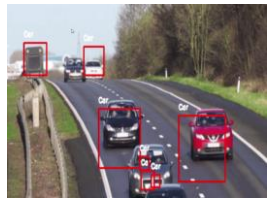
openCV활용

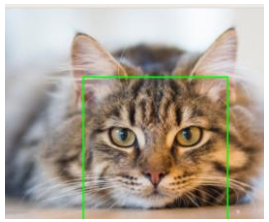
Cars

https://github.com/KrishArul26/Cars_Counting-Detection-using-haar-Cacade



Cats

https://blog.naver.com/chandong83/221484138901



스노우와 같은 합성

https://velog.io/@be1le/cv2-%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5-snow-%EC%95%B1-%EB%A7%8C%EB%93%A4%EC%96%B4-%EB%B3%B4%EA%B8%B0

졸음감지하기

http://www.nefus.kr/2021_Demonstration/Drowsy_Driver/index.html

MediaPipe

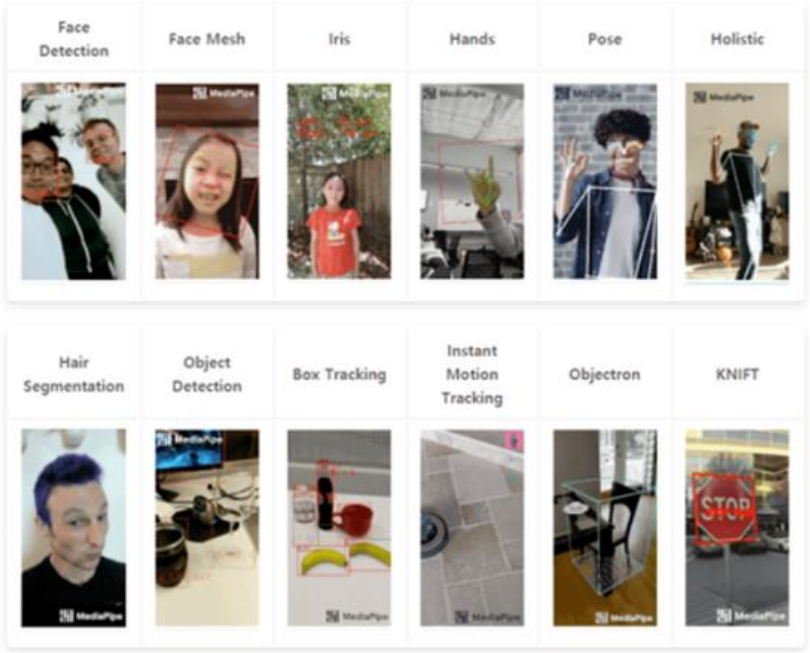
https://developers.google.com/mediapipe/solutions/vision/face_detector

https://github.com/ntu-rris/google-mediapipe/tree/main/code

[1] MediaPipe

MediaPipe란 구글에서 제공하는 AI 프레임워크로서, 비디오형식 데이터를 이용한 다양한 비전 AI 기능을 파이프라인 형태로 손쉽게 사용할 수 있도록 제공된다. AI 모델개발 및 수많은 데이터셋을 이용한 학습도 마친 상태로 제공되므로 라이브러브 불러 사용하듯이 간편하게 호출하여 사용하기만 하면 되는 형태로 비전 AI 기능을 개발할 수 있다.

기본적인 얼굴인식 이외에도 Pose 인식 등 다양한 비전AI 기능들이 제공되는데 사용할 수 있는 비전 AI 솔루션들은 다 다음과 같다.



통상의 C++이나 Python언어 이외도 안드로이드나 iOS 등 모바일 프로그램 개발에 활용할 수 도 있고, JavaScript를 이용하여 Web 페이지 형태로 구현할 수도 있다.

	Android	iOS	C++	Python	JS	Coral
Face Detection	✅	✅	✅	✅	✅	✅
Face Mesh	✅	✅	✅	✅	✅	
Iris	✅	✅	✅			
Hands	✅	✅	✅	✅	✅	
Pose	✅	✅	✅	✅	✅	
Holistic	✅	✅	✅	✅	✅	
Selfie Segmentation	✅	✅	✅	✅	✅	
Hair Segmentation	✅		✅			
Object Detection	✅	✅	✅			✅
Box Tracking	✅	✅	✅			
Instant Motion Tracking	✅					
Objectron	✅		✅	✅	✅	
KNIFT	✅					
AutoFlip			✅			
MediaSequence			✅			
YouTube 8M			✅			

[2] 모듈 정보 확인 및 모듈 개념 잡기

```
import mediapipe as mp
mp_face_detection = mp.solutions.face_detection
```

```
1  !pip show mediapipe
```
✓ 0.7s

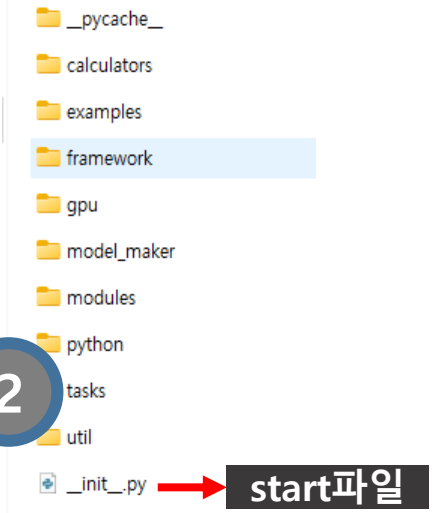
모듈정보확인

```
Name: mediapipe
Version: 0.10.3
Summary: MediaPipe is the simplest way for researchers and developers to build
Home-page: https://github.com/google/mediapipe
Author: The MediaPipe Authors
Author-email: mediapipe@google.com
License: Apache 2.0
Location: C:\Users\user\miniconda3\envs\media\Lib\site-packages
Requires: absl-py, attrs, flatbuffers, matplotlib, numpy, opencv-contrib-pyth
Required-by:
```

모듈위치

1

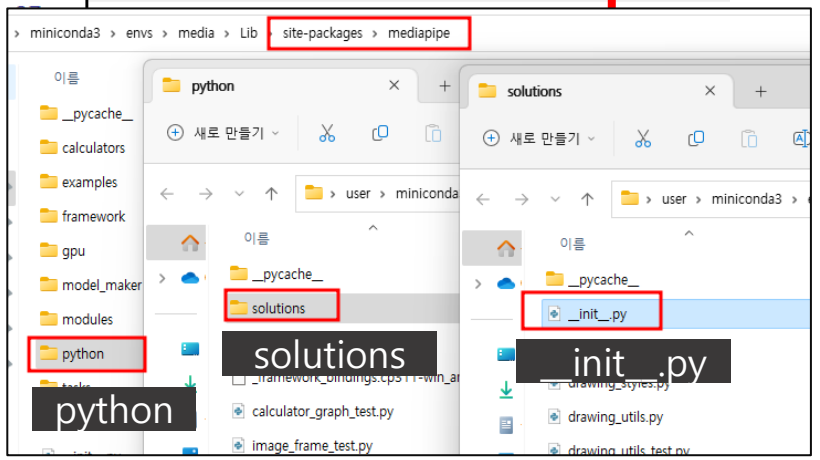
2




```
14
15  from mediapipe.python import *
16  import mediapipe.python.solutions as solutions
17  import mediapipe.tasks.python as tasks
18
19
20  del framework
21  del gpu
22  del modules
23  del python
24  del mediapipe
25  del util
26  __version__ = '0.10.3'
```

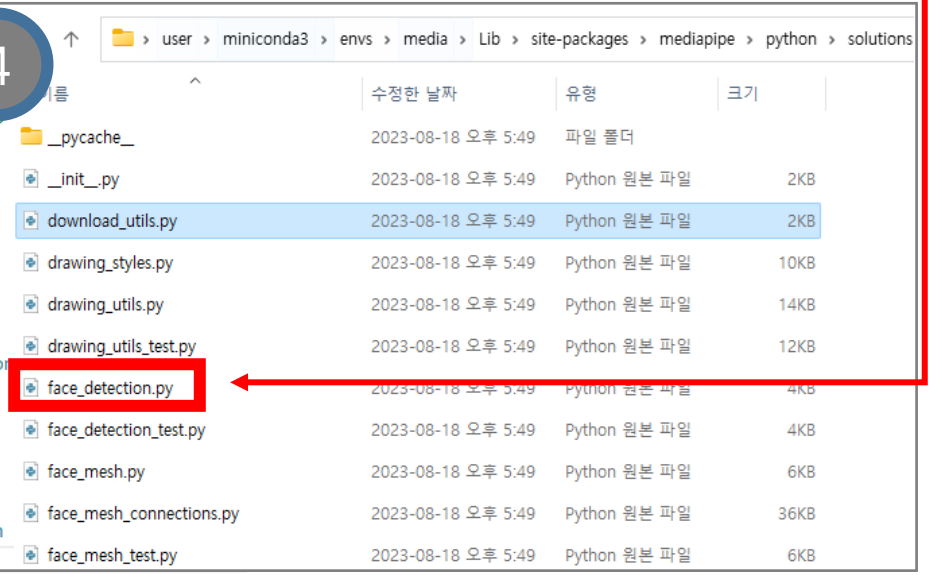
모듈

3




```
9   # Unless required by applicable law or agreed to
10  # distributed under the License is distributed
11  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
12  # See the License for the specific language governing
13  # limitations under the License.
14
15  """MediaPipe Solutions Python API."""
16
17  import mediapipe.python.solutions.drawing_styles
18  import mediapipe.python.solutions.drawing_utils
19  import mediapipe.python.solutions.face_detection
20  import mediapipe.python.solutions.face_mesh
21  import mediapipe.python.solutions.face_mesh_connections
22  import mediapipe.python.solutions.hands
23  import mediapipe.python.solutions.hands_connections
24  import mediapipe.python.solutions.holistic
25  import mediapipe.python.solutions.objectron
26  import mediapipe.python.solutions.pose
27  import mediapipe.python.solutions.selfie_segmentation
28
```

4



[3] face_detection.py : 반드시 확인하지 않아도 됨

```python
class FaceKeyPoint(enum.IntEnum):
  """The enum type of the six face detection key points."""
  RIGHT_EYE = 0
  LEFT_EYE = 1
  NOSE_TIP = 2
  MOUTH_CENTER = 3
  RIGHT_EAR_TRAGION = 4
  LEFT_EAR_TRAGION = 5


class FaceDetection(SolutionBase):
  """MediaPipe Face Detection.

  MediaPipe Face Detection processes an RGB image and returns a list of the
  detected face location data.

  Please refer to
  https://solutions.mediapipe.dev/face_detection#python-solution-api
  for usage examples.
  """

  def __init__(self, min_detection_confidence=0.5, model_selection=0):
    """Initializes a MediaPipe Face Detection object.

    Args:
      min_detection_confidence: Minimum confidence value ([0.0, 1.0]) for face
        detection to be considered successful. See details in
        https://solutions.mediapipe.dev/face_detection#min_detection_confidence.
      model_selection: 0 or 1. 0 to select a short-range model that works
        best for faces within 2 meters from the camera, and 1 for a full-range
        model best for faces within 5 meters. See details in
        https://solutions.mediapipe.dev/face_detection#model_selection.
    """

    binary_graph_path = _FULL_RANGE_GRAPH_FILE_PATH if model_selection == 1 else _SHORT_RANGE_GRAPH_FILE_PATH

    super().__init__(
        binary_graph_path=binary_graph_path,
        graph_options=self.create_graph_options(
            face_detection_pb2.FaceDetectionOptions(), {
                'min_score_thresh': min_detection_confidence,
            }),
        outputs=['detections'])
```

Face모델에서 출력되는 6개의 좌표값

상속클래스, super로 이 상속클래스 함수사용

설명서

1이면 Full, 0이면 short

Super(). ㅁ 상속받은 class의 함수를 사용

MODEL_SELECTION(모델 선택)

모델 인덱스는 0 또는 1입니다.

0을 사용하면 카메라 2m 이내의 **부분적 모델** 촬영에 적합하고,

1은 5m 이내에서 **전신 모델**을 촬영하는데 적합합니다.

지정하지 않을 경우의 **기본값**은 0입니다.

MIN_DETECTION_CONFIDENCE(최소 감지 신뢰값)

검출에 성공한 것으로 간주할 얼굴의 검출 모델의 신뢰값은([0.0, 1.0])입니다.

기본값은 0.5입니다.

Model_selection=0

Model_selection=1

[4] face_detection모듈을 이용한 코드

참고) https://puleugo.tistory.com/4

```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt

mp_face_detection=mp.solutions.face_detection
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=['./img/short.jpg']

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=1, min_detection_confidence=0.5) as face_detection:

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        results=face_detection.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
        print('#'*100)

        print(results.detections)

```


```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt

mp_face_detection=mp.solutions.face_detection
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=['./img/short.jpg']

with mp_face_d

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=0, min_detection_confidence=0.5) as face_detection:

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        results=face_detection.process(cv2.cvtColor(image, cv2.COLOR_BGR2RG
```

0으로 하면 score가 높아짐

```
####################################
[label_id: 0
score: 0.9207152
location_data {
  format: RELATIVE_BOUNDING_
  relative_bounding_box {
    xmin: 0.17662388
    ymin: 0.35764927
    width: 0.74172145
```

```
[label_id: 0
 score: 0.82186544
 location_data {
   format: RELATIVE_BOUNDING_BOX
   relative_bounding_box {
     xmin: 0.19477129
     ymin: 0.16733167
     width: 0.7221874
     height: 0.24313098
   }
   relative_keypoints {
     x: 0.39192796
     y: 0.2477236
   }
   relative_keypoints {
     x: 0.7193618
     y: 0.24726191
   }
   relative_keypoints {
     x: 0.5533507
     y: 0.30088544
   }
   relative_keypoints {
     x: 0.554466
     y: 0.34839734
   }
   relative_keypoints {
     x: 0.22187567
     y: 0.27636918
   }
   relative_keypoints {
     x: 0.8914497
     y: 0.27647448
   }
}]
```

얼굴임을 확신하는 %

모든크기는 0-1값

얼굴사각형 크기

RIGHT_EYE

LEFT_EYE

NOSE_TIP

MOUTH_CENTER

RIGHT_EAR_TRAGION

LEFT_EAR_TRAGION

[4] face_detection모듈을 이용한 코드

참고) https://puleugo.tistory.com/4

```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt

mp_face_detection=mp.solutions.fa
mp_drawing=mp.solutions.drawing_u


IMAGE_FILES=['./img/long.png']

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사
    model_selection=0, min_detection_confidence=0.5

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        results=face_detection.process(cv2.cvtColor
        print('#'*100)

        print(results.detections)
```

long.png

0.0s

```
####################################################################################################
None
```

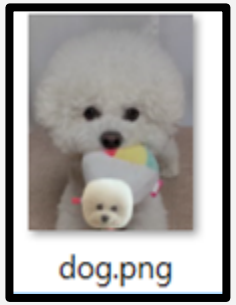
```python
import cv2
import mediapi
import matplot


mp_face_detecti
mp_drawing=mp.s


IMAGE_FILES=['./img/long.png']

with mp_face_detection.FaceDetec

    ### selection=1은 5m 이내의
    model_selection=1, min_detec

    for idx, file in enumerate(I
        image=cv2.imread(file)
        results=face_detection.p
        print('#'*100)

        print(results.detections
```

long.png

0.0s

```
########################################
[label_id: 0
score: 0.8843321
location_data {
  format: RELATIVE BOUNDING BOX
```

```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt

mp_face_detection=mp.solutions.face_detection
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=['./img/dog.png']

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=1, min_detection_confidence=0.5) as face_detection:

    for idx, file in enum
        image=cv2.imread(
        results=face_dete


        if not results.de
            print("Face n
        else:
            print(results
```



dog.png

```python
mp_face_detection=mp.solutions.face_detec
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=['./img/dog.png']

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2
    model_selection=0, min_detection_conf

    for idx, file in enumerate(IMAGE_FILE
        image=cv2.imread(file)
        results=face_detection.process(cv


        if not results.detections:
            print("Face not found in imag
        else:
            print(results.detections)
```

0.0s

```
[label_id: 0
score: 0.59620696
```

[4] face_detection모듈을 이용한 코드

참고) https://puleugo.tistory.com/4

```
IMAGE_FILES=['./img/sample.jpg']

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=1, min_detection_confidence=0.5) as face_detection:

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        results=face_detection.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))


        if not results.detections:
            print("Face not found in image")
        else:
            print(len(results.detections))
```

1.2s

```
2
```

첫번째인식얼굴

```
[label_id: 0
score: 0.84290385
location_data {
  format: RELATIVE_BOUNDING_BOX
  relative_bounding_box {
    xmin: 0.4062397
    ymin: 0.27737606
    width: 0.07183576
    height: 0.10767686
  }
  relative_keypoints {
    x: 0.42865098
    y: 0.30971527
  }
  relative_keypoints {
    x: 0.45559222
    y: 0.30989414
  }
  relative_keypoints {
    x: 0.43918645
    y: 0.33326507
  }
  relative_keypoints {
    x: 0.4418567
    y: 0.35496944
  }
  relative_keypoints {
    x: 0.41933575
    y: 0.3247264
  }
  relative_keypoints {
    x: 0.47998714
    y: 0.32117993
  }
}
```

두번째인식얼굴

```
, label_id: 0
score: 0.73074967
location_data {
  format: RELATIVE_BOUNDING_BOX
  relative_bounding_box {
    xmin: 0.25379795
    ymin: 0.2927487
    width: 0.08250666
    height: 0.123734
  }
  relative_keypoints {
    x: 0.28104448
    y: 0.33116978
  }
  relative_keypoints {
    x: 0.31091326
    y: 0.32574838
  }
  relative_keypoints {
    x: 0.3093937
    y: 0.35433245
  }
  relative_keypoints {
    x: 0.30499995
    y: 0.38063258
  }
  relative_keypoints {
    x: 0.24221723
    y: 0.35301334
  }
  relative_keypoints {
    x: 0.30845618
    y: 0.33971715
  }
}
]
```

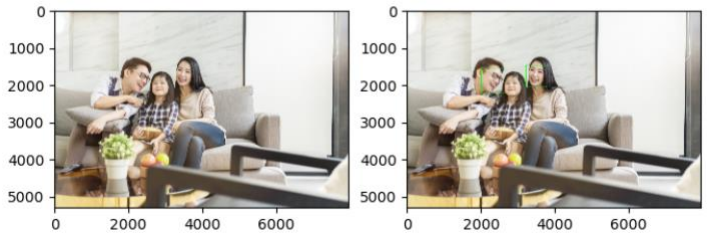
[5]얼굴에 박스 및 포인트 위치 잡기

```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt
from glob import glob

mp_face_detection=mp.solutions.face_detection
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=glob('./img/*.*')

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=1, min_detection_confidence=0.5) as face_detection:

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        image=cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results=face_detection.process(image)


        if not results.detections:

            print(str(idx) + '-->'+ file + ':Face not found in image')

        else:
            print(str(idx) + '-->'+  file + ':'+ str(len(results.detections)) + 'count')
            annotated_image = image.copy()
            for detection in results.detections:
                mp_drawing.draw_detection(annotated_image, detection,
                                          bbox_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255, 0), thickness=7))

            plt.figure(figsize=(8,20))
            plt.subplot(1,2,1);plt.imshow(image)
            plt.subplot(1,2,2);plt.imshow(annotated_image)
            plt.show()
```

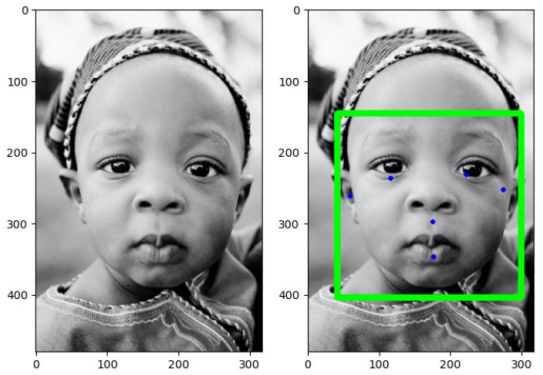
1번으로 작업시

```
0-->./img\dog.png:Face not found in image
1-->./img\long.png:1count
```

```
2-->./img\sample.jpg:2count
```



```
3-->./img\short.jpg:1count
```

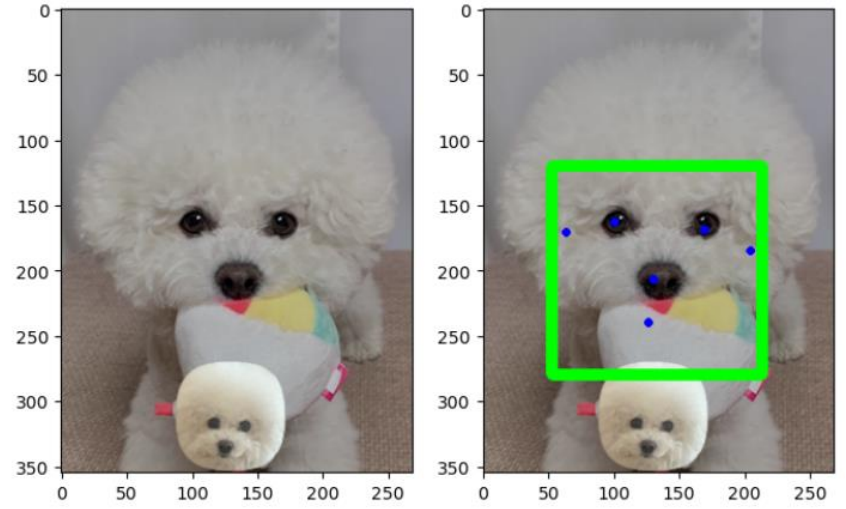


[5]얼굴에 박스 및 포인트 위치 잡기

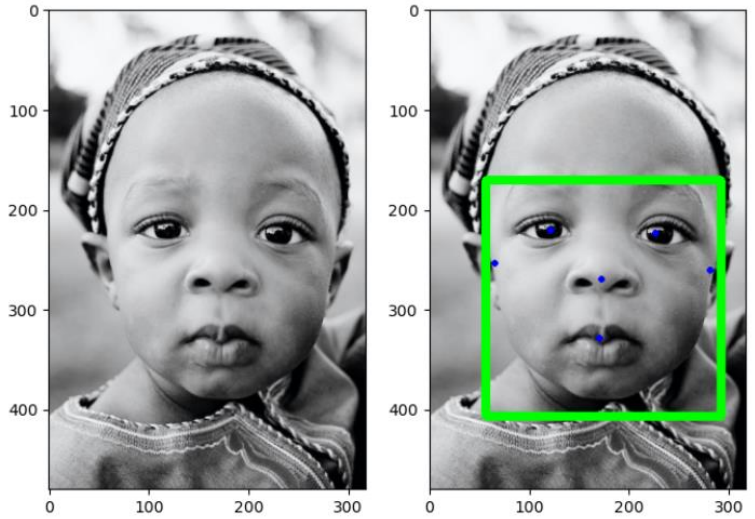
```python
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt
from glob import glob

mp_face_detection=mp.solutions.face_detection
mp_drawing=mp.solutions.drawing_utils


IMAGE_FILES=glob('./img/*.*')

with mp_face_detection.FaceDetection(

    ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
    model_selection=0, min_detection_confidence=0.5) as face_detection:

    for idx, file in enumerate(IMAGE_FILES):
        image=cv2.imread(file)
        image=cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results=face_detection.process(image)


        if not results.detections:

            print(str(idx) + '-->'+ file + ':Face not found in image')

        else:
            print(str(idx) + '-->'+  file + ':'+ str(len(results.detections)) + 'count')
            annotated_image = image.copy()
            for detection in results.detections:
                mp_drawing.draw_detection(annotated_image, detection,
                                          bbox_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255, 0), thickness=7))

            plt.figure(figsize=(8,20))
            plt.subplot(1,2,1);plt.imshow(image)
            plt.subplot(1,2,2);plt.imshow(annotated_image)
            plt.show()
```

0번으로 작업시

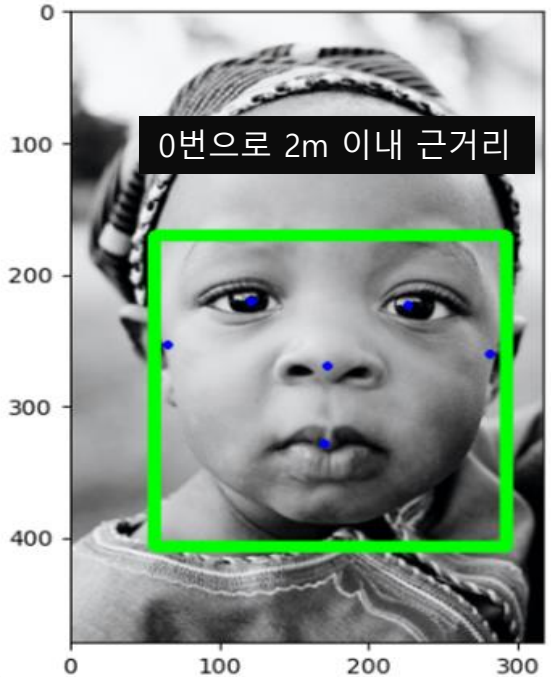
```
0-->./img\dog.png:1count
```



```
1-->./img\long.png:Face not found in image
2-->./img\sample.jpg:Face not found in image
3-->./img\short.jpg:1count
```



[5]얼굴에 박스 및 포인트 위치 잡기



[5]key 포인트 값 받기 - [Part2] 눈,코,입,귀등 포인트 위치값 출력

```
8
9  IMAGE_FILES=['./img/short.jpg']
10
11 with mp_face_detection.FaceDetection(
12
13     ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
14     model_selection=0, min_detection_confidence=0.5) as face_detection:
15
16     for idx, file in enumerate(IMAGE_FILES):
17         image=cv2.imread(file)
18         results=face_detection.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
19
20
21         if not results.detections:
22             print("Face not found in image")
23         else:
24             print(len(results.detections)) # 얼굴인식 갯수 출력
25
26             for detection in results.detections:
27
28                 for x in detection.location_data.relative_keypoints:
29                     print('-'*50)
30                     print(x)
31
32
```

✓ 0.0s

```
relative_keypoints {
  x: 0.42865098
  y: 0.30971527
}
relative_keypoints {
  x: 0.45559222
  y: 0.30989414
}
relative_keypoints {
  x: 0.43918645
  y: 0.33326507
}
relative_keypoints {
```

face_detection.py

```
def get_key_point(
    detection: detection_pb2.Detection, key_point_enum: 'FaceKeyPoint'
) -> Union[None, location_data_pb2.LocationData.RelativeKeypoint]:
  """A convenience method to return a face key point by the FaceKeyPoint type.

  Args:
    detection: A detection proto message that contains face key points.
    key_point_enum: A FaceKeyPoint type.

  Returns:
    A RelativeKeypoint proto message.
  """
  if not detection or not detection.location_data:
    return None
  return detection.location_data.relative_keypoints[key_point_enum]
```

```
----------------------------------------------------
x: 0.3815947
y: 0.4612929
```

RIGHT_EYE

```
----------------------------------------------------
x: 0.7119123
y: 0.46759182
```

LEFT_EYE

```
----------------------------------------------------
x: 0.54288363
y: 0.563848
```

NOSE_TIP

```
----------------------------------------------------
x: 0.5366237
y: 0.6869578
```

MOUTH_CENTER

```
----------------------------------------------------
x: 0.20462587
y: 0.5311299
```

RIGHT_EAR_TRAGION

```
----------------------------------------------------
x: 0.8856636
y: 0.5446642
```

LEFT_EAR_TRAGION

[5]key 포인트 값 받기 - [Part3] FaceKeyPoint를 이용하여 지정한 객체값 위치만 출력

```
20
21          if not results.detections:
22              print("Face not found in image")
23          else:
24              print(len(results.detections)) # 얼굴인식 갯수 출력
25
26              print(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.NOSE_TIP))
27
28              print(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint(2)))
29
```

✓ 0.0s

```
1
x: 0.54288363
y: 0.563848

x: 0.54288363
y: 0.563848
```

face_detection.py

```
class FaceKeyPoint(enum.IntEnum):
  """The enum type of the six face detection key points."""
  RIGHT_EYE = 0
  LEFT_EYE = 1
  NOSE_TIP = 2
  MOUTH_CENTER = 3
  RIGHT_EAR_TRAGION = 4
  LEFT_EAR_TRAGION = 5
```

```
9  IMAGE_FILES=['./img/short.jpg']
10 xList=['RIGHT_EYE','LEFT_EYE','NOSE_TIP','MOUTH_CENTER','RIGHT_EAR_TRAGION','LEFT_EAR_TRAGION']
11
12 with mp_face_detection.FaceDetection(
13
14     ### selection=1은 5m 이내의 전신, 0은 2m 이내의 사진, 기본값은 0
15     model_selection=0, min_detection_confidence=0.5) as face_detection:
16
17     for idx, file in enumerate(IMAGE_FILES):
18         image=cv2.imread(file)
19         results=face_detection.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
20
21
22         if not results.detections:
23             print("Face not found in image")
24         else:
25             print(len(results.detections)) # 얼굴인식 갯수 출력
26
27             for x in range(6):
28                 print('-->'+ xList[x])
29                 print(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint(x)))
30
```

출력물

```
-->RIGHT_EYE
x: 0.3815947
y: 0.4612929

-->LEFT_EYE
x: 0.7119123
y: 0.46759182

-->NOSE_TIP
x: 0.54288363
y: 0.563848

-->MOUTH_CENTER
x: 0.5366237
y: 0.6869578

-->RIGHT_EAR_TRAGION
x: 0.20462587
y: 0.5311299

-->LEFT_EAR_TRAGION
x: 0.8856636
y: 0.5446642
```