

심화 교육과정

공개된 데이터셋을 이용하여 분류 모델 실습

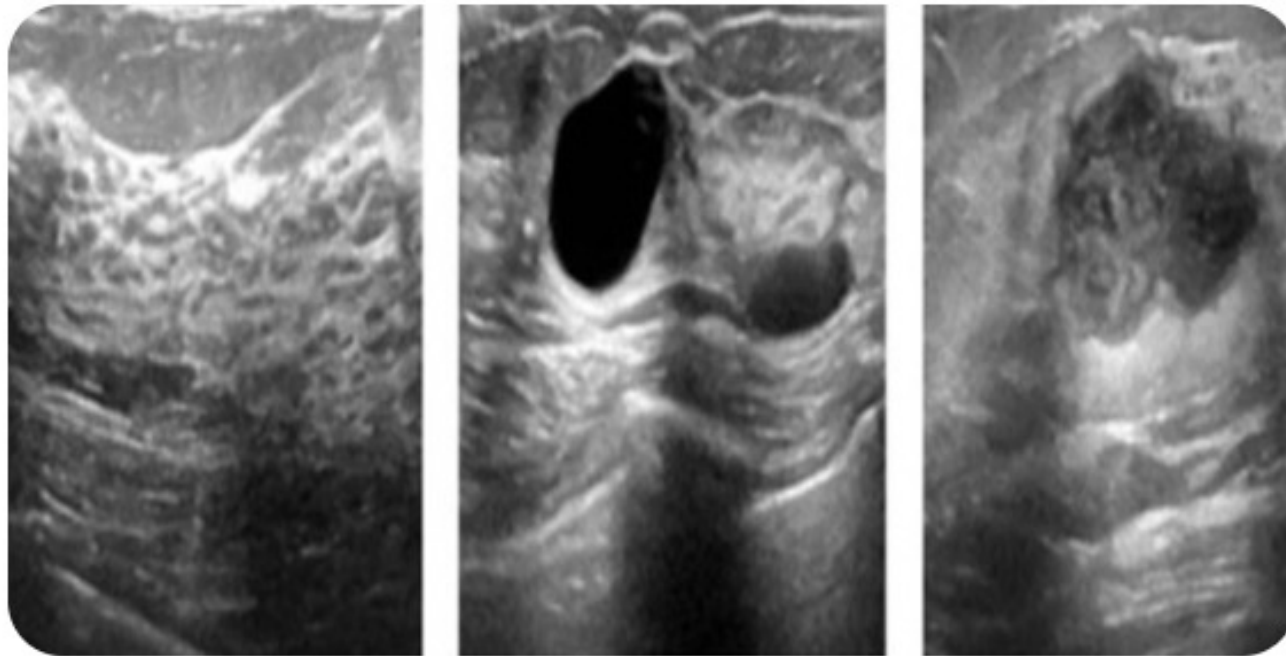
THINK LIFE SYNC AI



공개데이터를 활용한 분류 모델 실습

THINK LIFE SYNC AI

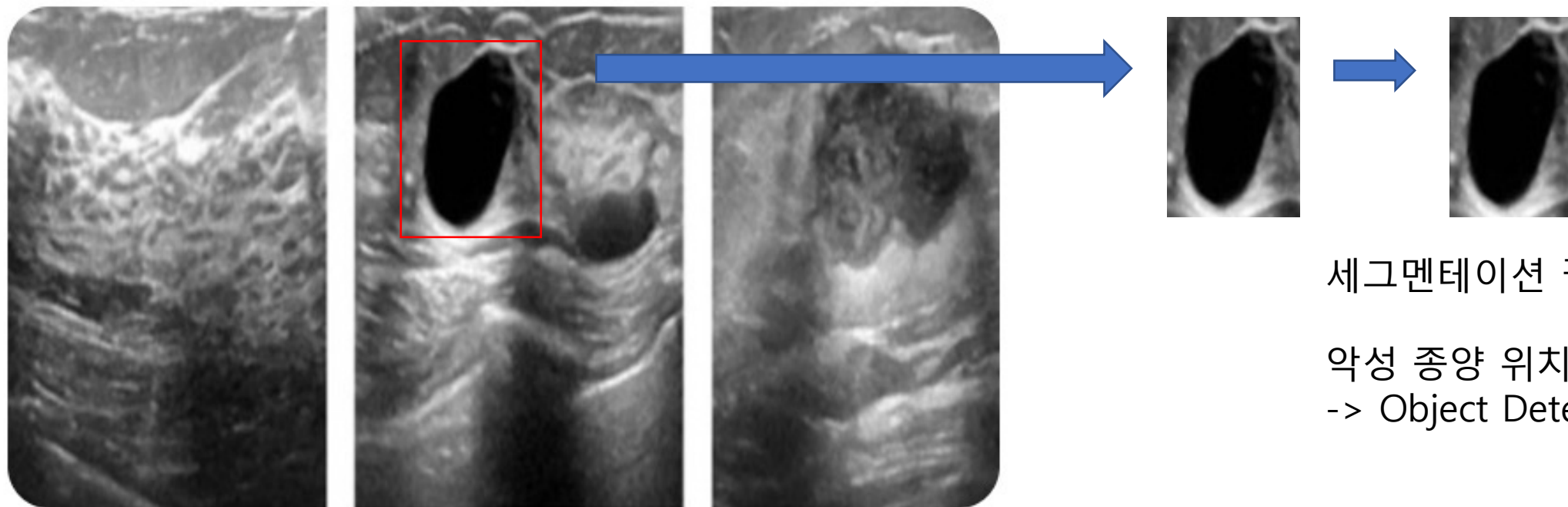
Breast Ultrasound Images Dataset



유방 초음파 데이터 세트는 정상, 양성 및 악성 영상의 세 가지 클래스로 분류됩니다.

Breast Ultrasound Images Dataset

Image -> 양성 음성 판별하는 Classification 분류 -> 양성 악성 종양 찾기 필요 -> Object Detection
세그멘테이션 필요하다



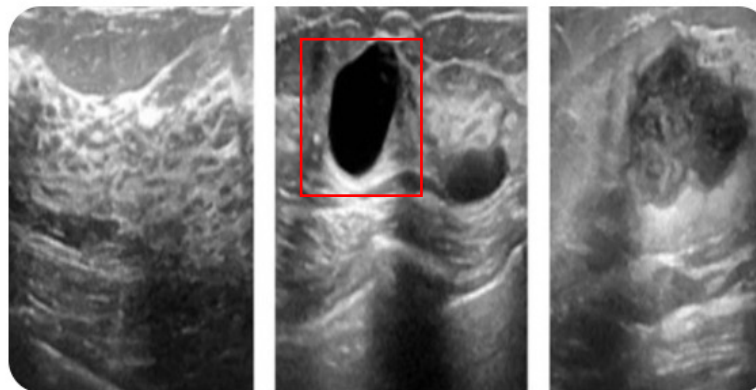
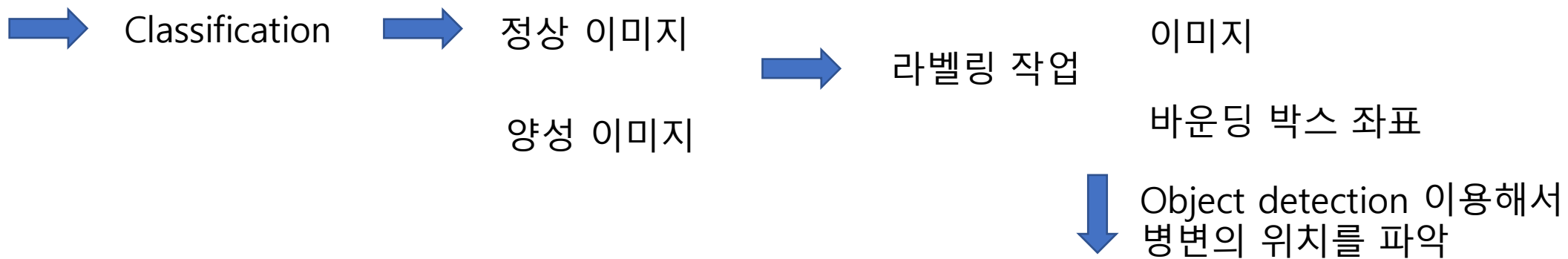
세그멘테이션 필요하다

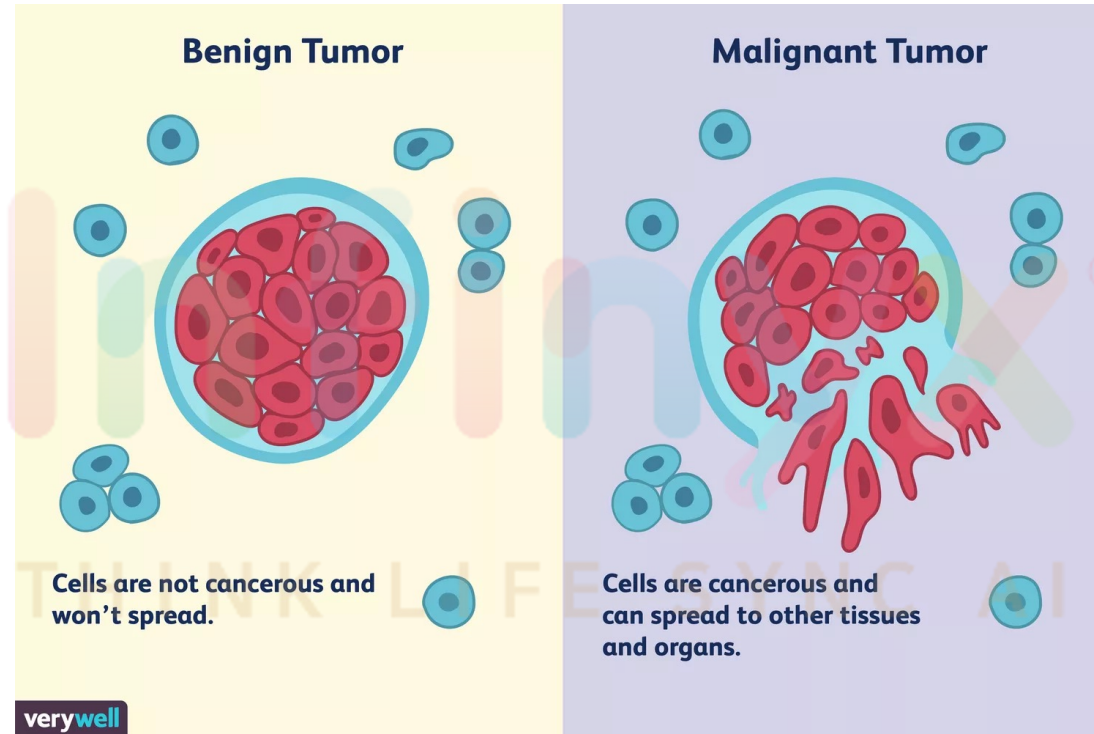
악성 종양 위치만 파악
-> Object Detection

Breast Ultrasound Images Dataset

이미지 분류 정상, 양성 폴더 [정상 5천건 / 비정상 5천건]

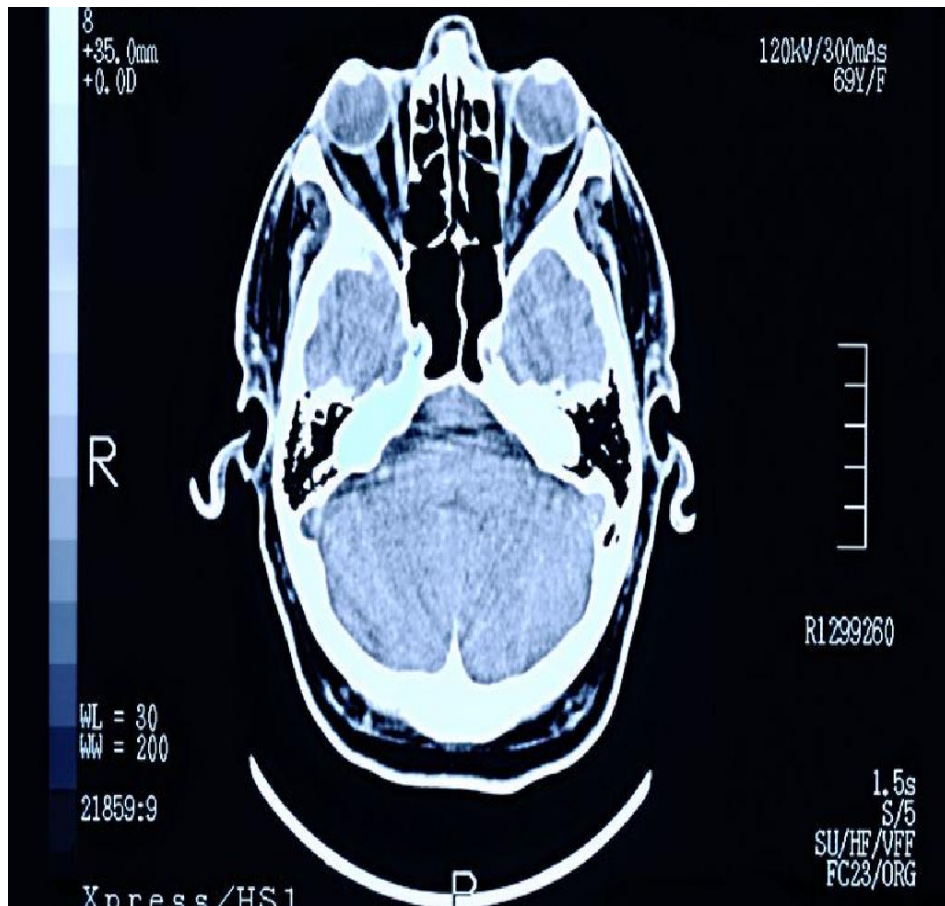
추가적인 문제 -> 박스처리가 되지 않은 이미지 10만건 (양성 정상 분류 x)





양성 및 악성 사례를 정확하게 식별하는 것이 목적이다. 양성종양은 주변 조직을 침범하거나 몸 주위에 퍼지지 않는 종양이다. 악성종양은 주변 조직을 침범하거나 몸 주위에 퍼질 수 있는 종양이다.

DICOM 파일이란 무엇입니까?



DICOM은 의학 분야의 Digital Imaging and Communications의 약자입니다.

이 형식의 파일은 DCM 또는 DCM30 (DICOM 3.0) 파일 확장자로 저장 될 가능성이 높지만 일부 파일 확장자는 전혀 없을 수 있습니다.

DICOM은 통신 프로토콜이자 파일 형식입니다. 즉, 초음파 및 MRI 이미지와 같은 의료 정보를 환자의 정보와 함께 하나의 파일에 저장할 수 있습니다.

이 형식은 모든 데이터가 함께 유지되도록하며 DICOM 형식을 지원하는 장치간에 정보를 전송할 수있는 기능을 제공합니다.

Pydicom 모듈

참고 : [pydicom github](#)

...

pydicom is a pure Python package for working with DICOM files. It lets you read, modify and write DICOM data in an easy "pythonic" way.

As a pure Python package, *pydicom* can run anywhere Python runs without any other requirements, although if you're working with *Pixel Data* then we recommend you also install *Numpy*.

...

```
# Using pip:
pip install pydicom
```

```
# Using conda:
conda install -c conda-forge pydicom
```

• dicom 파일 읽는 법

```
dcm = pydicom.dcmread(filename)
pirt(dcm)
```

```

>>> Dataset.file_meta
(0002, 0000) File Meta Information Group Length   UC: 18
(0002, 0001) File Meta Information Version         OB: '3.10001001'
(0002, 0002) Media Storage SOP Class UID          UC: Ultrasound Multi-frame Image Storage
(0002, 0003) Media Storage SOP Instance UID       UC: 999.999.133.1996.1.1808.1.6.29
(0002, 0004) Transfer Syntax UID                  UC: MRLE Lossless
(0002, 0012) Implementation Class UID            UC: 999.999.332346

(0008, 0000) Group Length                         UC: 226
(0008, 0001) Image Type                           CS:
(0008, 0002) SOP Class UID                         UC: Ultrasound Multi-frame Image Storage
(0008, 0003) SOP Instance UID                     UC: 999.999.133.1996.1.1808.1.6.29
(0008, 0020) Study Date                           DA: '19940823'
(0008, 0004) Study Time                           TM: '121504.0'
(0008, 0050) Accession Number                     SN:
(0008, 0006) Modality                             CS: 'US'
(0008, 0070) Manufacturer                         LO: ''
(0008, 0000) Referring Physician's Name           PR: ''
(0008, 1030) Study Description                     LO: 'Echocardiogram'
(0008, 1036) Series Description                   LO: '2AS FEM2'
(0018, 0000) Group Length                         UC: 70
(0018, 0001) Patient's Name                       PR: 'RUBO DEMO'
(0018, 0002) Patient's ID                         LO: '123456789'

```


학습의 기본 구성 OVERVIEW

1. Libraries, Seed
2. PyTorch Dataset
3. Neural Nets – Resnet 50 and vgg
4. Training
5. Eval



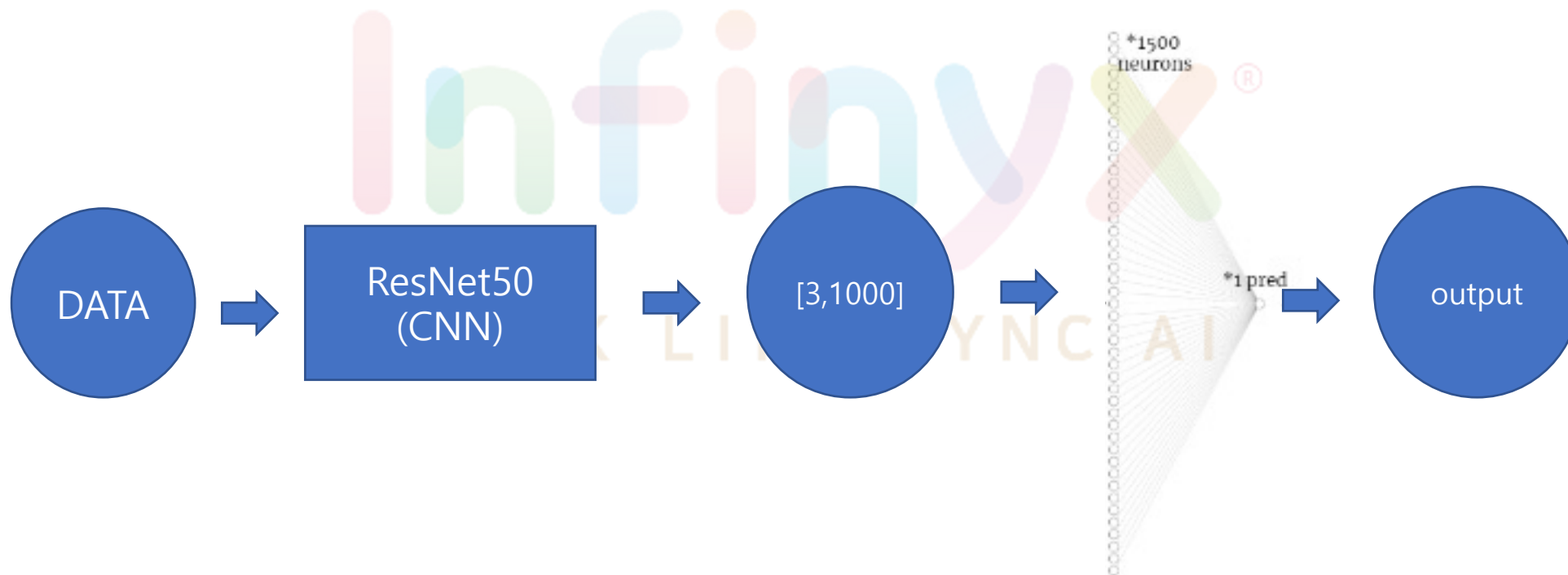
학습의 기본 구성 OVERVIEW

1. Libraries, Seed

```
def set_seed(seed = 1234):  
    # Sets the seed of the entire notebook so results are the same every time we run  
    # This is for REPRODUCIBILITY  
    np.random.seed(seed)  
    random.seed(seed)  
    torch.manual_seed(seed)  
    torch.cuda.manual_seed(seed)  
    # When running on the CuDNN backend, two further options must be set  
    torch.backends.cudnn.deterministic = True  
    # Set a fixed value for the hash seed  
    os.environ['PYTHONHASHSEED'] = str(seed)  
  
set_seed()  
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')  
print('Device available now:', device)
```

학습의 기본 구성 OVERVIEW

How is ResNet50() working ?





감사합니다.

THINK LIFE SYNC AI

Infinyx®