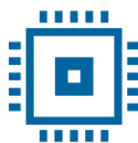




Universitatea  
Transilvania  
din Brașov



Universitatea  
Transilvania  
din Brașov  
FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR

Departamentul Automatică și Tehnologia Informației

Programul de studii: Sisteme Avansate în Automatică și Tehnologii Informatic

# Proiect Deep Learning Clasificarea nivelului de lichid din sticla

Conducător științific:

Dr. Ing. Anamaria VIZITIU

Student:

Ing. Bogdan Andrei PATRANIA

Brașov 2024

## Cuprins

1.Tema propusă .....	3
2.Baza de date.....	4
3.Biblioteca.....	5
4.Arhitectura.....	6
5.Rezultate experimentale .....	7
6.Compararea performanțelor .....	10
7.Bibliografie .....	13

## 1.Tema propusă

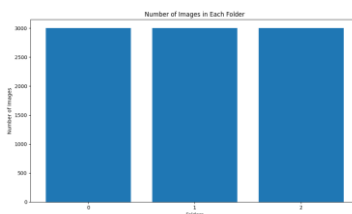
Tema propusă spre rezolvare constă în clasificarea nivelului de lichid din sticlă folosind trei tipuri de ipostaze de nivel de lichid. Utilizând aceste trei tipuri de nivel, se urmărește antrenarea unor modele, în urma cărora să se obțină un model capabil să distingă între aceste trei niveluri de lichid. Sarcina pe care modelul trebuie să o realizeze este cea de clasificare.

## 2.Baza de date

Baza de date folosită în cadrul acestui proiect a fost creată de mine și conține cele trei ipostaze pentru nivelul de lichid din sticlă: sub 50%, în jur de 50% și peste 50%. Baza de date cuprinde un total de 156 de imagini, distribuite astfel: 44 de imagini pentru clasa "în jur de 50%", 50 de imagini pentru clasa "peste 50%" și 62 de imagini pentru clasa "sub 50%". Imaginile sunt organizate în fișiere separate, fiecare având asociată o adnotare a clasei din care fac parte, iar detaliile claselor se găsesc în fișierul labels.csv, care specifică faptul că: clasa 0 corespunde sticlelor cu un nivel de lichid sub 50%, clasa 1 corespunde sticlelor cu un nivel de lichid în jur de 50%, iar clasa 2 corespunde sticlelor cu un nivel de lichid peste 50%.

Setul de date a fost împărțit astfel: 60% din imagini pentru antrenare, 20% pentru validare și 20% pentru testare. Această împărțire a fost realizată pentru a asigura un echilibru între antrenarea modelului și evaluarea performanței acestuia.

Imaginile din setul de date au fost augmentate pentru a obține un număr mai mare de imagini, adăugând 3000 de imagini suplimentare pentru fiecare clasă prin diverse operațiuni de augmentare: rotire, răsturnare pe orizontală, contrast aleatoriu, luminozitate aleatorie, culoare aleatorie, deformare (shear), distorsiune aleatorie. Acest lucru se poate observa în Figura 1.



Figur 1 Baza de date augmentată

Această augmentare a setului de date a fost esențială pentru a preveni problema de overfitting și pentru a asigura că modelul antrenat este robust și capabil să generalizeze bine pe date noi. Astfel, baza de date augmentată a contribuit semnificativ la îmbunătățirea performanței modelului în clasificarea nivelului de lichid din sticlă.

### 3.Biblioteca

Pentru crearea și antrenarea modelelor, am utilizat biblioteca TensorFlow, proiectul fiind realizat în limbajul Python. TensorFlow este o bibliotecă open-source dezvoltată de Google, destinată implementării algoritmilor de calcul numeric în cadrul aplicațiilor de învățare automată. Aceasta funcționează prin definirea operațiunilor matematice în grafuri de computație, unde nodurile reprezintă operațiuni matematice, iar muchiile sunt valori numerice transmise ca vectori multidimensionali, numiți tensori. Datorită arhitecturii sale flexibile, TensorFlow permite implementarea facilă a calculelor pe diverse platforme, inclusiv procesoare, unități de procesare grafică (GPU) și clustere de servere.

Pe lângă TensorFlow, am utilizat și Keras, o bibliotecă software gratuită care servește drept interfață pentru rețele neuronale artificiale. Keras funcționează ca o interfață pentru TensorFlow, oferind numeroase implementări ale componentelor esențiale ale rețelelor neuronale, cum ar fi straturi, funcții de activare și optimizatori. De asemenea, Keras pune la dispoziție o varietate de instrumente menite să simplifice gestionarea și prelucrarea datelor, facilitând astfel dezvoltarea rapidă și eficientă a modelelor de învățare automată.

Pentru a păstra și reutiliza modelul antrenat, am folosit funcționalitățile încorporate în TensorFlow și Keras pentru salvarea și încărcarea modelelor. Aceste funcționalități permit salvarea modelului în formatul specific al bibliotecii Keras (de exemplu, .h5 pentru modelele secvențiale) și încărcarea ulterioară a acestuia pentru evaluare sau utilizare practică

## 4.Arhitectura

Arhitectura modelului utilizat include multiple straturi de convoluție, straturi de pooling și straturi complet conectate, integrând tehnici de regularizare pentru a preveni cazul de overfitting.

Modelul începe cu două straturi de convoluție, fiecare având 60 de filtre cu dimensiunea de 5x5 și funcția de activare ReLU. Aceste straturi sunt urmate de un strat de pooling maxim cu dimensiunea 2x2, care reduce dimensiunile spațiale ale reprezentărilor învățate și ajută la generalizarea modelului, prevenind cazul de overfitting. Următoarele două straturi de convoluție au 30 de filtre fiecare, cu dimensiunea de 3x3 și funcția de activare ReLU, urmate din nou de un strat de pooling maxim cu dimensiunea 2x2. Aceste straturi suplimentare ajută la extragerea caracteristicilor mai complexe din imagini.

Pentru a reduce riscul de overfitting, modelul include un strat Dropout cu o rată de 0.5, ceea ce înseamnă că jumătate din nodurile din stratul precedent sunt ignorate aleatoriu în timpul fiecărui pas de antrenare. După stratul de convoluție și pooling, modelul folosește un strat Flatten pentru a transforma matricea de caracteristici 2D într-un vector 1D. Acesta este urmat de un strat dens (complet conectat) cu 500 de noduri și funcția de activare ReLU, și un alt strat Dropout cu o rată de 0.5.

Modelul se încheie cu un strat complet conectat (Dense) cu un număr de noduri egal cu numărul claselor (noOfClasses) și funcția de activare softmax, pentru a produce probabilitățile de clasificare pe fiecare clasă. Modelul este compilat folosind optimizatorul Adam cu o rată de învățare de 0.001 și funcția de pierdere categorical crossentropy, fiind evaluat pe baza acurateței.

Această arhitectură bine definită și metodele de regularizare utilizate contribuie la îmbunătățirea performanței modelului și la generalizarea acestuia pe date noi, esențială pentru clasificarea corectă a nivelului de lichid din sticlă. În plus, arhitectura modelului prezintă următoarele detalii referitoare la parametrii săi: totalul parametrilor este de 35,633,951 (135.93 MB), dintre care 11,877,983 (45.31 MB) sunt parametri antrenabili, iar 23,755,968 (90.62 MB) reprezintă parametri ai optimizatorului. Nu există parametri neantrenabili.

## 5.Rezultate experimentale

În urma antrenării, s-a obținut un model care a atins următoarele performanțe: o acuratețe de 99.94% pe setul de validare și o eroare de 0.0066. A fost salvat cel mai bun model din timpul antrenării. Curbele de antrenare și de validare au fost urmărite utilizând matplotlib pentru a plota evoluția acurateței și a pierderii (Figura 2 și Figura 3).

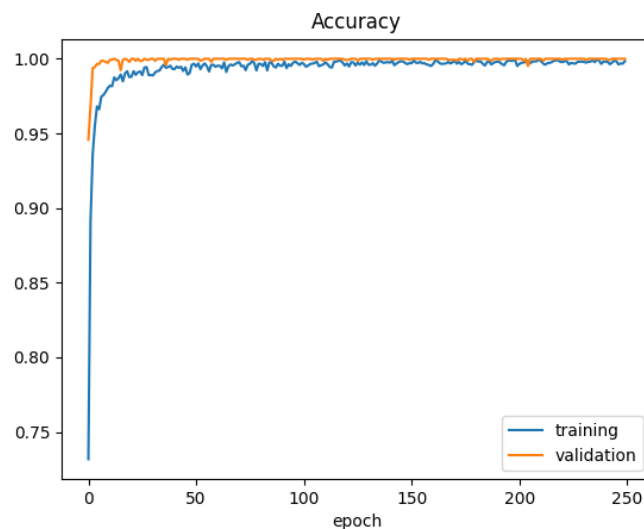


Figura 2 Evoluția acurateței

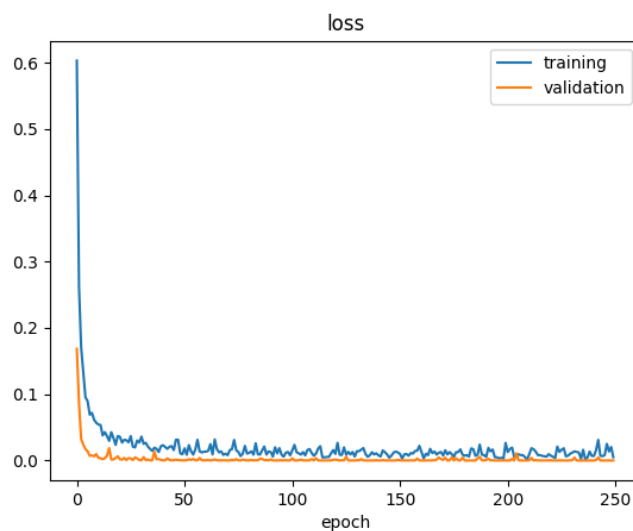
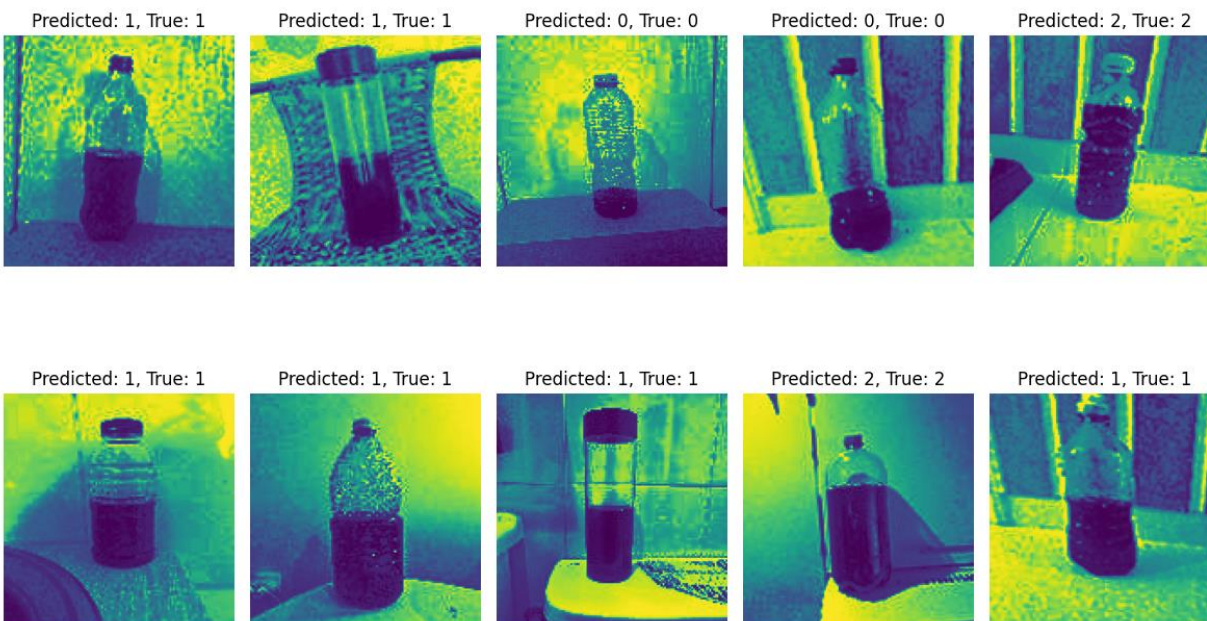


Figura 3 Evoluția pierderii

Modelul a fost testat pe baza de testare, demonstrând o capacitate excelentă de generalizare pe date noi. Performanța maximă a fost obținută atunci când modelul a fost antrenat pentru 250 de epoci, ceea ce indică faptul că modelul este foarte precis în clasificarea nivelului de lichid din sticlă.

Rezultatele predicției modelului pot fi vizualizate în Figura 4.



*Figura 4 Rezultate obținute în urma predicției*

În această figură, fiecare imagine este asociată unei clase specifice, conform categoriilor definite: clasa 0 reprezintă nivelul de lichid sub 50%, clasa 1 reprezintă nivelul de lichid în jur de 50%, iar clasa 2 reprezintă nivelul de lichid peste 50%.

Imaginile din această figură au fost pregătite anterior pentru a fi adecvate analizei ulterioare. În primul rând, imaginile au fost transformate în tonuri de gri, eliminând informația de culoare pentru a simplifica analiza. Apoi, a fost aplicată o tehnică de echilibrare a luminozității pentru a uniformiza distribuția intensității luminoase în imaginile procesate. În final, valorile pixelilor au fost normalizate pentru a fi cuprinse între 0 și 1, asigurând consistența și comparabilitatea acestora în cadrul modelului de învățare automată.



În figura 5, putem observa rezultatul unei demonstrații utilizând imagini noi preluate de pe internet, care reprezintă tipuri de sticle și lichide care nu au fost incluse în baza de date originală.



Figura 5 Rezultate demo

Clasificarea realizată este una corectă, ceea ce indică capacitatea modelului de a face predicții precise chiar și pe date noi, nevăzute în timpul antrenării.

## 6.Compararea performanțelor

În acest capitol, am comparat performanțele a două modele de rețele neuronale convoluționale utilizate pentru clasificarea nivelului de lichid din sticlă: un model personalizat și modelul preantrenat ResNet50. Deoarece baza de date folosită în acest proiect a fost creată de mine, am decis să rulez un model folosind ResNet50 pentru a evalua performanța acestuia în comparație cu modelul personalizat.

ResNet50 este o arhitectură de rețea neuronală profundă dezvoltată de echipa de cercetare de la Microsoft. ResNet, sau Residual Network, a fost introdusă pentru a aborda problema degradării performanței în rețelele foarte adânci. ResNet50 constă din 50 de straturi de rețea și utilizează blocuri reziduale care permit semnalului să sară peste unul sau mai multe straturi. Această abordare facilitează antrenarea rețelelor mult mai adânci fără a pierde din acuratețea modelului.

Parametrii arhitecturii ResNet50 utilizate sunt următorii:

- Total params: 24,638,341 (93.99 MB)
- Trainable params: 1,050,627 (4.01 MB)
- Non-trainable params: 23,587,712 (89.98 MB)
- Optimizer params: 2 (12.00 B)

În comparație, modelul meu personalizat are parametrii:

- Total params: 35,633,951 (135.93 MB)
- Trainable params: 11,877,983 (45.31 MB)
- Non-trainable params: 0 (0.00 B)
- Optimizer params: 23,755,968 (90.62 MB)

### Performanțele Modelului Personalizat

Modelul personalizat a fost antrenat și testat utilizând arhitectura detaliată anterior. Acesta a obținut rezultate remarcabile:

- **Test Score:** 0.006621099542826414

- **Test Accuracy:** 0.9994444251060486

Aceste rezultate indică o eroare minimă și o acuratețe foarte ridicată, ceea ce sugerează că modelul personalizat este extrem de eficient în clasificarea imaginilor din setul de date.

### Performanțele Modelului ResNet50

Modelul ResNet50, cunoscut pentru performanțele sale înalte în sarcini de clasificare a imaginilor, a fost de asemenea antrenat și testat pentru aceeași sarcină. Rezultatele obținute au fost următoarele:

- **Test Score:** 0.3265560567378998
- **Test Accuracy:** 0.8772222399711609

Deși ResNet50 a demonstrat o performanță bună, acuratețea obținută este semnificativ mai mică decât cea a modelului personalizat.

Curbele de acuratețe și pierdere pentru modelul ResNet50 sunt prezentate în Figura 6 și Figura 7.

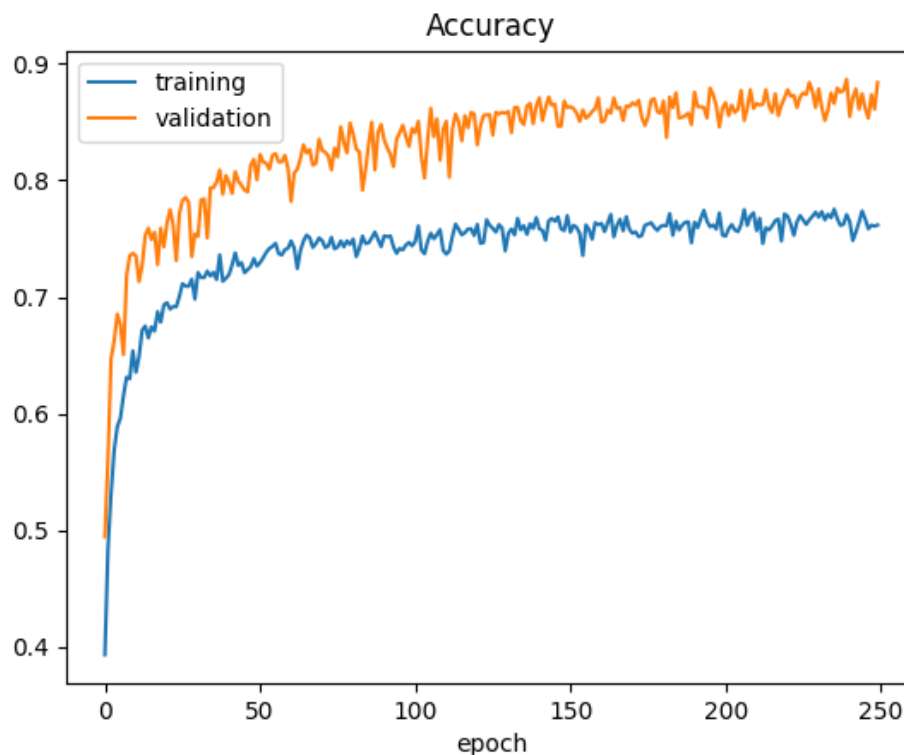
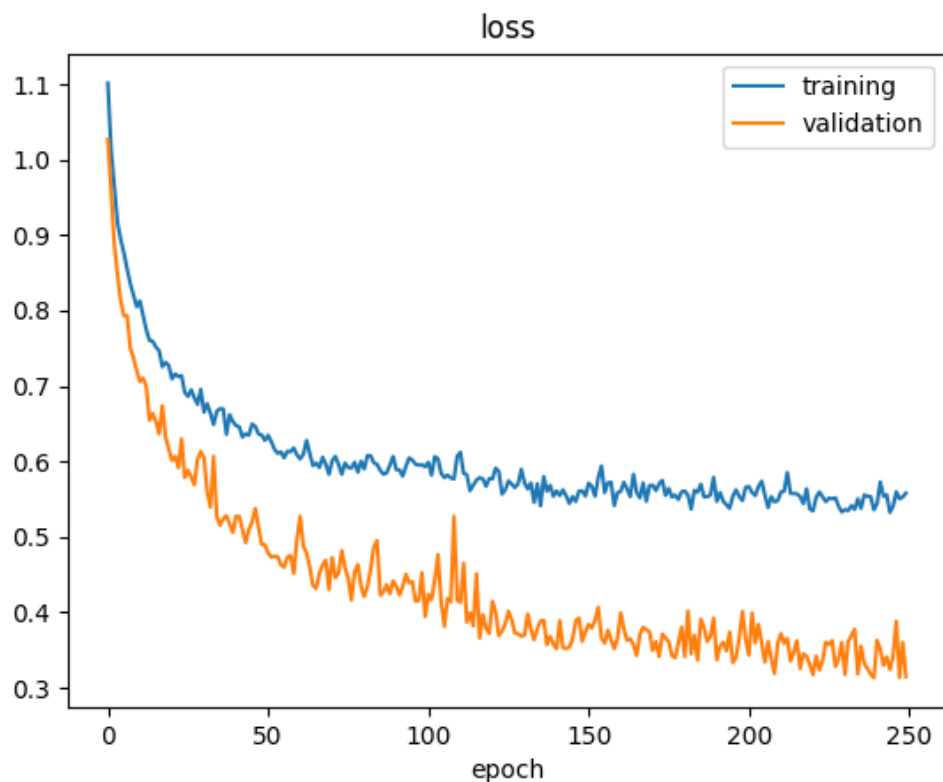


Figura 6 Evoluția acurateței model ResNet50



*Figura 7 Evoluția pierderii model ResNet50*

Aceste figuri arată fluctuații mai mari și o convergență mai lentă în comparație cu modelul personalizat, indicând că ResNet50 nu este la fel de bine adaptat pentru acest set de date specific.

Comparând cele două modele, putem concluziona că modelul personalizat a obținut rezultate superioare în ceea ce privește acuratețea și eroarea de testare. Acest lucru sugerează că arhitectura și tehnicile implementate în modelul personalizat sunt bine adaptate pentru sarcina de clasificare a nivelului de lichid din sticlă în acest context specific.

## 7. Bibliografie

- [1] Vizitiu, A. (n.d.). Notițe laborator: Deep Learning.
- [2] TensorFlow. (n.d.). În Wikipedia. <https://en.wikipedia.org/wiki/TensorFlow>
- [3] Keras. (n.d.). În Wikipedia. <https://en.wikipedia.org/wiki/Keras>
- [4] Keras Applications. (n.d.). ResNet50. <https://keras.io/api/applications/resnet/>
- [5] ChatGPT. (n.d.). OpenAI.