3D Convolutional Neural Networks for Human Action Recognition

1 Citation

Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." IEEE transactions on pattern analysis and machine intelligence 35.1 (2013): 221-231.

`http://www.cs.odu.edu/~sji/papers/pdf/Ji_ICML10.pdf`

2 Abstract

Let's use a CNN to recognize human actions in videos. We need 3D convolution because we have a temporal dimension in addition to the two spatial dimensions.

3 Introduction

Traditional action recognition techinques use hand-crafted features. Let's use a CNN instead. Using a CNN on static frames fails to account for motion, so we use 3D convolution. We evaluate on TRECVID (airport surveillence videos). We also evaluate on the KTH data.

4 3D Convolutional Neural Networks

The value of unit at position (x, y) in the jth feature map of the ith layer in 2D convolution is:

$$v_{ij}^{xy} = \tanh(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)})$$

We extend this to 3D:

$$v_{ij}^{xyz} = \tanh(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)})$$

Now we need to turn this into a CNN, which will involve using multiple conv layers (where later layers have more feature maps) and pooling layers.

The input to our CNN will take 7 frames of size 60×40.

Our first layer will apply some hardwired kernels to compute 33 maps with five channels: gray, gradient-x, gradient-y, optflow-x, and optflow-y.

Then, we have a 7 × 7 × 3 conv layer, applied to each input channel separately and produce two sets of 23 feature maps.

The next layer is 2 × 2 subsampling.

We then do 7 × 6 × 3 convolution on each channel and set of feature maps separately. We get six sets of 13 feature maps.

Next is 3 × 3 subsampling.

Next, since the temporal dimension is small now, we only do 7 × 4 conv layer and produce a 128 length feature vector. This goes through a fully connected layer and gets classified.

5 Related Work

The HMAX system is like a CNN, but the features are handcrafted and the system is not end-to-end trainable.

6 Experiments

Given the TRECVID videos, we use a human detector to find humans. We put bounding boxes around them and extract inputs to feed into our CNN. We don't take 7 consecutive frames though, we take every other frame (-6, -4, -2, 0, 2, 4, 6).

There are three action classes (cell to ear, object put, pointing).

We compare our CNN against a model that computes Bag-of-Words (BoW) on the input and feeds into one-vs-all SVMs. For the BoW, we extract SIFT and motion edge history images (MEHI) features. We beat the models we compare ourselves against.

Our metrics are precision, recall, and AUC.

We also evaluate on the KTH dataset (6 action classes). We tweak our CNN architecture to support the larger input size. We are almost as good as the HMAX model.

7 Conclusions and Discussions

Our 3D CNN beats other models on TRECVID and is competitive on KTH.