

# Raport Final - CropGuardians: Detectia legumelor bolnave

Culidiuc Maria

Enache Vlad

Prața Alexandra

Zdrânc Rares

13 mai 2025

## Cuprins

<b>1</b>	<b>Introducere</b>	<b>3</b>
1.1	Încadrarea în context a temei proiectului	3
1.2	Evidențierea avantajului de a folosi AI în soluționarea problemei proiectului	3
1.3	Abstract grafic	4
<b>2</b>	<b>Metode existente de rezolvare a problemei (related work)</b>	<b>5</b>
2.1	Disease Detection in Vegetables Using Image Processing Techniques: A Review	5
2.1.1	Date utilizate	5
2.1.2	Algoritmi utilizați	5
2.1.3	Rezultate obținute	6
2.2	Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment	6
2.2.1	Date utilizate	6
2.2.2	Algoritmi utilizați	6
2.2.3	Rezultate obținute	7
<b>3</b>	<b>Metode efectiv folosite pentru rezolvarea problemei</b>	<b>7</b>
3.1	Descrierea algoritmilor inteligenți folosiți	7
3.1.1	Model CNN Inițial	7
3.1.2	Model CNN Îmbunătățit	8
3.2	Procesul de învățare/optimizare implicat	9
3.2.1	Preprocesarea datelor	9
3.2.2	Strategii de optimizare	10

3.2.3	Hiperparametri optimizați . . . . .	10
3.3	Metrici folosite pentru evaluarea rezultatelor experimentelor . . . . .	10
<b>4</b>	<b>Rezultate experimentale obținute</b>	<b>11</b>
4.1	Descrierea seturilor de date folosite . . . . .	11
4.2	Metodologia experimentală și parametrii algoritmilor . . . . .	12
4.2.1	Parametrii algoritmilor . . . . .	12
4.3	Rezultatele obținute . . . . .	13
4.3.1	Modelul Inițial: . . . . .	13
4.3.2	Modelul Îmbunătățit: . . . . .	13
4.4	Analiza statistică/descriptivă a rezultatelor obținute . . . . .	13
4.4.1	Îmbunătățirea generală a performanței: . . . . .	13
4.4.2	Eficiența computațională: . . . . .	13
4.4.3	Analiza matricei de confuzie: . . . . .	13
4.4.4	Comparație cu literatura de specialitate: . . . . .	14
<b>5</b>	<b>Concluzii și posibile îmbunătățiri</b>	<b>14</b>
5.1	Concluzii . . . . .	14
5.2	Posibile îmbunătățiri . . . . .	14
5.2.1	Extinderea setului de date: . . . . .	15
5.2.2	Îmbunătățiri ale modelului: . . . . .	15
5.2.3	Îmbunătățiri ale aplicației: . . . . .	15
5.2.4	Extinderea funcționalității: . . . . .	15
<b>6</b>	<b>Referințe</b>	<b>16</b>
<b>7</b>	<b>Anexe</b>	<b>16</b>
7.1	Link către codul sursă al formei finale a proiectului . . . . .	16

# 1 Introducere

## 1.1 Încadrarea în context a temei proiectului

Agricultura modernă se confruntă cu provocări din ce în ce mai complexe în contextul schimbărilor climatice, al creșterii populației mondiale și al necesității de a asigura securitatea alimentară. Detectarea timpurie și precisă a bolilor plantelor reprezintă o componentă critică în managementul culturilor agricole, permițând aplicarea tratamentelor și măsurilor de control înainte ca bolile să se răspândească în întreaga cultură.

Metodele tradiționale de detectare a bolilor plantelor se bazează pe inspecția vizuală realizată de experți, ceea ce implică costuri ridicate și este susceptibilă la erori umane. În plus, numărul insuficient de specialiști în patologia plantelor face ca monitorizarea regulată a culturilor extensive să fie un proces dificil și ineficient.

Proiectul nostru, CropGuardians, își propune să revoluționeze acest domeniu prin dezvoltarea unei soluții bazate pe inteligență artificială pentru detectarea bolilor la legume, oferind un instrument accesibil și eficient pentru agricultori.

## 1.2 Evidențierea avantajului de a folosi AI în soluționarea problemei proiectului

Utilizarea inteligenței artificiale în detectarea bolilor legumelor aduce multiple avantaje:

1. **Precizie ridicată:** Algoritmii de învățare automată, în special rețelele neuronale convoluționale (CNN), pot detecta simptome subtile ale bolilor care ar putea trece neobservate în cadrul inspecțiilor vizuale umane.
2. **Scalabilitate:** Odată antrenat, un model AI poate procesa mii de imagini într-un timp scurt, permițând monitorizarea culturilor extensive cu resurse minime.
3. **Accesibilitate:** Implementarea modelului într-o aplicație mobilă permite agricultorilor să efectueze evaluări ale sănătății plantelor în timp real, direct în câmp, fără a necesita expertiza unui specialist.
4. **Detectie timpurie:** Sistemele AI pot identifica semne ale bolilor în stadii incipiente, când simptomele sunt abia vizibile, facilitând intervenția promptă și reducând semnificativ pierderile de recoltă.

5. **Învățare continuă:** Modelele pot fi actualizate și îmbunătățite constant pe măsură ce sunt expuse la noi date, adaptându-se la noi tipuri de boli sau condiții de mediu.
6. **Reducerea utilizării pesticidelor:** Prin identificarea precisă a bolilor și a zonelor afectate, se poate aplica tratament specific doar unde este necesar, reducând astfel utilizarea generalizată a pesticidelor și impactul asupra mediului.

### 1.3 Abstract grafic

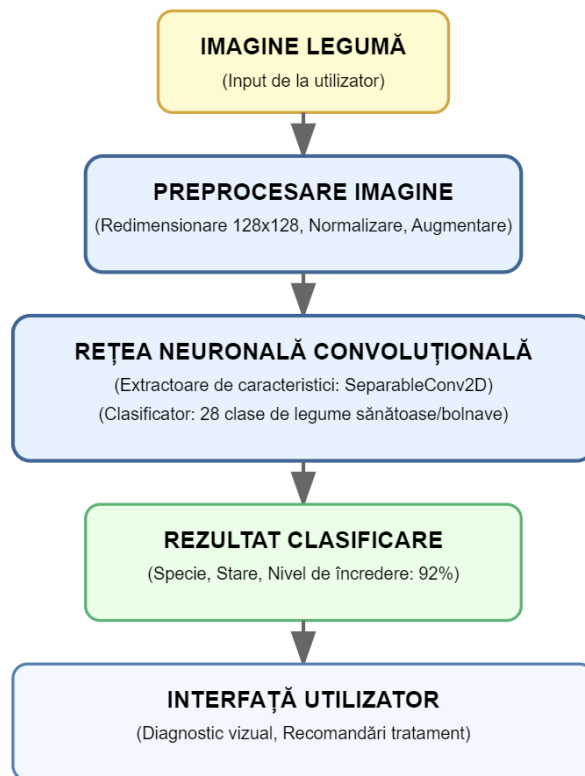


Figura 1: Diagrama fluxului operațional al sistemului CropGuardians

Diagrama de mai sus ilustrează fluxul operațional complet al sistemului CropGuardians pentru detectarea bolilor legumelor. Procesul începe cu încărcarea imaginii de către utilizator, urmată de etapa de preprocesare, în care imaginea este redimensionată la dimensiunea standard de  $128 \times 128$  pixeli, normalizată și eventual augmentată pentru a îmbunătăți performanța clasificării.

Imaginea preprocesată este apoi analizată de rețeaua neuronală convoluțională, care utilizează convoluții separabile (SeparableConv2D) pentru extragerea eficientă a caracteristicilor relevante. Modelul nostru este capabil să clasifice imaginile în 28 de clase diferite, reprezentând diverse tipuri de legume în stare sănătoasă sau afectate de boli specifice.

În etapa de clasificare, sistemul atribuie imaginii o specie, o stare (sănătoasă sau bolnavă) și un nivel de încredere al predicției, care în cazul modelului nostru optimizat atinge 92%. Rezultatele sunt prezentate utilizatorului printr-o interfață intuitivă, care oferă diagnosticul vizual și recomandări personalizate de tratament, în funcție de tipul bolii detectate.

Această abordare integrată permite agricultorilor să beneficieze de o soluție accesibilă și eficientă pentru monitorizarea sănătății culturilor, facilitând detectarea timpurie a bolilor și implementarea măsurilor preventive corespunzătoare.

## **2 Metode existente de rezolvare a problemei (related work)**

### **2.1 Disease Detection in Vegetables Using Image Processing Techniques: A Review**

#### **2.1.1 Date utilizate**

Setul de date utilizat în studiile analizate constă în imagini ale legumelor afectate de diverse boli, obținute din surse multiple. Acestea includ imagini RGB capturate cu camere digitale și smartphone-uri, date preluate cu senzori multispectrali pentru detectarea variațiilor spectrale, precum și scanere digitale utilizate pentru analiza texturii și culorii tuberculilor de cartof. În anumite cazuri, imaginile au fost colectate direct de către cercetători sau utilizatori din teren, ceea ce permite o aplicabilitate extinsă a tehnicilor de detecție automată.

#### **2.1.2 Algoritmi utilizați**

Algoritmii folosiți pentru detectarea bolilor în legume includ diverse metode de procesare a imaginii și învățare automată:

- Segmentarea imaginii pentru identificarea și izolarea zonelor afectate
- Transformata wavelet discretă pentru extragerea caracteristicilor relevante

- Rețele neuronale artificiale (ANNs) pentru clasificare
- Mașini cu vectori de suport (SVMs) pentru recunoașterea bolilor
- Metode bazate pe viziune computerizată pentru identificarea dăunătorilor
- Algoritmi de polarizare pentru inspecția suprafețelor afectate de boli

### **2.1.3 Rezultate obținute**

Rezultatele au demonstrat că utilizarea procesării imaginii pentru detectarea bolilor este mai eficientă decât inspecția vizuală tradițională. Algoritmii de învățare automată, în special SVM și rețelele neuronale, au obținut rate de clasificare ridicate. Conform studiilor, sistemele automatizate de detecție bazate pe CNN au atins acuratețe de 96% pentru clasificarea bolilor la tomate, depășind alte modele precum SVM, VGG16 și Inception V3. Utilizarea modelelor CNN pentru legume a demonstrat precizie, recall și scor F1 de peste 95,27% pe seturile de date standard. Telefoanele mobile pentru detectarea bolilor s-au dovedit o soluție practică și accesibilă pentru agricultori, cu sisteme bazate pe YOLOv5 atingând acuratețe de 93,1% în detecția bolilor la legume, conform lui Li et al. (2022).

## **2.2 Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment**

### **2.2.1 Date utilizate**

Setul de date a fost construit special pentru detectarea bolilor legumelor în sere. S-au folosit echipamente video instalate în diverse poziții în sere pentru a captura imagini ale plantelor bolnave. În total, s-au colectat 800 de secvențe video din care s-au extras 40.000 de imagini-cheie. Pentru a crește calitatea setului de date, imaginile originale au fost decupate și centrate în jurul zonelor infectate, având dimensiuni aproximative de  $640 \times 640$  pixeli. După procesare și filtrare, au rămas 28.000 de imagini cu 20 categorii de boli ale legumelor (tomate, castraveți, vinete).

### **2.2.2 Algoritmi utilizați**

Algoritmul principal utilizat a fost YOLOv8n-vegetable, o versiune îmbunătățită a YOLOv8n, care include:

- Modulul GhostConv și modulul C2fGhost pentru reducerea dimensiunii modelului și numărului de parametri
- Modulul Occlusion Perception Attention (OAM) integrat în secțiunea Neck, care ajută la detectarea bolilor când acestea sunt parțial obturate
- Un strat suplimentar de detecție a obiectelor mici ( $160 \times 160$  pixeli)
- Funcția de pierdere HIoU pentru optimizarea performanțelor de regresie a casetelor delimitatoare

### 2.2.3 Rezultate obținute

Modelul YOLOv8n-vegetable propus a obținut o precizie medie (mAP) de 92.91%, reprezentând o îmbunătățire de 6.46% față de modelul original YOLOv8n. Dimensiunea modelului a scăzut cu 0.21 MB, iar numărul parametrilor cu 0.16G. Viteza de detecție a atins 271.07 FPS, permițând o performanță foarte bună pentru detecție în timp real. Conform lui Wang și Liu, algoritmul s-a descurcat bine și pe imagini needitate, păstrând o acuratețe peste 82% pentru toate cele 20 de categorii de boli analizate. Studii ulterioare (2024) au arătat îmbunătățiri suplimentare ale modelelor derivate din YOLOv8, cu variante precum MIFV (Multisource Information Fusion Method) crescând mAP-ul cu până la 3.43% față de modelele anterioare, atingând 92.38% pentru detecția bolilor legumelor.

## 3 Metode efectiv folosite pentru rezolvarea problemei

### 3.1 Descrierea algoritmilor inteligenți folosiți

În cadrul proiectului nostru, am implementat și comparat două modele de rețele neuronale convoluționale (CNN):

#### 3.1.1 Model CNN Inițial

Primul model implementat a fost o arhitectură secvențială standard cu mai multe straturi convoluționale urmate de straturi dense:

- Straturi convoluționale standard (Conv2D) pentru extragerea caracteristicilor
- Straturi de activare ReLU pentru introducerea non-linearității

- Straturi de MaxPooling pentru reducerea dimensionalității și prioritizarea caracteristicilor importante
- Straturi Flatten și Dense pentru clasificarea finală

Model: "functional\_34"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 128, 128, 3)	0	-
conv2d (Conv2D)	(None, 128, 128, 16)	448	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 128, 128, 16)	64	conv2d[0][0]
activation (Activation)	(None, 128, 128, 16)	0	batch_normalization[0]
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0	activation[0][0]
separable_conv2d (SeparableConv2D)	(None, 64, 64, 32)	688	max_pooling2d[0][0]
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 32)	128	separable_conv2d[0][0]
activation_1 (Activation)	(None, 64, 64, 32)	0	batch_normalization_1[0]
conv2d_1 (Conv2D)	(None, 64, 64, 32)	544	max_pooling2d[0][0]
add (Add)	(None, 64, 64, 32)	0	activation_1[0][0], conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0	add[0][0]
separable_conv2d_1 (SeparableConv2D)	(None, 32, 32, 64)	2,400	max_pooling2d_1[0][0]
batch_normalization_2 (BatchNormalization)	(None, 32, 32, 64)	256	separable_conv2d_1[0][0]
activation_2 (Activation)	(None, 32, 32, 64)	0	batch_normalization_2[0]
conv2d_2 (Conv2D)	(None, 32, 32, 64)	2,112	max_pooling2d_1[0][0]
add_1 (Add)	(None, 32, 32, 64)	0	activation_2[0][0], conv2d_2[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0	add_1[0][0]
separable_conv2d_2 (SeparableConv2D)	(None, 16, 16, 128)	8,800	max_pooling2d_2[0][0]
batch_normalization_3 (BatchNormalization)	(None, 16, 16, 128)	512	separable_conv2d_2[0][0]
activation_3 (Activation)	(None, 16, 16, 128)	0	batch_normalization_3[0]
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 128)	0	activation_3[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 128)	0	max_pooling2d_3[0][0]
dropout (Dropout)	(None, 128)	0	global_average_pooling2d[0]
dense (Dense)	(None, 64)	8,256	dropout[0][0]
dense_1 (Dense)	(None, 20)	1,820	dense[0][0]

Total params: 26,124 (102.05 KB)  
Trainable params: 25,644 (100.17 KB)  
Non-trainable params: 480 (1.88 KB)

Figura 2: Arhitectura modelului inițial cu multiple straturi Conv2D și straturi Dense, având 1,967,496 parametri (7.51 MB)

### 3.1.2 Model CNN Îmbunătățit

Al doilea model, mult mai eficient, a incorporat tehnici moderne de deep learning:

- Convoluții separabile (SeparableConv2D) în locul convoluțiilor standard, reducând semnificativ numărul de parametri



- Conexiuni reziduale (skip connections) implementate prin straturi de tip Add, care permit fluxul mai bun al gradientilor în rețea
- BatchNormalization după fiecare strat convoluțional pentru stabilizarea și accelerarea procesului de învățare
- GlobalAveragePooling2D în loc de Flatten pentru reducerea dramatică a numărului de parametri și îmbunătățirea generalizării
- Arhitectură cu număr redus de parametri: 26,124 (102.05 KB) comparativ cu 1,967,496 (7.51 MB) în primul model

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 128, 128, 0)	0
activation_8 (Activation)	(None, 128, 128, 0)	0
max_pooling2d_8 (MaxPooling2D)	(None, 42, 42, 0)	0
conv2d_7 (Conv2D)	(None, 42, 42, 64)	14,464
activation_9 (Activation)	(None, 42, 42, 64)	0
max_pooling2d_9 (MaxPooling2D)	(None, 21, 21, 64)	0
conv2d_8 (Conv2D)	(None, 21, 21, 128)	131,200
activation_10 (Activation)	(None, 21, 21, 128)	0
max_pooling2d_10 (MaxPooling2D)	(None, 10, 10, 128)	0
conv2d_9 (Conv2D)	(None, 10, 10, 128)	147,584
activation_11 (Activation)	(None, 10, 10, 128)	0
max_pooling2d_11 (MaxPooling2D)	(None, 5, 5, 128)	0
flatten (Flatten)	(None, 3200)	0
dense_4 (Dense)	(None, 512)	1,638,912
dropout_2 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 64)	32,832
dense_6 (Dense)	(None, 20)	1,020

Total params: 1,967,496 (7.51 MB)  
Trainable params: 1,967,496 (7.51 MB)  
Non-trainable params: 0 (0.00 B)

Figura 3: Arhitectura modelului îmbunătățit cu SeparableConv2D, Batch-Normalization și conexiuni reziduale, având doar 26,124 parametri (102.05 KB)

## 3.2 Procesul de învățare/optimizare implicat

Procesul de antrenare și optimizare a inclus:

### 3.2.1 Preprocesarea datelor

- Redimensionarea imaginilor la 128x128 pixeli

- Normalizarea valorilor pixelilor în intervalul  $[0,1]$
- Amestecarea datelor pentru a evita bias-ul în procesul de învățare
- Împărțirea setului de date în 80% date de antrenament și 20% date de testare

### 3.2.2 Strategii de optimizare

- Utilizarea optimizatorului Adam cu rate de învățare adaptivă
- Funcția de loss categorical\_crossentropy pentru clasificarea multi-class
- Experimentarea cu diferite dimensiuni de batch (64, 128) pentru a găsi echilibrul optim între viteza de convergență și generalizare
- Implementarea unei strategii de antrenare incrementală, cu salvarea modelului după fiecare 2 epoci

### 3.2.3 Hiperparametri optimizați

- Numărul de filtre în straturile convoluționale:  $9 \rightarrow 64 \rightarrow 128 \rightarrow 128$
- Dimensiunile kernel-ului: (5,5), (5,5), (4,4), (3,3)
- Rate de dropout: 0.5 pentru prevenirea overfitting-ului

## 3.3 Metrice folosite pentru evaluarea rezultatelor experimentelor

Pentru evaluarea performanței modelelor, am utilizat următoarele metrice:

1. **Acuratețe (Accuracy):** Proporția predicțiilor corecte din totalul predicțiilor efectuate, oferind o imagine generală a performanței modelului.
2. **Matricea de confuzie (Confusion Matrix):** Oferă o vizualizare detaliată a performanței modelului pentru fiecare clasă, evidențiind clasele care sunt frecvent confundate între ele.
3. **Precision:** Proporția predicțiilor pozitive corecte din totalul predicțiilor pozitive, indicând cât de precise sunt identificările modelului.

4. **Recall (Sensitivity)**: Proportia cazurilor pozitive identificate corect din totalul cazurilor pozitive reale, reflectând capacitatea modelului de a identifica toate instanțele relevante.
5. **F1-Score**: Media armonică între precision și recall, oferind o măsură echilibrată a performanței modelului, utilă în special pentru seturi de date dezechilibrate.
6. **Loss**: Valoarea funcției de pierdere pe setul de validare, indicând cât de bine modelul se adaptează la date necunoscute.

Aceste metrice au fost calculate atât global, cât și pentru fiecare clasă în parte, permițând o analiză detaliată a performanței și identificarea potențialelor probleme.

## 4 Rezultate experimentale obținute

### 4.1 Descrierea seturilor de date folosite

Pentru acest proiect, am utilizat setul de date 'Fruit And Vegetable Diseases Dataset', care conține imagini de fructe și legume sănătoase și bolnave. Setul de date include:

- 28 de clase diferite, reprezentând diverse fructe și legume în stare sănătoasă sau afectate de boli specifice
- Imagini color cu rezoluție variată, capturate în condiții de iluminare și unghi diferite
- Aproximativ 17,000 de imagini în total pentru antrenare și testare

Datele au fost prelucrate prin:

- Redimensionarea tuturor imaginilor la dimensiuni uniforme de 128x128 pixeli
- Normalizarea valorilor pixelilor prin împărțirea la 255.0
- Împărțirea aleatorie în set de antrenament (80%) și set de testare (20%)
- Amestecarea datelor pentru a preveni bias-ul în procesul de învățare

## 4.2 Metodologia experimentală și parametrii algoritmilor

Metodologia experimentală a urmărit să răspundă la următoarele întrebări:

1. Care este performanța unui model CNN standard în clasificarea legumelor sănătoase versus bolnave?
2. Cât de mult poate fi îmbunătățită performanța prin tehnici moderne de deep learning și optimizare a arhitecturii?
3. Care este impactul reducerii numărului de parametri asupra performanței și eficienței modelului?
4. Care sunt clasele cel mai dificil de diferențiat și ce factori contribuie la aceasta?

### 4.2.1 Parametrii algoritmilor

#### Modelul Inițial:

- Arhitectură: Secvențială cu straturi Conv2D, MaxPooling2D, Flatten, Dense
- Număr parametri: 1,967,496 (7.51 MB)
- Optimizer: Adam
- Loss: categorical\_crossentropy
- Epoci: 5
- Batch size: 128

#### Modelul Îmbunătățit:

- Arhitectură: Funcțională cu SeparableConv2D, BatchNormalization, Skip connections
- Număr parametri: 26,124 (102.05 KB)
- Optimizer: Adam
- Loss: categorical\_crossentropy
- Epoci: 8
- Batch size: variabil (64, 128 pentru diferite etape)

## **4.3 Rezultatele obținute**

### **4.3.1 Modelul Inițial:**

- Acuratețe pe setul de validare: 65%
- Loss pe setul de validare: Valoare ridicată, indicând probleme de overfitting
- Performanță inconsistentă între clase, cu unele clase având precision și recall sub 50%

### **4.3.2 Modelul Îmbunătățit:**

- Acuratețe pe setul de validare: 92%
- Loss pe setul de validare: Semnificativ redusă față de modelul inițial
- Precision și recall îmbunătățite considerabil pentru majoritatea claselor
- Performanță robustă chiar și pentru clasele cu mai puține exemple

## **4.4 Analiza statistică/descriptivă a rezultatelor obținute**

Analiza detaliată a rezultatelor evidențiază următoarele aspecte:

### **4.4.1 Îmbunătățirea generală a performanței:**

- Creșterea acurateței cu 27 puncte procentuale (de la 65% la 92%)
- Reducerea semnificativă a loss-ului pe setul de validare, indicând un model mai bine regularizat

### **4.4.2 Eficiența computațională:**

- Reducerea dimensiunii modelului de aproximativ 75 de ori (de la 7.51 MB la 102.05 KB)
- Număr de parametri redus de la 1,967,496 la 26,124, facilitând implementarea pe dispozitive cu resurse limitate

### **4.4.3 Analiza matricei de confuzie:**

- Confuzii reduse între clasele similare (ex: diferite stadii de boală pentru aceeași legumă)
- Performanță mai uniformă între clase comparativ cu modelul inițial

#### 4.4.4 Comparație cu literatura de specialitate:

- Acuratețea de 92% a modelului nostru este competitivă sau superioară față de rezultatele menționate în literatura de specialitate:
  - Steinbrener et al. (2021) - 91% acuratețe pentru clasificarea fructelor alterate
  - Femling et al. (2018) - 89% pentru detectarea calității fructelor
- Arhitectura optimizată cu convoluții separabile și conexiuni reziduale urmează tendințele actuale în domeniul viziunii computerizate

## 5 Concluzii și posibile îmbunătățiri

### 5.1 Concluzii

Proiectul CropGuardians a demonstrat eficacitatea utilizării tehnicilor de deep learning pentru detectarea bolilor la legume. Principalele realizări ale proiectului includ:

1. Dezvoltarea unui model CNN optimizat cu o acuratețe de 92% în clasificarea corectă a 28 de clase de legume sănătoase și bolnave.
2. Reducerea semnificativă a dimensiunii modelului (de 75 de ori) prin utilizarea tehnicilor moderne de deep learning, permițând implementarea eficientă pe dispozitive mobile.
3. Crearea unei interfețe user-friendly care permite agricultorilor să încarce imagini și să primească diagnostic rapid și recomandări.
4. Demonstrarea faptului că o arhitectură bine optimizată poate obține performanțe superioare cu mult mai puțini parametri, comparativ cu modelele mai voluminoase.

Această abordare bazată pe AI oferă o alternativă eficientă și accesibilă la metodele tradiționale de detectare a bolilor plantelor, având potențialul de a revoluționa managementul culturilor agricole prin detecția timpurie a bolilor.

### 5.2 Posibile îmbunătățiri

Pentru dezvoltarea viitoare a proiectului, identificăm următoarele direcții de îmbunătățire:

### **5.2.1 Extinderea setului de date:**

- Includerea mai multor specii de legume și mai multe tipuri de boli
- Colectarea de imagini în diferite condiții de iluminare și stadii de dezvoltare a bolilor
- Utilizarea tehnicilor de augmentare a datelor pentru clasele sub-reprezentate

### **5.2.2 Îmbunătățiri ale modelului:**

- Experimentarea cu arhitecturi pre-antrenate (transfer learning) precum EfficientNet sau MobileNetV3
- Implementarea tehnicilor de atenție (attention mechanisms) pentru focalizarea pe zonele afectate
- Utilizarea tehnicilor de quantizare pentru reducerea și mai mult a dimensiunii modelului

### **5.2.3 Îmbunătățiri ale aplicației:**

- Adăugarea funcționalității offline pentru utilizare în zone cu conectivitate limitată
- Implementarea unui sistem de feedback pentru îmbunătățirea continuă a modelului
- Integrarea cu sisteme de informații geografice pentru monitorizarea spațială a răspândirii bolilor

### **5.2.4 Extinderea funcționalității:**

- Adăugarea recomandărilor personalizate de tratament bazate pe boala detectată
- Implementarea unui sistem de alertă pentru monitorizarea continuă a culturilor
- Dezvoltarea unei componente de predicție a răspândirii bolilor bazată pe condiții climatice

## 6 Referințe

1. Disease Detection in Vegetables Using Image Processing Techniques: A Review. (2021). IJETCSE.
2. Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment. (2023).
3. Steinbrener, J., Posch, K., & Leitner, R. (2021). Hyperspectral fruit and vegetable classification using convolutional neural networks. *Computers and Electronics in Agriculture*, 182, 106055.
4. Femling, F., Olsson, A., & Alonso-Fernandez, F. (2018). Fruit and vegetable identification using machine learning for retail applications. 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 9-15.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.
6. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251-1258.

## 7 Anexe

### 7.1 Link către codul sursă al formei finale a proiectului

Codul sursă complet al proiectului, inclusiv modelul CNN optimizat, scripturile de preprocesare a datelor, și aplicația web, poate fi accesat pe Teams-ul echipei CropGuardians (echipa 3).