Qualcomm Developer Project CM2290 Image Classification

Project Submission

|  |  |  |
| --- | --- | --- |
| **Attribution:** |  | |
| **Email address** | [zhangzz6687@thundersoft.com](mailto:zhangzz6687@thundersoft.com)  [hongliang.liu@thundersoft.com](mailto:hongliang.liu@thundersoft.com)  zhanglei0706@thundersoft.com | |
| **Project Title**\* | **CM2290 Image Classification** | |
| **Images**  *Upload up to 5 images of your project*  *Please submit/send the original JPEG/PNG files for all images included in the document* | CM2290.png  CM2290 [alt tag: “CM2290 Image Classification using the CM2290 which is designed based on the Qualcomm QCM2290/QCS2290 platform.”] **Type-C.png**   |  | | --- | | **typc** |   [alt tag: “**using the type-c line to develop on C865DK development board.** ”]  [alt tag: “use mini-hdmi line to connect display **.**”]  IP-Camera.png  ip-camera | |
| **Description**\*  *High level description of the project* ***(75 words or less)*** | Use the CM2290 to Use tflite model to identify especial things in Image.  In most cases, those objects should be less than three. | |
| **Objective**   * *What inspired you to create this project?* * *What is your desired outcome?* | 1. Recognize normal things such as keyboard and mouse. 2. For future Image analysis data statisticsd | |
| **Materials Required / Parts List / Tools** | Part Name | Link to purchase |
| CM2290 | https://www.thundercomm.com/zh/product/cm2290-c2290-som/ |
| Type-c line | https://detail.tmall.com/item.htm?id=44425281296&ali\_refid=a3\_430582\_1006:1103572855:N:8BFxSxK119dzkfQCc2yGI2us815vvcUHETWnj5g1swo=:6399b40850a40201c56536531a885bcf&ali\_trackid=1\_6399b40850a40201c56536531a885bcf&spm=a230r.1.14.11 |
| IP Camera | https://item.jd.com/ |
| Models | <https://github.com/tulasiram58827/ocr_tflite> |
|  |  |
| Description | Link |
| [Source Code](https://github.com/canyudeguang/Home_Automation) | <https://github.com/ThunderSoft-XA/CM2290-Image-Classificaion> |
| **Source Code / Source Examples / Application Executable**  *Link to open source / shareable code repository* |  |  |
|  |  |
|  |  |
| Resource Title | Link or File Name (and provide file) |
|  |  |
| **Additional Resources**  *List related links or resources such as websites, videos, presentations, or other materials* |  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Build / Assembly Instructions** | Sample outline:   1. Overall design framework and Test environment construction method.      1. Software Build Instructions   Prepare a PC (Ubuntu 16.04/ window10/ MAC);   1. Install app to CM2290 device   adb device; adb install CM2290 Image Classification.apk | |
|  | Sample outline:   1. How does it work?   During training, a Image classification model is provided Images and their associated labels. Each label is the name of a distinct object, that the model will learn to recognize. For Image action recognition, the Images will be of full of a special entity with a prue color background and the labels will be it’s discription.  The Image classification model can learn to predict whether new Images belong to any of the classes provided during training. This process is called inference. You can also use [transfer learning](https://colab.research.google.com/github/tensorflow/models/blob/master/official/projects/movinet/movinet_tutorial.ipynb) to identify new classes of Images by using a pre-existing model.  The model is a streaming model that receives continuous image and responds in real time. As the model receives a Image stream, it identifies whether any of the classes from the training dataset are represented in the Image. For each frame, the model returns these classes, along with the probability that the Image represents the class.  Main structure of java： ├── eBox //Main function directory│   ├── Activity //group-box control│   ├── Adapter│   ├── AI // AI task to do picture text recognition│   ├── Config //Configuration module│   ├── Constants│   ├── Data //some AI data structure│   ├── Database//Database processing│   ├── Gl //display module│   ├── Log│   ├── Model│   ├── Utils //some common functions│   └── VIew //display interface├── gateway //some info structure│   ├── data│   └── utils├── libyuv //the color conversion├── rtsp //rtsp client module└── util //common functions Function support by cpp：  ├── Affinity //CPU binding functions  ├── BasicUsageEnvironment  │   └── include  ├── groupsock //live555 feature  │   └── include  ├── libbitmap //same bitmap functions  ├── libyuv //mage color space conversion  │   └── libyuv  │   ├── build\_overrides  │   ├── docs  │   ├── include  │   │   └── libyuv  │   ├── infra  │   │   └── config  │   ├── source  │   ├── tools\_libyuv  │   │   ├── autoroller  │   │   │   └── unittests  │   │   │   └── testdata  │   │   ├── msan  │   │   ├── ubsan  │   │   └── valgrind  │   │   └── memcheck  │   ├── unit\_test  │   │   └── testdata  │   └── util  ├── liveMedia //live555 feature  │   └── include  ├── RtspClient //live555 feature  │   └── include  └── UsageEnvironment //live555 feature  └── include | |
| **Usage Instructions** | Sample outline:   1. Install app to CM 2290 device   adb install CM2290 Image Classification.apk   1. Start app.   1)connect wifi/wired network, start “Image Classification” app  ic_launcher.png  2)use the camera on CM 2290 to get Image   1. App can show name of the objects in image and show the data rate | |
| **Contributor(s) Info**  *Feel free to include headshots!* | Name | Title  Company |
|  |  |
|  |  |
|  |  |

––– Continued on next page –––

Filters and Tags for QDN projects page

|  |  |  |
| --- | --- | --- |
| **Platform/Hardware** | ☐ CSR 101x/102x Bluetooth  ☐ DragonBoard 410c  ☐ mangOH Red/Yellow | ☐ MDM920x LTE for IoT  ☐ QCA-402x WiFi/BLE/Zigbee  √ Qualcomm CM 2290 Som |
| **Software Tools** | ☐ 3D Audio Plugin for Unity  ☐ Adreno GPU SDK  ☐ Hexagon DSP SDK | ☐ Neural Processing SDK for AI  ☐ Snapdragon Profiler |
| **Operating System** | √ Android  ☐ Linux  ☐ ThreadX RTOS | ☐ Ubuntu Core  ☐ Windows 10 IoT Core |
| **Cloud Services/Platform** | ☐ Sierra Wireless AirVantage  ☐ Gizwits Cloud Platform  ☐ AT&T M2X  ☐ IBM Bluemix | ☐ IBM Watson IoT  ☐ Microsoft Azure IoT  ☐ Amazon AWS IoT |
| **Skill Level Required** | √ Advanced  ☐ Beginner  ☐ Intermediate |  |
| **Areas of Focus** | ☐ 3D Printing & Modeling  ☐ Alexa Voice Service  √ Artificial Intelligence  ☐ Bluetooth  √ Computer Vision  ☐ Digital Signage  ☐ Education  ☐ Embedded  ☐ Gaming | ☐ Healthcare  √ IoT  ☐ Robotics  ☐ Security  ☐ Sensors  ☐ Smart Cities  ☐ Smart Home  ☐ Toys  √ Smart Retail |

*By submitting your content (“Submission”), you are granting Qualcomm a royalty-free, perpetual, non-exclusive, unrestricted, worldwide license to: (a) post, use, copy, sublicense, adapt, transmit, publicly perform or display any such Submission, (b) use, reproduce, modify, adapt, publish, translate, create derivative works from, distribute, perform, play, host, communicate, make available and publish your Submission without restriction and (c) sublicense to third parties the unrestricted right to exercise any of the foregoing rights granted with respect to the Submission. The foregoing grants shall include the right to exploit any ideas, concepts, intellectual property, or proprietary rights in such Submission, including but not limited to rights under copyright, trademark, servicemark or patent laws under any relevant jurisdiction without Qualcomm owing any monies to you whatsoever. You represent and warrant that you own all right, title and interest in and to the Submission, or you have been granted sufficient rights in and to the Submission allowing the foregoing use of such Submission.*