Programi për Shkenca Kompjuterike dhe Inxhinieri - Master

# Big Data 2023 Lab Project
# Traffic Signs Recognition Project


**Mentori:**
Prof. Dipl.-Ing. Dr. techn. Yll Haxhimusa

**Studentja:**
BSc. Zana Shala Krasniqi


Qershor, 2023

# 1. Analizimi i datasetit

Dataseti GTSRB - German Traffic Sign Recognition Benchmark [1] është një set të dhënash shumë i përdorur si pikë referimi për detyrat e njohjes së shenjave të trafikut. Ai përmban imazhe të shenjave të trafikut të zakonshme në rrugët gjermane. Hulumtuesit dhe praktikantët përdorin datasetin GTSRB për të vlerësuar dhe krahasuar performancën e algoritmeve të njohjes së shenjave të trafikut.

Dataseti përmban më shumë se 50,000 imazhe të shenjave të ndryshme të trafikut. Madhësia e datasetit është rreth 300 MB.

Dataseti përbëhet nga 43 klasa të ndryshme të shenjave të trafikut. Çdo klasë paraqet një lloj të caktuar shenje të trafikut, si kufizimet e shpejtësisë, ndalimi, lejimi i kalimit etj. Shpërndarja e imazheve nëpër klasa mund të ndryshojë, me disa klasa që kanë më shumë imazhe se të tjerat. Ky heterogjenitet në shpërndarjen e klasave reflekton skenarët në botën reale ku disa lloje të shenjave të trafikut janë më të zakonshme se të tjerat. Dataseti shfaq variacione në kushte ndriçimi, kushte të motit, cilësi të imazhit dhe perspektivash. Kjo ndryshueshmëri ndihmon modelin të mësojë karakteristika të forta dhe të përgjithshme në imazhet reale të shenjave të trafikut.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0. | Speed limit (20km/h) | 11. | Right-of-way at the next intersection | 22. | Bumpy road | 33. | Turn right ahead |
| 1. | Speed limit (30km/h) | 12. | Priority road | 23. | Slippery road | 34. | Turn left ahead |
| 2. | Speed limit (50km/h) | 13. | Yield | 24. | Road narrows on the right | 35. | Ahead only |
| 3. | Speed limit (60km/h) | 14. | Stop | 25. | Road work | 36. | Go straight or right |
| 4. | Speed limit (70km/h) | 15. | No vehicles | 26. | Traffic signals | 37. | Go straight or left |
| 5. | Speed limit (80km/h) | 16. | Vehicles over 3.5 metric tons prohibited | 27. | Pedestrians | 38. | Keep right |
| 6. | End of speed limit (80km/h) | 17. | No entry | 28. | Children crossing | 39. | Keep left |
| 7. | Speed limit (100km/h) | 18. | General caution | 29. | Bicycles crossing | 40. | Roundabout mandatory |
| 8. | Speed limit (120km/h) | 19. | Dangerous curve to the left | 30. | Beware of ice/snow | 41. | End of no passing |
| 9. | No passing | 20. | Dangerous curve to the right | 31. | Wild animals crossing | 42. | End of no passing by vehicles over 3.5 tons |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10. | No passing for vehicles over 3.5 metric tons | 21. | Double curve | 32. | End of all speed and passing limits | | |

*Tabela 1. Klasat e datasetit GTSRB*

Dataseti ndahet në një dosje trajnimi dhe një dosje testimi. Dosja e trajnimit përmban imazhe për trajnimin e modelit, ndërsa dosja e testimit përdoret për të vlerësuar performancën e modelit. Është e rëndësishme të ndajmë datasetin në mënyrë të saktë për të siguruar vlerësim të saktë dhe për të evituar mbivendosjen.

Imazhet në dataset zakonisht ruhen në një format të zakonshëm të imazhit si JPEG, PNG ose BMP. Këto formate mbështeten gjerësisht nga libraritë për procesimin e imazheve dhe mund të lexohen dhe përpunohen lehtësisht.

Para se të fillojë trajnimi i modelit, është thelbësore të paraprocesohen të dhënat në dataset. Kjo mund të përfshijë veprime si ndryshimi i madhësisë së imazheve në një madhësi të caktuar, normalizimi i vlerave të pikselave dhe zgjerimi i të dhënave duke aplikuar transformime si rotacioni, skalimi dhe përmbysja. Paraprocesimi ndihmon në përmirësimin e performancës së modelit dhe përgjithësimin e tij.

Duke qenë se disa klasa në dataset mund të kenë më pak imazhe, teknikat e rritjes së të dhënave mund të aplikohen për të rritur artificialisht madhësinë e datasetit. Teknikat si rotacioni, translacioni, zmadhimi dhe shtimi i zhurmave mund të përdoren për të krijuar mostra trajnimi të reja, duke ulur kështu rrezikun e mbivendosjes dhe duke përmirësuar aftësinë e modelit për të trajtuar variacionet në të dhënat reale.

Për të vlerësuar performancën e modelit të trajnuar, mund të përdoren metrika të vlerësimit si saktësia, precizion, rikthim (recall) dhe skora F1. Këto metrika ofrojnë njohuri në aftësinë e modelit për të klasifikuar saktë shenjat e trafikut dhe për të trajtuar klasa të ndryshme.

Duke studiuar dhe analizuar datasetin GTSRB, fitojmë njohuri më të mirë për karakteristikat e tij, që mund të ndihmojnë në projektimin e një arkitekture efektive të modelit, zgjedhjen e teknikave të përshtatshme të paraprocesimit dhe vlerësimin e performancës së modelit me saktësi.

## 2. Modeli CNN (Convolutional Neural Network)

CNN është një lloj modeli i Deep Learning i cili është i dizajnuar posaçërisht për të trajtuar të dhënat e strukturuara e të organizuara në formën e rrjetëzave, siç janë imazhet. Ky model është shumë efektiv në detyrat si klasifikimi i imazheve, zbulimi i objekteve dhe segmentimi i imazheve.

Modeli CNN për njohjen e shenjave të trafikut duke përdorur datasetin GTSRB (German Traffic Sign Recognition Benchmark) përdor një rrjet të thellë me konvolucione për të klasifikuar shenjat e trafikut në kategori të ndryshme. Para zgjedhjes së modelit përfundimtar është eksperimentuar me tri CNN modele të ndryshme.

Dy modelet kanë arkitektura të ndryshme, të cilat mund të çojnë në ndryshime në performancën dhe aftësitë e tyre. Më poshtë është dhënë një përmbledhje e ndryshimeve kryesore midis modeleve.

2.1 CNN modeli I

Modeli i dhënë është një rrjet nervor konvolucional (CNN) me arkitekturën e mëposhtme. Fillimisht, për implementimin e këtij modeli janë përdorur libraritë: Pandas [2], TensorFlow [3], Sklearn [4], Matplotlib [5], Numpy [6] dhe PIL [7]. Forma e hyrjes së modelit është (Asnjë, 32, 32, 3), ku 3 tregon për kanalizimin RGB të imazheve të futura me madhësinë 32x32 piksel. Modeli fillon me një shtresë Conv2D me 32 filtra dhe madhësinë e kernelit (3, 3), e cila përdor aktivizimin ReLU për të nxjerrë veçoritë e para. Pas kësaj, vjen një shtresë MaxPooling2D me madhësinë (2, 2) që redukton dimensionin e hartave të veçorive. Pastaj, përsëriten një shtresë Conv2D me 64 filtra dhe një shtresë MaxPooling2D për të vazhduar zvogëlimin e dimensionit të veçorive. Një shtresë tjetër Conv2D me 128 filtra ndjek, pas së cilës vjen një shtresë MaxPooling2D tjetër. Pas kësaj, hartat e veçorive 2D shndërrohen në një vektor 1D përmes shtresës Flatten. Një shtresë e dendur me 128 njësi dhe aktivizim ReLU ndjek, e cila ndihmon në zbulimin e lidhjeve komplekse midis veçorëve. Përfundimisht, një shtresë e dendur me 43 njësi, që përkon me numrin e klasave, përdor aktivizimin softmax për të prodhuar parashikime të saktësisë së klasifikimit. Numri total i parametrave të trajnueshëm në model është 164,459.
Gjatë trajnimit, modeli përpilohet me optimizuesin Adam, duke përdorur ndër-entropinë kategorike si funksion të humbjes dhe saktësinë si metrikë. Më pas trajnohet për 10 epoka me një madhësi grupi prej 32.
Pas trajnimit, modeli vlerësohet në grupin e testit. Humbja dhe saktësia e testit llogariten dhe printohen.
Më në fund, një matricë konfuzioni llogaritet duke përdorur etiketat e vërteta dhe etiketat e parashikuara në grupin e testimit.

2.1.1 Rezultatet e modelit I

Në rezultatet e dhëna, modeli arrin një humbje testimi prej 0,4608 dhe një saktësi testimi prej 0,0066. Matrica e konfuzionit tregon shpërndarjen e etiketave të parashikuara nëpër etiketa të ndryshme të vërteta.

---

```
[[ 82 756 702  ...  72  61 106]
 [  0   1   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 ...
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]]
```

---

Numrat në rreshtat e matricës tregojnë numrin e pranisë së etiketave të parashikuara për secilën klasë të vërtetë. Në rreshtin e parë, kemi 82 raste ku klasa e parë është parashikuar saktë. Në rreshtin e dytë, kemi vetëm 1 rast ku klasa e dytë është parashikuar saktë. Vlerat e tjera në rreshtat tregojnë praninë e parashikimeve për klasat e tjera. Numrat në kolonat e matricës tregojnë numrin e pranisë së etiketave të vërtetë për secilën klasë të parashikuar. Në kolonën e

parë, kemi 82 raste ku klasa e parë është e pranishme. Në kolonën e dytë, kemi 756 raste ku klasa e dytë është e pranishme. Vlerat e tjera në kolona tregojnë praninë e etiketave të vërteta për klasat e tjera të parashikuara. Vlerat në diagonale tregojnë numrin e pranisë së etiketave të parashikuara saktë për secilën klasë të vërtetë. Në rastin tonë, vlerat diagonale janë në fillim të matricës dhe tregojnë numrin e parashikimeve të saktë për secilën klasë. Vlerat që janë jashtë diagonales tregojnë gabimet e parashikimeve për secilën klasë të vërtetë. Vlerat të cilat janë të mëdha në krahasim me vlerat e tjera mund të tregojnë vështirësi në parashikimin e këtyre klasave.
Në anën tjetër, saktësia e testit prej 0,0066 duket jashtëzakonisht e ulët, gjë që sugjeron se mund të ketë një problem me modelin ose procesin e vlerësimit. Prandaj, është vazhduar eksperimentimi me modele të tjera.

## 2.3 CNN modeli II

Modeli final është një rrjet konvolucional i thjeshtë me tre blloqe konvolucionale dhe disa shtresa të plotësisë (fully connected layers) në fund.
Rrjeti fillon me shtresën e parë konvolucionale, e cila ka 32 filtra me një madhësi (kernel size) 3x3. Pas konvolucionit, aktivizohet funksioni ReLU. Pastaj, përdoret MaxPooling me madhësinë 2x2 për të zvogëluar dimensionin e imazhit.
Rrjeta përdor tre blloqe konvolucionale të ngjashme. Çdo bllok ka një shtresë konvolucionale dhe një shtresë MaxPooling. Blloqet konvolucionale kanë respektivisht 64, 64 dhe 64 filtra me një madhësi 3x3.
Pas tre blloqeve konvolucionale, përdoret shtresa Flatten për të shndërruar imazhet 2D në vektorë 1D, duke i përgatitur ato për shtresat plotësuese.
Rrjeti përdor dy shtresa plotësuese me 64 dhe 43 neuronë. Shtresa e fundit përdor funksionin e aktivizimit Softmax për të përcaktuar probabilitetin e klasifikimit të imazheve në 43 klasa të ndryshme.
Modeli është kompiluar me optimizuesin Adam, funksionin humbje (loss function) "categorical_crossentropy" për klasifikim me shumë klasa dhe metrikën e saktësisë (accuracy) për vlerësimin e performancës.
Modeli është trajnuar nëpërmjet funksionit fit() duke përdorur të dhënat e trajnimit për një numër të caktuar epokash (epochs) dhe një madhësi të grupit (batch size) 32. Gjithashtu, është përdorur një pjesë e dhënash të validimit për të monitoruar performancën e modelit gjatë trajnimit.
Pas trajnimit, modeli është vlerësuar nëpërmjet funksionit evaluate() duke përdorur të dhënat e testimit. Kështu është llogaritur gabimi i testimit (test loss) dhe saktësia e testimit (test accuracy) të modelit.
Modeli është përdorur për të bërë parashikime mbi imazhet e testimit duke përdorur funksionin predict(). Rezultati i parashikimeve është një matricë e probabiliteteve për secilën klasë.
Matrica e probabiliteteve është konvertuar në etiketa të parashikuara të klasave duke përdorur argmax për secilin parashikim të imazhit të testimit.

Etiketat e testimit të koduara me one-hot janë konvertuar në etiketat e tyre origjinale të klasave duke përdorur argmax. Shfaqja e imazheve të testimit të klasifikuara: Një vizualizim është krijuar duke shfaqur imazhet e testimit të klasifikuara së bashku me etiketat e tyre të vërteta dhe etiketat e parashikuara.
Është llogaritur matrica e konfuzionit duke krahasuar etiketat e vërteta të klasave me etiketat e parashikuara të klasave. Në vijim janë diskutuar rezultatet e modelit III.

### 2.1.1 Diskutim i rezultateve të modelit II

Rezultatet e mëposhtme tregojnë performancën e modelit gjatë trajnimit. Në fillim, saktësia e trajnimit është rreth 71.43%, por rritet gradualisht gjatë epokave të tjera. Gjatë 10 epokave, saktësia e trajnimit arrin deri në 99.50%. Kjo tregon se modeli është në gjendje të mësojë dhe të adaptohet ndaj të dhënave të trajnimit.
Saktësia e validimit gjithashtu rritet nga 95.00% në fillim deri në 99.31% në epokën e fundit. Kjo tregon se modeli nuk po shkakton overfitting (mos-adaptimi i tepërt) ndaj të dhënave të trajnimit, pasi performon mjaft mirë edhe në të dhënat e validimit.
Saktësia e testimit është 95.13%, e cila është shumë pranë saktësisë së validimit. Kjo tregon se modeli ka aftësi të mirë për të klasifikuar imazhet e reja që nuk janë përdorur gjatë trajnimit.

```
Epoch 1/10
981/981 [==============================] - 44s 43ms/step - loss: 1.3093 - accuracy: 0.6245 - val_loss: 0.2951 - val_a
ccuracy: 0.9193
Epoch 2/10
981/981 [==============================] - 51s 52ms/step - loss: 0.1979 - accuracy: 0.9468 - val_loss: 0.1657 - val_a
ccuracy: 0.9552
Epoch 3/10
981/981 [==============================] - 45s 46ms/step - loss: 0.1006 - accuracy: 0.9744 - val_loss: 0.0859 - val_a
ccuracy: 0.9767
Epoch 4/10
981/981 [==============================] - 50s 51ms/step - loss: 0.0624 - accuracy: 0.9841 - val_loss: 0.0589 - val_a
ccuracy: 0.9839
Epoch 5/10
981/981 [==============================] - 49s 50ms/step - loss: 0.0450 - accuracy: 0.9878 - val_loss: 0.0526 - val_a
ccuracy: 0.9841
Epoch 6/10
981/981 [==============================] - 46s 47ms/step - loss: 0.0335 - accuracy: 0.9908 - val_loss: 0.0395 - val_a
ccuracy: 0.9899
Epoch 7/10
981/981 [==============================] - 44s 45ms/step - loss: 0.0295 - accuracy: 0.9918 - val_loss: 0.0479 - val_a
ccuracy: 0.9874
Epoch 8/10
981/981 [==============================] - 45s 46ms/step - loss: 0.0256 - accuracy: 0.9928 - val_loss: 0.0609 - val_a
ccuracy: 0.9833
Epoch 9/10
981/981 [==============================] - 57s 58ms/step - loss: 0.0187 - accuracy: 0.9952 - val_loss: 0.0535 - val_a
ccuracy: 0.9850
Epoch 10/10
981/981 [==============================] - 57s 58ms/step - loss: 0.0166 - accuracy: 0.9952 - val_loss: 0.0346 - val_a
ccuracy: 0.9912
395/395 [==============================] - 5s 13ms/step - loss: 0.3800 - accuracy: 0.9414
Gabimi i testimit: 0.3800
Saktësia e testimit: 0.9414
395/395 [==============================] - 5s 13ms/step
```

*Figura 1. Saktësia dhe gabimet e trajnimit dhe validimit përgjatë 10 epokave*

Matrica e konfuzionit tregon se si janë klasifikuar imazhet në testimin e modelit për secilën klasë. Vlerat në matricë tregojnë numrin e rasteve kur një imazh është klasifikuar si klasa e parashikuar (rrjeshta) në krahasim me klasën e vërtetë (kolona).

Diskutimi i matricës së konfuzionit mund të përfshijë këto aspekte: Në klasën e parashikuar 0, janë klasifikuar saktë 45 imazhe. Në klasën e parashikuar 1, janë klasifikuar saktë 699 imazhe. Ka disa raste gabimi, ku 6 imazhe janë klasifikuar gabimisht në klasën 2. Në klasën e parashikuar 2, janë klasifikuar saktë 746 imazhe. Ka disa raste gabimi, ku 3 imazhe janë klasifikuar gabimisht në klasën 1 dhe një imazh është klasifikuar gabimisht në klasën 4. Procesi vazhdon në të njëjtën mënyrë për klasat e tjera, ku matrica tregon numrin e imazheve të klasifikuara saktë dhe rastet e gabimeve të ndryshme. Në përgjithësi, matrica e konfuzionit tregon performancën e modelit në përputhje me klasat individuale. Duke i analizuar vlerat në matricë, mund të identifikohen klasat ku modeli ka performancë të mirë dhe klasat ku ka tendencë të klasifikojë gabimisht. Kjo informacion mund të përdoret për të përmirësuar modelin, duke i dhënë më shumë peshë ose fokus në klasat me gabime të shumta ose duke marrë masa të tjera për të rritur saktësinë e modelit.

---

```
[[ 50   3   0 ...   0   0   0]
 [  1 698   7 ...   3   0   0]
 [  0  23 725 ...   0   0   0]
 ...
 [  0   0   0 ...  87   0   0]
 [  0   0   0 ...   0  42   0]
 [  0   0   0 ...   0   0  90]]
```

---

Bazuar në matricën e konfuzionit të dhën më sipër, ka disa elemente që mund të vërehen: Vlerat në diagonale (nga këndi sipër majtë në këndin poshtë djathtas) tregojnë numrin e pranisë së etiketave të parashikuara saktë për secilën klasë të vërtetë. Në rastin tonë, vlerat diagonale janë të ndryshme nga zero, që tregon se modeli ka bërë parashikime saktë për disa klasa. Vlerat jashtë diagonales tregojnë numrin e gabimeve të parashikimeve për secilën klasë të vërtetë. Nëse ka vlera të mëdha jashtë diagonales për një klasë të caktuar, kjo mund të tregojë se modeli ka vështirësi në identifikimin e saktë të kësaj klase. Vlerat e rreshtave dhe kolonave: Shikoni vlerat në rreshtat dhe kolonat individuale për të kuptuar më shumë. Për shembull, në rreshtin e parë, numri i parashikimeve të saktë për klasën e parë është 50, ndërsa në rreshtin e dytë, numri i parashikimeve të saktë për klasën e dytë është 698. Kjo tregon dallimet në performancën e parashikimeve midis klasave. Vlerat zero tregojnë se nuk ka gabime të parashikimeve për disa kombinime të klasave. Kjo mund të tregojë se modeli ka performuar shumë mirë për këto kombinime të caktuara të klasave.

Për të përcaktuar arsyet pas klasifikimeve gabim, mund të merren parasysh disa faktorë. Disa shenja rrugore mund të kenë ngjashmëri vizuale, duke bërë të vështirë dallimin për modelin. Për shembull, shenjat me forma ose ngjyra të ngjashme mund të jenë më të prona për gabime klasifikimi.

Nëse disa klasa kanë një numër të kufizuar të shembujve trajnimi ose nëse shpërndarja e klasave në të dhënat e trajnimit është e padrejtë, modeli mund të mos ketë mësuar mjaftueshëm për ato klasa specifike, duke rezultuar në saktësi të ulët.

Shenjat rrugore mund të dalin në pamje të ndryshme për shkak të faktorëve si kushtet e ndriçimit, mbyllje të pjesëve të shenjave ose këndesh të ndryshme të kapjes. Nëse të dhënat e trajnimit nuk kapin mjaftueshëm variacione ose nëse të dhënat e testimit përmbajnë instance që ndryshojnë ndjeshëm nga të dhënat e trajnimit, modeli mund të hasë vështirësi në të përgjithshmen mirë.

Disa shenja rrugore mund të përmbajnë detaje të ndërlikuara ose simbole komplekse që janë të vështira për tu njohur edhe për njerëzit. Nëse modeli nuk i kap këto detaje në mënyrë efektive ose nëse kapaciteti i modelit është i kufizuar, kjo mund të çojë në klasifikime gabim.

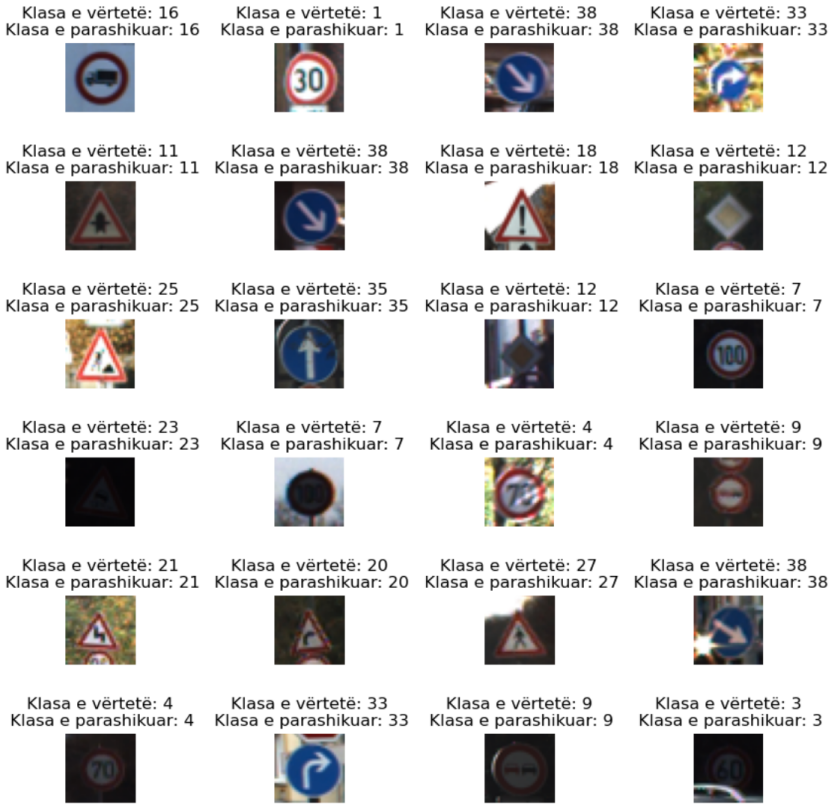


*Figura 2. Klasifikimi i fotove në bazë të klasave*

## 3. Përfundim

Në vijim do i jap sugjerimet e mia përfundimtare për kodin e dhënë, si dhe ide për përmirësimet e mundshme dhe algoritmet e tjerë që mund të performojnë më mirë. Gjithësisht, kodi duket të jetë i strukturuar mirë dhe ndjek praktikat më të mira për trajnimin e një rrjeti konvolucional (CNN) në të dhëna imazhi. Arkitektura e modelit përfshin shumë shtresa konvolucionale të ndjekura nga shtresa të maksimizimit, që çojnë në një shtresë të plotësisë së lidhur dhe një shtresë output. Kjo është një arkitekturë e përdorur shpesh për detyrat e klasifikimit të imazheve dhe duhet të funksionojë mirë. Kodi trajton ngarkimin e të dhënave, paraprosesimin, trajnimin e modelit dhe vlerësimin në mënyrë efektive. Një përmirësim i mundshëm është të përfshihen teknika të zgjerimit të të dhënave gjatë trajnimit. Zgjerimi i të dhënave mund të ndihmojë në rritjen e diversitetit dhe madhësisë së grupit të trajnimit, duke çuar në përmirësim të përgjithshëm dhe performancën e modelit. Mund të konsiderohet implementimi i ndalimit të hershëm gjatë trajnimit për të parandaluar mbivendosjen. Ndalimi i hershëm lejon të monitorohet humbja e vlerësimit dhe të ndalohet trajnimi kur performanca fillon të degradohet. Duke eksperimentuar me hiperparametra të ndryshëm, si shkalla e mësimit, madhësia e grupit dhe numri i epokave, mund të përmirësohet performanca e modelit. Mund të kryhet një kërkim i hiperparametrave duke përdorur teknika si kërkimi i rrjetit ose kërkimi i rastësishëm. Përveç CNN-ve, mund të eksplorohen arkitektura të tjera si ResNet, DenseNet, apo Inception, të cilat kanë treguar performancë të fortë në shumë detyra të klasifikimit të imazheve. Këto arkitektura shpesh përdorin lidhje të kthyera, strategji të avancuara të maksimizimit, ose ekstraktim të karakteristikave në shkallë të ndryshme për të përmirësuar saktësinë. Është gjithashtu e vlefshme të konsiderohen metoda të mbledhjes së modeleve, si kombinimi i parashikimeve nga modele të ndryshme apo implementimi i teknikave si bagging ose boosting. Metodat e mbledhjes së modeleve shpesh sjellin përmirësim të performancës dhe qëndrueshmërisë. Përveç qasjeve të thellësisë së mëtejshme, mund të eksploroni algoritme të tradicionshme të Machine Learning si Support Vector Machines (SVM), Random Forests, apo Gradient Boosting, veçanërisht nëse ka të dhëna trajnimi të kufizuara. Këto algoritme mund të jenë efektive në ekstraktimin e tipareve të ndërtuara dorazi nga imazhet. Së fundi, analiza e mostrave të klasifikuara gabimisht dhe kuptimi i modeleve ose karakteristikave që çojnë në parashikime të pasakta mund të ofrojë idetë për përmirësime të mundshme ose inxhinieri të tipareve.

## 4. Referencat

[1] MYKOLA (2018). GTSRB - German Traffic Sign Recognition Benchmark. https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

[2] McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56). https://pandas.pydata.org/docs/getting_started/index.html

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.. https://www.tensorflow.org/

[4] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.. https://scikit-learn.org/stable/

[5] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.. https://matplotlib.org/

[6] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. . https://numpy.org/doc/

[7] Umesh, P. (2012). Image Processing in Python. *CSI Communications*, *23*.https://pillow.readthedocs.io/en/stable/