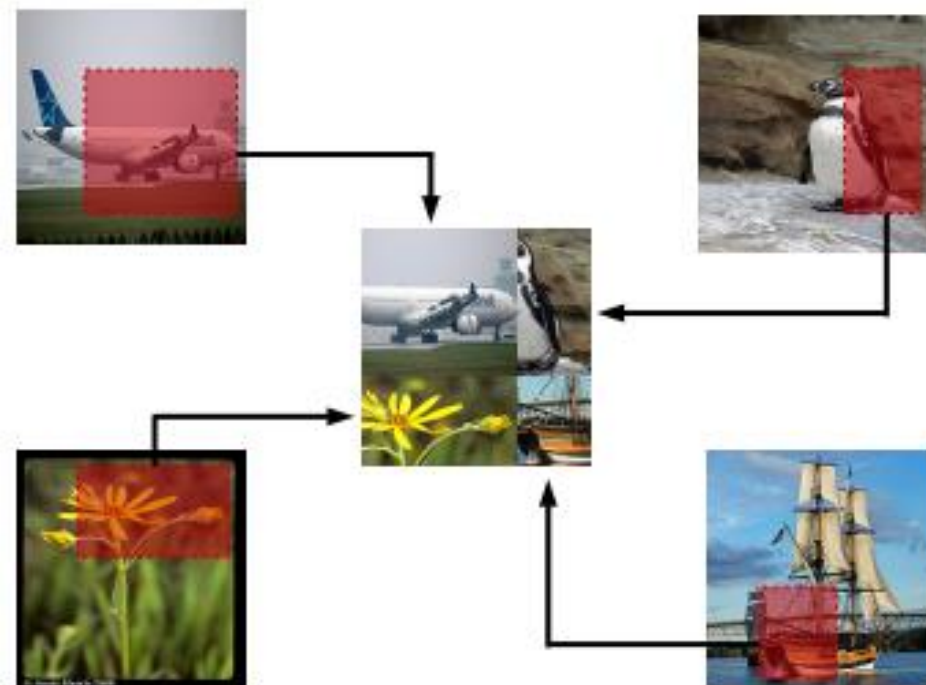


Random Image Cropping and Patching

- RICAP은 4개의 이미지를 random crop한 후 한개의 이미지로 합성하는 Data Augmentation 이다.
- 논문의 WideResNet50-2을 ResNet34에 적용시켰다.



$$\begin{aligned} v_{c_{ricap}} &= RICAP_{caption}(v_{c_m}, v_{c_n}, v_{c_o}, v_{c_p}) \\ &:= \sum_{k=\{m,n,o,p\}} W_k v_{c_k}, \\ &= \sum_{k=\{m,n,o,p\}} W_k GRU(c_k), \end{aligned}$$

Runs	1 st Run
Acc@1(%)	78.840
Acc@5(%)	94.720

Random Flare Data Augmentation

제안 방법(motivation)

- 렌즈 플레어는 강한 빛이 카메라의 렌즈에 산란되며 이미지에 남겨진 허상으로, 이미지와 영상에서 렌즈 플레어 현상은 자주 목격할 수 있다.

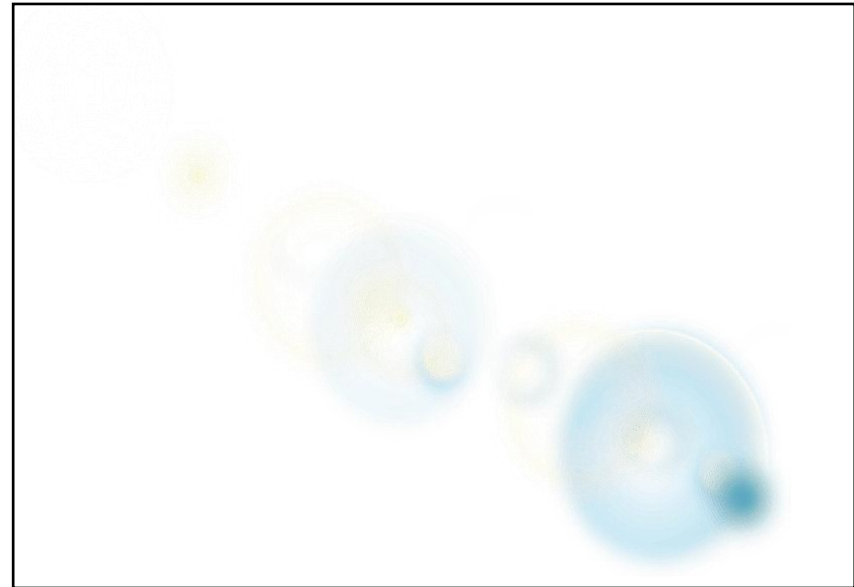
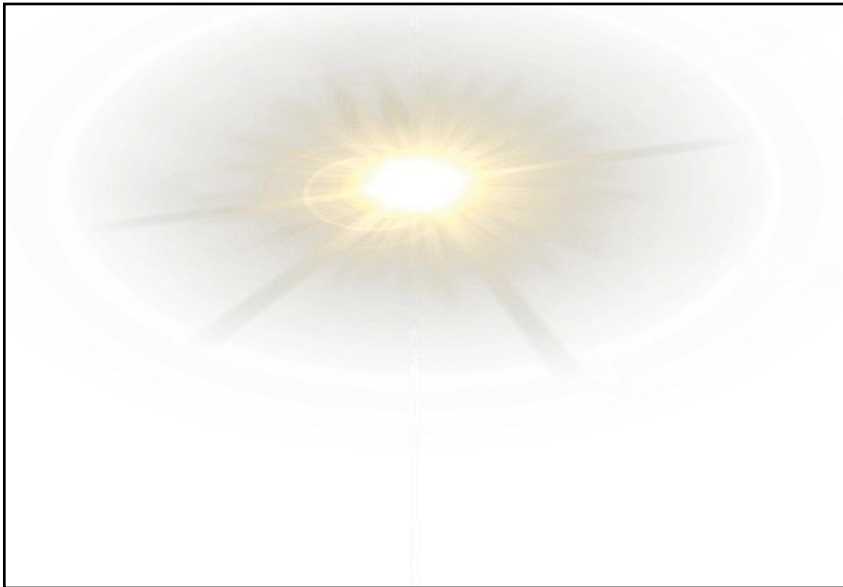


제안 방법(핵심 아이디어)

- 실제 상황에서 플레어가 발생하더라도 이미지 속 객체의 본질은 변화하지 않는다.
- 예상 효과
 - 이미지가 과하게 변형되지 않아 Data Augmentation에 의한 학습 오류가 적을 것이다.
 - 플레어를 적용시키면 원의 투명도와 색에 따라 부분적으로 Color Filtering류의 Augmentation 효과도 얻을 수 있을 것으로 예상된다.

제안 방법(구현)

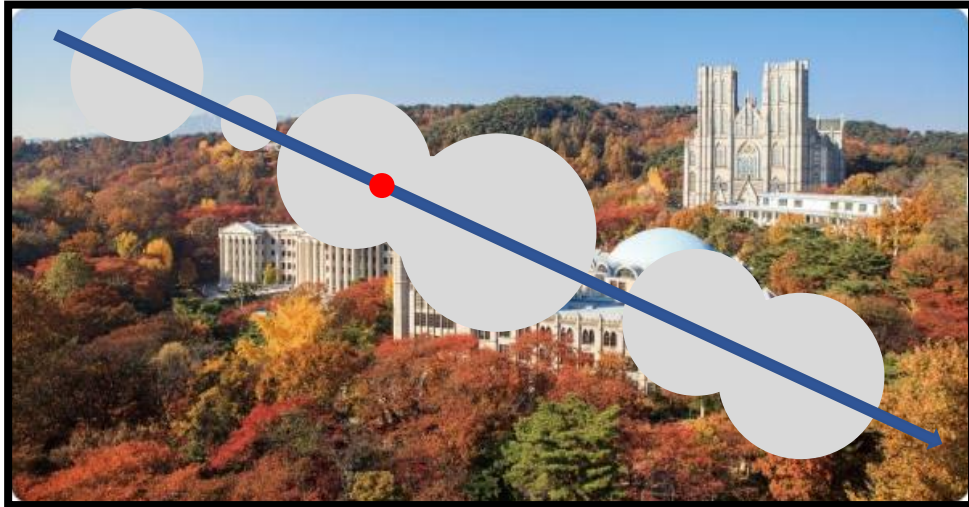
- 모든 이미지에 다운 받은 flare 이미지를 합성한다?
→ 여러가지 flare를 구하기 어렵고, random성이 떨어진다.



제안 방법(구현)

- 이미지를 복사한 overlay 레이어를 생성한다.
- overlay 위의 한 point를 잡아 top-left corner와 point를 연결하는 직선의 gradient를 구한다.
- 구한 직선 위에 임의의 개수만큼 임의의 색과 반경을 가지는 원을 그린다.
- Overlay를 일정 비율의 투명도로 원본 이미지와 합성한다.
- train_transform 의 첫 단계로 Flare Augmentation을 추가한다.

제안 방법(구현)



제안 방법(구현)

```
img = np.array(img) #PIL image를 numpy array로 변환
h, w, _ = img.shape #image의 height, width 추출

overlay = img.copy() #image와 합칠 overlay 레이어 선언

max_r = min(h,w)//6 #원의 최대 반지름을 제한

gradient = np.random.randint(h)/np.random.randint(1,w) #top-left corner와 random pixel의 gradient
alpha = 0.3 #overlay의 투명도, 즉 원들의 투명도

for _ in range(self.n_circles): #선언한 flare circle의 수 만큼 반복
    x = np.random.randint(w) #random한 x좌표 선택
    center = (int(gradient*x), x) #원의 중심 좌표 설정

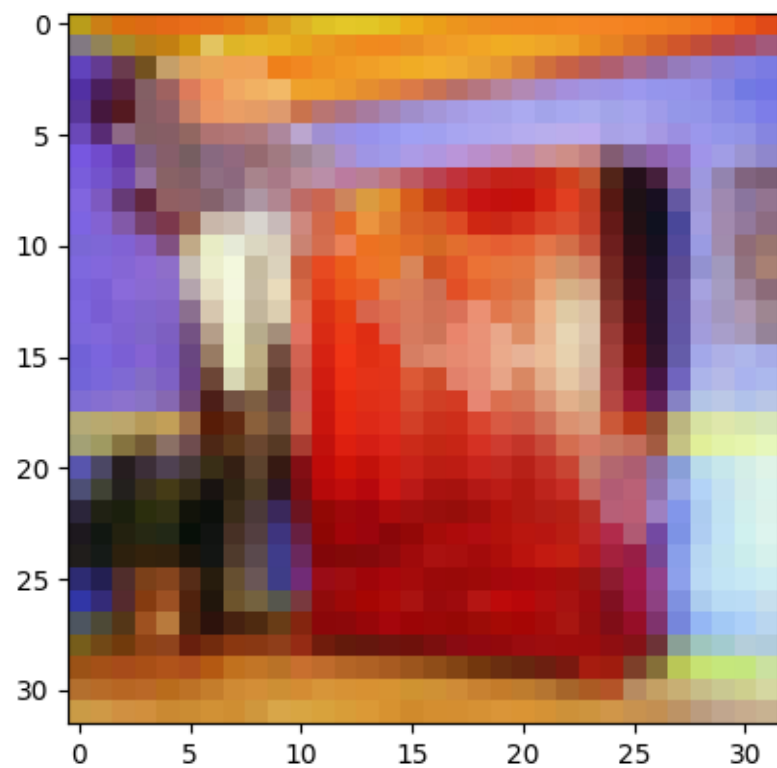
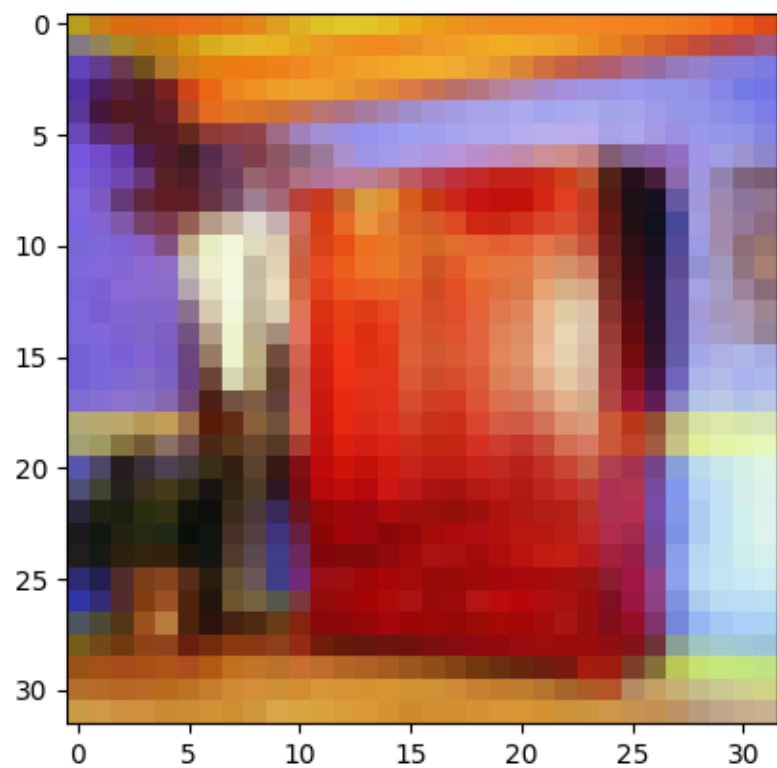
    c = np.random.randint(190,255) #원의 색(grey) 설정
    cv2.circle(overlay, center, np.random.randint(max_r), (c,c,c), -1) #overlay 레이어 위에 원을 그림

img = cv2.addWeighted(overlay, alpha, img, 1-alpha, 0) #overlay에 투명도 부여 후 원본 image와 합성
img = Image.fromarray(img) #numpy array를 PIL image로 변환
```

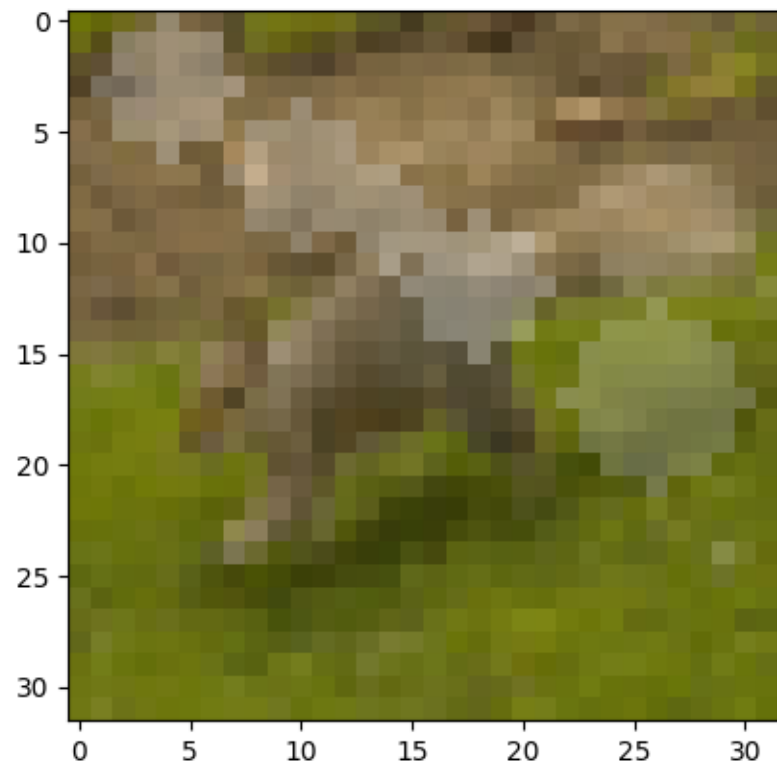
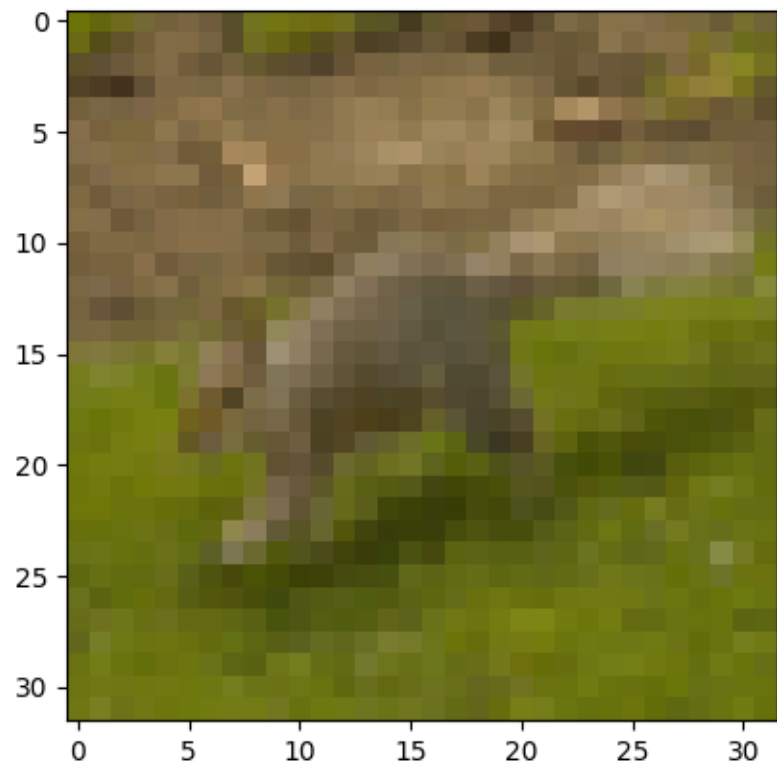

실험 조건

- Flare의 수는 16개, 투명도(alpha)는 30%로 설정
- Flare의 색은 모든 이미지에 어울리게 적용시키기 위해 rgb에 동일한 random 값을 부여
- cutout은 flare로 변경, n_holes와 length는 주석 처리
- 그 외 parameter는 모두 기존 Cutout과 동일하게 유지

실험 결과



실험 결과



실험 결과

- Accuracy가 Cutout이나 RICAP을 넘지 못했다.
- 플레어 모양이나 색에 변화를 주어서 더 realistic한 렌즈 플레어를 적용시킨다면 model의 accuracy를 보완할 수 있을 것이라 예상한다.

Runs	1 st Run	2 nd Run	3 rd Run
Acc@1(%)	77.92	78.050	78.120
Acc@5(%)	-	94.180	94.040

