DOI: https://doi.org/10.26565/2304-6201-2024-62-07

УДК (UDC) 004.8:342.9

Трусов Михайло Андрійович *Ведучий розробник програмного забезпечення, IT-компанія «EPAM Systems», Ukraine, вул. 23 Серпня, 33, Харків, Україна, 61045*
e-mail: trusov.michael@gmail.com

Узлов Дмитро Юрійович *к.т.н., доцент, в.о. декана факультету комп'ютерних наук Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022*
*e-mail:*dmytro.uzlov@karazin.ua
https://orcid.org/0000-00Q3-3308-424X

Перспективи використання моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях

Актуальність: впровадження моделей глибокого навчання для семантичної сегментації на автономних пристроях є перспективним напрямком розвитку інтелектуальних систем, здатних аналізувати візуальну інформацію без постійного підключення до зовнішніх ресурсів.
Метою роботи є дослідження можливостей та викликів використання моделей глибокого навчання для задач семантичної сегментації на автономних пристроях.
Методи дослідження включають теоретичний аналіз, систематизацію та узагальнення використання моделей глибокого навчання в автономних пристроях, а також параметрів, що впливають на обсяг пам'яті моделей, та особливостей імплементації натренованих моделей у власні програмні продукти.
Результати: виявлено значний потенціал технології глибокого навчання для створення автономних інтелектуальних систем. Визначено основні параметри, що впливають на ефективність роботи моделей на пристроях з обмеженими ресурсами. Запропоновано рекомендації щодо імплементації натренованих моделей у програмні продукти. Висвітлено сучасні підходи до шифрування моделей глибокого навчання.
Висновки: подальші розробки в області глибокого навчання для семантичної сегментації на автономних пристроях сприятимуть розвитку більш ефективних та автономних систем для широкого спектру застосувань, включаючи комп'ютерний зір, робототехніку тощо. Забезпечення безпеки доступу до навчених моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях вимагає комплексного підходу, що поєднує апаратні та програмні рішення.
Ключові слова: *глибоке навчання, семантична сегментація, автономні пристрої, оптимізація моделей, вбудовані системи, апаратне прискорення, шифрування даних.*

Як цитувати: Трусов М. А., Узлов Д. Ю. Перспективи використання моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2024. вип. 62. С.70-79. https://doi.org/10.26565/2304-6201-2024-62-07

How to quote: M. Trusov, D. Uzlov "Perspectives of using deep learning models for semantic image segmentation on autonomous devices", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 62, pp. 70-79, 2024. [In Ukrainian] https://doi.org/10.26565/2304-6201-2024-62-07

1. Вступ

Комп'ютерний зір є однією з найбільш динамічних і стрімко розвиваючихся галузей штучного інтелекту (ШІ), яка займається створенням технологій, здатних імітувати зорове сприйняття людини. Ця дисципліна об'єднує в собі елементи комп'ютерних наук, оптики, механіки та нейронауки для розробки алгоритмів, які можуть аналізувати, обробляти та інтерпретувати візуальні дані з різноманітних джерел [1]. Серед ключових напрямків цієї галузі особливе місце займає розробка та вдосконалення алгоритмів семантичної сегментації зображень [2]. Семантична сегментація, як інтегральна частина комп'ютерного зору, має за мету детальну класифікацію кожного пікселя на зображенні згідно з визначеними категоріями [3]. Вона відрізняється від процесу детекції об'єктів, який обмежується створенням контурів навколо них, надаючи більш глибоке розуміння форми та положення об'єктів у просторі зображення. Цей метод є одним з

трьох підходів у комплексному процесі сегментації зображень, що сприяє точнішому та ефективнішому розумінню візуальної інформації комп'ютерними системами. Моделі семантичної сегментації спрямовані на створення карти сегментації для вхідного зображення, яка є реконструкцією оригіналу, де кожен піксель кодується певним кольором відповідно до його семантичного класу [4]. Це дозволяє створювати маски сегментації, які виділяють окремі області зображення, відмежовані від інших областей. Так, наприклад, карта сегментації дерева в порожньому полі, ймовірно, міститиме три маски сегментації: для дерева, для землі та для неба на задньому плані, відповідно. Семантична сегментація, яка дозволяє комп'ютерним системам ідентифікувати та класифікувати об'єкти на зображеннях до відповідних категорій, є ключовою для створення інтелектуальних систем, здатних до глибокого розуміння візуального контенту. Підвищення точності та ефективності цих алгоритмів сприяє значному прогресу у вирішенні завдань, пов'язаних з обробкою великих обсягів візуальних даних, забезпечуючи більш точне та швидке узагальнення цільових об'єктів.

Проблематика семантичної сегментації є фокусом наукових досліджень ряду вітчизняних та зарубіжних вчених [5-7]. Зокрема, у вітчизняній науці дослідження зосереджені на адаптації та вдосконаленні сучасних методів семантичної сегментації для конкретних прикладних задач, таких як аналіз супутникових знімків, аналіз даних від БПЛА різних типів тощо. Окрім цього, такі галузі як робототехніка, медицина, автопромисловість в тому чи іншому вигляді використовують новітні технології, пов'язані з необхідністю швидкого та якісного аналізу зображення. У свою чергу, дослідження зарубіжних вчених охоплюють ширший спектр проблем та підходів, включаючи розробку нових архітектур нейронних мереж та методів навчання [8,9].

Загальноприйняті на сьогоднішній день концепції семантичної сегментації постулюють існування двох підходів до реалізації семантичної сегментації [10]. Зокрема, традиційні методи семантичної сегментації включають два основні процеси: виділення ознак (feature extraction) і класифікацію пікселів (pixel classification). Однак, традиційний підхід має кілька недоліків, зокрема високу залежність від доменної експертизи для визначення та виділення ознак, що може бути часозатратним та не завжди ефективним для складних зображень [11]. Крім того, цей підхід часто обмежений у своїй здатності адаптуватися до нових, невидимих даних, що призводить до зниження точності класифікації. У відповідь на ці обмеження, було розроблено методологію глибокого навчання, яка використовує глибокі нейронні мережі для автоматичного виявлення ознак, що значно підвищує точність та універсальність систем семантичної сегментації [1,2]. Завдяки своїй здатності до глибокого навчання на великих наборах даних, ці системи можуть ефективно адаптуватися до нових завдань, забезпечуючи високу точність навіть у складних візуальних умовах. Провідними архітектурами для семантичної сегментації з використанням глибокого навчання є Fully Convolutional Network (FCN), U-Net, DeepLab та PSPNet. Ці архітектури ефективно використовують згорткові нейронні мережі (CNN) для деталізованої екстракції характеристик та високоточної класифікації пікселів у складних візуальних сценах.
Однак, при використанні нейронних мереж для семантичної сегментації зображень, виникає питання про їх інтеграцію в автономні пристрої з обмеженим або відсутнім доступом до Інтернету. У світлі цього, метою даної роботи є дослідження потенціалу застосування моделей глибокого навчання в автономних пристроях, аналіз методів їх ефективної інтеграції та використання без постійного з'єднання з Інтернетом.

2. Використання моделей глибокого навчання в автономних пристроях для семантичної сегментації зображень

Аналіз останніх досліджень у галузі семантичної сегментації показав, що моделі глибокого навчання, зокрема повністю згорткові мережі (FCN) та інші типи нейронних мереж, можуть бути ефективно використані на автономних пристроях, які не мають доступу до Інтернету. Одним із способів реалізації цього є **деплой моделей** безпосередньо **на пристрої** (**on-device interference**), що дозволяє використовувати навчену модель для локального прогнозування без необхідності з'єднання з мережею. Це включає навчання моделі на потужному обладнанні, а потім її розгортання на кінцевому пристрої [12]. Таким чином, пристрій може виконувати локальні прогнози без необхідності постійного підключення до зовнішніх обчислювальних ресурсів. Це не тільки зменшує затримку при обробці даних, але й підвищує приватність, оскільки чутлива інформація не передається через мережу. Крім того, локальне виконання моделей дозволяє

працювати в умовах обмеженого або відсутнього підключення до Інтернету, що розширює можливості застосування ШІ в різних сценаріях.

Інша концепція, що базується на **граничних обчисленнях (edge computing)**, передбачає розміщення обчислювальних ресурсів ближче до джерела даних, що дозволяє обробляти інформацію локально. Наприклад, розумні камери можуть аналізувати відеопотоки безпосередньо на пристрої за допомогою попередньо навчених моделей, що значно зменшує залежність від постійного інтернет-з'єднання та сприяє обробці даних у реальному часі [13,14]. Окрім цього, граничні обчислення також дозволяють розподілити навантаження між різними пристроями, створюючи розподілену мережу систем ШІ, здатних взаємодіяти та обмінюватися інформацією для вирішення складних завдань.

Для ефективного виконання моделей нейронних мереж на автономних пристроях широко використовуються **спеціалізовані вбудовані системи**. Такі платформи, як NVIDIA Jetson, Google Coral та Intel Movidius, розроблені саме для цієї мети і можуть бути інтегровані в різноманітні автоматизовані пристрої для виконання складних завдань, включаючи семантичну сегментацію. Ці системи оптимізовані для виконання операцій, характерних для нейронних мереж, що дозволяє значно прискорити обчислення порівняно зі звичайними процесорами. Крім того, вони часто мають низьке енергоспоживання, що робить їх перспективною альтернативою для мобільних та автономних пристроїв [15].

Для забезпечення ефективної роботи нейронних мереж на пристроях з обмеженими ресурсами застосовується широкий спектр **методів оптимізації**. Ці техніки спрямовані на зменшення обчислювальної складності моделей при збереженні їх точності та продуктивності. Одним з ключових напрямків є стиснення моделей (model compression), яке включає в себе ряд підходів, таких як квантування, скорочення та дистиляція знань. Квантування передбачає зменшення точності представлення ваг та активацій нейронної мережі, що дозволяє суттєво скоротити обсяг пам’яті, необхідний для зберігання моделі, та прискорити обчислення [16]. Скорочення мережі полягає у видаленні надлишкових нейронів та зв'язків, що не мають значного впливу на кінцевий результат, тим самим зменшуючи кількість параметрів моделі. Дистиляція знань дозволяє передати "знання" від великої складної моделі до меншої, зберігаючи при цьому високу точність прогнозування.

Іншим важливим напрямком оптимізації є використання апаратного прискорення (hardware acceleration). Цей підхід передбачає застосування спеціалізованих обчислювальних пристроїв, таких як графічні процесори (GPU), тензорні процесори (TPU) або спеціалізовані мікросхеми для штучного інтелекту (ШІ-чіпи), які оптимізовані для виконання операцій, характерних для нейронних мереж. Графічні процесори, завдяки своїй паралельній архітектурі, здатні значно прискорити матричні обчислення, які є основою більшості операцій в нейронних мережах [17]. Тензорні процесори, розроблені спеціально для задач машинного навчання, забезпечують ще вищу ефективність за рахунок оптимізації під конкретні алгоритми глибокого навчання. Спеціалізовані ШІ-чіпи, такі як нейроморфні процесори, імітують структуру біологічних нейронних мереж, що дозволяє досягти надзвичайно високої енергоефективності при виконанні завдань штучного інтелекту [18].

Комбінація методів стиснення моделей та апаратного прискорення дозволяє досягти синергетичного ефекту, значно підвищуючи ефективність роботи нейронних мереж на вбудованих системах. Наприклад, квантовані моделі можуть бути ще більш ефективно виконані на спеціалізованих апаратних прискорювачах, оптимізованих під операції з низькою точністю. Крім того, розробляються нові архітектури нейронних мереж, які враховують особливості цільового апаратного забезпечення, що дозволяє максимально використовувати його можливості [19]. Такий комплексний підхід до оптимізації відкриває широкі перспективи для впровадження складних алгоритмів штучного інтелекту в мобільні та вбудовані пристрої, розширюючи сферу їх застосування та підвищуючи автономність та інтелектуальність edge-пристроїв.

Нарешті, ще одним аспектом сучасних вбудованих систем, що дозволяє пристроям виконувати складні завдання без постійного підключення до Інтернету, є так звана **обробка даних в автономному режимі (offline data processing)**. Цей підхід особливо корисний у ситуаціях, коли пристрій збирає дані та обробляє їх пакетами, виконуючи завдання сегментації в автономному режимі та лише час від часу підключаючись до Інтернету для оновлення або додаткової передачі даних. Такий метод забезпечує високу надійність, знижує затримку та покращує захист даних, оскільки чутлива інформація не передається через мережу. Однією з ключових переваг автономної

обробки даних є можливість виконання обчислень на місці, що зменшує залежність від зовнішніх обчислювальних ресурсів і підвищує швидкість реакції системи. Наприклад, у промислових застосуваннях, таких як автоматизовані виробничі лінії або системи контролю якості, локальна обробка даних дозволяє швидко виявляти дефекти та приймати рішення в реальному часі, що значно підвищує ефективність виробничих процесів. Крім того, автономні пристрої можуть працювати в умовах обмеженого або відсутнього підключення до Інтернету, що робить їх ідеальними для використання в віддалених або важкодоступних місцях.

Таким чином, можна стверджувати, що навчання моделей в автономному режимі та розгортання оптимізованих версій на периферійних пристроях або вбудованих системах є ефективним підходом для досягнення високоякісної семантичної сегментації без необхідності постійного підключення до Інтернету. Цей підхід має значні переваги, зокрема зниження затримок, підвищення надійності та забезпечення приватності даних, оскільки обробка здійснюється локально.

3. Параметри, які впливають на обсяг пам'яті, що займається натренованою моделлю

Мінімальний необхідний обсяг пам'яті для розгортання навчених моделей на пристрої залежить від кількох факторів, включаючи складність моделі, конкретну архітектуру та застосовані методи оптимізації. Розглянемо ці аспекти детальніше.

Розмір моделі є ключовим фактором. Прості моделі, такі як менші версії FCN або легкі архітектури (наприклад, MobileNet, SqueezeNet), можуть займати від кількох до десятків мегабайт. Натомість більші моделі, зокрема, повнорозмірні FCN або інші складні архітектури (наприклад, ResNet, VGG), можуть вимагати від сотень мегабайт до кількох гігабайт пам’яті [20].

Застосування методів оптимізації дозволяє суттєво зменшити розмір моделі. Квантизація, яка передбачає зниження точності ваги моделі (наприклад, з 32-бітної до 8-бітної), може зменшити розмір моделі до 4 разів. Обрізка (pruning) видаляє менш важливі ваги або нейрони, що також зменшує розмір моделі без значного впливу на продуктивність [21]. Дистиляція знань, при якій менша модель (учень) навчається імітувати більшу модель (вчитель), може призвести до значно меншої моделі з порівнянною продуктивністю.

Розглянемо конкретні приклади моделей. MobileNetV2, компактна модель, розроблена для мобільних пристроїв, зазвичай займає близько 10-15 МБ після квантизації [22]. Tiny YOLO, менша версія моделі ідентифікації об'єктів YOLO (You Only Look Once), займає приблизно 60 МБ. DeepLab, популярна модель для семантичної сегментації, має менші версії (наприклад, з використанням MobileNet як основи) розміром 20-30 МБ, а більші версії (наприклад, з використанням Xception або ResNet) – близько 100-200 МБ [23].

Окрім цього, певну увагу слід приділити додатковим вимогам. Середовище виконання або бібліотеки, необхідні для роботи моделі (наприклад, TensorFlow Lite, ONNX Runtime), також потребують певного обсягу пам'яті. Крім того, допоміжні дані, такі як мітки класів або конфігураційні файли, додають до загальних вимог до пам'яті.

Таким чином, можна оцінити вимоги до пам'яті наступним чином:

- для простих завдань та легких моделей: 10-50 МБ.
- для моделей середньої складності з деякою оптимізацією: 50-200 МБ.
- для складних, високопродуктивних моделей: від 200 МБ до кількох ГБ.

Наприклад, при розгортанні квантованої версії MobileNetV2 для семантичної сегментації можна очікувати наступні вимоги до пам'яті: розмір моделі - близько 10-15 МБ, накладні витрати фреймворку - 5-10 МБ, допоміжні дані - близько 1 МБ, що в сумі складає 16-26 МБ.

Таким чином, завдяки ефективним моделям та оптимізаціям, можливо розгортати нейронні мережі на пристроях з відносно невеликими вимогами до пам’яті, що робить можливим функціонування багатьох повністю автоматизованих пристроїв незалежно від підключення до Інтернету.

4. Особливості імплементації натренованих моделей у власні програмні продукти

Для використання навчених моделей у власних програмах необхідні бібліотеки або фреймворки, що полегшують взаємодію з цими моделями [24]. Розглянемо деякі поширені бібліотеки та фреймворки для різних платформ:

1. *TensorFlow Lite* – фреймворк, розроблений для розгортання моделей на мобільних та вбудованих пристроях. Він підтримує мови програмування Python, C++, Java та Swift. Ключовими

особливостями TensorFlow Lite є конвертація моделей TensorFlow у менший, оптимізований формат, надання API для інференсу на різних платформах (Android, iOS, Linux-based systems), а також підтримка апаратного прискорення (наприклад, через NNAPI, GPU).

2. *ONNX (Open Neural Network Exchange) Runtime* представляє собою кросплатформне високопродуктивне середовище для моделей формату ONNX. Він підтримує Python, C++, C#, Java, JavaScript та інші мови. ONNX Runtime дозволяє працювати з моделями з різних фреймворків (наприклад, PyTorch, TensorFlow), оптимізований для різних апаратних прискорювачів (GPU, TPU) і може використовуватися на різних платформах, включаючи Windows, Linux та macOS.

3. *PyTorch Mobile* призначений для розгортання моделей PyTorch на мобільних пристроях. Він підтримує Python, Java та C++. PyTorch Mobile надає інструменти для оптимізації та конвертації моделей PyTorch для мобільного розгортання, підтримує платформи Android та iOS, а також інтегрується з фреймворками розробки мобільних додатків.

4. *Core ML* є фреймворком машинного навчання Apple для додатків iOS та macOS. Він переважно використовується з Swift та Objective-C. Core ML інтегрується з Xcode для безперешкодного розгортання, оптимізований для апаратного забезпечення Apple (наприклад, Neural Engine) і підтримує конвертацію з різних форматів моделей (TensorFlow, PyTorch).

5. *NVIDIA TensorRT* представляє собою набор інструментів для високопродуктивного інференсу глибокого навчання на GPU NVIDIA. Він підтримує Python та C++. TensorRT оптимізує та розгортає моделі для інференсу на апаратному забезпеченні NVIDIA, підтримує різні фреймворки глибокого навчання (TensorFlow, PyTorch) і ідеально підходить для крайових пристроїв з модулями NVIDIA Jetson.

6. *OpenCV* – відкрита бібліотека комп'ютерного зору та машинного навчання. Вона підтримує C++, Python, Java та MATLAB. OpenCV надає можливості інференсу глибокого навчання через модуль DNN, підтримує моделі з різних фреймворків (Caffe, TensorFlow, PyTorch) і має кросплатформну підтримку (Windows, Linux, macOS, Android, iOS).

Нижче наведено приклад інтеграції з TensorFlow Lite (Python):

```
import tensorflow as tf
import numpy as np

# Load the TFLite model and allocate tensors.
interpreter = tf.lite.Interpreter(model_path="model.tflite")
interpreter.allocate_tensors()

# Get input and output tensors.
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

# Prepare input data
input_data = np.array(your_input_data, dtype=np.float32)

# Run inference
interpreter.set_tensor(input_details[0]['index'], input_data)
interpreter.invoke()

# Get output data
output_data = interpreter.get_tensor(output_details[0]['index'])
print(output_data)
```

Вибір конкретної бібліотеки або фреймворку залежить від специфіки проекту, цільової платформи та вимог до продуктивності.

5. Шифрування моделей глибокого навчання

При дослідженні використання моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях, критичного значення набуває забезпечення безпеки доступу

до навчених моделей. Розглянемо декілька сучасних підходів до забезпечення безпеки, які можуть бути ефективно застосовані у даному контексті:

1. Модулі апаратної безпеки (Hardware Security Modules, HSM)

HSM представляють собою спеціалізовані фізичні пристрої, які виконують криптографічну обробку та забезпечують надійне зберігання ключів [25]. Вони відіграють вирішальну роль у захисті моделей глибокого навчання, призначених для семантичної сегментації, шляхом зберігання та управління криптографічними ключами, що використовуються для шифрування та дешифрування моделей. Інтеграція HSM у автономні пристрої для реалізації семантичної сегментації зображень дозволяє проводити операції шифрування та дешифрування в безпечному середовищі [26]. Це запобігає несанкціонованому доступу до навчених моделей, оскільки вони можуть бути зашифровані та розшифровані лише всередині HSM. Такий метод значно підвищує захист інтелектуальної власності та конфіденційності даних, які використовуються під час навчання моделей.

2. Модуль довіреної платформи (Trusted Platform Module, TPM)

TPM є захищеним криптопроцесором, який може зберігати криптографічні ключі та виконувати криптографічні операції, створюючи надійну апаратну основу захисту [27]. Він є ключовим елементом у захисті моделей глибокого навчання для семантичної сегментації, оскільки забезпечує безпечне зберігання ключів, які є необхідними для доступу до цих моделей. Застосування TPM дозволяє шифрувати моделі та забезпечувати їх розшифрування лише за допомогою авторизованих операцій, що значно підвищує рівень безпеки. Такий підхід гарантує, що навіть при фізичному доступі до пристрою, несанкціоноване використання моделей стає вкрай утрудненим.

3. Комплексний підхід: шифрування та контроль доступу

Ефективний захист моделей глибокого навчання, які використовуються для семантичної сегментації зображень, вимагає комплексного підходу, що об'єднує якісне шифрування та ретельний контроль доступу. Шифрування моделей має бути здійснене за допомогою передових алгоритмів, таких як AES-256, що забезпечує надійний захист даних. Ключі для шифрування необхідно зберігати в безпечному середовищі, використовуючи модулі HSM або TPM, що гарантує їхню цілісність навіть у випадку несанкціонованого вилучення моделей з пристрою [28]. Такий комплексний підхід є ключовим для забезпечення високого рівня захисту моделей глибокого навчання в автономних пристроях.

6. Висновки

У роботі продемонстровано значний потенціал використання моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях. Ця технологія відкриває широкі можливості для розвитку інтелектуальних систем, здатних аналізувати візуальну інформацію без постійного підключення до зовнішніх обчислювальних ресурсів. Ключовим аспектом успішного впровадження таких моделей є оптимізація їх розміру та ефективності. Параметри, що впливають на обсяг пам'яті, займаний натренованою моделлю, включають складність архітектури, кількість шарів та нейронів, а також точність представлення ваг. Застосування методів оптимізації, таких як квантизація, обрізка та дистиляція знань, дозволяє значно зменшити розмір моделей без суттєвої втрати точності, що робить їх придатними для використання на пристроях з обмеженими ресурсами. Імплементація натренованих моделей у власні програмні продукти вимагає використання спеціалізованих бібліотек та фреймворків, таких як TensorFlow Lite, ONNX Runtime або PyTorch Mobile. Ці інструменти забезпечують ефективне розгортання моделей на різних платформах, включаючи мобільні пристрої та вбудовані системи. Важливим аспектом є також оптимізація моделей під конкретне апаратне забезпечення, що дозволяє максимально використовувати доступні обчислювальні ресурси. Критичним також є забезпечення безпеки доступу до навчених моделей глибокого навчання для семантичної сегментації зображень на автономних пристроях. Це вимагає комплексного підходу, що поєднує апаратні та програмні рішення.

Перспективи розвитку цієї галузі включають подальше вдосконалення архітектур нейронних мереж для підвищення ефективності семантичної сегментації, розробку нових методів оптимізації для зменшення обчислювальних вимог, а також створення спеціалізованих апаратних прискорювачів для ефективного виконання операцій глибокого навчання на автономних пристроях.

СПИСОК ЛІТЕРАТУРИ

1. R. Szeliski, Computer vision: algorithms and applications. Springer Nature, 2022. 925 p. https://link.springer.com/book/10.1007/978-3-030-34372-9.
2. Hao S., Zhou Y., Guo Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing*. 2020. V. 406. P. 302-321. https://www.sciencedirect.com/science/article/abs/pii/S0925231220305476.
3. Guo Y., Liu Y., Georgiou T., Lew M. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Info Retr*. 2018. V. 7. P. 87-93. https://link.springer.com/article/10.1007/s13735-017-0141-z.
4. Yu H., Yang Z., Tan L., Wang Y., Sun W., Sun M., Tang Y. Methods and datasets on semantic segmentation: a review. *Neurocomputing*. 2018. V. 304. P. 82-103. https://www.sciencedirect.com/science/article/abs/pii/S0925231218304077.
5. Куцик А.Я. Аналіз супутникових знімків на основі семантичної сегментації, кваліфікаційна робота, КПІ ім. І. Сикорського, 2020. https://ela.kpi.ua/items/5c1bdf70-a6a6-45ab-8a63-312296420f6c.
6. Рябко А.В. Аналіз та оцінка методів сегментації супутникових зображень, Всеукр. Науково-технічна конференція «Сталий розвиток систем зв'яязку, навігації, спостереження та організації повітряного руху CNS/ATM - 2023», 29-30 листопада 2023 р., с. 4. https://it-visnyk.kpi.ua/?page_id=2165.
7. Глубока Ю.О. Дослідження якості методів сегментації зображення людини в умовах дії адитивних завад, кваліфікаційна робота, Харківський національний університет радіоелектроніки, 2022. https://openarchive.nure.ua/entities/publication/6bad8449-2d47-4c70-87dc-927980da23e7.
8. Hua Y., Marcos D., Mou L., Zhu X., Tuia D. Semantic segmentation of remote sensing images with sparse annotations. *IEEE Geoscience and Remote Sensing Letters*. 2022. V. 19. P. 1-5. https://arxiv.org/abs/2101.03492.
9. Zhang Y., Chi M. Mask-R-FCN: a deep fusion network for semantic segmentation. *IEEE Access*. 2020. V. 8. P. 155753-155765. https://ieeexplore.ieee.org/document/9151932.
10. O'Mahony N., Campbell S., Carvalho A., et al. Deep learning vs. traditional computer vision, *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)*, Vol. 11, Springer International Publishing, 2020. P. 128-144. https://arxiv.org/abs/1910.13796.
11. Panella F., Lipani A., Boehm J. Semantic segmentation of cracks: data challenges and architecture. *Automation in Construction*. 2022. V. 135. P. 104110. https://www.sciencedirect.com/science/article/abs/pii/S0926580521005616.
12. Mairittha N., Mairittha T., Inoue S. On-device deep learning inference for efficient activity data collection. *Sensors (Basel)*. 2019. V. 19. P. 3434. https://www.mdpi.com/1424-8220/19/15/3434.
13. Cui T. Review of deep learning and mobile edge computing in autonomous driving. *Вісник Львівської політехніки*. 2022. V. 12. P. 208-218. https://science.lpnu.ua/sites/default/files/journal-paper/2023/jan/29757/221029maket-210-220.pdf.
14. Grigorescu S., Trasnea B., Cocias T., Macesanu G. A survey of deep learning techniques for autonomous driving. *J. Field Robotics*. 2020. V. 37. P. 362-386. https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918.
15. Zhang Z., Li J. A review of artificial intelligence in embedded systems. *Micromachines*. 2023. V. 14. P. 897. https://www.mdpi.com/2072-666X/14/5/897.
16. Merone M., Graziosi A., Lapadula V., Petrosino L., d'Angelis O., Vollero L. A practical approach to the analysis and optimization of neural networks on embedded systems. Sensors. 2022. V. 22. P. 7807. https://www.mdpi.com/1424-8220/22/20/7807.
17. Helms D., Amende K., Bukhari S., et al., Optimizing neural networks for embedded hardware. *SMACD/PRIME 2021, International Conference on SMACD and 16th Conference on PRIME*, online. 2021. P. 1-6. https://ieeexplore.ieee.org/document/9547911.
18. Song W. Hardware accelerator systems for embedded systems. *Advances in Computers*, vol. 122. Elsevier, 2021. P. 23-49. https://www.sciencedirect.com/science/article/abs/pii/S0065245820300917.
19. Yesuf M., Assefa B. Model compression techniques in deep neural networks. *Pan African Conference on Artificial Intelligence*. Cham: Springer Nature Switzerland. 2022. P. 169-190. https://link.springer.com/chapter/10.1007/978-3-031-31327-1_10.

20. Lohn A., Scaling AI. *Technical report, Center for Security and Emerging Technology*, 2023. https://cset.georgetown.edu/publication/scaling-ai/.
21. Acun B., Murphy M., Wang X., et al. Understanding training efficiency of deep learning recommendation models at scale. *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE. 2021. https://arxiv.org/abs/2011.05497.
22. Dong K., Zhou C., Rian Y., Li Y. MobileNetV2 model for image classification. *2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE, 2020. P. 476-480. https://ieeexplore.ieee.org/document/9422058.
23. Berthelier A., Chateau T., Duffner S., et al. Deep model compression and architecture optimization for embedded systems: a survey. *J. Signal Processing Systems*. 2021. V. 93. P. 863-878. https://link.springer.com/article/10.1007/s11265-020-01596-1.
24. Hadidi R., Cao J., Xie Y., et al. Characterizing the deployment of deep neural networks on commercial edge devices. *IEEE International Symposium on Workload Characterization (IISWC)*, IEEE. 2019. P. 35-48. https://ieeexplore.ieee.org/document/9041955.
25. Mavrovouniotis S. Ganley M. *Hardware security modules. Secure Smart Embedded Devices, Platforms and Applications*. New York, NY: Springer New York, 2013. P. 383-405.
26. Vembu S. K., Chattopadhyay A., Saha S. Authenticating edge neural network through hardware security modules and quantum-safe key management. *2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)*, Kolkata, India. 2024. P. 318-323. https://ieeexplore.ieee.org/document/10483401.
27. Ezirim K., Khoo W., Koumantaris G., et al. Trusted platform module – a survey. *The Graduate Center of The City University of New York*, 11. 2012. https://www.researchgate.net/profile/Kenneth-Ezirim/publication/287984174_Trusted_Platform_Module_-_A_Survey/links/567af54608ae197583812a7c/Trusted-Platform-Module-A-Survey.pdf.
28. Köylü T.Ç., Wedig Reinbrecht C.R., Gebregiorgis A., et al. A survey on machine learning in hardware security. *ACM Journal on Emerging Technologies in Computing Systems*. 2023. V. 19. P. 1-37. https://dl.acm.org/doi/10.1145/3589506.

REFERENCES

1. R. Szeliski, Computer vision: algorithms and applications. Springer Nature, 2022. 925 p. https://link.springer.com/book/10.1007/978-3-030-34372-9.
2. Hao S., Zhou Y., Guo Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing*. 2020. V. 406. P. 302-321. https://www.sciencedirect.com/science/article/abs/pii/S0925231220305476.
3. Guo Y., Liu Y., Georgiou T., Lew M. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Info Retr*. 2018. V. 7. P. 87-93. https://link.springer.com/article/10.1007/s13735-017-0141-z.
4. Yu H., Yang Z., Tan L., Wang Y., Sun W., Sun M., Tang Y. Methods and datasets on semantic segmentation: a review. *Neurocomputing*. 2018. V. 304. P. 82-103. https://www.sciencedirect.com/science/article/abs/pii/S0925231218304077.
5. Kutsik A. Ya. Analysis of satellite imagery based on semantic segmentation, bachelor thesis. Igor Sikorsky KPI, 2020. https://ela.kpi.ua/items/5c1bdf70-a6a6-45ab-8a63-312296420f6c. [in Ukrainian]
6. Ryabko A.V. Analysis and evaluation of satellite image segmentation methods, Ukrainian Scientific and Technical Conference 'Sustainable Development of Communication, Navigation, Surveillance Systems, and Air Traffic Organization CNS/ATM - 2023», November 29-30. P. 4. https://it-visnyk.kpi.ua/?page_id=2165. [in Ukrainian]
7. Glyboka Yu.O. Investigation of the quality of human image segmentation methods under the influence of additive noise, master thesis, Kharkiv National University of Radio Electronics, 2022. https://openarchive.nure.ua/entities/publication/6bad8449-2d47-4c70-87dc-927980da23e7. [in Ukrainian]
8. Hua Y., Marcos D., Mou L., Zhu X., Tuia D. Semantic segmentation of remote sensing images with sparse annotations. *IEEE Geoscience and Remote Sensing Letters*. 2022. V. 19. P. 1-5. https://arxiv.org/abs/2101.03492.
9. Zhang Y., Chi M. Mask-R-FCN: a deep fusion network for semantic segmentation. *IEEE Access*. 2020. V. 8. P. 155753-155765. https://ieeexplore.ieee.org/document/9151932.

10. O'Mahony N., Campbell S., Carvalho A., et al. Deep learning vs. traditional computer vision, *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)*, Vol. 11, Springer International Publishing, 2020. P. 128-144. https://arxiv.org/abs/1910.13796.
11. Panella F., Lipani A., Boehm J. Semantic segmentation of cracks: data challenges and architecture. *Automation in Construction*. 2022. V. 135. P. 104110. https://www.sciencedirect.com/science/article/abs/pii/S0926580521005616.
12. Mairittha N., Mairittha T., Inoue S. On-device deep learning inference for efficient activity data collection. *Sensors (Basel)*. 2019. V. 19. P. 3434. https://www.mdpi.com/1424-8220/19/15/3434.
13. Cui T. Review of deep learning and mobile edge computing in autonomous driving. *Вісник Львівської політехніки*. 2022. V. 12. P. 208-218. https://science.lpnu.ua/sites/default/files/journal-paper/2023/jan/29757/221029maket-210-220.pdf.
14. Grigorescu S., Trasnea B., Cocias T., Macesanu G. A survey of deep learning techniques for autonomous driving. *J. Field Robotics*. 2020. V. 37. P. 362-386. https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918.
15. Zhang Z., Li J. A review of artificial intelligence in embedded systems. *Micromachines*. 2023. V. 14. P. 897. https://www.mdpi.com/2072-666X/14/5/897.
16. Merone M., Graziosi A., Lapadula V., Petrosino L., d'Angelis O., Vollero L. A practical approach to the analysis and optimization of neural networks on embedded systems. Sensors. 2022. V. 22. P. 7807. https://www.mdpi.com/1424-8220/22/20/7807.
17. Helms D., Amende K., Bukhari S., et al., Optimizing neural networks for embedded hardware. *SMACD/PRIME 2021, International Conference on SMACD and 16th Conference on PRIME*, online. 2021. P. 1-6. https://ieeexplore.ieee.org/document/9547911.
18. Song W. Hardware accelerator systems for embedded systems. *Advances in Computers*, vol. 122. Elsevier, 2021. P. 23-49. https://www.sciencedirect.com/science/article/abs/pii/S0065245820300917.
19. Yesuf M., Assefa B. Model compression techniques in deep neural networks. *Pan African Conference on Artificial Intelligence*. Cham: Springer Nature Switzerland. 2022. P. 169-190. https://link.springer.com/chapter/10.1007/978-3-031-31327-1_10.
20. Lohn A., Scaling AI. *Technical report, Center for Security and Emerging Technology*, 2023. https://cset.georgetown.edu/publication/scaling-ai/.
21. Acun B., Murphy M., Wang X., et al. Understanding training efficiency of deep learning recommendation models at scale. *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE. 2021. https://arxiv.org/abs/2011.05497.
22. Dong K., Zhou C., Rian Y., Li Y. MobileNetV2 model for image classification. *2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE, 2020. P. 476-480. https://ieeexplore.ieee.org/document/9422058.
23. Berthelier A., Chateau T., Duffner S., et al. Deep model compression and architecture optimization for embedded systems: a survey. *J. Signal Processing Systems*. 2021. V. 93. P. 863-878. https://link.springer.com/article/10.1007/s11265-020-01596-1.
24. Hadidi R., Cao J., Xie Y., et al. Characterizing the deployment of deep neural networks on commercial edge devices. *IEEE International Symposium on Workload Characterization (IISWC)*, IEEE. 2019. P. 35-48. https://ieeexplore.ieee.org/document/9041955.
25. Mavrovouniotis S. Ganley M. *Hardware security modules. Secure Smart Embedded Devices, Platforms and Applications*. New York, NY: Springer New York, 2013. P. 383-405.
26. Vembu S. K., Chattopadhyay A., Saha S. Authenticating edge neural network through hardware security modules and quantum-safe key management. *2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)*, Kolkata, India. 2024. P. 318-323. https://ieeexplore.ieee.org/document/10483401.
27. Ezirim K., Khoo W., Koumantaris G., et al. Trusted platform module – a survey. *The Graduate Center of The City University of New York*, 11. 2012. https://www.researchgate.net/profile/Kenneth-Ezirim/publication/287984174_Trusted_Platform_Module_-_A_Survey/links/567af54608ae197583812a7c/Trusted-Platform-Module-A-Survey.pdf.
28. Köylü T.Ç., Wedig Reinbrecht C.R., Gebregiorgis A., et al. A survey on machine learning in hardware security. *ACM Journal on Emerging Technologies in Computing Systems*. 2023. V. 19. P. 1-37. https://dl.acm.org/doi/10.1145/3589506.

Trusov Mykhaylo *Chief software developer, Effective Programming for America, Ukraine, 23 Serpnya Str., 33, Kharkiv, Ukraine, 61045*

Uzlov Dmitro *Associate Professor of the Department of Theoretical and Applied Informatics, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine;*

Perspectives of using deep learning models for semantic image segmentation on autonomous devices

Relevance. The implementation of deep learning models for semantic segmentation on autonomous devices is a promising direction for the development of intelligent systems capable of analyzing visual information without constant connection to external resources. This enables the creation of more autonomous and efficient systems that can operate in real-time and under resource constraints. Such an approach is highly significant for various industries, including robotics, autonomous vehicles, medical diagnostics, and other fields where high accuracy and speed of image processing are required.
Goal. The goal of this work is to explore the possibilities and challenges of using deep learning models for semantic segmentation on autonomous devices. This includes analyzing the efficiency of the models, their adaptation to the limited resources of the devices, and developing methods to ensure the security of access to the trained models.
Research methods. The research methods include theoretical analysis, systematization, and generalization of the use of deep learning models in autonomous devices. Special attention is given to the parameters affecting the memory footprint of the models and the specifics of implementing trained models into proprietary software products. Additionally, modern approaches to encrypting models to ensure their security have been considered.
Results. A comparative analysis of traditional models and deep learning models for semantic segmentation of images has been conducted. Significant potential of deep learning technology for creating autonomous intelligent systems is identified. Various deep learning models currently used for semantic segmentation of images have been reviewed. The impact of key parameters on the efficiency of models on devices with limited resources has been determined, and the role of model size has been considered. Recommendations for implementing trained models into software products are presented, including optimizing models to reduce the size and increase the speed. Special attention has been paid to the analysis of encrypting trained models. It is shown that ensuring the security of access to trained deep learning models for semantic segmentation of images on autonomous devices requires a comprehensive approach that combines hardware and software solutions.
Conclusions. Further developments in the field of deep learning for semantic segmentation on autonomous devices will contribute to the development of more efficient and autonomous systems for a wide range of applications, including computer vision, robotics, and more. Ensuring the security of access to trained deep learning models for semantic segmentation of images on autonomous devices requires a comprehensive approach that combines hardware and software solutions. This not only protects the intellectual property of developers but also ensures the integrity and confidentiality of the data processed by autonomous devices when performing semantic segmentation tasks.

Keywords: *deep learning, semantic segmentation, autonomous devices, model optimization, embedded systems, hardware acceleration, data encryption*

Надійшла у першій редакції 01.05.2024, в останній - 10.06.2024.
The first version has been received on 01.05.2024, the final version – on 10.06.2024.