INTRO

Optical Character Recognition, or OCR, is a technology used to convert any kind of text or information stored in a digital document into machine-readable data.

Hence, **paper documents** and **hard copies** can be converted into computer-readable formats, allowing for further editing or data processing, **CHANGE** helping enterprises to move toward a paperless office.

CHANGE

With technology advancement, OCR tools have been increasingly relied upon to

- Extract text from images,
- data from PDF documents and
- to convert it to different file format.

CHANGE

It is now available on the Internet in the form of cloud-based services such as Cloud vision and API that integrate seamlessly with applications.

CHANGE

Modern OCR software uses AI and Machine Learning capabilities to perform even more advanced functions, such as

- identifying multiple languages,
- reading handwritten texts,
- writing styles, and
- handling common data constraints to list a few.

In order to increase the accuracy of OCR, deep learning has been incorporated into recent approaches to address the **wide variety of styles** of handwriting and printed material.

CHANGE

As deep learning requires a large amount of data to train a model, companies such as Google with large databases and high consumer traffic gain an advantage in producing good results with their OCR services.

CHANGE

PROJECT Brief & methodology

My application, OCR Card Reader makes use of standardized, pre-trained machine learning models

CHANGE

provided through the Google ML Kit (API) implemented in Android Studio IDE,

with the goal of automating the entry of data from **business cards** into the **device** contact using optical character recognition.

CHANGE

MLKIT

The ML Kit runs on devices, making real-time functionality available offline. It integrates machine learning into Android applications, with APIs for **Vision** and **Natural Language** to facilitate data extraction, which revolves around two main processes:

- Optical Character Recognition (OCR) followed by
- Natural Language Processing (NLP).

The first, is the process of converting text images into machine-encoded text, while the latter is the analysis of words to determine their meanings.

CHANGE

Computer vision techniques **such as** line and box detection are often used in conjunction with OCR, to extract the aforementioned data types such as tables and key-value pairs for broader comprehensive extraction.

CHANGE

In computer vision, convolutional neural networks (CNNs) are commonly used as a form of neural network variation, since they work in tensors (or high-dimensional vectors) rather than traditional layers for feature extraction. In conjunction with traditional network layers, CNNs perform very well in image-related tasks, and further provide the foundation for OCR and other feature recognition.

CHANGE

While images are independent of one another, text prediction is greatly improved by considering words before or after them., Alternatively, Natural language processing (NLP) relies on a different set of networks that analyze time series data.

LSTMs, a class of networks that use past data to predict future data, have gained in popularity in recent years. **Additionally**, bidirectional LSTMs are frequently used to improve prediction performance, considering both results prior to and after the prediction.

CHANGE

Data extraction mainly aims to convert data from unstructured documents into structured formats, with highly accurate retrieval of

- text,
- figures, and
- data structures contributing greatly to **statistical** and **contextual analysis**.

CHANGE

OCR Card Reader

The **OCR Card Reader** digitize visiting cards and convert their information into text, which can then be entered as a contact in the user's phone. The core part of the program includes the **text recognition** and **entity extraction**.

CHANGE

Until recently, both of these parts were achieved through OpenCV or TensorFlow lite, which included a steep learning curve and **required expertise** in machine learning and natural language processing. Putting this together was a tedious and time-consuming task that required tons of effort.

CHANGE

As a game changer, Google now powered ML KIT blends machine learning expertise with mobile developers in an easy-to-use package, enabling apps to become more

- engaging,
- personalized, and
- helpful with solutions that are optimized for mobile.

With its powerful machine learning models and advanced processing pipelines, it enables the use of powerful use cases within the application with its easy-to-use APIs.

CHANGE

Overview of the project working

CHANGE

When the application is first launched, the camera preview page is displayed, and a prompt is shown asking for the user's permission to use the camera and the device storage.

CHANGE

In response to receiving permission, the application launches the device's camera, and the user can select a card to initiate the **scanning process** by taking a picture. A progress dialog box is displayed, as the image is analyzed for text recognition.

If text is detected, a form containing the contact details is displayed, otherwise the user is notified that no text was detected.

CHANGE

The user can then edit the form as needed and choose to save the contact, which directs them to a system contact saving confirmation followed by a successful save toast message.

CHANGE

Project Structure

In implementation, an OCR Card reader project file consists of three main components, the AndroidManifest.xml file, the source code files, and the layout files.

CHANGE

In the androidmanifest.xml file, the hardware permission required by the application is defined first, followed by a declaration that ensures that the ml model is automatically downloaded from the MLKit to the device.

The app uses the following APIs to support its various functionalities,

- **ML KIT API**: to recognize text,
- **CameraX**: to input images,
- **Entity Extraction API**: to extract entities from text,
- **Activity Result API**: to capture results.

Additionally, I synchronized the project build by adding the required repositories and dependencies for the APIs.

CHANGE

There are three main sections of the application design: the main activity, where the user scans the card and is redirected to the form, the language selector interface, and the progress dialogue box.

CHANGE

A. Main Activity

In this section the navigation API is utilized, allowing users to interact with the application through actions. Implementation involves creating a navigation graph resource file that contains destination and actions, and the graph represents the navigation path from the camera to the component displaying the recognized text

CHANGE

B. Language selector interface

The application supports Latin-based languages for text recognition, and users can change their preferred language by selecting change language button which displays the language selection interface.

CHANGE

C. Progress dialogue box

Once the photo has been taken, a progress dialog box appears with an animated loading progress bar as the text is extracted

CHANGE

//Source Code Structure

The application can be decomposed into modules like

- Camera,
- Entity Extraction,
- Contact Saving,
- and Changing Language, in order to provide a better understanding of its source code structure.

CHANGE

Camera,

This module handles the camera functionality of the application. It is built using the cameraX library, which takes advantage of camera2's capabilities, while maintaining a simpler, use case-based approach that's lifecycle-aware, to provide a consistency and easy-to-use API surface for most Android devices. Use cases that I implemented include:

- **Preview:** This feature allows users to view what the camera is seeing.
- **Image Capture:** This allows user to click and save images.
- **Image Analysis:** This case provides you with a buffer of images that can be accessed and used in a machine learning task.

CHANGE

Entity Extraction

In this module, entities are precisely recognized by using the Entity Extraction API, which supports the following entities.

CHANGE

In addition to several languages supported for **entity extraction**, **language selection** determines address and phone number detection.

CHANGE

Contact Saving

This module saves contact information to the device contact list using the Contact provider API. The Contact Provider is an Android component that manages the device's central repository of data about people.

It maintains three types of information about a person, each corresponding to a table offered by the provider.

The three tables each have their own contract classes, which specify constants for content

URIs, column names, and column values;

Individuals are represented by a **contact table**, **raw contact tables** hold information about individuals grouped by user account and type, and the **data tables** contain information such as email addresses and phone numbers.

CHANGE

Changing Language,

In this module, everything related to the language is handled. It listens for the change language button and updates the text **recognition** language preferences when an action is triggered,

CHANGE

Demonstration of application test run is as follows:

For the purpose of demonstrating the functionality of the application, I ran 5 test runs for cards with different layouts and languages.

CHANGE

In the **first test** run, we test the app permissions, text recognition of the card's characters, entity extraction, and a prompt to remind the user to enter the name and phone number.

CHANGE

In the **second test** run I checked text recognition with different alignments and no text detection alert.

CHANGE

In the **third test** run I checked how the data for an individual would be saved with first name, se, and surname,

CHANGE

and the **fourth test** run was to check the model's accuracy. **Lastly**, I checked the working of text recognition for German language, and the language change interface and notification message also how the data for an individual would be saved with first name, second name, and surname,

CHANGE

Advantages

With MLKit's on-device models and processing pipeline, which can be accessed via a free API, the application works offline without transmitting sensitive information.

Data processing and **resource utilization** are more efficient, and multiple languages can be supported, resulting in huge cost savings.

- Works offline
- Privacy is retained
- Efficient & Accurate
- Supports multiple languages

CHANGE

Limitation

This application is limited by the fact that the Vision API currently supports only Latin languages offline, and requires Google Cloud to process languages with characters such as Chinese, Arabic, Korean, Hindi, etc.

In addition, the entity extractor supports limited entities. ML Kit models run on device or cloud, but when it comes to large models, there's a slight possibility the model will slow down the phone. Moreover, in the application the text is only detected when the camera and text the is oriented the same way

CHANGE

CONCLUSION

In computer vision and natural language processing, data-extraction pipelines have been heavily influenced by deep learning. Compared to conventional machine learning applications,

Android Studio provides developers with easy **integration** and **implementation** of various machine learning libraries and tools that are needed to develop an application.

Nevertheless, the beta version has its limitations and it still needs extensive updates to cater for more features.

Overall, the results of the test run were promising with highly accurate text recognition and entity extraction.

CHANGE