Abstract

We present (1) a new CNN architecture named CT-CNN and (2) Non-local Gating ensembles that can infer the multiple actions in short clip videos. In order to learn effective multi-label actions for the video clip, our model aggregates slow and fast network informations and the Non-local Gating inference. Thus, our model achieves more accurate final class confidence prediction of each segments in video. We ensembled multiple predictions of Video CNN models including ours, and 12 variants of Nonlocal Gating layers. We participate in the first Multi Moment in Time challenge \cite{10, 9} in ICCV 2019, for which ensemble of our model achieves one of the best performances.

1. Challenge Introduction

Multi-Moments in Time Challenge 2019 presents a multilabel extension to the Moments in Time Dataset \cite{10, 9} which includes annotation of multiple actions in each video. The goal of this challenge is to detect multiple event labels depicted in a 3 second video clip.

2. Approach

2.1. Preprocessing

Video Frame The size of each frame of video frame data consisting of 3 channels of RGB was all resized in 128 by 128. Resized video goes through the input of the video cnn after some processing in the data loader module. Since the number of frames/fps of each video is not constant, we proceeded to correct this. the temporal depth of the input into the CNN is fixed between 16 and 64. At this time, if the total number of frames is larger than the fixed temporal depth, we uniformly sample frames from the entire video. After adjusting the temporal depth, random cropping was performed so that each frame had a height and width of 112 for data augmentation. In the training session, additional horizontal random flipping was performed.

Between-Class Learning. We applied the idea of between-class learning \cite{13} To do this, the data loader loads two resized videos, matches temporal depth and spatial dimensions, and then mixes them frame-wise at a random rate. In the same way, the labels of two videos are also class-wise mixed at the same rate as the video, producing a real value between 0 and 1. We then learn these values as labels for the video that are frame-wise mixed.

Video Sound. Sound waveforms are obtained from the video data. The waveform was used to train the Envnetv2 model \cite{12}.
2.2. Video CNN Backbones

We extract feature maps from CNN as the backbone of a custom multilabel prediction layer variant, including the NonLocal Gating layer.

CT-CNN : Temporal Shift for Slow Fast Network. As one of the backbones for processing video data, we used variant of SlowFast network [2]. SlowFast network is a video CNN based on ResNet in two paths with different sampling rates of frames. Fast path concentrates on temporal features at high frame sampling rates, and slow path concentrates on spatial features at low frame sampling rates. We implement variant of SlowFast network so that spatial features flows across slow path with Temporal Shift Module layers [6].


2.3. Video Nonlocal Gating

We add NonLocal layer [15] after CNN feature map and following context gating layer [8] for multi-label prediction. Putting learnable pooling before context gating [8] generally helps to improve performance. For example, we denote NLG(NVLAD) as sequential layers of [NonLocal Layer [15], NetVLAD [1], MLP+ContextGating [8]].

2.4. Training

BCE Loss The CNN prediction value is passed through the sigmoid function to calculate Binary Cross Entropy(BCE) loss with ground truth label for 313 classes and to train the model in the direction of minimizing it. BCE loss was used by default in all cnn backbone and fully-connected layers and in Envnetv2 training sessions. We use batch shuffling in every training epoch.

LSEP Loss We use LSEP(log-sum-exp pairwise) loss and threshold estimation for each classes [5] to train CT-CNN model. LSEP loss is differentiable and smooth everywhere, which make model easier to optimize. The estimated thresholds of each classes are used to calculate another binary cross entropy loss, $L_{\text{thresh}}$. $L_{\text{thresh}}$ can be calculated as BCE loss but, its input logits are sigmoid activation of difference between confidence of each classes and estimated thresholds.

We use the SGD-momentum optimizer. We set the Initial learning rate as $lr = 0.001$ in our experiments. For regularization, we apply batch normalization [4], and use dropout [11] after dense layers.

3. Experiments

We report the experimental results of our models for the Multi-Moment in Time challenge. The challenge provides a multi-class video data set that includes 313 action classes, 1M training labels for 1M video, 10K valida-
Figure 3. Qualitative examples of the multi-label action recognition task. The left column shows correct examples, while the right column shows wrong examples. In each case, we show our best scored 6 predictions in descending order. We denote each action class as Name of label(index of label).

3.1. Qualitative Results

Figure 3 illustrates qualitative results of our ensemble model results with correct (left) or wrong (right) examples for each task.

3.2. Quantitative Results

Table 1 summarizes the quantitative results of our experiments. Table 1 shows the public validation results and ensemble weight for final submission. At Table 1 weights of model R(2+1)D is trained 2 more epoch than the other R(2+1)D models with NLG layer.

4. Conclusion

We proposed the NonLocal Gating model for 3D spatio-temporal video feature maps. We have observed performance improvement in ensemble of various NLGating + Pooling layers, even they based on the same CNN feature map. We plan to expand the applicability of the CT-CNN, NLG models; Unfortunately, it was not enough time to run a variety of experiments until the challenge due, but more sophisticated experiments will be updated to compare the various cnn models. Since our method is applicable to any multi-label prediction tasks, we plan to test our model on other large scale video datasets.

Table 1. Performance comparison for validation dataset. ensW means ensemble weight

<table>
<thead>
<tr>
<th>Model</th>
<th>ensW</th>
<th>GAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT-CNN(no Pretrain)</td>
<td>0.1614</td>
<td>0.35082</td>
</tr>
<tr>
<td>R(2+1)D</td>
<td>0.2134</td>
<td>0.52823</td>
</tr>
<tr>
<td>CNN+NLGating</td>
<td>0.1793</td>
<td>0.45770</td>
</tr>
<tr>
<td>CNN+NLGating(BLSTM) [1]</td>
<td>0.3968</td>
<td>0.46304</td>
</tr>
<tr>
<td>CNN+NLGating(LSTM)</td>
<td>0.1507</td>
<td>0.43770</td>
</tr>
<tr>
<td>CNN+NLG(NeXTVLAD) [7]</td>
<td>0.3809</td>
<td>0.49452</td>
</tr>
<tr>
<td>CNN+NLG(NeXTVLAD(med)) [7]</td>
<td>0.5383</td>
<td>0.50102</td>
</tr>
<tr>
<td>CNN+NLG(NeXTVLAD(big)) [7]</td>
<td>0.6764</td>
<td>0.50879</td>
</tr>
<tr>
<td>Ensemble</td>
<td>–</td>
<td>0.78371</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison for official test leaderboard.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRN [10]</td>
<td>0.24</td>
</tr>
<tr>
<td>CT-CNN</td>
<td>0.32</td>
</tr>
<tr>
<td>CT-CNN+Envnetv2+R(2 + 1)D-152 [14]</td>
<td>0.47</td>
</tr>
<tr>
<td>Ours + NLGating Models (Ensemble)</td>
<td>0.4857</td>
</tr>
</tbody>
</table>

References


