[졸업작품전 – 연구]

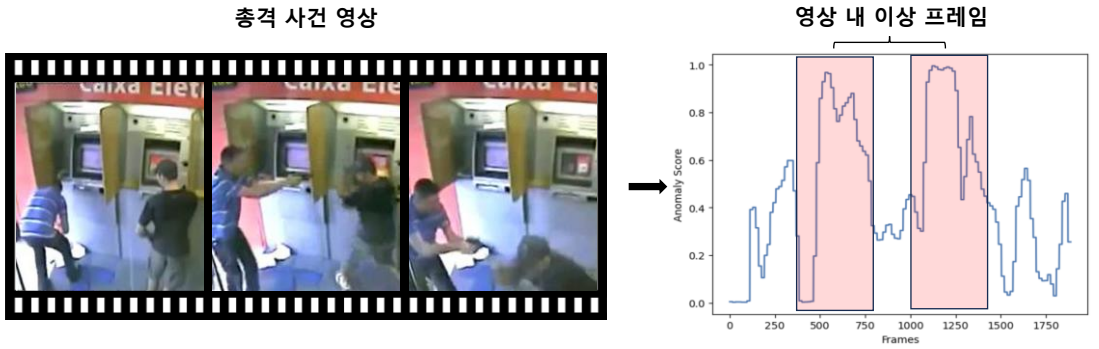
영상 기반 이상 상황 탐지를 위한 가상 데이터셋 구축과 도메인 적응

- 김성호: 전기공학과 2019060564
- 류세종: 융합전자공학부 2019076844

1. 연구 배경

1) 영상 기반 이상 상황 탐지

- 영상 내에서 발생하는 이상 상황을 탐지하는 기술
- CCTV 영상에서 이상 상황을 감지하는 것이 목표



2) 영상 기반 이상 상황 탐지 분야의 문제점

- 기존에 사용되던 데이터셋의 한계
  - 영상의 낮은 화질 (320 x 240) (UCF-Crime)
  - CCTV 영상과 다른 카메라 시점 (XD-Violence)
  - 폭력적인 데이터셋을 현실에서 구하거나 재현하기 어려움

⇒ **GTA**를 이용하여 높은 화질, CCTV 구도의 폭력 행동 **가상 데이터셋** 구축

⇒ **도메인 적응** 과정을 통해 가상 데이터와 현실 데이터 간의 차이 줄이기

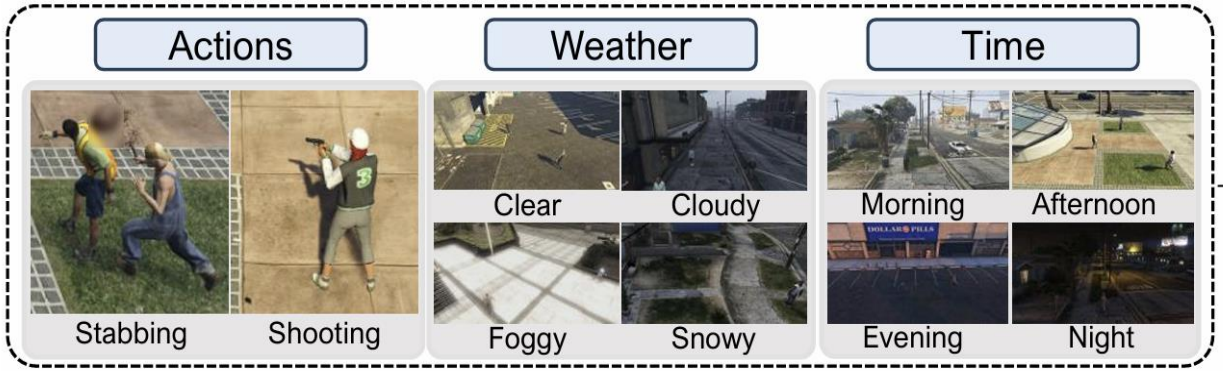
UCF-Crime 데이터셋

XD-Violence 데이터셋

2. 가상 데이터셋 구축

1) GTA를 이용한 폭력 행동 데이터셋 구축 (GTA-Crime)

- Grand Theft Auto V(GTA V), Rockstar Advanced Game Engine(RAGE), ScriptHook 활용
- GTA 게임에서 **칼부림(stabbing), 총격(shooting)**과 같은 폭력 사건 생성
  - 특정 시점에 칼로 찌르기나 총을 쏜 후 도망가도록 설계
- **다양한 배경 환경**에서 영상 생성
  - 게임 내의 75개의 장소
  - 랜덤 날씨 및 시간
  - 동일한 사건 다시점



Step 1. Set config　　Step 2. Save location & Assign ROI　　Step 3. Generate scene

2. 가상 데이터셋 구축

2) GTA-Crime 데이터셋

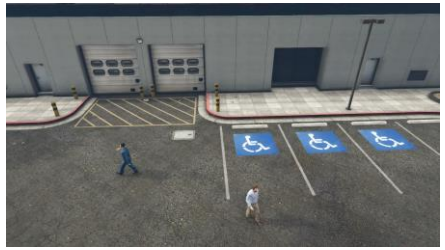
칼부림(Stabbing) 영상

road_2

beach_2

street_1

parkinglot_1

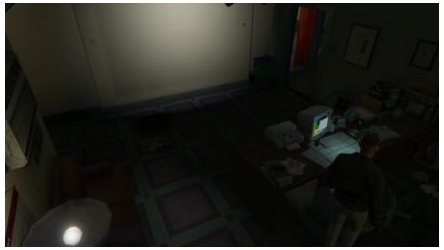


총격(Shooting) 영상

road_1

subway_4

in_5



clotheshop_2

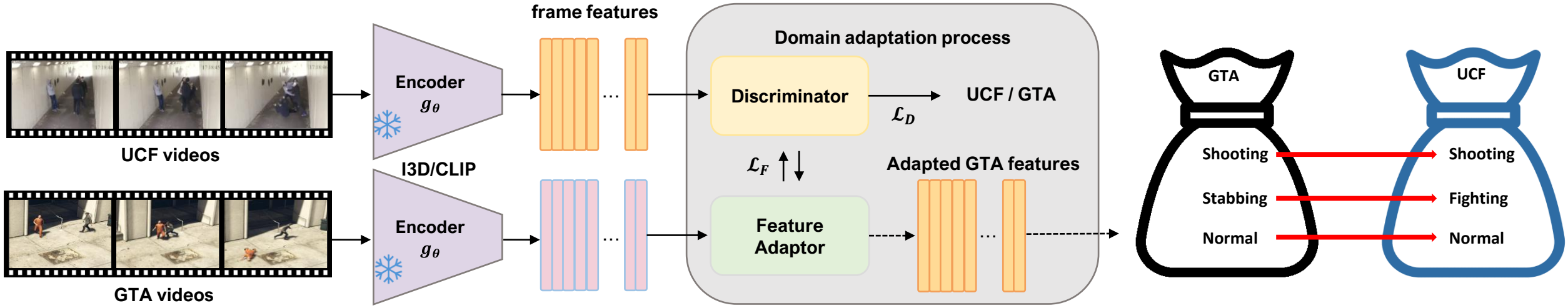
<GTA-Crime 데이터셋 구성>

category	# videos
Stabbing	124
Shooting	146
Normal	262

3. 도메인 적응

1) 적대적 학습을 이용한 피처 단위에서의 도메인 적응(domain adaptation, DA)

- 기존 방식인 비디오 단위에서의 도메인 적응은 **computational cost가 큼**
- **사전 학습된 I3D/CLIP 모델을** 사용하여 피처 추출 후 **피처 단위에서 도메인 적응**
- WGAN-GP loss 함수를 활용한 **적대적 학습** 사용
  - **Feature adaptor**: GTA 피처를 UCF 피처와 유사한 분포를 따르도록 변환
  - **Discriminator**: UCF 피처와 변환된 GTA 피처 구별
- 학습 시 GTA-Crime의 각 카테고리가 UCF-Crime의 각 카테고리에 대응되도록 함



3. 도메인 적응

2) 도메인 적응 결과 시각화

- t-SNE 알고리즘을 이용하여 GTA 피처(보라)와 UCF 피처(노랑) 분포의 시각화

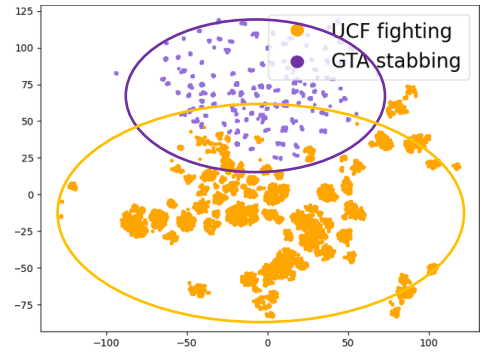
- 도메인 적응 이전
  - GTA 피처와 UCF 피처가 **구분되어** 있음

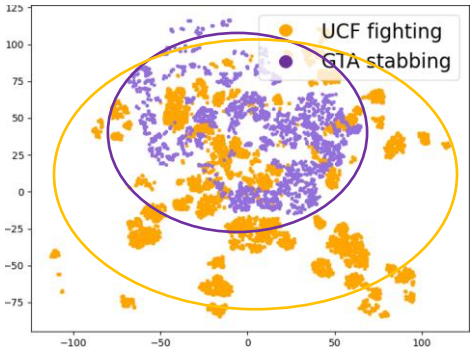
- 도메인 적응 이후
  - GTA 피처와 UCF 피처가 **섞여** 있음
  - GTA 피처의 분포가 UCF 피처와 **유사해짐**

<도메인 적응 전>

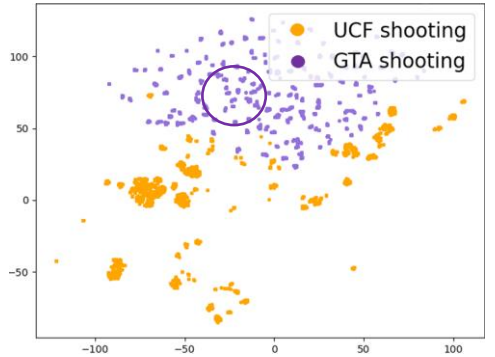
<도메인 적응 후>

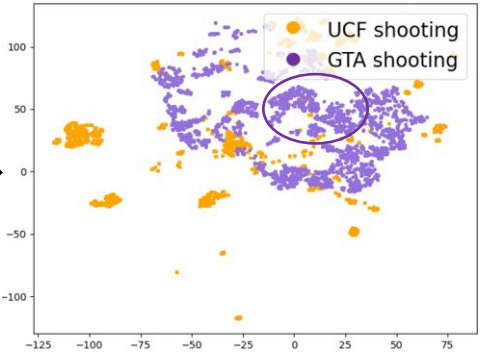
Stabbing



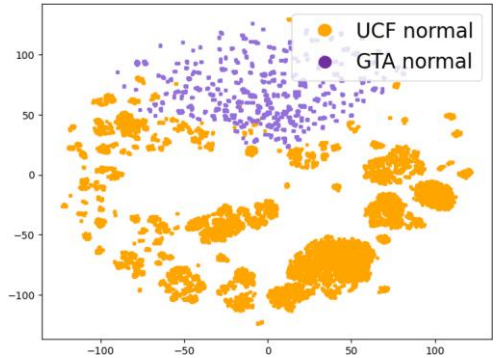


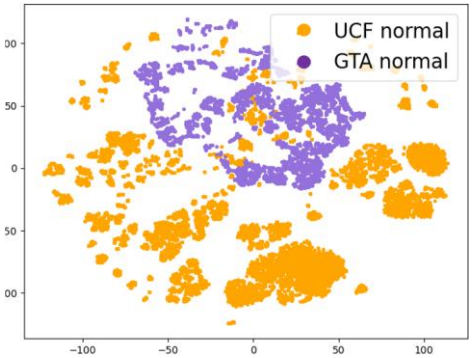
Shooting





Normal





4. 이상 상황 탐지 모델 적용

1) Video anomaly detection(VAD) 모델 적용

- VAD 모델을 사용하여 GTA-Crime(GTA) 데이터셋과 도메인 적응 효과 확인
  - RTFM, UR-DMU, MGFN, CLIP-TSA, VadCLIP (I3D 3개, CLIP 2개) 모델을 사용
  - UCF3: UCF-Crime 전체 데이터셋 중 Fighting, Shooting, Normal 세 개의 클래스 사용
    - ✓ 학습 데이터: UCF3+GTA 평가 데이터: UCF3
- 실험 결과 UCF3+GTA(with DA) 항목이 가장 높은 AUC 점수를 보임

  ⇒ **GTA-Crime 가상 데이터셋과 피처 단위에서의 도메인 적응이 유효함**을 보여줌

VAD model	UCF3	UCF3+GTA(w/o DA)	UCF3+GTA(w DA)
RTFM	85.43	84.98	**87.27**
UR-DMU	86.04	81.39	**86.47**
MGFN	82.55	79.37	**83.64**
CLIP-TSA	78.75	81.62	**82.66**
VadCLIP	73.59	74.60	**74.79**

4. 이상 상황 탐지 모델 적용

2) Ablation study

- 적대적 학습 방식을 통한 **피처 단위에서의 도메인 적응의 일반성** 검증
- WGAN-GP loss ⇒ **CycleGAN loss** 교체
- 실험결과 WGAN-GP loss를 사용했을 때와 비슷한 AUC 성능을 나타냄

  ⇒ **적대적 학습을 이용한 피처 단위에서의 도메인 적응의 유효성**을 보여줌

VAD model	UCF3	UCF3+GTA(w/o DA)	CycleGAN	WGAN-GP
RTFM	85.43	84.98	**87.28**	87.27
UR-DMU	86.04	81.39	86.35	**86.47**
MGFN	82.55	79.37	**84.35**	83.64
CLIP-TSA	78.75	81.62	81.65	**82.66**
VadCLIP	73.59	74.60	**76.84**	74.79

5. Contribution 및 추가 성과

ICASSP 2025 논문 제출

1. 실제 CCTV와 유사한 구도의 오픈 소스의 가상 데이터셋 및 제작 파이프라인[1] 공개
2. 피처 단위에서의 간단한 비디오 도메인 적응
3. 1 & 2를 이용하여 다양한 VAD 모델에서의 성능 향상

(ICASSP 2025) 2025 IEEE International Conference on Acoustics, Speech and Signal Processing

Date: 6-11 April 2025
Location: Hyderabad, India

Conference Paper Submission Deadline: 09 September 2024
Website Link: ICASSP 2025 Website

[1] https://github.com/ta-ho/GTA-Crime

Q & A

Appendix

1. WGAN-GP loss function

z: fake (GTA-Crime)
x: real (UCF-Crime)

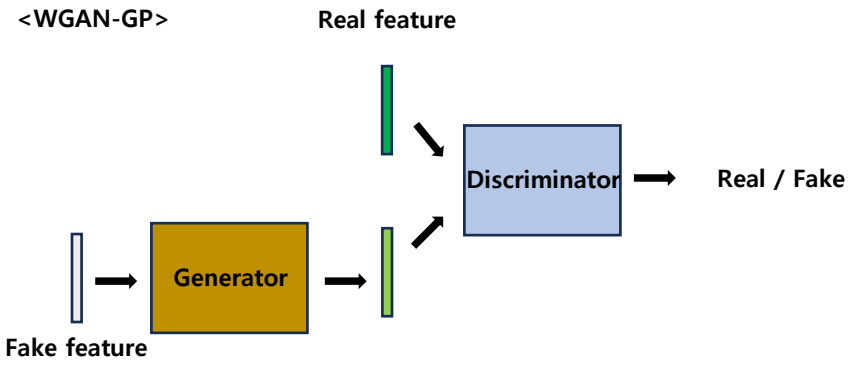
1. Feature Adapter loss

$$L_G = -\mathbb{E}_{z\sim\mathbb{P}_z}[D(G(z)]$$

2. Discriminator loss

WGAN loss | Gradient Penalty

$$L_D = \mathbb{E}_{z\sim\mathbb{P}_z}[D(G(z))] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(\| \nabla_{\tilde{x}} D(\tilde{x}) \|_2 - 1)^2]$$




```python
# critic iter
for j in range(args.critic_iter):
    real_data_v = autograd.Variable(ucf_features)
    netD.zero_grad()

    # train with real
    D_real = netD(real_data_v)
    D_real = D_real.mean()
    D_real.backward(mone)

    # train with fake
    fake = netG(gta_features)
    inputv = fake
    D_fake = netD(inputv)
    D_fake = D_fake.mean()
    D_fake.backward(one)

    # train with gradient penalty
    gp = gradient_penalty(netD, real_data_v.data, fake.data)
    gp.backward()

2)  D_cost = D_fake - D_real + gp # D(G(gta)) - D(ucf) + gp

    Wasserstein_D = D_real - D_fake # D(ucf) - D(G(gta))
    optimizerD.step()

    Wasserstein_D_list.append(Wasserstein_D.item())

### 2. update G network
for p in netD.parameters():
    p.requires_grad = False

netG.zero_grad()
fake = netG(gta_features)
G = netD(fake)
G = G.mean()
cos_sim = F.cosine_similarity(fake, ucf_features, dim=2).mean()
G.backward(mone)
1)  G_cost = -G
optimizerG.step()
```

1. CycleGAN loss function

z: fake (GTA-Crime)
x: real (UCF-Crime)

1. Feature Adapter loss

GAN loss (MSE loss)

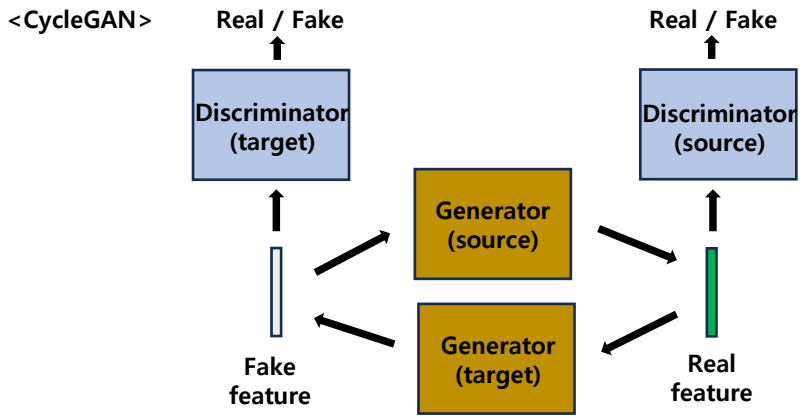
$$L_G = \mathbb{E}_{z\sim\mathbb{P}_z}[(D_S(G_S(z)) - 1)^2] + \mathbb{E}_{x\sim\mathbb{P}_r}[(D_T(G_T(x)) - 1)^2]$$
$$+ \mathbb{E}_{z\sim\mathbb{P}_z}[\|G_T(G_S(z)) - z\|_1] + \mathbb{E}_{x\sim\mathbb{P}_r}[\|G_S(G_T(x)) - x\|_1]$$

Cycle consistency loss (L1 loss)

2. Discriminator loss

$$L_D = \mathbb{E}_{x\sim\mathbb{P}_r}[(D_S(x) - 1)^2] + \mathbb{E}_{z\sim\mathbb{P}_z}[D_S(G_S(z))^2]$$
$$+ \mathbb{E}_{z\sim\mathbb{P}_z}[(D_T(z) - 1)^2] + \mathbb{E}_{x\sim\mathbb{P}_r}[D_T(G_T(x))^2]$$

GAN loss (MSE loss)




```
def backwardF( (parameter) criterionGAN: Any source_rec, target_real, source_fake, target_rec, ne
    lossF_S = criterionGAN(netD_S(target_fake), True)   # GAN loss netD_S(netF_S)
    lossF_T = criterionGAN(netD_T(source_fake), True)   # GAN loss netD_T(netF_T)
    lossCyc_S = criterionCycle(source_rec, source_real) * args.lambda_S     # forward cycle loss
    lossCyc_T = criterionCycle(target_rec, target_real) * args.lambda_T     # backward cycle loss

    if args.identity_flag == True:
        source_idt = netF_S(target_real)
        lossIdt_S = criterionIdt(source_idt, target_real) * args.lambda_S * args.lambda_idt
        target_idt = netF_T(source_real)
        lossIdt_T = criterionIdt(target_idt, source_real) * args.lambda_T * args.lambda_idt
    else:
        lossIdt_S = 0.0
        lossIdt_T = 0.0

    lossF = lossF_S + lossF_T + lossCyc_S + lossCyc_T + lossIdt_S + lossIdt_T

    return lossF, lossF_S, lossF_T, lossCyc_S, lossCyc_T, lossIdt_S, lossIdt_T


def backwardD(args, source_real, target_fake, source_rec, target_real, source_fake, target_rec, ne
    target_pred_real = netD_S(target_real)                # source true
    lossD_S_real = criterionGAN(target_pred_real, True)     # source false
    target_pred_fake = netD_S(target_fake.detach())
    lossD_S_fake = criterionGAN(target_pred_fake, False)
    lossD_S = (lossD_S_real + lossD_S_fake) * 0.5          # source combine

    source_pred_real = netD_T(source_real)                # target true
    lossD_T_real = criterionGAN(source_pred_real, True)
    source_pred_fake = netD_T(source_fake.detach())        # target false
    lossD_T_fake = criterionGAN(source_pred_fake, False)
    lossD_T = (lossD_T_real + lossD_T_fake) * 0.5          # target combine

    return lossD_S, lossD_T
```

3. Additional Experiment(1)

1. UCF-Crime의 모든 클래스를 사용하여 VAD 진행

VAD model	UCF-Crime	+GTA-Crime (w/o DA)	+GTA-Crime (w WGAN-GP)	+GTA-Crime (w CycleGAN)
VadCLIP	**87.61**	87.06	87.31	86.59

2. UCF-Crime 학습 데이터 수

category	Normal	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
# videos	800	48	45	41	47	87	29	45	127	145	27	29	95	45

GTA-Crime 추가: Fighting +124, Shooting, +146, Normal + 262

category	Normal	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
# videos	**1062**	48	45	41	47	87	29	**169**	127	145	**173**	29	95	45

3. Additional Experiment(2)

1. GTA-Crime 학습 및 GTA-Crime 평가

VAD model	AUC(overall)	AUC(shooting)	AUC(stabbing)
RTFM	84.02	78.96	84.05
UR-DMU	80.13	70.09	71.89
MGFN	76.18	77.93	86.55
CLIP-TSA	84.49	86.06	87.09
VadCLIP	71.36	78.84	64.35

4. 데이터셋 비교

1. GTA-Crime과 기존 데이터셋 비교

(△: 일부 영상만 해당)

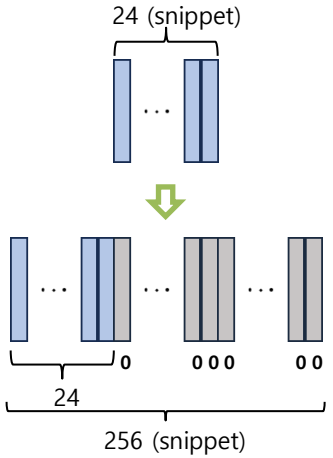
Dataset	Fatal violence types	CCTV view	Synthetic	# videos	Resolution
UCF-Crime	Fighting(△), Shooting	○	X	1900	320 x 240
XD-Violence	Fighting(△), Shooting	△	△	4754	Multiple
CCTV-Fights	Fighting(△)	△	X	1000	Multiple
GTAVEvent	None	○	○	54	2560 x 1440
Ubnormal	None	○	○	543	Multiple
GTA-Crime(ours)	**Stabbing, Shooting**	**○**	**○**	**532**	**1920 x 1080**

5. Snippet 구성 방식

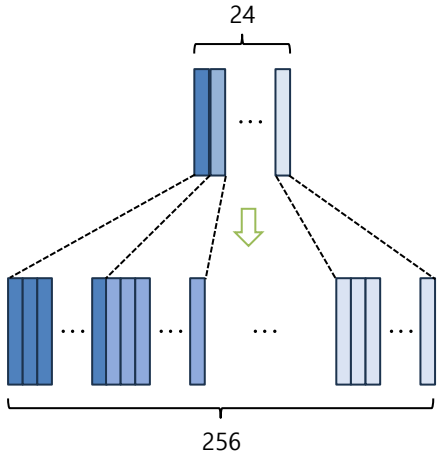
1. Snippet 구성방식: Uniform 방식, Padding 방식

- 비디오의 16개 프레임을 1개의 snippet으로 구성
  - 1 snippet = 16 frames (예시: 400 frames → 24 snippets)
- 학습 시 배치 구성을 위해 모든 비디오의 snippet 수를 일정하게 변환
  - VAD 모델에 적용하기 위해 같은 개수의 snippets으로 변환 필요
  - Pad, Uniform 방식으로 snippet 수 변경

Pad

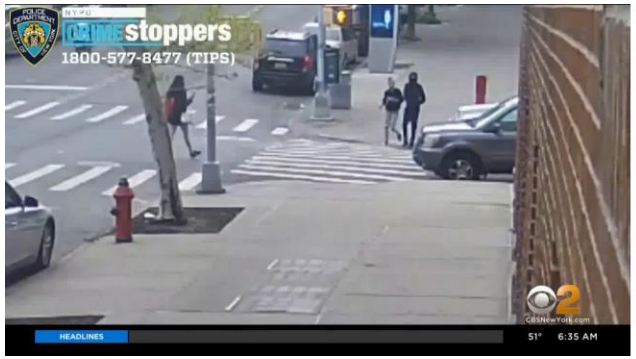


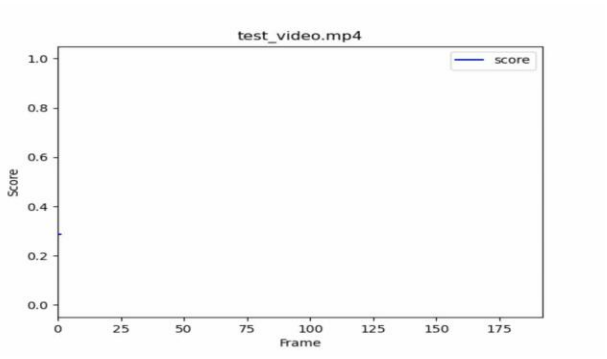
Uniform



6. 영상 이상 탐지 시각화

1. UCF3 학습





2. UCF3+GTA(w WGAN-GP) 학습