

Detecția și recunoașterea semnelor de circulație

Dumitriu Radu-Toader

Calculatoare și Tehnologie Informației
Facultatea de Automatică și Calculatoare
Iași, România
radu-toader.dumitriu@student.tuiasi.ro

Atanase Alexandru-Teodor

Calculatoare și Tehnologie Informației
Facultatea de Automatică și Calculatoare
Iași, România
alexandru-teodor.atanase@student.tuiasi.ro

Abstract—În zilele noastre, detecția semnelor de circulație joacă un rol crucial în autonomia mașinilor. Prima utilizare a detectării semnelor de circulație a fost în anii 1960, atunci când s-au dezvoltat primele sisteme de recunoaștere a semnelor de circulație utilizând tehnologii precum analiza de imagine și procesarea semnalului. Aceste sisteme au fost utilizate în principal în vehiculele de transport public, precum autobuze și metrou, pentru a ajuta la navigarea automată și la respectarea regulilor de circulație. Încă de atunci această tehnologie este în dezvoltare și presupune eficientizarea ei, optimizarea și nu în ultimul rând folosirea în cât mai multe domenii. Acest articol are ca scop informarea cititorului cu privire la modul în care este realizată detectarea semnelor de circulație, pornind de la încă de la nevoia care a stat la baza unei astfel de tehnologii și până la implementarea propriu-zisă.

Index Terms—rețele neuronale, detecție, semne de circulație

I. INTRODUCERE

Detecția semnelor de circulație este o componentă importantă a sistemelor de asistare a conducerii (ADAS) și a vehiculelor autonome. Această tehnologie permite vehiculului să recunoască și să interpreteze semnele de circulație, cum ar fi limitele de viteză, interdicțiile și indicațiile de direcție, pentru a se conforma regulilor de circulație și a evita accidentele.

Nevoia care a stat la baza detecției semnelor de circulație este aceea de a crește siguranța rutieră. Semnele de circulație sunt utilizate pentru a indica regulile de circulație și pentru a ghida șoferii în mod sigur prin trafic. Însă, datorită numărului mare de accidente rutiere care au loc în întreaga lume, a devenit evident faptul că este necesar să se găsească metode suplimentare pentru a asista șoferii.

Detecția semnelor de circulație este una dintre aceste metode, care permite vehiculului să recunoască și să interpreteze semnele de circulație și să se conformeze regulilor. Această tehnologie poate fi utilizată pentru a asista șoferul prin furnizarea de informații despre viteză și semnele de circulație din jurul vehiculului sau chiar de a interveni automat prin sisteme de frânare și accelerare pentru a evita accidentele.

În plus, cu dezvoltarea tehnologiei și a vehiculelor autonome, nevoia detecției semnelor de circulație se amplifică, deoarece vehiculele autonome au nevoie de informații precise despre semnele de circulație pentru a se conforma regulilor de circulație și pentru a evita accidentele.

Prin urmare, nevoia care a stat la baza detecției semnelor de circulație este aceea de a crește siguranța rutieră prin asistarea șoferului sau prin intervenții automate pentru a evita

accidentele, și de a asigura o circulație sigură pentru vehiculele autonome.

Există mai multe metode de detecție a semnelor de circulație, cum ar fi detecția prin imagini, care utilizează camere video montate pe vehicul pentru a captura imagini ale semnelor de circulație și a le analiza folosind algoritmi de recunoaștere a imaginilor. Alte metode incluse sunt detecția prin radar și lidar, care utilizează unde de radio sau laser pentru a detecta și măsura semnele de circulație.

Eficacitatea acestor metode variază în funcție de condițiile de iluminare, vreme și alte factori, cum ar fi poziționarea semnului, unghiul de vizualizare și calitatea imaginii. De aceea, sistemele moderne de detecție a semnelor de circulație combină mai multe metode, cum ar fi detecția prin imagini și radar, pentru a maximiza precizia și fiabilitatea.

De asemenea, este important de menționat că detecția semnelor de circulație poate fi utilizată pentru a asista șoferul prin a-i furniza informații despre viteză și semnele de circulație din jurul vehiculului sau chiar de a interveni automat prin sisteme de frânare și accelerare pentru a evita accidentele.

În concluzie, detecția semnelor de circulație este o tehnologie esențială pentru vehiculele autonome și ADAS, care contribuie la siguranța rutieră prin asistarea șoferului sau prin intervenții automate pentru a evita accidentele.

II. METODE EXISTENTE

Semnele de circulație sunt utilizate pentru a furniza informații și instrucțiuni conducătorilor de vehicule pentru a asigura siguranța și fluxul corect al traficului. Acestea pot fi clasificate în funcție de culoare, formă și conținutul lor simbolic.

Culoarea semnului de circulație poate furniza informații importante despre semnificația sa. De exemplu, semnele de circulație roșii au o semnificație de avertisment, indicând interzicerea sau restricțiile de circulație. Semnele de circulație albastre indică direcția de circulație, în timp ce semnele de circulație verzi indică ghidarea și informații despre kilometraj. Semnele de circulație portocalii sunt utilizate pentru a indica lucrările de construcții și mentenanță, iar semnele de circulație maro sunt utilizate pentru a indica zonele de recreere.

Forma semnului de circulație poate de asemenea furniza informații importante despre semnificația sa. De exemplu, un cerc cu o linie roșie care trece prin el indică o interdicție, în timp ce un triunghi echilateral cu vârful în sus indică o

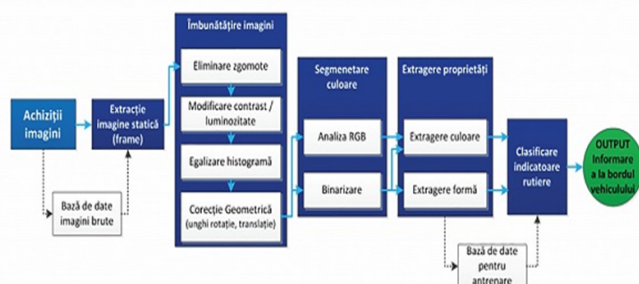
avertizare. Un octagon indica stop si un diamant indica lucrari de constructii sau mentenanta.

Tehnologia de astazi permite detectarea automata a semnelor de circulatie prin utilizarea algoritmilor de invatare automata si a camerelor video. Aceasta tehnologie poate fi utilizata pentru a asista conducatorii de vehicule in luarea deciziilor corecte si pentru a reduce accidentele de circulatie.

Culoare	Semnificație
Roșu	Avertisment
Albastru	Direcția
Verde	Ghidare și kilometraj
Portocaliu	Construcții și lucrări de mentenanță
Maro	Recreere
Galben	Avertizare
Alb	Auxiliar

Formă	Semnificație
Cerc	Interdicție
Triunghi echilateral cu vf. Sus	Avertizare
Triunghi echilateral cu vf. Jos	Cedare
Octagon	Stop
Diamant	Construcții și mentenanță
Dreptunghi	Ghidare și informare

Cum arată un sistem de detecție privit mai în detaliu:



Prezentarea imaginii de mai sus:

1. Camera de înaltă rezoluție și un unghi larg care detectează semnele de circulație și oferă imagini clare;
2. Segmentarea culorii, care este un pas important pentru a elimina toate obiectele de fond și informații neimportante în imagine;
3. Analiza formei semnelor de circulație: acest modul normalizează semnul de circulație recunoscut, astfel încât acesta să devină invariabil la transformări. Acest lucru înseamnă că semnul rezultat are o dimensiune fixă și este situat într-o poziție standard;
4. Baza de date cu imagini brute este pur și simplu o colecție de scene de trafic realizate de către cameră;
5. Baza de date cu imagini binare, care conține imagini binare de dimensiuni normalizate precum 36x36 pixeli. Această bază de date este folosită pentru extragerea unor caracteristici ale semnelor de circulație pentru validarea acestora;

6. Modulul de extragere a caracteristicilor imaginilor conține algoritmi care sunt utilizați pentru a extrage caracteristici de la oricare imagine formată în baza de date;

7. Modulul de clasificare a imaginii joacă un rol central în această etapă și semnele sunt clasificate în clase diferite, cum ar fi triunghiuri, cercuri, octogoane, etc. Pictograma analizată permite o etapă ulterioară de clasificare, prin analizarea pictogramei împreună cu textul din interiorul semnelor, fiind ușor a decide clasa individuală a semnelor în cauză.

Tehnologia de detectare a semnelor de circulație a evoluat semnificativ în ultimii ani. Sistemele moderne de detectare a semnelor de circulație pot identifica cu precizie diferite tipuri de semne, cum ar fi limitele de viteză, interdicțiile de viraj și indicatoarele de sens unic, folosind tehnologii precum inteligența artificială și analiza imaginilor. Aceste sisteme sunt de obicei integrate în camerele de bord ale mașinilor sau utilizate în vehiculele autonome pentru a ajuta la navigarea și la luarea deciziilor în timpul condusului.

Exista mai multe metode de detecție a semnelor de circulație, cum ar fi:

Detecția prin imagini: Aceasta metoda utilizează camere video montate pe vehicul pentru a captura imagini ale semnelor de circulație și a le analiza folosind algoritmi de recunoaștere a imaginilor.

Detecția prin radar: Aceasta metoda utilizează unde de radio pentru a detecta și măsura semnele de circulație.

Detecția prin lidar: Aceasta metoda utilizează laser pentru a detecta și măsura semnele de circulație.

Detecția prin infraroșu: Aceasta metoda utilizează detectarea temperaturii pentru a detecta semnele de circulație.

Detecția prin rețele neuronale: Aceasta metoda se bazează pe modele de inteligență artificială antrenate să detecteze și să interpreteze semnele de circulație.

Detecția prin sisteme GPS: Utilizează informații despre poziție pentru a detecta semnele de circulație.

Detecția combinată: Aceasta metoda combină mai multe metode de detecție, cum ar fi detecția prin imagini, radar, lidar sau infraroșu pentru a maximiza precizia și fiabilitatea.

În schimb metodele pe care ne vom concentra noi sunt cele bazate pe rețele neuronale sau machine learning cum ar fi: utilizarea unei rețele neuronale convoluționale (CNN) pentru a recunoaște forme și simboluri specifice semnelor de circulație din imagini capturate de camera vehiculului; utilizarea unei rețele neuronale de tip Recurrent Neural Network (RNN) pentru a analiza fluxul de imagini și a detecta semnele de circulație care pot fi prezente în mai multe cadre și nu în ultimul rând, YOLO (You Only Look Once), un algoritm de detecție a obiectelor bazat pe o rețea neurală care poate fi utilizat pentru a detecta și clasifica obiecte din imagini fiind dezvoltat pentru a detecta obiecte în imagini în timp real, cu o precizie ridicată.

A. Rețele Neuronale Convoluționale (CNN)

Acest tip de rețea neurală este special proiectată pentru a analiza imagini și este foarte precisă în detecția obiectelor.

Precizia unei rețele neuronale convoluționale depinde de numărul de straturi și de calitatea datelor de antrenament. Cu cât rețeaua are mai multe straturi și cu cât datele de antrenament sunt mai bune, cu atât precizia va fi mai mare.

Avantajele unei rețele neuronale convoluționale sunt:

- capacitatea de a recunoaște forme, culori și simboluri specifice semnelor de circulație
- capacitatea de a analiza imagini în timp real
- precizie înaltă în detectarea obiectelor

Dezavantajele unei rețele neuronale convoluționale sunt:

- necesitatea unui volum mare de date de antrenament pentru a obține o precizie ridicată
- necesitatea unui puternic calcul pentru a procesa imagini
- dificultatea de a generaliza modelele pre-antrenate pentru alte seturi de date sau alte aplicații

În concluzie, rețeaua neurală convoluțională este un algoritm puternic și precis pentru detectarea obiectelor în imagini, cum ar fi semnele de circulație. Cu toate acestea, pentru a obține o precizie ridicată, este necesar un volum mare de date de antrenament și un puternic calcul pentru a procesa imaginile. De asemenea, este dificil de a generaliza modelele pre-antrenate pentru alte seturi de date sau alte aplicații, dar acestea sunt aspecte care pot fi îmbunătățite prin cercetarea continuă în domeniu.

B. Recurrent Neural Network(RNN)

Precizia unei RNN depinde de calitatea datelor de antrenament și de complexitatea rețelei. Cu cât datele de antrenament sunt mai bune și cu cât rețeaua este mai complexă, cu atât precizia va fi mai mare.

Avantajele unei RNN sunt:

- capacitatea de a analiza și a prezice secvențe de date, cum ar fi fluxul de imagini sau sunet
- permite analiza secvențelor de date în timp real
- poate detecta și clasifica semnele de circulație din fluxul de imagini

Dezavantajele unei RNN sunt:

- necesită un volum mare de date de antrenament pentru a obține o precizie ridicată
- poate fi dificil de proiectat și antrenat, mai ales pentru rețele mai complexe
- poate suferi de probleme cu "uitarea" informațiilor în lungi secvențe de date datorită faptului că informațiile din trecut sunt utilizate pentru a prezice informațiile viitoare
- necesită o puternică putere de calcul pentru a procesa secvențele de date

În concluzie, RNN sunt o clasă puternică de rețele neuronale care pot fi utilizate pentru a analiza și a prezice secvențe de date cum ar fi fluxul de imagini sau sunet. Acestea pot fi utilizate pentru detectarea semnelor de circulație din fluxul de imagini. Cu toate acestea, necesită un volum mare de date de antrenament și un puternic calcul pentru a procesa secvențele de date.

C. You Only Look Once(YOLO)

Precizia YOLO depinde de calitatea datelor de antrenament și de configurația rețelei. Cu cât datele de antrenament sunt mai bune și cu cât rețeaua este mai bine configurată, cu atât precizia va fi mai mare.

Avantajele YOLO sunt:

- Viteza sa, deoarece algoritmul poate analiza întreaga imagine într-un singur pas
- capacitatea de a detecta obiecte chiar și în condiții de iluminare scăzută sau în imagini cu rezoluție mare
- capacitatea de a detecta și clasifica semne de circulație și alte obiecte din imagini

Dezavantajele YOLO sunt:

- necesită un volum mare de date de antrenament pentru a obține o precizie ridicată
- poate fi dificil de configurat pentru unele aplicații sau seturi de date
- poate fi sensibil la schimbările în iluminare sau unghiul de captură al imaginii

În concluzie, YOLO este un algoritm puternic și precis pentru detectarea obiectelor în imagini, cum ar fi semnele de circulație. Cu toate acestea, necesită un volum mare de date de antrenament și poate fi dificil de configurat pentru unele aplicații sau seturi de date.

III. METODOLOGIE

În această secțiune vom cuprinde aspecte importante precum: detalii despre cum a fost realizat studiul, inclusiv tehnicile și instrumentele utilizate, precum și metoda de colectare a datelor.

IV. REZULTATE

Codul folosit în antrenarea rețelei cu date se bazează o rețea neurală convoluțională (CNN) și Keras, care poate fi utilizat pentru a clasifica diferite semne de circulație.

La început, importă toate modulele necesare pentru a funcționa, printează un mesaj "Imported" și setează unele variabile pentru dimensiunea imaginilor, numărul de canale și numărul de clase. Apoi, codul citește imaginile din folderele specifice care conțin fiecare semn de circulație și le adaugă într-un array. În acest caz, imaginile sunt citite din folderele "D:\—Facultate—PIM—Workspace—PIM Proiect—input—Train/0/", unde 0 este numărul clasei sau mai bine zis al folderului din care face parte un anumit semn de circulație, în total fiind 43 de astfel de foldere. Fiecare imagine este redimensionată la dimensiunea specificată în variabile (height, width) și apoi adăugată în array-ul "data" și "labels" care conține etichetele corespunzătoare fiecărei imaginii.

Dupa citirea imaginilor, array-urile sunt amestecate aleator pentru a evita bias-ul în setul de antrenare și de validare. Apoi, se ia 20% din imaginile citite pentru setul de validare și restul pentru setul de antrenare.

În continuare, se face one-hot encoding pentru etichetele setului de antrenare și de validare. Acest lucru este necesar

pentru ca rețeaua neurală să poată procesa etichetele ca o serie de valori binare, în loc de etichete numerice.

One-hot encoding este o metodă de preprocesare a datelor utilizată pentru a transforma una sau mai multe variabile categoricale în formate numerice, care pot fi utilizate de un algoritm de învățare automată. Aceasta este utilizată pentru a pregăti datele pentru modele de învățare automată care se bazează pe date numerice.

În One-hot encoding, fiecare valoare unică a unei variabile categoricale este reprezentată printr-un vector binar, unde elementul corespunzător valorii este setat la 1, iar celelalte elemente sunt setate la 0. Aceasta este utilă pentru modele de învățare automată deoarece majoritatea algoritmilor nu pot lucra cu variabile categoricale direct. One-hot encoding permite transformarea acestor variabile în formate numerice pe care algoritmi pot lucra cu ele.

După aceasta, se construiește modelul CNN, care conține mai multe straturi de filtre convoluționale, pooling și dropout. Fiecare strat de filtre convoluționale este utilizat pentru a detecta caracteristici specifice din imagini, cum ar fi linii, curbe sau forme. Straturile de pooling sunt utilizate pentru a reduce dimensiunea imaginilor, în timp ce straturile de dropout sunt utilizate pentru a preveni overfitting-ul.

Modelul este apoi compilat cu o pierdere categorical crossentropy și un optimizator Adam. Categorical crossentropy este o pierdere utilizată pentru probleme de clasificare multiplă, unde se țin cont mai multe clase. Adam este un optimizator popular pentru probleme de învățare automată.

După compilare, modelul este antrenat pe setul de antrenare și se evaluează pe setul de validare. În acest caz, se utilizează 20 de epoci pentru antrenarea modelului. Epoca reprezintă un ciclu complet prin toate exemplele din setul de antrenare. Rezultatele sunt salvate într-o variabilă numită "istorie" pentru a fi utilizate mai târziu pentru a afișa graficele de acuratețe și pierdere.

După antrenare, se afișează graficele de acuratețe și pierdere utilizând matplotlib. Acuratețea este măsurată ca procentaj de exemple clasificate corect din setul de validare. Rezultatele obținute de noi sunt următoarele:

Când vine vorba de testarea rețelei se încarcă un model de învățare automată salvat anterior folosind TensorFlow's "load-model" funcție. Modelul este încărcat din dosarul specificat și afișează un mesaj "Model loaded" pentru a indica faptul că încărcarea a avut loc cu succes. După încărcarea modelului, se încarcă setul de date de testare din dosarul specificat.

Apoi, se încarcă o imagine din dosarul specificat și se redimensionează la dimensiunea dorită (30x30) folosind OpenCV și PIL.

După aceea, se utilizează modelul încărcat pentru a face predicții pe baza imaginii încărcate, rezultatul este stocat într-o variabilă numită "pred". Se afișează rezultatul predicției și clasa corespunzătoare acesteia.

În final, se verifică precizia modelului prin intermediul funcției "accuracy-score" din sklearn și se afișează rezultatul care este un vector de dimensiune 43 care conține probabilitățile pentru fiecare clasă posibilă. Fiecare element din vector este

```
Epoch 1/20
981/981 [=====] - 39s 39ms/step - loss: 1.2046 - acc: 0.6552 - val_loss: 0.1755 - val_acc: 0.9515
Epoch 2/20
981/981 [=====] - 39s 40ms/step - loss: 0.2501 - acc: 0.9239 - val_loss: 0.0714 - val_acc: 0.9829
Epoch 3/20
981/981 [=====] - 40s 41ms/step - loss: 0.1507 - acc: 0.9535 - val_loss: 0.0516 - val_acc: 0.9867
Epoch 4/20
981/981 [=====] - 40s 41ms/step - loss: 0.1068 - acc: 0.9677 - val_loss: 0.0347 - val_acc: 0.9931
Epoch 5/20
981/981 [=====] - 41s 42ms/step - loss: 0.0899 - acc: 0.9730 - val_loss: 0.0489 - val_acc: 0.9876
Epoch 6/20
981/981 [=====] - 41s 42ms/step - loss: 0.0809 - acc: 0.9757 - val_loss: 0.0289 - val_acc: 0.9927
Epoch 7/20
981/981 [=====] - 41s 42ms/step - loss: 0.0644 - acc: 0.9793 - val_loss: 0.0241 - val_acc: 0.9939
Epoch 8/20
981/981 [=====] - 41s 42ms/step - loss: 0.0553 - acc: 0.9833 - val_loss: 0.0323 - val_acc: 0.9917
Epoch 9/20
981/981 [=====] - 39s 40ms/step - loss: 0.0583 - acc: 0.9820 - val_loss: 0.0235 - val_acc: 0.9939
Epoch 10/20
981/981 [=====] - 40s 40ms/step - loss: 0.0601 - acc: 0.9819 - val_loss: 0.0268 - val_acc: 0.9941
Epoch 11/20
981/981 [=====] - 40s 41ms/step - loss: 0.0479 - acc: 0.9856 - val_loss: 0.0250 - val_acc: 0.9940
Epoch 12/20
981/981 [=====] - 40s 40ms/step - loss: 0.0498 - acc: 0.9854 - val_loss: 0.0198 - val_acc: 0.9953
Epoch 13/20
981/981 [=====] - 40s 40ms/step - loss: 0.0407 - acc: 0.9855 - val_loss: 0.0250 - val_acc: 0.9945
Epoch 14/20
981/981 [=====] - 40s 40ms/step - loss: 0.0504 - acc: 0.9855 - val_loss: 0.0191 - val_acc: 0.9963
Epoch 15/20
981/981 [=====] - 39s 39ms/step - loss: 0.0436 - acc: 0.9877 - val_loss: 0.0179 - val_acc: 0.9960
Epoch 16/20
981/981 [=====] - 40s 41ms/step - loss: 0.0356 - acc: 0.9898 - val_loss: 0.0204 - val_acc: 0.9950
Epoch 17/20
981/981 [=====] - 39s 39ms/step - loss: 0.0444 - acc: 0.9873 - val_loss: 0.0171 - val_acc: 0.9948
Epoch 18/20
981/981 [=====] - 39s 40ms/step - loss: 0.0440 - acc: 0.9869 - val_loss: 0.0182 - val_acc: 0.9954
Epoch 19/20
981/981 [=====] - 40s 40ms/step - loss: 0.0396 - acc: 0.9898 - val_loss: 0.0158 - val_acc: 0.9967
Epoch 20/20
981/981 [=====] - 40s 41ms/step - loss: 0.0406 - acc: 0.9887 - val_loss: 0.0213 - val_acc: 0.9963
395/395 [=====] - 3s 7ms/step
```

o valoare între 0 și 1, care indică probabilitatea ca imaginea testată să aparțină clasei corespunzătoare.

Se observa ca elementul cu indice 1 are valoarea 1, ceea ce înseamnă ca modelul este foarte sigur că imaginea testată aparține clasei 1 și se afișează drept clasă corespunzătoare acestui rezultat, numărul 1, de aici rezultând faptul că modelul a făcut o predicție cu succes pentru această imagine.

```
Model loaded
Images loaded
1/1 [=====] - 0s 142ms/step
[[7.7563690e-32 1.0000000e+00 5.7935969e-20 9.3097669e-37 2.1357776e-23
 4.5632330e-24 1.2571004e-33 2.5920246e-28 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 1.1105002e-34 1.7665849e-33 3.9827063e-38
 3.7331735e-34 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 5.7605093e-38 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 1.7880659e-38 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 1.5407319e-32 0.0000000e+00 1.8196913e-38]]
1
```

Unele dintre limitările sau problemele posibile ale acestui cod ar putea fi:

- numărul de clase (43) și dimensiunile imaginilor (30x30) ar trebui să fie adaptate în funcție de setul de date specific utilizat.
- procentul de date de validare (0.2) ar putea fi ajustat pentru a se potrivi cu nevoile specifice ale setului de date.
- numărul de epoci de antrenare (20) ar putea fi ajustat pentru a se potrivi cu nevoile specifice ale setului de date.
- acest cod utilizează biblioteca keras, dar nu verifică dacă este instalată în sistem

- codul utilizează biblioteca tensorflow, dar nu verifică dacă este instalată în sistem sau dacă este configurată pentru utilizarea GPU

V. DISCUȚII

Rezultatele obținute arată că modelul a fost încărcat și imaginile au fost încărcate cu succes. În cele din urmă, se afișează un vector de predicții pentru o imagine de testare, unde fiecare valoare reprezintă probabilitatea ca imaginea să fie asociată cu o anumită clasă. În acest caz, se poate observa că imaginea de testare este în clasa 1 cu o probabilitate de 100%. Acest rezultat poate fi considerat bun, deoarece se poate vedea că modelul a avut o acuratețe de 99% la antrenarea epocilor. Acest lucru sugerează că modelul a învățat să recunoască cu acuratețe clasele din setul de date de antrenare și poate generaliza aceste învățăminte la date noi. Totuși, aceste rezultate ar trebui validate cu un set de date de testare independent pentru a verifica dacă acuratețea modelului este satisfăcătoare pentru aplicația specifică.

VI. CONCLUZII

Recomandările pe viitor pentru îmbunătățirea acestor date ar include:

- utilizarea unui set de date de antrenare și testare mai mare și mai divers pentru a asigura că modelul poate generaliza la o varietate de semne de circulație și condiții de iluminare
- utilizarea unui set de date care include semne de circulație deteriorate sau parțial acoperite pentru a testa modelul pentru cazurile în care semnele de circulație sunt dificil de citit
- utilizarea unui set de date care include semne de circulație din diferite țări sau regiuni pentru a asigura că modelul poate recunoaște semne de circulație din diferite zone geografice
- experimentarea cu modele de rețele neuronale diferite sau hiperparametri pentru a vedea dacă acestea pot îmbunătăți performanța modelului
- utilizarea tehnicilor de îmbunătățire a datelor, cum ar fi augmentarea datelor, pentru a genera mai multe exemple de antrenament și a reduce overfitting-ul

În concluzie, codul furnizat construiește un model de rețea neuronală convoluțională pentru clasificarea semnelor de circulație și antrenează acest model folosind o cantitate limitată de date de antrenare. Rezultatele obținute arată că acest model a avut o performanță bună în ceea ce privește acuratețea pe setul de date de antrenare și testare, dar pentru aplicații practice ar trebui să fie testat pe un set de date mai mare și mai divers și experimentat cu modele și hiperparametri diferiți. Recomandările pentru viitor ar trebui să fie luate în considerare pentru a îmbunătăți performanța modelului și a face recunoașterea semnelor de circulație din ce în ce mai precisă și robustă.

VII. REFERINȚE

REFERENCES

- [1] IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 52, NO. 5, SEPTEMBER 2003 1329, Road-Sign Detection and Tracking, Chiung-Yao Fang, Associate Member, IEEE, Sei-Wang Chen, Senior Member, IEEE, and Chiou-Shann Fuh, Member, IEEE
- [2] IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 8, NO. 2, JUNE 2007, Road-Sign Detection and Recognition Based on Support Vector Machines, Saturnino Maldonado-Bascón, Member, IEEE, Sergio Lafuente-Arroyo, Pedro Gil-Jiménez, Hilario Gómez-Moreno, Member, IEEE, and Francisco López-Ferreras
- [3] Road Sign Detection in Images: A Case Study R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, N. Paparoditis, Université Paris-Est, LEPSIS, INRETS-LCPC, 58 Boulevard Lefebvre, F-75015 Paris, France ERA27 LCPC, 11 Rue Jean Mentelin, B.P. 9, 67035 Strasbourg IGN, MATIS, 73 Avenue de Paris, 94165 Saint-Mandé, France
- [4] Fast Shape-based Road Sign Detection for a Driver Assistance System, Gareth Loy, Computer Vision and Active Perception Laboratory Royal Institute of Technology (KTH), Nick Barnes, Autonomous Systems and Sensor Technologies Program National ICT, Australia