

# 비디오 스트리밍에서 이벤트 기반의 지각된 비디오 화질

## 추정 기법

### (Event-based Estimation of Perceived Video Quality for Video Streaming)

현종환\*, Yongfeng Huang\*\*, Jin Xiao\*\*, 홍원기\*,\*\*

\* 포항공과대학교 컴퓨터공학과 분산처리 및 네트워크관리 연구실

\*\* 포항공과대학교 정보전자융합공학부

{noraki, xgjonathan, jinxiao, jwkhong}@postech.ac.kr

## 요 약

멀티미디어 서비스를 관리하는 데 있어 네트워크 성능이 사용자 경험에 미치는 영향을 이해하는 것은 매우 중요하다. 본 논문에 제시된 비디오 화질 추정 기법은, 기계 학습 기법을 통해 자동적으로 분류된 비디오 왜곡 이벤트를 바탕으로 사용자가 인식하는 비디오 품질을 추정한다. 이는 사용자 경험이 이벤트 기반이며, 비디오 왜곡 이벤트와 사용자 MOS (Mean Opinion Score)간에 밀접한 상관관계가 있기 때문이다. 실험을 통해, 제시된 기법이 작은 데이터 셋에 대해서도 서로 다른 비디오 왜곡 이벤트를 정확하게 찾아낼 수 있으며, 각각의 왜곡 이벤트가 사용자 경험에 미치는 영향이 서로 다르다는 것을 보인다.

**Keywords:** 비디오 품질 관리, H.264/AVC, QoE, 기계 학습

## 1. 서론

멀티미디어 서비스는 오늘날의 네트워크에서 큰 비중을 차지하고 있으며, 이와 관련하여 발생하는 문제 중 하나가 네트워크를 통해 전송되는 비디오의 품질을 보장하는 것이다. 사용자가 느끼는 체감 품질(Quality of Experience: QoE)을 향상시키기 위해서는 인간이 지각하는 화질에 영향을 끼치는 요소가 무엇인지, 그리고 그것들이 어떻게 연관되어 있는지 이해하는 것이 중요하다. 이러한 요소들 중에는 네트워크와 관련된 요소들(e.g., 패킷 손실, 지연, 지터) 및 어플리케이션과 관련된 요소들(e.g., 초당 프레임 수, 비트레이트, 코덱 종류, 손실 복구 기술), 그리고 사용자 지각과 관련된 요소들(e.g., 비디오 내용에 대한 사용자 선호도)이 있다. 본 논문에서는 네트워크와 관련된 요소들, 그 중에서도 패킷 손실이 비디오 품질에 미치는 영향에 대해 살펴보고자 한다. 이는 지연이나 지터는 플레이어의 재생 버퍼에서 손실로 처리할 수 있기 때문이다.

사용자가 지각하는 비디오 품질 평가와 관련된 기존의 연구들은 측정 가능한 객관적 지표에 의존하고 있다. PSNR (Peak Signal to Noise Ratio), MSE (Mean Square Error), SSIM (Structural SIMilarity) [1], VQM (Video Quality Metric) [2] 등 대부분의 객관적 지표의 경우, 각 프레임 별로 지표가 계산되기 때문에 원본 프레임에 대한 정보가 필요하다. 이러한 접근 방법은 각 지표들 간의 상관관계를 이해하는 데 큰 도움이 되지만, 사람이 지각하는 화질(perceived quality)과 관련된 요소들을 제대로 표현할 수 없다는 문제점이 있다.

객관적 지표 이외에도 QoE를 사용하여 주관적인 비디오 품질 평가가 이루어지며, 이는 사용자의 직접적인 피드백을 통해 얻을 수 있다. 가장 일반적인 평가 방법으로는 MOS가 있으며 사람이 영상의 품질에 대해 1(가장 나쁨)부터 5(가장 좋음) 사이의 값을 매기는 방식이다 [3]. 하지만 MOS는 측정하기가 어려우며, 사람의 주관적인 판단에 의존하기 때문에 콘텐츠 서비스 제공자들이 측정 가능한 지표들로부터 QoE를 예측하기가 쉽지 않다. 객관적 지표와 주관적 지표의 장단점을 고려하면, 객관적 지표를 주관적 지표에 매핑시키는 것은 사용자의 QoE를 추정하는 좋은 방법이 될 수 있다. 이러한 매핑은 객관적

지표뿐만 아니라 사용자의 지각(perception)도 고려한다.

본 논문에서는 VIDEO quality Analyzer in Real-time (VIDAR) 프로젝트 [4]에 사용되는 비디오 인지 모델을 바탕으로 지각된 비디오 화질 추정기법을 제안한다. VIDAR는 분석적 접근 방법을 사용하여 여러 가지 네트워크 환경에서의 비디오 서비스에 대한 사용자 경험을 평가하는 프레임워크이다. 본 논문에서 제안하는 기법은 기계 학습 기법을 사용하여, 프레임 레벨에서의 객관적 지표를 사용자 MOS 특성을 반영하는 인지적 비디오 왜곡(perceivable video defects)과 매핑하는 것이다. 제시된 기법은 아래 두 가지 문제에 초점을 맞추고 있다.

- 기존의 객관적 지표는 사람의 인식과 관련된 요소들을 고려하지 않고 있다. 어떻게 하면 이러한 객관적 지표들을 사용하여 사용자 MOS를 추정하고 그 정확도를 높일 수 있는가?
- 제시된 기법은 실시간 멀티미디어 스트리밍에 적용되는데, 실시간 서비스를 어떻게 보장할 것인가?

이러한 문제를 해결하기 위해 본 논문에서는 다음과 같은 연구를 수행하였다. 먼저, 사용자 경험이 이벤트 기반인 점에 근거하여 프레임 품질 지표를 왜곡 이벤트(defective events)로 그룹핑하였다. 여기서 왜곡 이벤트는 재생 시간 기준으로 연속된 왜곡된 프레임들을 의미한다, 다음으로 다중 클래스 분류법(multi-class classification)을 사용하여 사용자가 느끼게 될 왜곡의 종류를 자동으로 판단하도록 하였다. 확실적인 패킷 손실과 다양한 길이를 가진 왜곡 이벤트로 인해 왜곡 패턴이 복잡하게 형성되기 때문에, 기존의 시계열 분류법(time series classification)은 앞서 제시한 문제들을 해결하기에 부적절하다. 따라서 본 논문에서는 시계열(time series) 데이터에서 서로 다른 왜곡 이벤트들의 특성을 잘 나타내주는 통계적 특성들을 추출하고, 그 중 핵심이 되는 특성들만 선정하였다. 동시에 SVM (Support Vector Machine) [5]의 RBF (Radial Basis Function) kernel에서의 파라미터 선택과 이진 분류기(binary classifier)를 종합하는 과정에서의 방법 선택과 같은 기계 학습 과정에서의 최적화를 수행하였다. 실험을 통해, 제시된 기법이 프레임 레벨에서의 객관적 지표들로 사용자 MOS를 비교적 정확하게 추정할 수 있으며 동시에 그 과정이 간결하면서도 빠르다는 것을 보일 것이다.

본 논문의 구성은 다음과 같다. 2장에서 H.264/AVC 및 관련 연구에 대해 소개하고, 3장에서는 VIDAR에 대해 간략히 설명한다. 4장에서는 왜곡 이벤트를 분류하는데 사용된 알고리즘에 대해 설명한다. 5장에서는 실험 결과를 제시하고 마지막으로 결론과 향후 과제를 제시한다.

## 2. 관련 연구

### 2.1 H.264/AVC

H.264/AVC는 ISO/IEC Moving Pictures Expert Group과 ITU-T Video Coding Experts Group이 공동으로 2003년에 개발한 비디오 코덱으로 현재 최신의 비디오 코덱 표준이다 [6]. H.264/AVC에서 비디오는 일련의 프레임으로 이루어져 있으며, 각 프레임은 여러 개의 슬라이스(slice)로 구성되어 있다. H.264/AVC는 또한 시간적 압축 기술도 포함하고 있는데, 이는 기존의 각 프레임 별 공간적 압축 뿐만 아니라 프레임 간의 중복 데이터도 고려한다는 것이다. 인코딩 과정에서 프레임 간의 예측(Inter prediction) 방식을 사용하기 때문에 비디오 전송 시 네트워크 상에서 패킷 손실이 발생하는 경우 디코딩 과정에서 에러 전파(error propagation)와 같은 문제가 발생할 수 있다. H.264/AVC는 에러 전파에 취약하기 때문에, 하나 이상의 슬라이스를 담은 패킷이 손실되는 경우 해당 슬라이스를 담은 프레임 뿐 아니라 해당 슬라이스를 참조하는 다른 슬라이스에도 영향을 끼치게 된다.

### 2.2 객관적인 비디오 품질 평가 방법

객관적인 비디오 품질 평가(Video Quality Assessment: VQA)는 전송된 비디오에 대한 모든 정보나 일부 정보를 필요로 하는지, 혹은 아무런 정보도 필요로 하지 않는가에 따라 전 기준(Full-reference: FR), 감소 기준(Reduced-reference: RR), 무 기준(No-reference: NR)으로 구분할 수 있다 [7]. FR과 RR은 정확도 측면에서 좀 더 나은 결과를 보여주지만 실제 상황에서는 적용하기 힘들고, NR은 가장 낮은 정확도를 나타내지만 실제 구현에 가장 적합한 방식이다.

비디오 품질 평가를 위한 서로 다른 FR 모델 가운데, PSNR이 가장 간단하면서도 널리 쓰인다. 하지만 [8]에 따르면 인간의 인지 능력과 비교하여 떨어지는 성능을 보인다. VQM은 인간이 인지하는 비디오 품질과 밀접한 상관관계를 가지지만, 높은 정보 처리 역량을 요구하기 때문에 실시간 처리가 힘들다. 널리 쓰이고 있는 또 다른 품질 지표로는, 비디오의 휘도 성분을 프레임별로 계산하는 SSIM 이 있다. SSIM은 정지 이미지 품질 평가(Still Image Quality Assessment)의 지표로서는 좋은 평가를 받지만, 시간적인 요소나 비디오의 내용과 관련된 요소를 고려하지 않는 단점이 있다. 본 연구에서는 SSIM을 객관적인 지표로 사용하고, SSIM이 가진 단점을 보완하기 위해 VIDAR의 주관적 모델(Vidi 모델)을 사용하기로 한다.

### 2.3 객관적 지표로부터 사용자 MOS 추론

비디오 품질에 대한 사용자 인식과 관련된 객관적 지표는 크게 어플리케이션 관련 지표(MSE, PSNR 등)와 네트워크 관련 지표(패킷 손실, 지연 등)로 구분된다.

어플리케이션과 관련된 MSE, PSNR, SSIM 등의 객관적 지표들은 시각적 특성이나 인간 시각 시스템(Human Visual System: HVS)의 인지적 특성을 고려하지 않기 때문에, 이러한 점을 보완하고 지표들을 주관적인 요소들과 매핑시키기 위한 연구들이 많이 이루어졌다.

MSE 기반의 간단한 인지적 지표는 다음 식과 같다 [9]:

$$MOS_p = 1 - k(MSE), \quad (1)$$

여기서  $k$ 는 spatial edge strength이며, Spatial edge는 물체의 가장자리, 표면의 주름과 같은 중요한 시각적 이벤트와 연관되어 있다 [10]. 압축된 비디오의 MOS를 추정하는 데 이 식을 사용할 수 있지만, 네트워크 상황에 따른 비디오 왜곡은 고려되지 않았다.

PSNR은 scaling factor  $a$ 와 shift factor  $b$ 를 affine minimum MSE estimator [11]에 적용시킴으로써 MOS scale로 정규화할 수 있다.

$$\widehat{MOS}_{PSNR}[n] = a \times PSNR[n] + b, \quad (2)$$

여기서  $n$ 은 프레임 번호이다. 정규화 이후, 사람의 인식에 관한 여러 가지 규칙을 적용하여 이 결과를 수정한다. 예를 들어, 점진적인 장면 전환은 최소값을  $k > 1$  와 곱해줌으로써 보상할 수 있다.  $k$ 는 모든 테스트 셋에 linear minimum mean square estimation을 적용하여 얻을 수 있다.

위에서 언급한 통계적 접근 방법들은 한두 가지의 인지적 특성만을 고려하고 있다. 통계적 접근 방법 외에도, 기계 학습이 좀 더 발전된 통계적 기법으로 Adaptive QoE 예측모델에 적용될 수 있다 [12]. SVM과 decision Tree는 모두 잘 알려진 ML 알고리즘으로, 비디오 품질에 대한 사용자의 지각을 예측하는 데 사용된다. 알고리즘에서 사용되는 데이터 인스턴스들은 네 가지 파라미터로 구성되는데, 비디오의 공간적 정보, 시간적 정보, 초당 프레임 수, 그리고 비트레이트이다. 하지만 결과물이 YES, NO로만 표현되어 사용자의 QoE를 포괄적으로 표현할 수 없다.

위에서 언급했던 어플리케이션 관련 지표를 사용한 연구 이외에도 네트워크 관련 지표에 ML 알고리즘을 적용하여 사용자의 QoE를 예측하려는 연구도 있다. Neural network-based reasoner는 링크 상에서 확률적으로 패킷 손실이 발생하는 상황에서 PSNR 기반으로 QoE를 최적화하려는 목적으로 디자인되었다 [13].

어플리케이션 관련 지표나 네트워크 관련 지표를 사용하여 사용자의 QoE를 예측하는 다른 연구들과는 달리, 본 논문에서는 사람의 경험이 대부분 이벤트 중심이라는 점에 착안하여 이벤트 기반의 사용자 QoE 예측 기법을 제안한다. 사용자는 이전 이벤트에 대한 반응을 기억해 냄으로써 현재의 경험에 반응하고 이를 평가하려 한다. 본 논문에서 다루고자 하는 방법은 이와 같이 사람의 인지적 특성을 고려하여 좀 더 나은 QoE 예측을 하고자 하는 것이다.

### 2.4 기계학습(Machine Learning: ML)

기계학습은 크게 supervised ML과 unsupervised ML로 나눌 수 있으며 본 연구에서는 supervised ML을 사용한다. Supervised ML은 주어진 인스턴스로부터 일반적인 가설을 세우는 것을 학습하는 과정이며, 가설을 바탕으로 이후의 인스턴스에 대해서 예측하는 것이다. 이 과정은 다음과 같은 세부 항목으로 구성된다: 데이터 셋 수집, 데이터 전처리(preprocessing), 특성(feature) 생성, 알고리즘 선택, 트레이닝, 테스트 데이터를 사용한 평가. 평가 결과에 따라 위 과정을 반복하며 정확도를 높일 수 있도록 조절한다.

Supervised ML 알고리즘은 logic-based, perceptron-based, statistical learning, SVM으로 구분된다. Logic-based 알고리즘 가운데 decision tree가 가장 많이 사용된다. Decision tree에서 각 노드는 인스턴스의 특성을 나타내며, 각 가지(branch)는 노드가 가질 수 있는 값을 나타낸다. 하지만 decision tree와 같은 logic-based 알고리즘은 수치 특성에 대해서는 제대로 동작하지 않는 문제가 있다. Perceptron-based 기법 가운데 널리 알려진 알고리즘 중 하나인 Artificial Neural Network (ANN)는 실생활 문제에 많이 적용되었으며, 서로 연관성이 적은 특성들에는 매우 비효율적이다. Naïve Bayesian Network (NB)는 statistical learning 알고리즘 중 가장 널리 알려진 알고리즘이다. 가장 큰 장점은 트레이닝에 걸리는 시간이 짧다는 것이며, 하나의 확률분포만을 가지고 클래스들을 분류한다는 점에서는 불완전하다고 여겨진다. Statistical method 가운데 k-Nearest Neighbor (kNN) 또한 널리 알려진 알고리즘이다. kNN은 서로 가까운 거리에 있는 인스턴스들끼리는 특성이 비슷하다는 점에 착안한 알고리즘이다. kNN 역시 서로 연관성이 적은 특성들에 민감하게 반응하지만, 사용하기 쉽고 설정해야 할 파라미터가 nearest neighbor의 개수 하나뿐이다. 가장 최근에

소개된 기법인 SVM은 고차원의 연속적인 특성을 다루는 경우 및 입력 특성과 출력 특성이 서로 non-linear 관계를 가지는 경우에 좋은 성능을 보인다.

각 기법들의 장단점을 비교하였을 때 모든 데이터 셋에 대해 가장 뛰어난 성능을 나타내는 알고리즘은 없다. 알고리즘 선정을 위한 가장 간단한 방법은 동일한 데이터 셋에 대해 여러 개의 후보 알고리즘을 테스트한 후 가장 정확한 결과를 보이는 알고리즘을 선정하는 것이다. 본 연구에서는 수치적 특성을 다루고 있다는 점을 고려하여, 가장 간단한 알고리즘인 kNN과 고차원의 데이터를 다루면서 일반적으로 가장 좋은 성능을 나타내는 SVM을 후보 알고리즘으로 선정하였다.

### 3. 접근 방법

#### 3.1 VIDAR (Video quality Analyzer in Real-time)

VIDAR는 실시간으로 비디오의 품질을 측정할 수 있는 시스템으로, 클라이언트에서 관찰되는 네트워크 QoS 조건들을 입력 받아 그것이 비디오 프레임의 품질에 미치는 영향과 비디오 품질에 대한 사용자의 인식에 미치는 영향을 예측할 수 있다(그림 1).

R3 모델은 클라이언트에 설치된 간단한 모니터 프로그램과 서버의 R3 Analyzer로 구성된다. 클라이언트 모니터는 비디오 디코더를 수정한 것으로 Error-correlation trees를 생성하여 서버의 R3 Analyzer로 전송한다. 네트워크에서 패킷 손실이 발생하면 해당 비디오 프레임에 에러가 발생하며, 이 프레임을 참조하는 GOP내의 다른 프레임에도 에러가 전파된다. 이러한 GOP 내에서의 에러 전파 과정을 tree 형태로 나타낸 것이 Error-correlation trees이다. 이를 분석함으로써 클라이언트에 전송된 비디오에 접근하지 않고도 서버에서 SSIM을 추정할 수 있으며, 이렇게 추정된 SSIM을 eSSIM이라고 정의한다 [4].

Vidi 모델은 각 프레임의 eSSIM과 왜곡된 비디오의 특정 이벤트를 매핑시킨다. 이러한 디자인이 필요한 이유는 다음과 같다. 각 프레임의 이미지 품질을 추정하는 것만으로는 사용자가 느끼는 비디오 품질을 예측하기 힘들다. 그 이유는 사용자가 느끼는 비디오의 품질은 비디오의 내용에 따라 다르기 때문이다. 예를 들어, 어두운 장면에서 발생한 비디오 왜곡은 사람의 눈에 잘 띄지 않으며 [1], 비디오의 잔상은 장면 전환 직후에 나타나는 프레임의 왜곡을 감출 수 있기 때문이다 [14]. Vidi 모델에서는 다음의 세 가지 특징을 고려한다: 휘도, 프레임 복잡도, 장면 전환. 이 세 가지 요소들을 조합함으로써 여러 가지 비디오 콘텐츠를 설명할 수 있다(e.g., 뉴스, 스포츠, 액션 및 드라마). Vidi 모델은 그 결과로 Video Index(Vidi) metric을 생성한다. Vidi metric은 탐지된 왜곡 이벤트의 유형과 횟수 및 그 강도(intensity)로 구성된다.

사용자 모델은 Vidi metric과 사용자 MOS를 매핑시켜 준다. 이 매핑은 얼마나 많은 종류의 왜곡이 사용자의 QoE와 그 강도에 영향을 끼치는지에 달려있다.

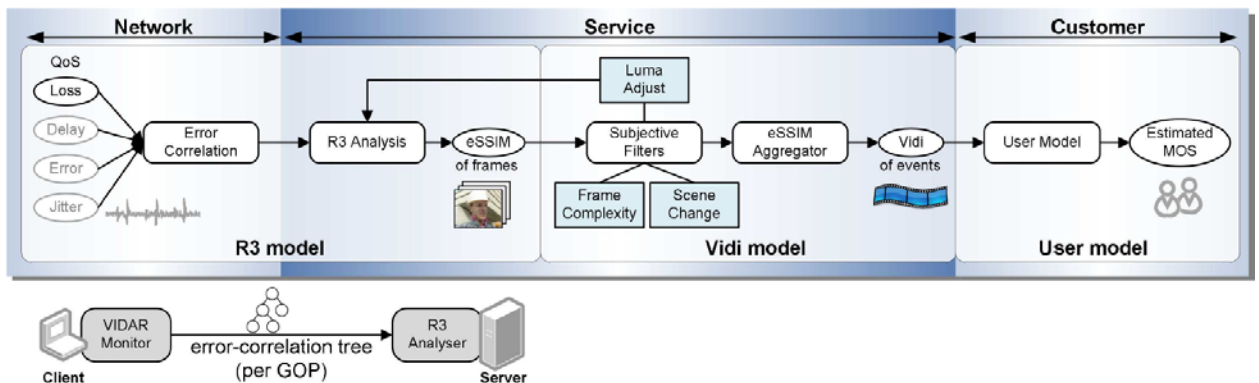


그림 1 Overview of VIDAR Framework

#### 3.2 이벤트 기반 분류 (Event-based classification)

이전 연구에 의하면 [4], 전송된 비디오의 왜곡은 다음과 같이 분류할 수 있다.

- **Distortion:** 인지할 수 있는 왜곡을 가진 프레임
- **Glitch:** distortion이 발생한 프레임이 연속적으로 나타나는 것.
- **Freezing:** 복제된 프레임이 연속적으로 나타나는 것, 혹은 손실된 프레임이 interleaving pattern으로 복제되는 것.
- **Discontinuity:** 연속하지 않은 두 프레임이 차례로 재생되는 것

이러한 왜곡 이벤트들은 eSSIM을 콘텐츠 특성 필터를 적용시킨 결과값을 사용하여 만들 수 있다(그림 1). 그 후 각각의 이벤트에 대해 인위적으로 라벨을 붙인다. 하지만 이 작업은 비효율적일 뿐만 아니라

오류가 발생할 가능성이 높다. 따라서 기계 학습 기법을 사용하여 왜곡 이벤트들을 자동으로 분류할 수 있는 시스템을 개발하는 것이 필요하다.

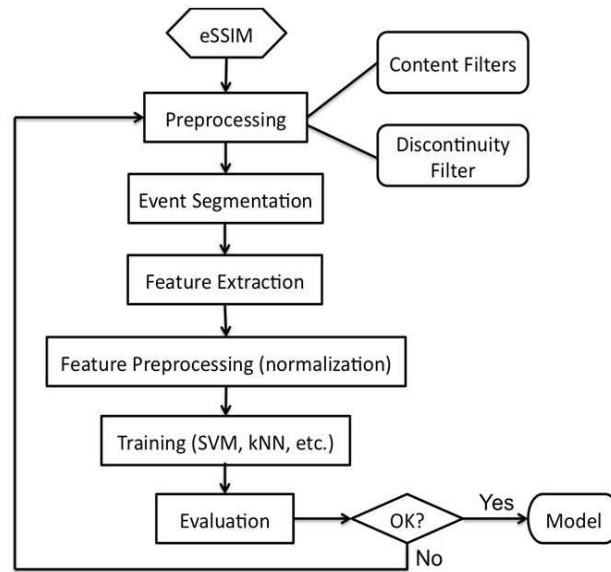


그림 2 eSSIM aggregator의 작업 흐름도

기본적인 분류 방법은 그림 2에 나타나 있으며, 기계학습 기법을 적용할 데이터 준비 단계(eSSIM Data Preprocessing, Event Segmentation, Feature Extraction, Feature Preprocessing)와 기계학습 기법 적용 및 최적화 단계(Training, Evaluation)로 구분된다. 본 연구에서는 여러 가지 왜곡 이벤트를 다루기 때문에 다중 클래스 분류 기법을 도입하였다. 이진 분류 알고리즘 후보군 가운데 ANN와 kNN이 다중 클래스 분류 기법으로 확장하기 용이하며, 이진 분류기인 SVM을 조합하여 다중 클래스 문제를 해결하였다.

#### 4. Defective Event의 분류

본 장에서는 다중 클래스 분류법을 사용하여 왜곡 이벤트를 분류할 수 있는 방법을 제시한다(그림 2). 정확도를 높이는 동시에 분류 작업을 효율적으로 진행하기 위해 여러 가지 최적화 기법이 적용되었다(e.g., SVM kernel에서의 파라미터 선택).

##### 4.1 eSSIM 데이터의 전처리

영상의 콘텐츠에 따라 사람의 지각 정도가 달라지는 것을 고려하여, eSSIM 데이터는 휘도, 프레임 복잡도, 장면 전환의 세 가지 콘텐츠 필터를 거친다(그림 1). 하지만, 분류의 정확도를 높이기 위해, 분류에 도움이 되는 몇 가지 요소들이 데이터에 추가될 수 있다. 세 가지 왜곡 타입 중, 모든 distortion, glitch 및 freeze는 eSSIM 데이터 내에 각각의 특징을 지니고 있다. 비교적 작은 eSSIM 값(eSSIM값은 0과 1 사이의 값을 가질 수 있으며, 1은 원본 프레임과 전송된 프레임이 완벽하게 일치하는 것을 의미한다. 0에 가까운 값은 distortion의 정도가 심한 것을 의미하며, 0은 해당 프레임이 중복되었거나 손실되었음을 의미한다.)을 가진 프레임이 연속되는 경우 distortion이 발생하였을 가능성이 높으며, glitch는 distortion의 경우보다 eSSIM값은 조금 더 크고 연속된 프레임의 개수가 더 적은 경우 발생할 수 있다. 그리고 freezing은 주로 복제된 프레임(eSSIM = 0)이 연속적으로 나타나는 경우 발생한다. Discontinuity와 다른 왜곡 타입과의 차이점은 discontinuity만 두 프레임끼리의 비교를 필요로 한다는 점이다. 이전 실험에 의하면, discontinuity는 항상 freezing과 함께 발생하며, 주로 다음과 같은 두 가지 경우에 discontinuity를 느낄 수 있다.

- **프레임 손실**: 만약 손실된 프레임의 이전 프레임과 이후 프레임이 서로 눈에 띄게 차이 나는 콘텐츠를 갖고 있을 경우 discontinuity를 느낄 수 있다.
- **프레임 복제**: 일련의 프레임이 그 이전의 한 프레임을 복제하는 경우, 그리고 그 이후의 프레임이 복제된 프레임과 눈에 띄게 차이 나는 콘텐츠를 담고 있을 경우 discontinuity를 느낄 수 있다.

eSSIM이 discontinuity 특성을 표현할 수 있도록 다음과 같은 discontinuity 지표를 추가하였다.

$$eSSIM_{disc} = ssim(frame_b, frame_a) - 1 \quad (3)$$

$ssim()$ 은 복제되거나 손실된 프레임의 이전 프레임( $frame_b$ )과 이후 프레임( $frame_a$ )의 SSIM값을 계산해주는 함수이다. SSIM이 0 ~ 1 사이의 값을 가지기 때문에  $eSSIM_{disc}$ 는 -1 ~ 0 사이의 값을 갖는다. 따라서 그 값이 0에 가까울수록  $frame_b$ 와  $frame_a$ 의 차이가 작다고 볼 수 있다. 위에서 설명한 두 상황에 대해 이 접근 방법을 적용하면 다음과 같이 정리할 수 있다.

- **프레임 손실**: 손실된 프레임의 원래  $eSSIM$  값 대신  $eSSIM_{disc}$ 를 사용한다.
- **프레임 복제**: 복제된 프레임에 대해서는  $eSSIM$ 값을 0으로 두고 복제된 이후에  $eSSIM_{disc}$ 를 더해준다.

이 작업은 두 가지 의미를 갖는다. 먼저,  $eSSIM_{disc}$ 를 계산하는 작업이 서버 측에서 이루어지는데, 이것은  $frame_b$ 와  $frame_a$ 를 원본 비디오로부터 얻을 수 있다는 점이다. 두 번째로는, 인간 시각 시스템 특성 상 장면 전환 직전의 몇 프레임은 제대로 인식할 수 없다 [14]. 따라서 위 작업을 이미 장면 전환 필터를 통과한  $eSSIM$ 값에 적용할 수 있다.

#### 4.2 이벤트 세분화 (Event segmentation)

이벤트 기반의 분류는 인간의 경험이 이벤트 기반이며 이전 이벤트들에 대해 반응하고 평가한다는 점에 착안하여 디자인되었다. 또한, 이벤트의 길이(본 논문에서 이벤트의 길이는 한 이벤트를 구성하는 프레임의 개수를 뜻한다)는 확장성(*scalability*)을 가지며 두 이벤트의 경계는 비교적 거리를 두고 있다. 하지만 이벤트 세분화는 기계 학습 트레이닝과 테스트를 위한 인스턴스를 생성하는 데 있어 필수적인 요소이다. 따라서 이전까지의 실험 결과에 따라 (이전 실험에서 *frame rate*는 초당 30프레임, 비트레이트는 초당 800 비트로 설정하였다) 비디오  $eSSIM$  시퀀스로부터 이벤트 인스턴스를 뽑아내기 위해 다음과 같은 룰을 설정할 수 있다.

이벤트의 길이는 10 ~ 100 프레임 사이이며, 서로 다른 두 이벤트 사이의 간격은 10 프레임 이상이어야 한다.

#### 4.3 특성 추출 (Feature extraction)

왜곡 이벤트는 시계열 데이터와 같이 다를 수 있다. 기존의 시계열 데이터에 대한 분류 알고리즘들은 주로 Euclidean Distance와 같은 거리 측정 방식을 주로 사용하였다. 하지만 이러한 방법은 본 논문에서 해결하고자 하는 문제에는 적용하기 힘들다. 그 이유는 우선 시계열 데이터가 고차원이기 때문에 연산 속도를 떨어뜨리는 문제를 야기한다. 또한, 실제 포인트들을 대상으로 알고리즘을 수행하는 경우, 데이터가 일부 누락되거나 노이즈가 존재하는 경우에는 다루기 힘든 특징이 있다. 그리고 서로 다른 길이의 데이터를 처리할 수 없으며 1:1 정렬을 요구한다. 하지만 본 연구에서는 이벤트들이 동일한 시작점을 갖고 있지 않으며 특정한 패턴을 따르지도 않는다. 이러한 특징이 나타나는 이유는 실험에 사용된 패킷 손실 모델, GOP의 크기, 이벤트 세분화 방법 때문이다. 따라서 데이터 누락과 노이즈를 포함한 서로 다른 길이를 가진 시계열 데이터를 다루기 위해서는, 통계학적인 접근을 통해 이벤트 데이터로부터 특성을 추출해야 한다. 기본적으로 왜곡 이벤트로부터 추출되는 특성들은 몇 가지 요소가 고려되어야 한다. 첫째, 이벤트 데이터의 일반적인 특성을 가장 잘 나타내야 한다. 둘째, 이벤트간에 유사점과 차이점을 명확히 판별할 수 있어야 한다. 마지막으로 이벤트의 길이가 서로 다르기 때문에 정규화된 방식으로 계산될 수 있어야 한다.

- 이벤트 데이터로부터 다음과 같은 특성들을 추출하였다. 1) 평균( $\mu$ ), 2) 표준 편차( $\sigma$ ), 3) 최소값, 4) 왜곡 비율(*defective ratio*):

$$ratio = \frac{N_{eSSIM < 0.95}}{n} \quad (4)$$

여기서  $N_{eSSIM < 0.95}$ 은  $eSSIM < 0.95$ 인 프레임의 개수이다. 이 수치는 왜곡의 정도를 프레임 개수를 통해 나타낸다.

- 5) 누락된 프레임과 중복된 프레임의 정도(*severity*):

$$severity = \frac{N_{eSSIM \leq 0.0}}{n} \quad (5)$$

여기서  $N_{eSSIM \leq 0.0}$ 은  $eSSIM \leq 0.0$ 인 프레임의 개수이다. 이 수치는 *freezing*의 정도를 나타낸다.

6) 왜도(*skewness*): 이벤트 데이터 확률 분포의 비대칭성을 측정하기 위한 특성이다. 이 값이 음수이면 확률 분포가 오른쪽으로 치우쳐져 있다는 것을 뜻하고, 양수일 때는 반대로 왼쪽으로 치우쳐져 있다고 볼 수 있다. 따라서 왜도와 아래 설명할 첨도는 이벤트 데이터의 확률 분포에 기반하여 계산되어야 한다. 이전 실험에 의하면, 다음과 같은 유닛에 의해 생성된다. [-1.0, 0.0, 0.1, 0.5, 0.8, 0.9, 0.95, 0.98, 1.0].

7) 첨도(kurtosis): 정규분포와 비교하여, 이벤트 데이터의 확률 분포의 형태(flat or peak)를 나타내주는 척도이다.

위에서 설명한 통계적 특성들을 고차원의 서로 다른 길이를 가진 왜곡 이벤트 데이터 셋에 적용하면 제한된 개수의 특성을 가진 새로운 데이터 셋을 얻을 수 있다.

#### 4.4 특성 전처리 (Preprocessing features)

앞에서 설명한 일곱 가지 특성들을 생성하는 도중, 각 수치들을  $n$ 으로 나누어 정규화하여 계산하는 방법도 고려하였다. 하지만 왜도와 첨도의 범위는 다른 특성들과는 상이하기 때문에 전체적인 분류 효율을 떨어뜨릴 수 있다. 예를 들어 Euclidean Distance 방법을 사용하면, 모든 차원을 동일하게 놓고 이를 kNN 알고리즘에 적용하여 서로 다른 두 인스턴스의 거리를 측정하기 때문에, 계산된 왜도와 첨도는 결과에 큰 영향을 준다. 따라서 다음 단계로 넘어가기 전에 데이터 셋을  $[-1, 1]$ 의 범위로 다시 정규화할 필요가 있다.

#### 4.5 특성 축소 (Feature reduction)

데이터 셋의 사이즈를 제한하기 위해 일곱 가지의 특성만을 추출하였지만, 이 특성들이 완전히 독립적이지는 않다. 예를 들어, 높은 왜곡률은 높은 평균 값을 수반한다. 여러 특성들간의 의존성은 종종 분류의 정확도를 떨어뜨린다 [15]. SVM과 비교하여 kNN은 노이즈가 포함된 데이터나 관계가 적은 특성들에 대해 민감하다 [5]. 따라서 특성들에 대한 greedy forward selection 알고리즘을 kNN에 적용하였다. Greedy forward selection 알고리즘은 가장 독립적인 특성을 하나 선택하고, 남은 특성들 가운데 가장 독립적인 특성들을 차례대로 선택하여 이미 선택된 특성들과 합쳐나가는 방식으로 동작한다. 최종적으로 모든 조합 가운데 데이터 셋을 가장 잘 분류해낼 수 있는 조합 하나가 선택된다.

#### 4.6 다중 클래스 분류 (SVM)

SVM은 여러 분야에서 널리 사용되고 있으며 성능도 뛰어난 분류 알고리즘이다. 입력 값과 출력 값 사이에 non-linear한 관계가 성립될 때 잘 동작하며 고차원의 데이터를 처리할 수 있다는 장점이 있다. 알고리즘의 성능은 다음과 같은 요소에 의해 결정된다: 데이터 전처리, kernel selection, SVM과 kernel의 파라미터 설정. 데이터 전처리에 대해서는 이미 다루었고 본 장에서는 kernel selection과 파라미터 설정에 대해 설명하고자 한다.

**Kernel selection:** SVM의 kernel function은 데이터 셋 인스턴스를 original feature space에서 고차원의 feature space로 매핑시키는 역할을 한다. 그 후 고차원 feature space에서 서로 다른 두 클래스를 분리하기 위해 decision boundary가 생성된다. Kernel function은 linear(i.e., linear kernel)와 non-linear(i.e., polynomial kernel, radial basis function (RBF))로 구분할 수 있다. 5장에서 최적의 파라미터를 설정하여 수행한 실험에 따르면, linear kernel은 가장 나쁜 결과(정확도:  $74.45\% \pm 0.63\%$ )를 보인데 비해 RBF kernel은 가장 좋은 결과(정확도:  $85.46\% \pm 0.29\%$ )를 나타냈으며, polynomial kernel은 두 번째로 좋은 결과(정확도:  $83.88\% \pm 0.36\%$ )를 보였다. Linear kernel이 나쁜 결과를 보여준 원인은 실험 데이터가 고차원의 feature space에서도 linear하게 분리할 수 없기 때문이다. 따라서 분류를 위해 RBF kernel을 선택하였으며, RBF kernel은 다음 수식으로 나타낼 수 있다.

$$k(\vec{x}, \vec{x}_i) = \exp(-\gamma \|\vec{x} - \vec{x}_i\|^2) \quad (6)$$

여기서  $\gamma = 1/2\sigma^2$ 이다.

**파라미터 설정:** RBF kernel을 사용하는 SVM은, RBF kernel에 사용되는 파라미터  $\gamma$ 와 SVM에 사용되는 파라미터  $C$ 를 설정하여야 한다.  $C$ 는 soft margin constant와 penalty factor로 사용되며, 각각의 인스턴스를 다르게 분류할 때 얼마나 엄격하게 분류할 지 조절하는 역할을 한다.  $\gamma$ 는 decision boundary의 탄력성(flexibility)을 조절한다.  $\gamma$ 가 작을 때는 decision boundary가 거의 linear하다.  $C$ 와  $\gamma$ 값이 작을 경우 underfitting이 발생하며,  $\gamma$ 값이 매우 큰 경우에는 overfitting이 발생한다.  $C$ 와  $\gamma$ 의 최적의 조합을 찾기 위해서는 복잡도가  $O(n^2)$ 인 grid search가 주로 사용된다. 우리가 제시한 모델은 실시간 스트리밍을 지원해야 하기 때문에 grid search는 적절하지 못하다. 하지만  $C$ 와  $\gamma$ 의 최적의 조합은 local area에 존재하는 것을 발견하였다. 따라서 Linear kernel SVM은 baseline으로 사용될 수 있다 [16]:

$$\log \sigma^2 = \log C - \log \tilde{C} \quad (7)$$

여기서  $\tilde{C}$ 는 Linear kernel SVM의 penalty parameter이다.  $C$ 와  $\gamma$ 의 최적의 조합을 가진 baseline은 위 식에 의해 정의되며  $\tilde{C}$ 는 linear SVM에 의해 최적화된다. 따라서,  $C$ 와  $\gamma$ 은 다음의 과정을 거쳐 얻을 수 있다. 1) Linear kernel SVM의 최적의  $\tilde{C}$ 값을 찾는다. 2) 1)에서 찾은  $\tilde{C}$ 값에 기반한 등식을 만족하는  $C$ 와  $\gamma$ 의 최적의

조합을 찾는다. 시간 복잡도는  $O(n)$ 으로 줄어든다.

**이진 분류기 조합:** 다중 클래스를 분류하는 조건을 만족하기 위해서는 여러 개의 이진 분류기를 조합하는 방법이 일반적으로 사용된다. 주로 1:All (OVA)과 1:1 (OVO)의 두 가지 조합 방법이 사용된다. OVA는  $M$ (클래스의 개수) 개의 이진 분류기를 조합하며, 각각의 분류기는 클래스를 positive class로, 남은 것을 negative class로 설정한다. OVO는  $M(M-1)/2$ 개의 이진 분류기를 조합하며, 각각의 분류기는 모든 클래스 중 두 개만 상호배타적으로 선택하여 하나는 positive class, 다른 하나는 negative class로 설정한다. 예를 들어, 본 논문에 사용된 데이터 셋에서는 distortion vs. glitch, distortion vs. freezing, glitch vs. freezing 세 개의 OVO 분류기를 사용하였다. 비교적 클래스 크기가 작고 샘플 사이즈가 편중되어 있기 때문에 [17], 본 논문에서는 OVO를 다중 클래스 분류법의 조합 방법으로 선택하였다.

#### 4.7 다중 클래스 분류(kNN)

kNN은 feature space에서 두 포인트의 상대적인 거리를 비교하여 포인트들을 분류하는 방법이다. 기본적인 원리는 가까운 거리에 있는 두 인스턴스는 특성이 비슷하다는 것이다. 이 때문에 kNN은 노이즈가 포함된 인스턴스나 불균형한 특성들에 민감하다. kNN을 사용할 때는  $k$ 와 거리 함수 두 가지를 결정해야 한다.  $k$ 값이 작은 경우 노이즈가 포함된 인스턴스에 더 민감하지만 클래스를 더 작은 영역으로 맞출 수 있다. 거리 함수는 같은 클래스에 포함된 인스턴스들 간의 거리를 최소화하고 다른 클래스에 포함된 인스턴스들 간의 거리를 최대화한다. 여기서는 grid search를 사용하여 최적의  $k$ 와 거리 함수를 선택하였다(Minkowsky, Manhattan, Chebychev, Euclidean) [5].

### 5. 실험

본 장에서는 실험 결과에 대해 설명한다. 먼저, 실험에 사용된 데이터 셋과 그 생성 방법에 대해 설명하고, 다중 클래스 분류의 결과를 비교한 후 본 논문에서 제시한 최적화 방법이 분류 효율에 미치는 영향에 대해 살펴본다. 마지막으로, 제시한 분류 방법과 사용자 MOS와의 관련성에 대해 살펴본다.

실험에 사용될 비디오는 서버에서 VLC media player의 x264 코덱을 사용하여 H.264/AVC로 인코딩하였다. 인코딩된 비디오는 RTP over UDP 를 사용하여 클라이언트로 스트리밍된다. 클라이언트는 스트리밍되는 비디오를 FFmpeg으로 디코딩한 후 VLC media player로 재생한다. 실험에는 총 다섯 가지 종류의 비디오가 사용되었다(표 1). GOP 의 크기가 비디오의 왜곡 정도에 영향을 끼치기 때문에 [4], 인코딩 시 각각의 비디오의 GOP크기를 두 가지(12, 24)로 설정하였다. 또한, 다양한 네트워크 패킷 손실 상황을 시뮬레이션하기 위해, 실험에서는 Bernoulli uniform loss model과 Gilbert-Elliott (GE) burst loss model 두 가지 패킷 손실 모델을 사용하였고, 패킷 손실률은 두 모델 모두 2%로 설정하였다. 실험을 통해 일반적으로 GE모델이 uniform 모델보다 더 심한 왜곡을 생성한다는 점을 알 수 있었다. 또한, GE모델은 더 많은 freezing을 유발하며, uniform 모델은 glitch를 더 많이 유발한다.

표 1 실험에 사용된 비디오

제목	장면 전환	움직임	내용
Foreman	0	중간	인물 및 배경
Mother and Daughter	0	없음	인물
Football	0	빠름	스포츠
Bus	0	중간	움직이는 차량
Inception	6	빠름	다양함

각 비디오 별로 두 가지 패킷 손실 모델 (Bernoulli uniform loss model, GE burst loss model)과 두 가지 GOP 값을 사용하여 실험을 진행하였다. R3 모델을 거쳐 전송된 비디오로부터 eSSIM 값들을 얻을 수 있고, 여기에 content filter와 discontinuity filter를 적용시킨 후 왜곡 이벤트를 탐지하였다. 그 결과 157 개의 왜곡 이벤트가 탐지되었으며, 그 중 distortion 이 87개, glitch가 27개, 그리고 freezing이 43개이다. 이러한 왜곡 이벤트들은 수동으로 라벨링되었다.

실험에 사용될 네 개의 다중 클래스 분류법을 정확도와 시간을 기준으로 비교한 결과는 아래와 같다.

- **SMO-G:** Sequential Minimal Optimization (SMO)에 RBF kernel을 사용하였고, grid search를 사용하여  $C$ 와  $\gamma$ 의 최적의 조합을 찾는다. SMO는 SVM의 트레이닝 과정에서 발생하는 최적화 문제를 효율적으로 해결하기 위한 알고리즘이다 [18].  $C$ 와  $\gamma$ 의 범위는 다음과 같다.

$$C \in \{2^0, 2^{0.5}, \dots, 2^{11.5}, 2^{12}\}, \gamma \in \{2^{-5}, 2^{-4.5}, \dots, 2^{6.5}, 2^7\}$$



- **SMO-L**: Linear kernel을 사용한 SMO이다. Linear kernel에서는 C만 설정하면 되며, 따라서 최적의 C 값을 찾기 위해 fine-grained 기법을 사용한다 ( $C \in \{2, 4, \dots, 198, 200\}$ ).
- **SMO-O**: RBF kernel을 사용한 SMO이며, C와  $\gamma$ 의 최적의 조합을 찾기 위해 4.6절에 설명된 방법을 사용한다. 먼저, SMO-L에 의해 선정된 최적의 C값을 사용하여  $\tilde{C}$  값을 선택한 후, fine-grained 기법을 사용하여 C와  $\gamma$ 의 최적의 조합을 찾는다 ( $\gamma \in \{0.05, 0.10, \dots, 1.95, 2.00\}$ ).
- **kNN**: k와 거리 함수의 최적의 조합을 찾기 위해 grid search를 사용하였다.

위 네 가지 방법 각각 교차 검증(cross-validation)을 수행하였다. 다중 클래스 분류법을 Weka 오픈소스 라이브러리 [19]를 사용하여 구현하였으며, 분류 결과는 표 2와 같다. 표를 보면, SMO-G가 가장 높은 정확도를 나타내나, 다른 방법들에 비해 시간이 상당히 많이 소요된다. 반면, SMO-O는 SMO-G와 거의 비슷한 정확도를 보이면서도 소요되는 시간은 훨씬 적은 것을 알 수 있다. 소요되는 시간만으로 성능을 평가한다면 kNN도 비교적 높은 정확도를 보인다. 하지만 kNN은 lazy-learning 알고리즘이므로 실행 시간이 트레이닝 시간에 비해 훨씬 많이 소요되며 데이터 셋의 크기가 클 경우에는 그 차이가 훨씬 커진다.

표 2 분류 결과

	SMO-G	SMO-L	SMO-O	kNN
정확도	85.46% $\pm$ 0.29%	74.45% $\pm$ 0.63%	84.57% $\pm$ 0.29%	83.00% $\pm$ 0.33%
시간(초)	445.2 $\pm$ 16.0	66.7 $\pm$ 2.7	94.5 $\pm$ 4.5	31.6 $\pm$ 3.2

위 실험에서는 실험의 편의를 위해 각 이진 분류기의 파라미터를 균일하게 설정하였다. 아래에 각각의 이진 분류기의 정확도에 영향을 끼치는 요인을 살펴보고자 한다. 표 3을 보면, glitch vs. freeze가 가장 높은 정확도를 보이는데, 이는 glitch와 freeze를 가장 잘 구분해낼 수 있다는 것을 의미한다. Distortion vs. glitch의 정확도가 낮게 나타난 이유는, 두 유형 간의 분류가 영상의 내용과 관련 있기 때문이다. 예를 들어, 프레임을 디코딩하는 과정에서 같은 횟수의 에러가 발생하였을 경우, 영상에서 배경에서 발생하는 왜곡보다 주 물체(e.g., Mother and Daughter 영상에서 어머니의 얼굴)에 발생하는 왜곡을 더욱 잘 인식할 수 있다. 반면, distortion vs. freezing의 정확도가 가장 낮은 이유는 두 왜곡 이벤트가 종종 같이 발생하기 때문이다.

표 3 이진 분류기의 정확도

	Dist. vs. Glitch	Dist. vs. Freeze	Glitch vs. Freeze
정확도	92.04%	85.38%	97.18%

그림 3은 각 이진 분류기가 일곱 가지 특성을 구분함에 있어 각자 어느 정도의 정확도를 보이는지 나타낸 그래프이다. 점선은 특성을 제거하지 않은 상태로 분류한 결과이고, 실선은 각각의 특성을 제거한 후 분류한 결과이다. 그래프에서 알 수 있듯이, 하나의 특성을 제외하는 것으로는 모든 이진 분류기의 정확도를 높일 수 없다. 이는 다중 클래스 분류법을 사용하기 위해서는 일곱 가지 특성 모두가 사용되어야 한다는 것을 의미한다. 하지만, 일부 분류기는 특성을 제거하는 것에 대해 민감하게 반응한다. 예를 들어, distortion vs. glitch는 ratio 특성에 대해 민감한데, 이는 glitch가 일반적으로 distortion에 비해 아주 작은 ratio 값을 가지기 때문이다. 따라서 ratio는 glitch와 distortion을 구분하는 데 필수적인 특성이다.

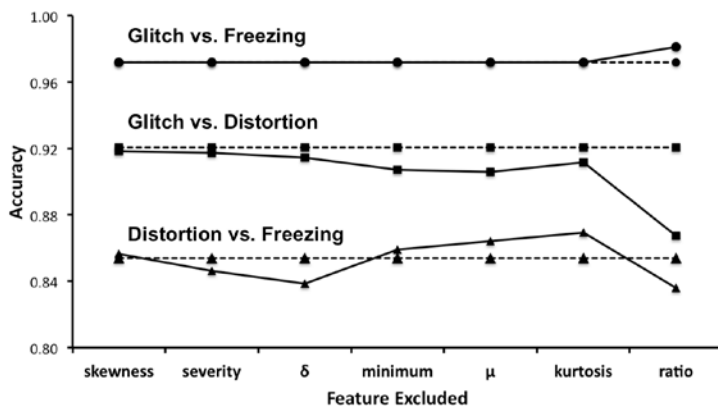


그림 3 서로 다른 이진 분류기의 특성 민감도

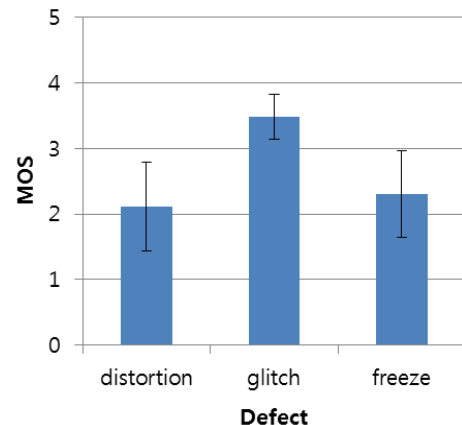


그림 4 왜곡 종류별 MOS

아래에 왜곡 이벤트가 사용자 MOS을 얼마나 잘 예측하는지 살펴본다. 실험은 다섯 명의 피실험자를 대상으로 진행되었으며, 데이터 셋 가운데 30개의 왜곡 이벤트를 임의로 선정하여 테스트 셋으로 삼았고, glitch, distortion 및 freezing을 각각 10개씩 선정하였다. 각 피실험자에게 임의의 순서로 정렬된 30개의 왜곡 이벤트를 보여준 후, 각 이벤트에 대해 MOS를 선정하도록 하였다. MOS의 평균값은 그림 4에 나타나 있다. 그 결과, distortion이 비디오 품질 인식에 가장 나쁜 영향을 주는 것으로 나타났으며 ( $2.11 \pm 0.67$ ) glitch는 비디오 품질 인식에 큰 영향을 주지 않는 것으로 나타났다 ( $3.48 \pm 0.34$ ). 이는 앞에서 정의한 왜곡 이벤트와도 일치하는 결과이다. 또한, freezing 직후에 발생하는 discontinuity가 낮은 MOS ( $2.3 \pm 0.66$ )의 원인이지만, 몇몇 경우에는 freezing 이후에 distortion도 함께 발생하며 이것 역시 MOS 값을 낮추는 데 기여했다.

정리하여 보면, 약 85% 정도의 정확도를 가지는 SMO-O는 eSSIM 값들로부터 추출한 특성들이 비교적 적은 트레이닝 셋에 대해서도 왜곡 이벤트의 공통점과 차이점을 판별해낼 수 있는 것으로 나타났다. 이는 기계학습 기반의 접근의 가장 큰 문제점인 큰 트레이닝 데이터 셋의 필요성을 경감해주는 효과가 있다. 마지막으로, 서로 다른 왜곡 이벤트가 사용자의 경험에 서로 다른 정도의 영향을 준다는 것을 밝혀냈다. 따라서 이벤트 클래스를 기반으로 왜곡을 MOS와 매핑시키는 것이 적절하다고 판단할 수 있다.

## 6. 결론

본 논문에서는 기계 학습 기법을 사용한 이벤트 기반의 지각된 비디오 화질 추정기법을 제안하였다. 이 기법에서는 다른 기계 학습 기반의 접근 방식과는 달리, 사용자의 경험이 이벤트에 기반한다는 점에 초점을 맞추었다. 구현의 편의를 위해 특성 선택, 파라미터 최적화 및 ML 분류방법 선택 등의 성능 향상에 초점을 맞추었다. 실험 결과, 분류되는 이벤트와 가장 큰 연관성을 가지는 특징을 선택하는 것이 중요하며, 주관적 필터를 적용하는 것은 데이터의 전처리를 위한 방법임을 알 수 있었다. 향후 연구에서 보다 많은 실험을 수행하고, 사용자 MOS를 예측하는 방법에 관하여 연구를 진행할 계획이다.

## 참고 문헌

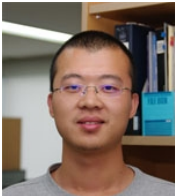
- [1] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.
- [2] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, September 2004.
- [3] ITU-T, "Methods of subjective determination of transmission quality," ITU-T Recommendation P.800, 1996.
- [4] A. Kwon, J. Xiao, S. S. Seo, J. W.-K. Hong, and R. Boutaba, "The impact of network performance on perceived video quality and user experience in H.264/AVC," in *IEEE/IFIP Network Operations and Management Symposium (NOMS), mini-conference*, 2012.
- [5] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, 2007.
- [6] T. Wiegand, G.-J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.
- [7] ITU-T, "User requirements for objective perceptual video quality measurement in digital cable television," ITU-T Recommendation J.1443m, 2000.
- [8] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, December 1995.
- [9] A. Bhat, I. Richardson, and S. Kannangara, "A new perceptual quality metric for compressed video," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009.
- [10] X. Ran and N. Farvardin, "A perceptually motivated three-component image model - part 1: Description of the model," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 401–415, April 1995.
- [11] O. Nemethova, M. Ries, M. Zavodsky, and M. Rupp, "PSNR-based estimation of subjective time-variant video quality for mobiles," in *MESAQUIN*, 2006.
- [12] V. Menkovski, A. Oredope, A. Liotta, and A. Cuadra, "Predicting quality of experience in multimedia streaming," in *International Conference on Advances in Mobile Computing and Multimedia*, 2009.
- [13] S. Latre, P. Simoens, B. D. Vleeschauwer, W. V. D. Meerssche, F. D. Turck, B. Dhoedt, P. Demeester, S. V. D. Berghe, and E. D. de Lumley, "An autonomic architecture for optimizing QoE in multimedia access networks," *Computer Networks*, vol. 53, pp. 1587–1602, July 2009.
- [14] A. R. Reibman and D. Poole, "Predicting packet-loss visibility using scene characteristics," in *Packet Video*, November 2007.
- [15] S. Markovitch and D. Rosenstein, "Feature generation using general construction functions," *Machine Learning*, vol. 49, pp. 59–98, 2002.
- [16] S. S. Keerthi and C.-J. Lin, "Asymptotic behavior of support vector machine with gaussian kernel," *Neural Computation*, July 2003.

- [17] F. Provost, "Learning with imbalanced data sets 101," in AAI 2000 Workshop on Imbalanced Data Sets, 2000.  
 [18] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in Advances in Kernel Methods - Support Vector Learning, 1998.  
 [19] Weka, "http://www.cs.waikato.ac.nz/ml/weka".



**현 중 환**

2011 포항공과대학교, 컴퓨터공학과 학사  
 2011 ~ 현재 포항공과대학교, 컴퓨터공학과 통합 과정  
 <관심분야> 네트워크 트래픽 모니터링, 네트워크 보안



**Yongfeng Huang**

2010 Xi'an Jiaotong University, BSc. in Computer Science  
 2010 ~ 2012 POSTECH, M.S. in Division of IT Convergence Engineering (Autonomics)  
 2012 ~ 현재 National University of Singapore 박사과정



**Jin Xiao**

2001 Univ. of Calgary, BSc. in Computer Science  
 2001 ~ 2010 Univ. of Waterloo, Ph.D. in Computer Science  
 2010 ~ 현재 포항공과대학교 정보전자융합공학부 연구교수



**홍 원 기**

1983 Univ. of Western Ontario, BSc in Computer Science  
 1985 Univ. of Western Ontario, MS in Computer Science  
 1985 ~ 1986 Univ. of Western Ontario, Lecturer  
 1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science  
 1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow

1992 ~ 1995 Univ. of Western Ontario, 연구교수  
 1995 ~ 현재 포항공과대학교 컴퓨터공학과 교수  
 2007~2011 포항공과대학교 정보통신대학원장  
 2007~2010 포항공과대학교 정보통신연구소 연구소장  
 2008~2010 포항공과대학교 컴퓨터공학과 주임교수  
 2008~2012 포항공과대학교 정보전자융합공학부장  
 2008~현재 포항공과대학교 정보전자융합공학부 교수  
 2012~현재 kt Advanced Institute of Technology CTO