E prescription

Speech To Text Approach:

First: Speech To Text using Google API:

using “SpeechRecognizer “Class, this class provides access to the speech recognition service. This service allows access to the speech recognizer. The implementation of this API is likely to stream audio to remote servers to perform speech recognition. As such this API is not intended to be used for continuous recognition, which would consume a significant amount of battery and bandwidth, and it’s totally free. There are many constants in “SpeechRecognizer”, but unfortunately, there isn't a single built-in constant or method to directly detect the spoken language within the Android “SpeechRecognizer” API. While some constants might seem promising, here's why they wouldn't achieve language detection:

- EXTRA_LANGUAGE: This constant allows you to specify a preferred language model for recognition. However, it doesn't provide feedback on the actual language spoken.
- Locale: While you can set the locale for the recognition, it only influences the recognition model used, not language detection after speech recognition is complete.

Second: Use Cloud Speech-To-Text for Speech Recognition On GCP:

It’s a service on GCP that enables developers to convert audio input to text using Google’s speech recognition technology inside without start any activity for result.
Speech-to-Text pricing:

Category	Models	Pricing			
		0-500,000 minutes / month	500,000-1,000,000 minutes / month	1,000,000-2,000,000 minutes / month	2,000,000+ minutes / month
Speech recognition (default)	Standard[1]	$0.016 / minute **	$0.010 / minute **	$0.008 / minute **	$0.004 / minute **
	Medical[2]	$0.078 / minute **	$0.078 / minute **	$0.078 / minute **	$0.078 / minute **
Dynamic batch speech recognition	Standard[1]	$0.003 / minute **	$0.003 / minute **	$0.003 / minute **	$0.003 / minute **

Standard[1] models include: `default`, `command_and_search`, `latest_short`, `latest_long`, `phone_call`, `video`, `chirp` (Speech-to-Text V2 only).
Medical[2] models include: `medical_conversation`, `medical_dictation`.

Drawing Approach:

Digital ink recognition:

In first version with Flutter: I was taking screen shoot to the drawing and using OCR to read the image and convert it to text, but it was not so accurate because it's hand writing so I added trained English Data Model to make output more accurate it's helped to improve the output but it was still not accurate hundred percent. So Recognizing digital ink with ML Kit on Android.

With ML Kit's digital ink recognition API you can recognize handwritten text and classify gestures on a digital surface in hundreds of languages, as well as classify sketches. The digital ink recognition API uses the same technology that powers handwriting recognition in Gboard, Google Translate

Digital ink recognition Capabilities:

- Converts handwritten text to sequences of unicode characters
- Runs on the device in near real time
- The user's handwriting stays on the device, recognition is performed without any network connection
- Supports 300+ languages and 25+ writing systems.
- Recognizes emojis and basic shapes
- Keeps on-device storage low by dynamically downloading language packs as needed

The recognizer takes an Ink object as input. Ink is a vector representation of what the user has written on the screen: a sequence of *strokes*, each being a list of coordinates with time information called *touch points*. A stroke starts when the user puts their stylus or finger down and ends when they lift it up. The Ink is passed to a recognizer, which returns one or more possible recognition results, with levels of confidence.

While the digital ink recognition API supports hundreds of languages, each language requires some data to be downloaded prior to any recognition. Around 20MB of storage is required per language, every time I check whether a model has been downloaded already

OCR Approach:

First: Text recognition v2 ML Kit:

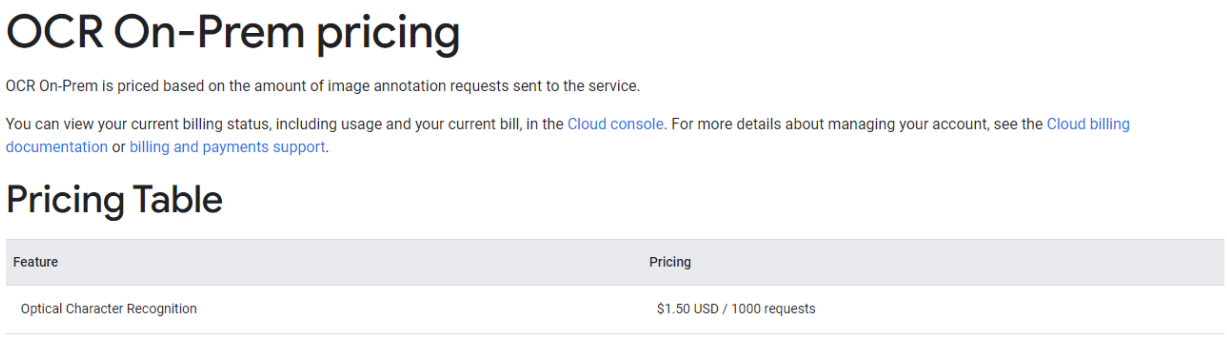
Key capabilities

- Recognize text across various scripts and languages Supports recognizing text in Chinese, Devanagari, Japanese, Korean and Latin scripts
- Analyzes structure of text Supports detection of symbols, elements, lines and paragraphs
- Identify language of text Identifies the language of the recognized text
- Real-time recognition Can recognize text in real-time on a wide range of devices

But there is issue of Text Recognition v2 struggling with lines containing excessive spaces in Kotlin that makes recognize receipt is impossible.
Text recognition v2 ML Kit it's totally free but doesn't support Arabic language and has poor performance when used with handwriting

Second: Google Cloud-based OCR with Arabic Support and hand writing:

Access advanced vision models via APIs to automate vision tasks, streamline analysis, and unlock actionable insights. Or build custom apps with no-code model training and low cost in a managed environment.



OCR On-Prem pricing

OCR On-Prem is priced based on the amount of image annotation requests sent to the service.

You can view your current billing status, including usage and your current bill, in the Cloud console. For more details about managing your account, see the Cloud billing documentation or billing and payments support.

Pricing Table

Feature	Pricing
Optical Character Recognition	$1.50 USD / 1000 requests

Third: Server-side Tesseract OCR (Advanced):
Install Tesseract OCR on your server with Arabic language support. Develop server-side logic to process images and extract text using Tesseract OCR. This approach involves server infrastructure management and complex integration with Kotlin application.
https://github.com/tesseract-ocr/tessdata/blob/main/eng.traineddata (English model)
https://github.com/tesseract-ocr/tessdata/blob/main/ara.traineddata (Arabic model)
note: I implemented receipt OCR(Asprise), but it works well only with shopping receipts. The high cost is also a concern.

Reduce the size of your ML Kit Android app's APKs:

Before you deploy to production an app that uses an ML Kit on-device model, consider following the advice on this page to reduce the download size of your app.

Move optional ML features to dynamic feature modules:
If you use ML Kit in a feature of your app that isn't its primary purpose, consider refactoring your app to move that feature and its ML Kit dependencies to a dynamic feature module.
Overview of Play Feature Delivery
Google Play's app serving model uses Android App Bundles to generate and serve optimized APKs for each user's device configuration, so users download only the code and resources they need to run your app.
Play Feature Delivery uses advanced capabilities of app bundles, allowing certain features of your app to be delivered conditionally or downloaded on demand. To do that, first you need to separate these features from your base app into feature modules.