UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Identificarea si monitorizarea factorilor de risc interni si externi pentru siguranta traficului auto - INOVSAFE PN-III-P1-1.1-PD-2021-0247

Echipa de cercetare: Răzvan Itu, Radu Dănescu

ETAPA 3 (01/01/2024 - 31/03/2024) - Testare, validare si integrare sistem (O1, O2, O3)

Introducere - rezumatul etapei

Etapa 3 a avut două activități majore:
1 - Migrarea sistemului cadru de procesare pe sisteme mobile de procesare (O1, O2) - partea 2
2 - Diseminare rezultate (O1, O2, O3) - partea 3

1 - Migrarea sistemului cadru de procesare pe sisteme mobile de procesare (O1, O2) - partea 2

Pentru migrrarea soluției propuse pe platforma Nvidia Jetson am folosit software-ul TensorRT oferit de Nvidia, în detrimentul folosirii TFLite menționat anterior în etapa precedentă. TensorRT este o unealtă pentru optimizarea și implementarea modelelor de “deep learning” pe Nvidia Jetson, în special în medii cu resurse limitate, cum ar fi sistemele de tip embedded. Importanța sa constă în capacitatea sa de a accelera semnificativ viteza de inferență în timp ce minimizează amprenta de memorie și consumul de energie.

Această optimizare este vitală pentru aplicații în timp real, cum ar fi soluția propusă și dezvoltată în cadrul acestui proiect (estimarea vitezei vehiculului în mașinile cu conducere autonomă), unde luarea deciziilor în timp util pe baza datelor de la senzori este crucială pentru asigurarea siguranței și eficienței.

Conversia modelului folosind TensorRT se face prin următorul cod:

```
import tensorflow.contrib.tensorrt as trt

trt_graph = trt.create_inference_graph(
   input_graph_def=frozen_graph,
   outputs=output_names,
   max_batch_size=1,
   max_workspace_size_bytes=1 << 25,
   precision_mode='FP16',
   minimum_segment_size=50
)
```

Acest cod ilustrează modul în care am folosit TensorRT pentru optimizarea și exportul modelului dezvoltat folosind TensorFlow/Keras. Prin crearea unui model de inferență optimizat, se realizează o performanță îmbunătățită a inferenței în timp real pe dispozitivele Nvidia Jetson. TensorRT optimizează modelul pentru precizia datelor și dimensiunea lotului, reducând consumul de memorie și accelerând timpul de inferență. Cei mai importanți parametri sunt următorii:

- *max_workspace_size_bytes*: parametru care stabilește dimensiunea maximă a spațiului de lucru în octeți pe care TensorRT poate să-l folosească pentru optimizare și l-am setat la 1 << 25 (echivalent cu 32 MB);

- *precision_mode*: parametru care specifică modul de precizie pentru optimizare, setat la 'FP16', indicând faptul că TensorRT va utiliza precizia cu virgulă mobilă de 16 biți pentru calcul, ceea ce poate reduce utilizarea memoriei și îmbunătăți performanța în comparație cu precizia completă (virgulă mobilă de 32 de biți);

Salvarea modelului se face folosind:

```
graph_io.write_graph(trt_graph, "./model/", "trt_graph.pb", as_text=False)
```

De asemenea, se poate folosi și următoarea variantă pentru conversie TensorRT:

```
converter = trt.TrtGraphConverterV2(
  input_saved_model_dir=SAVED_MODEL_DIR,
  precision_mode=trt.TrtPrecisionMode.FP16
)
converter.convert()

converter.save(converted_model_path)
```

Rezultatul conversiei modelului propus de mine:

```
################################################################################
TensorRT unsupported/non-converted OP Report:
 - NoOp -> 2x
 - Placeholder -> 2x
 - Identity -> 1x
 - Pack -> 1x
 - Reshape -> 1x
 - Shape -> 1x
 - StridedSlice -> 1x
--------------------------------------------------------------------------------
 - Total nonconverted OPs: 9
 - Total nonconverted OP Types: 7
For more information see https://docs.nvidia.com/deeplearning/frameworks/tf-trt-user-guide/index.html#supported-ops.
################################################################################

2024-02-16 11:51:12.147701: W tensorflow/compiler/tf2tensorrt/segment/segment.cc:1298] The environment variable TF_TRT_MAX_ALLOWED_ENGINES=20 has no effect since there are only 1 TRT Engines with  at least minimum_segment_size=3 nodes.
2024-02-16 11:51:12.147726: I tensorflow/compiler/tf2tensorrt/convert/convert_graph.cc:802] Number of TensorRT candidate segments: 1
```

2024-02-16 11:51:12.172584: I tensorflow/compiler/tf2tensorrt/convert/convert_graph.cc:916] Replaced segment 0 consisting of 48 nodes by TRTEngineOp_000_000.
TRTEngineOP Name Device # Nodes # Inputs # Outputs Input DTypes Output Dtypes Input Shapes Output Shapes
==========
TRTEngineOp_000_000 device:GPU:0 48 2 1 ['float32', 'f ... ['float32'] [[-1, 300, 300 ... [[-1, 3]]

- BiasAdd: 9x
- ConcatV2: 1x
- Const: 20x
- Conv2D: 5x
- MatMul: 4x
- Relu: 8x
- Reshape: 1x

==========
[*] Total number of TensorRT engines: 1
[*] % of OPs Converted: 82.76% [48/58]

Astfel, modelul exportat se poate folosi pe platforma Nvidia Jetson, pentru testare am folosit varianta Nano care este disponibilă în laboratorul de cercetare.

Testarea modelului pe platforma Jetson se face folosind limbajul Python, iar o parte din codul necesar pentru a executa o predicție/inferență a modelului este următorul:

```
tf_sess = tf.Session(config=tf_config)
tf.import_graph_def(trt_graph, name='')

pred = tf_sess.run(output_tensor, feed_dict)
```

Implementarea modelului de inferență TensorRT pe Nvidia Jetson permite utilizarea eficientă a capacităților hardware-ului pentru sarcini sensibile la timp, precum estimarea vitezei vehiculului în scenarii de trafic rutier.

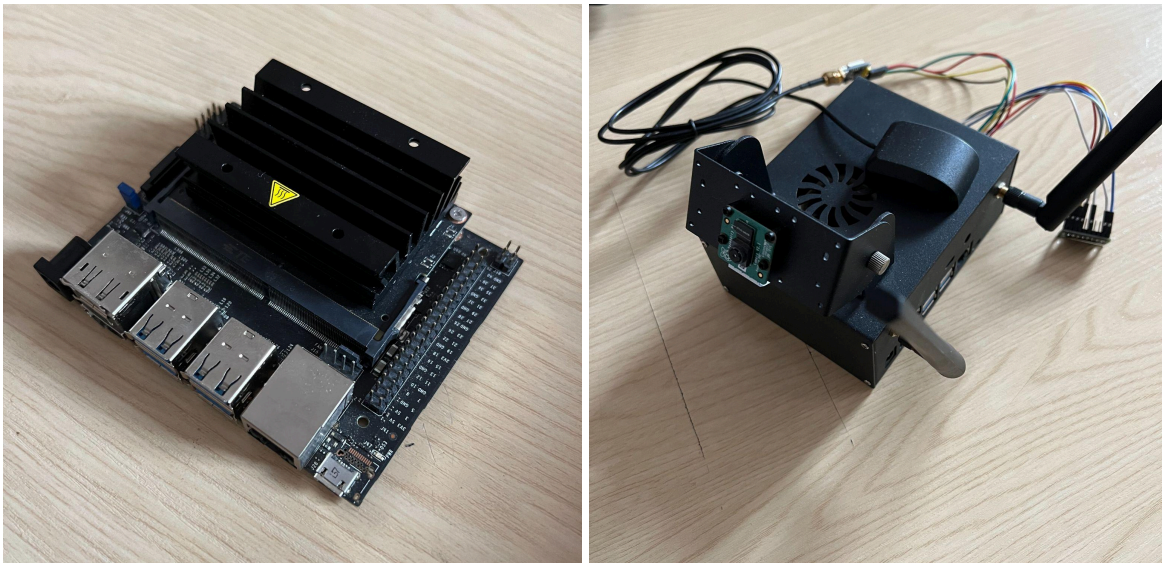

Fig. 1. Nvidia Jetson Nano (stânga), Nvidia Jetson Nano într-o carcasă și un modul de cameră video atașat (dreapta).

Testarea soluției s-a făcut folosind o cameră video conectată la Nvidia Jetson Nano (figura 1 dreapta), de asemenea prin porturile USB se poate adăga orice webcam.

2 - Diseminare rezultate (O1, O2, O3) - partea 3

Rezultatele au fost diseminate într-un jurnal indexat ISI (a fost aprobată lucrarea trimisă în urma modificărilor și ajustărilor cerute de recenzori și editori):

1. Razvan Itu, Radu Danescu, "Fully Convolutional Neural Network for Vehicle Speed and Emergency-Brake Prediction", Sensors, Vol. 24, No. 1, 2024, Art. No. 212.

Diseminare rezultate (tot proiectul)

Conferințe (indexate ISI):
1. Razvan Itu, Radu Danescu, "On-Board Estimation of Vehicle Speed and the Need of Braking Using Convolutional Neural Networks”, In Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics - Volume 1 (ICINCO), 2023, SciTePress, pp. 600-607.

2. Razvan Itu, Radu Danescu, “Predicting Emergency Braking in Vehicles Using a CNN with Sequential Image and Velocity Data”, 2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP 2023), 2023, pp. 41-47.

Jurnale ISI:
1. Razvan Itu, Radu Danescu, "Part-Based Obstacle Detection Using a Multiple Output Neural Network", Sensors, Vol. 22, No. 12, 2022, Art. No. 4312.

2. Razvan Itu, Radu Danescu, "Fully Convolutional Neural Network for Vehicle Speed and Emergency-Brake Prediction", Sensors, Vol. 24, No. 1, 2024, Art. No. 212.

Rezumat obiective

Obiectivele prezentate în propunerea de proiect sunt următoarele:

O1. Crearea unei platforme hardware și software pentru achiziția, vizualizarea, organizarea și procesarea datelor de imagine și senzoriale (en. “Creating a hardware and software platform for the image and sensorial data acquisition, visualization, organization and processing”).

O2. Extragerea caracteristicilor relevante ale scenei de trafic din imagini monoculare (en. “Extracting the relevant traffic scene features from monocular images”).

O3. Analiza stării ego-vehiculului și a comportamentului șoferului din imagini de trafic și datele vehiculului (en. “Analyzing ego-vehicle state and driver behaviour from traffic imagery and vehicle data”).

Gradul de realizare al obiectivelor:
Obiectivul 1 este realizat complet.
Obiectivul 2 este realizat complet.
Obiectivul 3 este realizat complet.

Indicatori de rezultat:

Articole ISI: 2 articol de jurnal aprobate și publicate(ISI Q1 - zona roșie)
Articole conferințe ISI: 2 articole susținute în conferințe, publicate și indexate

Au fost publicate 2 articole în jurnale indexate ISI, câte unul pentru fiecare an al proiectului, dar au fost prezentate și două lucrări în cadrul unor conferințe internaționale (indexate ISI).

Impactul rezultatelor obținute:

Rezultatele obținute vor avea impact în dezvoltarea și implementarea de sisteme de conducere autonomă a vehiculelor sau platformelor robotice, cu cel mai mare impact fiind legat de îmbunătățirea siguranței prin predicția vitezei vehiculului propriu direct din imaginile de trafic rutier, soluție care poate fi extinsă și pentru alte scenarii (exemplu robot autonom dintr-o fabrică). Astfel, cel mai semnificativ rezultat se referă la predicția vitezei proprii direct din imagini, ceea ce contribuie la estimarea stării curente a vehiculului propriu.

Rezumat activități:

1 - Migrarea sistemului cadru de procesare pe sisteme mobile de procesare (O1, O2) - partea 2
În această etapă, am implementat o variantă optimizată a modelului propus de mine. Această variantă este capabilă să execute predicții (inferență) pe platforme cu hardware redus de tip embedded, iar implementarea am făcut-o pe Nvidia Jetson Nano.

2 - Diseminare rezultate (O1, O2, O3) - partea 3
În această etapă, am finalizat publicarea unei lucrări într-un jurnal indexat ISI.

Concluzii

În această etapă am reușit implementarea cu succes migrarea și testarea soluției propuse pe o platformă mobilă (Nvidia Jetson). Totodată, am reușit să modific lucrarea trimisă și să obțin publicarea acestia într-o revistă indexata ISI. Am atins obiectivele propuse în cadrul proiectului de cercetare și am atins indicatorii de rezultat propuși.

Director proiect,

Sl.Dr.Ing. Razvan Itu