

PLASS-NIA 인공지능학습데이터구축사업 (2021)

4팀(김도현, 전은성)

목차

1. 멀티 GPU 사용 결과
2. 모델 테스트
3. 향후 계획



1. 멀티 GPU 사용 결과

```
1 import os
2 os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID" # Arrange GPU devices starting from 0
3 os.environ["CUDA_VISIBLE_DEVICES"]="0,1,2" # Set the GPUs 0, 1 and 2 to use
4
```

- train.py에 멀티 GPU 사용 코드 추가
- 실제 학습시간 단축에 영향이 없음

0	Tesla V100-PCIE...	Off	00000000:27:00.0	Off	0
N/A	40C	P0	188W / 250W	31427MiB / 32510MiB	71% Default
					N/A
1	Tesla V100-PCIE...	Off	00000000:28:00.0	Off	0
N/A	32C	P0	35W / 250W	309MiB / 32510MiB	0% Default
					N/A
2	Tesla V100-PCIE...	Off	00000000:43:00.0	Off	0
N/A	30C	P0	36W / 250W	309MiB / 32510MiB	0% Default
					N/A

Processes:						
GPU	GI	CI	PID	Type	Process name	GPU Memory
	ID	ID				Usage
0	N/A	N/A	61195	C	python	31423MiB
1	N/A	N/A	61195	C	python	305MiB
2	N/A	N/A	61195	C	python	305MiB

1. 멀티 GPU 사용 결과

```
(ENVC3D) NIA_AI_DATASET_2021-C3D@G292-Z40-00:~/C3D_model1$ CUDA_VISIBLE_DEVICES=0,1,2 python 5.\ train.py
```



- 터미널에서 멀티 GPU 사용을 하는 명령어로 train.py 학습
- 실제 학습시간 단축에 영향이 없음

0	Tesla V100-PCIE...	Off	00000000:27:00.0	Off	0
N/A	42C	P0	142W / 250W	31427MiB / 32510MiB	61% Default
					N/A
1	Tesla V100-PCIE...	Off	00000000:28:00.0	Off	0
N/A	33C	P0	35W / 250W	309MiB / 32510MiB	0% Default
					N/A
2	Tesla V100-PCIE...	Off	00000000:43:00.0	Off	0
N/A	30C	P0	36W / 250W	309MiB / 32510MiB	0% Default
					N/A

Processes:						
GPU	GI	CI	PID	Type	Process name	GPU Memory
	ID	ID				Usage
0	N/A	N/A	62754	C	python	31423MiB
1	N/A	N/A	62754	C	python	305MiB
2	N/A	N/A	62754	C	python	305MiB

2. 모델 테스트 (Top-5)

```
top_inds = pred[0].argsort()[::-1][:5]

if width>500:
    pont=2
    pos=80
else:
    pont=0.5
    pos=20

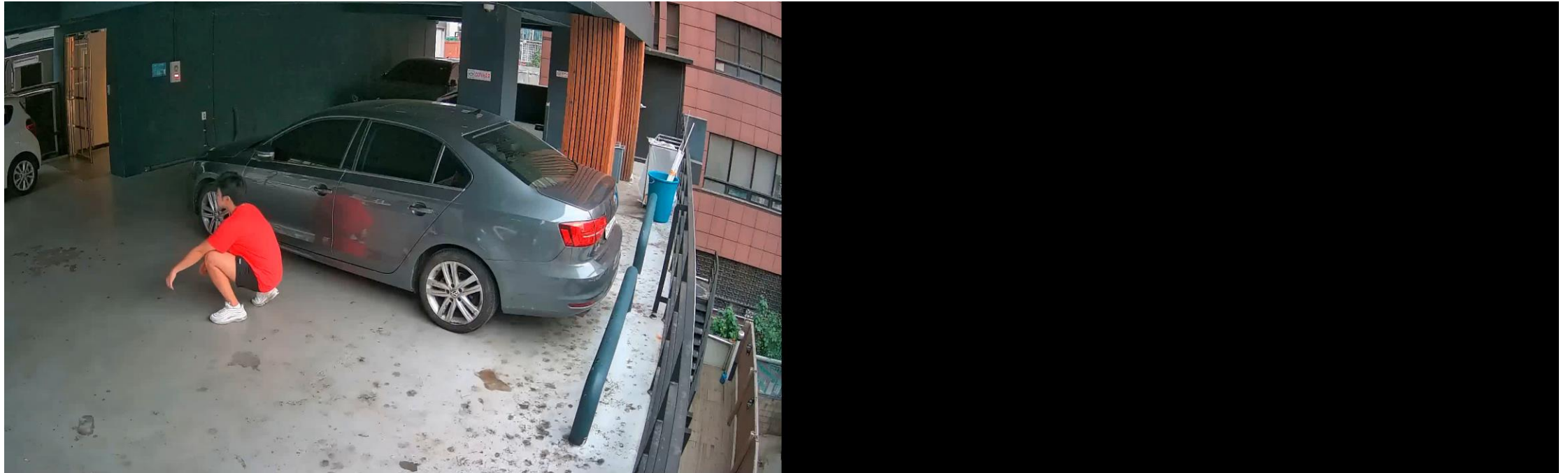
cv2.putText(img, "corr" +os.path.basename(dir_name), (20, pos),
            cv2.FONT_HERSHEY_SIMPLEX, pont, (255, 255, 255), int(pont*2))
fw.write(str(frame_num)+" corr" +os.path.basename(dir_name)+'\n')

cv2.putText(img, "answ" +"%-13s"%class_names[label].split(' ')[-1].strip()+" prob: %.4f" % pred[0][label], (20, pos*2),
            cv2.FONT_HERSHEY_SIMPLEX, pont, (255, 255, 255), int(pont*2))
fw.write(str(frame_num)+" answ" +"%-13s"%class_names[label].split(' ')[-1].strip()+" prob: %.4f" % pred[0][label]+'\\n')

for i,j in enumerate(top_inds):
    cv2.putText(img,"top"+str(i+1)+' ' +"%-13s"%class_names[j].split(' ')[-1].strip()+" prob: %.4f" %pred[0][j], (20, pos*(3+i)),
                cv2.FONT_HERSHEY_SIMPLEX, pont, (255, 255, 255), int(pont*2))
    fw.write(str(frame_num)+" top"+str(i+1)+' ' +"%-13s"%class_names[j].split(' ')[-1].strip()+" prob: %.4f" % pred[0][j]+'\\n')
```

- 프레임별로 가장 높은 확률을 나타내는 종목을 출력하는 기존 코드에서 Top-5까지 종목을 출력하도록 코드를 변경
- index별 정확도를 저장하고 있는 pred 배열을 내림차순으로 정렬후 top 5의 index를 top_inds 저장
- 기존처럼 영상 왼쪽에 검출내용 출력방식으로는 전부 표기가 불가능을 판단
- 영상 오른쪽에 영상과 같은 크기의 공간을 만들어 검출내용 출력

2. 모델 테스트 (Theft)



C042_A31_SY30_P09_B03_01DBS.avi

- 초반 영상 정확도: 0.9 이상
- 중반 영상 정확도: 1.0
- 후반 영상 정확도: 1.0

2. 모델 테스트 (Stealing)



Stealing092_x264.avi

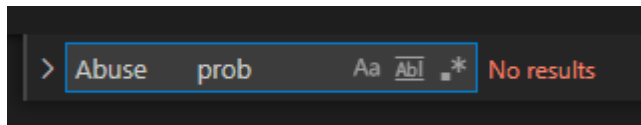
```
10856 corr) Stealing
10856 ans) Arson      prob: 0.8365
10856 top1) Arson      prob: 0.8365
10856 top2) Fighting   prob: 0.0512
10856 top3) Vandalism  prob: 0.0404
10856 top4) Burglary   prob: 0.0310
10856 top5) Stealing   prob: 0.0070
```

- 다른 종목 판별
- 특정 프레임에서 아주 낮은 정확도로 top5에 검출

2. 모델 테스트 (Abuse)

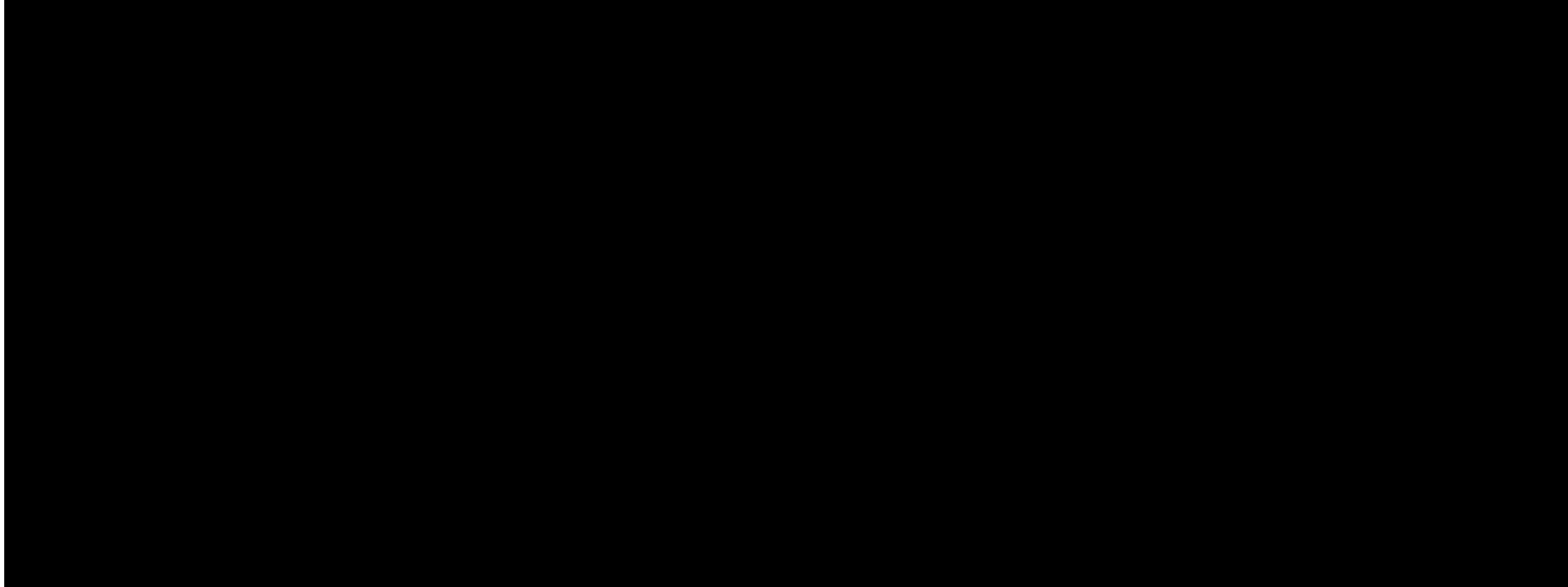


Abuse050_x264_test.avi

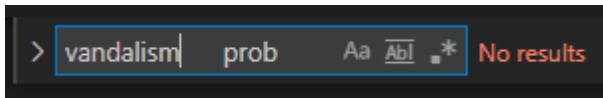


- 다른 종목 판별
- Abuse 판별 불가

2. 모델 테스트 (Vandalism)



Vandalism040_x264_test.avi



- 다른 종목 판별
- Vandalism 판별 불가

3. 향후 계획

- 멀티 GPU 사용률 최적화
- 모델 테스트 정확도 개선
- 개선 사항 수정
- 테스트 결과 확인

감사합니다

4팀(김도현, 전은성)