Multilabel Classification in Video

Andrei Nicolicioiu
andrei.nicolicioiu@gmail.com

Introduction - Multilabel classification

Ground Truth:	Predictions:
'Animal'	'Fishing'
'Outdoor recreation'	'Boat'
'Fishing'	'Outdoor recreation'
'River'	'Vehicle'
'Recreational fishing'	'Fish'
'Fishing rod'	'Animal'
	'Recreational fishing'
	'Lake'
	'River'
	'Fishing rod'
	'Fisherman'

Classification in Video

- Yue-Hei Ng, Joe, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4694-4702. 2015.

Beyond short snippets: Deep networks for video classificatio

- Yue-Hei Ng et al. [2015] - video classification
  - videos provides additional information through the temporal component
  - to understand the content of a video you have to combine information at multiple time steps
  - more computationally demanding

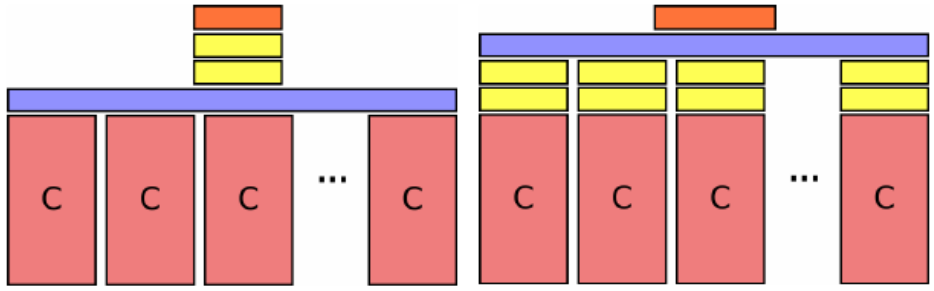
Beyond short snippets: Deep networks for video classificatio

- Yue-Hei Ng et al. [2015] - video classification
  - videos provides additional information through the temporal component
  - to understand the content of a video you have to combine information at multiple time steps
  - more computationally demanding
  - naive approach: use CNN to classify each frame and average the result

Beyond short snippets: Deep networks for video classificatio

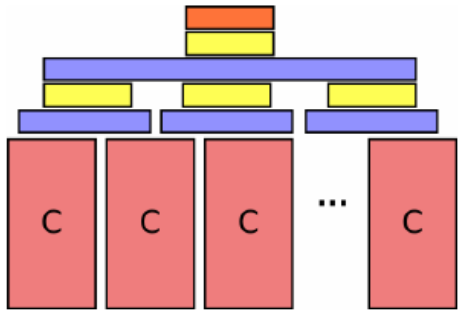
- Yue-Hei Ng et al. [2015] - video classification
  - videos provides additional information through the temporal component
  - to understand the content of a video you have to combine information at multiple time steps
  - more computationally demanding
  - naive approach: use CNN to classify each frame and average the result
  - need temporal information $=>$ aggregate frame features
    - apply different pooling strategies
    - recurrent network: learn how to integrate information over time

Pooling strategies

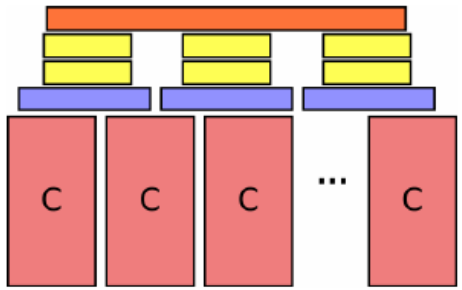


(a) Conv Pooling

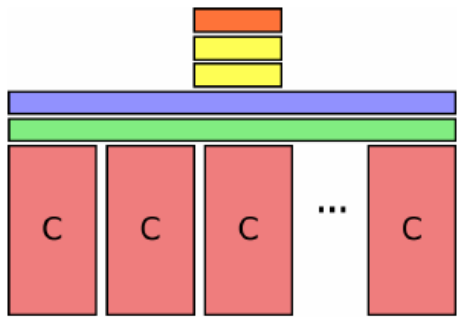
(b) Late Pooling



(c) Slow Pooling



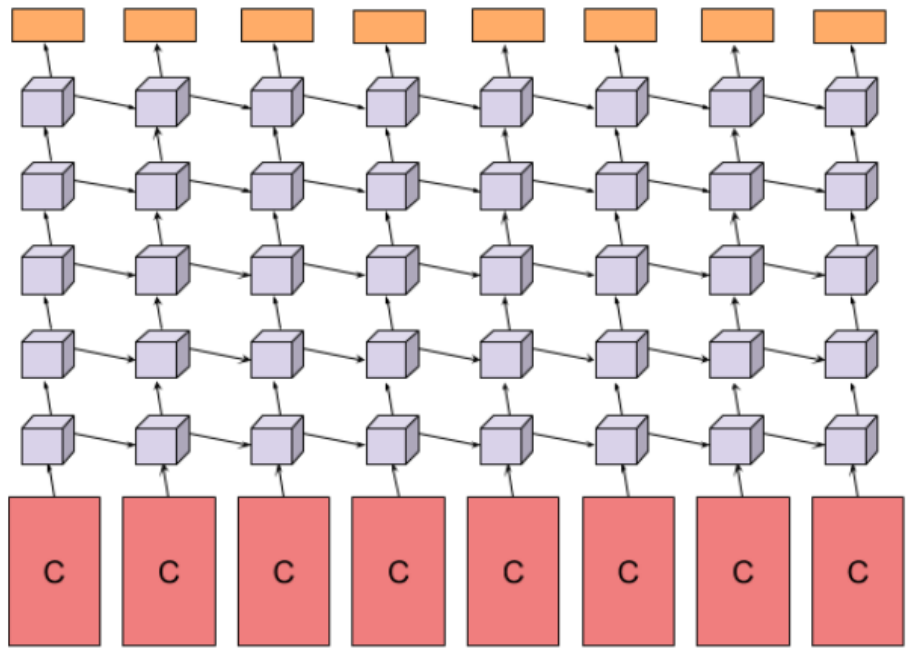
(d) Local Pooling



(e) Time-Domain Convolution

LSTM

- explicitly consider sequences of CNN activations and discover long-range temporal relationships
- use stack of 5 LSTM layers
- use a softmax classifier at each time step
- backpropagate label at every time frame
  - wight the gradients from 0 to 1 to emphasize the predictions in later frames
- at test time make weighted average of all predictions



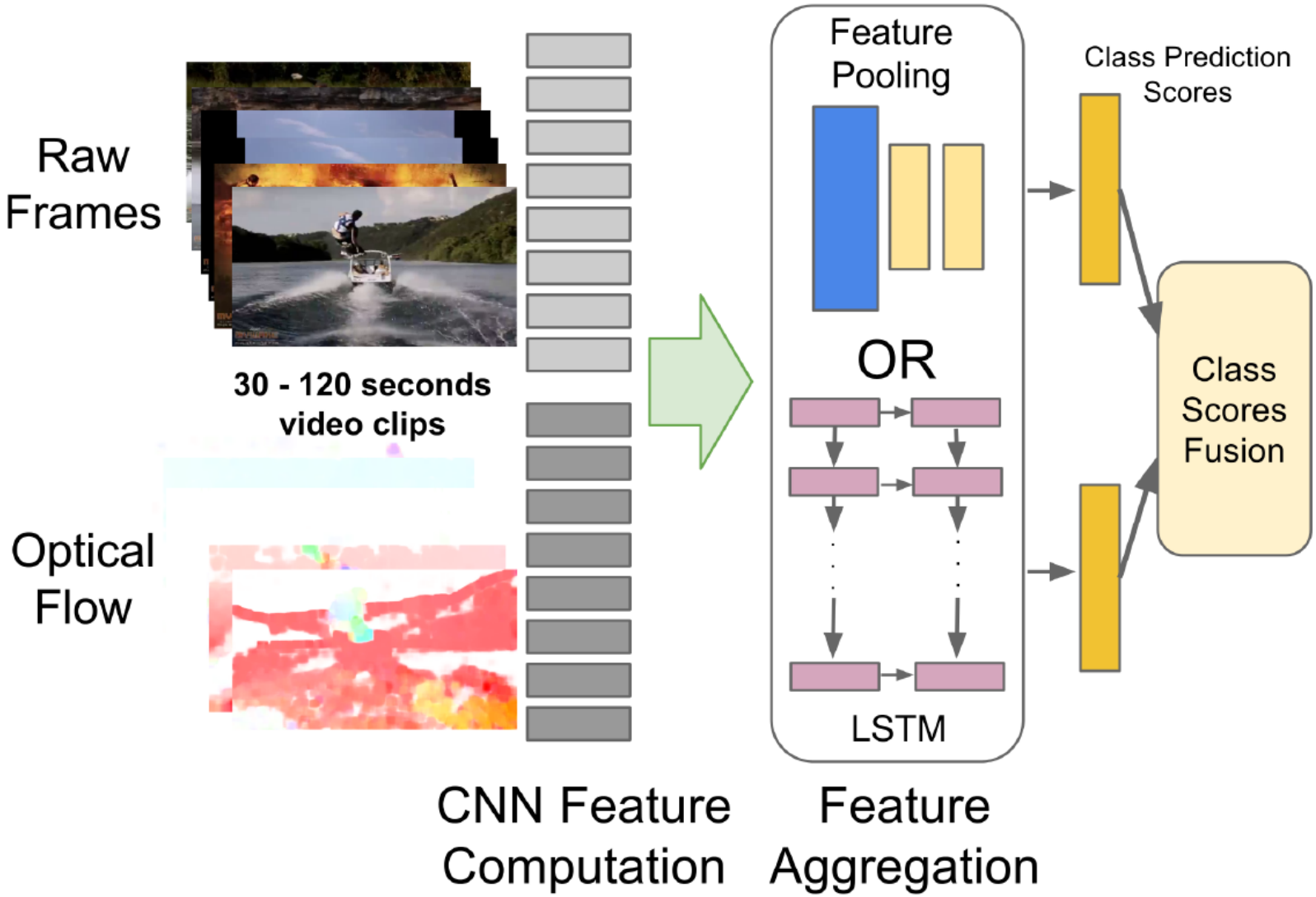
Optical Flow

- Images are sampled at low frame rate 1fps
  - Problem: no apparent motion information

Optical Flow

- Images are sampled at low frame rate 1fps
  - Problem: no apparent motion information
- Use optical flow
  - use 2 adjacent frames at 15 fps
  - represent optical flow with 3 channels and use the same network to make predictions
  - optical flow network can be initialized from one trained on images, with faster convergence

Architecture Overview



Training and inference

- Sports-1M dataset
  - 1.2 million YouTube sports videos
  - 487 classes, 5% of the videos have more than one class.
- UCF-101 Dataset
  - simpler dataset, camera motion is constrained
  - 13,320 videos of sports, musical instruments, and human-object interaction
  - 101 classes
- training on Sports-1M, testing on both

Results Sports-1M

Method	Clip Hit@1	Hit@1	Hit@5
Conv Pooling	**68.7**	**71.1**	**89.3**
Late Pooling	65.1	67.5	87.2
Slow Pooling	67.1	69.7	88.4
Local Pooling	68.1	70.4	88.9
Time-Domain Convolution	64.2	67.2	87.2

Results Sports-1M

Method	Frames	Clip Hit@1	Hit@1	Hit@5
LSTM	30	N/A	72.1	90.4
Conv pooling	30	66.0	71.7	90.4
	120	70.8	72.3	90.8

Method	Hit@1	Hit@5
LSTM on Optical Flow	59.7	81.4
LSTM on Raw Frames	72.1	90.6
LSTM on Raw Frames + LSTM on Optical Flow	73.1	90.5
30 frame Optical Flow	44.5	70.4
Conv Pooling on Raw Frames	71.7	90.4
Conv Pooling on Raw Frames + Conv Pooling on Optical Flow	71.8	90.4

Results UCF-101

Method	3-fold Accuracy (%)
Improved Dense Trajectories (IDTF)s [23]	87.9
Slow Fusion CNN [14]	65.4
Single Frame CNN Model (Images) [19]	73.0
Single Frame CNN Model (Optical Flow) [19]	73.9
Two-Stream CNN (Optical Flow + Image Frames, Averaging) [19]	86.9
Two-Stream CNN (Optical Flow + Image Frames, SVM Fusion) [19]	88.0
Our Single Frame Model	73.3
Conv Pooling of Image Frames + Optical Flow (30 Frames)	87.6
Conv Pooling of Image Frames + Optical Flow (120 Frames)	**88.2**
LSTM with 30 Frame Unroll (Optical Flow + Image Frames)	**88.6**

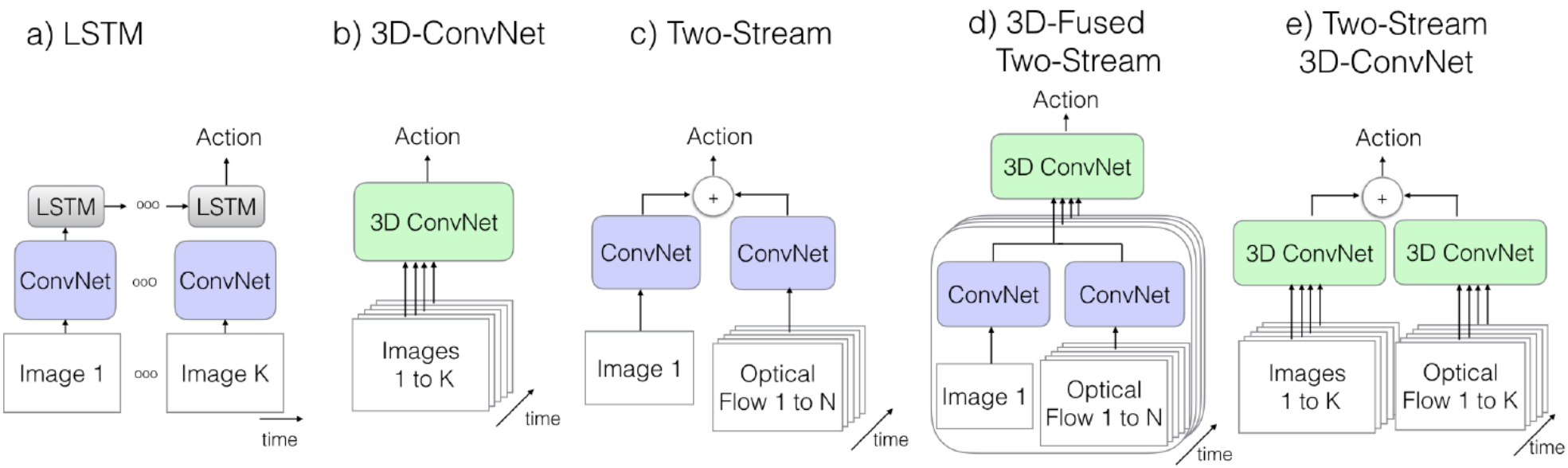
Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

- Carreira, Joao, and Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset." arXiv preprint arXiv:1705.07750 (2017).

Quo Vadis, Action Recognition?
A New Model and the Kinetics Dataset

- Carreira and Zisserman [2017]
- analyse state-of-the-art architectures on new Kinetics Human Action Video dataset
  - Kinetics: challenging dataset with with 400 human action classes and over 400 clips per class
- introduce Two-Stream Inflated 3D ConvNet (I3D)
  - network with 3D convolutional kernels initialised from pretrained 2D parameters
  - same architecture as established image classification networks

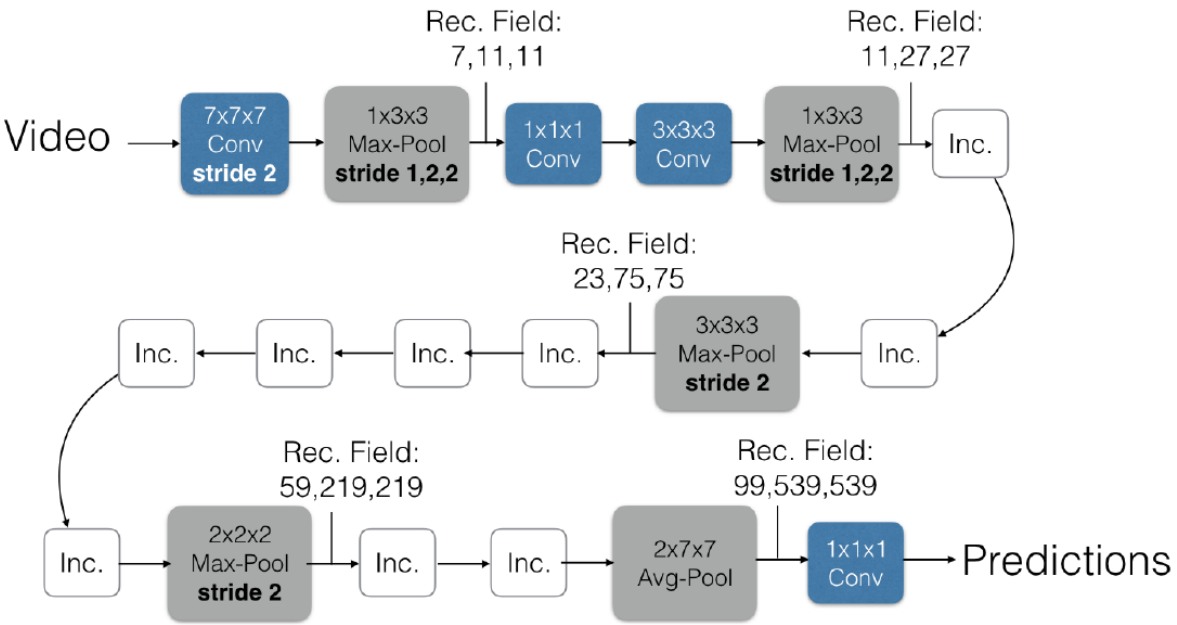
Old architectures



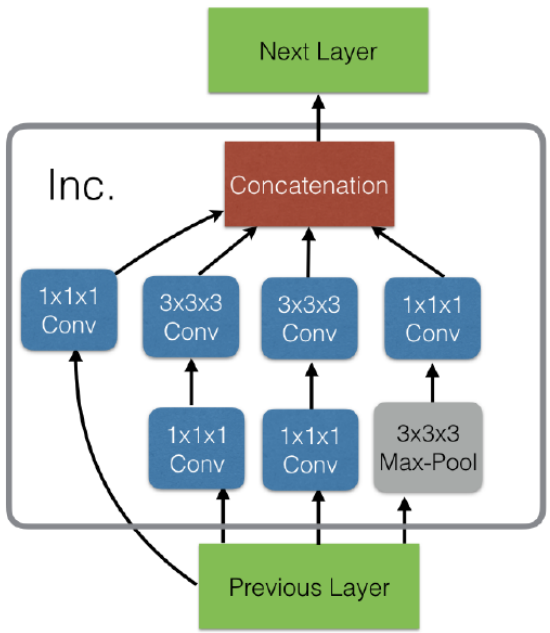
Inflated 3D ConvNet

- same architecture as Inception v1 but with 3D kernels
- 3D kernel initialized from of 2D pretrained on ImageNet
  - repeat the 2D kernels N times, rescale them by dividing by N

Inflated Inception-V1



Inception Module (Inc.)



Results

Architecture	UCF-101			
	Original	Fixed	Full-FT	Δ
(a) LSTM	81.0	81.6	82.1	-6%
(b) 3D-ConvNet	49.2	76.0	79.9	**-60.5%**
(c) Two-Stream	91.2	90.3	91.5	-3.4%
(d) 3D-Fused	89.3	88.5	90.1	-7.5%
(e) Two-Stream I3D	**93.4**	**95.7**	**96.5**	-47.0%

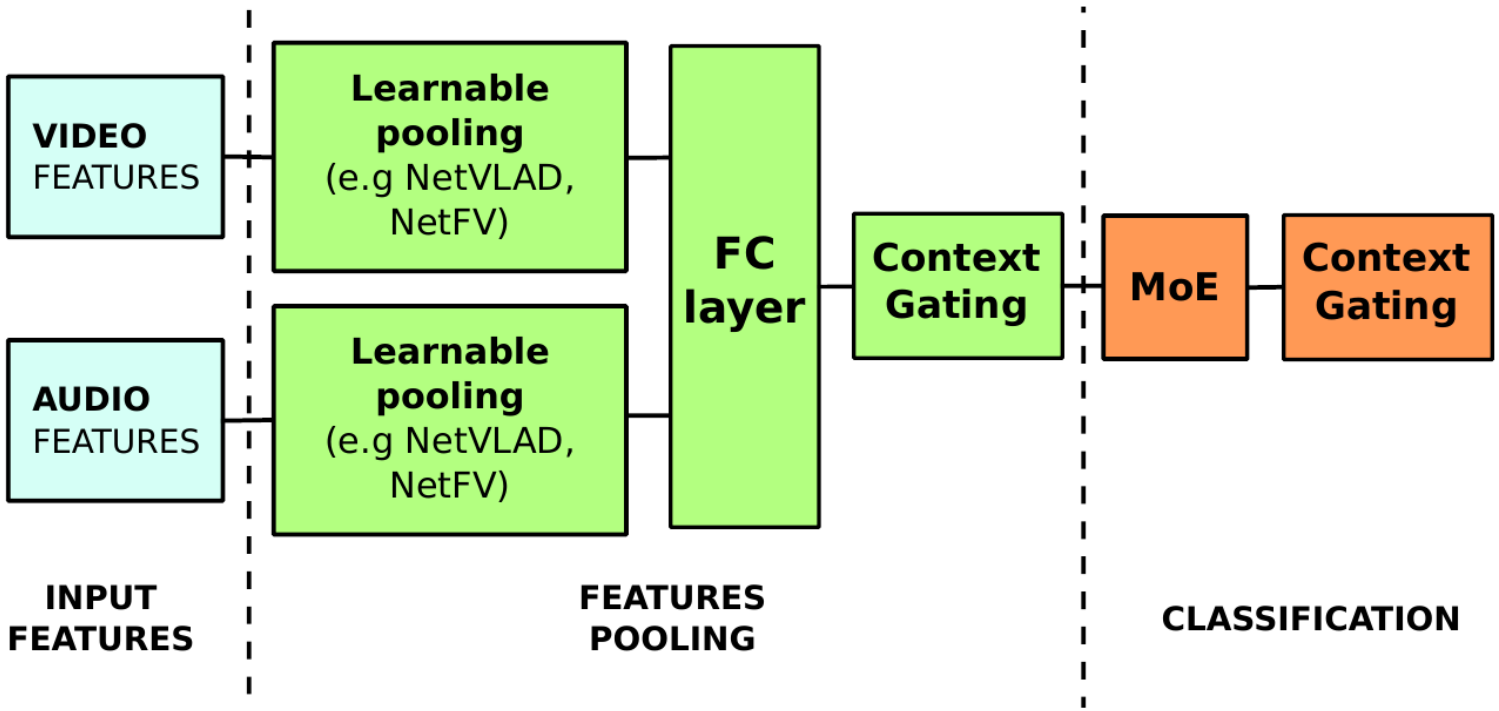
Learnable pooling with Context Gating for video classification

- Miech, Antoine, Ivan Laptev, and Josef Sivic. "Learnable pooling with Context Gating for video classification." arXiv preprint arXiv:1706.06905 (2017).

Learnable pooling with Context Gating for video classification

- Miech et al. [2017]
- winner of the Kaggle Youtube-8M Large-Scale Video Understanding challenge
  - Youtube-8M: collection of 8M videos, with features extracted at every frame by a Inception network
- explore combinations of learnable pooling techniques: Soft Bag-of-words, Fisher Vectors, NetVLAD, GRU and LSTM

Architecture overview

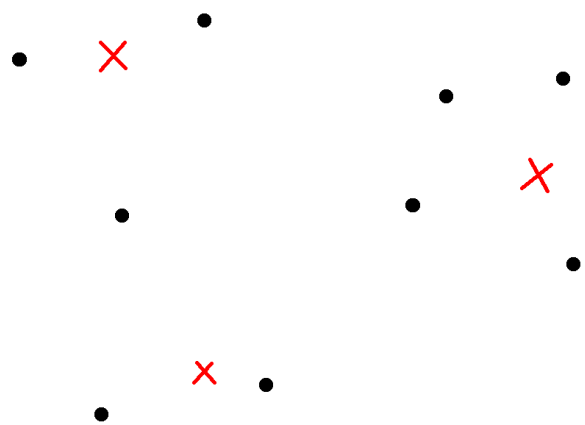


Feature Aggregation Methods

- in general: set of features $->$ vector representation
- in our case: get a vector representative of the video from frame features

Bag of words

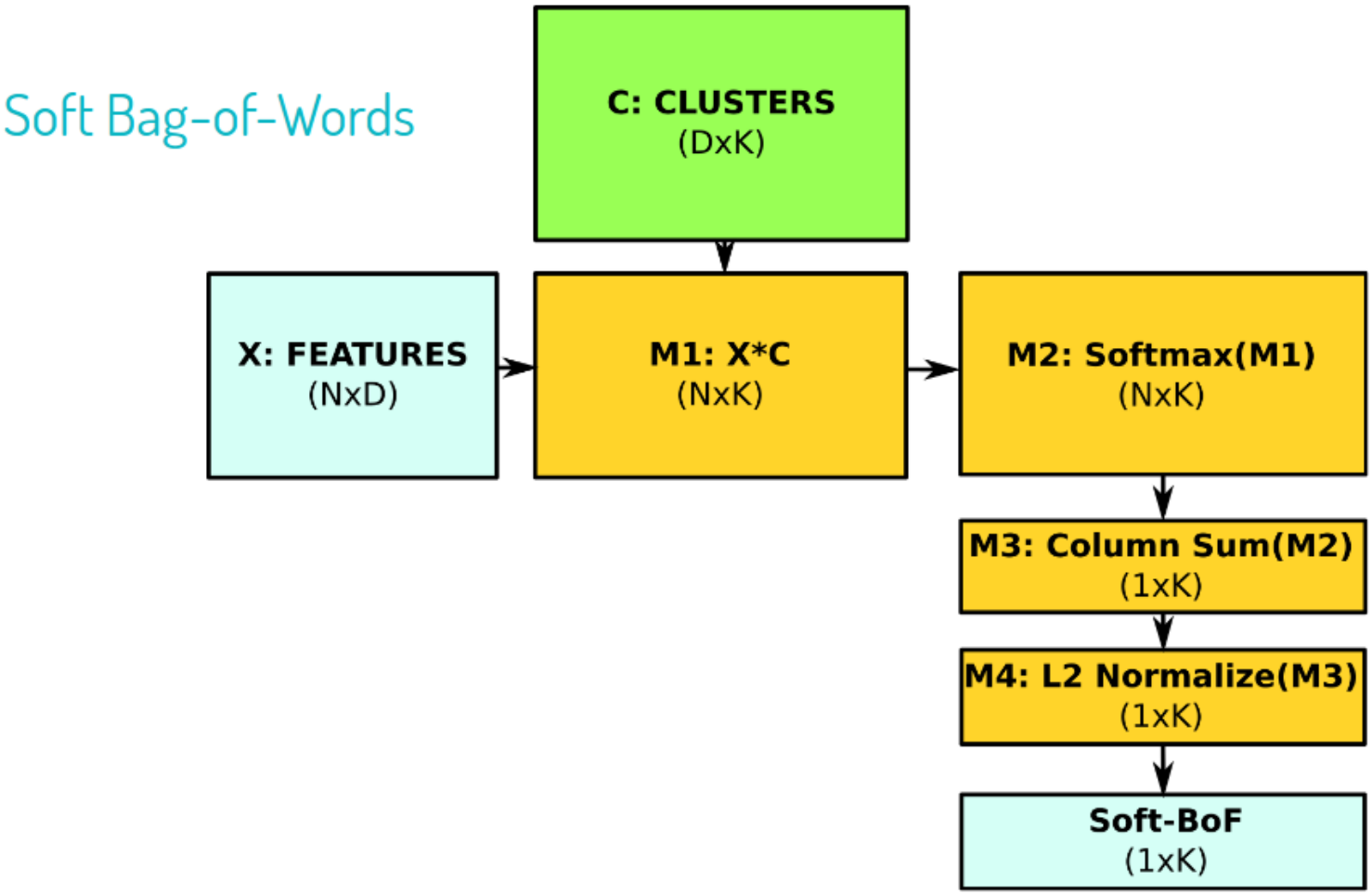
- choose K representative words from the space of features
- for each feature choose the nearest word
- count the features assigned to each word $=>$ K dimensional descriptor



Soft Bag of Words

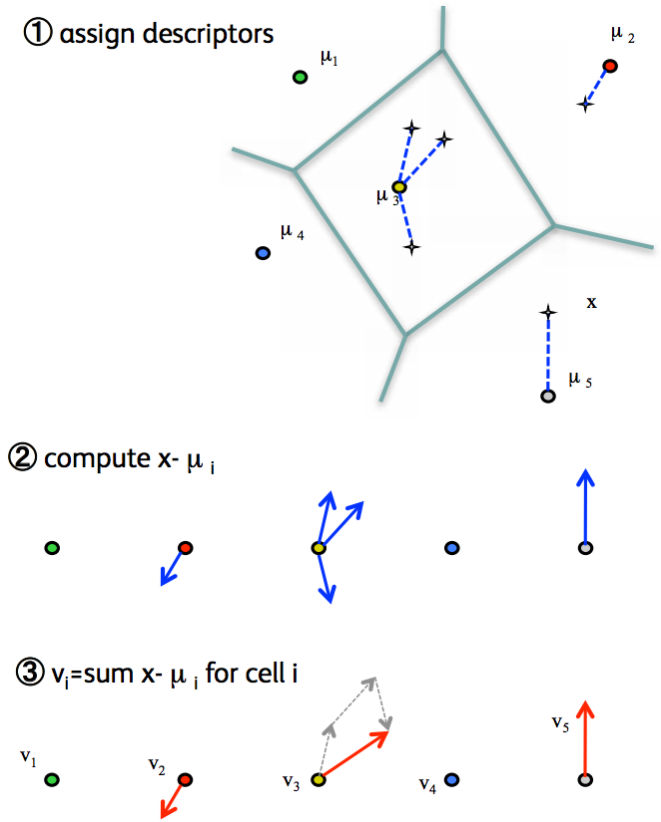
- $\gamma_t(k)$ assignment of feature x_t to cluster k
  - $\gamma_t(k) \in \{0, 1\}$ discrete case
  - $\gamma_t(k) \in [0, 1]$ continuous case = probability of $x_t \in$ cluster k
- $d[k] = \frac{1}{N} * \sum_t \gamma_t(k)$ K dimensional descriptor
- the cluster centers can be learned as a matrix M of dimension $N \times D$
- γ can be easily computed from M as the projection of the features to the clusters centers

Soft Bag of Words



Vlad Descriptor

- choose K representative words / cluster centers from the space of features
- compute residuals: differences between each feature and its cluster
- for each word add the residuals => K dimensional descriptor



[1]Image from: https://prateekvjoshi.com/2014/08/23/image-classification-

NetVlad Descriptor

Arandjelovic et al. [2016]

$$V(j,k) = \sum_{i=1}^{N} \gamma_i(k)(x_i(j) - c_k(j)) \tag{1}$$

$$\gamma_i(k) = \frac{e^{-\alpha\|x_i - c_k\|^2}}{\sum_{k'} e^{-\alpha\|x_i - c_{k'}\|^2}} \tag{2}$$

$$\gamma_i(k) = \frac{e^{2*\alpha c_k^T x_i - \alpha\|c_k\|}}{\sum_{k'} e^{2*\alpha c_{k'}^T x_i - \alpha\|c_{k'}\|}} = \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} \tag{3}$$

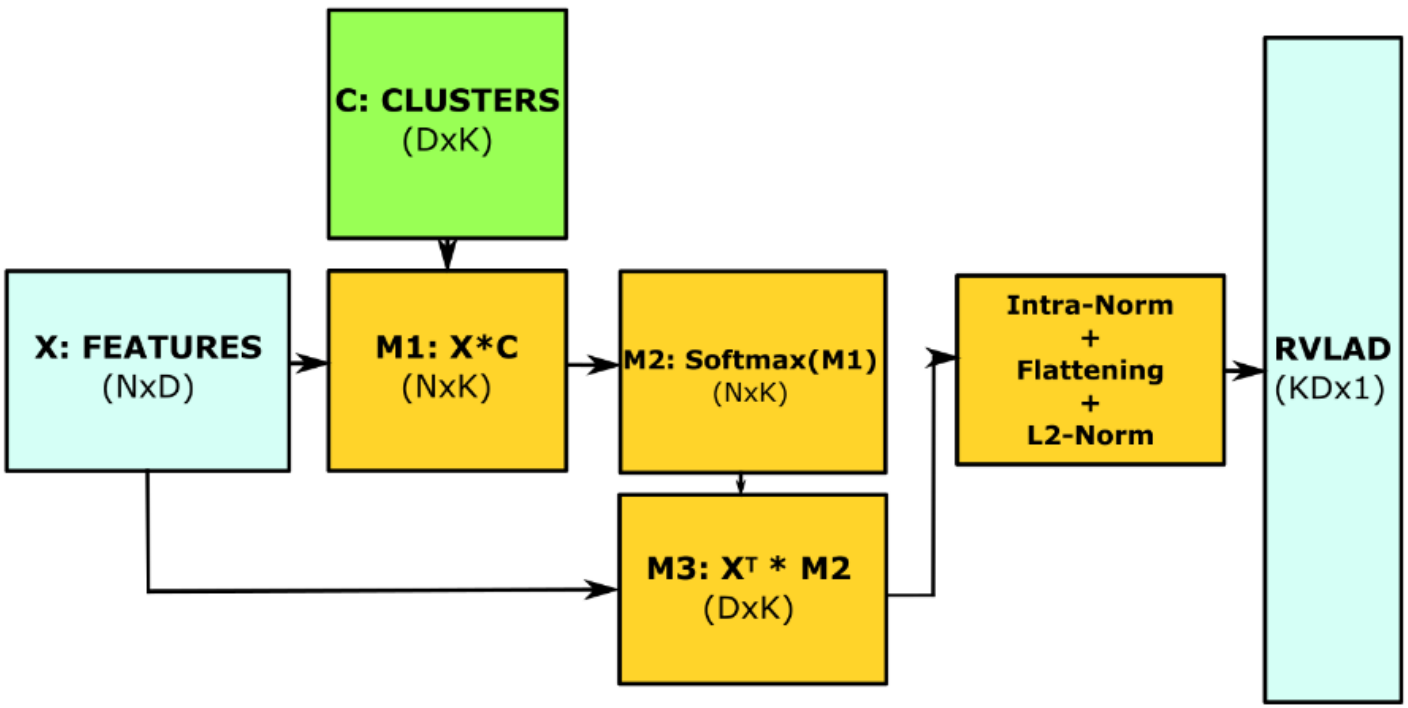
$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}}(x_i(j) - c_k(j)) \tag{4}$$

- w_k, b_k, c_k can be easily be learned

NetVlad Descriptor

- if we use the actual features instead of residuals
  - simplification of NetVLAD
  - less parameters
  - less computations

Residual-less NetVLAD
(NetRVLAD)



Fisher Vector

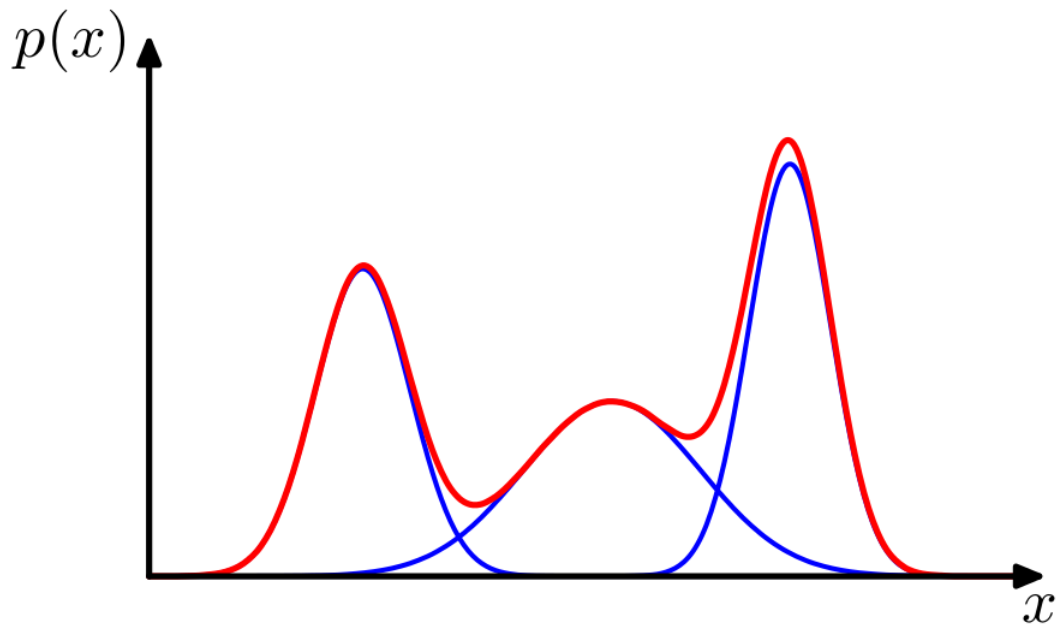
- Jaakkola and Haussler [1999] Perronnin and Dance [2007]
- use a generative model
- the log-loglikelihood $\mathcal{L} = \log p(X|\lambda)$ - how good the model fits the data
- the gradient of $\mathcal{L}$ w.r.t. to the parameters $\nabla_\lambda \log p(X|\lambda)$ expresses the direction that the parameters must change to best fit the data
- having two sets of features X_1 and X_2
  - if the features from set $X1$ are different from $X2$ then the direction of parameters changes are different for best fitting X_1 and X_2
  - $=>$ the direction is a good descriptor for distinguishing between sets of features
  - $=>$ use the normalised gradient $F^{-1/2}\nabla_\lambda \log p(X|\lambda)$, called Fisher vector

Mixtures of Gaussians - GMM

- mixtures of Gaussians is the superposition of K different Gaussian distribution
- a sample can be generated by each distribution

$$p(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \sigma_k) \tag{5}$$

- where w_k, μ_k, σ_k are mixing coefficients, mean and variance of the gaussians



Fisher Vector - GMM

- if we consider a Gaussian Mixture
  Model for generating the features we can derive the fisher vector

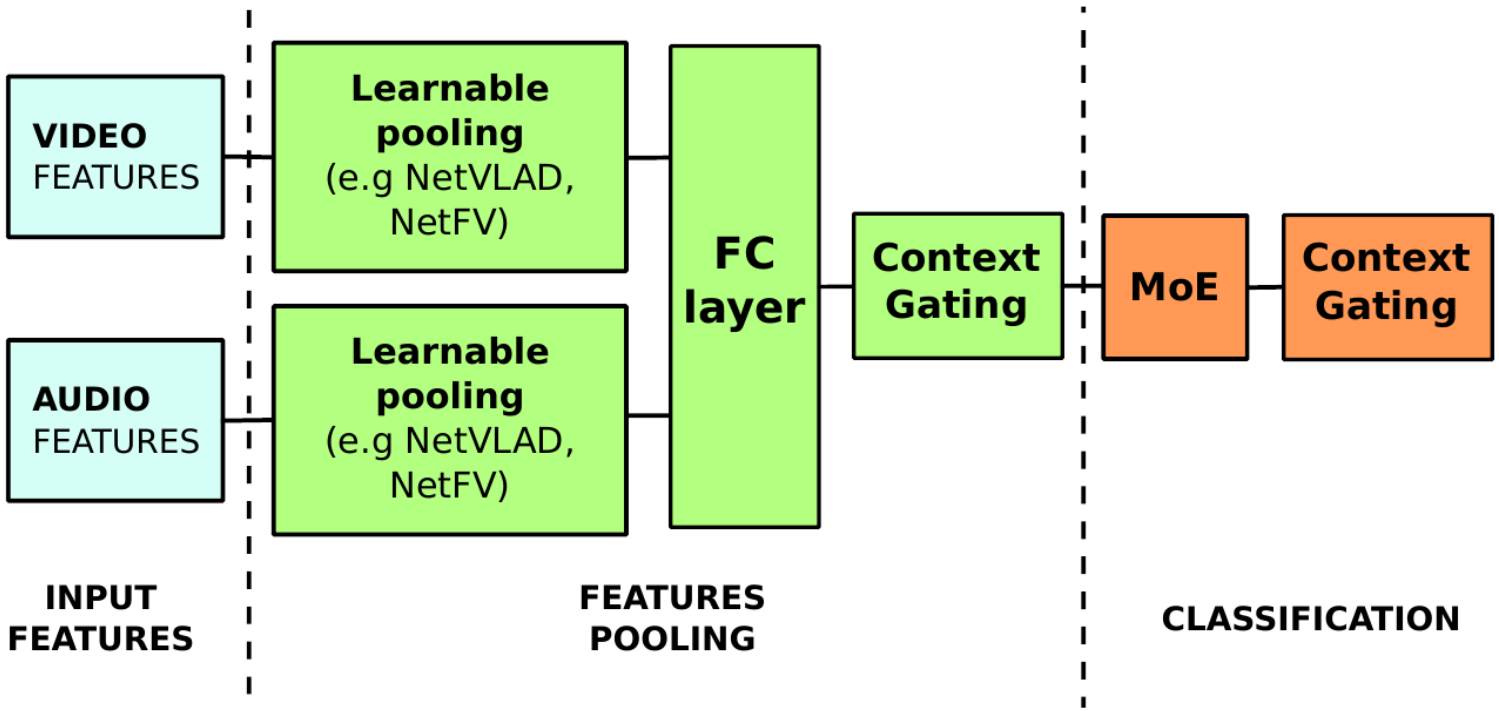
$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial w_i} = \sum_{t=1}^{T} \left[\frac{\gamma_t(i)}{w_i} - \frac{\gamma_t(1)}{w_1} \right] \text{ for } i \geq 2 \,,$$

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial \mu_i^d} = \sum_{t=1}^{T} \gamma_t(i) \left[\frac{x_t^d - \mu_i^d}{(\sigma_i^d)^2} \right] \,,$$

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial \sigma_i^d} = \sum_{t=1}^{T} \gamma_t(i) \left[\frac{(x_t^d - \mu_i^d)^2}{(\sigma_i^d)^3} - \frac{1}{\sigma_i^d} \right] \,.$$

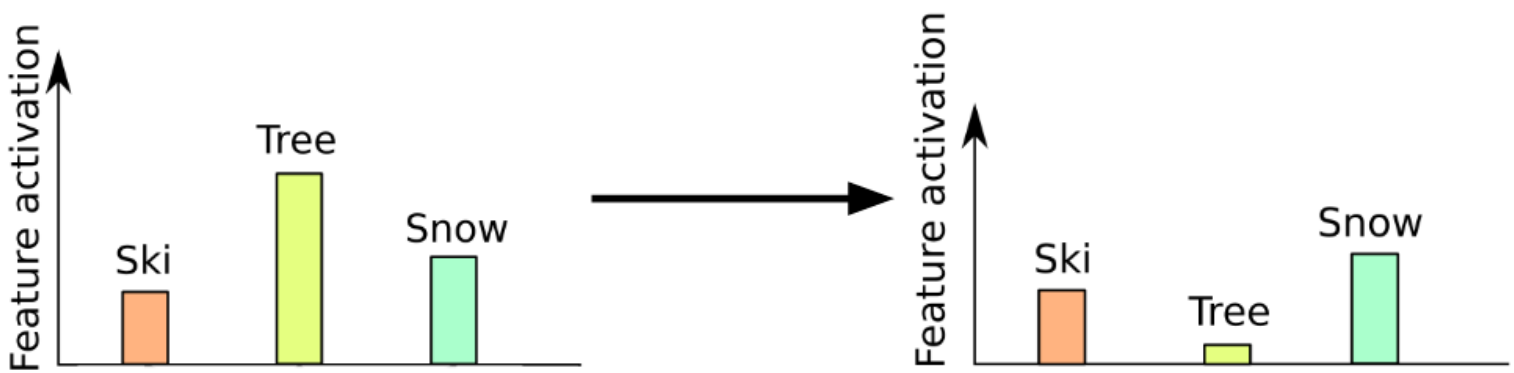
- $\gamma_t(i)$ is probability that x_t is generated by the i-th gaussian
- again we can lean w, μ, σ

Architecture overview



Context Gating

- $Y = \sigma(WX + b) \odot X$
- introduce non-linear interactions among activations
- we wish to recalibrate the strengths of different activations
- used before classification to capture the dependencies among the features
- used after classification re-weight the output probabilities for the different classes.



Results

Method	GAP
NetVLAD	82.2%
NetVLAD + CG after pooling	82.7%
NetVLAD + GLU after pooling, CG after MoE	82.7%
NetVLAD + CG after pooling and MoE	**83.0%**

Results

Method	GAP
Baseline 1 (Average pooling + Logistic Regression)	71.4%
Baseline 2 (Average pooling + MoE + CG)	74.1%
LSTM (2 Layers)	81.7%
GRU (2 Layers)	82.0%
Soft-DBoW (4096 Clusters)	81.6%
NetFV (128 Clusters)	82.2%
NetVLAD (256 Clusters)	82.4%
Gated Soft-DBoW (4096 Clusters)	82.0%
Gated NetFV (128 Clusters)	83.0%
Gated NetRVLAD (256 Clusters)	83.1%
Gated NetVLAD (256 Clusters)	**83.2%**

Results

Groundtruth: Car - Vehicle - Sport Utility Vehicle - Dacia Duster - Renault - 4-Wheel Drive

Top 6 scores: Car - Vehicle - Sport Utility Vehicle - Dacia Duster - Fiat Automobiles Volkswagen Beetles

Results

Groundtruth: Tree- Christmas Tree - Christmas Decoration - Christmas
Top 6 scores: Christmas - Christmas decoration - Origami - Paper - Tree
Christmas Tree

Deep Convolutional Ranking for Multilabel Image Annotationn

- Gong, Yunchao, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. "Deep convolutional ranking for multilabel image annotation." arXiv preprint arXiv:1312.4894 (2013).

Deep Convolutional Ranking for Multilabel Image Annotation

- Gong et al. [2013] method for image multi label classification
- investigate multiple loss function
- softmax loss
  - softmax loss
  - pairwise-ranking loss
  - warp loss

Deep Convolutional Ranking for Multilabel Image Annotation

- softmax loss

$$p_{ij} = \frac{e^{f_j(x_i)}}{\sum_{k=1}^{c} e^{f_k(x_i)}} \tag{6}$$

$$J = -\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{c} \bar{p_{ij}} log(p_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{c_+} \frac{1}{c_+} log(p_{ij}) \tag{7}$$

- $\bar{p_{ij}}$ normalized ground truth predictions
- not specific for multilabel problem

Deep Convolutional Ranking for Multilabel Image Annotation

- pairwise-ranking loss

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c_+} \sum_{k=1}^{c_-} \max(0, 1 - f_j(\boldsymbol{x}_i) + f_k(\boldsymbol{x}_i))$$

- does not directly optimize the top-k annotation accuracy

Weighted Approximate Ranking (WARP)

- warp loss

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c_+} \sum_{k=1}^{c_-} L(r_j) \max(0, 1 - f_j(\boldsymbol{x}_i) + f_k(\boldsymbol{x}_i)).$$

$$L(r) = \sum_{j=1}^{r} \alpha_j, \text{ with } \alpha_1 \geq \alpha_2 \geq \ldots \geq 0.$$

- $\alpha_i = \frac{1}{i}$
- if a positive label is not ranked top
  - $L()$ will assign a much larger weight to the loss
  - $=>$ pushes the positive label to the top
- specifically optimizes the top-k accuracy

Results

method / metric	per-class recall	per-class precision	overall recall	overall precision
Upper bound	99.57	28.83	96.40	46.22
Visual Feature + kNN	32.14	**22.56**	66.98	32.29
Visual Feature + SVM	34.19	18.79	47.15	22.73
CNN + Softmax	48.24	21.98	74.04	35.69
CNN + Ranking	42.48	22.74	72.78	35.08
CNN + WARP	**52.03**	22.31	**75.00**	**36.16**

Recap

- Beyond short snippets: Deep networks for video classification
  - test simple pooling strategies of the frame features
  - train a LSTM for learning to integrate information over time
  - use optical flow as additional info
- Quo Vadis, Action Recognition?
  - review old architectures
  - introduce Two-Stream Inflated 3D ConvNet
- Youtube8M Winner
  - aggregation Methods: Bag of Words, Vlad, Fisher Vector
  - context gating
- Multilabel on image classification
  - use WARP loss

Questions?

Thank you!

References I

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*, 2017.

Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*, pages 487–493, 1999.

References II

A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.