Deep Hallucination Classification

Boboc Ștefan
02.04.2022

Conținut

1. Introducere

Proiectul presupune clasificarea unor imagini - „deep image hallucination” - cu ajutorul unui model de ML. Imaginile pot fi încadrate pe 7 categorii (lable = [0,6]), fiecare imagine fiind de dimensiune 16x16 pixeli. Setul de antrenare este alcătuit din 8.000 de imagini, iar setul de validare este format din 1.173 imagini.

În urma mai multor încercări am decis ca pentru acest proiect să propun trei metode de abordare: Naive Bayes (NB), Support Vector Machine (SVM) și Multi-layer Perceptron Classifier

2. Naive Bayes

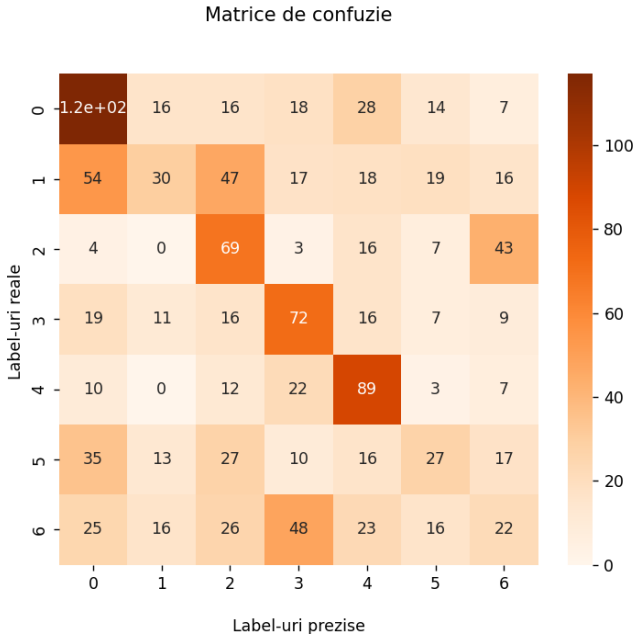
Pentru început am încercat o abordare a modelelor Naive Bayes, acestea făcând parte din categoria clasificatorilor probabilistici.

La această categorie, testând mai multe modele de clasificare NB, am observat că modelul MultinomialNB și cel CategoricalNB obțin cele mai bune rezultate. Diferența dintre MultinomialNB și CategoricalNB este faptul că, CategoricalNB, spre deosebire de MultinomialNB, în timpul procesării, el numără și apariția pixelilor în clasele complementare. Am adus diferite modificări asupra parametrului alpha -*Laplace smoothing parameter*-, un factor de probabilitate.

model \ alpha	0.4	0.6	0.9	1.0 (default)
MultinomialNB	0.3653	0.3653	0.3657	0.3657
CategoricalNB	0.3589	0.3597	0.3597	0.3589

Valori diferite ale lui alpha pe datele de validare

Din tabelul de mai sus se poate observa că modelul MultinomialNB obține cea mai bună performanță pentru *alpha = 1.0*.




```
Matrice de confuzie:
 [[117  16  16  18  28  14   7]
 [ 54  30  47  17  18  19  16]
 [  4   0  69   3  16   7  43]
 [ 19  11  16  72  16   7   9]
 [ 10   0  12  22  89   3   7]
 [ 35  13  27  10  16  27  17]
 [ 25  16  26  48  23  16  22]]
```

În urma testării modelului pe platforma Kaggle acesta a obținut scorul de acuratețe de 0.35511.

3. Support Vector Machine

După încercarea modelelor NB am testat și un model SVM. Am ales să folosesc clasificatorul SVC(). Acesta caută cele mai bune hyperplane între clasele de imagini. Modelul este utilizat din ScikitLearn care are o abordare one-to-one, încercând ca pentru fiecare label să traseze câte un hyperplan pentru a separa obiectele.

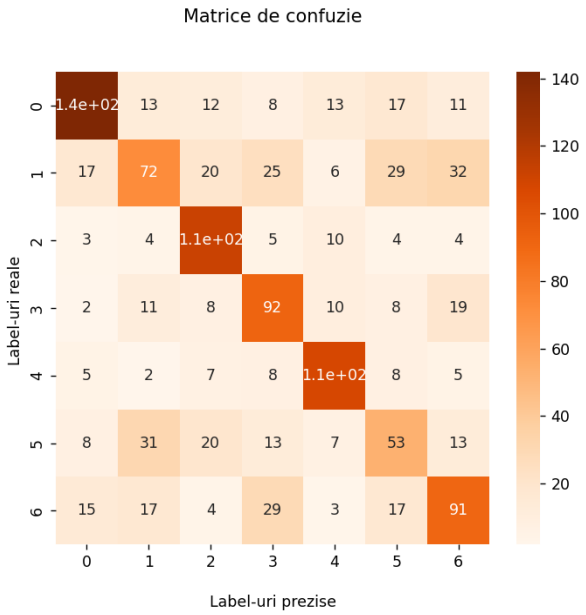
3.1. Pentru început, din partea de preprocesare a datelor din modelul NB am păstrat redimensionarea listei de dimensiune 4 într-o dimensiune 2 cu ajutorul funcției *flatten()*. După am luat diferite valori ale hyperparametrului C de regularizare și am observat că pentru o valoare cât mai apropiată de 5.6, modelul oferă cea mai bună performanță.

```
# default  ..... 0.541
# C = 1.3  ..... 0.55
# C = 1.35 ..... 0.553
# C = 1.4  ..... 0.552
# C = 3.0  ..... 0.552
# C = 4.0  ..... 0.554
# C = 5.0  ..... 0.559
# C = 5.6  ..... 0.564
# C = 6.0  ..... 0.560
```

3.2. După am aplicat asupra setului de date o organizare pe bin-uri. După mai multe teste am constatat că la o spațiere egală de 20, modelul oferă o performantă de 0.571.

No. bins	num = 15	num = 17	num = 18	num = 19	num = 20	num = 21
Acuratețe	0.559	0.563	0.568	0.560	0.571	0.55

Valori diferite ale dimensiunii bin-urilor




```
Matrice de confuzie:
 [[142  13  12   8  13  17  11]
 [ 17  72  20  25   6  29  32]
 [  3   4 112   5  10   4   4]
 [  2  11   8  92  10   8  19]
 [  5   2   7   8 108   8   5]
 [  8  31  20  13   7  53  13]
 [ 15  17   4  29   3  17  91]]
```

3.3. Dorind să măresc setul de date de testare am așezat fiecare imagine în oglindă pe axa verticală. Folosind operația de *slice* am mutat fiecare pixel al fiecărei imagini în capetele opuse.

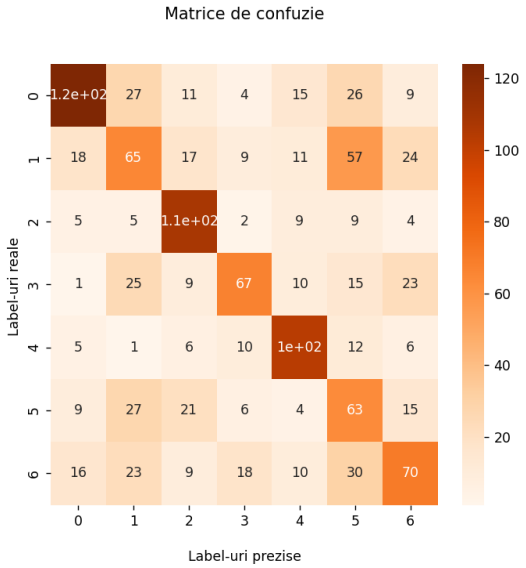
În urma testărilor, modelul a obținut pe datele de validare un scor de aproximare mai slab, acesta fiind de 0.52.

În final, am păstrat modelul SVC() cu preprocesarea datelor de la punctul 3.2., acesta fiind cel mai bun de până acum, iar pe datele de test a obținut pe platforma Kaggle, scorul de 0.57954.

4. Multi-layer Perceptron Classifier

Pentru acest model am folosit clasificatorul MLPClassifier(). Am ajustat parametrii: *batch_size=128* - mărimea batch-ului pentru antrenare; *hidden_layer_sizes=(1000,1000)* - numărul de straturi și neuroni ascunși; *early_stopping=True* - antrenarea se va termina dacă eroarea pe mulțimea de validare nu se îmbunătățește.

Aproximarea pe datele de validare a fost de 0.5080, iar pe datele de test de pe platforma Kaggle a fost de 0.53977.




```
Matrice de confuzie:
 [[124  27  11   4  15  26   9]
 [ 18  65  17   9  11  57  24]
 [  5   5 108   2   9   9   4]
 [  1  25   9  67  10  15  23]
 [  5   1   6  10 103  12   6]
 [  9  27  21   6   4  63  15]
 [ 16  23   9  18  10  30  70]]
```

5. Funcții auxiliare

Pentru testele pe mai multe valori ale tuturor hyperparametrilor am utilizat biblioteca *GridSearchCV* din pachetul *sklearn.model_selection*.

Pentru obținerea matricii de confuzie am utilizat biblioteca *metrics* din *sklearn*, iar pentru plotarea acesteia am utilizat *seaborn.*

6. Concluzie

Scorurile de pe Kaggle pentru cele trei modele sunt:

Model	Acuratețe
Naive Bayes - MultinomialNB()	0.35511
Support Vector Machine - SVC()	**0.57954**
Multi-layer Perceptron Classifier - MLPClassifier()	0.53977

cel mai bun dintre ele fiind modelul bazat pe **Support Vector Machine**.