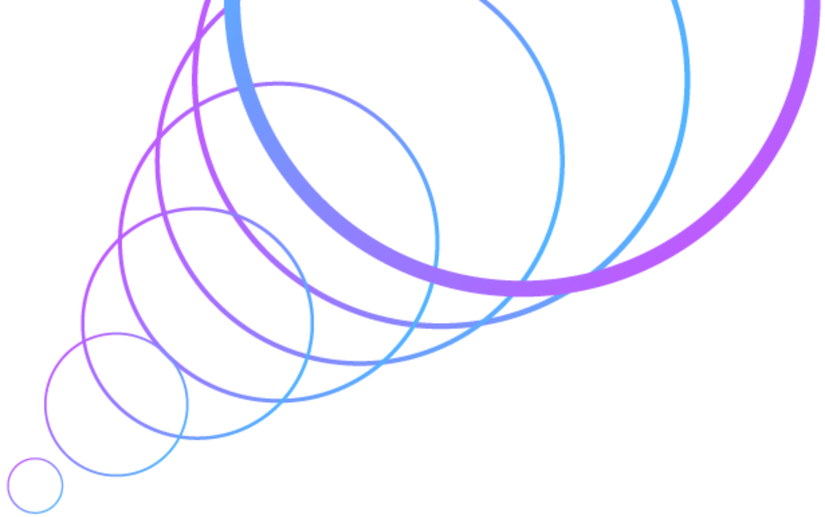
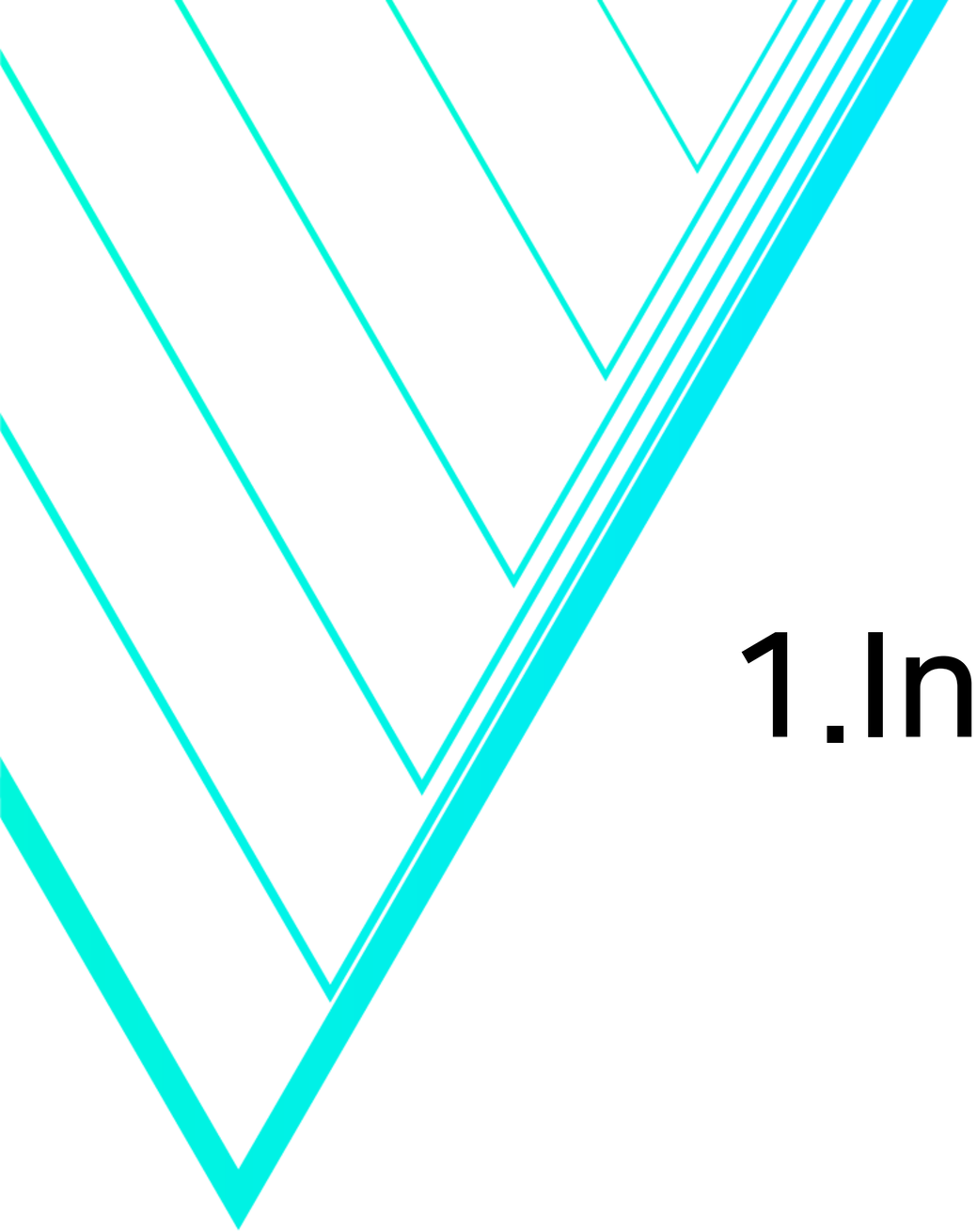


# Clova Face Kit: 10분 안에 누구나 적용하는 얼굴인식

강정현 정상엽 김범수 유준희
NAVER Clova

# CONTENTS



DEVIEW 2021

# 1.Introduction

# 1.1. Clova Face Kit?

DEVIEW 2021

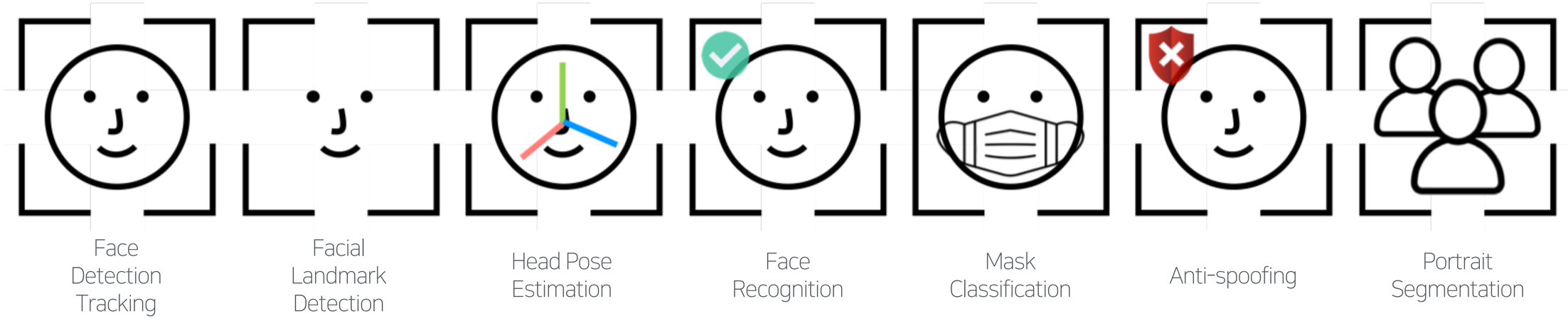
## 단말(edge)에서 구동되는 얼굴 인식 기술

| | |
|---|---|
| Total Solution<br>for face & human analysis | No need for **expensive GPU**<br>server |
| Support **cross-platform** | **Easy & Simple**<br>API |

# 1.2. 지원 기능

다양한 플랫폼 지원: Android, iOS, Linux, macOS, Windows, WebAssembly
다양한 인터페이스 지원: C++, JavaScript, Kotlin, Objective-C, Python

다양한 기능 지원:



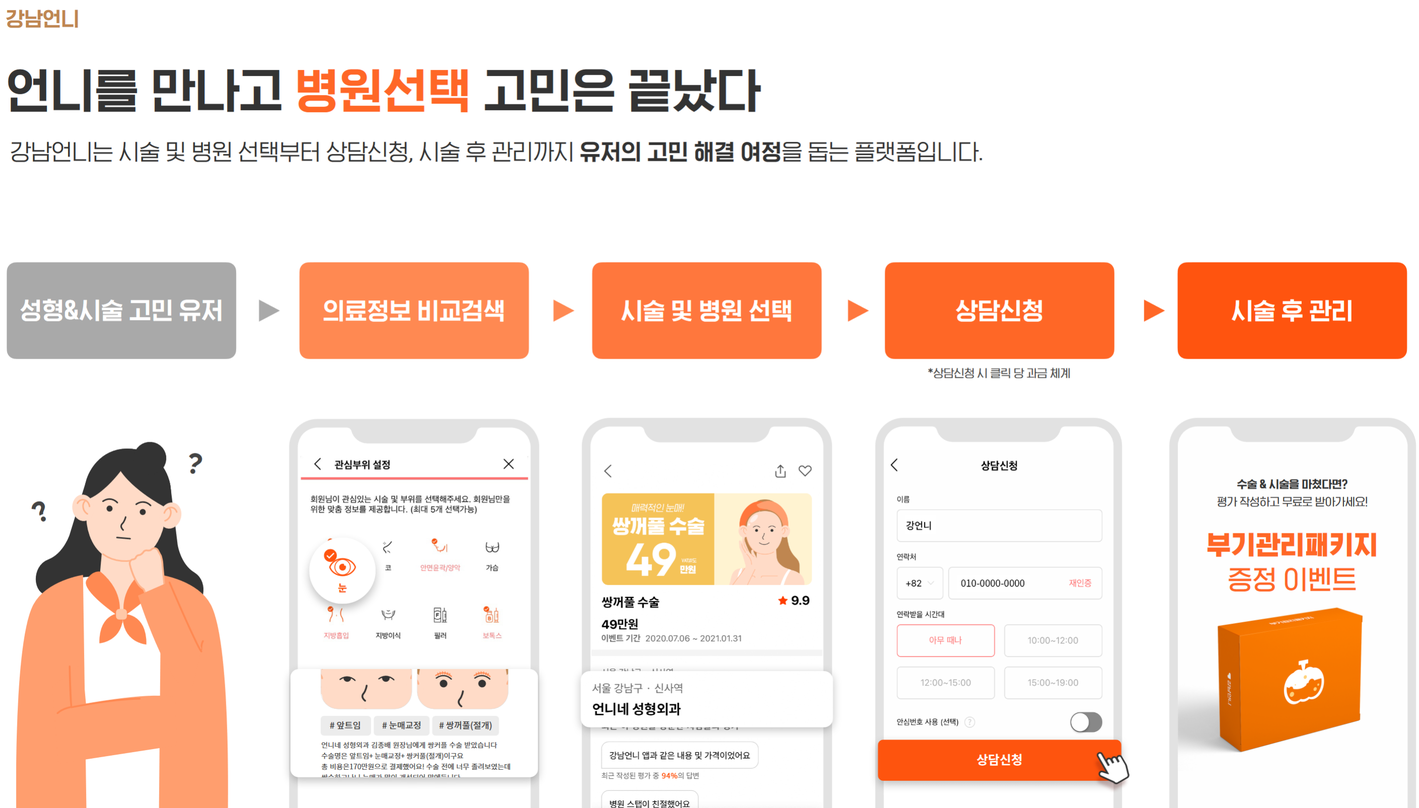
# 1.3. 사용처



- 강남언니
- Clova FaceSign
- LINE eKYC
- NAVER 포토클라우드

# 1.3. 강남언니

- 공급자 중심의 의료서비스를
- IT 기반 미용의료 정보 플랫폼을 통해
- 의료소비 주체가 중심 되는 시장으로



# 1.3. 강남언니 시술 후기



- 다른 사용자의 시술 후기는 시술 상품 선택에 큰 도움이 됨
- 사진이 포함된 후기는 신뢰도가 더욱 높아짐
- 사용자는 후기에 사진을 첨부할 때 일부 영역을 감추고 싶은 요구를 가짐
- 기존에는 각 사용자가 직접 각종 도구를 사용해 사진을 편집

# 1.3. Clova Face Kit 사용 후기 사진 보정


Clova
Face Kit



# 1.3. 강남언니의 Clova Face Kit 효과

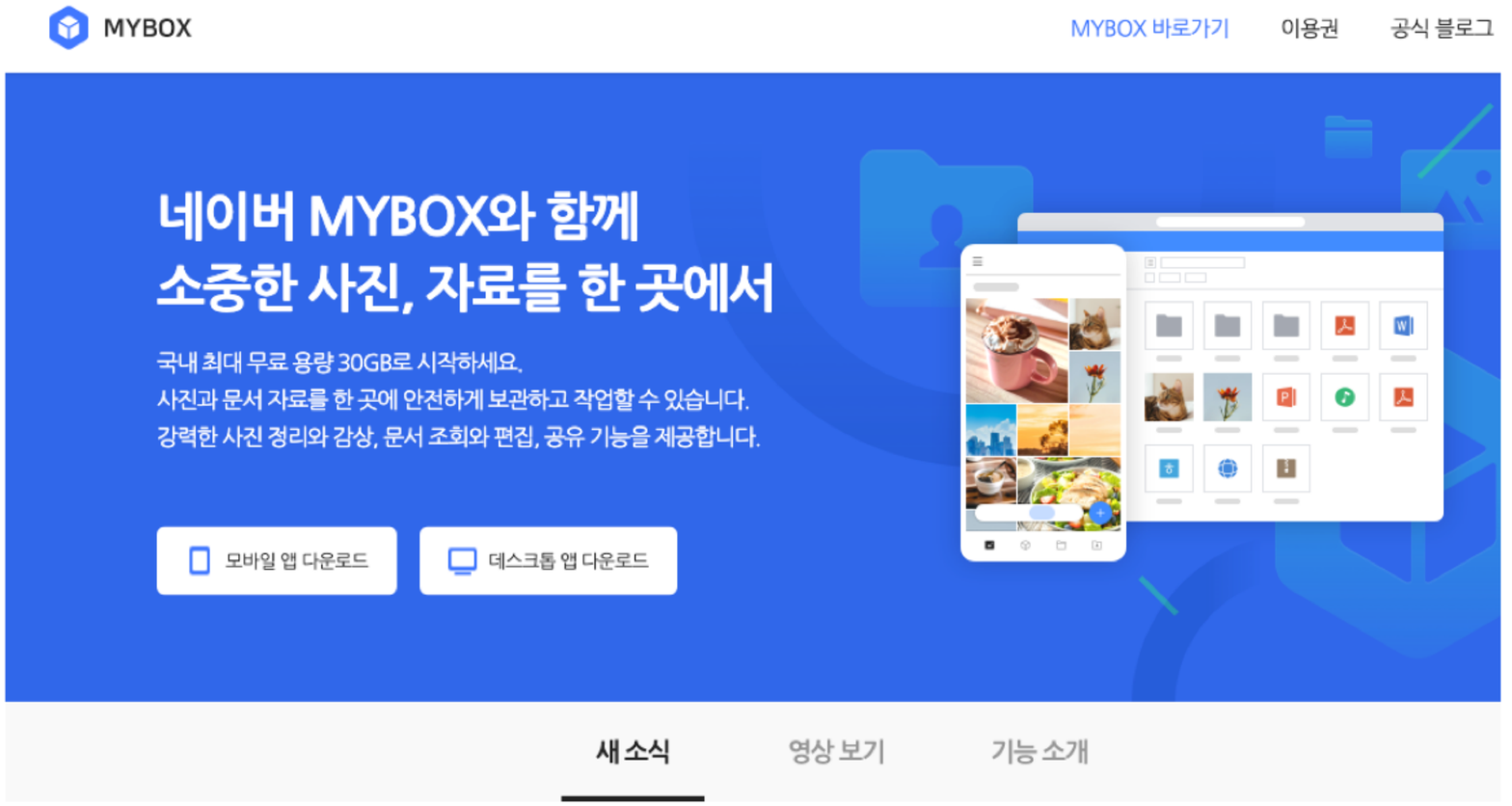
DEVIEW 2021

- 사용자는 민감한 얼굴 사진 중 후기에 해당하는 부분만 강남언니 서비스에 제공
- 강남언니 시스템 자원 대신 사용자의 단말기 자원 사용
- 일관된 사진 보정 효과
- 언제나 인적 자원이 부족한 스타트업이 준비된 도구를 사용해 매우 짧은 시간에 서비스 가치 창출

# 1.3. Clova FaceSign

# 1.3. NAVER 포토 클라우드





# 1.3. LINE eKYC

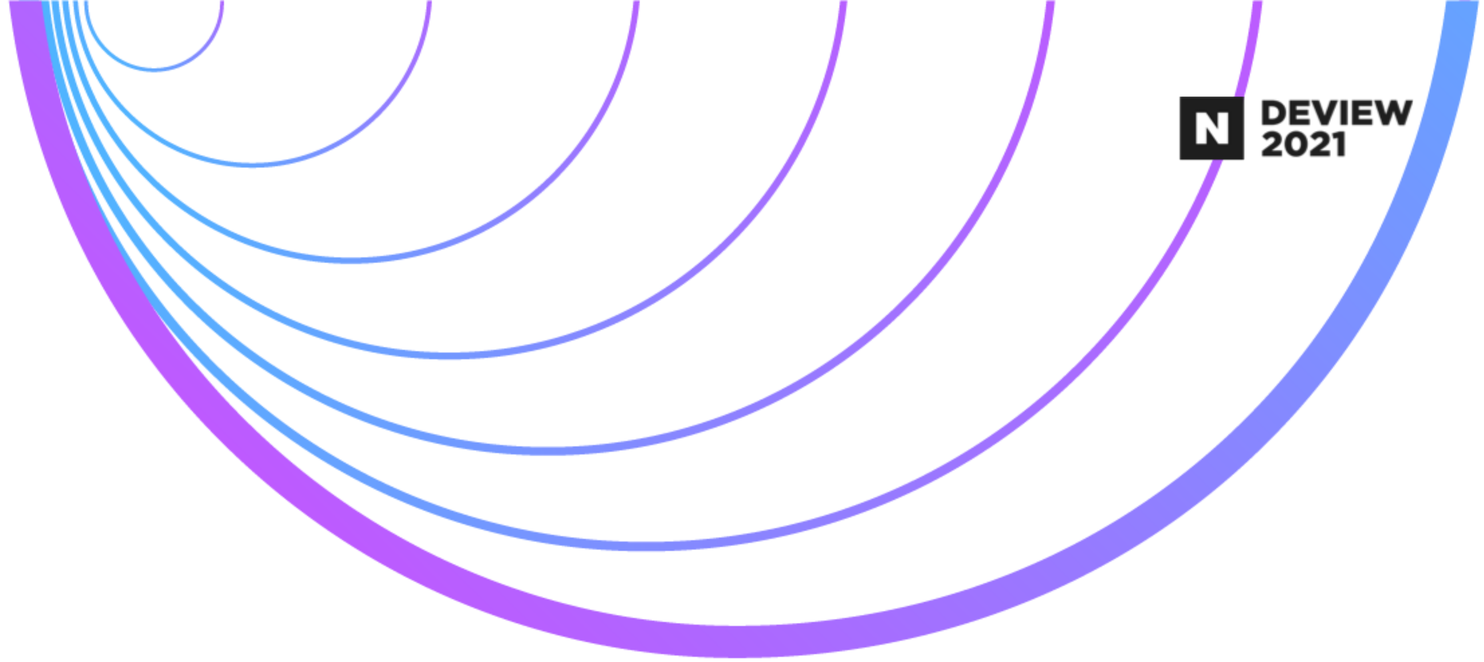

LINE eKYC
LINE eKYCとは
導入事例
ユースケース
特長
導入の流れ
LINE eKYCのお問い合わせ
LINEが開発した高精度のeKYCソリューション
安心・安全×スムーズなオンライン本人確認
LINE eKYC
資料請求・お問い合わせはこちら
自動で撮影を行います
3


# 2.Hands-On

# 2. Hands-On

DEVIEW 2021

1. Settings & Options
2. iOS
3. Android
4. Python
5. JavaScript



DEVIEW 2021

# 2.1. Settings & Options

# 2.1. Settings & Options

DEVIEW 2021

## Notice

- Clova Face Kit은 구 명칭인 ClovaSee로 부터 명칭을 변경하는 중입니다.
- Hands-On의 예제 곳곳에 ClovaSee의 흔적이 남아 있으나, 추후 변경 예정입니다.

# 2.1. Settings & Options



Settings: Clova Face Kit instance에 영향을 미치는 설정 값
Options: Clova Face Kit의 Frame 처리 1회에 대한 설정

- Platform 혹은 언어별로 명명 형식이 상이함. (camel case, snake case 등)
- Settings와 Options들의 값과 효과는 모두 동일함.
  - C/C++을 기반으로 하기 때문

# 2.1. Settings



## 모든 Settings는 SettingsBuilder로 부터!

예시)

```
Settings s = SettingsBuilder().SetIntermittentInformationRatio(v)
                              .SetNumberOfThreads(n)
                              .build()
```

# 2.1. Settings



```
SetIntermittentInformationRatio(uint8_t value);
```

- Clova Face Kit의 정보(특징점, 각도 등)을 몇 Frame마다 생성할지 설정
- 10을 설정하는 경우, 0~9 Frame에 대해서는 정보를 생성하지 않음.
- 얼굴을 찾는 것(bounding box)은 예외.
- Default: 1

```
SetNumberOfThreads(int value)
```

- Thread를 최대 몇개까지 사용할지 설정
- Default: 4

# 2.1. Settings



```
SetPerformanceMode(const Settings::PerformanceMode& value)
```

- 얼굴의 특징점을 찾는 방식을 설정.
- kAccurate106, kAccurate98, kFast  값을 사용 가능.

# 2.1. Options



## 모든 Options는 OptionsBuilder로 부터

```
Options op =
OptionsBuilder().OptionsSetInformationToObtain(v)
                .SetSmoothingRect(b)
                .build()
```

# 2.1. Options



`SetBoundingBoxThreshold(float value)`

- 얼굴 탐지결과 반환시 confidence가 value 이하의 결과값을 가진 bounding box들을 사용하지 않습니다.

`SetMinimumBoundingBoxSize(float value)`

- Frame을 기준으로 얼굴인식 결과의 최소 크기를 설정합니다.
- Value보다 작은 얼굴은 인식결과에서 제외합니다.

# 2.1. Options



`SetResizeThreshold(int value)`
- 주어진 Frame의 처리를 위해 Resize할 때 장축의 길이를 설정합니다. 예를 들어 값이 320이고 Frame의 크기가 620, 100이었다면, 내부 Resizing에서 Frame의 크기는 320, 50으로 변경됩니다.
- Default: 320

`SetSmoothingRect(bool value)`
- Frame의 변화에 따라 Bounding box들이 변화할 때 튀는 정도를 부드럽게 바꿉니다.
- Default: false

`SetSmoothingContour(bool value)`
- Frame의 변화에 따라 Lamdmark들이 변화할 때 튀는 정도를 부드럽게 바꿉니다.
- Default: true

# 2.1. Options



`SetInformationToObtain(uint8_t value)`

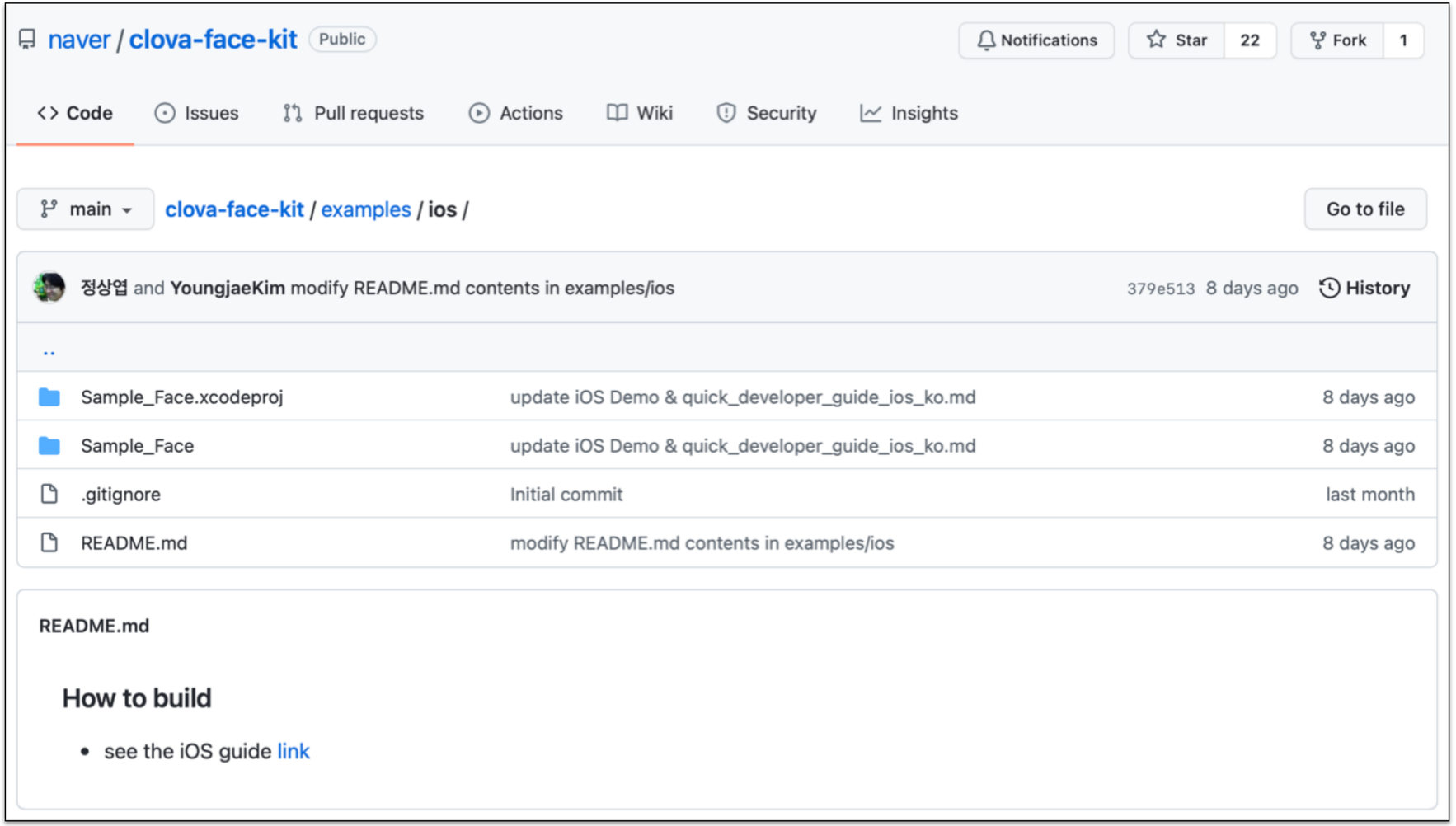
- Bounding box외에 추가로 얻을 정보를 flag로 전달합니다.

- kBoundingBoxes
  얼굴이 있는 영역의 사각형 정보를 반환합니다.
- kContours
  얼굴의 윤곽선 정보를 추가하여 반환합니다.
- kMasks
  마스크 착용 여부를 추가하여 반환합니다.
- kTrackingIDs
  얼굴에 추적 ID를 추가하여 반환합니다.
- kSpoofs
  얼굴이 진짜 얼굴인지 여부를 추가하여 반환합니다.
- kEulerAngles
  얼굴의 각도: Yaw, Roll, Pitch 값을 추가하여 반환합니다.
- kAll
  위 정보를 모두 추가하여 반환합니다.

# 2.2. iOS Hands-On

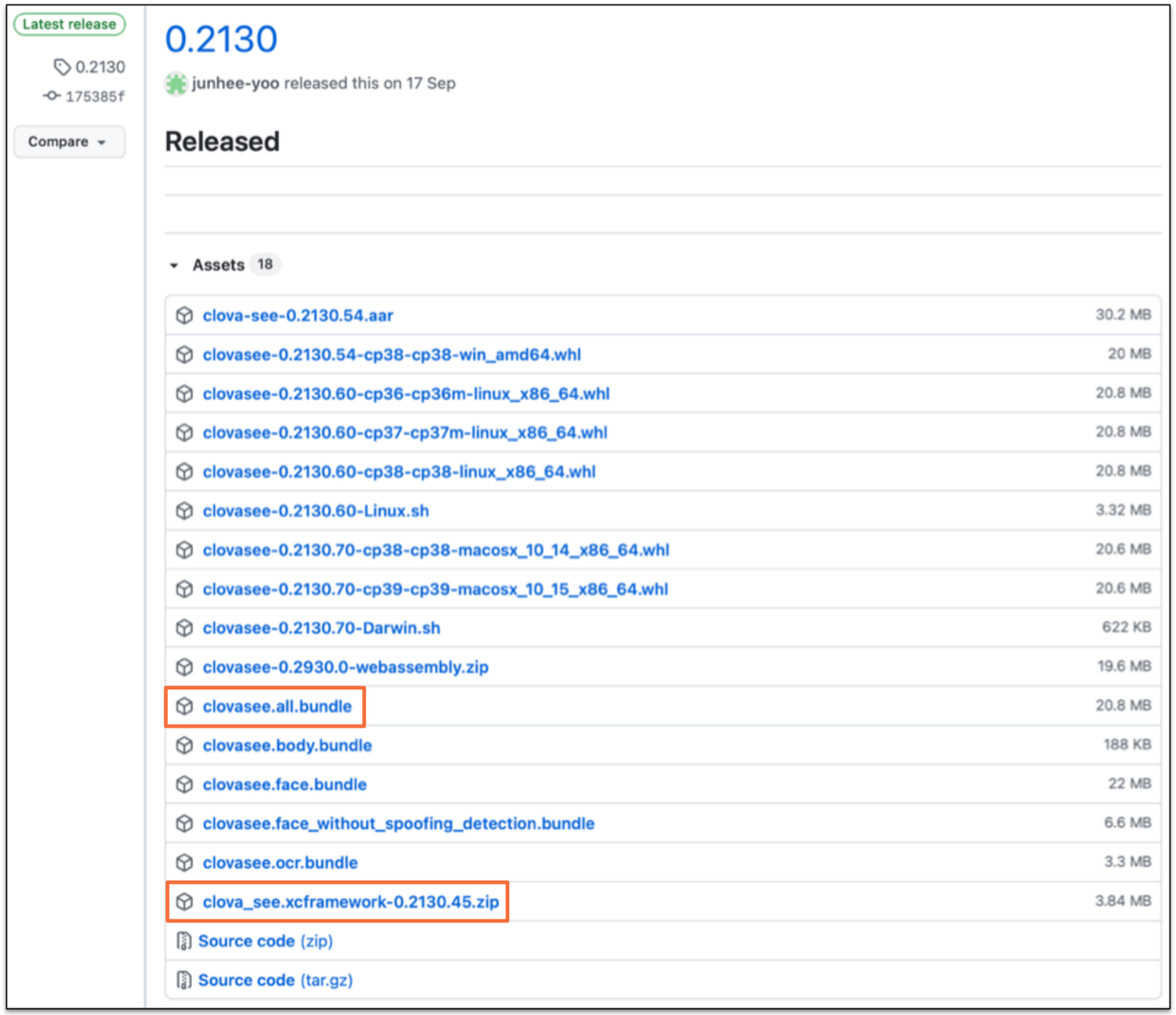
# 2.2. iOS Hands-On

DEVIEW 2021

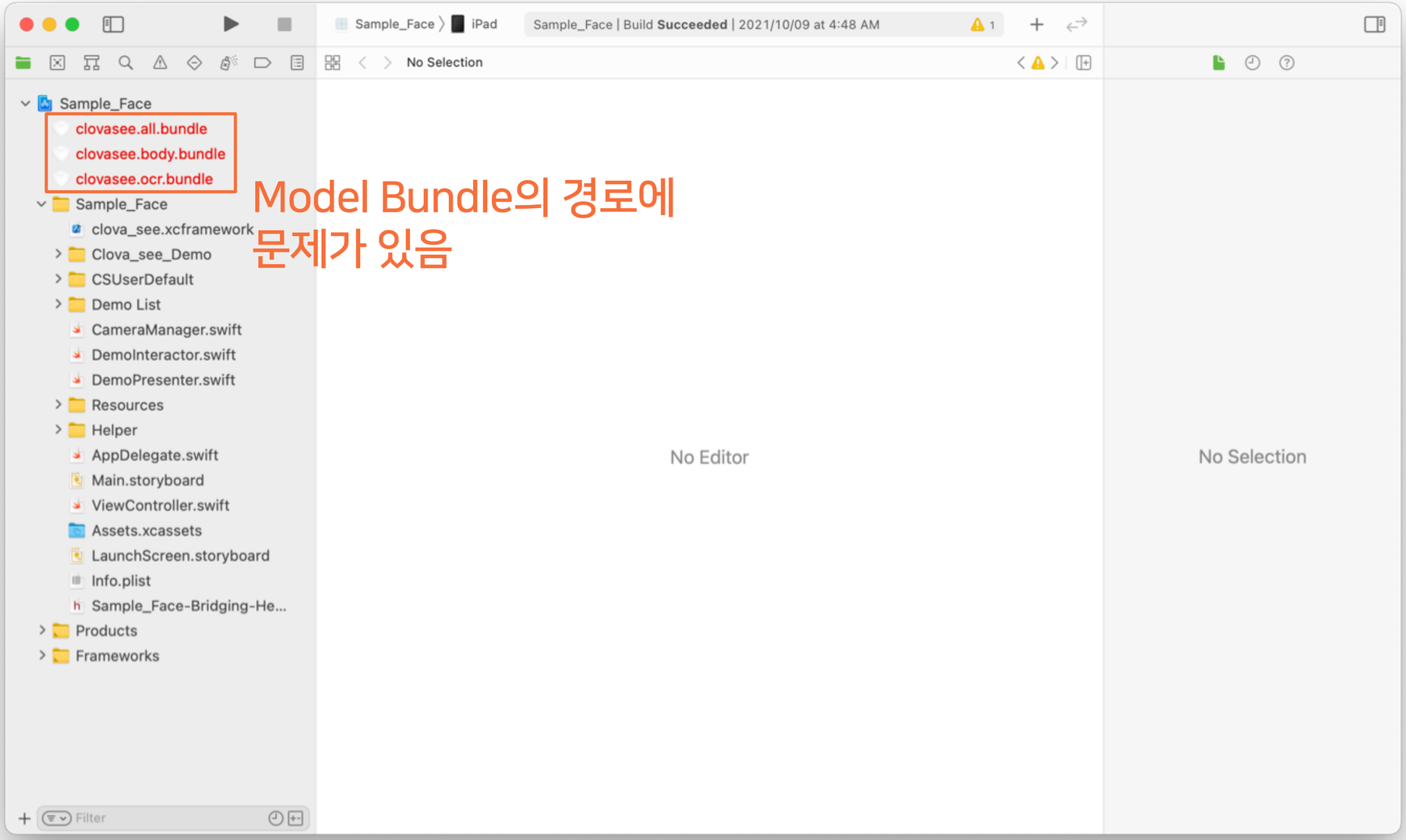


# 2.2. iOS Hands-On

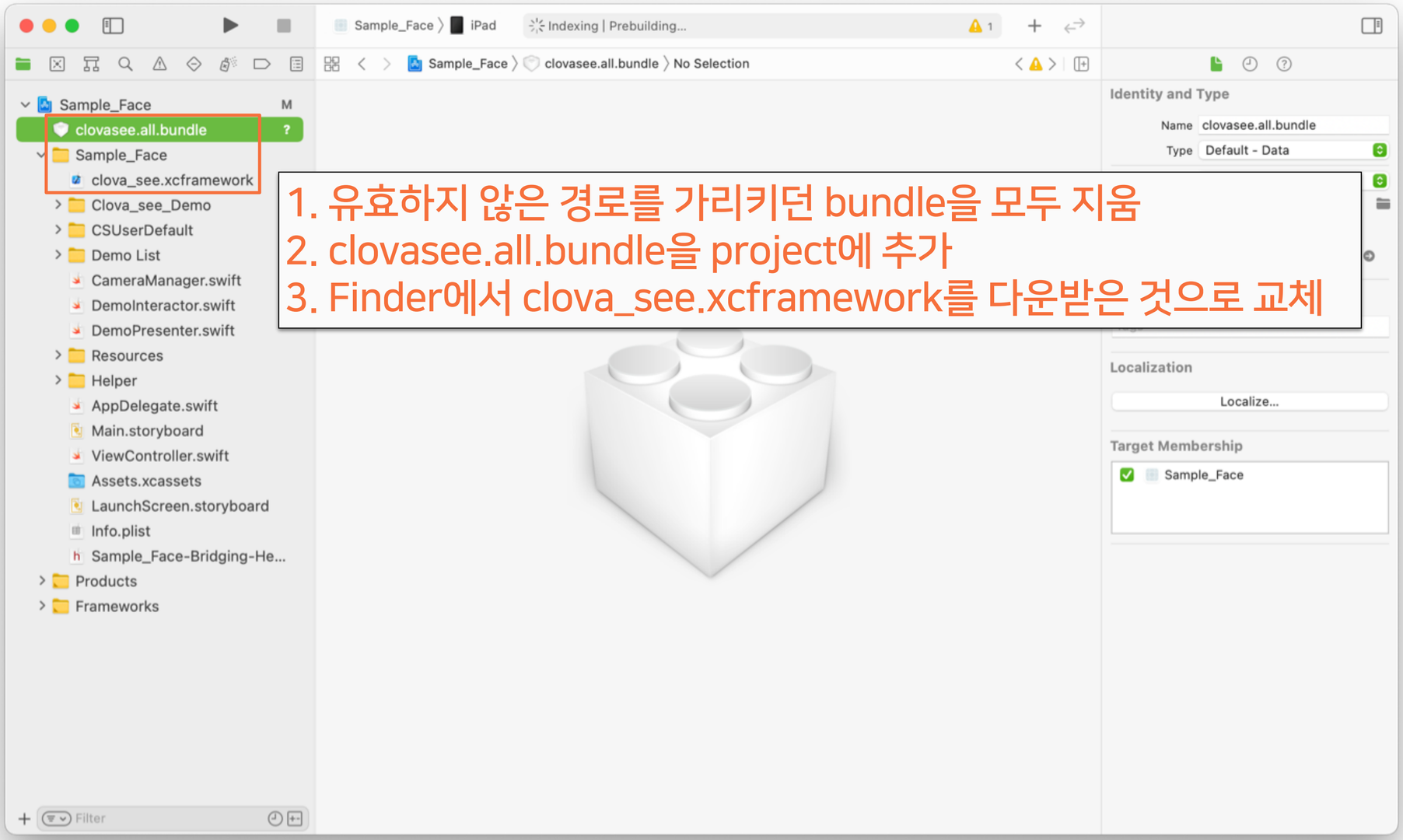




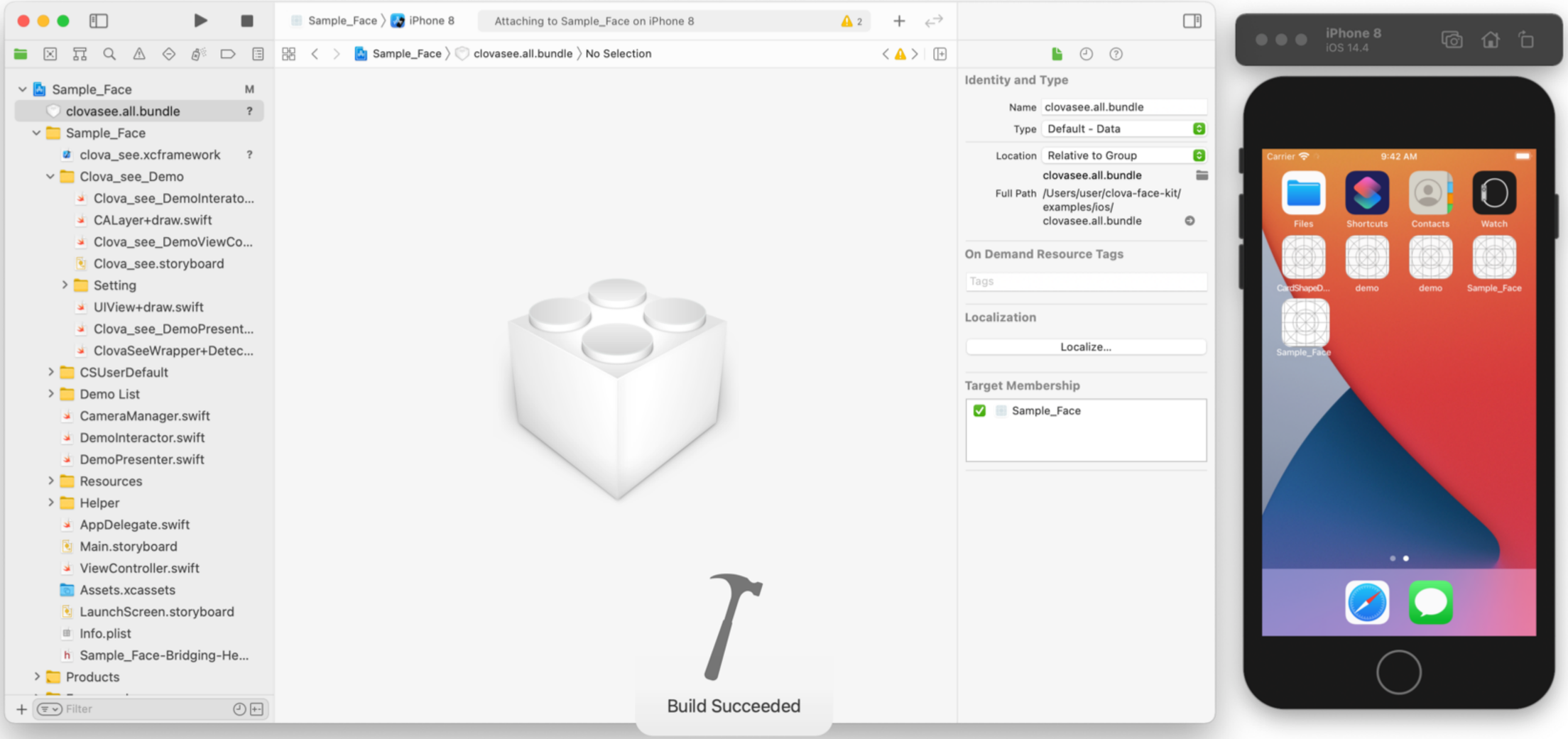
# 2.2. iOS Hands-On



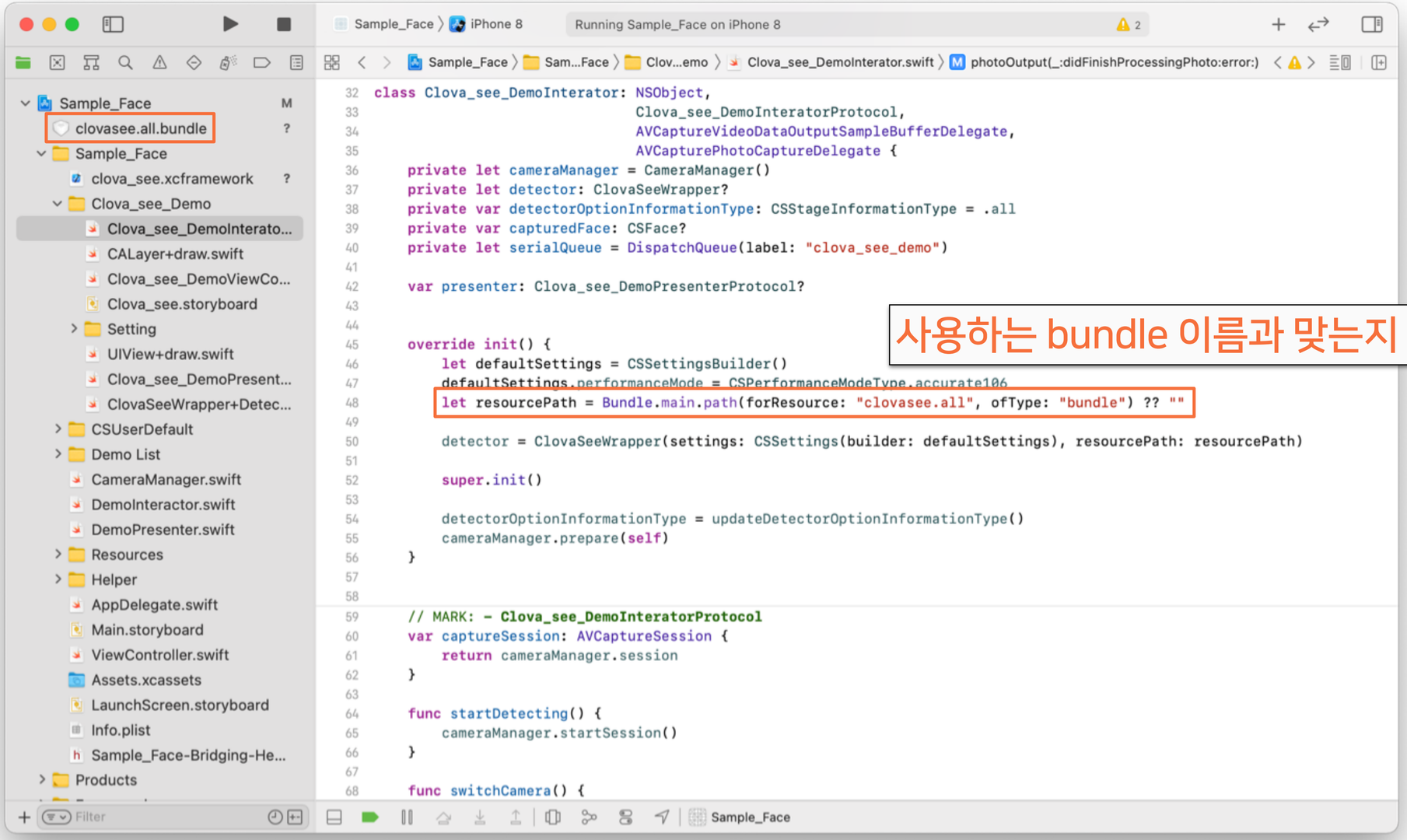
# 2.2. iOS Hands-On



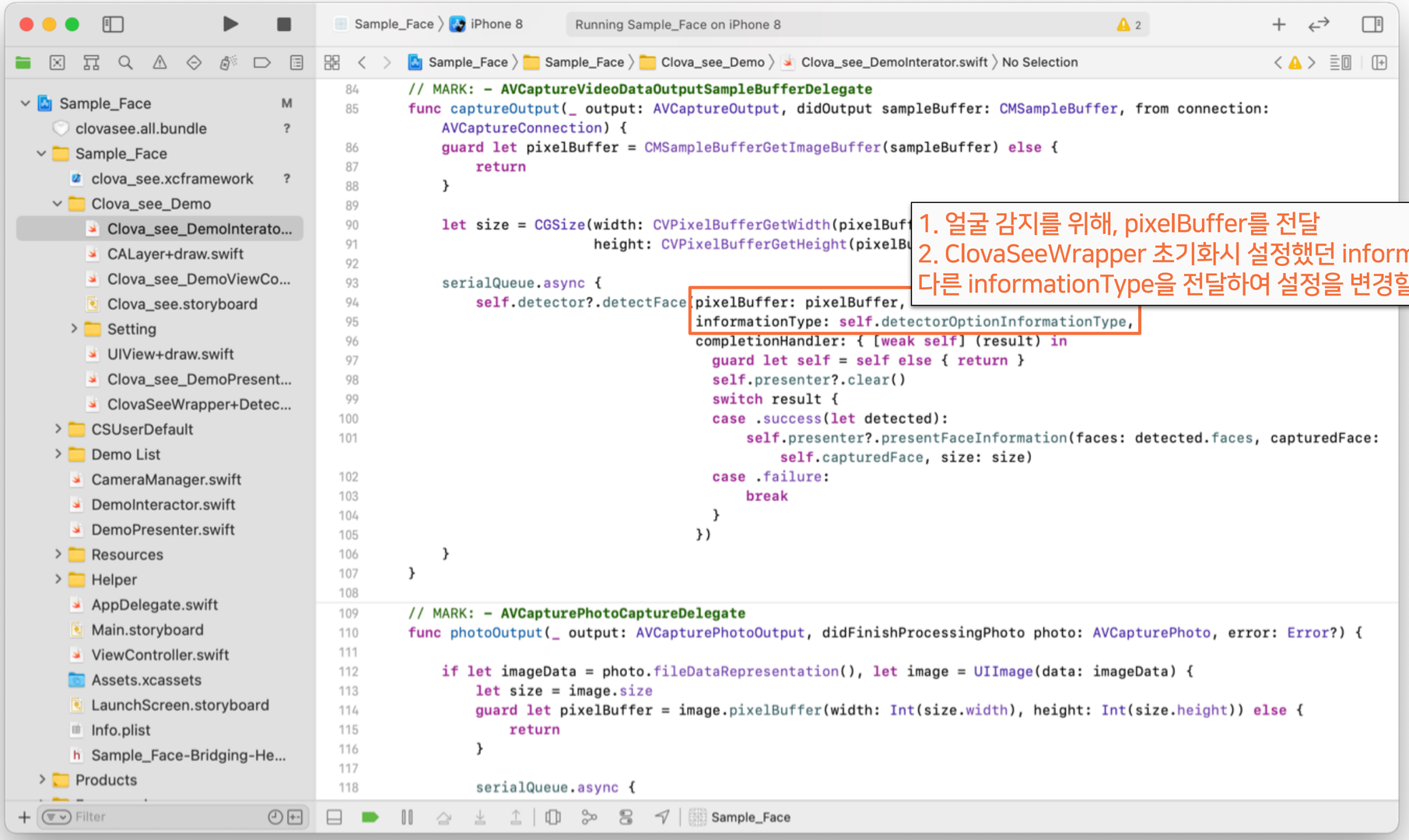
# 2.2. iOS Hands-On



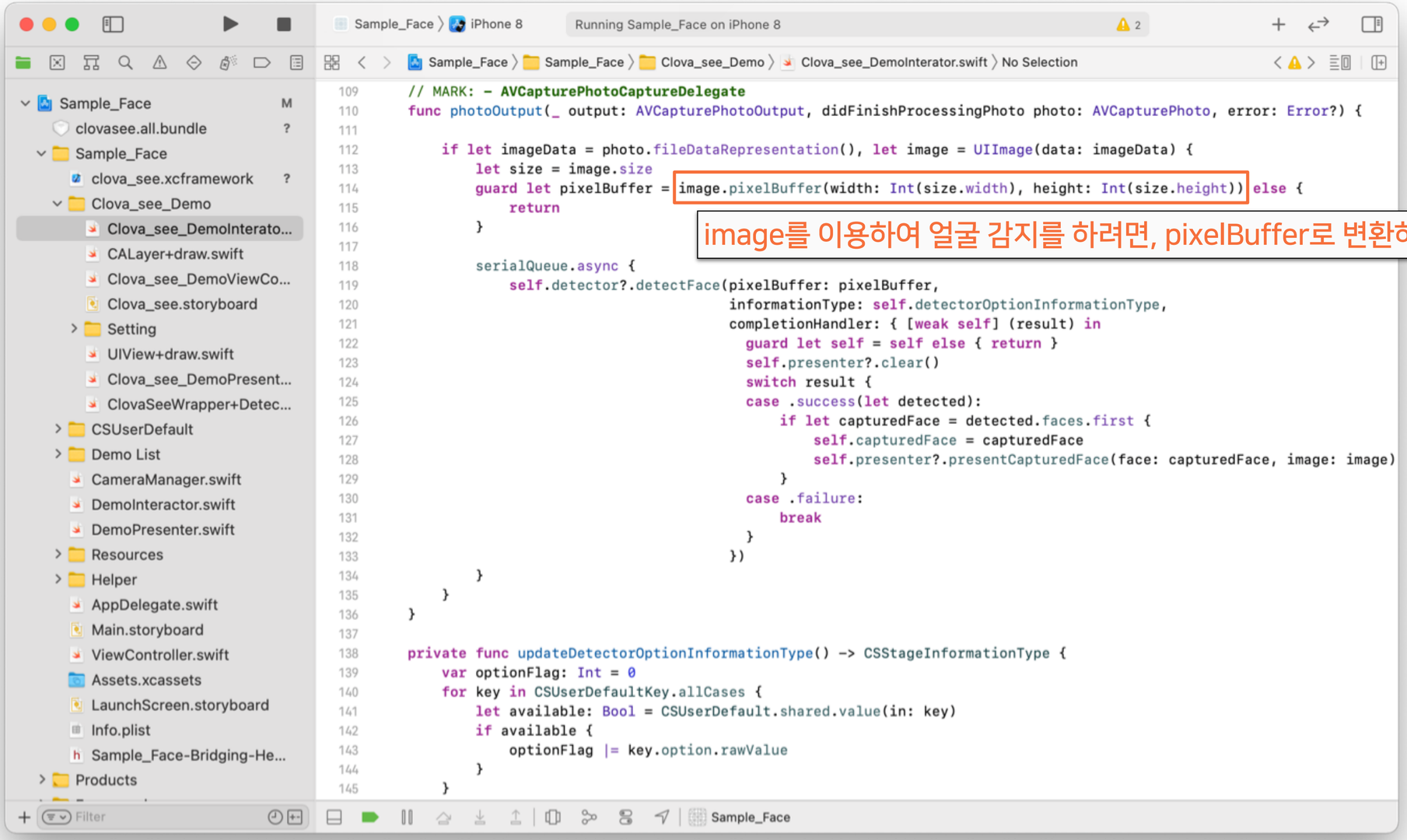
# 2.2. iOS Hands-On



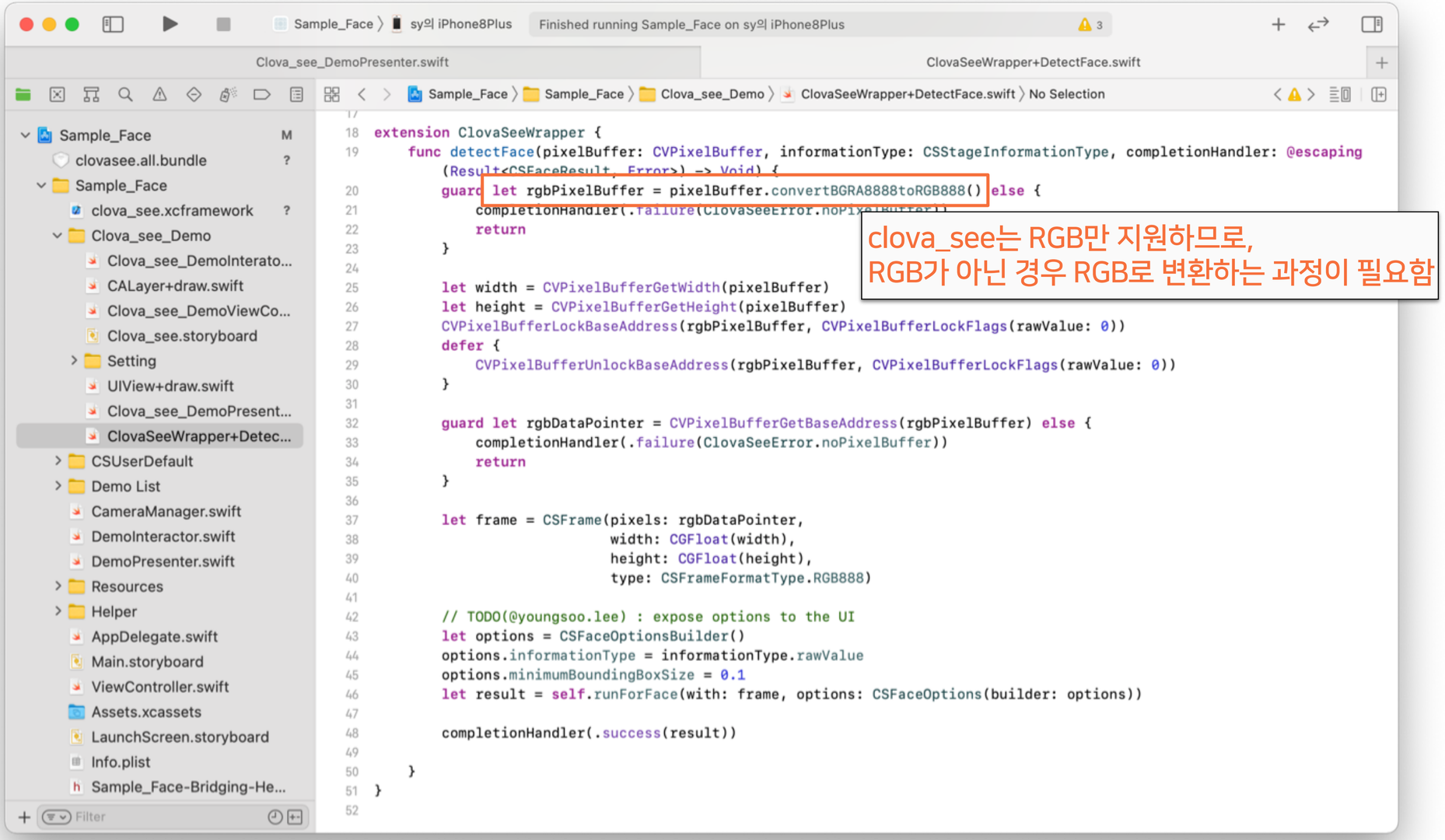
# 2.2. iOS Hands-On



# 2.2. iOS Hands-On

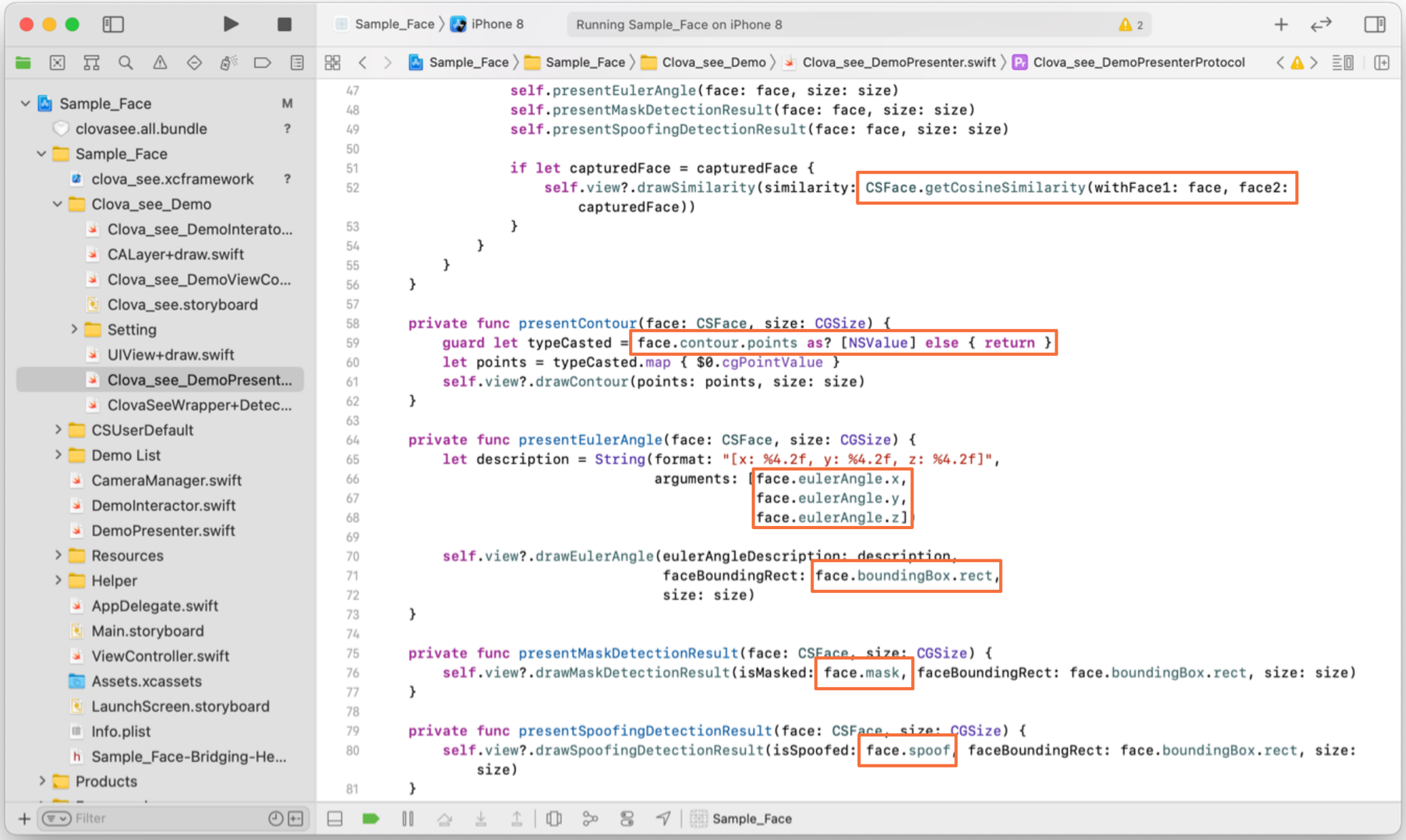


# 2.2. iOS Hands-On



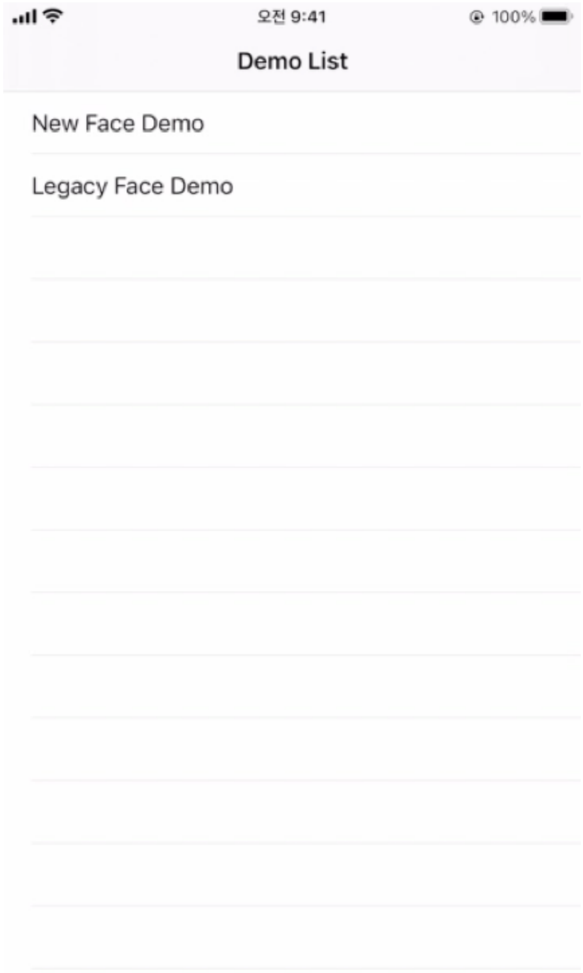

```
extension ClovaSeeWrapper {
    func detectFace(pixelBuffer: CVPixelBuffer, informationType: CSStageInformationType, completionHandler: @escaping
        (Result<CSFaceResult, Error>) -> Void) {
        guard let rgbPixelBuffer = pixelBuffer.convertBGRA8888toRGB888() else {
            completionHandler(.failure(ClovaSeeError.noPixelBuffer))
            return
        }

        let width = CVPixelBufferGetWidth(pixelBuffer)
        let height = CVPixelBufferGetHeight(pixelBuffer)
        CVPixelBufferLockBaseAddress(rgbPixelBuffer, CVPixelBufferLockFlags(rawValue: 0))
        defer {
            CVPixelBufferUnlockBaseAddress(rgbPixelBuffer, CVPixelBufferLockFlags(rawValue: 0))
        }

        guard let rgbDataPointer = CVPixelBufferGetBaseAddress(rgbPixelBuffer) else {
            completionHandler(.failure(ClovaSeeError.noPixelBuffer))
            return
        }

        let frame = CSFrame(pixels: rgbDataPointer,
                            width: CGFloat(width),
                            height: CGFloat(height),
                            type: CSFrameFormatType.RGB888)

        // TODO(@youngsoo.lee) : expose options to the UI
        let options = CSFaceOptionsBuilder()
        options.informationType = informationType.rawValue
        options.minimumBoundingBoxSize = 0.1
        let result = self.runForFace(with: frame, options: CSFaceOptions(builder: options))

        completionHandler(.success(result))

    }
}
```

# 2.2. iOS Hands-On



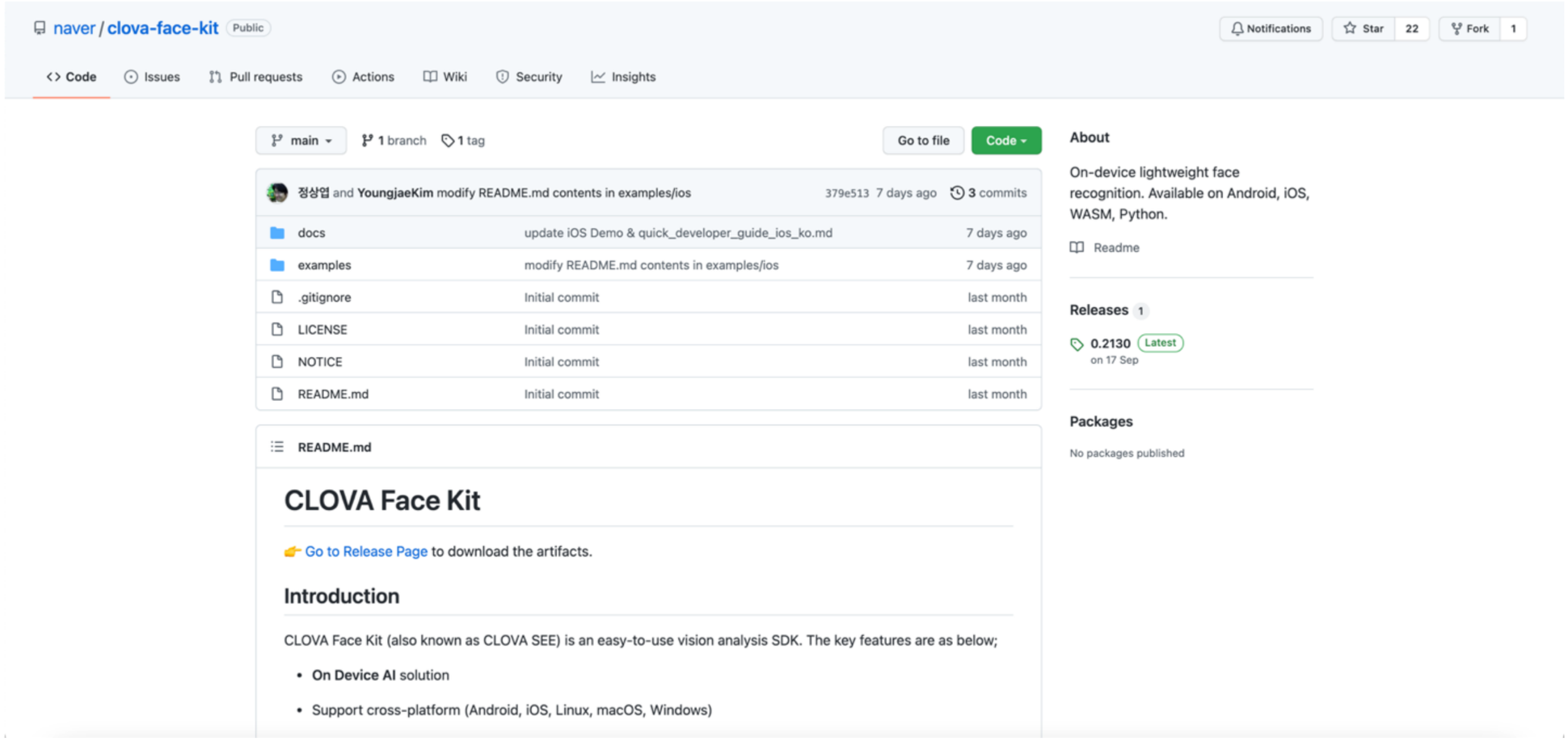
# 2.2. iOS Hands-On

DEVIEW 2021



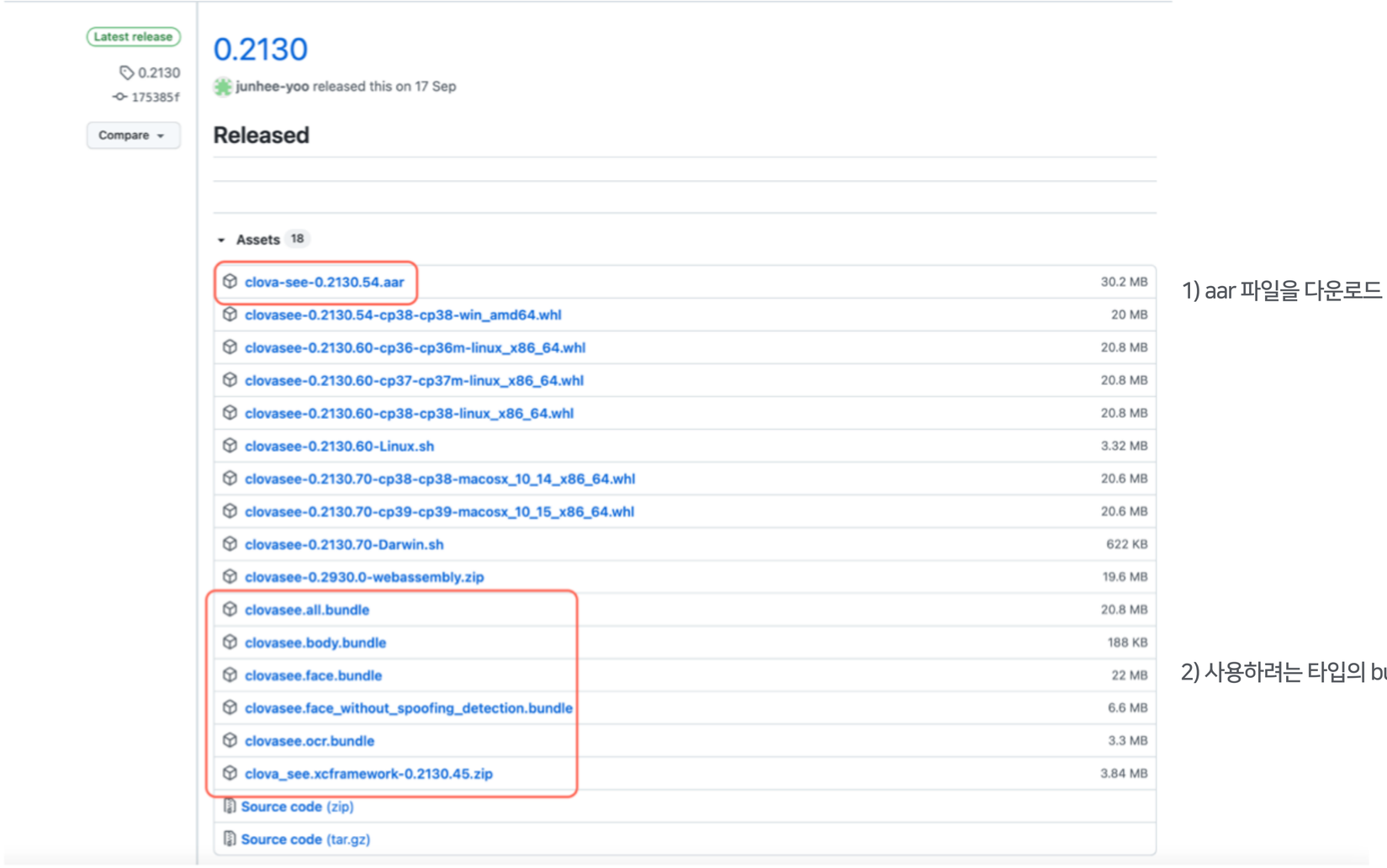
# 2.3. Android Hands-On

# 2.3.1 clova-face-kit download




https://github.com/naver/clova-face-kit

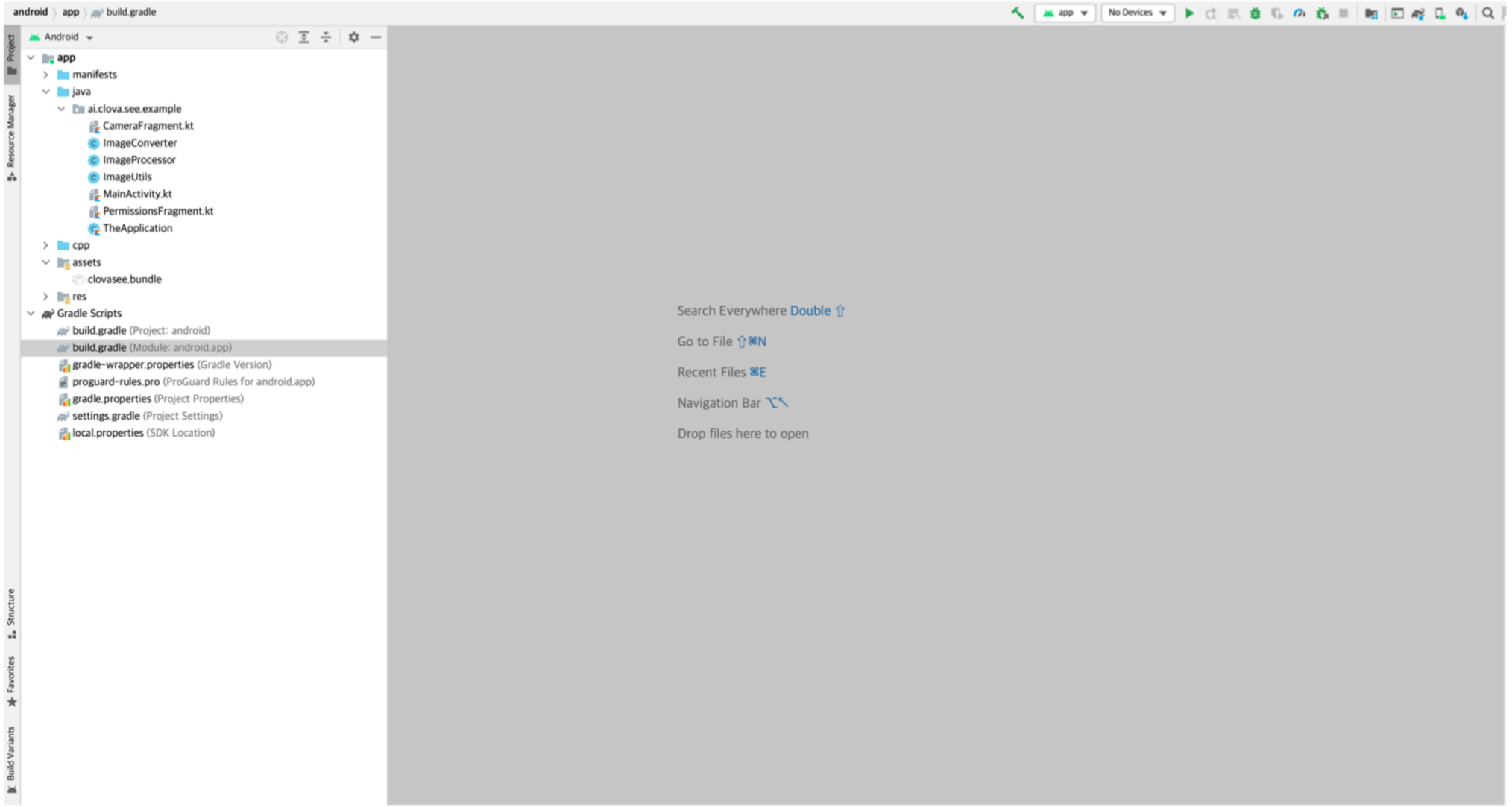
# 2.3.1 clova-face-kit download




1) aar 파일을 다운로드

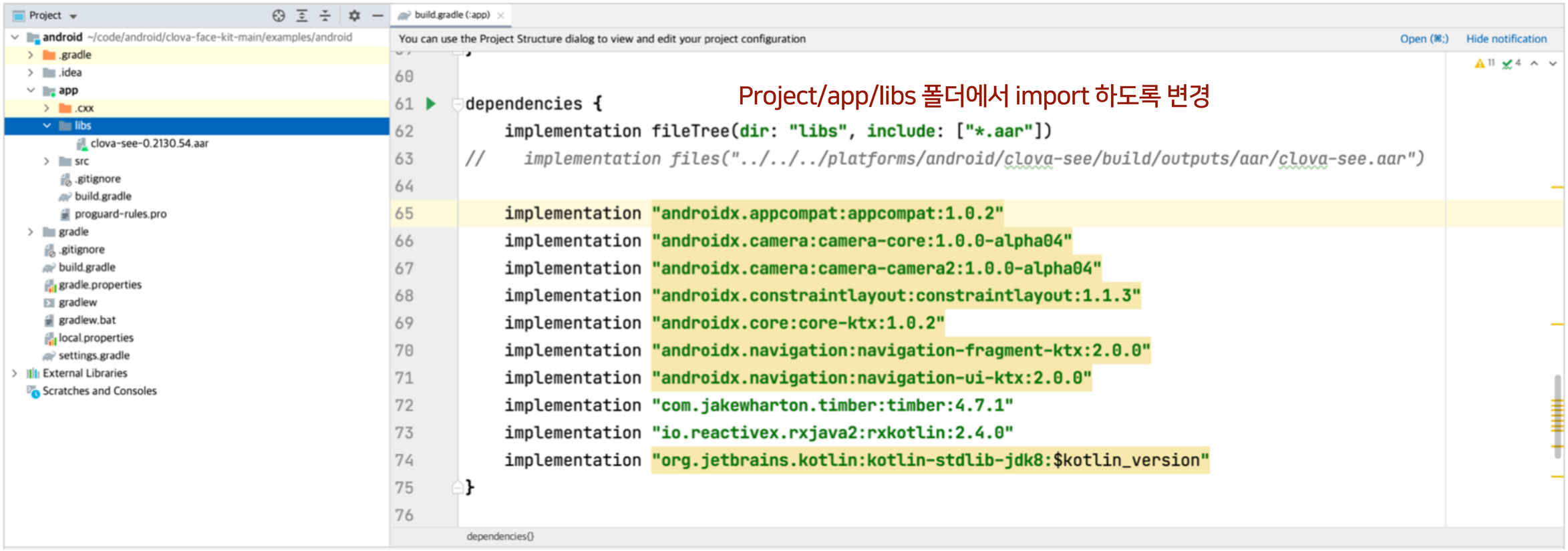
2) 사용하려는 타입의 bundle file 다운로드

# 2.3.2 clova-face-kit open

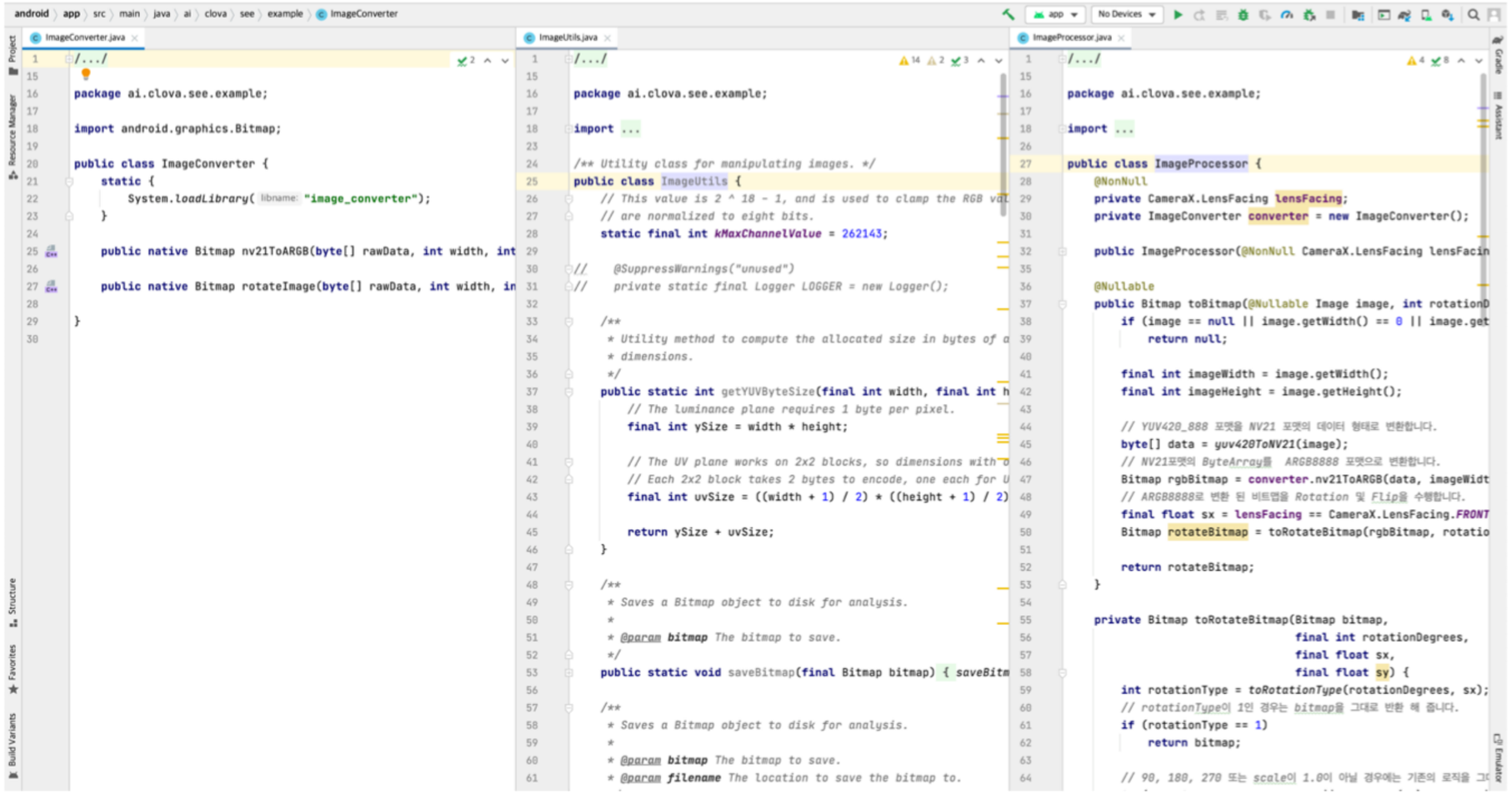


# 2.3.3 clova-face-kit source code





# 2.3.3 clova-face-kit source code



ImageConverter.java

```
/.../

package ai.clova.see.example;

import android.graphics.Bitmap;

public class ImageConverter {
    static {
        System.loadLibrary( libname: "image_converter");
    }

    public native Bitmap nv21ToARGB(byte[] rawData, int width, int

    public native Bitmap rotateImage(byte[] rawData, int width, in

}
```

ImageUtils.java

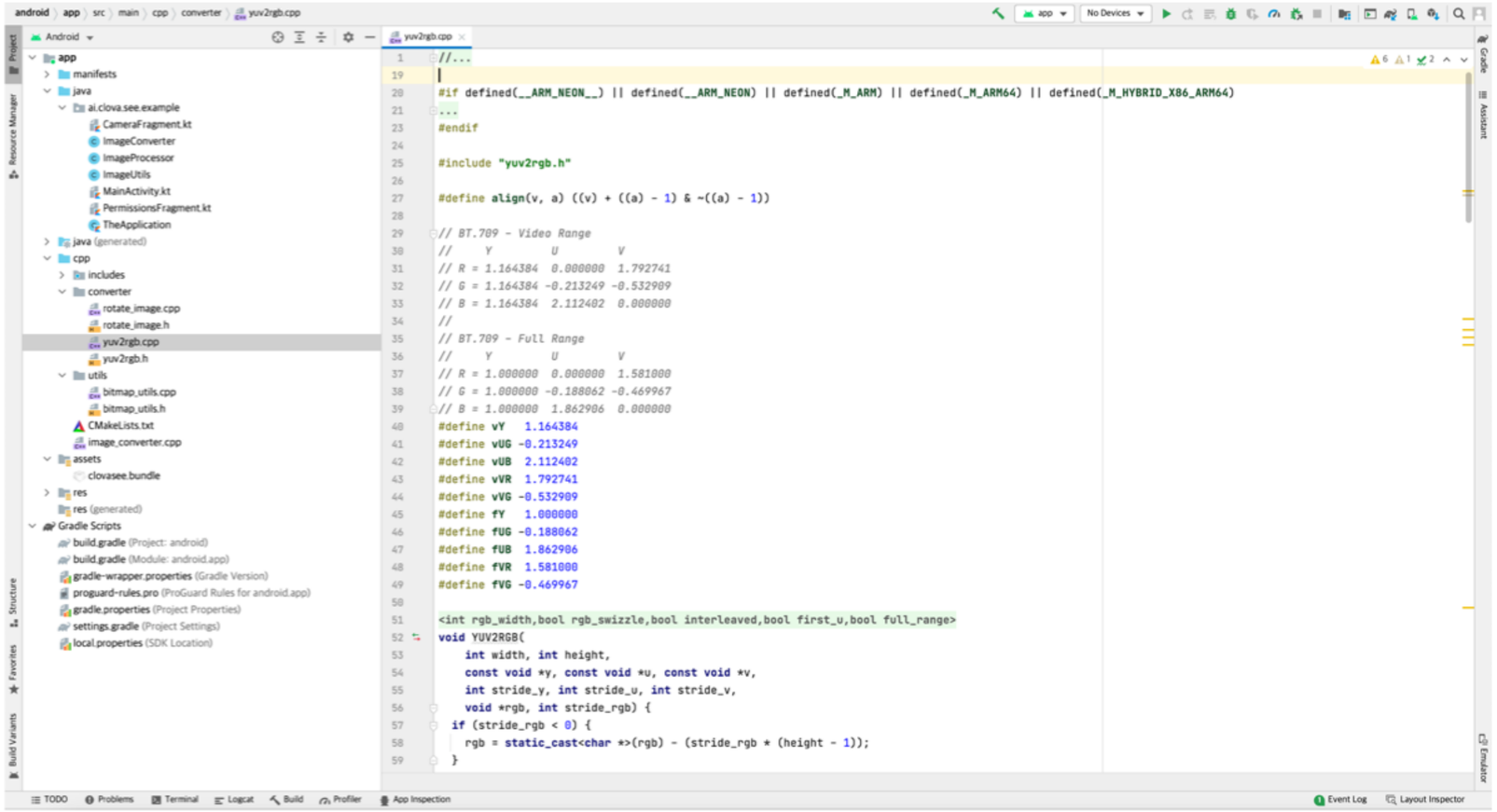
```
/.../

package ai.clova.see.example;

import ...

/** Utility class for manipulating images. */
public class ImageUtils {
    // This value is 2 ^ 18 - 1, and is used to clamp the RGB val
    // are normalized to eight bits.
    static final int kMaxChannelValue = 262143;

//    @SuppressWarnings("unused")
//    private static final Logger LOGGER = new Logger();

    /**
     * Utility method to compute the allocated size in bytes of a
     * dimensions.
     */
    public static int getYUVByteSize(final int width, final int h
        // The luminance plane requires 1 byte per pixel.
        final int ySize = width * height;

        // The UV plane works on 2x2 blocks, so dimensions with o
        // Each 2x2 block takes 2 bytes to encode, one each for U
        final int uvSize = ((width + 1) / 2) * ((height + 1) / 2)

        return ySize + uvSize;
    }

    /**
     * Saves a Bitmap object to disk for analysis.
     *
     * @param bitmap The bitmap to save.
     */
    public static void saveBitmap(final Bitmap bitmap) { saveBitm

    /**
     * Saves a Bitmap object to disk for analysis.
     *
     * @param bitmap The bitmap to save.
     * @param filename The location to save the bitmap to.
```

ImageProcessor.java

```
/.../

package ai.clova.see.example;

import ...

public class ImageProcessor {
    @NonNull
    private CameraX.LensFacing lensFacing;
    private ImageConverter converter = new ImageConverter();

    public ImageProcessor(@NonNull CameraX.LensFacing lensFacin

    @Nullable
    public Bitmap toBitmap(@Nullable Image image, int rotationD
        if (image == null || image.getWidth() == 0 || image.get
            return null;

        final int imageWidth = image.getWidth();
        final int imageHeight = image.getHeight();

        // YUV420_888 포맷을 NV21 포맷의 데이터 형태로 변환합니다.
        byte[] data = yuv420ToNV21(image);
        // NV21포맷의 ByteArray를  ARGB8888 포맷으로 변환합니다.
        Bitmap rgbBitmap = converter.nv21ToARGB(data, imageWidt
        // ARGB8888로 변환 된 비트맵을 Rotation 및 Flip을 수행합니다.
        final float sx = lensFacing == CameraX.LensFacing.FRONT
        Bitmap rotateBitmap = toRotateBitmap(rgbBitmap, rotatio

        return rotateBitmap;
    }

    private Bitmap toRotateBitmap(Bitmap bitmap,
                                  final int rotationDegrees,
                                  final float sx,
                                  final float sy) {
        int rotationType = toRotationType(rotationDegrees, sx);
        // rotationType이 1인 경우는 bitmap을 그대로 반환 해 줍니다.
        if (rotationType == 1)
            return bitmap;

        // 90, 180, 270 또는 scale이 1.0이 아닐 경우에는 기존의 로직을 그
```

CameraX에서 받은 ImageProxy를 bitmap으로 변환

# 2.3.3 clova-face-kit source code




yuv_420_888 ->

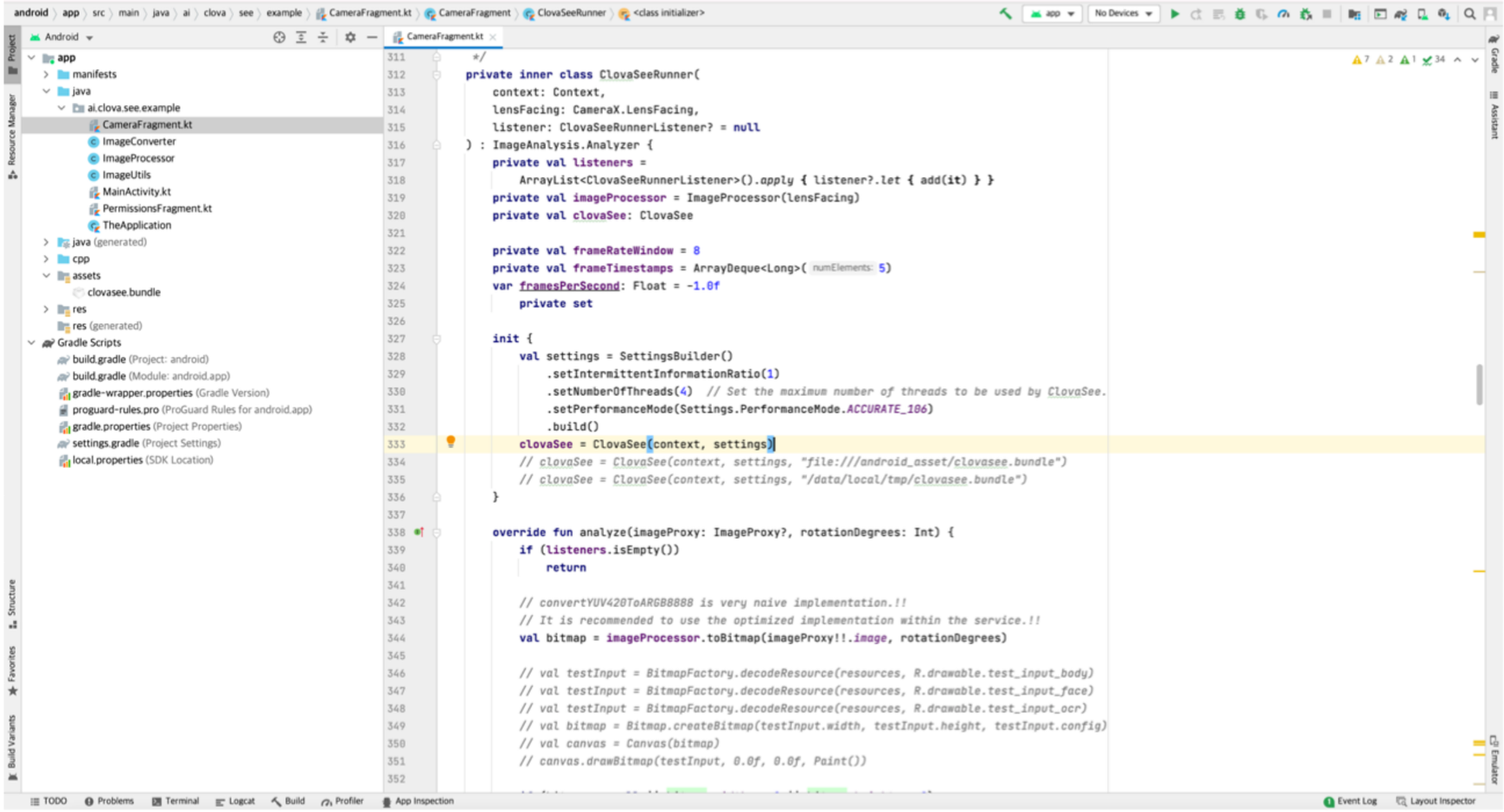
RGB로 변환 하는

코드는 C++로 작성

# 2.3.3 clova-face-kit source code

DEVIEW 2021

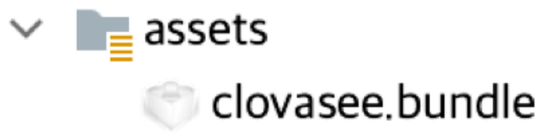
## Camera Fragment



# 2.3.3 clova-face-kit source code

DEVIEW 2021

```
init {
    val settings = SettingsBuilder()
        .setIntermittentInformationRatio(1)
        .setNumberOfThreads(4)  // Set the maximum number of threads to be used by ClovaSee.
        .setPerformanceMode(Settings.PerformanceMode.ACCURATE_106)
        .build()
    clovaSee = ClovaSee(context, settings)
    // clovaSee = ClovaSee(context, settings, "file:///android_asset/clovasee.bundle")
    // clovaSee = ClovaSee(context, settings, "/data/local/tmp/clovasee.bundle")
}
```



# 2.3.3 clova-face-kit source code

```kotlin
/**
 * ClovaSeeController
 */
interface ClovaSeeController {
    enum class RunType { FACE, BODY, OCR }

    fun isBypassed(): Boolean
    fun setBypassed(set: Boolean)
    fun getRunType(): RunType
    fun setRunType(set: RunType)
    fun isLogging(): Boolean
    fun setIsLogging(isLogging: Boolean)
}
```

FACE, BODY,OCR

총 3가지 타입이 있어요

# 2.3.3 clova-face-kit source code

```
ClovaSeeController.RunType.BODY -> {
    Pair(
        ClovaSeeRunResult(
            ai.clova.see.face.Result(emptyArray()),
            clovaSee.run(bitmap, ai.clova.see.body.OptionsBuilder().build()),
            ai.clova.see.ocr.Result(Document())
        ),
        clovaSee.getMeasureResult()
    )
}
ClovaSeeController.RunType.OCR -> {
    Pair(
        ClovaSeeRunResult(
            ai.clova.see.face.Result(emptyArray()),
            ai.clova.see.body.Result(Segment( size: 0)),
            clovaSee.run(bitmap, ai.clova.see.ocr.OptionsBuilder().build())
        ),
        clovaSee.getMeasureResult()
    )
}
else -> {
    val faceOptions = ai.clova.see.face.OptionsBuilder()
        .setBoundingBoxThreshold(0.7f)
        .setInformationToObtain(ai.clova.see.face.Options.CONTOURS or
                                ai.clova.see.face.Options.MASKS or
                                ai.clova.see.face.Options.EULER_ANGLES or
                                ai.clova.see.face.Options.TRACKING_IDS)
        .setResizeThreshold(320)
        .setMinimumBoundingBoxSize(0.1f)
        .build()
    Pair(
        ClovaSeeRunResult(
            clovaSee.run(bitmap, faceOptions),
            ai.clova.see.body.Result(Segment( size: 0)),
            ai.clova.see.ocr.Result(Document())
        ),
        clovaSee.getMeasureResult()
    )
}
```

사용하려는 type의 option을 설정해서

ClovaSee.run() 을 실행

# 2.3.3 clova-face-kit source code

DEVIEW 2021

```
val faceOptions = ai.clova.see.face.OptionsBuilder()
    .setBoundingBoxThreshold(0.7f)
    .setInformationToObtain(ai.clova.see.face.Options.CONTOURS or
                            ai.clova.see.face.Options.MASKS or
                            ai.clova.see.face.Options.EULER_ANGLES or
                            ai.clova.see.face.Options.TRACKING_IDS)
    .setResizeThreshold(320)
    .setMinimumBoundingBoxSize(0.1f)
    .build()
Pair(
    ClovaSeeRunResult(
        clovaSee.run(bitmap, faceOptions),
        ai.clova.see.body.Result(Segment( size: 0)),
        ai.clova.see.ocr.Result(Document())
    ),
    clovaSee.getMeasureResult()
)
```

# 2.3.3 clova-face-kit source code

DEVIEW 2021

```
public final class Options public constructor(boundingBoxThreshold: kotlin.Float /* = compiled code
    public companion object {
        public final val ALL: kotlin.Int /* compiled code */

        public final val BOUNDING_BOXES: kotlin.Int /* compiled code */

        public final val CONTOURS: kotlin.Int /* compiled code */

        public final val DEFAULT_BOUNDING_BOX_THRESHOLD: kotlin.Float /* compiled code */

        public final val DEFAULT_MINIMUM_BOUNDING_BOX_SIZE: kotlin.Float /* compiled code */

        public final val DEFAULT_RESIZE_THRESHOLD: kotlin.Int /* compiled code */

        public final val EULER_ANGLES: kotlin.Int /* compiled code */

        public final val FEATURES: kotlin.Int /* compiled code */

        public final val MASKS: kotlin.Int /* compiled code */

        public final val MOJOS: kotlin.Int /* compiled code */

        public final val SPOOFS: kotlin.Int /* compiled code */
```

# 2.3.3 clova-face-kit source code

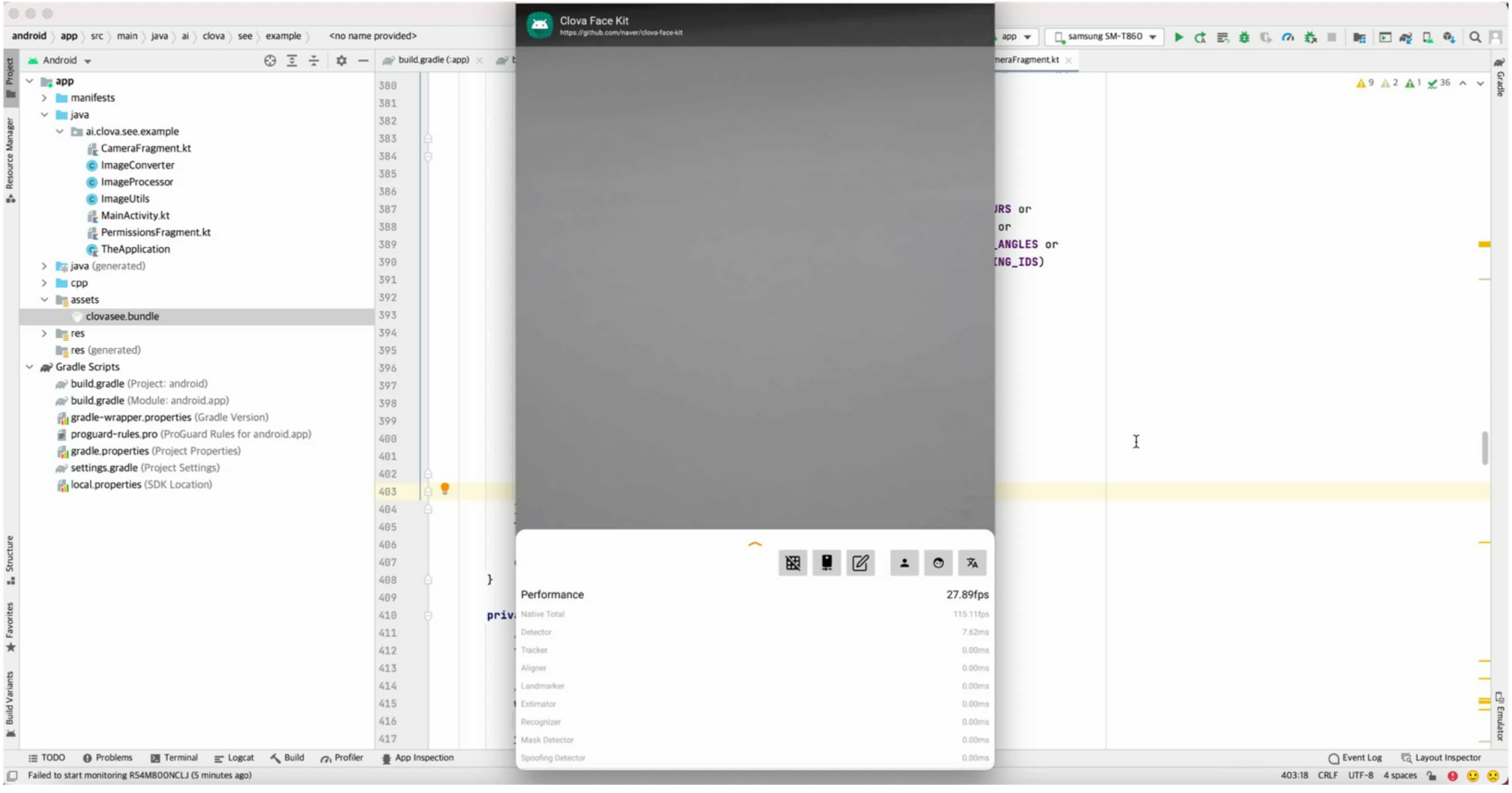
DEVIEW 2021

```
public final class Face public constructor(boundingBox: android.graphics.Rect, contour: ai.clova.see
    public companion object {
        @kotlin.jvm.JvmStatic public final fun getCosineSimilarity(face1: ai.clova.see.Face, face2:

        @kotlin.jvm.JvmStatic public final fun isSame(face1: ai.clova.see.Face, face2: ai.clova.see
    }

    public final val boundingBox: android.graphics.Rect /* compiled code */

    public final val contour: ai.clova.see.Contour /* compiled code */

    public final val eulerAngle: ai.clova.see.EulerAngle /* compiled code */

    public final val feature: ai.clova.see.Feature /* = kotlin.FloatArray */ /* compiled code */

    public final val mask: ai.clova.see.Mask /* = kotlin.Boolean */ /* compiled code */

    public final val mojo: ai.clova.see.Mojo /* = kotlin.ByteArray */ /* compiled code */

    public final val spoof: ai.clova.see.Spoof /* = kotlin.Boolean */ /* compiled code */

    public final val trackingID: ai.clova.see.TrackingID /* = kotlin.Int */ /* compiled code */
}
```

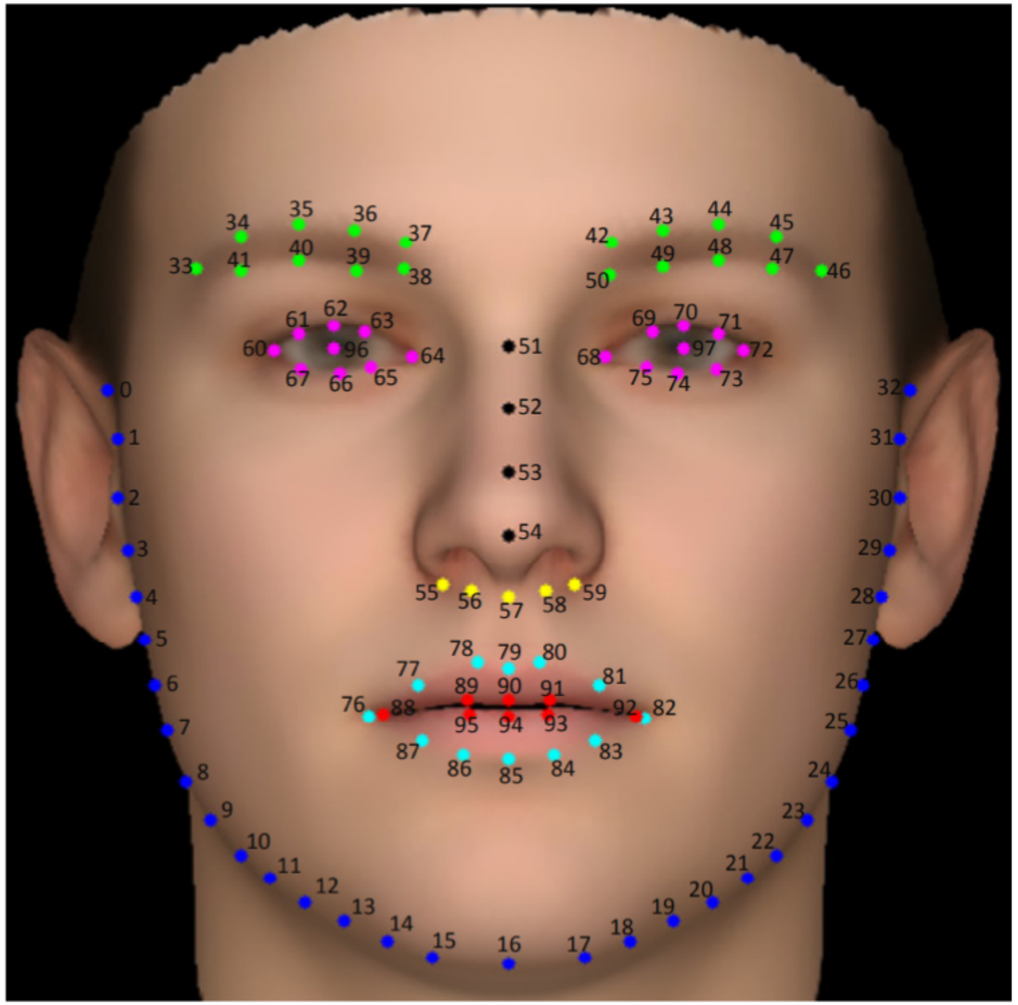
Face 결과 값

얼굴과 관련된 다양한 정보를 return 값으로 받을 수 있다

# 2.3.4 clova-face-kit 실행



# 2.3.5 clova-face-kit 응용




왼쪽 기준

눈 윗줄 (61~63) 과

눈 아랫줄 (65~67) 사이의 거리를

계산해서 눈 뜨고있는지 여부 계산

# 2.3.5 clova-face-kit 응용

DEVIEW 2021

```
/**
 * 왼쪽 눈을 뜨고 있는지 체크
 */
fun getLeftEyeOpen(face: Face): Double {
    face.contour.points.let { it: Points
        return (distance(
            it[61],
            it[67]
        ) + distance(
            it[62],
            it[66]
        ) + distance(
            it[63],
            it[65]
        ))
    }
}
```

```
/**
 * 오른쪽 눈을 뜨고 있는지 체크
 */
fun getRightEyeOpen(face: Face): Double {
    face.contour.points.let { it: Points
        return (distance(
            it[69],
            it[75]
        ) + distance(
            it[70],
            it[74]
        ) + distance(
            it[71],
            it[73]
        ))
    }
}
```

# 2.3.5 clova-face-kit 응용

DEVIEW 2021

```
public final class EulerAngle public constructor(x: kotlin.Float, y: kotlin.Float, z: kotlin.Float) {
    public final val pitch: kotlin.Float /* compiled code */

    public final val roll: kotlin.Float /* compiled code */

    public final val x: kotlin.Float /* compiled code */

    public final val y: kotlin.Float /* compiled code */

    public final val yaw: kotlin.Float /* compiled code */

    public final val z: kotlin.Float /* compiled code */
}
```

EulerAngle을 통해 얼굴의 방향을 알 수 있어요

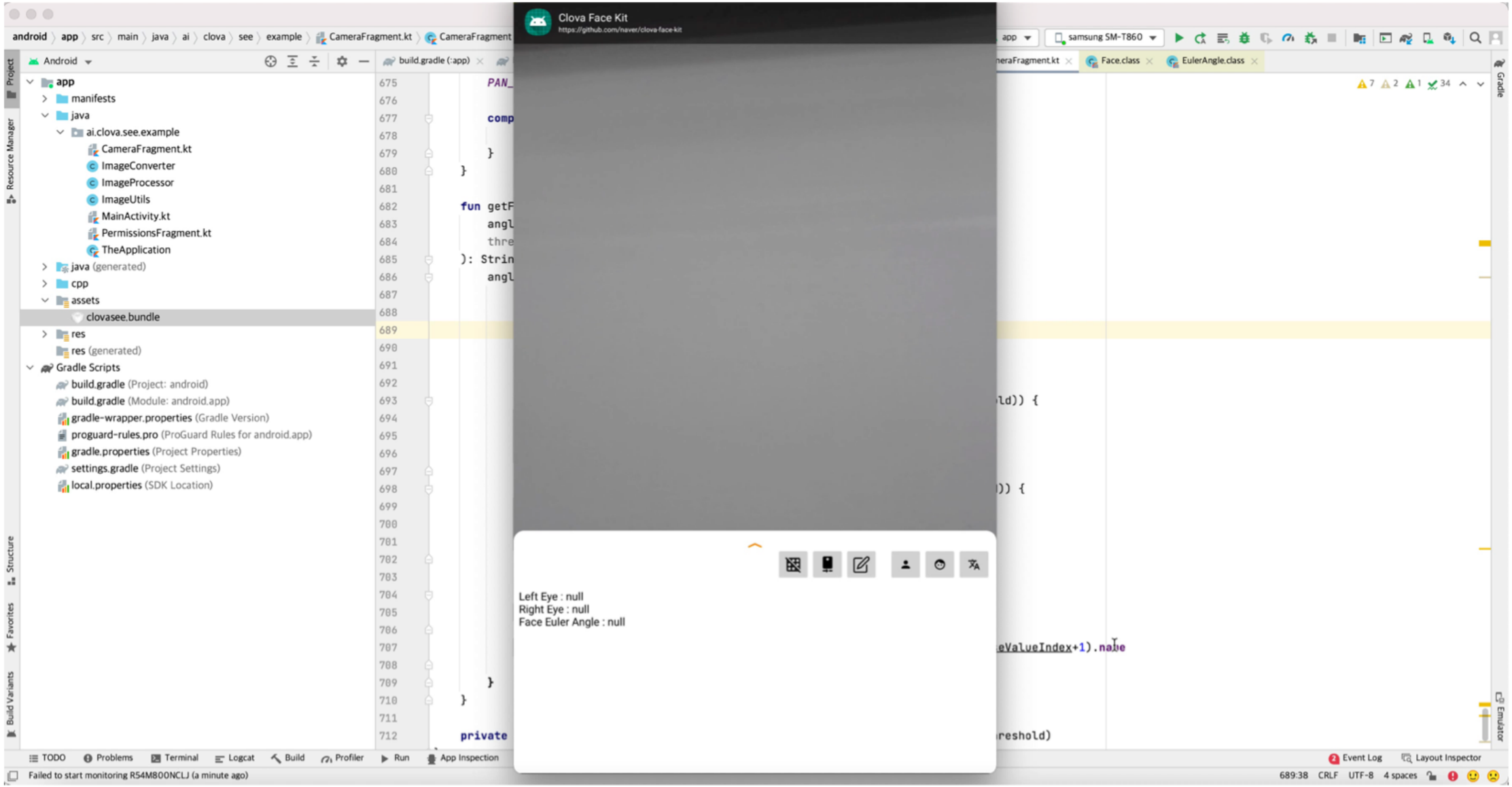
# 2.3.5 clova-face-kit 응용

DEVIEW 2021

```kotlin
/**
 * 얼굴 각도 체크
 */
enum class FacePose {
    FRONT,
    DOWN,
    UP,
    ROLL_LEFT,
    ROLL_RIGHT,
    PAN_RIGHT,
    PAN_LEFT
}
```

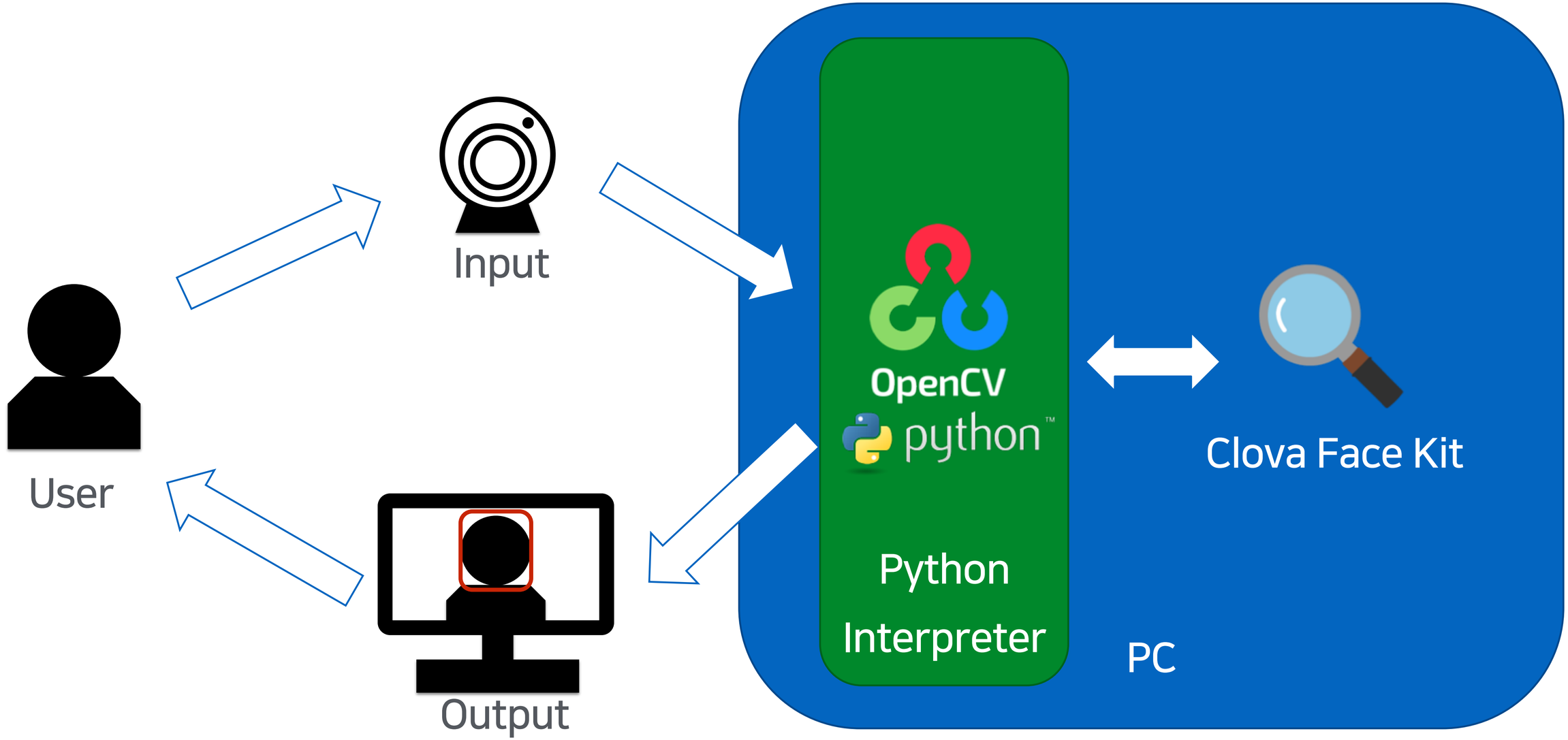
```kotlin
fun getFacePose(
    angle: Triple<Float, Float, Float>,
): String {
    angle.let { (pitch, roll, yaw) ->
        var value = pitch
        var valueThreshold = 8.0f
        val rollThreshold = 12.0f
        val yawThreshold = 12.0f
        var facePoseValueIndex = FacePose.DOWN.ordinal

        if (getPoseRatio(value, valueThreshold) < getPoseRatio(roll, rollThreshold)) {
            value = roll
            valueThreshold = rollThreshold
            facePoseValueIndex = FacePose.ROLL_RIGHT.ordinal
        }
        if (getPoseRatio(value, valueThreshold) < getPoseRatio(yaw, yawThreshold)) {
            value = yaw
            valueThreshold = yawThreshold
            facePoseValueIndex = FacePose.PAN_LEFT.ordinal
        }

        return if (getPoseRatio(value, valueThreshold) < 1.0) {
            FacePose.FRONT.name
        } else {
            FacePose.getFromValue(if (value < 0) facePoseValueIndex else facePoseValueIndex+1).name
        }
    }
}
```

# 2.3.6 clova-face-kit 응용 코드 실행



# 2.4. Python Hands-On

# 2.4. Python Hands-On



# 2.4. Python Hands-On

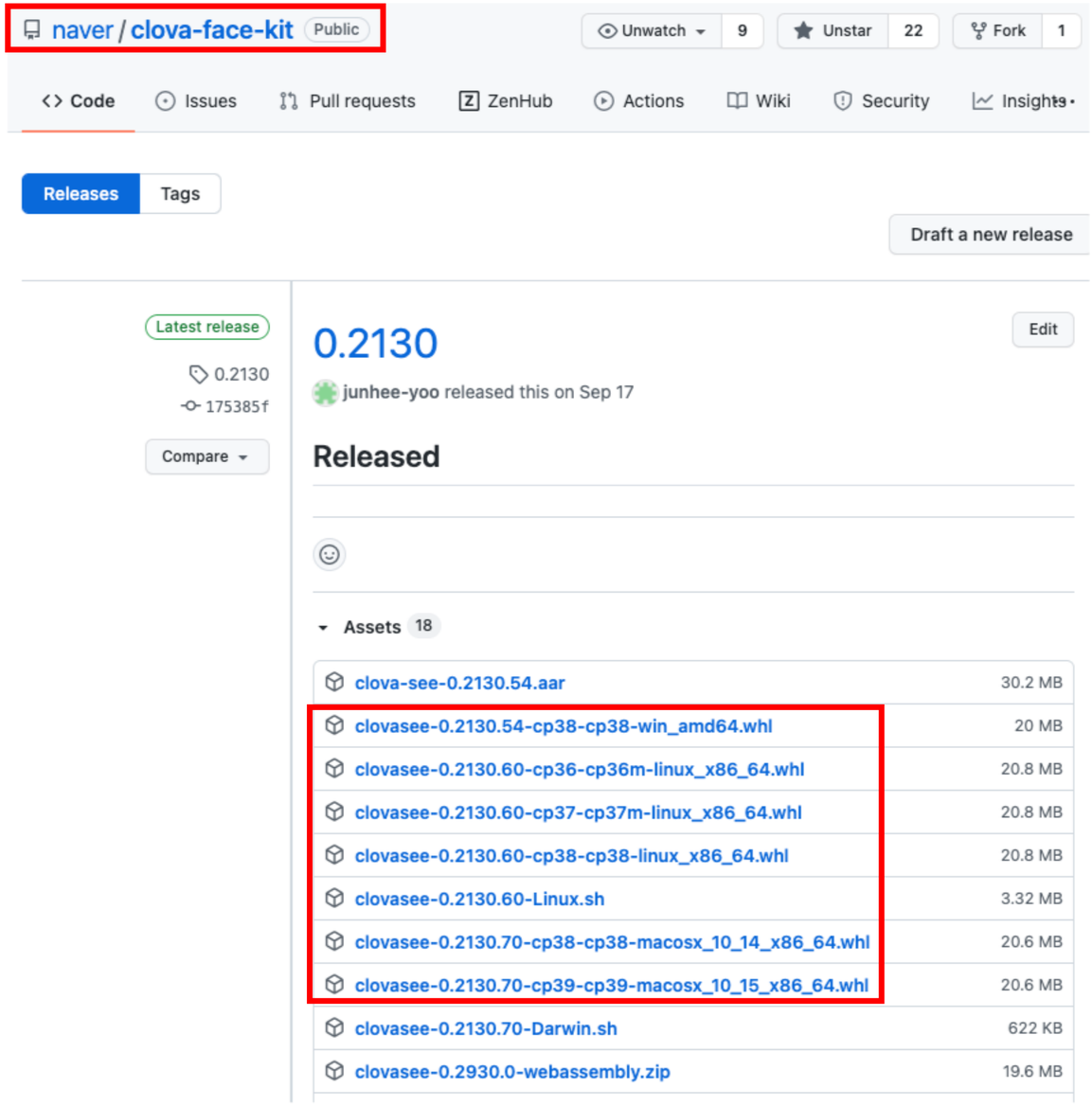
DEVIEW 2021

## Preview

# 2.4. Python Hands-On

DEVIEW 2021

## Download



# 2.4. Python Hands-On

DEVIEW 2021

Download

https://commons.wikimedia.org/wiki/File:Guy_fawkes_mask_by_nacreouss-d462juf.png

-> mask.png

# 2.4. Python Hands-On

DEVIEW 2021

## Environments

```
➜  deview_2021 pipenv shell
Creating a virtualenv for this project...
Pipfile: /Users/jhyoo/workspace/area51/deview_2021/Pipfile
Using /usr/local/bin/python3.9 (3.9.6) to create virtualenv...
⠸ Creating virtual environment...created virtual environment CPython3.9.6.final.0-64 in 598m
s
  creator CPython3Posix(dest=/Users/jhyoo/.local/share/virtualenvs/deview_2021-FYBE5rkn, cle
ar=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
app_data_dir=/Users/jhyoo/Library/Application Support/virtualenv)
    added seed packages: clovasee==0.2130.70, numpy==1.21.2, opencv_python==4.5.3.56, pip==2
1.2.4, pybind11==2.7.1, setuptools==57.4.0, setuptools==58.0.4, wheel==0.37.0
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator
,XonshActivator

✔ Successfully created virtual environment!
Virtualenv location: /Users/jhyoo/.local/share/virtualenvs/deview_2021-FYBE5rkn
Launching subshell in virtual environment...
 . /Users/jhyoo/.local/share/virtualenvs/deview_2021-FYBE5rkn/bin/activate
➜  deview_2021  . /Users/jhyoo/.local/share/virtualenvs/deview_2021-FYBE5rkn/bin/activate
(deview_2021) ➜  deview_2021 pip install opencv-python
```

# 2.4. Python Hands-On

DEVIEW 2021

## Imports

```python
import cv2

from clovasee import ClovaSee
from clovasee import Frame
from clovasee import Settings
from clovasee import SettingsBuilder
from clovasee import FaceOptions
from clovasee import FaceOptionsBuilder
```

# 2.4. Python Hands-On

DEVIEW 2021

## Main function

(setup)

```
def main():
  capture = cv2.VideoCapture(1)
  capture.set(cv2.CAP_PROP_FORMAT, cv2.CV_8UC3)

  settings = SettingsBuilder() \
      .set_number_of_threads(4) \
      .set_performance_mode(Settings.PerformanceMode.kAccurate106) \
      .build()
  clova_see = ClovaSee(settings)

  options = FaceOptionsBuilder() \
              .set_bounding_box_threshold(0.7) \
              .set_information_to_obtain(
                FaceOptions.kBoundingBoxes
                | FaceOptions.kContours \
                | FaceOptions.kEulerAngles \
              ) \
              .set_minimum_bounding_box_size(0.1) \
              .set_resize_threshold(320) \
              .build()

  mask = cv2.imread('mask.png', cv2.IMREAD_UNCHANGED)
```

# 2.4. Python Hands-On

DEVIEW 2021

## Main function

(loop)

```
while True:
  ret, snapshot = capture.read()
  if not ret:
    continue

  snapshot = cv2.flip(snapshot, 1)

  (height, width, _) = snapshot.shape
  frame = Frame(snapshot,
                width,
                height,
                Frame.Format.kBGR_888)
  faces = clova_see.run(frame, options).faces()

  for face in faces:
    draw_bounding_box(snapshot, face)
    draw_mask(snapshot, face, mask)
    draw_contour(snapshot, face)
    draw_euler_angle(snapshot, face)

  cv2.imshow("deview 2021", snapshot)

  key = cv2.waitKey(1)
  if key == ord('q'):
    break

capture.release()
cv2.destroyAllWindows()
```

# 2.4. Python Hands-On

DEVIEW 2021

## Draw functions

(bounding box, euler angle, contour)

```python
def draw_bounding_box(canvas, face):
  cv2.rectangle(canvas,
                (face.bounding_box().origin().x,
                 face.bounding_box().origin().y),
                (face.bounding_box().right_bottom().x,
                face.bounding_box().right_bottom().y),
                (0, 0, 255), 2)

def draw_euler_angle(canvas, face):
  cv2.putText(canvas,
              (f"x={face.euler_angle().x:.2f} "
               f"y={face.euler_angle().y:.2f} "
               f"z={face.euler_angle().z:.2f}"),
              (face.bounding_box().x, face.bounding_box().right_bottom().y+18),
              cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255))

def draw_contour(canvas, face):
  for point in face.contour().points:
    cv2.circle(canvas, (point.x, point.y), 1, (0, 255, 0))
```

# 2.4. Python Hands-On

DEVIEW 2021

## Draw functions

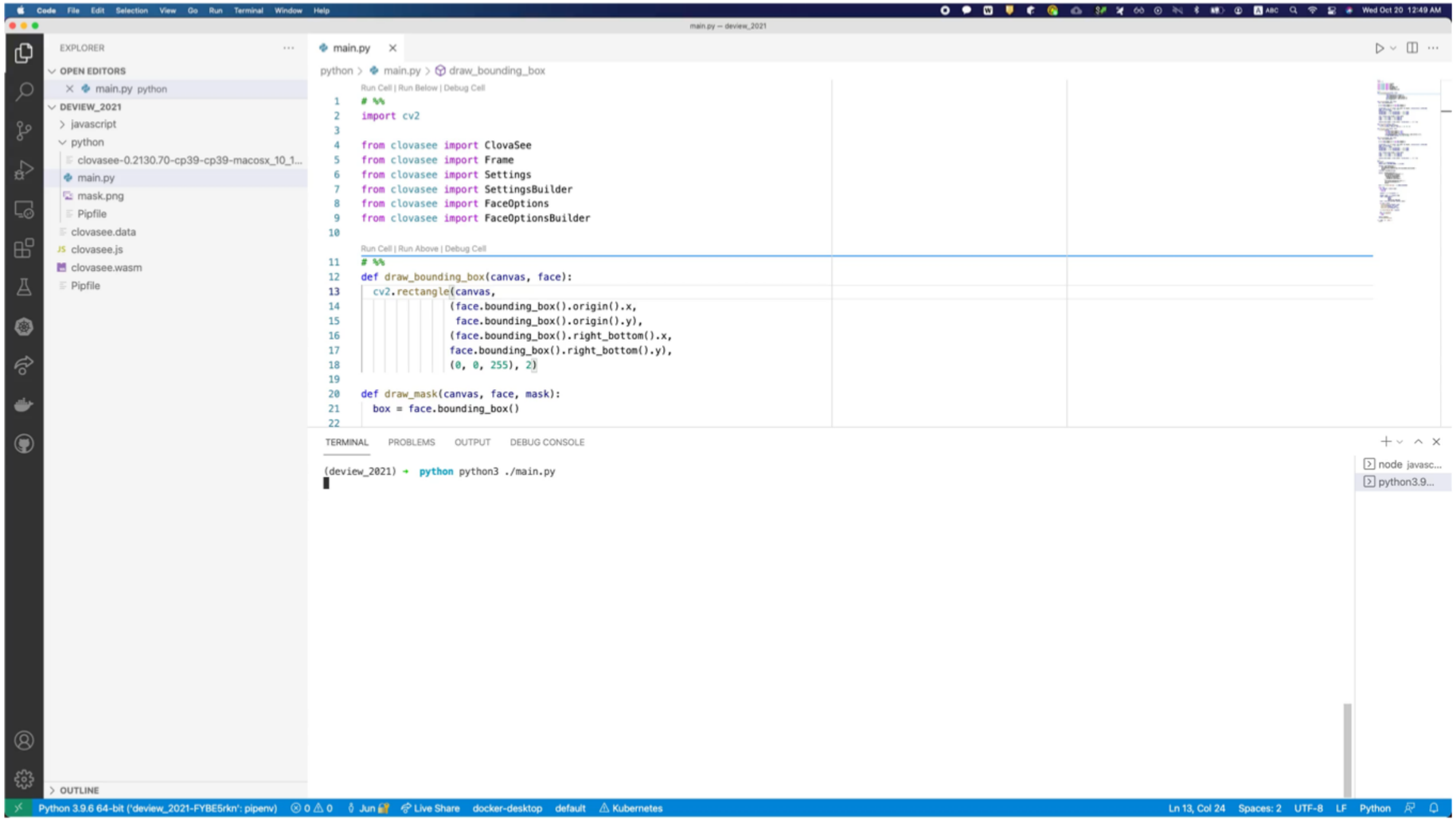
(mask)

```python
def draw_mask(canvas, face, mask):
  box = face.bounding_box()

  y = 0 if box.origin().y < 0 else box.origin().y
  x = 0 if box.origin().x < 0 else box.origin().x

  resized_mask = cv2.resize(mask, (box.width, box.height), interpolation=cv2.INTER_AREA)
  alpha = resized_mask[:, :, 3] / 255.0
  alpha_inv = 1.0 - alpha
  resized_mask[:, :, 0] = resized_mask[:, :, 0] * alpha
  resized_mask[:, :, 1] = resized_mask[:, :, 1] * alpha
  resized_mask[:, :, 2] = resized_mask[:, :, 2] * alpha

  crop = canvas[y:y+box.height, x:x+box.width]
  crop[:, :, 0] = crop[:, :, 0] * alpha_inv
  crop[:, :, 1] = crop[:, :, 1] * alpha_inv
  crop[:, :, 2] = crop[:, :, 2] * alpha_inv

  canvas[y:y+box.height, x:x+box.width] = resized_mask[:, :, :3] + crop
```

# 2.4. Python Hands-On

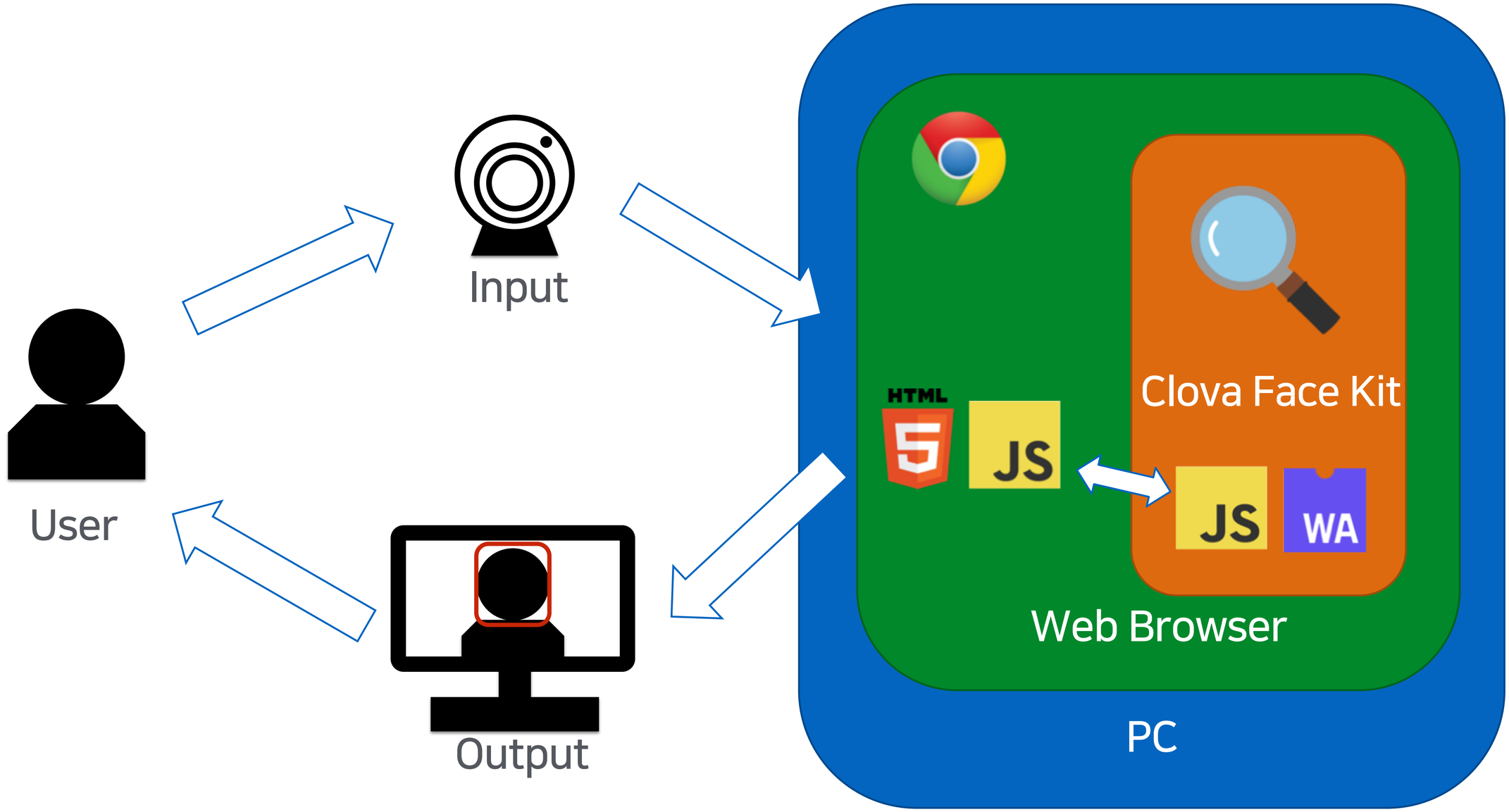
DEVIEW 2021

Result



# 2.5. JavaScript Hands-On

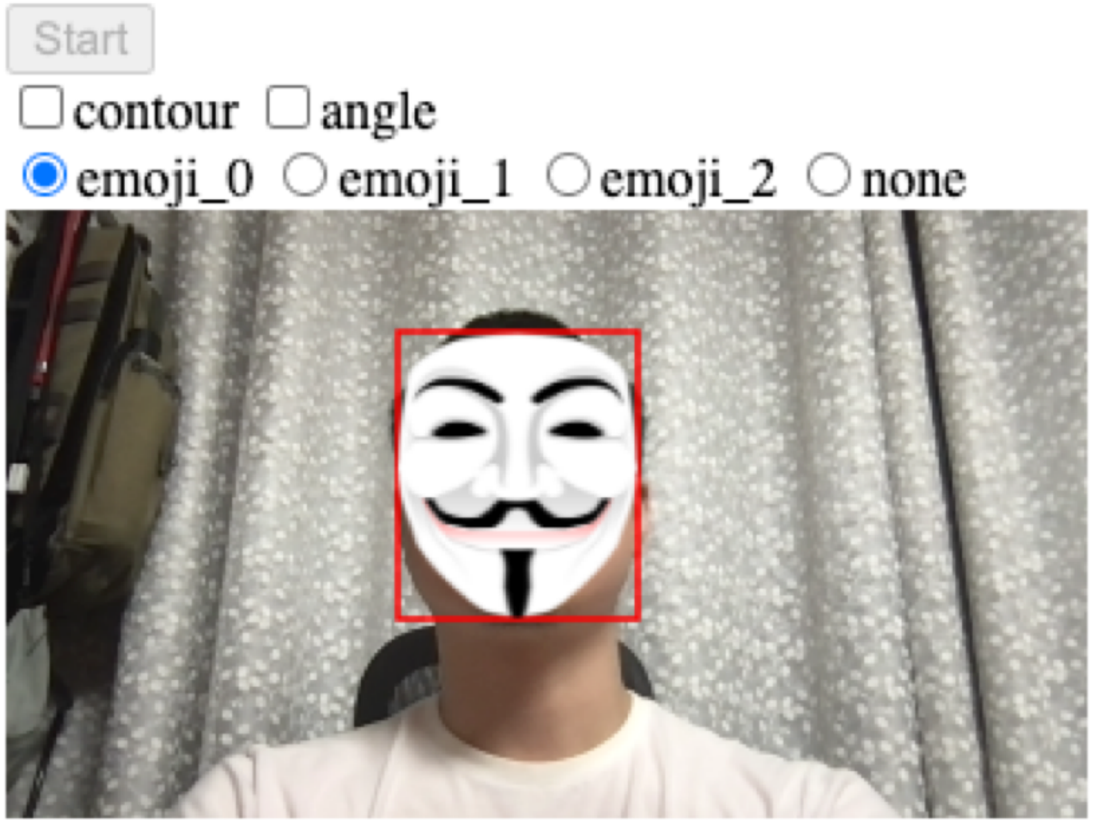
# 2.5. JavaScript Hands-On



# 2.5. JavaScript Hands-On

DEVIEW 2021

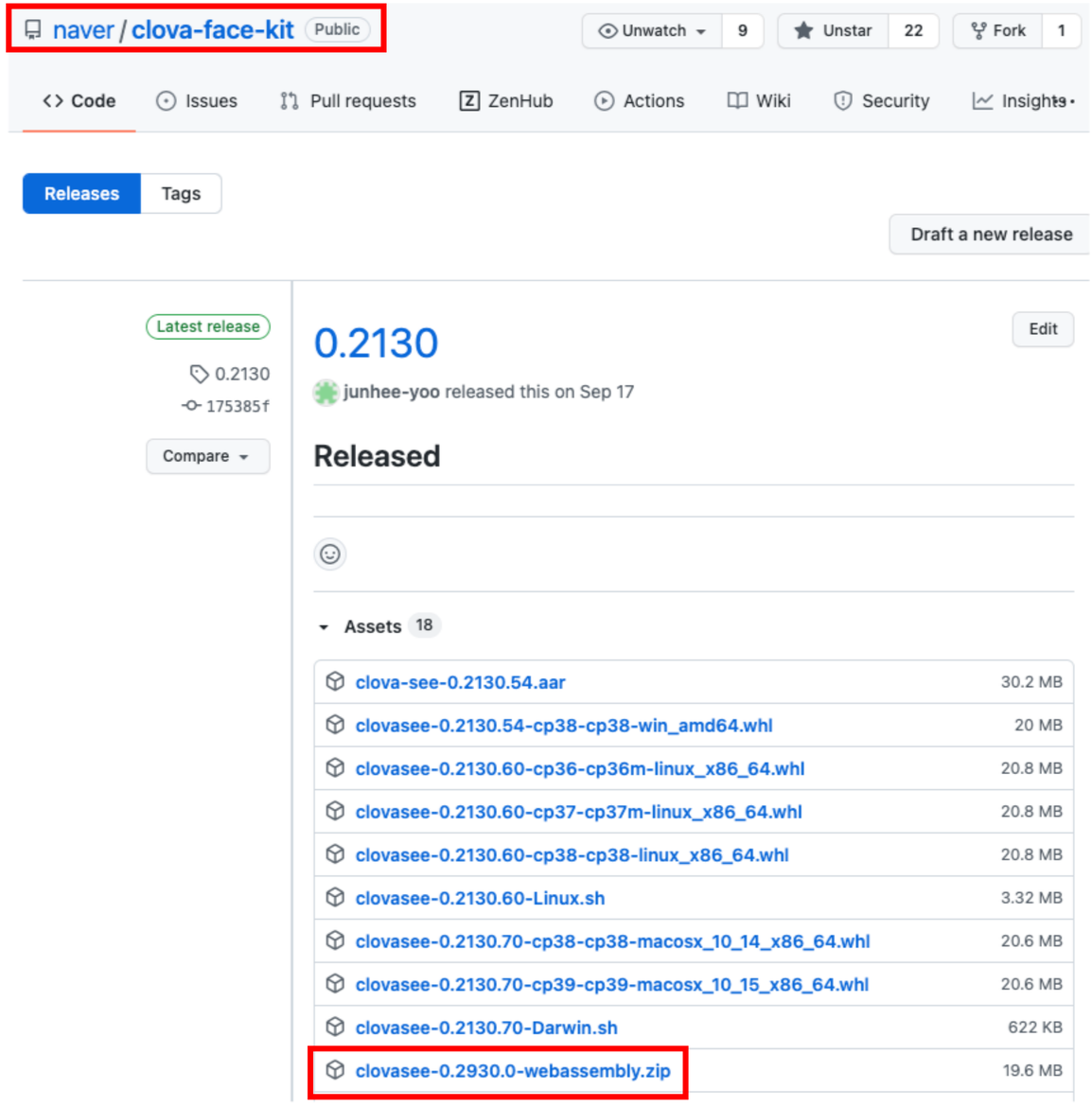
## Preview



# 2.5. JavaScript Hands-On

DEVIEW 2021

## Download



# 2.5. JavaScript Hands-On

DEVIEW 2021

## Environments

```
➜  deview_2021 unzip ~/Downloads/clovasee-0.2930.0-webassembly.zip
Archive:  /Users/jhyoo/Downloads/clovasee-0.2930.0-webassembly.zip
  inflating: clovasee.data
  inflating: clovasee.js
  inflating: clovasee.wasm
➜  deview_2021 npx serve .
Need to install the following packages:
  serve
Ok to proceed? (y) y
```

```
Serving!

- Local:            http://localhost:5000
- On Your Network:  http://192.168.55.42:5000

Copied local address to clipboard!
```

# 2.5. JavaScript Hands-On

DEVIEW 2021

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Clova See for DEVIEW 2021</title>
  <script type="application/javascript" src="clovasee.js"></script>
  <script type="application/javascript" src="capture.js"></script>
</head>
<body>
  <div class="contentarea">
    <div class="content">
      <video id="video" hidden=true>Video stream not available.</video>
      <button id="startbutton">Start</button>
      <div class="features">
        <label><input type="checkbox" id="bounding_box" name="bounding_box">bounding box</label>
        <label><input type="checkbox" id="contour" name="contour">contour</label>
        <label><input type="checkbox" id="angle" name="angle">angle</label>
      </div>
      <div class="selector">
        <label><input type="radio" id="emoji_0" name="selector" value="0"/>emoji_0</label>
        <label><input type="radio" id="emoji_1" name="selector" value="1"/>emoji_1</label>
        <label><input type="radio" id="emoji_2" name="selector" value="2"/>emoji_2</label>
        <label><input type="radio" id="emoji_3" name="selector" value="3" checked="checked"/>none</label>
      </div>
    </div>
    <div class="output">
      <canvas id="canvas" hidden=true></canvas>
      <img id="photo" alt="The screen capture will appear in this box.">
    </div>
  </div>
</body>
</html>
```

https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(async load)

```
clova().then(
  (instance) => {
    console.log('Wasm Loaded!');
    process(instance);
  }
);
```

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(create instance)

```
function process(clova) {
  // The width and height of the captured photo. We will set the
  // width to the value defined here, but the height will be
  // calculated based on the aspect ratio of the input stream.

  var width = 320;    // We will scale the photo width to this
  var height = 0;     // This will be computed based on the input stream

  // |streaming| indicates whether or not we're currently streaming
  // video from the camera. Obviously, we start at false.

  var streaming = false;

  // The various HTML elements we need to configure or control. These
  // will be set by the startup() function.
  var video = document.getElementById('video');
  var canvas = document.getElementById('canvas');
  var photo = document.getElementById('photo');
  var startbutton = document.getElementById('startbutton');

  let resources = new clova.Resources('clovasee.all.bundle');
  let settings = new clova.SettingsBuilder()
                          .setPerformanceMode(clova.PerformanceMode.ACCURATE106)
                          .build();
  let clovaSee = new clova.ClovaSee(settings, resources);

  var masks = [
    document.createElement("img"),
    document.createElement("img"),
    document.createElement("img"),
  ]
  masks.forEach((item, idx) => {
    item.src = "./mask_" + idx + ".png";
  });
  masks.push(document.createElement("img"));
```

https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(event listener)

```
// Set up our event listener to run the startup process
// once loading is complete.
startbutton.addEventListener('click', function(ev){
  startbutton.disabled = true;
  startup();
  ev.preventDefault();
}, false);
clearphoto();
```

https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(startup)

```
function startup() {
  navigator.mediaDevices.getUserMedia(
    {
      video: {width: {exact: 640}, height: {exact: 360}},
      audio: false
    }
  ).then(function(stream) {
    video.srcObject = stream;
    video.play();
  }).catch(function(err) {
    console.log("An error occurred: " + err);
  });

  video.addEventListener('canplay', function(ev){
    if (!streaming) {
      height = video.videoHeight / (video.videoWidth/width);

      // Firefox currently has a bug where the height can't be read from
      // the video, so we will make assumptions if this happens.

      if (isNaN(height)) {
        height = width / (4/3);
      }

      video.setAttribute('width', width);
      video.setAttribute('height', height);
      canvas.setAttribute('width', width);
      canvas.setAttribute('height', height);
      streaming = true;
    }
  }, false);

  requestAnimationFrame(takepicture);
}
```

https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(takepicture)

```
function takepicture() {
  var context = canvas.getContext('2d');
  if (width && height) {
    canvas.width = width;
    canvas.height = height;
    context.drawImage(video, 0, 0, width, height);

    var imgData = context.getImageData(0, 0, width, height);

    var frame = new clova.Frame(imgData.data,
                                width,
                                height,
                                clova.Format.RGBA_8888);

    var getBoundingBox = document.getElementById('bounding_box').checked;
    var getContour = document.getElementById('contour').checked;
    var getAngle = document.getElementById('angle').checked;

    var flag = 0;
    if(getBoundingBox) {
      flag |= clova.BOUNDING_BOXES;
    }
    if(getContour) {
      flag |= clova.CONTOURS;
    }
    if(getAngle) {
      flag |= clova.EULER_ANGLES;
    }
```

https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

## capture.js

(takepicture)

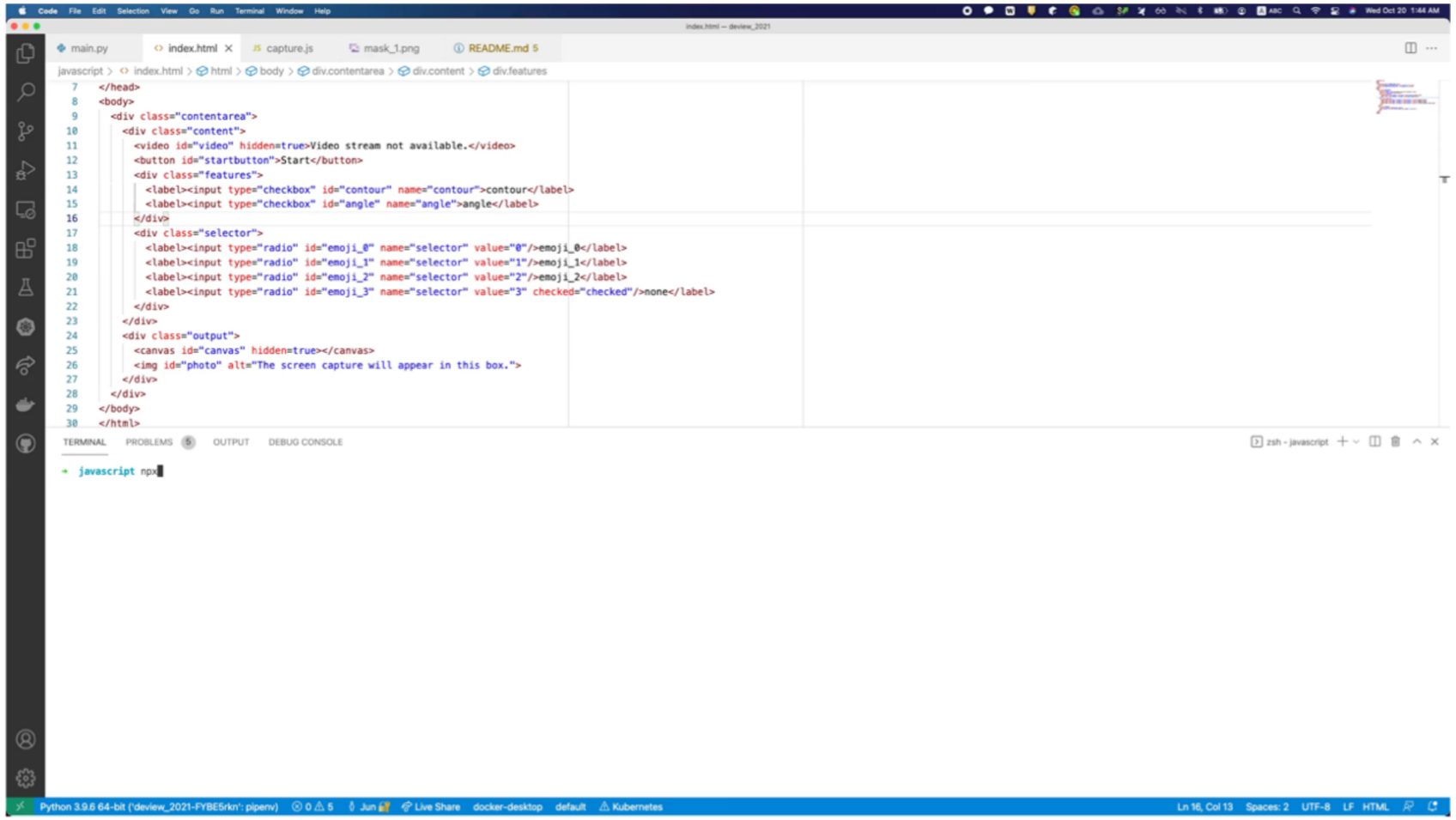
```
    var options = new clova.FaceOptionsBuilder()
                            .setInformationToObtain(flag)
                            .build();
    var faces = clovaSee.runForFace(frame, options).faces();

    context.lineWidth = 1.5;
    context.strokeStyle = 'red';
    var idx = document.querySelector('input[name="selector"]:checked').value;
    for (var i=0; i < faces.size(); ++i) {
      var face = faces.get(i);
      if(getBoundingBox) drawBoundingBox(context, face);
      drawMask(context, face, masks[idx]);
      if(getContour) drawContour(context, face);
      if(getAngle) drawAngle(context, face);
    }

    var data = canvas.toDataURL('image/png');
    photo.setAttribute('src', data);

    options.delete();
    frame.delete();
    faces.delete();

  } else {
    clearphoto();
  }

  requestAnimationFrame(takepicture);
}
```

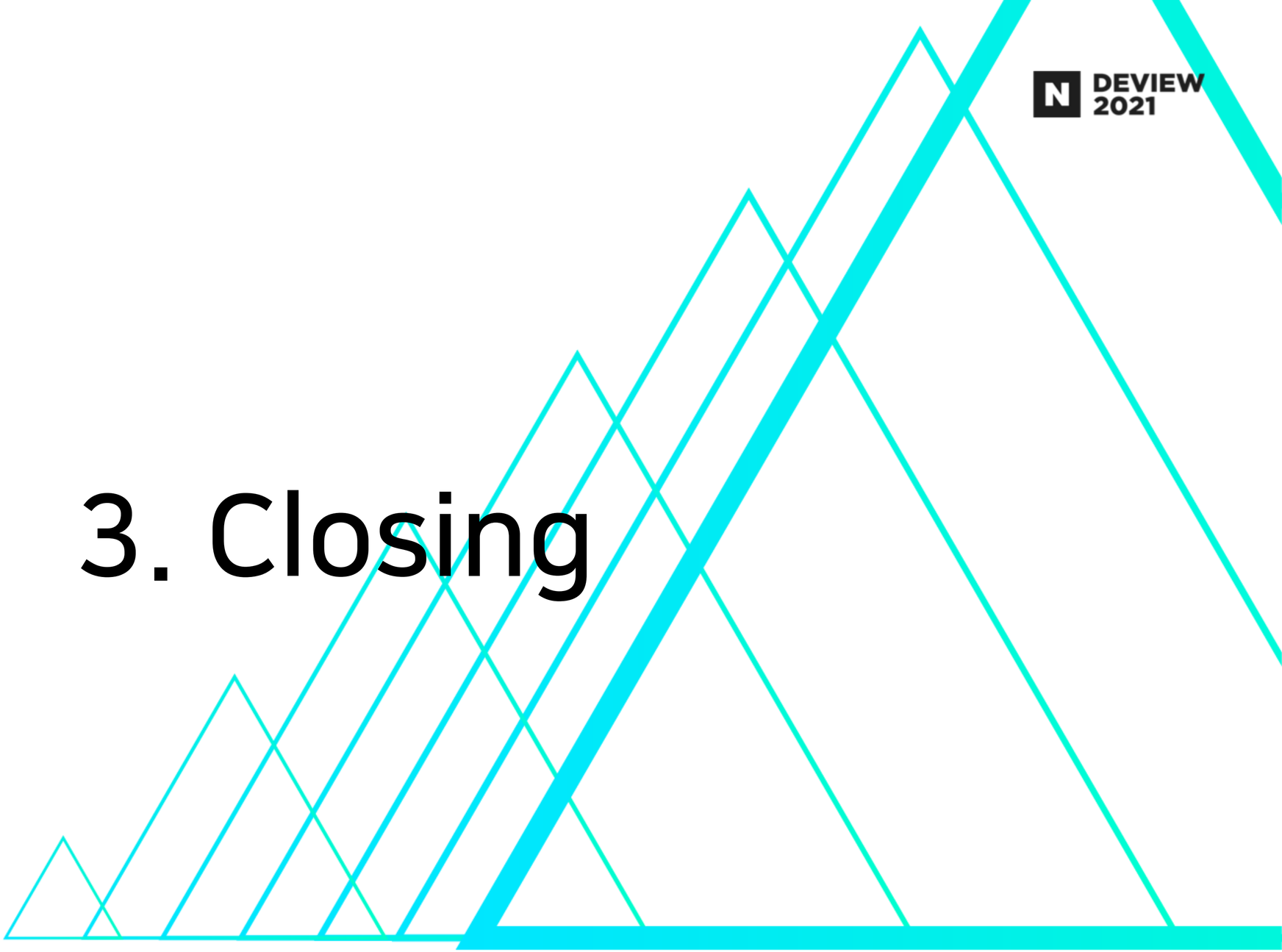
https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Taking_still_photos

# 2.5. JavaScript Hands-On

DEVIEW 2021

Result





# 3. Closing

# 3. 여기까지 오시느라 고생하셨습니다

DEVIEW 2021

당장 사용해보고 싶다면?

- https://github.com/naver/clova-face-kit

# Thank You

dl_clova_partnership@navercorp.com