COUPONER-X

—

Jaime Canizales, Marcos Fabian,
Henry Huang, & Jessenia Lopez

Advance Application: Capstone Major

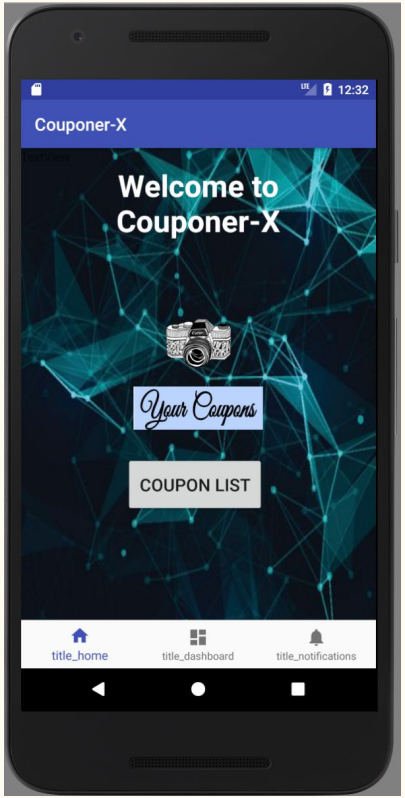
CSCI 49900 - 02

Our Goal

Create an app that allows users to store and keep track of different coupons without having to physically carry them.

- Efficient
  - Coupons can be viewed at any time
- User-friendly
  - Integrated OCR Implementation
- Great for Coupon Enthusiasts
  - Coupons will stay with you wherever you go

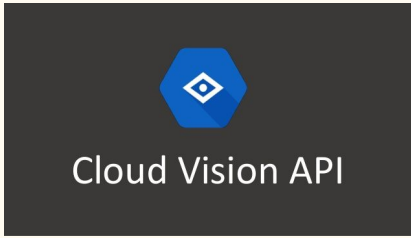
Couponer-X: The app



- Coupon Storer
  - Take pics and saves them
    - Images can be viewed on the app itself
- Coupon Organizer
  - Store Name
- Coupon Reminder
  - The app sends a notification a day prior to the expiration date
    - Note: we did not have enough time to implement this just yet

Technologies Used in this Application

- Languages Used: Android, Java, HTML, and Python
- Android Studio
  - Our code was written and ran on the emulator or a physical Android Device
  - ListView
  - ImageView
  - Bytearray
- SQLite Database
  - For the database
- Google Vision API
  - OCR

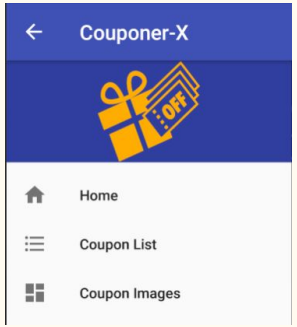


Work Distribution

- Jaime Canizales
  - OCR Implementation, and Barcode Detection
- Marcos Fabian
  - Camera Feature, Website,OCR Implementation and Debugger
- Henry Huang
  - Database Developer
- Jessenia Lopez
  - Layout, Debugger, and Merged the Code's

Architecture of the App

- Button's:
  - Camera
    - Images are stored in the gallery's directory
  - Your Coupons
    - Shows the images of the coupons
  - Manual Entry
    - User's can manually insert the coupon information
- Navigation Bar
  - Home
  - Coupon List
    - Click on a specific store name and you will see the:
      - Expiration Date
      - Barcode Number
  - Coupon Images



An Integrated Development Environment (IDE) where applications can be created for android platforms.

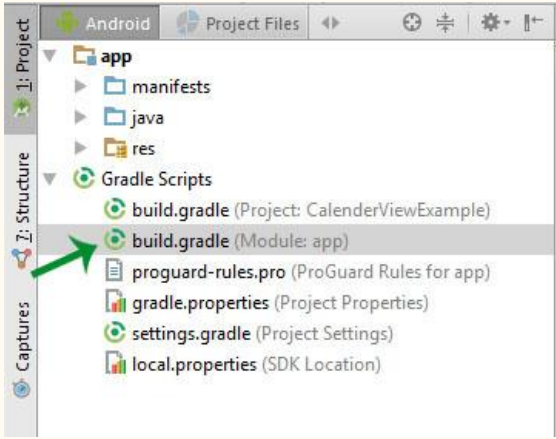
Features:

- APK File
- Gradle-Based Building System
- API Level
- Built-in Emulator
- Unified Environment
- Basic Android Studio Functionalities
- Has built in support for Google Cloud Platform

APK & Gradle Build System

You can do the following:

- APK: The application represented in the android device or emulator
- The Gradle Build System compress all the components used in the projects into and APK. (resources, source code, APIs)
- Customization and Configuration of the Building Process
- Multiple APK's can be created using the same code
- Reuse the code and resources



(AVD) Android Virtual Device & API Level

- Android comes with built-in emulator
- API Level: is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.
- Compatibility Problem: When the project to build has a different API Level than that of the android device or AVD
  - Emulator or device may not be updated
  - Emulator or device may be old
  - Emulator or device may not support some libraries
  - Emulator or device is not capable of completing task due to the harward

Basic Android Studio Functionalities

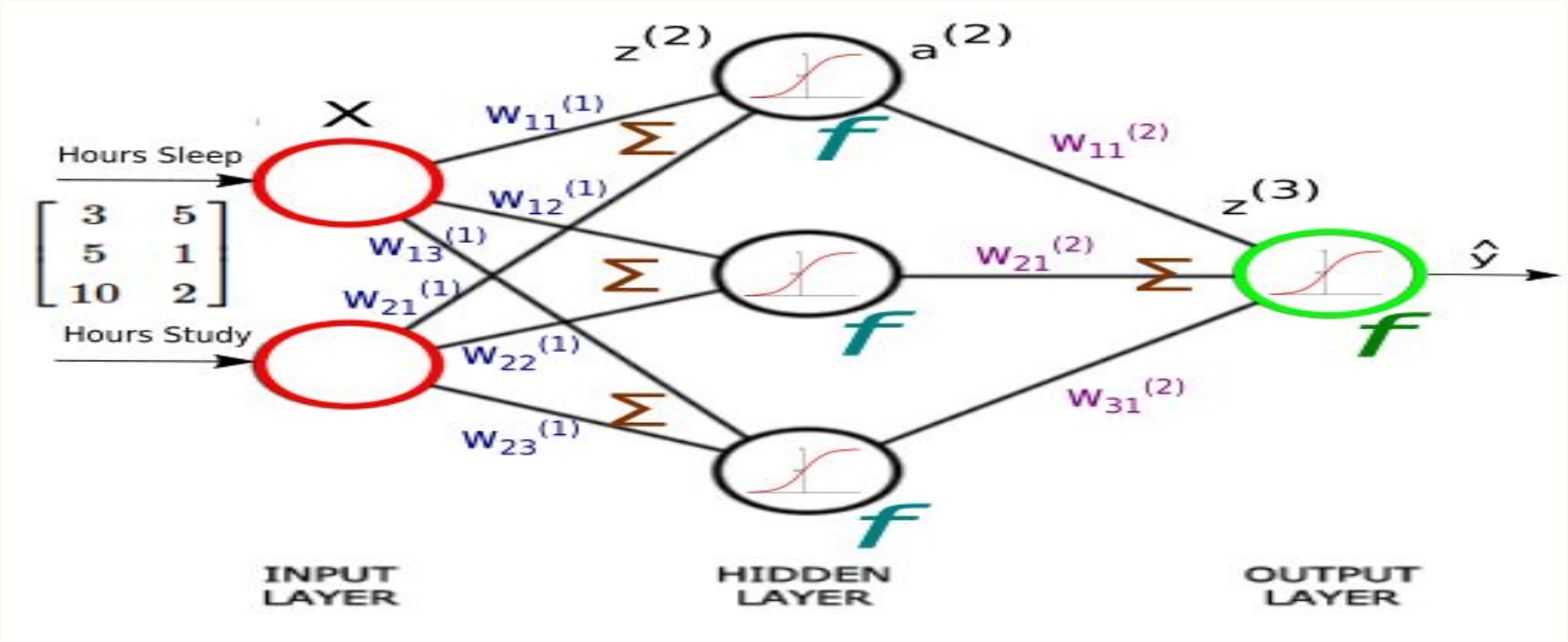
- Each screen is an activity which have .xml file (layout)
  - Layout: defines the visual structure of an interface
    - Allows the user to see the layout of each activity(screens)
- Each part of an activity have an implementation
  - a bottom have a function to decide what it should do when pressed
- Each activity is attached to a .java class
  - .java code can be implemented with an interface presented to users
  - Supports C/C++
- Intents allows activities to switch one another (.java class)
- SQLite Allows information to be saved in a database

An example of an Android Studio Library: SQLite

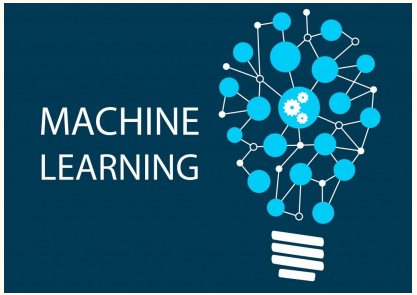
- Android Studios has built in SQLiteDatabase class
  - Allows users to create a database from the app
  - Allows the user to directly store data into the app
  - Relational database management system (DBMS).
  - SQLite is embedded in a program. Allows user to store information in phone.
- Allows users to save images into SQLite
  - Saves byte array (Blob) into database

Neural Network

Sigmoid function is 1/(1+e^-x)



Machine Learning Models



- A ML model is an algorithm such as a neural network, that has defined parameters, has been trained on a dataset and is capable of making predictions on a new instance

Google Cloud 

- A suite of cloud computing service
- Offers parallel computing services on tons of cpu, gpus and tpus
- Is home to many machine learning models ready to be used!
- And much more !

Google Play Services 

- A set of APIs for android

Why to use google play services ?

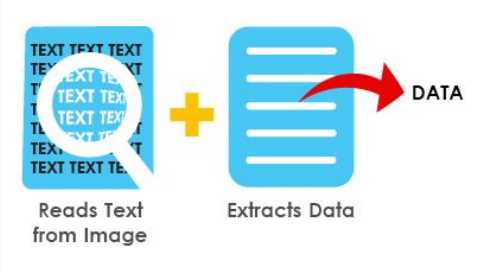
- Easy to implement
- Reliable and is often modified to improve performance and fix bugs
- Computationally efficient

Google Play Services Mobile Vision API 

- A computer vision API that uses trained ML models to provide useful information about an image.
- Most of those trained ML models use Neural Networks in some way.
- Many of them are stored in Google Cloud servers.
- Supports OCR, face detection, and scans barcodes.

How it works

1. Given an image or camera as input on your device.
2. Function call to API sends image to the location where the model(probably NN) is stored(NN is likely stored in Google Cloud).
3. ML model will predict and return an output based on your request and sends the output to your device.



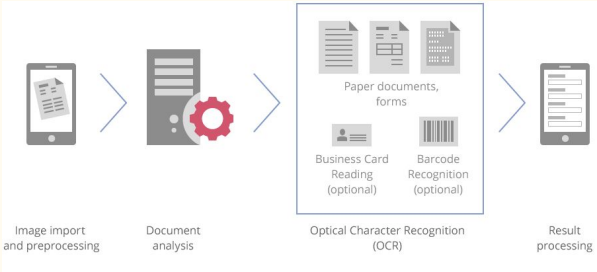
Mobile Vision API continued

When to use it:

- Whenever image processing that requires OCR, face detection, or scanning barcodes.

When not to use:

- Other types of computer vision applications not mentioned above.
- If you need to train your own ML model.



In the near future...

- Train a neural network to identify barcodes from an image, crop it, and save it to the database.
- Apply an OCR algorithm to correctly identify the store name, expiration date, and description of the coupon directly from the image.
- Organize the stored coupons in a simpler and user friendly fashion.
- Include Notifications that can be viewed on the Navigation Bar
    - so user's can get alerts for when a specific coupon is about to expire

App Demonstration: