[논문]

Virtual Piano using Computer Vision

- https://arxiv.org/pdf/1910.12539.pdf

- **궁금한 점**

  1). 실제 actual 데이터를 사용할 때, audio & video 의 **sync 를 맞추기 위해 어떤 노력**을 했는지
  → Audio 데이터를 사용하지 않고, **무음 비디오** 사용

  2). Video 에서 어떤 식으로 audio 정보를 추출했는지

  [논문 핵심]
  - 연주 비디오로부터 얼마나 많은 연주 Info 를 추출할 수 있는지
  - 따라서, **오디오 데이터가 따로 없음**

  3). 모델을 어떤 식으로 사용했는지
  → **CNN**
  → **LSTM + 3D-ConvNet, two-stream 3D-ConvNet**

  실시간 처리 고려

  4). 서로 다른 노래에 대해서 어떻게 모델에 train 시켰는지
  → 동일한 배경 이미지 (건반 이미지 사용)

- **요약**
  - **Video 데이터** 사용
    - Image 사용 X
    - Video (시간 & 공간에 대한 정보 포함)
  - (논문 배경)
    연주 : Audio 뿐만 아니라 Visual 적인 요소도 영향을 준다
  - [논문]
    - 이전 연구) 손의 움직임 + 키 On / Off + 얼마나 키를 눌렀는지
    - 해당 연구) 이전 연구 + **키 누름의 강도**
  - 비디오 연주 (시각 정보) 로만으로도
    [연주 관련된 여러 정보 분석]
    - **무음 비디오**로 촬영
    - 추출된 연주 Info
      ① **키 On / Off**
      ② **키 누름 지속성 (얼마나 키가 오래 눌렸는지)**
      ③ **키 누름 강도**

- **연구 및 모델**

1 연구

1). 키 On/Off

- RELU / 최대 풀링 / 드롭 아웃 레이어 x 2번
- **2D CNN**

[피아노 연주] → ((시각 정보)) → 컴퓨터 비전 알고리즘 → 키보드 & 특정 키의 위치

⇨ **2D CNN**

2). 키 강도

- 손의 움직임 → (영향을 미침) → 키 누름 강도

(움직임 == 시간 정보)

- **3D CNN → 2D Image 벡터로** 줄임

⇨ **3D CNN** + Early fusion 기술 (손의 움직임을 분석하는 모델) ⇨ **새로운 CNN 모델 (시공간 반영) 개발**

[비디오] ⇨ 시간 & 공간 정보 ⇨ CNN 모델 이 적합

(추후에 모델 개발에 참조)

Operation	Kernel Size	Output Size	Operation	Kernel Size	Output Size
Input feature vector		(n, 600)	Input feature vector		(n, 400)
Reshape		(n, 10, 60, 1d)	Reshape		(n, 10, 40, 1d)
Conv2D (Stride=1)	(3, 3, 1d, 16d)	(n, 10, 60, 16d)	Conv2D (Stride=1)	(3, 3, 1d, 16d)	(n, 10,40, 16d)
ReLU		(n, 10, 60, 16d)	ReLU		(n, 10, 40, 16d)
Max_Pool (Drop_Out)	(1, 2, 2, 1)	(n, 5, 30, 16d)	Max_Pool (Drop_Out)	(1, 2, 2, 1)	(n, 5, 20, 16d)
Conv2D (Stride=1)	(3, 3, 16d, 32d)	(n, 5, 30, 32d)	Conv2D (Stride=1)	(3, 3, 16d, 32d)	(n, 5, 20, 32d)
ReLU		(n, 5, 30, 32d)	ReLU		(n, 5, 20, 32d)
Max_Pool (Drop_Out)	(1, 2, 2, 1)	(n, 3, 15, 32d)	Max_Pool (Drop_Out)	(1, 2, 2, 1)	(n, 3, 10, 32d)
Reshape		(n, 3 x 15 x 32)	Reshape		(n, 3 x 10 x 32)
Dense		(n, 256)	Dense		(n, 256)
Dense		(n, 2)	Dense		(n, 2)

Table 1: Architecture for detecting whether keys are pressed or not

2 모델

1). CNN

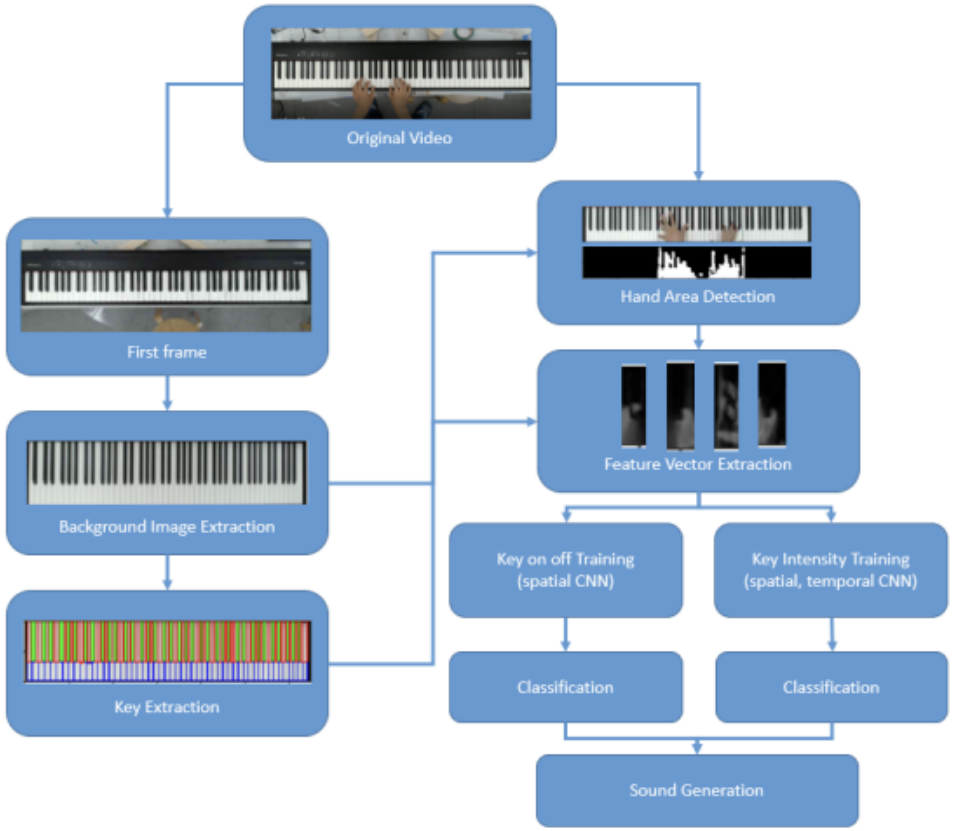
((비디오 기반)) → [시/공간 정보] → CNN --> **image fusing 방법**

(실시간에 대한 처리)를 위해 CNN 을 사용

- Image Fusing 방법
  - 다양한 layer 의 이미지 및 영상을 합성하여 노이즈 감소 / 선명도 증가
  - 여러 조건의 이미지를 결합하여 더 풍부한 정보를 추출

CNN (2 가지 사용)
- ① (공간) 정보만 필요한 경우
- ② (공간) & (시간) 정보 모두 필요한 경우

2). LSTM + 3D-ConvNet, two-stream 3D-ConvNet



- **Dataset**
  1). 데이터 수집
    - 웹 카메라로 (10명의 참가자) 로부터 **직접** 수집
    - 30 frame / 960 x 640 해상도
    - MIDI 데이터 수집
    - **28개의 비디오** →
  2). 데이터
    a). 건반 누름 감지
      * Key Off (0), On : (1)
    b). 강도 감지
      * 강도 레이블 : 강도 순으로 0 ~ 4 레벨
    c). 광학 흐름 지도를 사용한 강도 감지
  3). 데이터 불균형
    - **잘 나온 데이터에 대해서 더 높은 가중치를 부여**
  4). Optimizer
    - **Adam Optimizer 사용**

논문 전반적으로
- **엄청나게 많은 양의 비디오 데이터 셋을 필요로 하지 X**
- **하지만, 논문 연구에서 필요로 하는 정확한 데이터 셋을 사용**

- **정확도**
  - **92 ~ 94%**

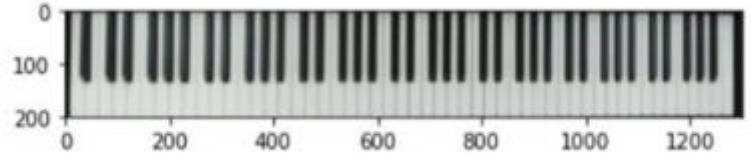
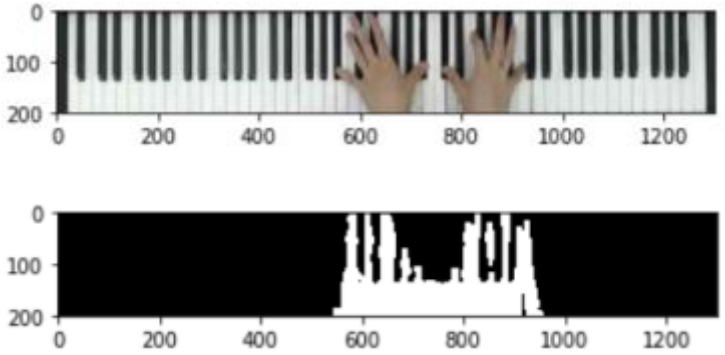


Figure 4: Result of finding keyboard area, which serve as our background image.



- **참고하면 좋을 점**
  - 비디오 데이터로부터 추출 할 수 있는 데이터 : (Key On / Off + 강도 + 키 누름 등)
  - Adam Optimizer 사용
  - 모델 (CNN) + LSTM 기술 사용