Weakly Supervised Video Action Recognition with Convex Clustering

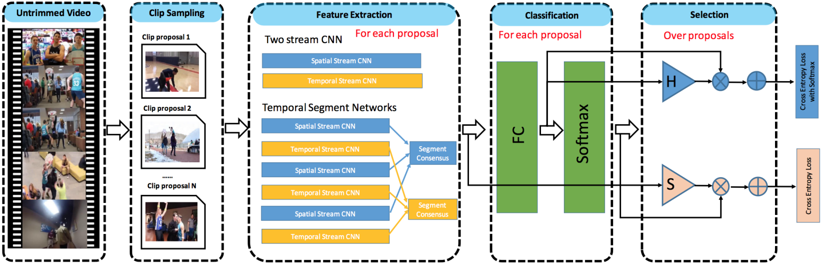
Zhenghao Peng[1,2], Xiaojiang Peng[2], Li Jiang[1], Xiaoou Tang[2], Yu Qiao[2]

[1]Shanghai Jiao Tong University,
[2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences

Video Action Recognition

- A classification problem to assign a given video the corresponding class(es).
- The input video may be untrimmed and can contains multiple action instances.
- Unlike object detection task, action recognition considers spatial-temporal information.
- Widely use in video recommendation and smart surveillance.
- Most videos on Internet have action labels while no temporal annotations. Weakly-Supervised learning is needed.

Previous Work - UntrimmedNet[2]

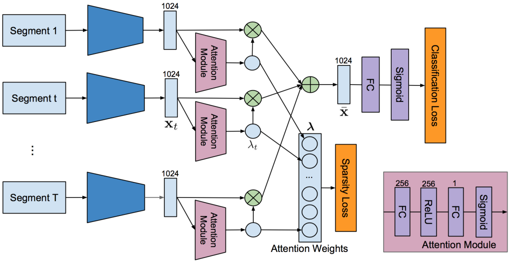


- Weakly-Supervised video action recognition and detection by using an attention module.
- Using Two-Stream-Like Network[1] as feature extractor.
- In training procedure, sample k segments in a video and learn an attention module which produces weights for each segment.
- In testing procedure, using weighted-sum of all segments in one video as output score for classification task and thresholding the segment-wise scores to produce temporal proposal for detection task.

[1] Temporal Segment Networks: Towards Good Practices for Deep Action Recognition

[2] UntrimmedNets for Weakly Supervised Action Recognition and Detection

Previous Work - Sparse Temporal Pooling Network[3]



Pros

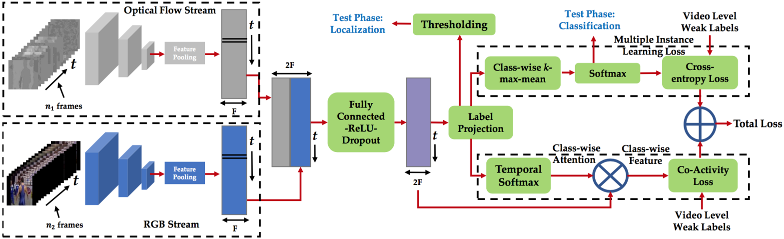
- Focus on improving UntrimmedNet's detection ability.
- Add sparse loss on attention module's output. $L_{sparse} = ||attentions||_1$
- Beside class-agnostic attention, class-relative Temporal Class Activation Map (T-CAM) is built and used for temporal proposal.

Cons

- The detection performance is highly restricted by the network's recognition ability.
- No consideration on the similarity between segments including same class of action.

[3] Weakly Supervised Action Localization by Sparse Temporal Pooling Network

Previous Work - W-TALC[4]



Pros

- Assume that the feature vector of same class of segments should be similar.
- Maximize cosine similarity for same class feature vectors pairs and minimize for those where different classes of actions occurring, by adding a loss to gross loss function and thus the network can be trained End-to-end.

Cons

The comparison required by the similarity calculation is limited among only a batch of data, instead of the whole training set.

[4] W-TALC: Weakly-supervised Temporal Activity Localization and Classification

Related Work - Weakly Supervised Object Detection[6]

- Intuitively, the feature vectors of bounding boxes bounding the same class of object should be similar to each other, and thus should be clustering in feature space.
- In the convex clustering loss L_{cc}, $p(h|x)$ is the weight of a bounding box in a image x. $q_{h',x'}$ is the "representativeness" of a window h' in term of containing information about the related class of object.
- $p(h|x)$ can be the output of attention module (trained end-to-end). $q_{h',x'}$ can be computed by convex clustering technique[5].
- Train the system in two stages: first train the object detection neural network for $p(h)$. Then fixed the network and train $q_{h'}$ for all possible bounding boxes.

$$L_{cc} = -\sum_{h,x} p(h|x) \log\Big(\sum_{h',x'} q_{h',x'} e^{-\alpha d(h,h')}\Big)$$

Pros

- A good assumption: Same object should share features among different images.
- The logarithm term is the "soft" version of "clustering center".
- $q_{h',x'}$ can serve as the identifier of representativeness of a window h'.

[5] Convex Clustering with Exemplar-Based Models

[6] Weakly Supervised Object Detection by Convex Clustering

Proposed Method - Overview

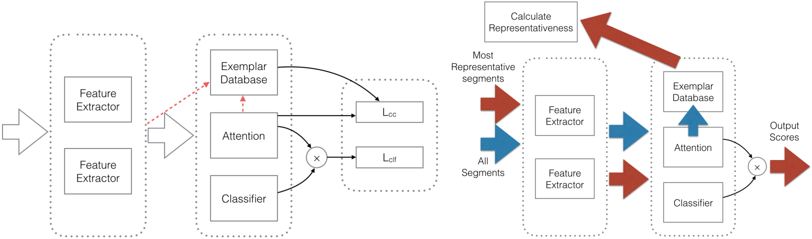
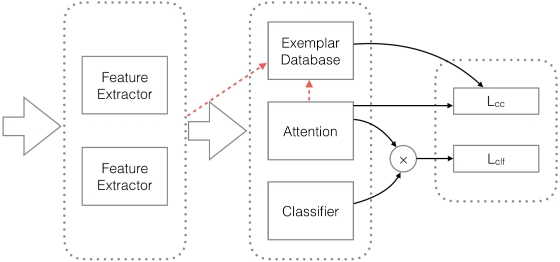


Figure 1: Left: Training Procedure, Right: Testing Procedure.

- Collect high activated segments and build *Exemplar Database*.
- Introduce convex clustering loss.
- Representative segments selection mechanism in testing.

Proposed Method - Components



- Feature Extractor: We use recently proposed I3D[7] models.
- Classifier and attention module: Provide Segment-level classification scores and weights.
- Exemplar Database $\mathcal{D}$: $\mathcal{D} \in \mathbb{R}^{c \times n \times m}$, c, n, m denote the number of classes, single class database size, and length of the feature vector, respectively. Contains the features of those most activated segments. Used for training (providing L_{cc}) and testing (for segments selection).
- Representativeness Matrix $\mathcal{Q}$: $\mathcal{Q} \in [0, 1]^{c \times n}$. The representativeness scores of all feature vectors in $\mathcal{D}$.

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

Proposed Method - Training Procedure (1)

Loss Function

Define:

$$L_{cc} = -\sum_{s \in x, x \text{ in batch}} p(s|x) \log\left(\sum_{s' \in \mathcal{D}} q_{s'} e^{-\alpha d(\phi(s), \phi(s'))}\right)$$

where $d(\cdot)$ is the Euclidean distance of two feature vectors. $\phi(\cdot)$ is the feature extractor. $p(s|x)$ is the attention weight for segment s in video x. The gross loss function is:

$$Loss = L_{clf} + L_{cc}$$

Training Procedure

(1) Train a baseline network without L_{cc}

(2) Collect Exemplar Database $\mathcal{D}$

(3) Fix the neural network, train the representativeness matrix $\mathcal{Q}$ based on $\mathcal{D}$

(4) Fix $\mathcal{Q}$, train neural network with L_{cc}

(5) Repeat (2)-(4)

Proposed Method - Training Procedure (2)

Collect Exemplar Database

Define the final classification score of a given segment i:

$$s_i = \frac{s_i^{rgb} \times a_i^{rgb} + s_i^{flow} \times a_i^{flow}}{2}$$

Where $s_\cdot^\cdot$ is the output score from classifier and $a_\cdot^\cdot$ is the attention weight.

In training period, due to the weakly-supervised setting, we know the exact class of a given video. Thus we can feed all segments in one video into the network and collect the final scores $\{s_i\}_{i=0}^{length}$.

For each video in training set, this procedure is repeated, all segments' score and related feature vector are collected. For each class, say c, the feature vectors are sorted based on their scores, and the features with the greatest n scores compose the database $\mathcal{D}_c$. All database $\mathcal{D}_c$ compose $\mathcal{D}$

Proposed Method - Training Procedure (3)

Train representativeness matrix $\mathcal{Q}$

Each element in $\mathcal{Q}$ represent a weight of a "Exemplar" feature vector. And they are subjected to:

$$\sum_{s} q_{c,s} = 1, \forall c$$

c denote the class.
In order to minimize L_{cc}, when the neural network is fixed (that is $p(s|x) isfixed$), we apply the following update rules:

$$t_{s,s'} = e^{-\alpha d(\phi(s),\phi(s'))}$$

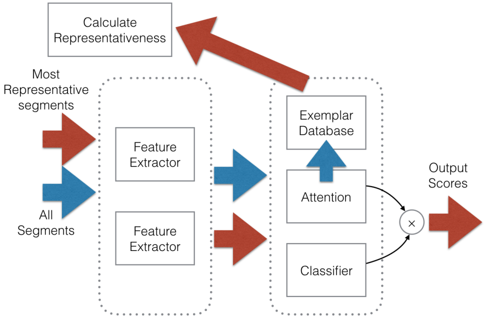
$$z_s^{(t)} = \sum_{s'} t_{s,s'} q_{s'}^{(t)}$$

$$\eta_{s'}^{(t)} = \sum_{s} p(s|x) \frac{t_{s,s'}}{z_s^{(t)}}$$

$$q_{s'}^{(t+1)} = \eta_{s'}^{(t)} q_{s'}^{(t)}$$

We stop the updating when $|max_s(\log \eta_s) - \sum_s q_s \log \eta_s|$ less than a threshold.

Proposed Method - Testing Procedure



(1) Input a video, run the neural network to get all scores and features of all segments $\{s_i, \phi_i\}_{i=0}^{length}$.

(2) For each class, calculate the L_{cc} w.r.t. the sub-database $\mathcal{D}_c$ and feature $\{\phi_i\}$ and take the k features with least L_{cc}. The union of all class compose filtered features set $\{\phi\}^*$. (it has size range from k to $c \times k$)

(3) Run the classifier and attention module to generate final score for $\{\phi\}^*$.

Conclusion

In this work, we propose a weakly supervised video action recognition framework which leverages attention module and a external representativeness database.