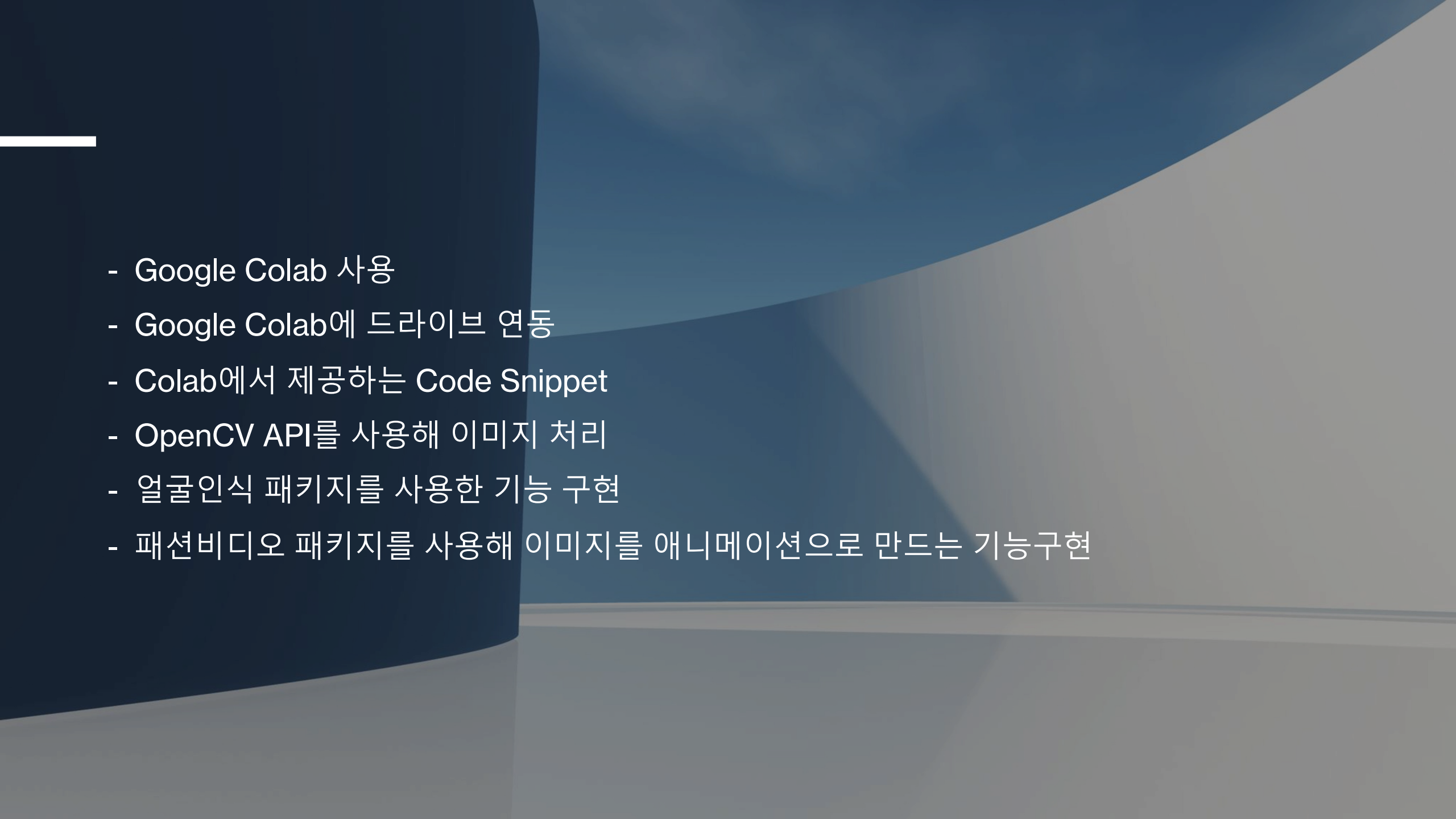


FIRST-ORDER- MOTION-MODEL

GOOGLE COLAB으로
FIRST-ORDER-MOTION-MODEL 사용하기

- 
- Google Colab 사용
 - Google Colab에 드라이브 연동
 - Colab에서 제공하는 Code Snippet
 - OpenCV API를 사용해 이미지 처리
 - 얼굴인식 패키지를 사용한 기능 구현
 - 패션비디오 패키지를 사용해 이미지를 애니메이션으로 만드는 기능구현

Google Colab

- 주피터 노트북과 유사한 코드블럭 사용 프로그램
- Code Snippet 기능을 통해 라이브러리를 쉽게 사용해볼 수 있음
- 머신러닝과 데이터분석 등에 용이함
- 구글 드라이브와 연동해 데이터 관리가 쉬움

Google Colab에 드라이브 연동

```
from google.colab import drive
drive.mount('/gdrive')
```

노트북에서 Google Drive 파일에 액세스하도록 허용하시겠습니까?

이 노트북에서 Google Drive 파일에 대한 액세스를 요청합니다. Google Drive에 대한 액세스 권한을 부여하면 노트북에서 실행되는 코드가 Google Drive의 파일을 수정할 수 있게 됩니다. 이 액세스를 허용하기 전에 노트북 코드를 검토하시기 바랍니다.

[아니요](#)

[Google Drive에 연결](#)

- colab에서 드라이브를 import
- 구글 드라이브 액세스를 허용
- 연동이 잘 안되면 `drive.flush_and_unmount()` 를 한번 실행한 후 다시 mount를 시도

Colab에서 제공하는 Code Snippet

- 카메라로 이미지를 캡처할 수 있는 기능
- 코드스니펫에 Camera Capture을 검색하면 자동으로 코드가 삽입된다
- 코드 실행 시 카메라가 켜지고 캡처 버튼으로 화면에 띄울 수 있다
- 이외에도 파일업로드, 구글시트-드라이브에서 파일 가져오기, 코드 숨기기 등 다양한 스니펫 기능이 있다

```
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo(filename='photo.jpg', quality=0.8):
    js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      // Resize the output to fit the video element.
      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.
      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
    display(js)
    data = eval_js('takePhoto({})'.format(quality))
    binary = b64decode(data.split(',')[1])
    with open(filename, 'wb') as f:
        f.write(binary)
    return filename
```

```
[ ] from IPython.display import Image
try:
    filename = take_photo()
    print('Saved to {}'.format(filename))

    # Show the image which was just taken.
    display(Image(filename))
except Exception as err:
    # Errors will be thrown if the user does not have a webcam or if they do not
    # grant the page permission to access it.
    print(str(err))
```

Saved to photo.jpg



OpenCV API를 사용해 이미지 처리

1. First-Order-Motion-Model 을 사용하기 위해서 깃헙 클론
2. 패키지 import
3. 나의 구글 드라이브에서 이미지, 비디오 가져오기
4. 이미지, 비디오 사이즈 조정하기
5. 이전에 학습해둔 모델 불러오기
6. 이미지 애니메이션으로 만들고 재생하기

OpenCV API를 사용해 이미지 처리

1. First-Order-Motion-Model 을 사용하기 위해서 깃헙 클론

2. 패키지 import

3. 나의 `!git clone https://github.com/AliaksandrSiarohin/first-order-model`

4. 이미지, 비디오 사이즈 조정하기

5. 이전에 학습해둔 모델 불러오기

6. 이미지 애니메이션으로 만들고 재생하기

OpenCV API를 사용해 이미지 처리

1. First-Order-Motion-Model 을 사용하기 위해서 깃헙 클론

2. 패키지 import

3. 나의 구글 드라이브에서 이미지, 비디오 가져오기

4. 이미지, 비디오 사이즈 조정하기

5. 이전에 학습해둔 모델 불러오기

6. 이미지 애니메이션으로 만들고 재생하기

```
import imageio
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from skimage.transform import resize
from IPython.display import HTML
import warnings
warnings.filterwarnings("ignore")
```


얼굴인식 패키지를 사용한 기능 구현

- 구글 드라이브에 저장한 정면 얼굴 이미지1과 말하는 사람 얼굴 비디오1을 imageio로 가져온 뒤 리사이징한다
- 학습된 VoxCeleb 모델의 체크포인트를 load_checkpoint() 함수로 이미지에 적용시킨다
- display() 함수를 사용해 출력한다



패션비디오 패키지를 사용해 이미지를 애니메이션으로 만드는 기능구현

- 구글 드라이브에 저장한 정면 모델 사진3과 움직이는 모델 비디오1을 imageio로 가져온 뒤 리사이징한다
- 나머지는 얼굴인식 패키지를 이용한 기능구현과 같은 방법을 사용한다



— 감사합니다