Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE23Radojcin

КОМПАРАТИВНА АНАЛИЗА ТЕХНОЛОГИЈА ЗА МАШИНСКО УЧЕЊЕ НА ИВИЦИ ПРИМЕНОМ NVIDIA JETSON TX2 УРЕЂАЈА

COMPARATIVE ANALYSIS OF MACHINE LEARNING AT THE EDGE TECHNOLOGIES USING THE NVIDIA JETSON TX2

Милош Радојчин, *Факултет техничких наука, Нови Сад*

Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА

Кратак садржај – *У овом раду дате су теоријске основе машинског учења на ивици и обраде тока података, значење термина Machine Learning Operations и опис Nvidia Jetson TX2 уређаја. Затим су анализиране технологије за машинско учење на ивици и дате су њихове компаративне анализе. Неке од ових технологија су примењене на решење демографске аналитике коришћењем Nvidia Jetson TX2 уређаја.*

Кључне речи: *Машинско учење, технологије, компаративна анализа, Nvidia Jetson TX2*

Abstract – *This paper presents the theoretical foundations of machine learning at the edge and data flow processing, the meaning of the term Machine Learning Operations and a description of the Nvidia Jetson TX2 device. Then, machine learning technologies at the edge are analyzed and their comparative analyzes are given. Some of these technologies were applied to the demographic analytics solution using Nvidia Jetson TX2 device.*

Keywords: *Machine learning, technologies, comparative analysis, Nvidia Jetson TX2*

1. УВОД

Крајњи уређаји, попут телефона и IoT сензора, генеришу податке који морају бити обрађени у реалном времену. Нови тренд уводи примену машинског учења за одбраду генерисаних података. Међутим, примена машинског учења и обрада података захтевају доста процесне моћи и брз одзив како би се извршавање обављало у реалном времену.

Како би се овај захтев испунио, за примену машинског учења на ивици (енгл. *machine learning at the edge*) неопходно је изабрати добар скуп технологија. Ово је важно јер у мору информација и података које се креирају великом брзином, у такозваним системима великих скупова података (енгл. *Big Data Systems*), потребно је имати добар алат са којим ће се креирати квалитетна решења и нови производи у софтверској индустрији.

У овом раду биће укратко дате теоријске основе машинског учења на ивици, затим биће укратко описан Nvidia Jetson TX2 уређај, увод у то шта је обрада тока података (енгл. *data stream*, *dataflow*) и значење термина Machine Learning Operations. Након тога биће наведене испробане технологије за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику и њихова компаративна анализа. Затим ће бити наведене изабране технологије и резултати постигнути њиховим коришћењем. На крају биће дат закључак.

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Душан Гајић, ванр. проф.

2. ТЕОРИЈСКЕ ОСНОВЕ

У овом поглављу дате су теоријске основе области из ког се имплементација система за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику састоји.

2.1. Машинско учење и његова примена на ивици

Машинско учење је еволуирајућа грана рачунарских алгоритама који су дизајнирани да опонашају људску интелигенцију учећи из окружења. Другим речима, машинско учење је грана вештачке интелигенције (енгл. *Artificial Intelligence*, AI) и рачунарске науке која је фокусирана на податке и алгоритме тако да имитира људе, постепено унапређујући сами себе. Машинско учење на ивици је област са одређеним бројем изазова али и могућности. Предности овог приступа су смањена количина података који се шаљу на централно место за њихово чување и тиме смањено време одзива, децентрализован приступ постаје робуснији на отказ у мрежи и приватност података је на већем нивоу. Постоји неколико метода машинског учења, и оне се деле на надгледано (енгл. *supervised*), ненадгледано (енгл. *unsupervised*) и учење условљавањем (енгл. *reinforcement learning*), у зависности од типа проблема који алгоритам решава.

2.2. Обрада тока података

Многи извори производе податке континуално. Примери укључују мреже, акције корисника, научне податке, мултимедије, трансакције и многе друге. Ови извори се зову извори података. Обрада токова података се у највишем употребљава за детекцију круцијалних ствари, као што су детекција превара, крађа, отказ надгледаног система, али и ради добијања статистичких података у жељене сврхе.

2.3. Machine Learning Operations

Термин Machine Learning Operations (MLOps) је дефинисан као проширење Development Operations (DevOps) методологије укључивањем средстава

машинског учења и науке о подацима као *first-class citizens* у оквиру DevOps екологије.

Принципи MLOps-а су оптимизација и аутоматизација процеса стављања нових ствари у продукцију и добијања повратне информације, континуални развој, испоручивање, тренирање модела и мониторинг података и метрика перформанси модела.

2.4. Nvidia Jetson TX2 уређај

Nvidia Jetson је серија напредних система коришћених од стране инжењера за креирање иновативних AI производа. Са Jetson платформом, која је данас једна од водећих у свету рачунарства на ивици, отворен је широк спектар за добијање искуства и креирање креативних пројектних решења. Платформа је састављена од малих јединица које се називају модули. Хардверска компонента која је срце ове платформе је Nvidia Pascal графичка картица, која је намењена за извршавање алгоритама машинског учења и неуронских мрежа, док софтверска компонента која представља подршку за изградњу и извршавање AI апликација је Nvidia JetPack SDK, што представља скуп пакета и библиотека, а међу најважније спадају OpenCV, TensorRT, cuDNN, CUDA и Nvidia Container Runtime.

3. ТЕХНОЛОГИЈЕ И ЊИХОВА КОМПАРАТИВНА АНАЛИЗА

У овом поглављу су наведене испробане технологије у имплементацији система за демографску аналитику и њихова компаративна анализа.

3.1. OpenCV

OpenCV (Open Source Computer Vision Library) је библиотека отвореног кода (енгл. *open source library*) за анализу и управљање сликама и видео садржајем представљена од стране Intel корпорације [1]. Саграђена је у намери да обезбеди механизам за обраду слике у апликацијама за машинско учење и убрза коришћене перцепције машина у комерцијалним производима [2]. Елементи обраде, као што су слика и видео, су представљени матрично што представља природну репрезентацију ових елемената и тиме чини употребу ове библиотеке једноставном.

3.2. YoloV5

YOLO (You Only Look Once) алгоритам, који је представљен 2015. године, донео је нови приступ преобликовањем детекције објеката као регресиони модел извршавајући се у једној неуронској мрежи. Основна идеја аутора YOLO алгоритма била је да ова архитектура израчуна све особине слике и направи предикције свих објеката истовремено, у једном пролазу. Због начина на који ради и резултата које постиже оцењен је као најбољи алгоритам за детекцију објеката [3].

3.3. Multi-Task Cascaded Convolutional Networks и Haar Cascade

MTCNN алгоритам је спорији због свог поступка који је у свакој фази захтеван. Међутим, даје веома добре резултате у погледу квалитета без обзира на квалитет улазне слике, у погледу угла лица, осветљености и заклоњености.

Haar Cascade алгоритам је брз алгоритам, али једна од мана овог алгоритма је што је у стању да детектује само лица спреда под добрим осветљењем, с обзиром на то да је на таквом скупу података трениран. Такође, често даје лажно позитивне резултате.

3.4. EfficientNetB7, VGG16 и InceptionV3

С обзиром на то да је EfficientNetB7 неуронска мрежа креирана коришћењем претраге неуронске архитектуре за проналажење добре основне мреже а затим и претрагом мреже величина за добијање оптималних вредности величина за скалирање по свим димензијама, она даје најбоље резултате. Разлог услед ког VGG16 неуронска мрежа даје боље резултате од InceptionV3 неуронске мреже јесте тај што је комплекснија у погледу броја рачунања што јој, без обзира што је потребно више времена за рачунање, помаже у остваривању бољих резултата. Табела 1 приказује постигнуте резултате сваке од мрежа понаособ на UTKFace скупу података.

Табела 1. *Постигнути резултати мрежа*

	accuracy	**loss**	**male f1-score**	**female f1-score**
EfficientNetB7	0.855	0.312	0.856	0.855
VGG16	0.833	0.384	0.852	0.809
InceptionV3	0.757	0.521	0.763	0.752

3.5. Apache Kafka и RabbitMQ

Apache Kafka и RabbitMQ системи могу се поредити у два погледа, у погледу квалитета и погледу квантитета.

Квалитативно поређење обухвата неколико особина: *time decoupling*, *routing logic*, *delivery guarantees*, *ordering guarantees*, *availability*, *transactions*, *dynamic scaling*.

Квантитативно поређење обухвата неколико особина: *latency*, *throughput*.

Поређењем система по овим особинама показало се да не постоји генерално бољи систем, већ је један систем у неким особинама бољи од другог па је за такве потребе и намењен.

Предности Kafka система: дуготрајно чување, репликација порука, Kafka Connect и компакција лог порука.

Предности RabbitMQ система: стандардизован протокол, подршка за коришћење више протокола истовремено, дистрибуирани модови топологије, свеобухватни алати за управљање и праћење, изолација окружења, праћење потрошача, мање коришћење диска, контрола произвођача, могућност ограничења величине редова и могућност ограничења животног века порука [4].

3.6. Apache Flink, Apache Kafka Streams и Apache Spark

Један од изазова у развоју архитектуре за обраду токова јесте избор праве технологије за различите

случајеве коришћења. Како свака од испробаних технологија функционише донекле по сличним принципима и архитектурама, корисно је издвојити особине сваке од технологија.

3.6.1. Програмски модел

Stream алат је базиран на микро-пакетном (енгл. *micro-batch*) принципу, што је проширење главног Spark интерфејса. Главна апстракција у Sparku је RDD (Resilient Distributed Dataset).

Flink је *native* алат за обраду података са подршком за обраду података у пакетима. Основни градивни блок Flink програма су токови и оператори трансформације пресликане у токове података.

Kafka Streams је део целокупног Kafka екосистема који се састоји од непромењивих записа, токова података звани *topic*-и, потрошача, произвођача, логова, партиција и кластера. Најважнија апстракција у Kafka Streams је ток који представља неограничен скуп непромењивих записа. Топологија представља граф процесора који су повезани токовима и који деле складишта стања, као и апстракцију кориштену да дефинише логику рачунања у токовима за апликације.

3.6.2. Партиционисање података

Spark аутоматски партиционише RDD компоненте и дистрибуира партиције преко различитих чворова. Hash и Range су честе стратегије за партиционисање подржане од Sparka.

Током извршавања, ток у Flinku садржи једну или више партиција и сваки оператор има један или више подзадатака који су му додељени. Ток може да трансформише податке између два оператора у "један-на-један" шаблону (енгл. *one-to-one pattern*), што очувава редослед елемената.

Kafka Streams партиционише податке ради процесирања док слој порука у Kafka екосистему партиционише податке за чување и транспорт порука. У оба случаја партиционисање омогућава локалитет, еластичност, скалабилност, високе перформансе и отпорност на грешке.

3.6.3. Управљање стањем

На високом нивоу апстракције, Sparkova Structured Streaming библиотека прати стање на сличан начин и у микро-пакетном и у серијском режиму. Стање апликације се прати коришћењем два спољашња система за складиштење, "дневником унапред" (енгл. *write-ahead log*) и складиштем стања.

Flinkov екосистем сачињен од модула и сервиса саграђен на његовом језгру нуди различите начине приступа и изолазије стања. Свака операција у току може дефинисати своје сопствено стање и ажурирати га континуално у намери да одржава резиме до сада виђених података.

Kafka Streams пружа апликацијама моћну, еластичну и високо скалабилну обраду са могућностима отпорности на грешке. Kafka Streams пружа складишта стања која могу бити употребљена од стране апликација за обраду података за складиште и добављају податке, што је важна способност за спровођење операција.

3.6.4. Гаранција процесирања

Sparkova Structured Streaming библиотека обезбеђује брзу, скалабилну обраду отпорну на грешке, *end-to-end exactly-once* обраду података помоћу микро-пакетног *engine*-а. Ове гаранције могу бити постигнуте избором релевантних режима базираних на захтевима апликација без промене Dataset или DataFrame операција.

Механизам за обраду порука у Flinku гарантује да ће и у случају пада, стање програма одржавати сваки запис из тока, тачно једном. Flink може да гарантује да се "тачно једном" стање ажурира у стање које дефинише корисник само када извор учествује у механизму контролних тачака (енгл. *checkpoints*).

Kafka Streams такође подржава *end-to-end exactly-once* семантику која гарантује да за било који запис прочитан из извора Kafka *topic*-а, резултат његове обраде ће бити рефлектован тачно једном у излазни *topic*.

3.6.5. Отпорност на грешке

Structured Streaming библиотека обезбеђује *end-to-end exactly-once* гаранцију отпорности на грешке кроз контролне тачке и *write-ahead logs* приступе. Structured Streaming систем контролних тачака користи већа складишта стања да чува тренутна стања оператора за дуготрајне операције.

Flink имплементира отпорност на грешке коришћењем комбинације репродукције тока и контролних тачака. Стриминг тока података у Flinku може да се надовеже са контролне тачке док се одржава доследност или семантика обраде "тачно једном".

Kafka Streams се надовезује на способност отпорности на грешке интегрисану у језгро Kafke. Kafkine партиције су веома доступне и репликабилне, као када је податак перзистентан и Kafki, доступан је чак и ако апликација падне и потребно га је поново обрадити [5].

3.7. MLflow

MLflow библиотека је дизајнирана да олакша развој машинског учења. То је платформа отвореног кода креирана да ради са било којом библиотеком машинског учења и било којим програмским језиком. MLflow дефинише компоненте које су дизајниране за адресирање фундаменталних изазова у свакој фази циклуса машинског учења, од развоја модела до стављања у продукцију, и то MLflow Tracking за праћење експеримената, MLflow Models за паковање модела, MLflow Projects за паковање кода и MLflow Registry за чување модела.

3.8. Apache Superset

Apache Superset је модерна веб-апликација пословне интелигенције. То је брза, лака, интуитивна апликација која садржи мноштво опција и која корисницима свих нивоа олакшава анализу и визуализацију података, од једноставних до веома детаљних графикона.

Superset пружа интуитиван интерфејс, широк спектар лепих визуализација, креирање визуализације без

кода, моћан SQL алат за припрему података и многе друге ствари.

3.9. MinIO

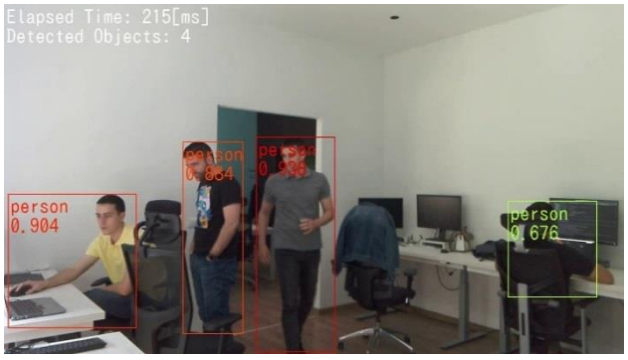
MinIO је систем високих перформанси за складиштење дистрибуираних објеката. MinIO се разликује по томе што је од свог почетка дизајниран да буде стандард у приватном и хибридном складишту објеката у облаку. Будући да је MinIO наменски направљен да служи само за складиштење објеката, једнослојна архитектура постиже сву потребну функционалност без компромиса. Резултат тога је *cloud-native* сервер објеката који је истовремено ефикасан, скалабилан и лаган. Иако се MinIO истиче у случајевима коришћења традиционалног складиштења објеката као што су секундарно складиштење, опоравак од катастрофе и архивирање, јединствен је у превазилажењу изазова повезаних за машинско учење, аналитику и радна оптерећења апликација у облаку.

4. РЕЗУЛТАТИ

За имплементацију система за демографску аналитику употребом Nvidia Jetson TX2 уређаја употребљен је следећи скуп технологија: OpenCV, YoloV5, Multi-Task Cascaded Convolutional Networks, EfficientNetB7, Apache Kafka, Apache Flink, MLflow, Apache Superset и MinIO.

Делови који обухватају учитавање слика са камере, закључивање (енгл. *inference*) неуронских мрежа и креирање материјала за каснију употребу обављају се на самом Jetson уређају, док остали део архитектуре се обавља на екстерној машини.

Слика 1 приказује пример резултата детекције особа на слици.



Слика 1. *Пример резултата детекције особа на слици*

5. ЗАКЉУЧАК

У овом раду су укратко дате теоријске основе машинског учења на ивици, затим је укратко описан Nvidia Jetson TX2 уређај, увод у то шта је обрада тока података и значење термина Machine Learning Operations. Након тога су наведене испробане технологије за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику и њихова компаративна анализа. Затим су наведене изабране технологије и резултати постигнути њиховим коришћењем.

Оно што ову архитектуру издваја од других често креираних архитектура јесте ниво машинског учења који је постигнут извршавањем машинског учења на Jetson уређају.

Примена ове имплементације на неки од случајева коришћења са циљем у реалној употреби представља тему за даљи развој. Унапређење модела машинског учења и имплементација интеракције више Jetson уређаја представљају тему за даља унапређења.

6. ЛИТЕРАТУРА

[1] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.

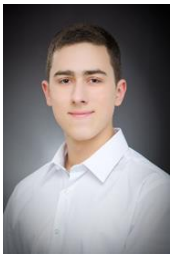
[2] Getting started with OpenCV and Python, https://medium.com/the-andela-way/simple-operations-on-images-using-opencv-d37b26e6e3ab

[3] Thuan, D. (2021). Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm.

[4] Dobbelaere, P., & Esmaili, K. S. (2017, June). Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper. In *Proceedings of the 11th ACM international conference on distributed and event-based systems* (pp. 227-238).

[5] Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. (2019). A survey of distributed data stream processing frameworks. *IEEE Access*, *7*, 154300-154316.

Кратка биографија:



Милош Радојчин рођен је у Новом Саду 1997. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке и информатика одбранио је 2021. год.

контакт: milos.radojcin@uns.ac.rs