

Рис.1. Реалізація проекту «Завод з виробництва автомобілів»

Клієнтський код повинен працювати як із фабриками, так і з продуктами тільки через їхні загальні інтерфейси. Це дозволить подавати у ваші класи будь-які типи фабрик і виробляти будь-які типи продуктів, без необхідності вносити зміни в існуючий код.

Патерн Абстрактна фабрика використовують, коли необхідно змінювати поведінку системи, зберігаючи інтерфейси та варіюючи об'єктами які створюються. Він дозволяє створювати групи взаємопов'язаних об'єктів, що реалізують загальну поведінку.

Висновки. Використання патернів проектування є необхідним при створенні програмного забезпечення, тому що надає можливість спроектувати оптимальну роботу майбутнього проекту та його коду.

Використання абстрактної фабрики має такі переваги: ізолює конкретні класи; спрощує заміну сімейств продуктів; гарантує поєднання продуктів; система не повинна залежати від того, як створюються, компонуються і представляються об'єкти, з яких вона складається; взаємопов'язані об'єкти, що входять в сімейство, повинні використовуватися разом і необхідно забезпечити виконання цього обмеження.

ЛІТЕРАТУРА

1. Проектирование программного обеспечения [Електронний ресурс] – Режим доступу: <https://unetway.com/tutorial/proektirovanie-programmnogo-obespecenia>
22. Шаблоны проектирования простым языком. Часть первая. Порождающие шаблоны [Електронний ресурс] – Режим доступу: <https://tproger.ru/translations/design-patterns-simple-words-1/>
23. Шаблоны проектирования простым языком. Часть вторая. Структурные шаблоны [Електронний ресурс] – Режим доступу: <https://tproger.ru/translations/design-patterns-simple-words-2/#22>
24. Шаблоны проектирования простым языком. Часть третья. Поведенческие шаблоны [Електронний ресурс] – Режим доступу: <https://tproger.ru/translations/design-patterns-simple-words-3/#32>
25. Шаблон проектирования [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/Шаблон_проектирования

РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ У ВІДЕОПОТОКАХ РЕАЛЬНОГО ЧАСУ

Кітєв О.М., факультет інформаційних технологій та дизайну, група 6зІР2, Херсонський національний технічний університет, м.Херсон, Україна,

Огнева О. Є., к.т.н., доцент кафедри програмних засобів і технологій, Херсонський національний технічний університет, м.Херсон, Україна

Вступ. Комп'ютерний зір, або здатність штучних інтелектуальних систем «бачити» як люди, протягом десятиліть викликає підвищений інтерес і ретельні дослідження. Як спосіб наслідування зорової системи людини,

дослідження в області комп'ютерного зору спрямовані на розробку машин, які можуть автоматизувати завдання, що вимагають візуального пізнання [1-3].

Постановка задачі. Смарт техніка швидко розвивається. Тепер із застосуванням комп'ютерного зору машина може ще й "бачити", орієнтуватися у навколишньому середовищі. Так званий штучний інтелект та машинне навчання дають змогу на розробку системи розпізнавання зображень та виявлення різних об'єктів [2-3].

Одною із актуальних сфер застосування є дорожній транспорт, а саме управління автономними транспортними засобами шляхом виявлення перешкод, людей і дорожніх знаків для забезпечення максимальної безпеки дорожнього руху. Ця властивість знаходить своє застосування в автомобілях із допоміжним керуванням та самокерованому транспорті. Імплементация цієї технології зменшує ризик аварій та дозволяє автоматизувати виробничий транспорт [4-5].

Наразі складність полягає у тому, що обробка графічної інформації в реальному часі потребує значних обчислень, а точність детектора має бути якомога більшою. Тому необхідні ефективні алгоритми та техніки. Реалізація саме таких ефективних рішень є темою даного дослідження.

Викладення основного матеріалу. Процес розшифровки зображень через великий обсяг багатовимірних даних, які потребують аналізу, є набагато складнішим, ніж інші форми обробки бінарної інформації. Він потребує об'ємних обчислень. Система повинна виявити, класифікувати об'єкти та окреслити їх межі. До того ж, в контексті руху автомобілів розпізнавання образів повинно відбуватися у режимі реального часу, щоб мати під контролем ситуацію на дорозі [2-5].

В якості вирішення проблеми представлено додаток, який постійно виявляє об'єкти (обмежувальні бокси та класи) у кадрах, які бачить задня камера пристрою, використовуючи квантовану модель SSD MobileNet.

Модель нейронної мережі розроблена на базі фреймворку TensorFlow. Інтерфейс додатку написан за допомогою Java, а нейронна мережа – на мові Python. У проекті використовується середовище розробки Android Studio для створення та розгортання програми на мобільному пристрої.

Характеристики детектора: швидкість обробки одного кадру 29 мс = 34 FPS (кадрів на секунду); **точність** mAP = 18 ; Максимальна кількість виявлених об'єктів в одному кадрі – 10.

Робота додатку була протестована в реальних умовах застосування, тобто вулицях та дорогах міста.

Результати зйомки представлені на рис.1.

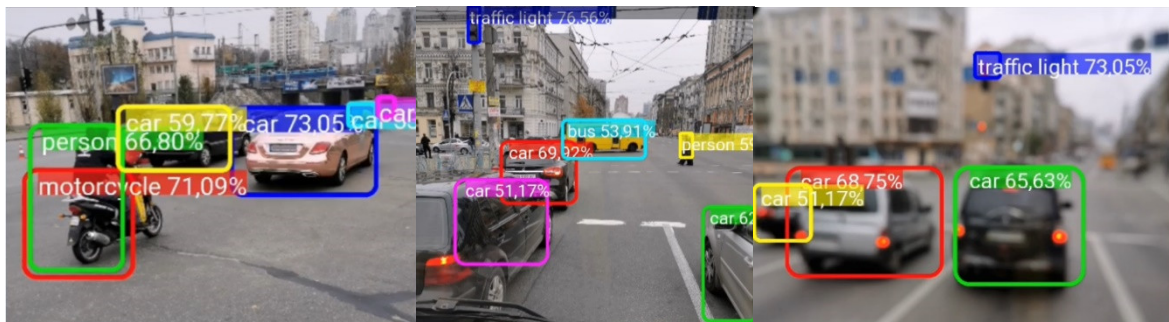


Рис.1. Презентація роботи системи

Висновки. Досліджена галузь є дуже молода, а її розвиток відбувається великими кроками. Зараз кілька разів на рік у світ виходить нова модель класифікації зображень та детекції об'єктів із кращими показниками.

В основі реалізованої системи детекції є нейронна мережа MobileNet, архітектура якої має відносно малу кількість параметрів. Разом із швидким методом детекції об'єктів SSD та фреймворком Tensorflow Lite, що дозволяє зменшити розмір моделі через заміну операцій з плаваючою точкою на цілі числа фіксованої довжини, система успішно розгортається у додатку смартфона.

ЛІТЕРАТУРА

1. Artificial neuron // Режим доступу: <https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e>
2. Convolutional neural network structure // Режим доступу: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
3. M.Nielsen How the backpropagation algorithm works // Режим доступу: <http://neuralnetworksanddeeplearning.com/chap2.html>
4. Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard. Handwritten digit recognition: Applications of neural net chips and automatic learning. 1989
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2012.