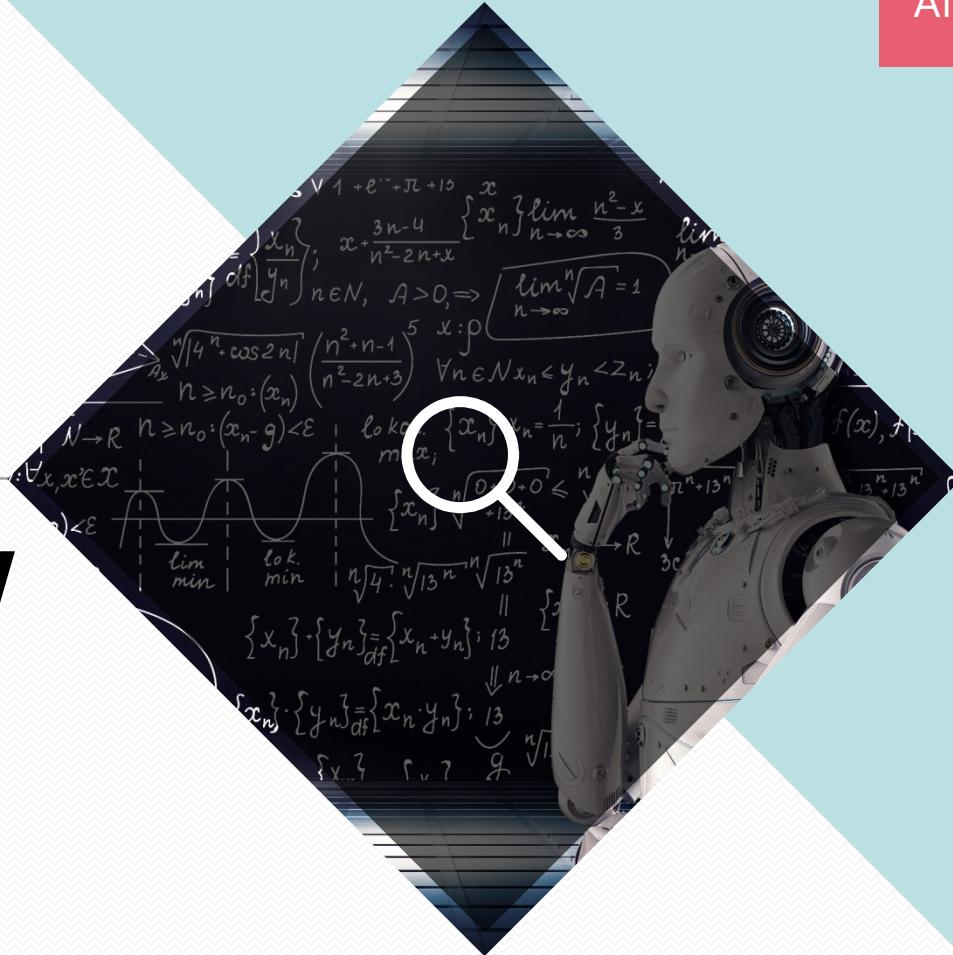


Quick AI Review & Mini Project



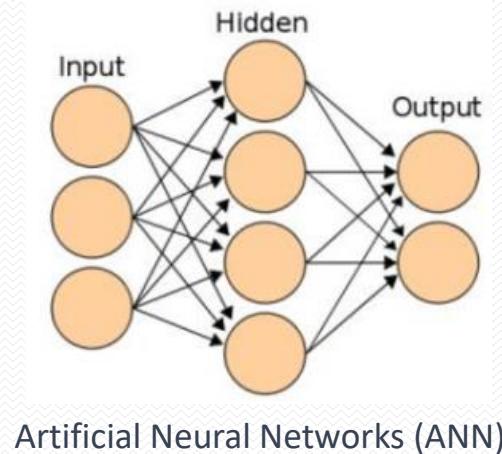
Contents

1. AI 용어/개념 정리
2. 전체적인 AI project 개발 방법
3. 전처리/후처리를 위한 OpenCV
4. OpenVINO 뭐지? 꼭 알아야 할까?
5. Practices
6. Mini Project

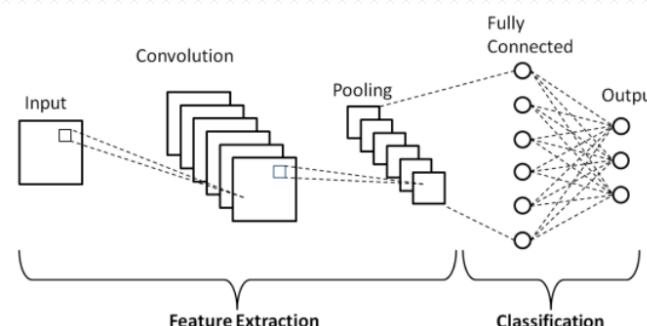
AI 용어정리 - ANN, CNN, RNN, Transformer

신경망 모델의 유형을 살펴보자. 크게 ANN / CNN / RNN / Transformer 등으로 나눈다.

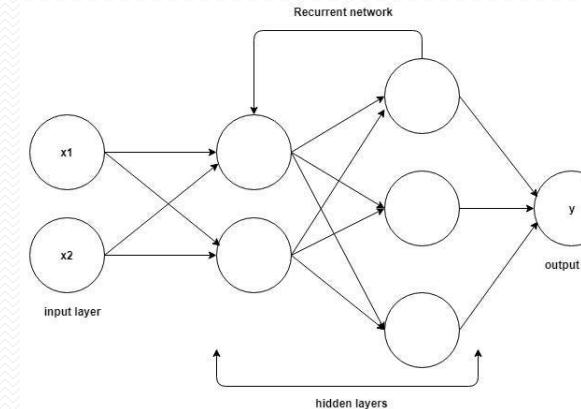
- ANN(인공 신경망), for classification.
- CNN(합성곱 신경망) , for feature extraction.
- RNN(순환 신경망), for sequential data.
- Transformer, for sequential data with better parallel processing & long-range data



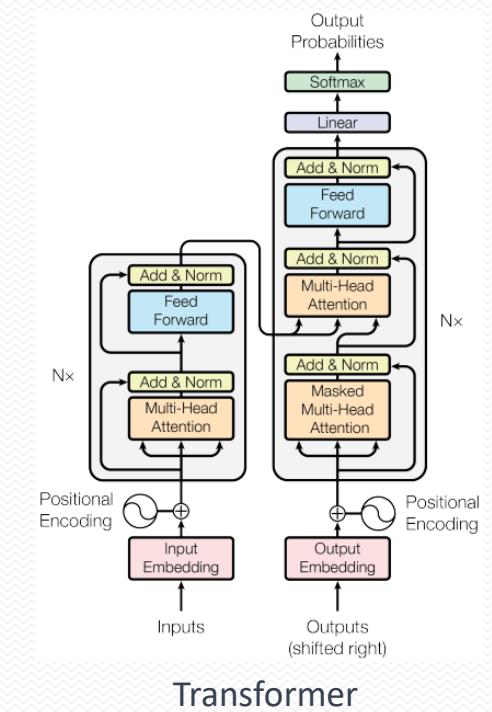
Artificial Neural Networks (ANN)



Convolutional Neural Network (CNN)



Recurrent Neural Network (RNN)

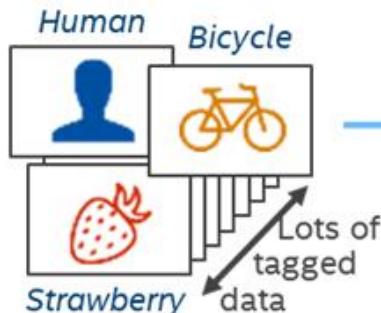


Transformer

AI 용어정리 - Training(훈련) vs Inference(추론)

Training은 모델이 학습하는 과정이며, Inference는 그 학습된 모델을 새로운 데이터에 적용하는 과정

TRAINING:

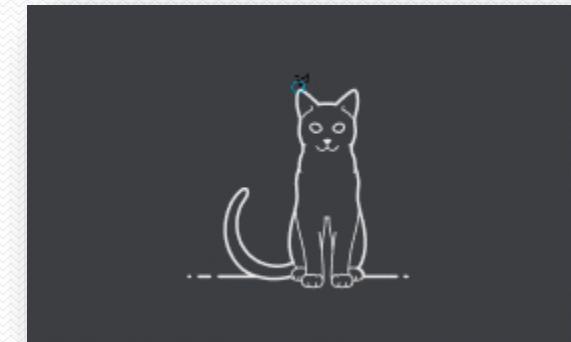
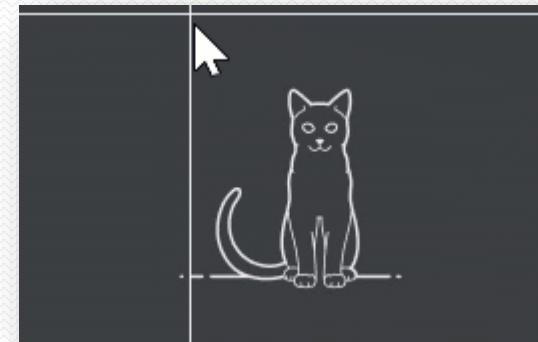


INFERENCE:

YOLOv3 모델을 한번 Inference 할때 약 1초정도 걸리는 HW로 해당 AI 모델을 10,000 개의 데이터셋으로 training 한다고 하면 얼마나 걸릴까?
상상해봅시다.

AI 용어정리 - Computer Vision AI Models

- **Image Classification** (이미지 분류)
- **Object Detection** (물체 감지)
- **Image Segmentation** (이미지 분할)



AI 용어정리 - NLP(자연어 처리) AI Models

ChatGPT 는 어떤 NLP AI Model 일까?

- **Text Classification(텍스트 분류):** 텍스트를 사전 정의된 카테고리로 분류
- **Question Answering(질문 응답):** 자연어 질문에 대한 답변을 제공
- **Translation(번역):** 하나의 언어로 된 텍스트를 다른 언어로 변환
- **Summarization(요약):** 긴 텍스트를 짧고 핵심적인 내용으로 요약
- **Conversational(대화형):** 자연스러운 대화를 생성하고 유지하는 모델
- **Text Generation(텍스트 생성):** 주어진 입력에 기반하여 새로운 텍스트를 생성

AI 용어정리 - Multimodal AI Models

유튜브 등에 사람들이 Multimodal 이런 표현을 쓰는.. 이게 뭘까요 ?

- "멀티모달 AI 모델"이라는 용어는 여러 종류의 입력 데이터를 처리할 수 있는 인공지능 모델임
- 이러한 모델은 텍스트, 이미지, 소리 등 다양한 형태의 데이터를 동시에 이해하고 분석함

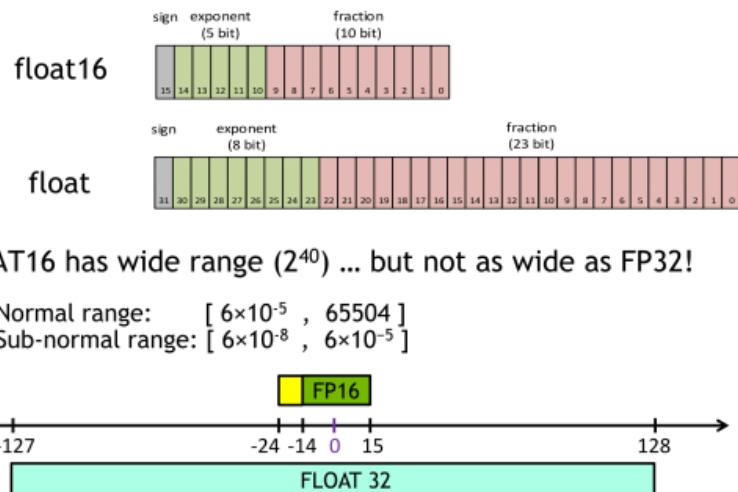
- Document Question Answering
- Text to Image
- Image to Text
- Image to Video
- Text to Video
- Text to 3D
- Image to 3D
- Many others..

AI 용어정리 - Data type & precision of AI

AI 모델에서 INT8/FP16/FP32 와 같은 용어들이 나오네. 이게 뭐지?

- INT8, FP16, FP32 등의 표현은 AI 모델의 가중치나 연산에서 사용되는 데이터의 형식과 정밀도를 의미
- 데이터의 형식과 정밀도는 모델의 학습과 추론에 큰 영향을 미치며, 높은 정밀도는 더 정확한 연산을 제공하지만, 더 많은 메모리와 연산 리소스를 필요로 함. 반면에 낮은 정밀도는 더 빠른 연산과 메모리 절약을 가능하게 하지만, 모델의 정확도가 줄어들 수 있음

HALF-PRECISION FLOAT (FLOAT16)



1. INT8 (8-bit integer)

- 정수 값을 8비트로 표현.
- 양자화된 모델에서 자주 사용되며, 경량화와 추론 속도 향상을 위해 사용

2. INT16 (16-bit integer)

- 정수 값을 16비트로 표현
- 일부 중간 정밀도 연산이나 특정 하드웨어에서의 연산을 위해 사용

3. INT32 (32-bit integer)

- 정수 값을 32비트로 표현
- 일반적인 CPU 연산 및 일부 고도의 정밀성이 필요한 연산에 사용

4. FP16 (16-bit floating point) or Half precision

- 숫자를 16비트 부동소수점으로 표현
- GPU 연산 최적화와 모델의 경량화를 위해 사용

5. FP32 (32-bit floating point) or Single precision

- 숫자를 32비트 부동소수점으로 표현
- 일반적인 GPU 연산 및 대부분의 훈련 연산에 사용

6. FP64 (64-bit floating point) or Double precision

- 숫자를 64비트 부동소수점으로 표현
- 고도의 정밀성이 필요한 연산 및 과학적 연구에서 사용

7. BF16 (Brain Floating Point 16)

- 16비트 부동소수점이지만, FP16과 다른 비트 구성을 가짐
- Google의 TPU와 같은 특정 하드웨어에서 사용

8. TF32 (TensorFloat-32)

- NVIDIA의 최신 GPU에서 사용되는 특수한 데이터 타입
- FP32의 정밀도와 FP16의 성능 사이의 중간 지점을 제공하기 위해 설계됨

AI 용어정리 - GFLOP

Giga Floating Point Operation

Amount of single precision giga-floating point operations calculated

Most of deep learning calculation is proceed in floating point

1 Giga FLOPS (GFLOPS) computer system is capable of performing **one billion floating-point operations per second**

Easy to represent a large dynamic range

Mostly used in picture information (graphics)

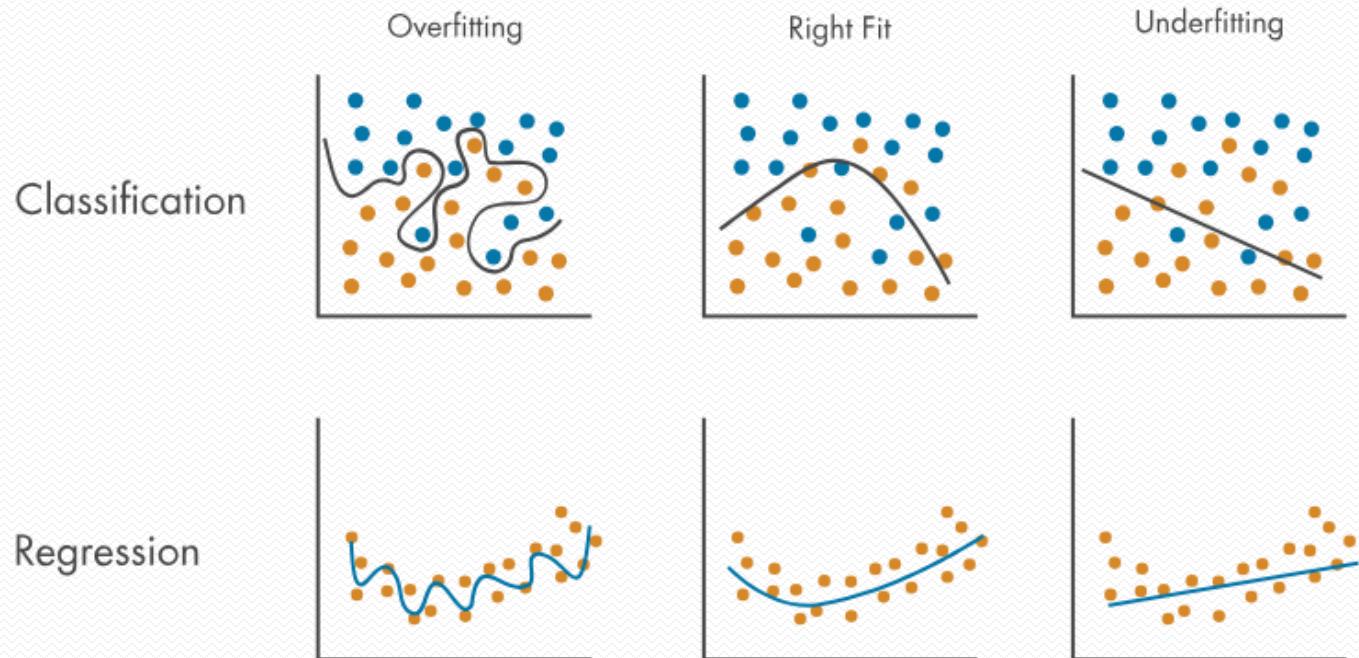
$$\text{FLOPS} = \text{cores} \times \text{clock} \times \frac{\text{FLOPs}}{\text{cycle}}$$

Unit	Flops
kFLOPS	10^3
MFLOPS	10^6
GFLOPS	10^9
TFLOPS	10^{12}
PFLOPS	10^{15}
EFLOPS	10^{18}
ZFLOPS	10^{21}

AI 용어정리 - Overfitting

Overfitting의 개념은 뭐고 왜 발생하나?

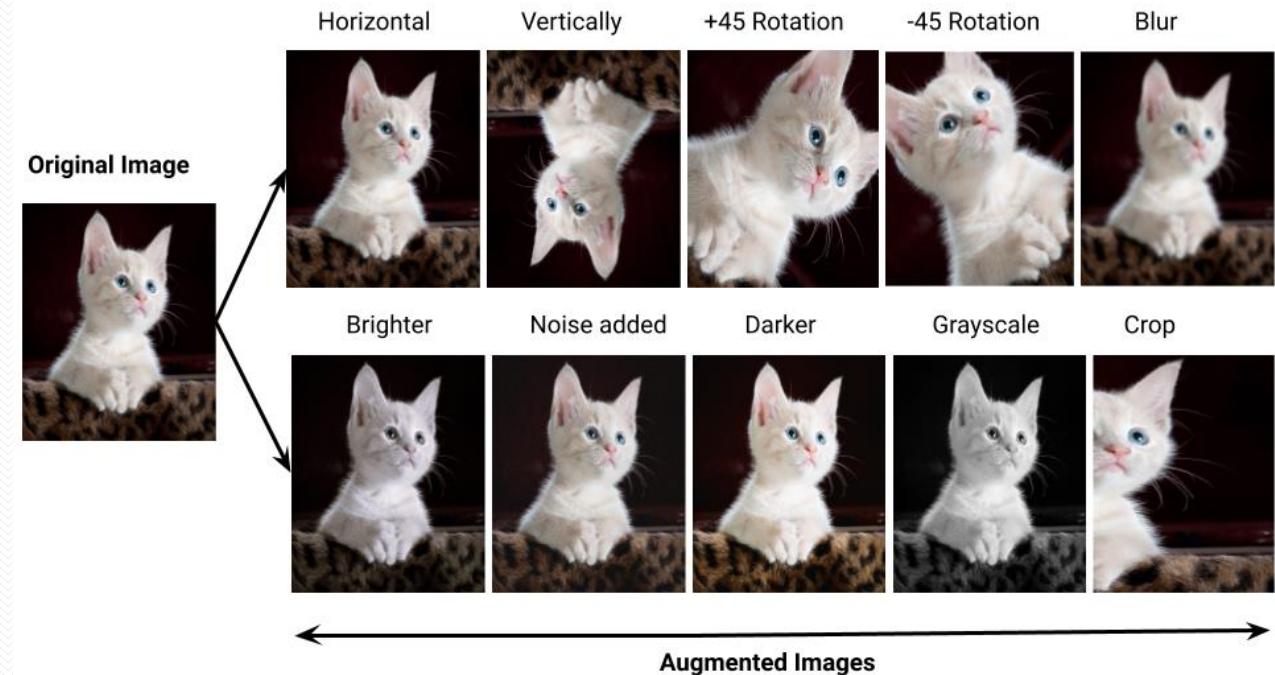
- 오버피팅(Overfitting)은 기계 학습 및 딥러닝에서 모델이 학습 데이터에 지나치게 최적화되어, 새로운 데이터에 대한 예측 성능이 떨어지는 현상을 의미
- 모델이 학습 데이터의 특정 노이즈나 상세한 패턴까지 너무 정확하게 학습하여 일반화 성능이 떨어진 상태
- 모델의 복잡도가 너무 높거나, 학습 데이터의 양이 부족할 때 주로 발생



AI 용어정리 - Data Augmentation

Data Augmentation 은 왜 필요하고 어떻게 가능할까?

- 데이터 증강(Data Augmentation)은 기존의 데이터셋을 활용하여 새로운 데이터를 생성하는 기법
- 주로 딥러닝이나 기계 학습에서 데이터의 양을 늘리기 위해 사용
- 데이터 증강은 원본 데이터에 약간의 변화(회전, 확대/축소, 반전 등)를 가하여 다양한 형태의 새로운 데이터를 만들어내는 방식
- 이를 통해 모델이 오버피팅(과적합)을 줄이고 일반화 성능을 향상시킴



AI 용어정리 - Transfer Learning

Transfer Learning의 개념을 알아보고 활용해 보자.

- Transfer Learning은 사전에 훈련된 모델을 사용하여 새로운 작업을 더 빠르고 효과적으로 학습하는 기계 학습의 방법
- 기본 아이디어는 이미 큰 데이터셋으로 학습된 모델의 지식을 다른 관련된 작업에 전이하여 사용하는 것
- 실제 사용사례: 이미지 인식 - 이미지 분류를 위해 사전에 훈련된 모델이 있습니다. 이 모델은 수백만 개의 이미지를 사용하여 다양한 물체를 인식하는 방법을 배웠습니다. 이제 특정한 작업, 예를 들어 의료 영상에서 종양을 탐지하는 작업을 수행하려고 합니다. 종양 탐지를 위한 데이터는 제한적이지만, 사전에 훈련된 이미지 분류 모델의 지식을 활용하여 종양 탐지 모델을 더 빠르게 학습시킬 수 있습니다. 이미지 분류 모델에서의 일반적인 이미지 인식 능력이 종양 탐지 작업에 도움을 주는 것

```
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.datasets import cifar10

# 데이터 불러오기 및 전처리
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
train_images = train_images / 255.0
test_images = test_images / 255.0

# 사전 훈련된 VGG16 모델 불러오기
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(32, 32, 3))

# 모델의 가중치를 고정 (학습되지 않도록 설정)
for layer in base_model.layers:
    layer.trainable = False

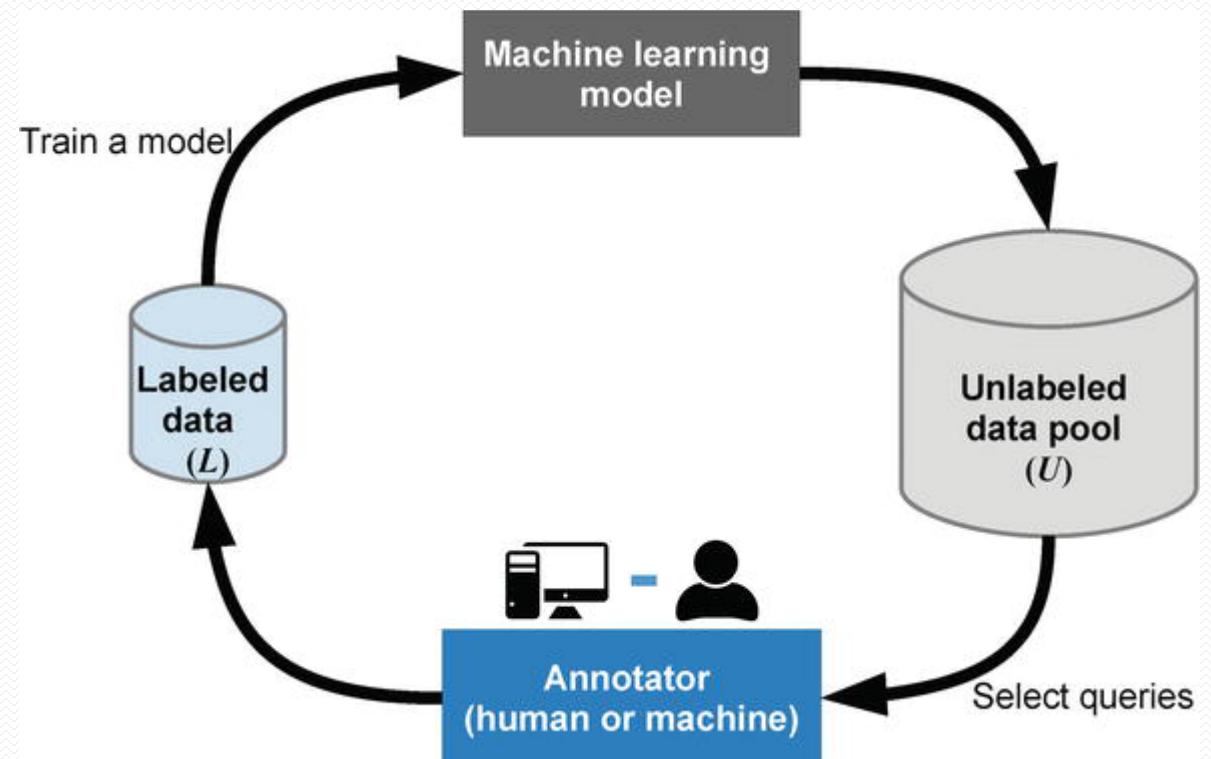
# 사용자 정의 분류기 추가
model = Sequential([
    base_model,
    Flatten(),
    Dense(512, activation='relu'),
    Dense(10, activation='softmax') # CIFAR-10은 10개의 클래스를 가짐
])

# 컴파일 및 학습
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=5, batch_size=64, validation_data=(test_images, test_labels))
```

AI 용어정리 - Active Learning

Active Learning 의 개념과 장점을 알아보자.

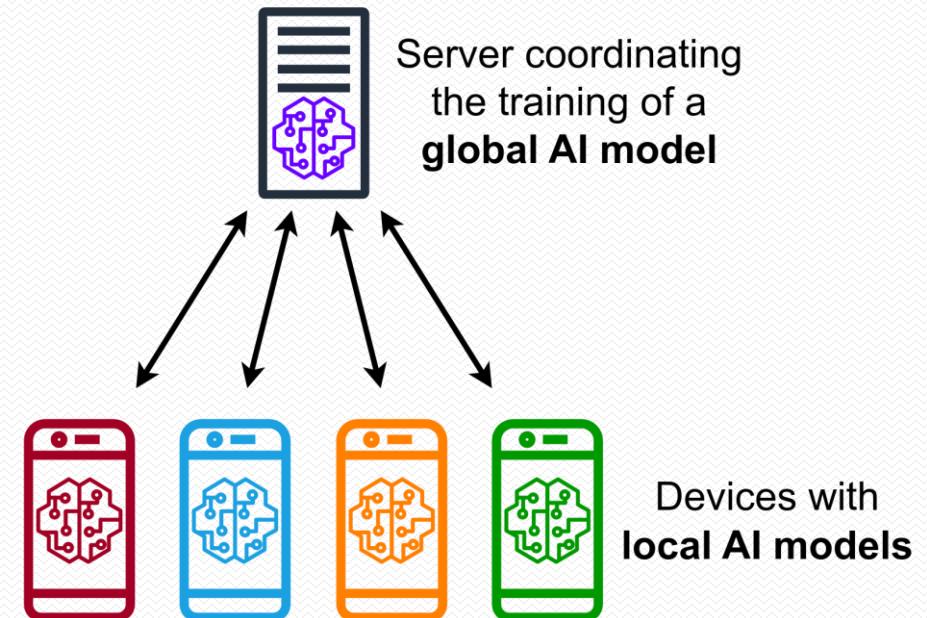
- 모델 스스로 학습에 가장 효과적인 데이터 샘플을 선택하여 라벨링을 요청하는 방식을 의미
- Example) 상상해보세요. 당신은 수천장의 사진을 가지고 있고, 이 사진들 중에서 고양이 사진만을 분류하려고 합니다. 그런데, 각 사진에 라벨을 붙이는 것은 시간과 비용이 많이 듭니다. 여기서 Active Learning을 사용하면, 처음에는 몇 장의 사진만 라벨링하고 이를 기반으로 초기 모델을 학습시킵니다. 그 다음, 모델은 자신이 확신이 가지 않는 사진들 중에서 가장 '의심스러운' 사진 몇장을 선택하여 사용자에게 라벨링을 요청합니다. 이렇게 선택적으로 라벨링을 진행하면서, 모델은 점점 더 정확해지게 됩니다. 결과적으로, 모든 사진을 라벨링하는 것보다 훨씬 적은 수의 사진만 라벨링하면서도 높은 정확도의 모델을 얻을 수 있습니다.



AI 용어정리 - Federated Learning

Federated Learning 의 개념과 사용법

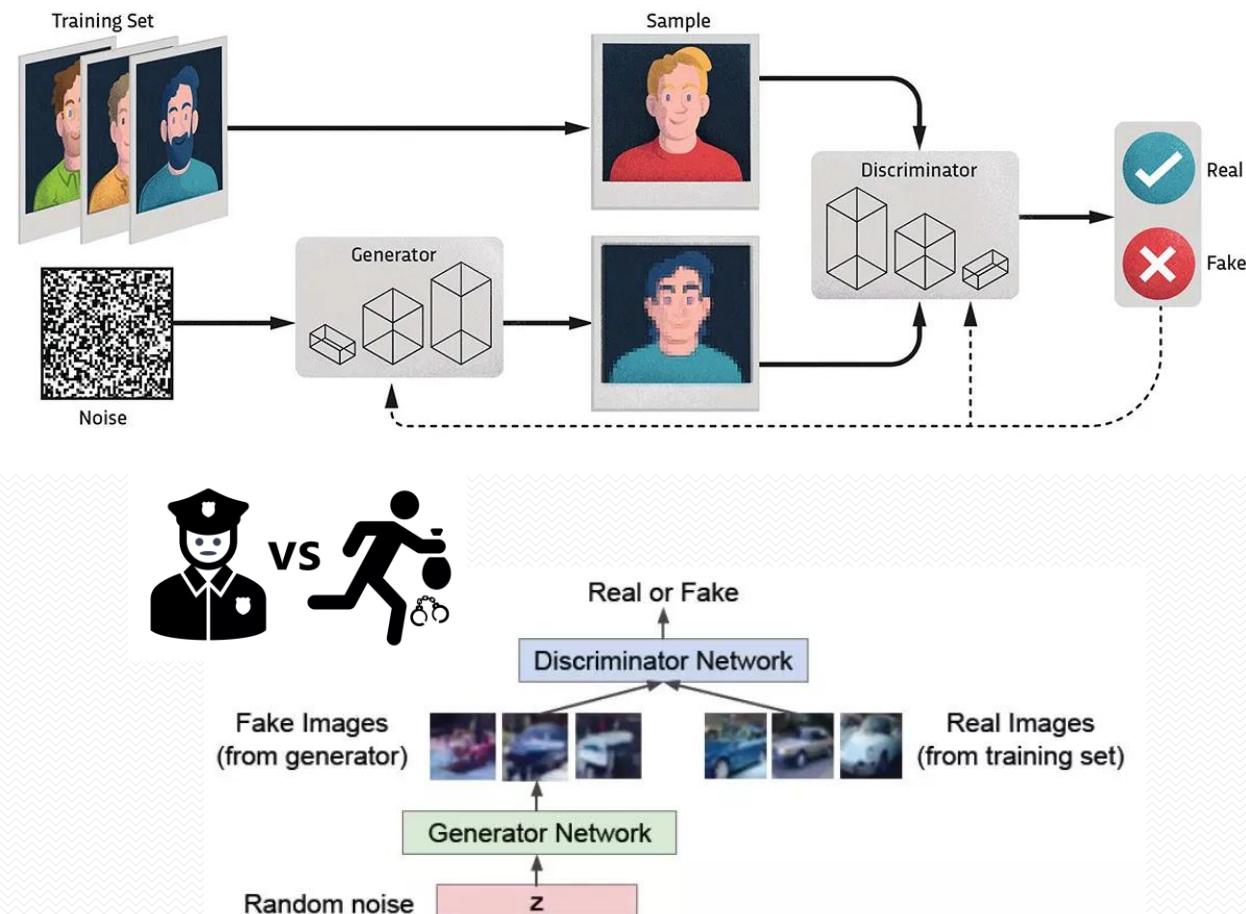
- Federated Learning은 분산된 여러 기기에서 모델 학습을 수행하고, 그 결과를 중앙 서버에서 집계하여 모델을 업데이트하는 방식의 기계 학습
- 핵심 원리
 - 1.로컬 학습: 각 기기는 자신의 데이터로 모델을 독립적으로 학습함
 - 2.모델 업데이트 공유: 각 기기는 로컬에서 학습한 모델의 업데이트를 중앙 서버로 전송
 - 3.집계 및 통합: 중앙 서버는 모든 기기로부터 받은 모델 업데이트를 통합하여 전체 모델을 업데이트
 - 4.업데이트된 모델 배포: 중앙 서버는 업데이트된 모델을 모든 기기에 배포
- 장점
 - 1.데이터 프라이버시: 데이터는 로컬 기기에 머무르므로 중앙 서버에 개인 데이터를 전송할 필요가 없어 의료, 금융 등 민감정보 포함하는 곳에 주로 사용
 - 2.효율적인 대역폭 사용: 모델의 파라미터만 전송되므로 대량의 데이터 전송이 필요 없음
 - 3.실시간 학습: 각 기기에서 실시간으로 데이터를 수집하고 학습할 수 있음



AI 용어정리 - Generative AI

생성형 AI는 기존의 AI 모델들과 어떤 차이가 있을까?

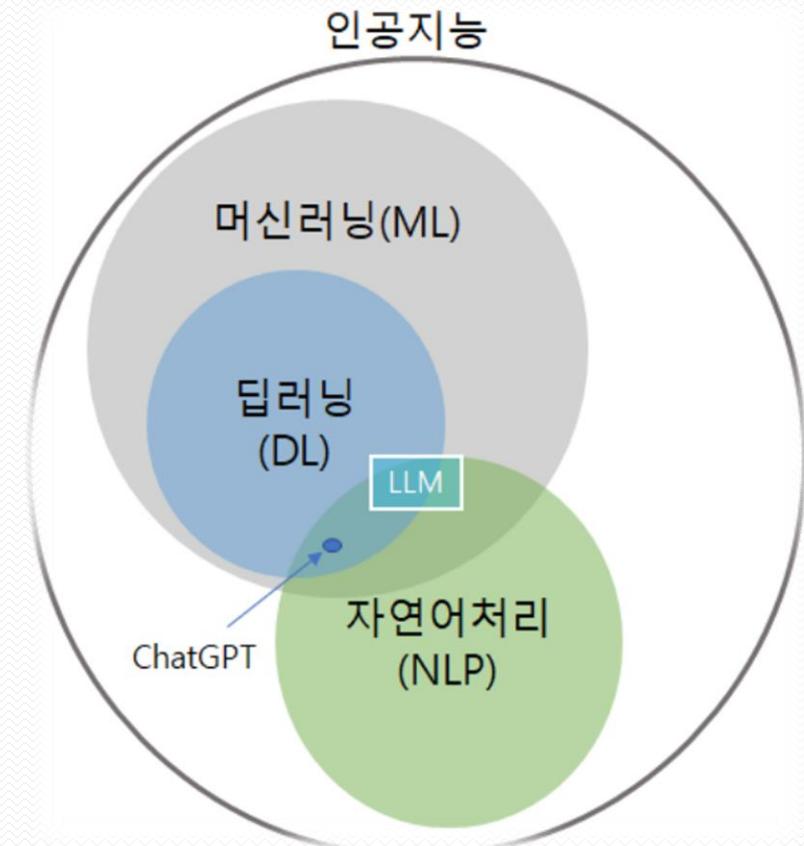
- Generative AI는 데이터를 기반으로 새로운 내용을 생성할 수 있는 인공지능의 한 분야
- 이미지, 음악, 텍스트 등의 새로운 콘텐츠를 생성하는 것이 포함되며 대표적으로 Generative Adversarial Networks (적대적 생성 신경망, GANs)가 있음
- 기존의 AI는 주로 데이터를 분류하거나 예측하는데 중점을 둔다면, Generative AI는 새로운 콘텐츠를 생성하는 것에 중점을 둠
- 훈련 방식의 차이
 - 기존의 AI: 주로 지도 학습 방식을 사용하며, 입력 데이터와 그에 해당하는 라벨을 사용하여 모델을 훈련
 - Generative AI: 주로 비지도 학습 또는 준지도 학습 방식을 사용하며, 데이터의 분포를 학습합니다. GAN의 경우, 생성자와 판별자 두 개의 네트워크가 경쟁하는 방식으로 훈련



AI 용어정리 - LLM (Large Language Model)

LLM 이라고 자꾸 사람들이 말한다. 대체 기존의 AI 와 뭐가 다른가?

- **NLP (자연어 처리, Natural Language Processing):**
 - 정의: NLP는 컴퓨터가 인간의 언어를 이해하고 처리할 수 있도록 도와주는 컴퓨터 과학 및 인공 지능의 하위 분야
 - 목적: 텍스트나 음성 데이터를 분석, 이해 및 생성하기 위한 기술들을 개발하는 것
 - 예시: 기계 번역, 감정 분석, 텍스트 요약, 음성 인식 등이 NLP의 응용 분야임
- **LLM (대규모 언어 모델, Large Language Model):**
 - 정의: LLM은 방대한 양의 텍스트 데이터를 학습하여 자연어 처리 작업을 수행하는 딥러닝 모델
 - 목적: 복잡한 언어 패턴을 학습하여 다양한 NLP 작업에서 높은 성능을 달성하는 것
 - 예시: OpenAI의 GPT (Generative Pre-trained Transformer) 시리즈와 같은 모델들이 LLM의 대표적인 예
- LLM이 NLP 분야의 일부분임



Contents

1. AI 용어/개념 정리

2. 전체적인 AI project 개발 방법

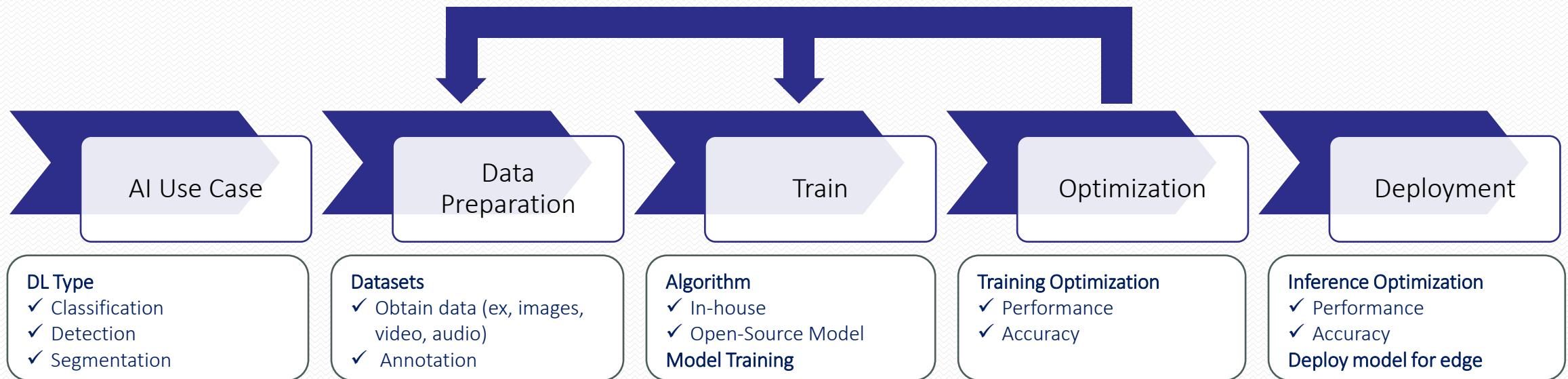
3. 전처리/후처리를 위한 OpenCV

4. OpenVINO 뭐지? 꼭 알아야 할까?

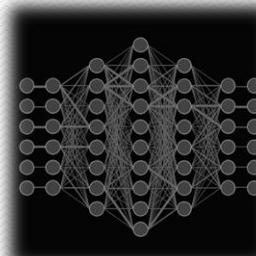
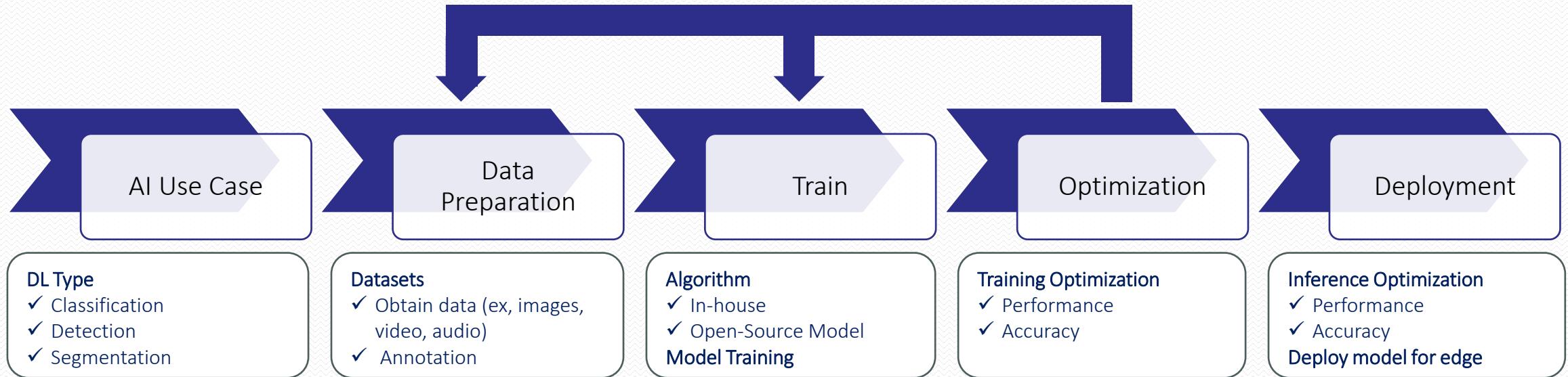
5. Practices

6. Mini Project

2. 전체적인 AI project 개발 방법



2. 전체적인 AI project 개발 방법



AI MODELS
OPTIMIZATION



Contents

1. AI 용어/개념 정리

2. 전체적인 AI project 개발 방법

3. 전처리/후처리를 위한 OpenCV

4. OpenVINO 뭐지? 꼭 알아야 할까?

5. Practices

6. Mini Project

Pre-processing (전처리) for Vision AI model

이미지 전처리가 왜 필요할까? OpenCV 를 꼭 써야하나?

- 결론: 전처리는 반드시 필요함. 꼭 OpenCV 로 안해도 되지만 이게 편하고 쉽다.
그래서 다들 많이 쓴다.

```
# Download the image from the openvino_notebooks storage
image_filename = download_file(
    "https://storage.openvinotoolkit.org/repositories/openvino_notebooks/data/data/image/intel_rnb.jpg",
    directory="data"
)

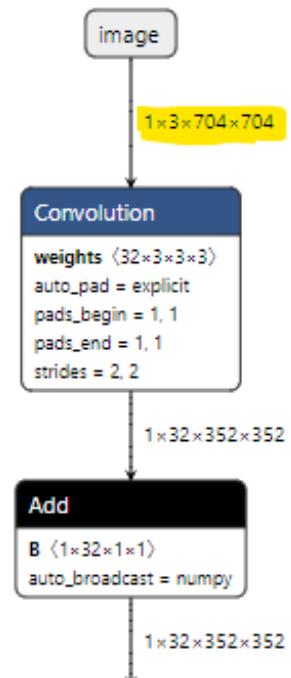
# Text detection models expect an image in BGR format.
image = cv2.imread(str(image_filename))

# N,C,H,W = batch size, number of channels, height, width.
N, C, H, W = input_layer_ir.shape

# Resize the image to meet network expected input sizes.
resized_image = cv2.resize(image, (W, H))

# Reshape to the network input shape.
input_image = np.expand_dims(resized_image.transpose(2, 0, 1), 0)

plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB));
```



Post-processing (후처리) for Vision AI model

후처리가 왜 필요할까? OpenCV 를 꼭 써야하나?

- 결론: 결과를 알기쉽게 보여주기 위해서 필요할 수 있음. 꼭 OpenCV로 안해도 되지만 이게 편하고 쉽다. 그래서 다들 많이 쓴다.

```
# Iterate through non-zero boxes.
for box in boxes:
    # Pick a confidence factor from the last place in an array.
    conf = box[-1]
    if conf > threshold:
        # Convert float to int and multiply corner position of each box by x and y ratio.
        # If the bounding box is found at the top of the image,
        # position the upper box bar little lower to make it visible on the image.
        (x_min, y_min, x_max, y_max) = [
            int(max(corner_position * ratio_y, 10)) if idx % 2
            else int(corner_position * ratio_x)
            for idx, corner_position in enumerate(box[:-1])
        ]

        # Draw a box based on the position, parameters in rectangle function are: image, start_point, end_point, color,
        rgb_image = cv2.rectangle(rgb_image, (x_min, y_min), (x_max, y_max), colors["green"], 3)

        # Add text to the image based on position and confidence.
        # Parameters in text function are: image, text, bottom-left_corner_textfield, font, font_scale, color, thickness:
        if conf_labels:
            rgb_image = cv2.putText(
                rgb_image,
                f"{conf:.2f}",
                (x_min, y_min - 10),
                cv2.FONT_HERSHEY_SIMPLEX,
                0.8,
                colors["red"],
                1,
                cv2.LINE_AA,
            )

return rgb_image
```



3. 이미지 - Basic Operation

- 이미지를 Read / Write / Display

```
import numpy as np
import cv2

# 이미지 파일을 Read
img = cv2.imread("my_input.jpg")

# Image 란 이름의 Display 창 생성
cv2.namedWindow("image", cv2.WINDOW_NORMAL)

# Numpy ndarray H/W/C order
print(img.shape)           ← dashed arrow from question 1

# Read 한 이미지 파일을 Display
cv2.imshow("image", img)

# 별도 키 입력이 있을때 까지 대기
cv2.waitKey(0)             ← dashed arrow from question 2

# output.png 로 읽은 이미지 파일을 저장
cv2.imwrite("output.png", img)

# Destroy all windows
cv2.destroyAllWindows()
```



Quiz

- print(img.shape) 의 출력 결과는 무슨 의미일까?
- 본인이 좋아하는 사진을 web에서 다운받아서 OpenCV API를 사용해서 Display 및 파일로 저장해 보자.
- 현재는 별도의 키 입력이 있을때 까지 cv2.waitKey(0) 함수에서 대기하게 된다. 코드를 추가해서 소문자 "s" 키를 입력받을때만 이미지 파일을 저장하고 다른 키가 입력되면 이미지 파일을 저장하지 않게 수정해 보자.

3. 이미지 - Basic Operation

▪ RGB/HSV Color Space (색 공간)

```
# 이미지 파일을 Read 하고 Color space 정보 출력
color = cv2.imread("strawberry.jpg", cv2.IMREAD_COLOR)
#color = cv2.imread("strawberry_dark.jpg", cv2.IMREAD_COLOR)
print(color.shape)

height, width, channels = color.shape
cv2.imshow("Original Image", color)

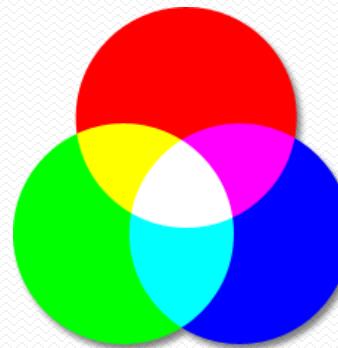
# Color channel 을 B,G,R 로 분할하여 출력
b,g,r = cv2.split(color)
rgb_split = np.concatenate((b,g,r),axis=1)
cv2.imshow("BGR Channels",rgb_split)

# 색공간을 BGR 에서 HSV 로 변환
hsv = cv2.cvtColor(color, cv2.COLOR_BGR2HSV)

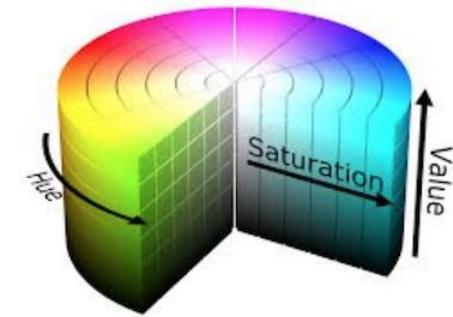
# Channel 을 H,S,V 로 분할하여 출력
h,s,v = cv2.split(hsv)
hsv_split = np.concatenate((h,s,v),axis=1)
cv2.imshow("Split HSV",hsv_split)
```

```
enum cv::ImreadModes {
    cv::IMREAD_UNCHANGED = -1,
    cv::IMREAD_GRAYSCALE = 0,
    cv::IMREAD_COLOR = 1,
    cv::IMREAD_ANYDEPTH = 2,
    cv::IMREAD_ANYCOLOR = 4,
    cv::IMREAD_LOAD_GDAL = 8,
    cv::IMREAD_REDUCED_GRAYSCALE_2 = 16,
    cv::IMREAD_REDUCED_COLOR_2 = 17,
    cv::IMREAD_REDUCED_GRAYSCALE_4 = 32,
    cv::IMREAD_REDUCED_COLOR_4 = 33,
    cv::IMREAD_REDUCED_GRAYSCALE_8 = 64,
    cv::IMREAD_REDUCED_COLOR_8 = 65,
    cv::IMREAD_IGNORE_ORIENTATION = 128
}
```

https://www.w3schools.com/colors/colors_picker.asp



RGB color space



HSV color space

Quiz

1. 위 색공간 이미지의 링크로 이동해서 각 색 공간의 표현 방법을 이해해 보자.
2. HSV color space 가 어떤 경우에 효과적으로 사용될까?
3. HSV로 변환된 이미지를 BGR 이 아닌 RGB로 다시 변환해서 출력해 보자.
4. COLOR_RGB2GRAY 를 사용해서 흑백으로 변환해 출력해 보자.

3. 이미지 - Basic Operation

- Crop / Resize (자르기 / 크기 조정)

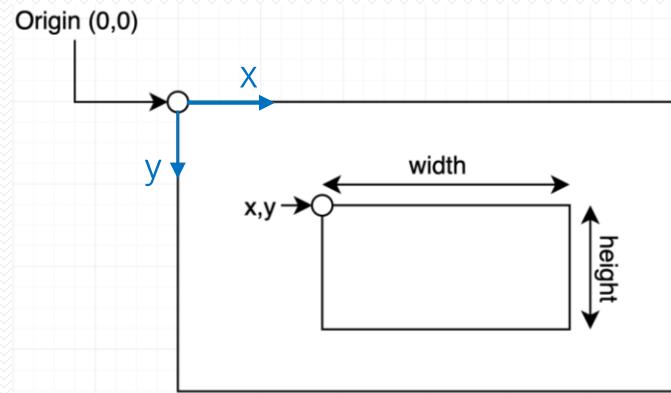
```
# 이미지 파일을 Read
img = cv2.imread("my_input.jpg")

# Crop 300x400 from original image from (100, 50)=(x,y)
cropped = img[50:450, 100:400]

# Resize cropped image from 300x400 to 400x200
resized = cv2.resize(cropped, (400,200))

# Display all
cv2.imshow("Original", img)
cv2.imshow("Cropped image", cropped)
cv2.imshow("Resized image", resized)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



< OpenCV에서 2D 이미지 좌표 개념 >

Quiz

1. Input image 를 본인이 좋아하는 인물 사진으로 변경해서 적용하자. 그리고 본인이 사용한 input image 의 size 를 확인해 보자.
2. 본인이 사용한 이미지의 얼굴 영역만 crop 해서 display 해 보자.
3. 원본 이미지의 정확히 1.5배만큼 이미지를 확대해서 파일로 저장해 보자.
4. openCV 의 rotate API 를 사용해서 우측으로 90도만큼 회전된 이미지를 출력해 보자.

4. 동영상 - Basic Operation

- 동영상 파일을 읽고 보여주기

```
import numpy as np
import cv2

# Read from the recorded video file
cap = cv2.VideoCapture("ronaldinho.mp4")

# 동영상 파일이 성공적으로 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()

    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")  
    break

    # Display
    cv2.imshow("Frame",frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Length	00:01:08
Frame width	1280
Frame height	720
Data rate	2562 kbps
Total bitrate	2565 kbps
Frame rate	24.00 frames/second



Quiz

- 동영상이 너무 빠르게 재생된다. 이유를 찾아보고 정상적인 속도로 재생될 수 있도록 수정해 보자.
- 동영상이 끝까지 재생되면 더이상 frame 을 읽어오지 못해 종료된다. 동영상이 끝까지 재생되면 다시 처음부터 반복 될 수 있도록 수정해 보자.
- 동영상 크기를 반으로 resize 해서 출력해 보자.
- 동영상 재생 중 'c' 키 입력을 받으면 해당 프레임을 이미지 파일로 저장하게 코드를 수정해 보자. 파일 이름은 001.jpg, 002.jpg 등으로 overwrite 되지 않게 하자.

5. 카메라 - Basic Operation

- 카메라로부터 input 을 받아 보여주고 동영상 파일로 저장하기

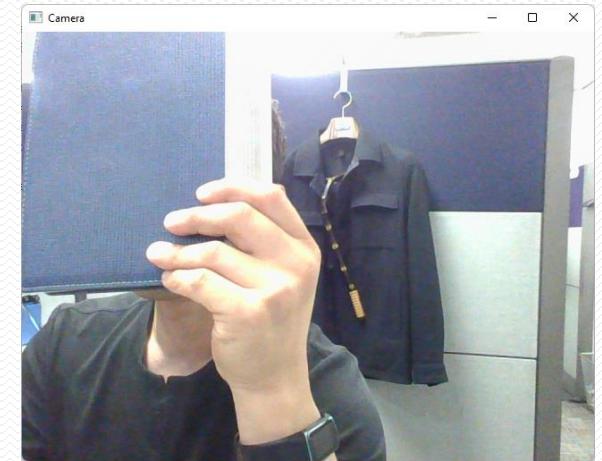
```
# Read from the first camera device
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

w = 640#1280#1920
h = 480#720#1080
cap.set(cv2.CAP_PROP_FRAME_WIDTH, w)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, h) ←

# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()
    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Display
    cv2.imshow("Camera",frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break
```



Quiz

- 가지고 있는 카메라의 지원 가능한 해상도를 확인 후 카메라 해상도를 변경해 보자.
- 카메라 Input 을 “output.mp4” 동영상 파일로 저장하도록 코드를 추가해 보자.

Webcam Manual Control

<https://github.com/kccistc/intel-03/blob/main/class01/webcam.md>

Installation

```
sudo apt install v4l-utils
```



Supported format / resolution / fps list

```
v4l2-ctl --device=/dev/video0 --list-formats-ext
```



Supported tuning parameters

```
v4l2-ctl --device=/dev/video0 --list-ctrls-menus
```



How to tune

- Get value

```
v4l2-ctl --device=/dev/video0 --get-ctrl brightness
v4l2-ctl --device=/dev/video0 --get-ctrl contrast
v4l2-ctl --device=/dev/video0 --get-ctrl saturation
v4l2-ctl --device=/dev/video0 --get-ctrl hue
v4l2-ctl --device=/dev/video0 --get-ctrl gamma
v4l2-ctl --device=/dev/video0 --get-ctrl power_line_frequency
v4l2-ctl --device=/dev/video0 --get-ctrl sharpness
v4l2-ctl --device=/dev/video0 --get-ctrl backlight_compensation
```



- Set value

```
v4l2-ctl --device=/dev/video0 --set-ctrl brightness=0
v4l2-ctl --device=/dev/video0 --set-ctrl contrast=0
v4l2-ctl --device=/dev/video0 --set-ctrl saturation=0
v4l2-ctl --device=/dev/video0 --set-ctrl hue=0
v4l2-ctl --device=/dev/video0 --set-ctrl gamma=0
v4l2-ctl --device=/dev/video0 --set-ctrl power_line_frequency=0
v4l2-ctl --device=/dev/video0 --set-ctrl sharpness=0
v4l2-ctl --device=/dev/video0 --set-ctrl backlight_compensation=1
```



Contents

1. AI 용어/개념 정리
2. 전체적인 AI project 개발 방법
3. 전처리/후처리를 위한 OpenCV
4. OpenVINO 뭐지? 꼭 알아야 할까?
5. Practices
6. Mini Project

What is OpenVINO?

- OpenVINO: Open Visual Inferencing and Neural network Optimization



- AI model 을 사용한 추론 시 Intel HW 의 성능을 최대로 활용 할 수 있게 해주는 SW Toolkit.
- Computer Vision 을 targeting 하였으나 발전하는 AI 추세에 맞게 NLP, Audio, Other generative AI models 지원.
- 꼭 배워야 AI를 사용 할 수 있나요?
몰라도 됩니다. 다른 AI SW toolkit 을 사용해서 구현 가능해요.
- 면접때 OpenVINO 를 배워서 AI 프로젝트를 수행했다고 해도 될까요?
정확한 사용 목적을 아는 상황에서 얘기하셔야 합니다. 아니면 말 안하느니만 못해요. OpenVINO 는 pretrained AI model 을 사용해 inference 하는 여러 방식 중 하나의 toolkit 이지만, 하나의 동일한 코드로 모든 Intel platform 에서 inference 성능을 극대화 시키는 것이 주 목적입니다.

How OpenVINO works?

https://github.com/openvinotoolkit/openvino_notebooks/blob/main/notebooks/102-pytorch-to-openvino/102-pytorch-to-openvino.ipynb

일반적인
Inference 과정

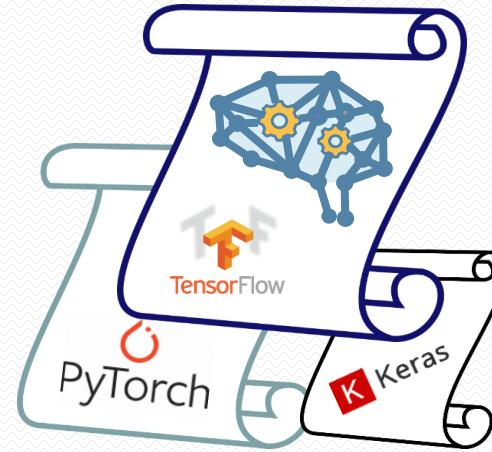


K Keras



Training

Trained
Model

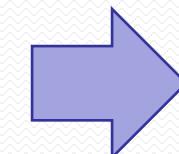
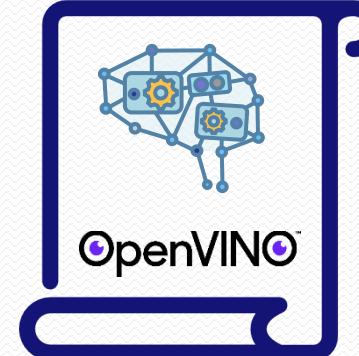


Inference



K Keras

OpenVINO 를
사용한 Inference

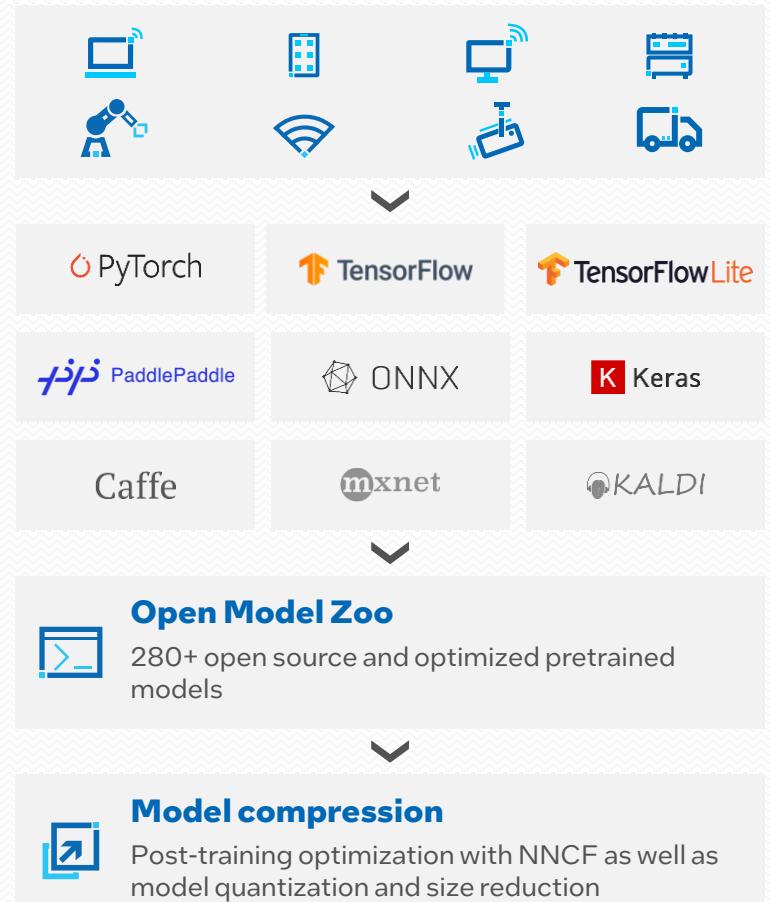


OpenVINO™

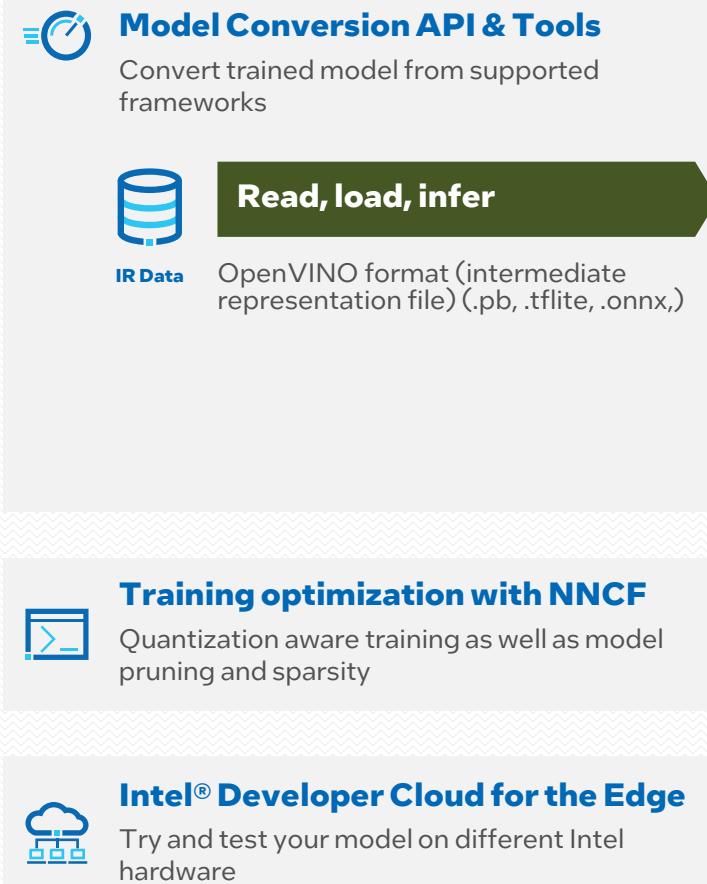
Inference

How OpenVINO works?

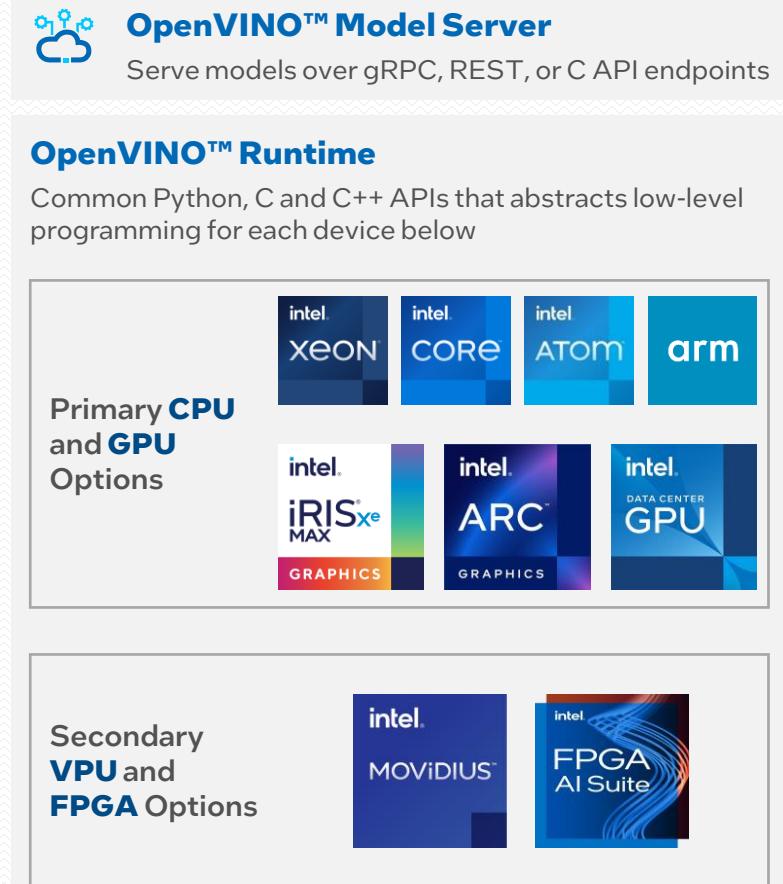
1 | MODEL



2 | OPTIMIZE



3 | DEPLOY



Benefits of OpenVINO

Fast, accurate real-world results with high-performance, deep learning inference

1 MODEL

PyTorch TensorFlow TensorFlowLite PaddlePaddle ONNX Keras Caffe mxnet KALDI



OpenVINO™

2 OPTIMIZE

Optimized Performance

CPU



GPU



VPU



FPGA



3 DEPLOY

Windows

Linux

macOS

Contents

1. AI 용어/개념 정리
2. 전체적인 AI project 개발 방법
3. 전처리/후처리를 위한 OpenCV
4. OpenVINO 뭐지? 꼭 알아야 할까?
5. Practices
6. Mini Project

Practice

https://github.com/openvinotoolkit/openvino_notebooks/blob/main/notebooks/401-object-detection-webcam/401-object-detection.ipynb

OpenVINO tutorial 의 코드 local python code 로 porting 하기

- Goal: 401 예제를 jupyter notebook 환경이 아닌 본인 local machine 의 python 가상환경에 python 코드로 porting 하여 동작시켜보기
- Mini Project 를 수행하기 위해 필요한 과정으로 기본적인 코드 이해가 바탕이 되어야 porting 이 가능함

Contents

1. AI 용어/개념 정리
2. 전체적인 AI project 개발 방법
3. 전처리/후처리를 위한 OpenCV
4. OpenVINO 뭐지? 꼭 알아야 할까?
5. Practices
6. Mini Project

Mini Project

3인 1팀으로 1개 이상의 pre-trained AI 모델들을 사용하여
본인들만의 application 만들기!

- 2개 이상의 Pre-trained AI 모델 사용 시 멋쟁이
- 가용한 모든 자원 활용~ Network, Webcam, Mic 기타 등등
- 실생활에 유용하게 쓰이거나 쓰일만한 idea 를 적용
- Examples
 - 감정이 화날때나 기쁠때 사진 찍어서 관련된 effect 주기
 - OCR 을 통해 특정 단어 인식하여 그에 따른 Service 를 제공
 - Object detection 을 통해 detection 된 물체의 mono depth 를 통해 거리가 너무 가깝거나 멀어지면 경고



THANK YOU

Who is the best?

Question

- <https://github.com/FinanceData/FinanceDataReader> 라이브러리를 이용해서 삼성전자 주식데이터를 크롤링하는 python 코드를 작성해줘
- 겨울에 관한 시를 작성해줘



<https://chat.openai.com/>



<https://clova-x.naver.com/>



<https://bard.google.com/?hl=ko>



<https://replicate.com/meta/llama-2-70b-chat>



Introduction to Hugging Face



An online community and platform that provides community-sourced building blocks for advanced deep learning applications. It is a central repository for models, data sets and documentation that helps users reuse artifacts and build their use cases.

Features

- An open-source community for sharing model artifacts
- Pretrained and packaged models for popular ML tasks
- Preformatted and labeled datasets
- A library and an API for easy training and inference
- Additional building blocks like tokenizers, metrics, etc

Pretrained Models in Hugging Face

- ## Features
- Hosts a huge repository of pretrained model checkpoints
 - Searchable by task, ML framework, architecture, and others
 - Model card provides background and documentation
 - Hosted inference API - to run a quick example
 - Providing Hugging Face transformer libraries built on top of PyTorch and TensorFlow

The screenshot illustrates the Hugging Face platform's features:

1. **Models**: A red box highlights the "Models" tab in the navigation bar.
2. **Tasks**: A red box highlights the "Tasks" section under the search bar.
3. **Model Card**: A red box highlights the "Model card" for the Intel/neural-chat-7b-v3-1 model.
4. **Inference with transformers**: A red box highlights the code snippet for generating a response using the transformers library.

Model card details for Intel/neural-chat-7b-v3-1:

This model is a fine-tuned model based on [mistralai/Mistral-7B-v0.1](#) on the open source dataset [Open-Orca/SlimOrca](#). Then we align it with DPO algorithm. For more details, you can refer our blog: [The Practice of Supervised Fine-tuning and Direct Preference Optimization on Intel Gaudi2](#).

Model date
Neural-chat-7b-v3-1 was trained between September and October, 2023.

Evaluation
We submit our model to [open_llm_leaderboard](#), and the model performance has been **improved significantly** as we see from the average metric of 7 tasks from the leaderboard.

```
model_name = 'Intel/neural-chat-7b-v3-1'
model = transformers.AutoModelForCausalLM.from_pretrained(model_name)
tokenizer = transformers.AutoTokenizer.from_pretrained(model_name)

def generate_response(system_input, user_input):

    # Format the input using the provided template
    prompt = f"### System:{system_input}\n### User:{user_input}\n### Assistant:\n"

    # Tokenize and encode the prompt
    inputs = tokenizer.encode(prompt, return_tensors="pt", add_special_tokens=False)

    # Generate a response
    outputs = model.generate(inputs, max_length=1000, num_return_sequences=1)
    response = tokenizer.decode(outputs[0], skip_special_tokens=True)

    # Extract only the assistant's response
    return response.split("### Assistant:\n")[-1]

# Example usage
system_input = "You are a math expert assistant. Your mission is to help users understand complex mathematical concepts."
user_input = "calculate 100 + 520 + 60"
response = generate_response(system_input, user_input)
print(response)
```

Datasets in Hugging Face



Search models, datasets, users...

Models

Datasets

Spaces

Docs

Solutions

Pricing



Tasks Sizes Sub-tasks Languages Licenses Other

Filter Tasks by name

Multimodal

- Feature Extraction
- Text-to-Image
- Image-to-Text
- Text-to-Video
- Visual Question Answering
- Graph Machine Learning

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Conversational
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity
- Table to Text
- Multiple Choice
- Text Retrieval

Audio

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Tabular

- Tabular Classification
- Tabular Regression
- Tabular to Text
- Time Series Forecasting

Reinforcement Learning

- Reinforcement Learning
- Robotics

Datasets 83,825

Filter by name

Full-text search

Sort: Trending

Dataset	Last Updated	Size	Downloads	Stars
wikimedia/wikipedia	5 days ago	3.24k	96	10
fka/awesome-chatgpt-prompts	Mar 7	882	3.9k	10
epfl-l1m/guidelines	5 days ago	158	41	10
nvidia/HelpSteer	6 days ago	1.27k	100	10
HuggingFaceH4/no_robots	23 days ago	2.73k	226	10
Anthropic/hh-rlhf	May 27	37.7k	770	10
Intel/orca_dpo_pairs	6 days ago	5.44k	44	10
OpenLLM-France/Claire-Dialogue-French-0.1	about 22 hours ago	40	15	10
jondurbin/cinematika-v0.1	5 days ago	220	13	10
togethercomputer/RedPajama-Data-V2	19 days ago	1.3M	210	10
gsdf/EasyNegative	Feb 12	3	1.08k	10
samsum	Dec 27, 2022	285k	189	10
berkeley-nest/Nectar	8 days ago	1.13k	10	10
Lin-Chen/ShareGPT4V	13 days ago	314	112	10
BAAI/CCI-Data	6 days ago	37	38	10
MMMU/MMMU	about 5 hours ago	3.69k	28	10
gaia-benchmark/GAIA	12 days ago	186	67	10
Open-Orca/OpenOrca	Oct 21	28.4k	926	10
Open-Orca/SlimOrca	Oct 12	3.66k	105	10
argilla/ultrafeedback-binarized-preferences	5 days ago	44	13	10
teknum/openhermes	Sep 8	492	87	10
allenai/ultrafeedback_binarized_cleaned	4 days ago	837	11	10
HuggingFaceH4/ultrafeedback_binarized	Oct 27	29.3k	70	10
togethercomputer/RedPajama-Data-1T	Jul 1	22.9k	928	10

Spaces in Hugging Face



Spaces

Discover amazing ML apps made by the community!

Search Spaces

Create new Space or [learn more about Spaces.](#)

Full-text search Sort: Trending

Spaces of the week 🔥

Name	Running on	Created by	Last updated	Stars
Seamless M4T v2	A106	facebook	about 23 hours ago	146
LEDITS++	A106	editing-images	4 days ago	143
Notus Chat	A106	argilla	4 days ago	20
LaVie	A106	Vchitect	1 day ago	83
Automatic Hallucination Detection	A106	mithril-security	12 days ago	8
Seamless Expressive	A106	facebook	5 days ago	60
dreamgaussian-mini	T4	dylanebert	4 days ago	32
Unofficial SDXL Turbo Img2Img Txt2Img	A106	diffusers	4 days ago	181

All running apps, trending first

Name	Running on	Created by	Last updated	Stars
Stable Video Diffusion	A106	multimodalart	6 days ago	613
Unofficial SDXL Turbo Img2Img Txt2Img	A106	diffusers	4 days ago	181
Open LLM Leaderboard	CPU UPGRADE	HuggingFaceH4	about 17 hours ago	6.48k
StyleTTS 2	T4	styletts2	5 days ago	236
Seamless M4T v2	A106	facebook	about 23 hours ago	146
LEDITS++	A106	editing-images	4 days ago	143
ChatGPT Free		ngocuanai	Oct 26	486
GPT Baker		abidlabs	9 days ago	195

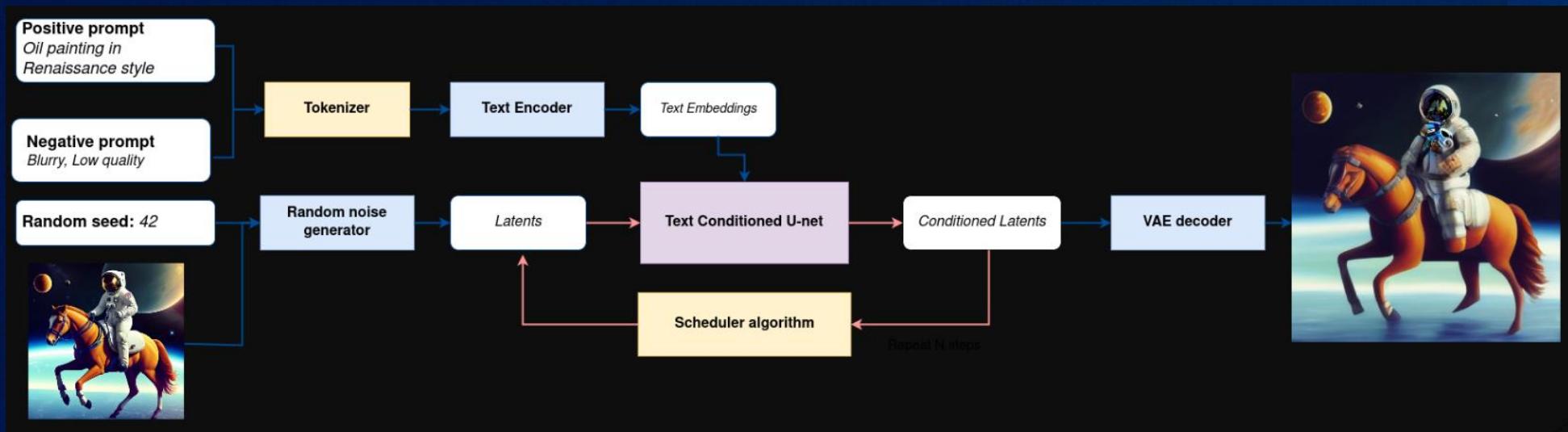
OpenVINO Tutorials with LLM

- 225 - Text-to-Image Generation with Stable Diffusion and OpenVINO
- 233 - Visual Question Answering and Image Captioning using BLIP
- 236 - Stable Diffusion v2.1 using Optimum-Intel OpenVINO
- 250 - Controllable Music Generation with MusicGen and OpenVINO
- 254 - Create an LLM-powered Chatbot using OpenVINO

225 - Text-to-Image Generation with Stable Diffusion and OpenVINO

Stable Diffusion is a text-to-image latent diffusion model. It is trained on 512x512 images from a subset of the LAION-5B database.

https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/225-stable-diffusion-text-to-image



233 - Visual Question Answering and Image Captioning using BLIP

BLIP is a language-image pre-training framework for unified vision-language understanding and generation.

https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/233-blip-visual-language-processing

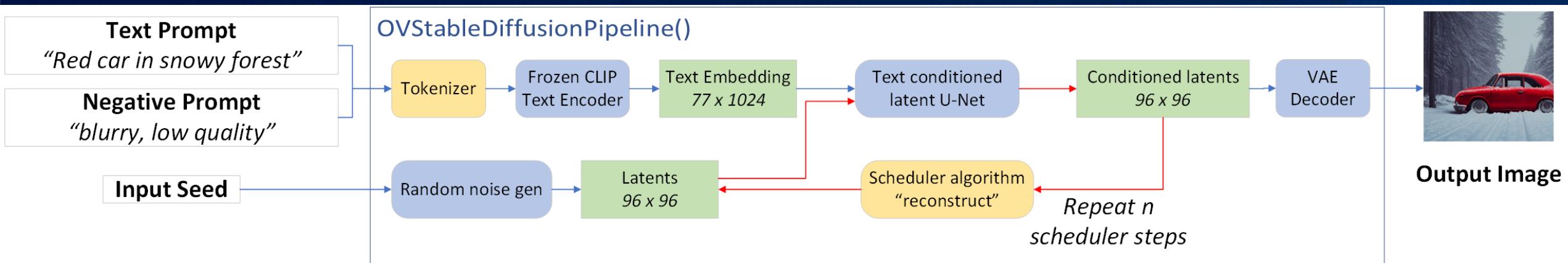


Image-Text Retrieval: “The man in blue shirt is wearing glasses.”

236 - Stable Diffusion v2.1 using Optimum-Intel OpenVINO

- Using full precision model in CPU with `StableDiffusionPipeline`
- Using full precision model in CPU with `OVStableDiffusionPipeline`
- Using full precision model in dGPU with `OVStableDiffusionPipeline`

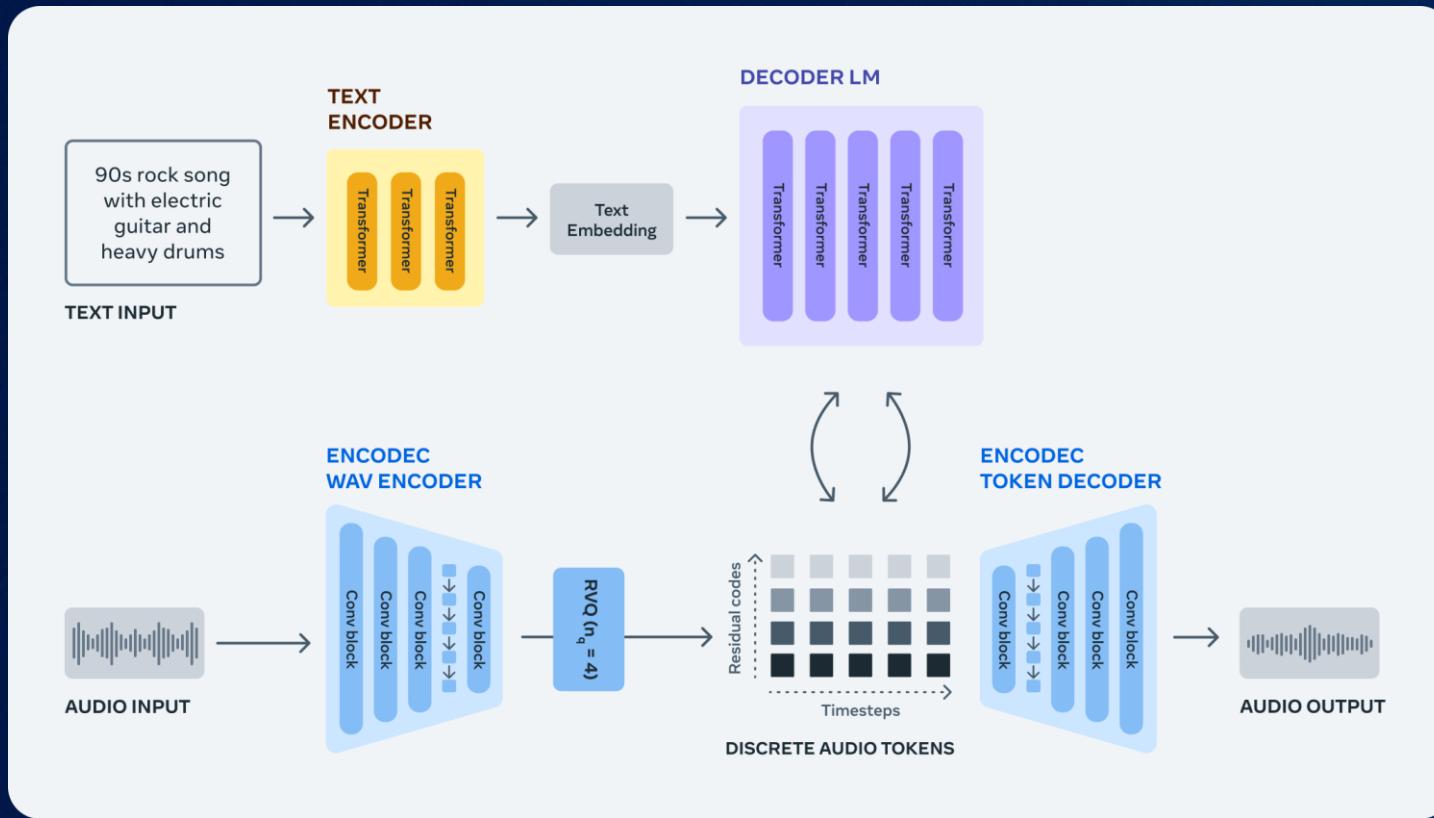
https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/236-stable-diffusion-v2



250 - Controllable Music Generation with MusicGen and OpenVINO

MusicGen is a single-stage auto-regressive Transformer model capable of generating high-quality music samples conditioned on text descriptions or audio prompts.

https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/250-music-generation



254 - Create an LLM-powered Chatbot using OpenVINO

Select model: [red-pajama-3b-chat](#) / [llama-2-7b-chat](#) / [mpt-7b-chat](#) / [zephyr-7b-beta](#) / [qwen-7b-chat](#) / [chatglm2-6b](#)

https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/254-llm-chatbot

OpenVINO llama-2-chat-7b Chatbot

processing | 19.6/51.5s

Chatbot

hi, I am Max. How are you doing?

Hello Max! *smiling* I'm just an AI, I don't have feelings like humans do, but I'm here to help you with any questions or concerns you may have. How can I assist you today? Please keep in mind that I'm just an AI and I cannot provide personal opinions or engage in discussions that promote harmful or unethical content. Is there anything else I can help you with?

I had an argue with my manager. How to apology? can you let me know the more than 5 ways for me?

Of course, I'd be happy to help! Here are some effective ways to apologize to your manager after an argument:

1. Be sincere and direct: Start by acknowledging the issue and expressing your regret directly and sincerely. For example, "Hey boss, My apologies for losing my cool during our meeting earlier. I know I shouldn't have reacted that way."
2. Take responsibility: Make it clear that you take

Chat Message Box

Submit Stop Clear

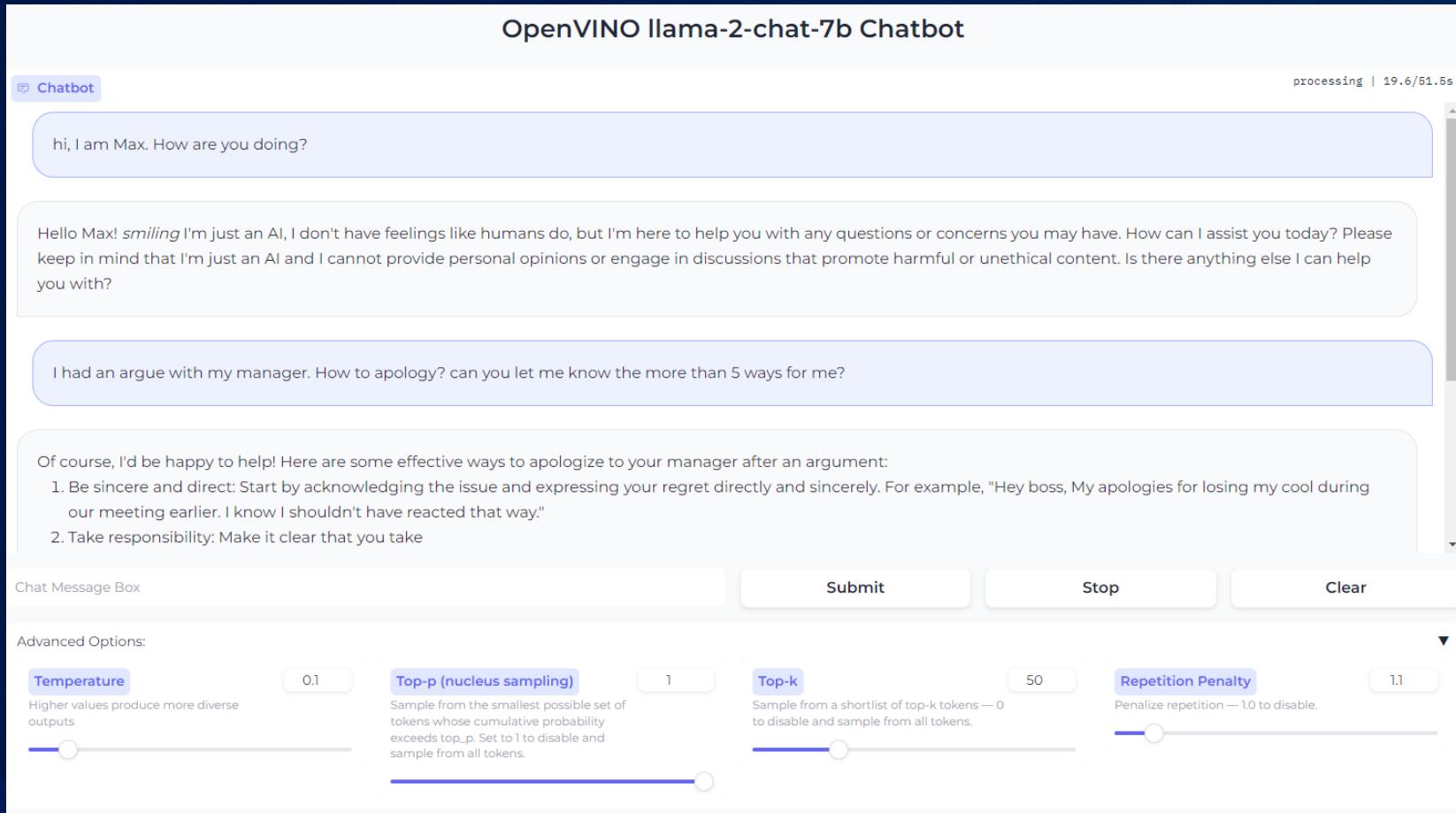
Advanced Options:

Temperature: 0.1 Higher values produce more diverse outputs.

Top-p (nucleus sampling): 1 Sample from the smallest possible set of tokens whose cumulative probability exceeds top_p. Set to 1 to disable and sample from all tokens.

Top-k: 50 Sample from a shortlist of top-k tokens — 0 to disable and sample from all tokens.

Repetition Penalty: 1.1 Penalize repetition — 1.0 to disable.



AI Milestones of the Last 15 Years



Li @ Sanford
launches
ImageNet



IBM Watson
wins Jeopardy



Amazon
introduces Alexa
voice assistant



Google's AlphaGo
defeats Lee Sedol,
grandmaster in Go



OpenAI GPT-2
generates human-like
text responses



Google AlphaFold
solves protein folding for
new drug discovery

'09

'10

'11

'12

'13

'14

'15

'16

'17

'18

'19

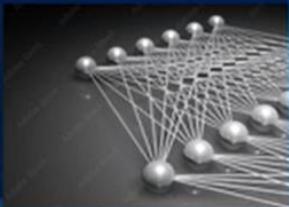
'20

'21

'22

'23

Bengio et al
apply ReLU to
deep learning



Hinton & Ng
learn to find cats
in videos



Tesla Autopilot
introduces AI-based
self-driving cars



Facebook
DeepFace
identifies human faces
with 97% accuracy



OpenAI'
DALL-E
generates realistic
images from text



OpenAI
ChatGPT
based on GPT-4



Gemini

Build with Gemini

Integrate Gemini models into your applications with Google AI Studio
and Google Cloud Vertex AI.

<https://deepmind.google/technologies/gemini/#build-with-gemini>