

# 자동차인공지능 2주차 과제

항공우주공학과

12190476 김호윤

# 문제정의 및 문제 해결 방식

## HW1: data augmentation



- 4중 for(while)문을 사용하여 data augmentation을 진행하시오.
  - Blur → 2
    - Filter size: 3x3
    - Filter size: 5x5
  - Noise → 10
    - Gaussian (Normal) noise: mean=0, std: 1~20, 2씩 증가
  - Translation → 10
    - (0,0) → (tx, ty)
    - tx:1~10, 1씩 증가
    - ty:1~10, 1씩 증가
  - Rotation → 11
    - (-5 ~5) degree, 1도 간격으로 증가

### 문제정의

- Data augmentation을 수행
- OpenCV를 활용해, 주어진 이미지 데이터셋을, 수만장의 이미지로 확장

### 문제 해결 방식

- 이를 위해 Blurring, Noise, Translation, Rotation 등 다양한 기능을 OpenCV의 각종 메서드들을 활용해 구현
- 구체적으로, cv2의 blur, warpAffine, getRotation2D 등의 메서드를 활용
- Gaussian Noise 획득을 위해 Numpy의 random 메서드도 사용

# 작성한 핵심 코드 설명(1)

```
for blur_option in range(2):
    if blur_option == 0:
        kernel = kernel_small
        blur_val = 3
    else:
        kernel = kernel_large
        blur_val = 5

    blurred = cv2.blur(orig_img, kernel)
    blur_path = os.path.join(save_folder, f'image_{img_idx}_blur_{blur_val}.jpg')
    cv2.imwrite(blur_path, blurred)
```

- 이미지에 대한 blur 처리를 하기 위해서 cv2의 blur 메서드를 사용

# 작성한 핵심 코드 설명(3)

```
for noise_idx in range(10):
    cur_std = 1 + noise_idx * 2
    noise = np.random.normal(n_mean, cur_std, (height, width, channels))
    noisy = blurred.astype(np.float32) + noise
    noise_path = os.path.join(save_folder, f'image_{img_idx}_blur_{blur_val}_noise_{noise_idx}.jpg')
    cv2.imwrite(noise_path, noisy)
```

- 이미지에 대한 Gaussian Noise 처리를 하기 위해서 Numpy와 cv2를 활용
- Numpy의 random 메서드를 통해서 noise를 생성하고, 이를 이미지에 적용함
- 이를 통해 최종적으로 Gaussian Noise가 포함된 이미지들을 획득

# 작성한 핵심 코드 설명(1)

```
for trans_idx in range(10):
    shift_x = trans_idx
    shift_y = trans_idx
    trans_mat = np.float32([[1, 0, shift_x], [0, 1, shift_y]])
    translated = cv2.warpAffine(noisy, trans_mat, (width, height))
    trans_path = os.path.join(save_folder, f'image_{img_idx}_blur_{blur_val}_noise_{noise_idx}_trans_{trans_idx}.jpg')
    cv2.imwrite(trans_path, translated)
```

- 이미지를 Translation 하기 위해서 cv2의 warpAffine 메서드를 활용해 translation을 진행
- Translation Matrix는 Numpy를 활용해 배열을 생성
- Cv2는 기본적으로 이미지를 Numpy 배열로 다루기 때문에 Numpy를 사용해도 무방함

# 작성한 핵심 코드 설명(4)

```
for rot_idx in range(11):  
    angle = -5 + rot_idx  
    rot_mat = cv2.getRotationMatrix2D((width / 2, height / 2), angle, 1)  
    rotated = cv2.warpAffine(translated, rot_mat, (width, height))  
    rot_path = os.path.join(  
        save_folder, f'image_{img_idx}_blur_{blur_val}_noise_{noise_idx}_trans_{trans_idx}_rot_{angle}.jpg'  
    )  
    cv2.imwrite(rot_path, rotated)
```

- 이미지에 Rotation을 주기 위해서, cv2의 getRotationMatrix2D 메서드를 사용해 Rotation Matrix를 설정
- 이후 warpAffine 메서드를 사용해서 이미지를 translation하고 rotation 함

# 결과물 (1)

- 최종적으로 21,798장의 이미지가 생성되었음
- 생성된 이미지의 예시는 다음과 같음



가장 왼쪽이 원본이며, blur, gaussian noise, translation, rotation 등의 연산을 적용했다.

## 결과물 (2)

- 추가적인 예시는 다음과 같음



가장 왼쪽이 원본이며, blur, gaussian noise, translation, rotation 등의 연산을 적용했다.



# 레퍼런스

- 자동차인공지능 2주차 Numpy 강의자료
- 자동차인공지능 2주차 OpenCV 강의자료