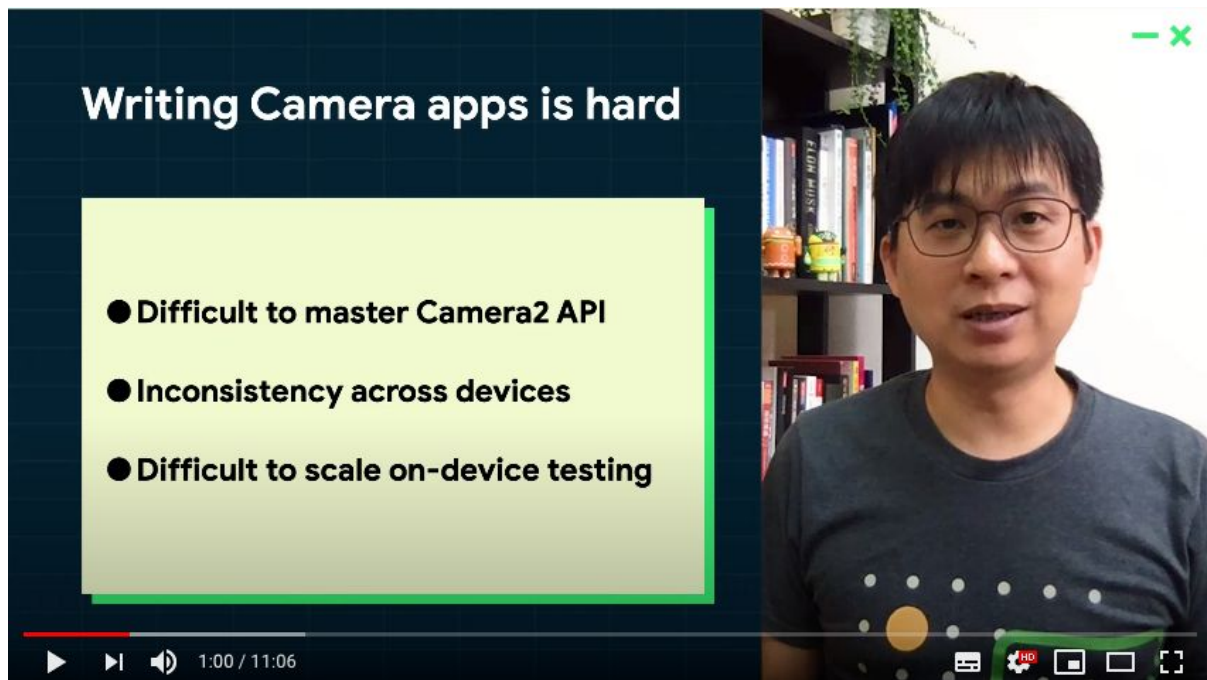


# Getting to know CameraX Beta

## 1. Writing Camera app is hard

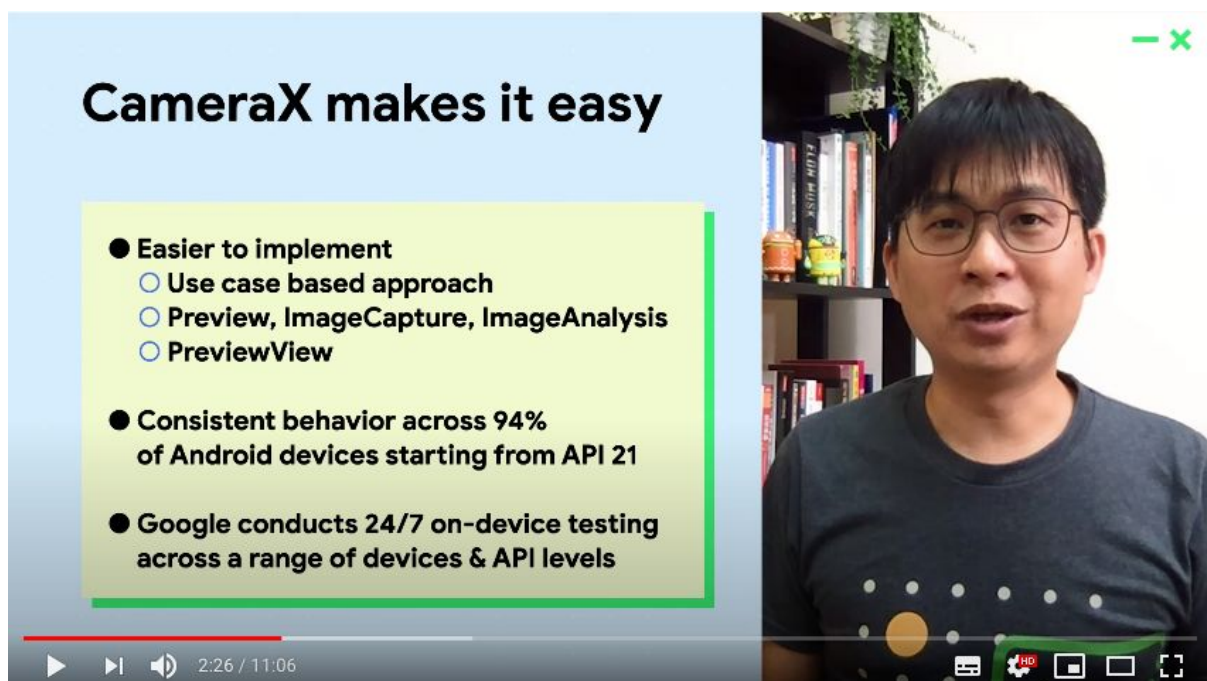


### Writing Camera apps is hard

- **Difficult to master Camera2 API**
- **Inconsistency across devices**
- **Difficult to scale on-device testing**

- Camera 2 API 는 제대로 다루기 어렵다.
- Android의 수많은 장치로 인해 일관된 경험을 제공하기 어렵다. 기기별 분기된 코드가 필요하게된다.
- 플레이스토어에 출시하게 된다면, 많은 실제 기기에서 테스트가 필요하며, 일관된 경험을 제공하기 위해서 기기별로 분기된 코드로 인해 테스트에 비용과 시간이 많이 소요될 수 있다.

## 2. CameraX makes it easy



### CameraX makes it easy

- **Easier to implement**
  - Use case based approach
  - Preview, ImageCapture, ImageAnalysis
  - PreviewView
- **Consistent behavior across 94% of Android devices starting from API 21**
- **Google conducts 24/7 on-device testing across a range of devices & API levels**

- CameraX 를 사용하여 위의 모든 문제를 해결했다.
- 구현이 쉬움

- 이해하기 쉽고 작성해야하는 총 코드 양을 크게 줄여주는 사용하기 쉬운 API를 제공
- 사용사례 기반으로 Camera2의 어려운 부분을 제거
- 현재 3가지 사용케이스를 제공함 - Preview, ImageCapture, ImageAnalysis
- PreviewView : 미리보기 변환의 복잡성을 처리하는 PreviewView
- API 21시작하여 94%의 안드로이드 기기에 일관된 경험을 갖도록 보장
- 테스트에 많은 투자, CameraX 테스트랩은 연중 무휴 가동, 광범위하고 다양한 버전의 Android 기기 보유, CameraX로 호환성 이슈를 처리

### 3. What's new in 1H 2020 (2020년 전반기 새로운 기능)



## What's new in 1H 2020

- Improved device compatibility with Beta!
- Preview:
  - PreviewView new APIs
  - OpenGL sample
- ImageAnalysis:
  - YUV to RGB conversion utility
  - ML Kit sample
  - TensorFlow lite sample

- 장치 호환성은 최우선 순위다. 우리는 장치 문제를 해결하고, 테스트랩 인프라를 개선하는데 많은 노력을 기울였다.

- Preview :
  - PreviewView 에 새로운 API 추가
  - OpenGL 샘플 추가
- ImageAnalysis
  - YUV to RGB 변환 유틸
  - ML Kit 샘플
  - TensorFlow lite 샘플

#### 4. Improved Device Compatibility with Beta

## Improved Device Compatibility with Beta!

- Fixed lots of device specific issues
  - Wrong preview aspect ratio (LG G3)
  - Failed to switch camera (Pixel 2)
  - Tap-To-Focus not working (Samsung Galaxy S7)
- More test coverage:  
rotation, UI widget integration, aspect ratio
- On-device test expanded 3X
  - 88 phones representing 400M active Android devices

- 많은 디바이스의 이슈 해결
- 더 많은 테스트 커버리지
- CameraX test lab 이 3배 성장

#### 5. PreviewView

## PreviewView

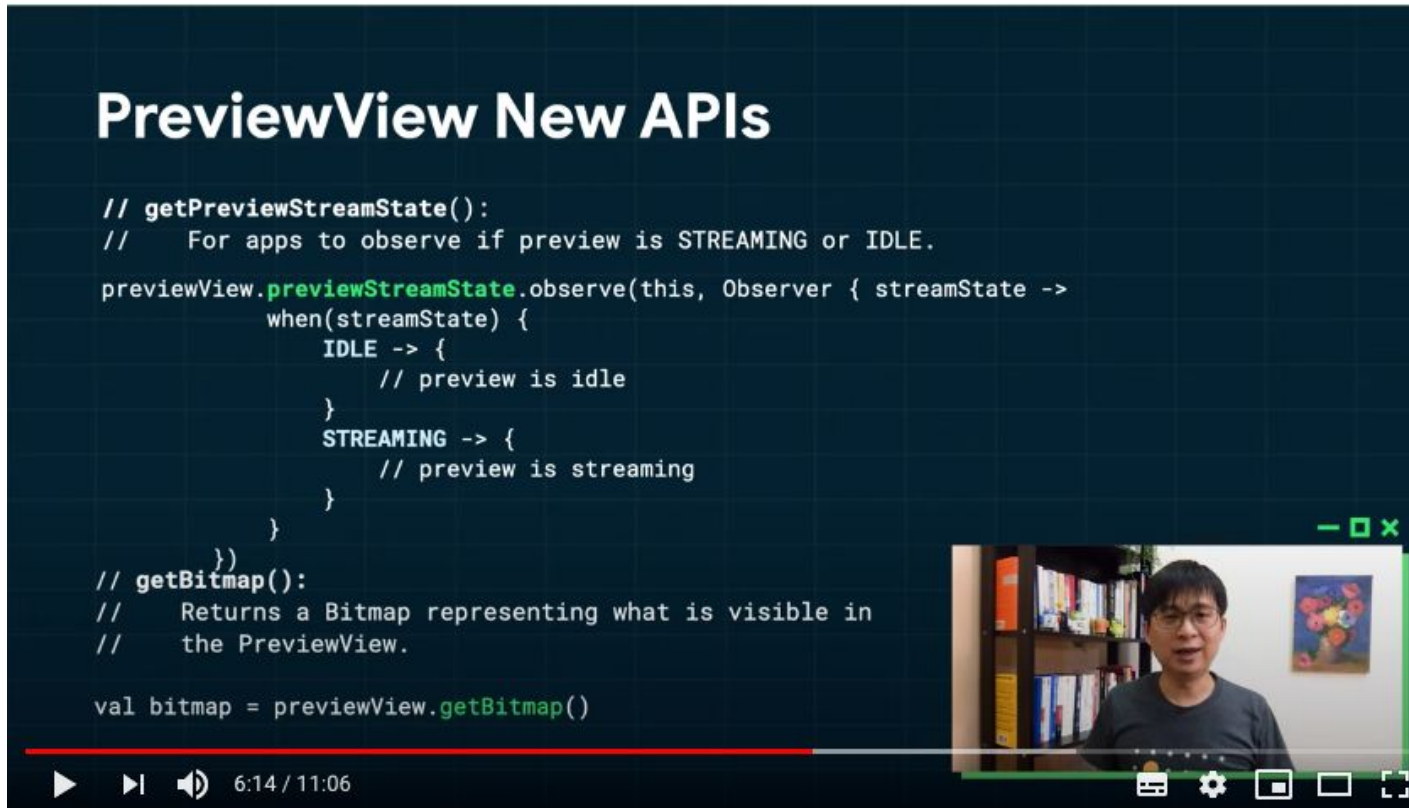
- No transform required
- Power efficient
- Low latency
- Supports `scaleType`
- Easy Tap-To-Focus impl

- transform 이 요구 되지 않음
- 효율적인 전력사용



- 낮은 지연 시간
- 이러한 것들은 내부적으로 SurfaceView를 사용하게되고, 장치가 SurfaceView와 호환이 되지 않는 경우 tetra view(?)로 자동으로 돌아간다.
- scaletype을 지원하므로 PreviewView에서 미리보기의 크기를 조정 가능
- 쉬운 Tap-To-Focus 구현

## 6. PreviewView New APIs



# PreviewView New APIs

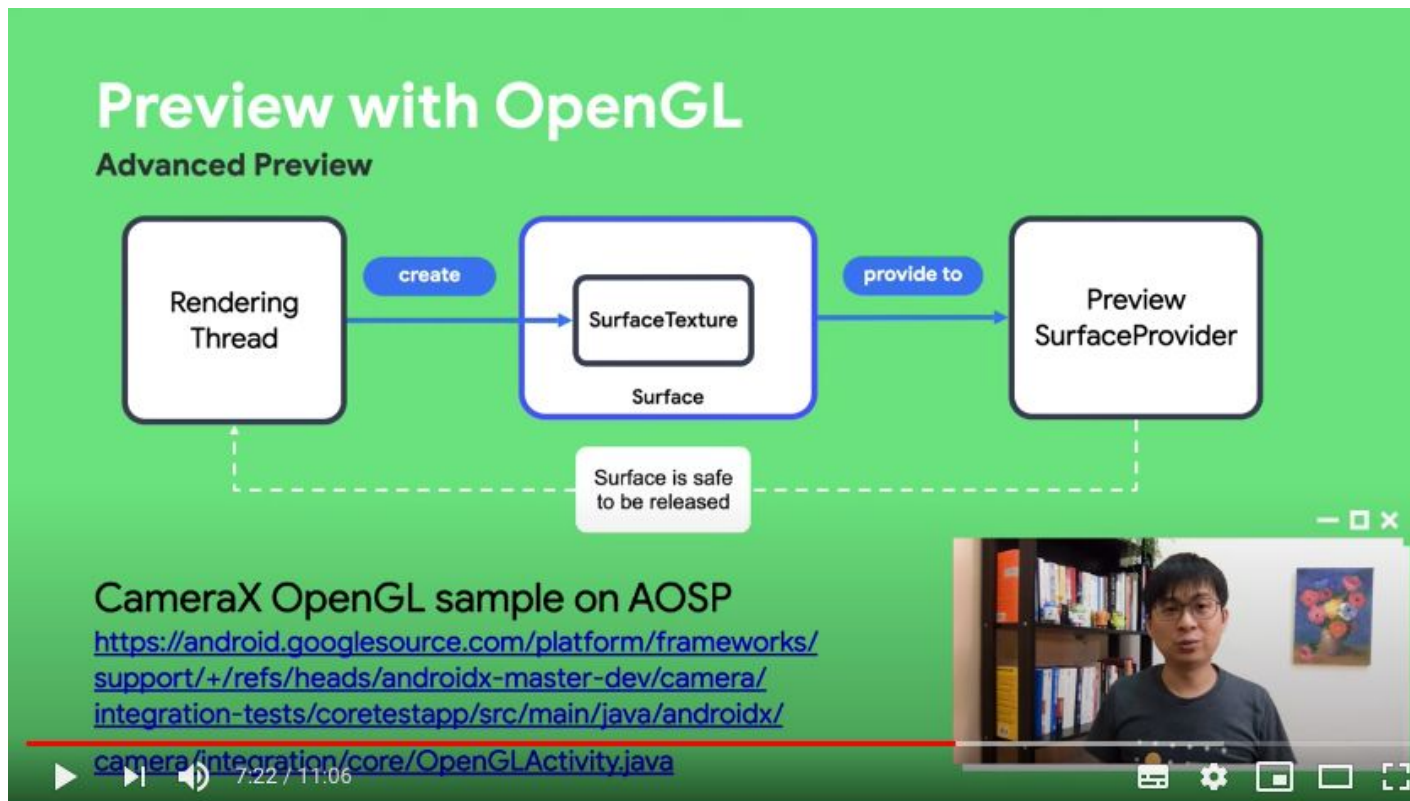
```
// getPreviewStreamState():
//   For apps to observe if preview is STREAMING or IDLE.
previewView.previewStreamState.observe(this, Observer { streamState ->
    when(streamState) {
        IDLE -> {
            // preview is idle
        }
        STREAMING -> {
            // preview is streaming
        }
    }
})

// getBitmap():
//   Returns a Bitmap representing what is visible in
//   the PreviewView.

val bitmap = previewView.getBitmap()
```

- PreviewView에 두 개의 API를 추가
- `getPreviewStreamState` API :
  - 미리보기가 streaming 중이거나 idle 상태 일 때 애플리케이션에 알릴 수 있음
  - 미리보기가 시작되는 동안 매우 일시적인 휴식 화면을 피하려는 애플리케이션에 유용
  - 예를 들어 미리보기가 idle 상태 일 때 PreviewView 위에 placeholder 이미지를 표시하고 streaming일때 placeholder 이미지를 숨김
- `getBitmap` API :
  - PreviewView에 표시된 현재 미리보기의 스냅샷을 가져옴

## 7. Preview with OpenGL - Advanced Preview

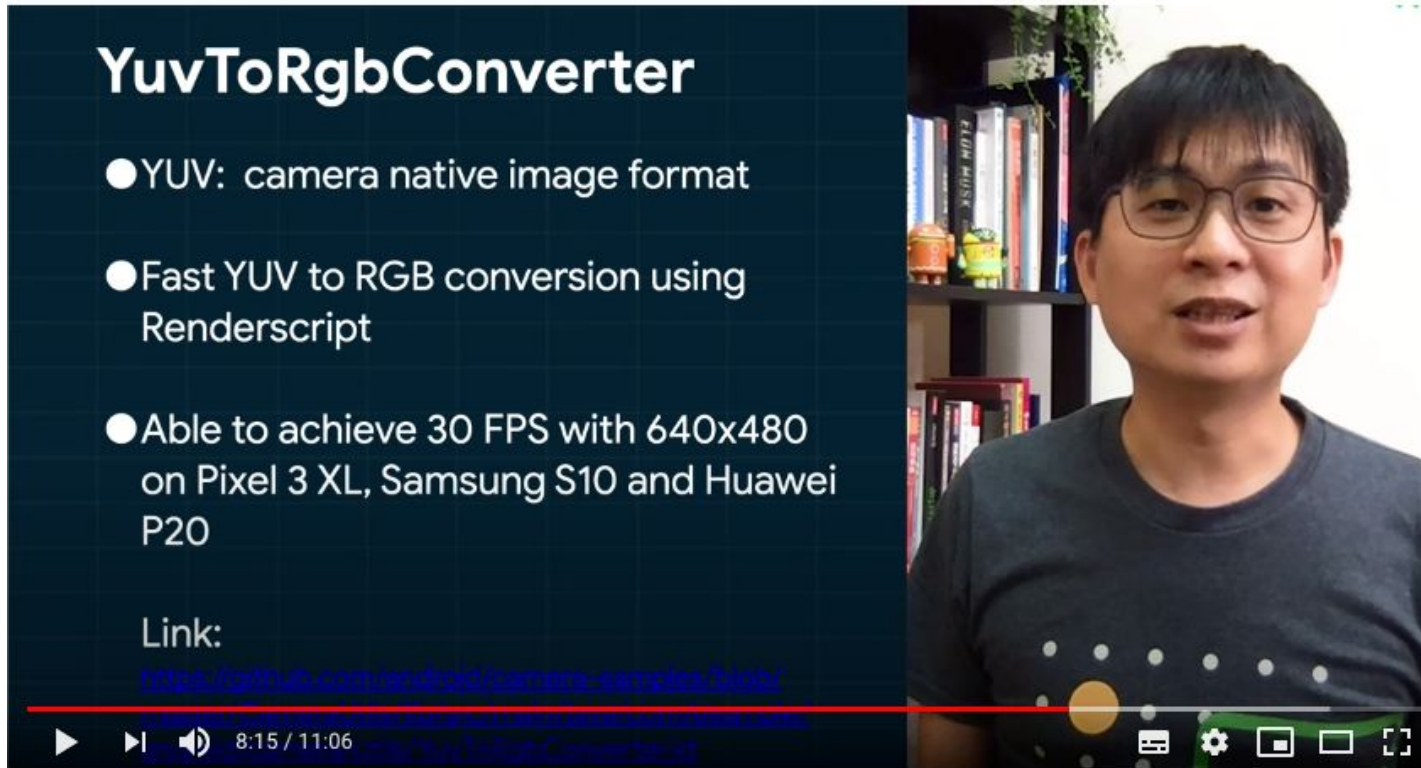


- 카메라 출력 위에 고급 효과를 주거나 앱에있는 기존 OpenGL 버킷이나 게임 엔진에 카메라 미리보기 통합할 때 미리보기 서비스 제공 업체 API를 통해 이를 수행하는 방법

1. Rendering Thread에 SurfaceTexture를 사용하여 Surface를 만들
2. Surface를 Preview SurfaceProvider에 제공
3. SurfaceProvider는 또한 표면을 해제해도 안전 할 때 애플리케이션에 신호를 제공
4. 애플리케이션은 이제 SurfaceTexture에서 카메라 프레임을 쉽게 수신 할 수 있으며 렌더링 삽입물은 OpenGL을 사용하여 SurfaceTexture에서 원하는 모든 작업을 수행가능.

<https://android.googlesource.com/platform/frameworks/support/+/refs/heads/androidx-master-dev/camera/integration-tests/coretestapp/src/main/java/androidx/camera/integration/core/OpenGLActivity.java>

## 8. YuvToRgbConverter



The image shows a YouTube video player. On the left, a presentation slide titled 'YuvToRgbConverter' is displayed. The slide has a dark blue background with white text. It lists three bullet points: 'YUV: camera native image format', 'Fast YUV to RGB conversion using Renderscript', and 'Able to achieve 30 FPS with 640x480 on Pixel 3 XL, Samsung S10 and Huawei P20'. Below the bullet points, it says 'Link:' followed by a GitHub URL. On the right, a man with glasses and a dark t-shirt is speaking. The video player interface at the bottom shows a progress bar at 8:15 / 11:06 and various control icons.

# YuvToRgbConverter

- YUV: camera native image format
- Fast YUV to RGB conversion using Renderscript
- Able to achieve 30 FPS with 640x480 on Pixel 3 XL, Samsung S10 and Huawei P20

Link:  
<https://github.com/android/camera-samples/blob/master/CameraUtils/lib/src/main/java/com/example/android/camera/Utils/YuvToRgbConverter.kt>

8:15 / 11:06

### - YUV

- camera native image format

- <https://ko.wikipedia.org/wiki/YUV>

### - RGB - <https://ko.wikipedia.org/wiki/RGB>

- Fast YUV to RGB conversion using RenderScript


- 640x480 해상도로 30FPS 달성 가능(Pixel 3XL, 갤럭시 S10, 화웨이 P20기준)

<https://github.com/android/camera-samples/blob/master/CameraUtils/lib/src/main/java/com/example/android/camera/Utils/YuvToRgbConverter.kt>

# Fun with ImageAnalysis

Samples for ImageAnalysis

- **ML Kit + CameraX :**
  - **Translate Showcase App**  
<https://github.com/googlesamples/mlkit/tree/master/android/translate-showcase>
- **Tensorflow Lite + CameraX**
  - **Real-time object detection**  
<https://github.com/android/camera-samples/tree/master/CameraXTfLite>

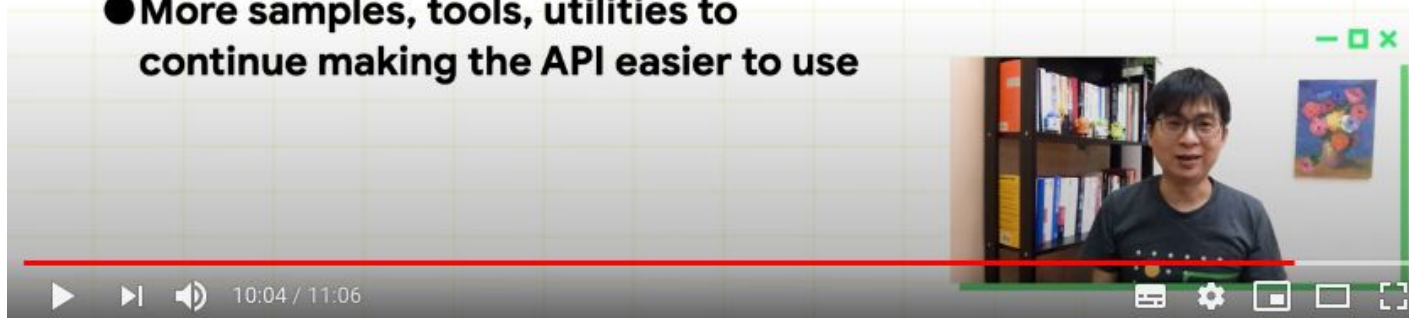


9:12 / 11:06

- ML Kit + CameraX:
  - Translate Showcase App  
<https://github.com/googlesamples/mlkit/tree/master/android/translate-showcase>
- Tensorflow lite + CameraX
  - Real-time object detection  
<https://github.com/android/camera-samples/tree/master/CameraXTfLite>

# What to expect

- Address critical issues you've filed
- Improved device compatibility & performance
- More samples, tools, utilities to continue making the API easier to use



- 제출된 critical issue 해결
- 기기 호환성, 성능 향상
- 더 많은 샘플, 도구, 유틸리티 지속적으로 쉽게 사용할 수 있는 API를 만듦

## Resources:

- Check out the 11 Weeks of Android website → <https://goo.gle/30FDT8S>
- CameraX OpenGL sample → <https://goo.gle/2Cm9piP>
- YuvToRgbConverter → <https://goo.gle/32Babmv>
- MLKit + CameraX Translate Showcase App → <https://goo.gle/2CRFC18>
- Tensorflow Lite + CameraX sample app → <https://goo.gle/2WBLWkA>
- CameraX documentation → <https://goo.gle/3jlk65H>
- CameraX Codelab → <https://goo.gle/3eOdRnv>
- CameraX Discussion Group → <https://goo.gle/39bsosi>
- CameraX public issue tracker → <https://goo.gle/2BfeQ2f>