Abstract

이 연구는 대규모 감독된 비디오 데이터셋에서 훈련된 깊은 3차원 합성곱 네트워크(3D ConvNets)를 사용하여 시공간 특징 학습을 위한 간단하면서도 효과적인 접근 방식을 제시한다.

연구 결과는 다음 3가지 주요 포인트로 요약된다.

1. 3D ConvNets는 시공간 특징 학습에 있어 2D ConvNets보다 더 적합하다는 것을 발견했다. 이는 3D ConvNets가 비디오와 같은 시간적 요소를 포함하는 데이터의 공간적 및 시간적 패턴을 더 잘 포착할 수 있음을 의미한다.

2. 모든 층에서 작은 3×3×3 convolution kernel을 사용하는 구조가 3D ConvNets에 대해 가장 높은 성능을 보이는 구조 중 하나로 나타났다. 이는 복잡성을 줄이면서도 효과적인 특징 학습을 가능하게 한다.

3. C3D는 3D Convolution이며, C3D로 학습된 feature는 단순한 선형 분류기(SVM)를 사용함에도 불구하고 4개의 다른 벤치마크에서 최신 방법들을 능가하는 성능을 보였으며, 나머지 2개 벤치마크에서는 현재 최고의 방법들과 비교할 만한 수준의 성능을 보였다. 또한, 이 feature들은 매우 compact(간결)하며(UCF101 데이터셋에서 단 10개의 차원으로 52.8% 정확도 달성) ConvNets의 빠른 추론 덕분에 계산도 매우 효율적이다.

즉, 이 연구는 비디오 분석 및 관련 분야에서 3D ConvNets의 이점과 효율성을 강조하며, 시공간 데이터 처리를 위한 새로운 방향을 제시한다.

◎ 3D ConvNets와 C3D(Convolution 3D)의 차이점

3D ConvNets는 비디오 데이터에서 시공간 feature를 학습하기 위한 일반적인 아키텍처의 범주이다. C3D는 3D ConvNets의 한 구현으로, 고정된 크기의 kernel을 사용하는 특정 아키텍처이다. 즉, C3D는 3D ConvNets의 한 예시 혹은 구현이다.

◎ C3D가 더욱 복잡한 분류기를 사용한다면 성능이 더 올라갈까?

DNN 기반 분류기, RNN, LSTM 등 SVM 보다 복잡한 분류기를 사용한다면 성능이 올라간다.

LSTM과 같은 순환 신경망을 사용함으로써, C3D가 추출한 특징들 사이의 시간적 관계를 보다 정확하게 모델링할 수 있다. 그리고 단순히 행동을 분류하는 것뿐만 아니라, 행동의 순서, 지속 시간, 그리고 행동 사이의 상호 작용과 같은 더 복잡한 문제를 해결할 수 있다.

하지만, 더욱 복잡한 분류기를 쓴다면 계산 비용의 증가와 과적합의 위험 같은 단점이 따른다.

Introduction

효과적인 비디오 discriptor는 다음 4가지 특성이 따른다.

1. Generic
   다양한 유형의 비디오를 잘 나태내고, 구별할 수 있어야 한다.

2. Compact
   방대한 양의 비디오를 다루므로 간결한 기술자는 처리, 저장, 작업을 보다 확장 가능하게 한다.

3. Efficient
   실제 설계 시스템에서 매분 수천개의 비디오가 처리되기에 계산하기 효율적이어야 한다.

4. Simple
   복잡한 인코딩과 분류기를 사용하는 것보다 단순한 모델(선형 분류기)을 사용하여도 잘 작동해야 한다.

이미지 기반의 깊은 feature들은 동작 모델링의 부족으로 비디오에 적합하지 않다. 따라서 깊은 3D ConvNets를 사용하여 시공간 feature를 학습한다.

이렇게 학습된 feature들은 단순한 선형 분류기와 함계 사용될 때 좋은 성능을 냈다. 3D ConvNets에서 나오는 feature들은 비디오에서 객체, 장면, 행동과 관련된 정보를 포함하며, 미세 조정이 필요없이 다양한 작업에 유용하다.

다음 3가지는 이 연구에서 발견한 내용이다.

1. 우리는 3D convolution deep net-works가 외관과 동작을 모델링하는 좋은 특징 학습 기계임을 실험적으로 보여줍니다.

2. 모든 레이어에서 3x3x3 kernel이 제한된 세트의 탐색된 아키텍처 중에서 가장 잘 작동한다.

3. 그렇게 나온 feature들은 4가지 다른 작업과 6가지 벤치마크에서 다른 최고성능의 방법들을 능가하거나 접근하는 성능을 보여주며, 간결하며, 계산하기 효율적이다.

Dataset Task	Sport1M action recognition	UCF101 action recognition	ASLAN action similarity labeling	YUPENN scene classification	UMD scene classification	Object object recognition
Method	[29]	[39]([25])	[31]	[9]	[9]	[32]
Result	**90.8**	75.8 (89.1)	68.7	96.2	77.7	12.0
C3D	85.2	**85.2 (90.4)**	**78.3**	**98.1**	**87.7**	**22.3**

Table 1. **C3D compared to best published results**. C3D outperforms all previous best reported methods on a range of benchmarks except for Sports-1M and UCF101. On UCF101, we report accuracy for two groups of methods. The first set of methods use only RGB frame inputs while the second set of methods (in parentheses) use all possible features (e.g. optical flow, improved Dense Trajectory).

Table1은 다양한 데이터 셋에서 C3D모델과 이전에 발표된 최고성능의 방법들을 비교한 결과이다.

Sport1M과 UCF101 데이터셋에 대해서 행동 인식에 대한 작업을 진행하였으며, 벤치마크가 Sport1M의 최고성능 90.8보다 높진 않으나, 그에 준하는 점수를 냈다.

UCF101의 경우 두 가지 방법의 결과를 나타내었는데, 첫 번째(괄호가 없는)는 오직 RGB프레임 입력만 사용하였고, 두 번째(괄호가 있는)는 가능한 모든 feature(optical flow, iDT(improved dense trajectory))을 사용했다.

◎ Optical flow : 비디오 내의 객체나 피사체가 움직임을 감지하는 기술이다. 움직이는 객체의 속도와 방향을 계산하여 객체가 시간에 따라 어떻게 움직이는지 이해할 수 있다.

◎ iDT(improved Dense Trajectory 개선된 밀집 궤적) : optical flow 정보를 포함하여 여러 가지 추가 정보(피사체의 모양 변화나 카메라의 움직임)를 사용함으로써 움직임을 보다 정확하게 추적할 수 있다.

ASLAN은 행동 유사성 라벨링 작업에 사용되며, 비디오 쌍이 같은 동작을 포함하는지 여부를 결정한다.

YUPENN, UMD 데이터셋들은 장면 분류 작업에 사용됩니다. 주어진 이미지 또는 비디오가 어떤 장면 유형(도시, 자연)에 속하는지 분류한다.

Object는 객체 인식 작업에 사용되며, 이미지 내 특정 객체를 식별하고 분류하는 것을 목표로 한다.

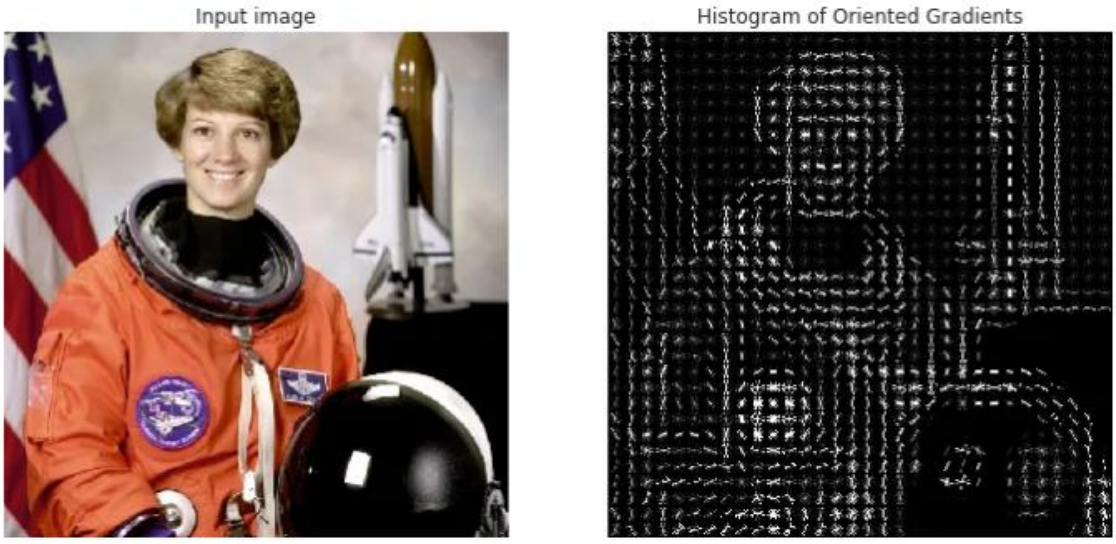
Related Work

CV분야에서 비디오는 행동 인식, 이상 탐지, 비디오 검색, 이벤트 및 액션 탐지 등 다양한 방법들이 비디오 표현에 제안되었다.

다음은 다양한 학자들이 비디오 표현 연구에 관한 내용이다.

1. Laptev, Linde-berg -> 3D 해리스 코너 컴출기를 확장하여 STIPs(spatio-temporal interest points 공간-시간 관심 지점)를 제안했으며, SIFT와 HOG(histogram of oriented gradients 방향 그래디언트 히스토그램)은 행동 인식을 위해 각각 SIFT-3D와 HOG-3D로 확장되었다.

◎ HOG

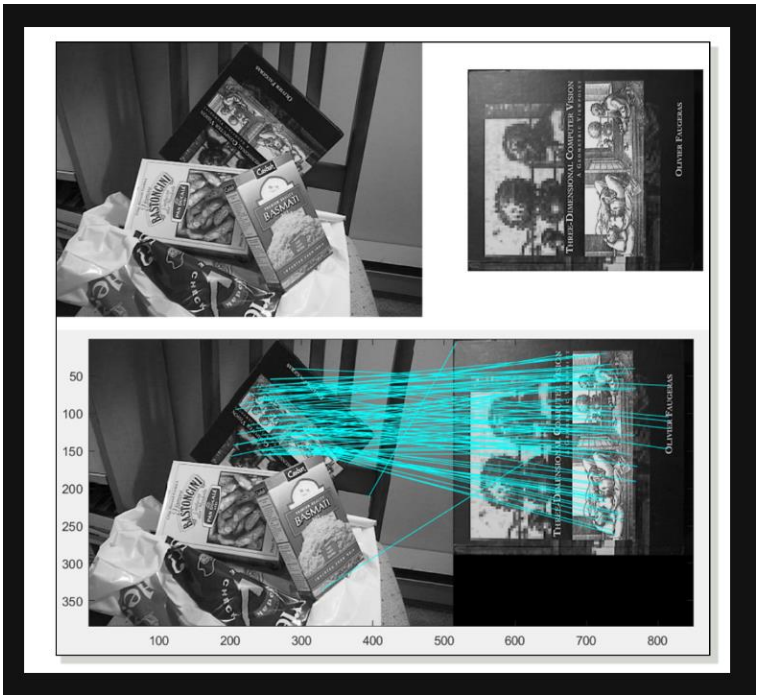


Credit: iq.opengenus.org

이미지 내의 지역적인 객체 형태와 구조를 기술하기 위해 방향 그래디언트의 분포를 사용하는 방법이다. 각 지역 블록 내에서 이미지의 그래디언트 방향을 히스토그램으로 나타냄으로써, 해당 블록의 형태적 특성을 기술한다.

HOG는 주로 정지 이미지에서 객체 인식에 사용되었으나, HOG-3D로 확장되면서 비디오 내의 시간적 변화를 포함한 3차원 데이터 분석에도 적용될 수 있게 되었다. HOG-3D는 공간적 그래디언트 뿐만 아니라 시간적 그래디언트 정보도 포함하여, 비디오 내에서 움직임과 행동을 더 잘 기술할 수 있게 된다.

◎ SIFT



scale-invariant feature transform은 David Lowe에 의해 개발된 컴퓨터 비전에서 사용되는 알고리즘 중 하나로, 이미지 내에서 특징점(Feature Points)을 추출하고 기술하는 데 사용된다. 위와 같이 SIFT 알고리즘을 통해 추출한 feature들을 매칭키셔서 두 image에서 서로 대응되는 부분을 찾을 수 있다.
SIFT는 특히 이미지의 스케일과 회전에 불변하는 특성을 가지며, 뷰포인트 변화나 조명 변화에도 강인한 특징을 추출할 수 있는 것이 특징이다. 이러한 특성 때문에 SIFT는 이미지 매칭, 패턴 인식, 객체 인식 등 다양한 컴퓨터 비전 분야에서 널리 활용된다.
3차원 데이터를 다루기 위해, SIFT 기술을 3차원으로 확장한 것이 SIFT-3D이다.

2. Dollar -> 행동 인식을 위해 Cuboids feature를 제안했다.
   ◎ cuboids feature : 이 feature는 비디오에서 특정 시간 동안 발생하는 공간적 변화와 움직임을 3차원 형태로 나타내는데,
   여기서 'Cuboid'는 직육면체 형태의 시공간 블록을 의미한다. 이러한 시공간 블록은 비디오 시퀀스 내에서 움직임이나 액션의 중요한 정보를 포함하고 있다.

3. Sadanand, Corso -> 행동 인식을 위해 ActionBank를 구축했다.
   ◎ action bank : 비디오를 분석할 때 단일 action 검출기에 의존하는 대신, 매우 다양한 액션 검출기들의 응답을 종합함으로써 풍부하고 다양한 액션 정보를 포착하는 것이다.
   이 방법은 단일 액션 또는 간단한 움직임 패턴을 넘어서, 비디오 내에서 복잡하고 다양한 행동들을 인식하는 데 특히 유용하다.

4. Wang -> 최근 수작업으로 만든 feature인 iDT(improved Dense Trajectory 개선된 밀집 궤적)를 제안했다. iDT는 시간 신호를 공간 신호와 다르게 처리할 수 있다. 이 방법은 3D 해리스 코너 검출기를 확정하여 비디오 프레임에서 밀집 샘플링된 feature point를 시작으로 optical flow를 사용하여 그 feature point를 추적한다.
   ◎ '밀집 샘플링된 feature point를 시작'이란, 비디오 내 다양한 장명에서 대표적으로 움직임이 관측될 수 있는 점들을 많이 선택하여 시작점으로 삼는 다는 것이다. iDT는 성능이 좋지만, 계산이 매우 복잡하여 대규모 데이터셋에서는 처리가 불가능하다는 단점이 있다

GPU의 발전으로 인해 강력한 병렬처리 가능해지고, 대량의 학습 데이터셋의 사용성이 높아짐에 따라 CNN이 시각 인식 분야에서 발전을 이루고 있다. CNN은 이미지와 비디오에서 인간 자세 추정에도 적용되었다.

Zhou의 연구에서 이러한 이미지 feature 학습이 사용된 깊은 네트워크가 전이 학습에서 좋은 성능을 보이며, 비디오 학습에도 비지도 학습이 적용되었다.

Le의 연구에서 stacked ISA를 사용하여 비디오의 시공간 feature를 학습하였다. 행동 인식에서 좋은 결과를 보였지만, 계산량이 많아서 대규모 데이터셋에는 적용이 어렵다는 단점이 있다.

◎ Stacked ISA : 비디오나 이미지와 같은 시각적 데이터에서 시공간 feature를 학습하기 위한 방법 중 하나이다.

ISA (Independent Subspace Analysis)는 데이터 내의 숨겨진 통계적으로 독립적인 부분 공간을 찾아내는 데 목적을 두고 있으며, 이를 통해 복잡한 구조를 가진 데이터를 더 잘 이해하고 분석할 수 있다.

Stacked ISA는 이러한 ISA를 여러 층(layer)으로 쌓아 올린 구조를 의미한다. 각 층은 데이터의 다양한 추상화 수준에서 특징을 추출하며, 이를 통해 더욱 복합적이고 고차원적인 특징을 학습할 수 있다.

또한 CNN은 인간 동작 인식과 의료 이미지 분할에도 사용되며, 시공간 feature 학습을 위해 Restricted Boltzmann Machines(제한된 볼츠만 기계)와 함께 사용되었다.

◎ Restricted Boltzmann Machines : 딥러닝과 기계학습 분야에서 사용되는 확률적 그래픽 모델이다. RBM으로 불리며, 두 그룹의 뉴런으로 구성되어 있다. 하나는 보이는(가시적) 뉴런이고, 다른 하나는 숨겨진 뉴런이다.

이 두 그룹의 뉴런은 서로 연결되어 있지만, 같은 그룹 내의 뉴런끼리는 연결되어 있지 않다. 이러한 구조적 제약 때문에 "제한된(Restricted)"이라는 이름이 붙었다.

최근 Karpathy가 대규모 비디오 데이터셋에 깊은 네트워크를 학습시켜 비디오 분류 작업에 적용했다.

Simonyan과 Zisserman은 2개의 stream network를 사용하여 동작 인식에서 최고의 결과를 달성했다.

◎ 2개의 Stream network : 비디오 내의 동작을 인식하기 위해 설계되었다. 이 모델은 크게 두 가지 병렬 구조로 구성된다. 하나는 공간적 특징을 처리하는 "공간 스트림(Spatial Stream)"이고, 다른 하나는 시간적 특징을 처리하는 "시간 스트림(Temporal Stream)"이다.

이 두 스트림은 병렬로 운영되며 각각의 스트림이 추출한 특징을 결합하여 최종적인 동작 인식 결과를 도출한다.

이러한 접근방식들 중에서 3D ConvNets 접근 방식은 우리와 밀접하게 관련되어 있다. 예를 들어 human detector와 head tracking을 통해 비디오에서 인간 대상을 분할한다. 분할된 비디오에서 3개의 합성곱 레이어를 가진 3D 합성곱 신경망에 입력되어 동작을 분류한다.

반면, 이 논문의 방법은 전체 비디오 프레임을 입력으로 사용하며, 어떠한 전처리에 의존하지 않아서 대규모 데이터셋으로 쉽게 확장이 가능하다.

이렇게 프레임을 사용한다는 점에서 위의 Karpathy와 Simonyan과 Zisserman의 연구와 유사성을 띄지만, 2D 합성곱과 2D 풀링 연상만을 사용하여 구축하였다는 점에서 차이점이 있다.

반면 이 논문에서는 모든 레이어에 시간 정보를 전파하는 3D 합성곱과 3D 풀링 연산을 수행한다.

또한, 공간과 시간 정보를 점진적으로 풀링하고 더 깊은 네트워크를 구축함으로써 최고의 결과를 달성한다.

3D convolution and pooling

이제 3D ConvNets의 기본 작동원리를 설명한다.

3D ConvNets에서는 컨볼루션과 풀링 연산이 시공간적으로 수행되는 반면, 2D ConvNets에서는 오직 공간적으로만 수행된다.

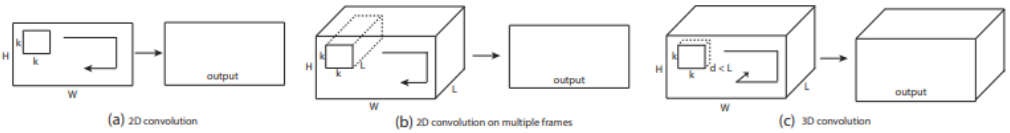
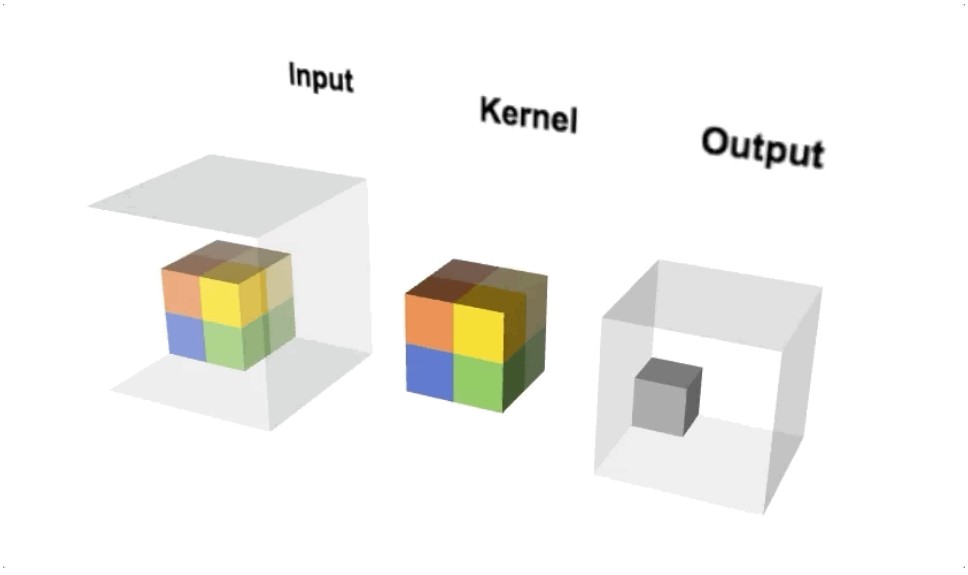


Figure 1. **2D and 3D convolution operations**. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

Figure1이 이 차이를 잘 보여주는데, a는 이미지에 2D 합성곱을 적용하여 이미지를 결과로 내놓는다.

b는 비디오에 2D 합성곱을 적용하여 이미지를 결과로 내놓는다. 이때 비디오의 다수의 프레임을 다수의 채널로 적용하였다.



c는 비디오에 3D 합성곱을 적용하여 또다른 볼륨을 결과로 내놓는다. 이때 입력 신호의 시간적 정보가 보존된다.

2D ConvNets는 이미지를 결과로 하며, 합성곱 연산 후 바로 입력 신호의 시간적 정보를 잃어버리지만, 3D 합성곱이 입력 신호의 시간적 정보를 보존하여 출력 볼륨을 생성한다. 이러한 현상은 풀링에도 적용이 된다.

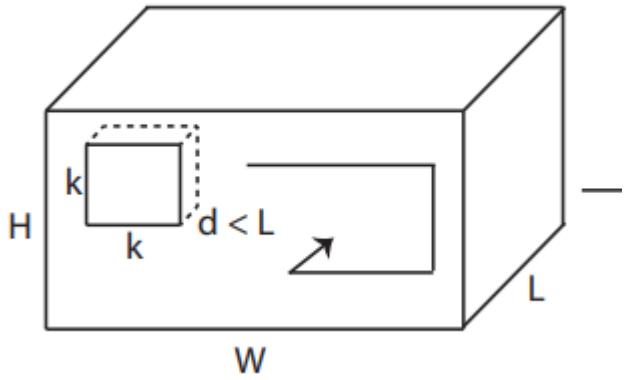
예를 들어, 걷기와 달리기 행동을 인식하려고 할 때, 2D ConvNets에서는 각 프레임을 독립적인 이미지로 처리해서 시간적 정보를 잃어서 걷다가 달리는 행동의 변화를 인식하기 힘들지만, 3D ConvNets를 사용하면 시간적 정보가 보존되므로 행동 변화를 정확하게 포착한다.

2D 합성곱에서 시간적 stream network가 여러 프레임을 입력으로 받지만, 첫번째 합성곱 레이어 이후에 시간적 정보가 소멸된다.

Slow fushion 모델은 첫 3개의 합성곱 레이어에서 3D 합성곱과 평균 풀링을 사용하여 좋은 성능을 냈지만, 3번째 합성곱 이후에 모든 시간적 정보를 잃어버렸다.

대규모 비디오 데이터셋에서 깊은 네트워크를 훈련하는 것은 매우 시간이 많이 소요되기 때문에, 우선 중간 규모 데이터셋인 UCF101을 실험하여 최적의 구조를 찾는다. 그리고 합성곱 커널을 3x3으로 고정하고 시간적 깊이만 변화시킨다.

Notations



비디오 클립을 c(채널 수) x l(프레임 수) x h(프레임 높이) x w(프레임 너비)의 크기로 언급한다.

3D 합성곱 커널과 풀링 크기를 d(시간적 깊이) x k(커널의 공간적 크기) x k(커널의 공간적 크기)로 표기한다.

Common network settings

네트워크는 비디오 클립을 입력으로 받아 101가지 다른 행동에 속하는 클래스 레이블을 예측하

도록 설정되었다. 모든 비디오 프레임은 128 × 171로 크기 조정되었다. 이는 대략 UCF101 프레임의 절반 해상도이다. 비디오는 중첩되지 않는 16프레임 클립으로 분할되며, 이 클립들은 네트워크의 입력으로 사용된다.

입력 차원은 3(채널 수) × 16(프레임 수) × 128(프레임 높이) × 171(프레임 너비)입니다. 우리는 또한 훈련 중 입력 클립의 3 × 16 × 112 × 112 크기의 무작위 크롭을 사용하여 jittering을 사용합니다.

Examples:

```
>>> # With square kernels and equal stride
>>> m = nn.Conv3d(16, 33, 3, stride=2)
>>> # non-square kernels and unequal stride and with padding
>>> m = nn.Conv3d(16, 33, (3, 5, 2), stride=(2, 1, 1), padding=(4, 2, 0))
>>> input = torch.randn(20, 16, 10, 50, 100)
>>> output = m(input)
```

Pytorch의 torch.nn.Conv3d 사용예시이다. Conv3d의 입력차원은 보통 5(배치크기, 깊리, 높이, 너비, 채널 수)이다.

◎ Jittering : 지터링은 데이터 증강 기법 중 하나로, 모델이 과적합을 방지하고 일반화 성능을 향상시키는 데 도움을 주는 방법이다. 특히 비디오 또는 이미지 데이터를 다룰 때, 입력 데이터에 약간의 변형을 주어 모델이 더 다양한 형태의 데이터를 학습할 수 있도록 한다. 원본 비디오 클립의 무작위 위치에서 112 × 112 크기의 부분을 선택한다. 이 과정은 각 색상 채널마다 동일하게 적용되므로, 최종적으로 선택되는 부분의 차원은 3 × 16 × 112 × 112가 된다. 이는 원본 해상도에서 무작위로 선택된 작은 부분을 의미한다.

5개의 합성곱 레이어, 5개의 풀링 레이어를 가지며, 각 합성곱 레이어가 작업을 수행한 이후 바로 풀링 레이어로 전달되어 처리된다. 행동 레이블을 예측하기 위해 2개의 FC(fully connected) layer와 softmax loss layer를 가진다. 5개의 합성곱 레이어에 대한 필터의 개수는 각각 64, 128, 256, 256, 256이다. 모든 커널은 d(시간 깊이)의 크기를 가진다.

모든 합성곱 레이어는 적절한 패딩과 1의 스트라이드를 적용하여 입력에서 출력까지 크기가 변하지 않는다.

모든 풀링 레이어의 커널 크기는 2(시간적 깊이) x 2(커널의 공간적 크기) x 2(커널의 공간적 크기)인 max pooling과 1의 스트라이드를 적용하여 8배로 크기가 줄어든다.

첫 번째 풀링 레이어는 시간적 신호를 너무 이르게 합치지 않으면서 16프레임 클립 길이를 만족시키기 위해 1 × 2 × 2의 커널 크기를 가진다.

2개의 FC layer는 2048개의 출력을 가진다. 30개의 미니 배치를 이용하며, 초기 학습률은 0.003이

며, 4 에포크마다 10으로 나누며 16 에포크 후에 훈련이 중단된다.

Varying network architectures

이 논문에서 깊은 네트워크를 통해 시간적 정보를 어떻게 집계하는지 초점을 맞춘다. 보다 좋은 3D ConvNet 구조를 찾기 위해, 다른 설정을 고정하고 합성곱 레이어의 커널의 d_i(시간적 깊이)만 변화시킨다. 다음 2가지 유형의 구조를 실험한다.

1. 동일한 시간 깊이 : 모든 합성곱 레이어가 같은 시간 깊이를 가진다.
2. 변화하는 시간 깊이 : 시간 깊이가 레이어들 사이에서 변화한다.

동일한 시간 깊이의 경우, 시간 깊이 d를 각각 1, 3, 5, 7 4가지로 다른 네트워크를 실험한다. 이 네트워크들은 depth-d로 명명한다. depth-1의 경우 깊이가 1이므로 비디오에 2D 합성곱을 적용한 것과 동등하다.

변화하는 시간 깊이의 경우, 첫번째로 시간 깊이가 3-3-5-5-7로 증가하는 것과 두번째로 7-5-5-3-3으로 감소하는 것 2가지를 실험한다.

모든 네트워크들이 마지막 풀링 레이어에서 같은 크기의 출력 신호를 가지기 때문에 FC layer에서 동일한 수의 매개변수를 가진다. 커널의 시간 깊이가 다르기 때문에 매개변수의 수는 합성곱 레이어에서만 다르다. 이는 시간 깊이의 차이가 클수록 매개변수의 차이도 크다.

예를 들어 depth-7은 depth-1 보다 51,000개 더 많다. 이는 총 매개변수의 0.3% 미만이라서 학습 능력이 비교적 유사하다.

Exploring kernel temporal depth

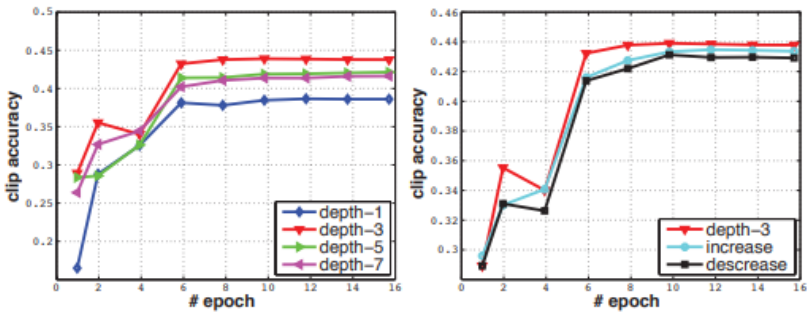


Figure 2. **3D convolution kernel temporal depth search.** Action recognition clip accuracy on UCF101 test split-1 of different kernel temporal depth settings. 2D ConvNet performs worst and 3D ConvNet with $3 \times 3 \times 3$ kernels performs best among the experimented nets.

Figure2는 네트워크들을 UCF101의 train split1에서 학습하고 UCF101 test split 1에서 다양한 아키텍처의 클립 정확도를 보여준다.

왼쪽 그래프는 동일한 시간 깊이를 가진 네트워크의 결과를 보여주고, 오른쪽 그래프는 커널 시간 깊이가 변하는 네트워크의 결과를 보여준다.

동일한 시간 깊이에서는 depth-3가 가장 좋은 성능을 보인다. 반면, depth-1은 다른 네트워크에 비해 현저히 나쁜데, 이는 운동 모델링의 부족 때문이라고 생각한다.

시간 깊이가 변하는 네트워크와 비교했을 때, depth-3이 가장 좋은 성능을 보이지만, 차이는 더 작다.

Spatiotemporal feature learning

Network architecture

3 × 3 × 3(depth-3)의 합성곱 커널이 3D ConvNet에 가장 적합하다는 것을 알았다.

8개의 합성곱 레이어, 5개의 풀링 레이어를 가진 3D ConvNet을 디자인했으며, 이어서 두 개의 FC layer와 소프트맥스 출력 레이어가 뒤따른다.

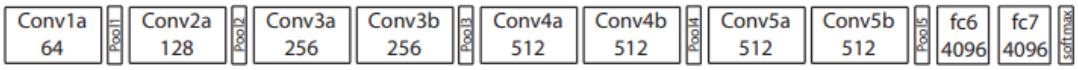


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling kernels are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

네트워크 아키텍처는 Figure3과 같다.

모든 3D convolution kernel size : 3 × 3 × 3이며, stide : 1 × 1 × 1

모든 3D pooling layer size : 2 × 2 × 2, stride : 2 × 2 × 2

pool1은 초기 단계에서 시간적 정보를 보존하려는 의도로 1 × 2 × 2의 커널 크기와 1 × 2 × 2의 스트라이드를 가진다. 각 완전 FC layer는 4096개의 출력 유닛을 가진다.

◎ 3차원 공간에서의 합성곱 연산을 설명할 때, 스트라이드는 각각의 차원(가로, 세로, 깊이)에 대해 얼마나 많이 이동할지를 나타낸다. 따라서 "1 x 1 x 1"은 합성곱 커널이 각 차원을 한 칸씩 이동한다는 것을 의미한다.

Dataset

Sports-1M 데이터셋에서 C3D를 훈련시킨다. 이 데이터셋은 110만 개의 스포츠 비디오로 구성되어 있다. 각 비디오는 487개의 스포츠 카테고리 중 하나에 속한다.

Training

train은 Sports-1M train split에서 이루어진다.

훈련 비디오마다 무작위로 5개의 2초 길이 클립을 추출한다. 클립들은 프레임 크기가 128×171이 되도록 크기 조정된다.

훈련 시, 입력 클립을 공간적 및 시간적 jittering을 위해 16×112×112 크기로 무작위로 자른다. 또한, 50% 확률로 수평으로 뒤집는다.

미니배치의 크기가 30이며, SGD(확률적 경사 하강법)로 수행된다.

초기 학습률은 0.003아며, 매 150,000 반복마다 2로 나눈다. 최적화는 1,900,000 반복(약 13 에포크)에서 중단된다(전체 훈련을 약 13에포크 진행한다). I380K에서 사전학습된 모델도 함께 실험했다.

Sports-1M classification results

Method	Number of Nets	Clip hit@1	Video hit@1	Video hit@5
DeepVideo's Single-Frame + Multires [18]	3 nets	42.4	60.0	78.5
DeepVideo's Slow Fusion [18]	1 net	41.9	60.9	80.2
Convolution pooling on 120-frame clips [29]	3 net	**70.8***	**72.4**	**90.8**
C3D (trained from scratch)	1 net	44.9	60.0	84.4
C3D (fine-tuned from I380K pre-trained model)	1 net	46.1	61.1	85.2

Table 2. **Sports-1M classification result**. C3D outperforms [18] by 5% on top-5 video-level accuracy. (*)We note that the method of [29] uses long clips, thus its clip-level accuracy is not directly comparable to that of C3D and DeepVideo.

Table2에서 DeepVideo와 Convolution pooling을 C3D 네트워크와 비교해서 결과를 보여준다.

클립당 단일 중앙 자르기만을 사용하고 이를 네트워크를 통과시켜 클립 예측을 한다. 비디오 예측의 경우, 비디오에서 무작위로 추출된 10개의 클립의 클립 예측을 평균한다.

DeepVideo와 C3D는 짧은 클립을 사용하는 반면, Convolution pooling은 훨씬 긴 클립을 사용한다. DeepVideo는 더 많은 split을 사용하는 반면, C3D는 각각 1개와 10개의 split을 사용한다.

scratch를 통해 훈련된 C3D는 84.4% 정확도를 보이며, I380K에서 사전 학습된 경우 비디오 top5 정확도에서 85.5% 정확도를 보인다. 두 C3D 네트워크가 DeepVideo 네트워크를 능가한다.

Convolution pooling보다 5.6% 낮다. 하지만, 이 방법은 120프레임의 긴 클립을 사용하기에 훨씬 짧은 클립에서 작동하는 C3D와 DeepVideo와 직접 비교할 순 없다. 그리고 클립과 비디오에 대한 top1정확도 차이가 1.6% 정도로 작다는 점을 주목한다.

◎ top1 : 모델이 가장 높은 확률로 예측한 클래스가 실제 정답 클래스와 정확히 일치할 때를 의미한다. 즉, 모델의 예측 중 가장 확신도가 높은 예측이 정확해야 top1 정확도가 올라간다.

◎ top5 : 모델이 예측한 확률이 가장 높은 상위 5개 클래스 중 하나가 실제 정답 클래스와 일치할 때를 의미한다. 모델이 제시한 상위 5개의 예측 중 어느 하나라도 정답이면, 그 예측은 top5 정확도가 올라간다.

C3D video descriptor

훈련된 C3D는 다른 비디오 분석 작업을 위한 feature 추출기로 사용될 수 있다. C3D feature를 추출하기 위해, 비디오는 16프레임 길이의 클립으로 분할되며, 두 연속 클립 사이에는 8프레임의 중복이 있다.

이 클립들은 C3D 네트워크에 전달되어 fc6 activation을 추출한다. 이 클립 fc6 activation은 평균화되어 4096차원의 비디오 설명자를 형성하며, 이어서 L2-정규화를 거친다.

◎ fc6 activation : FC(fully connected) layer6의 약자로 신경망의 6번째 FC layer를 의미한다. Activation은 활성화 함수를 의미한다. 즉, fc6 activation은 6번째 FC layer에서 뉴련들의 활성화 상태이며, 그 계층을 통과한 후 각 뉴런들의 출력값을 의미한다.

◎ L2 정규화 : 벡터의 각 원소를 그 벡터의 L2 노름(유클리드 거리)로 나누어 주는 과정이다. 이는 벡터의 길이를 1로 만들어 주어, 다양한 벡터들이 동일한 스케일을 갖도록 조정하는 방법이다. L2 정규화는 벡터 내의 값들을 상대적으로 비교 가능하게 만들어 주며, 기계 학습에서 특히 데이터의 스케일 차이를 줄이는 데 유용하게 사용된다.

What does C3D learn?

C3D는 처음 몇 프레임에서 외관에 집중을 시작하고 이후 프레임에서 두드러진 움직임을 추적한다.

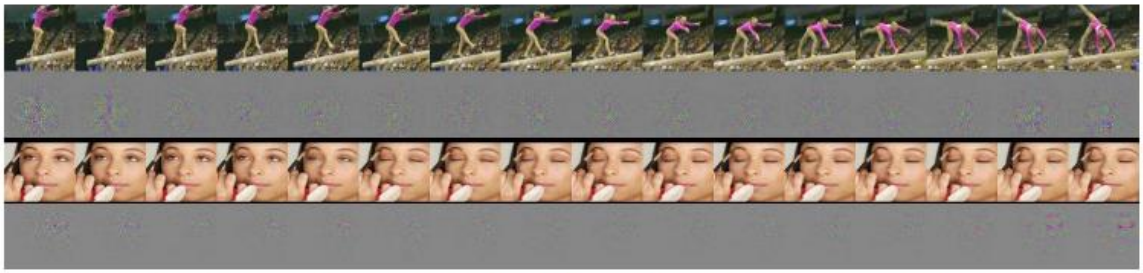


Figure 4. **Visualization of C3D model, using the method from [46].** Interestingly, C3D captures appearance for the first few frames but thereafter only attends to salient motion. Best viewed on a color screen.

Figure4는 이미지 공간으로 다시 투영된 가장 높은 활성화를 가진 두 C3D conv5b feature 맵의 deconvolution을 시각화한다.

◎ conv5b : conv는 컨볼루션을 의미하며, 5는 이 계층이 모델 내에서 다섯 번째 컨볼루션 계층임을 나타낸다. b는 이 계층이 동일 순서 내에서 두 번째 버전 또는 분기(branch)임을 나타내는데 사용될 수 있다. 예를 들어, conv5a 다음에 오는 계층이라면 conv5b라고 할 수 있다.

◎ deconvolution : 컨볼루션 연산의 역과정을 수행하는 방법 중 하나이며, transposed convolution이라 불리기도 한다.
컨볼루션 연산은 일반적으로 입력 데이터의 크기를 줄이는 특성이 있다. 예를 들어, 이미지 위에 컨볼루션 필터를 적용하면, 결과적으로 축소된 특성 맵(feature map)을 얻게 된다. 이 과정은 데이터의 중요한 특성을 추출하고, 모델의 parameter 수를 줄이며, 계산 효율성을 높이는 역할을 한다.
반면, 디컨볼루션은 이 과정의 역을 수행하여, 작은 특성 맵으로부터 원래의 크기 또는 더 큰 크기의 데이터를 복원하거나 생성한다. 이를 통해, 모델은 저수준의 특성으로부터 고수준의 데이터를 재구성할 수 있다. 예를 들어, image segmentation(이미지를 구성하는 픽셀들을 여러 개의 부분으로 나누는 과정)에서는 디컨볼루션을 사용하여 작은 특성 맵에서 세밀한 segmentation map을 생성한다.

첫 번째 예에서, feature는 전체 사람에 초점을 맞추고 나머지 프레임에서 장대 높이뛰기의 움직임을 추적한다. 비슷하게 두 번째 예에서는 먼저 눈에 초점을 맞춘 다음 메이크업을 바르는 동안 눈 주변에서 발생하는 움직임을 추적한다. 따라서 C3D는 표준 2D ConvNets와 다르게 움직임과 외관에 선택적으로 주목한다.

Action recognition

Dataset

UCF101 데이터셋에서 C3D feature를 평가한다. 이 데이터셋은 101가지 인간 동작 카테고리의 13,320개 비디오로 구성된다. 이 데이터셋에 제공된 세 가지 분할 설정을 사용한다.

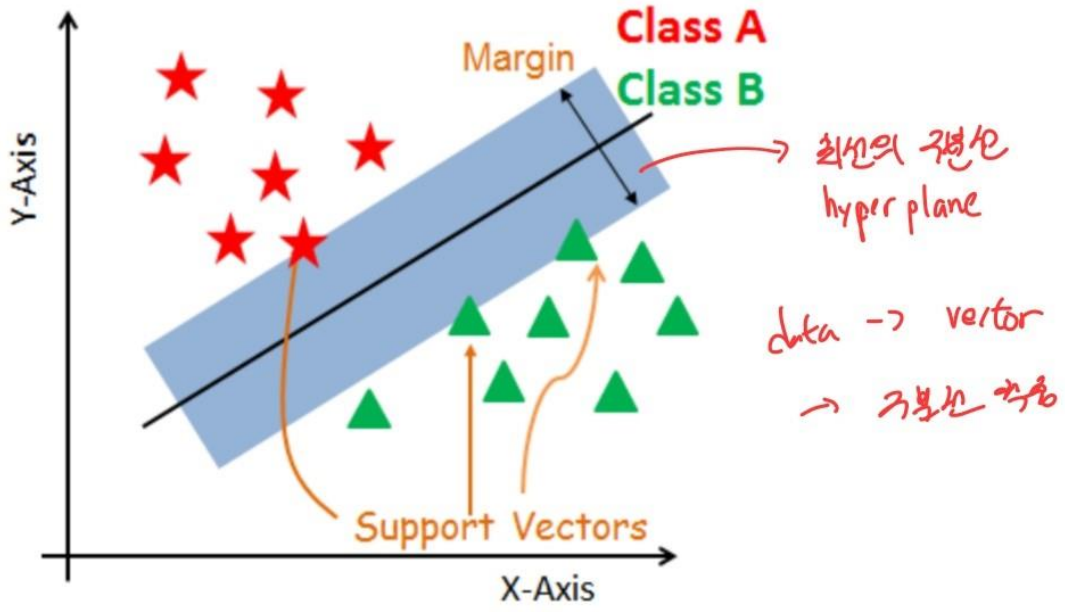
Classification model

C3D를 통해 feature를 추출하고 이를 다중 클래스 선형 SVM에 입력하여 모델을 훈련한다. 3가지 다른 네트워크를 사용하여 C3D를 실험한다

1. I380K에서 훈련된 C3D
2. Sports-1M에서 훈련된 C3D
3. I380K에서 훈련되고 Sports-1M에서 미세 조정된 C3D

L2 정규화 과정을 거친 후, 이러한 정규화된 C3D descriptor들을 서로 연결한다.

◎ SVM



SVM은 지도학습에서 분류 혹은 회귀 문제를 해결하기 위해 사용되는 머신러닝 모델 중 하나이다. 학습된 hyperplane을 기준으로 샘플들을 분류한다. hyperplane은 두 클래스 사이의 margin을 최대화 하는 방향으로 학습된다.

Baselines

현재 최고의 수작업 feature로 알려진 iDT(improved Dense Trajectories)와 널리 사용되는 딥 이미지 feature(Caffe의 Imagenet 사전 훈련 모델을 사용한 Imagenet) 2가지를 C3D feature를 비교한다.

iDT의 경우, 궤적, HOG, HOF, MBHx, MBHy의 각 iDT feature 채널에 대해 5000 크기의 codebook을 사용하는 단어 가방 표현을 사용한다. 우리는 각 채널의 히스토그램을 L1-노름(벡터의 각 요소의 절대값의 합)을 사용해 별도로 정규화하고 이러한 정규화된 히스토그램을 연결하여 비디오에 대한 25,000 feature 벡터를 형성한다.

Imagenet 기준선의 경우, C3D와 유사하게, 우리는 각 프레임에 대한 Imagenet fc6 feature를 추출하고, 이 프레임 feature들을 평균 내어 video descriptor를 만든다. 이 두 기준선에 대해서도 공정한 비교를 위해 다중 클래스 선형 SVM이 사용된다.

Result

Method	Accuracy (%)
Imagenet + linear SVM	68.8
iDT w/ BoW + linear SVM	76.2
Deep networks [18]	65.4
Spatial stream network [36]	72.6
LRCN [6]	71.1
LSTM composite model [39]	75.8
C3D (1 net) + linear SVM	82.3
C3D (3 nets) + linear SVM	**85.2**
iDT w/ Fisher vector [31]	87.9
Temporal stream network [36]	83.7
Two-stream networks [36]	88.0
LRCN [6]	82.9
LSTM composite model [39]	84.3
Conv. pooling on long clips [29]	88.2
LSTM on long clips [29]	88.6
Multi-skip feature stacking [25]	89.1
C3D (3 nets) + iDT + linear SVM	**90.4**

Table 3. **Action recognition results on UCF101**. C3D compared with baselines and current state-of-the-art methods. Top: simple features with linear SVM; Middle: methods taking only RGB frames as inputs; Bottom: methods using multiple feature combinations.

Table3는 C3D와 두 가지 baseline 및 현재 최고의 방법들과 비교한 동작 인식 정확도를 제시한다.

상단 부분은 두 baseline의 결과를 보여준다.

중간 부분은 입력으로 오직 RGB 프레임만을 사용하는 방법들의 결과를 비교한다.

하단 부분은 모든 가능한 feature 조합(optical flow, iDT)을 사용하는 현재 최고의 방법들의 결과를

비교한다.

C3D 중에서 1net에서 3net으로 증가하여 차원이 3배로 증가하여 정화도가 82.3%에서 85.2%로 향상되었다.

C3D를 iDT와 결합하면 정확도가 더욱 향상되어 90.4%에 이른다. C3D를 iDT와 결합하는 것은 두 기술이 서로 매우 보완적이기 때문에 유익하다. 실제로, iDT는 optical flow tracking과 low level gradients의 히스토그램을 기반으로 하는 수작업으로 만들어진 feature이며, C3D는 고수준의 추상적, 의미적 정보를 포착한다.

C3D를 3개의 네트워크와 함께 사용했을 때, 85.2%의 성능을 달성하여 iDT(76.2%)와 이미지넷 기준 모델(68.8%)보다 각각 9%와 16.4% 향상되었다.

오직 RGB 입력 설정에서, CNN 기반 접근법(85.2%)과 비교할 때, C3D는 의 Deep networks(65.4%)와 Spatial stream network(72.6%)를 각각 19.8%와 12.6%로 뛰어넘었다.

두 네트워크는 AlexNet 구조를 사용하며, 각각 Sport-1M과 이미지넷 pre-trained model에서 fine tuning되었다. 하지만, C3D는 fine tuning 없이 Sports-1M에서 훈련되었다.

C3D는 iDT와 결합하면 Two-stream network, 다른 iDT 기반 방법들(iDT w/Fisher vector, Multi-skip feature stacking) 그리고 장기 모델링에 초점을 맞춘 방법(Conv. pooling on long clips)을 능가할 수 있다. 좋은 성능 수치와 더불어, C3D는 다른 방법들에 비해 단순하다는 장점도 가지고 있다.

C3D is compact

C3D의 feature가 compact하다는 것을 평가하기 위해, PCA를 사용하여 feature를 낮은 차원으로 투영하고 선형 SVM을 사용하여 UCF101에서 투영된 기능의 분류 정확도를 본다. iDT와 Imagenet에서 동일한 과정을 적용한 결과가 Figure5이다.

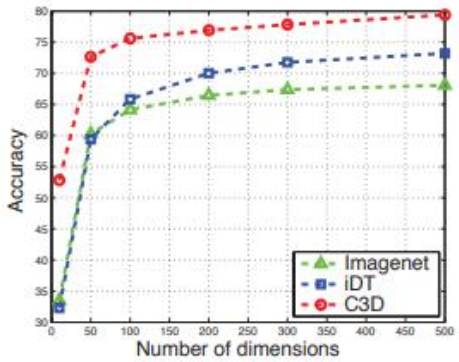


Figure 5. **C3D compared with Imagenet and iDT in low dimensions**. C3D, Imagenet, and iDT accuracy on UCF101 using PCA dimensionality reduction and a linear SVM. C3D outperforms Imagenet and iDT by 10-20% in low dimensions.

Figure5는 PCA 차원축소, linear SVM을 사용한 iDT와 Imagenet을 C3D가 10~20% 낮은 차원에서 성능을 능가했다.

극단적으로 10차원에서 C3D의 정확도는 52.8%로 Imagenet과 iDT의 정확도인 약 32%보다 20% 이상 더 높다. 50차원과 100차원에서, C3D는 각각 72.6%와 75.6%의 정확도를 얻어, Imagenet과 iDT보다 약 10-12% 더 높다. 마지막으로, 500차원에서, C3D는 79.4%의 정확도를 달성하여 iDT보다 6% 더 높고 Imagenet보다 11% 더 높다. 이는 C3D의 기능이 compact하면서도 차별적이다는 것을 나타낸다. 이는 낮은 저장 비용과 빠른 검색이 중요한 대규모 검색 애플리케이션에 매우 유용하다.

그리고 또 다른 데이터셋에서 학습된 C3D feature의 임베딩을 시각화함으로써, 비디오에 대한 좋은 일반적 기능인지를 정성적으로 평가한다. 우리는 UCF101에서 무작위로 100K 클립을 선택한 다음, Imagenet 및 C3D의 기능을 사용하여 해당 클립에 대한 fc6 기능을 추출한다. 이 기능들은 그 다음 t-SNE를 사용하여 2차원 공간으로 투영된다.

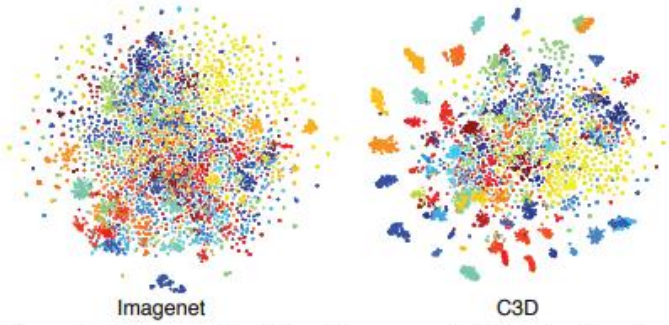


Figure 6. **Feature embedding**. Feature embedding visualizations of Imagenet and C3D on UCF101 dataset using t-SNE [43]. C3D features are semantically separable compared to Imagenet suggesting that it is a better feature for videos. Each clip is visualized as a point and clips belonging to the same action have the same color. Best viewed in color.

Figure6는 Imagenet과 C3D를 UCF101 데이터셋에서 t-SNE를 사용한 특징 임베딩을 시각화한 것이다. C3D feature는 이미지넷에 비해 의미론적으로 분리가 가능하다고 제안되며, 비디오에 대한 더 나은 feature로 여겨진다. 각 클립은 점으로 시각화되며, 동일한 동작에 속하는 클립은 같은 색을 가진다. 색상으로 보는 것이 가장 좋다.

데이터셋 간에 좋은 일반화 능력을 보이는지 확인하기 위해 어떠한 fine tuning도 하지 않았다는 점에 주목할 가치가 있다. 우리는 정량적으로 관찰하여 C3D가 Imagenet보다 더 낫다고 판단한다.

◎ Feature embedding : 특징 임베딩은 고차원의 데이터를 저차원의 공간으로 매핑하는 기술이다. 이 과정에서 데이터의 중요한 특성이나 구조를 보존하려고 한다. 임베딩은 단순한 차원 축소를 넘어, 데이터 사이의 복잡한 관계나 패턴을 학습하고 이를 저차원에서 표현하려는 목적을 가진다.

◎ t-SNE : 고차원 데이터를 2차원 또는 3차원 공간에 시각화하기 위해 널리 사용되는 기술이다. 고차원의 데이터 포인트 사이의 유사성을 확률분포로 모델링하고, 이를 저차원 공간에서도 유사한 확률분포를 가지도록 매핑한다. 이 과정에서 고차원에서의 유사한 데이터 포인트들이 저차원에서도 가까이 위치하게 되어, 데이터의 구조나 군집을 시각적으로 이해하기 쉬워진다.

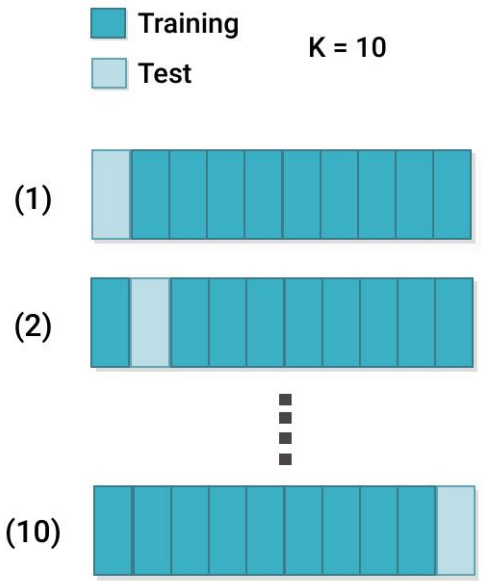
Action Similarity Labeling

Dataset

ASLAN 데이터셋은 432개의 행동 분류로부터 온 3,631개의 비디오로 구성되어 있다. 주어진 비디오 쌍이 같은 행동에 속하는지 또는 다른 행동에 속하는지를 예측하는 것이 과제이다. 우리는 데이터셋과 함께 제공된 분할을 사용하여 지정된 10-fold cross validation을 사용한다.

이 문제는 행동 인식과는 다르며, 과제는 실제 행동 라벨을 예측하는 것이 아니라 행동 유사성을 예측하는 데 초점을 맞춘다. 이 과제는 테스트 세트에 "*never seen before*" 행동의 비디오가 포함되어 있기 때문에 상당히 도전적이다.

◎ 10-fold cross validation : 10겹 교차 검증은 기계 학습에서 모델의 성능을 평가하기 위해 사용되는 방법 중 하나이다. 이 방법은 전체 데이터셋을 10개의 동일한 크기를 가진 부분집합으로 나누고, 이 중 9개를 훈련 데이터로 사용하고 남은 1개를 검증 데이터로 사용하는 과정을 10번 반복한다.
각 반복에서 검증 데이터로 사용되는 부분집합은 바뀌며, 모든 부분집합이 정확히 한 번씩 검증 데이터로 사용된다(검증 데이터가 겹칠 수 없다). 이렇게 함으로써, 모델이 다양한 데이터 조합에 대해 얼마나 잘 작동하는지 평가할 수 있다. 10-겹 교차 검증을 통해 얻어진 성능 지표들의 평균값은 모델의 전반적인 성능을 나타내는 데 사용된다.



Features

비디오를 16프레임 클립으로 나누고, 겹치는 8프레임을 가지고 있습니다. 각 클립에 대해 C3D 특성(prob, fc7, fc6, pool5)을 추출한다. 비디오의 특성은 각 유형의 특성에 대해 별도로 클립 특성의 평균을 계산하고, 이어서 L2 정규화를 통해 계산된다.

Classification model

우리는 [21]에서 사용된 것과 같은 설정을 따릅니다. 비디오 쌍이 주어지면, 우리는 [21]에서 제공된 12가지 다른 거리를 계산한다. 4가지 유형의 특성을 가지고 있으므로, 각 비디오 쌍에 대해 48

차원(12(거리) × 4(특성) = 48) 특성 벡터를 얻는다. 이 48개의 거리는 서로 비교할 수 없기 때문에, 각 차원이 평균이 0이고 단위 분산을 가지도록 독립적으로 정규화한다. 마지막으로, 이 48차원 특성 벡터에 대해 선형 SVM이 훈련되어 비디오 쌍을 같은 것 또는 다른 것으로 분류한다.

[21] : ASLAN(Action Similarity Labeling Challenge 동작 유사성 라벨링 도전)데이터셋을 소개하고, 이를 사용하여 동작 유사성을 판단하는 방법론과 알고리즘의 성능을 평가한다.

현재 방법과 비교하는 것 외에도, 우리는 이미지 기반의 특성을 사용하는 강력한 base line과 C3D를 비교합니다. Base line은 우리의 C3D와 같은 설정을 가지고 있지만, feature는 Imagenet feature이다.

Result

Method	Features	Model	Acc.	AUC
[21]	STIP	linear	60.9	65.3
[22]	STIP	metric	64.3	69.1
[20]	MIP	metric	65.5	71.9
[11]	MIP+STIP+MBH	metric	66.1	73.2
[45]	iDT+FV	metric	68.7	75.4
Baseline	Imagenet	linear	67.5	73.8
Ours	**C3D**	linear	**78.3**	**86.5**

Table 4. **Action similarity labeling result on ASLAN**. C3D significantly outperforms state-of-the-art method [45] by 9.6% in accuracy and by 11.1% in area under ROC curve.

Table4에서 최신 방법들과 비교한다. 대부분의 현재 방법들이 다양한 수작업 feature, 강력한 인코딩 방법들(VLAD, Fisher Vector), 그리고 복잡한 학습 모델들을 사용하는 반면, 우리의 방법은 비디오 전체에 걸친 C3D 특징의 단순 평균화와 선형 SVM을 사용한다.

C3D는 정확도에서 9.6%, ROC 곡선 아래 영역(AUC)에서 11.1%로 최신 방법[45](iDT+FV)을 크게 앞선다. Imagenet baseline은 최신 방법[45]보다 단지 1.2% 낮게 수행되는데, 이는 운동 모델링의 부재로 인해 C3D보다 10.8% 떨어지는 수치이다.

[45] : 동작 유사성 라벨링을 위한 Large margin dimensionality reduction for action similarity labeling(대 마진 차원 축소 기법)에 관한 연구이다.

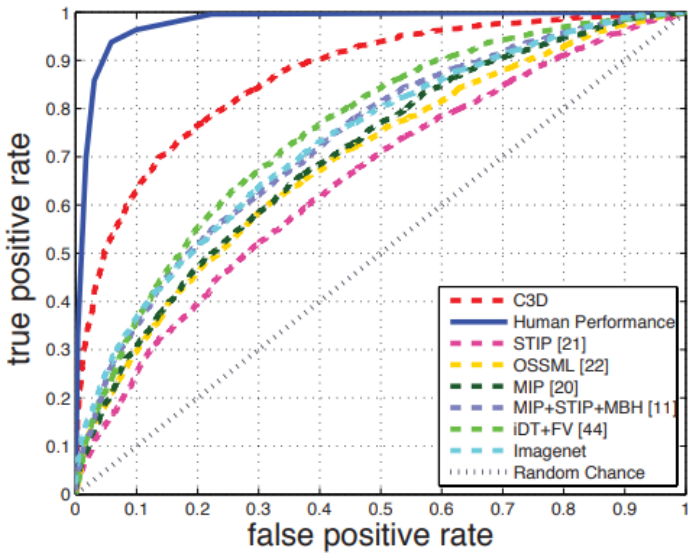


Figure 7. **Action similarity labeling result**. ROC curve of C3D evaluated on ASLAN. C3D achieves 86.5% on AUC and outperforms current state-of-the-art by 11.1%.

Figure7은 현재 방법들과 인간 성능에 비해 C3D의 ROC 곡선을 비교한 것이다. C3D는 현저한 향상을 이루었으며, 이는 현재 최신 방법에서 human performance(98.9%)까지의 절반 지점에 해당한다.

◎ Human performance : C3D 같은 컴퓨터 비전 알고리즘과 비교되는 인간의 시각적 인식 능력의 최고 수준을 지칭한다.

Scene and Object Recognition

dataset

동적 장면 인식을 위해, YUPENN과 Maryland 두 벤치마크에서 C3D를 평가한다.

YUPENN은 14개의 장면 카테고리에 대한 420개의 비디오로 구성되어 있고, Maryland는 13개의 장면 카테고리에 대한 130개의 비디오로 구성되어 있다.

Object recognition을 위해, 우리는 매일 사용하는 42종류의 객체가 포함된 egocentric(1인칭 시점 데이터셋) 데이터셋에서 C3D를 테스트합니다. 모든 비디오가 1인칭 시점에서 녹화되었으며, 훈련 데이터셋에서 가지고 있는 비디오들과는 매우 다른 외형 및 동작 특성을 가지고 있다.

classification model

두 데이터셋 모두에서, 우리는 feature 추출과 선형 SVM을 사용한 분류에 동일한 설정을 사용하고, 이 데이터셋의 저자들이 설명한 것과 같은 leave-one-out 평가 프로토콜을 따른다.

◎ Leave-one-out 프로토콜 : 교차 검증(cross-validation)의 한 형태로서, 모델의 성능을 평가하는 데 사용된다.
이 방법은 데이터셋의 모든 샘플을 대상으로 하여, 한 번에 하나의 샘플만을 테스트 데이터로 사용하고 나머지는 훈련 데이터로 사용하는 방식이다.
즉, 데이터셋에 N개의 샘플이 있을 경우, N번의 실험을 진행하게 되며, 각 실험에서 단 하나의 샘플이 테스트용으로 분리되고, 나머지 N-1개의 샘플은 훈련용으로 사용된다. 이 과정을 모든 샘플에 대해 반복한 후, 각 실험의 결과를 종합하여 모델의 전반적인 성능을 평가한다.

object 데이터셋의 경우, 표준 평가는 프레임을 기반으로 한다. 그러나, C3D는 특징을 추출하기 위해 16 프레임 길이의 비디오 클립을 사용한다. 우리는 C3D 특징을 추출하기 위해 모든 비디오에 16 프레임의 window를 slide한다. (16프레임 길이의 연속된 클립(window)을 하나의 단위로 사용한다.)

우리는 각 클립에 대한 ground truth 라벨을 클립의 가장 빈번하게 발생하는 라벨로 선택한다.

◎ Ground truth label : 데이터 샘플이 실제로 어떤 클래스에 속하는지를 정확히 나타내는 라벨을 의미한다. 이는 학습 데이터를 준비할 때, 각 데이터 포인트에 대해 전문가가 판단하여 부여한 정확한 분류나 측정값이다. 예를 들어, 이미지 분류 작업에서 각 이미지가 어떤 객체를 포함하고 있는지를 나타내는 라벨(예: 고양이, 개 등)이 ground truth 라벨이 된다.

클립에서 가장 빈번한 라벨이 8 프레임보다 적게 발생하는 경우, 우리는 그것을 객체가 없는 부정적인 클립으로 간주하고 훈련 및 테스트에서 모두 제외한다.

우리는 선형 SVM을 사용하여 C3D 특징을 훈련 및 테스트하고 객체 인식 정확도를 확인한다. 우리는 [32]에서 제공된 동일한 분할을 따릅니다. 또한, 우리는 이 3개의 벤치마크에서 Imagenet feature를 사용하는 baseline과 C3D를 비교한다.

[32] : egocentric 시각에서 다루는 객체들을 인식하는 문제에 초점을 맞추는 연구

Result

Dataset	[4]	[41]	[8]	[9]	Imagenet	**C3D**
Maryland	43.1	74.6	67.7	77.7	**87.7**	**87.7**
YUPENN	80.7	85.0	86.0	96.2	96.7	**98.1**

Table 5. **Scene recognition accuracy**. C3D using a simple linear SVM outperforms current methods on Maryland and YUPENN.

human performance (98.9%).

Table5에서 C3D 결과를 보고하고 현재 최고의 방법들과 비교한다.

장면 분류에서, C3D는 현재 최고의 방법[9]을 Maryland에서 10%, YUPENN에서는 1.9% 앞서는 성능을 보여준다.

[9] : 동적 장면 인식을 위한 새로운 접근 방식을 제시한다. 주요 내용은 '시공간 에너지 가방(Bags of Spacetime Energies)'라는 개념을 도입하여 동적인 장면을 인식하는 기술에 관한 것이다.

주목할 만한 점은 C3D가 클립 특징의 단순 평균화와 함께 선형 SVM만을 사용하는 반면, 두 번째로 좋은 방법[9]은 다양한 복잡한 특징 인코딩(FV(Fisher vector), LLC(Locality-constrained Linear Coding), dynamic pooling)을 사용한다는 것이다. Imagenet 기준선은 Maryland에서 C3D와 유사한 성능을 달성하고 YUPENN에서는 C3D보다 1.4% 낮다.

Object object recognition
[32]
12.0
22.3

object recognition에서, C3D는 22.3%의 정확도를 얻고 강력한 SIFT-RANSAC 특징 매칭에 RBF-커널을 사용한 비교 방법[32]보다 10.3% 앞서는 성능을 보여준다.

Imagenet baseline과 비교했을 때, C3D는 여전히 3.4% 떨어진다. 이는 C3D가 Imagenet이 사용하는 전체 크기 해상도(256x256)에 비해 더 작은 입력 해상도(128x128)를 사용하기 때문에 설명될 수 있다. C3D가 Sports1M 비디오에서만 훈련되었고 어떤 세밀한 조정도 없이 Imagenet은 1000개의 객체 카테고리에 완전히 훈련되었기 때문에, 이 작업에서 C3D가 잘 작동할 것으로 기대하지 않았다. 이 결과는 비디오에서 외관과 움직임 정보를 포착하는 C3D의 일반성을 보여준다.

Runtime analysis

우리는 C3D와 iDT, Temporal stream network의 실행 시간을 비교했다. iDT의 경우, 저자들이 친절하게 제공한 코드를 사용했다. [36]에 대해서는 평가할 수 있는 공개 모델이 없다. 그러나 이 방법

은 Brox의 optical flow을 입력으로 사용한다.

저자들이 제공한 CPU 구현과 OpenCV에서 제공하는 GPU 구현을 사용하여 Brox 방법의 실행 시간을 평가할 수 있었다. UCF101 데이터셋 전체에서 특징을 추출하기 위한(입출력 포함) 위에서 언급한 세 가지 방법의 실행 시간을 단일 CPU 또는 단일 K40 Tesla GPU를 사용하여 Table6에 보고한다.

Method Usage	iDT CPU	Brox's CPU	Brox's GPU	C3D GPU
RT (hours)	202.2	2513.9	607.8	2.2
FPS	3.5	0.3	1.2	313.9
x Slower	91.4	1135.9	274.6	1

Table 6. **Runtime analysis on UCF101**. C3D is 91x faster than improved dense trajectories [44] and 274x faster than Brox's GPU implementation in OpenCV.

Temporal stream network는 이미지 한 쌍에 대해 입출력을 제외한 계산 시간이 0.06초라고 보고했다. 우리의 실험에서, Brox의 GPU 구현은 이미지 쌍당 0.85-0.9초가 걸린다는 것을 포함하여 측정했다. 이는 iDT와의 비교가 공정하지 않음에 유의해야 한다. 왜냐하면 이 방법은 CPU만을 사용하기 때문이다. 이 방법의 GPU 구현을 찾을 수 없었으며, 이 알고리즘을 GPU에 병렬로 구현하는 것은 단순하지 않다.

C3D는 실시간보다 훨씬 빠르게, 초당 313 프레임을 처리하는 반면, 다른 두 방법은 4 프레임 미만의 처리 속도를 가지고 있다.

Conclusions

이 작업에서는 대규모 비디오 데이터셋에서 훈련된 3D ConvNets을 사용하여 비디오의 시공간 특성을 학습하는 문제를 해결하려고 시도했다.

3D ConvNets에 대한 최적의 시간적 커널 길이를 찾기 위해 체계적인 연구를 수행했다.

C3D가 외모와 움직임 정보를 동시에 모델링할 수 있으며, 다양한 비디오 분석 작업에서 2D ConvNet 특성을 능가함을 보였다.

선형 분류기와 함께 사용된 C3D 특성이 다른 비디오 분석 벤치마크에서 현재 최고의 방법들을 능가하거나 접근할 수 있음을 보였다.

마지막으로, C3D feature은 효율적이고, compact하며, 사용하기 극도로 간단하다.

Sports-1M dataset을 통해 학습된 C3D의 Sport classification