

UNIVERSITATEA BABEŞ BOLYAI, CLUJ NAPOCA, ROMÂNIA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

MEDICAL ASSISTENT

– MIRPR –

Membrii Echipei

Gherasim Georgian, Informatică, 233

Hârşan Mihnea, Informatică, 233

Irimiciuc Andreea, Informatică, 237

Abstract

Cancerul vezicii urinare este una dintre cele mai frecvente și mai costisitoare afecțiuni maligne umane de tratat. Succesul tratamentului depinde de stadiul tumorii primare și de starea ganglionilor limfatici regionali [Vezi rata de cazuri noi și decese și epidemiologia cancerului vezicii urinare.](#)

Asistentul automatizat abordează problema procesării și clasificarea imaginilor venind astfel în sprijinul studenților de la medicină pentru o mai bună înțelegere și interpretare a diagnosticului prezentând vizual informații caracteristice. Această aplicație are rolul de a îmbunătăți calitatea procesului de învățare și asimilarea de noi cunoștințe într-o manieră mai atractivă cu ajutorul algoritmilor inteligenți.

Metoda noastră se bazează pe segmentarea imaginilor pe regiuni folosind măști pentru organele investigate. Pentru crearea modelului de localizare a organelor, segmentarea ROI, am utilizat o rețea de tip darknet. Structura algoritmului este de forma U-Net cu scurtături pe o structură de Convolutional Neural Network. Pentru detectarea obiectelor în timp real am folosit și algoritmul YOLO și biblioteca Tensorflow pentru contruirea modelelor.

În final, am obținut o precizie și o acuratețe mare de 0.788 datorită utilizării de scurtături în rețea și un timp de răspuns mic datorită folosirii unui algoritm eficient de ROI.

Contents

1	Introducere	1
1.1	Descrierea aplicației	1
1.2	Flow-ul aplicației	2
2	Problema științifică	4
2.1	Descrierea problemei	4
3	Stadiul artei/Lucrări conexe	5
4	Abordare propusă	7
4.1	U-net	18
4.2	YOLO	19
4.3	Tensorflow	21
5	Aplicație (validare numerică)	22
5.1	Metodologie	22
5.2	Date	22
5.3	Rezultate	23
6	Concluzii și lucrări viitoare	29

List of Tables

4.1	Parametrii modelului standard de antrenament Yolo	19
5.1	Primul model de antrenament	26
5.2	Al doilea model de antrenament	27
5.3	Parametrii modelului de antrenament	28

List of Figures

1.1	Mindmap	2
1.2	Schema flow-ului aplicației	2
1.3	Diagrama flow-ului aplicației	3
3.1	SOTA (State of the art)	6
4.1	Masck Defect Class 1 prima fază	8
4.2	Masck Defect Class 2 prima fază	8
4.3	Masck Defect Class 1 a doua fază	9
4.4	Masck Defect Class 2 a doua fază	9
4.5	Masck Defect Class 1 a treia fază	10
4.6	Masck Defect Class 2 a treia fază	10
4.7	Masck Defect Class 1 a patra fază	11
4.8	Masck Defect Class 2 a patra fază	11
4.9	Masck Defect și Predicted Mask Class 1 a prima fază (2 pacienți)	12
4.10	Masck Defect și Predicted Mask Class 1 a doua fază (2 pacienți)	12
4.11	Masck Defect și Predicted Mask Class 1 a treia fază (2 pacienți)	13
4.12	Masck Defect Mask Class 1 a prima fază (19 pacienți)	13
4.13	Masck Defect Class 1 a doua fază (19 pacienți)	14
4.14	Masck Defect Mask Class 1 a treia fază (19 pacienți)	14
4.15	Masck Defect Mask Class 1 a patra fază (19 pacienți)	15
4.16	Masck Defect Class 1 a cincea fază (19 pacienți)	15
4.17	Masck Defect Mask Class 1 a şasea fază (19 pacienți)	16
4.18	Masck Defect Mask Class 1 a şaptea fază (19 pacienți)	16
4.19	Masck Defect Class 1 a opta fază (19 pacienți)	17
4.20	Masck Defect Mask Class 1 a noua fază (19 pacienți)	17
4.21	Convolutional Networks pentru segmentarea imaginilor biomedicale	18
4.22	Algoritm YOLO pentru detectarea obiectelor	19
4.23	Valoarea loss-ului pentru iterațiile de YOLO de la 800 la 900	20
4.24	Clasificare pe straturi a imaginii cu Tensorflow	21
5.1	Rezultate algoritm de segmentare pentru loss cu 250 de epoci	23
5.2	Rezultate algoritm de segmentare pentru model loss cu 100 de epoci după aprox. 0.01	23
5.3	Îmbunătire algoritm de segmentare pentru loss	23
5.4	Rezultate algoritm de segmentare pentru accuracy cu 250 de epoci	24
5.5	Rezultate algoritm de segmentare pentru model accuracy cu 100 de epoci după aprox. 0.01	24
5.6	Rezultate algoritm de segmentare pentru model accuracy și model loss cu 50 de epoci pentru 2 pacienți	25

5.7 Rezultate algoritm de segmentare pentru model accuracy și model loss cu 100 de epoci pentru 19 pacienți	25
--	----

List of Algorithms

1	Detectarea predicțiile obiectelor	20
---	---	----

Chapter 1

Introducere

1.1 Descrierea aplicației

Pentru a ajuta la procesul de învățare a studenților de la medicină aceștia dispun de o aplicație mobilă care le prezintă vizual informații relevante despre vezica urinară și peretele acestuia cu straturile corespunzatoare, precum și a unor leziuni posibile. Prin urmare, plecând de la informațiile RMN (Imagistica prin rezonanță magnetică), studenții pot să vizualizeze 2D imagini cu vezica urinară sau tumori posibile. Un student poate să încarce o arhivă de tipul .rar cu imagini în formatul .jpeg și să verifice dacă există sau nu malformații ale viziunii urinare.

- Care este problema (științifică)?

Problema abordează procesarea limbajului natural, înțelegerea limbajului natural și clasificarea imaginilor. Utilizatorul va încărca o arhivă de imagini cu vezica urinară în format .jpeg pentru detectarea posibilelor anomalii. Algoritmii inteligenți de pe server vor deduce dacă sunt malformații la nivelul vezicii urinare și vor returna un fișier .gif către utilizator facilitând astfel procesul de învățare pentru studenții de la medicină.

- De ce este important?

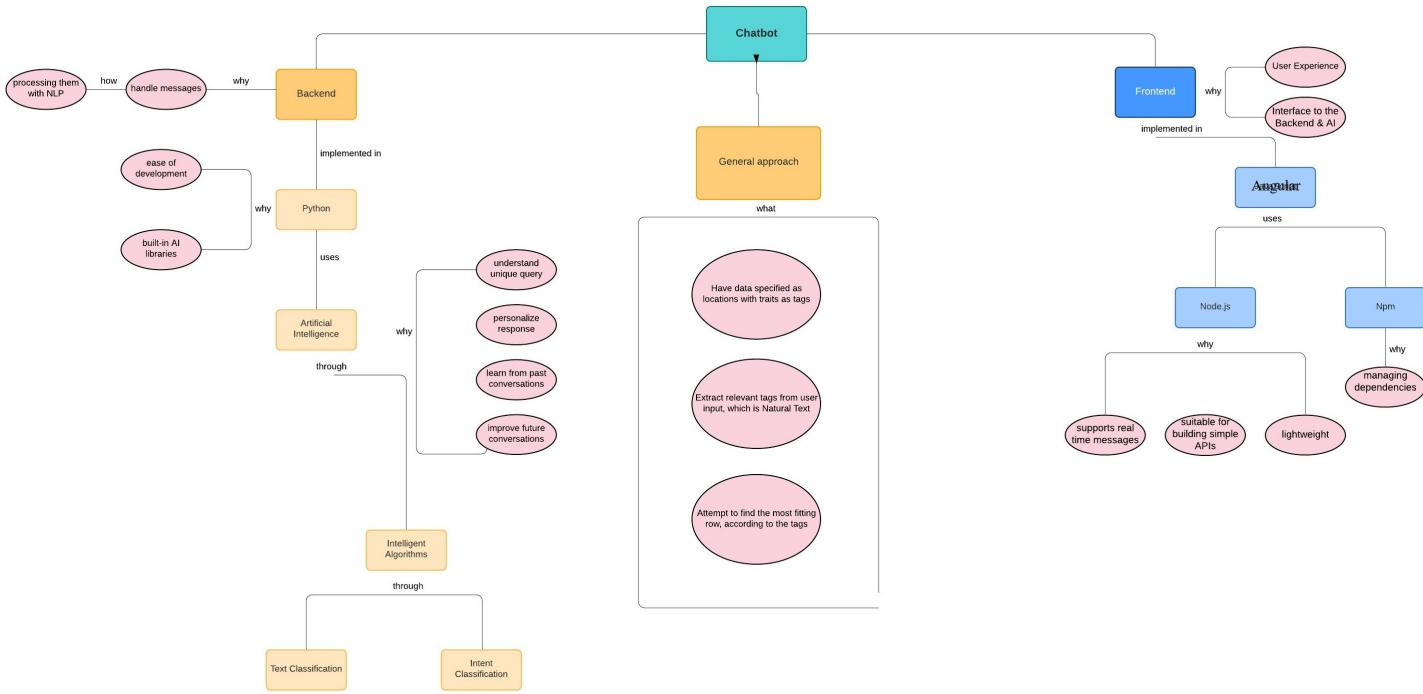
Această aplicație este un asistent automatizat pentru studenții de la medicină, ceea ce înseamnă că va prezenta vizual informații relevante despre vezica urinară și peretele acesteia cu straturile corespunzatoare, precum și a unor leziuni posibile sau tumori.

- Care este abordarea de bază?

Construim un prototip care demonstrează funcționalitățile de bază ale aplicației. Aceasta include încărcarea unei arhive de imagini și recunoașterea tumorilor maligne a vezicii urinare din setul

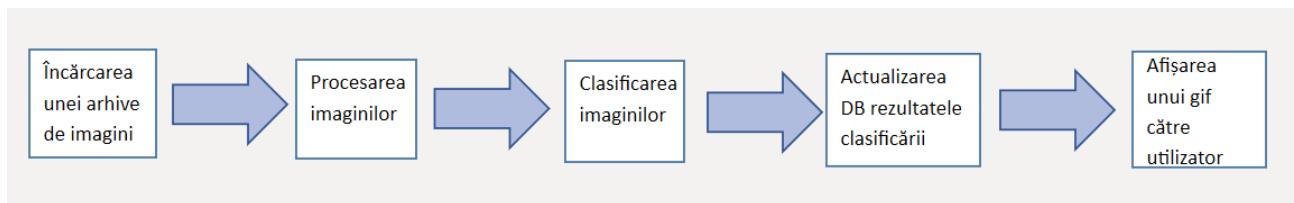
de date încărcat de către utilizator folosind API-ului Upload împreună cu AI-ul nostru pentru a oferi informații potrivite.

Figure 1.1: Mindmap



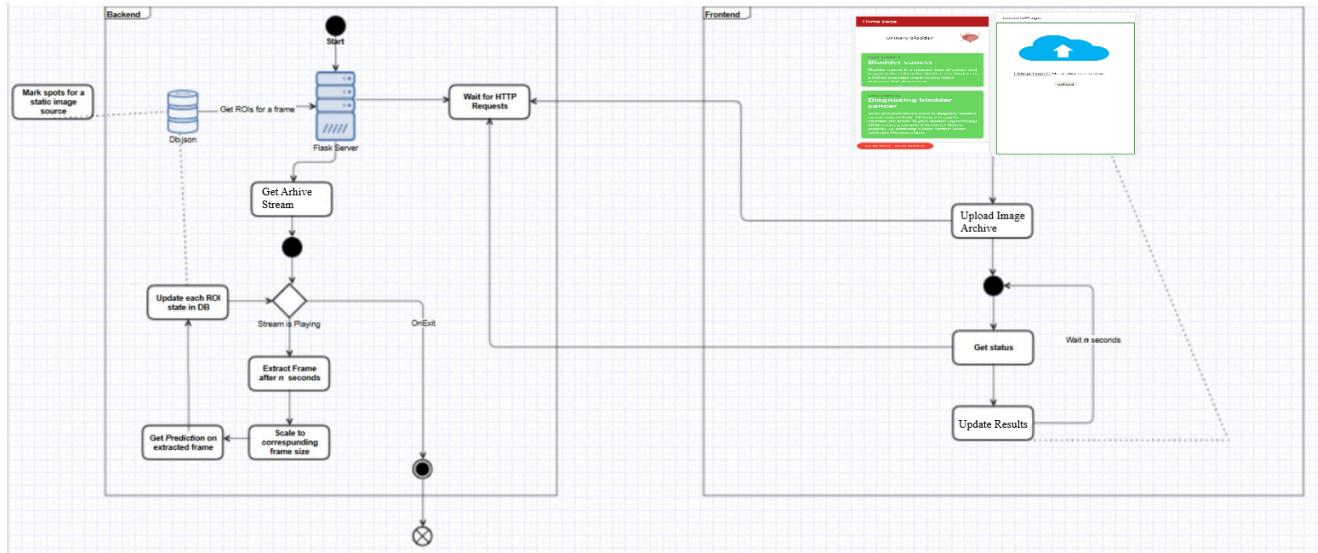
1.2 Flow-ul aplicației

Figure 1.2: Schema flow-ului aplicației



După ce aplicația pornește studentul alege opțiunea de încarcare a unei arhive de imagini în format .jpeg prin butonul "Upload". Arhiva respectivă cu imagini se trimite la server, unde imaginile vor fi preluate de algoritmi inteligenți care detectează posibilele malformații ale vezicii urinare bazându-se pe informațiile furnizate de utilizator. Serverul returnează o imagine de tip .gif care scoate în evidență posibilele anomalii.

Figure 1.3: Diagrama flow-ului aplicăției



Chapter 2

Problema științifică

2.1 Descrierea problemei

Algoritmii de inteligență artificială au rol important pentru culegerea datelor, contribuie la procesele de diagnosticare, la dezvoltarea protocolului de tratament, la dezvoltarea medicamentelor sau și în medicina personalizată. În analizarea problemei vom utiliza 4 cazuri:

- cazul normal
- tumoare malignă sau neoplasm malign
- pietre urinare
- cistita emfizematoasă

Din punct de vedere plastic, pe baza unei imagini RMN a unei vezici urinare se poate observa unul dintre cele 4 cazuri analizând imaginea la nivelul pixelilor utilizând algoritmi inteligenți pe imagini.

Din punctul de vedere formal avem proiectarea unui algoritm ce realizează clasificarea vezicii urinare dintr-o imagine RMN în 4 clase. Algoritmul analizează vezica cu ajutorul unei rețele CNN (vezi Figure 4.21) și determină dacă vezica este sanatoasă sau nu.

Chapter 3

Stadiul artei/Lucrări conexe

În contextul cercetării noastre ne-am bazat pe lucrări, cum ar fi: pentru detectarea tumorilor pe creier [Vezi detectarea tumorii cerebrale](#), segmentarea automată a prostatei și a zonelor de prostată [Vezi segmentarea prostatei și a zonelor de prostată](#). De asemenea, ne-am axat pe algoritmi și metode inteligente de învățare.

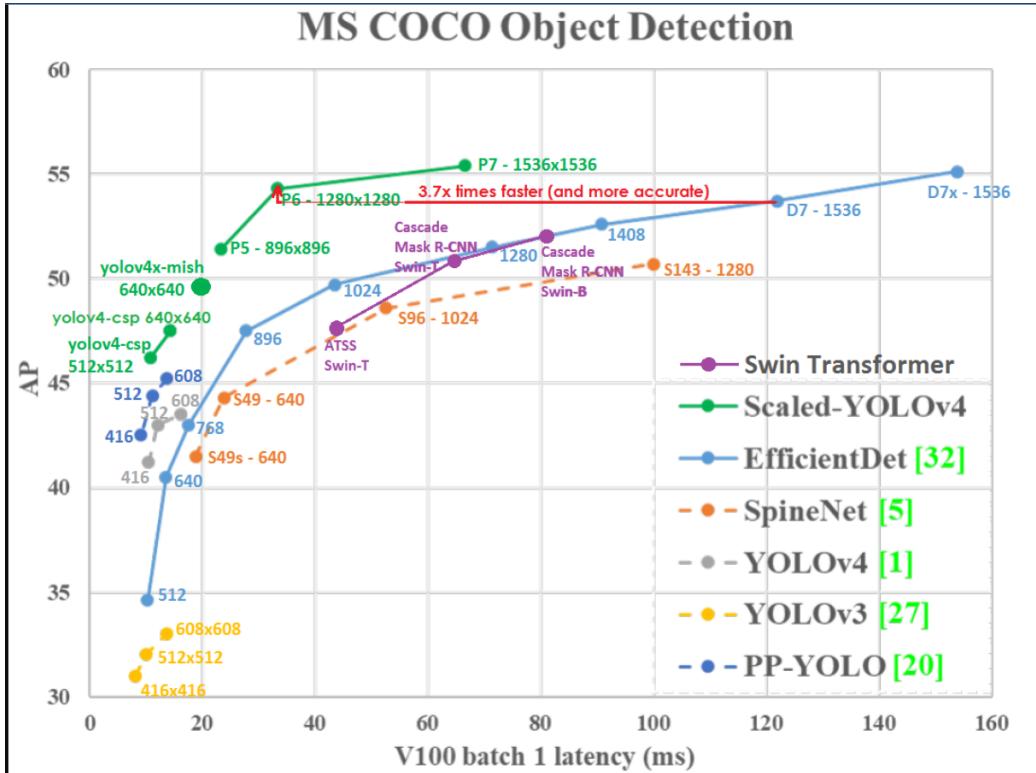
În primul stadiu al cercetării am utilizat informații și surse de date de pe [Data analysis](#) pentru diferite etape ale tumorii pentru vezica urinară [Vezi standalizarea cancerului vezicii urinare](#) la care am aplicat algoritmi pentru segmentarea imagilor și antrenarea algoritmului folosind măști create de mână, aproximativ 50 de măști în prima fază.

În a doua parte a cercetării ne-am axat pe un set de date de la 19 pacienți primit de la un centru medical. Algoritmul a fost optimizat și antrenat pe noul set de date cu măștile generate, un total de 446 de măști pe baza datelor noi.

În final, s-a obținut o precizie și o acuratețe foarte bună, fiind mai mare de 0.7.

Mai jos este prezentat o comparație de algoritmi:

Figure 3.1: SOTA (State of the art)



Putem observa ca Scaled-YOLOv4 este de 3.7x mai rapid decât EfficientDet pentru un batch de 100 de imagini și average precision de 55.

Chapter 4

Abordare propusă

Abordarea propusă de noi se bazează pe segmentarea imaginii pe regiuni. Scopul segmentării imaginii este de a înțelege și extrage informații din imagini la nivel de pixel.

Segmentarea imaginilor medicale este un pas important de procesare a imaginii. Compararea imaginilor pentru a evalua calitatea segmentării este o parte esențială a măsurării progresului în acest domeniu de cercetare [Vezi Deep Learning de segmentare a imaginilor medicale](#).

Segmentarea imaginii poate fi utilizată pentru recunoașterea și localizarea obiectelor, ceea ce oferă o valoare extraordinară în multe aplicații.

Folosind segmentarea imaginii, o rețea neuronală va fi antrenată pentru a produce o mască [Vezi Mask Converter](#) în funcție de pixelii imaginii. În cazul în care dimensiunea imaginii mai mare, atunci se va redimensiona pe lățime și lungime înainte de aplicarea algoritmului.

O mască reprezintă ieșirea produsă de modelul de segmentare a imaginii, iar în rezolvarea noastră am utilizat o mască pentru pentru cavitatea vezicii, tumoare și pentru peretele vezicii urinare. Pentru a reprezenta această mască, trebuie mai întâi să aplatizam imaginea într-o matrice. În prima fază de lucru a proiectului, măștile au fost create de mâna din lipsa de date, mai apoi au fost generate pe baza setului de date primit de la un centru medical.

Tehnicile moderne de segmentare a imaginilor se bazează pe abordarea de învățare profundă, care utilizează arhitecturi comune, cum ar fi CNN, FCNs (Fully Convolution Networks) și Deep Encoders Decoders [Vezi Encode și Decode](#). La acest proiect, arhitectura Res-U-Net [Vezi ResUNet-a Baseline pe TensorFlow](#) este folosită pentru a îndeplini această sarcină.

Pentru segmentarea ROI s-a folosit rețeaua neuronală darknet [Vezi Darknet](#) pentru a crea modelul de localizare a vezicii, identificarea peretelui, respectiv a cavității vezicii sau a tumorii.

În continuare sunt prezentate date parțiale de măști create în primul stadiu de cercetare realizate de mână din lipsa de date.

Figure 4.1: Mask Defect Class 1 prima fază

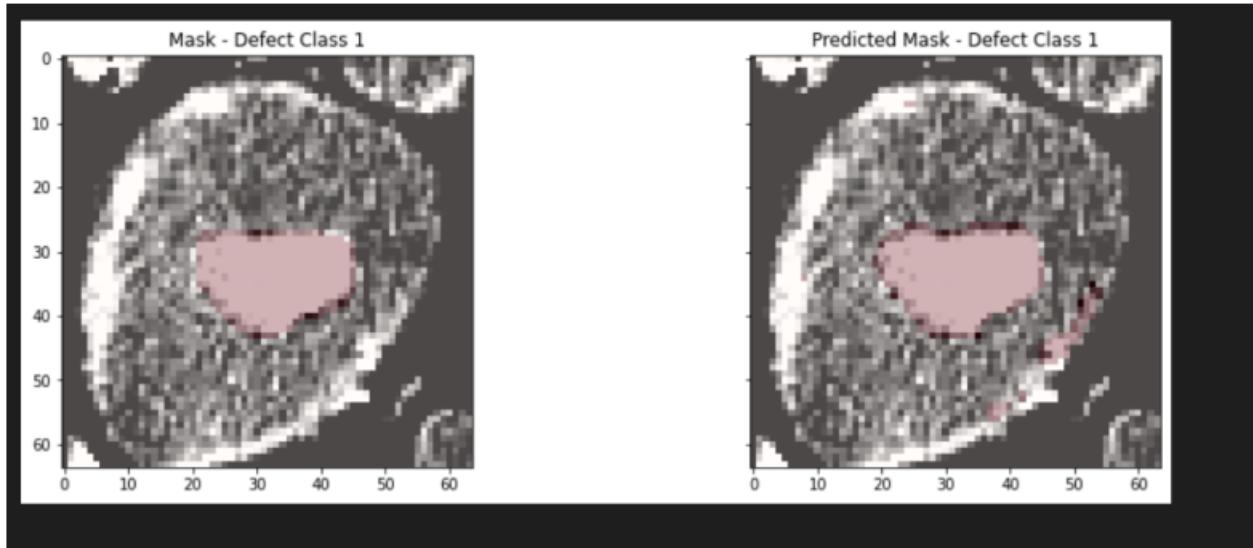


Figure 4.2: Mask Defect Class 2 prima fază

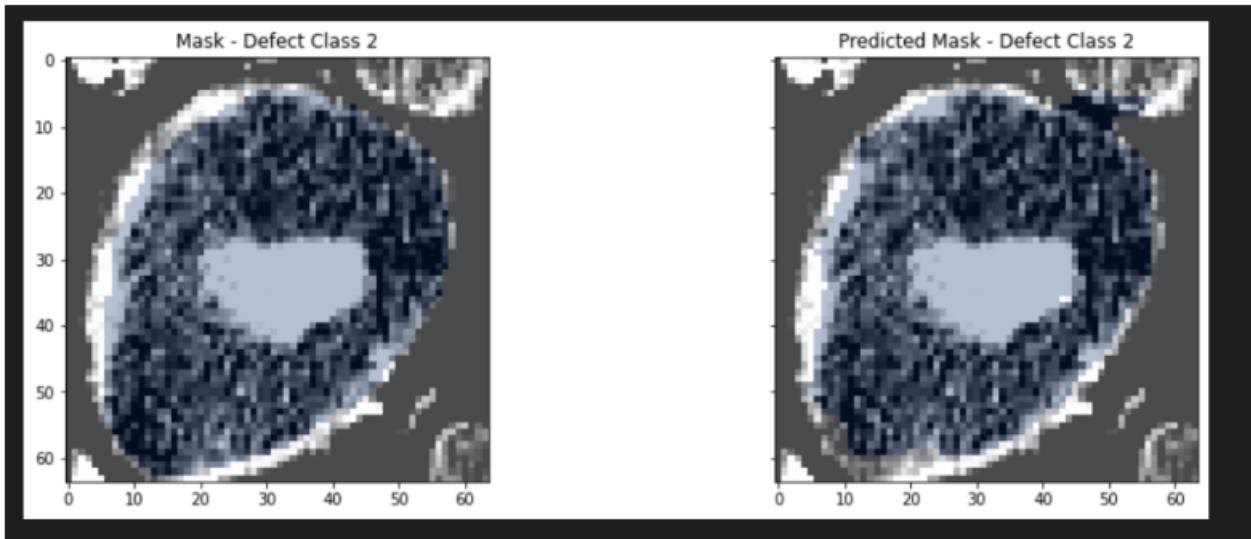


Figure 4.3: Mask Defect Class 1 a doua fază

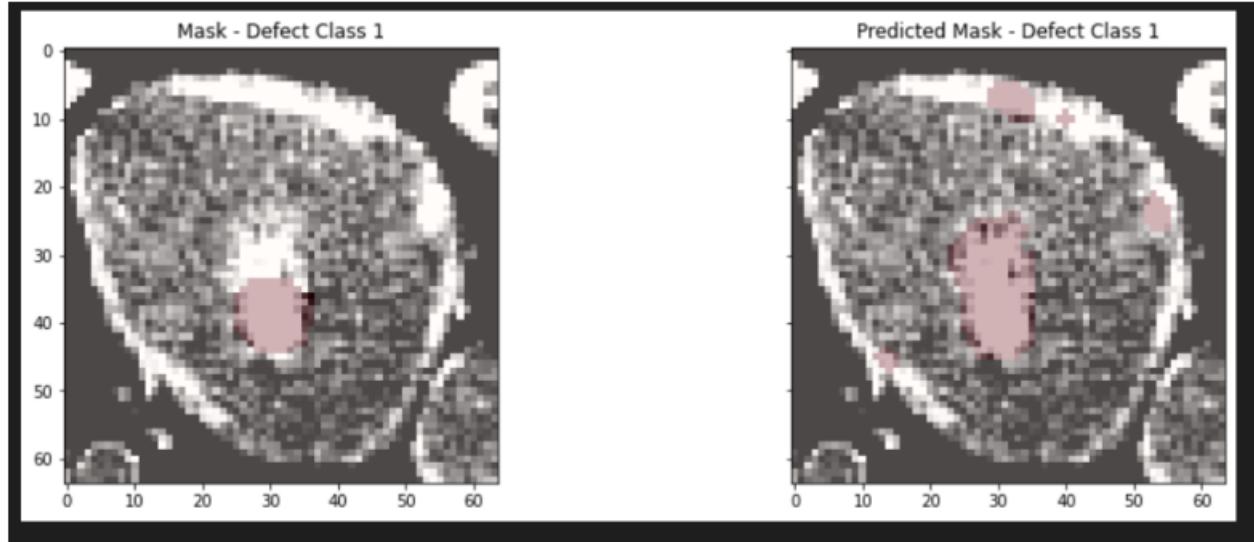


Figure 4.4: Mask Defect Class 2 a doua fază

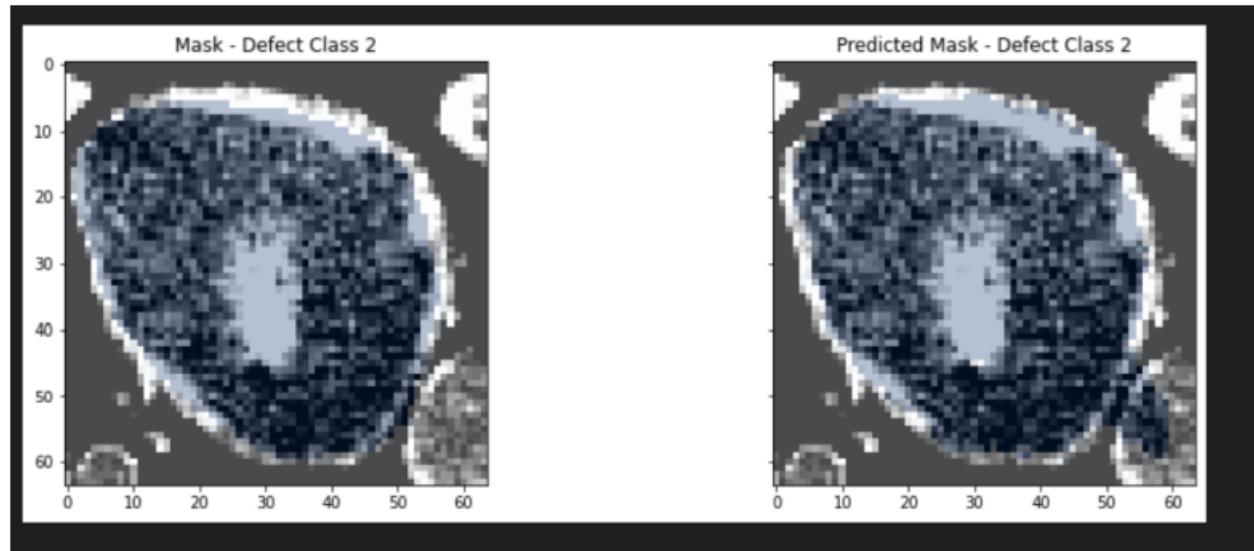


Figure 4.5: Mask Defect Class 1 a treia fază

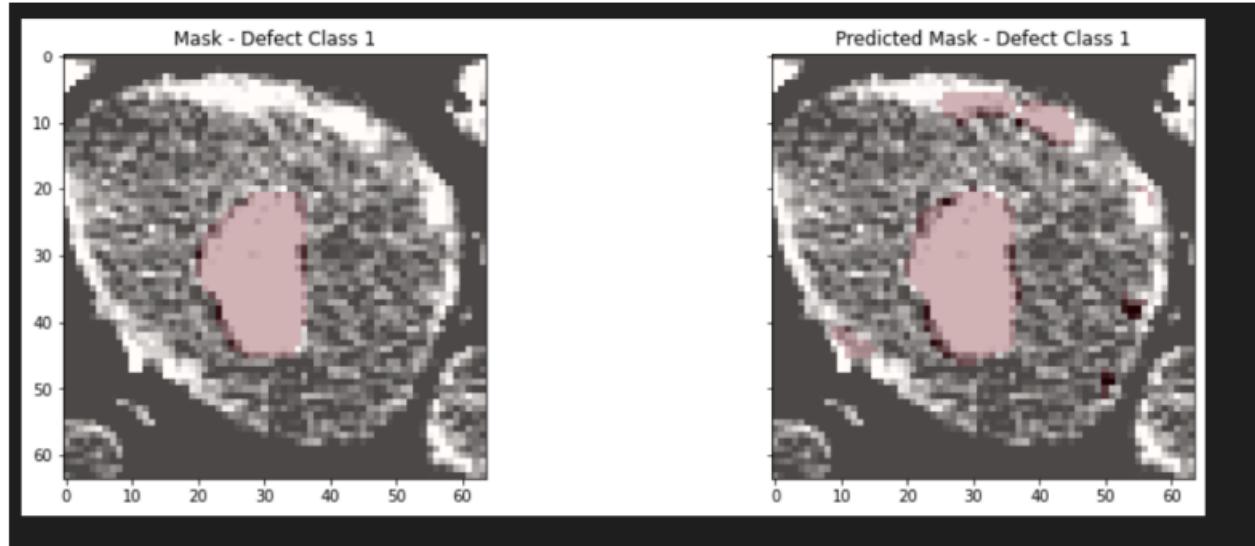


Figure 4.6: Mask Defect Class 2 a treia fază

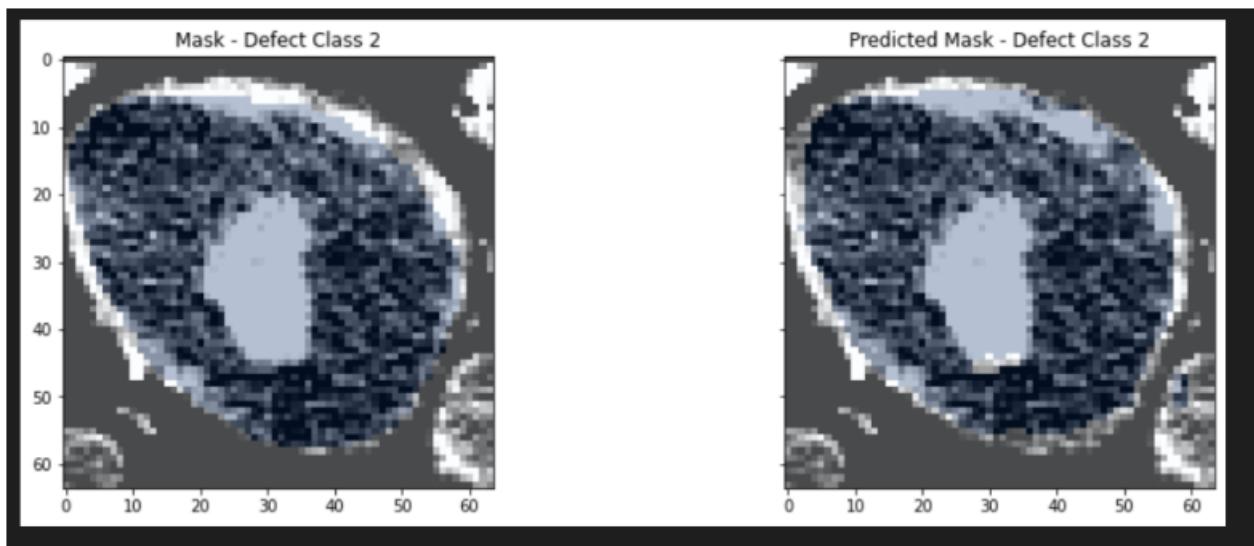


Figure 4.7: Mask Defect Class 1 a patra fază

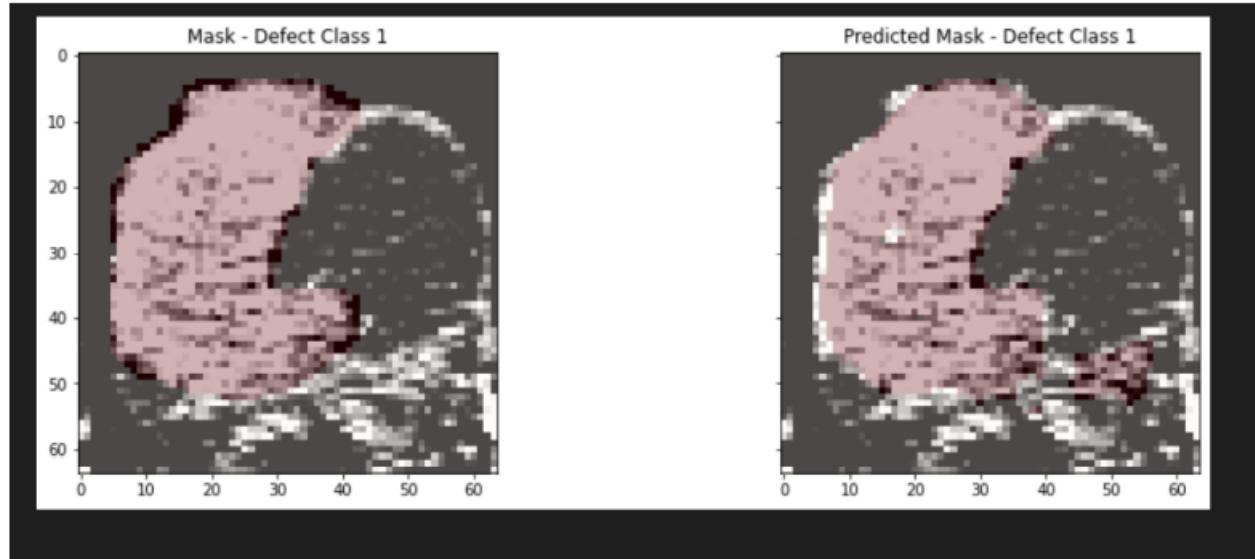
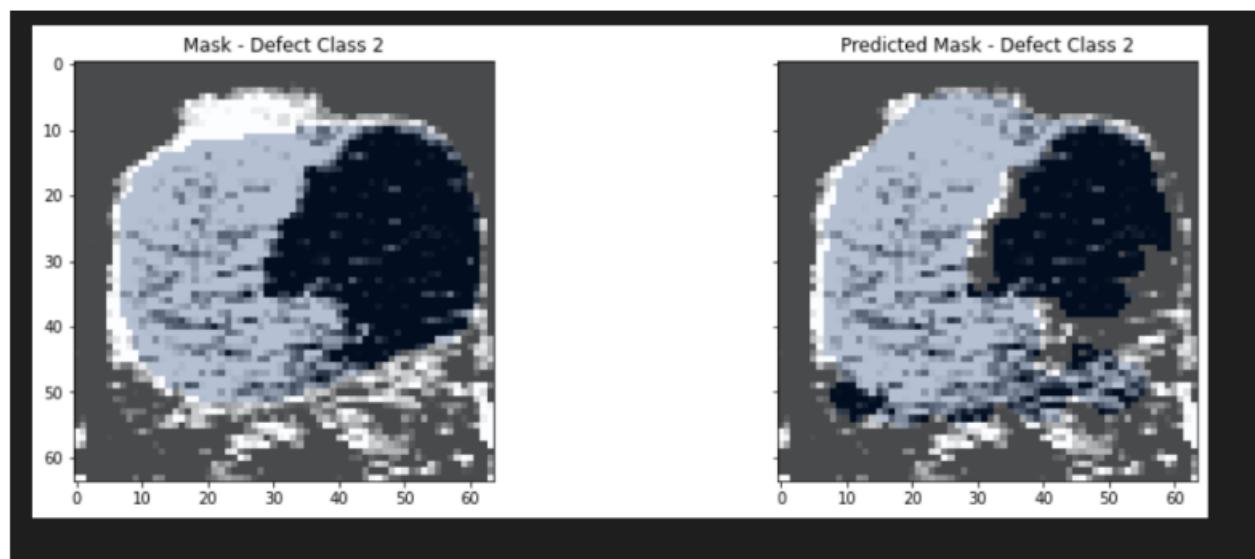


Figure 4.8: Mask Defect Class 2 a patra fază



De asemenea, date parțiale de măști pe baza datelor primite de la un centru medical pentru 2 pacienți.

Figure 4.9: Mask Defect și Predicted Mask Class 1 a prima fază (2 pacienți)

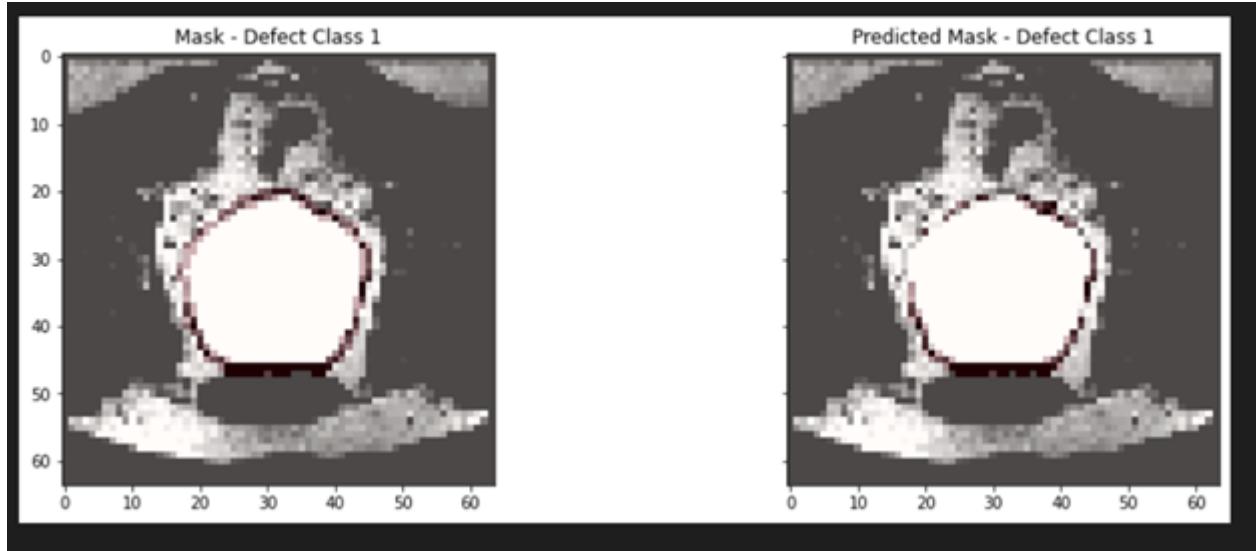


Figure 4.10: Mask Defect și Predicted Mask Class 1 a doua fază (2 pacienți)

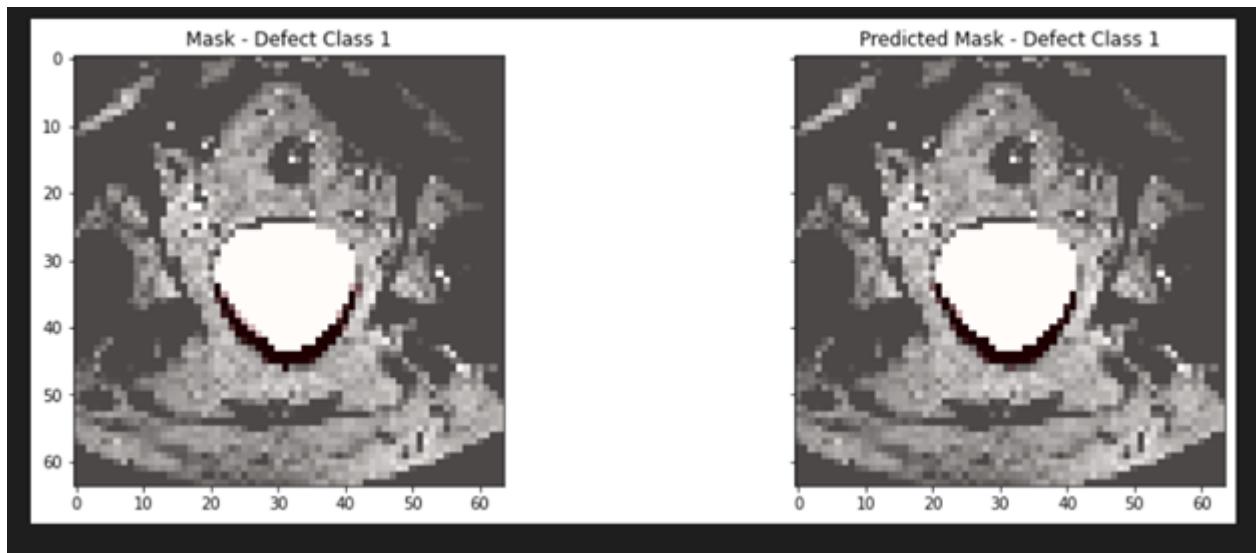
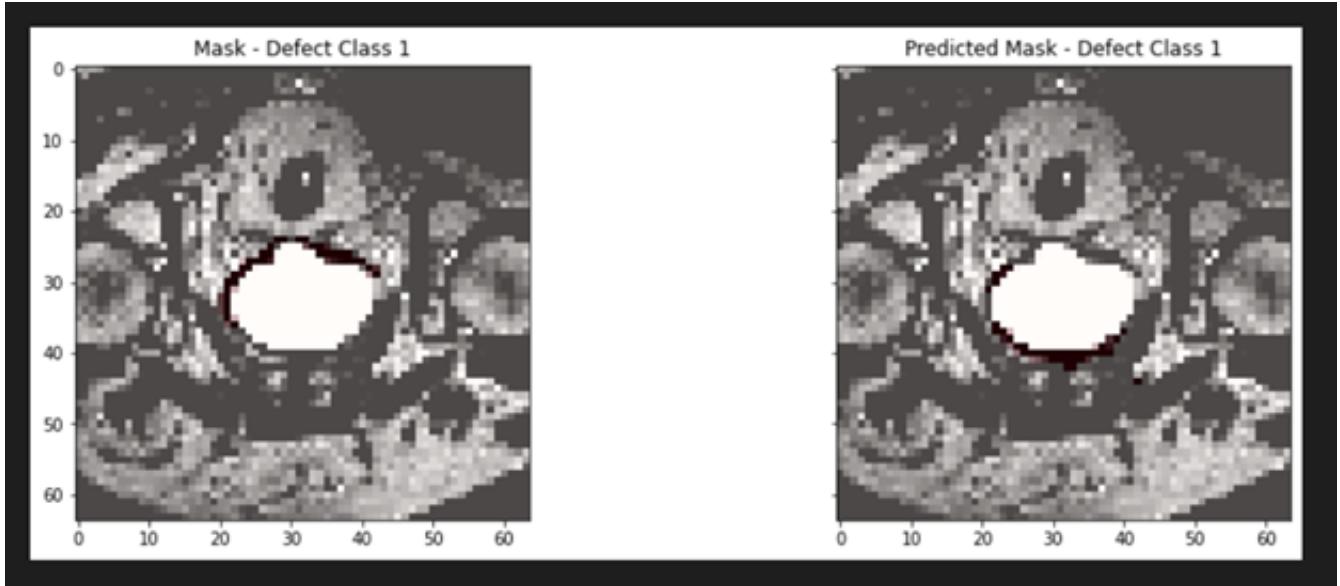


Figure 4.11: Mask Defect și Predicted Mask Class 1 a treia fază (2 pacienți)



Totodată, date parțiale de măști pe baza datelor primite de la un centru medical pentru 19 pacienți.

Figure 4.12: Mask Defect Mask Class 1 a prima fază (19 pacienți)

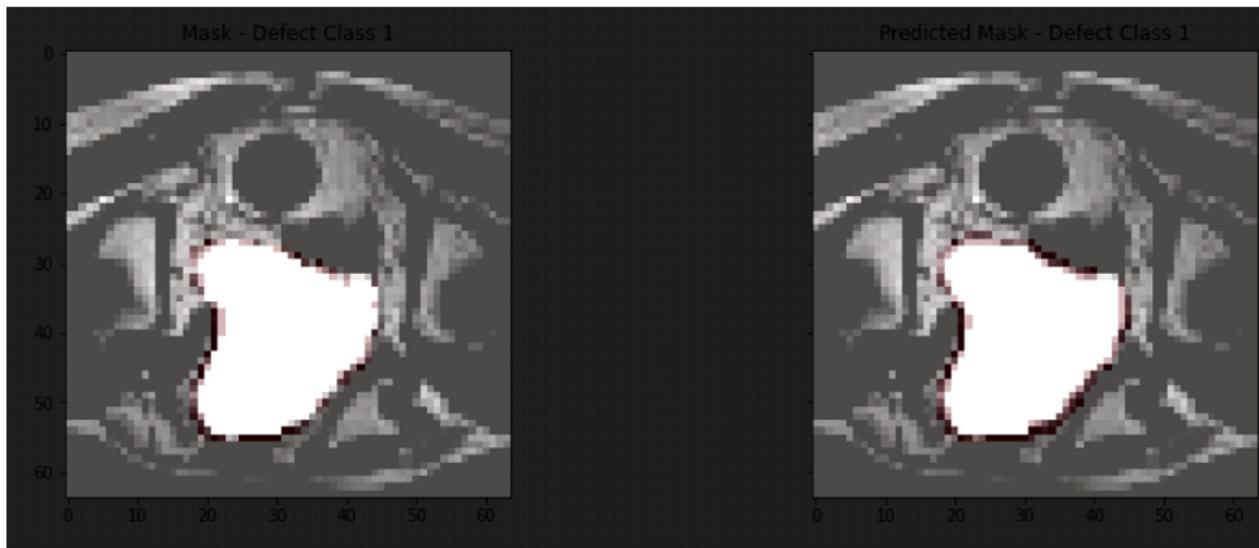


Figure 4.13: Mask Defect Class 1 a doua fază (19 pacienți)

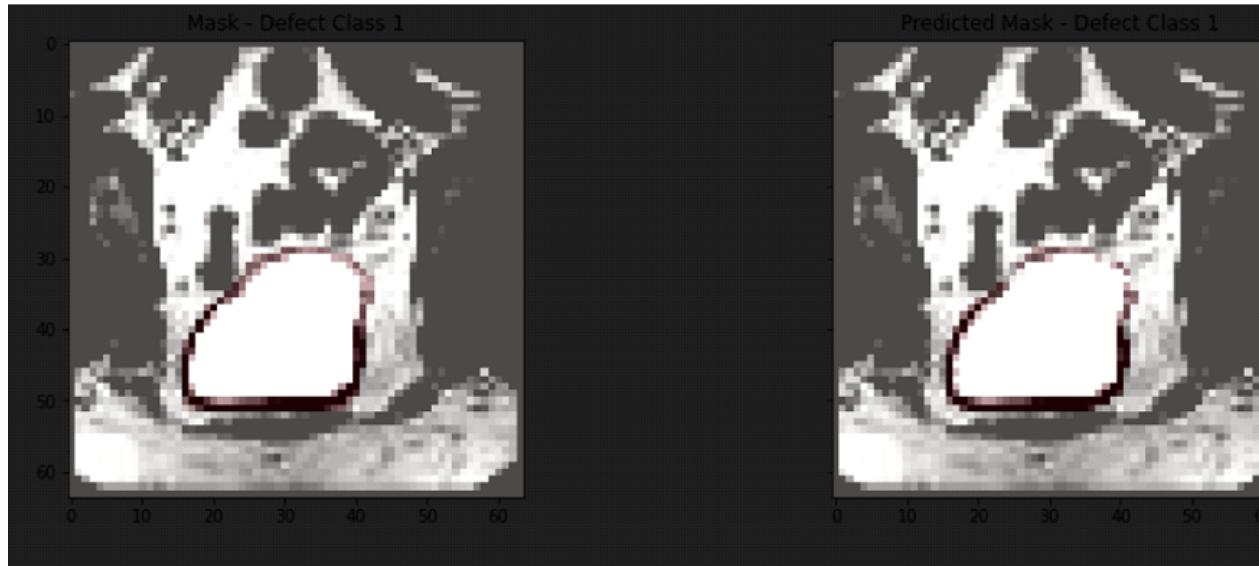


Figure 4.14: Mask Defect Mask Class 1 a treia fază (19 pacienți)

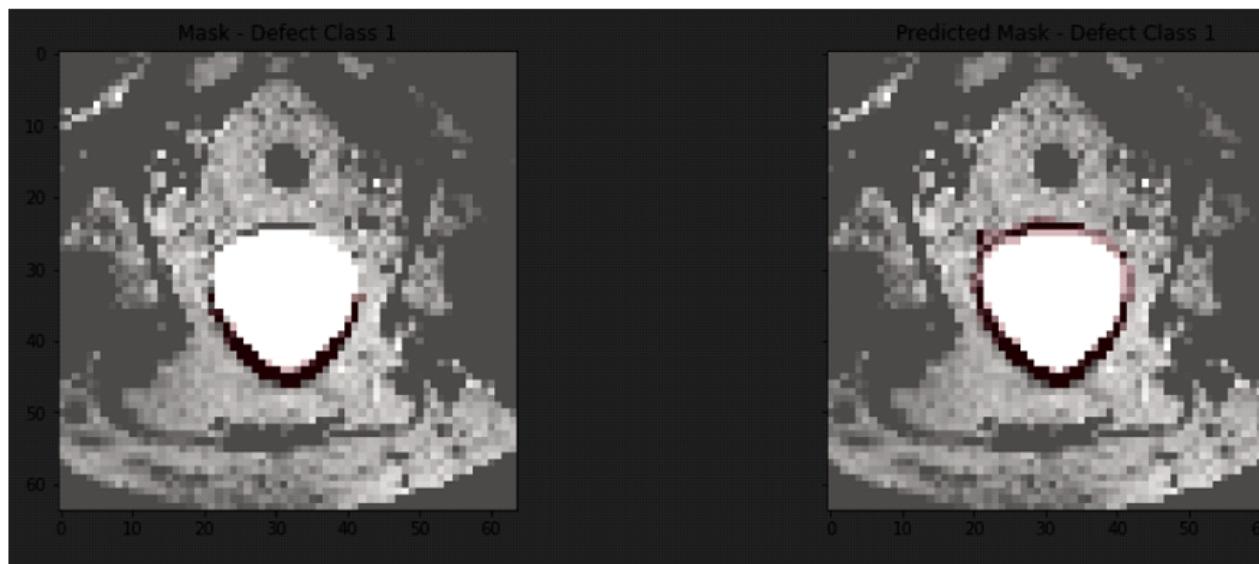


Figure 4.15: Mask Defect Mask Class 1 a patra fază (19 pacienți)

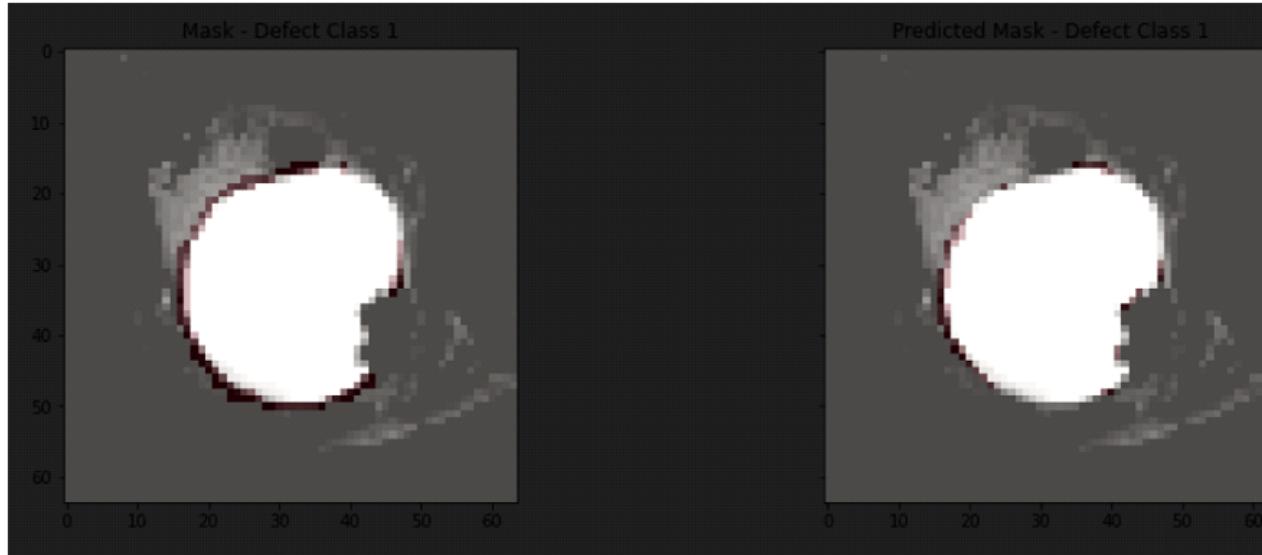


Figure 4.16: Mask Defect Class 1 a cincea fază (19 pacienți)

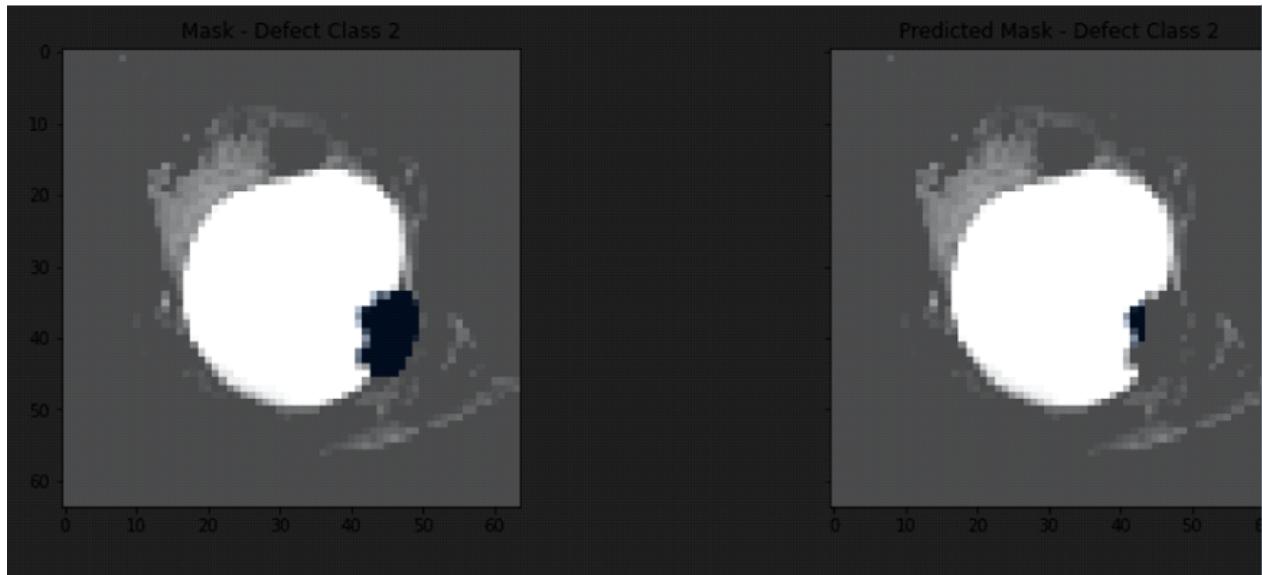


Figure 4.17: Mask Defect Mask Class 1 a şasea fază (19 pacienți)

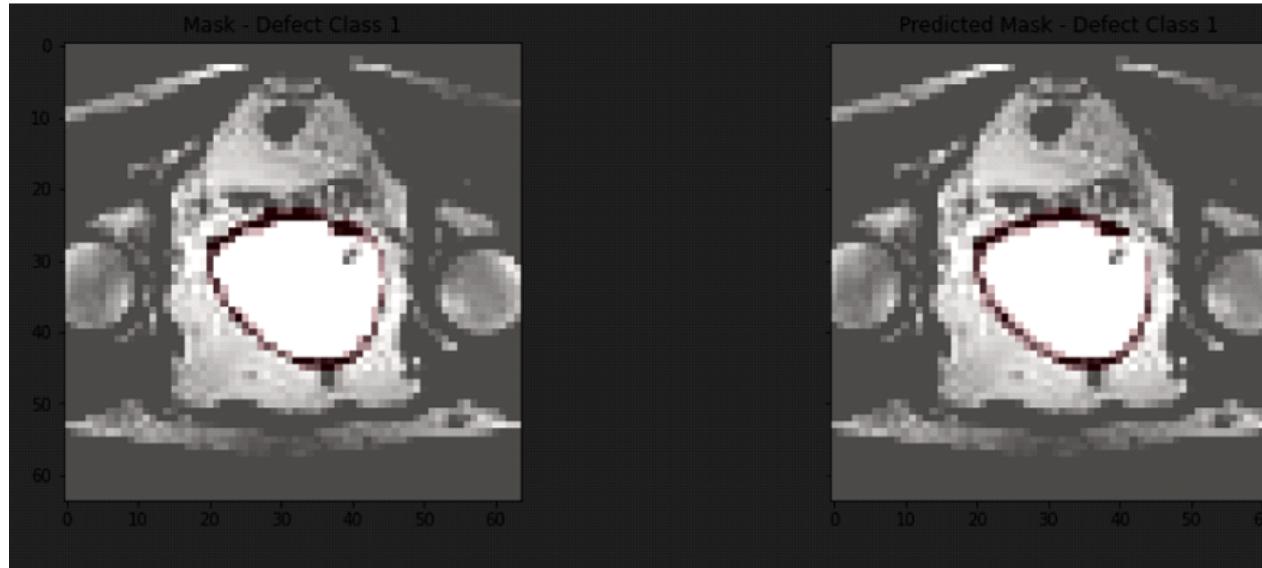


Figure 4.18: Mask Defect Mask Class 1 a şaptea fază (19 pacienți)

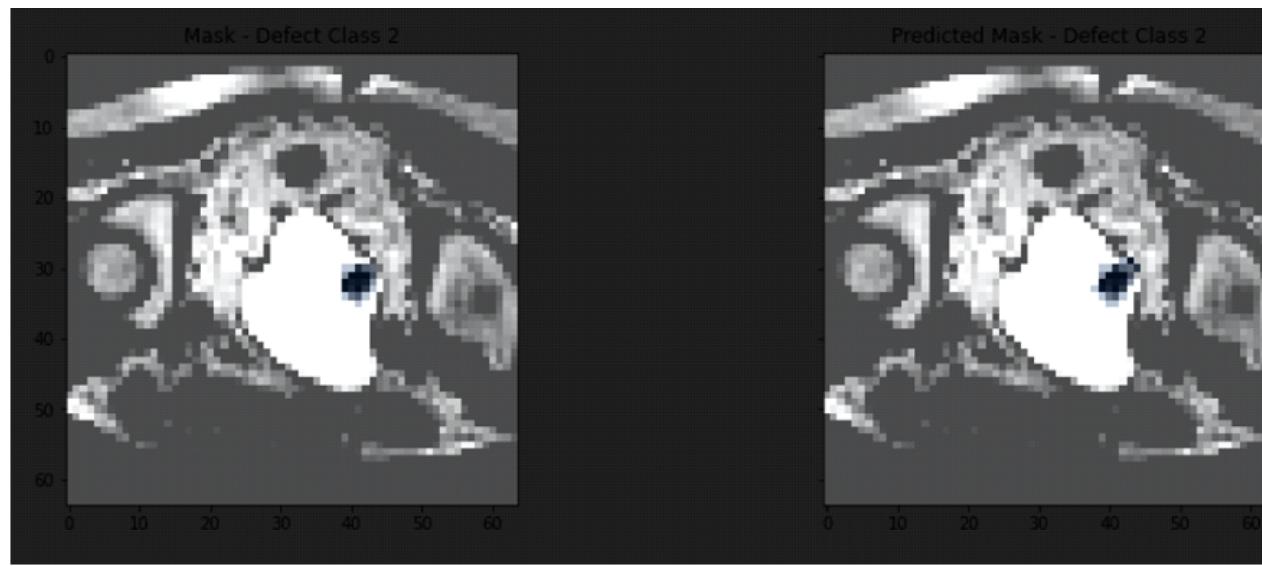


Figure 4.19: Mask Defect Class 1 a opta fază (19 pacienți)

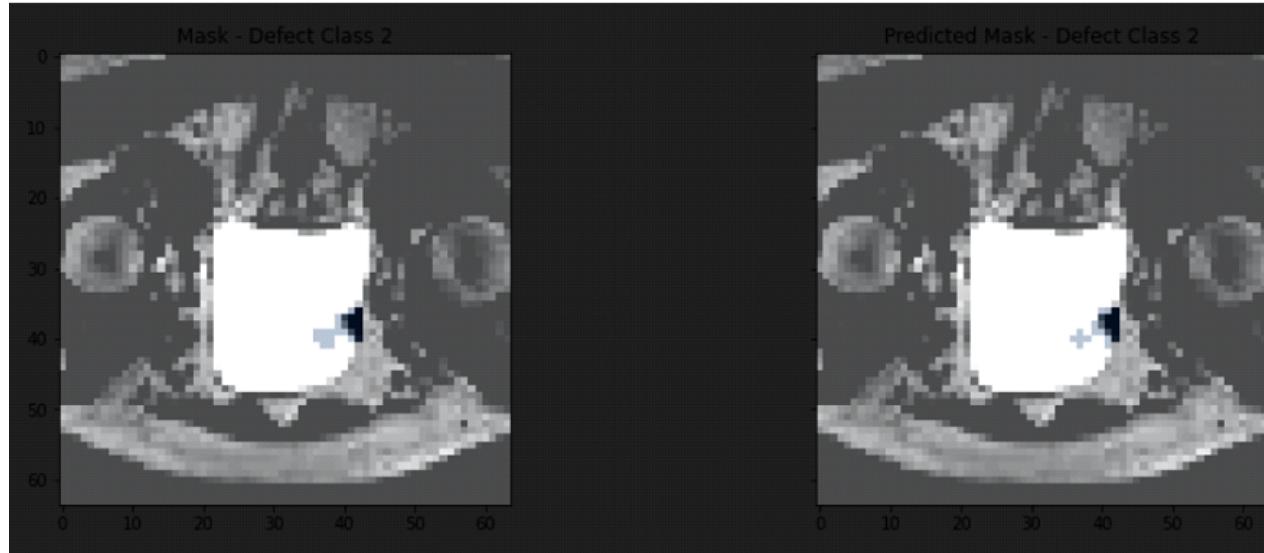
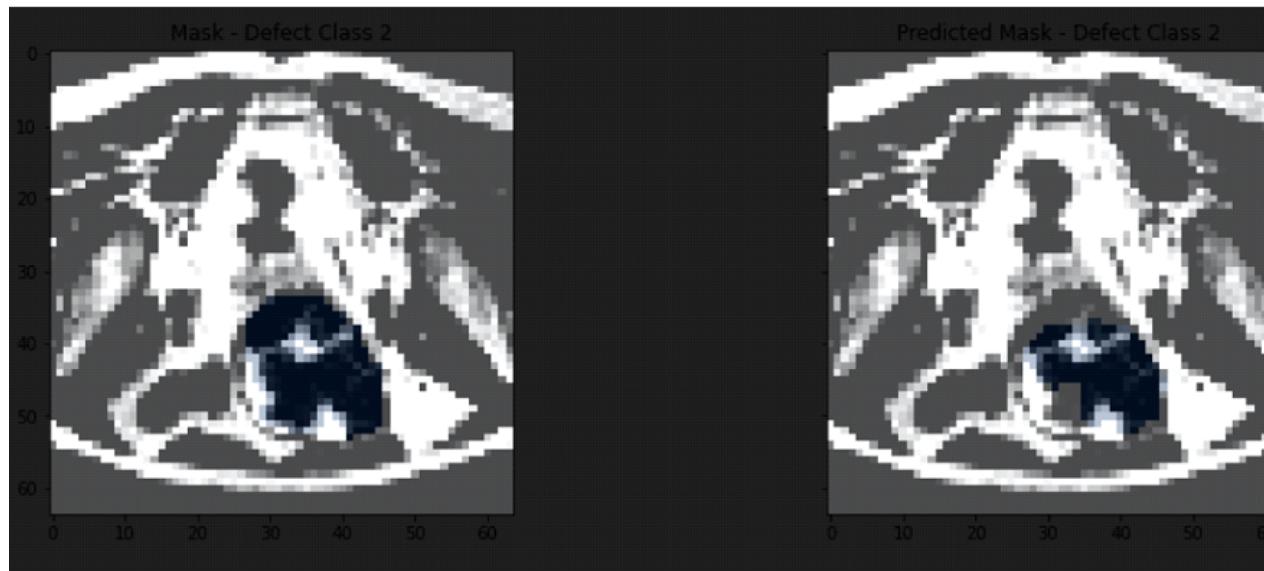


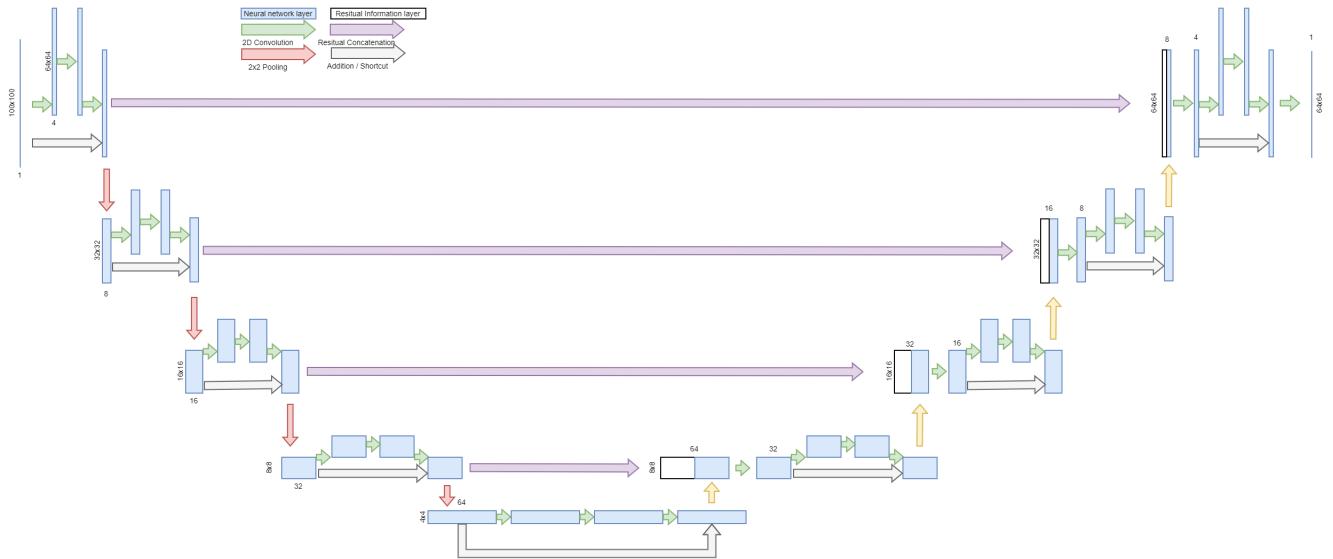
Figure 4.20: Mask Defect Mask Class 1 a noua fază (19 pacienți)



4.1 U-net

Structura de algoritm este de forma U-Net [Vezi U-Net](#) cu scurtături pe o structură de CNN (Convolutional Neural Network) deoarece oferă o fidelitate bună pentru masca finală chiar dacă este un nivel scăzut de noduri comparativ cu alte structuri la același nivel de training.

Figure 4.21: Convolutional Networks pentru segmentarea imaginilor biomedicală



Operația de conoluție se bazează pe luarea unei secțiuni de 3×3 din tensor centrată pe un anumit nod și aplicarea unei funcții pe zona respectivă, rezultatul urmând să fie valoarea nodului în stratul următor.

Pooling-ul este o operație a CNN care constă în luarea unei zone de 2×2 și alegerea valorii maxime din zona respectivă pentru a forma noul layer. Astfel se micșorează mărimea stratului pe 2 dimensiuni cu un factor de 50%.

Operatia de Shortcut sau de adăugare oferă o direcție mai scurtă pentru informația de intrare. Aceasta se realizează prin adăugarea celor două matrici ce reprezintă valorile nodurilor. Scopul operației este de a eficientiza antrenarea prin păstrarea fidelității datelor mai adânc în network.

Structura generală este formată dintr-un encoder care transformă imaginea dintr-un spațiu bidimensional mare pe 2 dimensiuni (imagine 100×100) într-un spațiu 3d mare pe o dimensiune ($4 \times 4 \times 64$) abstractizând informația imaginii prin straturi convoluționale și un decoder care formează operația inversă. Aceste două elemente sunt cuplate transversal pe straturile de aceeași mărime pentru a obține o fidelitate mai clară a mații prin concatenarea tensorilor din partea de encoder cu tensori care au trecut prin operația de upscale.

Mai concret, CNN-ul are 4 straturi de encoding, și anume un strat de bridge și patru straturi de decoder. Fiecare strat de encoding este format din 2 convoluții, o adăugare și o convoluție finală urmată de o operație de activare non liniară ReLU și o normalizare. Între straturile de Encoder este o operație de Pooling, iar între straturile de Decoder este o operație de Upscale. Imediat după operația de upscale se realizează o concatenare cu rezultatele stratului echivalent din encoder. Scopul concatenării este destul de similar cu cel al shortcut-ului, mai exact de a reduce detaliul imaginii originale în imaginea degradată generată de encoder.

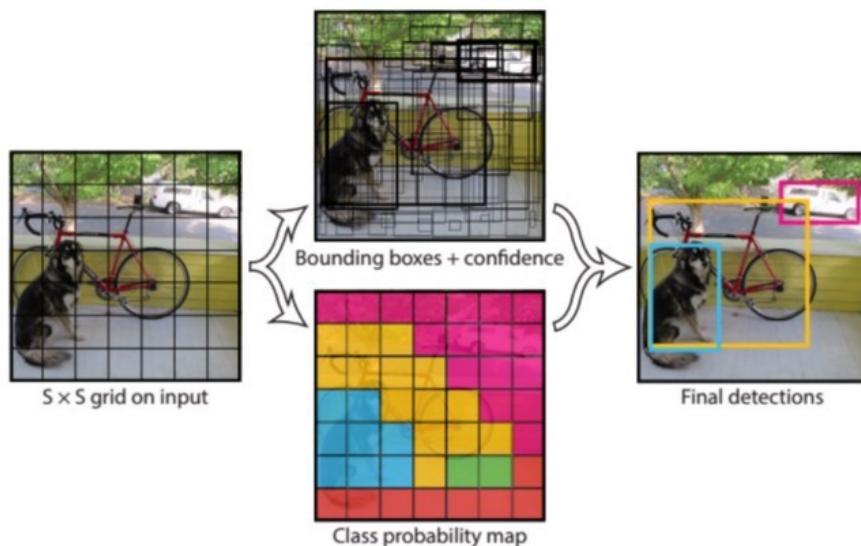
4.2 YOLO

YOLO este un algoritm care folosește rețele neuronale pentru a oferi detectarea obiectelor în timp real. YOLO este o abreviere pentru termenul "You Only Look Once".

Table 4.1: Parametrii modelului standard de antrenament Yolo

Parameter	Value
Learning Rate	0.001
Score Threshold	0.5
IOU Threshold	0.5
Input Image Size	416x416
Number of Classes	1

Figure 4.22: Algoritmul YOLO pentru detectarea obiectelor



Algorithm 1 Detectarea predicțiile obiectelor

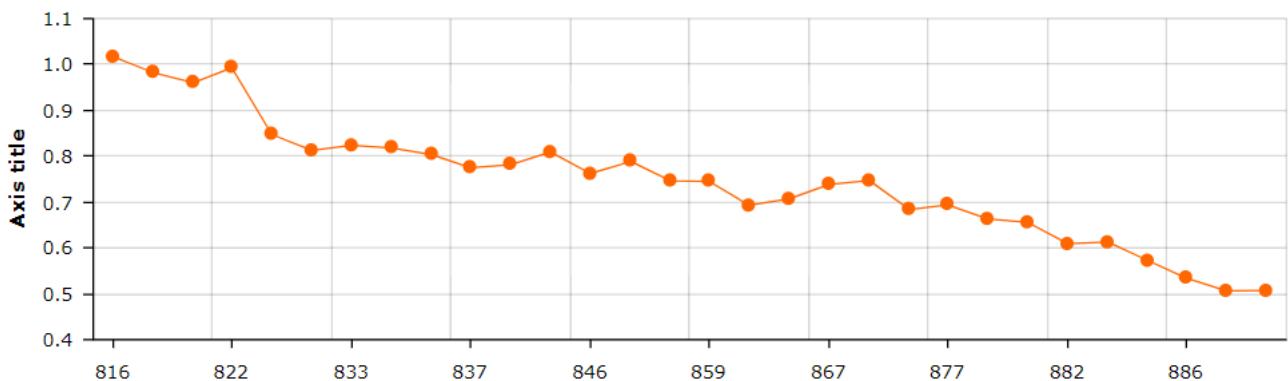
```

BEGIN
    @ Read and Transform Image
    @ Divide the image in a SxS grid
    @ Go through the darknet53 extractor to obtain features
    @ Send the features to the detection convolutional layers at multiple scales
    @ Detect bounding box
    @ Calculate various scores:
        boxConfidenceScore = Pr(object) * IoU
        conditionalClassProbability = Pr(class|object)
        classConfidenceScore = Pr(class) * IoU
    @ DetectBoxes(threshold, IoU)
    @ Apply NonMaxSupression
    if detection is over IoU and threshold then
        @ Return(detection);
    end if
END

```

În comparație cu alte rețele de clasificare a propunerilor de regiune (RCNN rapid) care efectuează detectarea pe diferite propuneri de regiune ajungând să efectueze predicții de mai multe ori pentru diferite regiuni dintr-o imagine, arhitectura Yolo seamănă mai mult cu FCNN (rețea neuronală complet conlovuțională). YOLO transmite imaginea ($n \times n$) prin FCNN, iar ieșirea este o predicție ($m \times m$). Această arhitectură împarte imaginea de intrare în grila $m \times m$ iar pentru fiecare generație de grilă sunt 2 casete de delimitare și probabilități de clasă pentru casetele respective.

Figure 4.23: Valoarea loss-ului pentru iteratiile de YOLO de la 800 la 900



Yolov4 este o îmbunătățire a algoritmului Yolov3, având o îmbunătățire a preciziei de medii (mAP) cu până la 10 la sută și a numărului de cadre pe secundă cu 12%.

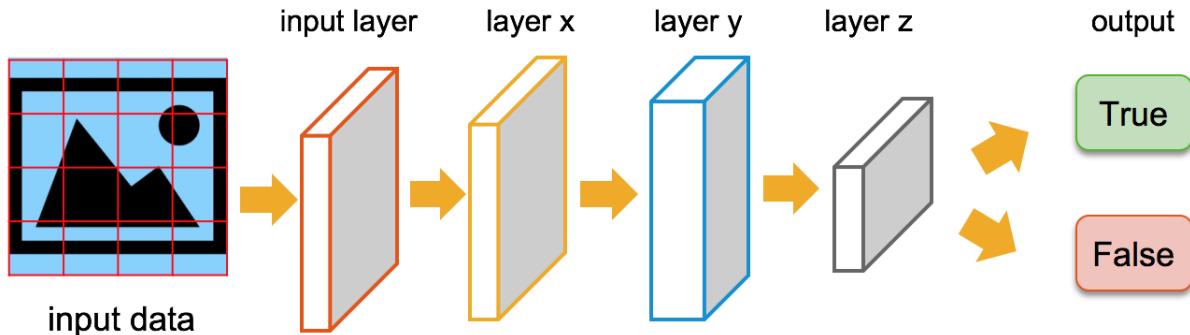
Compararea YOLO cu alți algoritmi.

SOTA(State of the art): vezi Figure 3.1

4.3 Tensorflow

Tensorflow este o bibliotecă de inteligență artificială open source, care utilizează grafice ale fluxului de date pentru a construi modele. Aceasta permite dezvoltatorilor să creeze rețele neuronale la scară largă cu multe straturi.

Figure 4.24: Clasificare pe straturi a imaginii cu Tensorflow



TensorFlow este folosit în principal pentru:

- Clasificare
- Percepție
- Înțelegere
- Descoperire
- Creare

Chapter 5

Aplicație (validare numerică)

5.1 Metodologie

- Care sunt criteriile utilizate pentru a evalua metoda?

Am folosit o structura de U-net bazată pe un encoder, Stem și Decoder. Fiecare din cele 4 nivele de encoding, respectiv decoding este format din 2 straturi convoluționale (Conv2D) și un shortcut intern care se adaugă la rezultatul acestora.

Downscaling-ul se realizează prin schimbarea stride-ului în cadrul ultimei convoluții de la 1 la 2, iar stem-ul este format din aceeași structură ca și Decoder-ul, cu excepția concatenării datelor reziduale de la nivelul corespunzător din Encoder. Structura nivelelor acestora este de 2 convoluții, un shortcut intern care se adaugă la rezultatul acestora, respectiv un layer de Upsampling.

Pentru a menține integritatea informației, fiecare layer trece printr-un strat de normalizare și pentru a crește fidelitatea imaginii în fiecare nivel de encoder, respectiv fiecare nivel de decoder, astfel se realizează o activare non liniară RELU.

5.2 Date

Pe parcursul proiectului, datele de antrenament s-au îmbunătățit, optimizând astfel predicția și acuratețea algoritmului.

Pentru început am utilizat bucăți de date de la diferiți pacienți la care am creat aproximativ 50 de măști de mâină.

Mai apoi, ne-am axat pe date concrete primite de la un centru medical, mai exact date de antrenament de la 19 pacienți și 446 de măști.

5.3 Rezultate

Pentru început, primele rezultate din primul stadiu de lucru pentru loss și accuracy.

Pentru loss avem:

Figure 5.1: Rezultate algoritm de segmentare pentru loss cu 250 de epoci

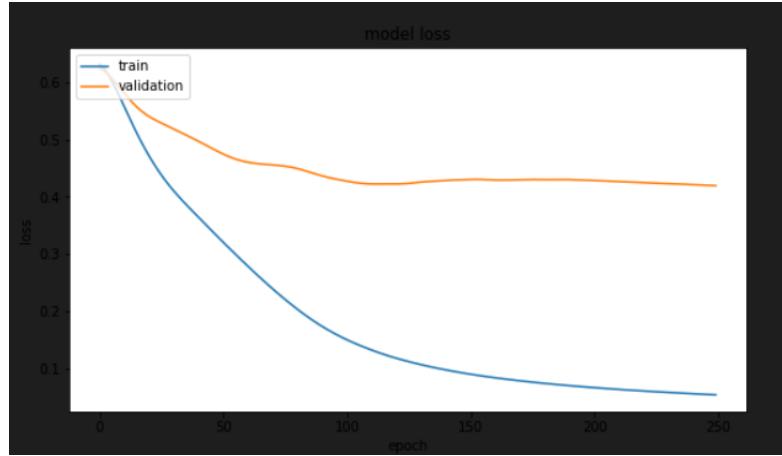
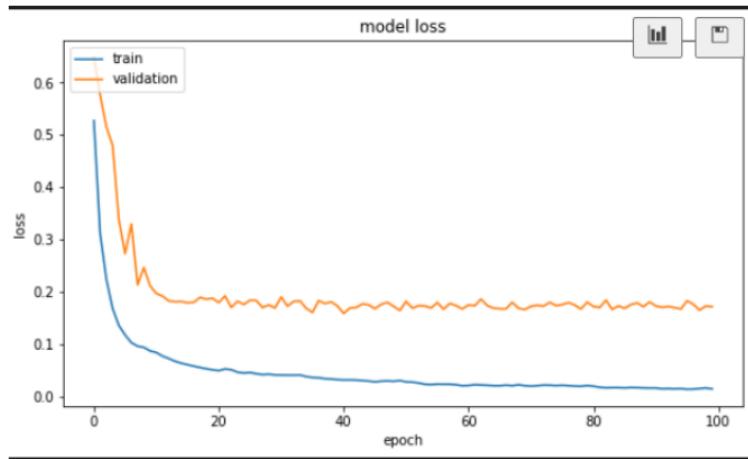


Figure 5.2: Rezultate algoritm de segmentare pentru model loss cu 100 de epoci după aprox. 0.01



Îmbunătățire loss

Figure 5.3: Îmbunătățire algoritm de segmentare pentru loss

```
33/33 [=====] - 2s 54ms/step - loss: 0.0143 - tversky: 0.9964 - val_loss: 0.1/16 - val_tversky: 0.8998
```

Pentru accuracy avem:

Figure 5.4: Rezultate algoritm de segmentare pentru accuracy cu 250 de epoci

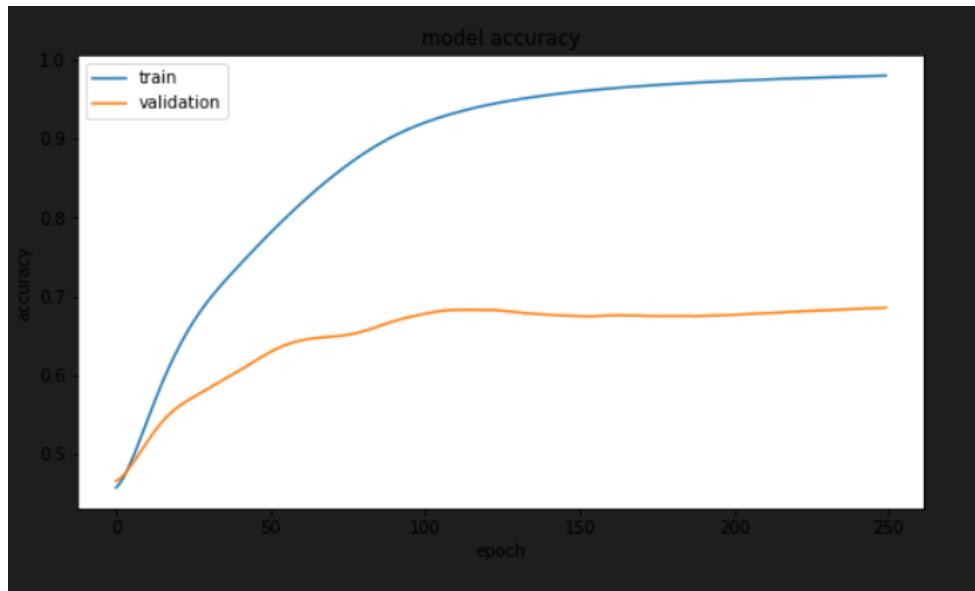
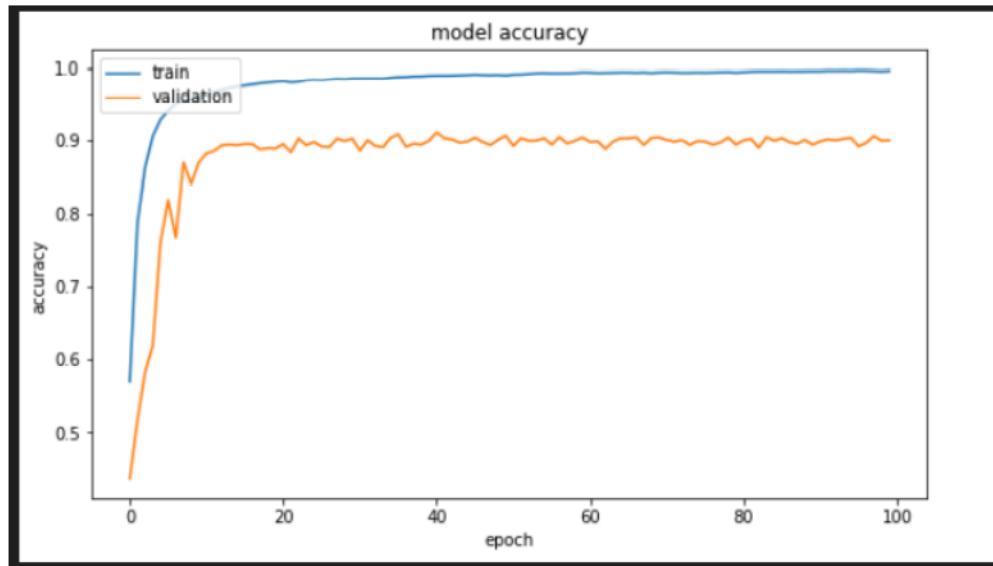


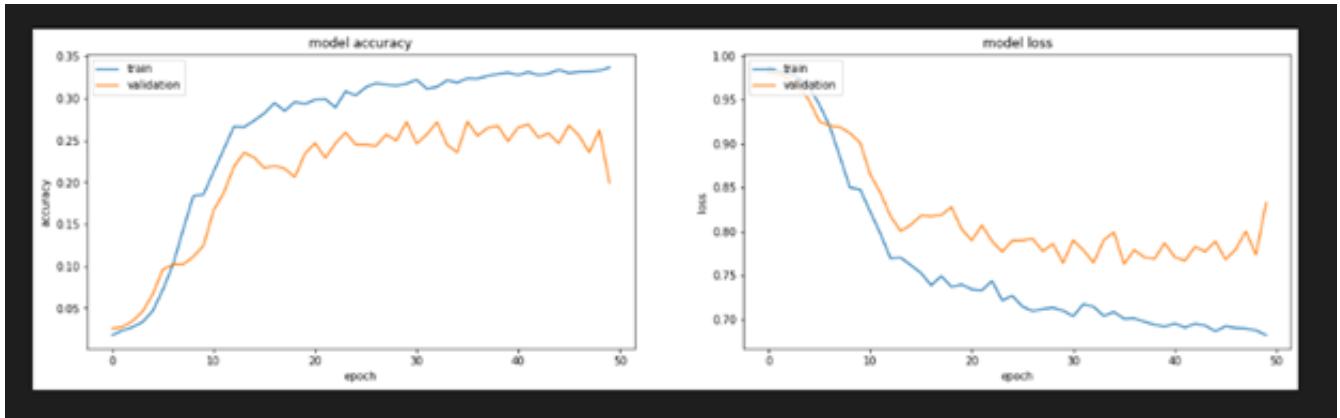
Figure 5.5: Rezultate algoritm de segmentare pentru model accuracy cu 100 de epoci după aprox. 0.01



În continuare, rezultatele pentru loss și accuracy din faza a două a cercetării pe baza setului de date de la centrul medical.

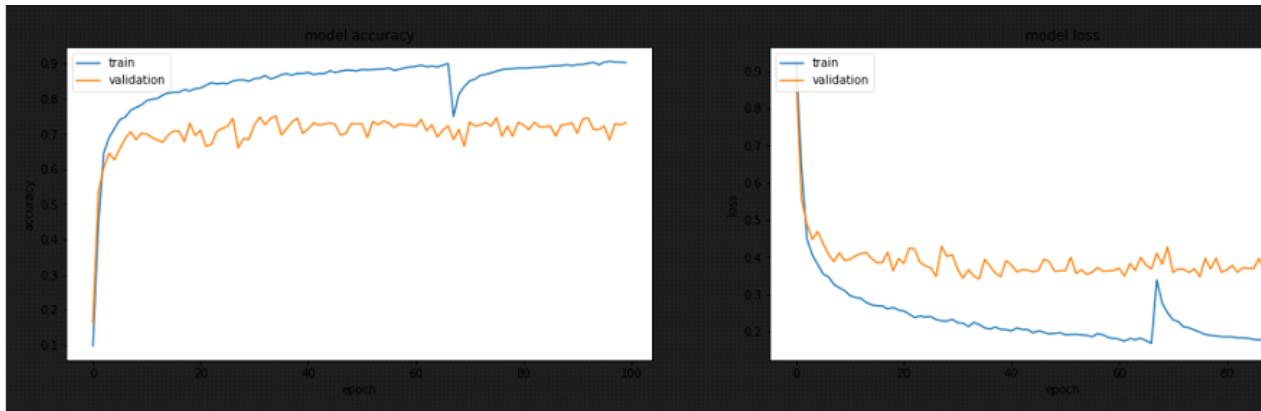
Pentru 2 pacienți avem următoarele rezultate:

Figure 5.6: Rezultate algoritm de segmentare pentru model accuracy și model loss cu 50 de epoci pentru 2 pacienți



Pentru 19 pacienți avem următoarele rezultate:

Figure 5.7: Rezultate algoritm de segmentare pentru model accuracy și model loss cu 100 de epoci pentru 19 pacienți



Mai jos sunt prezentate două modele de antrenament pe baza setului de date de la centrul medical. Cele două modele au fost testate pentru 100 de epoci, cu mențiunea că în tabel nu au fost trecute și epocile care aveau aceeași valoare de loss.

Modelul 1

Numărul de date de antrenament: 19 pacienți

Numărul de teste: 100

Numărul de măți: 446 (118 tumoare, 328 perete)

Rezultate validare: 35% date de validare (40 tumoare, 114 perete)

Rezultate antrenament: 65% date de antrenament(78 tumoare, 214 perete)

Table 5.1: Primul model de antrenament

Epochs	Loss
1	0.92
2	0.63
3	0.44
4	0.40
5	0.38
6	0.35
7	0.34
8	0.32
9	0.31
11	0.29
13	0.28
14	0.27
16	0.26
20	0.25
22	0.24
25	0.23
28	0.22
33	0.21
37	0.20
45	0.19
55	0.18
61	0.17
67	0.16
100	0.15

Modelul 2

Numărul de date de antrenament: 19 pacienți

Numărul de teste: 100

Numărul de măți: 446 (118 tumoare, 328 perete)

Rezultate validare: 50% date de validare (59 tumoare, 164 perete)

Rezultate antrenament: 50% date de antrenament (59 tumoare, 164 perete)

Table 5.2: Al doilea model de antrenament

Epochs	Loss
1	0.91
2	0.69
3	0.48
4	0.44
5	0.39
6	0.36
7	0.34
8	0.33
9	0.32
10	0.31
11	0.30
12	0.29
14	0.27
15	0.26
17	0.25
21	0.24
25	0.23
26	0.22
29	0.21
35	0.20
43	0.19
46	0.19
58	0.17
73	0.16
76	0.15
100	0.14

Table 5.3: Parametrii modelului de antrenament

Parameter	Value
Loos: focal_tversky_loss	gamma = 0.75
Matrics: tversky	alpha = 0.7, smooth = 1e-6
Optimizer: tf.keras.optimizers.Adam	lr = 0.05, epsilon = 0.2

Chapter 6

Concluzii și lucrări viitoare

Lucrarea noastră a încercat să ofere un punct de vedere clar și ușor din perspectiva detectării tumorilor maligne a vezicii urinare pentru a facilita procesul de învățare și a crește productivitatea acestuia pentru studenții de la medicină și nu numai.

Puterea rezultatelor noastre constă în utilizarea algoritmului YOLO, care este un detector puternic recunoscut pentru obiecte. Am folosit ultima sa versiune pentru a avea cele mai recente îmbunătățiri atunci când vorbim despre detectarea posibilelor malformații ale vezicii urinare și a peretelui.

În final, am obținut o precizie și o acuratețe mare de 0.788 datorită utilizării de scurtături în rețea și un timp de răspuns mic ca urmare a folosirii unui algoritm eficient de ROI.

- Punctele tari:

- acuratețea detectării este mare datorită utilizării de scurtături în rețea
- timpul de răspuns este mic datorită utilizării unui algoritm eficient de ROI
- interacțiunea cu aplicația este ușoară datorită folosirii unei platforme compatibilă cu telefoane mobile
- designul este foarte accesibil și ușor de utilizat

- Punctele slabe:

- lipsa datelor
- necesitatea unei conexiuni la internet

- Oportunități:

- creșterea datelor de antrenament poate conduce la creșterea predicției
- îmbunătățirea calității procesului de învățare

- creșterea productivității de învățare
- oricine poate utiliza aplicația

- Amenințări:

- existența altor aplicații de același gen

Pentru viitor ne propunem să creștem numărul de date de antrenament pentru a crește acuratețea algoritmului și predicția aplicației. Totodată, dorim să analizăm și cazurile de pietre urinare și cistita emfizematoasă.

Bibliography

- [1] Sungmin Woo, Valeria Panebianco, Yoshifumi Narumi, Francesco Del Giudice, Valdair F. Muglia, Mitsuru Takeuchi, Soleen Ghafoor, Bernard H. Bochner, Alvin C. Goh, Hedvig Hricak, James W.F. Catto and Hebert Alberto Vargas, Diagnostic Performance of Vesical Imaging Reporting and Data System for the Prediction of Muscle-invasive Bladder Cancer: A Systematic Review and Meta-analysis, 2020
- [2] Valeria Panebianco, Yoshifumi Narumi, Ersan Altun,c Bernard H. Bochner, Jason A. Efstathiou, Shaista Hafeez, Robert Huddart, Steve Kennish, Seth Lerner, Rodolfo Montironi, Valdair F. Muglia, Georg Salomon, Stephen Thomas, Hebert Alberto Vargas, J. Alfred Witjes, Mitsuru Takeuchi, Jelle Barentsz and James W.F. Cattor, Multiparametric Magnetic Resonance Imaging for Bladder Cancer: Development of VI-RADS, 2018
- [3] Nader Aldoj, Federico Biavati, Florian Michallek, Sebastian Stober and Marc Dewey, Automatic prostate and prostate zones segmentation of magnetic resonance images using DenseNet-like U-net, 2020
- [4] Samuel J Galgano, Kristin K Porter, Constantine Burgan and Soroush Rais-Bahrami, The Role of Imaging in Bladder Cancer Diagnosis and Staging, 2020
- [5] See Hyung Kim, Validation of vesical imaging reporting and data system for assessing muscle invasion in bladder tumor, 2020
- [6] Ravi K. Kaza , Joel F. Platt, Richard H. Cohan, Elaine M. Caoili, Mahmoud M. Al-Hawary and Ashish Wasnik, Dual-Energy CT with Single- and Dual-Source Scanners: Current Applications in Evaluating the Genitourinary Tract, 2012
- [7] Anushri Parakh, Hyunkwang Lee, Jeong Hyun Lee, Brian H. Eisner, Dushyant V. Sahani and Synho Do, Urinary Stone Detection on CT Images Using Deep Convolutional Neural Networks: Evaluation of Model Performance and Generalization, 2019

- [8] Ekhtiar Syed, ResUNet-a Baseline on TensorFlow, 2019