Action Recognition with an Inflated 3D CNN

목해민

Contents

1 논문 분석
| 모델 비교, Results

2 I3D
| 모델 선정 이유

3 오픈 소스
| 하드웨어 성능, Dataset

4 Test
| test결과, 모델 추가학습

5 데이터 분석
| 다양한 영상의 예측 값 분석

Video recognition

동영상 데이터는 기본적으로 공간적, 시간적 요소로 분해될 수 있다. **공간적 부분은** 동영상에서 묘사된 장면과 물체에 관한 정보를 담고있다. **시간적 부분은** 관찰자(카메라)와 물체의 움직임에 관한 정보를 담고 있다.

인간 행동 인식은 비디오 이해에서 중요한 역할을 하기 때문에 최근 몇 년 동안 활발한 연구 분야가 되었습니다. 일반적으로 인간의 행동은 외모, 깊이, 시각적 흐름 및 신체 골격과 같은 여러 양식에서 인식할 수 있습니다.

비디오 도메인에서 충분히 큰 데이터 세트에 대한 동작 분류 네트워크를 교육하는 것이 다른 임시 작업 또는 데이터 세트에 적용될 때 유사한 성능 향상을 제공하는지 여부는 열린 질문입니다. 비디오 데이터 세트 구축의 어려움은 동작 인식에 대한 가장 인기 있는 벤치마크가 10k 비디오 정도로 작다는 것을 의미했습니다.

논문 분석_모델비교

모델 비교 분석

a [5, 15, 37]	2D ConvNet + LSTM	[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. "Long-term recurrent convolutional networks for visual recognition and description." CVPR 2015. [15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "Large-scale video classification with convolutional neural networks." CVPR 2014. [37] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. "Beyond short snippets: Deep networks for video classification." CVPR 2015.
b (C3D)[31]	3D ConvNet	[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning spatiotemporal features with 3d convolutional networks." ICCV 2015.
c [27]	Two-Stream	[27] K. Simonyan and A. Zisserman. "Two-stream convolutional networks for action recognition in videos." NIPS 2014.
d [8]	3D-Fused Two-Stream	[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. "Convolutional two-stream network fusion for video action recognition." CVPR 2016.
e (I3D)[this]	Two-Stream 3D ConvNet	[this] J. Carreira and A. Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset." CVPR 2017.

논문 분석_모델비교

a. 2D ConvNet + LSTM

http://koreascience.or.kr/article/JAKO201931765018039.page

이 논문에서는 작물 분류를 목적으로 작물의 시공간 특징을 고려할 수 있는 딥러닝 모델 2D convolution with bidirectional long short-term memory(2DCBLSTM)을 제안하였다. 제안 모델은 우선 작물의 공간 특징을 추출하기 위해 **2차원의 합성곱 연산자**를 적용하고, **추출된 공간 특징을 시간 특징을 고려할 수 있는 양방향 LSTM 모델**의 입력 자료로 이용한다.

Table 5. Overall accuracy of 2DCBLSTM with respect to different parameters

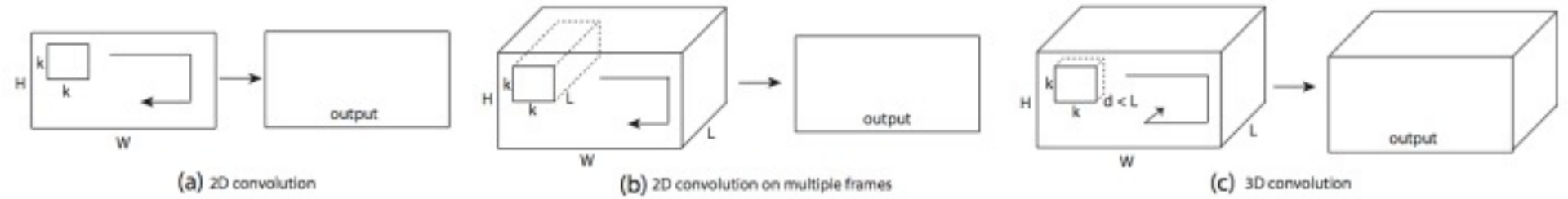
Case	The number of layers			Direction	Overall accuracy (%)
	Convolution	LSTM	Average pooling		
A	1	4	1	Forward	93.58
B	6	4	2	Forward	90.78
C	3	4	1	Forward	95.44
D	3	2	1	Forward	93.57
E	3	4	1	Bidirectional	96.59

- case B의 분류 정확도가 상 대적으로 가장 낮게 나타났는데 5×5의 작은 패치 크기 로 인해 풀링 계층 수가 늘어날수록 정보 손실이 크게 발 생한 것에 기인한 것으로 판단
- LSTM 계층 수를 증가시켰을 경우 특정 분류 항 목을 분류하지 못하는 결과를 보였는데, 이러한 결과는 LSTM의 내부 구조가 많은 수의 파라미터를 필요로 하기 때문에 LSTM 계층 수가 늘어날수록 기울기 손실 (vanishing gradient) 문제가 발생한 것이 주된 이유로 판단

논문 분석_모델비교

b. 3D ConvNet

https://arxiv.org/pdf/1412.0767.pdf



3d ConvNet은 3d convolution과 3d pooling 덕에 temporal information을 더 잘 모델링 할 수 있다.

•3d conv에서 conv, pooling은 시공간적으로 작동 가능. 2d conv는 오직 공간적으로만

- (a) 이미지에 2d적용: 아웃풋은 image
- (b) 복수의 이미지에 2d적용(다른 channel로 적용해서): 아웃풋은 image
  - 또한 2d conv는 매번 conv작동될 때마다 시간적 정보를 잃는다
  - 오직 3d conv만이 시간적 정보를 보존한다
- (c) 그 결과 3d conv에서만 아웃풋이 volume형태

논문 분석_모델비교

c. Two-Stream

https://proceedings.neurips.cc/paper/2014/fi
5c4359f4ae7bd7ba1-Paper.pdf

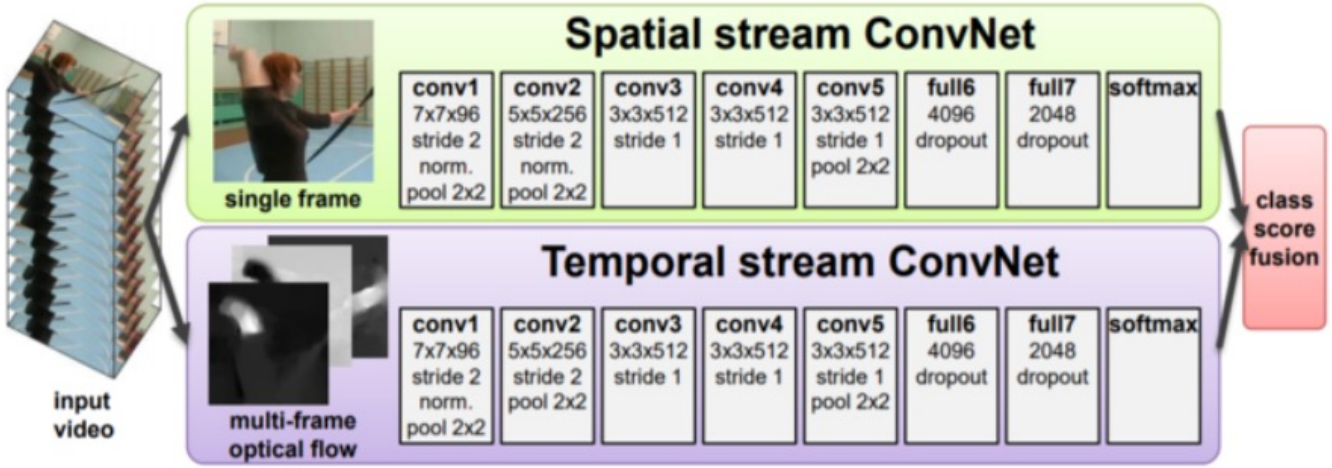


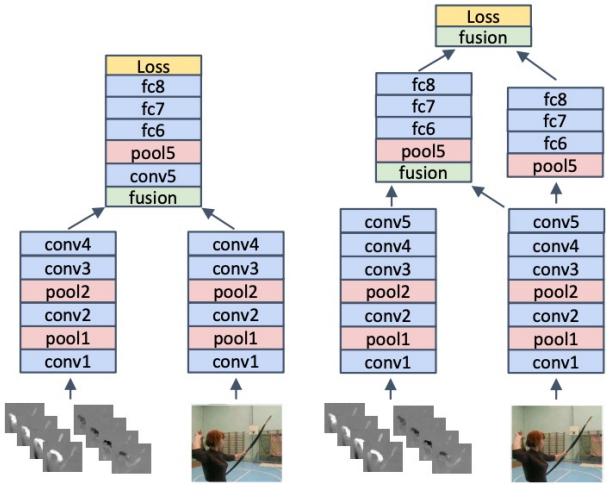
Figure 1: **Two-stream architecture for video classification.**

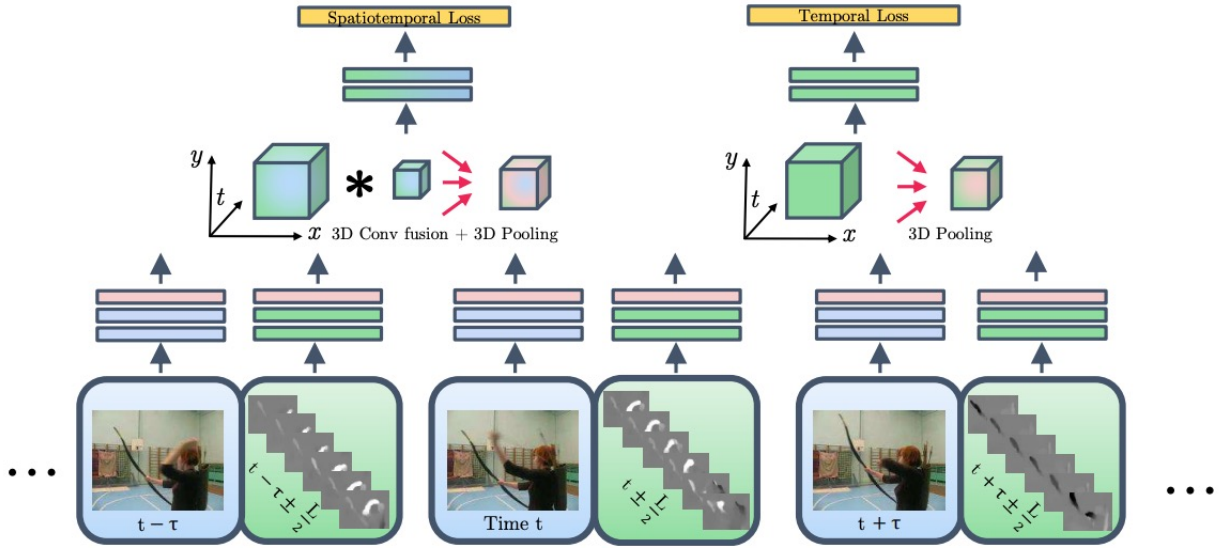
영상은 자연스럽게 공간적 요소와 시간적 요소로 분해될 수 있다. 공간 부분은 개별 프레임의 모습으로 영상에 묘사된 장면과 사물에 대한 정보를 담고 있다. 프레임을 가로지르는 모션 형태의 시간적 부분은 관찰자(카메라)와 객체의 움직임을 전달합니다. 이에 따라 비디오 인식 아키텍처를 고안하여 그림 1과 같이 두 개의 스트림으로 나눕니다. 각 스트림은 deep ConvNet을 사용하여 구현되며, softmax 점수는 late fusion으로 결합된다. 두 가지 융합 방법을 고려한다 : 누적된 L2 정규화 softmax 점수에 대한 다중 클래스 선형 SVM을 기능으로 평균화하고 훈련한다.

논문 분석_모델비교

d. 3D-Fused Two-Stream

https://openaccess.thecvf.com/content_cvpr_2016/papers/Feichtenhofer_Convolutional_Two-Stream_Network_CVPR_2016_paper.pdf





시공간 융합 ConvNet은 미세한 시간 척도(t ±L/2)에서 단기 정보를 캡처하는 2-스트림 ConvNet을 거친 시간 척도(t + T τ)에서 일시적으로 인접한 입력에 적용합니다. 두 스트림은 공간 스트림(파란색)과 시간 스트림(녹색)의 고도로 추상적인 특징과 x, y, t의 로컬 가중 조합 간의 대응 관계를 학습할 수 있는 3D 필터에 의해 융합됩니다. 퓨전 스트림과 시간적 스트림의 결과 특징은 시공간적으로 3D 풀링되어 입력 비디오를 인식하기 위한 시공간적(왼쪽 상단) 및 순수 시간적(오른쪽 상단) 특징을 학습합니다.

왼쪽 예는 네 번째 conv-layer 이후의 퓨전을 보여줍니다.
융합 지점에서 단일 네트워크 타워만 사용됩니다.
오른쪽 그림은 두 네트워크 타워가 유지되는 두 계층(conv5 이후 및 fc8 이후)에서의 융합을 보여줍니다.
하나는 하이브리드 시공간 네트워크이고 다른 하나는 순수 공간 네트워크입니다.

논문 분석

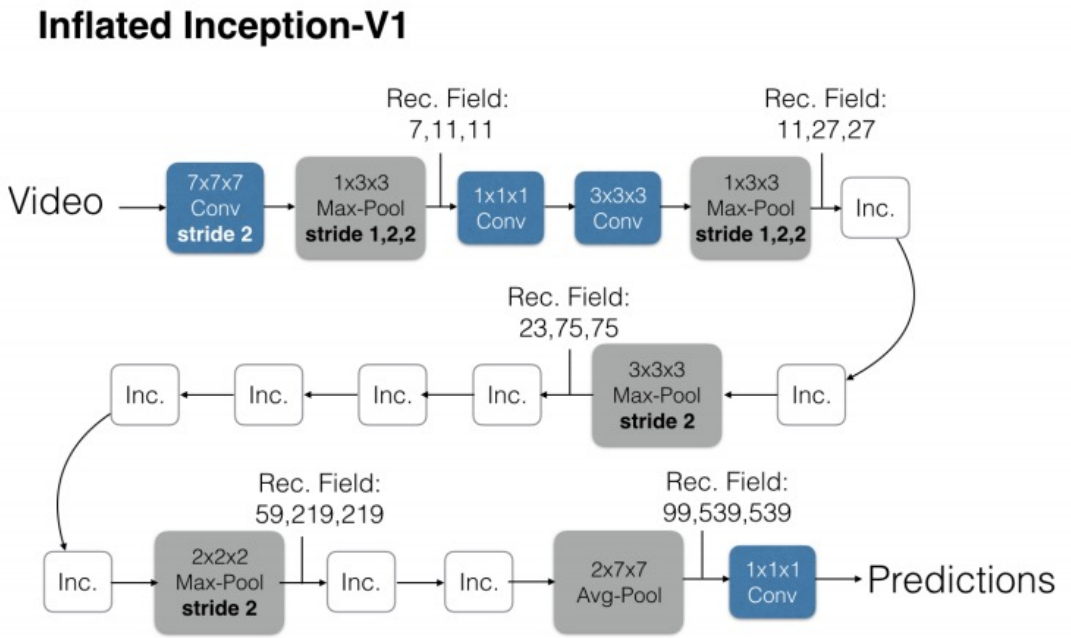
Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset
https://arxiv.org/abs/1705.07750

Introduction

- Imagenet 으로 이미지 분류로 알게된 사실 → 굉장히 큰 데이터셋으로 pre-training하면 다른 도메인의 문제로도 확대 적용하는데 굉장히 많은 도움 된다
- 네트워크 구조 바뀌어도 도움된다
- 큰 비디오 데이터셋이 있다면 → pre-training 으로 퍼포먼스 높일 수 있지 않을까
- kinetics라는 큰 비디오 데이터셋 만듦
  - SOTA 아키텍쳐 다시 구현
  - HMDB-51, UCF-101인 작은 데이터셋으로 pre-training
- I3D 모델 제안

논문 분석

e. I3D



저자가 제안하는 방법은 2D ConvNet의 ImageNet 데이터셋으로 classification 문제를 푼 모델을 이용해 문제를 푸는 방식

Inflating 2D ConvNets into 3D

ImageNet 데이터를 이용해 2D classification 문제에 대해 학습시킨 2D ConvNet을 3D ConvNet으로 바꾸는 방법이다.

저자는 이러한 NxN filters를 NxNxN filter로 바꾸는 방식을 inflating이라고 불렀습니다.

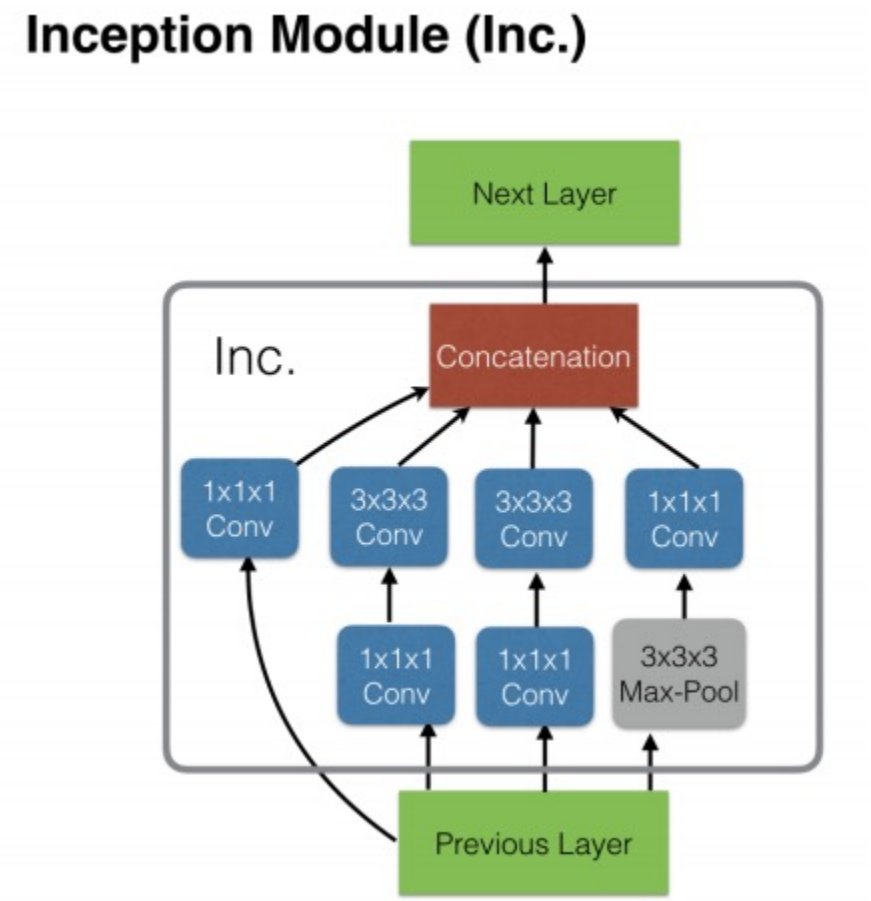
디테일한 방식으로는 필터의 dimension을 늘려준 뒤, weight를 1/N로 나눠준다.

Optical flow 정보를 넣어줌으로써 motion 정보에 대해 더 잘 예측할 수 있도록 하였다.

두개의 I3D 모델의 각각 입력을 RGB, optical flow를 넣어주어 학습을 시키고, prediction의 평균값을 취해 최종 prediction을 했다.

논문 분석

e. I3D



Training Detail :

- ImageNet pretrained Inception-V1 사용
- 모든 Conv layer에 BN(Batch Norm),ReLU 사용
- SGD+momentum 0.9 사용
- 입력 영상은 256pixel로 resize 후 랜덤하게 224 pixel로 cropping을 함
- Random하게 left-right flipping 진행
- 입력 영상은 25fps이며 총 길이는 64 frame

논문 분석_*Results*

Method	#Params	Training		Testing	
		# Input Frames	Temporal Footprint	# Input Frames	Temporal Footprint
ConvNet+LSTM	9M	25 rgb	5s	50 rgb	10s
3D-ConvNet	79M	16 rgb	0.64s	240 rgb	9.6s
Two-Stream	12M	1 rgb, 10 flow	0.4s	25 rgb, 250 flow	10s
3D-Fused	39M	5 rgb, 50 flow	2s	25 rgb, 250 flow	10s
Two-Stream I3D	25M	64 rgb, 64 flow	2.56s	250 rgb, 250 flow	10s

Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

논문의 test에 따르면 RGB 하나만 넣어준 경우보단 **RGB+Flow**인 경우에서 항상 더 좋은 성능이 나왔습니다.

논문 분석_*Results*

Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

Scartch로 부터 학습한 경우와 Image pretreained를 사용한 모델의 성능차를 보여줌으로써 ImageNet Pretrained 된 모델의 사용이 당위성을 보여줌
괄호 안의 숫자는 top5-accuracy, 나머지는 모두 top-1

논문 분석_*Results*

Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

ImageNet + Kinetics + UCF-101

Kinetics + UCF-101

ImageNet + Kinetics + HMDB-51

Kinetics + HMDB-51

모든 아키텍처가 Kinetics의 추가 비디오 데이터에 대한 사전 교육을 통해 이점을 얻지만 일부는 특히 I3D-ConvNet 및 3D-ConvNet(후자는 훨씬 낮은 기초부터 시작). Kinetics(Fixed)에서 사전 훈련한 후 모델의 마지막 계층만 훈련하면 I3D 모델에 대해 UCF-101 및 HMDB-51에서 직접 훈련하는 것보다 훨씬 더 나은 성능을 얻을 수 있습니다.

논문 분석_*Results*

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

UCF-101 및 HMDB-51에 대한 I3D 모델의 성능과 이전의 최첨단 방법 비교. Kinetics 데이터 세트에서 사전 교육할 때 결과를 포함합니다.(ImageNet 사전 교육 포함 및 제외)

데이터세트에서 가장 성능이 좋은 방법 : RGB 및 광학 흐름 스트림에서 ResNet-50 모델을 사용하고 UCF-101에서 94.6%, 70.3%를 얻는 방법

RGB-I3D 또는 RGB-Flow 모델 중 하나만 Kinetics에서 pre-trained을 하면 모든 모델 또는 모델 조합에서 이전에 게시된 모든 성능을 능가합니다. 우리의 결합된 Two-Stream아키텍처는 이전 모델에 비해 이점을 상당히 넓혀 UCF-101에서 98.0, HMDB-51에서 80.9로 전체 성능을 가져옵니다.

Kinetics 사전 훈련된 I3D 모델과 이전 3D ConvNets(C3D) 사이의 차이는 훨씬 더 크지만 C3D는 더 많은 비디오, Sports1M의 1M 예제와 내부 데이터 세트, 그리고 IDT와 앙상블되고 결합된 경우에도 훈련됩니다. 이는 Kinetics의 더 나은 품질로 설명할 수 있지만 단순히 I3D가 더 나은 아키텍처이기 때문에 설명할 수도 있습니다.

논문 분석_결론

많은 작업에 대해 ImageNet에서 ConvNet을 pre-training하는 데 그러한 이점이 있었던 것처럼 (큰 비디오 데이터 세트) **Kinetics에 대한 pre-training에 상당한 이점**이 있다는 것은 분명합니다. 이것은 유사한 작업에 대해 하나의 데이터 세트(Kinetics)에서 다른 데이터 세트(UCF-101/HMDB-51)로 학습을 전이하는 것을 보여줍니다(동작 클래스는 다르지만). 그러나 Semantic Video Segmentation, Video Object Detection, or Optical Flow Computation과 같은 다른 비디오 작업에 Kinetics 사전 훈련을 사용하는 이점이 있는지 여부는 여전히 확인해야 합니다.

이 모델을 선정한 사유

Action recognition에 관한 논문을 찾아보다가 **“Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”**을 읽어보게 되었다.
해당 논문은 기존의 방법들을 소개해주고, 새로운 Kinetics Human Action Video 데이터 세트에 비추어 최첨단 아키텍처를 재평가한다.
2D ConvNet 인플레이션을 기반으로 하는 새로운 Two-Stream Inflated 3D ConvNet(I3D)을 사용하여 Kinetics에 대한 사전 교육 후 I3D 모델이 최신 동작 분류를 상당히 개선하여 HMDB-51에서 80.9%, UCF-101에서 98.0%에 도달했음을 보여준다.
98.0% 라는 높은 결과값에 반해 이 I3D 모델을 선정하게 되었다.

하드웨어 성능

```
!nvidia-smi
```

```
Sat Nov 26 06:14:08 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   62C    P0    27W /  70W |   9080MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

```
[33] !python --version
```

```
Python 3.7.15
```

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

```
[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 8390439943574947052
 xla_global_id: -1, name: "/device:GPU:0"
 device_type: "GPU"
 memory_limit: 14415560704
 locality {
   bus_id: 1
   links {
   }
 }
 incarnation: 12211492864148384895
 physical_device_desc: "device: 0, name: Tesla T4, pci bus id: 0000:00:04.0, compute capability: 7.5"
 xla_global_id: 416903419]
```

DataSet

UCF-101

이 오픈소스는 UCF-101 dataset을 사용

UCF101은 YouTube에서 수집한 사실적인 동작 동영상의 동작 인식 데이터 세트로 101개의 동작 범주가 있습니다. 이 데이터 세트는 50개의 조치 범주가 있는 UCF50 데이터 세트의 확장입니다.

101개 동작 카테고리의 13320개 비디오를 통해 UCF101은 동작 측면에서 가장 큰 다양성을 제공하며 카메라 움직임, 개체 모양 및 포즈, 개체 크기, 관점, 어수선한 배경, 조명 조건 등의 큰 변화가 있는 가장 큰 다양성을 제공합니다. 현재까지 설정된 도전적인 데이터. 사용 가능한 동작 인식 데이터 세트의 대부분은 사실적이지 않고 배우가 연출하기 때문에 UCF101은 새로운 현실적인 동작 범주를 학습하고 탐색하여 동작 인식에 대한 추가 연구를 장려하는 것을 목표로 합니다.

- Human-Object Interaction
- Body-Motion Only
- Human-Human Interaction
- Playing Musical Instruments
- Sports

테스트 결과

https://upload.wikimedia.org/wikipedia/commons/9/9d/Jumping_with_bicycle_Novgorod_2018.ogv

프레임을 50으로 했을 때에는 결과가 riding unicycle : 54.9% , riding a bike : 28.85% 이지만
프레임을 200으로 했을 때에는 결과가 riding a bike : 56.71%, riding unicycle : 37.91% 이다.
동영상의 길이가 길수록 더 정확한 결과를 도출해내는 것을 확인할 수 있다.

```
[27] sample_video = load_video(video_path)[:50]
     sample_video.shape

     (50, 224, 224, 3)
```

```
[28] to_gif(sample_video)
```

```
predict(sample_video)

Top 5 actions:
  riding unicycle       : 54.90%
  riding a bike         : 28.85%
  faceplanting          : 11.51%
  jogging               :  1.57%
  riding scooter        :  0.79%
```

```
[24] sample_video = load_video(video_path)[:100]
     sample_video.shape

     (100, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)

Top 5 actions:
  riding unicycle       : 48.35%
  riding a bike         : 45.21%
  faceplanting          :  3.58%
  riding mountain bike  :  1.27%
  jogging               :  0.72%
```

```
[21] sample_video = load_video(video_path)[:200]
     sample_video.shape

     (200, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)

Top 5 actions:
  riding a bike         : 56.71%
  riding unicycle       : 37.91%
  faceplanting          :  3.47%
  riding mountain bike  :  1.56%
  biking through snow   :  0.18%
```

테스트 결과

https://upload.wikimedia.org/wikipedia/commons/9/98/RSA-EP-Swimming-video-001.ogv

```
[19] video_path = "RSA-EP-Swimming-video-001.ogv"
```

```
[20] sample_video = load_video(video_path)[:100]
     sample_video.shape
```

```
(100, 224, 224, 3)
```

```
[21] to_gif(sample_video)
```

자유형하는 영상을 들고왔는데
예측한 결과로는
Top 1 : 수영 접영
Top 2 : 수영 평영
이라는 결과가 나왔다.

```
predict(sample_video)
```

```
Top 5 actions:
  swimming butterfly stroke: 56.51%
  swimming breast stroke: 11.49%
  feeding fish           :  8.28%
  swimming backstroke    :  5.66%
  somersaulting          :  2.82%
```

```
[19] video_path = "RSA-EP-Swimming-video-001.ogv"
```

```
[23] sample_video = load_video(video_path)[:200]
     sample_video.shape
```

```
(200, 224, 224, 3)
```

```
to_gif(sample_video)
```

영상의 길이를 늘렸더니
결과가 더 분산돼서 나왔다.

이럴때에는 어떻게 해야할까?
-> 수영에 대한 학습을 더 해야할 것 같다

```
[25] predict(sample_video)
```

```
Top 5 actions:
  swimming butterfly stroke: 28.75%
  feeding fish           : 19.09%
  swimming backstroke    : 12.54%
  swimming breast stroke: 10.60%
  snorkeling             :  6.55%
```

모델 추가 학습

Kinetics-400 I3D -> Kinetics-600 I3D

본래에는 Kinetics-400을 이용한 I3D 였지만, Kinetics-600으로 바꿔주니,
Top 1 : ice swimming :80.28%가 나왔다.

-> Kinetics-600 에는 자유형(수영)에 관련된 학습이 없었다.
그러므로 이 예측 값이 최선이라는 것을 알았다.

```
to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  ice swimming           : 80.28%
  swimming front crawl   :  5.17%
  capsizing              :  3.63%
  diving cliff           :  2.71%
  swimming butterfly stroke:  2.51%
```

모델 추가 학습

Kinetics-400 I3D -> Kinetics-600 I3D

본래에는 Kinetics-400을 이용한 I3D 였지만, Kinetics-600으로 바꿔주니,
Top 1 : jumping bicycle 이 나왔다.

영상을 보시면 아시겠지만, 더욱더 정확한 값이라고 할 수 있다.

```
[25] to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  jumping bicycle       : 49.00%
  falling off bike      : 42.09%
  riding a bike         :  5.96%
  riding unicycle       :  2.82%
  biking through snow   :  0.05%
```

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/7/70/Horse_race_in_Vienna.webm

```
[32] video_path = "Horse_race_in_Vienna.webm"
```

```
[33] sample_video = load_video(video_path)[:200]
     sample_video.shape
```

```
(200, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  riding mule                 : 38.32%
  riding or walking with horse: 30.63%
  jaywalking                  : 13.21%
  playing polo                :  4.48%
  jogging                     :  3.65%
```

Horse race 에 관련된 영상을 넣어보았다.
피사체가 멀리서 부터 가까워지는 영상인데,
예측 값은 조금 분산되었지만, 정확한 결과 값이라고 할 수 있다.

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/0/0e/Nationale_Military_met_cross_en_steeplechase-525023.ogv

```
[37] video_path = "Nationale_Military_met_cross_en_steeplechase-525023.ogv"

[41] sample_video = load_video(video_path)[:100]
     sample_video.shape

     (100, 224, 224, 3)

     to_gif(sample_video)
```

다른 horse race 영상을 넣어봤다.
동영상의 특성상 짧은 길이로는 결과를 예측하기에 어려움이 있는 것 같다.
100프레임으로 데이터를 분석하니 엉뚱한 예측 값이 나왔지만,
200프레임으로 분석하니 정확한 예측 값이 나왔다.

```
     predict(sample_video)

     Top 5 actions:
       javelin throw          : 20.30%
       shooting goal (soccer): 12.91%
       riding or walking with horse:  9.71%
       flying kite            :  6.39%
       throwing discus        :  4.86%
```

```
[37] video_path = "Nationale_Military_met_cross_en_steeplechase-525023.ogv"

[38] sample_video = load_video(video_path)[:200]
     sample_video.shape

     (200, 224, 224, 3)

     to_gif(sample_video)
```

```
     predict(sample_video)

     Top 5 actions:
       riding or walking with horse: 42.08%
       javelin throw          :  9.96%
       riding mule            :  6.44%
       playing polo           :  5.79%
       flying kite            :  3.94%
```

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/1/1e/Banzai_Skydiving.webm

```
[10] video_path = "Banzai_Skydiving.webm"
```

```
[11] sample_video = load_video(video_path)[:100]
     sample_video.shape
```

```
(100, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  dancing charleston    : 13.51%
  playing chess         :  9.55%
  finger snapping       :  8.54%
  shaking hands         :  8.48%
  robot dancing         :  7.64%
```

```
[10] video_path = "Banzai_Skydiving.webm"
```

```
[14] sample_video = load_video(video_path)[:200]
     sample_video.shape
```

```
(200, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  skydiving             : 11.68%
  base jumping          :  9.95%
  playing chess         :  7.04%
  dancing ballet        :  5.10%
  flying kite           :  3.06%
```

Skydiving 영상을 넣어보았다.
이번에는 잘 예측하였지만, 예측률이 너무 낮다.

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/9/9b/9_Month_Milestone-_Crawls.webm

```
[9]  video_path = "9_Month_Milestone-_Crawls.webm"
```

```
[10] sample_video = load_video(video_path)[:100]
     sample_video.shape
```

```
(100, 224, 224, 3)
```

```
to_gif(sample_video)
```


```
predict(sample_video)
```

```
Top 5 actions:
  crawling baby          : 52.57%
  mountain climber (exercise): 23.76%
  yoga                   :  8.78%
  installing carpet      :  4.94%
  stretching leg         :  3.08%
```

피사체가 매우 가까운 영상을 넣어보았다.
예측 값이 잘 나오긴 했지만, top2의 예측 값도 수치가 작은편은 아니여서 모델의 학습이 더 필요하다고 생각된다.

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/6/6d/Reves_dos_manos_con_pelota.ogv

```
[10] video_path = "Reves_dos_manos_con_pelota.ogv"
```

```
[11] sample_video = load_video(video_path)[:100]
     sample_video.shape
```

```
(100, 224, 224, 3)
```

```
to_gif(sample_video)
```

Tennis 하는 영상을 넣어보았다.
피사체가 가깝고, 불필요한 동작을 하지 않아서 100프레임으로 실행시켜보았다.
이때까지의 데이터 분석 중 가장 만족스러운 예측 값이 나온것 같다.

```
predict(sample_video)
```

```
Top 5 actions:
  playing tennis        : 99.72%
  catching or throwing softball:  0.18%
  playing badminton     :  0.08%
  throwing ball (not baseball or American football):  0.01%
  catching or throwing baseball:  0.01%
```

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/9/97/Tanzanian_women_competing_pushups.ogv

```
[23] video_path = "Tanzanian_women_competing_pushups.ogv"
```

```
[27] sample_video = load_video(video_path)[:250]
     sample_video.shape
```

```
(250, 224, 224, 3)
```

```
to_gif(sample_video)
```

여성 두 명에서 pushup 하는 영상이다.
정석 pushup자세가 아니여서 그런지 제대로 된 예측 값을 내지 못하였지만
Top 2 에 Push Up을 예측했다.

```
predict(sample_video)
```

```
Top 5 actions:
  yoga                 : 70.73%
  push up              : 16.24%
  building sandcastle  :  3.48%
  stretching arm       :  2.03%
  contorting           :  1.36%
```

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/6/6a/Отжимания_на_тренажере_X1-PRO.ogv

```
[37] video_path = "%D0%9E%D1%82%D0%B6%D0%B8%D0%BC%D0%B0%D0%BD%D0%B8%D1%
```

```
[38] sample_video = load_video(video_path)[:100]
     sample_video.shape
```

```
(100, 224, 224, 3)
```

```
to_gif(sample_video)
```

```
predict(sample_video)
```

```
Top 5 actions:
  push up                : 99.97%
  lunge                  :  0.01%
  mountain climber (exercise):  0.01%
  yoga                   :  0.00%
  dancing ballet         :  0.00%
```

정면에서 정석 pushup자세를 하는 영상을 가져와봤다.
정확한 예측 값을 도출하였다.

데이터 분석

https://upload.wikimedia.org/wikipedia/commons/9/98/Interval_Push-ups.webm

```
[42] video_path = "Interval_Push-ups.webm"

[51] sample_video = load_video(video_path)[:200]
     sample_video.shape

     (200, 224, 224, 3)

     to_gif(sample_video)
```

```
     predict(sample_video)

     Top 5 actions:
       yoga                 : 68.07%
       push up              : 21.03%
       stretching leg       :  4.10%
       bending back         :  1.17%
       mountain climber (exercise):  1.00%
```

```
[42] video_path = "Interval_Push-ups.webm"

[48] sample_video = load_video(video_path)[:300]
     sample_video.shape

     (300, 224, 224, 3)

     to_gif(sample_video)
```

```
     predict(sample_video)

     Top 5 actions:
       push up              : 48.17%
       yoga                 : 36.44%
       stretching leg       :  6.70%
       mountain climber (exercise):  2.03%
       lunge                :  0.92%
```

이번에는 측면 정석 pushup영상을 넣어보았다.
200프레임까지는 yoga를 예측하여서
300프레임으로 출력하니 pushup을 예측하였다.
측면 영상보다 정면 영상을 더 정확하게 예측하는 것 같다