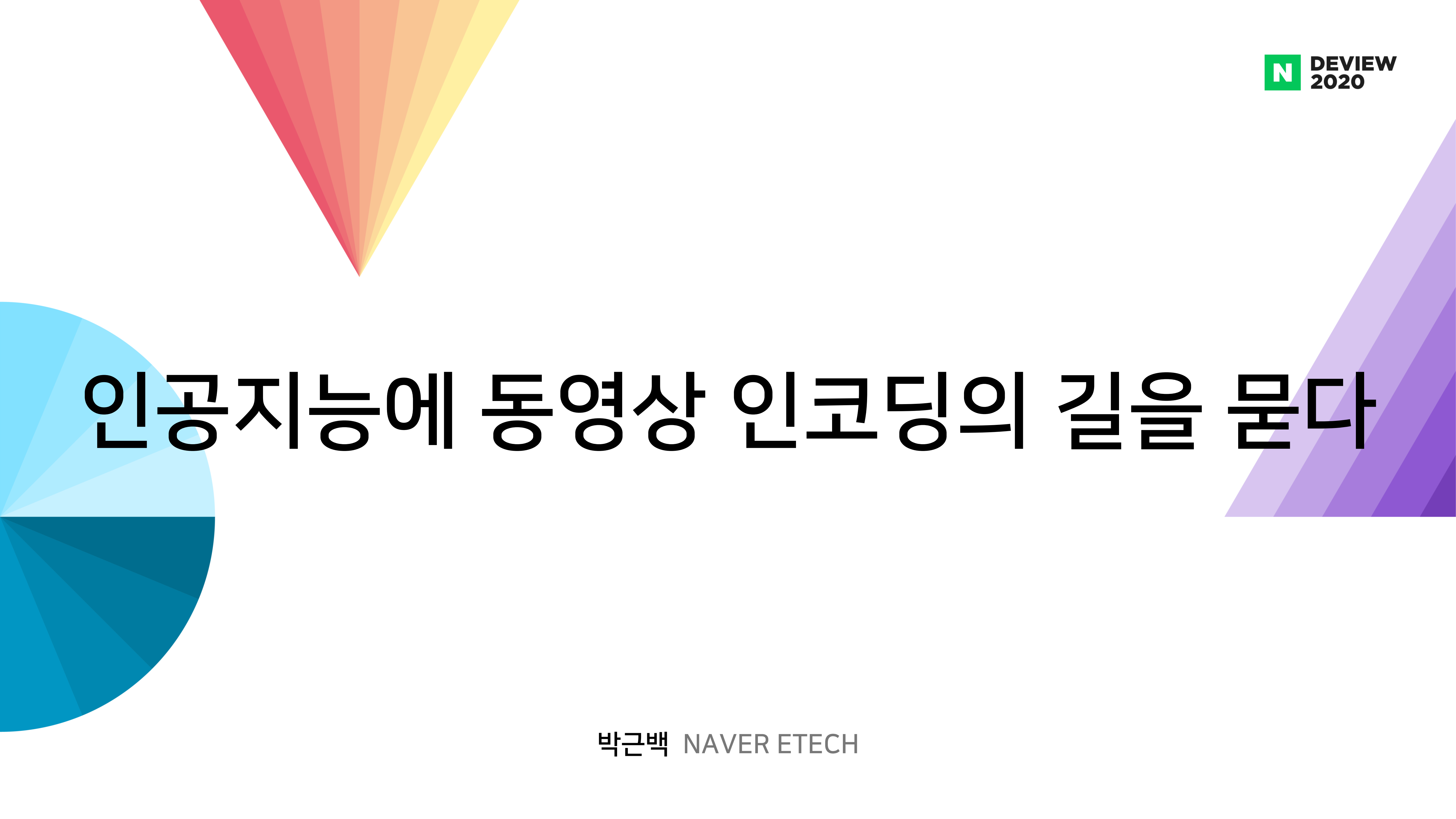


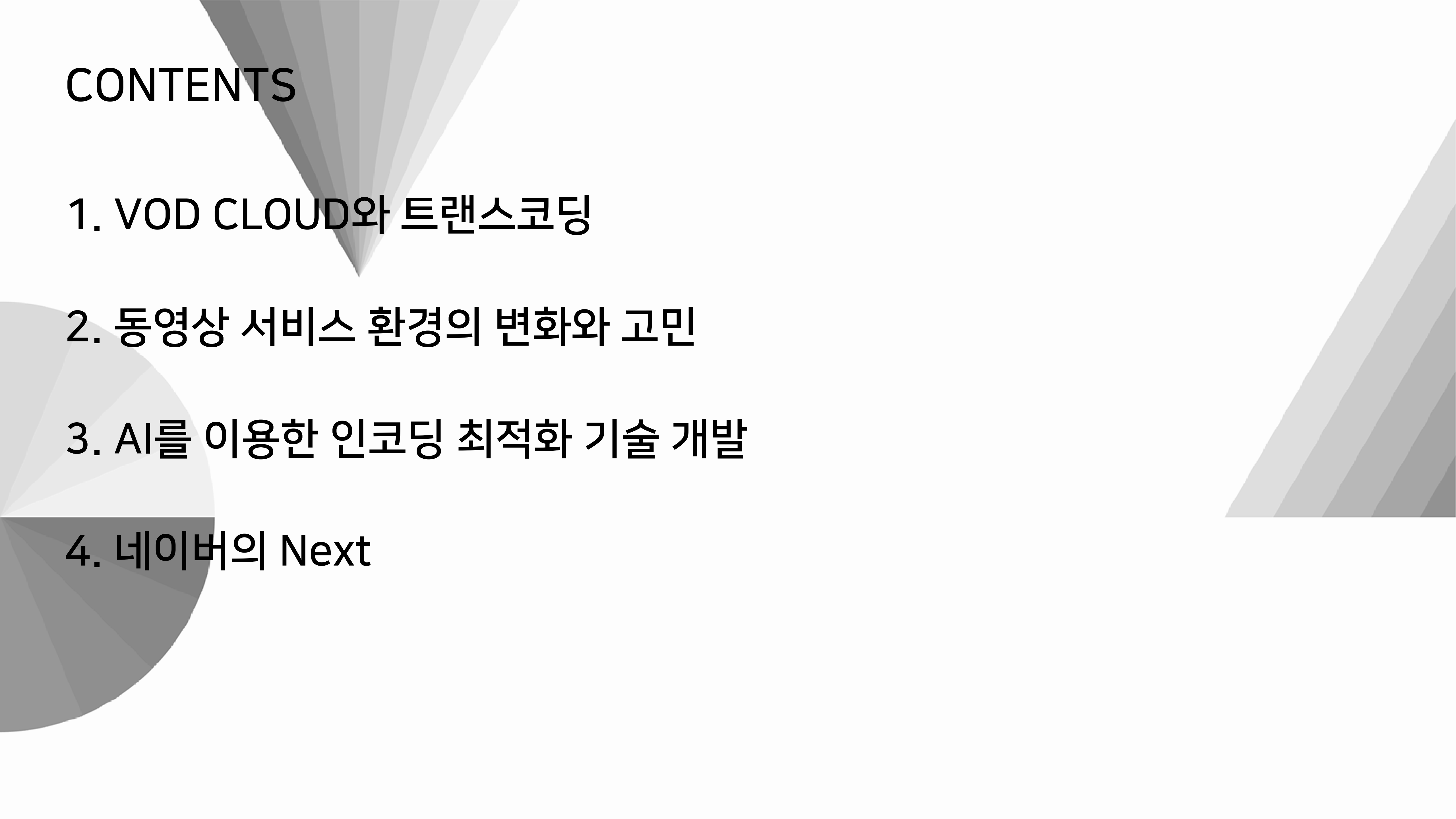


인공지능에 동영상 인코딩의 길을 묻다

NAVER ETECH. 는 미디어의 생산 <-> 클라우드 <-> 소비기술을 연구 개발합니다.

단단한 기술력을 바탕으로, 미디어의 Immersive Experience, AI., XR. 기술을
End-to-End로 제공하는 글로벌 미디어 플랫폼을 지향합니다.





CONTENTS

1. VOD CLOUD와 트랜스코딩

2. 동영상 서비스 환경의 변화와 고민

3. AI를 이용한 인코딩 최적화 기술 개발

4. 네이버의 Next

1. VOD CLOUD와 트랜스코딩

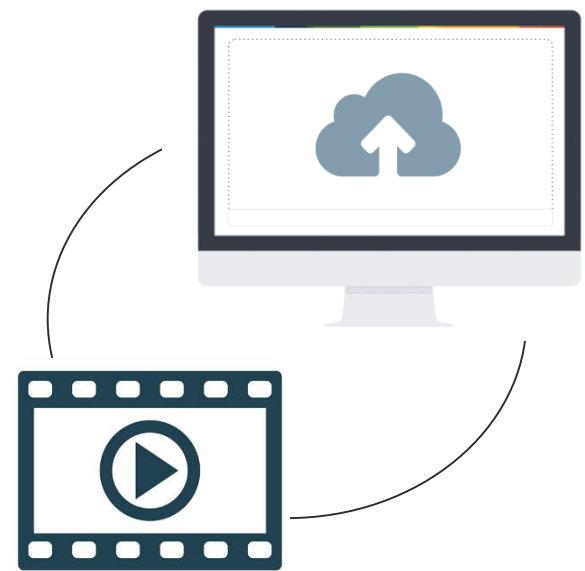
네이버 VOD CLOUD

네이버 전사 VOD 서비스 제공



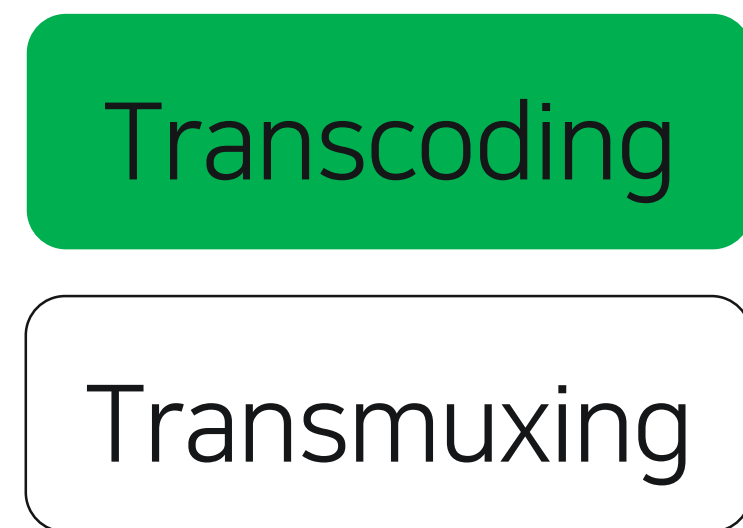
VOD 서비스 FLOW

INGEST



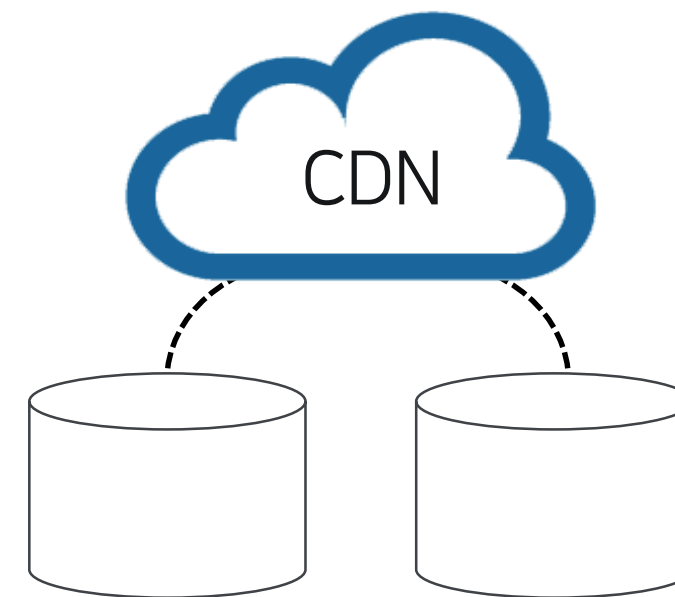
다양한 기기에서 캡처
동영상 업로드

CONVERT



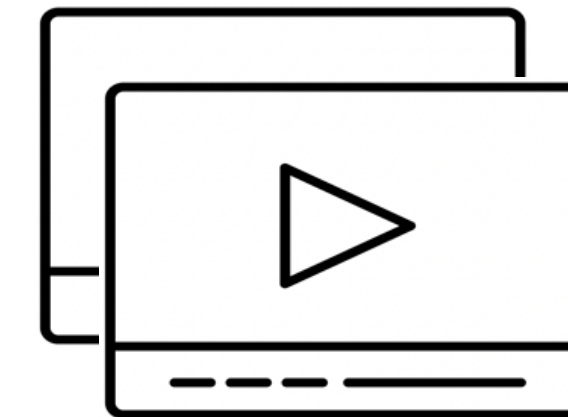
코덱, 컨테이너 변환
HLS, DASH

STORE &
DELIVER



Streaming, AD,
DRM License

PLAYBACK



다양한 환경에서
재생

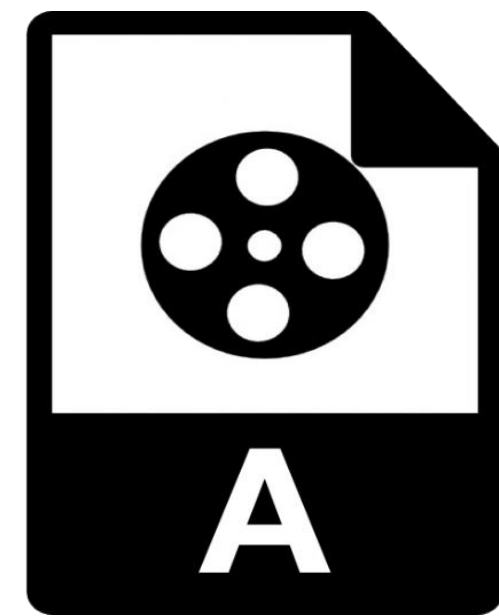
ANALYZE



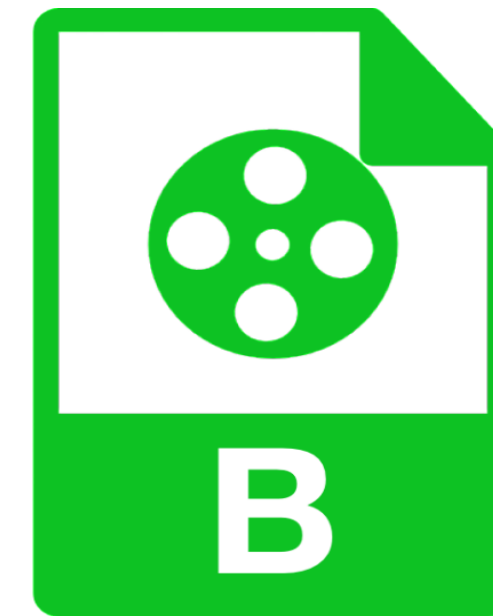
품질, 소비 분석

VOD 트랜스코딩

트랜스코딩은 콘텐츠를 다시 만드는 과정



MKV
MPEG-2
1080P
30Mbps
60fps
96kHz
5.1ch

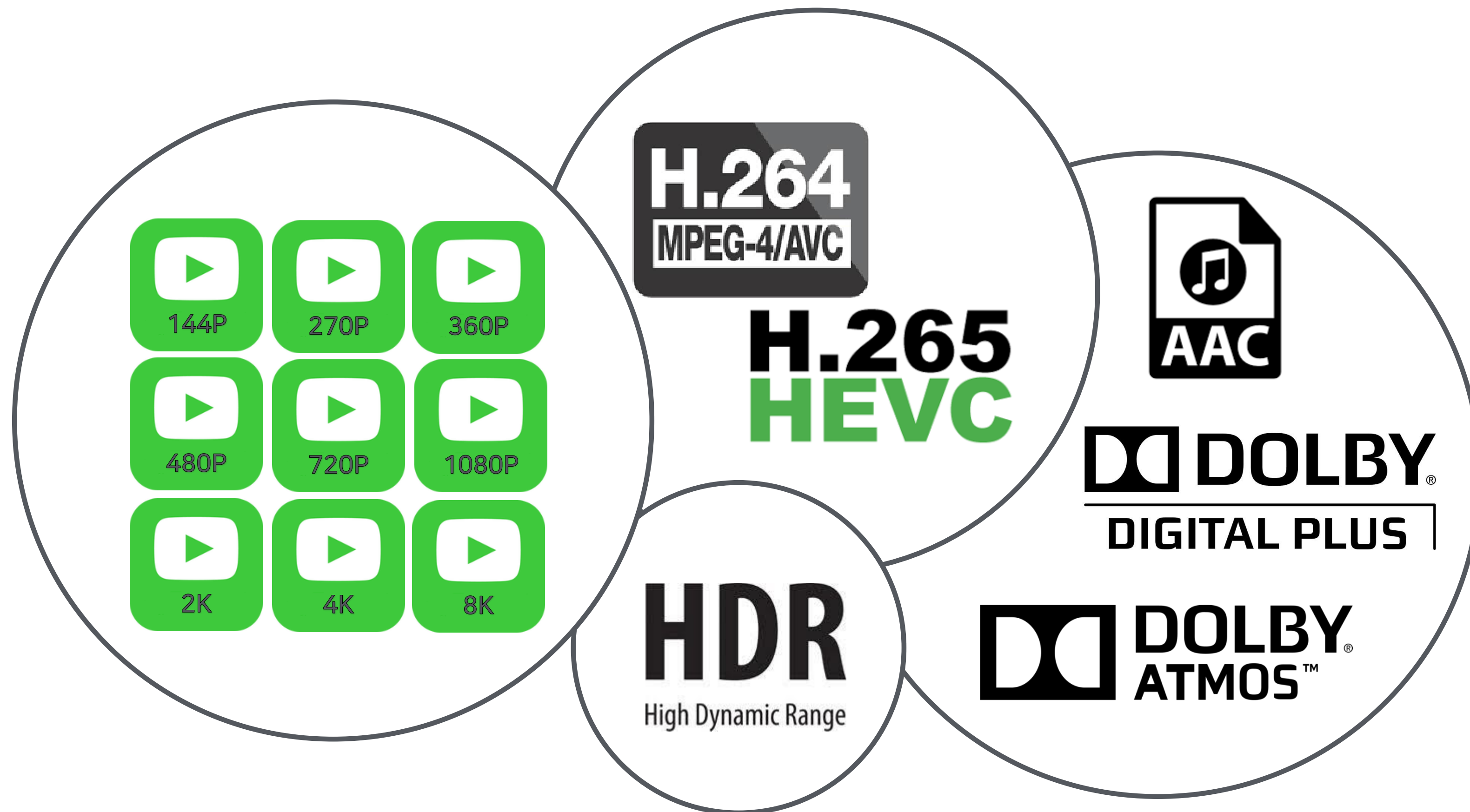


MP4
H.264
720P
5Mbps
30fps
44.1kHz
2ch

Container
CODEC
Resolution
Bit-rate
Frame-rate
Sampling-rate
Channels

다양한 환경의 사용자를 위한 준비

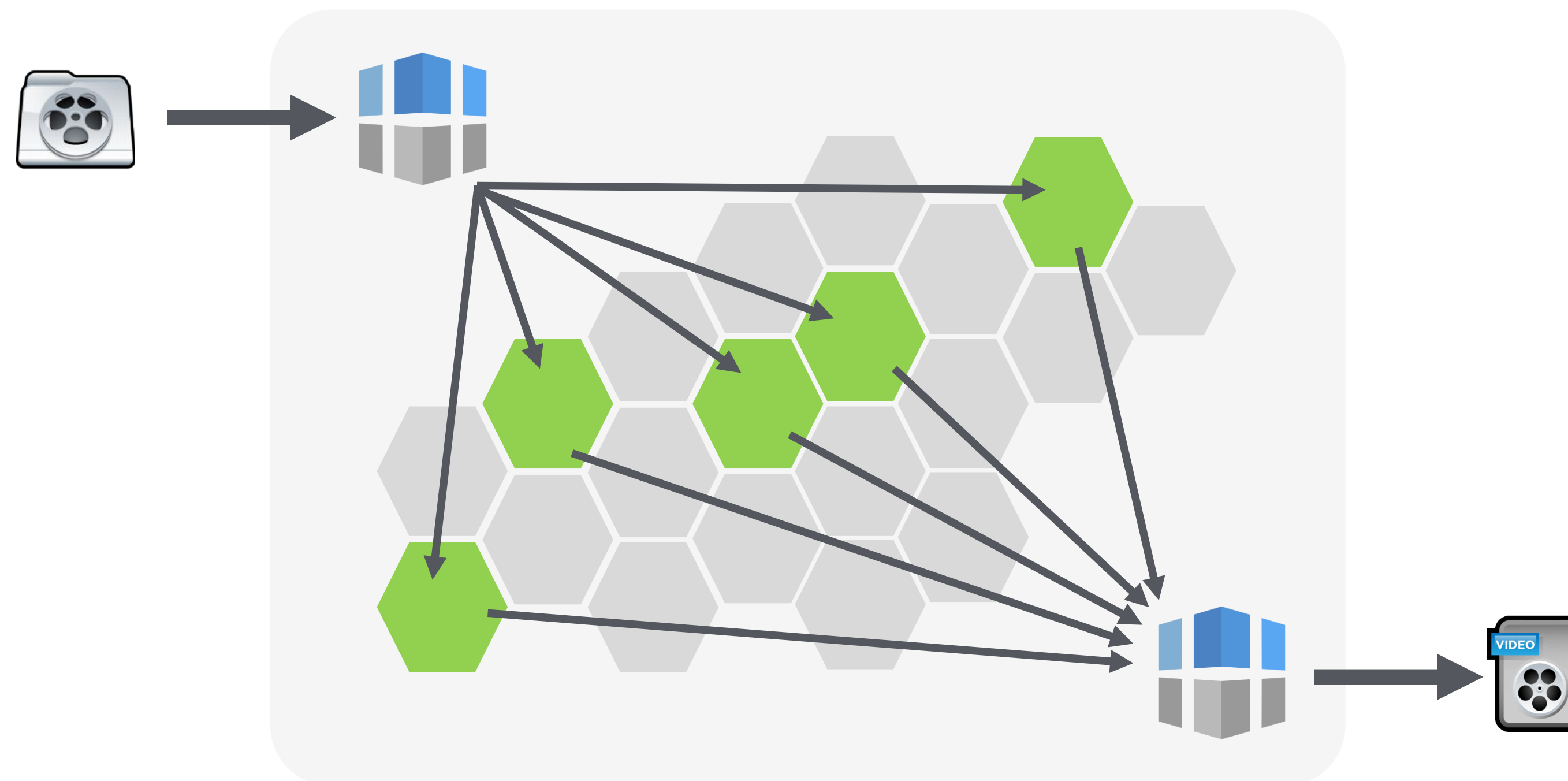
하나의 영상을 여러 벌의 파일로 준비



트랜스코딩 처리 시간의 개선

트랜스코딩 분산 처리를 통한 개선

- <https://d2.naver.com/helloworld/3661677>



2. 동영상 환경의 변화와 고민



촬영 디바이스의 발전

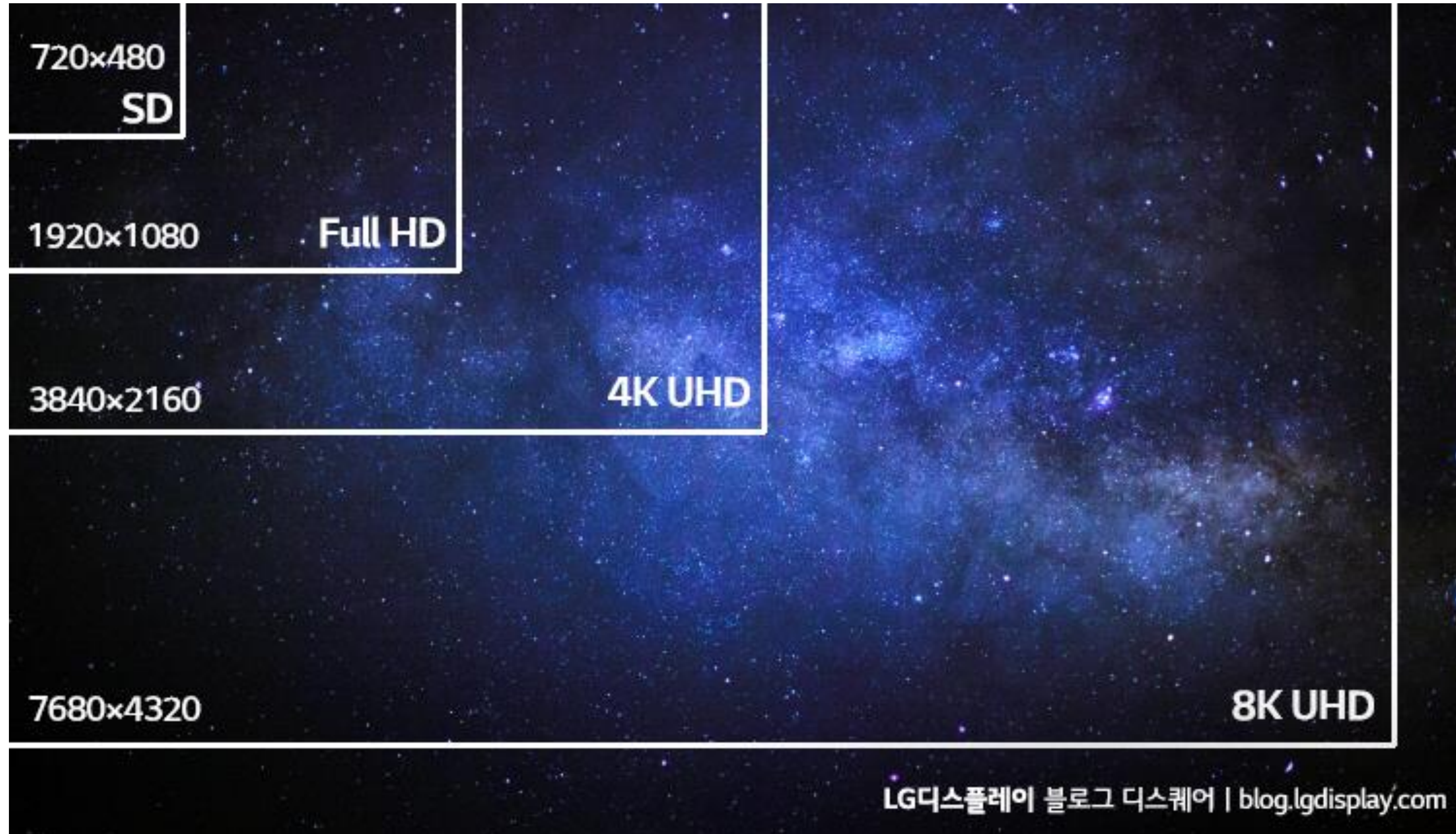
N DEVIEW
2020



Canon EOS R5



디스플레이의 발전



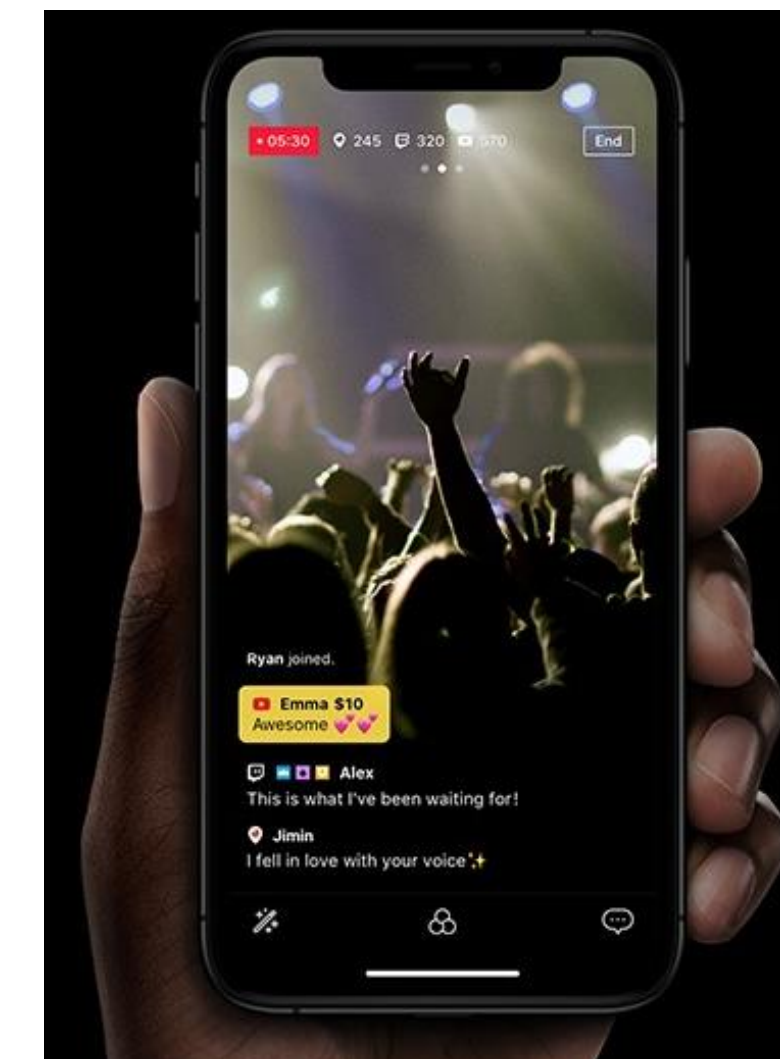
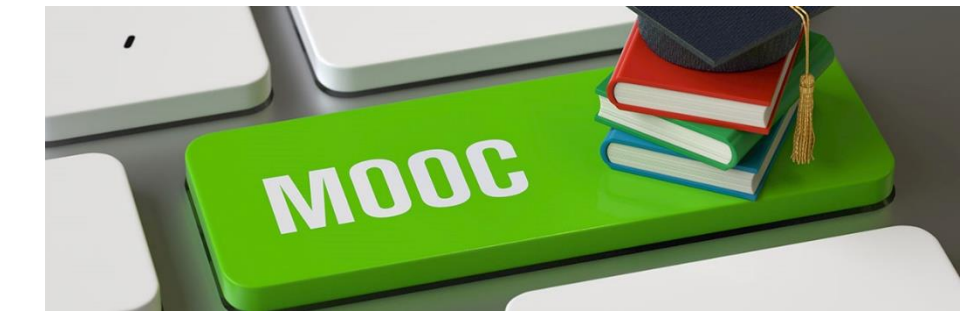
이제는 동영상의 시대로

커뮤니케이션 도구

동영상으로 쉽게 배운다

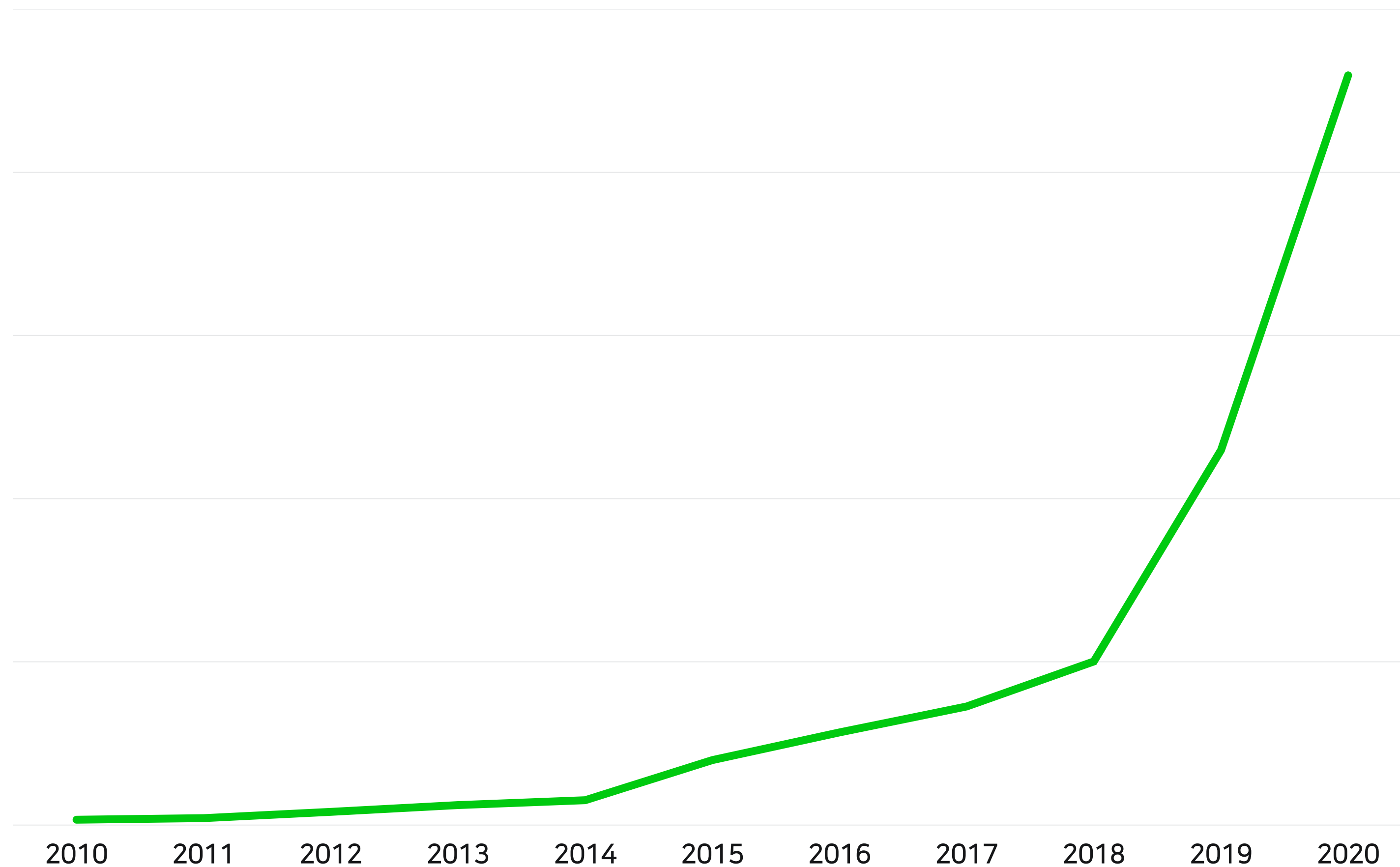
방송 주체의 변화

그리고.. Covid-19



VOD CLOUD 스토리지 사용량

네이버 VOD 스토리지 증가 추이



스토리지만 문제일까요?

고해상도, 고화질을 위한 데이터 증가는 사용자에게도 큰 부담

이런 상황을 개선할 수 있는 방안은..

어떤 영상의 화질이 더 좋나요?

N DEVIEW
2020



실제 비트레이트는



8Mbps



5.8Mbps



5Mbps



3Mbps

실제 비트레이트는



8Mbps **비트레이트 낭비**



5.8Mbps **품질과 비트레이트 균형**



5Mbps **비트레이트 낭비**



3Mbps **품질과 비트레이트 균형**

실제 비트레이트는

- 일정 화질 이상에서는 사람의 눈으로 화질의 구별이 어려움



8Mbps 비트레이트 낭비



5.8Mbps 품질과 비트레이트 균형

- 일정 화질 이상의 비트레이트는 낭비되게 됨



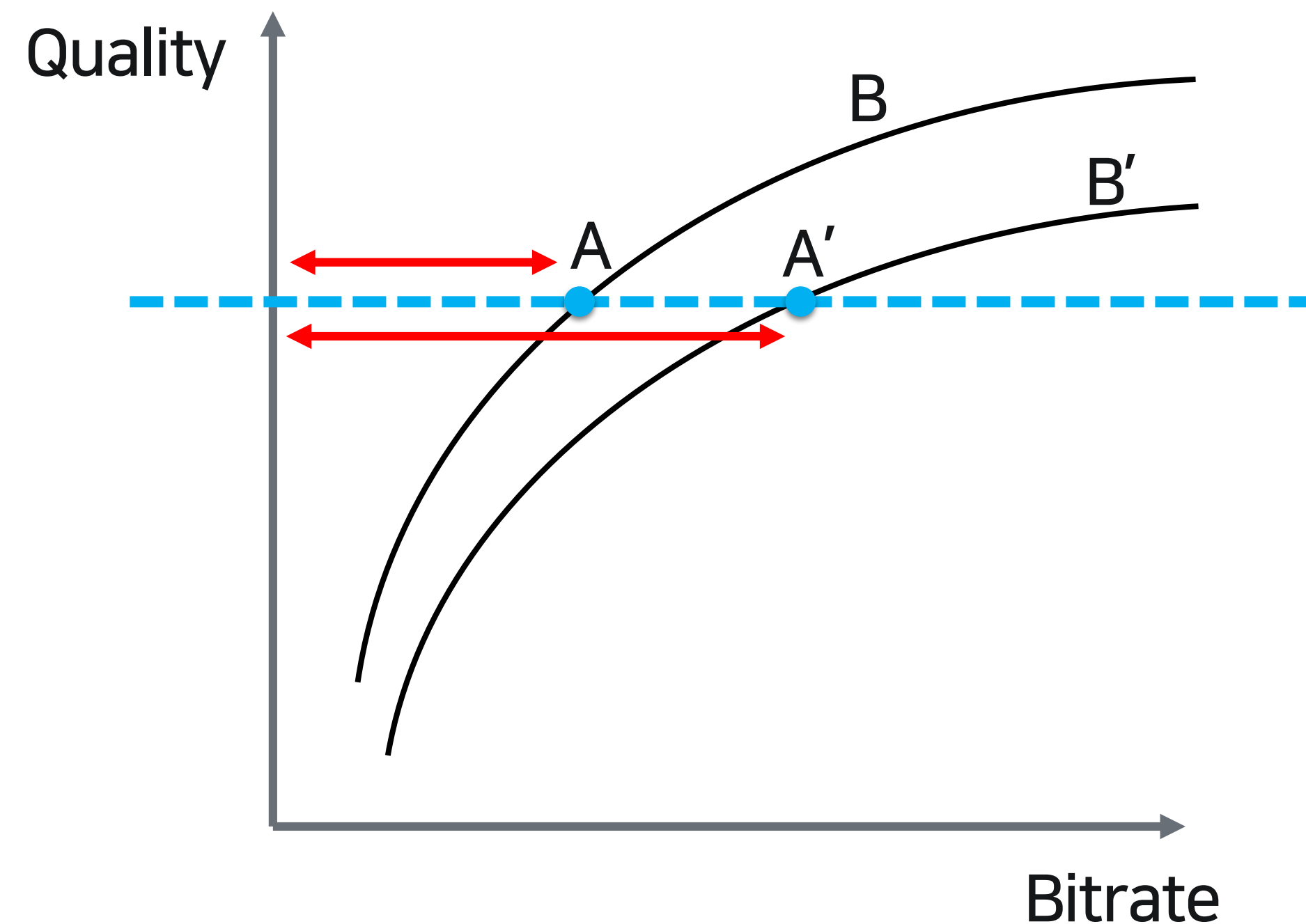
5Mbps 비트레이트 낭비



3Mbps 품질과 비트레이트 균형

비트레이트와 화질

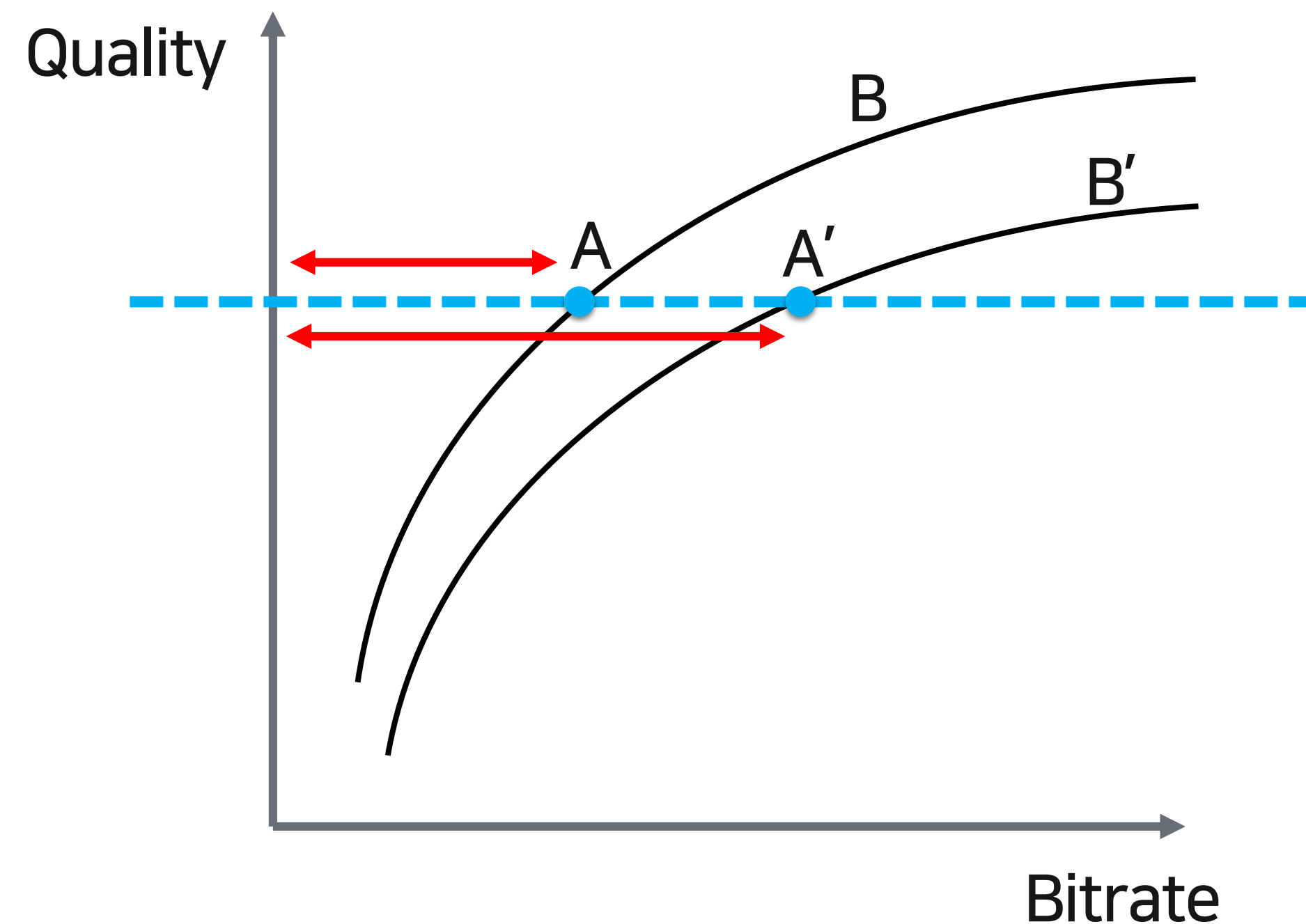
사람의 눈으로 화질 차이를 구별할 수 있는 비트레이트 범위는 영상에 따라 아주 다양함



비트레이트와 화질

사람의 눈으로 화질 차이를 구별할 수 있는 비트레이트 범위는 영상에 따라 아주 다양함

→ 일정 화질 이상으로 낭비되는 비트레이트를 줄이는데 필요



비트레이트 낭비되는 부분을 줄이면

- 1) 사용자 데이터 비용 절약
- 2) 사용자 경험 개선
- 3) 스토리지/CDN 비용 절감

우리가 추구하는 목표는

사용자가 부담하는 데이터 비용은 가능한 줄이면서
제공하는 영상의 품질을 높게 유지하는 것

→ 사용자의 좋은 경험을 유지하면서

→ 비트레이트 개선을 위한 인코딩 최적화 연구

3. AI를 이용한 인코딩 최적화 기술 개발

인코딩 최적화 기술 리서치

Per-Title Encoding

- Title별 하나의 Optimal Bitrate Ladder 생성
- **영상의 평균적인 특성(Complexity)**을 반영

< Bitrate Ladder >

Resolution	Static Bitrate
1080p	4000
720p	2000
480p	1000
270p	500

단위 : kbps

인코딩 최적화 기술 리서치

Per-Title Encoding

- Title별 하나의 Optimal Bitrate Ladder 생성
- **영상의 평균적인 특성(Complexity)**을 반영

< Bitrate Ladder >

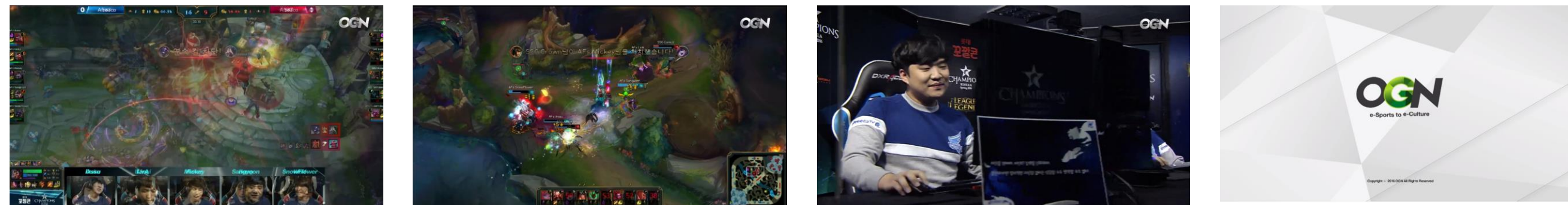
Resolution	Static Bitrate	PerTitle Bitrate	
		Animation	Action
1080p	4000	2200	4500
720p	2000	1105	2250
480p	1000	605	1105
270p	500	350	580

단위 : kbps

인코딩 최적화 기술 리서치

Per-Title Encoding

- Title별 하나의 Optimal Bitrate Ladder 생성
- **영상의 평균적인 특성(Complexity)**을 반영



Time(sec)

인코딩 최적화 기술 리서치

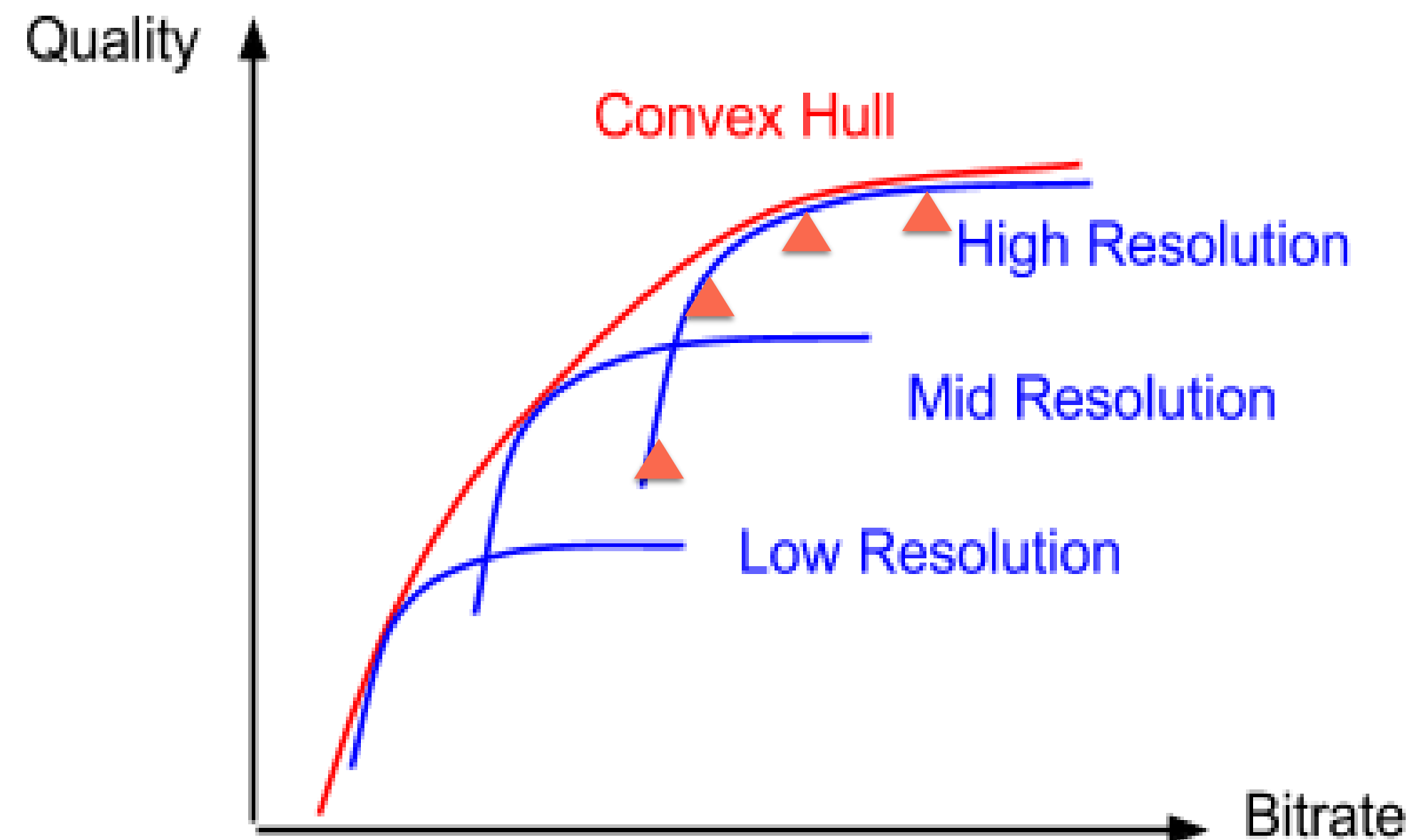
Dynamic Optimizer

- 영상의 구간별 Optimal Bitrate Ladder 생성
- 영상을 구간별로 나누고, 각 구간의 Optimal Bitrate Ladder에 따라 인코딩하고 다시 하나로 합치는 과정이 필요

인코딩 최적화 기술 리서치

Dynamic Optimizer

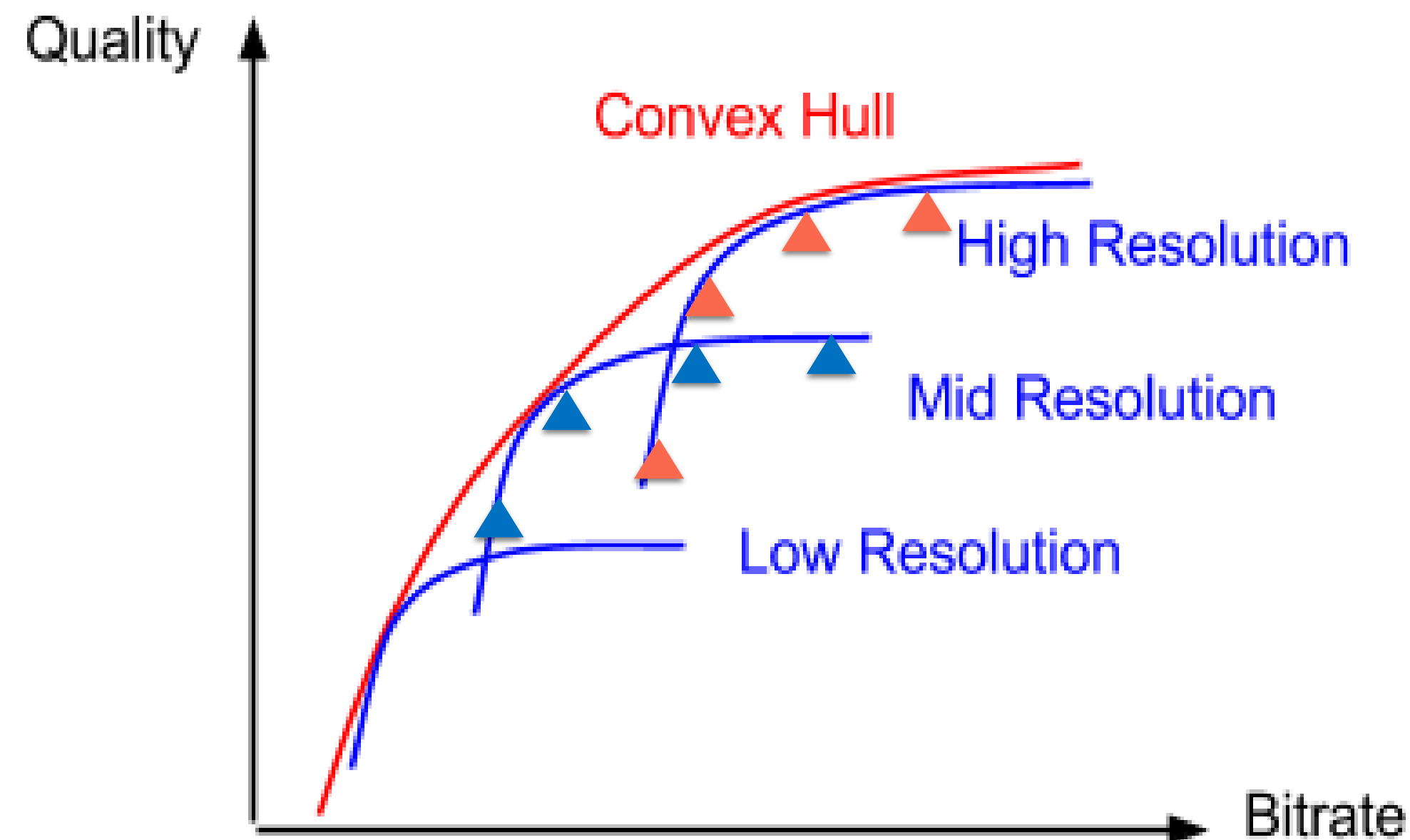
- 영상의 구간별 Optimal Bitrate Ladder 생성
- Optimal Bitrate Ladder 생성 방법
 - Convex Hull을 이용



인코딩 최적화 기술 리서치

Dynamic Optimizer

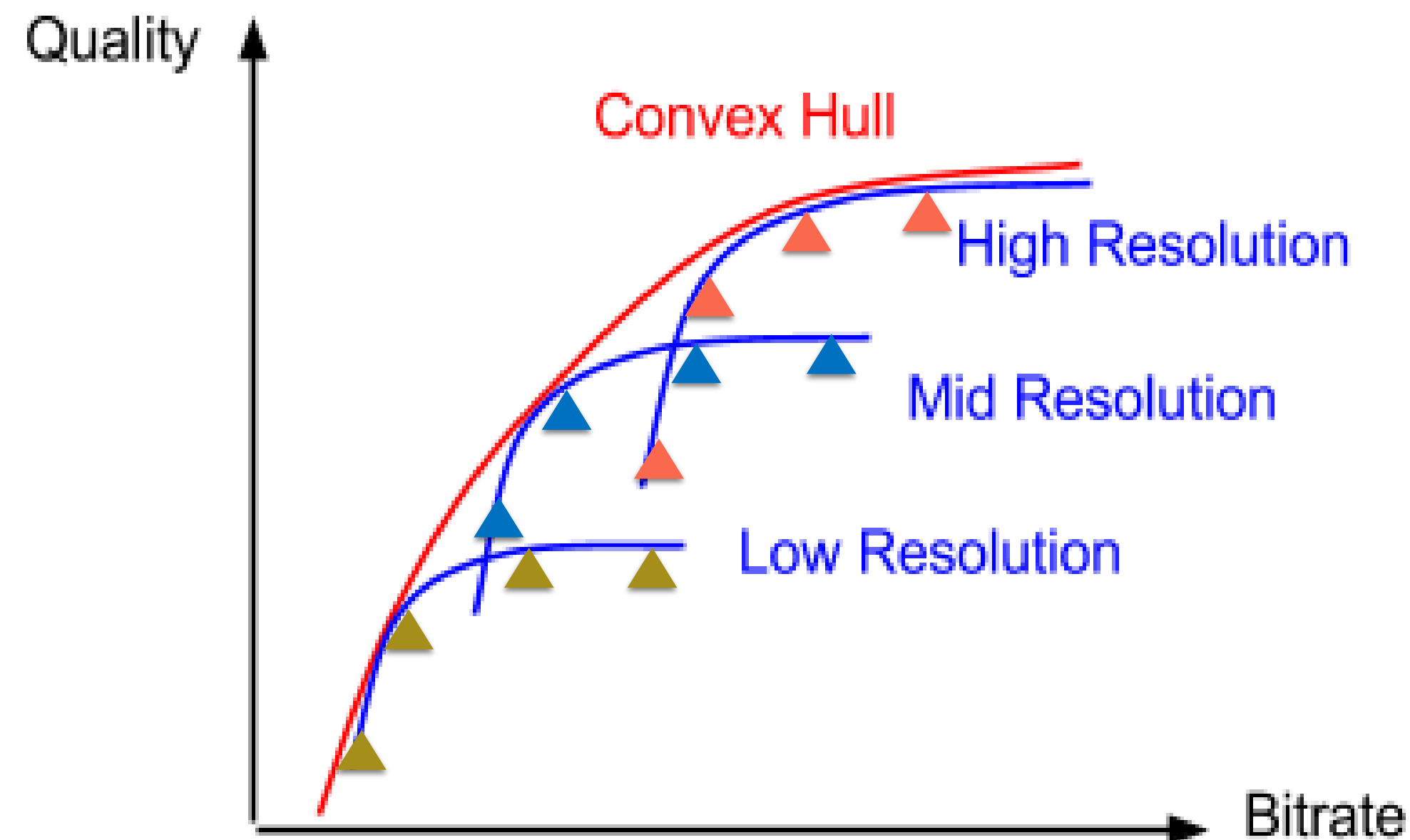
- 영상의 구간별 Optimal Bitrate Ladder 생성
- Optimal Bitrate Ladder 생성 방법
 - Convex Hull을 이용



인코딩 최적화 기술 리서치

Dynamic Optimizer

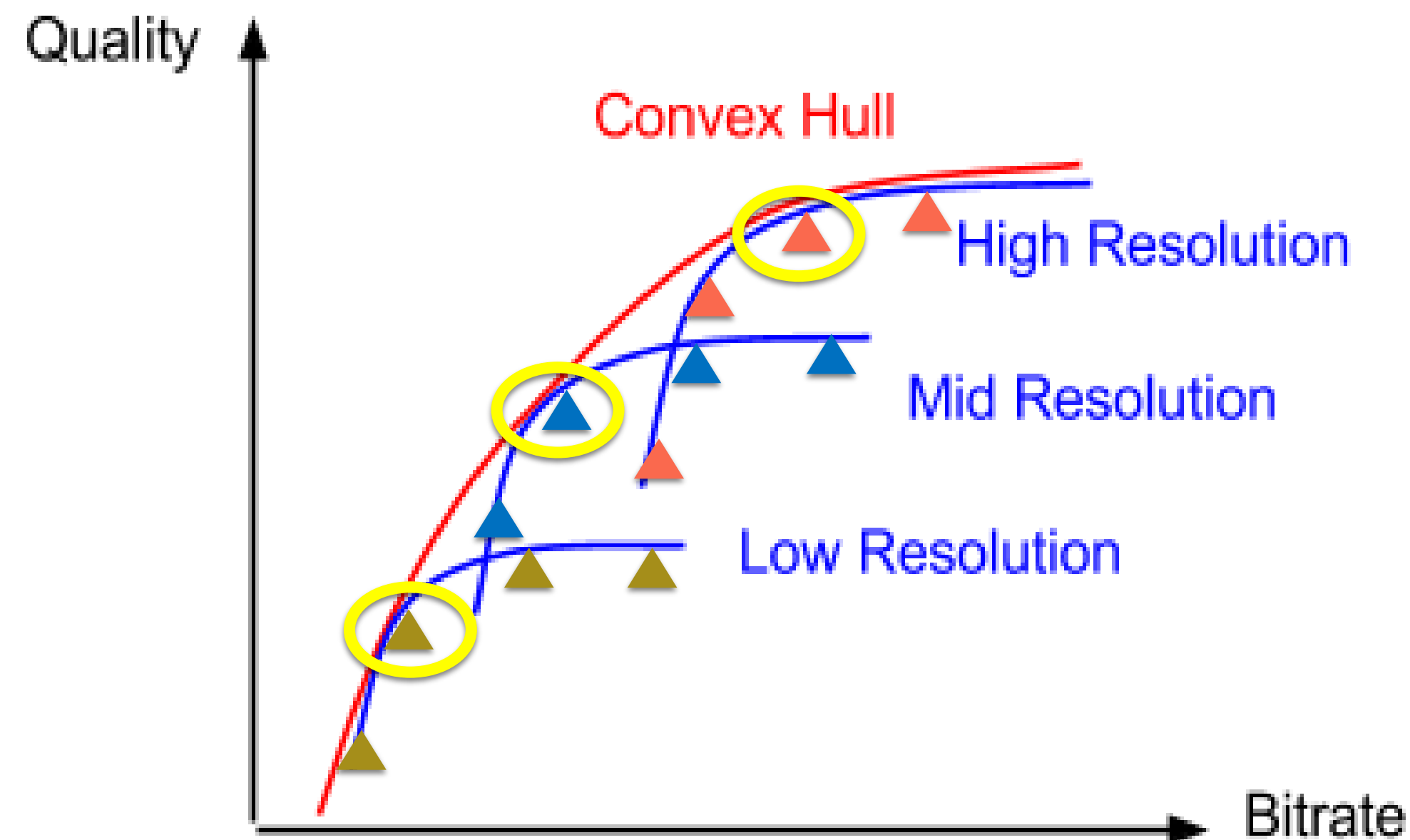
- 영상의 구간별 Optimal Bitrate Ladder 생성
- Optimal Bitrate Ladder 생성 방법
 - Convex Hull을 이용



인코딩 최적화 기술 리서치

Dynamic Optimizer

- 영상의 구간별 Optimal Bitrate Ladder 생성
- Optimal Bitrate Ladder 생성 방법
 - Convex Hull을 이용



인코딩 최적화 기술 리서치

Dynamic Optimizer

- 영상의 구간별 Optimal Bitrate Ladder 생성
 - Optimal Bitrate Ladder 생성 방법
 - Convex Hull을 이용
 - 수십배 이상의 인코딩 비용이 발생함
- => 구간별 특성을 찾기 위한 연구

인코딩 최적화 기술 리서치

- 영상/구간별 **적정 비트레이트**를 찾기 위한 연구
- 최적화의 범위가 전체 영상 → **구간별** 최적화로 세분화
- 영상/구간의 **특성**을 찾고, 그에 맞는 **인코딩 옵션**(비트레이트)를 찾는 것이 핵심

→ 영상 구간별 특성에 맞는 최적화 옵션 도출 연구

영상의 특성에 맞는 최적화 옵션을
도출하기 위해서 필요한 것은?

영상의 특성에 맞는 최적화 옵션 도출

- 1) **영상의 특성**을 어떻게 판단할 것인가
- 2) **최적화 옵션**은 무엇으로 할 것인가
- 3) 영상의 특성에 맞는 최적화 옵션을 **어떻게 도출**해낼 것인가

1) 영상의 특성을 어떻게 판단할 것인가

영상의 특성

- 동영상 : 여러 장면이 모여서 연속적으로 보여지는 결과물
- 영상의 특성 = Spatial Feature + Temporal Feature

1) 영상의 특성을 어떻게 판단할 것인가

영상의 특성

- 동영상 : 여러 장면이 모여서 연속적으로 보여지는 결과물

Spatial
Feature



Temporal
Feature



1) 영상의 특성을 어떻게 판단할 것인가

영상의 특성을 대표하는 Feature

- Spatial Feature
 - Block Partition, Coefficient
 - Quantization Parameter
- Temporal Feature
 - Motion Vector
 - Reference Frame Index

코덱에 의존적인 정보..



1) 영상의 특성을 어떻게 판단할 것인가

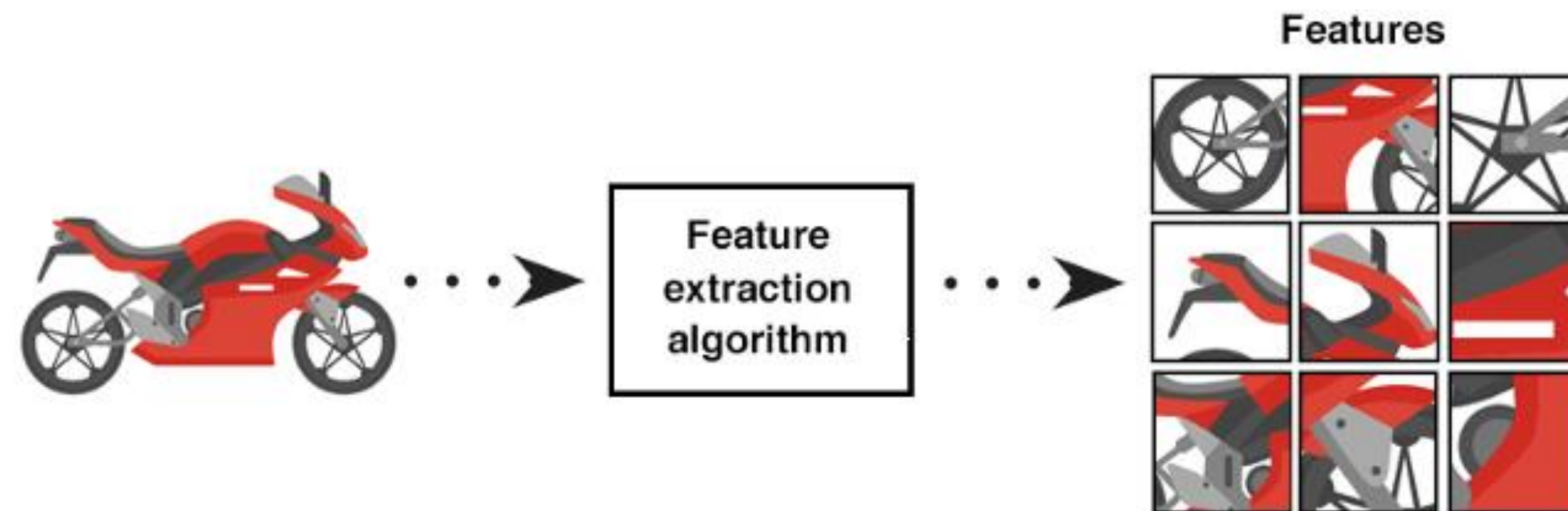
다양한 원본 영상에 사용 가능한 피쳐

- 디코딩 후 픽셀 도메인에서 피쳐를 뽑으면 원본 코덱과 무관
- 정확히 어떤 피쳐가 영상의 특성을 잘 대표하는지는 명확하지 않음

1) 영상의 특성을 어떻게 판단할 것인가

다양한 원본 영상에 사용 가능한 피쳐

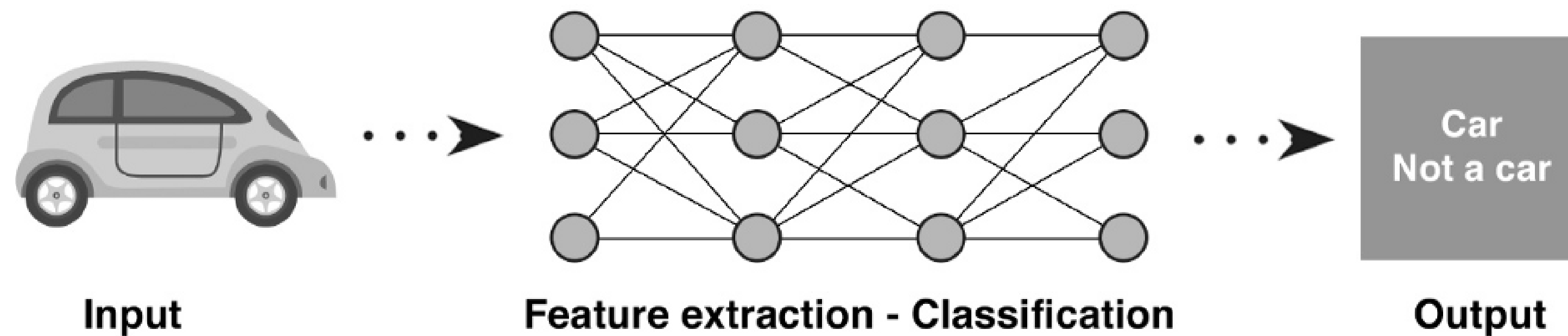
- 디코딩 후 픽셀 도메인에서 피쳐를 뽑으면 원본 코덱과 무관
- 정확히 어떤 피쳐가 영상의 특성을 잘 대표하는지는 명확하지 않음



1) 영상의 특성을 어떻게 판단할 것인가

다양한 원본 영상에 사용 가능한 피쳐

- 딥러닝은 유용한 피쳐를 추출하는 방법을 데이터로부터 학습



2) 최적화 옵션은 무엇으로 할 것인가

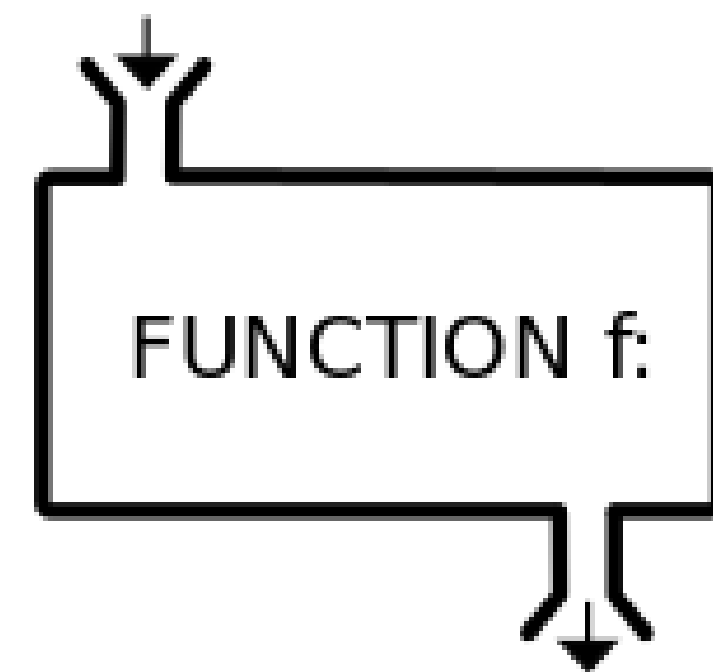
인코딩 Rate Control 옵션

CBR/VBR (Constant/Variable Bitrate)	CQP(Constant Quantization Parameter)	CRF (Constant Rate Factor)
- 화질이 아닌 특정 비트레이트로 인코딩 - 용량 예측가능 - 인코딩 결과 화질은 알 수 없음 - 화질이 고르지 않음(구간에 따라 화질 차이가 있음)	- 영상 전체적으로 균일한 화질 - 어느정도 화질이 될지는 알 수 없음 - 영상의 복잡도에 따라 인코딩 용량이 달라짐 - Value : 0~51	- CQP 대비 시각적으로 좀더 균일한 화질 - 사람의 지각적인 인식 특성 반영 - 용량 및 퀄리티에 효율적 - Value : 0~51

3) 영상의 특성에 맞는 최적화 옵션 도출 방법

Video Feature

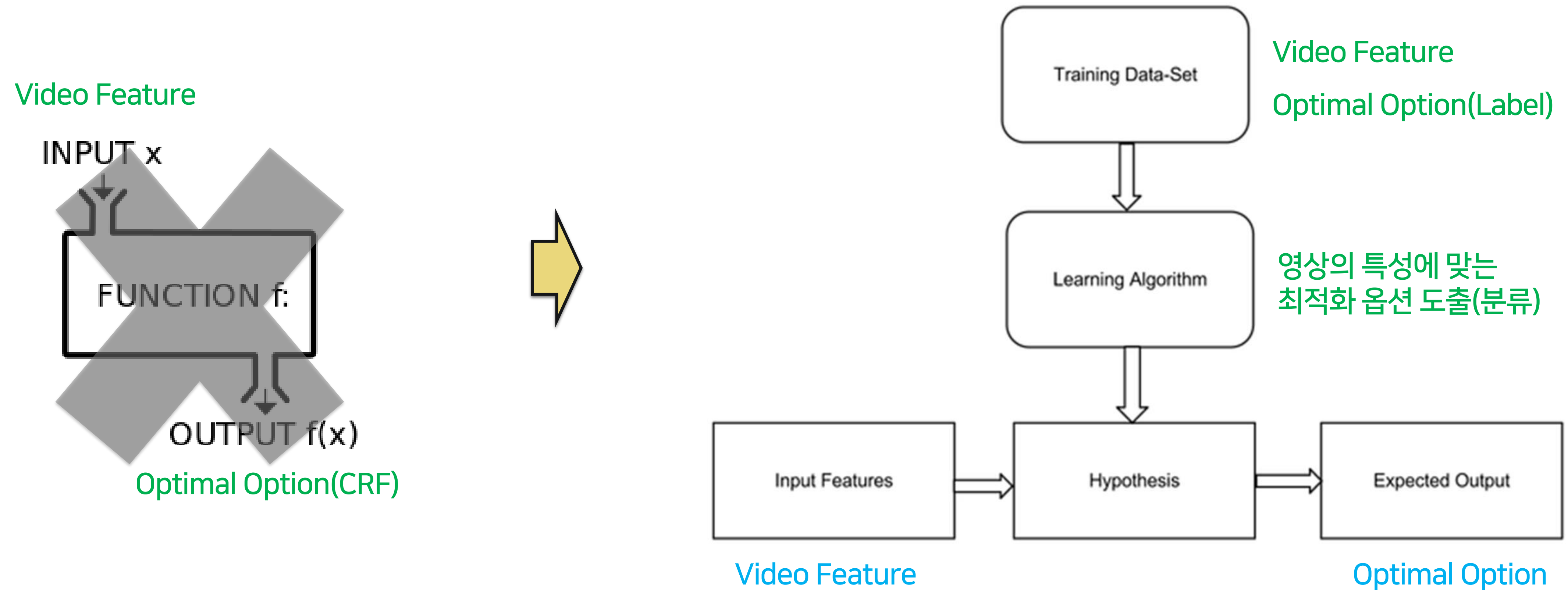
INPUT x



OUTPUT $f(x)$

Optimal Option(CRF)

3) 영상의 특성에 맞는 최적화 옵션 도출 방법



영상의 특성에 맞는 최적화 옵션 도출

1) 영상의 특성을 어떻게 판단할 것인가

=> 딥러닝으로 영상의 피쳐 추출

2) 최적화 옵션은 무엇으로 할 것인가

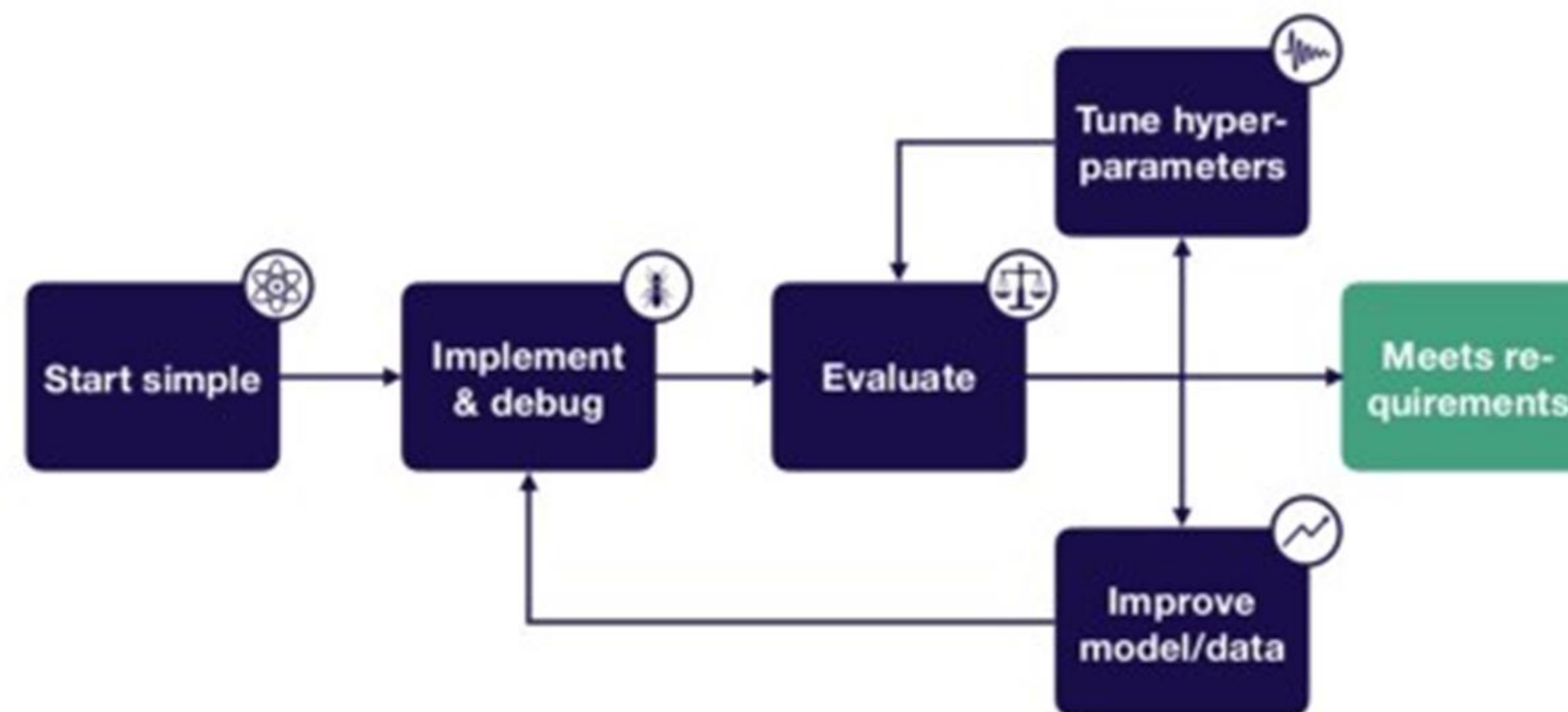
=> CRF(Constant Rate Factor)

3) 영상의 특성에 맞는 최적화 옵션을 어떻게 도출해낼 것인가

=> 지도학습으로 영상의 특성에 맞는 최적화 옵션 분류 모델

딥러닝 개발 검토

왜 잘되는지 왜 안되는지 명확하게 설명하기 어려운 부분이 존재
우선은 간단하게 시작하고(Start Simple), 점차 복잡도를 높여 나가자
가볍게 구현하고 검증된 라이브러리 사용하고, 먼저 동작하는 모델 만들자



프로토타이핑

심플한 프로토타입 통해서 전체 컨셉, 플로우를 확인해보자

1) 모델의 목표

- 입력받은 영상의 Complexity 특성을 상/중/하 로 분류

2) 개발 코드

- Youtube 8M Starter Code 참고

3) Dataset

- 샷 세그먼트 단위 10초 영상 1만개
- 라벨 : CRF 인코딩 용량 기준으로 Complexity 분류

프로토타이핑

- 연구 내용 적용 => 확인 => 개선의 주기를 빠르게 가져가기 위해 최소한의 파이프라인을 구축 (2주 단위 Iteration으로 반복)
- 심플하게 복잡도를 구분하는 모델에서의 가능성 파악
- 데이터셋 추가 확보 및 다양한 모델로 확대 적용

AI 인코딩 모델 개발

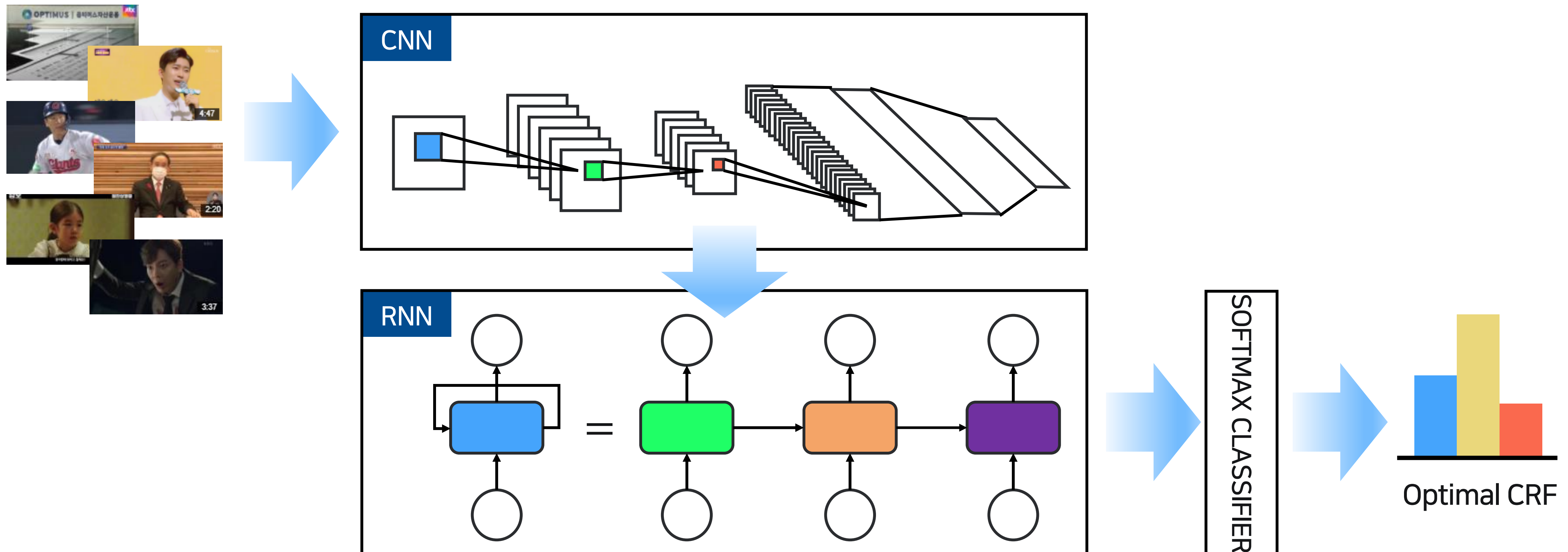
모델의 목표

영상의 특성을 판단하고 그에 맞는 최적의 인코딩 옵션을 예측
실서비스 적용 가능 수준 확보

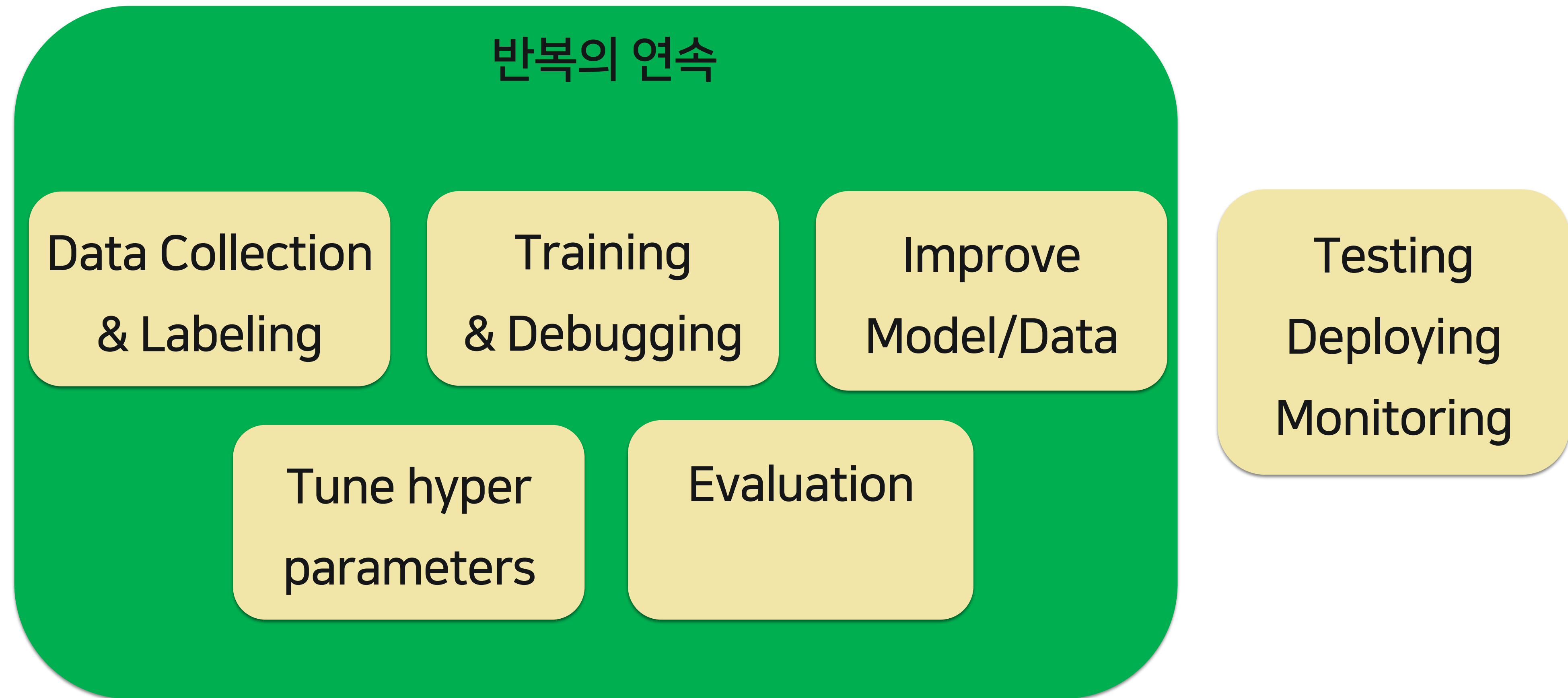
- ✓ 비트레이트 20~30%이상 개선
- ✓ Throughput을 극대화 할 수 있도록 인퍼런스 수행시간 최적화

모델의 구체화

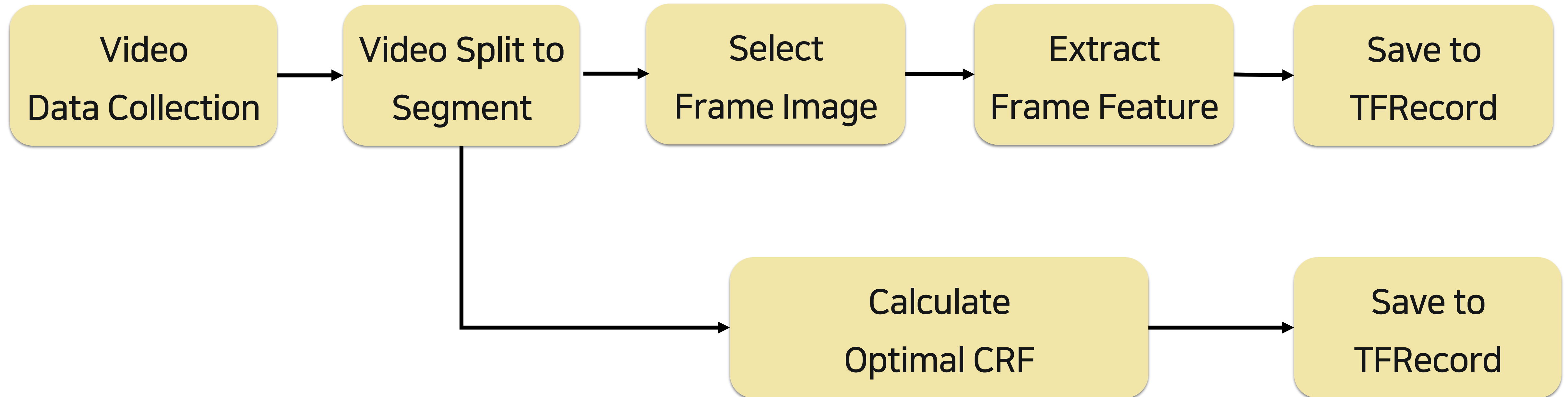
딥러닝(CNN+RNN)으로 영상의 피쳐 추출
영상의 피쳐별로 Optimal CRF 클래스로 분류하도록 지도 학습



모델 개발 과정

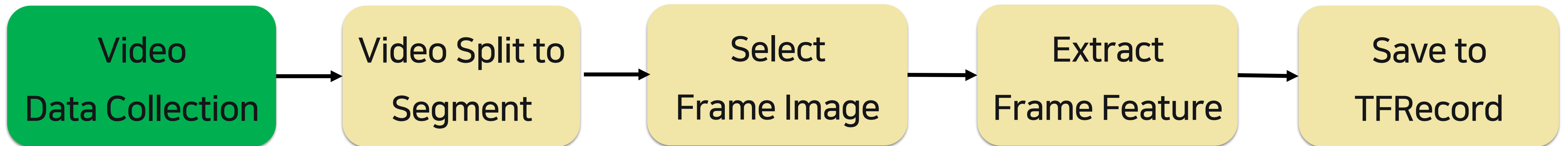


데이터 처리 과정



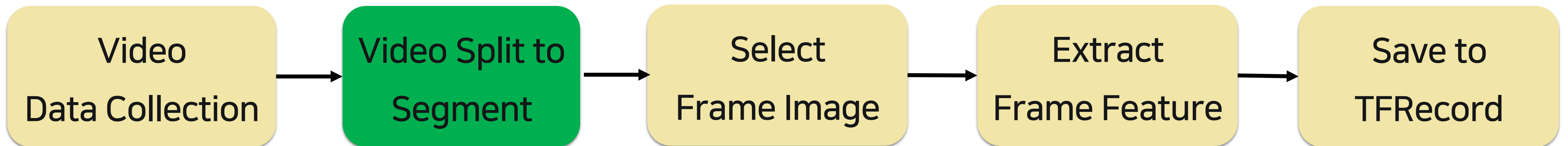
데이터 처리 과정

네이버에서 사용 가능한 특정 서비스 대상
다양한 특성의 영상을 포함
최신 데이터로 구성



데이터 처리 과정

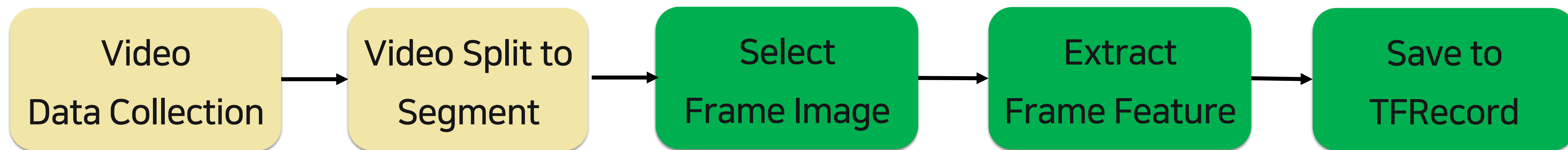
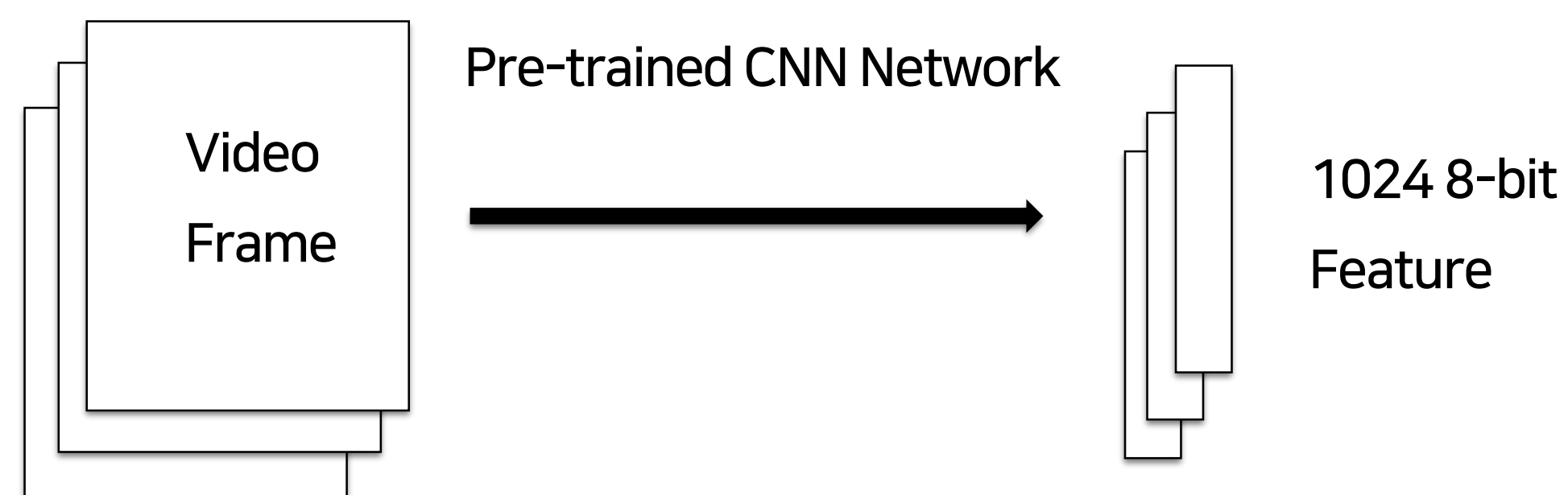
각 영상을 세그먼트 단위로 나누고 세그먼트를 인코딩하여 저장
영상별 세그먼트 개수를 동일하게
세그먼트 단위로 100만 데이터셋 확보



데이터 처리 과정

Frame-Level Feature (Feature Embedding)

- 미리 학습된 CNN 모델을 이용해서 1024 feature/frame

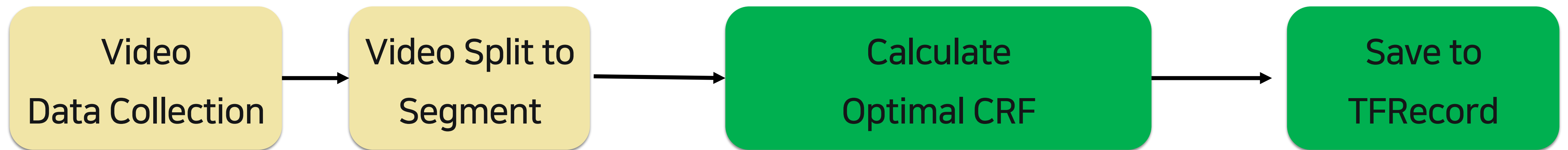
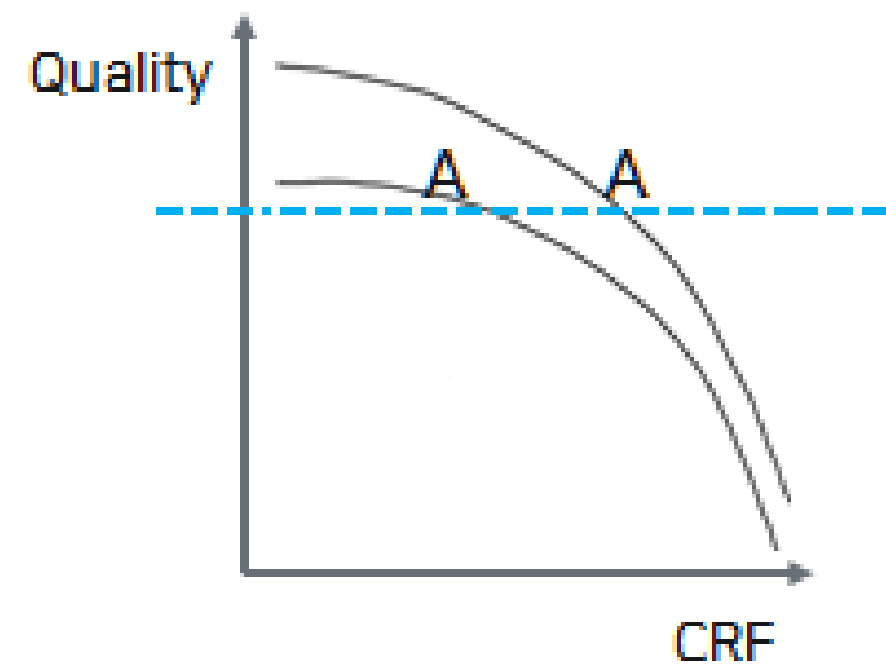


Transfer learning(전이학습) : 다른 데이터셋에 훈련된 모델의 가중치를 가지고 와서, 해결하고자 하는 문제에 맞게 재사용 함

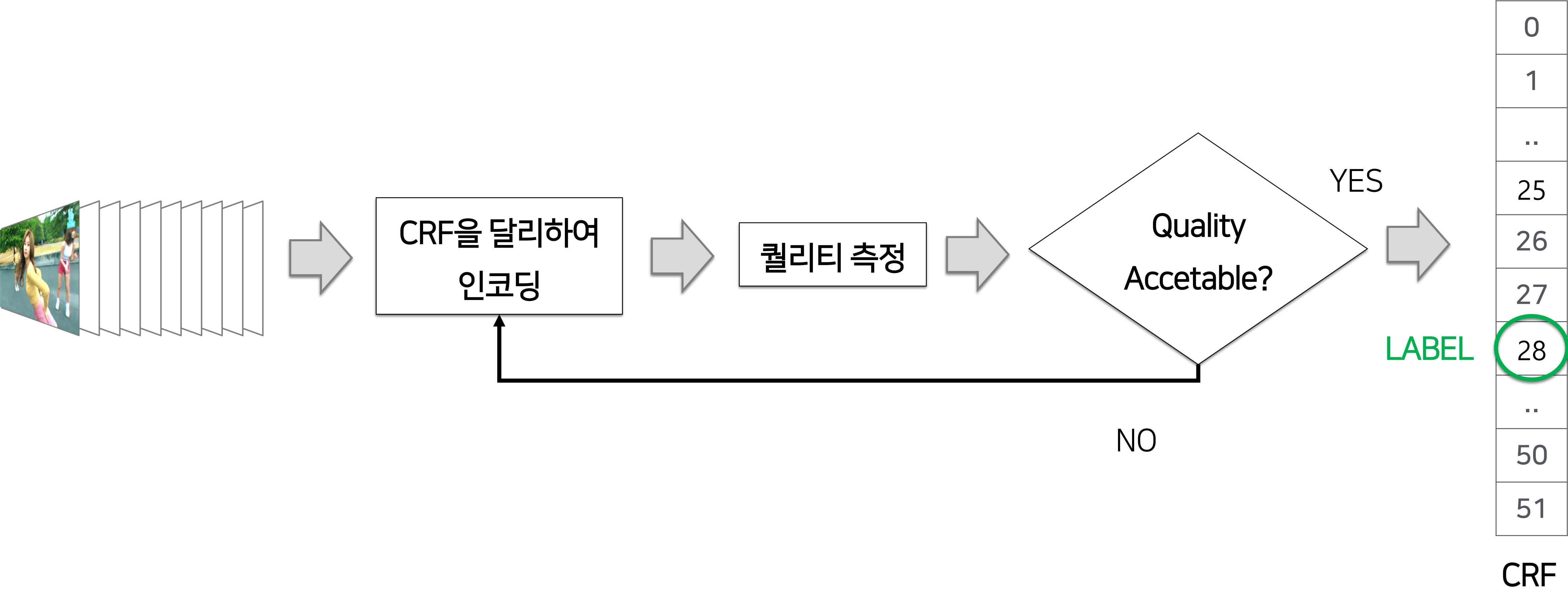
라벨 데이터 생성

라벨 : 각 세그먼트의 Optimal CRF 값(0~51)

- 각 세그먼트에 대해 일정한 화질을 만족하는 CRF 값을 찾음



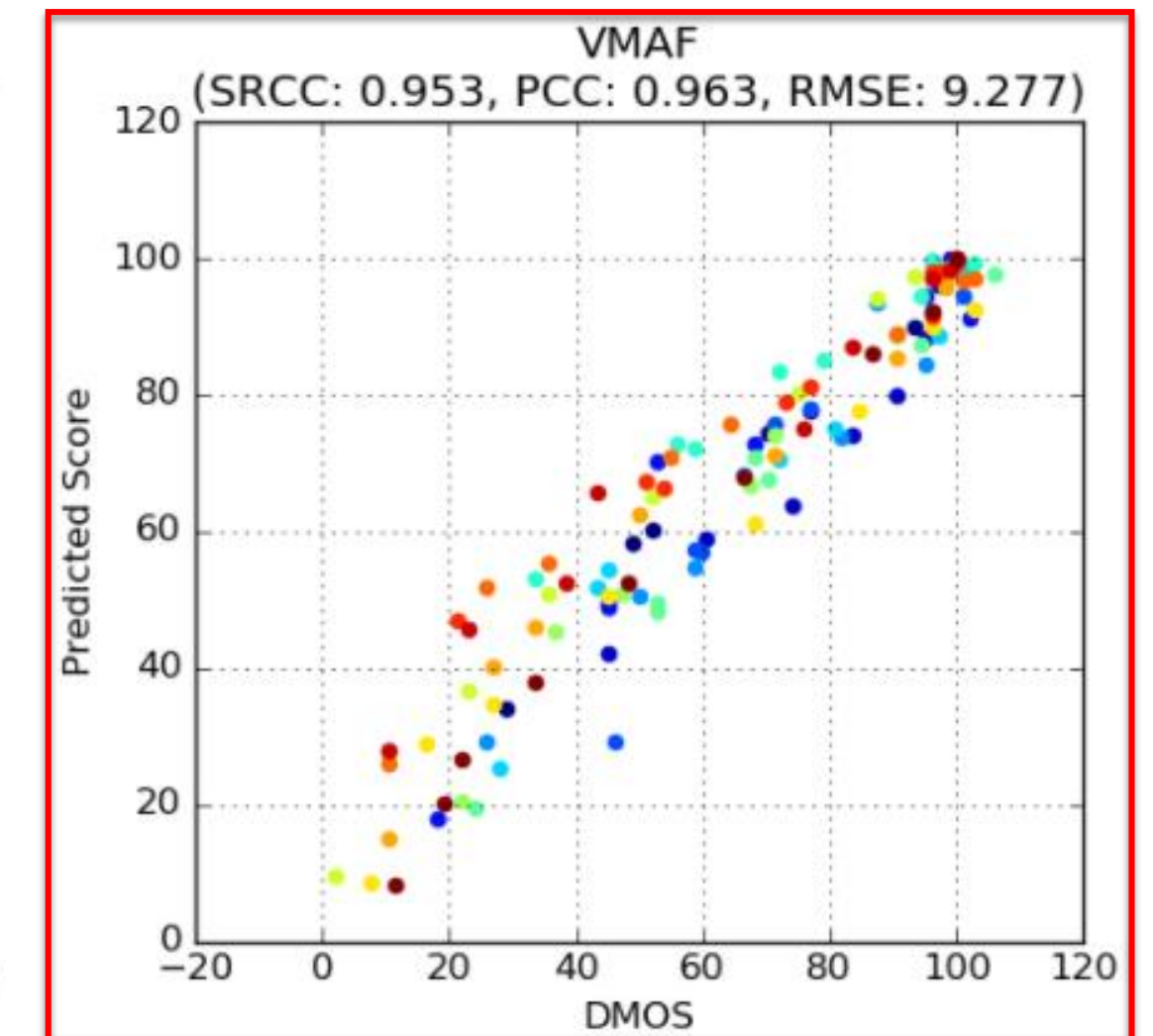
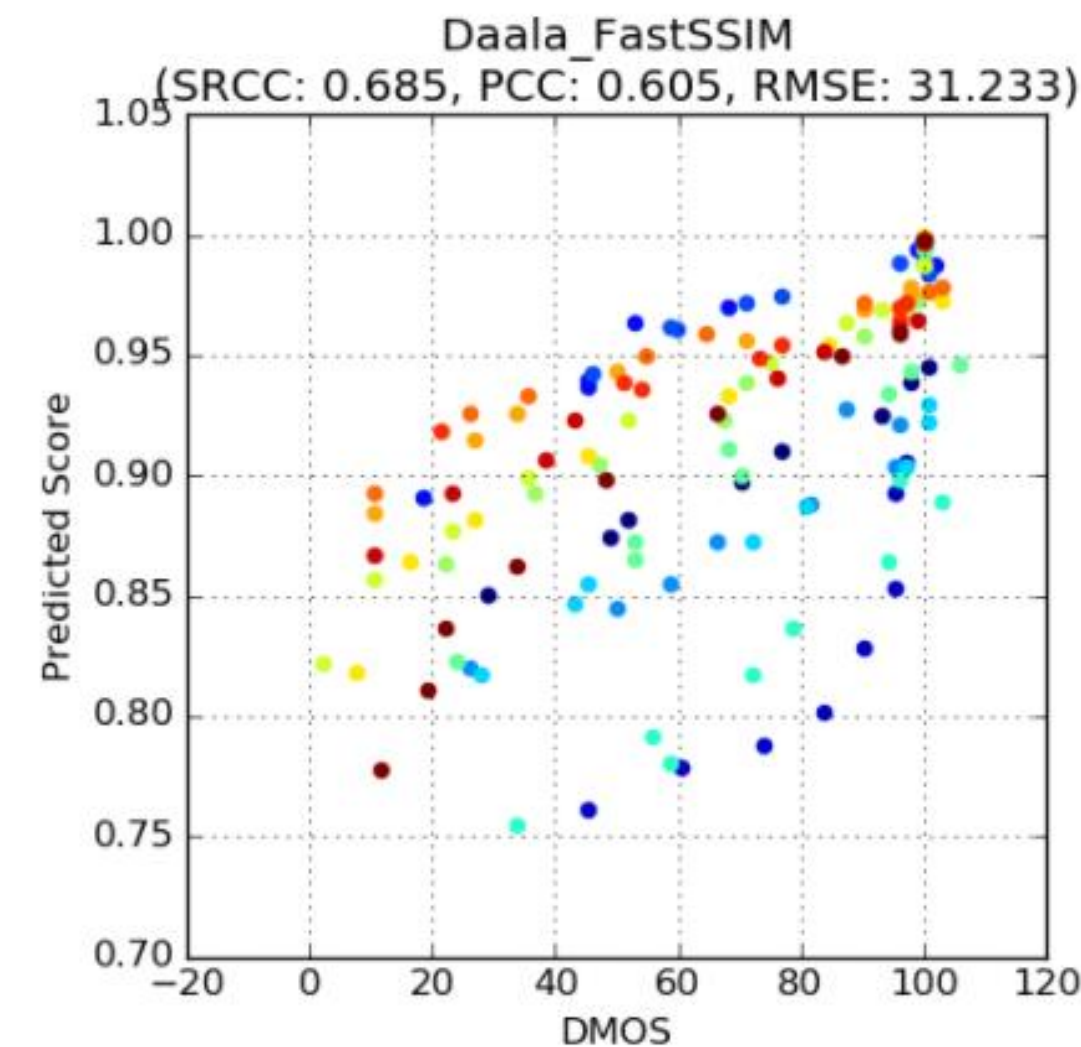
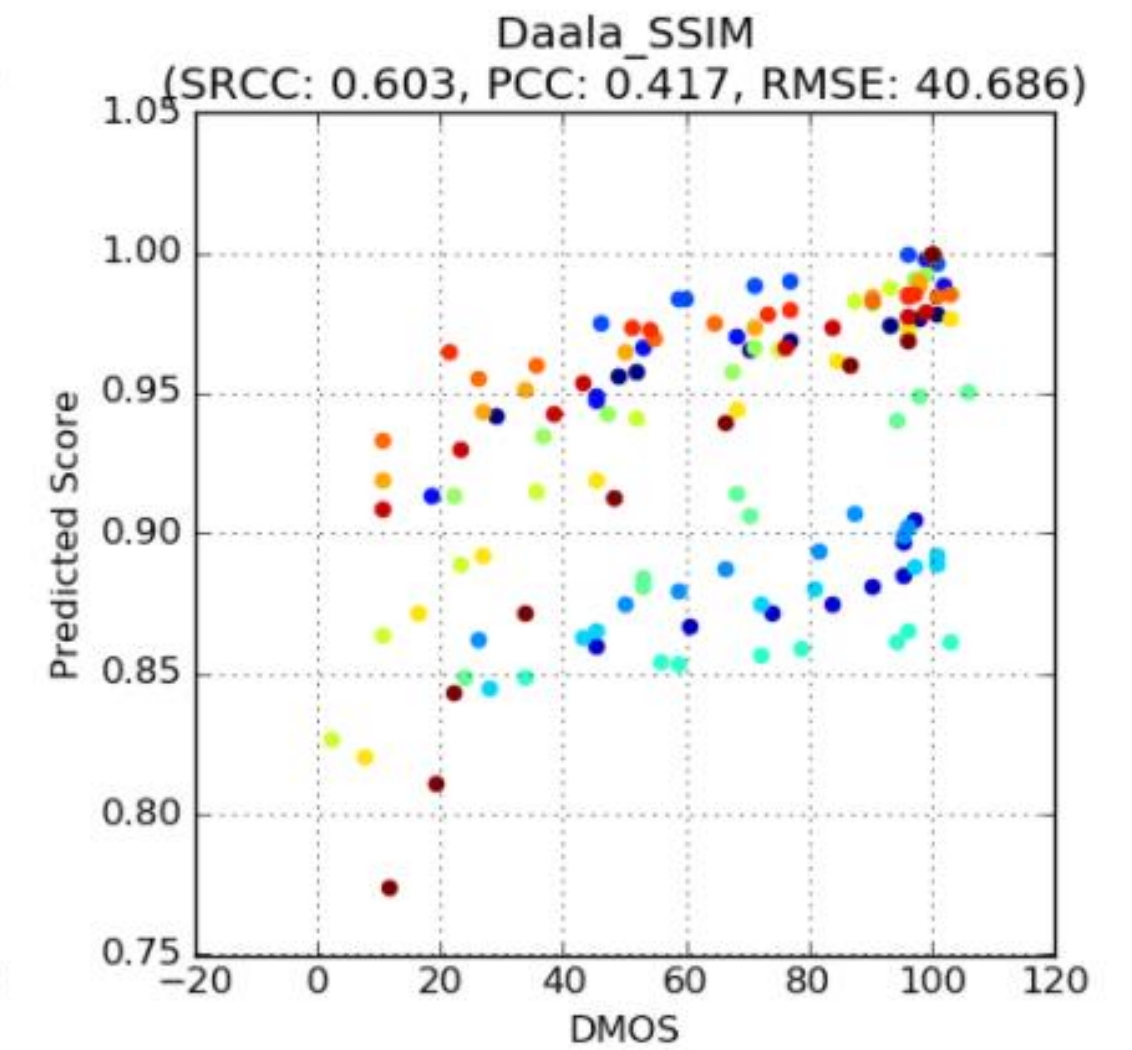
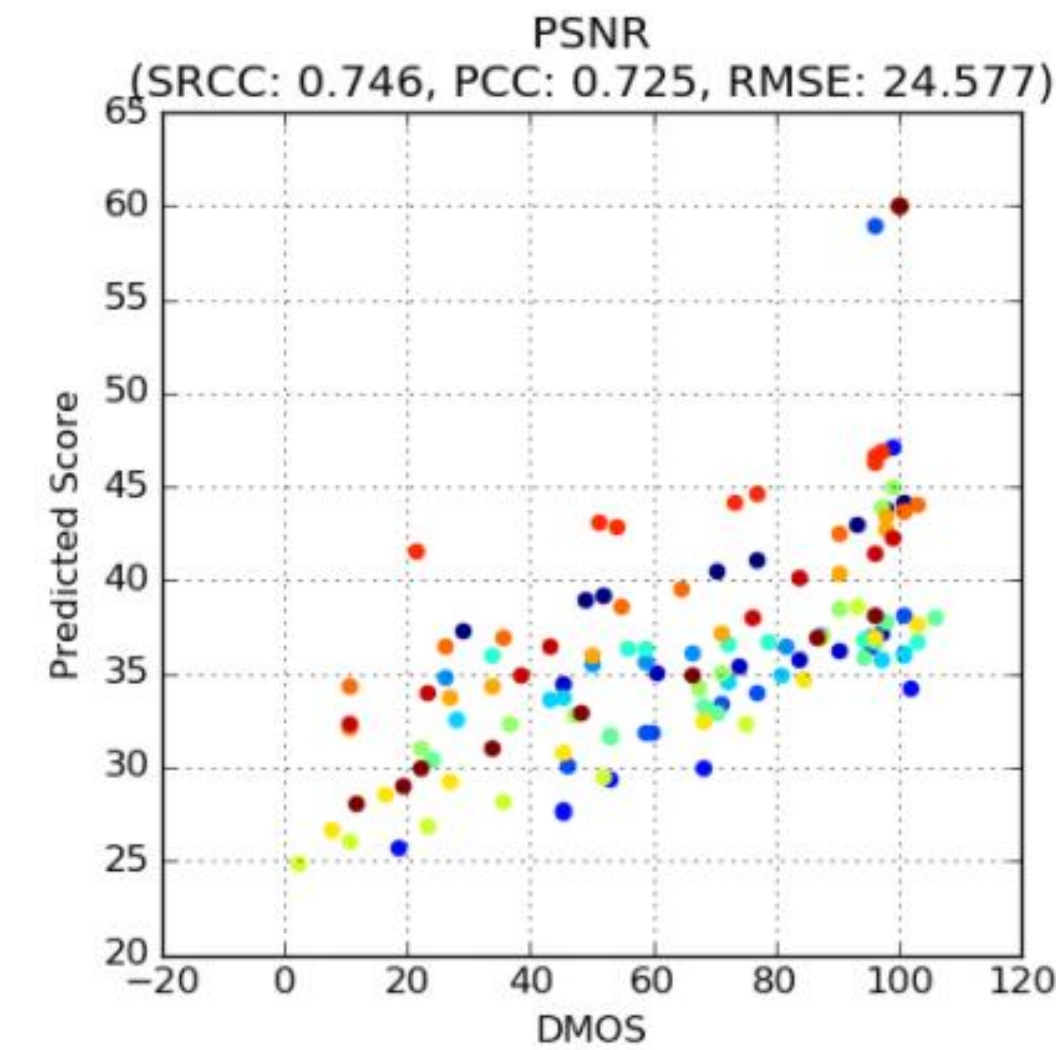
라벨 데이터 생성



라벨 데이터 생성

인코딩 품질의 기준은?

- VMAF가 사람의 지각적인 부분을 가장 잘 반영
- VMAF로 인코딩 결과의 품질 측정



데이터 정제와 검증

- 라벨 데이터 생성 오류 수정
- 라벨 클래스 불균형 개선
- 데이터 샘플링 오류 개선
- 세그먼트 생성 알고리즘 검증(결과물 전수 검증)

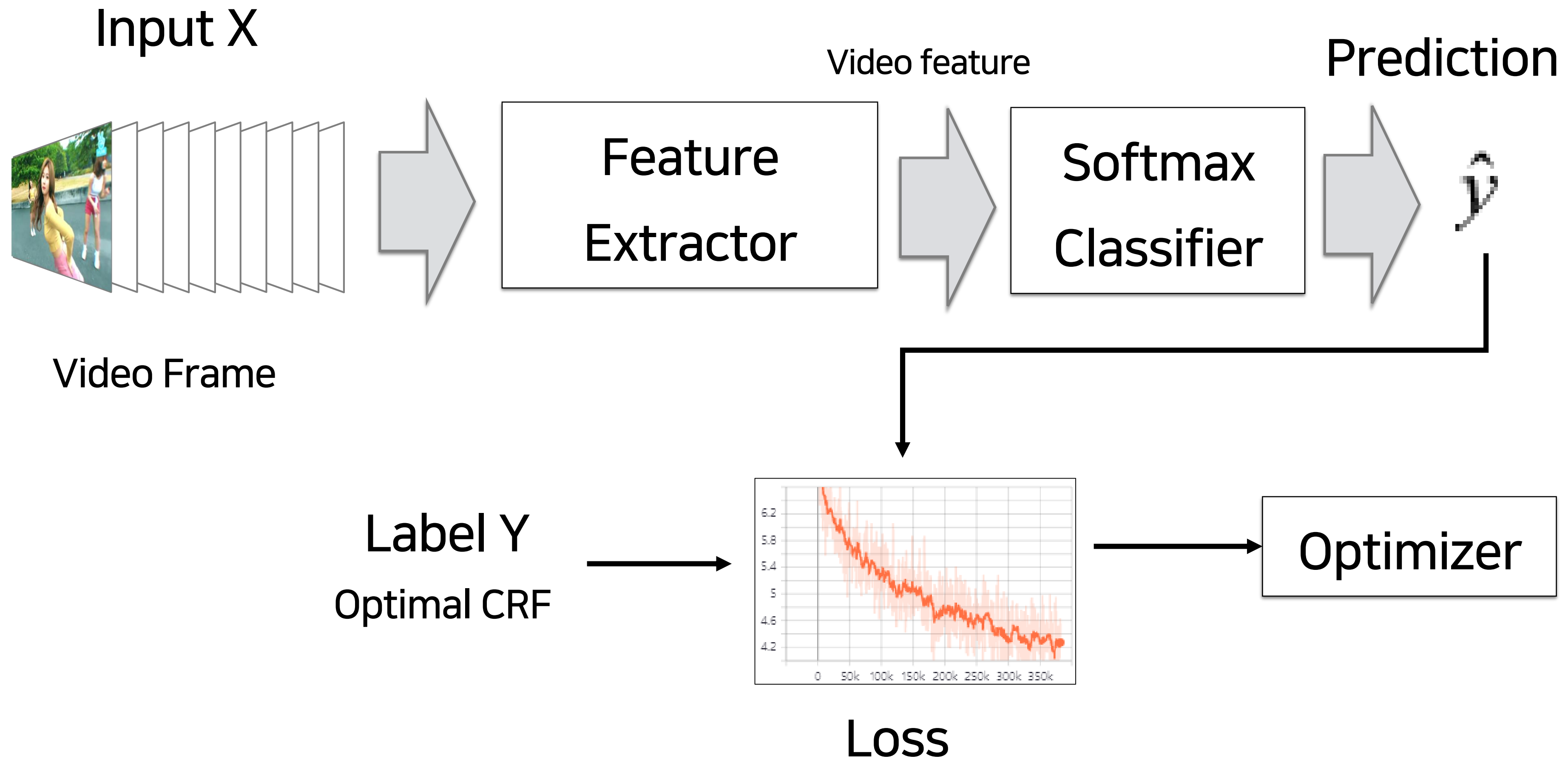
데이터 정제와 검증

- 라벨 데이터 생성 오류 수정
- 라벨 클래스 불균형 개선
- 데이터 샘플링 오류 개선
- 세그먼트 생성 알고리즘 검증(결과물 전수 검증)

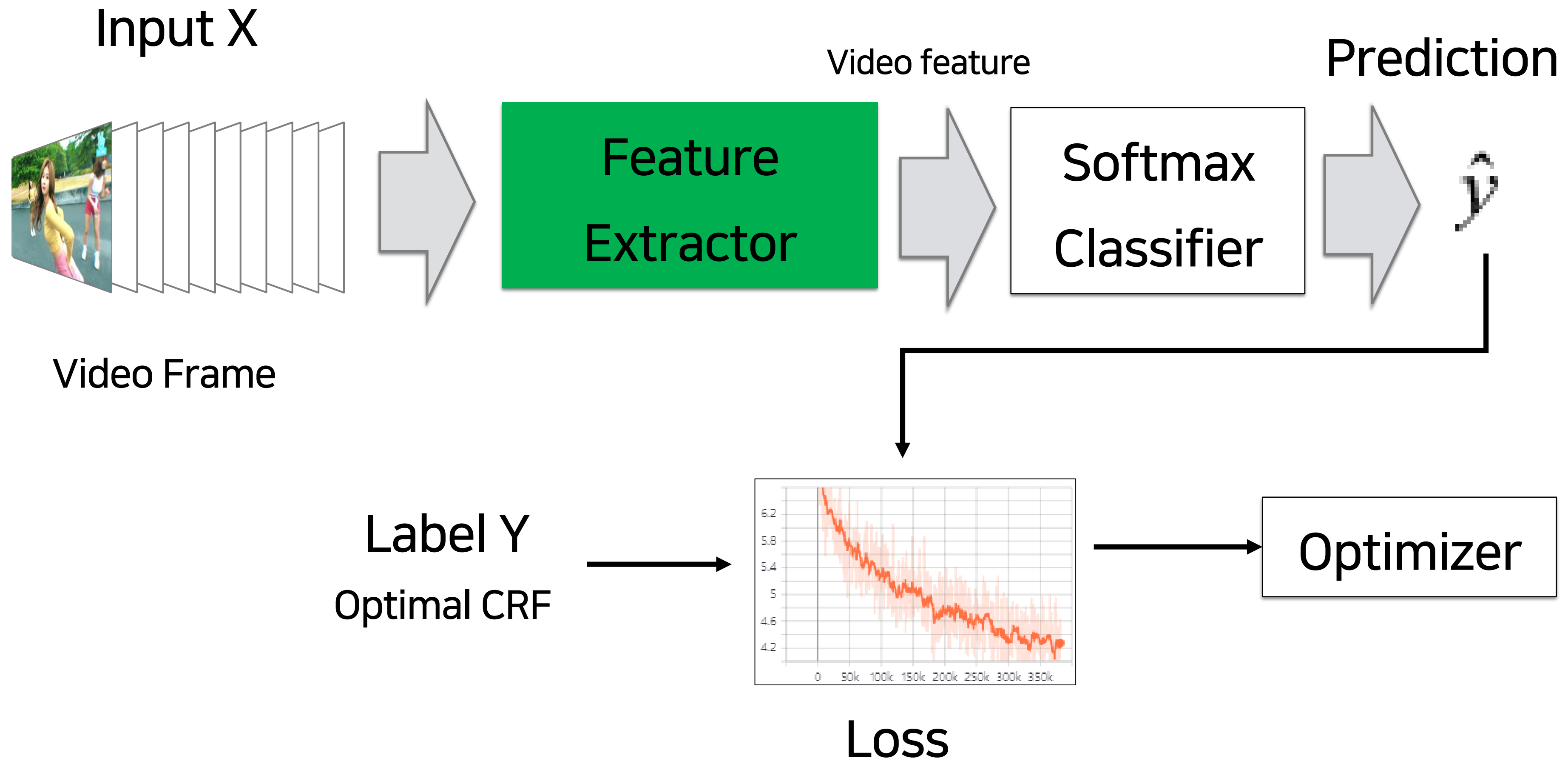
→ 데이터 검증 & 재생성 반복

→ 검증된 데이터셋을 잘 만드는 것이 중요!

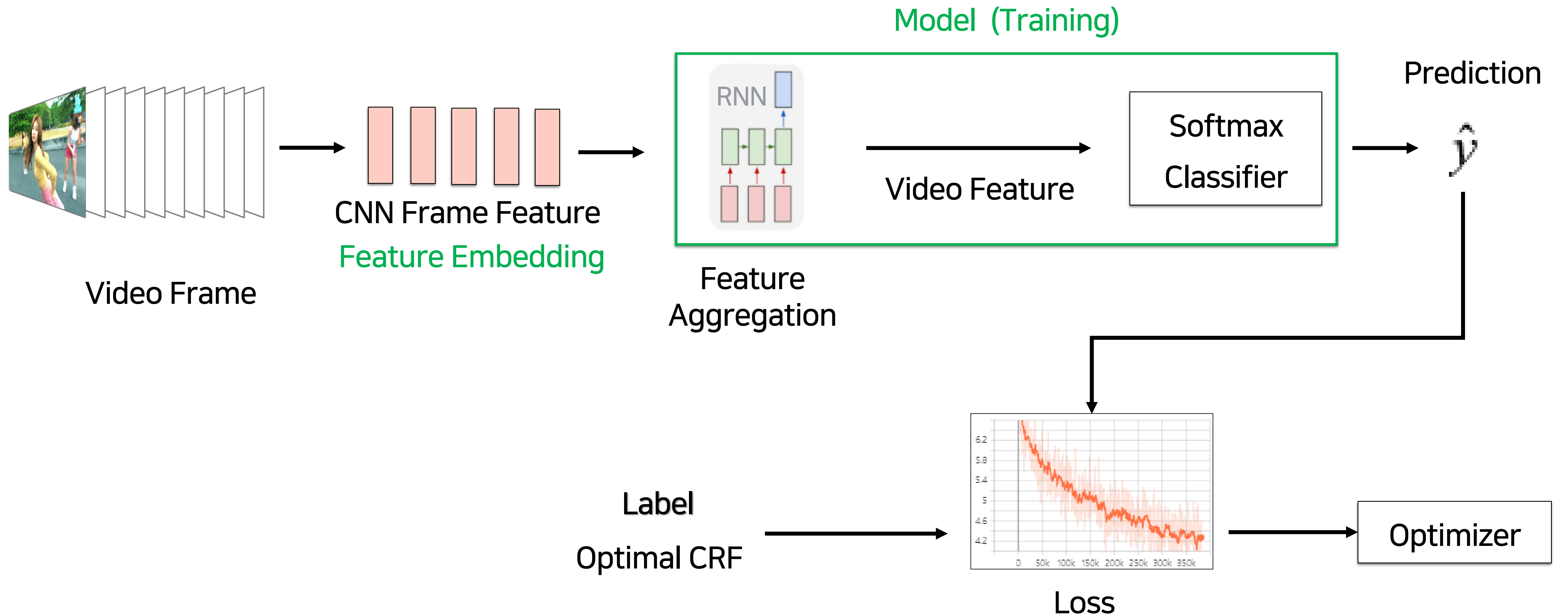
모델 아키텍처



모델 아키텍처



모델 아키텍처



모델 아키텍처 개선

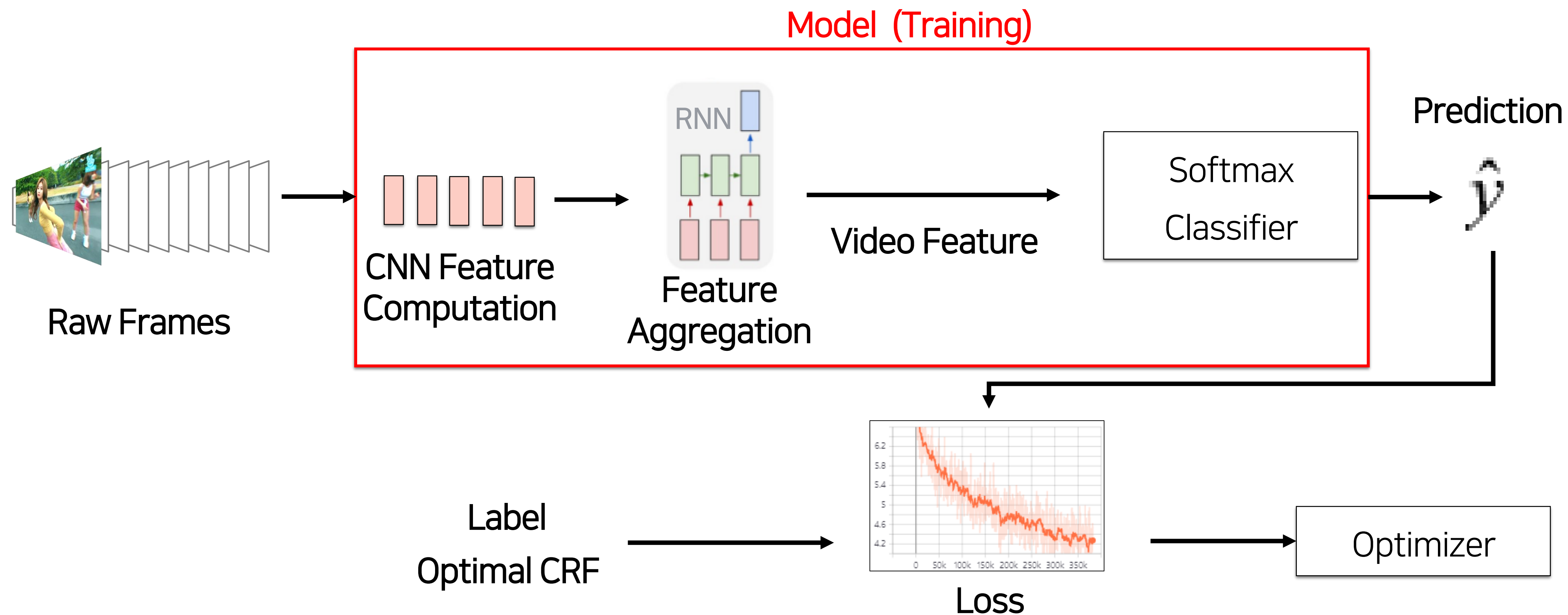
Frame Feature (Feature Embedding)

- ImageNet based Pre-trained CNN model
- Complexity와 같은 영상의 특성과는 일부 차이가 존재

→ CNN Fine Tuning & End-to-End Training

모델 아키텍처 개선

End-to-End Training with CNN



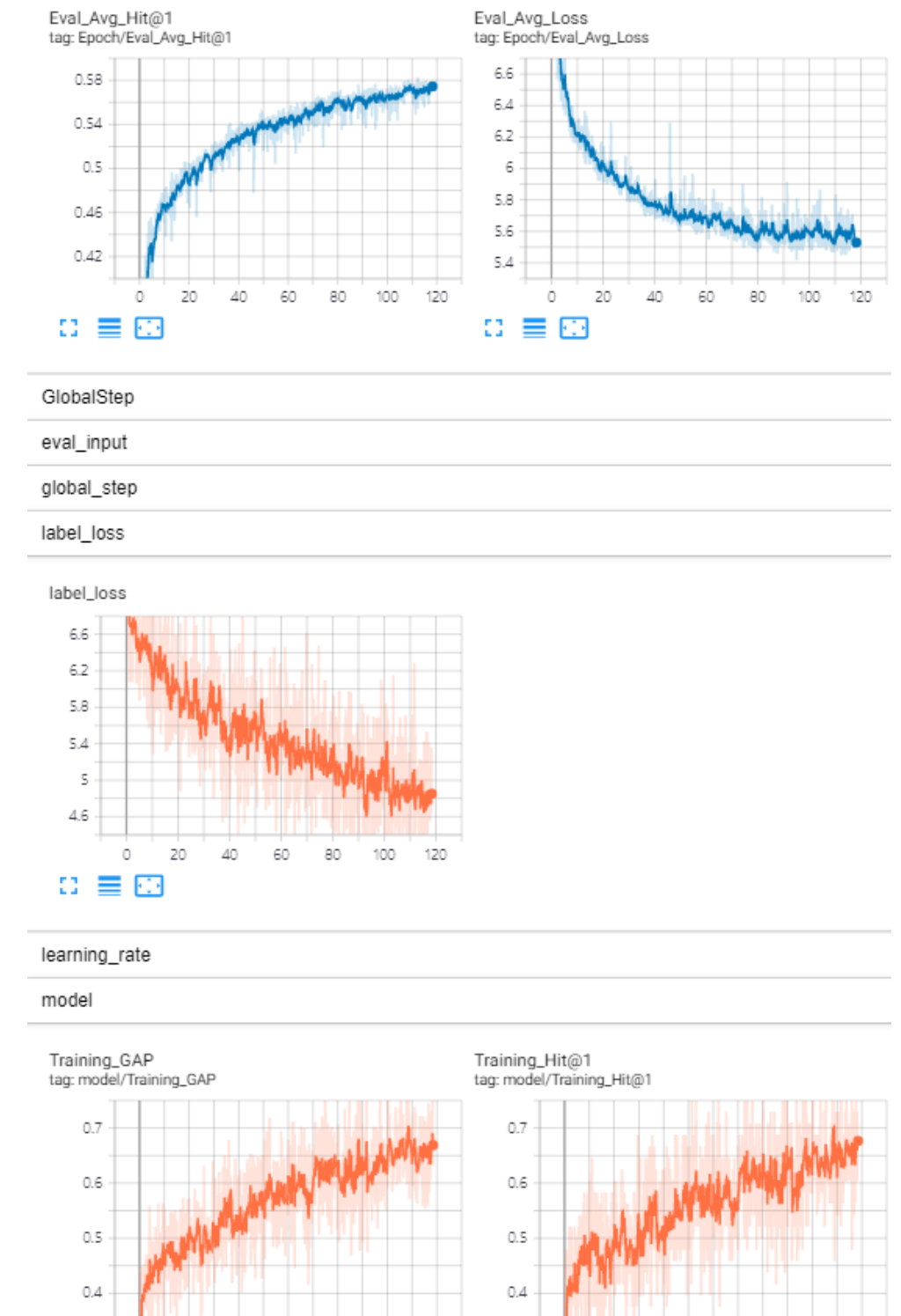
모델 아키텍처 개선

End-to-End Training

- 신경망이 너무 깊거나 메모리 문제로 트레이닝이 어려운 경우가 있음
- Lightweight CNN Model부터 점차 키워나감
- 트레이닝 메모리 문제가 없도록 Frame Image Size와 Frame Image 개수(Frame Image 간격)를 최적화

모델 개발 및 트레이닝

- Tensorflow 1.15 기반으로 개발
- Tensorflow Slim 제공 모델 및 파라미터 이용
- Training/Validation/Test Dataset
- Tensorboard 통해서 트레이닝 과정 모니터링
- GPU 이용해서 Training과 Evaluation 동시 수행



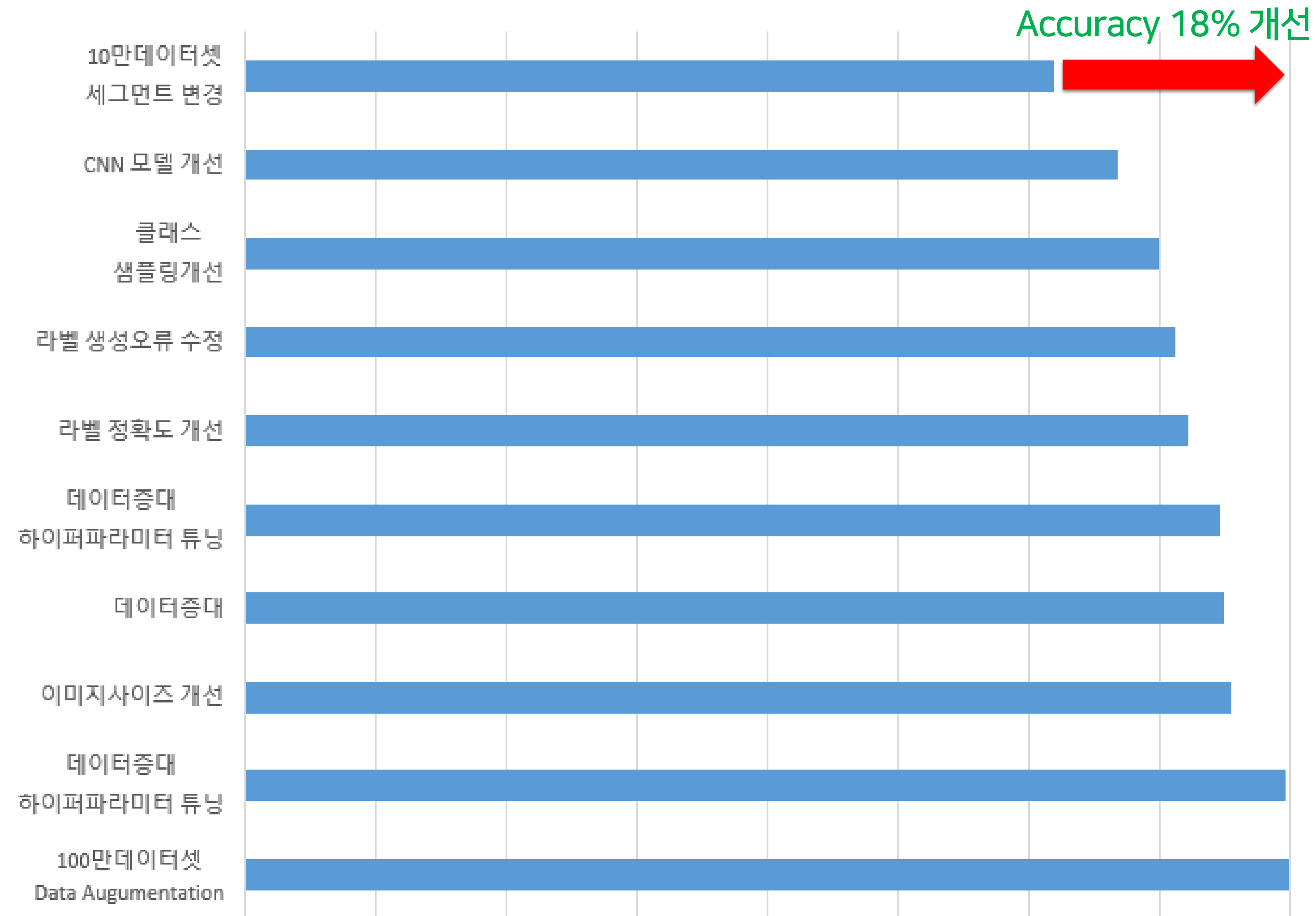
모델 개발 및 트레이닝

100만
데이터셋

데이터준비
400시간

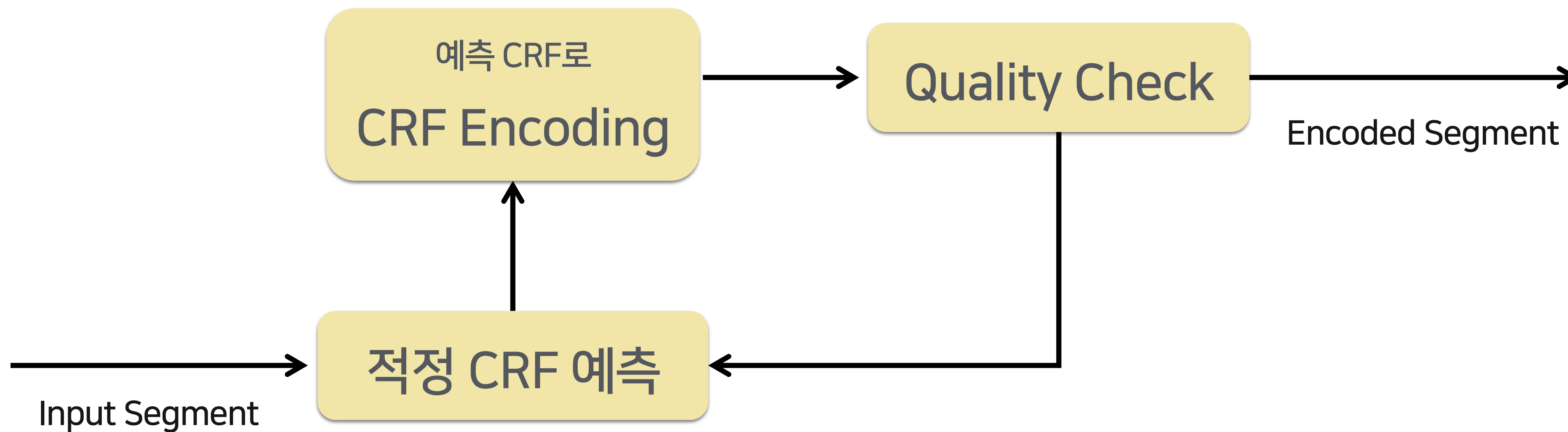
모델 트레이닝
10,000시간

모델 성능 개선



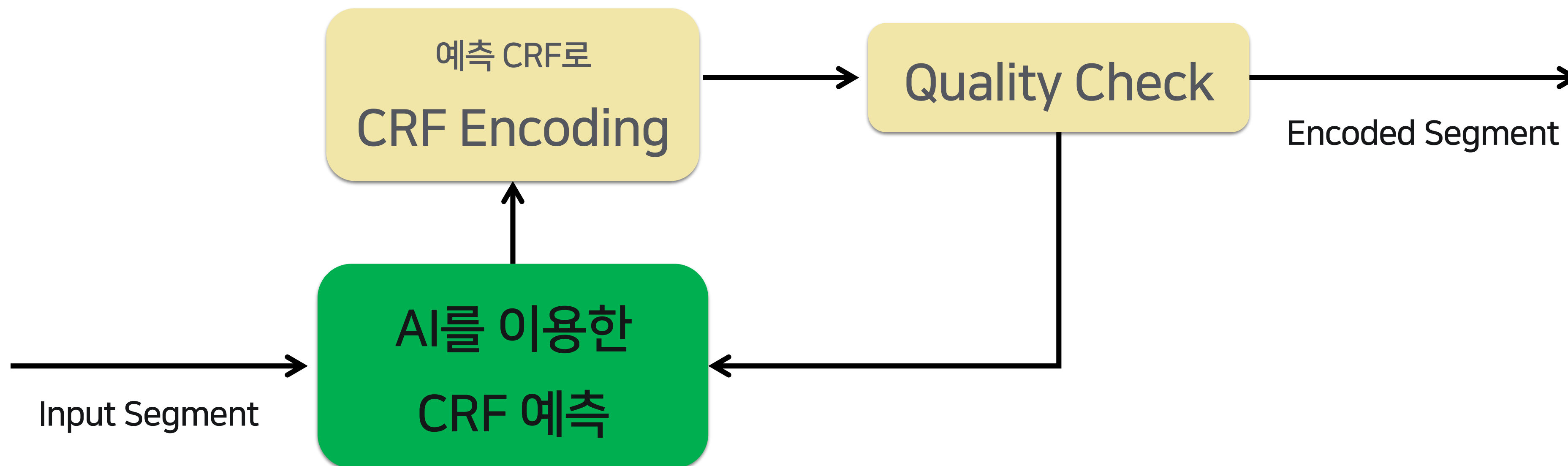
서비스 적용

전체 영상의 일정한 화질을 위해서는 반복적인 과정 필요



서비스 적용

전체 영상의 일정한 화질을 위해서는 반복적인 과정 필요
AI를 통한 예측으로 반복적인 과정을 최소화



서비스 적용 이슈

- 1) 예측 정확도가 100% 아님
- 2) 해상도별 서비스 비트레이트를 초과
 - 최소의 추가 인코딩으로 비트레이트 범위와 목표 화질을 달성

서비스 적용 이슈

- 1) 예측 정확도가 100% 아님
- 2) 해상도별 서비스 비트레이트를 초과
→ 최소의 추가 인코딩으로 비트레이트 범위와 목표 화질을 달성
- 3) 세그먼트 단위
→ 분산인코더 기준으로 세그먼트 단위 조정

4. 네이버의 Next

서비스 적용과 개선 효과

- 분산 트랜스코더 + AI 옵션 최적화
- 20~25%의 비트레이트 개선
- 사용자 데이터 사용량 절감
- QOE 개선을 통한 만족도 향상 기대

인코딩 개선을 위한 지속적인 연구

인코딩 최적화 모델 개선

- 적용 해상도의 확대
 - 4K 이상 모델 지원
- 세그먼트 단위의 개선
 - Shot-based segment
- 모델의 추가 성능 개선
 - 장르별, 서비스별 성능 최적화
 - 최신 데이터에 맞춰 지속적인 성능 개선

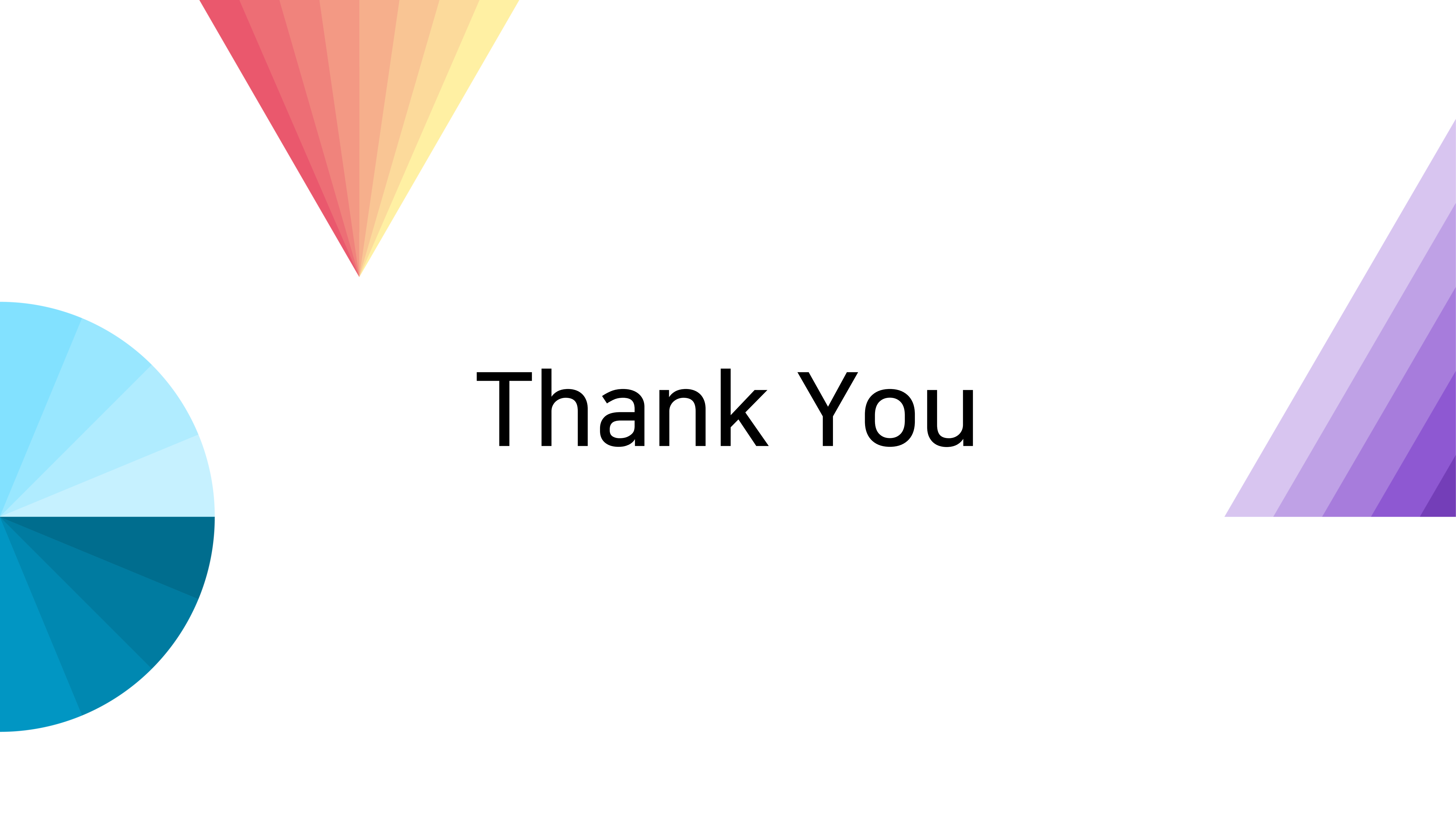
인코딩 개선을 위한 지속적인 연구

Encoding Ladder 최적화 연구

신규 포맷 최적화 연구



Q & A



Thank You