|  |  |
| --- | --- |
| **summary:** | HMS FaceOff Dress Trial |
| **description:** | HMS Codelab Integration Document |
| **HMS kits:** | Image Kit – Image Cropping Service  ML Kit - Image Segmentation |
| **status:** | Published |
| **feedback link:** | [https://developer.huawei.com/consumer/en/support/feedback](https://developer.huawei.com/consumer/cn/support/feedback) |
| **authors:** | HUAWEI |

# HMS FaceOff Dress Trial

## **Introduction**

Duration: 5:00

### Service Scenario Description

HMS FaceOff Dress Trial app is the application for dress trial with different type of dress trial room. User can go to the different type of dress rooms (room1, room2, room3, room4) to trial dresses. Each room have multiple dresses of different type of clothes to trial by users

In this code lab application we have implemented below features with HMS kits.

|  |  |
| --- | --- |
| **Features** | **HMS Kits** |
| Image crop | Image kit – Image Cropping Service |
| Image Flip | Image kit – Image Cropping Service |
| Image Rotation | Image kit – Image Cropping Service |
| Image Segmentation | ML Kit - Image Segmentation |

### What You Will Create

In this code lab, you will create demo project and use the APIs of Image kit and ML kit (Image Segmentation) you can explore the following processes in the demo project:

* Select different Type of Dress
* Capture image and process for cropping , flip, rotation image
* Image Segmentation to get segmented foreground image
* Super impose dress image and face image

### What You Will Learn

In this code lab, you will learn how to:

* Integrate Image Kit (Image Cropping Service)
* Integrate ML Kit (Image Segmentation)

## **What You Will Need**

Duration: 1:00

### Hardware Requirements

* A computer (desktop or laptop) that runs the Windows 10 operating system
* A Huawei phone with HMS Core (APK) 5.0.0.300 or later installed

|  |
| --- |
| **Note**: Please prepare the preceding hardware environment and relevant devices in advance. |

### Software Requirements

* [Android Studio 3.X](https://developer.android.com/studio)
* JDK 1.8 and later
* SDK Platform 28 and later
* Gradle 4.6 and later

|  |
| --- |
| **Note**: Please prepare the preceding software environment in advance. |

## **Preparing for the Integration**

Duration: 20:00

Use the below link for preparing for the integration

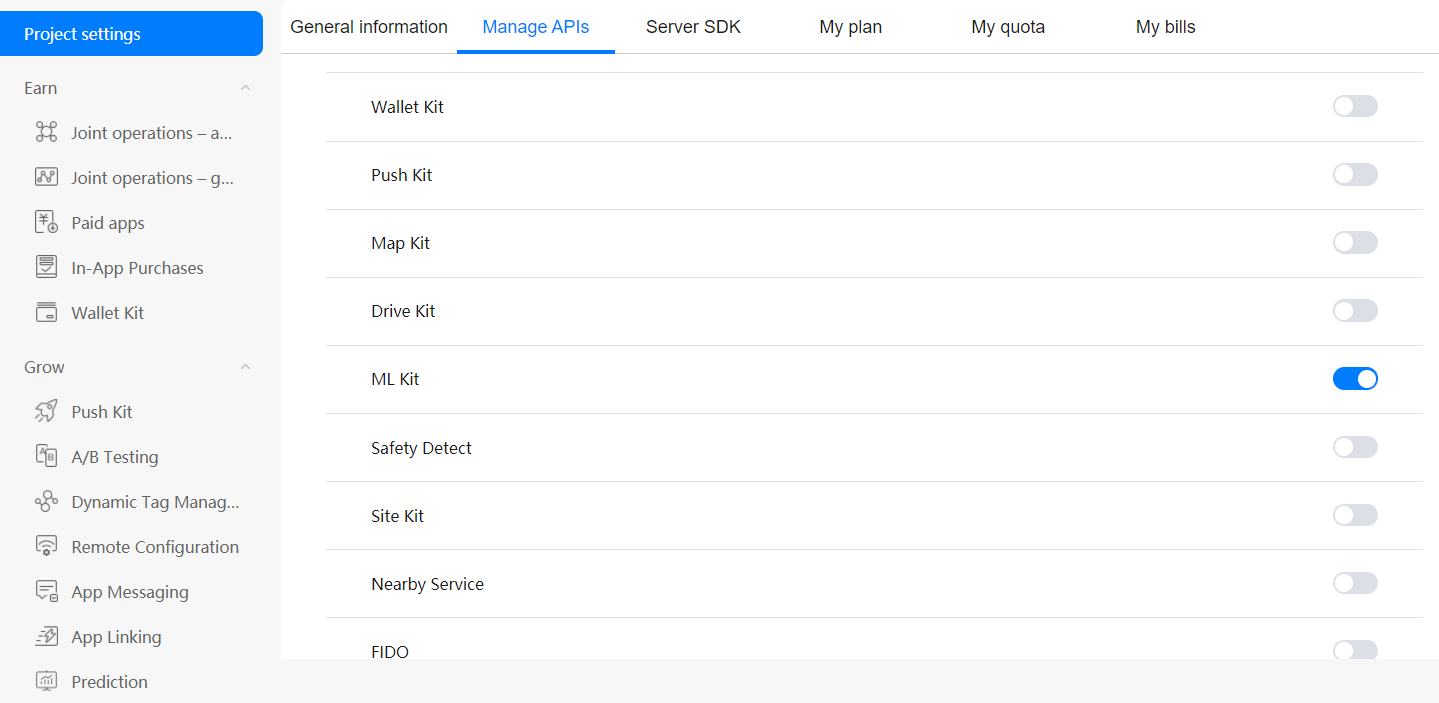
<https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/config-agc-0000001050196065>

## **Enable HUAWEI Service(s) in AGC console**

Duration: 5:00

1. Enable the API permission for below kits from **Project Setting** > **Manage APIs** and enable the API permission.

* ML Kit



|  |
| --- |
| **Note**: This permission will be enabled by default. If not enable it manually. |

Now, you have successfully enabled HUAWEI Services required for your app.

## GUI Development

Duration: 10:00

### Home Screen

User can go to the different type of dresses room (room1, room2, room3, room4) to trial dresses. Each room have multiple dresses of different type of clothes to trial by users

### Dress Trial Room

Once click on Room images (Room1, Room2, Room3, Room4)

1. It will show the dress trial room where the user can select dresses from bottom bar dress section and selected dress can become big size
2. If user want to try with another room, User can click on back button to navigate home screen



## Integration of HMS Kits

Duration: 35:00

### Integration of Image Kit (Image Cropping Service)

Use below link to Integrate HMS core SDK for following kits

[Image Kit - Image Cropping Service](https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides-V5/vision-service-dev_crop-0000001054686036-V5)

This service provides the cropping function for users to resize images. Image cropping service APIS are implemented based on **CropLayoutView**, which is a custom view.

Step 1- Add **CropLayoutView** using package **com.huawei.hms.image.vision.crop.CropLayoutView**, into the XML of the layout.

|  |
| --- |
| <com.huawei.hms.image.vision.crop.CropLayoutView  android:id="@+id/cropImageView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  </com.huawei.hms.image.vision.crop.CropLayoutView> |



Step 2 – Create the object of **croplayoutview**. And set the bitmap image to be process the image (crop, flip, rotation)

**Java**

|  |
| --- |
| cropLayoutView = findViewById(R.id.*cropImageView*); cropLayoutView.setImageBitmap(new Util().rotateImage(bmp, ROTATION\_ANGLE\_IN\_DEGREES)); cropLayoutView.setAspectRatio(ASPECT\_RATIO\_X, ASPECT\_RATIO\_Y); cropLayoutView.setFixedAspectRatio(Constants.*STATUS\_FAILURE*); |

**Kotlin**

|  |
| --- |
| cropLayoutView = findViewById(R.id.*cropImageView*); cropLayoutView?.setImageBitmap(Util().rotateImage(bmp, ROTATION\_ANGLE\_IN\_DEGREES.toFloat()));  cropLayoutView?.setAspectRatio(ASPECT\_RATIO\_X, ASPECT\_RATIO\_Y); cropLayoutView?.setFixedAspectRatio(Constants. *STATUS\_FAILURE*); |

Step 3 – make rotation, flip and save the cropped image

* API for rotation the image by 90 degrees.

**Java**

|  |
| --- |
| cropLayoutView.rotateClockwise(); |

**Kotlin**

|  |
| --- |
| cropLayoutView?.rotateClockwise(); |

* API for flip the image for horizontal mirroring.

**Java**

|  |
| --- |
| cropLayoutView.flipImageHorizontally(); |

**Kotlin**

|  |
| --- |
| cropLayoutView?.flipImageHorizontally(); |

* Save Cropped Image

**Java**

|  |
| --- |
| Bitmap croppedImage = cropLayoutView.getCroppedImage(); |

**Kotlin**

|  |
| --- |
| val croppedImage = cropLayoutView?.getCroppedImage(); |



|  |
| --- |
| **Note**: For more details about Image Cropping Service please refer the below link  <https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/vision-service-dev_crop-0000001054686036> |

### Integration of ML kit (Image Segmentation)

Use below link to Integrate HMS core SDK for following kits

[ML Kit – Image Segmentation](https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/image-segmentation-0000001050040109)

The image segmentation service segments same elements (such as human body, plant, and sky) from an image. The return values of human body segmentation include the coordinate array of the human body, human body image with a transparent background, and gray-scale image with a white human body and black background.

There are two types of image segmentation service model packages: human body segmentation model package and multiclass segmentation model package.

In this project you can implement **human body segmentation model package** which is used to segment the human body and background

Step1 - Create an image segmentation analyzer.

**Java**

|  |
| --- |
| public void createAnalyzer() {   MLImageSegmentationSetting setting =  new MLImageSegmentationSetting.Factory()  .setExact(false)  *// Set the human body segmentation mode.* .setAnalyzerType(MLImageSegmentationSetting.*BODY\_SEG*)  .setScene(MLImageSegmentationScene.*FOREGROUND\_ONLY*)  .create();  analyzer = MLAnalyzerFactory.*getInstance*().getImageSegmentationAnalyzer(setting); } |

**Kotlin**

|  |
| --- |
| fun createAnalyzer() {  val setting = MLImageSegmentationSetting.Factory()  .setExact(IsStatus)  .setAnalyzerType(MLImageSegmentationSetting.BODY\_SEG)  .setScene(MLImageSegmentationScene.FOREGROUND\_ONLY)  .create()  analyzer = MLAnalyzerFactory.getInstance().getImageSegmentationAnalyzer(setting) } |

Step2 – Process Image Segmentation –

* Create an **MLFrame** objectby using **fromBitmap()** methods for the analyzer to detect image
* Call the [**asyncAnalyseFrame**](https://developer.huawei.com/consumer/en/doc/development/HMSCore-References-V5/mlimagesegmentationanalyzer-0000001050169515-V5#EN-US_TOPIC_0000001050169515__section17489546143417)method to perform image segmentation. It is recommended that the image size be greater than or equal to 224 x 224 px.

**Java**

|  |
| --- |
| public void processSegmantationResult(Bitmap bitmap) {  MLFrame frame = MLFrame.*fromBitmap*(bitmap);  Task<MLImageSegmentation> task = analyzer.asyncAnalyseFrame(frame);  task.addOnSuccessListener(  new OnSuccessListener<MLImageSegmentation>() {  @Override  public void onSuccess(MLImageSegmentation segmentation) {  if (segmentation != null) {  displaySuccess(segmentation);  } else {  Log.*e*(*TAG*, "onSuccess: MLImageSegmentation result is null ");  }  }  })  .addOnFailureListener(  new OnFailureListener() {  @Override  public void onFailure(Exception e) {  Log.*e*(*TAG*, "onFailure: " + e);  }  }); } |

**Kotlin**

|  |
| --- |
| fun processSegmantationResult(bitmap: Bitmap?) {  val frame = MLFrame.fromBitmap(bitmap)  val task = analyzer!!.asyncAnalyseFrame(frame)  task.addOnSuccessListener **{** segmentation **->** segmentation?.let **{** displaySuccess(it) **}** ?: Log.d(TAG, Constants.NULL\_MSG)  **}** .addOnFailureListener **{** e **->** Log.d(TAG, Constants.ON\_FAILURE\_STR, e) **}** } |

Step3 – Get Bitmap image after returning segmentation object at **onSuccess** method. Use **getForeground()** to get foreground bitmap object

**Java**

|  |
| --- |
| private void displaySuccess(MLImageSegmentation imageSegmentationResult) {  Bitmap bitmapFore = imageSegmentationResult.getForeground();  if (bitmapFore != null) {  isListner.getSegmentationImage(bitmapFore);  stopAnalyser();  } else {  Log.*e*(*TAG*, "displaySuccess:bitmapFore is null. ");  } } |

**Kotlin**

|  |
| --- |
| private fun displaySuccess(imageSegmentationResult: MLImageSegmentation) {  val bitmapFore = imageSegmentationResult.getForeground()  if (bitmapFore != null) {  isListner.getSegmentationImage(bitmapFore)  stopAnalyser()  } else {  Log.d(TAG, Constants.BITMAP\_NULL)  } } |

Step4 - After the detection is complete, stop the analyzer to release detection resources.

**Java**

|  |
| --- |
| public void stopAnalyser() {  if (analyzer != null) {  try {  analyzer.stop();  } catch (IOException e) {  Log.*e*(*TAG*, "stopAnalyser: " + e);  }  } } |

**Kotlin**

|  |
| --- |
| fun stopAnalyser() {  if (analyzer != null) {  try {  analyzer!!.stop()  } catch (e: IOException) {  Log.d(TAG, Constants.EXCEPTION\_MSG, e)  }  } } |

|  |
| --- |
| **Note**: For more details about Image Segmentation Service please refer the below link  <https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/image-segmentation-0000001050040109> |



## Congratulations

Duration: 2:00

Well done. You have successfully built a dress trial app and learned how to:

* + Implement image cropping service using HMS Image Kit.
  + Implement Image Segmentation using ML Kit.

## References

Duration: 2:00

1. [Image kit - Image cropping Service](https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/vision-service-dev_crop-0000001054686036)
2. [ML Kit – Image Segmentation](https://developer.huawei.com/consumer/en/doc/development/HMSCore-Guides/image-segmentation-0000001050040109)

The source code can be found at:

Download sample [source](https://github.com/huaweicodelabs/FaceOffDressTrial) code