

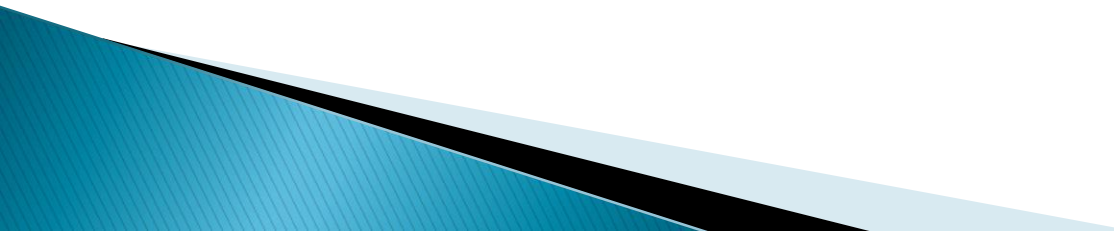
Metode de vedere artificială și învățare automată pentru clasificarea obiectelor în industria porțelanului

Realizat de: Copîndean Alin Gabriel, Golgoț Mihai Radu

Profesor coordonator: Conf.univ.dr. Bîrluțiu Adriana

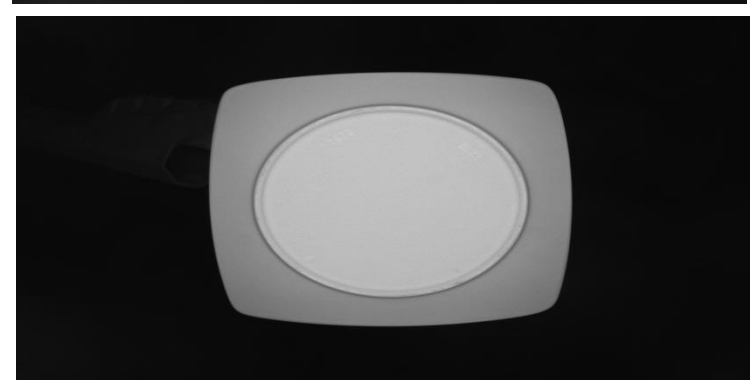
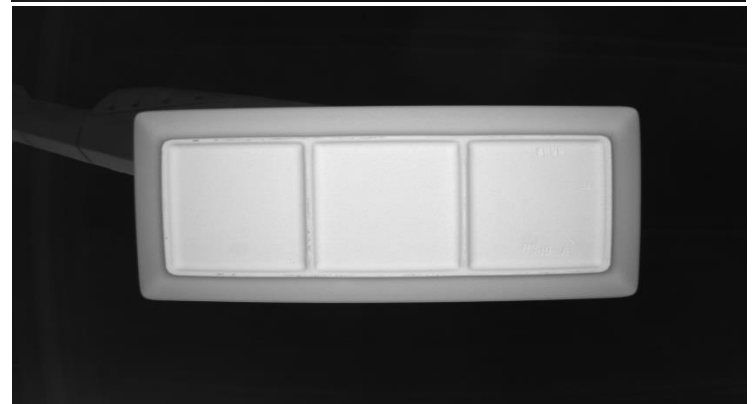


Motivația și obiectivul realizării proiectului

- Cercetare și învățare
 - Automatizarea proceselor în industria porțelanului
 - Reducerea timpului de producție
 - Reducerea costurilor salariale
- 

Setul de date folosit

- Companie din industria portelanului
- 141 de imagini, care sunt împărțite în 3 categorii:
 - farfurii rotunde,
 - farfurii dreptunghiulare,
 - farfurii pătrate



Tehnologii folosite

- Limbajul de programare **Python**
- Mediul de programare folosit : **Spyder**
- Bibliotecile folosite: **OpenCV, Numpy, Tensorflow, Glob**



Abordări pentru rezolvarea problemei

- ▶ **Abordarea axată pe tehnici de vedere artificială**
- ▶ **Abordarea axată pe tehnici de învățare automată**

Abordări pentru rezolvarea problemei

- ▶ **Abordarea axată pe tehnici de vedere artificială**
- ▶ Abordarea axată pe tehnici de învățare automată

Abordarea axată pe tehnici de vedere artificială

1. Citirea imaginilor – *imread* din openCV

cv2.imread(filename)

Abordarea axată pe tehnici de vedere artificială

1. Citirea imaginilor – *imread* din openCV
2. Delimitarea conturului – *findContours*, *drawContours*



Abordarea axată pe tehnici de vedere artificială

1. Citirea imaginilor -imread din openCV
2. Delimitarea conturului
3. Calcularea dimensiunii conturului - un parametru al funcției *findContours*.

*cv2.findContours(binary,cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)*

Abordarea axată pe tehnici de vedere artificială

1. Citirea imaginilor – imread din openCV
2. Delimitarea conturului
3. Calcularea dimensiunii conturului – un parametru al funcției *findContours*
4. Clasificarea imaginii farfuriei într-una dintre categorii folosind următoarea instrucțiune:

```
if len(c)>1000:
```

```
    if len(c)>=3000:
```

```
        print("Farfuria din imaginea %s are forma rotunda"%
```

```
(filename))
```

```
elif len(c)<=1450 and len(c)>=min_drept-13:
```

```
    print("Farfuria din imaginea %s are forma dreptunghiulara"%
```

```
(filename))
```

```
elif len(c)<=2999 and len(c)>=1451:
```

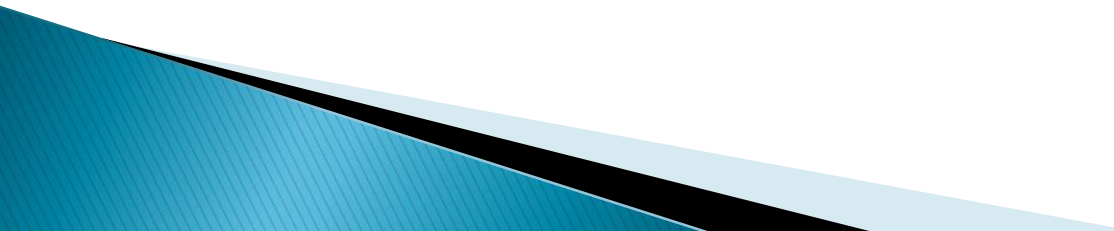
```
    print("Farfuria din imaginea %s are forma hexagonala"%
```

```
(filename))
```

```
else:
```

```
    print("Farfuria din imaginea %s NU are nici o forma dintre
```

```
cele invatate."% (filename))
```



Rezultatul Metodei 1

La executarea aplicației conturul farfuriei este marcat.

De asemenea, ne este returnată și clasa din care face parte farfuria din imaginea respectivă.



```
Farfuria din imaginea Test_Final\cerc1 (43).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (44).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (45).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (46).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (47).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (48).jpg are forma rotunda
Farfuria din imaginea Test_Final\cerc1 (49).jpg are forma rotunda
Farfuria din imaginea Test_Final\dreptunghi (21).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (22).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (24).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (25).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (26).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (27).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (28).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (29).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (30).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (32).jpg are forma dreptunghiulara
Farfuria din imaginea Test_Final\dreptunghi (33).jpg are forma dreptunghiulara
```

Abordări pentru rezolvarea problemei

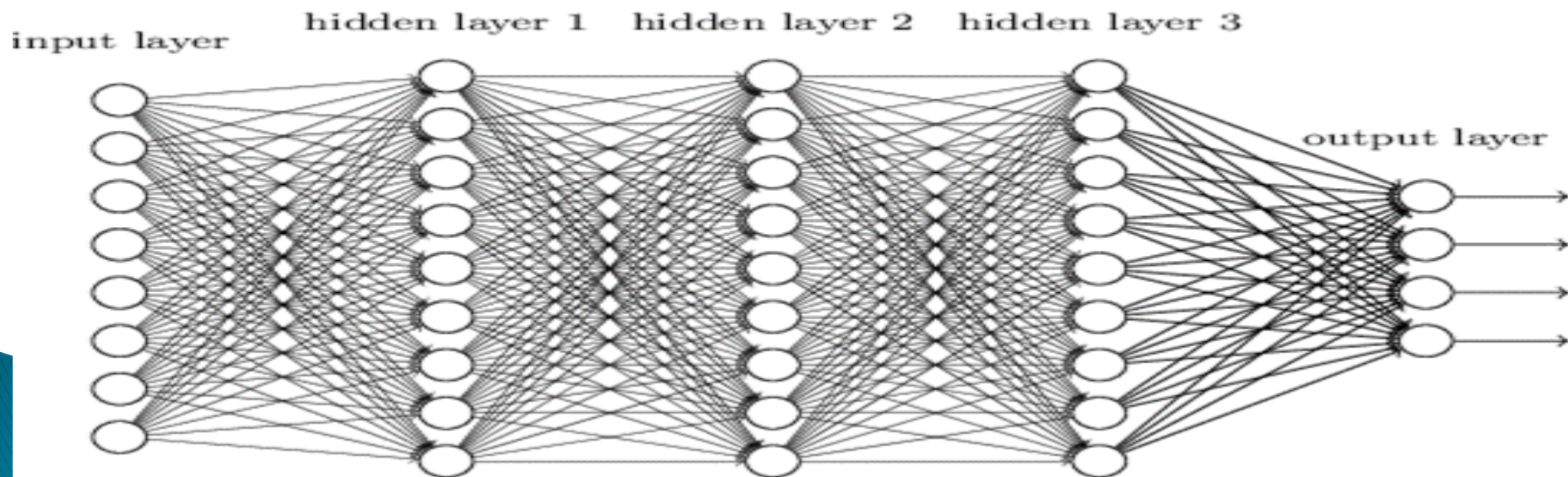
- ▶ Abordarea axată pe tehnici de vedere artificială
- ▶ **Abordarea axată pe tehnici de învățare automată**

Abordarea axată pe tehnici de învățare automată

1. Seturi disjuncte de antrenare si testare
 - Setul de date de antrenare:
 - 63 de imagini
 - 21 din fiecare clasă
2. Citirea imaginilor-imread din OpenCV, gray.scale
3. Redimensionăm imaginea 120X120 sub forma unei matrici – cv2.resize
4. Învățarea modelului de clasificare

Modelul de clasificare

- ▶ O rețea neuronală adâncă este o tehnologie care folosește mai multe tipuri de straturi care comunică între ele, fiecare cu precedentul strat. Avem un strat de intrare, straturile ascunse și un strat de ieșire.



Modelul de clasificare

- ▶ Rețea neuronală adâncă
- ▶ Biblioteca tensorflow este folosită pentru crearea modelului de clasificare, folosind funcția *tf.learn.dnn()*. Această funcție crează o rețea neuronală artificială cu mai multe straturi.

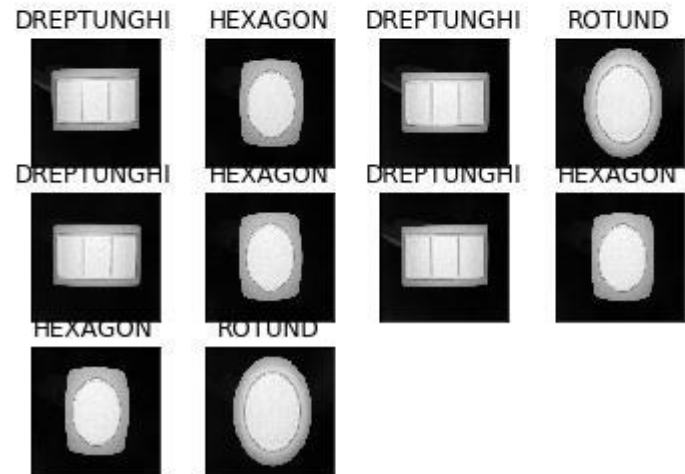


TensorFlow

Rezultatul aplicației

- Rezultatul este afișat cu ajutorul funcției *model.predict()* sub forma unei matrici cu 3 coloane
- Fiecare coloană reprezintă tipurile de farfurii
- Valorile de pe linii conțin probabilitățile de a face parte din una dintre cele 3 categorii

```
[0.03621624 0.9324939 0.03128983]  
[0.01007884 0.01735226 0.97256887]  
[0.03202585 0.9389889 0.02898526]  
[0.91955215 0.02842229 0.05202558]  
[0.03876398 0.9252014 0.03603461]  
[0.01833073 0.02071527 0.96095407]  
[0.03203131 0.9388454 0.0291232 ]  
[0.0126402 0.01846875 0.968891 ]  
[0.01097048 0.01706885 0.9719607 ]  
[0.9311663 0.02716203 0.04167164]
```



Concluzii și direcții viitoare de dezvoltare

- ▶ Înlocuirea forțelor de lucru umane cu mașinile
- ▶ Scurtarea timpului de producție
- ▶ Extinderea aplicației prin:
 - clasificarea altor tipuri de obiecte
 - Investigarea alor algoritmi de învățare automată și tehnici de vedere artificială

Bibliografie

- ▶ [1] Brownlee, J. (2017). Deep Learning With Python.
- ▶ [2] Gareth James, D. W. (2013). An Introduction to Statistical Learning.
- ▶ [3] Kamber, J. H. (2006). Data mining-Concepts and Techniques.
- ▶ [4] Mordvintsev, A. (2017). OpenCV-Python Tutorials Documentation.
- ▶ [5] Onița Daniela, G. I. (2017). Detectia defectelor în industria porțelanului folosind rețele neuronale adânci. Sesiunea de comunicări a studenților.
- ▶ [6] www.tensorflow.org.

Vă mulțumim!

