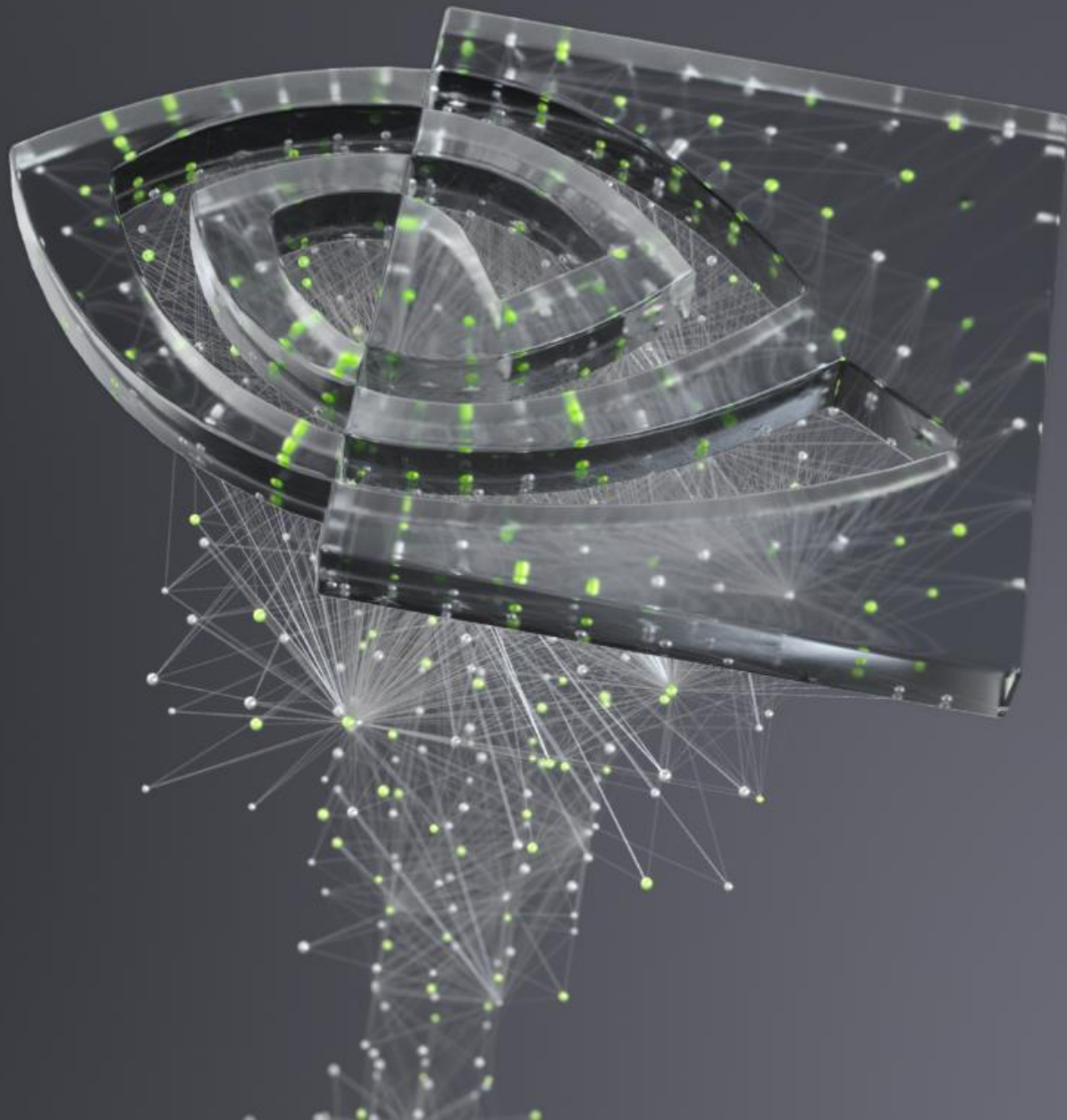




DEEP
LEARNING
INSTITUTE

딥러닝의 기초

4부: 데이터 증강 및 배포



목차

1부: 딥러닝 소개

2부: 뉴럴 네트워크의 트레이닝 방식

3부: CNN(Convolutional Neural Network)

4부: 데이터 증강 및 배포

5부: 사전 트레이닝된 모델

6부: 고급 아키텍처

목차 – 4부

- 데이터 증강
- 모델 배포

HANDS-ON 요약

분석

- CNN으로 검증 정확도를 높임
- 여전히 트레이닝 정확도가 검증 정확도보다 높음

솔루션

- 정제 데이터가 더 나은 예시를 제공
- 데이터세트의 다양성이 모델의 일반화에 도움이 됨





데이터 증강
DATA AUGMENTATION

데이터 증강 (DATA AUGMENTATION)



이미지 반전 (IMAGE FLIPPING)

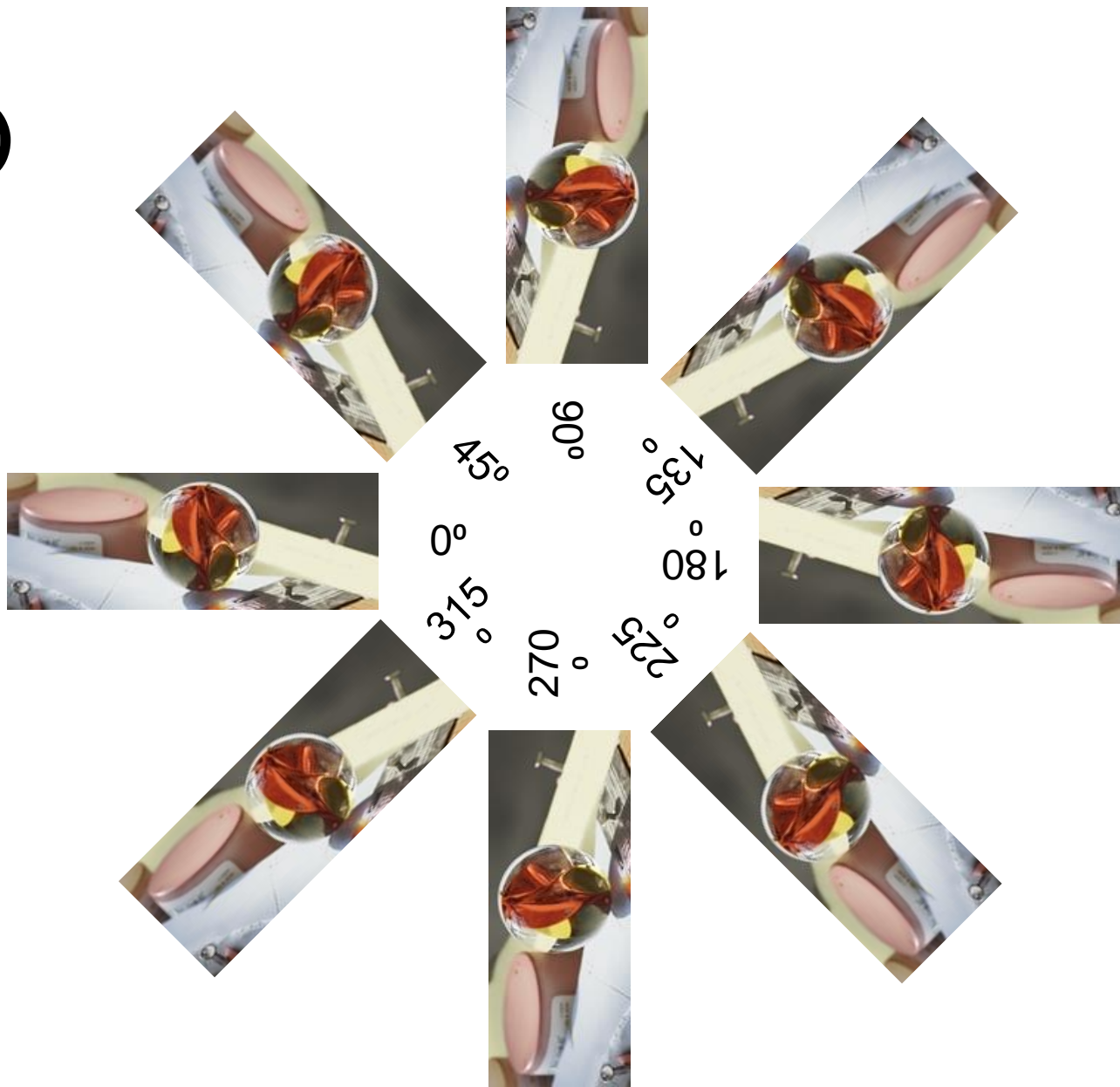
가로 반전



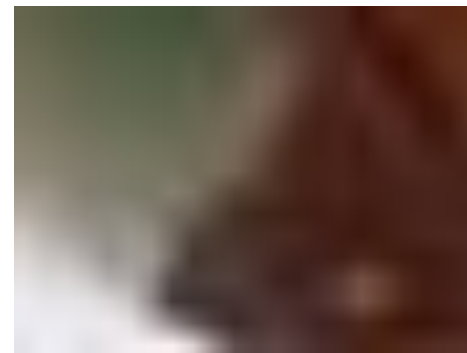
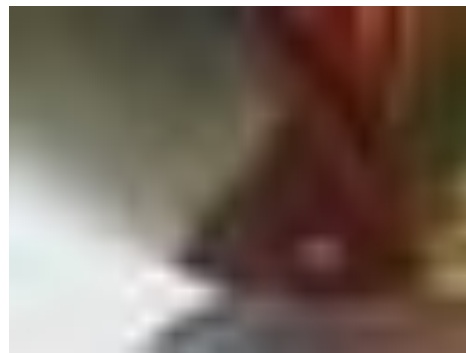
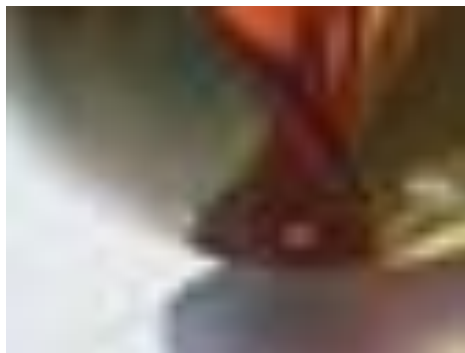
세로 반전



회전 (ROTATION)



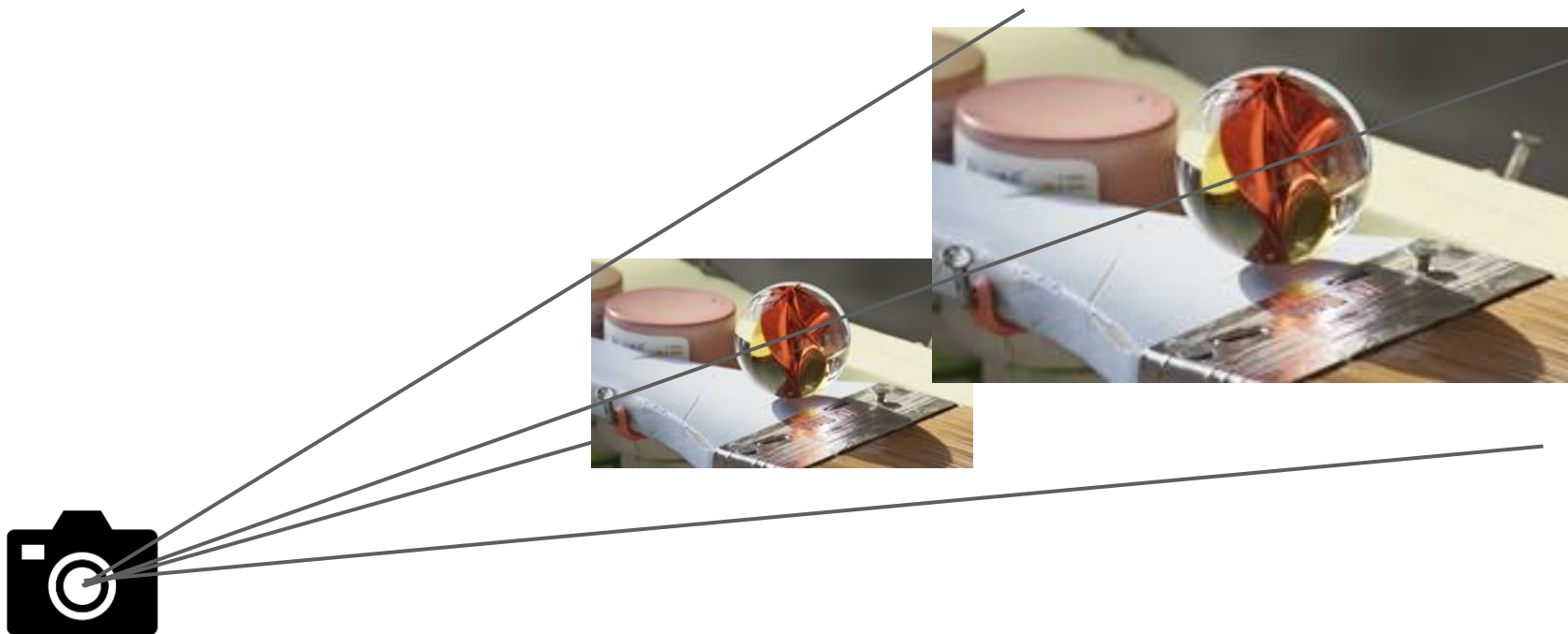
확대/축소 (ZOOMING)



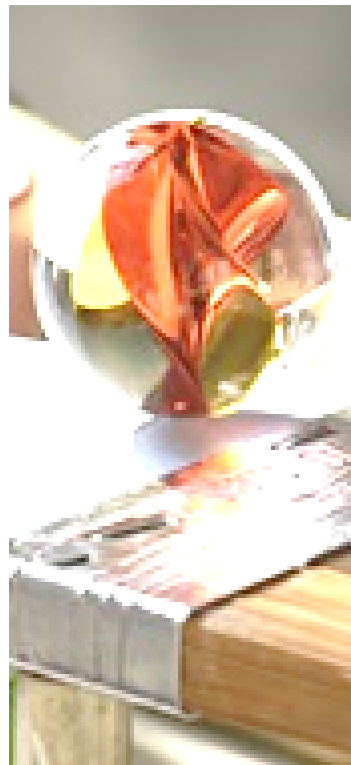
너비 및 높이 이동



호모그래피 (HOMOGRAPHY)



밝기 (BRIGHTNESS)



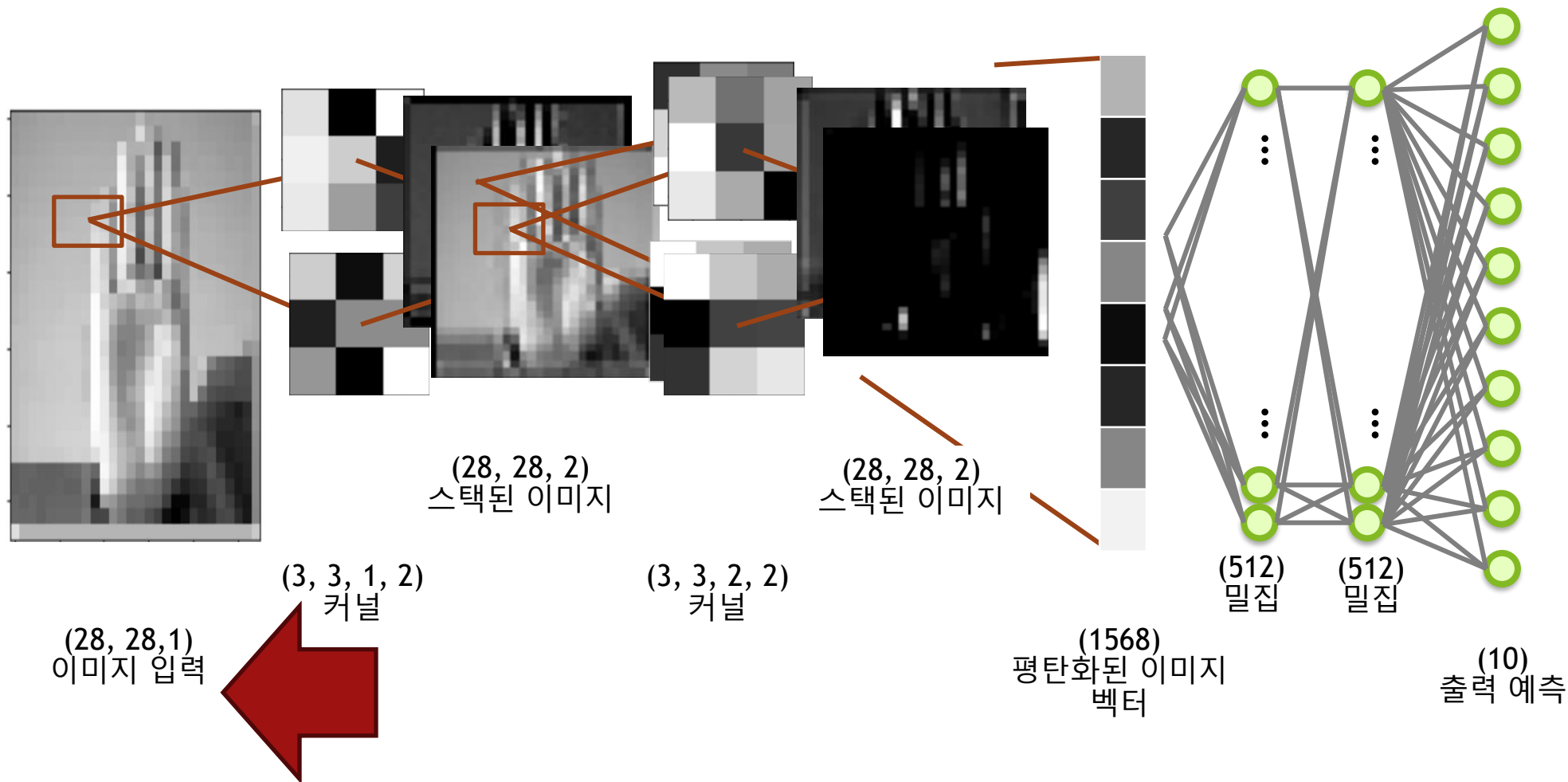
채널전환 (CHANNEL SHIFTING)





모델 배포 MODEL DEPLOYMENT

모델 배포 (MODEL DEPLOYMENT)



모델 배포 (MODEL DEPLOYMENT)

트레이닝
배치 입력



합성곱

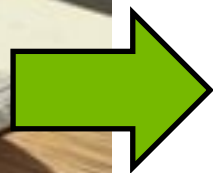
Max Pooling(최대 풀링)

...

모델 배포 (MODEL DEPLOYMENT)



(287, 433, 3)



(220, 155, 3)



(220, 155, 1)



(1, 220, 155, 1)

크기 조정

회색조

'배치'

tensorflow 이미지 데이터 증강 클래스 : ImageDataGenerator

- `flow_from_directory(directory)`
 - 폴더(directory) 형태로 된 데이터를 바로 가져와서 사용 가능
 - 무작위 샘플에 대해 트레이닝 가능
- `flow(x, y)`
 - 증강 데이터의 배치 생성(iterator)
 - `img_iter = datagen.flow(x_train, y_train, batch_size=batch_size)`
- 데이터 증강 설정
 - ```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(
 rotation_range=10, # randomly rotate images in the range (degrees, 0 to 180)
 zoom_range=0.1, # Randomly zoom image
 width_shift_range=0.1, # randomly shift images horizontally (fraction of total width)
 height_shift_range=0.1, # randomly shift images vertically (fraction of total height)
 horizontal_flip=True, # randomly flip images horizontally
 vertical_flip=False, # Don't randomly flip images vertically
)
```
- API : [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator)
- 참고 : <https://laheepapa.tistory.com/63>



# 데이터 증강 클래스 : ImageDataGenerator

- **트레이닝 데이터셋을 생성기에 맞추기**

```
datagen.fit(x_train)
```

- **증강 데이터셋으로 트레이닝**

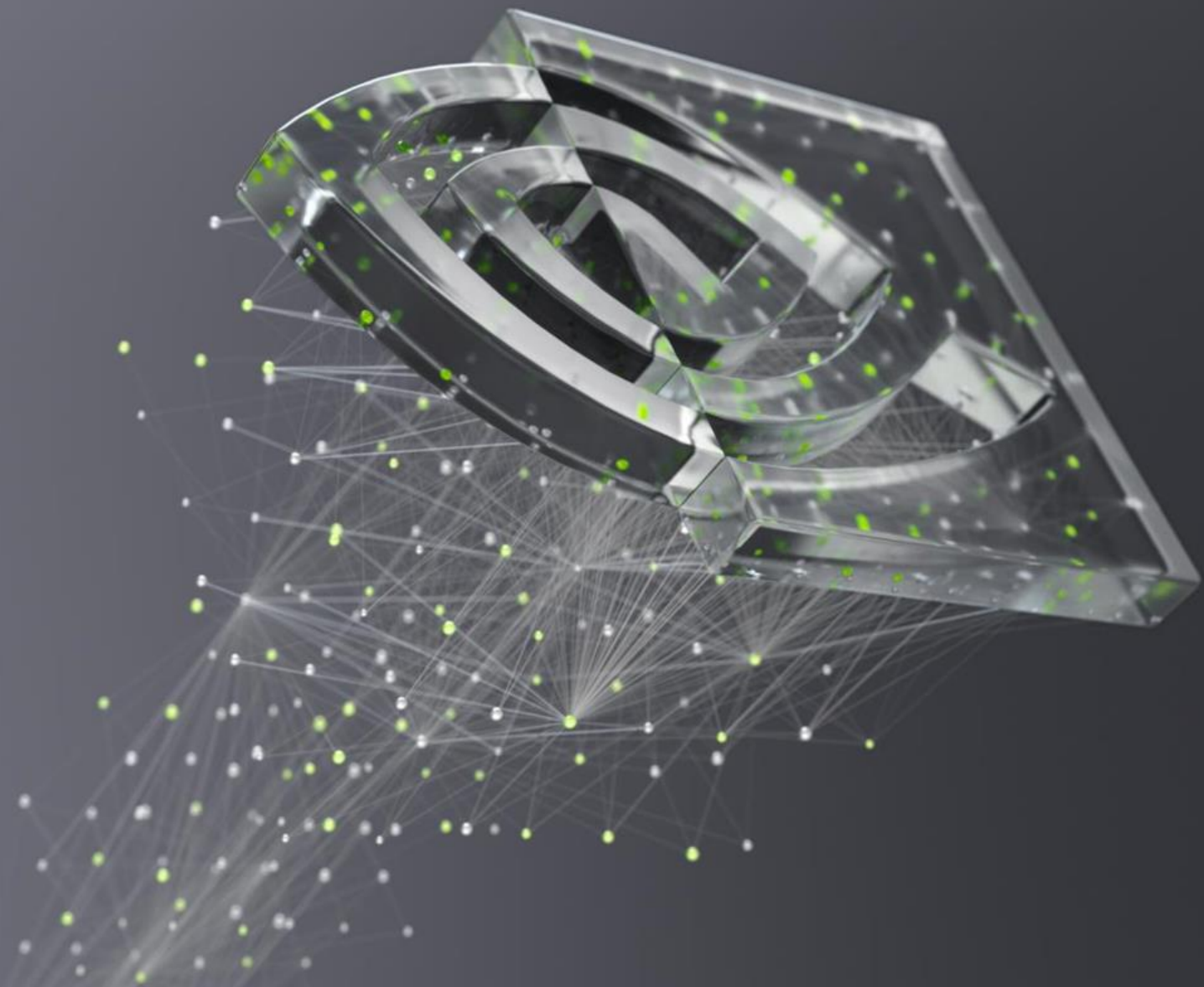
```
batch_size = 32
```

```
img_iter = datagen.flow(x_train, y_train, batch_size=batch_size)
```

```
model.fit(img_iter,
 epochs=20,
 steps_per_epoch=len(x_train)/batch_size,
 validation_data=(x_valid, y_valid))
```



시작하겠습니다!



DEEP  
LEARNING  
INSTITUTE