

«Штучний інтелект може сприяти персоналізованому підходу, — зазначив директор Центру ігрової науки та засновник Enlearn Зоран Поповіц. — Він здатний надати той навчальний план, якого учень потребує в цю мить».

Отже, навчання на основі штучного інтелекту, тільки починає свій шлях, даний спосіб поки що досить рідко використовують у навчанні, але вчителі уже мають можливість спробувати функціонал цього надзвичайно потужного інструменту. Вже є готові апаратні і програмні інструменти які можна і потрібно застосовувати. Штучний інтелект в майбутньому дозволить знайти індивідуальний підхід до кожного учня та максимально ефективно проводити навчальний процес.

Список використаних джерел

1. Як штучний інтелект може допомогти освіті [Електронний ресурс]. – Режим доступу: <https://osvitoria.media/experience/yak-shtuchnyj-intelekt-mozhe-dopomogty-osviti/> (дата звернення 18.10.2019).
2. Подуфалов М. С. Використання штучного інтелекту в розвиваючих комп'ютерних іграх [Електронний ресурс]. – Режим доступу: [file:///C:/Users/Vasya/Downloads/2134-7746-1-PB%20\(1\).pdf](file:///C:/Users/Vasya/Downloads/2134-7746-1-PB%20(1).pdf) (дата звернення 28.10.2019).
3. Перспективи для штучного інтелекту [Електронний ресурс]. – Режим доступу: <https://reklamaster.com/marketing-and-advertising/perspektivi-dlja-shtuchnogo-intelektu-shho-gotuie-2019/> (дата звернення 18.10.2019).

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ОПТИЧНОГО РОЗПІЗНАВАННЯ ТЕКСТУ ЗА ДОПОМОГОЮ FIREBASE ML KIT

Велещук Олександр Іванович

магістрант спеціальності «Середня освіта. Інформатика»,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
velsashok@gmail.com

Карабін Оксана Йосифівна

кандидат педагогічних наук, доцент кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
karabinoksana@gmail.com

Державні структури, бізнес, наукові та навчальні організації використовують велику кількість паперових документів. Тому існує потреба в оцифруванні таких документів, аналізі і обробці інформації яка в них міститься. Для вирішення цієї задачі використовують оптичне розпізнавання символів (англ. optical character recognition, OCR) — це механічне або електронне переведення зображень рукописного, машинописного або друкованого тексту в послідовність кодів, що використовуються для представлення в текстовому редакторі [1]. Отриману текстову інформацію використовують для: пошуку по ключових словах, перекладу або озвучення тексту, збереження інформації в компактнішій формі

Система оптичного розпізнавання символів є доволі складною. Потрібно пройти декілька етапів:

Знайти позицію тексту на зображенні.

Здійснити сегментацію тексту, тобто розбиття забраження на рядки, слова і окремі символи.

Здійснити класифікацію окремих символів. На цьому етапі зображення окремих символів передаються алгоритму класифікації який здійснює розпізнавання.

Щоб покращити точність використовують словники. При спробі розпізнати наступний символ враховується те, як були розпізнанні попередні.

Оскільки реалізація такої системи вимагає багато часу і зусиль, на практиці, зазвичай, використовують готові рішення. Одним з таких рішень є Firebase ML Kit — це набір із засобів розробки (SDK) від Google, який дозволяє використовувати машинне навчання у android і ios додатках. Для початку роботи із ML Kit не потрібно мати глибокі знання в галузі нейронних мереж чи оптимізації моделей для мобільних пристроїв. З іншого боку, для досвідчених розробників, ML kit надає можливість використовувати власні TensorFlow Lite моделі.

Основні можливості:

Готові рішення для типових задач — ML Kit надає набір готових для використання API для поширених задач, а саме: розпізнавання тексту, виявлення обличчя, розпізнавання орієнтирів (англ. *identifying landmarks*), розпізнавання штрих-кодів, визначення мови тексту, розпізнавання об'єктів [3].

Обчислення на пристрої або сервері — виконання обчислень на пристрої дозволяє програмі працювати швидко та без інтернет з'єднання. При цьому зростає розмір програми, оскільки вона міститиме модель машинного навчання. Виконання обчислень на сервері дозволяє досягти вищої точності завдяки доступу до більшого об'єму даних і використанню складніших моделей [3].

Можливість використовувати власні моделі — якщо існуючих засобів ML kit не вистачає для вирішення певної проблеми є можливість використати власну TensorFlow Lite модель [3].

Підключення Firebase ML kit

Спочатку потрібно створити новий android проект. Після цього потрібно створити проект на сайті <https://console.firebase.google.com/>. Отриманий файл конфігурації *google-services.json* потрібно помістити у папку *app* android проекту. Слідуючи інструкціям в проект потрібно додати декілька залежностей: *google-services* plugin і *firebase-ml-vision*. Для того щоб виконувати розпізнавання на пристрої потрібно додати спеціальний мета-тег в *AndroidManifest.xml*.

```
<application ...>
...
<meta-data
  android:name="com.google.firebase.ml.vision.DEPENDENCIES"
  android:value="ocr" />
</application>
```

Рис. 1. Приклад мета-тегу який вказує що потрібно завантажити модель після встановлення додатку

Після встановлення додатку, модель буде автоматично завантажена. Якщо не додавати цей запис, програма почне завантаження моделі під час першого запуску

процесу розпізнавання. Спроби розпізнавання тексту здійснені до завершення завантаження не повернуть жодного результату.

Розпізнавання тексту

Для того щоб розпізнати текст потрібно на основі зображення створити об'єкт *FirebaseVisionImage*. Для його створення можна використати *Bitmap*, *media.Image*, *ByteBuffer*, або шлях до файлу на пристрої. Створений екземпляр *FirebaseVisionImage* потрібно передати у метод *processImage* об'єкта *FirebaseVisionTextRecognizer*.

```
private fun recognizeTextFromImage() {
    1 val firebaseVisionImage = FirebaseVisionImage.fromFilePath(this, photoFileUri)
    2 val detector = FirebaseVision.getInstance().onDeviceTextRecognizer
    3 detector.processImage(firebaseVisionImage)
    4 .addOnSuccessListener { firebaseVisionText ->
        displayRecognizedText(firebaseVisionText)
    }
    .addOnFailureListener { e ->
        Toast.makeText(context: this, e.message, Toast.LENGTH_LONG).show()
    }
}
```

Рис. 2. Приклад використання *FirebaseVisionImage* та *FirebaseVisionTextRecognizer*

На рисунку 2 зображено:

Створення об'єкта *FirebaseVisionImage*

Створення об'єкта *FirebaseVisionTextRecognizer*

Виклик метода *processImage*

Після успішного розпізнавання буде викликаний обробник події *success*.

Як аргумент обробнику події *success* буде передано об'єкт *FirebaseVisionText*. Цей об'єкт містить весь текст розпізнаний на основі зображення, також він містить список об'єктів *TextBlock*. Кожен *TextBlock* представляє прямокутний блок тексту, який містить список об'єктів *Line*. Кожен об'єкт *Line* у свою чергу містить список об'єктів *Element*, який представляє окреме слово (сюди входять також дати, числа і т. д.). Варто взяти до уваги що списки об'єктів *TextBlock*, *Line*, *Element* можуть бути порожніми. Для кожного об'єкта (*TextBlock*, *Line*, *Element*) можна отримати текст а також координати прямокутника який містить цей текст [4].

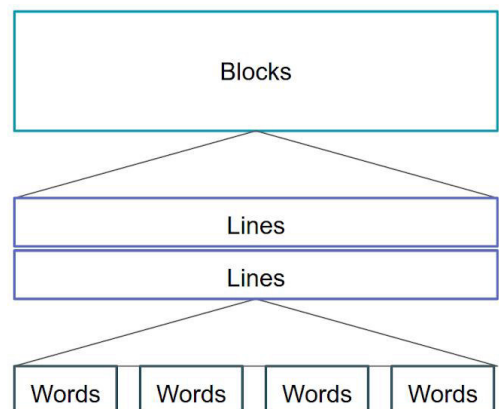
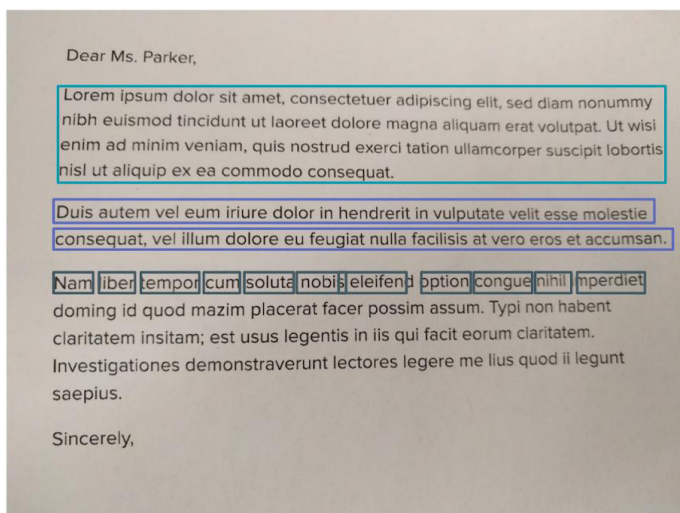


Рис. 3. Ієрархія об'єктів для представлення тексту розпізнаного із зображення

```
private fun displayRecognizedText(firebaseVisionText: FirebaseVisionText) {
    val textBlocks = firebaseVisionText.textBlocks

    if (textBlocks.size == 0) {
        Toast.makeText(
            context: this,
            text: "Текст не виявлено, спробуйте ще раз",
            Toast.LENGTH_LONG
        ).show()
    } else {
        for (block in textBlocks) {
            editText.append(block.text)
        }
    }
}
```

Рис. 4. Приклад роботи із результатом розпізнавання тексту



Рис 5. Скріншот готового додатку

У процесі дослідження розглянуто актуальність оптичного розпізнавання тексту. Наведено 4 етапи із яких складається процес розпізнавання. Розглянуто набір засобів розробки (SDK) Firebase ML Kit, який надає готові рішення для типових задач машинного навчання, в тому числі розпізнавання тексту. Описано спосіб підключання ML Kit до android проекту. Описано мета-тег який дозволяє завантажити модель на пристрій після інсталяції додатку. Наведено приклад використання класів *FirebaseVisionImage* і *FirebaseVisionTextRecognizer* для розпізнавання тексту із зображення. Також описано класи які представляють результат процесу розпізнавання. Наведено приклад їх використання. Результатом дослідження є мобільний додаток для оптичного розпізнавання тексту.

Список використаних джерел

1. Оптичне розпізнавання символів. URL: https://uk.wikipedia.org/wiki/Оптичне_розпізнавання_символів. (дата звернення: 29.09.2019).
2. Add Firebase to your Android project. URL: <https://firebase.google.com/docs/android/setup>. (дата звернення: 29.09.2019).

3. Recognize Text in Images with ML Kit on Android. URL: <https://firebase.google.com/docs/ml-kit/android/recognize-text>. (дата звернення: 29.09.2019).

4. Text Recognition API Overview. URL: <https://developers.google.com/vision/android/text-overview>. (дата звернення: 29.09.2019).

ПРОБЛЕМА ПІДВИЩЕННЯ ЯКОСТІ НАВЧАННЯ СИСТЕМ 3D ПРОЕКТУВАННЯ ЯК ЗАСОБУ ПРОФЕСІЙНОЇ ДІЯЛЬНОСТІ МАЙБУТНІХ ДИЗАЙНЕРІВ

Гірняк Наталя Петрівна

магістрантка спеціальності 014.09 Середня освіта (Інформатика),
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
kavka1973@ukr.net

Поява нових інформаційних комп'ютерних технологій, їх розвиток та удосконалення, проникнення в усі сфери людської діяльності обумовлюють переосмислення і трансформацію чинних підходів до підготовки майбутніх фахівців, зокрема професійної підготовки дизайнерів.

Інтенсивного розвитку та поширення набувають комп'ютерні графічні системи. З їх допомогою можна здійснювати проектувальну діяльність та створювати зображення, які практично неможливо відрізнити від реальних об'єктів. Комп'ютерні технології дозволяють виконувати складні проекти на основі самої лише ідеї. Це економить матеріальні, часові та інтелектуальні затрати. Актуальним на сьогодні є використання систем тривимірного (3D) комп'ютерного проектування, за допомогою яких не тільки створюють віртуальні об'єкти та 3D зображення, але й, використовуючи технології тривимірного друку, втілюють їх у реальність. Це і зумовлює велику поширеність комп'ютерних графічних технологій та їх застосування в усіх галузях людської діяльності: дизайні, інженерії, освіті, мистецтві, архітектурі, тощо.

Питання графічної підготовки у педагогічній літературі та наукових працях розглядають у різних напрямках. Так, загальні аспекти розвитку просторового мислення та просторових операцій висвітлюють Н. Бондар, А. Корнєєва, І. Нищак, О. Райковська, Ю. Фещук; формування графічних знань і вмінь за допомогою інформаційних технологій обґрунтовують П. Буянов, О. Глазунова, Р. Горбатюк, М. Козяр, В. Кондратова, Н. Поліщук, Ю. Рамський, І. Семенов, М. Юсупова; проблеми твердотілого об'ємного комп'ютерного моделювання вирішують П. Бездітко, В. Вірченко, Я. Кіницький, А. Краснюк, О. Крестьянполь, М. Мазур, В. Малащенко, Я. Підгайчук, М. Семенюк, О. Стрілець та інші; процес комп'ютерної анімації розглядають учені В. Головань, О. Дроздов, Д. Одновол, І. Теплицький, О. Теплицький, О. Шаповал; роботу з елементами комп'ютерного дизайну висвітлюють науковці С. Вяткін, Д. Захаренко, В. Карабчевський, С. Магдаліна, О. Романюк, К. Ручкін, А. Тимошенко; комп'ютерний графічний дизайн розглядають Г. Веселовська, В. Даниленко, В. Железнякова, В. Молочкова, М. Рябчиков, Б. Шашлова та інші. Незважаючи на значну кількість досліджень, проблеми навчання систем тривимірного