

ПІДХІД ДО РОЗПІЗНАВАННЯ СТУДЕНТІВ ЗА ФОТО ДЛЯ СИСТЕМИ МОНІТОРИНГУ ВІДВІДУВАННЯ ЗАНЯТЬ

Садовников Б.І.

Науковий керівник – к.т.н., доцент Мазурова О.О.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки,14, каф. Програмної інженерії,

The main aim of this work is to create an optimal algorithm of person recognition in context of student attendance monitoring system. Recognition in real time of a large number of people requires quit high amount of hardware resources. However, there are number of basic optimization methods that can change it. We will discuss complex measures which involve more than one optimization method. In order to resolve this issue we will develop custom approach especially for our system. Basics of this approach are distribution of photo preprocessing work between IoT devices and keeping balance between possible losses of data and accuracy.

На сьогоднішній день освіта грає важливу роль у житті суспільства, її високий рівень є основною запорукою процвітання суспільства. Одним із підходів до підвищення ефективності навчального процесу є автоматизація рутинної роботи, наприклад, поіменної перевірки наявності на занятті студентів.

Таким чином, актуальною задачею є розробка системи моніторингу відвідування занять студентами. Ключовою задачею для такої системи є розпізнавання студентів за фото, а з урахуванням високих навантажень, ще й оптимізації такого розпізнання.

Для отримання максимальної точності розпізнавання було запропоновано використовувати нейронну мережу, яка дозволить на основі фото, що можуть бути отримані з IoT приладу, визначати наявність студентів на заняттях. Під час навчання нейронної мережі необхідно проводити попередню обробку зображень, таку як повороти, віддзеркалення, масштабування, зміщення. Це підвищить вірогідність розпізнання на фото з частково спотвореним обличчям.

Була запропонована наступна архітектура програмної системи:

- REST API для взаємодії з клієнтами та IoT приладом;
- сервер бази даних; за результатами аналізу та моделювання предметної області була розроблена схема БД, яка містить інформацію про предмети, студентів, викладачів, розклад занять, оцінки, присутність студентів на заняттях;
- сервіс для розпізнавання студентів; тут зберігаються фото студентів для можливого перенавчання нейронної мережі, що може статися у випадку зміни її реалізацію або у випадку збоїв з боку апаратного забезпечення.

Отже, для вирішення задачі розпізнання студентів за фото було запропоновано наступний підхід:

- перед відправкою фото на розпізнавання, IoT пристрій перевіряє його на наявність обличчя[1]. Його наявність перевіряє тренерований класифікатор за ознаками Хаара. Основною ознакою є наявність на фото двох суміжних регіонів у області щік та очей;

- у випадку наявності вилучити обличчя з фото;

- провести нормалізацію для підвищення ефективності розпізнання.

Оскільки нейронні мережі працюють з відносними значеннями кольорів, а саме у діапазоні від 0 до 1. Крім цього, має сенс перевести зображення у відтінки сірого. Ми зменшимо кількість даних які має обробити нейрона мережа на $\frac{3}{4}$ (замість трьох значень кольору та значення прозорості ми отримаємо значення інтенсивності). Це не значно зменшить точність розпізнавання у обмін на досить великий приріст продуктивності;

- розпізнати обличчя за допомогою нейронної мережі.

Для реалізації запропонованого підходу була розроблена програмна система моніторингу відвідувань занять, що містить:

- серверну частину, створену за технологією Golang;

- сервер БД на базі MySQL та Redis для кешування інформації про поточні заняття;

- міні комп'ютер Raspberry Pi 3 B з операційною системою Raspbian в якості IoT пристрою;

- додаток для IoT написаний на C++;

- клієнтська частина системи – веб сайт з використанням технологій ReactJS, HTML, CSS;

- Android-додаток з використанням Kotlin.

Список використаних джерел

1. Szeliski, R. Computer Vision: Algorithms and Applications. — Berlin: Springer, 2010. — 812 p.;

2. Haykin, S. Neural Networks and Learning Machines. – Pearson: New Jersey, 2008 – 936 p.