

# Edge-Cloud Collaborative Systems for Live Video Analytics

## ◆ On-device approach

: 비디오가 캡처되는 기기 자체에서 처리

- ✓ 장점 - 자체에서 처리하기 때문에 효과적,
  - 네트워크 영향을 덜 받음
  - 프라이버시 우려가 적음
- ✓ 한계점 - 단일모델의 execution 만 가능
  - 복잡한 task는 무리가 됨
  - resolution이 낮은 것만 빨리 됨

## ◆ Edge-cloud collaborative : 데이터를 클라우드로 넘겨서, 클라우드가 처리한 후 기기로 다시 넘기는 형식

- ✓ On device가 가지는 한계점들을 극복해 줄 가능성이 있음
- ✓ Powerful computing resources - 20배 정도
- ✓ 네트워크 문제 - 현재 많이 고민되고 있음
- ✓ 프라이버시 문제 - 해결하기 위한 다양한 노력

⇒ 주목 해봐야 할 만한 approach 이다.



But, processing latency benefit is quickly compromised by data transmission latency

⇒ 핵심 문제 : **어떻게 하면 네트워크로 데이터를 효과적으로 보낼 것인가?**

- 1. Split inference approach
- 2. DNN-aware compression approach

# Edge-Cloud Collaborative Systems for Live Video Analytics

## ◆ Split inference approach

: 처리 구조화를 잘 하자.

### ❖ Layer-aware Split Inference (vertically)

- ✓ Edge-only inference : 기기 자원 제약  
= large latency, energy consumption
- ✓ Cloud-only inference : 네트워크 제약  
= network condition에 vulnerable 함
- ✓ **Split inference** : 둘을 잘 나눠서 같이함  
= fast, energy-efficient, robust  
빠르면서 효과적임

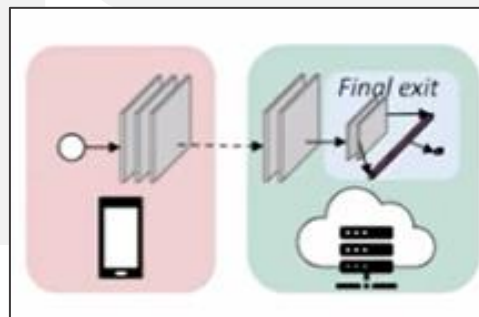
### ❖ Content-aware Split Inference (horizontally)

- ✓ 이미지의 내용별로 나눔
- ✓ Face detection, Lightweight  
큰, 쉬운 영역 -> On-device
- ✓ Heavy, Identity clarification  
작은, 세밀한 영역 -> Cloud

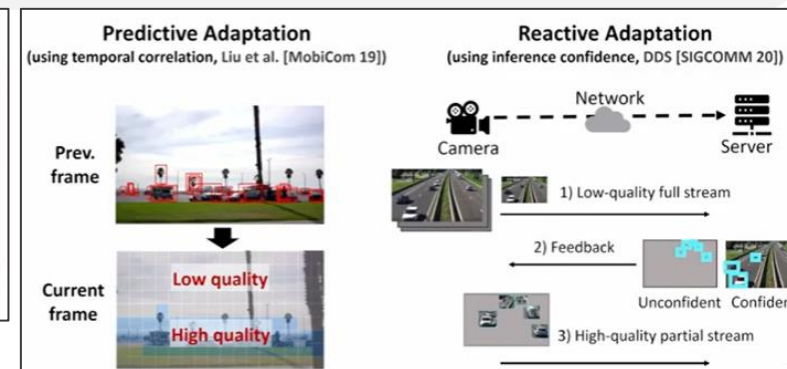
## ◆ DNN-aware compression approach

: 데이터를 압축해서 보내자

- ✓ hard objects -> high-quality
- ✓ easy objects -> 더 압축해서 low-quality  
=> 어떻게 어렵고 쉬운지 아는지?
- ❖ Predictive vs. Reactive Adaptations
  - Predictive - 과거의 frame을 참고하여
  - Reactive Adaptation
    - 1) low-quality로 한번 보냄
    - 2) Feedback (confidence 활용, 탐지 안되는 영역 다시하라고 함)
    - 3) high-quality로 다시 보냄

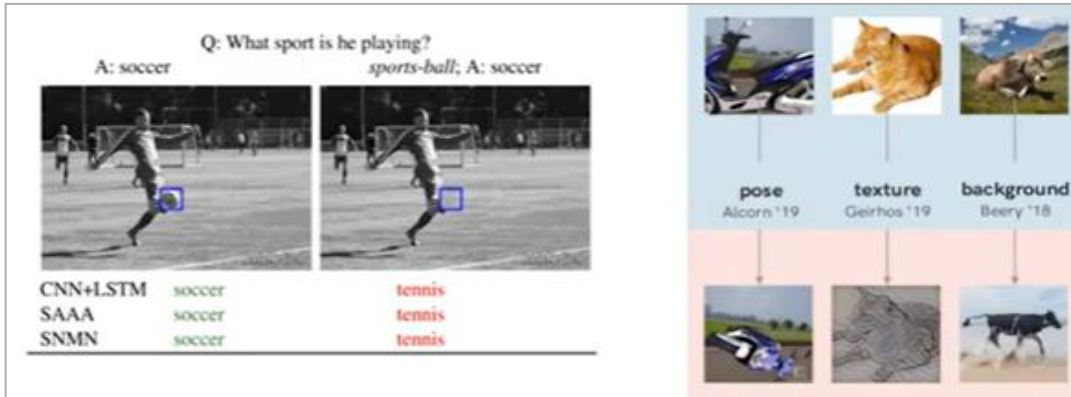


> Split inference



# Mitigating Shortcut Learning of NLP models

## 1 - What is the shortcut learning problem?



Deep neural model이 학습을 하는 과정에서 shortcut의 전략을 학습하게 됨. 각종 benchmarks에선 잘 결정하는데 challenging한 testing 환경에선 잘못 판단하는 것.

Ex) 축구사진에서 축구공이 사라지면 테니스로 잘못 판단

## 2 - Why neural models learn shortcuts?

- Annotation artifacts => NLI task에서 많이 일어남
- NLI task : 두개의 문장(전제, 가설) 관계가 Entailment(함의), Neutral(중립), Contradiction(모순)인지 분류
- Contradiction 에 never, no, nothing과 같은 단어들이 많이 분포함  
=> annotation 단계에서 편견이 발생하고, 모델이 학습함
- Classification 문제 : 두 개의 문장 사이의 관계가 아니라 hypothesis에 있는 단어만으로 추론
- training set에 따라 왜 이런 일이 일어나는지 분석 – shortcut / challenging version dataset  
=> Training set에서 challenging한 example 만드는 것도 중요하다.

# Mitigating Shortcut Learning of NLP models

## 3 - How to mitigate such problem for robust models?

### Directions

**Avoiding annotation artifacts**

- Guidance for clean/reliable labels
- Debaised/balanced labels
- Adding challenging examples

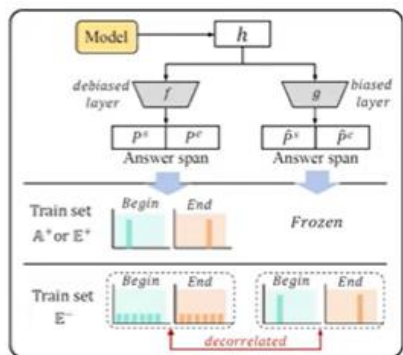


**Debiasing models**

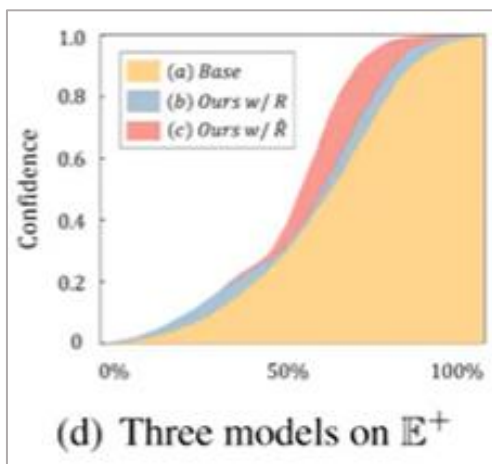
- Removing spurious correlations
- Improving generalization
- Improving robustness



### > 최종 full 모델



biased layer,  
debaised layer

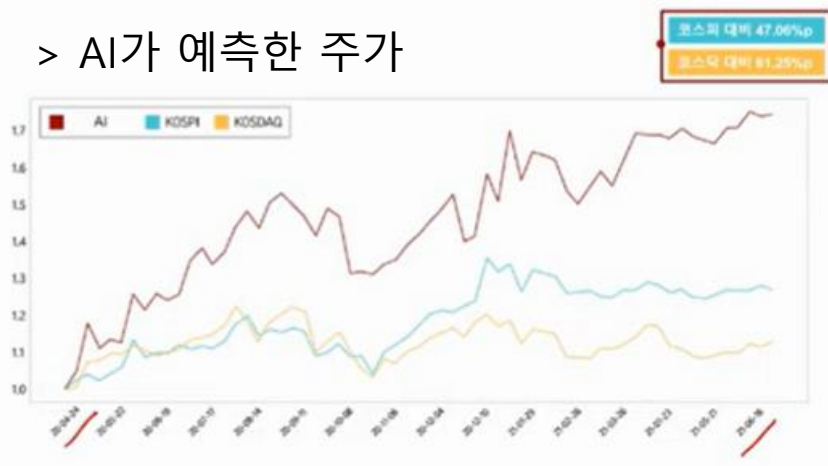


- avoiding shortcuts in QA model - evidentially 이용
  - problem : reasoning shortcut = 증거 일부만 읽음
  - solution : **training evidentiality**  
= 정답이 있더라도 증거가 있는지 모델이 판단해야 한다.
  - Evidence-negative set : 정답 포함하지만 증거가 없음
  - Evidence-positive set : 정답과 증거 모두 있음
- ⇒ 증거가 충분할 때 정답을 맞추게 하고,  
증거가 없을 때 정답을 맞추지 못하게 한다.

1. single-paragraphs로만 train한 모델 => shortcut 학습이 됨
2. evidence-negative set으로 regularization한 모델  
=> 증거가 없을 때 confidence가 0으로 가까워짐
3. 최종 full 모델
  - Evidence-negative set에서 낮은 confidence를 유지하면서
  - Evidence-positive set에서 confidence를 더 증가시킴

# Stock Prediction with AI

> AI가 예측한 주가



## ■ Why AI for Trading? (정당성)

- 사람은 기억력에 한계가 있고, 공포심이 있음
- AI는 패턴을 조기에 탐지가능
- 수십년 간의 데이터 활용 가능
- 최고의 전략으로 빠르게 찾을 수 있음
- Mission-critical services는 오류가 있으면 안되는데 (주로 보조역할로 사용) trading 분야에서는 오류가 있어도 된다.

## □ 주가 예측하는 연구

- 주식가격 예측하기 = Challenging but rewarding
- 주식이 다음날 오를지 내릴지 결정
- Previous models
  - Univariate models - 예측할 주식만 고려
  - Multivariate models with fixed correlations - 여러가지 주식 + 상관관계 고려

=> 어떻게 하면 주식들 간의 동적인 **correlation**을 사전지식 없이 잘 학습할 수 있을까?

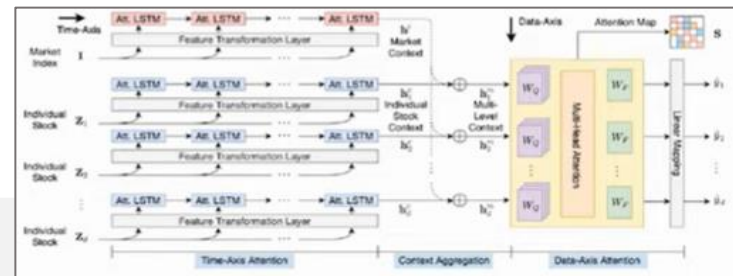
# Stock Prediction with AI

## □ Proposed method

### ▪ DTML(Data-axis Transformer with Multi-Level contexts)

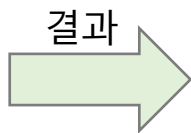
- Time-axis attention - 어떤 날짜가 다음날 예측할 때 도움이 될지 동적으로 학습
- Context aggregation - 개별 주식 + 시장의 트렌드
- Data-axis attention - 여러 주식 간의 상관관계를 동적으로 학습

=> 시간축 + global market + 종목 간의 연관관계 학습



## □ DTML 실험

- Dataset : 2(기존) + 4(새로운)  
= train / valid / test sets
- Evaluation metrics 평가 지표 :
  - Simple accuracy (ACC)
  - The Matthews correlation coefficient (MCC)



### 1. Accuracy 정확한지

- 모든 지표에서 highest ACC and MCC

### 2. Profit 실제 성과

- 실제로 0.8% ~ 13.3%의 수익을 얻음

### 3. Correlations 종목간 관계 잘 찾는지

- 어떤 종목이 영향을 주는지, 중요한 날짜 구별

### 4. Ablation study 각 구성요소가 도움이 되는지

- 각 요소들을 뺀 것보다 모든 요소를 더했을 때 정확도가 가장 높다



# Geometry in the Deep Learning Era

## ❖ Computer vision in action

- Object detection
- Segmentation
- Pose detection

## ❑ 3D vs 2D

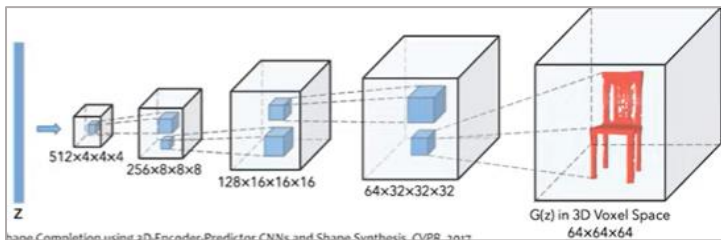
- 3D - 실제 metric 반영 (몇 cm에 있는지)
  - 빈공간이 존재
  - 이미지보다 비싸고 메모리 많이 필요
- 2D - 수집하기 쉬움
  - metric 정보가 없음
  - Semantic과 딥러닝에 좋음
- 3D와 2D를 잘 결합해서 각 장점을 살려서 어떻게 사용할지

## ❑ 3D / 2D measurement to perception

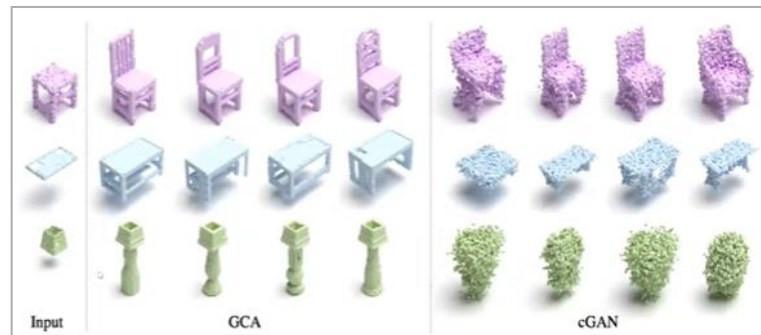
### 1. 3D shape completion (ICLR 2021)

=> 다양하고 디테일이 살아있는 3d shape 만들기 위해서

- 3d Generation Models - 해상도 문제
    - Sparsity : high 해상도 -> low 점유율
    - Connectivity : 점점 grow하는 형태
  - Generative Cellular Automata (GCA) 사용
    - 점진적으로 자라나게 하기 위해
    - initial shape => 그 주위로만 자라남
- => 여러 번 자라나다 보면 최종적인 shape에 다가감



### > 3D Generation Models



### > GCA

# Geometry in the Deep Learning Era

## 2. 2D-3D localization (ICCV 2021)

: 3d + 2d data 어떻게 잘 조합할 것인가

- 3d data에 2d 이미지를 매칭하는 formulation
- Panorama 이미지 사용
  - Global context 캡처
    - = 반복적 물체와 작은 변화에 robust
- Starting point 부터 색깔을 비교하며 loss를 계산 => smallest lost 고름

=> 2D localization on 3D map - 실용적인 usage

=> 파노라마 이미지 사용이

visual localization의 한계점을 극복함

=> 빠르고 정확함

=> 딥뉴럴 network를 사용하지 않음

## 3. Dynamic entity extraction (CVPR 2021)

: 3d 정보 없이 이미지만 가지고 물리적인 법칙을 표현할 수 있을까

- Agent (active)
- Objects (passive)
- Background (static)

=> agent를 따로 쪼개서 비디오만으로도 physical interaction을 잘 표현할 수 있었다.

