

## **Використання моделей штучного інтелекту у мовах програмування kotlin, java**

Вінницький національний технічний університет

### **Анотація**

*Пояснено способи використання моделей ШІ на мовах програмування kotlin, java використовуючи бібліотеку DJL.*

**Ключові слова:** *DJL, deep learning, AI, kotlin, java, model.*

### **Abstract**

*Explained how to use AI models in the kotlin, java programming languages using the DJL library.*

**Keywords:** *DJL, deep learning, AI, kotlin, java, model.*

### **Вступ**

Використання ШІ для вирішення поставлених задач набуває все більшої популярності через стрімкий розвиток напрямків та рішень у сфері досліджень штучного інтелекту. Найпопулярнішою мовою програмування для написання алгоритмів навчання та використання моделей ШІ є мова програмування Python, яка через свій простий синтаксис та широку підтримку зі сторони розробників стала інструментом для відображення широкого спектру математичних теорій та формул, на яких базується будь яка модель ШІ. В наш час такі моделі використовують для вирішення великого спектру задач, але найбільш розповсюдженими такі області[1]:

- *Машинне навчання (Machine Learning): алгоритми та моделі, які дозволяють комп'ютерам навчатися з даних і вдосконалювати свої здатності з досвідом.*
- *Обробка природної мови (Natural Language Processing, NLP): Фокусується на розумінні, аналізі та генерації людської мови комп'ютерами.*
- *Комп'ютерний зір (Computer Vision): Дослідження та розробка технологій, які дають комп'ютерам здатність аналізувати та розуміти зображення та відео.*
- *Автоматизація роботи (Robotic Process Automation, RPA): Використання ШІ для автоматизації рутинних бізнес-процесів та задач.*
- *Автономні системи (Autonomous Systems): Розробка систем, які можуть діяти та приймати рішення без прямого втручання людини.*
- *Розпізнавання образів та звуку: Дослідження і розробка систем, які можуть розпізнавати образи або звуки та витягати з них корисну інформацію.*

Використання ШІ значно спрощує виконання деяких задач, робить можливим впровадження нового функціоналу, та економить час та гроші. Але однією із проблем використання будь-яких загальновідомих моделей ШІ є реалізація їх запуску на Python. Більшість розробників все ж дають можливість реалізації запуску своїх моделей на інших платформах шляхом випуску бібліотек для розробки. Метою даної статті є реалізація запуску моделей ШІ з допомогою бібліотеки DJL на мовах програмування kotlin та java.

## Що таке модель ШІ

Перш ніж перейти до реалізації слід розібрати сам предмет нашої роботи, а саме модель ШІ.[2] Модель ШІ – це алгоритм, або ж конструкція алгоритмів, який використовується для розуміння або емуляції процесів. Відкривши модель у програмі для візуалізації нейронних мереж ми можемо побачити що модель представляє собою набір зв'язаних блоків що мають свої входи(input), та виходи(output). Кожен із цих блоків має свою роль та важливість під час виконання системи алгоритмів

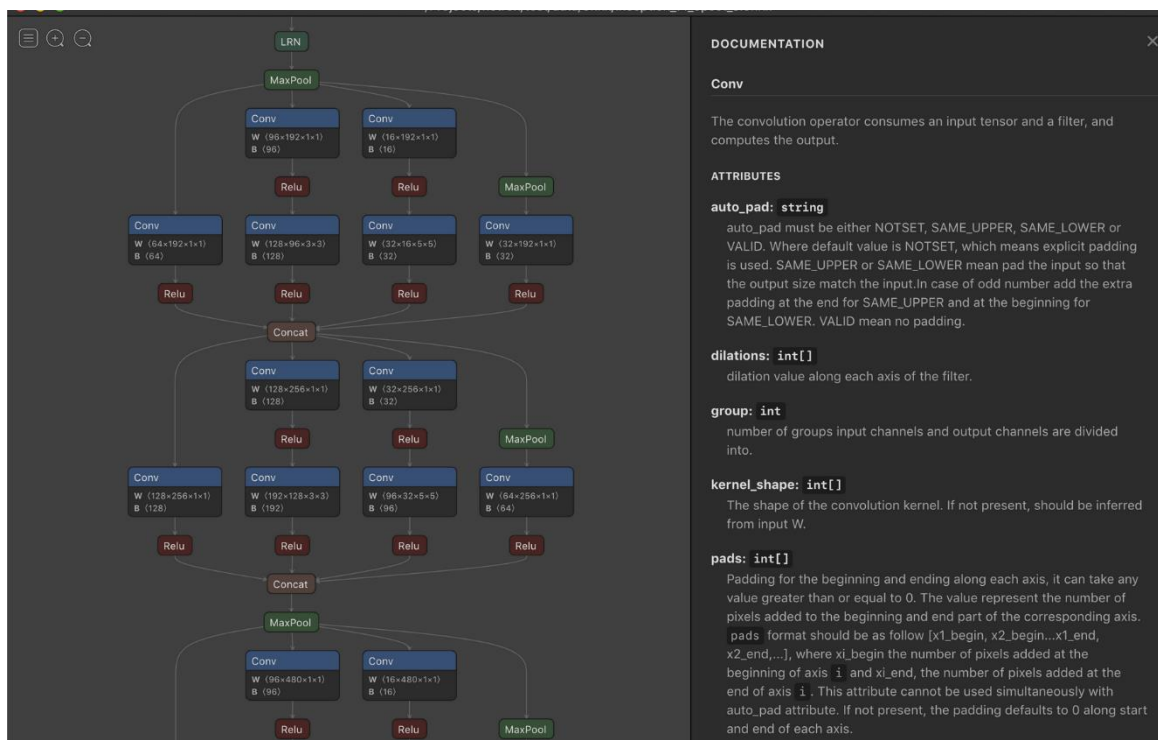


Рис. 1 – Візуалізація моделі ШІ

Кожен алгоритм алгоритм має свій Input та мінімум один Output, за якими і здійснюється опис даної моделі для подальшого її запуску. На приклад моделі, що працюють із фото для Input приймають матриці даних для подальшої їх обробки і часто мають значення float[]. Самі ж моделі часто поділяються на “Легші” та “Важчі”, і мають свою градацію залежно від розробника. “Легші” моделі відрізняються від “Важчих” меншою кількістю алгоритміки, або використанням більш спрощених формул та функцій, що призводить до зменшення її ваги, навантаження на залізо, але з іншої сторони зменшення точності на більших вимогах до набору даних для навчання та валідації.

## Реалізація запуску моделі ШІ

Для запуску моделі ШІ на java або kotlin використаємо бібліотеку DJL, яка є системою для запуску підготовлених для неї двигунів, які є сформованими пакетами для розробки, що постачаються самими авторами різних моделей ШІ, та підключаються окремо. Для прикладу будемо використовувати двигун PyTorch та навчену на розпізнавання об'єктів модель ШІ, яку заздалегідь перевели у формат .pytorch та прописали synset.txt файл, у якому записані усі об'єкти які вміє розпізнавати дана модель.

1. Створюємо новий об'єкт класу Criteria, який описує основні параметри моделі та її поведінку. Сама Criteria є generic класом, що реалізовує паттерн builder, який і дозволяє налаштувати конфігурацію. Важливо пам'ятати input та output нашої моделі для правильної інтерпретації наших даних, так як generic типи цього класу є input та output для нашої моделі
2. Так як наша модель приймає фіксовану матрицю 640x640 потрібно додати pipeline що буде змінювати будь яке зображення під прийнятний для моделі набір даних
3. Для правильної поведінки моделі та правильного розпізнавання її input та output використовуємо клас Translator, який описує поведінку з наданими та отриманими даними. DJL постачає великий вибір трансляторів, але нічого не заважає зробити свій
4. Прописуємо шлях до моделі та шлях до її synset.txt

```
21 @Configuration
22 class ProcessConfig {
23     @Value("${image.scale.size}")
24     private var scaleSize: Int = 0
25
26     @Bean
27     fun criteria(): Criteria<Image, DetectedObjects> {
28         val pipeline = Pipeline()
29         pipeline.add(Resize(scaleSize))
30         pipeline.add(ToTensor())
31         val translator: ObjectDetectionTranslator = YoloV5Translator
32             .builder()
33             .setPipeline(pipeline)
34             .optThreshold(0.8f)
35             .optSynsetArtifactName("anonymize_model/synset.txt")
36             .build()
37         return Criteria.builder()
38             .setTypes(Image::class.java, DetectedObjects::class.java)
39             .optProgress(ProgressBar())
40             .optModelUrls("IdeaProjects/teza/src/main/resources/")
41             .optModelName("anonymize_model/anonymize.torchscript")
42             .optEngine("PyTorch")
43             .optTranslator(translator)
44             .build()
45     }
46 }
```

Рис. 2 – Приклад Criteria для використовуваної моделі

Після написання правильної конфігурації ми можемо запустити модель використовуючи клас ModelZoo

```
fun detect(images: Set<Path>): Map<Path, DetectedObjects> {
    val imageFactory = ImageFactory.getInstance()
    val model = ModelZoo.loadModel(criteria)
    val predictor = model.newPredictor()
    val map: MutableMap<Path, DetectedObjects> = images.stream()
        .collect(Collectors.toMap({ it }, {
            predictor.predict(imageFactory.fromFile(it))
        }))
    model.close()
    log.info("Model successfully loaded, started prediction!")
    return map
}
```

Рис. 3 – запуск моделі

## Висновок

Розглянуто спосіб запуску моделей ШІ на мовах програмування java та kotlin використовуючи бібліотеку DJL. Обговорено принципи роботи моделей ШІ та застосування їх для вирішення задач.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Stuart J. Russell, Peter Norvig. Artificial Intelligence: A Modern Approach.* — 3. — Pearson, 2015. — [ISBN 978-9332543515](#). (дата звернення: 19.03.2024).
2. Piuri, Vincenzo; Shaw, Rabindra Nath; Ghosh, Ankush; Islam, Rabiul (2022). *AI and IoT for smart city applications.* Singapore. [ISBN 978-981-16-7498-3](#). (дата звернення: 19.03.2024).
3. *DJL official docs* — <https://djl.ai/website/javadoc.html> (дата звернення: 19.03.2024).

**Слободян Ігор Олександрович** – студент групи ІАКІТР-23м, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [ihor.slob@gmail.com](mailto:ihor.slob@gmail.com)

Науковий керівник: **Кветний Роман Наумович** – професор кафедри Автоматизації та інтелектуальних інформаційних технологій, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [rkvetny@vntu.edu.ua](mailto:rkvetny@vntu.edu.ua)

**Slobodyan Ihor O.** — Department of Intellectual Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail : [ihor.slob@gmail.com](mailto:ihor.slob@gmail.com)

Supervisor: **Kvetny Roman N.** - professor of the Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [kulyk.y.a@vntu.edu.ua](mailto:kulyk.y.a@vntu.edu.ua)