Apprentissage continu de représentations visuelles

ENSIMAG
2023-2024

Karteek Alahari & Diane Larlus

https://project.inria.fr/bigvisdata/

Aperçu du cours

Cours 1 : introduction

- Définition ‘Big Data’, applications, la chaine, lien avec la recherche d’information par le contenu
- List des thèmes abordés: apprentissage supervisé, auto-supervisé, adaptation de domaine, apprentissage continu, problèmes en vidéos
- Définition de l’apprentissage supervisé et de ses étapes majeures: collection des données, choix d’une représentation, choix d’un modèle, apprentissage, sélection d’un modèle
- Collection des données: difficulté et ambiguïté de l’annotation d’images, exemples de grandes bases d’images standard
- Décomposition de l’erreur
- Représentation des données: approches traditionnelles: représentations puis apprentissage, *mid-level representations*, apprentissage de bout-en-bout
- Définition de l’apprentissage profond

Aperçu du cours

Introduction

- Définition 'Big Data', applications, la chaine, lien avec la recherche d'information par le contenu
- List des thèmes abordés: apprentissage supervisé, auto-supervisé, adaptation de domaine, apprentissage continu, problèmes en vidéos
- Définition de l'apprentissage supervisé et de ses étapes majeures: collection des données, choix d'une représentation, choix d'un modèle, apprentissage, sélection d'un modèle
- Collection des données: difficulté et ambiguïté de l'annotation d'images, exemples de grandes bases d'images standard
- Décomposition de l'erreur
- Représentation des données: approches traditionnelles: représentations puis apprentissage, *mid-level representations*, apprentissage de bout-en-bout
- Définition de l'apprentissage profond

Adaptation de domaine

Quelques notions utiles

Comment créer des représentations visuelles ?

- Entrainer un modèle sur de grandes quantités de données annotées
- Utiliser ce modèle pour produire une représentation vectorielle pour chaque image proposée en entrée du modèle

Comment réutiliser des représentations visuelles ?

- Utiliser des représentations visuelles directement, optionnellement apprendre un modèle de décision par-dessus
- Utiliser le modèle précédent comme point de départ pour l'apprentissage et l'ajuster (*fine-tuning*) pour la tâche cible
- Appliquer une méthode d'adaptation, par exemple: adaptation de domaine

Comment créer des représentations visuelles quand on n'a pas d'annotations ?

- Introduction à l'apprentissage auto-supervisée

Aperçu du cours

Reconnaissance d’actions

- Space-time interest points [Laptev, IJCV’05]

- Dense trajectories [Wang and Schmid, ICCV’13]

Apprentissage auto-supervisé

- Different types of approaches
  - Discriminative (e.g., jigsaw) vs Generative (e.g., masked models)
- MoCo, SimCLR vs Barlow Twins, BYOL, DINO
- BERT/BEiT, MAE
- Evaluating representations

Multimodality
- e.g., CLIP

Dans ce cours

- Apprentissage supervisé (*supervised learning*)
  - Variantes, ex. Semi-supervisé (*semi-supervised*)
- L'adaptation de domaine (*domain adaptation*)
- Apprentissage auto-supervisé (*self-supervised*)
- **Problèmes en vidéos**
- **Apprentissage continu (*continual learning*)**

Reconnaissance d’actions

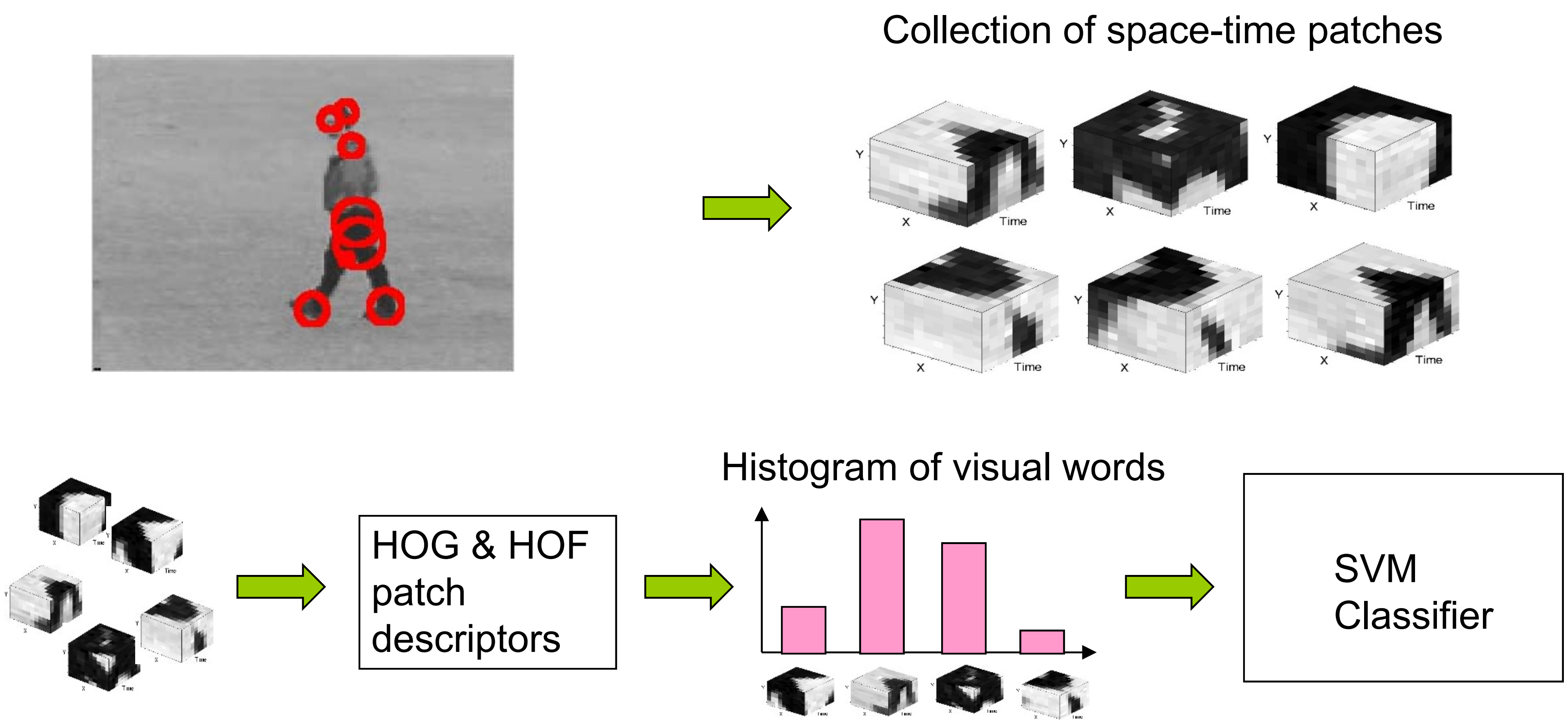
Quelques exemples d’approaches basées sur CNN

Action classification in videos

- Space-time interest points [Laptev, IJCV'05]
- Dense trajectories [Wang and Schmid, ICCV'13]
- Video-level CNN features

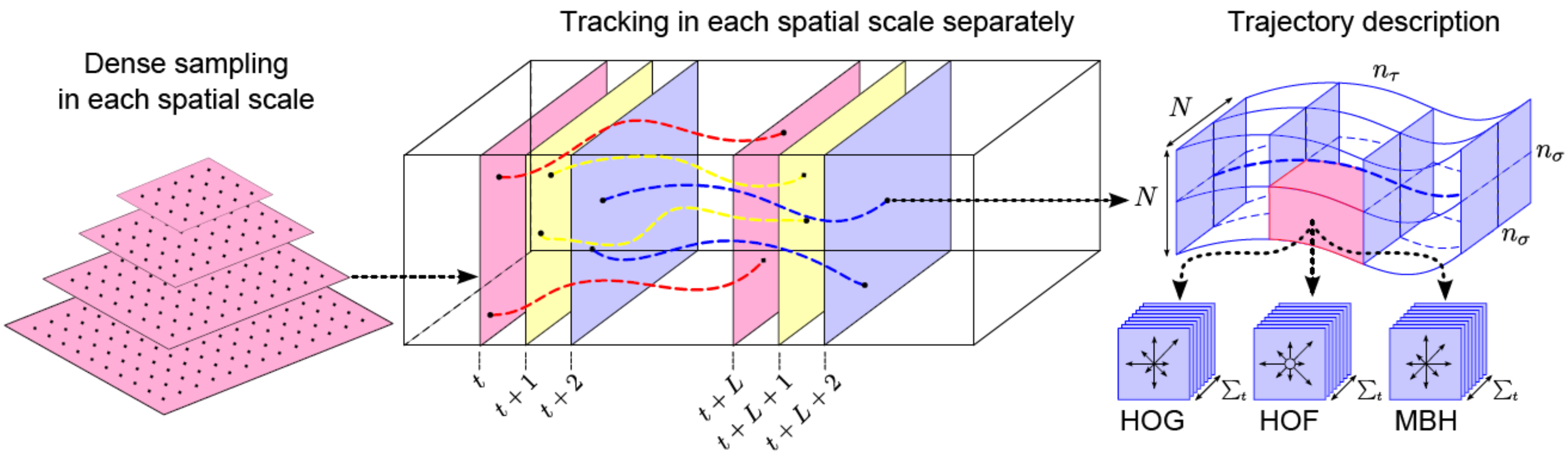
Action classification

- Bag of space-time features + SVM [Schuldt'04, Niebles'06, Zhang'07]



State of the art for video description

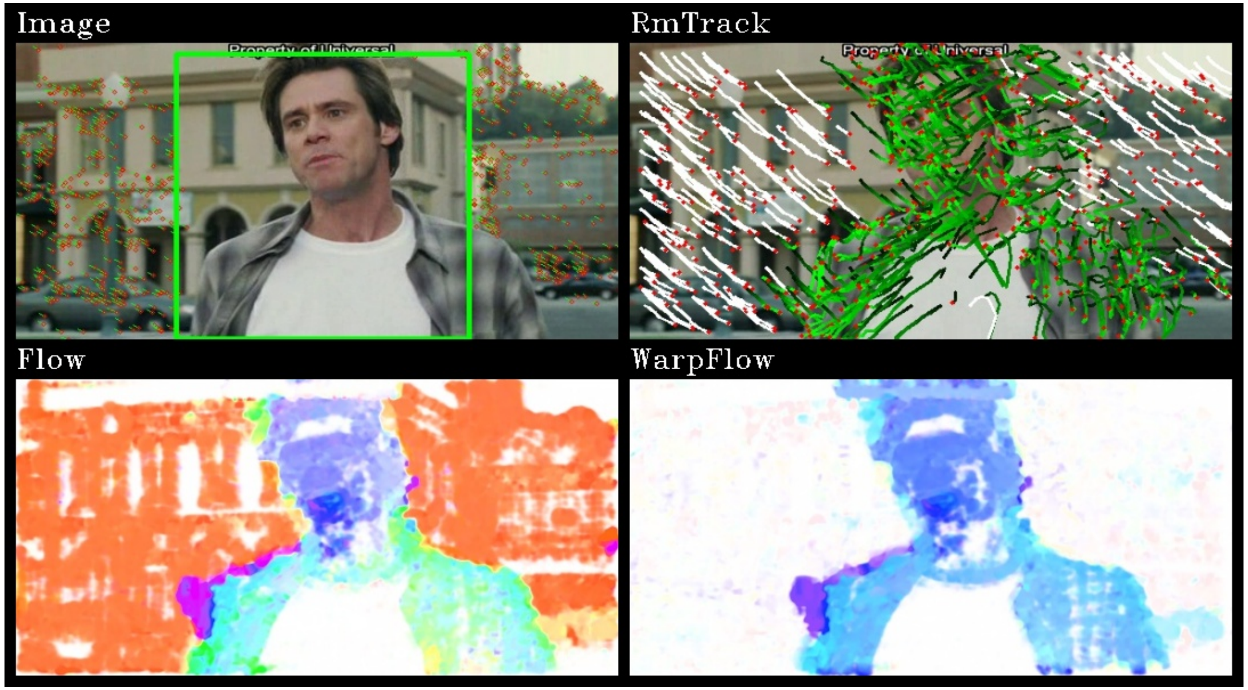
- Dense trajectories [Wang et al., IJCV'13] and Fisher vector encoding [Perronnin et al. ECCV'10]



- Orderless representation

Improved dense trajectories

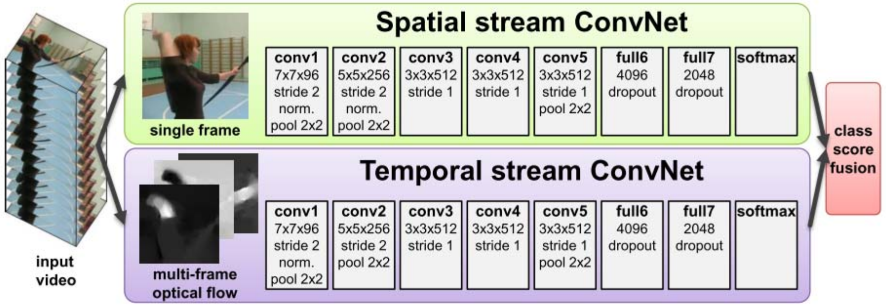
- Improve dense trajectories by explicit camera motion estimation
- Detect humans to remove outlier matches for homography estimation
- Stabilize optical flow to eliminate camera motion



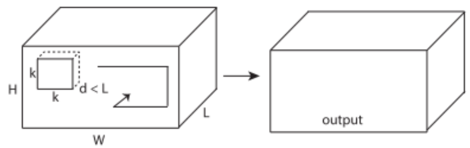
[Wang and Schmid. Action recognition with improved trajectories. ICCV'13]

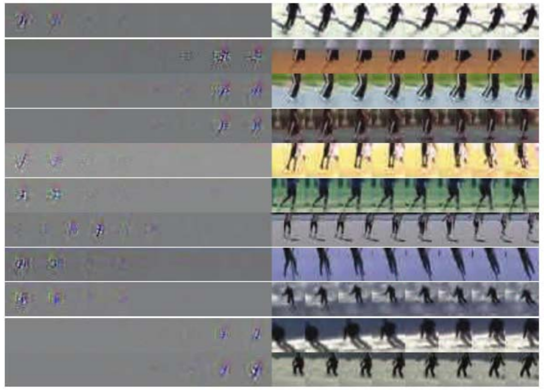
Recent CNN methods

Two-Stream Convolutional Networks for Action Recognition in Videos [Simonyan and Zisserman NIPS14]

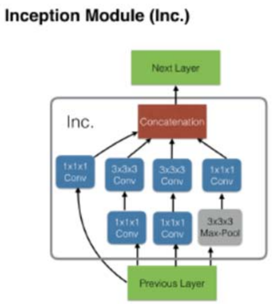


Learning Spatiotemporal Features with 3D Convolutional Networks [Tran et al. ICCV15]



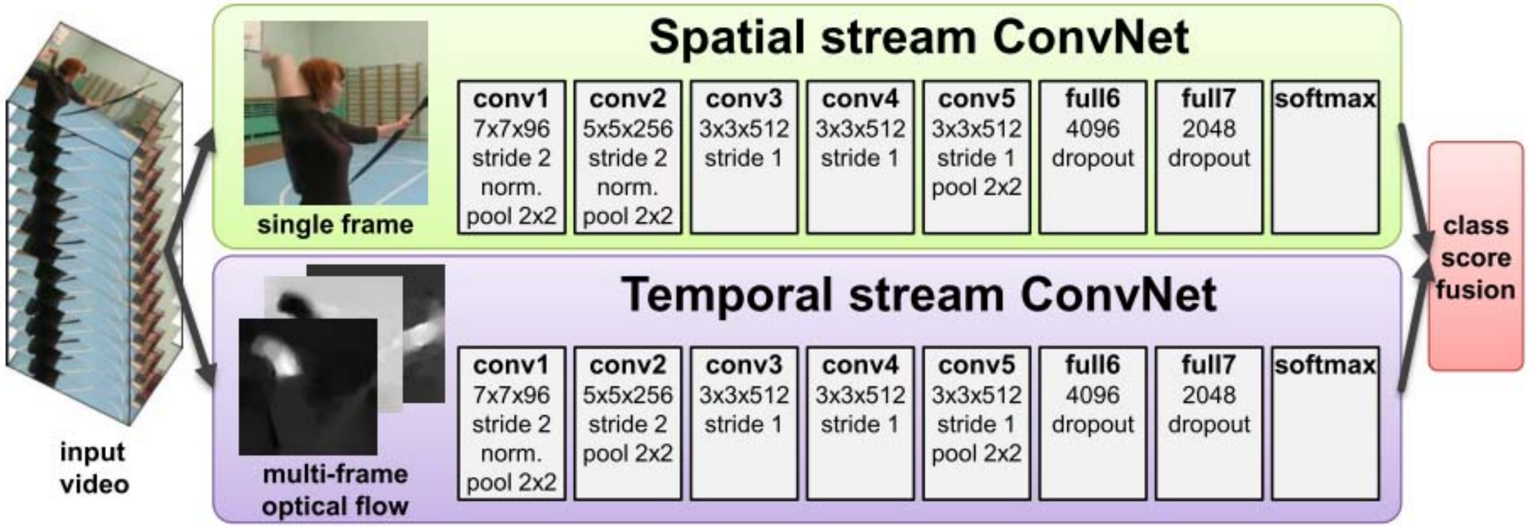


Quo vadis action recognition? A new model and the Kinetics dataset [Carreira et al. CVPR17]



Recent CNN methods

Two-Stream Convolutional Networks
for Action Recognition in Videos
[Simonyan and Zisserman NIPS14]



Method

50,000 ft

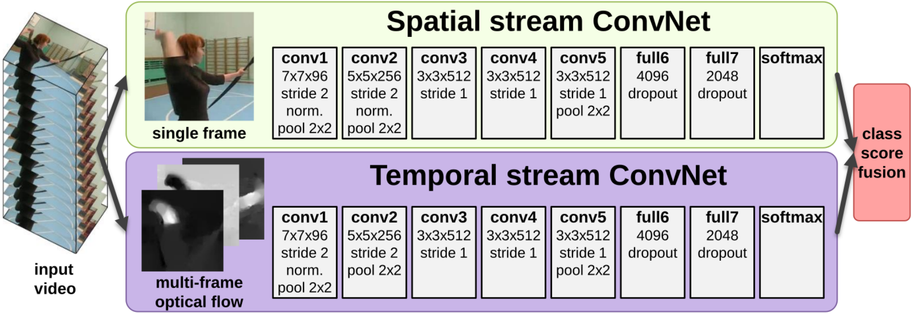
Hyposthesis

The human brain uses separate pathways to recognize objects and motions.

Idea

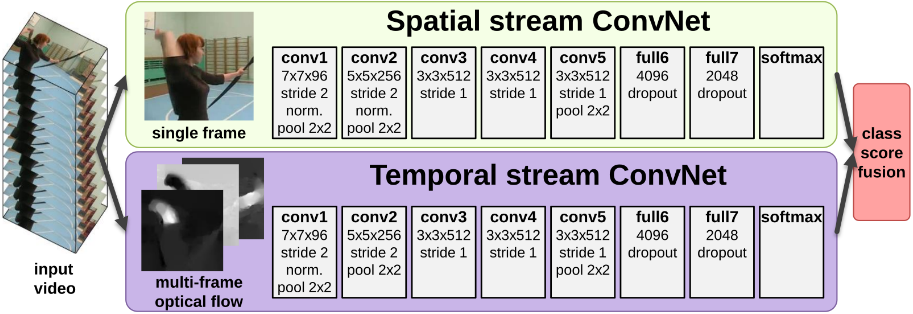
Make a network that mimics this strategy.

Network Architecture I



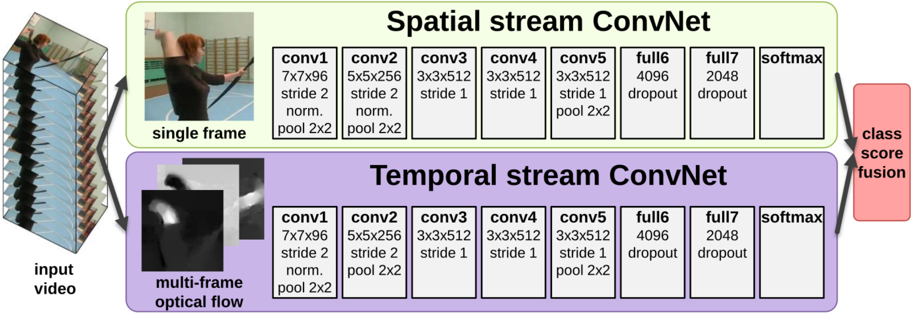
- Still images go into **Spatial Network**
  - Input is a single frame
  - CNP-C**N**P-C-C-CP-FD-FD-S

Network Architecture II



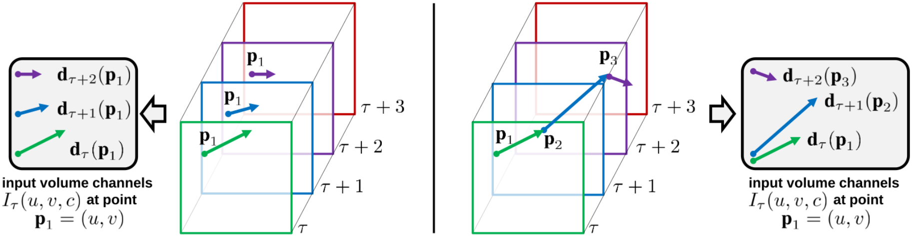
- Flow description goes into **Temporal Network**
  - CNP-CP-C-C-C-CP-FD-FD-S
  - Input is a stack of flow for L frames

Network Architecture III



- No combination of layer outputs up until output layer
  - Outputs of both networks are class scores
  - Combination by averaging or linear SVM
    - Combination via F-Layer had problems with overfitting

Flow Stacking I



Two methods to build input for flow network:

- [u, v, τ:$\tau + L$] describes flow at point [u, v] over time
  - i.e. use flow directly as input
- [u, v, τ:$\tau + L$] describes trajectory starting at [u, v, τ]

Flow Stacking II

Also use Backward Flow

- Also calculate backward flow
- Use $\frac{L}{2}$ frames forward and backward each

Camera Movement

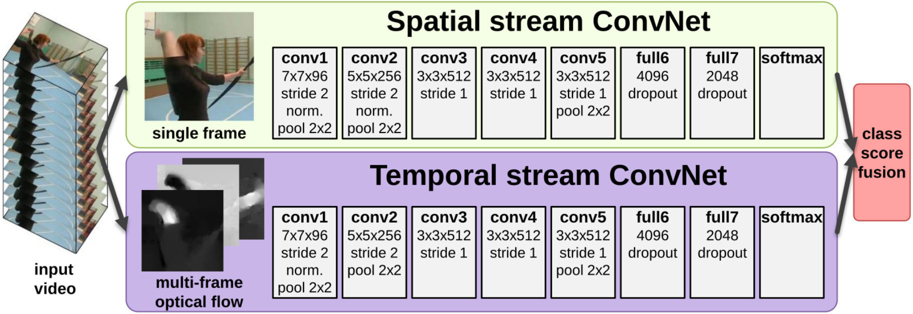
- Subtract mean flow for simple camera movement correction

Multitask Learning

- Use more than one dataset for training
- One softmax output layer per dataset
- Combine loss functions
    - Loss for videos of “other” datasets is zero
    - Sum up loss/gradient across batch/training set

Implementation Details

Implementation Details: Networks



- ReLUs
- Max-pooling on 3×3, stride 2
- Local Response Normalization [1]
  - Normalize activation by sum of activations of “neighbouring” filters

[1] Krizhevsky, Sutskever, Hinton: ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Mini-batch SGD

- Momentum 0.9
- Batch size 256
- Learning rate
  - Full Training:
    - 10^{-2} for 50K $\rightarrow 10^{-3}$ for 20K $\rightarrow 10^{-4}$ for 10K
    - $\Rightarrow$ 80K iterations
  - Fine tuning:
    - 10^{-2} for 14K $\rightarrow 10^{-3}$ for 6K

Input Processing

- Select 256 random videos for each mini-batch
- Select random anchor frame for each videos
- Scale so that smaller spatial dimension is 256
- **Spatial net input:** Crop random 224×224 patch, flip, jitter
- **Temporal net input:**
  - anchor “stack of flow” with length $2L$ at chosen frame
  - crop random 224×224 tube
  - random flipping

Testing

- Sample 25 anchor frames at equally spaced times
  - For temporal net, extract stack of flow around
- Crop to corners & to center $\rightarrow 224 \times 224$
- Flip each image/tube horizontally
- $\Rightarrow 25 \cdot 2 \cdot 5 = 250$ inputs for each network
- Average over resulting class scores

The Rest

- Optical flow straight from OpenCV
  - Precomputed and stored in 8-bit resolution $\rightarrow$ only 27GB
- Patched Caffe to run on multiple graphics cards

Experiments & Results

Spatial Net

Evalutation on UCF101

- Training from scratch
  - Overfits

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

Spatial Net

Evalutation on UCF101

- Training from scratch
  - Overfits
- Pretrain on ILSVRC-2012, fine-tune on UCF101
  - Works, but careful about over regularizing!

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

Spatial Net

Evalutation on UCF101

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

- Training from scratch
  - Overfits
- Pretrain on ILSVRC-2012, fine-tune on UCF101
  - Works, but careful about over regularizing!
- Pretrain on ILSVRC-2012, re-train softmax layer
  - Works, use this from now on

Temporal Net

Evalutation on UCF101

- $L = 5$ much better than $L = 1$
- $L = 10$ a bit better yet
- Stacking largely irrelevant

Input configuration	Mean subtraction	
	off	on
Single-frame optical flow (L = 1)	-	73.9%
Optical flow stacking (1) (L = 5)	-	80.4%
Optical flow stacking (1) (L = 10)	79.9%	**81.0%**
Trajectory stacking (2) (L = 10)	79.6%	80.2%
Optical flow stacking (1) (L = 10) ,bi-dir.	-	**81.2%**

Implemented Slow Fusion[2]

- Yields 56% accuracy (in line with that paper)
- Conclusion: motion needs to be presented appropriately

[2] Large-scale Video Classification with Convolutional Neural Networks

Multi-task Learning of Temporal Net

Evalutation on HMDB-51

Training setting	Accuracy
Training on HMDB-51 without additional data	46.6%
Fine-tuning a ConvNet, pre-trained on UCF-101	49.0%
Training on HMDB-51 with classes added from UCF-101	52.8%
Multi-task learning on HMDB-51 and UCF-101	**55.4%**

- Using more data helps!
- At least in this direction
  - UCF101 alone: 81%
  - UCF101+HMDB-51: 81.**5**%

Combined Networks

Spatial ConvNet	Temporal ConvNet	Fusion Method	Accuracy
Pre-trained + last layer	bi-directional	averaging	85.6%
Pre-trained + last layer	uni-directional	averaging	85.9%
Pre-trained + last layer	uni-directional, multi-task	averaging	86.2%
Pre-trained + last layer	uni-directional, multi-task	SVM	**87.0%**

Figure: Fused Results on UCF101 Split 1

Observations

- Fusion improves on each individual network
- SVM is better than averaging
- Multi-task learning helps
- Bi-directional flow does *not* help

Comparison to State of the Art (Mean over Splits)

Method	UCF-101	HMDB-51
Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
Spatio-temporal HMAX network [11, 16]	-	22.8%
"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-
Spatial stream ConvNet	73.0%	40.5%
Temporal stream ConvNet	83.7%	54.6%
Two-stream model (fusion by averaging)	86.9%	58.0%
Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

- Spacial net: pretrained + last layer
- Temporal net: unidirectional stacked flow, centered, multi-task

- Each net individually better than "The Other Paper" \o/
- Combination even better
- But not quite state of the art on HMDB

Recent CNN methods

Learning Spatiotemporal Features with 3D Convolutional Networks [Tran et al. ICCV15]

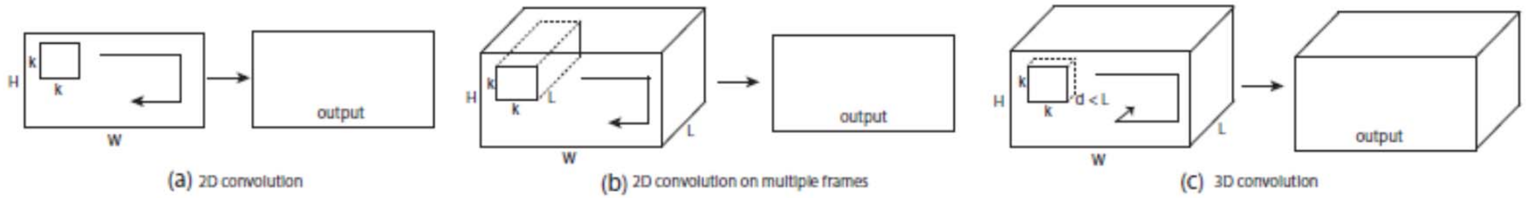
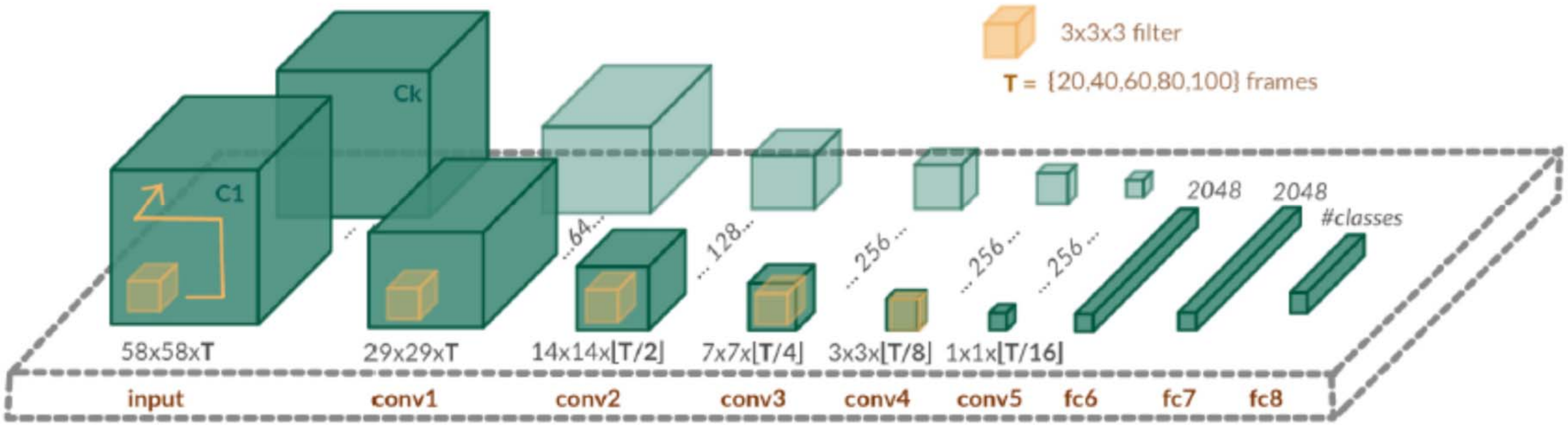
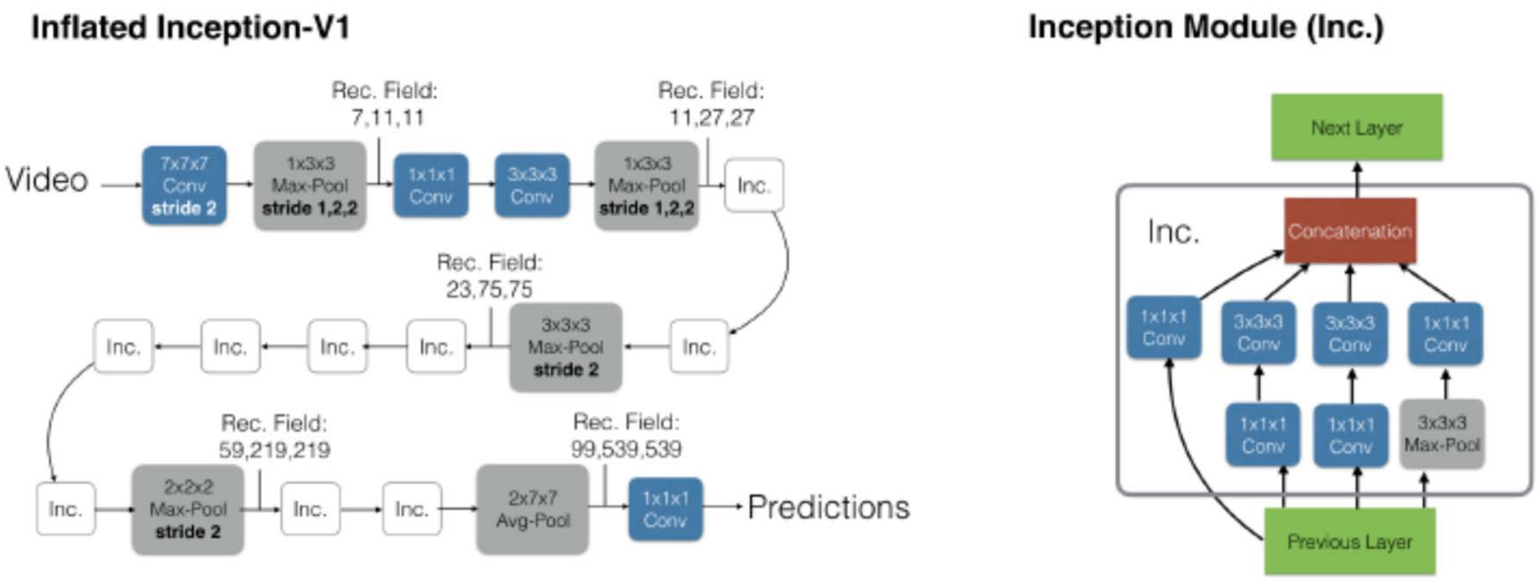


Figure 1. **2D and 3D convolution operations.** a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving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



Recent CNN methods

Quo vadis, action recognition? A new model and the Kinetics dataset
[Carreira et al. CVPR17]



Pre-training on the large-scale Kinetics dataset 240k training videos
→ significant performance grain

Summary

- 3D convolution capture spatio-temporal dynamics well
- Importance of sufficient training data