ICIP'17

4D Effect Classification by Encoding CNN Features

Thomhert S. Siadari[1], Mikyong Han[2], Hyunjin Yoon[1,2]

thomhert@etri.re.kr

Korea University of Science and Technology[1]

Electronics and Telecommunications Research Institute[2]

UST 과학기술연합대학원대학교 UNIVERSITY OF SCIENCE & TECHNOLOGY

ETRI Electronics and Telecommunications Research Institute

Motivation

Source: https://www.youtube.com/watch?v=kGW-vNsxkv4

- Effect type: motion, vibration, wind, flash, scent, snow, etc.
- Chairs: move, vibrate, shake

+ Increase immersive experience

- Manual production for effect labeling

Problem definition

- Action classification

Cricket Bowling

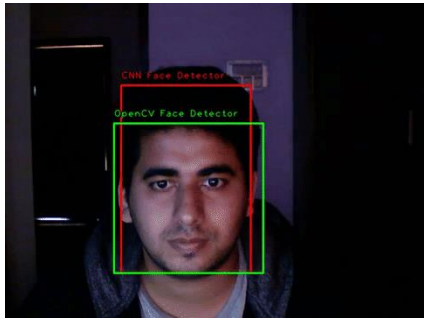
Clean & Jerk

- 4D effect classification

Effect or Non-Effect

Convolutional Neural Networks (CNNs)

- CNNs have wide applications, such as in image recognition, video recognition, recommender systems, and natural language processing.



Face detection in video (https://www.youtube.com/watch?v=Zedlhu_QCjE)

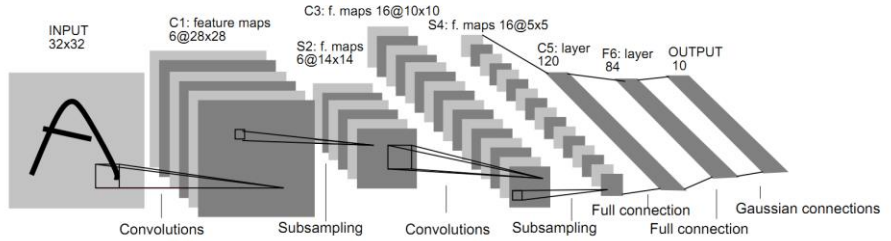
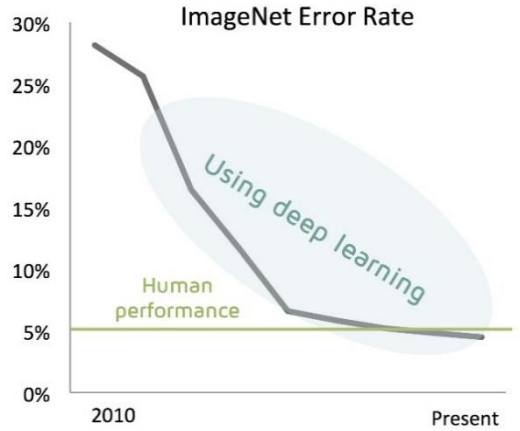


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

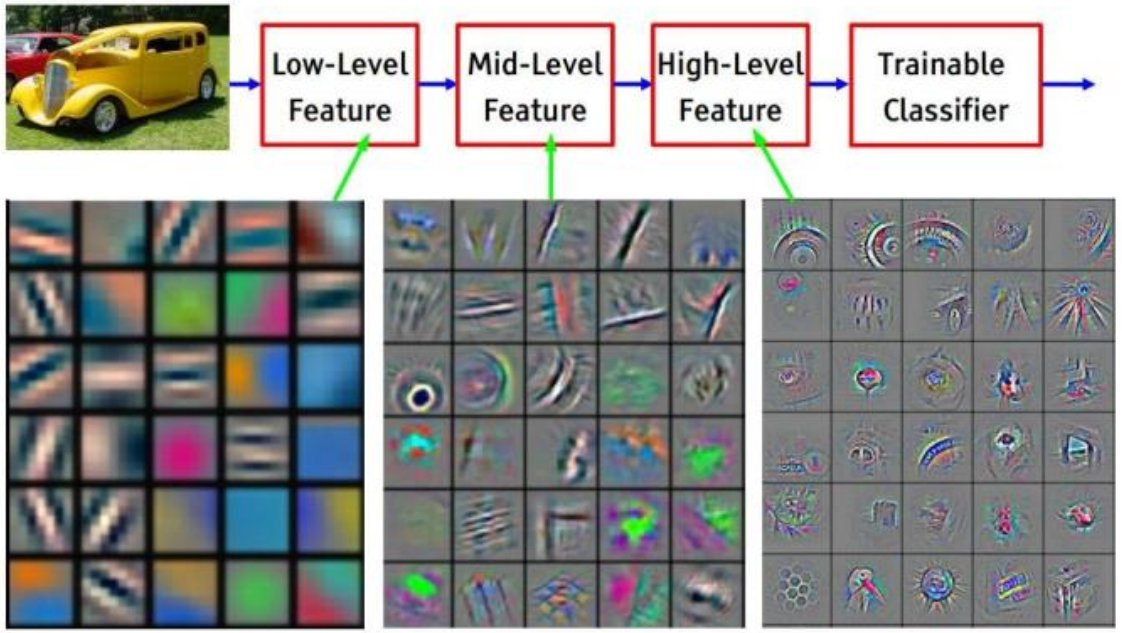
- One of ImageNet competition task is classify images from ImageNet dataset accurately.
  - Class: 1,000
  - #Training images: 1.2M
  - #Validation images: 50K



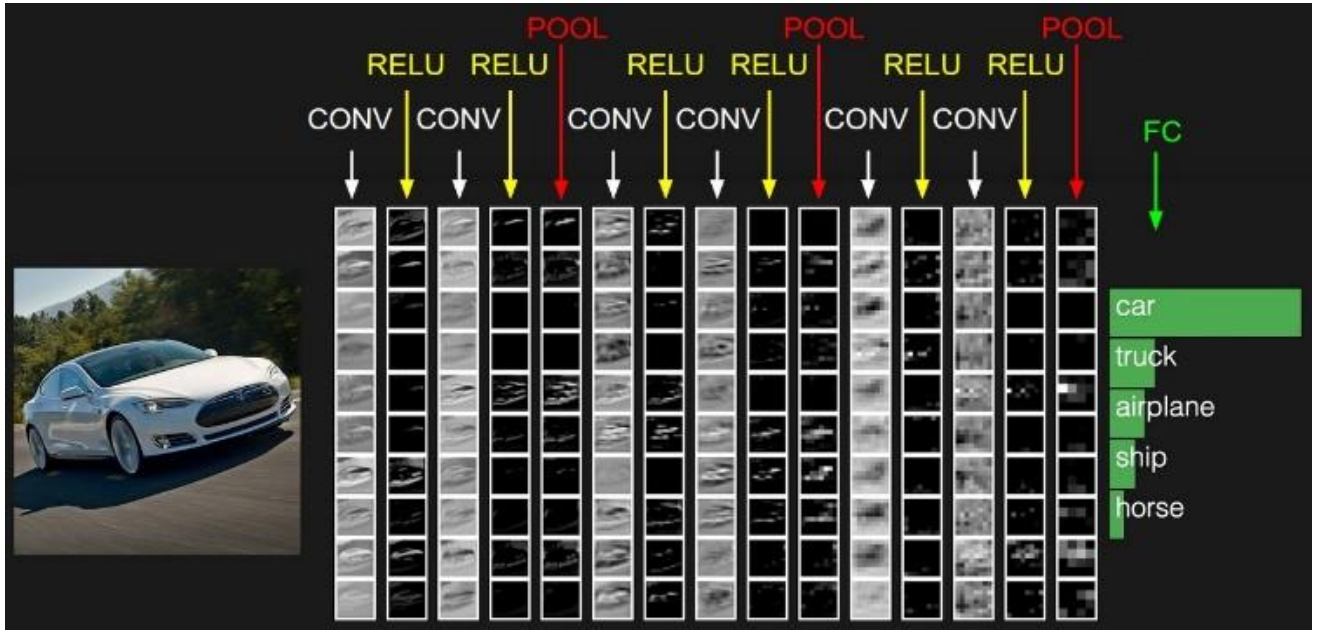
Imagenet Challenge History (https://www.nervanasys.com/deep-learning-and-the-need-for-unified-tools/)

CNNs: feature representation

- Using CNNs for feature extractor



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



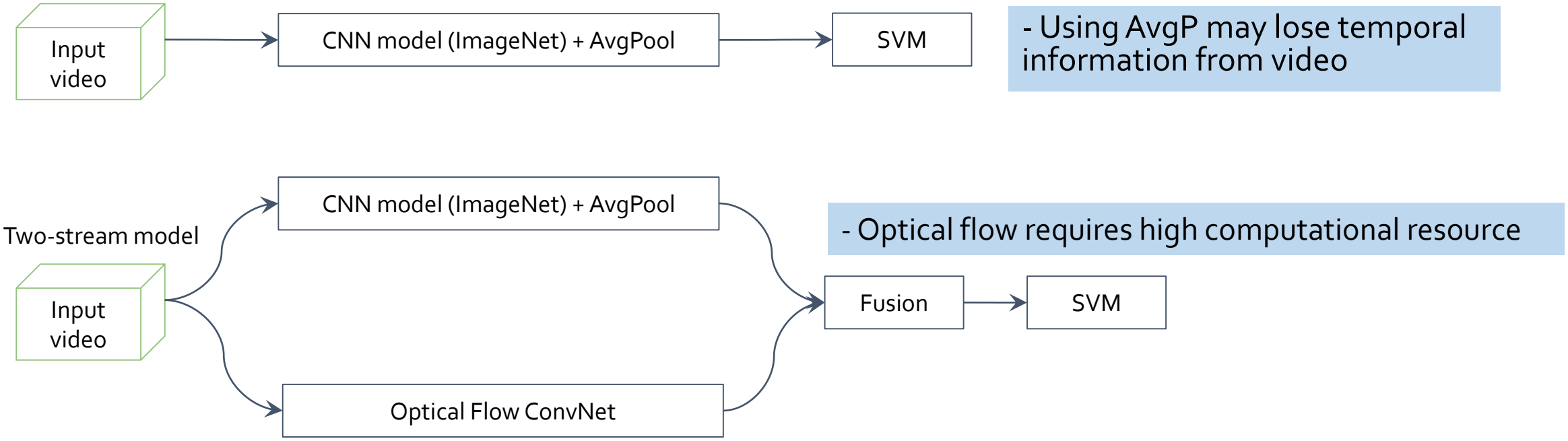
(reference: http://cs231n.github.io/convolutional-networks/)

Goal

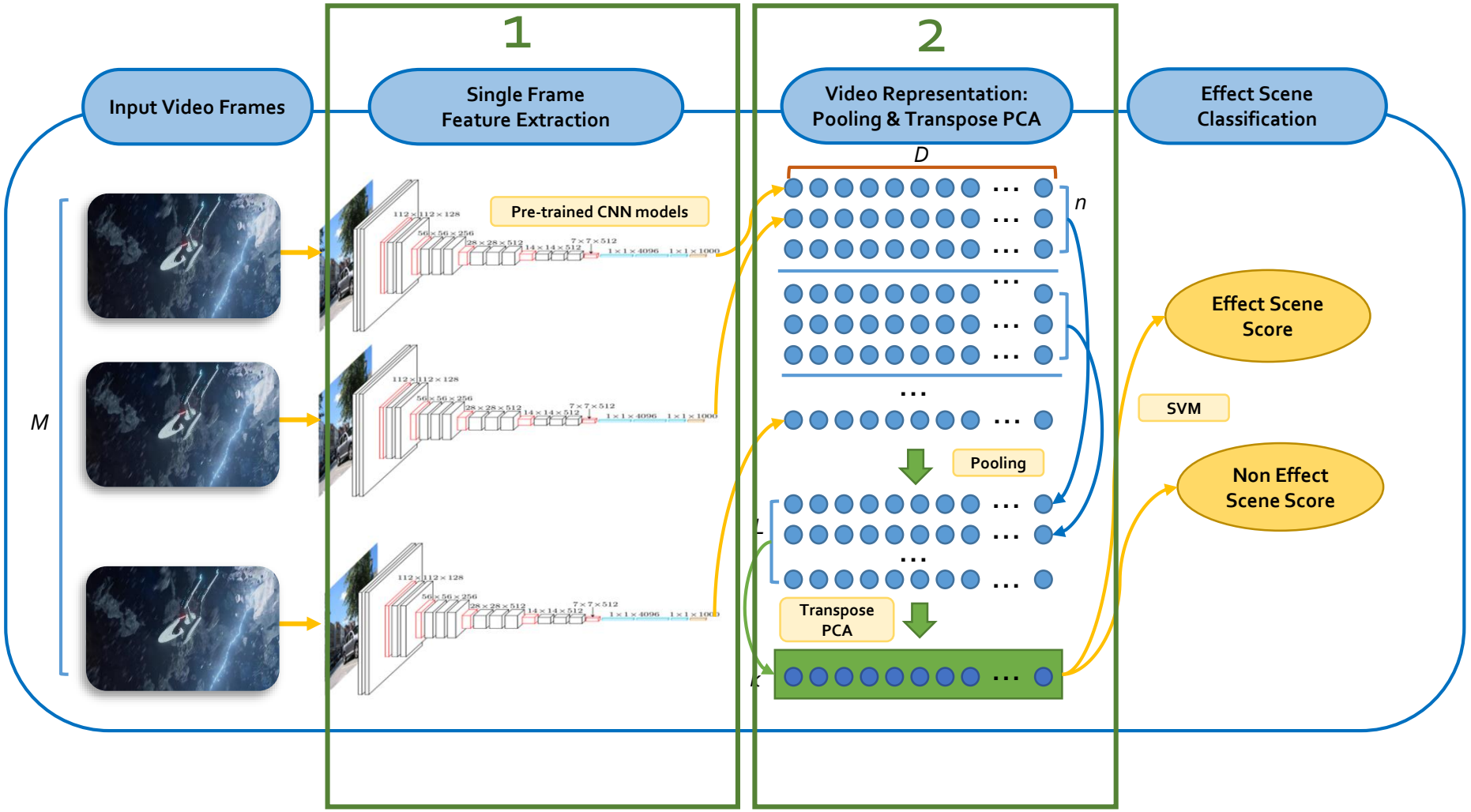
- Can we automate the human-labor process of 4D effect annotation?
- Can we understand videos and classify those videos into their effect or non-effect?

Research approach so far..

- How do researchers address video classification using pre-trained CNN models?



Proposed method



Step 1: Feature extraction

CaffeNet

```
data    (50, 3, 227, 227)
conv1   (50, 96, 55, 55)
pool1   (50, 96, 27, 27)
norm1   (50, 96, 27, 27)
conv2   (50, 256, 27, 27)
pool2   (50, 256, 13, 13)
norm2   (50, 256, 13, 13)
conv3   (50, 384, 13, 13)
conv4   (50, 384, 13, 13)
conv5   (50, 256, 13, 13)
pool5   (50, 256, 6, 6)
fc6     (50, 4096)
fc7     (50, 4096)
fc8     (50, 1000)
prob    (50, 1000)
```

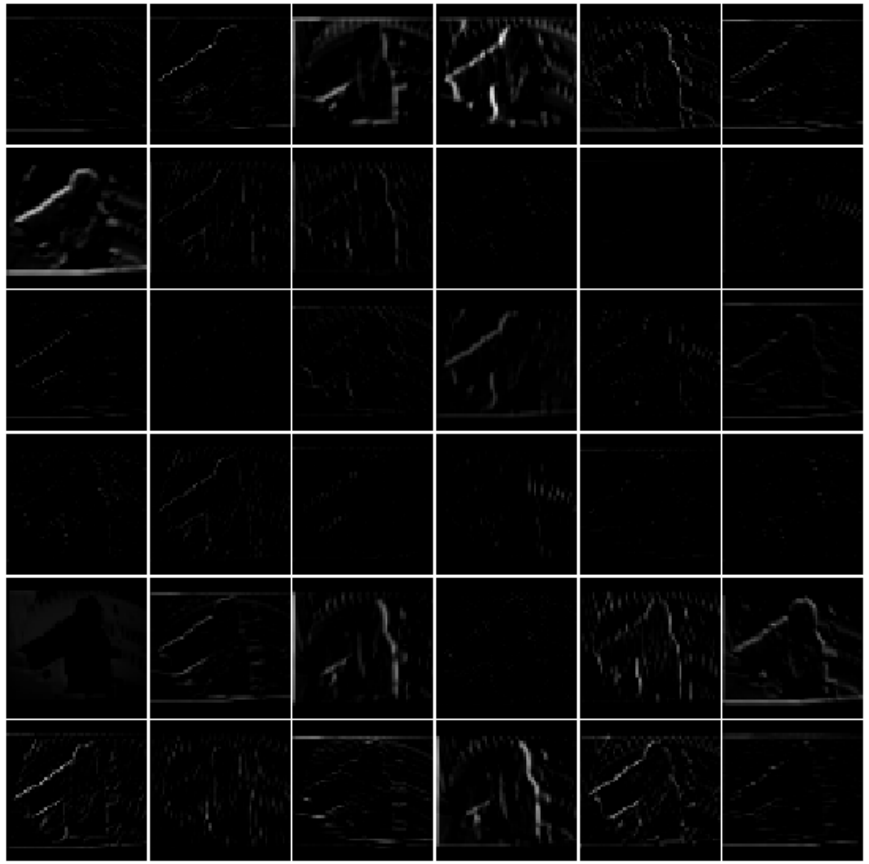
VGGNet

```
data    (2, 3, 224, 224)
conv1_1 (2, 64, 224, 224)
conv1_2 (2, 64, 224, 224)
pool1   (2, 64, 112, 112)
conv2_1 (2, 128, 112, 112)
conv2_2 (2, 128, 112, 112)
pool2   (2, 128, 56, 56)
conv3_1 (2, 256, 56, 56)
conv3_2 (2, 256, 56, 56)
conv3_3 (2, 256, 56, 56)
pool3   (2, 256, 28, 28)
conv4_1 (2, 512, 28, 28)
conv4_2 (2, 512, 28, 28)
conv4_3 (2, 512, 28, 28)
pool4   (2, 512, 14, 14)
conv5_1 (2, 512, 14, 14)
conv5_2 (2, 512, 14, 14)
conv5_3 (2, 512, 14, 14)
pool5   (2, 512, 7, 7)
fc6     (2, 4096)
fc7     (2, 4096)
fc8     (2, 1000)
prob    (2, 1000)
```

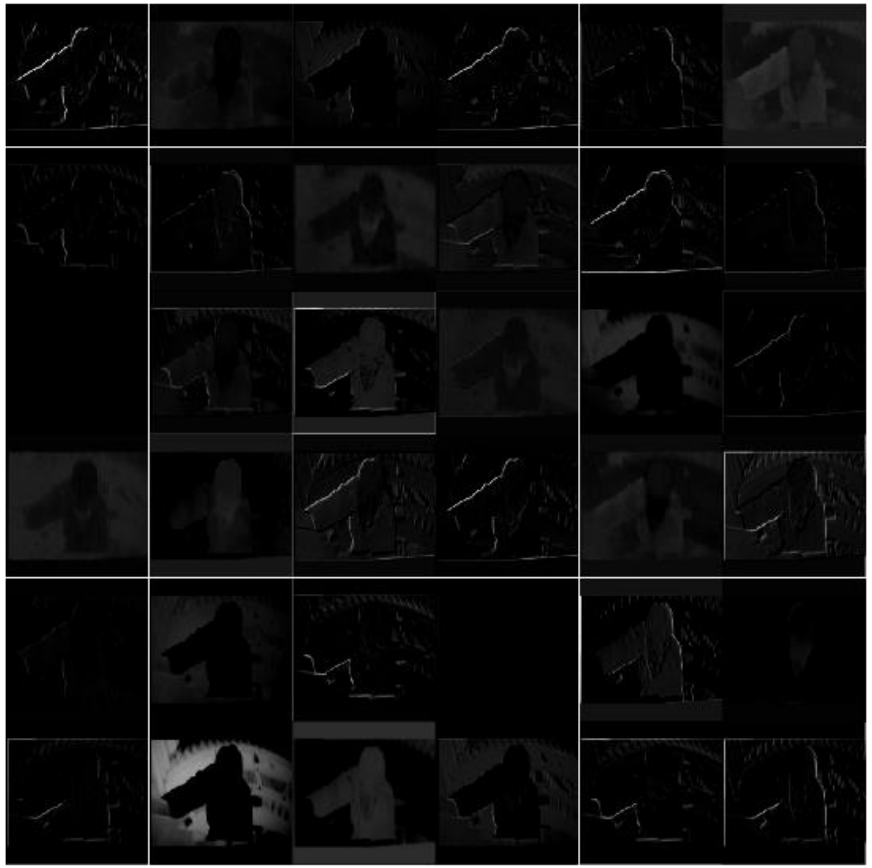
Visualization: an input frame

Features: First Conv Layer

CaffeNet

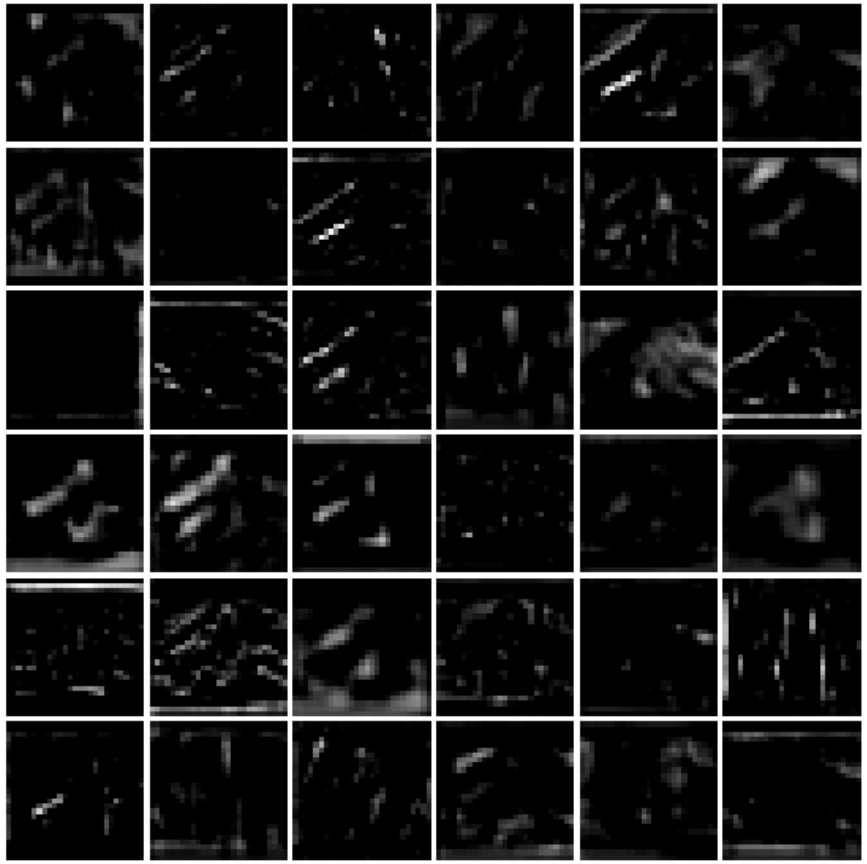


VGGNet

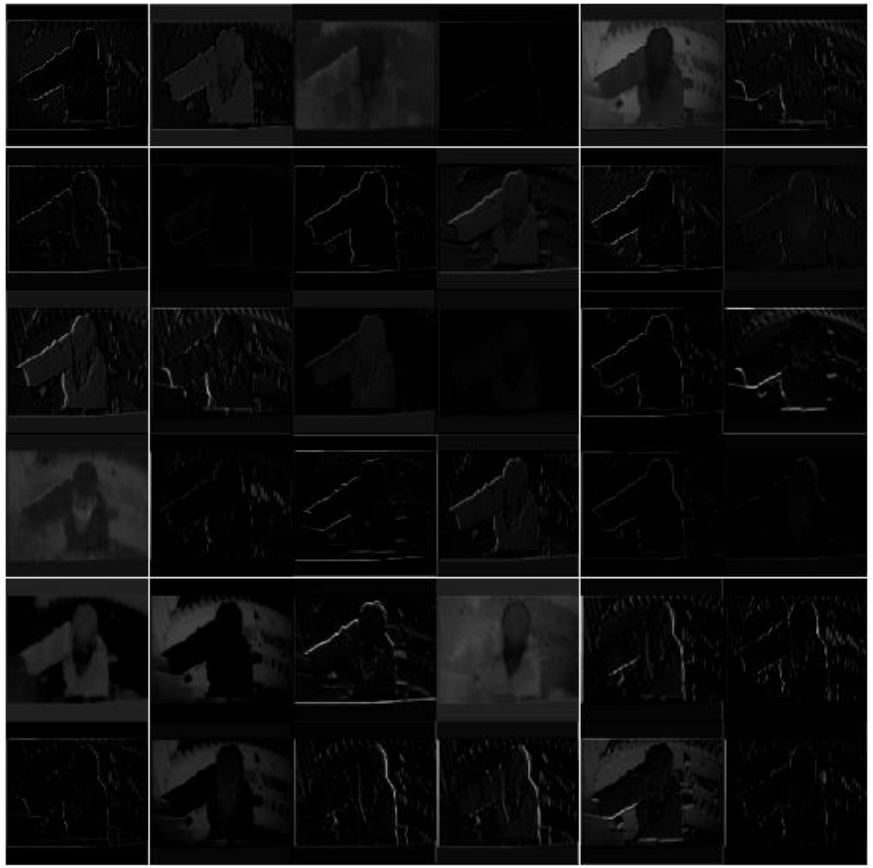


Features: Second Conv Layer

CaffeNet

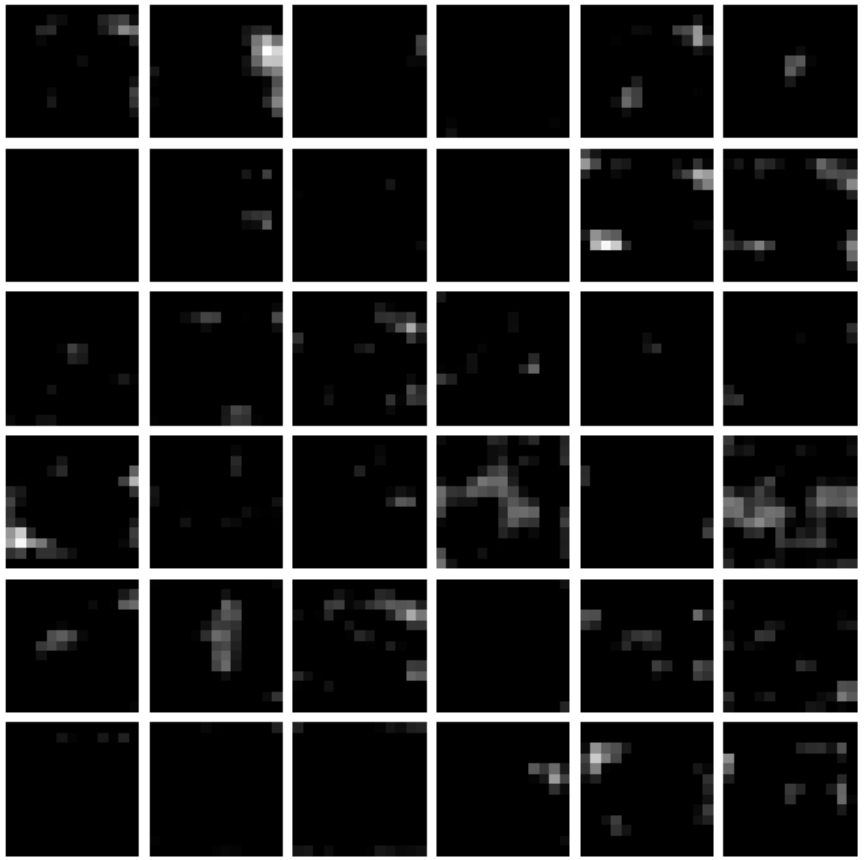


VGGNet

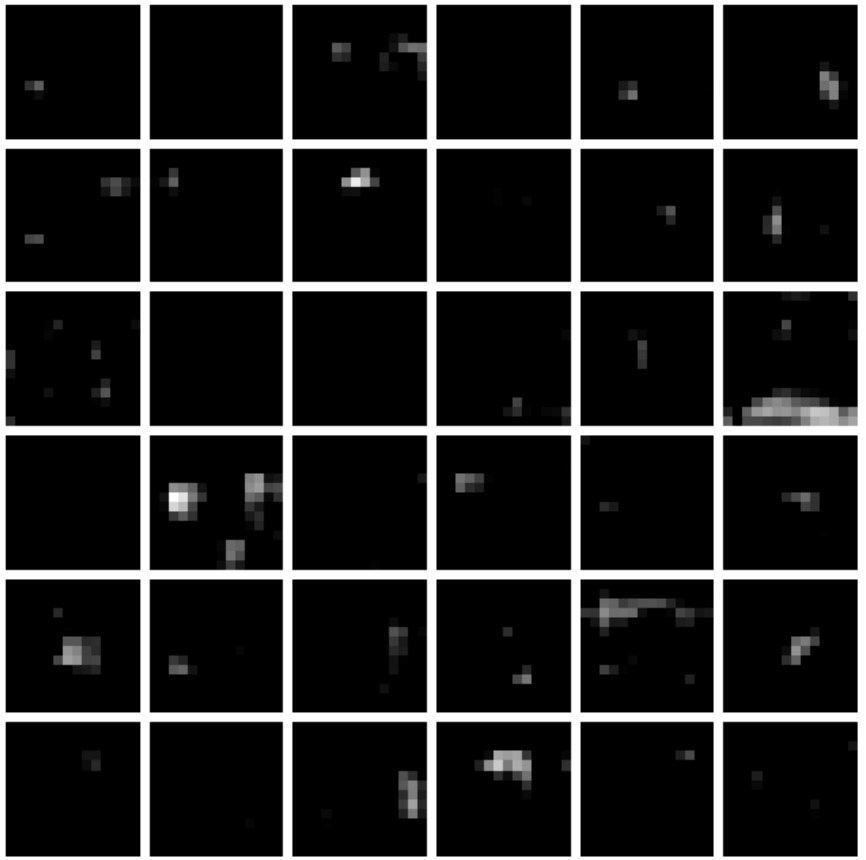


Features: Last Conv layer

CaffeNet



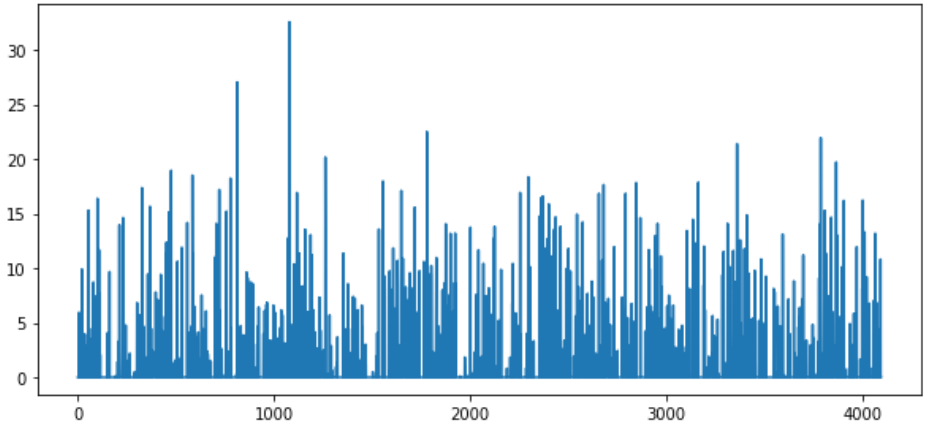
VGGNet



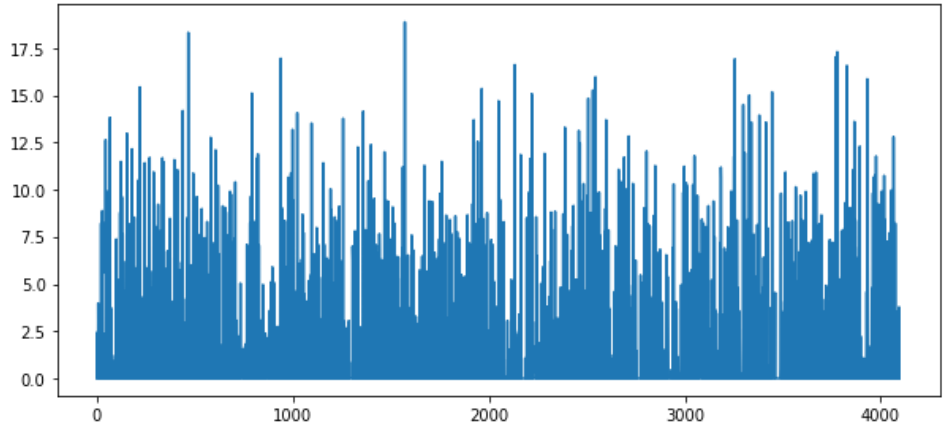
Step 1: feature extraction

- Fc6 feature maps as input for video-level represenation

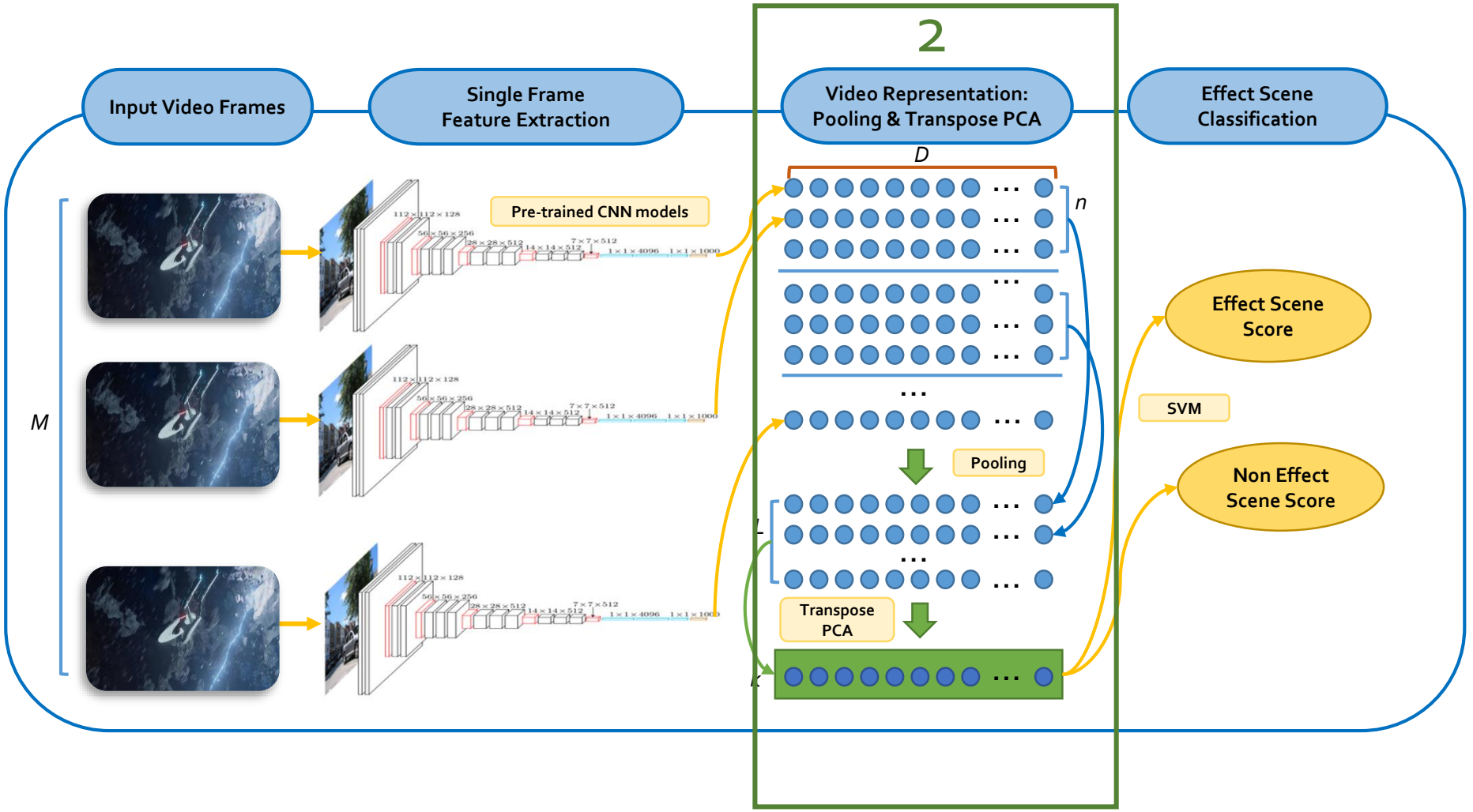
CaffeNet



VGGNet

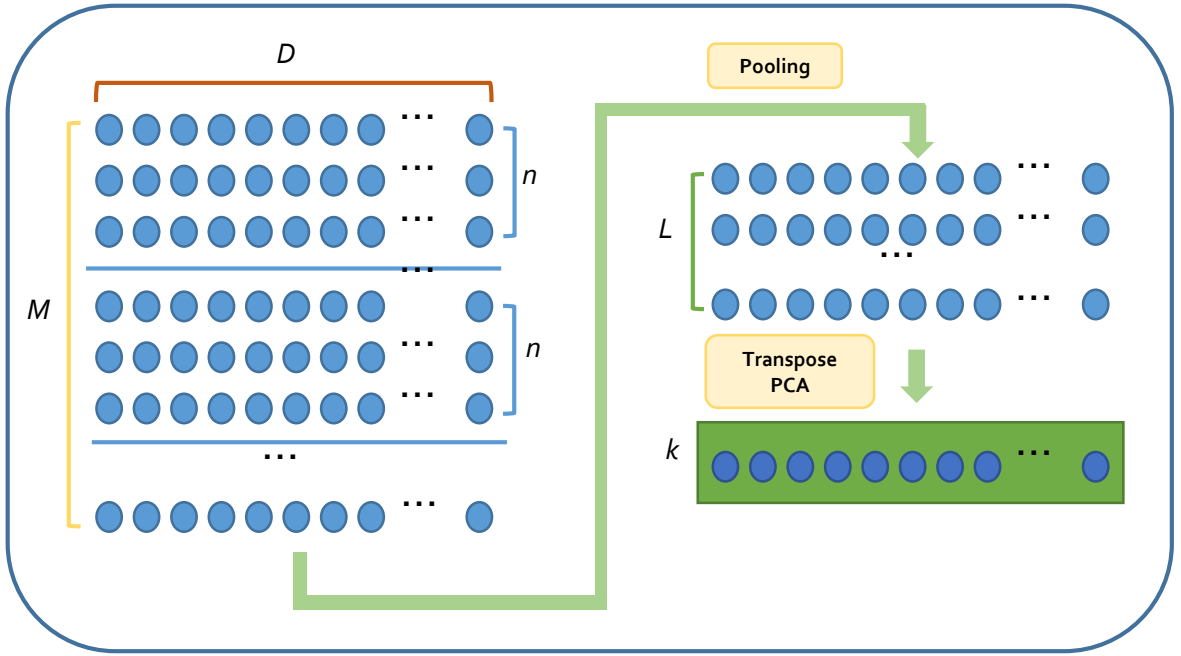


Step 2: video-level representation



Step 2: video representation

- Remember output of CNNs: feature maps 1x4096
- D = 4096
- M = video lengths
- n = variable to make multi-frame
- L = M/n
- k = number of components to keep



Performance Evaluation: Dataset

	Effect				Non-effect
	Motion	Vibration	Wind	Flash	
Number of video clips	137	75	93	42	238
Avg. frame number	611				329
Avg. length (s)	21				11

Performance Evaluation: setting

- CAFFE framework
- Python (numpy, scikit learn library)
- NVIDIA GPU GeForce TITAX X 12GB
- Linear SVM
- Excluding too short and too long videos
- Evaluation metrics: mean average precision and F1 score

Results(1)

- Architectures and features: binary classification (effect or non effect)

Model-Feature	mAP (%)	F1-score
VGGNet-fc6	63.87	0.64
VGGNet-fc7	61.53	0.63
CaffeNet-fc6	66.67	0.68
CaffeNet-fc7	63.63	0.69

- CaffeNet-fc6 has the highest score

Results(2)

- Video pooling representation: binary classification (effect or non effect)

Model-Feature	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet-fc6	63.87	66.76	0.64	0.67
CaffeNet-fc6	66.67	68.42	0.68	0.69

- Proposed approach achieved better result

Results(3)

- Binary vs multi-class classification

Model-Class	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet-binary	63.87	66.76	0.64	0.67
VGGNet-multi	37.83	41.1	0.35	0.41
CaffeNet-binary	66.67	68.42	0.68	0.69
CaffeNet-multi	31.29	33.72	0.29	0.34

Results(4)

- Fine-tuning attempt: binary classification (effect or non effect)

Model	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet	63.87	66.76	0.64	0.67
VGGNet-FT	68.44	69.46	0.66	0.67
CaffeNet	66.67	68.42	0.68	0.69
CaffeNet-FT	55.55	58.33	0.61	0.65

- Fine-tuning does not turn out well due to two reasons
  - Lack of dataset

Summary

- A framework to classify 4D effect videos
- Comprehensive experiments on different CNN architectures and layer-wise features for binary classification
  - CaffeNet vs VGGNet
  - Fc6 feature maps is better than fc7 feature maps
  - Multi-class classification attempt
  - Fine-tuning attempt
- A new video representation pooling to increase classification performance
  - Outperformed Average Video Pooling

Q & A?

- Reminder:
  - A framework to classify 4D effect videos
  - Comprehensive experiments on different CNN architectures and layer-wise features
    - CaffeNet vs VGGNet
    - Fc6 feature maps is better than fc7 feature maps
    - Multi-class classification attempt
    - Fine-tuning attempt
  - A new video representation pooling to increase classification performance
    - Outperformed Average Video Pooling