Monthly Report
2026-9호

www.copyright.or.kr

저작권 기술 트렌드



COPYRIGHT TECH TREND

한국저작권위원회
KOREA COPYRIGHT COMMISSION

본 보고서는 EC21R&C(컨설팅사)에서 작성하였고, 국내외 저작권 기술·산업 동향을 조사한 자료로 한국저작권위원회 의견이 반영되어 있지 않습니다.

워터마킹으로 추적하는 불법 유출 콘텐츠의 출처

뉴스 브리프

2026년 9월, 크로아티아와 슬로베니아의 방송사가 포뮬러 1 생중계를 보호하기 위해 실시간 불법 유통 모니터링과 포렌식 워터마크를 결합한 솔루션을 도입했다. 스포츠 생중계는 방송사가 보유한 가장 가치 있는 자산 중 하나이지만, 무단 송출이 어디에서 시작되었는지 확인할 방법이 없다면 대응은 사후에 머물 수밖에 없다. 솔루션 공급사는 스포츠 생중계의 불법 유통이 갈수록 정교해지며, AI에 의해 주도되고 있다고 진단한다. 결국 관건은 이미 퍼져 나간 콘텐츠에서 그 출처를 되짚는 기술이다. 본 보고서는 영상의 각 프레임에 식별 정보를 삽입해 출처와 편집 여부를 함께 확인하는 플로우마크와, 압축을 견디는 워터마크 삽입 기법인 COVER를 살펴본다.

뉴스 플러스

Ⅰ. 서론: 스포츠 생중계와 불법 재배포 추적

• 포뮬러 1 중계 보호를 위한 대응책 도입

2026년 9월, 크로아티아의 방송사 RTL 크로아티아(RTL Croatia)와 슬로베니아의 방송사 프로 플러스(Pro Plus)가 송출하는 2026 시즌 포뮬러 1(Formula 1) 중계를 보호하기 위해 콘텐츠 보호 솔루션 나그라벤투리(Nagra Venturi)를 도입했다.[1] 솔루션을 공급한 곳은 스위스의 콘텐츠 보안 기업 나그라비전(Nagravision)이며, 도입은 프랑스의 영상 기술 기업 아테미(Ateme)와의 협업으로 완료되었다. 해당 솔루션은 불법 유통을 실시간으로 모니터링하는 동시에 네트워크 구간에서 포렌식 워터마크(forensic watermark)를 삽입해, 무단 재배포가 일어나는 지점을 찾아낸다. 포렌식 워터마크란 콘텐츠에 배포 경로를 식별할 정보를 삽입해 두었다가, 유출된 콘텐츠에서 그 정보를 읽어 출처를 되짚는 기술을 가리킨다.

1) Nick Snow, “Nagravision secures F1 content for RTL Croatia, Pro Plus Slovenia”, Advanced Television, 2026.09.14., https://www.advanced-television.com/2026/09/14/nagravision-secures-f1-content-for-rtl-croatia-pro-plus-slovenia/

프로 플러스의 최고기술책임자 다니엘 코프(Daniel Cof)는 포뮬러 1 중계권을 자사가 보유한 가장 가치 있는 자산 가운데 하나로 꼽으면서, 중계가 시청자에게 도달하는 경로를 지키는 일은 "단순한 기술 문제가 아니라 사업의 우선순위"라고 밝혔다.[2] RTL 크로아티아의 시메 유를리나(Sime Jurlina)도 프리미엄 스포츠 콘텐츠를 보호함으로써 중계 투자 가치를 지키고, 시청자에게 더 나은 환경을 제공할 수 있다고 설명했다. 나그라비전의 최고운영책임자인 모르텐 솔바켄(Morten Solbakken)은 AI의 주도로 스포츠 생중계의 불법 유통이 갈수록 정교해지고 있어, 실시간 콘텐츠 보호가 중요해졌다고 진단했다.

• 사후 차단만으로 끊어지지 않는 불법 송출의 고리

스포츠 생중계의 가치는 경기가 열리는 바로 그 시간에 집중되며, 경기가 끝난 뒤의 영상은 같은 내용을 담고 있어도 그 가치가 현저히 줄어든다. 불법 재배포가 특히 문제가 되는 이유도 여기에 있다. 생중계와 동시에 이뤄지는 불법 송출은 잠재 시청자를 빼앗으며, 뒤늦게 차단하더라도 이미 소진된 가치는 복구할 수 없다. 방송사가 중계권에 투자한 가치를 회수할 수 있는 시간은 경기 시간 내로 한정되어 있는 셈이다.

그럼에도 대응은 대체로 사후에 이루어지며, 같은 일이 되풀이된다. 불법 송출을 찾아 차단하더라도 같은 경기의 다음 순간에 다른 주소로 다시 올라오고, 다음 경기에서도 같은 양상이 반복되기 때문이다. 불법 송출을 하나씩 찾아 지우는 방식으로는 그것이 흘러나오기 시작한 지점 자체를 건드리지 못한다. 이번에 도입된 솔루션이 개별 송출의 차단보다 무단 재배포의 출처를 찾는 데 초점을 둔 것도 이러한 맥락에서 이해할 수 있다. 프로 플러스가 도입의 성과로 운영 방식을 크게 바꾸지 않고도 무단 재배포의 출처를 더 잘 파악하게 되었고 그만큼 빠르게 대응할 수 있게 되었다는 점을 꼽은 것 역시 같은 맥락이다.

결국 관건은 이미 퍼져 나간 콘텐츠를 보고 그것이 어디에서 비롯되었는지를 되짚을 수 있는가에 있다. 그런데 유출된 콘텐츠는 원본 그대로의 모습을 하고 있지 않다. 유출 지점이 원본 파일이든 송출 구간이든, 재유통 과정에서 콘텐츠는 여러 차례 다시 압축되고 화면이 잘리거나 크기가 바뀐 채 퍼져 나간다. 이러한 변형을 거치고도 콘텐츠 안에 단서가 남아 있어야 비로소 출처를 되짚는 일이 가능해진다.

• 출처를 되짚는 두 갈래의 기술

유출된 콘텐츠에서 출처를 되짚으려면 콘텐츠 안에 단서가 남아 있어야 한다. 이를 위해 배포 전에 식별 정보를 삽입해 두었다가 유출된 사본에서 그 정보를 읽어내는 기술이 워터마크이며, 이번에 도입된 솔루션이 사용하는 방식도 여기에 해당한다. 관건은 그 정보가 재유통 과정을 거치고도 남아 있는가에 있다. 특히 영상이 저장되거나 공유될 때마다 거치는 압축은 사람 눈에 잘 띄지 않는 정보를 우선 지우는데, 워터마크는 본래 눈에 띄지 않도록 삽입되므로 함께 지워지기 쉽다. 그래서 최근 연구는 “워터마크를 어디에 삽입해야 압축 과정을 견디는가?”라는 물음에 집중하고 있다.

2) Nick Snow, “Nagravision secures F1 content for RTL Croatia, Pro Plus Slovenia”, Advanced Television, 2026.09.14., https://www.advanced-television.com/2026/09/14/nagravision-secures-f1-content-for-rtl-croatia-pro-plus-slovenia/

본 보고서는 이 물음에 서로 다른 답을 내놓은 연구 두 건을 살펴본다. 첫 번째 연구는 플로우마크(FlowMark)로서 워터마크 삽입 위치를 화면 안에서 찾되, 적합한 영역을 모델 스스로 판단하도록 한다. 영상 식별자와 프레임 번호를 나누어 담아 압축과 편집을 거친 뒤에도 출처와 편집 여부를 함께 확인할 수 있도록 한다. 두 번째 연구인 COVER(Codec-Robust Video Watermarking)는 워터마크 삽입 위치를 화면 바깥으로 옮겨, 영상을 압축된 형태로 다루는 별도의 공간에 삽입한다. 이어지는 본론에서는 각 기술의 작동 원리와 성능 및 한계를 차례로 분석한다.

II. 본론 1: 프레임마다 서명을 남기는 영상 워터마크, 플로우마크

• 영상 워터마크가 풀지 못했던 과제

워터마크를 영상에 적용할 때 가장 먼저 맞닥뜨리는 문제는 화면의 깜박임이다. 기존 방식 대부분은 화면 전체에 걸쳐 워터마크를 삽입하는데, 이를 영상의 각 프레임에 따로 적용하면 프레임마다 삽입된 신호의 모양이 조금씩 달라진다. 정지된 개별 사진에서는 눈에 띄지 않던 미세한 차이도, 초당 수십 장이 이어지는 영상에서는 미세한 떨림으로 드러난다. 플로우마크 연구진은 이를 깜박임(flicker)이라 부르며, 기존 방식이 프레임 사이의 일관성을 고려하지 않은 데서 비롯된 문제라고 지적한다.

두 번째 문제는 압축이다. 영상은 H.264나 H.265 같은 코덱(codec), 곧 영상을 작은 용량으로 줄이는 압축 방식을 거쳐 유통되는데, 이 과정에서 사람 눈에 잘 띄지 않는 정보가 우선 제거된다. 워터마크는 본래 눈에 띄지 않도록 삽입되므로 압축이 걸러내려는 정보와 성격이 비슷하고, 그만큼 함께 지워지기 쉽다. 여기에 화면을 자르거나 회전시키는 변형과 더불어 유튜브(YouTube)나 페이스북(Facebook)과 같은 플랫폼의 자체 압축 과정을 거치면 워터마크가 살아남을 여지는 한층 더 줄어든다.

세 번째 문제는 삽입 위치를 정하는 방식에 있다. 선행 연구 중에서는 화면 전체가 아니라 특정 영역에만 워터마크를 삽입해 화질 저하를 줄이려는 연구가 있었으나, 그 영역을 사람이 직접 지정해 주어야 하는 한계가 있었다. 어떤 연구는 영역을 무작위로 선택하는 것을 제안하기도 하였는데, 이러한 선행 연구의 방식은 실제 영상 제작과 유통 과정에 그대로 적용하기는 어렵다. 한편, 깜박임을 줄이기 위해 여러 프레임에 같은 내용을 반복해 삽입하는 방식도 사용되긴 하였으나, 이 경우 모든 프레임에 같은 정보가 삽입되므로 프레임 단위로 무엇이 바뀌었는지는 확인할 수 없게 된다.

• 삽입 위치를 스스로 찾아내는 마스크 예측기

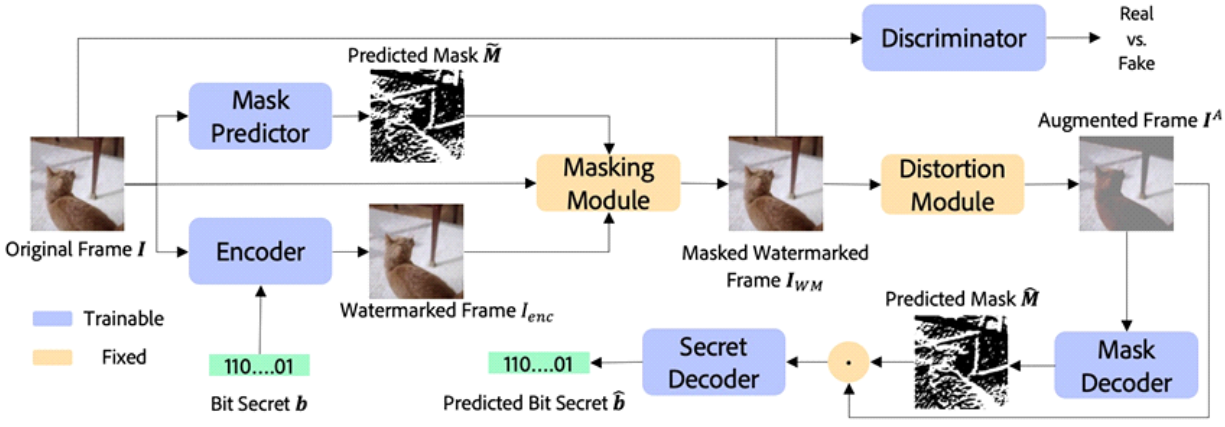
플로우마크의 출발점은 워터마크를 어디에 삽입할지를 모델이 스스로 정하도록 만드는 것이다. 이를 담당하는 구성 요소가 마스크 예측기(Mask Predictor)로, 입력된 프레임을 살펴 화면의 각 지점이 워터마크를 삽입하기에 얼마나 적합한지를 판단한다. 그 결과로 화면과 같은 크기의 마스크(mask), 곧 삽입할 영역과 삽입하지 않을 영역을 구분한 지도가 만들어진다. 마스크 예측기는 질감이 복잡하거나 움직임이 안정적인 영역을 골라내는 경향을 보이는데, 이러한 영역일수록 워터마크가 눈에 덜 띄면서도 압축에 견디기 쉽기 때문이다. 이 판단 기준은 사람이 정해준 규칙이 아니라 학습 과정으로 정해진다.

마스크만으로는 눈에 띄는 흔적을 충분히 제거할 수 없다. 그래서 플로우마크는 화면의 밝기에 따라 워터마크의 강도를 다르게 조절한다. 화면의 밝기와 질감에 따라 워터마크의 시각적 노출 정도가 달라진다. 특히 어둡고 평평한 영역에서는 미세한 변화가 눈에 잘 띄고 압축 과정에서 잡티가 생기기도 쉬우므로, 이런 영역에서는 워터마크의 강도를 줄인다. 반대로 밝고 질감이 있는 영역에서는 워터마크 강도를 유지해, 복원에 필요한 만큼의 신호가 남도록 한다.

깜박임에는 다른 방식으로 대응한다. 플로우마크는 인접한 두 프레임 사이에서 화면이 변한 정도가 원본의 변화와 비슷하게 유지되도록 하는 조건을 학습에 반영한다. 원본에서 거의 움직이지 않은 부분은 워터마크 삽입 후에도 거의 움직이지 않아야 하고, 크게 움직인 부분은 그만큼 움직여야 한다는 뜻이다. 이 조건이 지켜지면 삽입된 워터마크가 영상 본래의 움직임을 따라가게 되므로 화면의 떨림이 줄어든다.

학습은 세 단계로 나누어 진행한다. 첫 단계에서는 아무런 변형도 가하지 않은 상태로 삽입과 복원의 기본 동작을 익힌다. 두 번째 단계에서는 밝기와 색을 바꾸거나 화면을 자르고 회전시키는 등 이미지 수준의 변형을 거치도록 하며, 이 단계부터 마스크 예측기의 학습을 멈춤으로써 삽입 위치의 판단 기준이 흔들리지 않도록 한다. 세 번째 단계에서는 H.264, H.265, VP9, AV1과 같은 실제 영상 코덱의 압축을 점점 강하게 적용하고, 유튜브나 페이스북에 올렸을 때 일어나는 재압축을 가정한 조건까지 포함한다.

[그림 1] 플로우마크의 전체 작동 구조



출처: Vishal Asnani 외 2인, “FlowMark: Mask-Guided Video Watermarking”, arXiv, 2026.07.06., https://arxiv.org/pdf/2607.05261

• 영상 식별자와 프레임 번호를 나누어 담는 비트 설계

플로우마크는 한 프레임에 128비트의 정보를 삽입한다. 여기서 비트(bit)는 0과 1 가운데 하나를 담는 정보의 최소 단위로, 128비트란 0과 1을 128자리로 늘어놓은 값을 뜻한다. 플로우마크는 이 128자리를 한 덩어리로 쓰지 않고 앞의 112자리와 뒤의 16자리로 나누어 사용한다. 앞부분에는 영상마다 고유하게 부여된 식별자를 담고, 뒷부분에는 그 프레임이 영상에서 몇 번째인지를 나타내는 번호를 담는다.

이렇게 나누어 담으면 두 가지 정보를 한꺼번에 확인할 수 있다. 영상 식별자는 모든 프레임에서 같은 값으로 유지되므로, 유출된 영상의 어느 한 부분만 확보해도 그것이 어떤 영상에서 나왔는지 알 수 있다. 프레임 번호는 순서대로 하나씩 늘어나므로, 복원된 번호의 흐름이 끊기거나 뒤바뀌어 있다면 그 영상이 편집을 거쳤다는 뜻이 된다. 프레임이 빠지거나 추가되었는지, 순서가 뒤섞였는지를 번호 배열만으로 판단할 수 있는 것이다.

이러한 설계를 통해 앞서 언급한 기존 방식의 한계에 대응할 수 있다. 여러 프레임에 같은 값을 반복해 삽입하는 방식은 깜박임을 줄이는 대신 프레임 단위의 검증을 제약하여, 영상이 어디에서 왔는지는 알아도 그 안에서 무엇이 바뀌었는지는 확인하기 어렵다는 한계가 있었다. 플로우마크는 프레임마다 다른 값을 삽입하면서도, 앞서 살펴본 마스크와 시간적 조건을 통해 깜박임을 억제한다. 연구진은 플로우마크 구조를 통해 영상 수준의 출처와 프레임 수준의 순서를 워터마크 안에 함께 담을 수 있다고 설명한다.

• 화질 및 강건성 실험 결과

플로우마크는 사진과 영상을 제공하는 유료 서비스인 어도비 스톡(Adobe Stock)의 영상 자료로 학습한 뒤, 정지 화면과 영상 두 종류의 자료 모음에서 성능을 측정했으며, 영상 평가에는 초당 24프레임의 영상 155편을 사용하였다. 평가 항목은 크게 두 갈래다. 하나는 워터마크를 삽입한 뒤 화질이 얼마나 유지되는지이고, 다른 하나는 여러 변형을 거친 뒤 본래 워터마크를 얼마나 정확히 복원하는지다. 복원 정확도는 비트 정확도(bit accuracy)로 측정하는데, 이는 삽입한 128자리 가운데 몇 자리를 올바르게 복원했는지를 비율로 나타낸 값이다.

화질 측면에서 플로우마크는 논문에서 설정한 비교 실험의 화질 지표에서 우수한 결과를 보였다. 원본과 워터마크를 삽입한 화면의 차이를 나타내는 최대 신호 대 잡음비(Peak Signal-to-Noise Ratio, 이하 PSNR)*는 정지 화면에서 49.08dB, 영상에서 50.41dB로 측정되었다. PSNR 값은 클수록 원본과의 차이가 작다는 뜻인데, 플로우마크에 비해 비교 대상은 30dB대에서 48dB대 사이에 분포했다. 영상의 체감 화질을 사람의 평가에 가깝게 수치화한 지표에서는 100점 만점에 98.95점을 기록했고, 연구진은 일반적인 시청 환경에서 원본과 구별하기 어려운 수준이라고 설명한다. 아무런 변형을 가하지 않은 조건에서 비트 정확도는 128자리를 모두 되찾는 수준이었다.

변형을 가한 조건은 밝기와 색을 바꾸는 종류, 화면을 자르거나 회전시키고 크기를 바꾸는 종류, 코덱으로 압축하는 종류, 끝으로 이들을 함께 적용하는 조건 네 가지이다. 여기서 비교 대상의 한계가 드러난다. 압축 조건에서 정지 화면 기준 0.99를 기록한 어떤 방식은 화면을 자르거나 회전시키는 조건에서 0.52로, 여러 변형을 함께 적용한 조건에서는 0.50으로 떨어졌다. 128자리 가운데 절반만 맞힌다는 것은 0과 1을 무작위로 찍은 것과 다를 바 없어, 사실상 워터마크를 되찾지 못했다는 뜻이 된다. 반면, 연구진은 플로우마크가 이 네 가지 조건하에서 모두 높은 비트 정확도를 유지했다고 보고한다.

연구진은 영상에 특유한 편집과 실제 유통 경로도 시험 조건에 포함했다. 프레임의 순서를 바꾸거나, 새로운 프레임을 끼워 넣거나, 일부를 지우거나, 재생 속도를 바꾸거나, 프레임과 프레임 사이를 채워 넣는 편집이 여기에 해당한다. 또한, 유튜브와 페이스북에 실제로 업로드해 각 플랫폼이 자체 기준으로 다시 압축한 영상에서도 워터마크를 되찾을 수 있는지 확인했으며, 연구진은 이 조건에서 플로우마크가 비교 대상 가운데 가장 높은 검출 성능을 보였다고 밝힌다.

* 최대 신호 대 잡음비(Peak Signal-to-Noise Ratio, PSNR): 원본 영상과 처리를 거친 영상의 차이를 데시벨 단위로 나타낸 지표로, 값이 클수록 원본에 가까움을 의미함

• 남은 한계와 이어지는 과제

워터마킹 방식에도 한계가 존재한다. 워터마크는 콘텐츠가 배포되기 전에 삽입되어야 하며, 삽입되지 않은 채 유출된 영상에서는 되짚을 단서 자체가 존재하지 않는다. 또한, 복원 과정도 완벽하지는 않다. 워터마크를 삽입한 영역을 다시 찾아내는 정확도는 정지 화면에서 0.95, 영상에서 0.96으로, 아무런 변형을 가하지 않은 조건에서도 1에는 미치지 못한다.

실험 조건도 결과를 해석할 때 함께 고려할 필요가 있다. 학습과 평가에 쓰인 화면 해상도는 모두 256×256 크기로 조정된 것이고, 영상 평가에 사용된 자료는 155편이며, 학습에는 16프레임 길이의 짧은 구간이 쓰였다. 실제 유통되는 영상의 해상도와 길이는 이보다 훨씬 크고 길다. 학습에는 A100 GPU 8대가 동원되었고, 세 단계를 합쳐 50만 스텝의 학습이 이루어졌다.

한편, 이 연구가 상정한 유출 경로는 파일이나 스트림 형태로 퍼져 나가는 상황이다. 시험된 변형은 압축, 화면의 변형, 시간 축의 편집, 플랫폼의 재압축으로, 모두 워터마크가 화면 위에 삽입된다는 전제 아래하에 설계된 조건들이다. 그런데 압축 과정에서 삭제되는 것이 바로 화면의 미세한 정보이므로, 삽입 자리를 화면 안에서 고르는 한 이 문제는 완전히 해소되기 어렵다. 이어지는 본론 2에서 다룰 기술은 삽입 자리를 화면 바깥으로 옮김으로써 이러한 한계에 대응한다.

II. 본론 2: 압축을 견디는 자리에 삽입하는 영상 워터마크, COVER

• 코덱 압축이라는 미해결 과제

일반적으로 영상은 저장되거나 공유되기 전에 코덱으로 다시 압축된다. COVER 연구진은 바로 이 지점에서 기존 워터마크가 지워진다고 지적한다. 워터마크는 화면의 밝기와 색을 조금씩 바꾸는 방식으로 삽입되는데, 코덱이 용량을 줄이기 위해 가장 먼저 덜어내는 것도 그런 미세한 변화이기 때문이다. 그래서 삽입 직후에는 아무 문제 없이 읽히던 워터마크가, 시청자에게 도달할 무렵에는 사라졌을 수 있다.

연구진은 기존 연구가 이 문제에 정면으로 대응하지 않았다고 본다. 압축을 여러 왜곡 가운데 하나로 목록에 올려놓았을 뿐이며, COVER연구진은 실제 코덱 압축을 핵심 설계 목표로 삼았다는 점에서 기존 접근과 차별화된다고 설명한다. 여기에는 두 가지 사정이 있다. 하나는 코덱이 여러 방식의 손실을 한꺼번에 적용하고 그 정도를 영상 내용과 목표 용량에 맞춰 조절하므로, 하나의 고정된 왜곡으로 대신할 수 없다는 점이다. 다른 하나는 학습 방식에 관한 것이다.

인공지능 모델은 결과가 얼마나 어긋났는지를 되짚어 스스로를 조금씩 고쳐 가며 학습한다. 그런데 실제 코덱은 이 되짚기가 통하지 않는 방식으로 작동한다. 압축을 거친 뒤 워터마크가 얼마나 지워졌는지는 알 수 있어도, 그 결과를 삽입 단계로 되돌려 무엇을 고쳐야 하는지 알려 줄 수가 없다. 그래서 실제 코덱을 학습 과정 안에 그대로 넣을 수 없고, 코덱을 대신할 무언가를 두어야 한다.

• 화면이 아닌 잠재 공간에 삽입하는 방식

COVER의 해법은 워터마크를 삽입하는 자리를 옮기는 것이다. 화면의 미세한 변화가 압축에 지워지기 쉽다면, 그런 변화에 영향을 받지 않는 다른 자리를 찾자는 발상이다. COVER가 택한 자리는 잠재 공간(latent space)이다. 잠재 공간에는 영상 생성 모델이 영상을 다룰 때 쓰는 압축된 형태의 표현이 저장되는데, 화면을 화소 단위로 늘어놓는 대신 영상의 구조와 움직임을 간추려 담는다. COVER는 이미 학습을 마친 영상 생성 모델의 잠재 공간을 그대로 빌려 쓰며, 생성 모델 자체는 더 이상 학습시키지 않고 고정한다.

이 자리를 고른 이유는 코덱이 무엇을 남기고 무엇을 버리는지에서 원인을 찾을 수 있다. 코덱은 시청자가 화면을 알아보는 데 필요한 큰 틀은 남기고, 잘 알아차리지 못하는 미세한 변화를 덜어낸다. 그런데 영상 생성 모델의 잠재 공간도 이와 비슷하게 영상의 중요한 구조를 중심으로 짜여 있다. 따라서 잠재 공간 안에 워터마크를 삽입하면, 코덱이 굳이 덜어낼 필요가 적은 자리에 놓이게 된다. 연구진은 이러한 설계를 코덱이 용량을 나누어 쓰는 방식을 따라간 설계라고 설명한다.

삽입과 확인은 다음과 같은 과정으로 이루어진다. 먼저 영상을 잠재 공간으로 옮긴 뒤, 그 위에 워터마크에 해당하는 값을 덧얹는다. 그다음 이를 다시 영상으로 되돌리면 워터마크가 삽입된 영상이 된다. 확인할 때는 받은 영상을 같은 잠재 공간으로 다시 옮긴 뒤, 그 안에서 삽입한 값을 읽어낸다. 화면의 화소를 뜯어보는 대신 잠재 공간 수준에서 읽기 때문에, 압축이 이미 지워 버린 화면의 미세한 변화를 되살릴 필요가 없다.

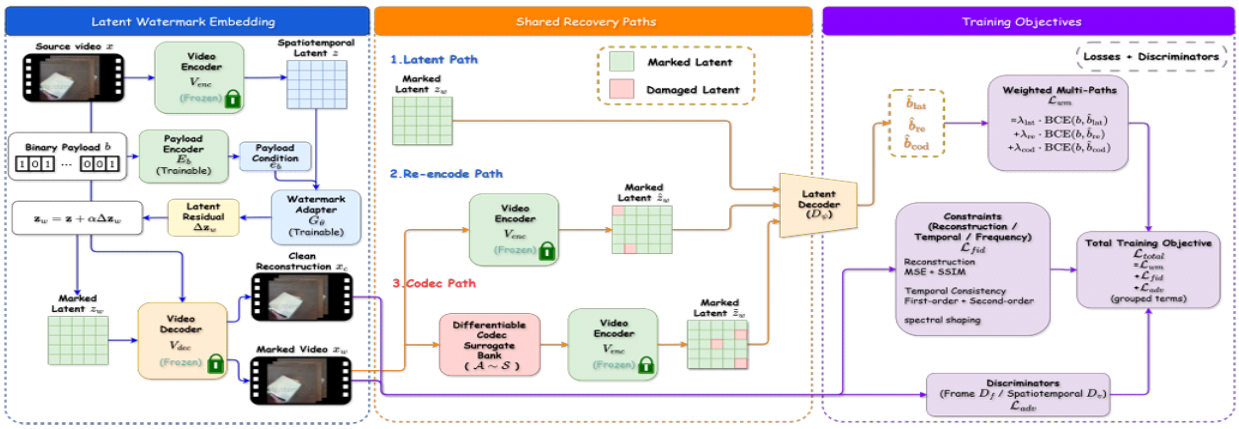
• 코덱을 흉내 낸 학습과 세 갈래 복원 경로

실제 코덱을 학습에 넣을 수 없으므로, COVER는 코덱의 작동을 모사하는 대체 모델(surrogate model)을 사용했다. 특정 코덱 하나를 그대로 재현하는 대신, 여러 코덱이 공통으로 일으키는 손상의 유형을 네 가지로 추려 각각을 모사하도록 했다. 값의 정밀도를 떨어뜨리고 화면을 격자 모양으로 뭉개는 유형과 미세한 무늬를 흐리는 유형, 해상도를 줄였다가 다시 키우는 유형, 색 정보를 줄이는 유형의 네 가지이다. 학습할 때마다 이들을 여러 개 이어 붙인 조합이 하나씩 뽑혀 적용되므로, 모델은 손상이 겹쳐서 오는 상황을 매번 마주하게 된다.

읽어내는 쪽은 세 갈래 경로로 훈련된다. 첫째는 워터마크를 삽입한 잠재 공간에서 곧바로 값을 읽는 경로로, 값이 제대로 담겼는지를 확인한다. 둘째는 그것을 영상으로 되돌렸다가 다시 잠재 공간으로 옮긴 뒤 읽는 경로다. 오가는 과정에서 생기는 어긋남을 견디도록 하기 위해서다. 셋째는 되돌린 영상에 앞서 만든 대체물을 적용해 손상시킨 뒤 다시 옮겨 읽는 경로로, 압축을 견디도록 하는 역할을 한다. 세 가지 경로는 읽어내는 장치 하나를 함께 사용하는데, 이는 어느 조건에서든 같은 방식으로 읽을 수 있도록 하기 위해서이다.

학습에는 워터마크를 얼마나 정확히 읽어내는가와 함께 화질을 지키는 조건이 추가된다. 구체적으로, 원본과 워터마크를 삽입한 영상의 차이를 작게 유지하는 조건, 프레임이 이어질 때 화면의 움직임이 원본과 어긋나지 않도록 하는 조건이 필요하다. 여기에 한 가지 조건이 더해지는데, 워터마크의 신호를 화면이 빠르게 변하는 부분이 아니라 완만하게 변하는 부분으로 몰아주는 조건이다. 압축이 가장 먼저 덜어내는 부분이 빠르게 변하는 부분이므로, 그 반대편에 신호를 두어 살아남을 여지를 넓히려는 것이다.

[그림 2] COVER의 전체 작동 구조



출처: Yuxin Cao 외 4인, “COVER: Codec-Robust Video Watermarking with Generative Video Priors”, arXiv, 2026.09.17.(Figure 2), https://arxiv.org/abs/2609.26236

• **실제 코덱에서의 성능과 한계**

평가는 실제 코덱으로만 이루어졌다. 학습에 쓴 대체물이 아니라 H.264, H.265, AV1, VP9, 네 종류의 코덱을 쓰되, 각 코덱마다 압축 강도를 세 단계로 나누어 모두 12가지 조건을 만들었다. 시험 대상으로는, 글을 입력하면 영상을 만들어 내는 모델로 생성한 영상 500편을 사용하였으며, 영상과 조건의 조합마다 서로 다른 값을 여러 차례 삽입해 읽어냈다. COVER의 평균 비트 정확도는 93.72%로 12가지 조건 가운데 11가지에서 가장 높았다.

차이가 두드러지는 지점은 압축이 강한 조건이다. 압축이 약한 조건에서는 여러 방식이 모두 97%를 넘어 우열을 가리기 어렵다. 반면, 압축이 강할수록 격차가 벌어지며, 가장 불리한 조건에서도 정확도가 기존 방식보다 높았다. 워터마크는 가장 약한 지점에서의 성능이 실제 기술적 실효성을 좌우하므로, 연구진은 이 대목을 주요 성과로 꼽는다.

다만, 화질 성능은 뒤떨어지는 것으로 나타났다. 워터마크를 삽입한 영상의 PSNR은 28.30dB로, 비교 대상 가운데 두 방식에 미치지 못한다. 연구진은 강건성을 얻는 대신 화질을 내준 결과라고 설명하고 있으며, 두 가지를 같은 조건에서 견주는 별도의 시험이 필요하다고 덧붙이고 있다. 하지만 이 수치에는 영상 생성 모델이 영상을 잠재 공간으로 옮겼다가 되돌릴 때 생기는 오차가 포함되어 있다. 워터마크를 삽입하기 전부터 이미 존재하는 차이라는 뜻이다.

COVER의 가장 큰 제약은 담을 수 있는 정보의 양이다. COVER가 삽입하는 값은 8비트로, 0과 1을 여덟 자리로 늘어놓은 것에 지나지 않는다. 연구진 스스로도 이를 첫 번째 한계로 든다. 여덟 자리 가운데 하나만 어긋나도 확인에 실패하므로, 비트 정확도가 높다고 하더라도 영상 한 편 단위의 판정은 대개 빗나가게 된다. 이 밖에도 의도적인 워터마크 제거 공격, 화면을 자르거나 돌리는 편집, 더 긴 영상과 더 높은 해상도는 이번 평가에서 다루지 않았다.

Ⅲ. 결론: 사후 대응에서 출처 추적으로

• **삽입 위치를 달리한 두 접근의 대응 관계**

앞서 설명한 두 연구는 “압축을 거치고도 워터마크가 남는 삽입 위치가 어디인가?”라는 같은 물음에 서로 다른 답을 내놓았다. 플로우마크는 그 위치를 화면 안에서 찾았다. 어느 영역이 적합한지를 모델이 스스로 판단하게 하고, 어두운 영역에서는 신호를 줄이며, 프레임이 이어질 때의 움직임을 원본에 맞추어 깜박임을 억제했다. COVER는 삽입 위치를 화면 밖으로 옮겼다. 영상 생성 모델이 영상을 간추려 담는 잠재 공간에 워터마크를 삽입해, 압축이 덜어내는 미세한 변화에 흔들리지 않도록 했다.

이 선택의 차이는 각 기술이 할 수 있는 일의 범위를 갈라놓는다. 플로우마크는 한 프레임에 128비트를 담아 영상 식별자와 프레임 번호를 함께 싣는다. 그래서 어떤 영상인지와 더불어 그 안에서 무엇이 바뀌었는지까지 확인할 수 있다. COVER가 삽입하는 값은 8비트에 그쳐 프레임 단위의 확인은 하지 못한다. 대신 압축이 강하게 걸린 조건에서 워터마크 값을 더 잘 지켜 낸다.

시험한 조건도 서로 다른 방향에서 구성되었다. 플로우마크는 압축과 함께 화면을 자르거나 돌리는 변형, 프레임을 빼거나 끼워 넣는 편집, 플랫폼의 재압축까지 다루었다. COVER는 코덱 압축 하나에 집중했다. 서론에서 살펴본 사례는 네트워크 구간에서 워터마크를 삽입해 무단 재배포의 출처를 찾는 방식이었다.

• 두 연구가 남긴 과제와 향후 발전 방향

두 연구는 각자의 한계를 분명히 밝히고 있다. 플로우마크는 학습과 평가가 256×256 크기로 조정된 화면과 155편의 영상, 16프레임 길이의 짧은 구간상에서 이루어졌다. 실제 유통되는 영상은 해상도가 더 높고 재생 시간이 더 길 수 있다. COVER 역시 한 가지 해상도와 정해진 길이의 영상에서만 평가되었으며, 연구진은 더 긴 영상과 더 높은 해상도를 향후 연구 과제로 제시한다.

두 연구의 보고된 실험에서는 의도적인 워터마크 제거 공격에 대한 검증이 충분히 제시되지 않았다. COVER 연구진은 배포 과정에서 일어나는 손상만을 대상으로 삼았고 지우려는 공격은 범위 밖이라고 명시한다. 플로우마크의 시험 조건에도 그러한 시도는 포함되어 있지 않다. 두 기술 모두 유통 과정의 자연스러운 변형을 견디는 데까지 나아갔을 뿐, 의도를 가진 제거에는 아직 답하지 않은 상태다.

COVER 연구진이 제시한 다음 단계의 연구는 삽입하는 값을 늘려 영상 한 편 단위의 판정을 안정시키는 일과, 프레임과 프레임 사이의 관계까지 대체물에 반영하는 일이다. 통제된 조건에서 얻은 수치는 기술이 무엇을 할 수 있는지를 보여 주지만, 실험 수치가 현장에서 그대로 재현되리라고 보장할 수는 없다. 서론에서 살펴본 사례가 이미 도입되어 운영되고 있다는 사실과, 두 연구가 아직 각자의 실험 조건 안에 머물러 있다는 사실은 실용화까지 가야 할 길이 아직 많이 남았음을 의미한다.

참고문헌

- Nick Snow, “Nagravision secures F1 content for RTL Croatia, Pro Plus Slovenia”, Advanced Television, 2026.09.14., https://www.advanced-television.com/2026/09/14/nagravision-secures-f1-content-for-rtl-croatia-pro-plus-slovenia/
- Julian Clover, “Nagravison secures Formula 1 streams for RTL Croatia and Pro Plus Slovenia”, Broadband TV News, 2026.09.14., https://www.broadbandtvnews.com/2026/09/14/nagravison-secures-formula-1-streams-for-rtl-croatia-and-pro-plus-slovenia/
- Vishal Asnani 외 2인, “FlowMark: Mask-Guided Video Watermarking”, arXiv, 2026.07.06., https://arxiv.org/abs/2607.05261
- Yuxin Cao 외 4인, “COVER: Codec-Robust Video Watermarking with Generative Video Priors”, arXiv, 2026.09.17., https://arxiv.org/abs/2609.26236