

POLITEHNICA BUCHAREST NATIONAL UNIVERSITY FOR SCIENCE AND TECHNOLOGY

Faculty of Automatic Control and Computers
Computer Science and Engineering Department



Detectarea Automată a Tumorilor în Imagini CT Folosind
Algoritmul Support Vector Machine (SVM)

Îmbunătățirea Acurateții Diagnosticului cu Ajutorul Învățării
Automate (Machine Learning)

VĂDANA IOAN-GRIGORE

2023

CONTINUT

1. [DESCRIEREA PROIECTULUI](#)
2. [ACHIZITIONAREA ȘI ORGANIZAREA DATELOR](#)
3. [PREGĂTIREA DATELOR](#)
4. [ALGORITM ȘI PARAMETRI](#)
5. [ANTRENAREA MODELULUI](#)
6. [EVALUAREA MODELULUI](#)
7. [REZULTATE](#)
8. [BIBLIOTECILE PYTHON](#)
9. [CONCLUZII](#)

DESCRIEREA PROIECTULUI

Context și Justificare:

Proiectul se axează pe dezvoltarea unei soluții de învățare automată pentru clasificarea imaginilor Computer Tomograf (CT), cu scopul de a identifica prezența sau absența tumorilor cerebrale. Această inițiativă răspunde unei nevoi critice în domeniul diagnosticării medicale, unde precizia și rapiditatea sunt esențiale. Având în vedere volumul mare de imagini CT pe care radiologii trebuie să le analizeze, un astfel de instrument automatizat poate oferi un suport semnificativ, reducând riscul de eroare umană și accelerând procesul de diagnosticare.

Obiectivele Proiectului:

Principalul obiectiv este de a antrena un model de Machine Learning capabil să diferențieze eficient între imagini CT cu și fără tumori. Acest lucru nu doar că ar putea îmbunătăți acuratețea diagnosticării, dar ar putea și să ajute la prioritizarea cazurilor care necesită atenție medicală urgentă.

Metodologie:

Pentru atingerea acestui obiectiv, am ales să utilizez algoritmul SVM (Suport Vector Machine) datorită eficienței sale recunoscute în clasificarea datelor de înaltă dimensiune, cum sunt imaginile. Am implementat și un proces riguros de preprocesare și standardizare a datelor pentru a asigura calitatea inputului pentru modelul SVM.

```
# Train an SVM model
svm = SVC(kernel='linear')
svm.fit(X_train_scaled, y_train)
```

Importanța și Impactul Proiectului:

Implementarea cu succes a acestui model are potențialul de a transforma modul în care sunt evaluate și diagnosticate imaginile CT în context medical. Acest proiect demonstrează cum tehnologiile avansate pot fi utilizate pentru a sprijini și îmbunătăți procesele critice de îngrijire a sănătății.

ACHIZIȚIONAREA ȘI ORGANIZAREA DATELOR

Structurarea și Etichetarea Datelor:

Odată colectate, imaginile [1] (în număr de 3762) au fost structurate în directoare separate în funcție de prezența sau absența tumorilor - 'tumor' sau 'no tumor' - pentru a simplifica încărcarea și preprocesarea în etapele ulterioare ale proiectului. Această organizare a facilitat sarcina de încărcare și procesare a imaginilor în etapele ulterioare.

```
# Load images
tumor_images, tumor_labels = load_images_from_folder( folder: r'C:\Users\Grig\PycharmProjects\ML\Data\tumor', label: 1)
no_tumor_images, no_tumor_labels = load_images_from_folder( folder: r'C:\Users\Grig\PycharmProjects\ML\Data\no_tumor', label: 0)
```

Pregătirea seturilor de date:

Setul complet de date a fost împărțit în trei subseturi: antrenament, validare și testare. Această împărțire a fost esențială pentru evaluarea obiectivă a modelului, asigurând că acesta este antrenat și testat pe date diferite, ceea ce contribuie la evitarea suprapotrivirii și la îmbunătățirea capacității de generalizare.

```
X_train, X_temp, y_train, y_temp = train_test_split(*arrays: data, labels, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(*arrays: X_temp, y_temp, test_size=0.5, random_state=42)
```

PREGĂTIREA DATELOR

Redimensionarea Imaginilor:

Toate imaginile au fost redimensionate la dimensiuni uniforme de 256x256 pixeli. Această standardizare asigură că fiecare imagine are același număr de caracteristici, facilitând procesarea ulterioară și analiza de către modelul SVM.

Normalizarea Valorilor Pixelilor:

Imaginile au fost apoi normalizate astfel încât valorile pixelilor să fie în intervalul 0-1. Acest pas este crucial pentru modelele de învățare automată, deoarece normalizează variația în intensitatea luminii și contrastul între diferite imagini.

Aplatizarea Imaginilor:

Pentru a le face compatibile cu modelul SVM, imaginile 2D au fost transformate în vectori 1D printr-un proces numit aplatizare. Aceasta este o etapă tehnică necesară, deoarece SVM-ul lucrează cu date în format vectorial

ALGORITM ȘI PARAMETRI

Am ales SVM ca algoritm de bază pentru acest proiect datorită eficienței sale recunoscute în clasificarea datelor de dimensiuni mari, precum imaginile. SVM este particular potrivit pentru acest tip de sarcină deoarece excellează în găsirea unui hiperplan optim care separă clasele de date, în cazul nostru, imaginile cu tumori de cele fără tumori.

Parametrii SVM

- **Kernel SVM:** Pentru modelul nostru, am ales un kernel liniar, adecvat pentru datele mele care nu necesită transformări complexe ale spațiului de caracteristici. Kernelul liniar este eficient și reduce riscul de overfitting pentru seturi de date mari.
- **Parametrii de Regularizare:** În modelul acesta este folosită valoarea implicită din **scikit-learn**, care este 1.0. Acest parametru controlează trade-off-ul între clasificarea corectă a punctelor de antrenament și maximizarea marginii de decizie.

Standardizarea Datelor:

Pentru a optimiza performanța SVM, am standardizat setul de date folosind **StandardScaler**, care ajustează caracteristicile la o distribuție cu medie zero și deviație standard unitară.

```
# Standardize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)
```

ANTRENAREA MODELULUI

Procesul de Antrenare:

Antrenarea modelului SVM este un pas crucial în acest proiect. Acest proces implică furnizarea setului de date de antrenament la model, permițându-i să învețe cum să diferențieze între imagini cu și fără tumori.

```
# Train an SVM model
svm = SVC(kernel='linear')
svm.fit(X_train_scaled, y_train)
```

Importanța Datelor de Antrenament:

Calitatea și cantitatea datelor de antrenament sunt esențiale. Prin utilizarea unui set de date divers și reprezentativ, ne asigurăm că modelul poate generaliza bine pe date noi, o caracteristică vitală pentru aplicabilitatea clinică.

Evaluarea Performanței în Timpul Antrenării:

Pe parcursul antrenării, monitorizăm performanța modelului pe setul de validare. Aceasta ne ajută să detectăm și să evităm overfitting-ul (suprapotrivirea) - o problemă comună în învățarea automată, unde modelul se potrivește prea bine cu datele de antrenament și nu performează bine pe date noi.

EVALUAREA MODELULUI

Utilizarea Setului de Validare:

Odată ce modelul este antrenat, următorul pas esențial este evaluarea performanței acestuia. Folosim setul de validare, care nu a fost implicat în antrenarea modelului, pentru a testa cât de bine poate modelul să clasifice date noi și necunoscute. Acest lucru ne oferă o estimare realistă a modului în care modelul ar putea performa în aplicații practice.

```
# Validate the model
y_val_pred = svm.predict(X_val_scaled)
print("Validation Results:")
print(classification_report(y_val, y_val_pred))
```

Analiza Performanței:

Evaluăm modelul folosind mai multe metrice, cum ar fi acuratețea, precizia, recall-ul și scorul F1. Aceste metrice ne oferă o imagine completă a performanței modelului, permițându-ne să identificăm punctele forte și aspectele care necesită îmbunătățiri.

```
print("Test Results:")
print("Test Accuracy:", accuracy_score(y_test, y_test_pred))
print(classification_report(y_test, y_test_pred))
```

Matricea de Confuzie:

Utilizăm, de asemenea, o matrice de confuzie pentru a vizualiza performanța modelului în clasificarea fiecărei clase. Aceasta ne ajută să înțelegem tipurile de erori comise de model.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

cm = confusion_matrix(y_test, y_test_pred)
plt.figure(figsize=(10,7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.title('Confusion Matrix')
plt.show()
```

Rezultatele evaluării ne ajută să determinăm dacă modelul este gata să fie testat pe setul de date de test sau dacă sunt necesare ajustări suplimentare.

REZULTATE

Analiza Rezultatelor de Validare si Testare:

```
C:\Users\Grig\PycharmProjects\ML\venv\Scripts\python.exe C:\Users\Grig\PycharmProjects\ML\test.py
Validation Results:
```

	precision	recall	f1-score	support
0	0.90	0.86	0.88	272
1	0.86	0.90	0.88	257
accuracy			0.88	529
macro avg	0.88	0.88	0.88	529
weighted avg	0.88	0.88	0.88	529

```
Test Results:
Test Accuracy: 0.8981132075471698
```

	precision	recall	f1-score	support
0	0.90	0.90	0.90	259
1	0.90	0.90	0.90	271
accuracy			0.90	530
macro avg	0.90	0.90	0.90	530
weighted avg	0.90	0.90	0.90	530

Rezultatele validării indică o performanță echilibrată a modelului, cu o acuratețe generală de 88%. Precizia și recall-ul sunt aproape simetrice pentru ambele clase, sugerând că modelul nu este părtinitor față de niciuna dintre categorii. Un scor F1 de 0.88 pentru ambele clase indică o bună armonie între precizie și recall, semn că modelul gestionează bine atât cazurile pozitive, cât și cele negative.

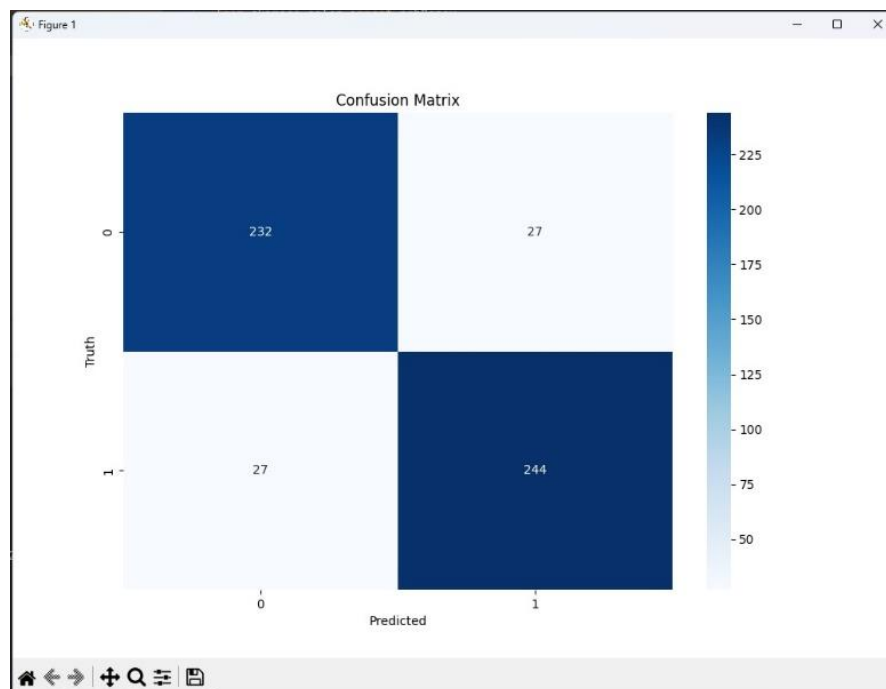
Rezultatele testului arată o ușoară îmbunătățire, cu o acuratețe totală de aproape 90%. Acest lucru confirmă robustețea modelului, arătând că poate generaliza bine pe date noi, o caracteristică esențială pentru aplicarea în scenarii reale.

Interpretarea Metricilor Statistice

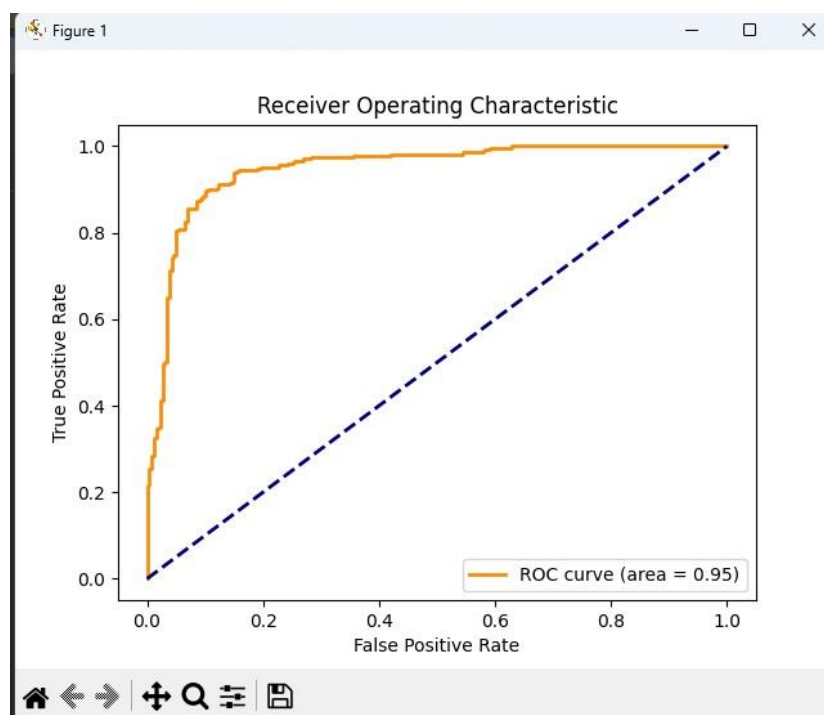
- **Precision** (Precizie) reprezintă procentajul de predicții pozitive corecte din totalul predicțiilor pozitive făcute de model. În contextul nostru, o precizie înaltă înseamnă că atunci când modelul prezice că o imagine are o tumoră, este foarte probabil să fie corect.
- **Recall** (Sensibilitate) reprezintă procentajul de cazuri pozitive reale care au fost corect identificate de model. Un recall înalt este esențial în diagnosticarea medicală pentru a minimiza rata cazurilor pozitive care sunt trecute cu vederea.
- **F1-score** este media armonică între precizie și recall, oferind un singur scor care echilibrează ambele metrice.

Interpretarea imaginilor pentru performanței acestuia:

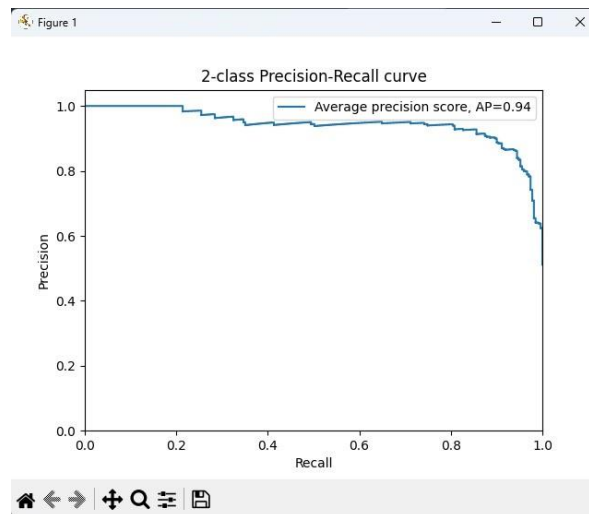
- **Matricea de Confuzie:** Arată că modelul a confundat o mică proporție de imagini pentru fiecare clasă, cu un număr egal de fals pozitive și fals negative, indicând echilibru.



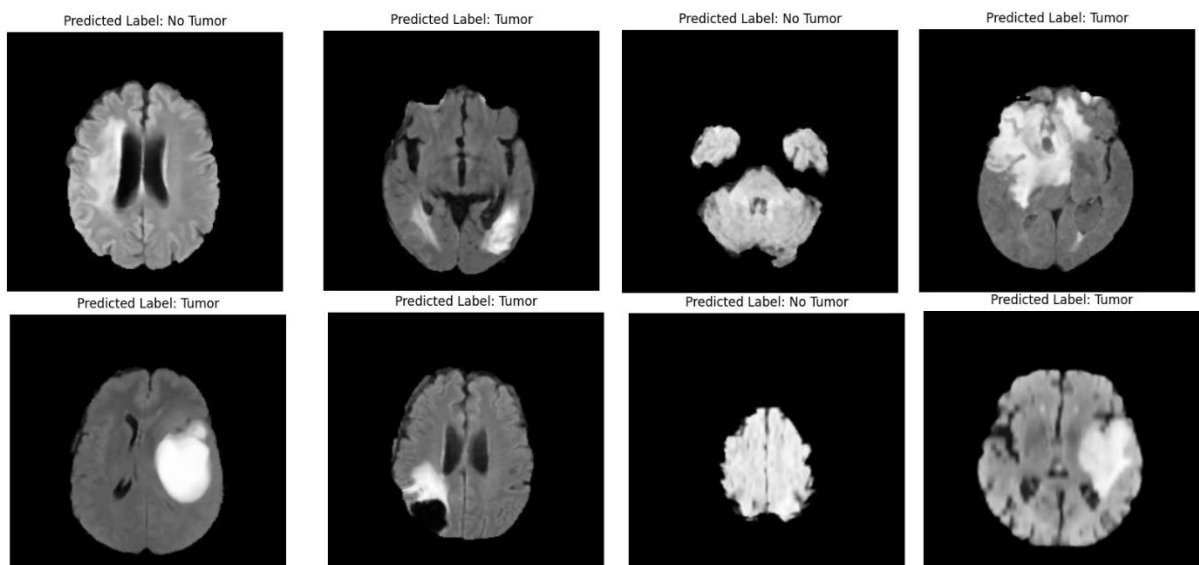
- **Curba ROC:** Cu o valoare AUC (Aria de Sub Curba ROC) de 0.95, curba ROC demonstrează o performanță excelentă a modelului, cu o capacitate foarte bună de a distinge între clasele pozitive și negative.



- **Curba Precision-Recall:** Un scor mediu de precizie de 0.94 pe curba Precision-Recall subliniază că modelul are o rată mare de așteptare a cazurilor pozitive, un factor important în contextul medical.



Analiza Vizuală a Imaginilor (exemplu pentru 8 imagini):



- Primele două imagini marcate ca "No Tumor" și "Tumor" par să fie clasificate corect, bazându-ne pe etichetele prezise. Prima imagine arată un creier fără anomalii evidente, în timp ce a doua arată o anomalie întunecată care poate reprezenta o tumoră.
- Următoarele două imagini marcate ambele ca "No Tumor" arată o separare clară între țesuturile sănătoase și zonele afectate, ceea ce sugerează că modelul a învățat să identifice anumite trăsături specifice tumorilor.
- Imaginile marcate ca "Tumor" prezintă diverse forme și dimensiuni ale anomaliilor, ceea ce indică faptul că modelul nu se bazează pe un singur tipar pentru a clasifica o imagine ca prezentând o tumoră.

BIBLIOTECILE PYTHON

Pentru construirea, antrenarea și evaluarea modelului SVM, am utilizat biblioteca **scikit-learn**. Aceasta a oferit instrumentele necesare pentru a implementa SVM și a calcula metricile de performanță, precum și pentru a împărți datele și a efectua standardizarea.

Biblioteca **scikit-image** a fost esențială pentru preprocesarea imaginilor, oferind funcții pentru citirea imaginilor, conversia lor în tonuri de gri și redimensionarea lor într-un format standardizat.

NumPy este o bibliotecă fundamentală pentru calcul științific în Python, permițându-ne să lucrăm eficient cu matricile de imagini și să efectuăm operații matematice necesare în preprocesare și evaluare.

Pentru vizualizarea rezultatelor, am utilizat **Matplotlib** și **Seaborn**, care sunt biblioteci de construire a graficelor în Python. Acestea au fost folosite pentru a genera matricea de confuzie, curba ROC și curba Precision-Recall.

CONCLUZII

În acest proiect, am reușit să dezvoltăm un model SVM care clasifică cu succes imagini CT în funcție de prezența sau absența tumorilor cerebrale. Modelul nostru a arătat o capacitate solidă de discriminare între imagini pozitive și negative, cu metrici de performanță care indică o acuratețe, precizie și recall echilibrate. Vizualizările generate, inclusiv matricea de confuzie și curbele ROC și Precision-Recall, au furnizat o confirmare vizuală a competențelor modelului.

Bibliografie:

[1]:

<https://www.kaggle.com/datasets/jakeshbohaju/brain-tumor>