Learning Spatiotemporal Features for Infrared Action Recognition with 3D Convolutional Neural Networks

Zhuolin Jiang, Viktor Rozgic, Sancar Adali

07-21-2017

Raytheon
BBN Technologies

Motivations

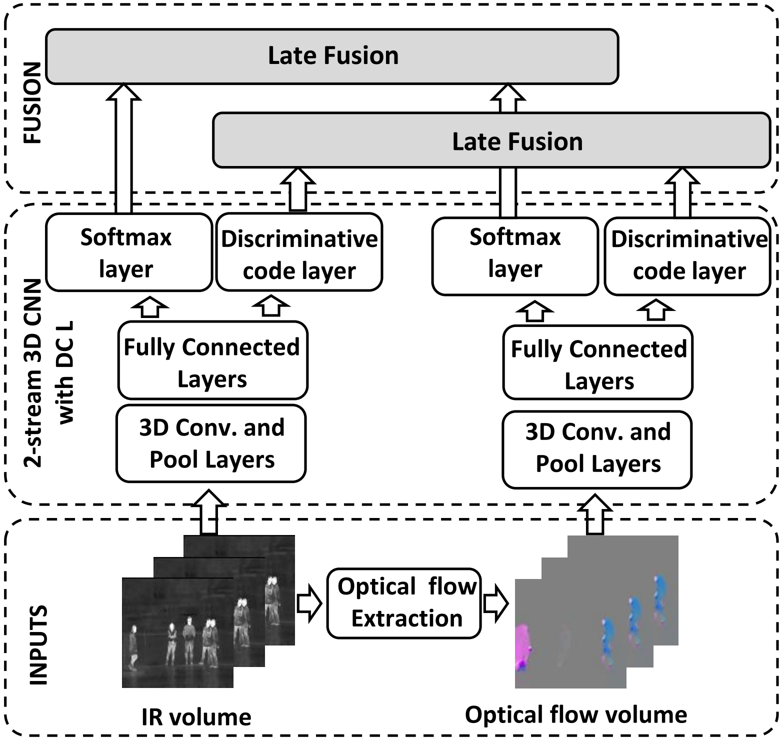
- Compared to visible spectrum cameras, Infrared (IR) imaging enables more robust action recognition due to lower sensitivity to lighting conditions and appearance variability

- While action recognition task on videos collected from visible spectrum imaging has received much attention, action recognition in IR videos is significantly less explored

Our Approach

- We develop a two-stream 3D CNN to learn spatiotemporal features from infrared videos. This two-stream model learns representations that capture *spatial* and *temporal* information simultaneously

- We combine the *discriminative code loss* with softmax classification loss, to train the 3D CNN. This discriminative code layer generates class-specific representations for infrared videos



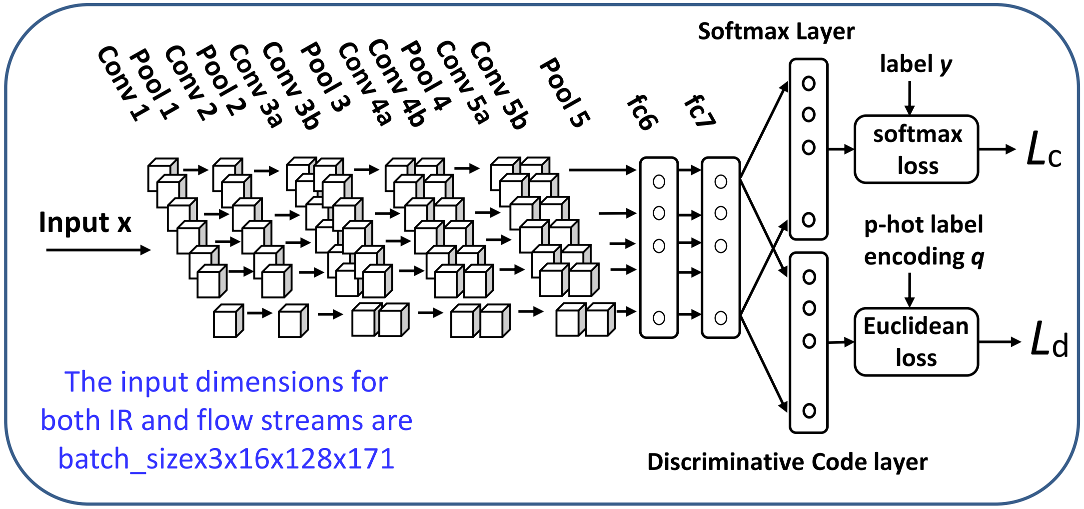
- We pretrained 3D CNN models on the large-scale Sports-1M action dataset with *videos from the visible light spectrum,* and finetuned them on the infrared dataset.

3D Convolutional Neural Network with Discriminative Code Layer

- We add a *discriminative code layer* on top of the last fully-connected layer. The overall loss function in network training:

$$L = L_c + \alpha L_d$$

  - L_c is the softmax classification loss
  - L_d is the discriminative code loss



- **Discriminative code loss**

$$L_d = L_d(\mathbf{x}_d^{(n+1)}, y) = \|\mathbf{q}^{(n)} - \mathbf{A}\mathbf{x}^{(n)}\|_2^2,$$

where $\mathbf{x}^{(n)}$ is output of the n-th layer, and $q^{(n)}$ is the target discriminative code or p-hot label encoding.

- **Target discriminative code** §
  - ✓ Each neuron is associated with a certain class label
  - ✓ ideally, only activates to samples from that class.

For example, given six neurons $\{d_1 \ldots d_6\}$ and five samples $\{y_1 \ldots y_5\}$,

$$\mathbf{Q}^{(l)} = \begin{array}{c} \begin{array}{ccc} \overbrace{y1}^{\text{Class 1}} & \overbrace{y2 \quad y3}^{\text{Class 2}} & \overbrace{y4 \quad y5}^{\text{Class 3}} \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \end{array} \begin{array}{l} \\ \left.\begin{array}{l} d1 \end{array}\right\} \text{group1, Class 1} \\ \left.\begin{array}{l} d2 \\ d3 \\ d4 \end{array}\right\} \text{group2, Class 2} \\ \left.\begin{array}{l} d5 \\ d6 \end{array}\right\} \text{group3, Class 3} \end{array}$$

§ Z. Jiang, Y. Wang, L. Davis, W. Andrews, V. Rozgic. "Learning Discriminative Features via Label Consistent Neural Network". WACV, 2017

Experimental Results

- Evaluated datasets
  - ✓ InfAR video dataset (12 action classes with 50 videos in each class)

- Baselines
  - ✓ Low-level descriptor features
    - ○ dense SIFT (D-SIFT), opponent SIFT (O-SIFT), and improved dense trajectories features (IDT)
  - ✓ Semantic concept/attribute features
    - ○ 2,784 concept detectors trained on the VideoStory dataset using D-SIFT, O-SIFT or IDT, separately.

Experimental Results

- Recognition performance comparisons in terms of average precisions (%)

Method	AP (%)
D-SIFT [1]	46.7
D-SIFT based concepts	46.7
O-SIFT [21]	47.5
O-SIFT based concepts	47.1
IDT [24]	43.3
IDT based concepts	44.6
Early fusion of all concepts	47.5
Late fusion of all features	47.9

- Recognition results of 3D-CNNs trained with or without discriminative code loss, and using different classification methods

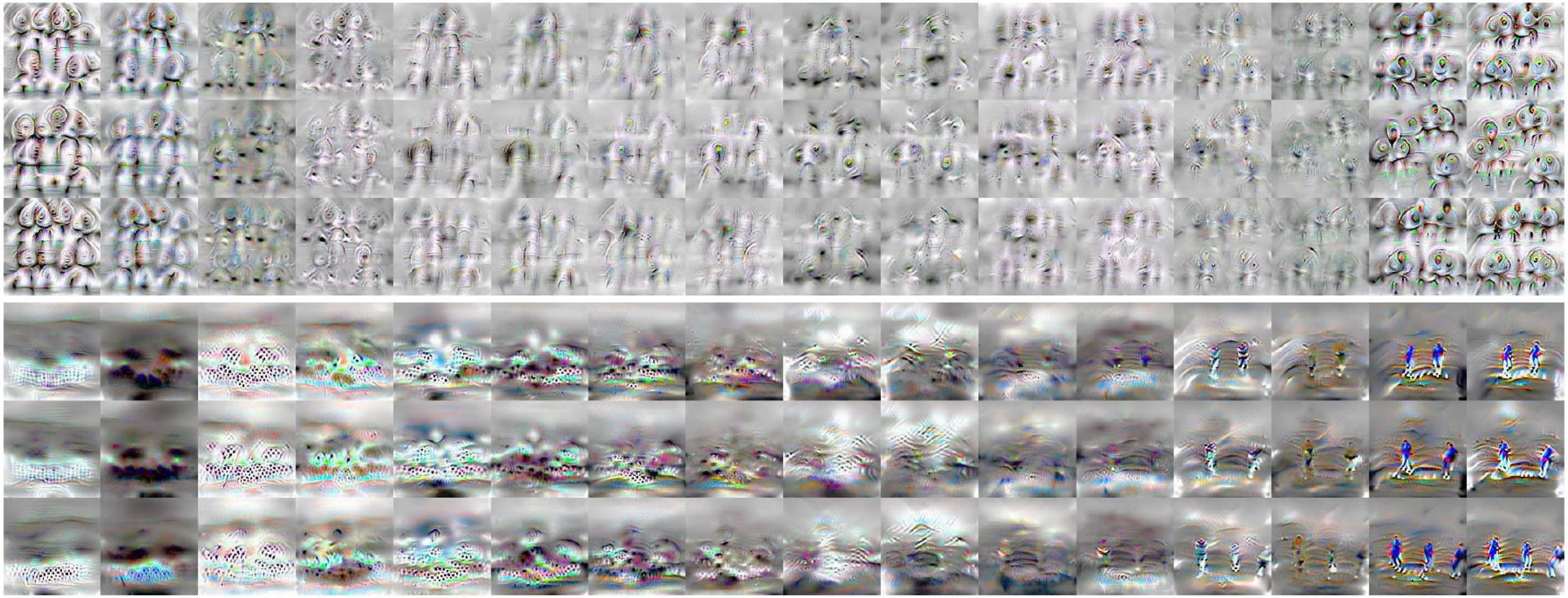
Method	AP (%)
IR net without DCL	48.75
IR net (softmax)	52.91
IR net (k-NN)	54.58
Flow net without DCL	69.58
Flow net (softmax)	72.91
Flow net (k-NN)	75.42
Two-stream-CNN-1 [5]	32.08
Two-stream-CNN-2 [5]	76.66

Two-stream (IR+Flow) 2D-CNN

Two-stream (motion-history-image+Flow) 2D-CNN

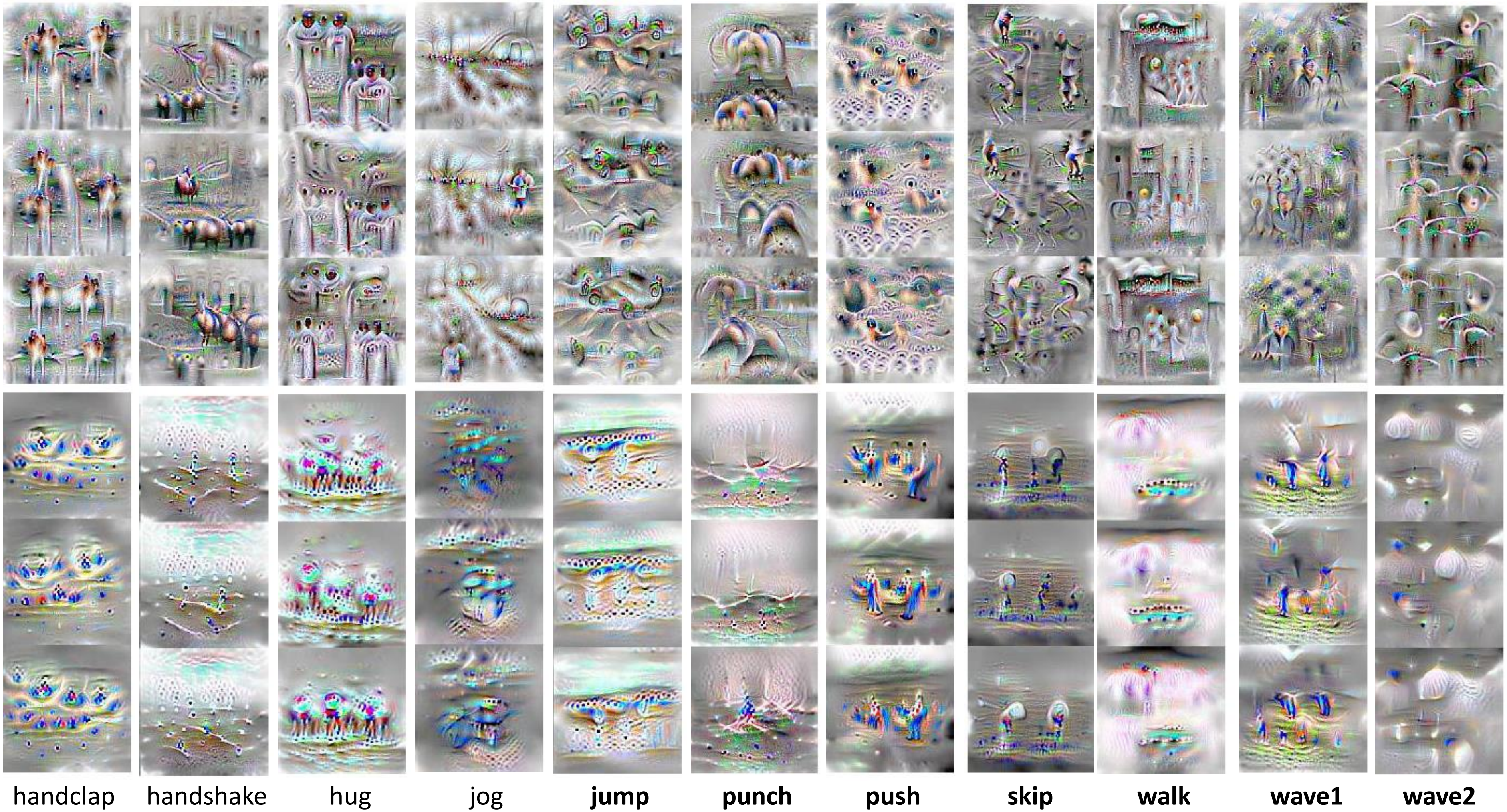
Experimental Results

- Visualization of three learned neurons for action ‘fight’ from the discriminative code layers in the IR and flow nets. Input is a 16-frame sequence of randomly initialized images



Neurons 0-2, assigned to class ‘fight’ (first three rows: IR net, other rows: flow net)

Experimental Results



The last frame of 16-frame long optimized image sequence, the other 11 classes

Conclusion

- We introduce a two-stream 3D convolutional neural network for action recognition in infrared videos.
- Each stream was trained with *softmax classification loss* and *discriminative code loss* making the extracted representations of infrared videos become more discriminative.
- Both nets are initialized by pretraining on high-resource visible spectrum videos, and finetuned on the low-resource infrared videos.

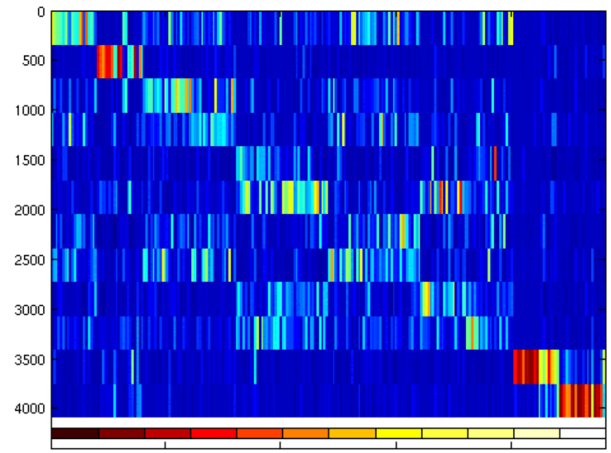
Thank you!

Experimental Results

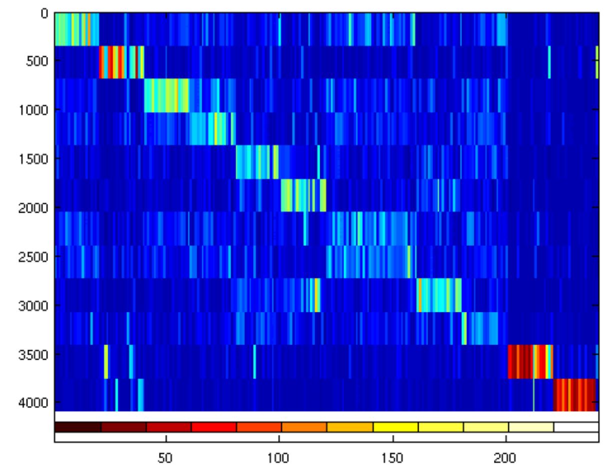
- Recognition performances of fusion with 3D CNN features from IR and Flow nets

Method	AP (%)
Late fusion 1	74
Late fusion 2	77.5
Single-layer NN fusion	71.25
Two-layer NN fusion	70.42

- Visualization of learned discriminative codes of testing videos



(a) IR stream



(b) Flow stream

Network Training

- Compared to standard CNN, the gradient term $\frac{\partial L}{\partial \mathbf{x}^{(n)}}$ changes, and two gradient terms $\frac{\partial L}{\partial \mathbf{A}}$, $\frac{\partial L}{\partial \mathbf{x}_d^{(n+1)}}$ are introduced.

$$\boxed{\frac{\partial L}{\partial \mathbf{x}_d^{(n+1)}} = \alpha \frac{\partial L_d}{\partial \mathbf{x}_d^{(n+1)}},} \quad \frac{\partial L}{\partial \mathbf{x}_c^{(n+1)}} = \frac{\partial L_c}{\partial \mathbf{x}_c^{(n+1)}}$$

$$\boxed{\frac{\partial L}{\partial \mathbf{A}} = 2\alpha(\mathbf{A}\mathbf{x}^{(n)} - \mathbf{q}^{(n)})\mathbf{x}^{(n)\mathrm{T}}}, \frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L_c}{\partial \mathbf{W}}$$

$$\frac{\partial L}{\partial \mathbf{x}^{(n)}} = \frac{\partial L}{\partial \mathbf{x}_c^{(n+1)}} \frac{\partial \mathbf{x}_c^{(n+1)}}{\partial \mathbf{x}^{(n)}} + \boxed{2\alpha(\mathbf{A}\mathbf{x}^{(n)} - \mathbf{q}^{(n)})^{\mathrm{T}}\mathbf{A}}$$

- Once $\frac{\partial L}{\partial \mathbf{x}^{(n)}}$ is known, $\frac{\partial L}{\partial \mathbf{W}^{(i)}}$ and $\frac{\partial L}{\partial \mathbf{x}^{(i-1)}}$ can be computed using the backward recurrence:

$$\frac{\partial L}{\partial \mathbf{W}^{(i)}} = \frac{\partial L}{\partial \mathbf{x}^{(i)}} \frac{\partial \mathbf{x}^{(i)}}{\partial \mathbf{W}^{(i)}},$$

$$\frac{\partial L}{\partial \mathbf{x}^{(i-1)}} = \frac{\partial L}{\partial \mathbf{x}^{(i)}} \frac{\partial \mathbf{x}^{(i)}}{\partial \mathbf{x}^{(i-1)}}, \quad \forall i \in \{1, ..., n\}$$