίζεται από δύο ακραίες τιμές π.χ. βαρύς/ελαφρύς, φτηνός/ακριβός, κ.α. Από τον ειδικό ζητείται να αποδώσει μια τιμή σε κάθε έννοια για όλα τα στοιχεία της περιοχής. Στο πλέγμα που δημιουργείται εξετάζεται αν κάποιο ζευγάρι εννοιών είναι παρόμοιο κατά τη σύγκριση των οριζοντίων γραμμών του πλέγματος, ώστε να παραλειφθούν κάποιες παραπλήσιες έννοιες. Τέλος, υπολογίζεται, σε ένα νέο πλέγμα, πόσο όμοια ή ανόμοια είναι τα στοιχεία της περιοχής μεταξύ τους. Η τεχνική είναι ιδιαίτερα χρήσιμη στην εκμαίευση υποκειμενικών δεδομένων. (ο. π.)

Εκτός από τις κλασικές μεθόδους εκμαίευσης της γνώσης, υπάρχουν οι ημιαυτόματες μέθοδοι κατά τις οποίες ο ειδικός εισάγει απευθείας τη γνώση στο σύστημα χρησιμοποιώντας ειδικό λογισμικό, πχ TEIRESIAS, OPAL κλπ, ή αυτόματες μέθοδοι όπου το σύστημα χρησιμοποιεί τεχνικές μηχανικής μάθησης. (ο. π.)

Το δεύτερο στάδιο απόκτησης της γνώσης είναι η **ανάλυση της γνώσης (knowledge analysis)** που εκμαιεύθηκε από τον άνθρωπο ειδικό με σκοπό τη δημιουργία ενός **μοντέλου της γνώσης (knowledge modeling)**. Ευριστικές διαδικασίες, έννοιες ή δομές ταξινόμησης αναλύονται και τυποποιούνται σε αναπαραστάσεις που μπορεί να είναι της μορφής των ευριστικών κανόνων, των πλαισίων, των αντικειμένων, των σημασιολογικών δικτύων ή των σχημάτων ταξινόμησης. Αυτές οι αναπαραστάσεις στη συνέχεια μετασχηματίζονται σε συγκεκριμένα σχήματα που υποστηρίζονται από το εργαλείο ανάπτυξης του Έμπειρου Συστήματος. Στο στάδιο αυτό είναι δυνατόν να χρησιμοποιηθούν μεθοδολογίες που τυποποιούν τη μοντελοποίηση της γνώσης, όπως η KADS. (ο. π.)

### **Σχεδίαση**

Κατά τη φάση της σχεδίασης προσδιορίζεται η μορφή με την οποία θα γίνει η αναπαράσταση της γνώσης, όπως κανόνες, πλαίσια κτλ. Στο στάδιο αυτό θα γίνει επιπλέον η επιλογή του εργαλείου ανάπτυξης Έ-

μπειρων Συστημάτων που θα χρησιμοποιηθεί, και το οποίο θα κρίνει σε μεγάλο βαθμό τη μορφή της αναπαράστασης της γνώσης, καθώς και η συλλογιστική που θα χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων. Μέρος της σχεδίασης αποτελεί ο προσδιορισμός της εσωτερικής δομής των Γεγονότων, με τρόπο συνεπή ώστε να συμβάλλει στην εύκολη κατανόηση και επεκτασιμότητα της εφαρμογής. Επιθυμητή είναι και η ανάπτυξη ενός προσεγγιστικού περιβάλλοντος διεπαφής, για το οποίο πρέπει να ζητηθεί η γνώμη των χρηστών πριν ξεκινήσει η φάση της υλοποίησης του συστήματος. (ο. π.)

Στο στάδιο αυτό πρέπει να ληφθούν σοβαρά υπόψη οι αποφάσεις που είχαν ληφθεί κατά το στάδιο της ανάλυσης του προβλήματος σχετικά με την τεχνολογία που θα χρησιμοποιηθεί και το μοντέλο της γνώσης που είχε επιλεγεί. Επίσης πρέπει να εξασφαλιστεί ότι η σχεδίαση προδιαγράφει ένα σύστημα που θα ικανοποιεί τις δεσμεύσεις απέναντι στις απαιτήσεις των χρηστών. (ο. π.)

### ***Υλοποίηση***

Στο στάδιο της υλοποίησης κωδικοποιείται το μοντέλο της γνώσης χρησιμοποιώντας εργαλεία ανάπτυξης Έμπειρων Συστημάτων, με βάση τις αποφάσεις που ελήφθησαν κατά το στάδιο της σχεδίασης. Η υλοποίηση ξεκινά με την ανάπτυξη ενός πρωτότυπου συστήματος επίδειξης. Αν το πρωτότυπο σύστημα αποδειχθεί ότι όντως ικανοποιεί τις απαιτήσεις που τέθηκαν στη φάση της ανάλυσης του συστήματος και επαληθεύει τη γνώση που αποκτήθηκε από τον ειδικό και μοντελοποιήθηκε από το μηχανικό γνώσης, μέσω των διαδικασιών επαλήθευσης και ελέγχου, καθοδηγεί την περαιτέρω ανάπτυξη του συστήματος. Διαφορετικά, ανάλογα με το είδος και το μέγεθος των αποκλίσεων, είναι δυνατό να οδηγήσει σε επανασχεδιασμό του συστήματος. (ο. π.)

### ***Επαλήθευση και έλεγχος αξιοπιστίας***

Η διαδικασία επαλήθευσης του συστήματος περιλαμβάνει τον έλεγχο της συμβατότητας του συστήματος με τις αρχικές προδιαγραφές. Η επαλήθευση επικεντρώνεται κάθε φορά σε ένα από τα δομικά στοιχεία του

συστήματος. Έτσι μπορεί να στοχεύει είτε στην επιβεβαίωση της συνέπειας και πληρότητας της κωδικοποίησης της γνώσης που περιέχεται στο Έμπειρο Σύστημα, είτε στην επαλήθευση της λειτουργίας του μηχανισμού εξαγωγής συμπερασμάτων, είτε στον έλεγχο της λειτουργίας του περιβάλλοντος διεπαφής. Ο έλεγχος πραγματοποιείται από το μηχανικό της γνώσης με τη βοήθεια εργαλείων, όπως π.χ. των CHECK, TEIRESIAS κ.α. (ο. π.)

Ο έλεγχος της αξιοπιστίας συνίσταται στην επιβεβαίωση της ορθότητας και της γενικότητας της γνώσης που περιέχει το Έμπειρο Σύστημα. Κατά τη διαδικασία ελέγχου της αξιοπιστίας το σύστημα επιλύει ένα σύνολο από υποδειγματικές περιπτώσεις (test cases). Στη συνέχεια οι λύσεις συγκρίνονται με λύσεις που δόθηκαν από διάφορους ειδικούς του τομέα, με επιθυμητό αποτέλεσμα την ταυτοποίηση των απαντήσεων. Οι υποδειγματικές περιπτώσεις πρέπει να είναι διαφορετικές από αυτές που χρησιμοποιήθηκαν στις προηγούμενες φάσεις ανάπτυξης του συστήματος. Επιπλέον, σκοπός του ελέγχου αξιοπιστίας είναι να εξασφαλιστεί η ανθεκτικότητα και η δυνατότητα απόκρισης του συστήματος σε μη-προσδοκώμενα δεδομένα (ο. π.)

## 2.6.4 Έμπειρα Συστήματα και συμβατικά προγράμματα

Οι βασικές διαφορές των Έμπειρων Συστημάτων με τα κλασικά αλγοριθμικά προγράμματα επεξεργασίας δεδομένων παρουσιάζονται στον Πίνακα 1.

ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ	ΣΥΜΒΑΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ
Προσομοιώνουν τον τρόπο επίλυσης ενός προβλήματος	Προσομοιώνουν το ίδιο το πρόβλημα
Παράσταση και χειρισμός γνώσης σε επίπεδο συμβόλων	Παράσταση και χειρισμός δεδομένων σε επίπεδο αριθμητικών υπολογισμών
Βάση γνώσης	Βάση δεδομένων
Χρήση ευριστικών μεθόδων	Χρήση αλγορίθμων
Δυνατότητα επεξήγησης	Ανυπαρξία επεξήγησης
Χειρισμός ασαφούς, αβέβαιης και μη-πλήρους γνώσης	Δυσχέρεια στο χειρισμό ασαφούς, αβέβαιης και μη-πλήρους γνώσης
Συμπερασματική διαδικασία	Επαναληπτική διαδικασία

Πίνακας 1. Διαφορές Έμπειρων Συστημάτων και κλασικών αλγοριθμικών προγραμμάτων

## 2.6.5 Πλεονεκτήματα/Μειονεκτήματα του ΕΣ σε σχέση με τον Άνθρωπο-Ειδικό

Τα πλεονεκτήματα και μειονεκτήματα ενός Έμπειρου Συστήματος σε σχέση με τον άνθρωπο ειδικό φαίνονται στον Πίνακας 2.

ΑΝΘΡΩΠΟΣ ΕΙΔΙΚΟΣ		ΕΜΠΕΙΡΟ ΣΥΣΤΗΜΑ	
Μειονεκτήματα	Γνώση διαθέσιμη όταν ο ίδιος είναι παρών	Πλεονεκτήματα	Γνώση πάντα διαθέσιμη.
	Δυσκολία μεταφοράς-αποτύπωσης γνώσης		Ευκολία μεταφοράς-αποτύπωσης γνώσης
	Συναισθηματικές παρορμήσεις		Εργάζεται με συνέπεια
	Η απόδοσή του επηρεάζεται από εξωγενείς παράγοντες		Εργάζεται οπουδήποτε
	Υψηλό κόστος		Χαμηλό κόστος λειτουργίας / Υψηλό κόστος ανάπτυξης
	Υποκειμενικότητα		Αντικειμενικότητα αν η γνώση προέρχεται από πολλούς ειδικούς
ΑΝΘΡΩΠΟΣ ΕΙΔΙΚΟΣ		ΕΜΠΕΙΡΟ ΣΥΣΤΗΜΑ	
Πλεονεκτήματα	Δημιουργικότητα, Ευρύνοια	Μειονεκτήματα	Απουσία έμπνευσης, Περιορισμένο πεδίο σκέψης
	Κοινή λογική		Δυσχέρεια στη μεταφύτευση της κοινής λογικής
	Γνώση των ορίων και δυνατοτήτων τους (μετα-γνώση)		Έλλειψη μετα-γνώσης
	Εκφραστική και λειτουργική επεξήγηση του τρόπου σκέψης τους		Μηχανική επεξήγηση του τρόπου λήψης απόφασης
	Ο έλεγχος της γνώσης γίνεται υποσυνείδητα		Πρέπει η γνώση να ελέγχεται για ορθότητα, πληρότητα και συνέπεια
	Αυτονομία στη μάθηση		Πρέπει να προγραμματιστούν για να μαθαίνουν αυτόματα
	Απόκριση σε πραγματικό χρόνο		Δυσκολία απόκρισης σε πραγματικό χρόνο

Πίνακας 2. Πλεονεκτήματα και μειονεκτήματα ενός Έμπειρου Συστήματος

## 2.7 Διάγνωση ασθενειών των φυτών

Οι σημαντικότερες παραγωγικές και οικονομικές απώλειες στη γεωργία οφείλονται κυρίως στις ασθένειες του φυτού. (Indhu Mathi, Kavya, Meera, Sankar Ganesh, 2018) Οι ασθένειες των φυτών μπορούν να αυξήσουν το κόστος της γεωργικής παραγωγής και μπορεί να οδηγήσουν σε μια συνολική οικονομική καταστροφή ένα παραγωγό, εάν δεν θεραπευθούν κατάλληλα σε πρώιμα στάδια. (Petrellis, 2015) Για τον έλεγχο της εξάπλωσης της νόσου, οι αγρότες πρέπει να παρακολουθούν συνεχώς την υγεία των φυτών, η οποία απαιτεί μεγαλύτερο κόστος. Επίσης, οι αγρότες πρέπει να εκπαιδεύονται και ακόμη και να ενημερώνονται για την ασθένεια και την επίδρασή της στα φυτά. Τα πιο σημαντικά συμπτώματα μιας ασθένειας, όπως οι βλάβες στα φύλλα, στα φρούτα, στους μίσχους κ.λ.π. είναι ορατά. (Indhu Mathi, Kavya, Meera, Sankar Ganesh, 2018)

Η πρόσληψη επαγγελματιών γεωργών μπορεί να μην είναι εφικτή ειδικά σε απομακρυσμένες απομονωμένες γεωγραφικές περιοχές. (Petrellis, 2015) Η απομακρυσμένη παρακολούθηση μέσω της μηχανικής όρασης μπορεί να προσφέρει μια εναλλακτική επιλογή και μια τέτοια προσέγγιση μπορεί ούτως ή άλλως να χρησιμοποιηθεί από έναν επαγγελματία για να προσφέρει τις υπηρεσίες του με χαμηλότερο κόστος. Μπορεί να χρειαστεί να πραγματοποιηθεί μοριακή ανάλυση προκειμένου να επιβεβαιωθεί εάν ένα φυτό επηρεάζεται από μια συγκεκριμένη ασθένεια, αλλά η επεξεργασία εικόνας μπορεί να δώσει μια πρώτη ένδειξη για το τι πραγματικά συμβαίνει στο χωράφι. (Petrellis, 2017)

Η διάγνωση της ασθένειας των φυτών μπορεί να γίνει με βάση αρκετά συμπτώματα. Η εξέλιξη των συμπτωμάτων στο χρόνο μπορεί να διαφέρει σημαντικά ανάλογα με τους βιοτικούς παράγοντες και αυτοί μπορούν να ταξινομηθούν ως πρωτογενείς ή δευτερογενείς. Περισσότερα από ένα παθογόνα μπορούν να μολύνουν ταυτόχρονα ένα φυτό. Τα συμπτώματα που εμφανίζονται σε αυτή την περίπτωση μπορεί να διαφέρουν από τα συμπτώματα που προκαλούνται από τα μεμονωμένα παθογόνα. Τα συμπτώματα ενός παθογόνου μπορεί συχνά να εκφράζονται ως μυκητιακές ή βακτηριακές κηλίδες στα φύλλα. Τα φύλλα μπορεί να αλλοιωθούν ή μπορεί να εμφανιστεί ωίδιο. Μπορεί επίσης να εμφανιστούν δομές σπορίων. (Indhu Mathi, Kavya, Meera, Sankar Ganesh, 2018)

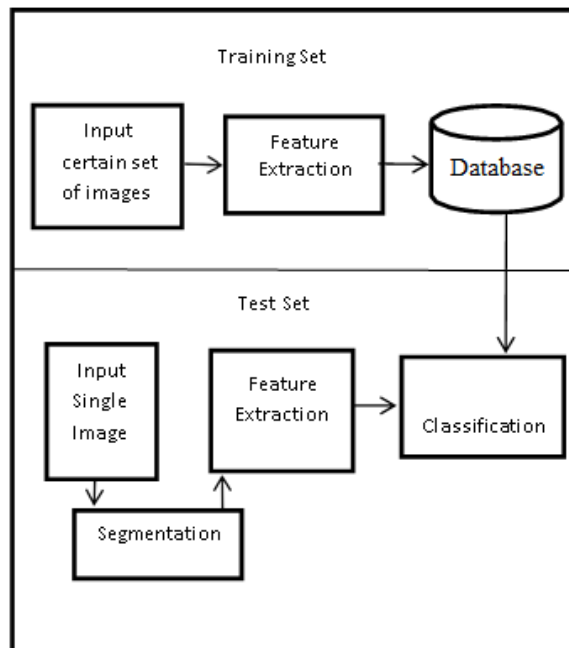
Η αναγνώριση ασθενειών ή ανεπάρκειας γίνεται συνήθως από τους αγρότες με συχνή παρακολούθηση των φύλλων των φυτών, των λουλουδιών, των καρ-

πών ή των μίσχων. Για τους αγρότες μικρής κλίμακας, είναι εφικτή η έγκαιρη αναγνώριση της ασθένειας και αυτοί είναι σε θέση να ελέγχουν τα έντομα με οργανικά παρασιτοκτόνα ή με τη χρήση ελάχιστης ποσότητας χημικών παρασιτοκτόνων. Για τους αγρότες μεγάλης κλίμακας, δεν είναι εφικτή η συχνή παρακολούθηση και η έγκαιρη αναγνώριση των ασθενειών και αυτό οδηγεί σε σοβαρή εστία εμφάνισης ασθενειών και ανάπτυξης παρασίτων που δεν μπορούν να ελεγχθούν με βιολογικά μέσα. Σε αυτή την περίπτωση, οι αγρότες υποχρεώνονται να χρησιμοποιήσουν τις δηλητηριώδεις χημικές ουσίες για να εξαλείψουν την ασθένεια προκειμένου να διατηρήσουν την απόδοση των καλλιεργειών. Αυτό το πρόβλημα μπορεί να επιλυθεί με την αυτοματοποίηση της διαδικασίας παρακολούθησης με τη χρήση προηγμένων τεχνικών επεξεργασίας εικόνας. (Jayaprakash, Veni, 2016)

Η ψηφιακή επεξεργασία εικόνας αναφέρεται στην επεξεργασία ψηφιακών εικόνων μέσω ψηφιακών υπολογιστών. Πρόκειται για μια διαδικασία μετατροπής μιας εικόνας σε ψηφιακή μορφή και την εκτέλεση ορισμένων λειτουργιών για να αποκτηθεί μια βελτιωμένη εικόνα και να εξαχθούν χρήσιμες πληροφορίες από αυτήν. Υπάρχουν διάφορα πεδία εφαρμογής της επεξεργασίας εικόνας όπως η πυρηνική ιατρική, η αστρονομική παρατήρηση, η αναγνώριση υπογραφών, η ανίχνευση πινακίδων, η γεωργία κλπ. Η ψηφιακή επεξεργασία εικόνας χρησιμοποιείται ευρέως στον αγροτικό τομέα. Χρησιμοποιείται σε έλεγχο συγκομιδής, ταξινόμηση φρούτων, σπορά και συλλογή τροφίμων, ανίχνευση ασθενειών των φυτών κλπ. (Khan, Oberoi, 2019)

Η ανίχνευση ασθένειας των φυτών είναι η τεχνική που εφαρμόζεται για την ανίχνευση ασθενειών από τα μολυσμένα φυτά. Οι τεχνικές ανίχνευσης ασθενειών των φυτών αποτελούνται από δύο φάσεις, στην πρώτη φάση γίνεται κατάτμηση της εικόνας εισόδου και ανίχνευση του ασθενικού τμήματος της εικόνας εισόδου. Η τεχνική της εξαγωγής χαρακτηριστικών, η οποία εφαρμόζεται στη δεύτερη φάση, θα εξαγάγει τα χαρακτηριστικά της εικόνας και επίσης θα ταξινομήσει τα εξαγόμενα χαρακτηριστικά χρησιμοποιώντας διάφορους ταξινομητές. (Εικόνα 16) (ο. π.)





Εικόνα 16. Γενική διαδικασία ανίχνευσης ασθενειών φυτών με επεξεργασία εικόνας

Υπάρχουν διάφοροι τρόποι ανίχνευσης παθολογίας των φυτών. Ορισμένες ασθένειες δεν έχουν ορατά συμπτώματα και οι αγρότες δεν μπορούν να τις αναγνωρίσουν εύκολα. Σε αυτές τις περιπτώσεις, είναι συνήθως απαραίτητο ένα είδος εξελιγμένης ανάλυσης, συνήθως μέσω ισχυρών μικροσκοπίων. Σε άλλες περιπτώσεις, τα σημάδια μπορούν να ανιχνευθούν μόνο με μηχανή απεικόνισης που καλύπτει σχεδόν ολόκληρο το ηλεκτρομαγνητικό φάσμα που κυμαίνεται από γάμμα έως ραδιοκύματα που δεν είναι ορατά στον άνθρωπο. Οι γεωργοί χρησιμοποιούν τη μέθοδο παρατήρησης δια γυμνού οφθαλμού και την εμπειρία τους για την ανίχνευση των ασθενειών, αλλά είναι πολύ δύσκολο να εντοπιστεί η ασθένεια σε πολύ πρώιμο στάδιο με αυτή τη μέθοδο. Για τον εντοπισμό διαφορετικών ασθενειών των φυτών μπορεί να εφαρμοστεί αυτοματοποιημένο σύστημα χρησιμοποιώντας διαφορετικούς ταξινομητές. (ο. π.)

Οι ασθένειες των φυτών αποτελούν έναν από τους σημαντικότερους λόγους που οδηγούν στην καταστροφή των φυτών και των καλλιεργειών. Η ανίχνευση αυτών των ασθενειών σε πρώιμα στάδια μας επιτρέπει να τις ξεπεράσουμε και να τις αντιμετωπίσουμε κατάλληλα. Αυτή η διαδικασία απαιτεί έναν εμπειρογνώμονα να εντοπίσει την ασθένεια, να περιγράψει τις μεθόδους θεραπείας και προστασίας. Ο προσδιορισμός της θεραπείας με ακρίβεια εξαρτάται από τη μέθοδο που χρησιμοποιείται στη διάγνωση των ασθενειών. Τα έμπειρα συστήματα βοηθούν πολύ στον

εντοπισμό αυτών των ασθενειών και περιγράφουν τις μεθόδους θεραπείας που πρέπει να ακολουθηθούν λαμβάνοντας υπόψη την ικανότητα των χρηστών να χειρίζονται και να αλληλεπιδρούν με το έμπειρο σύστημα εύκολα και με σαφήνεια. Αυτό προϋποθέτει ότι ο χρήστης είναι ικανός να χρησιμοποιεί έμπειρα συστήματα. (Abu-Naser, Kashkash, Fayyad, 2008)

Ο εντοπισμός των ασθενειών των φυτών δεν είναι εύκολο έργο, χρειάζεται εμπειρία και γνώση των φυτών και των ασθενειών τους. Επιπλέον, απαιτεί ακρίβεια στην περιγραφή των συμπτωμάτων των ασθενειών των φυτών. Ένας άνθρωπος μπορεί να βασισθεί σε ένα σύστημα που διαθέτει εμπειρία και γνώση (έμπειρο σύστημα) για να βοηθηθεί να εντοπίσει οποιοδήποτε είδος ασθένειας, να πάρει τη σωστή απόφαση και να επιλέξει τη σωστή θεραπεία. (ο. π.)

Οι μέθοδοι που χρησιμοποιούν τα έμπειρα συστήματα διαφέρουν από το ένα σύστημα στο άλλο, διότι αυτό εξαρτάται από την πρωτογενή γνώση του χρήστη. Η λήψη αποφάσεων εξαρτάται κυρίως από τον τρόπο απόκτησης αυτής της γνώσης. (ο. π.)

Οι γνώσεις του έμπειρου συστήματος είναι διαθέσιμες ακόμα και όταν ο άνθρωπος ειδικός μπορεί να μην είναι διαθέσιμος και έτσι η γνώση είναι διαθέσιμη ανά πάσα στιγμή και σε πολλά μέρη, όπου είναι απαραίτητο. Τα έμπειρα συστήματα αντλούν τα δεδομένα τους για τη λήψη αποφάσεων από προτροπές του χρήστη σε μια διεπαφή χρήστη ή από αρχεία δεδομένων που είναι αποθηκευμένα στον υπολογιστή. (ο. π.)

Επιπλέον είναι δυνατή η αποθήκευση πολλών από τις πληροφορίες που χρειάζεται ένας εμπειρογνώμονας για τη λήψη αποφάσεων και έτσι μπορεί να είναι διαθέσιμες και για άλλους, επομένως, η έννοια της βασισμένης στη γνώση γεωργίας έχει μια επαρκή προοπτική να βελτιώσει την γεωργική παραγωγή. (ο. π.)

## **2.8 Κοντινές ερευνητικές προσπάθειες**

Στο [19] περιγράφεται ένας αποδοτικός αλγόριθμος αυτοματοποιημένης διάγνωσης και ταξινόμησης των ασθενειών των φυτών που βασίζεται στην ανίχνευση ορατών συμπτωμάτων μιας ασθένειας όπως είναι οι αλλοιώσεις στα φύλλα, τα φρούτα, τους μίσχους κλπ. Ο χρήστης πρέπει να φορτώσει την εικόνα του φυτού με τις αλλοιώσεις από τις οποίες θα ανιχνευθεί η ασθένεια. Στη συνέχεια, η εικόνα

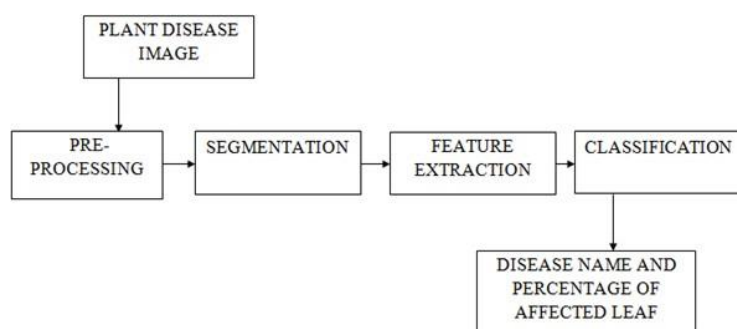
χωρίζεται σε συστάδες υγιούς περιοχής των φύλλων, μολυσμένης περιοχής λόγω των παθογόνων και πληγείσας περιοχής φύλλων λόγω των παθογόνων. Με βάση το επιλεγμένο σύμπλεγμα, αποκτάται το χαρακτηριστικό του επιλεγμένου συμπλέγματος. Το χαρακτηριστικό που προκύπτει συγκρίνεται με τον τυπικό χαρακτήρα της ασθένειας και δίνεται το όνομα της ασθένειας που επηρέασε το φυτό. Λαμβάνεται επίσης το ποσοστό των προσβεβλημένων φύλλων.

### **Πλεονεκτήματα**

- Αυτή η τεχνολογία βελτιώνει την αναγνώριση των ακμών
- Εξουδετερώνει το μειονέκτημα της μείωσης του μεγέθους και της αργής συνδεσιμότητας των συσκευών.

Τα βήματα του αλγορίθμου του συστήματος φαίνονται παρακάτω (Εικόνα 17)

- εικόνα των ασθενειών των φυτών
- προεπεξεργασία
- τμηματοποίηση (K-means Clustering)
- εξαγωγή χαρακτηριστικών (GLCM, Gray Level Co-occurrence Matrix)
- ταξινόμηση (SVM, Support Vector Machine)
- ονομασία της νόσου και ποσοστό των προσβεβλημένων φύλλων



Εικόνα 17. Block διάγραμμα των βημάτων του αλγορίθμου

Στο [26] περιγράφεται μια εφαρμογή για Windows Phone. Βασίζεται στην επεξεργασία εικόνας που αναλύει τα χαρακτηριστικά χρώματος των κηλίδων στα μέρη των φυτών. Η αναγνώριση μιας ασθένειας μπορεί συχνά να βασίζεται σε συ-

μπτώματα όπως αλλοιώσεις ή κηλίδες σε διάφορα μέρη ενός φυτού. Το χρώμα, το εμβαδό και ο αριθμός αυτών των κηλίδων μπορεί να καθορίσει σε μεγάλο βαθμό την ασθένεια που έχει μολύνει ένα φυτό. Μετά, αν είναι απαραίτητο, μπορεί να ακολουθήσουν μοριακές αναλύσεις και δοκιμές οι οποίες είναι υψηλού κόστους. Η συγκεκριμένη εφαρμογή είναι σε θέση να αναγνωρίσει τις αμπελουργικές ασθένειες μέσω φωτογραφιών των φύλλων με ακρίβεια μεγαλύτερη από 90% χρησιμοποιώντας ένα μικρό σετ εκπαίδευσης (Εικόνα 18). Αυτή η εφαρμογή μπορεί εύκολα να επεκταθεί για διαφορετικές ασθένειες φυτών και για διαφορετικές πλατφόρμες έξυπνων τηλεφώνων.

<i>Grape Disease</i>	<i>Successful Classification (Same TH used with the training set)</i>	<i>Successful Classification (10% lower TH)</i>	<i>Successful Classification (10% higher TH)</i>
Downy Mildew	90%	70%	70%
Powdery Mildew	98%	72%	68%
Phomopsis	88%	92%	84%
Esca	98%	98%	80%

Εικόνα 18. Αποτελέσματα ταξινόμησης ασθενειών

Οι συσκευές android είναι πολύ χρήσιμες όταν πρόκειται για συστήματα ταυτοποίησης ασθενειών των φύλλων των φυτών. Σε αυτή την περίπτωση θα πρέπει να δημιουργηθεί και η κατάλληλη εφαρμογή android (Εικόνα 19). Το πλήρες σύστημα που περιγράφεται στο [15] σχεδιάστηκε χρησιμοποιώντας το matlab και το λειτουργικό σύστημα android. Τα πειράματα διεξάγονται σε επεξεργαστή μονού πυρήνα για την πραγματική μέτρηση του υπολογιστικού κόστους και του παράγοντα χρόνου που σχετίζεται με αυτό. Δείγματα ασθενικών φύλλων λαμβάνονται χρησιμοποιώντας την κάμερα από διαφορετικά smartphones σε διαφορετικές αναλύσεις. Το προτεινόμενο σύστημα είναι μια λύση λογισμικού για τον αυτόματο υπολογισμό και την ανίχνευση χαρακτηριστικών υφής και σχήματος για ασθένειες των φύλλων.



Εικόνα 19. Το όνομα της ασθένειας και τα μέτρα πρόληψης σαν έξοδος στην οθόνη του android κινητού

Τα βήματα της λειτουργίας του συστήματος είναι:

1. Διαχείριση βάσεων δεδομένων με εικόνες φύλλων
2. Επιλογή εικόνας
3. Κατάτμηση K-means
4. Εξαγωγή χαρακτηριστικών GLCM
5. Διαδικασία Ταξινόμησης με SVM classifier
6. Ανάκτηση

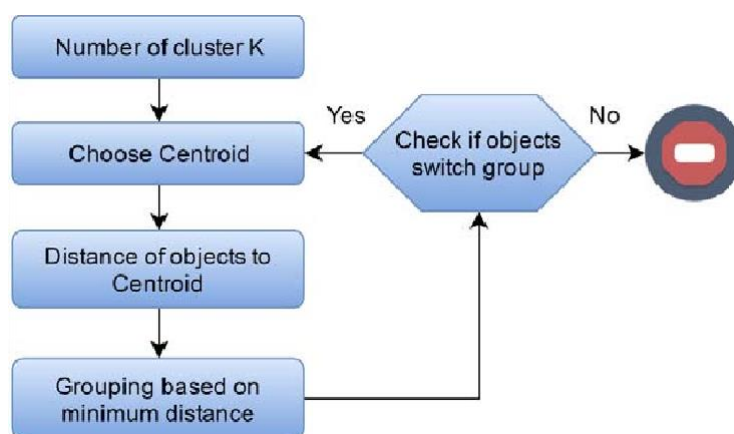
Στο [20] περιγράφεται ένα σύστημα που αυτοματοποιεί τη διαδικασία παρακολούθησης των φύλλων των φυτών κάνοντας χρήση προηγμένων τεχνικών επεξεργασίας εικόνας με στόχο την αναγνώριση των ασθενειών των φύλλων μάγκο. Το αυτοματοποιημένο σύστημα προτείνεται για χρήση σε συσκευές ή πλατφόρμες χαμηλού κόστους, όπως οι συσκευές android και το raspberry pi. Στοχεύει επίσης στην ταξινόμηση και αναγνώριση των ασθενειών των φύλλων μάγκο για την ιδιική γεωργία. Ο αλγόριθμος K-means επιλέγεται για την κατάτμηση της εικόνας της ασθένειας και η ταξινόμηση και η ταυτοποίηση της ασθένειας πραγματοποιείται χρησιμοποιώντας τον ταξινομητή SVM. Η ταυτοποίηση των ασθενειών με βάση

την ανάλυση των κηλίδων ή την αποχρωματισμό των φύλλων θα είναι καλή για ορισμένες από τις ασθένειες των φυτών, αλλά μερικές άλλες ασθένειες που θα παραμορφώσουν το σχήμα των φύλλων δεν μπορούν να προσδιοριστούν με βάση την ίδια μέθοδο. Σε αυτή την περίπτωση πρέπει να γίνει η ταυτοποίηση με βάση το σχήμα των φύλλων.

Το σύστημα βασίζεται σε:

- (i) Ταυτοποίηση των ασθενειών χρησιμοποιώντας τις βιβλιοθήκες OpenCV
- (ii) Προσδιορισμό της ασθένειας με βάση το σχήμα του φύλλου.

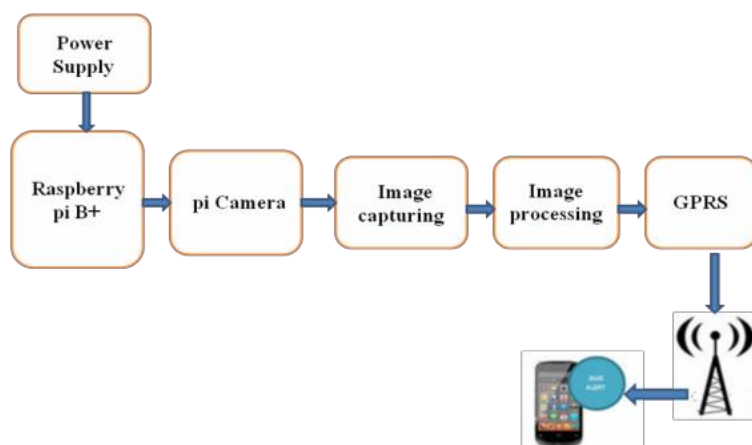
Τα βήματα που εμπλέκονται στην ανίχνευση ασθένειας είναι η λήψη ψηφιακών εικόνων, η προεπεξεργασία εικόνας (απομάκρυνση θορύβου, μετασχηματισμός χρώματος και ισοστάθμιση ιστογράμματος), k-means κατάτμηση (Εικόνα 20), εξαγωγή χαρακτηριστικών και ταξινόμηση χρησιμοποιώντας τον αλγόριθμο SVM ο οποίος είναι αλγόριθμος εποπτευόμενης μάθησης.



Εικόνα 20. Τα βήματα του αλγορίθμου κατάτμησης K-means

Στο [16] περιγράφεται ένα αυτοματοποιημένο σύστημα που χρησιμοποιεί μηχανική όραση για να ανιχνεύει, να αναγνωρίζει και να ταξινομεί την ασθένεια που έχει επηρεάσει την καλλιέργεια, ώστε να αποφεύγεται η ανθρώπινη παρέμβαση και επομένως να οδηγεί σε μια απόφαση αμερόληπτη και ακριβή σχετικά με την ασθένεια και την περαιτέρω αποτίμησή της. Το σύστημα παρέχει τη δυνατότητα λήψης

εικόνας, επεξεργασίας και λήψης αποτελεσμάτων μέσω τεχνικών επεξεργασίας εικόνας. Βασίζεται στη βιβλιοθήκη openCV και στη γλώσσα προγραμματισμού Python. Η ανίχνευση, η ταξινόμηση των ασθενειών των φύλλων που επηρεάζουν τις καλλιέργειες φρούτων χρησιμοποιώντας εργαλεία επεξεργασίας εικόνας και όλες οι πληροφορίες σχετικά με την ασθένεια αποστέλλονται στο κινητό τηλέφωνο του αγρότη μέσω του GPRS. Για την αύξηση της ταχύτητας και της ακρίβειας της ανίχνευσης και της ταξινόμησης των ασθενειών των φύλλων χρησιμοποιείται η μονάδα Raspberry pi. Αρχικά λαμβάνονται σε πραγματικό χρόνο οι εικόνες διαφόρων φύλλων χρησιμοποιώντας μια φωτογραφική μηχανή συνδεδεμένη σε raspberry pi. Στη συνέχεια εφαρμόζονται διάφορες μέθοδοι επεξεργασίας εικόνας στις εικόνες που έχουν ληφθεί για να πάρουν χρήσιμα χαρακτηριστικά που είναι σημαντικά για την επόμενη διαδικασία ανάλυσης. Η προεπεξεργασία συνίσταται επίσης στη μετατροπή εικόνων RGB σε γκρι, καθώς η εικόνα στην κλίμακα του γκρι παρέχει τέλεια ακρίβεια στην ανίχνευση ελαττωμάτων. Η διαδικασία εξαγωγής χαρακτηριστικών συνίσταται στην πραγματική ανίχνευση ασθένειας από μια εικόνα, συγκρίνοντας την εικόνα με εικόνες που δεν έχουν υποστεί βλάβη. Μετά από αυτό, εφαρμόζονται πολλές αναλυτικές τεχνικές ή μέθοδοι για να ταξινομηθούν οι εικόνες σύμφωνα με το συγκεκριμένο πρόβλημα. Τέλος η ασθένεια που έχει ανιχνευθεί και ταξινομηθεί θα αποσταλεί στο κινητό τηλέφωνο (Εικόνα 21).

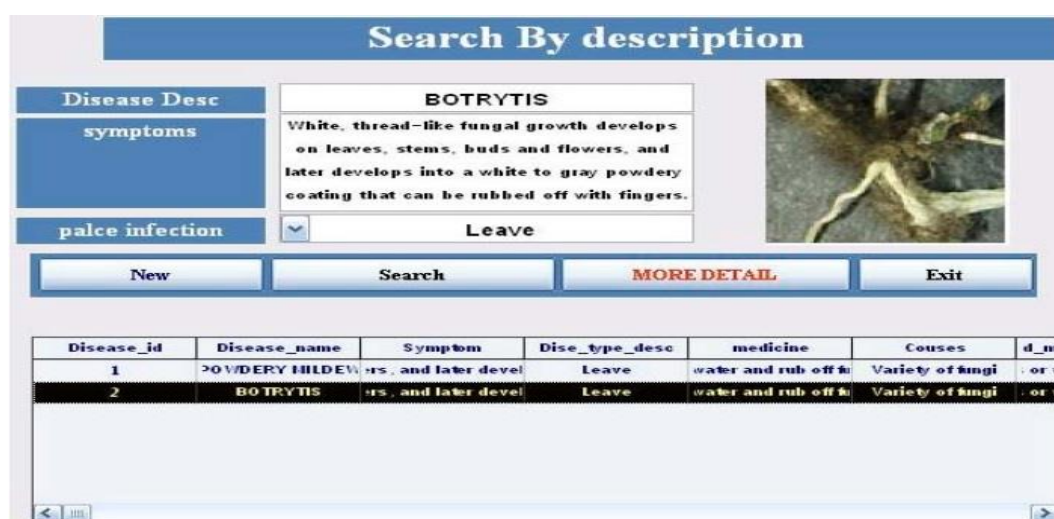


Εικόνα 21. Σύστημα αυτόματης ανίχνευσης και ταξινόμησης των ασθενειών των φύλλων

Ο σχεδιασμός και η εφαρμογή αυτών των τεχνολογιών θα συμβάλουν σημαντι-

κά στην εφαρμογή επιλεκτικής γεωργίας, μειώνοντας το κόστος και οδηγώντας έτσι στη βελτίωση της παραγωγικότητας.

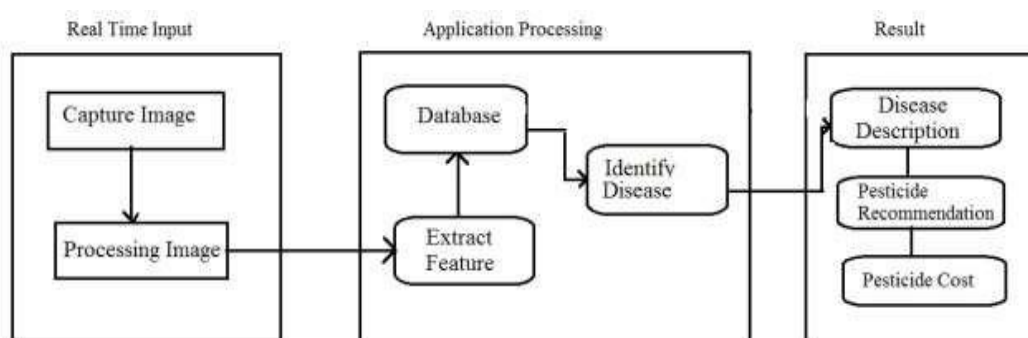
Στο [14] παρουσιάζεται ο σχεδιασμός και η ανάπτυξη ενός έμπειρου συστήματος που χρησιμοποιεί δύο διαφορετικές μεθόδους διάγνωσης ασθενειών των φυτών. Η πρώτη είναι η περιγραφική μέθοδος (βήμα προς βήμα) και η δεύτερη είναι η μέθοδος γραφικής αναπαράστασης. Κατά το σχεδιασμό αυτού του έμπειρου συστήματος χρησιμοποιείται η γλώσσα προγραμματισμού Visual Basic για τη δημιουργία του Graphical User Interface (GUI) και το πρόγραμμα CLIPS για τη δημιουργία του συστήματος λήψης αποφάσεων σύμφωνα με τις πληροφορίες που λαμβάνει ο χρήστης. Μια προκαταρκτική αξιολόγηση του συστήματος έδειξε ότι το έμπειρο σύστημα με τη γραφική αναπαράσταση είναι πιο ευνοϊκό από το περιγραφικό. Αυτό οφείλεται στις δυσκολίες στην περιγραφή των συμπτωμάτων της ασθένειας. Από την άλλη πλευρά, μια γραφική εικόνα των συμπτωμάτων δεν απαιτεί μεγάλη περιγραφή από τον χρήστη. Τα Έμπειρα Συστήματα θεωρούνται μία από τις πιο επιτυχημένες μεθόδους που χρησιμοποιούνται για να βοηθήσουν και να υποστηρίξουν τους χρήστες στη λήψη των σωστών αποφάσεων, εάν αυτοί δεν διαθέτουν τη γνώση για τη διάγνωση ασθενειών των φυτών (Εικόνα 22). Τα υπάρχοντα έμπειρα συστήματα εξοικονομούν πολύ χρόνο και προσπάθεια για τον εντοπισμό των ασθενειών των φυτών λόγω του μηχανισμού που χρησιμοποιείται για τη λήψη των δεδομένων και για τη λήψη των αποφάσεων.



Εικόνα 22. Προβολή αποτελεσμάτων αναζήτησης για να διασφαλιστεί η ακρίβεια της απόφασης



Στο [29] περιγράφεται η ανάπτυξη ενός συστήματος ημι-πραγματικού χρόνου, που χρησιμοποιεί έννοιες επεξεργασίας εικόνας και εξόρυξης δεδομένων, για να παρέχει σύσταση φυτοφαρμάκων και εκτίμηση κόστους φυτοφαρμάκων για τους αγρότες για τη διαχείριση των ασθενειών των καλλιεργειών. Όσον αφορά τις μεθοδολογίες διάγνωσης για την ανίχνευση ασθενειών σε καλλιέργειες, μελετήθηκε η επεξεργασία εικόνας στη διάγνωση ασθενειών και το eAGROBOT. Έτσι δημιουργήθηκε μια εφαρμογή android για την παρακολούθηση αποτελεσματικών παραμέτρων των χαρακτηριστικών για τη διάγνωση ασθενειών και τη δημιουργία συμβουλευτικών συστημάτων. Το σύστημα σχεδιάστηκε και αναπτύχθηκε χρησιμοποιώντας το android studio ως λογισμικό front-end και το SQLite ως back-end. Οι εικόνες και τα διορθωτικά μέτρα των ασθενειών θα αποθηκεύονται σε βάση δεδομένων και θα μπορούν να ανακτηθούν όποτε είναι απαραίτητο. Η πρόκληση είναι να κά-νουμε τους αγρότες να ακούν το σύστημα διάγνωσης ασθενειών των φυτών και να λαμβάνουν συμβουλές σχετικά με τις ασθένειες. Εδώ υπάρχει η απαίτηση να ανα-πτυχθεί το έμπειρο σύστημα στις τοπικές γλώσσες, έτσι ώστε οι αγρότες να μπο-ρούν να εκμεταλλεύονται το έμπειρο σύστημα μόνοι τους και να λαμβάνουν συμ-βουλές ειδικών από το σύστημα. Η αρχιτεκτονική του συστήματος κατηγοριοποιεί-ται σε τρία τμήματα: a) real time input section, b) application processing, c) result (Εικόνα 23).



Εικόνα 23. Η αρχιτεκτονική του συστήματος

Στο [18] αναπτύχθηκαν μοντέλα συνελκτικού νευρωνικού δικτύου για την

ανίχνευση και διάγνωση των ασθενειών των φυτών χρησιμοποιώντας απλές εικόνες υγείων και ασθενικών φύλλων φυτών, μέσω μεθοδολογιών deep learning. Η εκπαίδευση των μοντέλων πραγματοποιήθηκε με τη χρήση μιας ανοικτής βάσης δεδομένων 87.848 εικόνων, που περιέχει 25 διαφορετικά φυτά σε ένα σύνολο 58 ξεχωριστών κατηγοριών συνδυασμών [φυτών, ασθενειών], συμπεριλαμβανομένων υγείων φυτών. Αρκετές αρχιτεκτονικές μοντέλων εκπαιδεύτηκαν, με την καλύτερη επίδοση να επιτυγχάνει ποσοστό επιτυχίας 99,53% στην ταυτοποίηση του αντίστοιχου συνδυασμού [φυτών, ασθενειών] (ή υγιούς φυτού). Το σημαντικότερο ποσοστό επιτυχίας καθιστά το μοντέλο ένα πολύ χρήσιμο εργαλείο συμβουλευτικής ή έγκαιρης προειδοποίησης και μια προσέγγιση που θα μπορούσε να επεκταθεί περαιτέρω για να υποστηρίξει ένα ολοκληρωμένο σύστημα αναγνώρισης των ασθενειών των φυτών ώστε να λειτουργεί σε πραγματικές συνθήκες καλλιέργειας.

Οι μετρικές για ένα μοντέλο συνελκτικού νευρωνικού δικτύου (CNN) είναι:

- (i) τα ποσοστά επιτυχούς ταξινόμησης των κατηγοριών [φυτών, ασθενειών]: αυτός είναι ο αριθμός των σωστά ταξινομημένων εικόνων επί του συνολικού αριθμού εικόνων,
- (ii) η αντίστοιχη μέση τιμή λαθών των μοντέλων: πρόκειται για τη μέση τιμή απωλειών ανά παρτίδα, σε όλες τις παρτίδες του συνόλου δοκιμών,
- (iii) η περίοδος εκπαίδευσης στην οποία επιτεύχθηκαν οι καλύτερες επιδόσεις,
- (iv) ο μέσος χρόνος ανά περίοδο (σε δευτερόλεπτα) που απαιτείται για την εκπαίδευση κάθε μοντέλου.

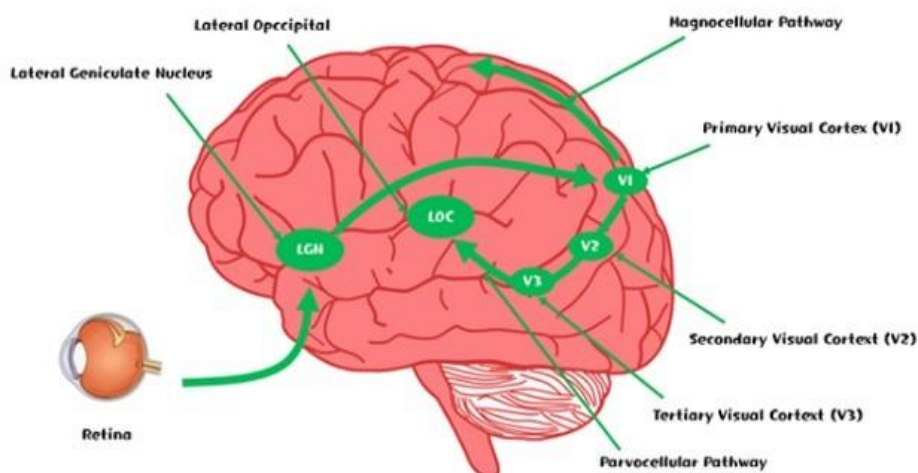
Η Εικόνα 24 δείχνει μερικές περιπτώσεις ταξινόμησης τυχαία επιλεγμένων εικόνων σε όλο το dataset δοκιμών. Η ταξινόμηση μιας εικόνας διαρκεί κατά μέσο όρο περίπου 2 ms στην ίδια GPU που χρησιμοποιήθηκε για την εκπαίδευση.



Εικόνα 24. Παραδείγματα σωστών κατηγοριοποιήσεων διαφόρων εικόνων του dataset δοκιμών

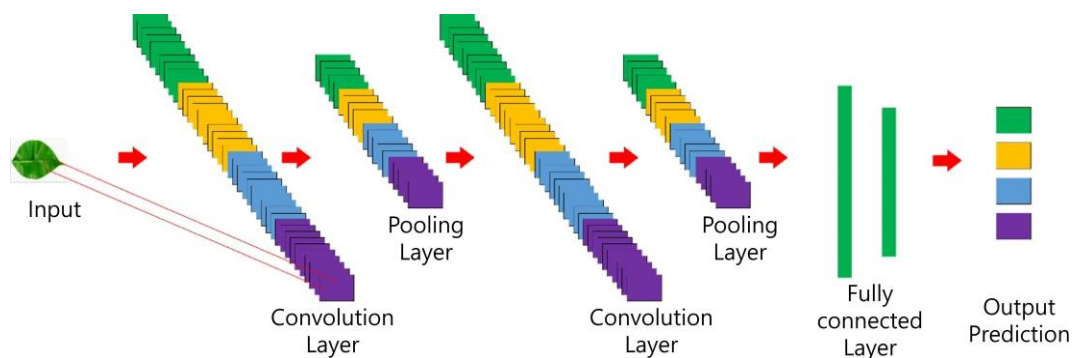
Στο [21] προτείνεται μια νέα μέθοδος ταξινόμησης των φύλλων χρησιμοποιώντας το μοντέλο CNN και δημιουργήθηκαν δύο μοντέλα προσαρμόζοντας το βάθος του δικτύου. Έχει αξιολογηθεί η απόδοση κάθε μοντέλου ανάλογα με τον αποχρωματισμό ή την καταστροφή των φύλλων. Το ποσοστό αναγνώρισης που επιτεύχθηκε ήταν μεγαλύτερο από 94%, ακόμη και όταν το 30% του φύλλου έχει υποστεί βλάβη.

Τα νευρωνικά δίκτυα μιμούνται την ανθρώπινη οπτική επεξεργασία της νευρικής δομής του εγκεφάλου, όπως φαίνεται στην Εικόνα 25.



Εικόνα 25. Η δομή του οπτικού συστήματος του ανθρώπου

Όπως φαίνεται στην Εικόνα 26, η δομή του συνελικτικού νευρωνικού δικτύου (CNN) εξάγει τα χαρακτηριστικά πραγματοποιώντας τη λειτουργία συνέλιξης στην εικόνα εισόδου, εξάγει τις μέγιστες ή μέσες τιμές χαρακτηριστικών στο επίπεδο συγκέντρωσης και στη συνέχεια τις ταξινομεί στο πλήρως συνδεδεμένο επίπεδο.



Εικόνα 26. Βασική δομή ενός νευρωνικού δικτύου συνέλιξης

Το μοντέλο CNN που χρησιμοποιείται είναι το GoogleNet. Με την εμφάνιση αυτού του μοντέλου, οι ερευνητές έχουν αναπτύξει βαθύτερες δομές δικτύου που δεν αυξάνουν την υπολογιστική πολυπλοκότητα. Το GoogleNet χρησιμοποιεί αρχικές ενότητες που χρησιμοποιούν πολλαπλές περιελίξεις παράλληλα, για να εξαγάγουν διάφορα σημεία χαρακτηριστικών.

Στο [17] χρησιμοποιείται το Tensorflow, μια βιβλιοθήκη ανοιχτού κώδικα ροής δεδομένων και μηχανικής μάθησης, για να δημιουργήσει ένα Convolutional Neural Network (CNN) ταξινόμησης εικόνων για την ταξινόμηση των ιθαγενών φυτών στο San Luis Obispo (και την Καλιφόρνια εν γένει). Το Tensorflow, εκτός του ότι παρέχει στους προγραμματιστές ένα απλό τρόπο δημιουργίας επιπέδων νευρωνικών δικτύων, μπορεί επίσης να χρησιμοποιηθεί σε κινητές πλατφόρμες όπως το Android. Ο απώτερος στόχος είναι ο σχεδιασμός και η βελτιστοποίηση ενός συνελικτικού νευρωνικού δικτύου για ταξινόμηση των φυτών και τελικά η δημιουργία μιας απλής εφαρμογής ταξινόμησης για κινητές συσκευές γύρω από το εκπαιδευμένο δίκτυο. Η εφαρμογή για κινητά θα επιτρέψει στους χρήστες να δοκιμάσουν

και να ταξινομήσουν τα φυτά ακόμα κι αν βρίσκονται σε εξωτερικούς χώρους ή βρίσκονται εκτός σύνδεσης.

Το CNN και ο ταξινομητής έχουν τη δυνατότητα να χρησιμοποιηθούν σε εργαστηριακό περιβάλλον από βοτανολόγους ή βιολόγους φυτών, αν και οποιοσδήποτε επιστήμονας θα επιθυμούσε πιθανώς να χρησιμοποιήσει το δικό του σύνολο δεδομένων. Το έργο πιθανότατα θα ήταν χρήσιμο στους επιστήμονες για την ταξινόμηση μεγάλων ποσοτήτων δεδομένων και τη δημιουργία στατιστικών για το σύνολο δεδομένων. Μια άλλη χρήση αυτού του έργου στην τρέχουσα κατάσταση του, θα μπορούσε να είναι για επιστήμονες φυτών που θέλουν να βοηθήσουν στην αναγνώριση ασθενειών σε φυτά, εάν το CNN εκπαιδεύεται με ένα σύνολο δεδομένων προσαρμοσμένο για αυτό. Μόλις δημιουργηθεί μια εφαρμογή για κινητές συσκευές γύρω από το CNN, οι βοτανολόγοι ή άλλοι επιστήμονες που ασχολούνται με φυτά σε χωράφια, θα μπορούσαν επίσης να επωφεληθούν από την ικανότητα να εντοπίσουν γρήγορα εάν ένα φυτό είναι ασθενικό.

Το σενάριο Python που χρησιμοποιεί το εκπαιδευμένο μοντέλο Tensorflow είναι πολύ απλό, επειδή ο περισσότερος χρόνος που δαπανήθηκε ήταν για τη συλλογή των δεδομένων και την εκμάθηση του πώς σχεδιάζεται ένα CNN. Ο ταξινομητής έχει σχεδιαστεί για να λαμβάνει έναν κατάλογο εικόνων, ένα αρχείο κειμένου των ετικετών που χρησιμοποιούνται στο δίκτυο και το ίδιο το εκπαιδευμένο μοντέλο ως εισόδους. Ο ταξινομητής ελέγχει τις εικόνες με το συγκεκριμένο μοντέλο και εμφανίζει τα αποτελέσματα συγκρίνοντας τη σωστή ετικέτα με τις τέσσερις πρώτες κατηγορίες βάσει του επιπέδου εμπιστοσύνης των προβλέψεων (Εικόνα 27).

```

Lasthenia_ferrisiae_23.jpg
Results:
  Lasthenia ferrisiae:      99.9%
  Ericameria arborescens:   0.1%
  Helenium bigelovii:      0.0%
  Dimorphotheca sinuata:    0.0%

Abronia_latifolia_6.jpg
Results:
  Abronia pogonantha:      99.7%
  Abronia latifolia:       0.2%
  Erysimum insulare:       0.0%
  Eschscholzia californica: 0.0%

Calandrinia_breweri_14.jpg
Results:
  Claytonia gypsophiloides: 91.2%
  Aster:      8.7%
  Cirsium californicum:     0.1%
  Calandrinia breweri:      0.0%

Glebionis_4.jpg
Results:
  Glebionis:      89.0%
  Aquilegia eximia: 10.9%
  Erysimum insulare: 0.0%
  Chasmanthe floribunda: 0.0%

```

Εικόνα 27. Η έξοδος του ταξινομητή εικόνων

Η βαθιά εκμάθηση (DL) αποτελεί μια σύγχρονη τεχνική επεξεργασίας εικόνας, με μεγάλες δυνατότητες. Αφού εφαρμόστηκε με επιτυχία σε διάφορους τομείς, εισήλθε πρόσφατα και στον τομέα της γεωργίας. Στο [22] διεξήχθη έρευνα για τις ερευνητικές προσπάθειες που χρησιμοποιούν συνελκτικά νευρωνικά δίκτυα (CNN), τα οποία αποτελούν μια συγκεκριμένη κατηγορία Deep Learning (DL), που εφαρμόζονται σε διάφορες προκλήσεις παραγωγής γεωργικών προϊόντων και τροφίμων. Η εργασία εξετάζει τα γεωργικά προβλήματα που είναι υπό έρευνα, τα χρησιμοποιούμενα μοντέλα, τις πηγές δεδομένων που χρησιμοποιούνται και τη συνολική ακρίβεια που επιτυγχάνεται σύμφωνα με τις μετρήσεις επιδόσεων που χρησιμοποιούνται από τους συγγραφείς. Τα συνελκτικά νευρωνικά δίκτυα συγκρίνονται με άλλες υπάρχουσες τεχνικές και παρατίθενται τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης του CNN στη γεωργία. Επιπλέον, εξετάζονται οι μελλοντικές δυνατότητες αυτής της τεχνικής, μαζί με τις προσωπικές εμπειρίες των συγγραφέων μετά την υιοθέτηση του CNN για την προσέγγιση ενός προβλήματος αναγνώρι-



σης της ελλείψεως βλάστησης από μια φυτεία ζαχαροκάλαμου στην Κόστα Ρίκα. Τα γενικά ευρήματα δείχνουν ότι το CNN αποτελεί μια πολλά υποσχόμενη τεχνική με υψηλή απόδοση όσον αφορά την ακρίβεια και την ακρίβεια ταξινόμησης, ξεπερνώντας τις υπάρχουσες κοινώς χρησιμοποιούμενες τεχνικές επεξεργασίας εικόνας. Ωστόσο, η επιτυχία κάθε μοντέλου CNN εξαρτάται σε μεγάλο βαθμό από την ποιότητα του συνόλου δεδομένων που χρησιμοποιούνται.

Για την καλύτερη κατανόηση των δυνατοτήτων και της αποτελεσματικότητας του CNN, πραγματοποιήθηκε ένα μικρό πείραμα, το οποίο ασχολούνταν με ένα πρόβλημα που δεν αναφέρθηκε στη σχετική εργασία: την αναγνώριση της ελλείψεως βλάστησης από ένα χωράφι, όπως φαίνεται στην Εικόνα 28.



Εικόνα 28. Ταυτοποίηση της ελλείψεως βλάστησης από ένα χωράφι. Οι περιοχές με την ένδειξη (1) αντιπροσωπεύουν παραδείγματα φυτών ζαχαροκάλαμου, ενώ οι περιοχές με την ένδειξη (2) αποτελούν παραδείγματα εδάφους. Οι περιοχές που φέρουν την ένδειξη (3) απεικονίζουν παραδείγματα ελλιπούς βλάστησης, δηλ. θα έπρεπε να ήταν ζαχαροκάλαμο, αλλά είναι έδαφος. Τέλος, οι περιοχές με την ετικέτα (4) είναι παραδείγματα «άλλων», που είναι άσχετα τμήματα εικόνας





## 3 Προτεινόμενο σύστημα

Στη συνέχεια περιγράφεται αναλυτικά το μοντέλο του συστήματος, δίνονται οι τεχνικές λεπτομέρειες, παρουσιάζονται διαγράμματα ροής, αναλύεται η λειτουργία του και προβάλλονται τα οφέλη από τη χρήση του.

### 3.1 System model

Το σύστημα που προτείνεται φαίνεται στην (Εικόνα 29) και αποτελείται από 3 βασικά μέρη:

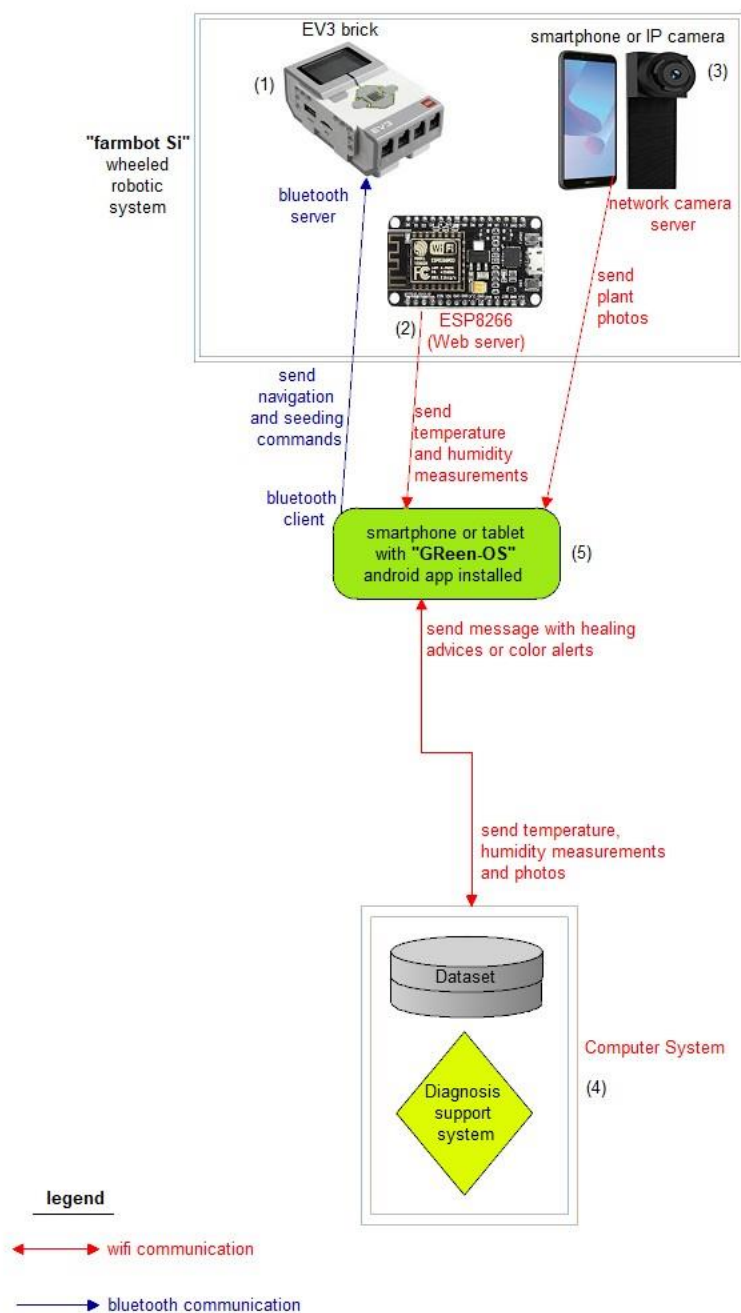
**A)** την τροχήλατη ρομποτική κατασκευή με όνομα **“farmbot-Si”** που περιλαμβάνει:

- (α) ένα EV3 brick που λαμβάνει εντολές μέσω bluetooth από την εφαρμογή "GReen-OS" για χειροκίνητη ή αυτόματη πλοήγηση
- (β) μια πλακέτα που βασίζεται στον μικροεπεξεργαστή arduino που λαμβάνει μετρήσεις θερμοκρασίας και υγρασίας μέσω του αισθητήρα DHT22 και τις μεταφέρει στην εφαρμογή "GReen-OS"
- (γ) ένα smartphone ή μια IP camera που είναι σε θέση να παίρνει φωτογραφίες των φυτών, να τις μεταφέρει στην εφαρμογή "GReen-OS" και να τις αποθηκεύει στην κάρτα sd του smartphone

**B)** ένα υπολογιστικό σύστημα που φιλοξενεί:

- (α) ένα σετ δεδομένων (όπου θα αποθηκευτούν τυποποιημένες φωτογραφίες υγιών και ασθενικών φυτών)
- (β) το σύστημα υποστήριξης διάγνωσης ασθενειών (DiSS).

**Γ)** την android εφαρμογή με όνομα **“GReen-OS”**, που είναι το «μυαλό» του IoT συστήματος.



Εικόνα 29. μοντέλο του προτεινόμενου συστήματος

## 3.2 Τεχνική περιγραφή

### 3.2.1 Περιγραφή αρχιτεκτονικής

Α) Η τροχήλατη ρομποτική κατασκευή “farmbot-Si” που θα πλοηγείται εντός του θερμοκηπίου θα είναι σε θέση να διεκπεραιώνει σπορά ακριβείας και έλεγχο της προόδου της καλλιέργειας. Αποτελείται από τρία υποσυστήματα:

**στοιχείο (1):** τον προγραμματιζόμενο ελεγκτή EV3 της LEGO

**στοιχείο (2):** την πλακέτα μικροελεγκτή ESP-8266

**στοιχείο (3):** το smartphone (ή IP camera)

**στοιχείο (1):** Το EV3 brick είναι αρμόδιο για την πλοήγηση και τις λειτουργίες του οχήματος, δηλαδή κάνει τον έλεγχο των τροχών, τον έλεγχο του μηχανισμού σποράς, κ.τ.λ. Επικοινωνεί μέσω Bluetooth («ασύρματη» τεχνολογία χαμηλού κόστους και χαμηλής ισχύος) με την εφαρμογή “GReen-OS”, η οποία μπορεί να είναι εγκατεστημένη σε κινητό τηλέφωνο ή tablet και στέλνει τις ανάλογες εντολές για χειροκίνητη ή αυτόματη κίνηση, για την σπορά ακριβείας με την βοήθεια κατάλληλου σπορέα και για την επιθεώρηση των φυτών. Το EV3 λειτουργεί ως bluetooth server και το smartphone είναι ο bluetooth client.

**στοιχείο (2):** Είναι η αυτόνομη ενεργειακά πλακέτα μικροελεγκτή ESP-8266 (κάνει την ίδια δουλειά με πλακέτα arduino με συνδεδεμένο WiFi shield, τις οποίες και αντικαθιστά) η οποία τοποθετείται εντός της ρομποτικής κατασκευής. Στην πλακέτα είναι συνδεδεμένος αισθητήρας DHT22 ο οποίος μετρά θερμοκρασία και υγρασία. Η πλακέτα ESP-8266 θα προγραμματιστεί κατάλληλα ώστε να υποστηρίξει Web Service (Web Server), που θα δέχεται τις μετρήσεις θερμοκρασίας και υγρασίας από τον αισθητήρα DHT22 και θα τις μεταβιβάζει (μέσω WiFi) στην πλατφόρμα “GReen-OS” (Web client).

**στοιχείο (3):** Είναι ένα smartphone (ή IP camera) το οποίο, με τη βοήθεια εξωτερικής εφαρμογής που το κάνει να λειτουργεί ως IP camera server, παίρνει φωτογραφίες από τα φυτά και τις στέλνει (μέσω WiFi και του εσωτερικού δικτύου) στο smartphone που τρέχει την εφαρμογή “GReen-OS” (ως client της εξωτερικής εφαρμογής), αποθηκεύοντάς τες ταυτόχρονα στην sd card του.

**B)** Το υπολογιστικό σύστημα στοιχείο (4) που θα λειτουργεί ως Diagnosis Support System περιλαμβάνει:

**(α)** έναν αποθηκευτικό χώρο όπου θα βρίσκονται οι πρότυπες φωτογραφίες των υγείων και ασθενικών φυτών

**(β)** το Diagnosis Support System (DiSS) που θα ενσωματώνει το Συνελικτικό Νευρωνικό Δίκτυο με τη βοήθεια του οποίου οι φωτογραφίες των φυτών που έρχονται (μέσω WiFi) στο DiSS από την εφαρμογή “GReen-OS”, θα υπόκεινται σε επεξεργασία και θα εξάγεται το συμπέρασμα αν το φυτό είναι υγιές ή ασθενικό και από ποιά ασθένεια πάσχει. Το συμπέρασμα θα επιστρέφει από το DiSS στην εφαρμογή “GReen-OS” με μορφή μηνύματος ή χρωματικού alert με υποδείξεις για την αντιμετώπιση της ενδεχόμενης ασθένειας ή των μη ενδεδειγμένων περιβαλλοντικών συνθηκών.

**Γ)** Η εφαρμογή android “GReen-OS” στοιχείο (5) που θα είναι το “μυαλό” της IoT πλατφόρμας. Στέλνει εντολές για κίνηση στο EV3 και δέχεται από την Web Service του ESP8266 μετρήσεις θερμοκρασίας-υγρασίας από τον αισθητήρα DHT22 αλλά και φωτογραφίες των φυτών από συγκεκριμένες θέσεις (μέσω smartphone ή IP κάμερας). Στην συνέχεια τις στέλνει σε fog FTP Server (που φιλοξενεί Dataset και DiSS) με σκοπό να γίνει η επεξεργασία τους και να δεχτεί πίσω μήνυμα ή χρωματικό alert και υποδείξεις τις οποίες θα στείλει με την σειρά του ως ενημέρωση στον υπεύθυνο του θερμοκηπίου. Το “GReen-OS” μπορεί να το χρησιμοποιεί όποιος βρίσκεται στο χώρο του θερμοκηπίου για να επιβλέπει την πλοήγηση του EV3 (μέσω bluetooth επικοινωνίας) αλλά ταυτόχρονα (real time) και οποιοσδήποτε άλλος θέλει, από οπουδήποτε εκτός θερμοκηπίου (μέσω WiFi), να παρακολουθεί και να ενημερώνεται για το τί συμβαίνει στο θερμοκήπιο σε πραγματικό χρόνο, πχ. ο ιδιοκτήτης του και πρακτικά όποιος έχει εγκατεστημένη την εφαρμογή. Το “GReen-OS” είναι ο κεντρικός κόμβος, ο οποίος επικοινωνεί μέσω δύο καναλιών (bluetooth και WiFi) με τα υπόλοιπα τμήματα (EV3 και DiSS) και συγκεντρώνει, αλλά και διαχέει, όλα τα δεδομένα - πληροφορίες από και προς όλα τα συστήματα.

### 3.2.2 Διαγράμματα ροής

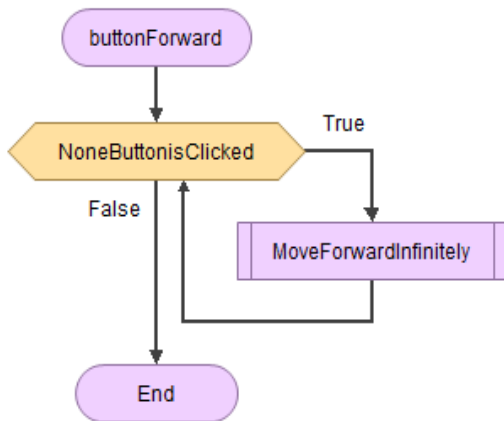
Στην συνέχεια περιγράφονται, υπό μορφή διαγραμμάτων ροής, οι 3 βασικές ενέργειες τις οποίες μπορεί να διεκπεραιώσει η ρομποτική κατασκευή “farmbot-Si” καθοδηγούμενη από την IoT πλατφόρμα “GReen-OS”. Οι ενέργειες αυτές, που υλοποιούνται προγραμματιστικά με την μορφή διαδικασιών, είναι οι ακόλουθες:

- i) Διαδικασία 1: Χειροκίνητη πλοήγηση
- ii) Διαδικασία 2: Αυτόματη σπορά
- iii) Διαδικασία 3: Αυτόματη επιθεώρηση

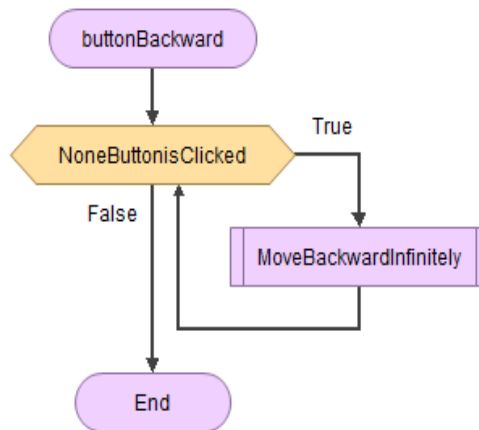
Η Διαδικασία 1, αυτή της χειροκίνητης πλοήγησης, αναλύεται σε 5 διαγράμματα ροής όπως φαίνεται στις παρακάτω εικόνες:

- Εικόνα 30 Διάγραμμα ροής 1α, συνεχής κίνηση μπροστά
- Εικόνα 31 Διάγραμμα ροής 1β, συνεχής κίνηση πίσω
- Εικόνα 32 Διάγραμμα ροής 1γ, στροφή αριστερά
- Εικόνα 33 Διάγραμμα ροής 1δ, στροφή δεξιά
- Εικόνα 34 Διάγραμμα ροής 1ε, διακοπή κίνησης

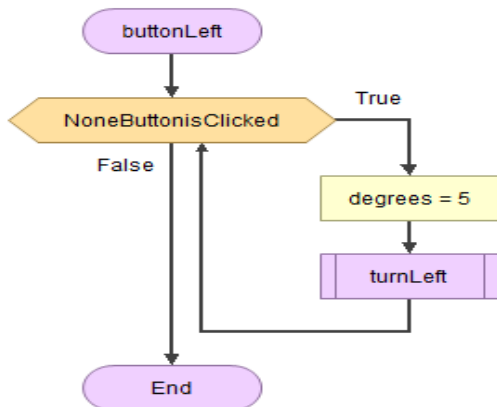
### Διαδικασία 1: χειροκίνητη πλοήγηση



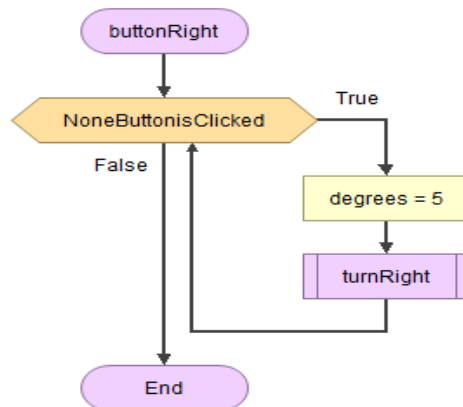
Εικόνα 30. Διάγραμμα ροής 1α, συνεχής κίνηση μπροστά



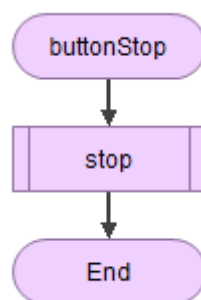
Εικόνα 31. Διάγραμμα ροής 1β, συνεχής κίνηση πίσω



Εικόνα 32. Διάγραμμα ροής 1γ, στροφή αριστερά



Εικόνα 33. Διάγραμμα ροής 1δ, στροφή δεξιά



Εικόνα 34. Διάγραμμα ροής 1ε, διακοπή κίνησης

### **Επεξήγηση Διαγραμμάτων ροής 1**

Διάγραμμα ροής 1α: Όταν πατηθεί το button Forward, το ρομπότ αρχίζει να κινείται μπροστά συνεχώς (με κάποια ταχύτητα που ρυθμίζεται), μέχρι να πατηθεί κάποιο άλλο button ή το button Stop.

Διάγραμμα ροής 1β: Όταν πατηθεί το button Backward, το ρομπότ αρχίζει να κινείται πίσω συνεχώς (με κάποια ταχύτητα που ρυθμίζεται), μέχρι να πατηθεί κάποιο άλλο button ή το button Stop.

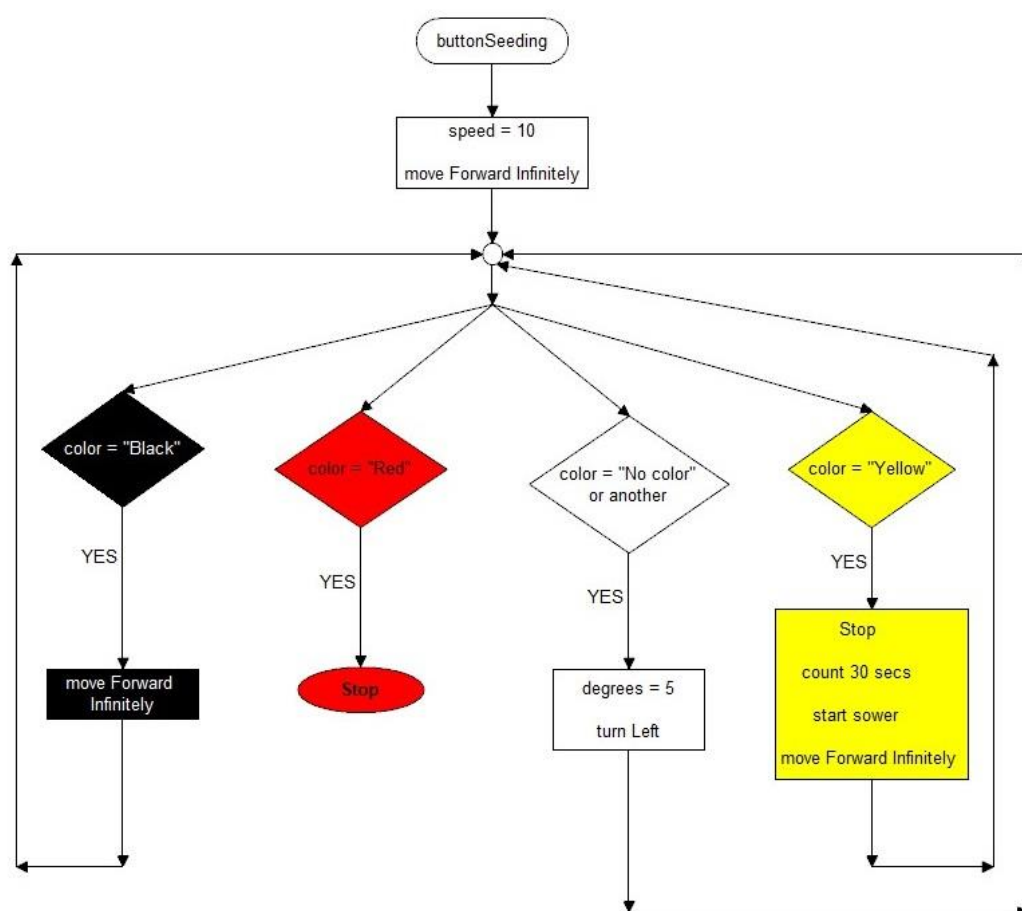
Διάγραμμα ροής 1γ: Όταν πατηθεί το button Left, το ρομπότ στρίβει κατά 5 μοίρες αριστερά και σταματάει, εκτός αν πατηθεί κάποιο άλλο button.

Διάγραμμα ροής 1δ: Όταν πατηθεί το button Right, το ρομπότ στρίβει κατά 5 μοίρες δεξιά και σταματάει, εκτός αν πατηθεί κάποιο άλλο button.

Διάγραμμα ροής 1ε: Όταν πατηθεί το button Stop, το ρομπότ σταματάει και τερματίζεται όποια λειτουργία έκανε πριν.

Η Διαδικασία 2, αυτή της αυτόματης σποράς (Διάγραμμα ροής 2), αναλύεται στην παρακάτω Εικόνα 35:

## Διαδικασία 2: αυτόματη σπορά



Εικόνα 35. Διάγραμμα ροής 2, αυτόματη σπορά ακριβείας

### Επεξήγηση Διαγράμματος ροής 2

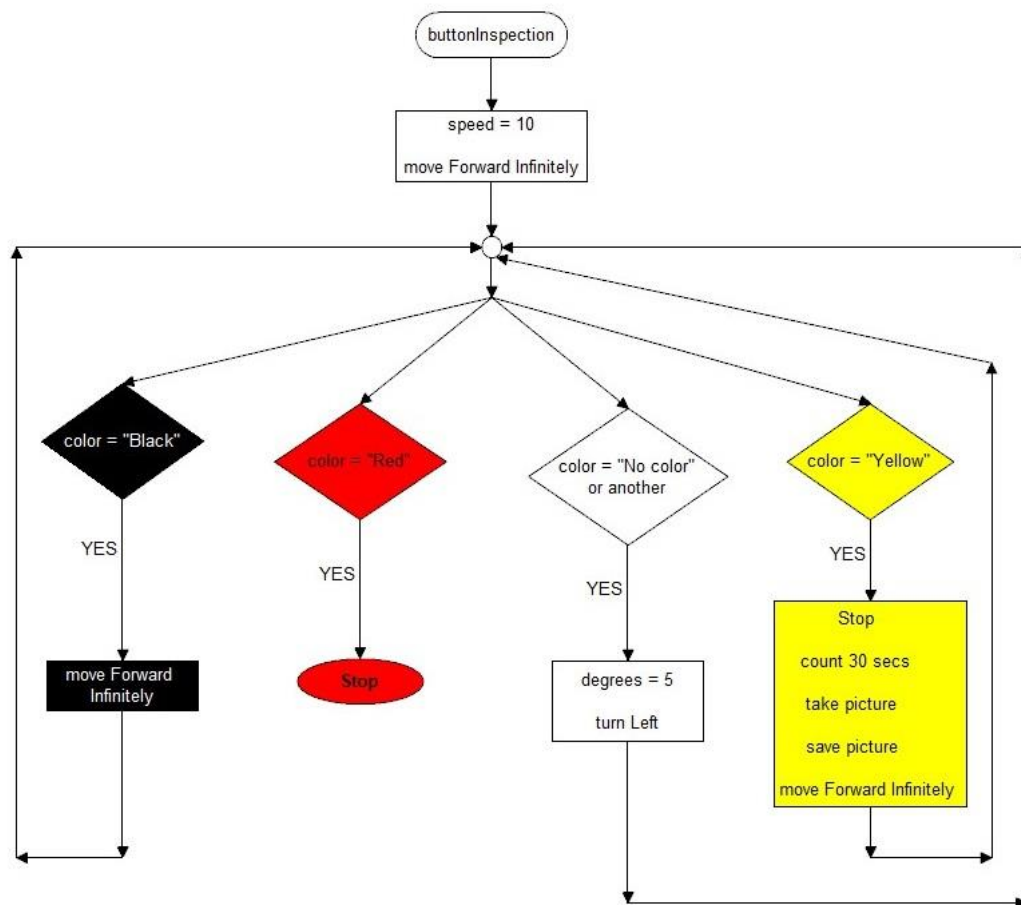
Αρχικά τοποθετούμε το ρομπότ στο σημείο έναρξης με τον αισθητήρα χρώματος να “βλέπει” τον μαύρο οδηγό. Όταν πατηθεί το button Seeding τότε ρυθμίζεται αυτόματα η ταχύτητα στην τιμή 10 και το ρομπότ αρχίζει να κινείται μπροστά συνεχώς. Όσο ο αισθητήρας χρώματος του ρομπότ βλέπει χρώμα “Black” το ρομπότ θα συνεχίσει να κινείται μπροστά συνεχώς παρακολουθώντας τον μαύρο οδηγό. Όσο εντοπίζει “No color” ή οποιοδήποτε άλλο χρώμα εκτός από “Yellow” ή “Red” ή “Black” θα στρίβει αριστερά κατά 5 μοίρες τη φορά ώστε να ξαναεντοπίσει τον μαύρο οδηγό. Αν εντοπίσει χρώμα “Yellow” υποτίθεται ότι βρίσκεται σε σημείο που πρέπει να γίνει σπορά, οπότε σταματάει, περιμένει 30 seconds, δουλεύει το μοτέρ του σπορέα, γίνεται σπορά και μετά συνεχίζει παρα-



κολουθώντας πάλι τον μαύρο οδηγό. Αν εντοπίσει χρώμα “Red” τότε υποτίθεται ότι έφθασε στο σημείο τερματισμού οπότε σταματάει.

Η Διαδικασία 3, αυτή της αυτόματης επιθεώρησης των φυτών (Διάγραμμα ροής 3), αναλύεται στην παρακάτω Εικόνα 36:

### Διαδικασία 3: αυτόματη επιθεώρηση



Εικόνα 36. Διάγραμμα ροής 3, αυτόματη επιθεώρηση (φωτογράφιση) φυτών

### **Επεξήγηση Διαγράμματος ροής 3**

Αρχικά τοποθετούμε το ρομπότ στο σημείο έναρξης με τον αισθητήρα χρώματος να “βλέπει” τον μαύρο οδηγό. Όταν πατηθεί το button Inspection τότε ρυθμίζεται αυτόματα η ταχύτητα στην τιμή 10 και το ρομπότ αρχίζει να κινείται μπροστά συνεχώς. Όσο ο αισθητήρας χρώματος του ρομπότ βλέπει χρώμα “Black” το ρομπότ θα συνεχίσει να κινείται μπροστά συνεχώς παρακολουθώντας τον μαύρο οδηγό. Όσο εντοπίζει “No color” ή οποιοδήποτε άλλο χρώμα εκτός από “Yellow” ή “Red” ή “Black” θα στρίβει αριστερά κατά 5 μοίρες τη φορά ώστε να ξαναεντοπίσει τον μαύρο οδηγό. Αν εντοπίσει χρώμα “Yellow” υποτίθεται ότι βρίσκεται σε σημείο που υπάρχει φυτό και πρέπει να τραβηχτεί φωτογραφία, οπότε σταματάει, περιμένει 30 seconds, κάνει capture το καρέ που “βλέπει” εκείνη τη στιγμή το smartphone που είναι στο ρομπότ και που παίζει τον ρόλο του camera Server, αποθηκεύει το καρέ που τραβήχτηκε στην sd card και μετά συνεχίζει παρακολουθώντας πάλι τον μαύρο οδηγό. Αν εντοπίσει χρώμα “Red” τότε υποτίθεται ότι έφθασε στο σημείο τερματισμού οπότε σταματάει.

## **3.3 Τι πετυχαίνουμε με το σύστημα**

### **3.3.1 Συγκεντρωτικός έλεγχος από μια φορητή συσκευή**

#### **Η IoT πλατφόρμα ως middleware**

Οι πλατφόρμες IoT έχουν τη μορφή ενδιάμεσου λογισμικού IoT, ο σκοπός του οποίου είναι να λειτουργήσει ως μεσολαβητής μεταξύ του υλικού και των επιπέδων της εφαρμογής. Οι πρωταρχικές του λειτουργίες περιλαμβάνουν τη συλλογή δεδομένων από τις συσκευές μέσω διαφορετικών πρωτοκόλλων και τοπολογιών δικτύου, τον απομακρυσμένο έλεγχο και την απομακρυσμένη ρύθμιση παραμέτρων της συσκευής, τη διαχείριση συσκευών και τις ενημερώσεις υλικολογισμικού over-the-air. Επιτυγχάνεται έτσι συγκεντρωτικός έλεγχος από ένα σημείο

για προσαρμογή των πρωτοκόλλων και των μοντέλων δεδομένων για τη συγκέντρωση πληροφοριών και τη διαχείριση των επικοινωνιών.

Για χρήση σε ετερογενή οικοσυστήματα IoT του πραγματικού κόσμου, το ενδιάμεσο λογισμικό του IoT αναμένεται να υποστηρίξει την ολοκλήρωση με σχεδόν οποιαδήποτε συνδεδεμένη συσκευή και να συνδυαστεί με εφαρμογές τρίτων που χρησιμοποιεί η συσκευή. Αυτή η ανεξαρτησία από το υλικό που βρίσκεται "από κάτω" και από το λογισμικό που βρίσκεται "από πάνω" δίνει την δυνατότητα σε μια ενιαία πλατφόρμα IoT να διαχειρίζεται κάθε είδους συνδεδεμένη συσκευή με τον ίδιο απλό τρόπο. (Εικόνα 37)



Εικόνα 37. Τα δομικά στοιχεία και η αρχιτεκτονική μιας IoT πλατφόρμας

Στην περίπτωσή μας, δημιουργήθηκε η IoT πλατφόρμα **“GReen-OS”** με την βοήθεια της οποίας ελέγχουμε την κίνηση του EV3 based τροχήλατου ρομπότ **“farmbot-Si”** (για αυτόματη και χειροκίνητη πλοήγηση, σπορά ακριβείας, επιθεώρηση φυτών), ελέγχουμε πλήρως την μετάδοση και αποθήκευση όλων των δεδομένων μεταξύ των συσκευών ρομπότ – smart device – υπολογιστικό σύστημα,

ενημερωνόμαστε για την υγεία των φυτών και την πορεία της καλλιέργειας, δεχόμαστε συμβουλές θεραπείας ασθενειών και δημιουργούμε ανάλογες αναφορές.

### **3.3.2 Άμεση ενημέρωση του αγρότη**

Η σημερινή γεωργία βρίσκεται σε αγώνα δρόμου. Οι γεωργοί πρέπει να αυξήσουν περισσότερο το προϊόν στο επιδεινούμενο έδαφος, στη μειωμένη διαθεσιμότητα της γης και στην αυξημένη διακύμανση των καιρικών συνθηκών. Η γεωργία με τη χρήση του IoT επιτρέπει στους αγρότες να παρακολουθούν το προϊόν τους και τις περιβαλλοντικές συνθήκες του σε πραγματικό χρόνο. Λαμβάνουν γρήγορα γνώσεις από μέσα, μπορούν να προβλέψουν ζητήματα πριν συμβούν και να λάβουν τεκμηριωμένες αποφάσεις για το πώς να τις αποφύγουν. Επιπλέον, οι λύσεις IoT στη γεωργία εισάγουν αυτοματοποίηση, για παράδειγμα, η άρδευση με βάση τη ζήτηση, η λίπανση και η συγκομιδή από ρομπότ.

Ένα από τα οφέλη της χρήσης του IoT στη γεωργία είναι η αυξημένη ευελιξία των διαδικασιών. Χάρη σε συστήματα παρακολούθησης και πρόβλεψης σε πραγματικό χρόνο, οι αγρότες μπορούν να ανταποκριθούν γρήγορα σε κάθε σημαντική αλλαγή του καιρού, της υγρασίας, της ποιότητας του αέρα καθώς και της υγείας κάθε φυτού ή του εδάφους. Στις συνθήκες ακραίων καιρικών αλλαγών, οι νέες δυνατότητες βοηθούν τους επαγγελματίες της γεωργίας να σώσουν τις καλλιέργειες.

Χάρη στην IoT πλατφόρμα "GReen-OS", ο υπεύθυνος της εκμετάλλευσης μπορεί να βλέπει στο smartphone του μια αναφορά με την κατάσταση της υγείας των φυτών σε πραγματικό χρόνο και έτσι να πάρει αμέσως αποφάσεις για τη βελτίωση της παραγωγικότητας και επομένως της κερδοφορίας της καλλιέργειας. Μπορεί να έχει στην διάθεσή του εγκαίρως τα απαραίτητα διορθωτικά μέτρα για την θεραπεία αρχομένων ασθενειών που εντοπίστηκαν αυτοματοποιημένα και με εγκυρότητα, να δει στατιστικά που αφορούν την πορεία της καλλιέργειας και να προσδιορίσει χρόνο και ποσότητα συγκομιδής, χωρίς να είναι απαραίτητη η φυσική του παρουσία στο χώρο του θερμοκηπίου.

### 3.3.3 Προγραμματισμός συγκομιδής

Οι ιδιοκτήτες και οι διαχειριστές αγροκτημάτων δείχνουν ενδιαφέρον και εντυπωσιάζονται από το μέγεθος των μετρήσεων που μπορούν να λάβουν οι αισθητήρες και από το πόσο ικανά είναι τα συστήματα πρόγνωσης και διαχείρισης.

Οι εφαρμογές έξυπνης γεωργίας παρέχουν πολλά οφέλη σε διάφορες γεωργικές δραστηριότητες όπως στον ακριβή προγραμματισμό της συγκομιδής. Για να αποκομίσουν τα μέγιστα οφέλη από τις καλλιέργειες, οι αγρότες πρέπει να γνωρίζουν πότε το οικόπεδο είναι έτοιμο για συγκομιδή. Με αυτόν τον τρόπο, οι αγρότες μπορούν να προγραμματίσουν συγκομιδή για την κάθε καλλιέργεια.

Οι τεχνικές μετά τη συγκομιδή και παρόμοιες τεχνικές θα μπορούσαν να βοηθήσουν στη βιομηχανία τροφίμων, βελτιστοποιώντας την επεξεργασία, μειώνοντας το κόστος και περιορίζοντας την αλλοίωση καθώς το προϊόν κατευθύνεται προς τον καταναλωτή. Η απόδοση της επένδυσης (Return on Investment, RoI) σε συστήματα γεωργίας ακριβείας μπορεί να ποσοτικοποιηθεί με βάση πραγματικά νούμερα. Στην πραγματικότητα, η αγορά αρχίζει να παρουσιάζει μελέτες περίπτωσης στις οποίες το RoI είναι πραγματικό.

Με την βοήθεια της εφαρμογής “**GReen-OS**” ο υπεύθυνος του θερμοκηπίου είναι σε θέση να γνωρίζει το ακριβές χρονικό σημείο της συνολικής διάρκειας της καλλιέργειας καθώς και την απόδοση της καλλιέργειας σε αυτή τη χρονική στιγμή. Τα δεδομένα αυτά μπορούν να συνδυαστούν για την εκτίμηση της καταλληλότερης ημερομηνίας συγκομιδής με την παρακάτω μορφή:

- Ημερομηνία σποράς: .....
- Ημερομηνία αναφοράς: .....
- Τρέχουσα εβδομάδα: πχ 10<sup>η</sup> από 16
- Αριθμός υγιών φυτών: .....
- Ποσοστό υγιών φυτών: .....
- Εκτιμώμενη ημερομηνία συγκομιδής : .....
- Εκτιμώμενη ποσότητα συγκομιδής: .....

Με αυτή την ενημέρωση στη διάθεσή του ο υπεύθυνος του θερμοκηπίου μπορεί να επικοινωνήσει εγκαίρως με τους αρμόδιους για την παραλαβή του τελικού προϊόντος ώστε αυτό να διατεθεί στους καταναλωτές στην πιο

φρέσκια μορφή του. Τα δεδομένα αυτά μπορούν να αποθηκευτούν με την μορφή ιστορικού ώστε να αξιοποιούνται για την δημιουργία στατιστικών.

### 3.3.4 Απομακρυσμένη εποπτεία καλλιέργειας

Τα θερμοκήπια χρειάζονται συχνή παρακολούθηση, προκειμένου να διασφαλιστεί ότι οποιαδήποτε βλάβη εντοπίζεται αμέσως και ότι μια σειρά από συνθήκες είναι συνεχώς υπό έλεγχο.

Με την βοήθεια της εφαρμογής “**GReen-OS**” ο υπεύθυνος του θερμοκηπίου έχει την δυνατότητα να παρακολουθεί από απόσταση και μέσα από το κινητό του τις περιβαλλοντικές συνθήκες (θερμοκρασία - υγρασία) που επικρατούν στο χώρο και να παίρνει εγκαίρως τις ανάλογες αποφάσεις σε πραγματικό χρόνο και από οπουδήποτε. Αυτά τα περιβαλλοντικά δεδομένα μπορεί να τα χρησιμοποιήσει για να εντοπίσει πιθανές αιτίες για τις εμφανιζόμενες ασθένειες των φυτών και έτσι να βελτιώσει τις συνθήκες ανάπτυξής τους. Έχει επίσης την δυνατότητα για εύελικτη οπτικοποίηση των δεδομένων για παρακολούθηση μελλοντικών καλλιεργειών σε πραγματικό χρόνο αλλά και με διατήρηση ιστορικού.

### 3.3.5 Αυτοματοποιημένη σπορά ακριβείας

Η γεωργία ακριβείας είναι μια γεωργική στρατηγική που περιλαμβάνει βελτιστοποίηση των αποδόσεων των εισροών και ποικίλες μεθόδους καλλιέργειας, συμπεριλαμβανομένου του οργώματος, της φύτευσης, της συγκομιδής και της διαχείρισης λιπασμάτων, σπόρων, αποβλήτων και φυτοφαρμάκων για την κάλυψη των συνθηκών καλλιέργειας σε ολόκληρη την καλλιεργούμενη έκταση.

Η γεωργία ακριβείας περιλαμβάνει κυρίως πολύπλοκες διαδικασίες που μπορούν να υλοποιηθούν μόνο με τη χρήση μηχανών.

**Κέρδος για το περιβάλλον:** προωθώντας τη βιοποικιλότητα, εξαλείφοντας τη συντριπτική πλειοψηφία χημικών αποβλήτων και εισροών πόρων, καθώς και τις ζημιές στα χωράφια που προκαλούνται από τους ελκυστήρες (τα ρομπότ είναι πολύ ελαφρύτερα απ' ό,τι για παράδειγμα ένα τρακτέρ John Deere).

**Κέρδος για τον αγρότη:** μειώνοντας το κόστος έως και 60% (λιγότερο δαπανηρές εισροές, όπως χημικές ουσίες και λιπάσματα) ... συν αυτά τα ρομπότ επιτελούν πολλά εργατικά καθήκοντα όπως το ξεχορτάρισμα - θεωρητικά απελευθερώνοντας τον χρόνο των αγροτών για άλλες εργασίες.

## Πλεονεκτήματα των ρομπότ γεωργίας ακριβείας

Τα κυριότερα πλεονεκτήματα των ρομπότ γεωργίας ακριβείας είναι τα ακόλουθα:

- Ελαφρύ βάρος
- Εξοικονόμηση εργασίας
- Μείωση των εκπομπών διοξειδίου του άνθρακα
- Αύξηση της απόδοσης της καλλιέργειας
- Οικονομικά στη χρήση

Τα πλήρως αυτόνομα ρομποτικά συστήματα που μοιάζουν με ελκυστήρες εκτελούν λειτουργίες όπως η δημιουργία συμπαγούς εδάφους και η μείωση του πορώδους του εδάφους. Το ρομπότ πλοηγείται στην καλλιεργούμενη έκταση και υλοποιεί την εργασία που του έχει ανατεθεί. Βοηθάει τους αγρότες να αφαιρούν ζιζάνια, να χρησιμοποιούν χημικά ή να συγκομίζουν τα φυτά και έχει ενσωματωμένα συστήματα πλοήγησης για να εντοπίζει την θέση του.

Το συγκεκριμένο **“farmbot-Si”** ρομπότ έχει την δυνατότητα να πλοηγείται αυτόνομα εντός του θερμοκηπίου, χρησιμοποιώντας αισθητήρα χρώματος για να παρακολουθεί οδηγούς που βρίσκονται στο έδαφος και επιτελεί αυτόματα σπορά ακριβείας. Η διαδικασία ξεκινάει με παρέμβαση του χρήστη με το πάτημα ενός button στην εφαρμογή **“GReen-OS”**. Στα σημεία που πρέπει να γίνει σπορά υπάρχουν σημάδια κίτρινου χρώματος επάνω στους οδηγούς. Μόλις αυτά τα σημεία γίνουν αντιληπτά από τον αισθητήρα χρώματος το ρομπότ σταματάει, ξεκινάει να λειτουργεί ο σπορέας, πέφτει ο σπόρος, σταματάει ο σπορέας, το ρομπότ συνεχίζει την κίνησή του και έτσι γίνεται σπορά ανά 40 cm ή σε οποιοδήποτε άλλο σημείο επιλέξει ο υπεύθυνος του θερμοκηπίου να τοποθετήσει κίτρινο σημάδι. Η διαδικασία σποράς ολοκληρώνεται όταν εντοπιστεί σημάδι κόκκινου χρώματος. Με την βοήθεια της συγκεκριμένης εφαρμογής μπορεί να γίνει σπορά σε σειρές οποιουδήποτε μήκους και σχήματος χωρίς να απαιτείται οποιαδήποτε αλλαγή στον προγραμματισμό του ρομπότ.

### 3.3.6 Δυνατότητα έγκαιρου και έγκυρου εντοπισμού αρχομένων ασθενειών των φυτών

Με την βοήθεια του Diagnosis Support System (DiSS) μπορεί αυτοματοποιημένα να γίνει διάγνωση της κατάστασης των φυτών. Η διάγνωση βασίζεται στις φωτογραφίες των φυτών που τραβιούνται επίσης αυτόματα από το “**farmbot-Si**” ρομπότ και κατόπιν στέλνονται στο έμπειρο σύστημα. Εκεί γίνεται ο διαχωρισμός που κατατάσσει τα φυτά σε υγιή ή ασθενικά και για τα ασθενικά εντοπίζεται και αναφέρεται και η ασθένεια από την οποία πάσχουν. Το συγκεκριμένο έμπειρο σύστημα μπορεί να αναγνωρίσει σημάδια που ανήκουν σε τρεις (3) ασθένειες: a) “bacterial spot”, b) “yellow leaf curl virus”, c) “late blight”.

### 3.3.7 Παροχή κατάλληλων συμβουλών θεραπείας των ασθενειών

Αφού γίνει η διάγνωση της ασθένειας από το έμπειρο σύστημα προτείνονται αυτοματοποιημένα τα κατάλληλα εξειδικευμένα διορθωτικά μέτρα που πρέπει να ληφθούν ώστε να αντιμετωπιστεί εγκαίρως η συγκεκριμένη ασθένεια. Το όνομα της ασθένειας και τα διορθωτικά μέτρα είναι άμεσα διαθέσιμα στον υπεύθυνο του θερμοκηπίου μέσα από την εφαρμογή “**GReen-OS**” με την μορφή αναφοράς. Η ενσωμάτωση των διορθωτικών μέτρων για κάθε ασθένεια στο Diagnosis Support System (DiSS) καθώς και των κατάλληλων δεδομένων για την δημιουργία του διαγνωστικού μοντέλου αποτελεί προϊόν συνεργασίας με ειδικούς επιστήμονες (γεωπόνους) που μετέδωσαν την τεχνογνωσία τους και εξασφάλισαν έτσι την ακρίβεια, την εγκυρότητα και την αξιοπιστία του έμπειρου συστήματος.

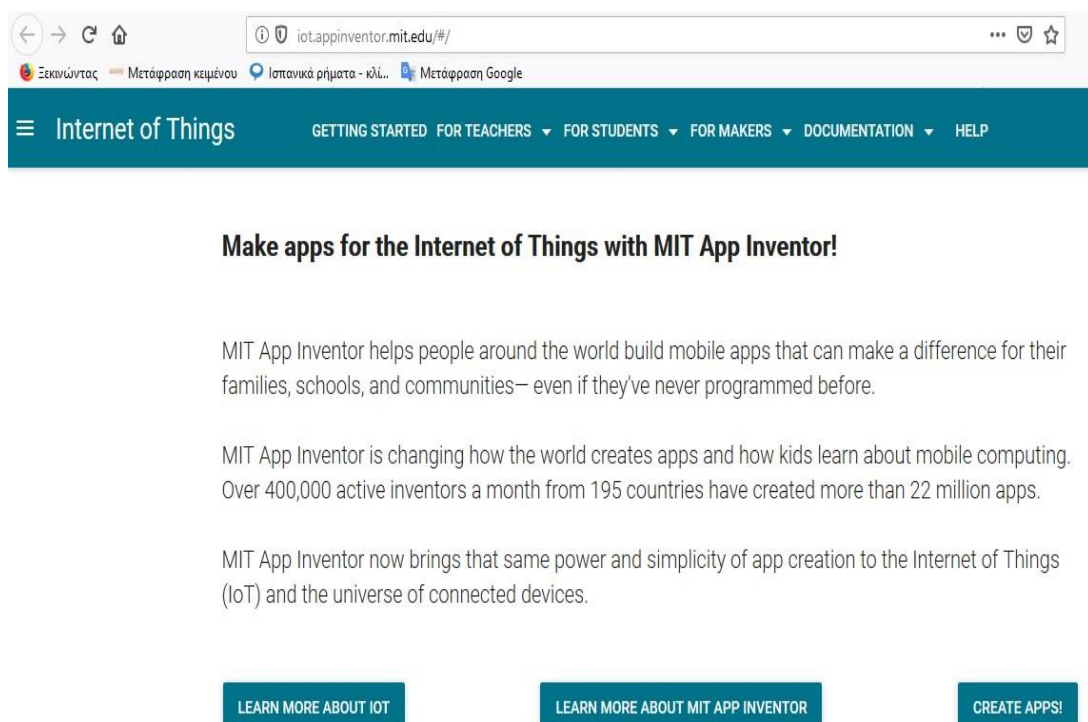


## 4 Εργαλεία υλοποίησης

Παρακάτω αναφέρονται τα εργαλεία λογισμικού που χρησιμοποιήθηκαν για να υλοποιηθούν οι επιμέρους λειτουργίες που επιτελεί το προτεινόμενο σύστημα, παρουσιάζονται οι απαραίτητες ρυθμίσεις ώστε να συνεργάζονται μεταξύ τους τα υποσυστήματα και περιγράφονται οι εξωτερικές (third-party) εφαρμογές που απαιτήθηκαν.

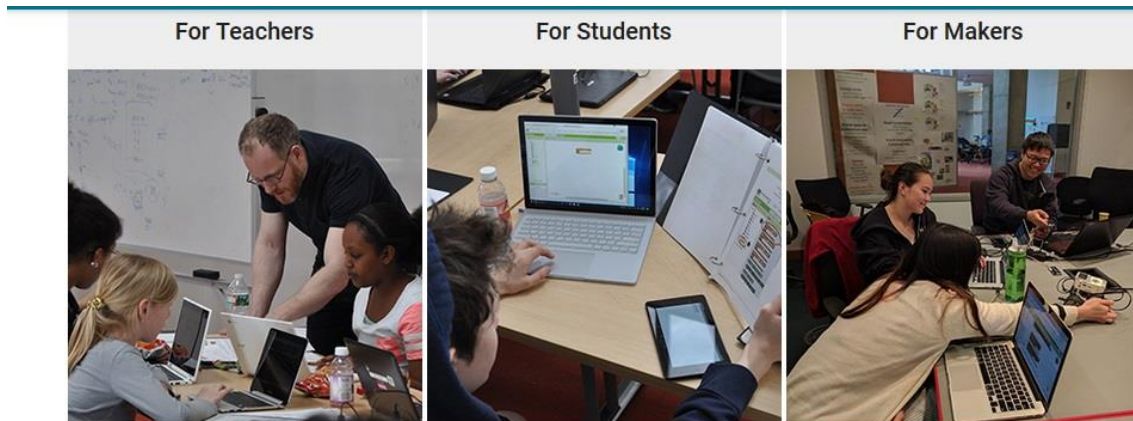
### 4.1 Προγραμματισμός της android εφαρμογής “GReen-OS”

Το “GReen-OS” είναι μια εφαρμογή για smartphones και tablets με λειτουργικό σύστημα android. Δημιουργήθηκε με την βοήθεια του προγραμματιστικού εργαλείου **MIT APP INVENTOR** που είναι ένα ελεύθερο, διαδικτυακό και οπτικό προγραμματιστικό περιβάλλον με πλακίδια (blocks) το οποίο το παρέχει το Massachusetts Institute of Technology (MIT). Το app inventor πρόσφατα εισήλθε και στον χώρο του Internet of Things (Εικόνα 38)



Εικόνα 38. Η πλατφόρμα του app inventor για το Internet of Things

Με την πλατφόρμα αυτή μπορούν να συνδεθούν πολλές συσκευές του πραγματικού κόσμου μεταξύ τους και να αλληλεπιδρούν. (Εικόνα 39)



### **Bring computing off the screen and into the world of everyday things!**

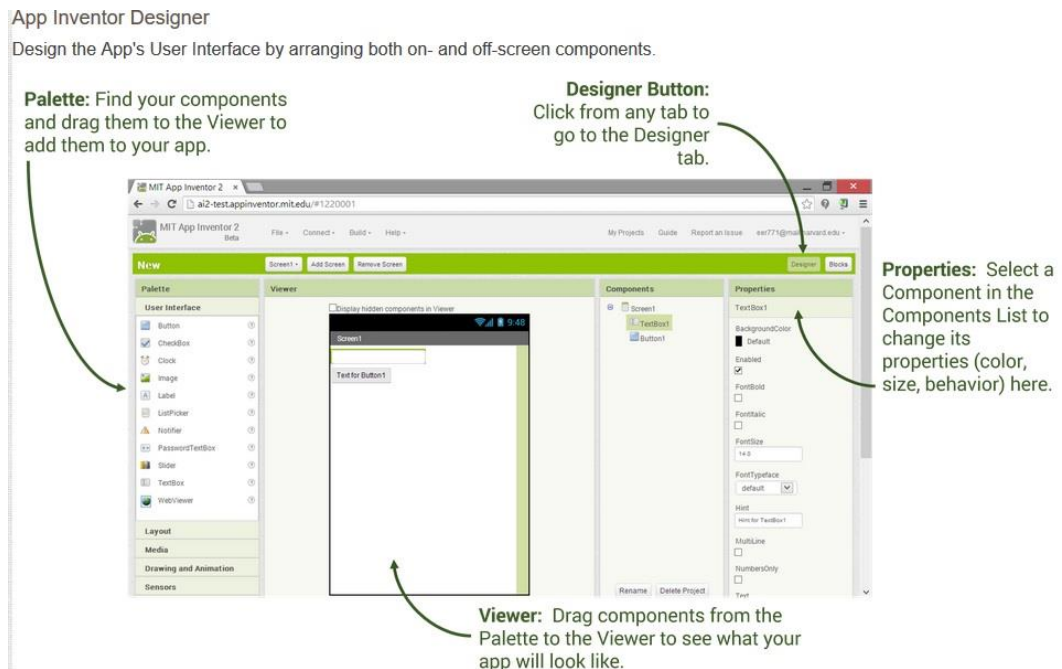
You can develop IoT applications the same way you develop any other MIT App Inventor mobile apps, but now you can use the apps to interact with objects all around you.

Εικόνα 39. Αλληλεπίδραση με συσκευές του πραγματικού κόσμου

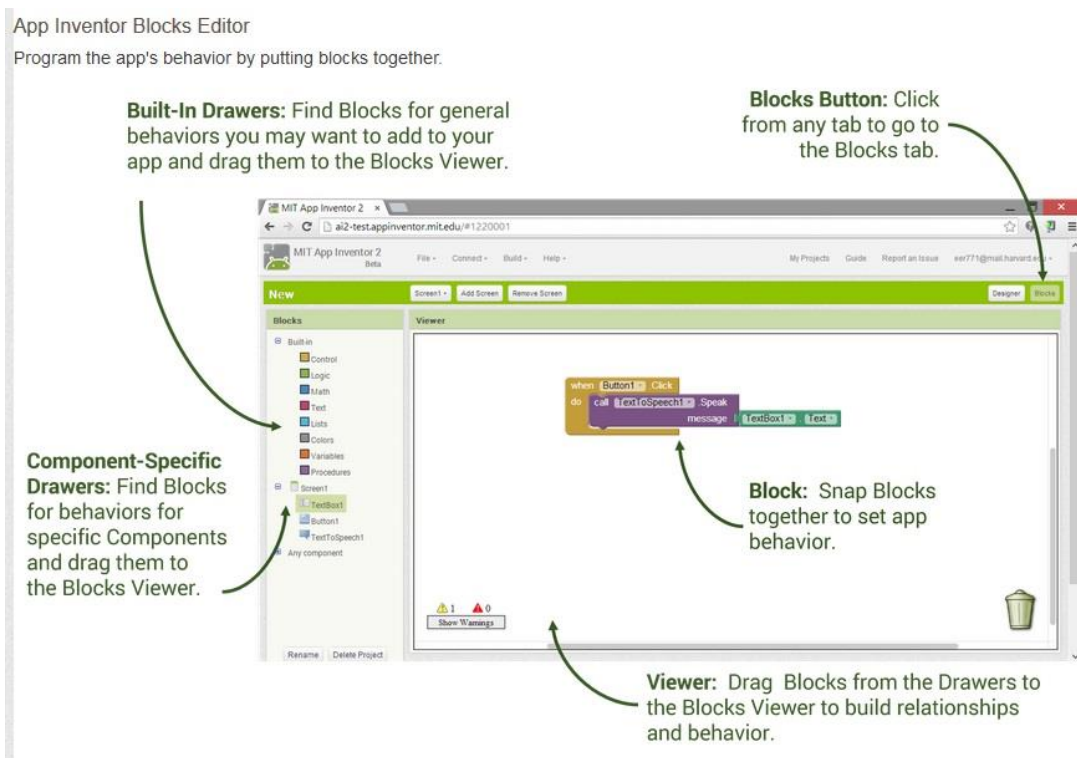
Ακόμα και ένας αρχάριος χρήστης μπορεί να συνδεθεί στο App Inventor και με διαδικασία «σύρε και άφησε» να συνδυάσει πλακίδια και να αναπτύξει εφαρμογές για φορητές συσκευές με λειτουργικό σύστημα android. Τα πλακίδια ενώνονται μόνο όταν προκύπτει συντακτικά σωστό πρόγραμμα και η τελική εφαρμογή μπορεί να εκτελεστεί και να δοκιμαστεί είτε απευθείας σε συσκευή που είναι συνδεδεμένη με τον υπολογιστή του χρήστη (ενσύρματα με USB ή ασύρματα με WiFi) είτε σε ενσωματωμένο emulator (προσομοιωτή κινητού τηλεφώνου).

Η κλασική δομή του περιβάλλοντος του App Inventor αποτελείται από: (α) τον Designer (Σχεδιαστή) (Εικόνα 40), όπου ο χρήστης επιλέγει τα συστατικά μέρη για την εφαρμογή που αναπτύσσει, και (β) τον Blocks Editor (Συντάκτη πλακιδίων) (Εικόνα 41), όπου ο χρήστης συνδυάζει οπτικά τα πλακίδια του προγράμματος, για να ορίσει τη συμπεριφορά των μερών της εφαρμογής (μοιάζει με τη συναρμολόγηση ενός

πάζλ). Τα πλακίδια είναι ταξινομημένα σε διαφορετικά χρώματα ανάλογα με τη λειτουργία που επιτελούν.



Εικόνα 40. Το περιβάλλον του Designer (Σχεδιαστή)



Εικόνα 41. Το περιβάλλον του Blocks Editor (Συντάκτη Πλακιδίων)

## Διαδικασία δημιουργίας μιας εφαρμογής στο App Inventor

1. Αρχικά, επισκεπτόμαστε τον επίσημο ιστότοπο του App Inventor, ο οποίος περιέχει υλικό στην αγγλική γλώσσα με υποστηρικτικές οδηγίες, οδηγούς εκμάθησης, βιβλιοθήκες, ομάδες συζητήσεων κ.ά.
2. Για να έχουμε δικαίωμα πρόσβασης στο προγραμματιστικό περιβάλλον, θα πρέπει να διαθέτουμε λογαριασμό στην Google. Για όσους δεν έχουν λογαριασμό, η εγγραφή είναι εύκολη και δωρεάν. Επιλέγουμε τον σύνδεσμο Create και στο παράθυρο που μας ανοίγει κάνουμε είσοδο με τα στοιχεία του λογαριασμού μας.
3. Στην αρχική οθόνη που εμφανίζεται επιλέγουμε New Project (νέο έργο), οπότε και μας ζητείται να δώσουμε ένα όνομα για την εφαρμογή που πρόκειται να δημιουργήσουμε.
4. Ανοίγει η καρτέλα Designer για να σχεδιάσουμε την εμφάνιση της εφαρμογής μας επιλέγοντας τα απαραίτητα συστατικά στοιχεία και ορίζοντας ιδιότητες γι' αυτά.

Ο Designer αποτελείται από τα παρακάτω κύρια πλαίσια:

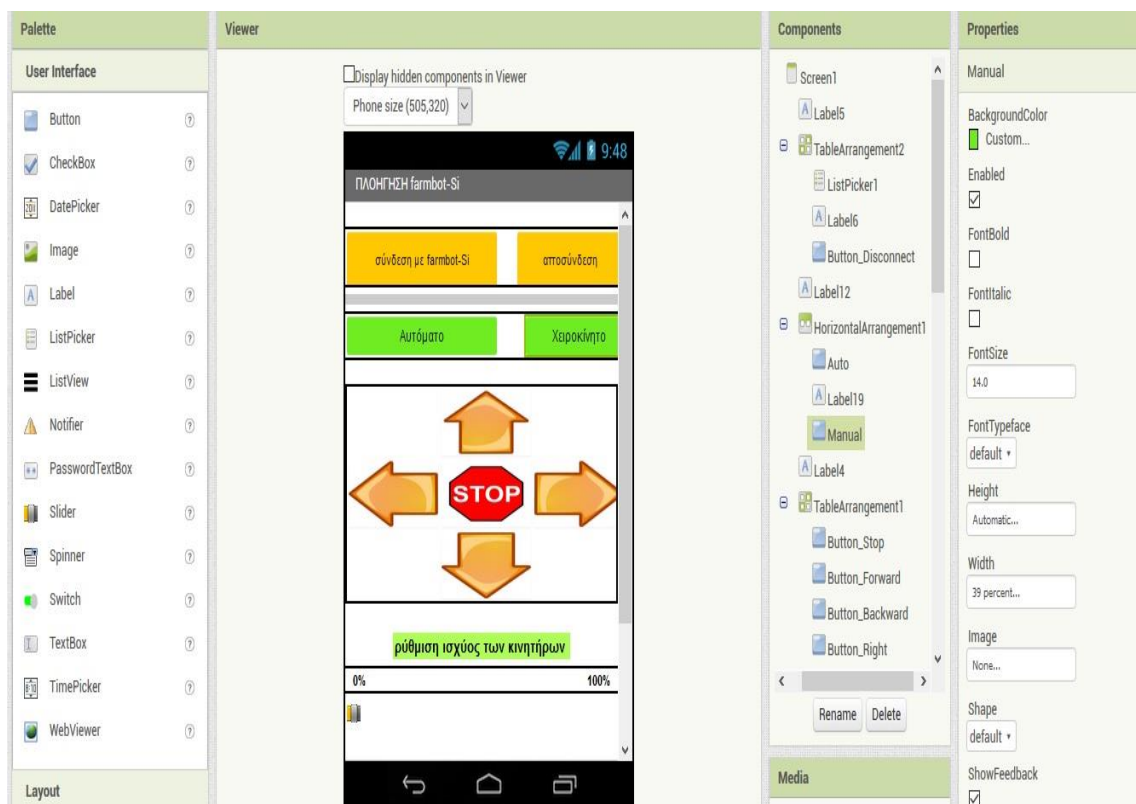
- Palette (συλλογή συστατικών): περιέχει όλα τα στοιχεία, χωρισμένα σε κατηγορίες (User Interface, Layout, Media κ.ά.) που μπορούμε να εισάγουμε στην εφαρμογή μας με απλό σύρσιμο.
  - Viewer (οθόνη συσκευής): εδώ τοποθετούμε στη θέση που θέλουμε τα συστατικά στοιχεία της εφαρμογής με τη διαδικασία «σύρε και άφησε» από το πλαίσιο Palette.
  - Components (επιλεγμένα συστατικά): μια δενδροειδής δομή των στοιχείων που έχουμε επιλέξει.
  - Properties (ιδιότητες): το πλαίσιο παραμετροποίησης του κάθε συστατικού (π.χ. χρώμα, μέγεθος, συμπεριφορά).
5. Μόλις ολοκληρώσουμε τη σχεδίαση της εφαρμογής μας και την παραμετροποίηση των συστατικών της μέσω των ιδιοτήτων τους, ανοίγουμε την καρτέλα Blocks. Ο προγραμματισμός γίνεται στο πλαίσιο Viewer (**Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**), όπου σύρουμε από το πλαίσιο Blocks τα κατάλληλα πλακίδια και τα συνδυάζουμε, για να ορίσουμε τις συμπεριφορές και τις συσχετίσεις της εφαρμο-

γής μας. Τα πλακίδια είναι χρωματιστά και χωρίζονται σε δύο κατηγορίες: α) τα ενσωματωμένα (Built-in), που ορίζουν γενικές συμπεριφορές στην εφαρμογή μας και β) τα σχετικά με συγκεκριμένα συστατικά της εφαρμογής, που ορίζουν συμπεριφορές γι' αυτά.

6. Στο τελευταίο βήμα εγκαθιστούμε και ελέγχουμε την εφαρμογή μας με σύνδεση σε κάποια φορητή συσκευή. Επιλέγουμε από το μενού Connect: AI Companion για σύνδεση μέσω WiFi, με την απαραίτητη προϋπόθεση να το έχουμε πρώτα εγκαταστήσει στη συσκευή μας ή USB για ενσύρματη σύνδεση ή Emulator για προσομοίωση φορητής συσκευής στον υπολογιστή μας.

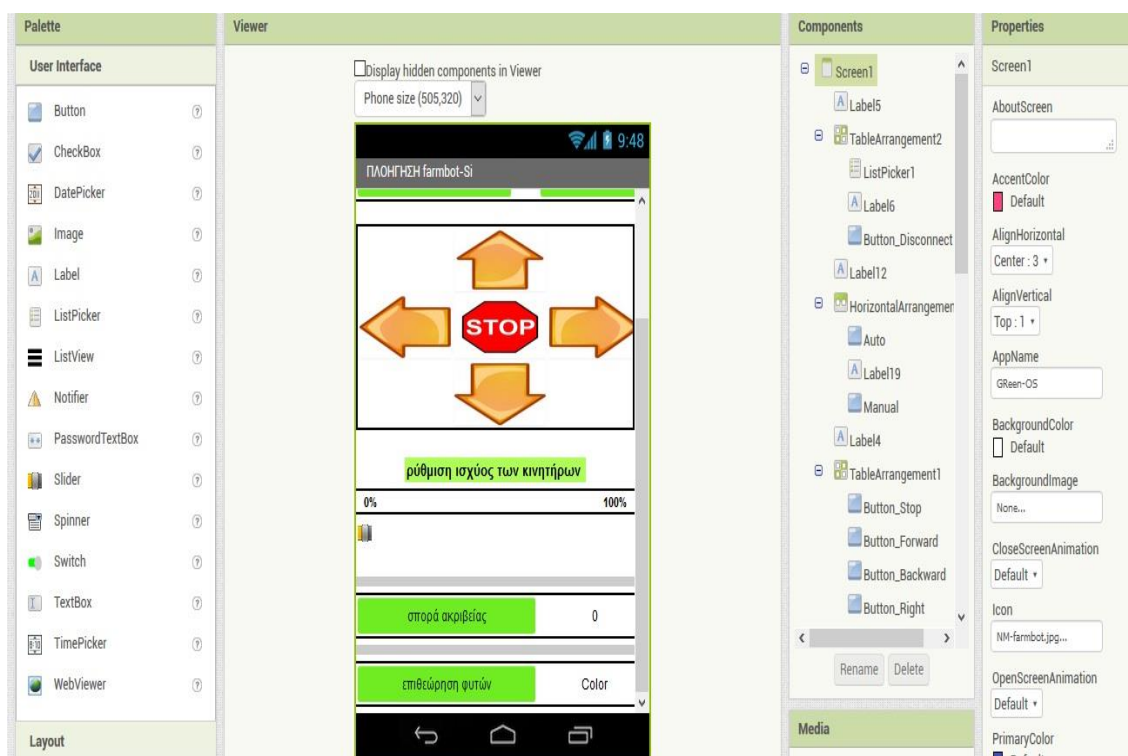
Ακολουθώντας την παραπάνω διαδικασία υλοποιήθηκε η εφαρμογή “GReen-OS” όπως φαίνεται στις παρακάτω εικόνες. Στην (Εικόνα 42) απεικονίζεται το screen με τα buttons για την δημιουργία ζεύγους bluetooth, ενώ στην (Εικόνα 43) διακρίνονται τα buttons για την αυτόματη σπορά ακριβείας και την αυτόματη επιθεώρηση των φυτών.

Έγινε προσπάθεια ώστε να δημιουργηθεί το ανάλογο απλό και φιλικό human-robot user interface.



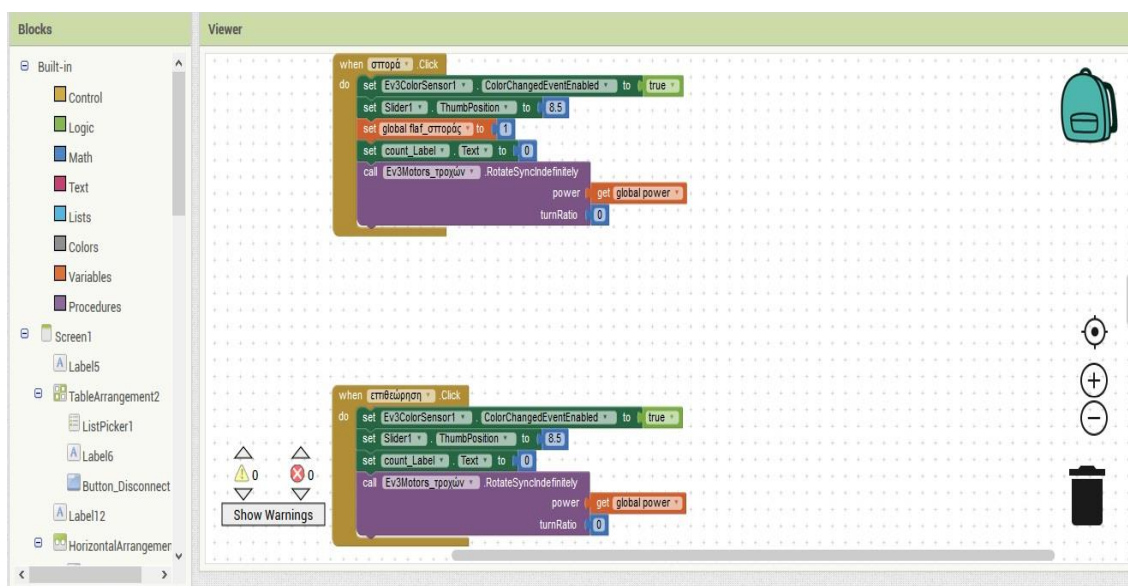
Εικόνα 42. Το screen με τα buttons για την δημιουργία ζεύγους bluetooth





Εικόνα 43. Το screen για αυτόματη σπορά ακριβείας και επιθεώρηση των φυτών

Παρακάτω δίνονται και κάποια τμήματα του κώδικα με την μορφή blocks. (Εικόνα 44)



Εικόνα 44. Τα blocks για την αυτόματη σπορά ακριβείας και την επιθεώρηση των φυτών

Η εφαρμογή “GReen-OS” που δημιουργήθηκε έχει σχετικά μικρό μέγεθος (μικρότερο από 20 MB) και μπορεί να “τρέξει” άνετα και σε smartphones ή tablets με σχετικά μικρή υπολογιστική ισχύ. (Εικόνα 45)



Εικόνα 45. Η εφαρμογή "GReen-OS" εγκατεστημένη σε smartphone

## 4.2 Προγραμματισμός-Ρυθμίσεις της μονάδας EV3

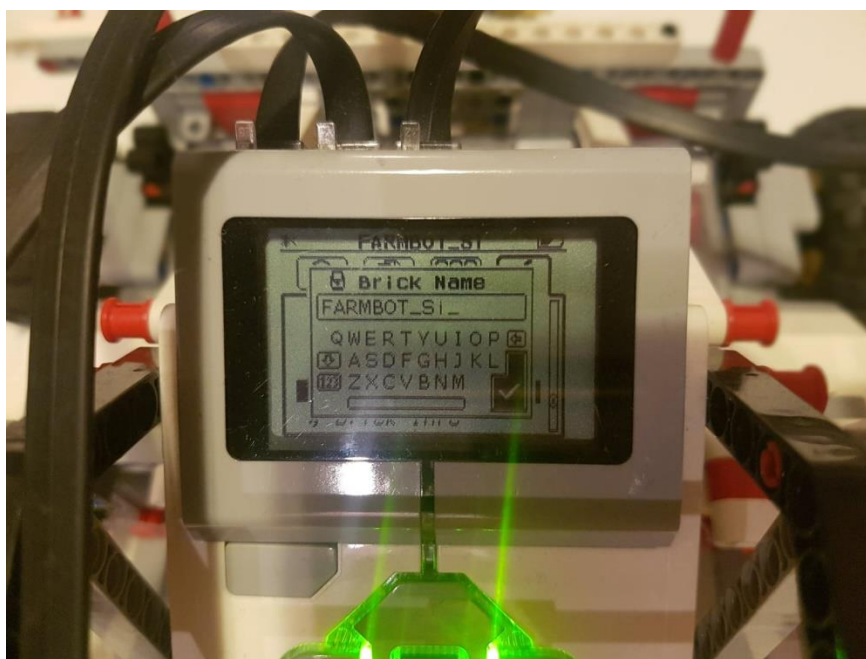
Η πλοήγηση του ρομπότ “farmbot-Si” εντός του θερμοκηπίου, είτε χειροκίνητη είτε αυτόματη, προϋποθέτει την ασύρματη επικοινωνία μεταξύ του smartphone και του ρομπότ. Αυτή η επικοινωνία εξασφαλίζεται μέσω ζεύγους bluetooth. Ο κώδικας για την πλοήγηση “τρέχει” στην εφαρμογή “GReen-OS” στο smartphone και οι εντολές μεταφέρονται μέσω bluetooth στην προγραμματιζόμενη μονάδα EV3 και στην συνέχεια στους κινητήρες που κινούν το ρομπότ.

Για να δημιουργηθεί το ζεύγος θα πρέπει να δώσουμε ένα όνομα στη μονάδα EV3. Αυτό γίνεται επιλέγοντας από το menu την ρύθμιση brick name (Εικόνα 46) και γρά-

φοντας το όνομα “farmbot-Si” (Εικόνα 47). Επίσης ενεργοποιούμε και την θύρα bluetooth (Εικόνα 48). Με τις ρυθμίσεις αυτές το EV3 ρομπότ είναι ανιχνεύσιμο και διαθέσιμο για την δημιουργία ζεύγους.

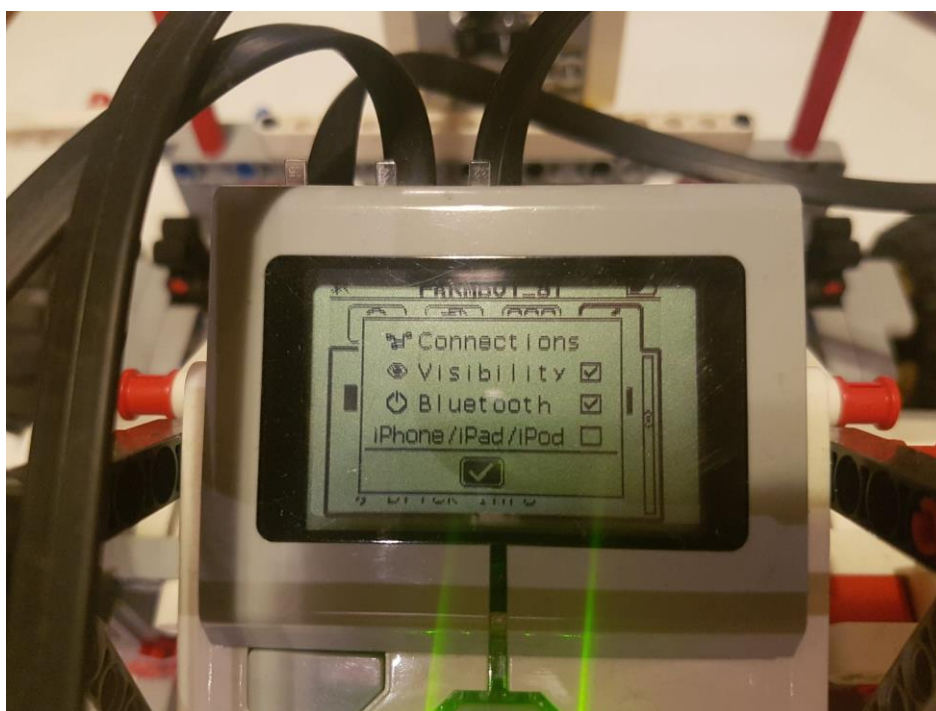


Εικόνα 46. Η ρύθμιση για το brick name



Εικόνα 47. Η οθόνη πληκτρολόγησης του ονόματος





Εικόνα 48. Η οθόνη ενεργοποίησης της θύρας bluetooth

### 4.3 Προγραμματισμός της πλακέτας ESP 8266 ως REST Web Server

Οι arduino based wifi πλακέτες ESP8266 γίνονται όλο και πιο δημοφιλείς μεταξύ των κατασκευαστών λόγω του χαμηλού κόστους και του ισχυρού on-board προγραμματιζόμενου μικροελεγκτή.

Το κόστος είναι μια τάξη μικρότερο από τις λύσεις που χρησιμοποιήθηκαν πιο παλιά, συμπεριλαμβανομένου του Arduino + Wifi Shield ή ενός Arduino Yun.

Συγκρίνοντας το ESP με άλλες λύσεις Wi-Fi στην αγορά, είναι μια εξαιρετική επιλογή για τα περισσότερα έργα "Internet of Things". Κοστίζει μόνο λίγα δολάρια και μπορεί να ενσωματωθεί σε προηγμένα έργα. Επιπλέον, είναι συμβατό με το IDE του arduino.

Συνδέουμε το ESP8266 με έναν από τους φθηνότερους ψηφιακούς αισθητήρες θερμοκρασίας και υγρασίας της σειράς DHT και δημιουργούμε ένα project που μπορεί να αναπτυχθεί κυριολεκτικά οπουδήποτε για να μεταδίδει δεδομένα αισθητήρων.

Θα δημιουργηθεί ένα sketch, σε προγραμματιστικό περιβάλλον Arduino IDE, που θα διαβάζει μετρήσεις θερμοκρασίας και υγρασίας από τον αισθητήρα DHT22 και θα τις προβάλλει με τη βοήθεια μιας REST Web Service που “ τρέχει ” σε μια συγκεκριμένη διεύθυνση IP στο τοπικό δίκτυο. Για να δούμε τις μετρήσεις μπορούμε να χρησιμοποιήσουμε ένα πρόγραμμα φυλλομετρητή (browser).

Η μετάδοση που εφαρμόζεται εδώ χρησιμοποιεί τον Web Server κώδικα του ESP8266 και ανταποκρίνεται σε αιτήματα ιστού (όπως σε ένα πρόγραμμα περιήγησης ή σε έναν πελάτη ιστού) για την επιστροφή δεδομένων θερμοκρασίας και υγρασίας (σε μορφή τύπου REST).

Το REST αντιπροσωπεύει το REpresentational State Transfer. Το REST είναι μια αρχιτεκτονική βασισμένη σε Web πρότυπα και χρησιμοποιεί πρωτόκολλο HTTP. Περιστρέφεται γύρω από τον πόρο, όπου κάθε στοιχείο είναι ένας πόρος και ένας πόρος είναι προσβάσιμος από μια κοινή διεπαφή χρησιμοποιώντας τυπικές μεθόδους HTTP.

Στην αρχιτεκτονική REST, ένας διακομιστής REST παρέχει απλώς πρόσβαση σε πόρους και ο REST πελάτης αποκτάει πρόσβαση και τροποποιεί τους πόρους. Εδώ κάθε πόρος προσδιορίζεται από URIs / global IDs. Το REST χρησιμοποιεί διάφορες αναπαραστάσεις για να αναπαραστήσει έναν πόρο όπως text, JSON, XML. Η JSON είναι η πιο δημοφιλής αναπαράσταση.

## HTTP μέθοδοι

Οι τέσσερις HTTP μέθοδοι που ακολουθούν χρησιμοποιούνται συνήθως στην αρχιτεκτονική που βασίζεται σε REST.

- **GET** – Παρέχει πρόσβαση μόνο για ανάγνωση σε έναν πόρο.
- **POST** – Χρησιμοποιείται για τη δημιουργία ενός νέου πόρου.
- **DELETE** – Χρησιμοποιείται για την κατάργηση ενός πόρου.
- **PUT** – Χρησιμοποιείται για την ενημέρωση ενός υπάρχοντος πόρου ή τη δημιουργία ενός νέου πόρου.

Ο προγραμματισμός του module ESP8266 γίνεται με χρήση του arduino IDE και διασυνδέοντας τον αισθητήρα θερμοκρασίας / υγρασίας DHT22. Χρησιμοποιώντας άλλους αισθητήρες και τις αντίστοιχες βιβλιοθήκες τους, άλλα ηλεκτρονικά συστήματα μπορούν να διασυνδεθούν με το ESP8266 και να παρακολουθούνται μέσω wifi.

Το hardware παρουσιάζεται σε ένα breadboard, καθώς δείχνει καλύτερα τις συνδέσεις και τον τρόπο σύνδεσης των ψηφιακών ακίδων με τον αισθητήρα. Το τελικό project μπορεί να έχει μόνο ηλεκτρική τροφοδοσία, έναν ρυθμιστή, τη μονάδα ESP8266 και έναν αισθητήρα, το οποίο θα μπορούσε να χωρέσει σε ένα πολύ μικρό κουτί.

## **Ο κώδικας**

Η διαδικασία κωδικοποίησης χρησιμοποιεί το πρόσθετο για το περιβάλλον ανάπτυξης του Arduino 1.6.x. της πλακέτας ESP8266.

Ο καλύτερος τρόπος για να το εγκαταστήσουμε είναι να ακολουθήσουμε τον οδηγό της Adafruit σχετικά με την προσθήκη υποστήριξης για πλακέτες όπως το ESP8266 για το arduino IDE.

## **Απαιτούμενες Βιβλιοθήκες**

Οι βιβλιοθήκες ESP8266WiFi, WiFiClient και ESP8266WebServer είναι απαιτούμενες.

Η άλλη βιβλιοθήκη που χρειάζεται είναι η βιβλιοθήκη DHT της Adafruit.

Πρέπει να επανεκκινήσουμε το περιβάλλον προγραμματισμού IDE του arduino για να είναι διαθέσιμη η βιβλιοθήκη.

## **Το πρόγραμμα**

Το παρακάτω πρόγραμμα αποθηκεύεται ως **ESP8266\_DHT22\_WebServer.ino**

```
/* DHTServer - ESP8266 Webserver with a DHT sensor as an input
```

Based on ESP8266Webserver, DHTexample, and BlinkWithoutDelay

Version 1.0 5/3/2014 Version 1.0 Mike Barela for Adafruit Industries

```
*/  
  
#include <ESP8266WiFi.h>  
  
#include <WiFiClient.h>  
  
#include <ESP8266WebServer.h>  
  
#include "DHT.h"  
  
#define DHTTYPE DHT22  
  
#define DHTPIN 2    // DHT Pin  
  
  
  
  
const char* ssid    = "GN_NETWORK.";   
  
const char* password = "qwe098gn";  
  
  
ESP8266WebServer server(80);  
  
  
// Initialize DHT sensor  
  
// NOTE: For working with a faster than ATmega328p 16 MHz Arduino chip, like an  
ESP8266,  
  
// you need to increase the threshold for cycle counts considered a 1 or 0.  
  
// You can do this by passing a 3rd parameter for this threshold. It's a bit  
  
// of fiddling to find the right value, but in general the faster the CPU the  
  
// higher the value. The default for a 16mhz AVR is a value of 6. For an  
  
// Arduino Due that runs at 84mhz a value of 30 works.  
  
// This is for the ESP8266 processor on ESP-01  
  
DHT dht(DHTPIN, DHTTYPE); // 11 works fine for ESP8266
```

```

float humidity, temp_f; // Values read from sensor

String webString=""; // String to display

// Generally, you should use "unsigned long" for variables that hold time

long previousMillis = 0; // will store last temp was read

const long interval = 2000; // interval at which to read sensor

void handle_root() {

    server.send(200, "text/plain", "Hello from the weather esp8266, read from /temp or
/humidity");

    delay(100);

}

void setup()

{

    // You can open the Arduino IDE Serial Monitor window to see what the code is doing

    Serial.begin(9600); // Serial connection from ESP-01 via 3.3v console cable

    // Connect to WiFi network

    WiFi.begin(ssid, password);

    Serial.print("\n\r\n\rWorking to connect");

    // Wait for connection

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);

```

```

    Serial.print(".");

}

Serial.println("");

Serial.println("DHT Weather Reading Server");

Serial.print("Connected to ");

Serial.println(ssid);

Serial.print("IP address: ");

Serial.println(WiFi.localIP());


server.on("/", handle_root);


server.on("/temp", [](){ // if you add this subdirectory to your webserver call, you get
text below :)

    gettemperature();    // read sensor

    //webString="Temperature: "+String((int)temp_f)+" C"; // Arduino has a hard time
with float to string

    webString=String((float)temp_f);

    server.send(200, "text/plain", webString);          // send to someones browser when
asked

});


server.on("/humidity", [](){ // if you add this subdirectory to your webserver call, you
get text below :)

    gettemperature();    // read sensor

    //webString="Humidity: "+String((int)humidity)+"%";

```

```

    webString=String((float)humidity);

    server.send(200, "text/plain", webString);          // send to someones browser when
asked

});

server.begin();

dht.begin();    // initialize DHT sensor

}

void loop()

{

    server.handleClient();

}

void gettemperature() {

    // Wait at least 2 seconds seconds between measurements.

    // if the difference between the current time and last time you read

    // the sensor is bigger than the interval you set, read the sensor

    // Works better than delay for things happening elsewhere also

    long currentMillis = millis();

    if(currentMillis - previousMillis > interval) {

        // save the last time you read the sensor

        previousMillis = currentMillis;

        // Reading temperature for humidity takes about 250 milliseconds!

```

```

// Sensor readings may also be up to 2 seconds 'old' (it's a very slow sensor)

humidity = dht.readHumidity();    // Read humidity (percent)

temp_f = dht.readTemperature();  // Read temperature as Celsius


// Check if any reads failed and exit early (to try again).

if (isnan(humidity) || isnan(temp_f)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

}

}

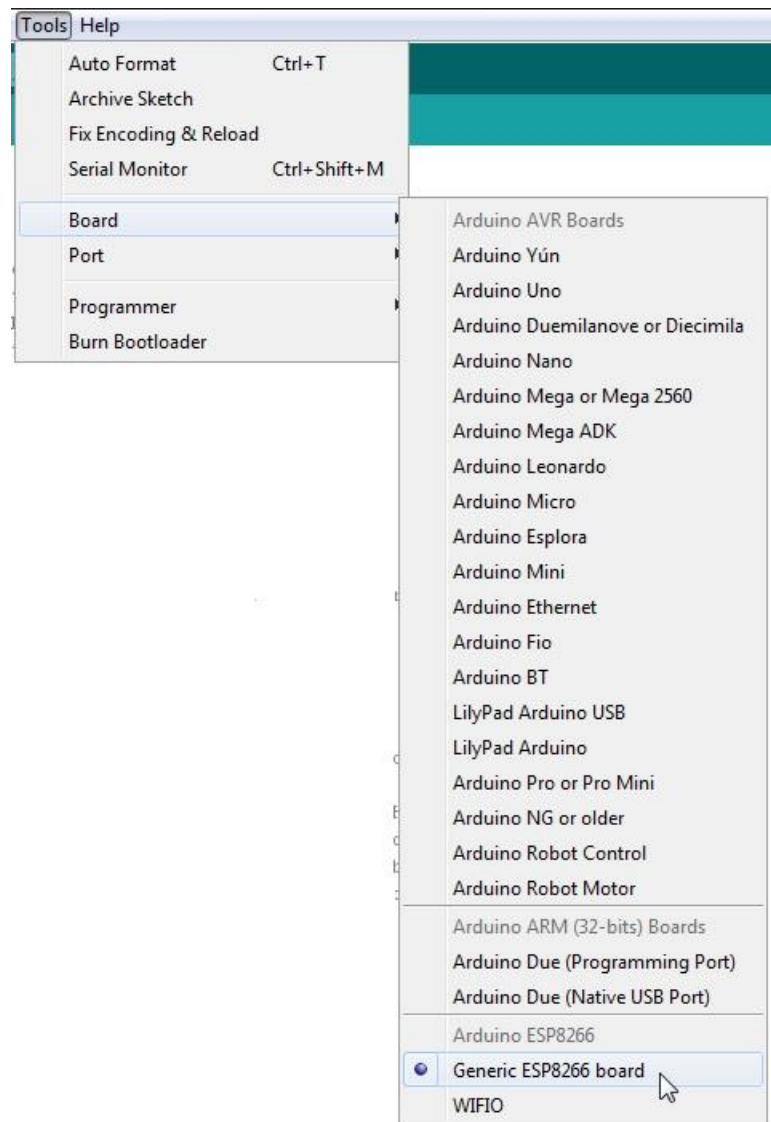
}

```

## IDE Setup

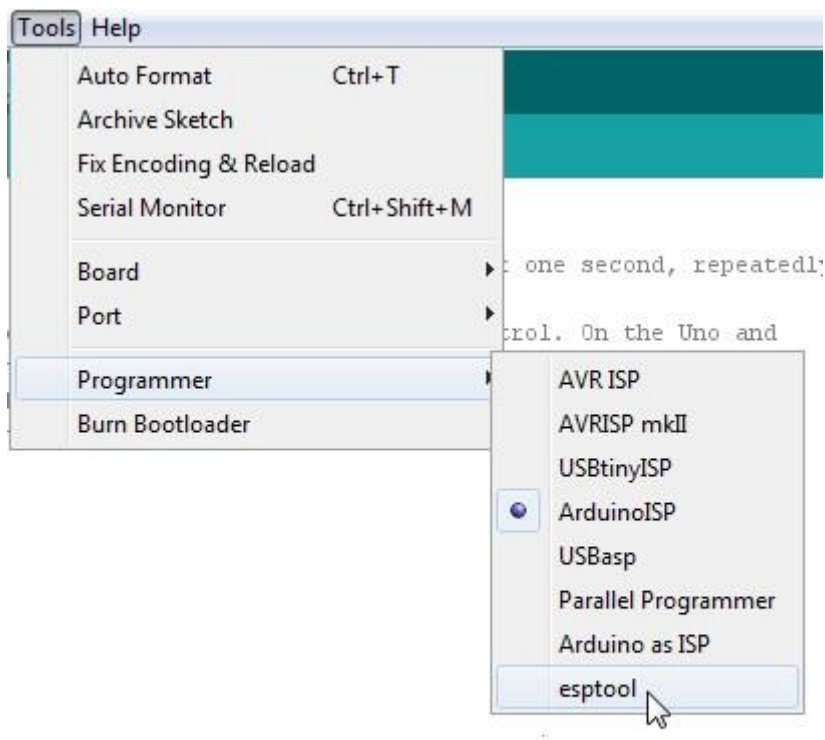
Το επόμενο βήμα είναι να επιλεγθούν τα στοιχεία μενού για το ESP. Όταν έχουμε εγκατεστημένη την υποστήριξη για το ESP8266, μπορούμε να επιλέξουμε το Generic ESP8266 ως πλακέτα (Εικόνα 49).





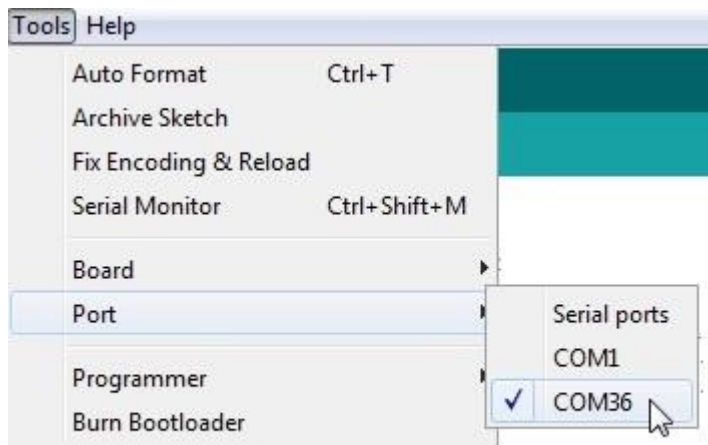
Εικόνα 49. Επιλογή της κατάλληλης πλακέτας ESP8266

Από την καρτέλα Tools -> Programmer menu item, επιλέγουμε esptool (Εικόνα 50)



Εικόνα 50. Επιλογή του esptool

Με το project μας να τροφοδοτείται και το USB σειριακό καλώδιο συνδεδεμένο, θα χρειαστεί να επιλέξουμε τη σειριακή θύρα στην οποία εμφανίζεται το καλώδιο. Στα Windows, χρησιμοποιούμε την οθόνη "Συσκευές και εκτυπωτές" για να δούμε το καλώδιο (Εικόνα 51).

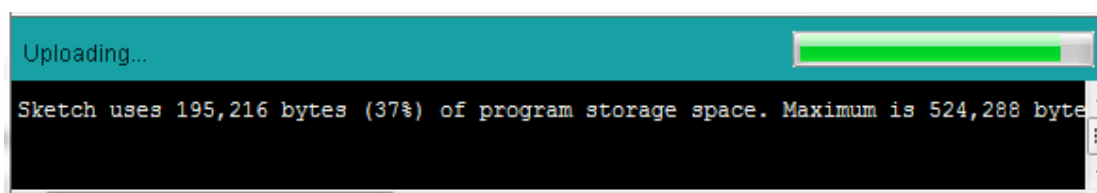


Εικόνα 51. Επιλογή της συνδεδεμένης σειριακής πόρτας επικοινωνίας με την πλακέτα

Τώρα θα πρέπει να είμαστε σε θέση να μεταγλωττίσουμε το πρόγραμμα που φορτώθηκε στην τελευταία σελίδα. Κάνουμε κλικ στο στρογγυλό εικονίδιο Ελέγχου για να βεβαιωθούμε ότι το πρόγραμμα μεταγλωττίζεται.

Εάν έχουμε λάθη που αναφέρουν DHT τότε η DHT βιβλιοθήκη της Adafruit δεν αναγνωρίστηκε (επιστρέφουμε και ακολουθούμε τα βήματα εγκατάστασης ξανά).

Εάν όλα έχουν ρυθμιστεί σωστά, το πρόγραμμα θα μεταγλωττιστεί κανονικά (Εικόνα 52):



Εικόνα 52. Η μεταγλώττιση του προγράμματος

## Uploading στο ESP8266

Για να μεταφορτώσουμε το μεταγλωττισμένο πρόγραμμά μας στο ESP8266:

1. Αποσυνδέουμε την τροφοδοσία από το ESP.

2. Φέρνουμε το GPIO0 στη γείωση (πατάμε ΚΑΙ ΚΡΑΤΑΜΕ ΠΑΘΗΜΕΝΟ το pushbutton ).
3. Συνδέουμε την τροφοδοσία στο ESP.
4. Απελευθερώνουμε το pushbutton
5. Κάνουμε κλικ στο στρογγυλό εικονίδιο δεξιού βέλους για να μεταφορτώσουμε το πρόγραμμα (Εικόνα 53)



Εικόνα 53. Το στρογγυλό κουμπί για την έναρξη της μεταφόρτωσης του προγράμματος

Αν έχουμε σφάλματα στις επικοινωνίες, ελέγχουμε τις συνδέσεις μας και σιγουρευόμαστε ότι έχουμε επιλεγμένη την κατάλληλη σειριακή θύρα. Ελέγχουμε ότι τα καλώδια σωστού χρώματος από το σειριακό καλώδιο είναι συνδεδεμένα στη μονάδα ESP8266.

Όταν αρχίσει να λειτουργεί, θα πρέπει να δούμε να αναβοσβήνει το μικροσκοπικό μπλε φως στην ηλεκτρονική μονάδα ESP8266 και την ακόλουθη οθόνη στο IDE του Arduino (Εικόνα 54):

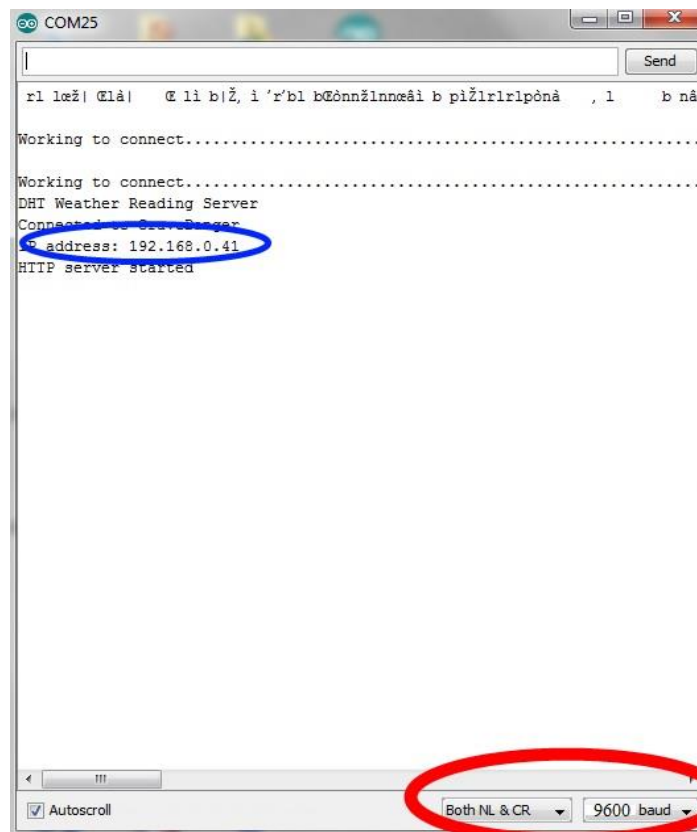
```
Sketch uses 174,082 bytes (33%) of program storage space. Maximum is 524,288 bytes.
Uploading 29072 bytes from C:\Users\hwiguna\AppData\Local\Temp\build491626908110780874
.....
Uploading 145048 bytes from C:\Users\hwiguna\AppData\Local\Temp\build49162690811078087
.....
```

Εικόνα 54. Η διαδικασία μεταφόρτωσης

Μόλις σταματήσει η μεταφόρτωση, το πρόγραμμα είναι φορτωμένο και εκτελείται. Εάν δεν φαίνεται να εκτελείται, μπορούμε να αποσυνδέσουμε την τροφοδοσία από την πρίζα και να την ξανασυνδέσουμε.

## Server Running

Ανοίγουμε το σειριακό τερματικό του arduino IDE χρησιμοποιώντας το μενού Tools -> Serial Monitor. Ρυθμίζουμε το baud rate στο κάτω δεξιό μέρος (κόκκινος κύκλος παρακάτω) σε 9.600 baud. Θα πρέπει να δούμε μια οθόνη όπως αυτή παρακάτω (Εικόνα 55):



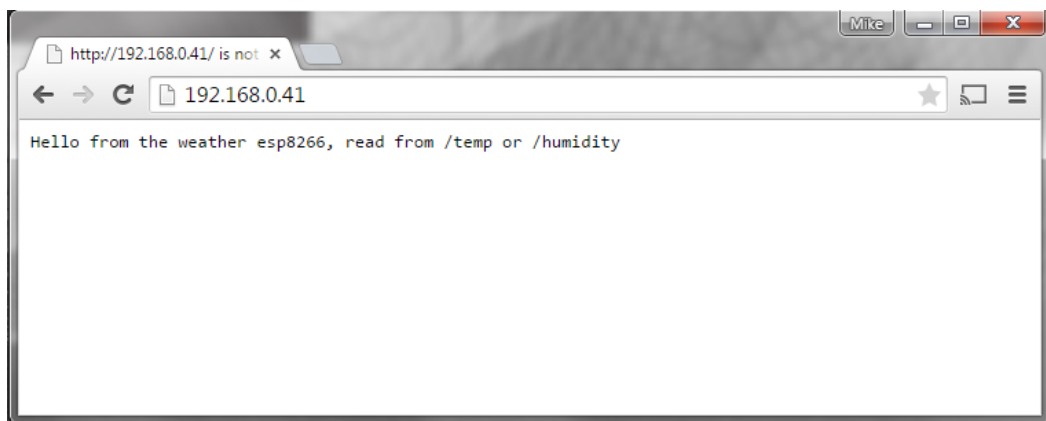
Εικόνα 55. Η οθόνη του σειριακού monitor κατά τη λειτουργία του server

Όταν ενεργοποιηθεί το project, θα λάβουμε μερικούς τυχαίους χαρακτήρες, στη συνέχεια μερικές τελείες, καθώς το ESP8266 συνδέεται με το δρομολογητή όπου καθορίσαμε το SSID και τον κωδικό πρόσβασης στο arduino sketch.

Όταν το ESP8266 συνδεθεί στον δρομολογητή μας, θα πει "DHT Weather Reading Server". Σημειώνουμε τη διεύθυνση IP (στον μπλε κύκλο) για το επόμενο βήμα.

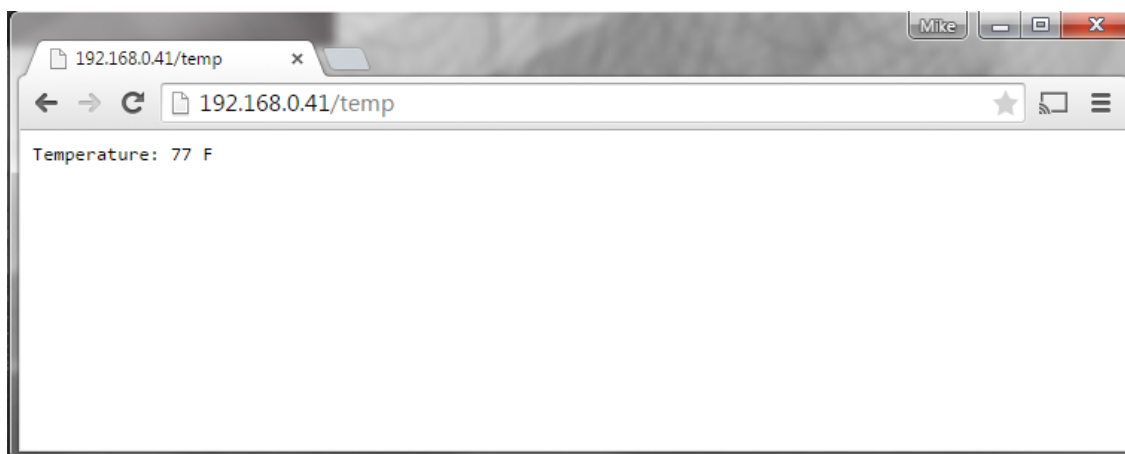
## Χρήση του Webserver

Ανοίγουμε το αγαπημένο μας πρόγραμμα περιήγησης στο Web σε οποιαδήποτε συσκευή του δικτύου του δρομολογητή μας. Πληκτρολογούμε τη διεύθυνση της πλακέτας ESP8266, που σημειώσαμε στο προηγούμενο βήμα, στη γραμμή διευθύνσεων (Εικόνα 56):



Εικόνα 56. Η διεύθυνση της ιστοσελίδας στην οποία "ακούει" ο Web Server

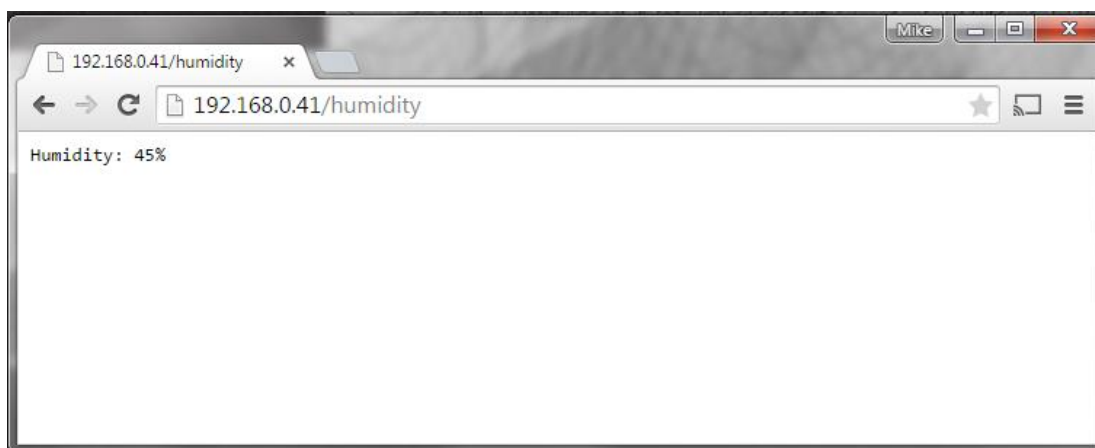
Αν όλα πήγαν καλά είμαστε συνδεδεμένοι με το project. Τώρα για να διαβάσουμε τη θερμοκρασία, πληκτρολογούμε την διεύθυνση IP (στο δίκτυό μας 192.168.0.41) με "/temp" μετά τη διεύθυνση (Εικόνα 57):



Εικόνα 57. Προβολή της τιμής της θερμοκρασίας

Αν θέλουμε μια πιο παραδοσιακή απάντηση REST (μόνο την τιμή και όχι το κείμενο που μας λέει ότι είναι η θερμοκρασία), μπορούμε να επεξεργαστούμε την κατάλληλη γραμμή του προγράμματος για να την αφαιρέσουμε.

Τώρα ζητάμε την υγρασία προσθέτοντας `"/ humidity"` στην διεύθυνση IP του ESP8266 του project μας (Εικόνα 58):



Εικόνα 58. Προβολή της τιμής της υγρασίας

Έχουμε τώρα ένα module ESP8266 που τρέχει τον δικό μας προσαρμοσμένο κώδικα που διαβάζει έναν αισθητήρα. Δεν χρειάζεται arduino ή άλλος μικροελεγκτής.

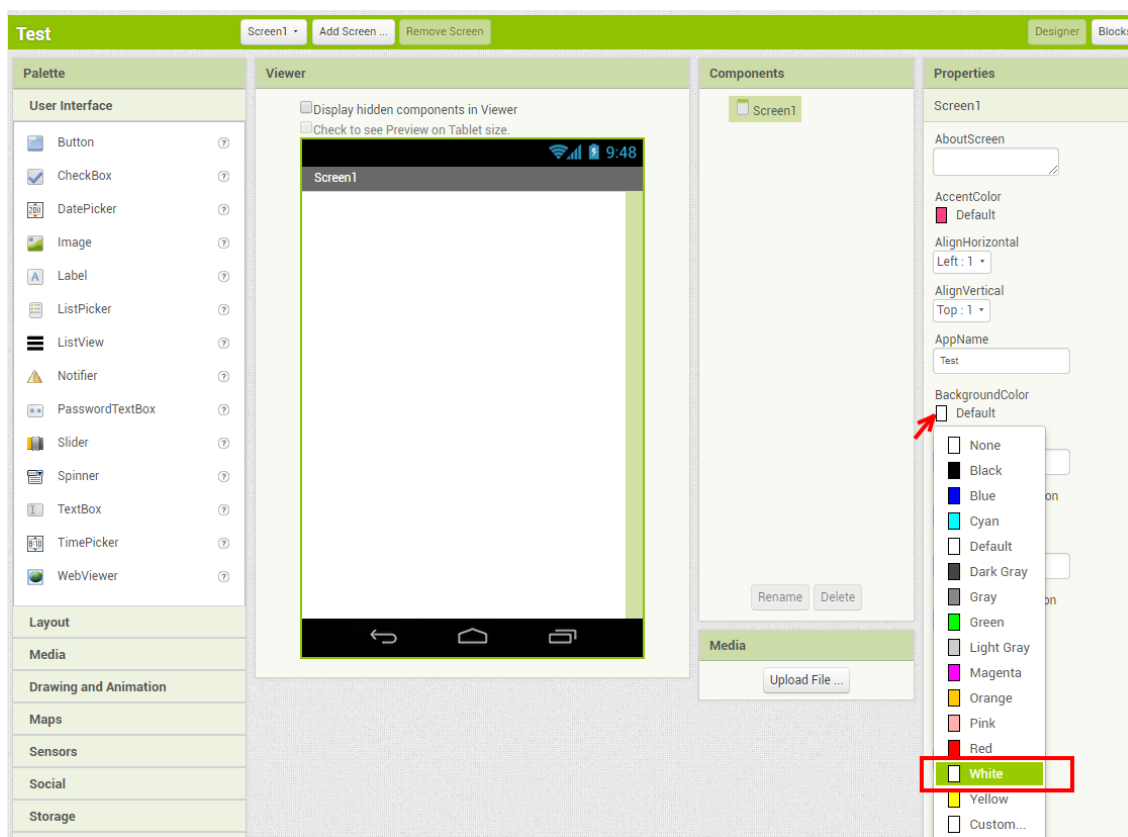
#### 4.4 Προγραμματισμός της REST client εφαρμογής “url\_request”

Η REST Web Service που περιγράφηκε στην προηγούμενη παράγραφο θα συνδυαστεί με μια REST client εφαρμογή για φορητές συσκευές. Η REST client εφαρμογή θα δημιουργηθεί με το προγραμματιστικό περιβάλλον “appinventor”, θα διαβάζει τις μετρήσεις από τον αισθητήρα DHT22 και θα ονομαστεί **“url\_request”**. Η εφαρμογή αυτή θα καλεί την REST Web Service και θα προβάλει τις μετρήσεις θερμοκρασίας και υγρασίας στην οθόνη του κινητού μας ή του tablet όπου «τρέχει» η εφαρμογή **“GReen-OS”**.

Θα κατασκευαστεί μια απλή εφαρμογή χρησιμοποιώντας το App Inventor, η οποία θα εμφανίζει ζωντανά δεδομένα από έναν συνδεδεμένο αισθητήρα DHT22. Ο συνδεδεμένος αισθητήρας δημοσιεύει τα δεδομένα του σε έναν διακομιστή ιστού. Αυτή η εφαρμογή θα διαβάζει τα δεδομένα χρησιμοποιώντας ένα στοιχείο WebViewer και ένα timer (a Clock component) για ανανέωση τιμών.

## Δημιουργία της εφαρμογής με το App Inventor

Πηγαίνουμε στο [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu), κάνουμε log-in και ξεκινάμε ένα νέο project. Πληκτρολογούμε ένα όνομα για το project μας. Αλλάζουμε το φόντο της οθόνης σε λευκό (Εικόνα 59):

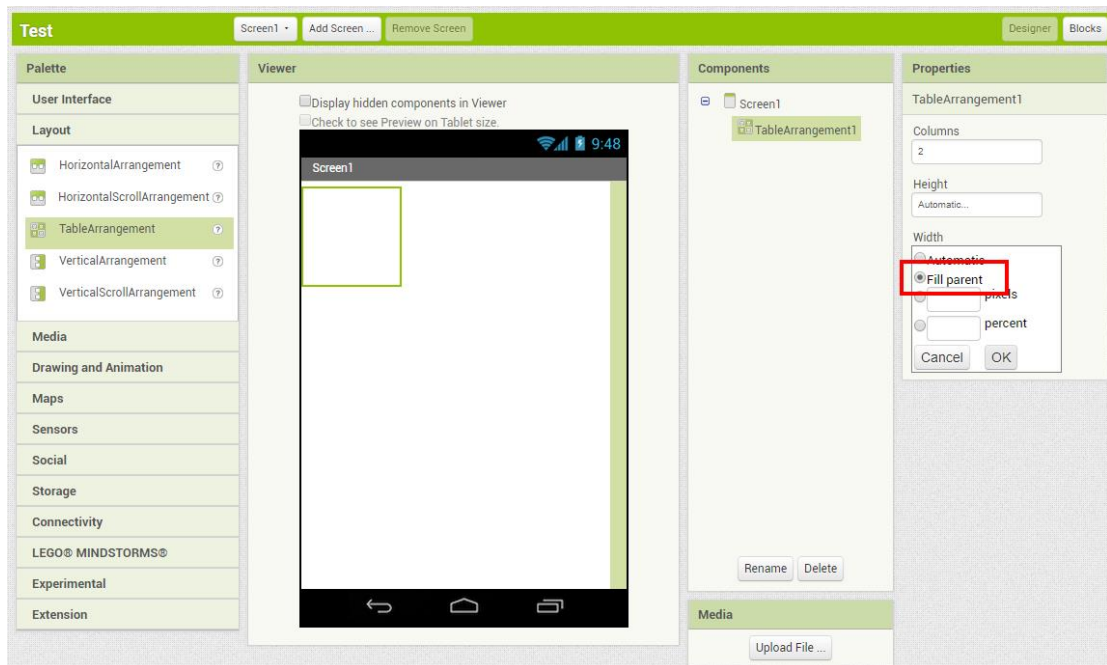


Εικόνα 59. Οι ρυθμίσεις για την αλλαγή του φόντου του screen1

Εάν μετακινηθούμε προς τα κάτω στη λίστα με τις Ιδιότητες, μπορούμε να αλλάξουμε τον Τίτλο (τώρα Screen1) σε "temperature-humidity".

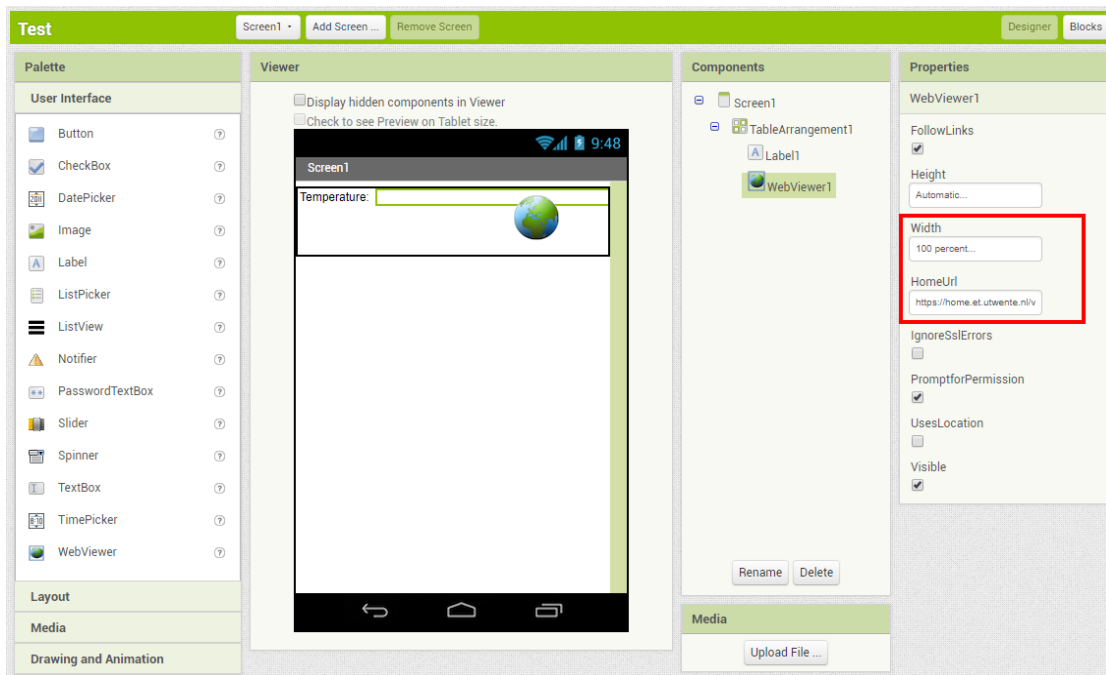


Στη συνέχεια, στο Palette, ανοίγουμε το "Layout" και προσθέτουμε ένα "TableArrangement" (το σύρουμε στο Viewer). Ορίζουμε το πλάτος του σε "Fill Parent" (Εικόνα 60):



Εικόνα 60. Η ρύθμιση του πλάτους του TableArrangement1

Στην καρτέλα Palette ανοίγουμε το "User Interface" και σύρουμε ένα "Label" και ένα "WebView" στο Viewer (βεβαιωνόμαστε ότι και τα δύο καταλήγουν μέσα στο TableArrangement). Αλλάζουμε το Text του Label σε "Temperature", ορίζουμε το πλάτος του WebView στο 100% και εισάγουμε τη διεύθυνση URL στο πεδίο 'HomeUrl'. Αυτή μπορεί να είναι "https://home.et.utwente.nl/val/temp" για δοκιμαστικούς σκοπούς (Εικόνα 61).

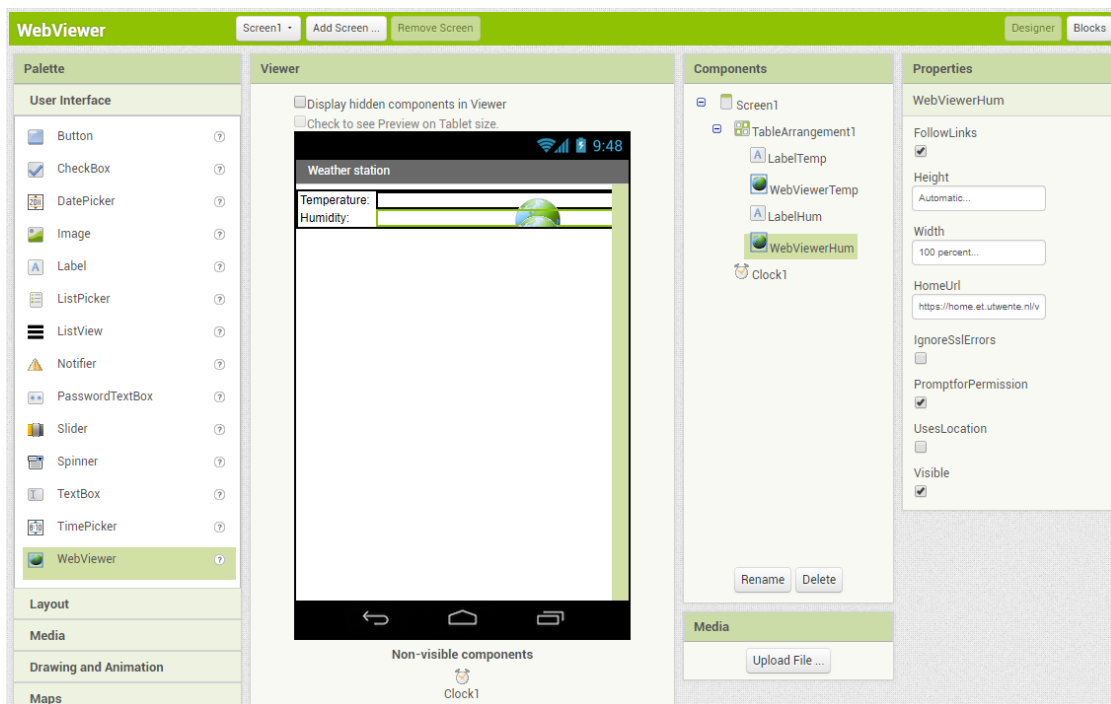


Εικόνα 61. Εισαγωγή διεύθυνσης URL στο WebViewer1 component

Μπορούμε να αλλάξουμε τα ονόματα των components σε πιο meaningful ονόματα στο Components window, χρησιμοποιώντας το κουμπί Rename.

Στη συνέχεια, προσθέτουμε ένα άλλο "Label" και ένα "WebViewer" (μέσα στο TableArrangement) για την υγρασία και ορίζουμε τις ιδιότητές τους.

Το δεύτερο WebViewer παίρνει "https://home.et.utwente.nl/val/humidity" ως το "HomeUrl"(Εικόνα 62).



Εικόνα 62. Εισαγωγή διεύθυνσης URL στο WebViewerHum component

Αποθηκεύουμε το project μέσω Projects > Save Project στο main menu.

## Refreshing the values

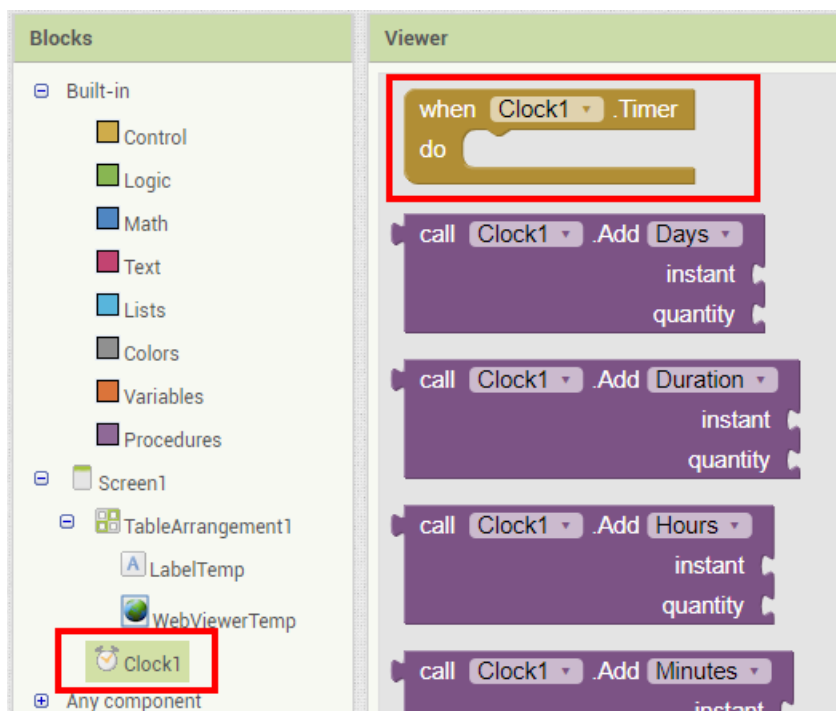
Για να ανανεώνουμε τις τιμές τακτικά, θα προσθέσουμε ένα timer στην εφαρμογή. Από το Palette, επιλέγουμε το "Clock" (είναι κάτω από το "Sensor") και το μεταφέρουμε στο Viewer. Δεν θα γίνει ένα ορατό συστατικό του User interface (γι' αυτό και παρατίθεται στην ενότητα "Non-visible components"). Επιλέγουμε το Clock και ρυθμίζουμε την ιδιότητά του 'TimerInterval' στο 10000. Αυτό θα κάνει το clock 'tick' κάθε 10 δευτερόλεπτα (η τιμή είναι σε milliseconds). Για δοκιμαστικούς σκοπούς αυτή είναι μια καλή τιμή. Αργότερα, ίσως να θέλουμε να το αυξήσουμε, για παράδειγμα 300000 (5 λεπτά).

Τώρα θα προγραμματίσουμε το Clock μας. Το App Inventor χρησιμοποιεί Blocks για τον προγραμματισμό. Κάνουμε εναλλαγή στη διεπαφή Blocks κάνοντας κλικ στο κουμπί 'Blocks' στην πράσινη επικεφαλίδα (στην πάνω δεξιά γωνία) (Εικόνα 63):



Εικόνα 63. Το κουμπί για την μετάβαση στο περιβάλλον προγραμματισμού

Από την καρτέλα "Blocks" επιλέγουμε το "Clock1" (Εικόνα 64) και σύρουμε το block "when ... do" στο Viewer (Εικόνα 65):

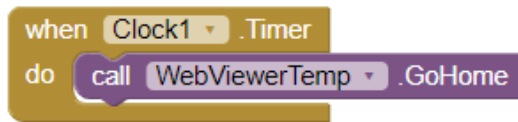


Εικόνα 64. Επιλογή του Clock1 component



Εικόνα 65. Επιλογή του block "when ... do" για το Clock1

Στη συνέχεια, στην καρτέλα "Blocks" επιλέγουμε "WebViewTemp" (ή οποιοδήποτε είναι το όνομα) και σύρουμε το block "call WebViewTemp .GoHome" στο Viewer. Το τοποθετούμε μέσα στο block "when ... do"(Εικόνα 66):



Εικόνα 66. Ο κώδικας για την ανανέωση της ιστοσελίδας που προβάλλει την θερμοκρασία

Στη συνέχεια, επιλέγουμε 'WebViewHum' και σύρουμε το block "call WebViewHum .GoHome" μέσα στο block "when Clock1 do" επίσης.

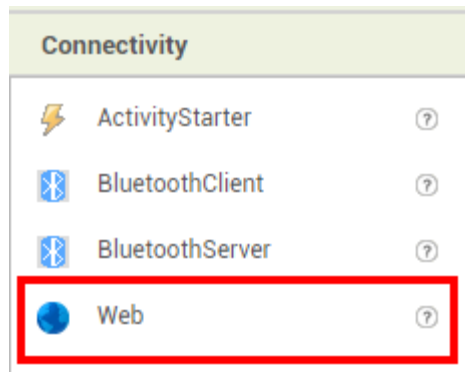
Τώρα έχουμε προγραμματίσει και τα δύο WebViews να ανανεώνονται κάθε 10 δευτερόλεπτα. Τώρα μπορούμε να το δοκιμάσουμε: αν έχουμε τον αισθητήρα σε λειτουργία, μπορούμε να αναπνέουμε επάνω του, έτσι μεταβάλλονται οι τιμές θερμοκρασίας και υγρασίας. Ελέγχουμε την εφαρμογή εάν εμφανίζει τις νέες τιμές. Εάν δεν έχουμε αισθητήρα που να δουλεύει, μπορούμε να αλλάξουμε την τιμή του αισθητήρα, αναθέτοντας σε αυτό μια νέα "ψεύτικη" τιμή. Για παράδειγμα, για να ρυθμίσουμε τη θερμοκρασία στους 10 βαθμούς, θα επισκεφθούμε απλά αυτή τη διεύθυνση URL στον browser μας: "https://home.et.utwente.nl/val/temp/10".

## **Ανάγνωση των πραγματικών τιμών και επεξεργασία**

Το να προβάλλουμε απλώς τις τιμές από τον ιστό δεν είναι αυτό που θέλουμε. Θέλουμε να διαβάσουμε τις πραγματικές τιμές και να τις επεξεργαστούμε.

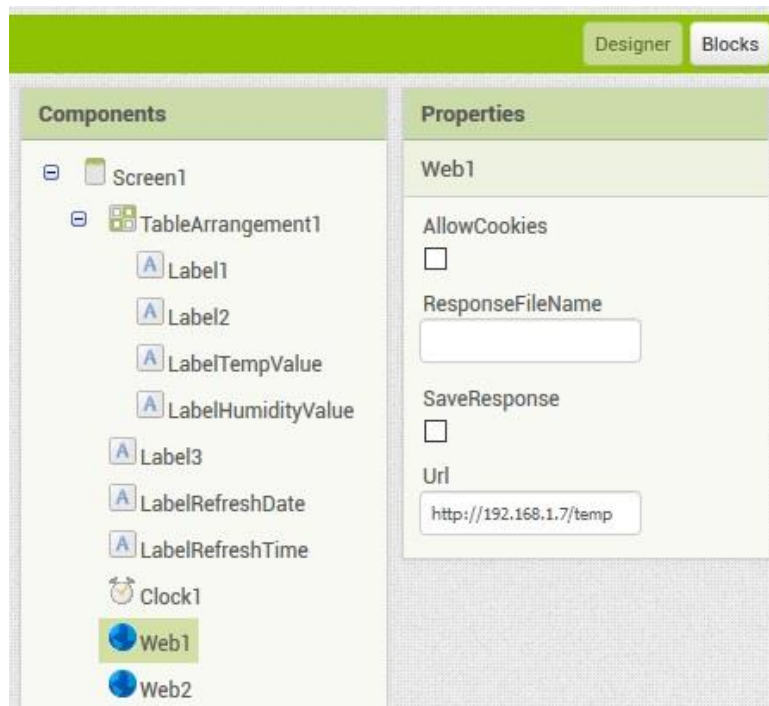
Για την περίπτωσή μας, να κάνουμε κάτι σαν: αν η θερμοκρασία είναι πάνω από 25 βαθμούς, να εμφανίζεται μια προειδοποίηση. Αυτό δεν μπορούμε να το κάνουμε χρησιμοποιώντας απλά ένα WebView component.

Για αυτό, θα χρησιμοποιήσουμε το Web Component (Εικόνα 67):



Εικόνα 67. Η επιλογή του Web component

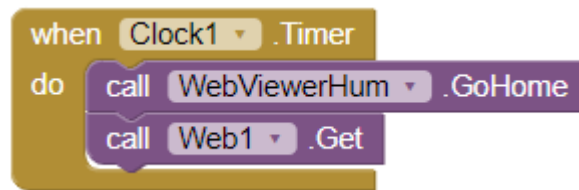
Το σύρουμε στον Viewer. Ως τιμή για την ιδιότητά του 'Url' στην περίπτωση μας, για τον REST Web Server μας, ορίζουμε: `http://192.168.1.7/temp` (Εικόνα 68).



Εικόνα 68. Ορισμός συγκεκριμένης διεύθυνσης Url για το Web1 component

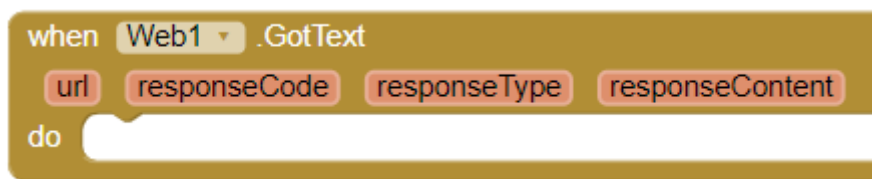
Επειδή θα διαβάσουμε την τιμή χρησιμοποιώντας ένα Web Component, το Web Viewer δεν χρειάζεται πλέον για την εμφάνιση της θερμοκρασίας. Έτσι αφαιρούμε αυτό το component και το αντικαθιστούμε με ένα συνηθισμένο Label το οποίο και ονομάζουμε 'LabelTempValue'.

Στη συνέχεια, στην καρτέλα "Blocks" αντικαθιστούμε το "call WebViewerTemp.GoHome" με "call Web1.Get" (Εικόνα 69):



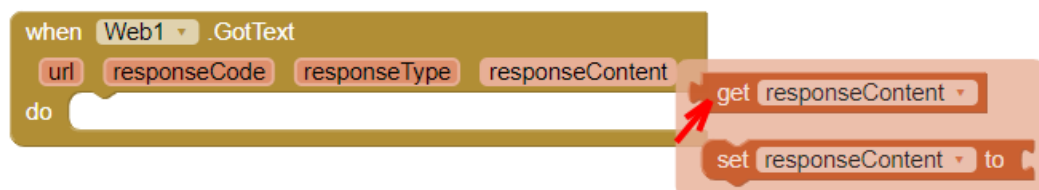
Εικόνα 69. Προσθήκη της μεθόδου get στον κώδικα για το Clock1

Αυτό θα εκτελέσει τη get-method του component Web1 και μπορούμε να απαντήσουμε σε αυτό, προσθέτοντας μια διαδικασία που θα χειριστεί τα εισερχόμενα δεδομένα από το get-request: κάνουμε κλικ στο στοιχείο Web1 και σύρουμε το block "when Web1.GoText" μέσα στο Viewer (Εικόνα 70):



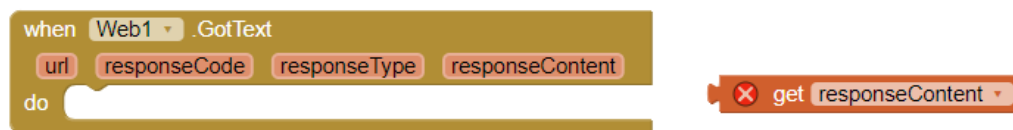
Εικόνα 70. Διαδικασία διαχείρισης των εισερχόμενων δεδομένων από το get-request

Μπορούμε να φέρουμε τα εισερχόμενα δεδομένα από το get-request μέσω της παραμέτρου responseContent. Τοποθετούμε το δείκτη του ποντικιού πάνω από την παράμετρο 'responseContent' του block (Εικόνα 71):



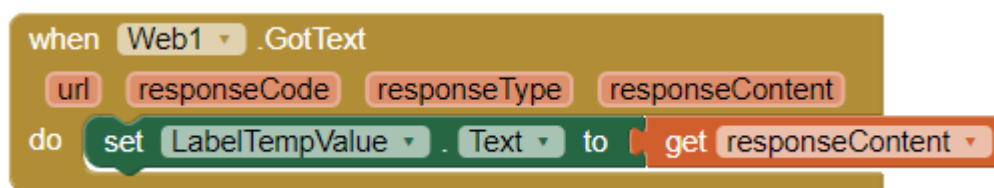
Εικόνα 71. Επιλογή της παραμέτρου responseContent

Και το σύρουμε σε ένα άδειο σημείο (Εικόνα 72):



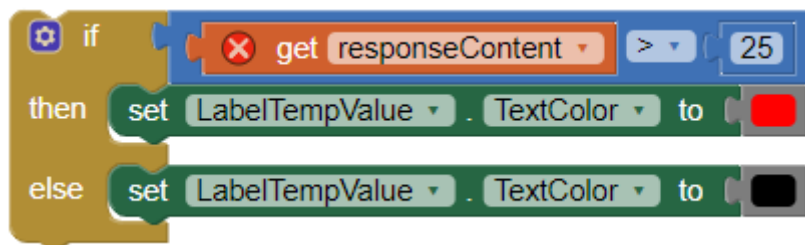
Εικόνα 72. Διαθεσιμότητα της παραμέτρου responseContent

Τώρα, επιλέγουμε το label LabelTempValue και επιλέγουμε το block "set LabelTempValue. Text to" και επισυνάπτουμε την ελεύθερη παράμετρο "responseContent" σε αυτό, σύροντάς την στη θέση της (Εικόνα 73):



Εικόνα 73. Χρήση της παραμέτρου responseContent για διαχείριση εισερχόμενων δεδομένων

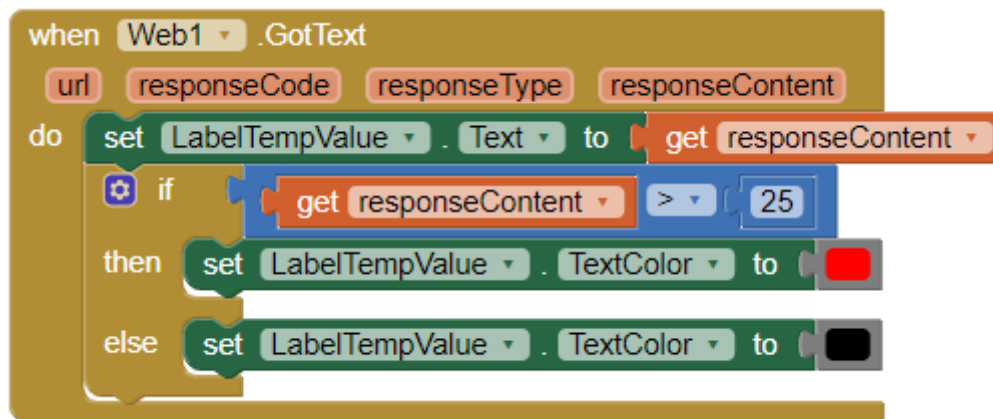
Τέλος, μπορούμε να ελέγξουμε τη θερμοκρασία. Αν η θερμοκρασία είναι πάνω από 25 βαθμούς, θα εμφανιστεί η τιμή της με κόκκινο χρώμα. Για αυτό μπορούμε να χρησιμοποιήσουμε μια δομή if-statement (Εικόνα 74):



Εικόνα 74. Η δομή if-statement με χρήση κόκκινου χρώματος για υψηλές θερμοκρασίες

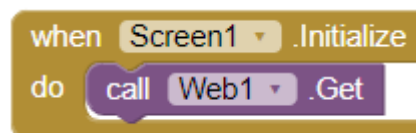


Τότε το προσθέτουμε στο block "when Web1 .GotText"(Εικόνα 75):



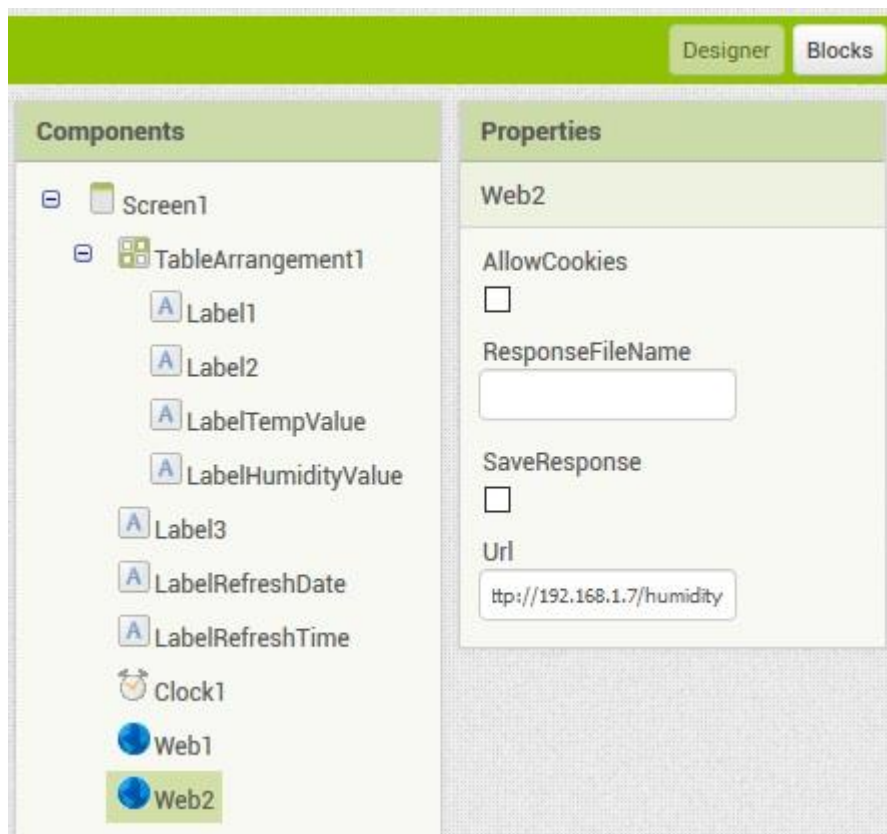
Εικόνα 75. Χρήση της δομής if-statement για δημιουργία alarm για την υψηλή θερμοκρασία

Κατά τη δοκιμή της εφαρμογής, παίρνει λίγο χρόνο πριν εμφανιστεί η τιμή της θερμοκρασίας. Αυτό οφείλεται στο γεγονός ότι προσθέσαμε μια κλήση στο Web1.Get στον timer, αλλά αυτός πυροδοτείται μόνο μετά από ... 10 δευτερόλεπτα. Θα πρέπει επίσης να πραγματοποιήσουμε μια κλήση αμέσως μόλις φορτώνεται το screen. Για να το κάνουμε αυτό, κάνουμε κλικ στο Screen1, μετά σύρουμε το "when Screen1 . Initialize ... do" στον Viewer και προσθέτουμε μια κλήση στο Web1.Get (Εικόνα 76):



Εικόνα 76. Κλήση της μεθόδου Web1.Get αμέσως μόλις φορτώνεται το Screen1

Με τον ίδιο τρόπο προσθέτουμε τα κατάλληλα blocks για το Web2 component θέτοντας την ιδιότητα 'Url' σε: <http://192.168.1.7/humidity> (Εικόνα 77).



Εικόνα 77. Ορισμός συγκεκριμένης διεύθυνσης Url για το Web2 component

## Εμφάνιση της τελευταίας ενημέρωσης

Είναι χρήσιμο σε αυτήν την εφαρμογή να βλέπουμε την τελευταία φορά που ανα-νεώθηκε η τιμή. Χρειαζόμαστε τον πραγματικό χρόνο ανανέωσης της θερμοκρασίας και της υγρασίας.

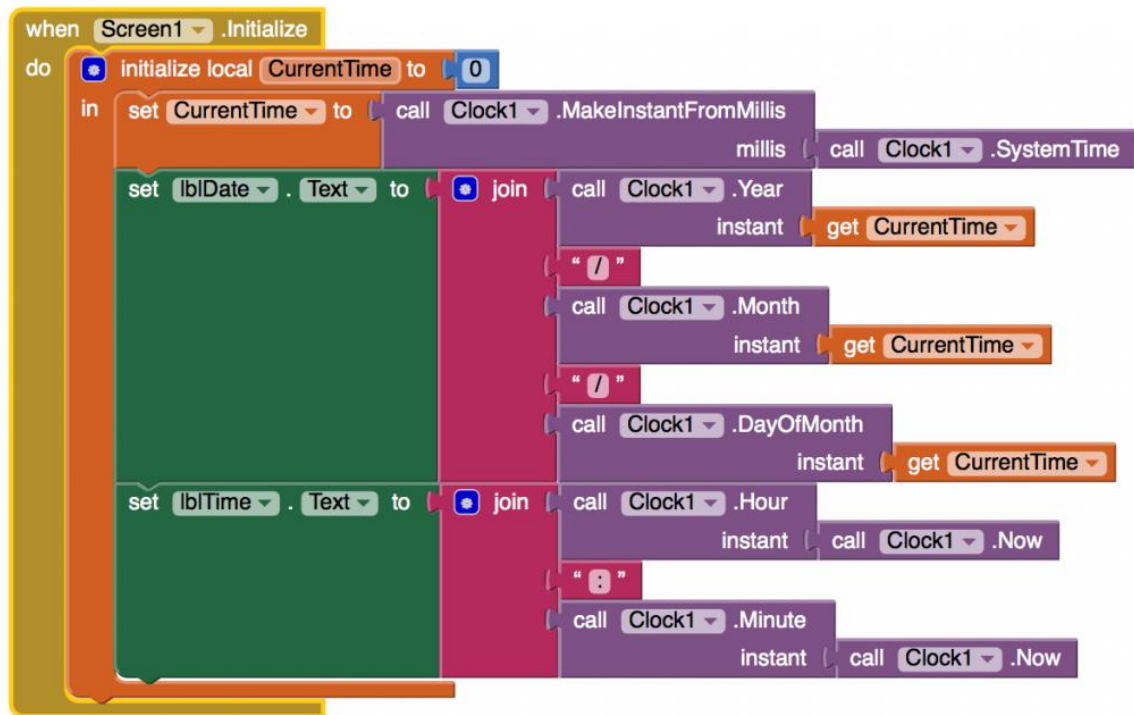
Για να αποκτήσουμε την τρέχουσα ημερομηνία και ώρα, η εφαρμογή μας καλεί το Clock1 component. Οι χρόνοι αποθηκεύονται σε εσωτερική μορφή που βασίζεται σε milliseconds, από το έτος 1970. Αυτή η ιδιότητα μορφή παράγεται από το Clock control, αλλά πρέπει να μετατραπεί σε τιμές έτους, μήνα, ημέρας και ώρας και λεπτών.

Το Step 1 είναι να αποκτήσουμε το *CurrentTime*. Αυτό γίνεται με την αναφορά της ιδιότητας Clock1.SystemTime (η οποία είναι σε μορφή millisecond) και στη συνέχεια μεταφέροντας αυτή την τιμή στη μέθοδο Clock1.MakeInstantFromMills και την αποθήκευση του αποτελέσματος στη μεταβλητή CurrentTime.

Το Step 2 μετατρέπει την τιμή *CurrentTime* χρησιμοποιώντας επιπλέον *Clock1* μεθόδους, για το *Year*, *Month* and *DayofMonth*.

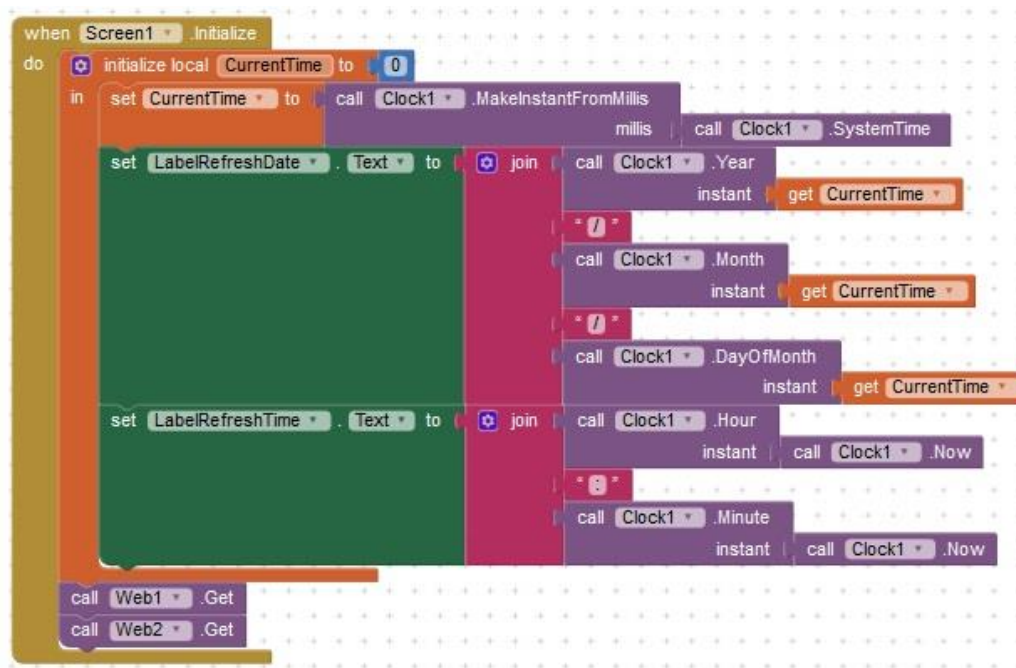
Το Step 3 αποκτά τις ώρες και τα λεπτά, αλλά το κάνει χρησιμοποιώντας μια εναλλακτική μέθοδο κλήσης *Clock1.Now* για να λάβουμε την τρέχουσα ώρα.

Και οι δύο μέθοδοι παρουσιάζονται στα παρακάτω blocks παραδειγμάτων (Εικόνα 78):

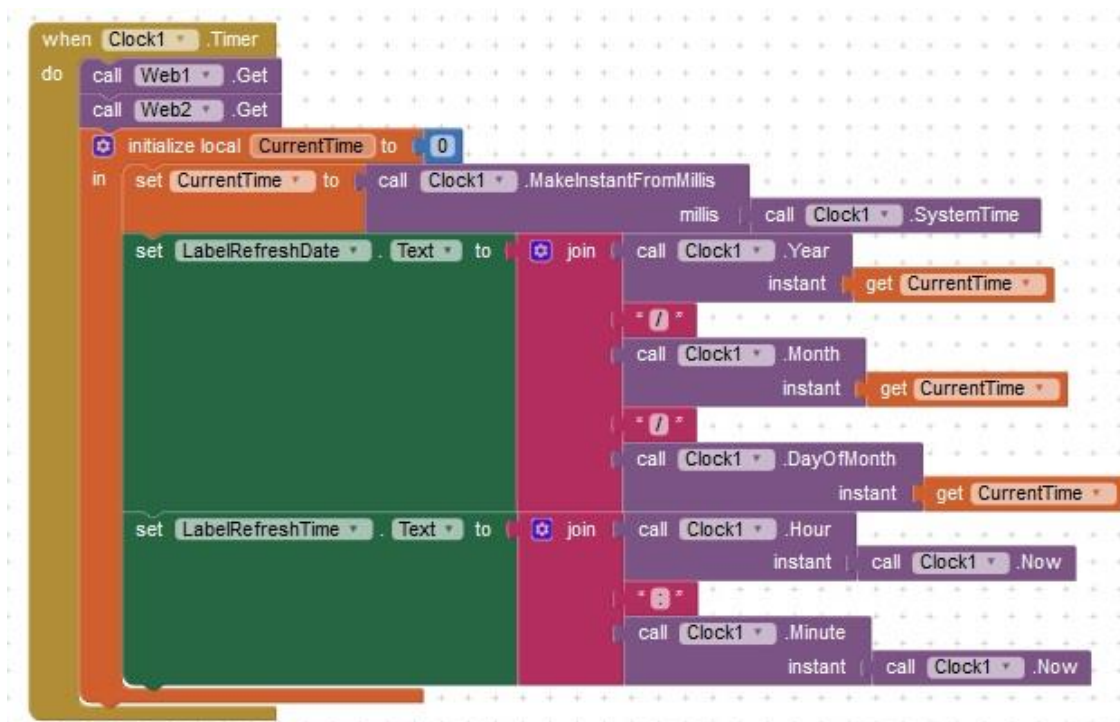


Εικόνα 78. Ο κώδικας για την τρέχουσα ημερομηνία και ώρα

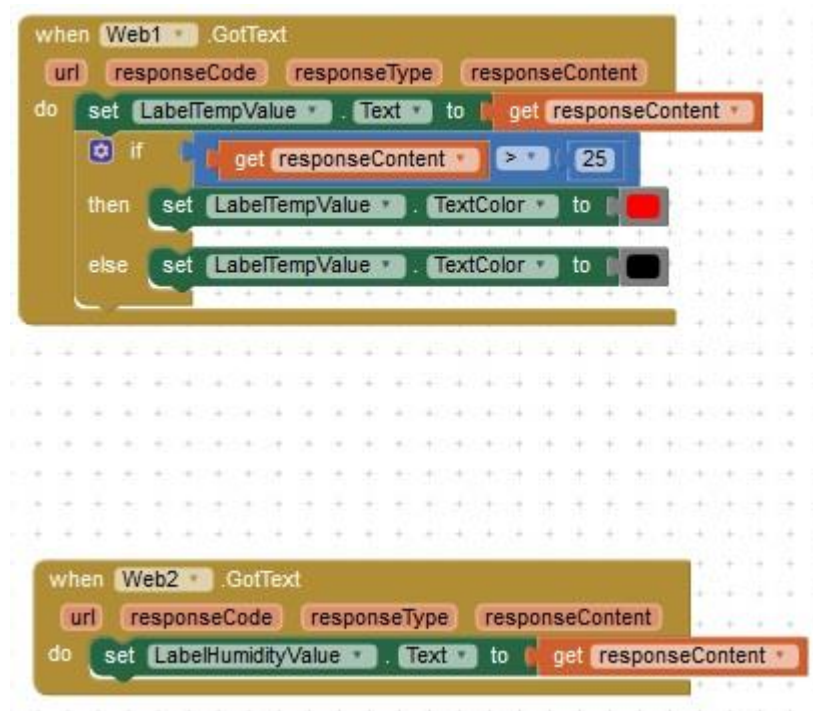
Τα blocks κώδικα για την πλήρη εφαρμογή φαίνονται στις 3 παρακάτω εικόνες: (Εικόνα 79), (Εικόνα 80), (Εικόνα 81).



Εικόνα 79. Τμήμα του κώδικα που τρέχει με το αρχικό φόρτωμα της εφαρμογής

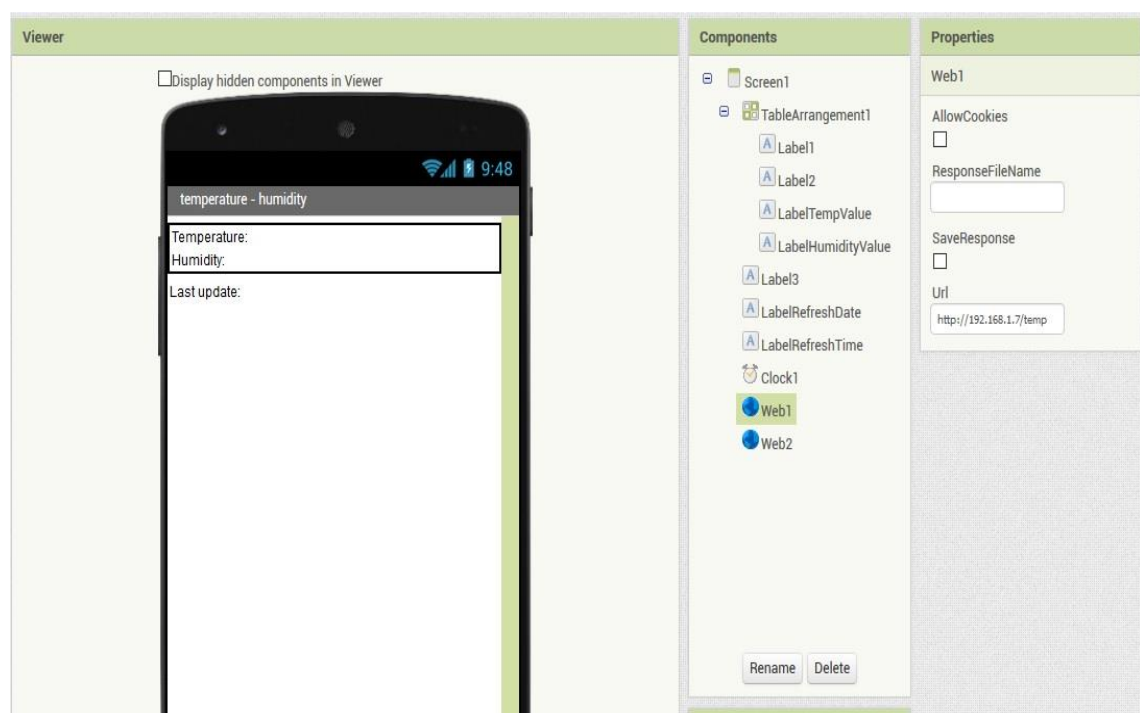


Εικόνα 80. Τμήμα του κώδικα που τρέχει όταν πυροδοτείται το Clock1



Εικόνα 81. Τμήμα του κώδικα για την διαχείριση των εισερχόμενων δεδομένων

Η εφαρμογή “url\_request” σε προβολή σχεδίασης φαίνεται παρακάτω (Εικόνα 82):



Εικόνα 82. Η εφαρμογή σε προβολή σχεδίασης

Ένα στιγμιότυπο οθόνης του τελικού αποτελέσματος (Εικόνα 83):



Εικόνα 83. Η εφαρμογή σε λειτουργία στην οθόνη του κινητού

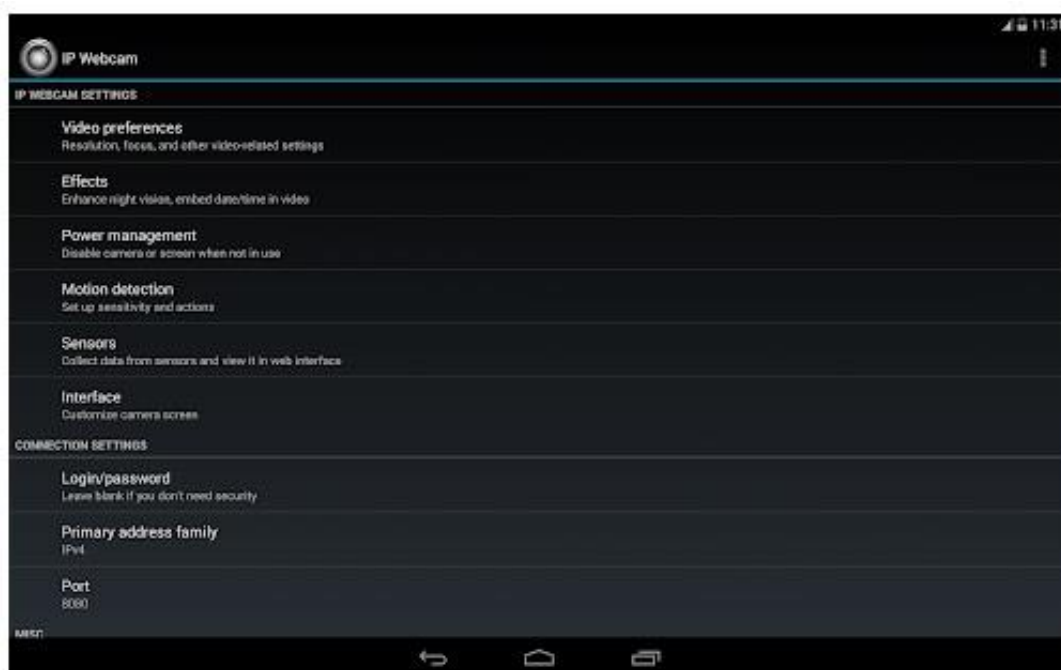
Στα πλαίσια της παρούσας εργασίας ο κώδικας της εφαρμογής “url\_request” ενσωματώνεται στην εφαρμογή “GReen-OS”.

## 4.5 Βοηθητικές εφαρμογές

### 4.5.1 mobile app “IP Webcam”

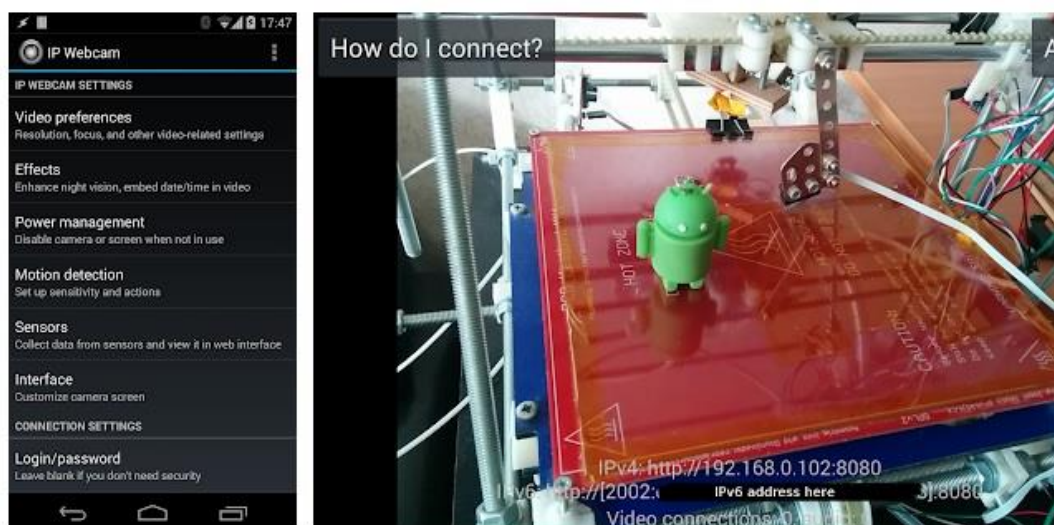
Θα χρησιμοποιηθεί η android εφαρμογή "IP Webcam" η οποία μετατρέπει το κινητό τηλέφωνό μας σε δικτυακή κάμερα με πολλαπλές επιλογές προβολής. (Εικόνα 84).





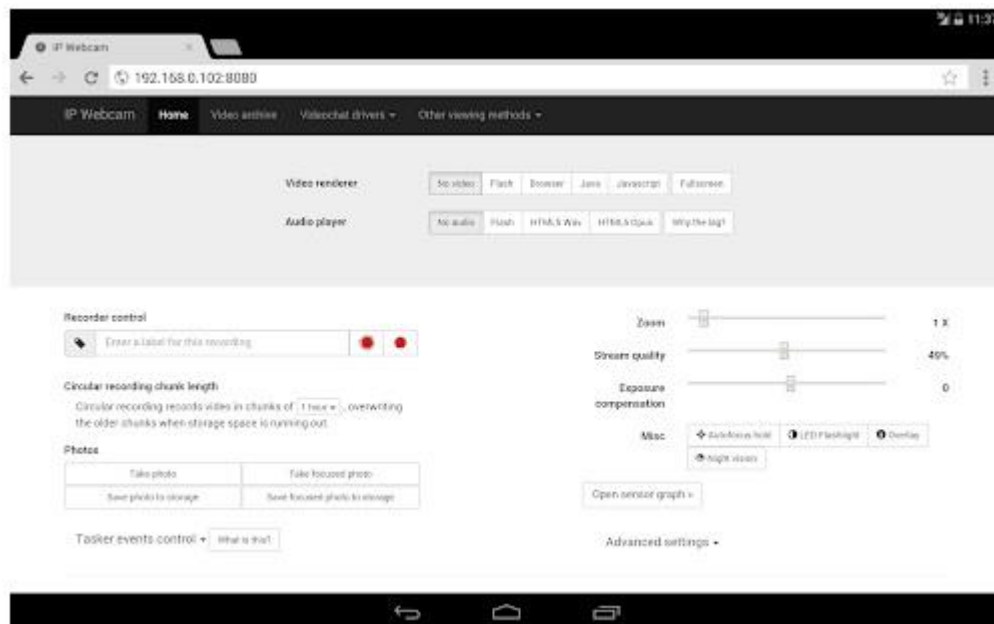
Εικόνα 84. Οι ρυθμίσεις που μας παρέχει η εφαρμογή

Με τη βοήθεια της εφαρμογής μπορούμε να δούμε video και φωτογραφίες που «τραβάει» η κάμερά μας σε οποιαδήποτε πλατφόρμα με VLC player ή web browser. (Εικόνα 85).



Εικόνα 85. Η διεπαφή με τη διεύθυνση IP του server για την εμφάνιση στον browser

Μπορούμε να δούμε Stream video μέσα στο WiFi δίκτυό μας χωρίς πρόσβαση στο διαδίκτυο. (Εικόνα 86).



Εικόνα 86. Οι διαθέσιμες ρυθμίσεις για την προβολή video ή φωτογραφιών

Πληροφορίες σχετικές με την android εφαρμογή υπάρχουν στο Google play store (Εικόνα 87).

IP Webcam turns your phone into a network camera with multiple viewing options. View your camera on any platform with VLC player or web browser. Stream video inside WiFi network without internet access.

Optional Ivideon cloud broadcasting is supported for instant global access.

Two-way audio supported in tinyCam Monitor on another android device.

Use IP Webcam with third-party MJPG software, including video surveillance software, security monitors and most audio players.

Features include:

- Video upload to Dropbox, SFTP, FTP and Email using Filoader plugin
- Several web renderers to choose from: Flash, Javascript or built-in
- Video recording in WebM, MOV, MKV or MPEG4 (on Android 4.1+)
- Audio streaming in wav, opus and AAC (AAC requires Android 4.1+)
- Motion detection with sound trigger, Tasker integration.
- Date, time and battery level video overlay.
- Sensor data acquisition with online web graphing.
- Videochat support (video stream only for Windows and Linux via an universal MJPEG video streaming driver)
- Cloud push notifications on motion and sound, cloud recording for motion-triggered records, online video broadcasting powered by Ivideon.
- Extensive baby and pet monitor features: night vision, motion detection, sound detection.

Εικόνα 87. Πληροφορίες και τεχνικά χαρακτηριστικά της εφαρμογής IP Webcam



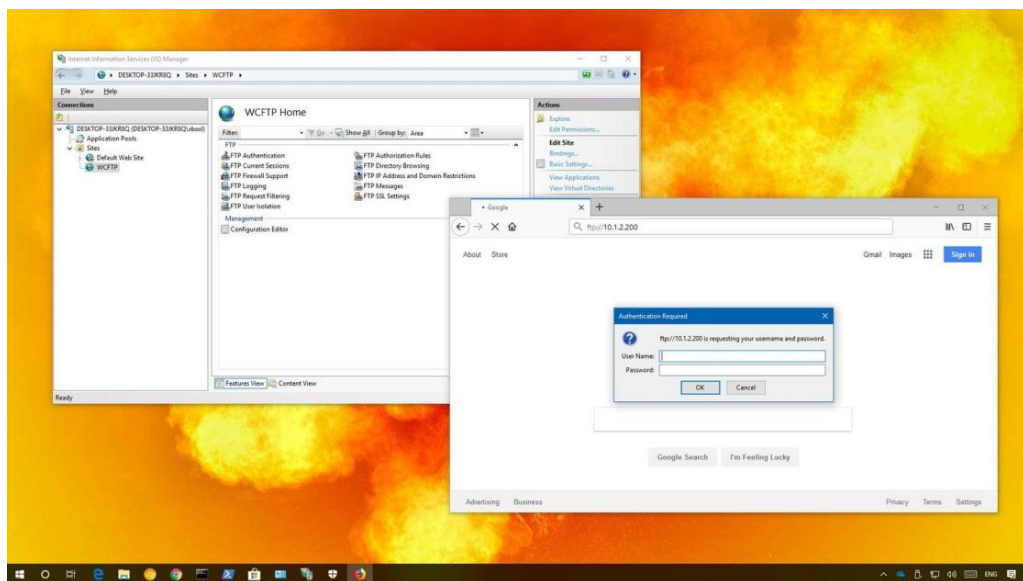
Στην περίπτωση μας η εφαρμογή “IP Webcam” χρησιμοποιείται σε συνεργασία με την εφαρμογή “GReen-OS”, μέσω του κατάλληλου κώδικα, ώστε να φωτογραφηθούν αυτόματα τα φυτά από το κινητό που βρίσκεται στο ρομπότ “farmbot-Si” και οι φωτογραφίες να αποθηκευτούν σε συγκεκριμένο φάκελο “plants” σε άλλο κινητό που τρέχει την εφαρμογή “GReen-OS” (Εικόνα 88).



Εικόνα 88. Ο κώδικας για τη λήψη και αποθήκευση των φωτογραφιών των φυτών

#### 4.5.2 FTP Server στα Windows 10

Θα δημιουργηθεί το δικό μας ιδιωτικό cloud για να γίνεται διαμοιρασμός και μεταφορά αρχείων φωτογραφιών μεταξύ smartphone και υπολογιστή, χωρίς περιορισμούς, χρησιμοποιώντας τη δυνατότητα διακομιστή FTP των Windows 10 (Εικόνα 89).



Εικόνα 89. Λειτουργία FTP διακομιστή στα Windows 10

Η εγκατάσταση ενός διακομιστή FTP (File Transfer Protocol) στα Windows 10 είναι ίσως μία από τις πιο βολικές λύσεις για τη μεταφόρτωση και λήψη αρχείων από σχεδόν οπουδήποτε στον υπολογιστή μας χωρίς τους περιορισμούς που συνήθως υπάρχουν με τις υπηρεσίες αποθήκευσης cloud.

Χρησιμοποιώντας ένα διακομιστή FTP, δημιουργούμε βασικά ένα ιδιωτικό cloud στο οποίο έχουμε απόλυτο έλεγχο. Δεν έχουμε μηνιαία ανώτατα όρια μεταφορών και οι ταχύτητες μπορούν να είναι γρήγορες (ανάλογα με τη συνδρομή μας στο διαδίκτυο).

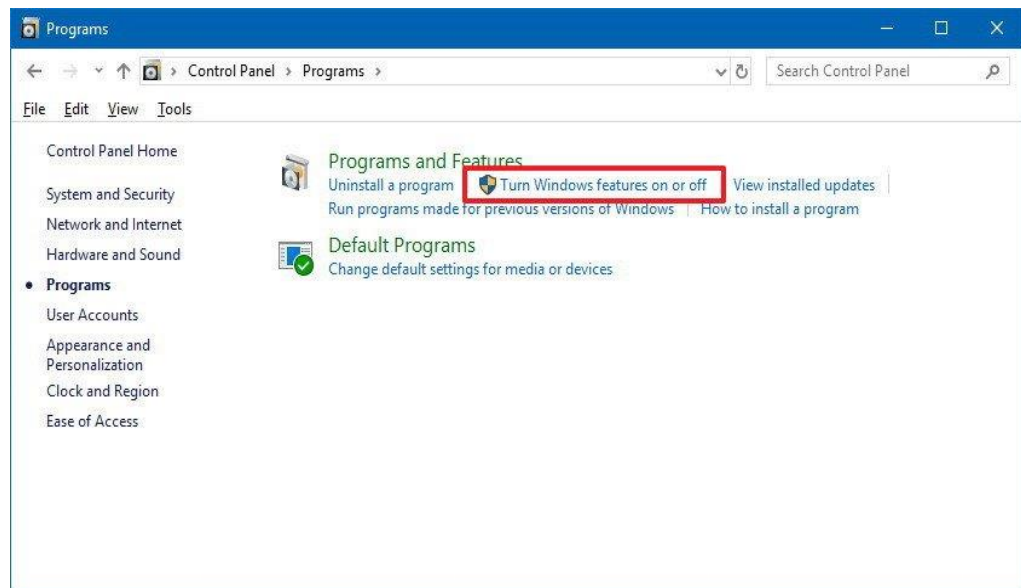
Επίσης, δεν υπάρχουν περιορισμοί τύπου ή μεγέθους αρχείων, πράγμα που σημαίνει ότι μπορούμε να μεταφέρουμε ένα αρχείο κειμένου 1KB καθώς και ένα αρχείο backup 1TB και μπορούμε να δημιουργήσουμε όσους λογαριασμούς θέλουμε για να επιτρέψουμε στον οποιονδήποτε να αποθηκεύει και να μοιράζεται αρχεία με οποιονδήποτε.

## Τρόπος εγκατάστασης των components του διακομιστή FTP στα Windows 10

Παρόλο που τα Windows 10 περιλαμβάνουν υποστήριξη για τη δημιουργία ενός διακομιστή FTP, χρειάζεται να προσθέσουμε τα απαιτούμενα components χειροκίνητα.

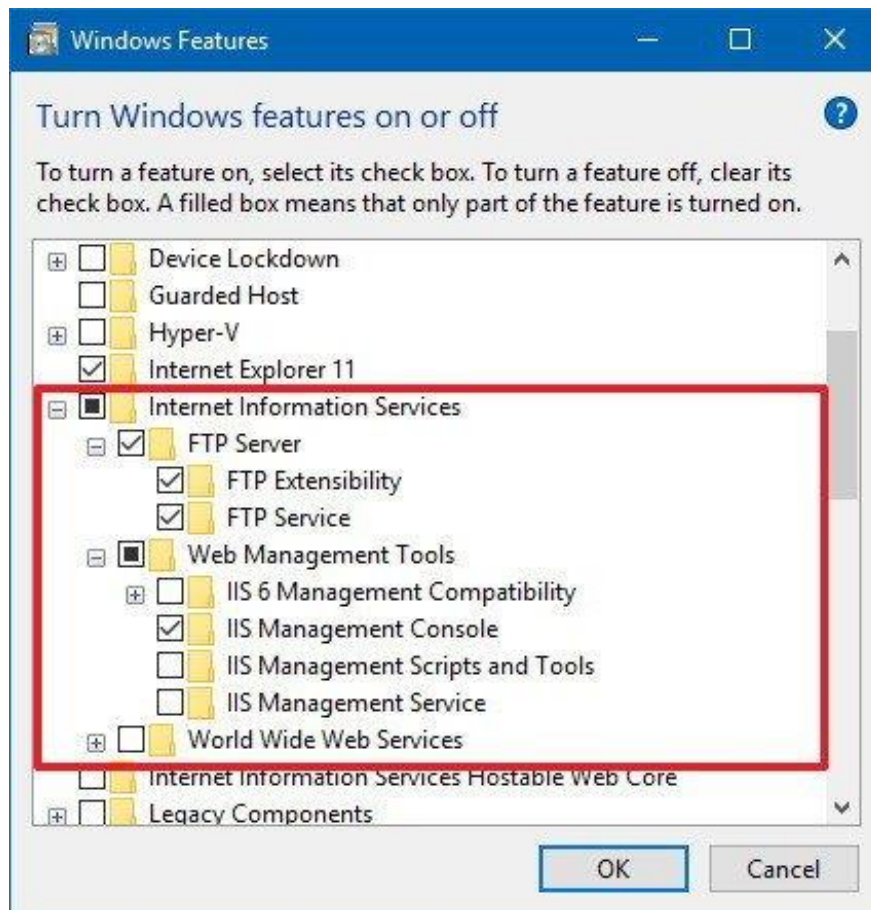
Για να εγκαταστήσουμε τα components του διακομιστή FTP, πρέπει να γίνουν τα εξής:

1. Ανοίγουμε το **Control Panel**.
2. Κάνουμε click στο **Programs**.
3. Στην ενότητα "**Programs and Features**" κάνουμε click στο σύνδεσμο **Turn Windows features on or off** (Εικόνα 90).



Εικόνα 90. Ενεργοποίηση των features των Windows

4. Αναπτύσσουμε τη λειτουργία "**Internet Information Services**" και αναπτύσσουμε την επιλογή **διακομιστή FTP**.
5. Κάνουμε check στις επιλογές **FTP Extensibility** και **FTP Service**.
6. Κάνουμε check στην επιλογή **Web Management Tools** με τις προεπιλεγμένες επιλογές, αλλά βεβαιωνόμαστε ότι είναι ενεργοποιημένη η επιλογή **IIS Management Console**. (Εικόνα 91)



Εικόνα 91. Ενεργοποίηση features του FTP Server

6. Κάνουμε click στο button **OK**.
7. Κάνουμε click στο button **Close**.

Αφού ολοκληρώσουμε τα βήματα, τα components για τη δημιουργία ενός διακομιστή FTP θα εγκατασταθούν στη συσκευή μας.

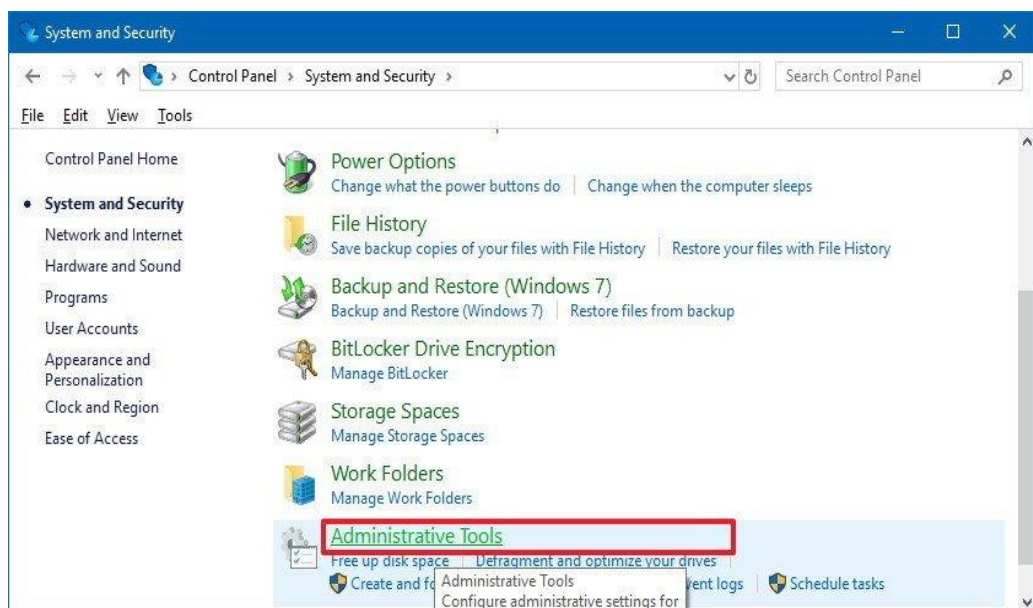
## Τρόπος ρύθμισης παραμέτρων ενός FTP server site στα Windows 10

Μετά την εγκατάσταση των απαιτούμενων components, μπορούμε να προχωρήσουμε στη διαμόρφωση ενός διακομιστή FTP στον υπολογιστή μας, ο οποίος περιλαμβάνει τη δημιουργία ενός νέου FTP site, τη ρύθμιση κανόνων τείχους προστασίας και την παροχή δυνατότητας εξωτερικών συνδέσεων.

### Ρύθμιση ενός FTP site

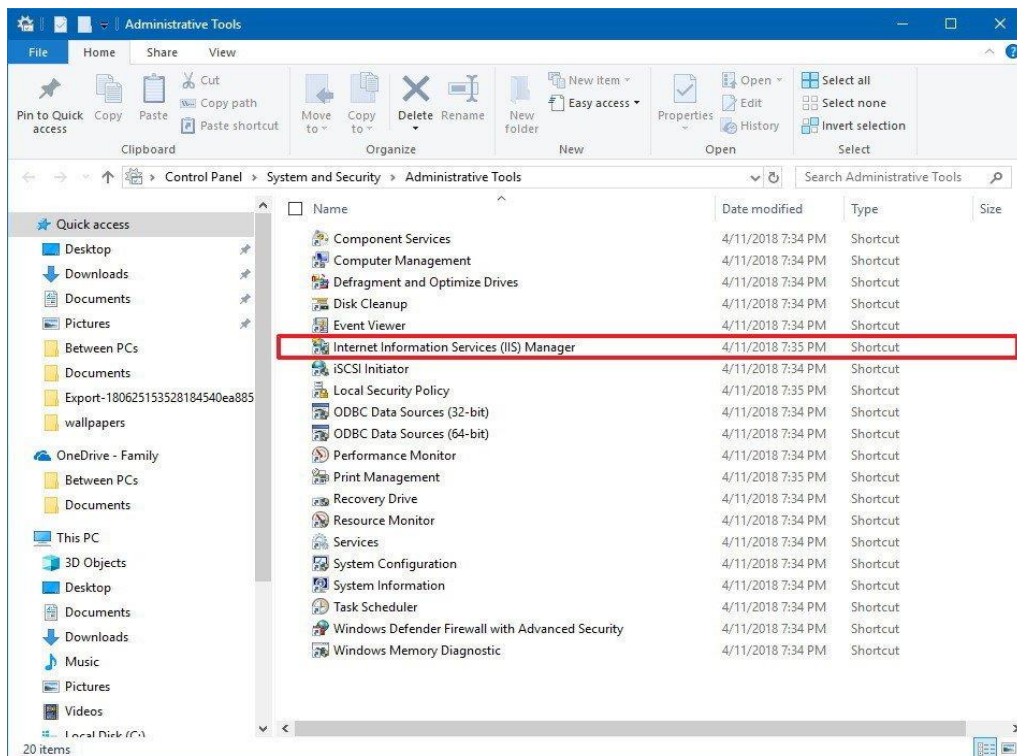
Για να ρυθμίσουμε ένα FTP site, πρέπει να γίνουν τα εξής:

1. Ανοίγουμε το **Control Panel**.
2. Κάνουμε click στο **System and Security**.
3. Κάνουμε click στο **Administrative Tools**. (Εικόνα 92)



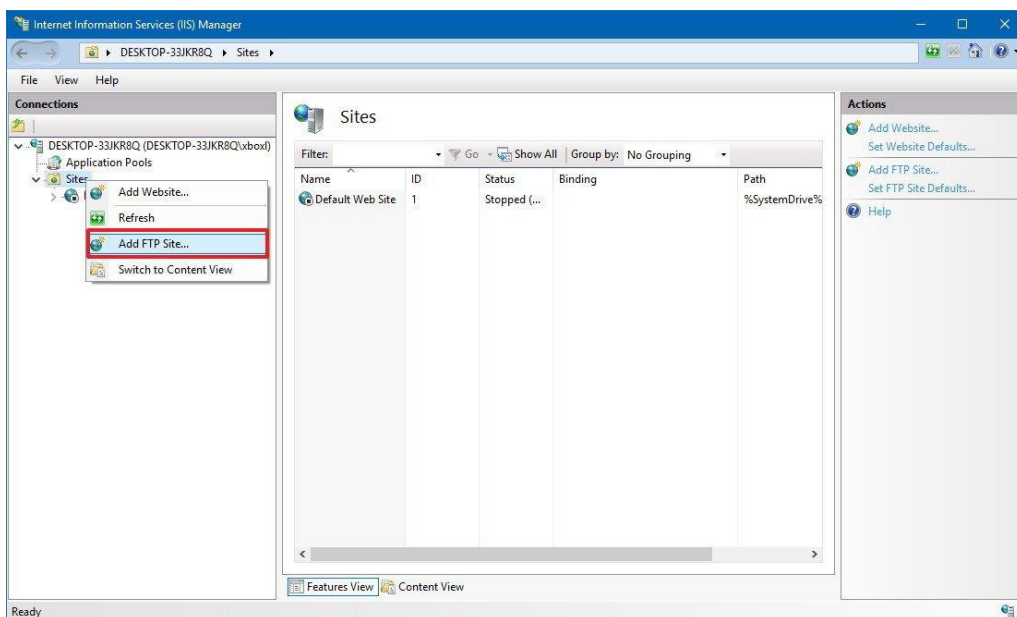
Εικόνα 92. Ενεργοποίηση των Administrative tools

4. Κάνουμε διπλό click στο εικονίδιο **Internet Information Services (IIS) Manager**. (Εικόνα 93)



Εικόνα 93. Ενεργοποίηση του Internet Information Services (IIS) Manager

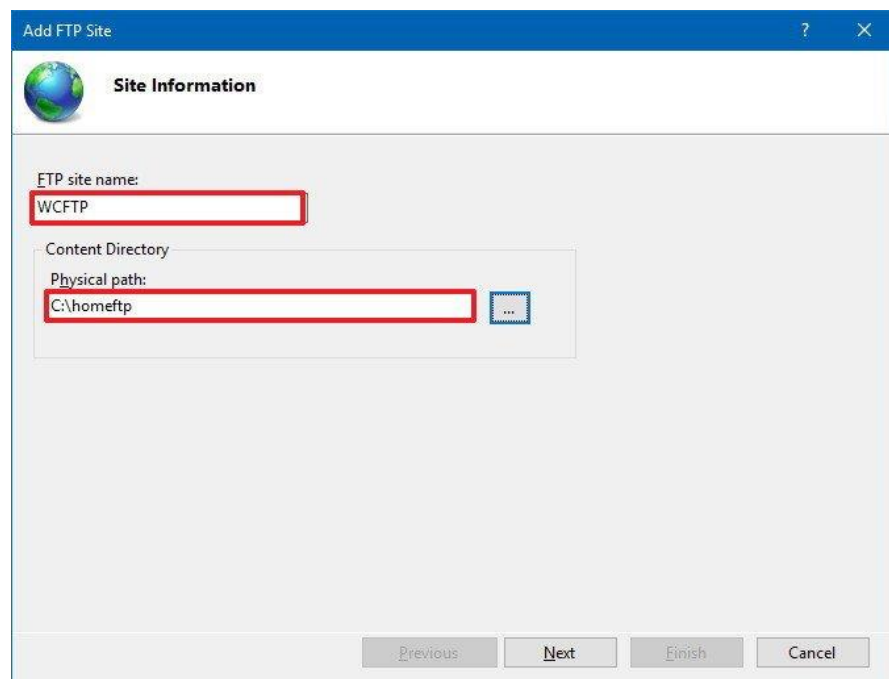
5. Στο παράθυρο "**Connections**", κάνουμε δεξί click στην επιλογή **Sites** και επιλέγουμε το **Add FTP Site**. (Εικόνα 94)



Εικόνα 94. Προσθήκη FTP Site

6. Στο όνομα του FTP site, πληκτρολογούμε ένα σύντομο περιγραφικό όνομα για τον server.
7. Στην ενότητα "**Content Directory**", στην κάτω ενότητα "**Physical path**", κάνουμε click στο κουμπί στα δεξιά για να εντοπίσουμε το φάκελο που θέλουμε να χρησιμοποιήσουμε για την αποθήκευση των αρχείων FTP. (Εικόνα 95)

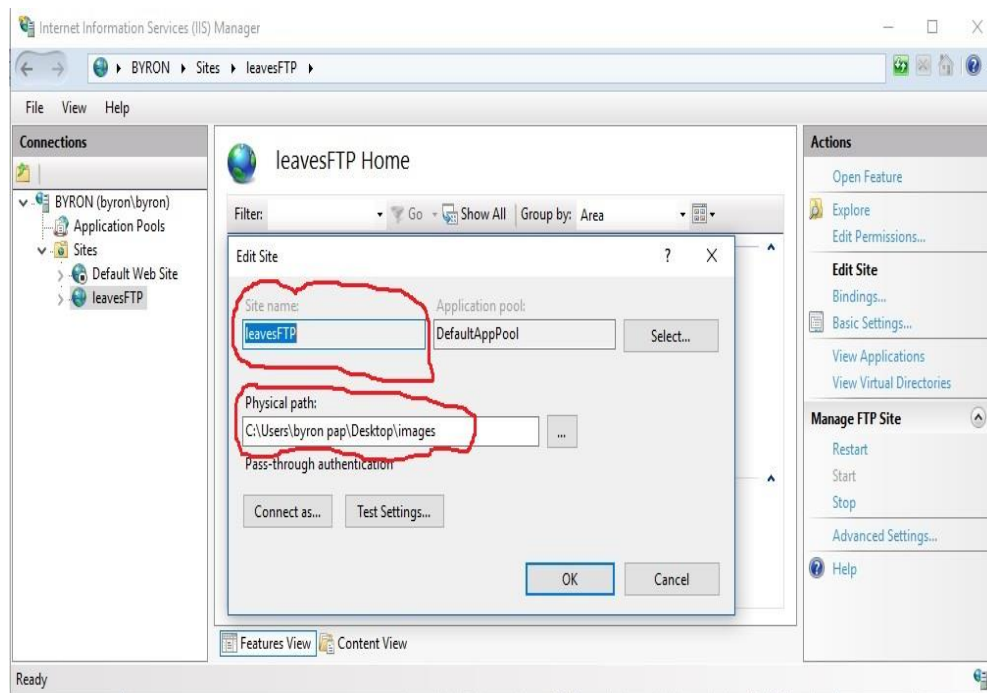
**Quick Tip:** Συνιστάται να δημιουργήσουμε ένα φάκελο στο root directory της κύριας μονάδας δίσκου του συστήματος ή σε έναν εντελώς διαφορετικό σκληρό δίσκο. Διαφορετικά, εάν ορίσουμε τον home φάκελο σε έναν από τους προεπιλεγμένους φακέλους μας κατά την προσθήκη πολλών λογαριασμών, οι άλλοι χρήστες δεν θα έχουν άδεια πρόσβασης στο φάκελο. (Μπορούμε να προσαρμόσουμε τα δικαιώματα φακέλου, αλλά δεν συνιστάται.)



Εικόνα 95. Εντοπισμός του φακέλου αποθήκευσης των αρχείων FTP

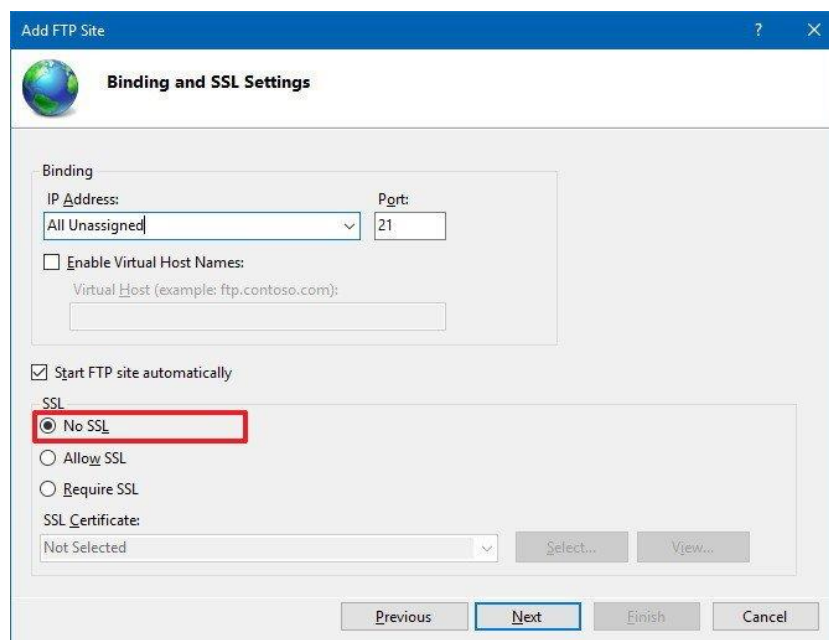
Στην περίπτωση μας σαν FTP site name πληκτρολογούμε το "**leavesFTP**" και επιλέγουμε ως physical path το "**C:\Users\byron pap\Desktop\images**" (Εικόνα 96).





Εικόνα 96. Επιλογή FTP site name και physical path

7. Κάνουμε click στο κουμπί **Next**.
8. Χρησιμοποιούμε τις προεπιλεγμένες **Binding** settings επιλογές.
9. Κάνουμε επιλογή στο **Start FTP site automatically**.
10. Στην ενότητα "**SSL**", επιλέγουμε το **No SSL**. (Εικόνα 97)

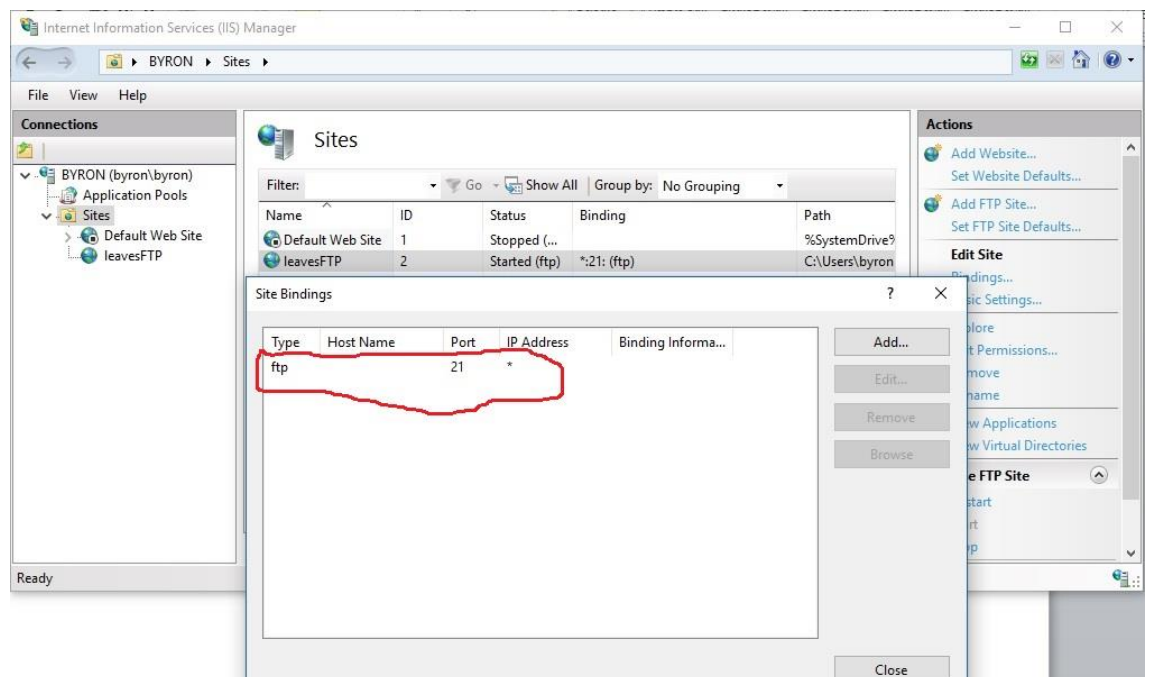


Εικόνα 97. Γενικές ρυθμίσεις Binding και SSL

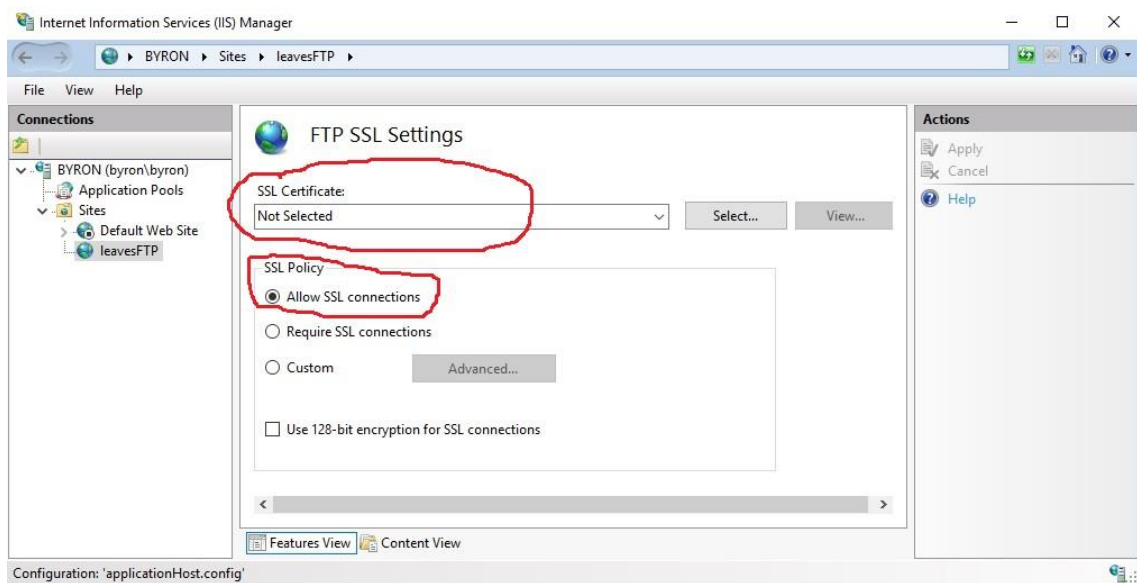


**Σημαντικό:** Σε ένα επιχειρηματικό περιβάλλον ή σε ένα διακομιστή FTP που θα φιλοξενήσει ευαίσθητα δεδομένα, η καλύτερη πρακτική είναι να διαμορφώσουμε το site να απαιτεί SSL έτσι ώστε να αποτρέψει τη μετάδοση δεδομένων σε μορφή καθαρού κειμένου.

Στην περίπτωση μας έχουμε τις εξής **Binding** (Εικόνα 98) και **SSL** (Εικόνα 99) settings.

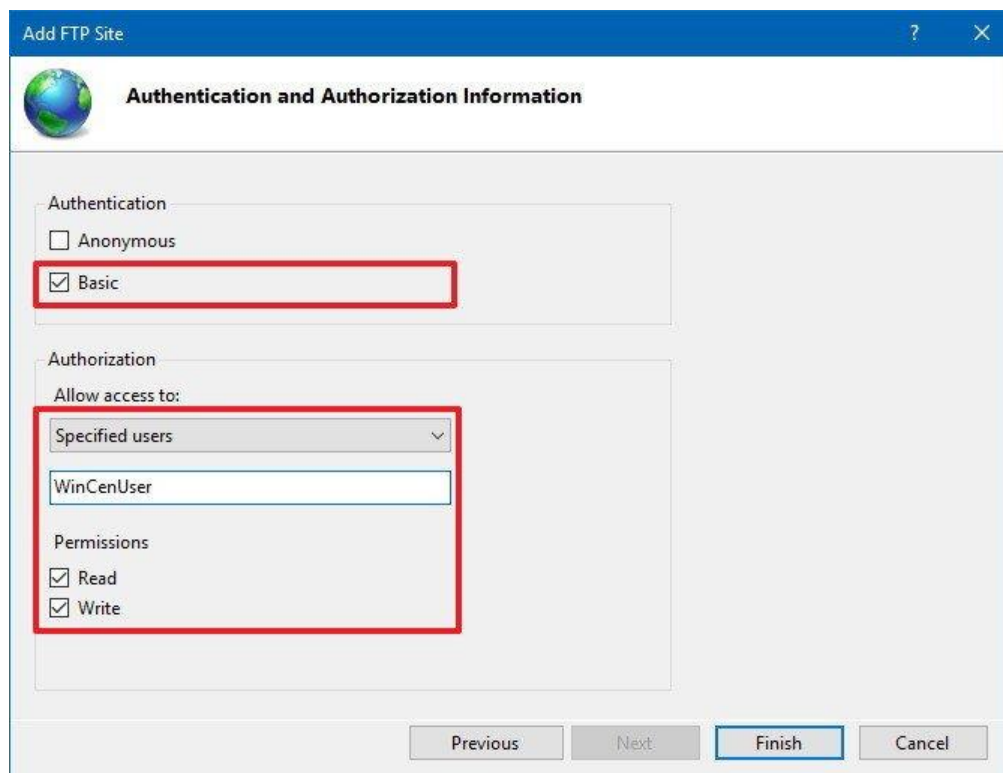


Εικόνα 98. Ρυθμίσεις Bindings στο δικό μας site



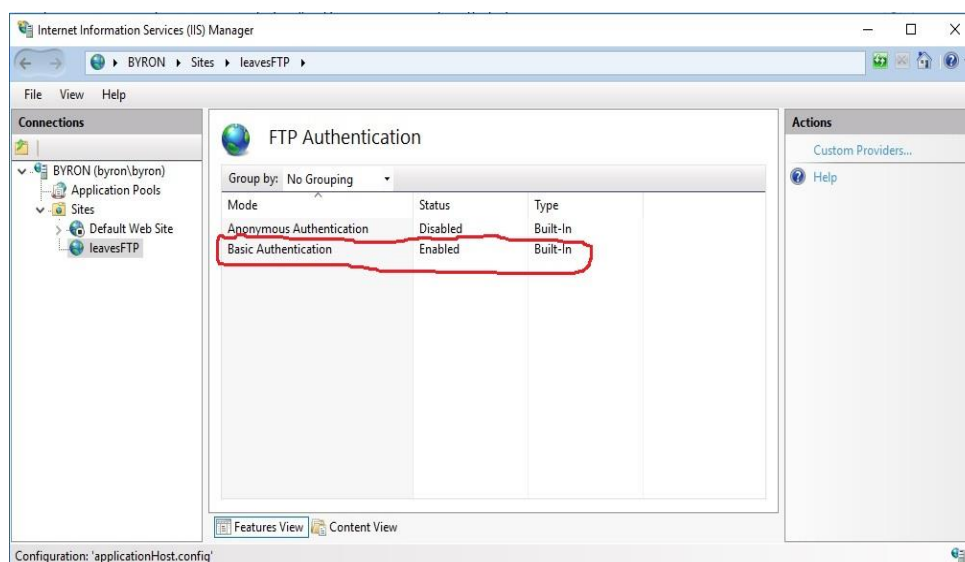
Εικόνα 99. Ρυθμίσεις SSL στο δικό μας site

11. Κάνουμε click στο button **Next**.
12. Στην ενότητα "**Authentication**", επιλέγουμε το **Basic**.
13. Στην ενότητα "**Authorization**", χρησιμοποιούμε το drop-down menu και επιλέγουμε **Specified users**.
14. Πληκτρολογούμε τη διεύθυνση ηλεκτρονικού ταχυδρομείου του λογαριασμού μας στα Windows 10 ή το όνομα του τοπικού λογαριασμού μας για να επιτρέψουμε την πρόσβαση στον διακομιστή FTP.
15. Κάνουμε επιλογή στα **Read** και **Write**. (Εικόνα 100)

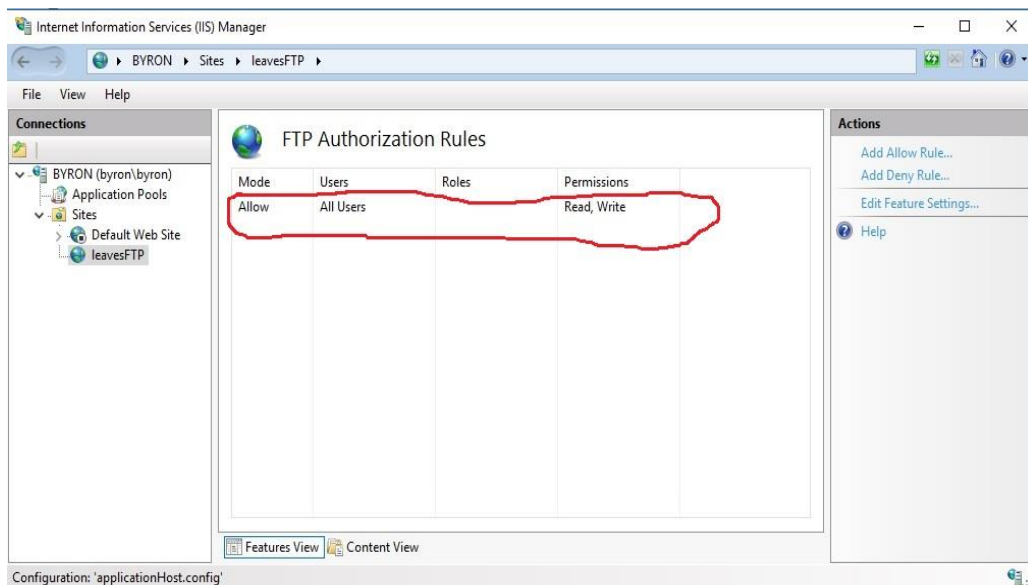


Εικόνα 100. Γενικές ρυθμίσεις Authentication και Authorization

Στην περίπτωση μας έχουμε τις εξής **Authentication** (Εικόνα 101) και **Authorization** (Εικόνα 102) settings.



Εικόνα 101. Ρυθμίσεις Authentication στο δικό μας site



Εικόνα 102. Ρυθμίσεις Authorization στο δικό μας site

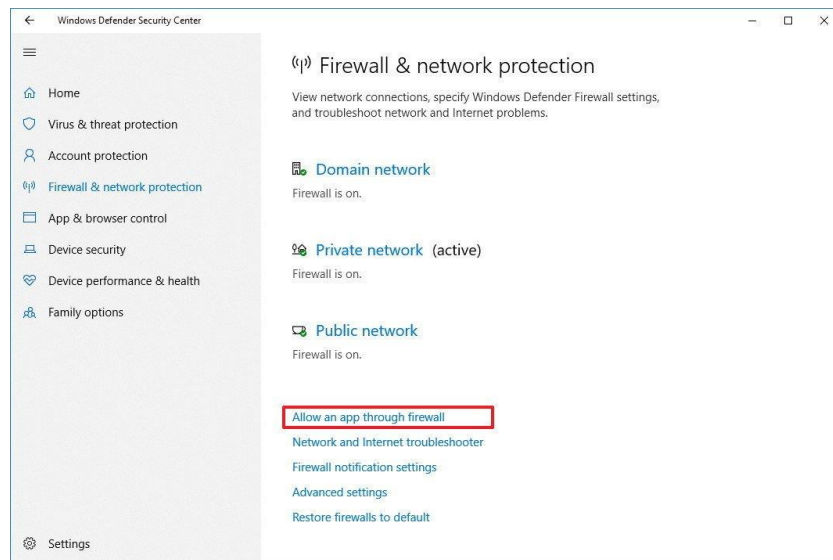
16. Κάνουμε click στο button **Finish**.

Αφού ολοκληρώσουμε τα παραπάνω βήματα, το FTP site είναι πλέον πλήρως λειτουργικό στον υπολογιστή μας.

## Ρύθμιση των κανόνων του τείχους προστασίας

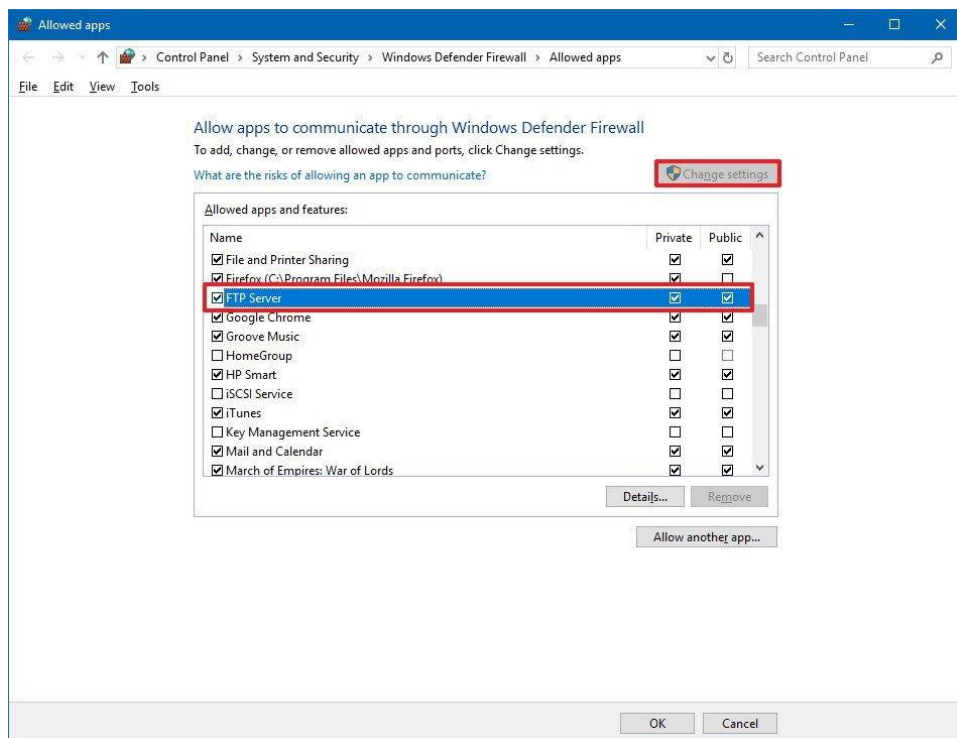
Εάν εκτελούμε το ενσωματωμένο τείχος προστασίας στα Windows 10, οι συνδέσεις με το διακομιστή FTP θα αποκλειστούν από προεπιλογή μέχρι να επιτρέψουμε χειροκίνητα την υπηρεσία, ακολουθώντας τα εξής βήματα:

1. Ανοίγουμε το **Windows Defender Security Center**.
2. Κάνουμε click στο **Firewall & network protection**.
3. Κάνουμε click στο **Allow an app through firewall**. (Εικόνα 103)



Εικόνα 103. Τροποποίηση των ρυθμίσεων του τείχους προστασίας

4. Κάνουμε click στο button **Change settings**.
5. Κάνουμε επιλογή στον **FTP Server**, όπως επίσης στα αντίστοιχα check boxes για να επιτρέψουμε πρόσβαση **Private** και **Public**. (Εικόνα 104)



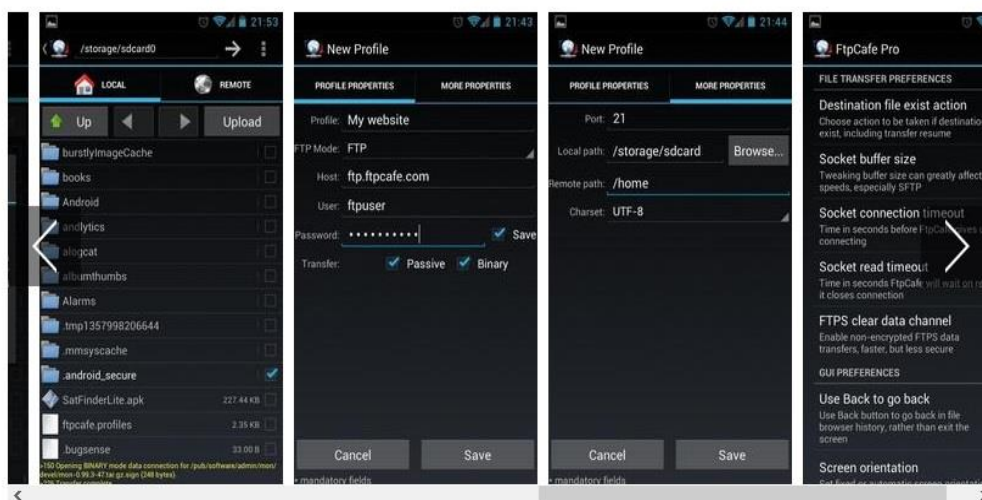
Εικόνα 104. Παροχή Private και Public πρόσβασης στον FTP Server

Μόλις ολοκληρώσουμε τα βήματα, ο διακομιστής FTP πρέπει τώρα να είναι προσβάσιμος από το τοπικό δίκτυο.

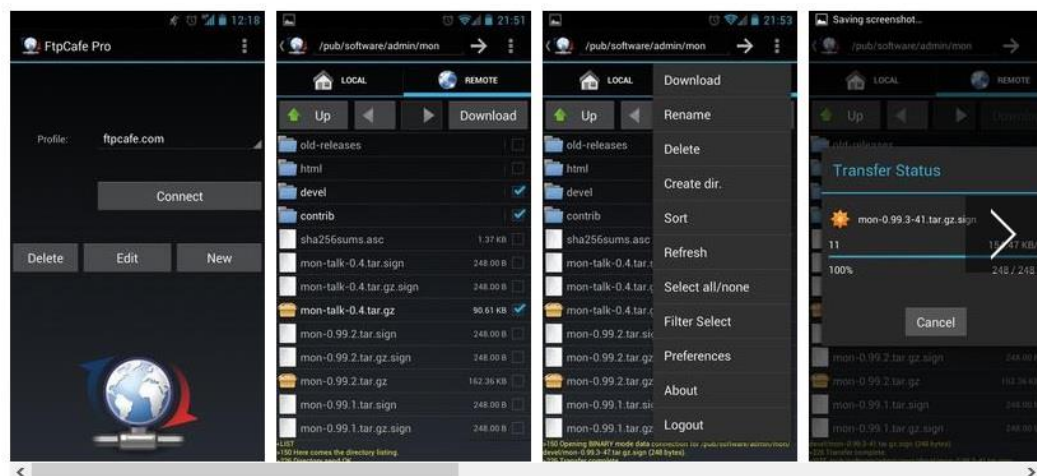
Σε περίπτωση που χρησιμοποιούμε third-party security software, βεβαιωνόμαστε ότι έχουμε ελέγξει το website υποστήριξης για πιο συγκεκριμένες λεπτομέρειες σχετικά με την προσθήκη κανόνων τείχους προστασίας.

#### 4.5.3 mobile app “FtpCafe” (FTP client)

Θα χρησιμοποιηθεί η android εφαρμογή “FtpCafe” με τη βοήθεια της οποίας μπορούμε να μεταφέρουμε πολλαπλά αρχεία και φακέλους μέσα από μια καθαρή και απλή διεπαφή χρήστη. Η εφαρμογή είναι ένας FTP client και συνεργάζεται με οποιονδήποτε FTP Server. Κάνοντας τις απαραίτητες ρυθμίσεις στον client στο κινητό (Εικόνα 105) μπορούμε να συνδεθούμε με τον FTP server και να ξεκινήσουμε την διαδικασία μεταφοράς αρχείων φωτογραφιών (Εικόνα 106)



Εικόνα 105. Ρυθμίσεις στον FTP client στο κινητό μας



Εικόνα 106. Διαδικασία μεταφοράς αρχείων φωτογραφιών στον FTP Server

Πληροφορίες σχετικές με την android εφαρμογή υπάρχουν στο Google play store (Εικόνα 107).

#### The description of FtpCafe

File transfers using:

- FTP
- FTPS (Implicit and Explicit FTP over SSL).
- SFTP (FTP over SSH). Login with password or RSA/DSA OpenSSL (Traditional SSLeay PEM) or ConnectBot (PKCS#8 PEM) private key.

- multiple file and directory transfers
- resume transfers support
- clean and simple UI

For more options use system menu, or long touch for file context menu.

[Show less ^](#)

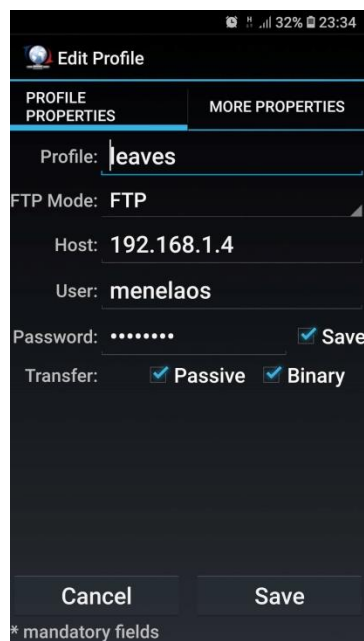
#### FtpCafe 2.7.4 Update

2019-04-08

Fixed dark text on action menu in Android 9

Εικόνα 107. Περιγραφή λειτουργίας της android εφαρμογής FtpCafe

Στην περίπτωση μας, κάνοντας τις απαραίτητες ρυθμίσεις στην εφαρμογή “FtpCafe” (Εικόνα 108), γίνεται μαζική αποστολή των φωτογραφιών των φυτών από τον φάκελο “plants” του κινητού τηλεφώνου (Εικόνα 109) προς τον φάκελο “images” του υπολογιστή που περιέχει το Diagnosis Support System (DiSS) (Εικόνα 110).

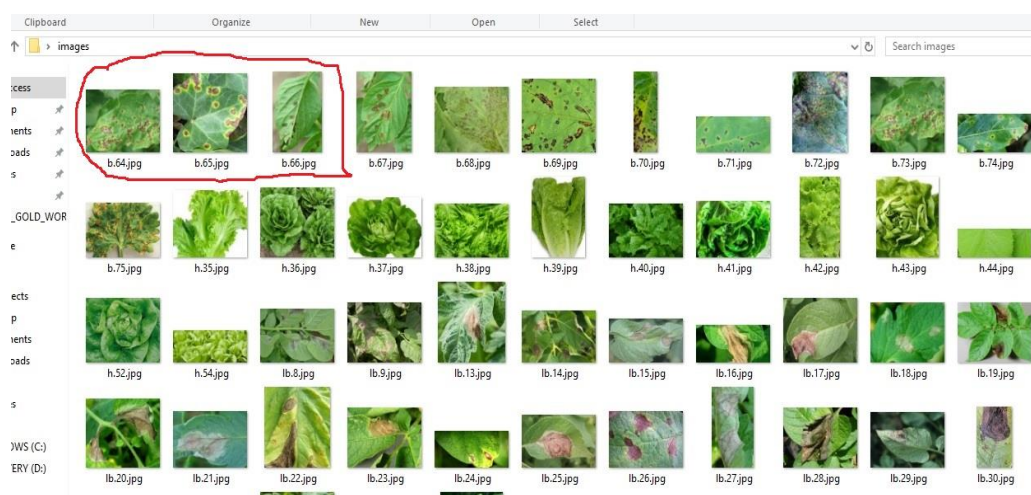


Εικόνα 108. Ρυθμίσεις για την επικοινωνία του FTP client με τον FTP Server





Εικόνα 109. Επιλογή των φωτογραφιών από τον φάκελο plants του κινητού τηλεφώνου



Εικόνα 110. Λήψη των φωτογραφιών στον φάκελο images του FTP Server

## 4.6 Προγραμματισμός του Diagnosis Support System (DiSS)

### 4.6.1 Προγραμματισμός του μοντέλου του ταξινομητή εικόνων με χρήση συνελικτικού νευρωνικού δικτύου (CNN)

Η **Μηχανική Μάθηση** είναι πλέον ένα από τα πιο "καυτά" θέματα σε όλο τον κόσμο. Αναφέρεται και ως η νέα ηλεκτρική ενέργεια στον σημερινό κόσμο. Αλλά για να είμαστε ακριβείς η **μηχανική μάθηση** είναι απλώς ένας τρόπος να διδάσκεις τη μηχανή τροφοδοτώντας την με μεγάλο όγκο δεδομένων.

Το CNN είναι ένας αλγόριθμος μηχανικής μάθησης που χρησιμεύει στο να κατανοήσουν οι μηχανές τα χαρακτηριστικά μιας εικόνας με προοπτική να θυμηθούν τα χαρακτηριστικά γνωρίσματα για να μαντέψουν το όνομα μιας νέας εικόνας που τροφοδοτείται στη μηχανή.

Ένας **ταξινομητής** μπορεί να διακρίνει εάν μια δεδομένη φωτογραφία είναι από υγιές ή ασθενικό φύλλο φυτού και από ποιά ασθένεια αυτό υποφέρει ανάλογα με τα δεδομένα που τον τροφοδοτούμε. Στα πλαίσια της εργασίας χρησιμοποιήθηκε ένας από τους διάσημους αλγόριθμους μηχανικής μάθησης για την ταξινόμηση εικόνων, δηλ. το συνελικτικό νευρωνικό δίκτυο (ή το CNN). Ένα συνελικτικό νευρωνικό δίκτυο είναι ένας τύπος νευρωνικού δικτύου (Deep Neural Network, DNN) εμπνευσμένο από το ανθρώπινο οπτικό σύστημα που χρησιμοποιείται για την επεξεργασία εικόνων. Τα CNN είναι εποπτευόμενα δίκτυα πολλαπλών επιπέδων τα οποία μπορούν να μαθαίνουν καινούρια χαρακτηριστικά αυτόματα από σύνολα δεδομένων. Τα τελευταία χρόνια, τα CNNs έχουν επιτύχει πάρα πολύ υψηλές επιδόσεις σε όλες σχεδόν τις σημαντικές εργασίες ταξινόμησης. Είναι σε θέση να εκτελούν εξαγωγή χαρακτηριστικών και ταξινόμηση κάτω από την ίδια αρχιτεκτονική.

#### Συστατικά στοιχεία των Convnets

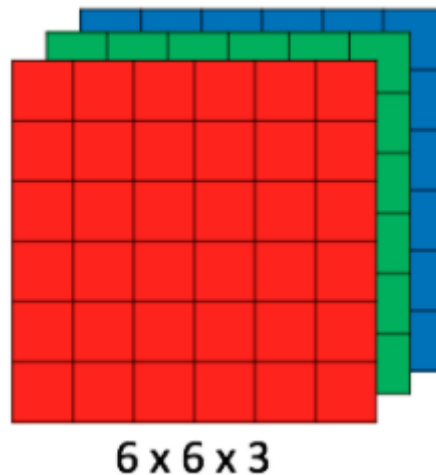
Υπάρχουν τέσσερα (4) συστατικά στοιχεία ενός Convnet

1. Convolution
2. Non Linearity (ReLU)
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)

## Κατανόηση του Συνελικτικού Νευρωνικού Δικτύου (CNN)

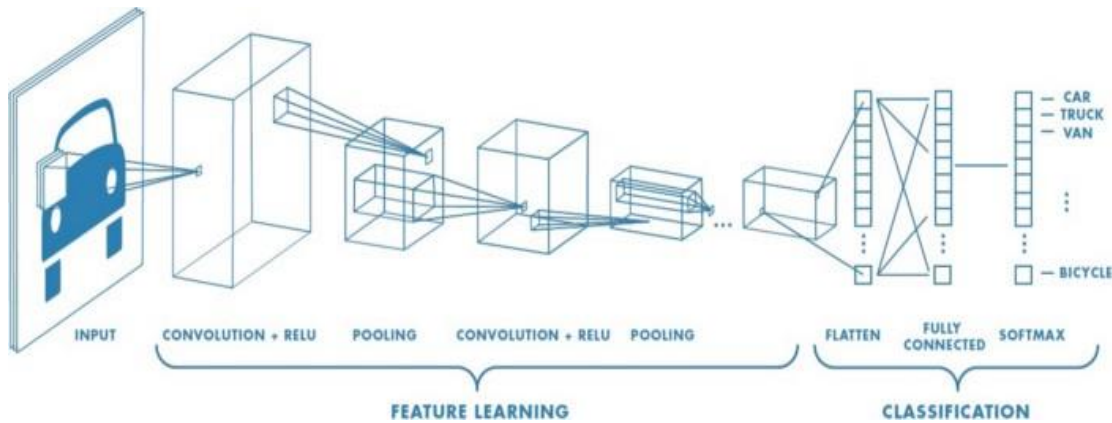
Στα νευρωνικά δίκτυα, το Convolutional νευρωνικό δίκτυο (ConvNets ή CNNs) είναι μία από τις κύριες κατηγορίες για την αναγνώριση εικόνων, την ταξινόμηση εικόνων. Η ανίχνευση αντικειμένων, η αναγνώριση προσώπων κλπ. είναι μερικές από τις περιοχές όπου χρησιμοποιούνται ευρέως τα CNN.

Τα CNN ταξινόμησης εικόνων παίρνουν μια εικόνα εισόδου, την επεξεργάζονται και την ταξινομούν κάτω από ορισμένες κατηγορίες (π.χ., σκύλος, γάτα, τίγρης, λιοντάρι). Οι υπολογιστές βλέπουν μια εικόνα εισόδου ως πίνακα εικονοστοιχείων ανάλογα με την ανάλυση εικόνας. Με βάση την ανάλυση εικόνας, θα δούμε  $h \times w \times d$  ( $h$  = ύψος,  $w$  = πλάτος,  $d$  = διάσταση). Π.χ. μια εικόνα  $6 \times 6 \times 3$  πίνακα μήτρας RGB (το 3 αναφέρεται σε τιμές RGB) και μία εικόνα  $4 \times 4 \times 1$  πίνακα μήτρας εικόνας γκρι (Εικόνα 111).



Εικόνα 111. Array of RGB Matrix

Τεχνικά, για να εκπαιδευθούν και να δοκιμαστούν τα μοντέλα CNN βαθιάς εκμάθησης, κάθε εικόνα εισόδου θα περάσει μέσα από μια σειρά από επίπεδα συνέλιξης με φίλτρα (Kernels), επίπεδα Pooling, πλήρως συνδεδεμένα επίπεδα (FC) και θα εφαρμοστεί η συνάρτηση Softmax για να ταξινομήσουν ένα αντικείμενο με πιθανοτικές τιμές μεταξύ 0 και 1. Το παρακάτω σχήμα (Εικόνα 112) είναι μια πλήρης ροή του CNN για την επεξεργασία μιας εικόνας εισόδου και την ταξινόμηση των αντικειμένων με βάση τις τιμές.

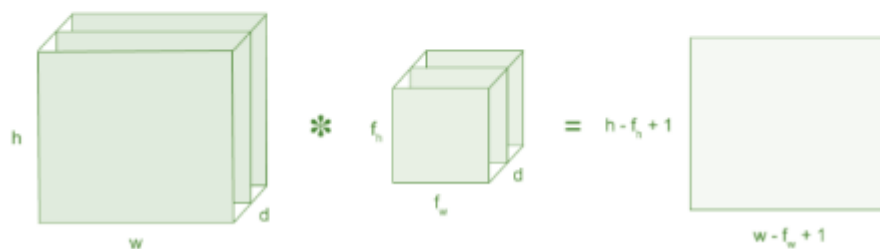


Εικόνα 112. Νευρωνικό δίκτυο με πολλά convolutional layers

### Επίπεδο Convolution (Συνέλιξη)

Το Convolution είναι το πρώτο επίπεδο που εξάγει χαρακτηριστικά από μια εικόνα εισόδου. Η συνέλιξη διατηρεί τη σχέση μεταξύ των εικονοστοιχείων μαθαίνοντας χαρακτηριστικά της εικόνας χρησιμοποιώντας μικρά τετράγωνα δεδομένων εισόδου. Πρόκειται για μια μαθηματική πράξη που παίρνει δύο εισόδους, τη μήτρα εικόνας και το φίλτρο ή πυρήνα (Εικόνα 113).

- An image matrix (volume) of dimension **( $h \times w \times d$ )**
- A filter ( **$f_h \times f_w \times d$** )
- Outputs a volume dimension  **$(h - f_h + 1) \times (w - f_w + 1) \times 1$**



Εικόνα 113. Η μήτρα εικόνας πολλαπλασιάζεται με τον πυρήνα ή τη μήτρα φίλτρου

Ας εξετάσουμε ένα πίνακα εικόνας  $5 \times 5$  του οποίου οι τιμές εικονοστοιχείων είναι 0, 1 και ένα πίνακα φίλτρου  $3 \times 3$  όπως φαίνεται παρακάτω (Εικόνα 114):

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0



1	0	1
0	1	0
1	0	1

**5 x 5 – Image Matrix**

**3 x 3 – Filter Matrix**

Εικόνα 114. Η μήτρα εικόνας πολλαπλασιάζεται με τον πυρήνα ή τη μήτρα φίλτρου

Στη συνέχεια, φαίνεται παρακάτω η συνέλιξη της μήτρας εικόνας 5 x 5 που πολλαπλασιάζεται με τη μήτρα φίλτρου 3 x 3. Η έξοδος είναι μια μήτρα 3 x 3 που ονομάζεται "Feature Map" (Εικόνα 115)

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x2</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x2</sub>	0 <sub>x0</sub>	1 <sub>x2</sub>	1	1
0	0	1	1	0
0	1	1	0	0








**Image**

4		

**Convolved  
Feature**

Εικόνα 115. Η 3 x 3 μήτρα εξόδου

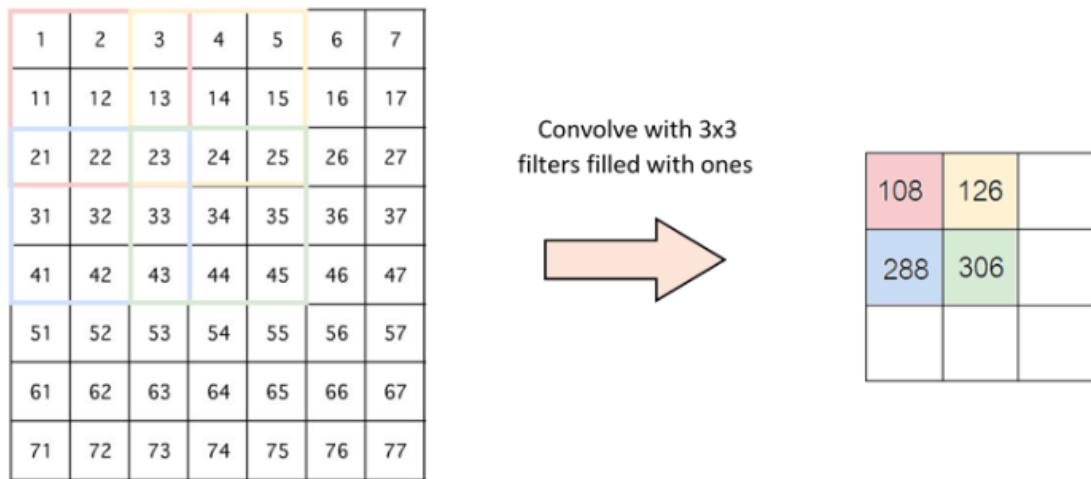
Η συνέλιξη μιας εικόνας με διαφορετικά φίλτρα μπορεί να εκτελέσει λειτουργίες όπως ανίχνευση άκρων, θόλωση και sharpen εφαρμόζοντας τα ανάλογα φίλτρα. Το παρακάτω παράδειγμα (Εικόνα 116) δείχνει διάφορες εικόνες συνέλιξης μετά την εφαρμογή διαφορετικών τύπων φίλτρων (πυρήνες).

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Εικόνα 116. Μερικά κοινά φίλτρα

## Strides

Το Stride είναι ο αριθμός των μετατοπίσεων των εικονοστοιχείων στη μήτρα εισόδου. Όταν το βήμα είναι 1 τότε μεταφέρουμε τα φίλτρα κατά 1 εικονοστοιχείο τη φορά. Όταν το βήμα είναι 2 τότε μεταφέρουμε τα φίλτρα κατά 2 εικονοστοιχεία κάθε φορά και ούτω καθεξής. Το παρακάτω σχήμα (Εικόνα 117) δείχνει πως θα λειτουργούσε η συνέλιξη με βήμα 2.



Εικόνα 117. Λειτουργία συνέλιξης με βήμα 2

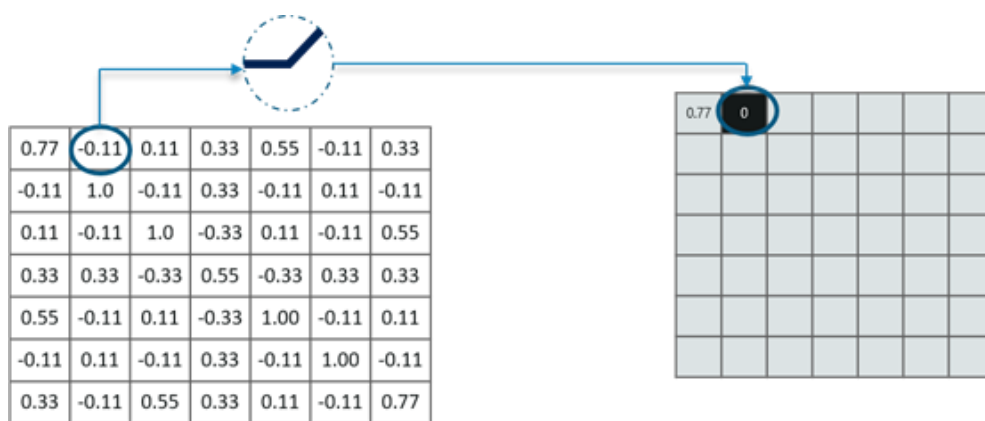
## Padding

Μερικές φορές το φίλτρο δεν ταιριάζει απόλυτα με την εικόνα εισόδου. Έχουμε δύο επιλογές:

- Βάζουμε στην εικόνα μηδενικά (zero-padding) ώστε να ταιριάζει
- Αφαιρούμε το τμήμα της εικόνας, όπου το φίλτρο δεν ταιριάζει. Αυτό ονομάζεται έγκυρο padding το οποίο διατηρεί μόνο το έγκυρο μέρος της εικόνας.

## Non Linearity (ReLU)

Το ReLU αντιπροσωπεύει τη **Rectified Linear Unit** για μια μη γραμμική λειτουργία. Η έξοδος είναι  $f(x) = \max(0, x)$ . (Εικόνα 118)



Εικόνα 118. Λειτουργία της συνάρτησης ReLU

Ο σκοπός της εφαρμογής της συνάρτησης ανορθωτή είναι να αυξήσει τη μη γραμμικότητα στις εικόνες μας. Ο λόγος που θέλουμε να το κάνουμε αυτό είναι επειδή οι εικόνες είναι μη γραμμικές από τη φύση τους.

Όταν εξετάζουμε οποιαδήποτε εικόνα, θα διαπιστώσουμε ότι περιέχει πολλά μη γραμμικά χαρακτηριστικά (π.χ. η μετάβαση μεταξύ των εικονοστοιχείων, τα όρια, τα χρώματα κ.λπ.).

Ο ανορθωτής χρησιμεύει για να σπάσει ακόμη περισσότερο τη γραμμικότητα, ώστε να αντισταθμίσουμε τη γραμμικότητα που επιβάλλουμε σε μια εικόνα όταν την βάζουμε στη λειτουργία της συνέλιξης. Στο τέλος της λειτουργίας της συνέλιξης, η έξοδος υπόκειται σε μια συνάρτηση ενεργοποίησης για να εφαρμοστεί η μη γραμμικότητα. Η συνήθης συνάρτηση ενεργοποίησης για τα convnet είναι η Relu. Ο κύριος στόχος είναι να αφαιρεθούν όλες οι αρνητικές τιμές από την συνέλιξη. Όλες οι θετικές τιμές παραμένουν ίδιες, αλλά όλες οι αρνητικές τιμές μετατρέπονται σε μηδέν.

Υπάρχουν και άλλες μη γραμμικές λειτουργίες όπως το tanh ή το sigmoid οι οποίες μπορούν επίσης να χρησιμοποιηθούν αντί της ReLU. Οι περισσότεροι επιστήμονες δεδομένων χρησιμοποιούν τη ReLU, αφού η απόδοση της ReLU είναι καλύτερη από των άλλων δύο.

## Pooling Layer

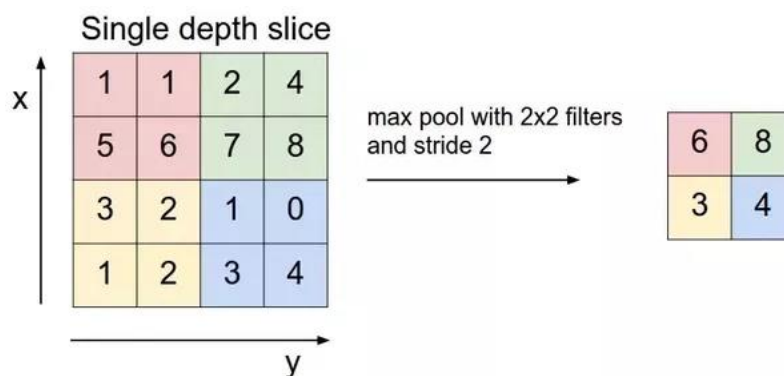
Τα επίπεδα Pooling θα μειώσουν τον αριθμό των παραμέτρων όταν οι εικόνες είναι πολύ μεγάλες. Το χωρικό pooling που ονομάζεται επίσης υποδειγματοληψία μειώνει το



dimensionality κάθε χάρτη, αλλά διατηρεί τις σημαντικές πληροφορίες. Το χωρικό pooling μπορεί να είναι διαφόρων τύπων:

- Max Pooling
- Average Pooling
- Sum Pooling

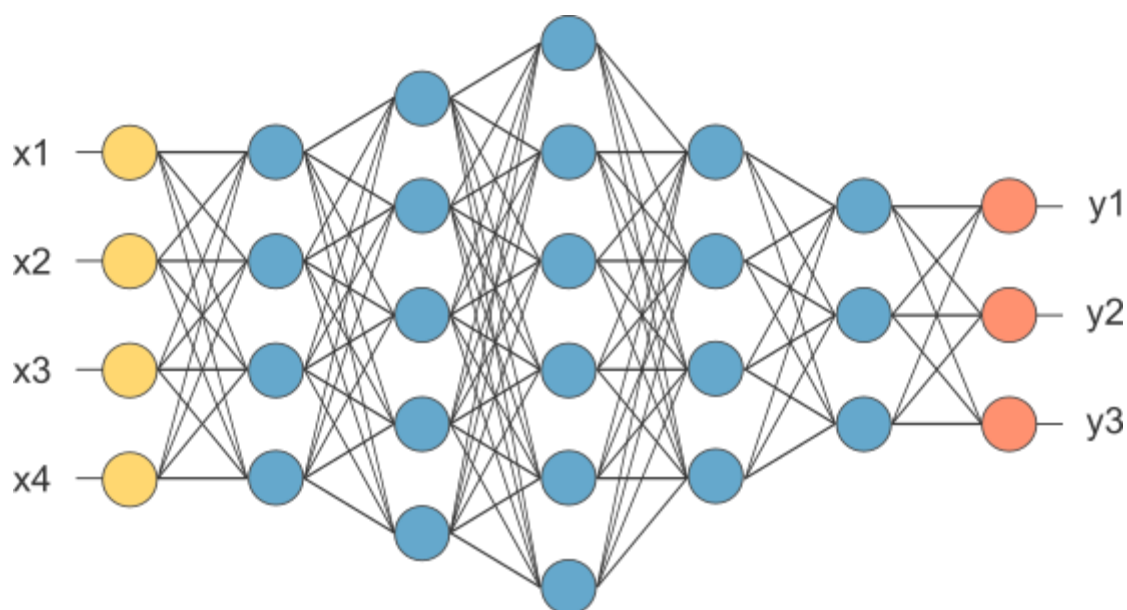
Το Max pooling παίρνει το μεγαλύτερο στοιχείο από τον rectified χάρτη χαρακτηριστικών (Εικόνα 119). Όπως παίρνουμε το μεγαλύτερο στοιχείο θα μπορούσαμε επίσης να πάρουμε το average pooling. Το άθροισμα όλων των στοιχείων στο χάρτη χαρακτηριστικών καλείται sum pooling.



Εικόνα 119. Max Pooling

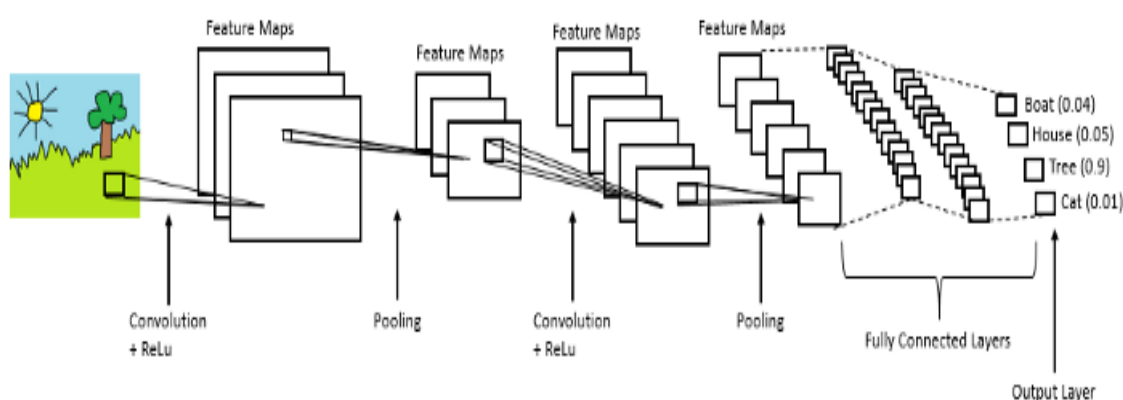
### Fully Connected Layer

Το επίπεδο που ονομάζουμε FC, κατατάσσει τη μήτρα μας σε διάνυσμα και την τροφοδοτεί σε ένα πλήρως συνδεδεμένο επίπεδο όπως το νευρωνικό δίκτυο (Εικόνα 120).



Εικόνα 120. After pooling layer, flattened as FC layer

Στο παραπάνω διάγραμμα, ο πίνακας χαρακτηριστικών θα μετατραπεί ως διάνυσμα ( $x_1, x_2, x_3, \dots$ ). Με τα fully connected layers, συνδυάζουμε αυτά τα χαρακτηριστικά μαζί για να δημιουργήσουμε ένα μοντέλο. Τέλος, έχουμε μια συνάρτηση ενεργοποίησης όπως η softmax ή η sigmoid για να ταξινομήσουμε τις εξόδους ως γάτα, σκύλος, αυτοκίνητο, φορτηγό κ.λπ. (Εικόνα 121)



Εικόνα 121. Πλήρης αρχιτεκτονική του CNN

Ο στόχος του fully connected layer είναι να πεπλατύσει τα χαρακτηριστικά υψηλού επιπέδου που μαθαίνονται από τα επίπεδα συνέλιξης και να συνδυάσει όλα τα χαρα-

κτηριστικά. Περνάει την πεπλατυσμένη έξοδο στο επίπεδο εξόδου όπου χρησιμοποιείται ένας ταξινομητής softmax για την πρόβλεψη της ετικέτας της κλάσης εισόδου.

## Σύνοψη

- Παρέχουμε την εικόνα εισόδου στο επίπεδο συνέλιξης
- Επιλέγουμε παραμέτρους, εφαρμόζουμε φίλτρα με strides, padding εάν απαιτείται. Εκτελούμε συνέλιξη στην εικόνα και εφαρμόζουμε την ενεργοποίηση ReLU στη μήτρα.
- Εκτελούμε pooling για να μειώσετε το μέγεθος των διαστάσεων
- Προσθέτουμε όσα επίπεδα συνέλιξης χρειαστεί
- Κάνουμε flatten την έξοδο και την τροφοδοτούμε σε ένα πλήρως συνδεδεμένο επίπεδο (FC Layer)
- Εξάγουμε την κλάση χρησιμοποιώντας μια συνάρτηση ενεργοποίησης (Logistic Regression με συναρτήσεις κόστους) και ταξινομούμε τις εικόνες.

## Διαφορά μεταξύ παραδοσιακού νευρωνικού δικτύου και συνελκτικού νευρωνικού δικτύου

Το πιο κρίσιμο συστατικό του μοντέλου είναι το επίπεδο της συνέλιξης. Αυτό το μέρος στοχεύει στη μείωση του μεγέθους της εικόνας για ταχύτερους υπολογισμούς των βαρών και στη βελτίωση της γενίκευσης.

Κατά τη διάρκεια της φάσης συνέλιξης, το δίκτυο κρατάει τα βασικά χαρακτηριστικά της εικόνας και αποκλείει τον άσχετο θόρυβο. Για παράδειγμα, το μοντέλο μαθαίνει πώς να αναγνωρίζει έναν ελέφαντα από μια εικόνα με ένα βουνό στο βάθος. Εάν χρησιμοποιήσουμε ένα παραδοσιακό νευρωνικό δίκτυο, το μοντέλο θα αποδώσει βάρος σε όλα τα εικονοστοιχεία, συμπεριλαμβανομένων και εκείνων από το βουνό, που δεν είναι απαραίτητα και μπορεί να παραπλανήσει το δίκτυο.

Αντ' αυτού, ένα συνελκτικό νευρωνικό δίκτυο θα χρησιμοποιήσει μια μαθηματική τεχνική για την εξαγωγή μόνο των πιο σχετικών εικονοστοιχείων. Αυτή η μαθηματική λειτουργία ονομάζεται συνέλιξη. Αυτή η τεχνική επιτρέπει στο δίκτυο να μαθαίνει όλο και πιο σύνθετα χαρακτηριστικά σε κάθε επίπεδο. Η συνέλιξη χωρίζει τον πίνακα σε μικρά κομμάτια για να μάθει τα περισσότερα βασικά στοιχεία μέσα σε κάθε κομμάτι.

### Απαραίτητες Βιβλιοθήκες για την υλοποίηση του προτεινόμενου CNN:

- **TFLearn** – μια αρθρωτή και διαφανής βιβλιοθήκη Deep Learning, που διαθέτει API υψηλότερου επιπέδου για το Tensorflow, προκειμένου να διευκολύνει και να επιταχύνει τις πειραματικές διαδικασίες, παραμένοντας όμως πλήρως διαφανής και συμβατή με αυτό. Το TFLearn είναι χτισμένο στην κορυφή του Tensorflow και χρησιμοποιείται για τη δημιουργία επιπέδων του CNN.

Στα χαρακτηριστικά του TFLearn περιλαμβάνονται:

- Εύκολο στη χρήση και στην κατανόηση υψηλού επιπέδου API για την υλοποίηση deep neural networks, με εκπαιδευτικό υλικό και παραδείγματα.
  - Γρήγορη κωδικοποίηση με ενσωματωμένα αρθρωτά επίπεδα νευρωνικών δικτύων, κανονικοποιητές, βελτιστοποιητές, μετρικές κλπ ...
  - Πλήρης διαφάνεια έναντι του Tensorflow. Όλες οι λειτουργίες είναι χτισμένες πάνω από τους tensors και μπορούν να χρησιμοποιηθούν ανεξάρτητα από το TFLearn.
  - Ισχυρές βοηθητικές λειτουργίες για την εκπαίδευση οποιουδήποτε γραφήματος TensorFlow, με υποστήριξη πολλαπλών εισόδων, εξόδων και βελτιστοποιητών.
  - Εύκολη και όμορφη γραφική απεικόνιση, με λεπτομέρειες σχετικά με τα βάρη, τις κλίσεις, τις δράσεις και πολλά άλλα.
  - Εύκολη τοποθέτηση συσκευών για χρήση πολλαπλών CPU / GPU.
- **tqdm** – Κάνουμε αμέσως τους βρόχους μας να δείχνουν έναν έξυπνο μετρητή προόδου, με μια απλή σχεδίαση. Ο καλύτερος / μόνος τρόπος για να προσθέσουμε γραμμές προόδου στον κώδικα Python, από το αραβικό "taqadum" που σημαίνει "πρόοδος", είναι μία από τις πολλές βιβλιοθήκες της Python που είναι τόσο χρήσιμες όσο και εύκολες στη χρήση τους.
  - **numpy** – χρησιμοποιείται για την επεξεργασία των πινάκων εικόνας. Είναι μια βιβλιοθήκη για τη γλώσσα προγραμματισμού Python, προσθέτοντας υποστήριξη για μεγάλες, πολυδιάστατες συστοιχίες και μήτρες, μαζί με μια μεγάλη συλλογή μαθηματικών λειτουργιών υψηλού επιπέδου για να λειτουργούν σε αυτές τις συστοιχίες.

- **open-cv** – χρησιμοποιείται για επεξεργασία της εικόνας όπως μετατροπή της σε κλίμακα του γκρι και κλπ. Πρόκειται για μια βιβλιοθήκη Python που έχει σχεδιαστεί για να λύσει τα προβλήματα μηχανικής όρασης. Το OpenCV αναπτύχθηκε αρχικά το 1999 από την Intel, αλλά αργότερα υποστηρίχθηκε από τη Willow Garage.

Το OpenCV υποστηρίζει μια μεγάλη ποικιλία γλωσσών προγραμματισμού όπως C ++, Python, Java κλπ. Υποστηρίζεται από πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows, Linux και MacOS.

Το OpenCV Python δεν είναι παρά μια κλάση περιτυλίγματος για την αρχική βιβλιοθήκη C ++ για να χρησιμοποιηθεί με την Python. Χρησιμοποιώντας το, όλες οι δομές πινάκων OpenCV μετατρέπονται από / σε Numpy πίνακες.

Αυτό διευκολύνει την ενσωμάτωσή της με άλλες βιβλιοθήκες που χρησιμοποιούν το NumPy. Για παράδειγμα, βιβλιοθήκες όπως οι SciPy και Matplotlib.

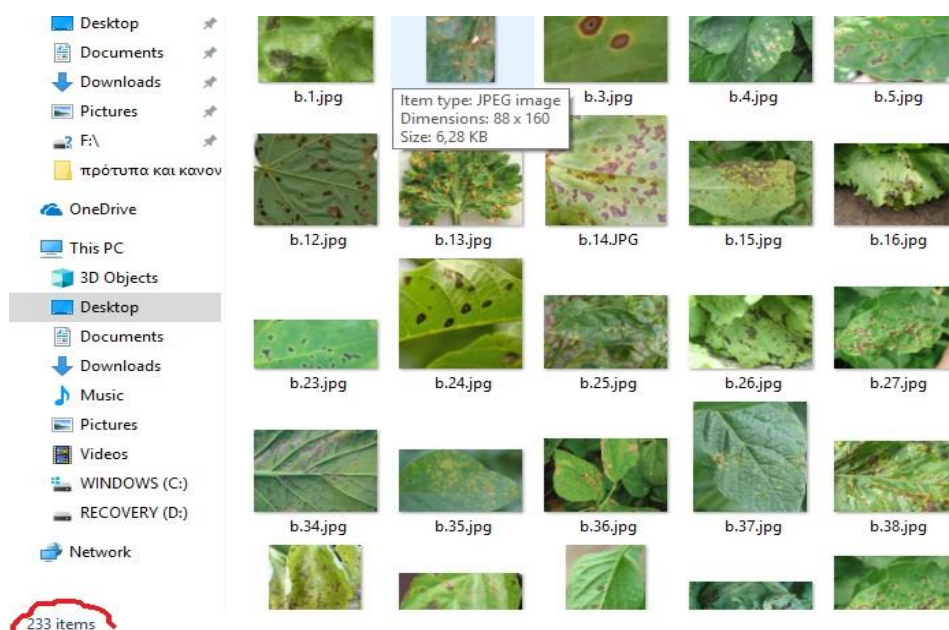
- **os** – για να έχουμε πρόσβαση στο σύστημα αρχείων για να διαβάσουμε την εικόνα από τον κατάλογο train και test από το υπολογιστικό μας σύστημα.
- **random** – για να ανακατέψουμε τα δεδομένα για να ξεπεραστεί η πόλωση.
- **matplotlib** – για να εμφανιστεί το αποτέλεσμα της πρόγνωσής μας. Πρόκειται για μια σχεδιαστική βιβλιοθήκη για τη γλώσσα προγραμματισμού Python και την μαθηματική της επέκταση.
- **tensorflow** – είναι μια βιβλιοθήκη Python για γρήγορους αριθμητικούς υπολογισμούς που δημιουργήθηκε και κυκλοφόρησε από την Google. Είναι μια βιβλιοθήκη θεμέλιο που μπορεί να χρησιμοποιηθεί για τη δημιουργία μοντέλων Deep Learning απευθείας ή με τη χρήση βιβλιοθηκών περιτυλίγματος που απλοποιούν τη διαδικασία που χτίζεται πάνω από το TensorFlow. Μπορεί να λειτουργεί σε συστήματα με μια CPU, σε GPUs καθώς και σε κινητές συσκευές και μεγάλης κλίμακας καταναμημένα συστήματα εκατοντάδων μηχανών. Χρησιμοποιείται μόνο για τη χρήση του tensorboard για να συγκρίνουμε τις καμπύλες απωλειών και adam των αποτελεσμάτων μας ή με τις ληφθείσες καταγραφές.

Ο ταξινομητής εικόνας δημιουργήθηκε σε Python IDE (IDLE 3.6 64-bit) χρησιμοποιώντας το TFLearn (πάνω από το Tensorflow) για τον εντοπισμό της ασθένειας των φύλλων των φυτών. Ο κώδικας βρίσκεται στο αρχείο cnp.py.

Για το dataset χρησιμοποιήθηκαν οι ακόλουθοι κατάλογοι:

- **train dataset** Χωρίζεται σε train set και validation set με αναλογία 80% - 20% αντίστοιχα. Αυτό γίνεται κατά την διαδικασία του model building για να αποφύγουμε το overfitting.

Για την περίπτωση μας υπάρχουν 233 φωτογραφίες (Εικόνα 122) αποθηκευμένες στον κατάλογο train, άρα θα χρησιμοποιηθούν οι τελευταίες 46 ως validation set ( $233 \cdot 20\% = 46,6$ ) (Εικόνα 123).



Εικόνα 122. Ο φάκελος train που περιέχει 233 φωτογραφίες

```

File Edit Format Run Options Window Help
convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 6
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

#Fully Connected Layer with SoftMax as Activation Function
convnet = fully_connected(convnet, 4, activation='softmax')
#Regression for ConvNet with ADAM optimizer
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss='categori

model = tflearn.DNN(convnet, tensorboard_dir='log')

if os.path.exists('{} .meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

# Splitting Our Test and Train Data
train = train_data[:-46]
test = train_data[-46:]

#This is our Training data
X = np.array([i[0] for i in train]).reshape(-1, IMG_SIZE, IMG_SIZE, 3)
Y = [i[1] for i in train]

#This is our Training data
test_x = np.array([i[0] for i in test]).reshape(-1, IMG_SIZE, IMG_SIZE, 3)

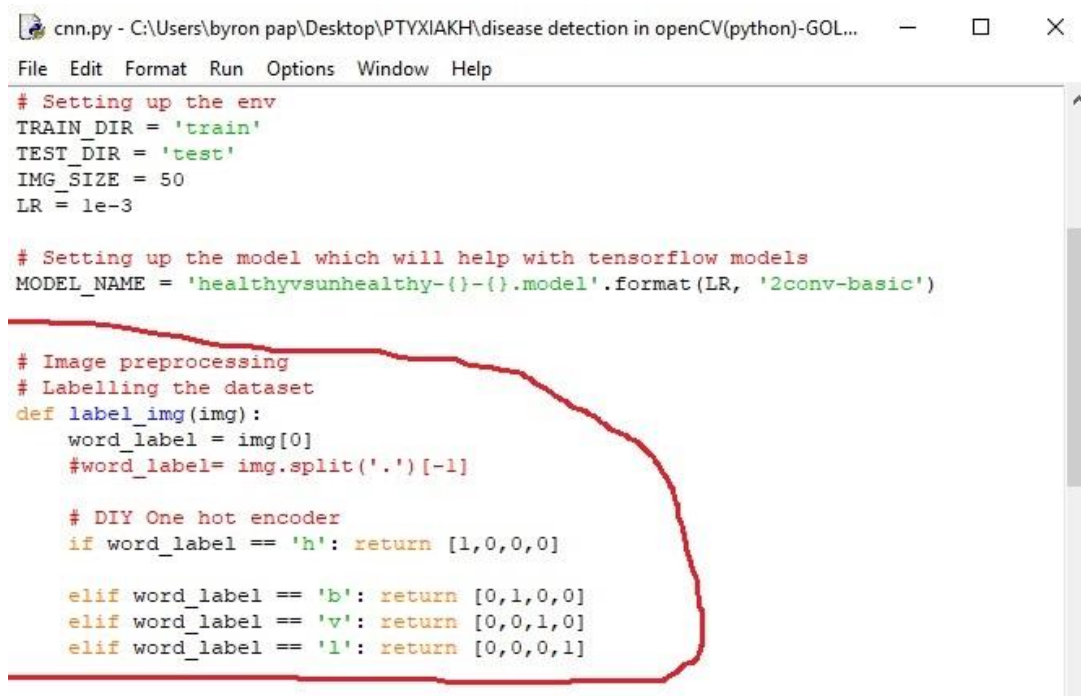
```

Εικόνα 123. Οι τελευταίες 46 φωτογραφίες χρησιμοποιούνται ως validation set

Στον παραπάνω κώδικα ως test εννοείται το validation set.

- **test dataset** Το test dataset είναι ανεξάρτητο του train dataset. Χρησιμοποιείται για να εντοπιστούν τα χαρακτηριστικά απόδοσης (accuracy).

Τώρα, μετά τη λήψη του dataset απαιτείται να κάνουμε μια μικρή προεπεξεργασία των δεδομένων και να δώσουμε ετικέτες σε κάθε εικόνα που μας δίνεται κατά την εκπαίδευση του dataset. Για να το πετύχουμε αυτό, παρατηρούμε ότι το όνομα κάθε εικόνας του dataset εκπαίδευσης ξεκινάει είτε με “h” για “healthy” ή “b” για “bacterial spot” ή “v” για “viral (yellow leaf curl virus)” ή “l” για “late blight”, έτσι μπορεί να κατασκευαστεί ένα hot encoder για να κατανοήσουν οι μηχανές τις ετικέτες (h[1, 0,0,0] ή b[0, 1,0,0] ή v[0,0,1,0] ή l[0,0,0,1]) (Εικόνα 124).



```
cnnp.py - C:\Users\byron pap\Desktop\PTYXIAKH\disease detection in openCV(python)-GOL...
File Edit Format Run Options Window Help
# Setting up the env
TRAIN_DIR = 'train'
TEST_DIR = 'test'
IMG_SIZE = 50
LR = 1e-3

# Setting up the model which will help with tensorflow models
MODEL_NAME = 'healthyvsunhealthy-{}-{}.model'.format(LR, '2conv-basic')

# Image preprocessing
# Labelling the dataset
def label_img(img):
    word_label = img[0]
    #word_label= img.split('.')[0]

    # DIY One hot encoder
    if word_label == 'h': return [1,0,0,0]

    elif word_label == 'b': return [0,1,0,0]
    elif word_label == 'v': return [0,0,1,0]
    elif word_label == 'l': return [0,0,0,1]
```

Εικόνα 124. Image preprocessing για τη δημιουργία hot encoder

Τα TRAIN\_DIR και TEST\_DIR ρυθμίζονται από τον χρήστη και είναι εύκολο να παίξει κανείς με τις βασικές hyperparameters όπως epoch, learning rate, κλπ για να βελτιώσει την ακρίβεια. Οι εικόνες μετατρέπονται σε έγχρωμες με σκοπό να σχηματιστούν οι 3-d μήτρες στις οποίες θα εφαρμοστεί το CNN.

Το προτεινόμενο μοντέλο αποτελείται από πέντε επίπεδα συνέλιξης, ακολουθούμενα από ένα επίπεδο max-pooling. Ακολουθεί ένα πλήρως συνδεδεμένο επίπεδο MLP (multilayer perceptron) 1024 νευρώνων. Το τελικό επίπεδο είναι επίσης ένα πλήρως συνδεδεμένο επίπεδο 4 νευρώνων, όσοι και οι κατηγορίες εξόδου. Η συνάρτηση ενεργοποίησης ReLu εφαρμόζεται στην έξοδο κάθε επιπέδου συνέλιξης και στο πρώτο πλήρως συνδεδεμένο επίπεδο.

Το πρώτο επίπεδο συνέλιξης φιλτράρει την εικόνα εισόδου με 32 πυρήνες μεγέθους 3x3. Αφού εφαρμοστεί max-pooling, η έξοδος δίνεται ως είσοδος για το δεύτερο επίπεδο συνέλιξης με 64 πυρήνες μεγέθους 3x3. Μετά την εφαρμογή του max-pooling, η έξοδος δίνεται ως είσοδος για το τρίτο επίπεδο συνέλιξης με 128 πυρήνες μεγέθους 3x3. Μετά την εφαρμογή του max-pooling, η έξοδος δίνεται ως είσοδος για το τέταρτο επίπεδο συνέλιξης με 32 πυρήνες μεγέθους 3x3. Το τελευταίο επίπεδο συνέλιξης έχει 64 πυρήνες μεγέθους 3x3 ακολουθούμενο από ένα πλήρως συνδεδεμένο επίπεδο 1024



νευρώνων. Το επόμενο επίπεδο είναι το τελευταίο και είναι ένα πλήρως συνδεδεμένο επίπεδο το οποίο χρησιμοποιεί την συνάρτηση ενεργοποίησης softmax για να παράγει μια κατανομή πιθανότητας των τεσσάρων (4) κατηγοριών εξόδου.

Το μοντέλο εκπαιδεύεται χρησιμοποιώντας adaptive moment estimation (Adam) με μέγεθος παρτίδας 500 για 3 epochs.

Στην ανάλυση εκπαίδευσης μπορούμε να εξάγουμε 4 μετρικές:

- i) loss,
- ii) accuracy,
- iii) validation loss,
- iv) validation accuracy

Έχουν αναπτυχθεί αρκετές τεχνικές για να ξεπεραστεί η υπερφόρτωση, όπως η αύξηση των δεδομένων. Η αύξηση δεδομένων γίνεται μόνο στα δεδομένα εκπαίδευσης. Δεδομένου ότι το CNN απαιτεί μεγάλο όγκο δεδομένων, η αύξηση δεδομένων χρησιμοποιείται για την αύξηση των δεδομένων εκπαίδευσης. Με την αύξηση των δεδομένων, μπορούμε να αποτρέψουμε το νευρωνικό μας δίκτυο να μαθαίνει άσχετα μοτίβα, ενισχύοντας ουσιαστικά τη συνολική απόδοση. Πραγματοποιήθηκε ένα πείραμα για να αυξηθούν τα δεδομένα εκπαίδευσης, περιστρέφοντας, αναστρέφοντας, ανασχεδιάζοντας τις εικόνες. Το αποτέλεσμα δείχνει ότι από μόνη της η αύξηση των δεδομένων επιλύει σημαντικά την υπερφόρτωση. Αυξάνει επίσης την ακρίβεια επικύρωσης.

Στην παρακάτω (Εικόνα 125) φαίνεται ένα τμήμα του κώδικα από το αρχείο cnn.py σε γλώσσα python. Με το αρχείο αυτό δημιουργείται το συνελκτικό νευρωνικό δίκτυο του ταξινομητή, το οποίο διαβάζει τις φωτογραφίες, εξάγει τα χαρακτηριστικά τους και τα ταξινομεί σε 4 διαφορετικές κατηγορίες:

- i. “healthy”
- ii. “bacterial” (bacterial spot)
- iii. “viral” (yellow leaf curl virus)
- iv. “lateblight” (late blight)

```

convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

#Conv Layer 1
convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 2
convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 3
convnet = conv_2d(convnet, 128, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 4
convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 5
convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

#Conv Layer 6
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

#Fully Connected Layer with SoftMax as Activation Function
convnet = fully_connected(convnet, 4, activation='softmax')
#Regression for ConvNet with ADAM optimizer
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss='categori

model = tflearn.DNN(convnet, tensorboard_dir='log')

if os.path.exists('{} .meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

```

Εικόνα 125. Ο κώδικας python του αρχείου cnn.py για τη δημιουργία του CNN

Κάθε φορά που κάποιος «τρέχει» το αρχείο αυτό, το νευρωνικό δίκτυο ξαναεκπαιδύεται να μαθαίνει καινούρια χαρακτηριστικά για τις κατηγορίες αυτές. Η διαδικασία επανεκπαίδευσης δημιουργεί ένα αρχείο που αποθηκεύει το εκπαιδευμένο μοντέλο. Αυτό το εκπαιδευμένο μοντέλο περιλαμβάνει όλα τα χαρακτηριστικά που εξάγονται κατά τη διάρκεια της διαδικασίας εκπαίδευσης για όλες τις κατηγορίες. Έτσι, κάθε φορά που επιθυμούμε να γίνει μια διάγνωση, το εκπαιδευμένο μοντέλο συγκρίνεται με μια καινούρια εικόνα, αναζητά τα σχετιζόμενα χαρακτηριστικά, εφαρμόζει τη συνάρτηση softmax στην έξοδο του fully connected layer και προβάλλει μια λίστα από κατηγορίες

(κλάσεις) με τις αντίστοιχες πιθανότητες τους ταξινομημένες κατά φθίνουσα σειρά. Η κλάση που έχει την υψηλότερη πιθανότητα είναι η πρόβλεψη του δικτύου.

Το συγκεκριμένο νευρωνικό δίκτυο εκπαιδεύτηκε σε ένα train dataset 233 φωτογραφιών και πέτυχε μια ακρίβεια ταξινόμησης της τάξης του 98,65% (Εικόνα 126).

```
:00, 165.91it/s] 94%|#####3| 218/233 [00:01<00:00, 162.33it/s]100%|#####  
###| 233/233 [00:01<00:00, 176.53it/s]  
model loaded!  
-----  
Run id: healthyvsunhealthy-0.001-2conv-basic.model  
Log directory: log/  
-----  
Training samples: 187  
Validation samples: 46  
--  
Training Step: 4231 | total loss: [1m][32m0.22025][0m][0m | time: 1.640s  
[2K] Adam | epoch: 001 | loss: 0.22025 - acc: 0.9686 -- iter: 064/187  
[A][A Training Step: 4232 | total loss: [1m][32m0.20739][0m][0m | time: 2.340s  
[2K] Adam | epoch: 001 | loss: 0.20739 - acc: 0.9717 -- iter: 128/187  
[A][A Training Step: 4233 | total loss: [1m][32m0.19511][0m][0m | time: 4.140s  
[2K] Adam | epoch: 001 | loss: 0.19511 - acc: 0.9746 | val_loss: 0.07662 - val  
_acc: 1.0000 -- iter: 187/187  
--  
Training Step: 4234 | total loss: [1m][32m0.18370][0m][0m | time: 0.650s  
[2K] Adam | epoch: 002 | loss: 0.18370 - acc: 0.9771 -- iter: 064/187  
[A][A Training Step: 4235 | total loss: [1m][32m0.17313][0m][0m | time: 1.450s  
[2K] Adam | epoch: 002 | loss: 0.17313 - acc: 0.9794 -- iter: 128/187  
[A][A Training Step: 4236 | total loss: [1m][32m0.16319][0m][0m | time: 3.150s  
[2K] Adam | epoch: 002 | loss: 0.16319 - acc: 0.9815 | val_loss: 0.06292 - val  
_acc: 1.0000 -- iter: 187/187  
--  
Training Step: 4237 | total loss: [1m][32m0.15347][0m][0m | time: 0.620s  
[2K] Adam | epoch: 003 | loss: 0.15347 - acc: 0.9833 -- iter: 064/187  
[A][A Training Step: 4238 | total loss: [1m][32m0.14441][0m][0m | time: 1.270s  
[2K] Adam | epoch: 003 | loss: 0.14441 - acc: 0.9850 -- iter: 128/187  
[A][A Training Step: 4239 | total loss: [1m][32m0.13558][0m][0m | time: 2.970s  
[2K] Adam | epoch: 003 | loss: 0.13558 - acc: 0.9865 | val_loss: 0.05116 - val  
_acc: 1.0000 -- iter: 187/187  
--  
>>> |
```

Εικόνα 126. Το μοντέλο εκπαιδεύεται και έτσι αυξάνεται η ακρίβεια ταξινόμησης

#### 4.6.2 Προγραμματισμός της διεπαφής χρήστη για τη διάγνωση των ασθενειών

Για να γίνει η διάγνωση της ασθένειας των φύλλων των φυτών δημιουργείται το κατάλληλο αρχείο σε Python IDE (IDLE 3.6 64-bit). Ο κώδικας βρίσκεται στο αρχείο

ui.py. Τώρα, έχοντας στην διάθεσή του το εκπαιδευμένο δίκτυο, μπορεί ο καθένας να επιλέξει καινούριες εικόνες για διάγνωση και να δει πώς το μοντέλο ταξινομεί αυτές τις εικόνες. Για τη διάγνωση χρησιμοποιείται η μέθοδος `model.predict([data])[0]` της βιβλιοθήκης `tflearn` (Εικόνα 127).

```
for num, data in enumerate(verify_data):

    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3, 4, num + 1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE, IMG_SIZE, 3)
    # model_out = model.predict([data])[0]
    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 0:
        str_label = 'healthy'
    elif np.argmax(model_out) == 1:
        str_label = 'bacterial'
    elif np.argmax(model_out) == 2:
        str_label = 'viral'
    elif np.argmax(model_out) == 3:
        str_label = 'lateblight'
```

Εικόνα 127. Χρήση της μεθόδου `model.predict` για τη διάγνωση

Για να είναι εύχρηστη η εφαρμογή ακόμα και από κάποιον αρχάριο δημιουργείται μια φιλική διεπαφή χρήστη. Οι εικόνες εξόδου δεν θα είναι πολύ καθαρές δεδομένου ότι όλες οι εικόνες μειώνονται στο μέγεθος των 50X50 pixels ώστε το υπολογιστικό σύστημα να μπορεί να τις επεξεργαστεί γρήγορα μέσω της εναλλαγής μεταξύ ταχύτητας και απώλειας.

Τα βήματα που θα ακολουθήσει ο χρήστης για να πάρει το αποτέλεσμα της διάγνωσης από το έμπειρο σύστημα DiSS είναι τα ακόλουθα:

α) αρχικά εκτελεί το αρχείο `ui.py` που αποτελεί τον πυρήνα του συστήματος διάγνωσης (DiSS) και ξεκινάει το γραφικό περιβάλλον επικοινωνίας της διεπαφής (Εικόνα 128).

β) στη συνέχεια επιλέγει από τον υπολογιστή του (Εικόνα 129) μια φωτογραφία του φυτού για το οποίο θέλει να γίνει διάγνωση.

γ) το επόμενο βήμα είναι να ζητήσει διάγνωση του φυτού της φωτογραφίας πατώντας το button “Analyze Image” (Εικόνα 130). Εδώ συμβαίνουν τα εξής:

Έχοντας το μοντέλο εκπαιδευμένο, το χρησιμοποιούμε για να κάνουμε προβλέψεις για τις εικόνες των φυτών.

Όταν σκεφτόμαστε ένα μοντέλο που επιλέγει μεταξύ διαφόρων κλάσεων εξόδου, έχουμε στο μυαλό μας την «εμπιστοσύνη» αυτής της επιλογής ως κατανομή πιθανοτήτων σε σχέση με τις πιθανές επιλογές. Για να πάρουμε αυτή τη κατανομή, χρειαζόμαστε μια συνάρτηση για να μετατρέψουμε το επίπεδο εξόδου του νευρωνικού δικτύου μας. Η συνάρτηση που χρησιμοποιείται για το σκοπό αυτό είναι η softmax.

Προκειμένου να έχουμε την πρόβλεψη, διαβάζουμε και προ-επεξεργαζόμαστε την εικόνα εισόδου με τον ίδιο τρόπο όπως κάναμε στην εκπαίδευση του μοντέλου.

Η πρόβλεψη γίνεται εξάγοντας τις τιμές των χαρακτηριστικών από την εικόνα εισόδου και συγκρίνοντάς τις με τις αντίστοιχες τιμές των χαρακτηριστικών από τις εικόνες κάθε μιας από τις 4 κλάσεις. Σε τεχνικό επίπεδο, αφού περάσουμε την εικόνα εισόδου μέσω όλων των επιπέδων του δικτύου μας, προσθέτουμε τις τιμές του vector table κάθε κλάσης και το άθροισμα το διαιρούμε με το άθροισμα των τιμών του vector table της εικόνας εισόδου. Το πηλίκο μας δίνει την πιθανότητα ταιριάσματος για κάθε κλάση και έτσι παράγεται ο πίνακας πιθανοτήτων για όλες τις κλάσεις.

Στην περίπτωσή μας η πρόβλεψη είναι ένας πίνακας από 4 αριθμούς. Αυτοί περιγράφουν την «εμπιστοσύνη» του μοντέλου ότι η εικόνα εισόδου αντιστοιχεί σε καθεμία από τις 4 διαφορετικές κατηγορίες (“healthy”, “bacterial”, “viral”, “lateblight”).

Για να δούμε ποια ετικέτα έχει την υψηλότερη τιμή εμπιστοσύνης χρησιμοποιείται η συνάρτηση **np.argmax(model\_out)**

Η τιμή που προκύπτει είναι και η κλάση στην οποία το μοντέλο είναι πιο σίγουρο ότι ανήκει η εικόνα εισόδου.

δ) όταν ολοκληρωθεί η διάγνωση της κατάστασης (υγιής ή ασθενική) (Εικόνα 131), θα εμφανισθούν τα ανάλογα μηνύματα για το Status και το Disease Name στην οθόνη.

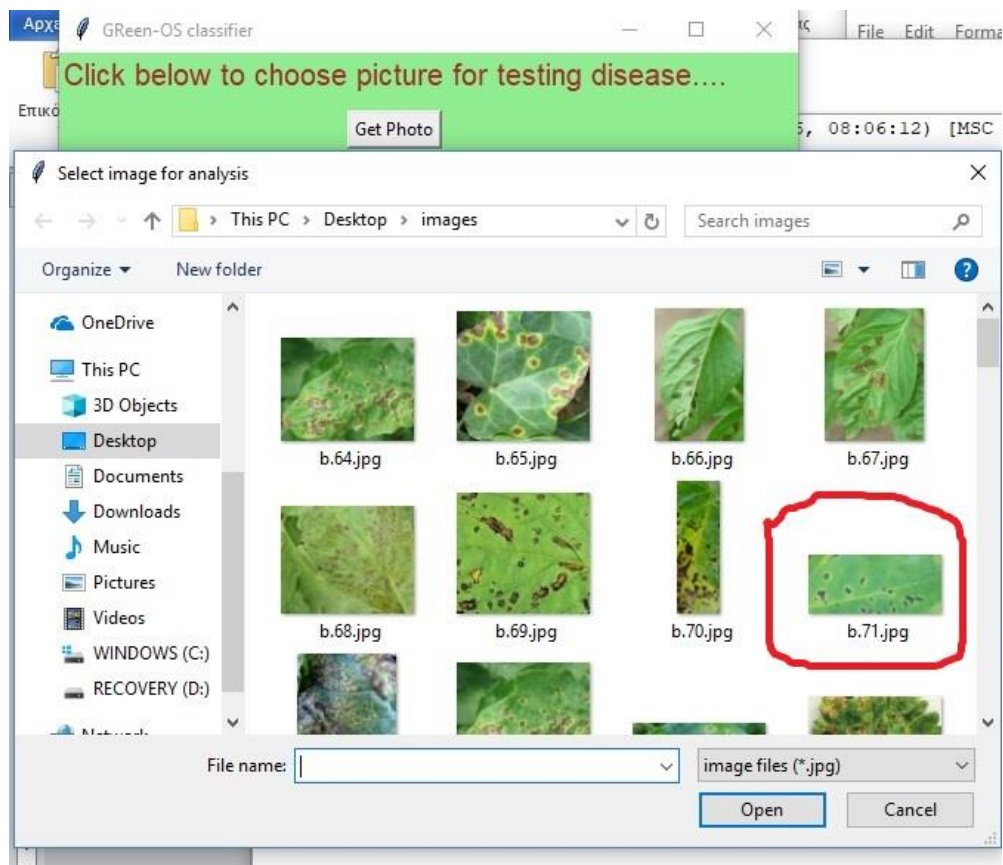
ε) τέλος το Diagnosis Support System (DiSS) παρέχει την δυνατότητα στον χρήστη να δει στην οθόνη του υπολογιστή του συμβουλές θεραπείας (διορθωτικά μέτρα) για τα ασθενικά φυτά. Οι συμβουλές αυτές προέρχονται από τη **Βάση Γνώσης** (knowledge

base), όπως καταγράφηκε με τη βοήθεια του ανθρώπου – ειδικού (γεωπόνου) και γίνονται διαθέσιμες στον χρήστη με το πάτημα του button “Remedies” (Εικόνα 132).

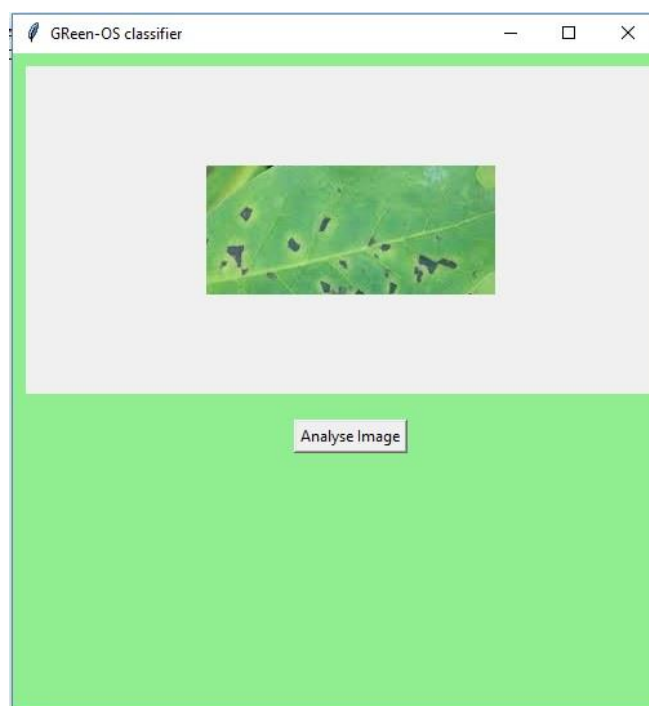


Εικόνα 128. Η απλή και φιλική διεπαφή χρήστη





Εικόνα 129. Επιλογή της φωτογραφίας από τον φάκελο images



Εικόνα 130. Το κουμπί analyze image για τη διάγνωση της ασθένειας



Εικόνα 131. Αναφορά της κατάστασης της υγείας και δυνατότητα λήψης συμβουλών



Εικόνα 132. Προβολή των προτεινόμενων διορθωτικών μέτρων



## 5 Συμπεράσματα - Δυνατότητες επέκτασης

Στη συνέχεια αναφέρονται τα τμήματα του προτεινόμενου συστήματος που τελικά υλοποιήθηκαν, καταγράφονται δυσκολίες και συμπεράσματα που προέκυψαν κατά την φάση της υλοποίησης και προτείνονται βελτιώσεις και επεκτάσεις.

### 5.1 Συμπεράσματα

Η εξέλιξη του IoT μέσα στα επόμενα χρόνια αναμένεται να δημιουργήσει νέες εφαρμογές, οι οποίες θα συνδέονται με φυσικά αντικείμενα και θα μπορούν να παρέχουν ευφυείς αποφάσεις. Οι ειδικοί εξετάζουν από τώρα τον συνδυασμό της Τεχνητής Νοημοσύνης με το IoT, κάνοντας λόγο για το «Intelligence of Things», το οποίο θα απλοποιήσει ακόμη περισσότερο την καθημερινότητά μας.

Για τις ανάγκες της εργασίας σχεδιάστηκε και υλοποιήθηκε πλατφόρμα IoT η οποία ελέγχει μια τροχήλατη ρομποτική κατασκευή μέσω μιας android εφαρμογής για φορητές συσκευές και με την υποστήριξη έμπειρου συστήματος που λειτουργεί ως fog computing παρέχουν αυτοματοποιημένες αγροτικές εργασίες, παρακολούθηση της παραγωγής, διάγνωση ασθενειών, παροχή διορθωτικών συμβουλών και άμεση ενημέρωση του υπεύθυνου του θερμοκηπίου απομακρυσμένα και σε πραγματικό χρόνο.

Το προγραμματιστικό εργαλείο appinventor αποδείχθηκε εξαιρετικό καθώς διαθέτει τα κατάλληλα συστατικά που χρειάζονται για να επικοινωνήσει, μέσω bluetooth, η android εφαρμογή “GReen-OS” με την ρομποτική κατασκευή “farmbot-Si”. Γράφοντας τον κατάλληλο κώδικα οι μετρήσεις θερμοκρασίας-υγρασίας που στέλνονται από την arduino based πλακέτα ESP8266 σε Web Server εντός του τοπικού δικτύου ενσωματώνονται στην εφαρμογή “GReen-OS” και είναι άμεσα διαθέσιμες. Οι φωτογραφίες των φυτών αποστέλλονται μέσω πρωτοκόλλου FTP εύκολα στο diagnosis support system με την βοήθεια εξωτερικών εφαρμογών android. Η εκπαίδευση και η χρήση του έμπειρου συστήματος διάγνωσης ασθενειών επιτυγχάνεται εύκολα μέσα από μια φιλική διεπαφή χρήστη.

Κατά τη διάρκεια της κατασκευής του συστήματος έγινε αντιληπτή η μεγάλη σημασία που έχουν οι φωτογραφίες του train dataset για την καλύτερη εκπαίδευση του συνελκτικού νευρωνικού δικτύου που αποτελεί το diagnosis support system. Όσο πε-

ρισσότερες και αντιπροσωπευτικές είναι οι φωτογραφίες, τόσο αυξάνεται η ακρίβεια και η αξιοπιστία του διαγνωστικού συστήματος.

Το πιο σημαντικό συστατικό του Έμπειρου Συστήματος είναι η Βάση Γνώσης. Για το κτίσιμο της Βάσης Γνώσης συμπεριλήφθησαν πληροφορίες που αντλήθηκαν από τη βιβλιογραφία σχετικά με τις ασθένειες αλλά και πληροφορίες που προέκυψαν ύστερα από επικοινωνία και συνεντεύξεις με ειδικούς του κλάδου της φυτοπαθολογίας. Αυτό την καθιστά ένα πολύτιμο αντικείμενο που μπορεί να μεταφερθεί και να χρησιμοποιηθεί σε άλλα Έμπειρα Συστήματα.

Τα συνελκτικά νευρωνικά δίκτυα (CNN) έχουν επιδείξει εξαιρετικές επιδόσεις (ξεπερνώντας αυτές των ανθρώπων) σε προβλήματα αναγνώρισης αντικειμένων και ταξινόμησης εικόνων. Σε αυτή την εργασία χρησιμοποιείται συνελκτικό νευρωνικό δίκτυο για την ανίχνευση και ταξινόμηση των ασθενειών των φύλλων των φυτών. Το νευρωνικό δίκτυο εκπαιδεύεται χρησιμοποιώντας εικόνες που έχουν ληφθεί τόσο υπό φυσικό όσο και υπό προστατευμένο περιβάλλον και έχει επιτευχθεί ακρίβεια ταξινόμησης σε ποσοστό 98,65%. Αυτό δείχνει την ικανότητα του CNN να εξαγάγει σημαντικά χαρακτηριστικά σε φυσικό περιβάλλον τα οποία απαιτούνται για την ταξινόμηση των ασθενειών των φυτών. Τα πειράματα δείχνουν επίσης ότι η εφαρμογή της αύξησης των δεδομένων στο dataset εκπαίδευσης βελτιώνει την απόδοση του δικτύου όταν το dataset είναι πολύ μικρό.

Η δημιουργία αναφοράς, η διατήρηση ιστορικού και η ενημέρωση του υπεύθυνου του θερμοκηπίου επιτυγχάνονται μέσα από την εφαρμογή “GReen-OS” εύκολα με χρήση real time βάσης δεδομένων.

## 5.2 Δυνατότητες επέκτασης

Εδώ επιλέγεται ο αισθητήρας DHT11 / DHT22 επειδή διαβάζει δύο τιμές και χρησιμοποιεί ψηφιακή λογική. Εύκολα θα μπορούσε να επεκταθεί η λειτουργικότητα της IoT πλατφόρμας με χρήση αισθητήρα μέτρησης οπτικής ακτινοβολίας ο οποίος θα βοηθήσει στην ενσωμάτωση φωτοβολταϊκών πάνελς για την εκμετάλλευση ανανεώσιμων πηγών ενέργειας. Με αυτή την τεχνολογία θα φορτίζει η μπαταρία που τροφοδοτεί το ρομπότ κατά την διάρκεια της κίνησής του και θα υπάρχει και δυνατότητα αποθήκευσης ηλεκτρικής ενέργειας.

Επέκταση θα μπορούσε να γίνει και στις αγροτικές εργασίες που παρέχει η ρομποτική κατασκευή προσθέτοντας δυνατότητες λήψης δείγματος από το υπέδαφος για ανάλυση.

Για να προχωρήσουμε περαιτέρω, θα μπορούσαν να παρέρχονται components για frontend και analytics, on-device επεξεργασία δεδομένων και cloud-based ανάπτυξη της IoT πλατφόρμας.

Θα μπορούσε κάποιος, σε μια δεύτερη φάση βελτίωσης του έμπειρου συστήματος, να το εμπλουτίσει με επιπλέον φωτογραφίες ή να αυξήσει τις επεξηγηματικές δυνατότητες του συστήματος συμπληρώνοντας τα αρχεία κειμένου (txt files) και προσθέτοντας πολυγλωσσική υποστήριξη. Τέλος, σημαντική βελτίωση του συστήματος θα μπορούσαμε να έχουμε κάνοντας την αξιολόγηση του συστήματος, συγκρίνοντας δηλαδή τα αποτελέσματα που δίνει, με εκείνα που θα προκύψουν από τη μικροβιολογική εξέταση για κάθε περίπτωση.

Υπάρχει η δυνατότητα να αυξηθεί το επίπεδο αυτοματισμού της IoT πλατφόρμας και των επιμέρους συστατικών της με μικρές επεμβάσεις στον κώδικα, κάτι που θα προσφέρει περισσότερη αυτονομία στο σύστημα. Ωστόσο δεν πρέπει να παραμελείται η σημασία της ανθρώπινης παρέμβασης η οποία κρίνεται αναπόσπαστο στοιχείο της φιλοσοφίας ενός “cyber-physical” system.

Σε ένα ευρύτερο πλαίσιο επέκτασης θα μπορούσε να σχεδιασθεί εκ νέου η IoT πλατφόρμα υιοθετώντας περισσότερο “ανοιχτές λύσεις” και να δημιουργηθεί μια πιο “compact” ρομποτική κατασκευή η οποία θα ελέγχεται από πλακέτα βασισμένη σε raspberry pi πάνω στην οποία θα τρέχει και το Σύστημα Υποστήριξης Διάγνωσης (DiSS). Το προεπιλεγμένο υλικολογισμικό (firmware) του raspberry pi είναι closed source, αλλά είναι διαθέσιμη μια ανεπίσημη έκδοση open source. Η υλοποίηση με raspberry pi παρέχει το πλεονέκτημα ότι μπορεί να χρησιμοποιηθεί οποιοδήποτε λειτουργικό σύστημα (Raspbian, Ubuntu, Windows 10 IoT Core, RISC OS) και αρκεί μια μόνο γλώσσα προγραμματισμού, η **python**, με την βοήθεια της οποίας μπορεί να υλοποιηθεί εξ ολοκλήρου η IoT πλατφόρμα, να γίνεται ο έλεγχος και η κίνηση των μοτέρ της ρομποτικής κατασκευής, να ελέγχεται η κάμερα για την λήψη των φωτογραφιών των φυτών και τέλος να “τρέχει” και ο κώδικας που υλοποιεί την λειτουργία του Συνελικτικού Νευρωνικού Δικτύου (CNN) για την διάγνωση των ασθενειών των φυτών.



# Πηγές

## Ξενόγλωσσες Πηγές

- [1] Feigenbaum E.A., (1978) DENDRAL and Meta-DENDRAL: Their Applications Dimension, Artificial Intelligence, 11, pp. 5-24.
- [2] Fonseca Daniel J., Navarrese Daniel., (2002) Artificial Neural Network for job shop simulation.
- [3] Haugeland J., (1989) Artificial Intelligence: The Very Idea, MIT Press.
- [4] Luger G.F., Stubblefield J., (1993) Artificial Intelligence: Structures and Strategies for Complex Problem-Solving, 4th edition, Addison-Wesley Longman.
- [5] Rich E., Knight K., (1991) Artificial Intelligence, McGraw-Hill.
- [6] Rich E.A., (1983) Potato Diseases. Academic Press, pp.238
- [7] Winston P.H., (1992) Artificial Intelligence, 3rd edition, Addison-Wesley.

## Ελληνικές Πηγές

- [8] Αποστόλου Α., (2004) Εφαρμογές Τεχνητών Νευρωνικών Δικτύων Στην Επίλυση Προβλημάτων Χρονοπρογραμματισμού Συστημάτων Παραγωγής, Ε.Μ.Π.
- [9] Αργιάλας Δ., (2001) «Introduction to Knowledge-Based Expert System», Πανεπιστημιακές Παραδόσεις στο μάθημα των Υπολογιστικών Μεθόδων στη Γεωπληροφορική, Ε.Μ.Π.
- [10] Βλαχάβας Ι., Κεφάλας Π., Βασιλειάδης Ν., (2002) Τεχνητή Νοημοσύνη, Εκδόσεις Γαργατάνη

- [11] Γιαλούρης Κ., (2011) Εισαγωγή στην Τεχνητή Νοημοσύνη και στα Έμπειρα Συστήματα, Πανεπιστημιακές Σημειώσεις στο μάθημα Έμπειρα Συστήματα, Γ.Π.Α.
- [12] Κρικέτος Β., Παστράς Κ., (1991) Εγχειρίδιο Εισαγωγής στα Έμπειρα Συστήματα, Εκδόσεις Εταιρία Ανάπτυξης της Ναυτικής Τεχνολογίας Α.Ε.
- [13] Χαντζάρα Α., (2008) Ανάπτυξη Έμπειρου Συστήματος λήψης αποφάσεων Ναυτικής επιχειρηματικότητας.

## Ηλεκτρονικές Πηγές

- [14] Abu-Naser S. S., Kashkash K. A., Fayyad M., (2008) Developing an Expert System for Plant Disease Diagnosis, διαθέσιμο στο διαδικτυακό τόπο:  
<https://scialert.net/fulltextmobile/?doi=jai.2008.78.85>
- [15] Chaitra K. M., Faiza A., Harshitha I. P., Meghana D. M., Rachitha M. V., (2018) Plant Leaf Disease Identification System for Android, διαθέσιμο στο διαδικτυακό τόπο:  
[https://www.technoarete.org/common\\_abstract/pdf/IJERCSE/v5/i6/Ext\\_49037.pdf](https://www.technoarete.org/common_abstract/pdf/IJERCSE/v5/i6/Ext_49037.pdf)
- [16] Chopade P., Bhagyashri K., (2016) Image processing Based Detection and classification of leaf disease on fruits crops, διαθέσιμο στο διαδικτυακό τόπο:  
<https://pdfs.semanticscholar.org/7544/3f21031b2da53263ab177d51aa2ec0428fee.pdf>
- [17] Dangtongdee K., Kurfess F., (2018) Plant Identification Using Tensorflow, διαθέσιμο στο διαδικτυακό τόπο:  
<https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1271&context=cpesp>
- [18] Ferentinos K., (2018) Deep learning models for plant disease detection and diagnosis, διαθέσιμο στο διαδικτυακό τόπο:  
[https://www.researchgate.net/publication/322941653\\_Deep\\_learning\\_models\\_for\\_plant\\_disease\\_detection\\_and\\_diagnosis](https://www.researchgate.net/publication/322941653_Deep_learning_models_for_plant_disease_detection_and_diagnosis)
- [19] Indhu Mathi k., Kavya V. B., Meera P., Sankar Ganesh K., (2018) Lesion Characterisation For Plant Disease Diagnosis, διαθέσιμο στο διαδικτυακό τόπο:  
<http://www.irjaet.com/Volume4-Issue-2/paper27.pdf>
- [20] Jayaprakash S., Veni S., (2016) OpenCV Based Disease Identification of Mango Leaves, διαθέσιμο στο διαδικτυακό τόπο:  
<http://www.enggjournals.com/ijet/docs/IJET16-08-05-417.pdf>

- [21] Jeon W., Rhee S., (2017) Plant Leaf Recognition Using a Convolution Neural Network, διαθέσιμο στο διαδικτυακό τόπο: [https://www.researchgate.net/publication/315911462\\_Plant\\_Leaf\\_Recognition\\_Using\\_a\\_Convolution\\_Neural\\_Network](https://www.researchgate.net/publication/315911462_Plant_Leaf_Recognition_Using_a_Convolution_Neural_Network)
- [22] Kamilaris A., Boldu F., (2018) A review of the use of convolutional neural networks in agriculture, διαθέσιμο στο διαδικτυακό τόπο: [https://www.researchgate.net/publication/325974939\\_A\\_review\\_of\\_the\\_use\\_of\\_convolutional\\_neural\\_networks\\_in\\_agriculture](https://www.researchgate.net/publication/325974939_A_review_of_the_use_of_convolutional_neural_networks_in_agriculture)
- [23] Khan U., Oberoi A., (2019) Plant Disease Detection Techniques: A Review, διαθέσιμο στο διαδικτυακό τόπο: [https://www.academia.edu/38815011/Plant\\_Disease\\_Detection\\_Techniques\\_A\\_Review](https://www.academia.edu/38815011/Plant_Disease_Detection_Techniques_A_Review)
- [24] Maheswari T., Madhuri P., (2018) Design and Implementation of Plant Disease Detection Algorithm, διαθέσιμο στο διαδικτυακό τόπο: <http://www.ijetjournal.org/Special-Issues/NCETIMES/NCETIMES11.pdf>
- [25] Petrellis N., (2015) Plant Disease Diagnosis Based on Image Processing, Appropriate for Mobile Phone Implementation, διαθέσιμο στο διαδικτυακό τόπο: [http://ceur-ws.org/Vol-1498/HAICTA\\_2015\\_paper28.pdf](http://ceur-ws.org/Vol-1498/HAICTA_2015_paper28.pdf)
- [26] Petrellis N., (2017) A Smartphone Image Processing Application for Plant Disease Diagnosis, διαθέσιμο στο διαδικτυακό τόπο: [https://www.researchgate.net/publication/317298608\\_A\\_smart\\_phone\\_image\\_processing\\_application\\_for\\_plant\\_disease\\_diagnosis](https://www.researchgate.net/publication/317298608_A_smart_phone_image_processing_application_for_plant_disease_diagnosis)
- [27] Sahil V. K., Mohit A. R., Bhargav R. P., Tatwadarshi P. N., (2018) A Survey of Image Processing and Identification Techniques, διαθέσιμο στο διαδικτυακό τόπο: [https://www.researchgate.net/publication/328190963\\_A\\_Survey\\_of\\_Image\\_Processing\\_and\\_Identification\\_Techniques](https://www.researchgate.net/publication/328190963_A_Survey_of_Image_Processing_and_Identification_Techniques)
- [28] Subhashini S. J., Nagalakshmi K., Vijayadurga A., Poongodi P., Yokes K., (2018) A Survey on Leaf Disease Detection by Using K-means Algorithm, διαθέσιμο στο διαδικτυακό τόπο: <https://www.ijltemas.in/DigitalLibrary/Vol.7Issue4/101-104.pdf>
- [29] Thakre G., More A., Gajakosh K., Yewale M., Shamkuwar D., (2017) A Study on Real Time Plant Disease Diagnosis System, διαθέσιμο στο διαδικτυακό τόπο: <https://pdfs.semanticscholar.org/31e5/53c300f34c2f09e6fbb2227e9998abdf3476.pdf>
- [30] Δημόπουλος Φ., (2012) Ανάπτυξη Εμπείρου Συστήματος για τις ασθένειες και τους εχθρούς της πατάτας, διαθέσιμο στο διαδικτυακό τόπο: [http://dspace.aua.gr/xmlui/bitstream/handle/10329/5694/Dimopoulos\\_F.pdf?sequence=3](http://dspace.aua.gr/xmlui/bitstream/handle/10329/5694/Dimopoulos_F.pdf?sequence=3)
- [31] Διαμαντίδης Δ., (2018) Ακόμη πιο «πράσινη» η γεωργία με τη ρομποτική,

διαθέσιμο στο διαδικτυακό τόπο:

<http://greenagenda.gr/27%CE%B7-agrotica-%CE%B1%CE%BA%CF%8C%CE%BC%CE%B7-%CF%80%CE%B9%CE%BF-%CF%80%CF%81%CE%AC%CF%83%CE%B9%CE%BD%CE%B7-%CE%B7-%CE%B3%CE%B5%CF%89%CF%81%CE%B3%CE%AF%CE%B1-%CE%BC%CE%B5-%CF%84%CE%B7/>

[32] Δίκτυο smart-AKIS, (2016), Τι είναι η έξυπνη γεωργία,

διαθέσιμο στο διαδικτυακό τόπο:

<https://www.smart-akis.com/index.php/el/network-el/what-is-smart-farming-el/>

[33] Κουρούδης Σ., (2017) Γεωργία ακριβείας: Το μέλλον του αγροτικού τομέα,

διαθέσιμο στο διαδικτυακό τόπο: <http://thrakika.gr/?p=14357>

[34] Ομάδα γεωπόνων της Farmacon, (2017) Η τεχνητή νοημοσύνη στη γεωργία... Ψηφιακές καλλιεργητικές δεξιότητες, διαθέσιμο στο διαδικτυακό τόπο:

<https://blog.farmacon.gr/katigories/tehniki-arthrografia/georgia-akriveias/item/1715-i-texniti-noimosyni-sti-georgia-psifiakes-kalliergitikes-deksiotites>

[35] Ομάδα γεωπόνων της Farmacon, (2018) Έξυπνη γεωργία σημαίνει αυτοματοποιημένη και συνδεδεμένη γεωργία, διαθέσιμο στο διαδικτυακό τόπο:

<https://blog.farmacon.gr/katigories/tehniki-arthrografia/georgia-akriveias/item/2008-eksypni-georgia-simainei-aftomatopoiimeni-kai-syndedemeni-georgia>

[36] Ομάδα γεωπόνων της Farmacon, (2018) 9 τελευταίες τεχνολογικές εξελίξεις στον αγροτικό τομέα, διαθέσιμο στο διαδικτυακό τόπο:

<https://blog.farmacon.gr/katigories/tehniki-arthrografia/georgia-akriveias/item/1981-9-teleftaies-technologikes-ekselikseis-ston-agrotiko-tomea>

[37] Ομάδα γεωπόνων της Farmacon, (2018) Ανάλυση δεδομένων χωραφίου και εφαρμογή λιπασμάτων, διαθέσιμο στο διαδικτυακό τόπο:

<https://blog.farmacon.gr/katigories/tehniki-arthrografia/threpsi-lipansi/item/2116-analysis-dedomenon-xorafioy-kai-efarmogi-lipasmaton>



# Παράρτημα Α

Δομή Πτυχιακής και έκταση κεφαλαίων.

Κεφάλαια	Περιεχόμενο	(σελίδες)
1 Εισαγωγή	Αναφορά στον σκοπό και στους επιμέρους στόχους της πτυχιακής. Στο τέλος του, παρατίθεται σύντομα το τι περιέχει κάθε κεφάλαιο της πτυχιακής.	1 – 2
2	Παρουσίαση/Περιγραφή του ευρύτερου γνωστικού πεδίου στο οποίο εντάσσεται η πτυχιακή καθώς και του ειδικότερου θέματος που πραγματεύεται η πτυχιακή. Αναφορά σε κοντινές ερευνητικές προσπάθειες.	3 – 68
3	α) Περιγραφή του μοντέλου του προτεινόμενου συστήματος β) Γενικότερη περιγραφή (αρχιτεκτονική του συστήματος, διαγράμματα ροής) γ) Αναφορά στο τι επιτυγχάνεται με το σύστημα.	69 - 84
4	α) Εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση. β) Σημαντικές ρυθμίσεις. γ) Περιγραφή της υλοποίησης - εικόνες (screenshots), σύντομα τμήματα σημαντικού κώδικα.	85 – 164
5 Συμπεράσματα	Επίλογος - Συμπεράσματα - Μελλοντική εργασία Κριτική αναφορά στα πεπραγμένα της εργασίας, Προβλήματα, Δυσκολίες που αντιμετωπίστηκαν, Επεκτάσεις, Θέματα που δεν λύθηκαν και τίθενται ως μελλοντικός στόχος άλλων πτυχιακών.	165 – 168
Βιβλιογραφία	Βιβλία - εργασίες - Web links που χρησιμοποιήθηκαν και κρίνονται απαραίτητα.	169 – 172
Παραρτήματα	A) Δομή Πτυχιακής B) Κώδικας	173 - 190
<b>ΣΥΝΟΛΟ από εξώφυλλο σε εξώφυλλο</b>		<b>1 - 191</b>



# Παράρτημα Β

Παρακάτω δίνεται ο κώδικας του αρχείου `cnn.py` (Το συνελκτικό νευρωνικό δίκτυο που κάνει την εξαγωγή κρίσιμων χαρακτηριστικών από τις εικόνες και τις ταξινομεί σε 4 διαφορετικές κατηγορίες).

```
# author: Papastergiou George
```

```
# Python program to create
```

```
# Image Classifier using CNN
```

```
# Importing the required libraries
```

```
import cv2          # working with, mainly resizing, images
```

```
import numpy as np   # dealing with arrays
```

```
import os            # dealing with directories
```

```
from random import shuffle # mixing up or currently ordered data that might lead our  
network as tray in training.
```

```
from tqdm import tqdm     # a nice pretty percentage bar for tasks.
```

```
import matplotlib.pyplot as plt
```

```
#Creating the neural network using tensorflow
```

```
# Importing the required libraries
```

```
import tflearn
```

```
from tflearn.layers.conv import conv_2d, max_pool_2d
```

```
from tflearn.layers.core import input_data, dropout, fully_connected
```

```
from tflearn.layers.estimator import regression
```

```
import tensorflow as tf
```

```

# Setting up the env
TRAIN_DIR = 'train'
TEST_DIR = 'test'
IMG_SIZE = 50
LR = 1e-3

# Setting up the model which will help with tensorflow models
MODEL_NAME = 'healthyvsunhealthy-{}-{}.model'.format(LR, '2conv-basic')

# Image preprocessing
# Labelling the dataset
def label_img(img):
    word_label = img[0]
    #word_label= img.split('.')[0]

# DIY One hot encoder
if word_label == 'h': return [1,0,0,0]

elif word_label == 'b': return [0,1,0,0]
elif word_label == 'v': return [0,0,1,0]
elif word_label == 'l': return [0,0,0,1]

# Creating the training data
# the actual reading of training and test data.
# Every image will be resized to 50 x 50 pixels.
def create_train_data():

```

```
# Creating an empty list where we should the store the training data
```

```
# after a little preprocessing of the data
```

```
training_data = []
```

```
# tqdm is only used for interactive loading
```

```
# loading the training data
```

```
for img in tqdm(os.listdir(TRAIN_DIR)):
```

```
    # labeling the images
```

```
    label = label_img(img)
```

```
    path = os.path.join(TRAIN_DIR,img)
```

```
    img = cv2.imread(path,cv2.IMREAD_COLOR)
```

```
    # resizing the images for processing them in the covnet
```

```
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
```

```
    # final step-forming the training data list with numpy array of the images
```

```
    training_data.append([np.array(img),np.array(label)])
```

```
# shuffling of the training data to preserve the random state of our data
```

```
shuffle(training_data)
```

```
# saving our trained data for further uses if required
```

```
np.save('train_data.npy', training_data)
```

```
return training_data
```

```
# Processing the given test data
```

```
# Almost same as processing the training data but
```

```

# we dont have to label it.

def process_test_data():

    testing_data = []

    for img in tqdm(os.listdir(TEST_DIR)):

        path = os.path.join(TEST_DIR,img)

        img_num = img.split('.')[0]

        img = cv2.imread(path,cv2.IMREAD_COLOR)

        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))

        testing_data.append([np.array(img), img_num])


    shuffle(testing_data)

    np.save('test_data.npy', testing_data)

    return testing_data


# If dataset is not created:

train_data = create_train_data()

# test_data = create_test_data()


# If we have already created the dataset:

# train_data = np.load('train_data.npy')

# test_data = np.load('test_data.npy')


# train_data = np.load('train_data.npy')

# test_data = np.load('test_data.npy')

# Building A ConvNet using tensorflow

tf.reset_default_graph()

```

```
tf.logging.set_verbosity(tf.logging.ERROR)
```

```
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')
```

```
#Conv Layer 1
```

```
convnet = conv_2d(convnet, 32, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 2
```

```
convnet = conv_2d(convnet, 64, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 3
```

```
convnet = conv_2d(convnet, 128, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 4
```

```
convnet = conv_2d(convnet, 32, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 5
```

```
convnet = conv_2d(convnet, 64, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 6
```

```
convnet = fully_connected(convnet, 1024, activation='relu')
```

```
convnet = dropout(convnet, 0.8)
```

```
#Fully Connected Layer with SoftMax as Activation Function
```

```
convnet = fully_connected(convnet, 4, activation='softmax')  
#Regression for ConvNet with ADAM optimizer  
convnet = regression(convnet, optimizer='adam', learning_rate=LR,  
loss='categorical_crossentropy', name='targets')
```

```
model = tflearn.DNN(convnet, tensorboard_dir='log')
```

```
if os.path.exists('{ }.meta'.format(MODEL_NAME)):  
    model.load(MODEL_NAME)  
    print('model loaded!')
```

```
# Splitting Our Test and Train Data
```

```
train = train_data[:-46]  
test = train_data[-46:]
```

```
#This is our Training data
```

```
X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,3)  
Y = [i[1] for i in train]
```

```
#This is our Training data
```

```
test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,3)  
test_y = [i[1] for i in test]
```

```
# Training Our Model To Recognize Diseases
```

```
model.fit({'input': X}, {'targets': Y}, n_epoch=3, validation_set=({'input': test_x}, {'tar-  
gets': test_y}),  
snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
```



```
# Saving Our Model(Important Step)
```

```
model.save(MODEL_NAME)
```

Παρακάτω δίνεται ο κώδικας του αρχείου ui.py (Δημιουργεί μια διεπαφή χρήστη ώστε με εύχρηστο τρόπο να μπορεί μια καινούρια φωτογραφία να συγκριθεί με αυτές που έχει το προεκπαιδευμένο μοντέλο, να διαγνωσθεί αν είναι υγιής ή αν πάσχει από κάποια ασθένεια και στην περίπτωση αυτή να προταθούν τα ανάλογα διορθωτικά μέτρα).

```
# author: Papastergiou George
```

```
# user interface to support plant leaf disease detection
```

```
# Setting up
```

```
import tkinter as tk
```

```
from tkinter.filedialog import askopenfilename
```

```
import shutil
```

```
import os
```

```
import sys
```

```
from PIL import Image, ImageTk
```

```
window = tk.Tk()
```

```
window.title("GRreen-OS classifier")
```

```
window.geometry("500x510")
```

```
window.configure(background="lightgreen")
```

```

title = tk.Label(text="Click below to choose picture for testing disease....", background
= "lightgreen", fg="Brown", font=("", 15))
title.grid()
def bact():
    window.destroy()
    window1 = tk.Tk()

    window1.title("GReen-OS classifier")

    window1.geometry("500x510")
    window1.configure(background="lightgreen")

    def exit():
        window1.destroy()
    rem = "The remedies for Bacterial Spot are:\n\n "
    remedies = tk.Label(text=rem, background="lightgreen",
        fg="Brown", font=("", 15))
    remedies.grid(column=0, row=7, padx=10, pady=10)
    rem1 = " Discard or destroy any affected plants. \n Do not compost them. \n Rotate
your tomato plants yearly to prevent re-infection next year. \n Use copper fungicides"
    remedies1 = tk.Label(text=rem1, background="lightgreen",
        fg="Black", font=("", 12))
    remedies1.grid(column=0, row=8, padx=10, pady=10)

    button = tk.Button(text="Exit", command=exit)
    button.grid(column=0, row=9, padx=20, pady=20)

    window1.mainloop()

```

```

def vir():

    window.destroy()

    window1 = tk.Tk()


    window1.title("GReen-OS classifier")


    window1.geometry("650x510")

    window1.configure(background="lightgreen")


    def exit():

        window1.destroy()

        rem = "The remedies for Yellow leaf curl virus are: "

        remedies = tk.Label(text=rem, background="lightgreen",

                             fg="Brown", font=("", 15))

        remedies.grid(column=0, row=7, padx=10, pady=10)

        rem1 = " Monitor the field, handpick diseased plants and bury them. \n Use sticky

yellow plastic traps. \n Spray insecticides such as organophosphates, carbamates during

the seedling stage. \n Use copper fungicides"

        remedies1 = tk.Label(text=rem1, background="lightgreen",

                              fg="Black", font=("", 12))

        remedies1.grid(column=0, row=8, padx=10, pady=10)


        button = tk.Button(text="Exit", command=exit)

        button.grid(column=0, row=9, padx=20, pady=20)


    window1.mainloop()


def latebl():

    window.destroy()

    window1 = tk.Tk()

```

```
window1.title("GReen-OS classifier")
```

```
window1.geometry("520x510")
```

```
window1.configure(background="lightgreen")
```

```
def exit():
```

```
    window1.destroy()
```

```
    rem = "The remedies for Late Blight are: "
```

```
    remedies = tk.Label(text=rem, background="lightgreen",
```

```
                        fg="Brown", font=("", 15))
```

```
    remedies.grid(column=0, row=7, padx=10, pady=10)
```

```
    rem1 = " Monitor the field, remove and destroy infected leaves. \n Treat organically  
with copper spray. \n Use chemical fungicides,the best of which for tomatoes is chloro-  
thalonil."
```

```
    remedies1 = tk.Label(text=rem1, background="lightgreen",
```

```
                        fg="Black", font=("", 12))
```

```
    remedies1.grid(column=0, row=8, padx=10, pady=10)
```

```
    button = tk.Button(text="Exit", command=exit)
```

```
    button.grid(column=0, row=9, padx=20, pady=20)
```

```
    window1.mainloop()
```

```
def analysis():
```

```
    import cv2 # working with, mainly resizing, images
```

```
    import numpy as np # dealing with arrays
```

```
    import os # dealing with directories
```

```
from random import shuffle # mixing up or currently ordered data that might lead
our network as tray in training.
```

```
from tqdm import \
```

```
tqdm # a nice pretty percentage bar for tasks.
```

```
verify_dir = 'C:/Users/byron pap/Desktop/plant_project_test_pictures' # it must be
changed to match appropriate dir
```

```
IMG_SIZE = 50
```

```
LR = 1e-3
```

```
MODEL_NAME = 'healthyvsunhealthy-{}-{}.model'.format(LR, '2conv-basic')
```

```
def process_verify_data():
```

```
    verifying_data = []
```

```
    for img in tqdm(os.listdir(verify_dir)):
```

```
        path = os.path.join(verify_dir, img)
```

```
        img_num = img.split('.')[0]
```

```
        img = cv2.imread(path, cv2.IMREAD_COLOR)
```

```
        img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
```

```
        verifying_data.append([np.array(img), img_num])
```

```
    np.save('verify_data.npy', verifying_data)
```

```
    return verifying_data
```

```
verify_data = process_verify_data()
```

```
#verify_data = np.load('verify_data.npy')
```

```
import tflearn
```

```
from tflearn.layers.conv import conv_2d, max_pool_2d
```

```
from tflearn.layers.core import input_data, dropout, fully_connected
```

```
from tflearn.layers.estimator import regression
```

```
import tensorflow as tf
```

```
tf.reset_default_graph()
```

```
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')
```

```
#Conv Layer 1
```

```
convnet = conv_2d(convnet, 32, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 2
```

```
convnet = conv_2d(convnet, 64, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 3
```

```
convnet = conv_2d(convnet, 128, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 4
```

```
convnet = conv_2d(convnet, 32, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 5
```

```
convnet = conv_2d(convnet, 64, 3, activation='relu')
```

```
convnet = max_pool_2d(convnet, 3)
```

```
#Conv Layer 6
```

```
convnet = fully_connected(convnet, 1024, activation='relu')
```

```
convnet = dropout(convnet, 0.8)
```

```
#Fully Connected Layer with SoftMax as Activation Function
```

```
convnet = fully_connected(convnet, 4, activation='softmax')
```

```

#Regression for ConvNet with ADAM optimizer

convnet = regression(convnet, optimizer='adam', learning_rate=LR,
loss='categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')

if os.path.exists('{} .meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

import matplotlib.pyplot as plt

fig = plt.figure()

for num, data in enumerate(verify_data):

    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3, 4, num + 1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE, IMG_SIZE, 3)
    # model_out = model.predict([data])[0]
    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 0:
        str_label = 'healthy'
    elif np.argmax(model_out) == 1:
        str_label = 'bacterial'
    elif np.argmax(model_out) == 2:

```

```

        str_label = 'viral'

    elif np.argmax(model_out) == 3:

        str_label = 'lateblight'


    if str_label == 'healthy':

        status = "HEALTHY"

    else:

        status = "UNHEALTHY"


    message = tk.Label(text='Status: '+status, background="lightgreen",
                        fg="Brown", font=("", 15))

    message.grid(column=0, row=3, padx=10, pady=10)

    if str_label == 'bacterial':

        diseasename = "Bacterial Spot "

        disease = tk.Label(text='Disease Name: ' + diseasename, back-
ground="lightgreen",
                        fg="Black", font=("", 15))

        disease.grid(column=0, row=4, padx=10, pady=10)

        r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))

        r.grid(column=0, row=5, padx=10, pady=10)

        button3 = tk.Button(text="Remedies", command=bact)

        button3.grid(column=0, row=6, padx=10, pady=10)

    elif str_label == 'viral':

        diseasename = "Yellow leaf curl virus "

        disease = tk.Label(text='Disease Name: ' + diseasename, back-
ground="lightgreen",
                        fg="Black", font=("", 15))

        disease.grid(column=0, row=4, padx=10, pady=10)

```



```

        r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))

        r.grid(column=0, row=5, padx=10, pady=10)

        button3 = tk.Button(text="Remedies", command=vir)

        button3.grid(column=0, row=6, padx=10, pady=10)

    elif str_label == 'lateblight':

        diseasename = "Late Blight "

        disease = tk.Label(text='Disease Name: ' + diseasename, back-
ground="lightgreen",

                                fg="Black", font=("", 15))

        disease.grid(column=0, row=4, padx=10, pady=10)

        r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))

        r.grid(column=0, row=5, padx=10, pady=10)

        button3 = tk.Button(text="Remedies", command=latebl)

        button3.grid(column=0, row=6, padx=10, pady=10)

    else:

        r = tk.Label(text='Plant is healthy', background="lightgreen", fg="Black",

                                font=("", 15))

        r.grid(column=0, row=4, padx=10, pady=10)

        button = tk.Button(text="Exit", command=exit)

        button.grid(column=0, row=9, padx=20, pady=20)

def openphoto():

    dirPath = "C:/Users/byron pap/Desktop/plant_project_test_pictures" # it must be
changed to match appropriate dir

    fileList = os.listdir(dirPath)

    for fileName in fileList:

        os.remove(dirPath + "/" + fileName)

```

```

# C:/Users/byron pap/Desktop/images is the location of the image that you want to
test..... you can change it according to the image location you have

fileName = askopenfilename(initialdir='C:/Users/byron pap/Desktop/images', ti-
tle='Select image for analysis ',

                             filetypes=[('image files', '.jpg')])

dst = "C:/Users/byron pap/Desktop/plant_project_test_pictures" # it must be changed
to match appropriate dir

shutil.copy(fileName, dst)

load = Image.open(fileName)

render = ImageTk.PhotoImage(load)

img = tk.Label(image=render, height="250", width="500")

img.image = render

img.place(x=0, y=0)

img.grid(column=0, row=1, padx=10, pady = 10)

title.destroy()

button1.destroy()

button2 = tk.Button(text="Analyse Image", command=analysis)

button2.grid(column=0, row=2, padx=10, pady = 10)

button1 = tk.Button(text="Get Photo", command = openphoto)

button1.grid(column=0, row=1, padx=10, pady = 10)


window.mainloop()

```