2020-12-27 1:27:28 PM

Network flow problem treated as that all edge classification

An embedding is obtained for each edge corresponding to a pair of detections:

> Appearance feature is extracted from a ResNet50 CNN using the Image patch corresponding to each detection as input
>
> simple handcrafted geometric features encoding the size, position and time distance between pairs of detections is fed into an MLP to get a corresponding geometric embedding
>
> L2 norm between the feature vectors corresponding to the detection pair is concatenated with the 4D box difference features along with the time difference and the whole apparently 6D feature vector is fed into another MLP will obtain the final Edge embedding

The so called message passing networks are used to update these appearance and geometric features by iteratively passing them through multiple MLPs

Each message passing step has 2 updates – one from edges to nodes and one from nodes to edges – each one represented by a separate MLP

Apparently, the number of iterations of this message passing mechanism determines the maximum distance of the nodes whose information is able to influence any given node and is in that sense equivalent to the CNN receptive field

Some rather crude temporal element is introduced in the message passing by aggregating the features from past and future frames using separate MLPs and then feeding these into another MLP to get the final features

Edges between detection from nonconsecutive frames also considered to handle missing detections but seems to have dubious validity or long term occlusions and false negatives that seem to be so common in many detectors

flow conservation constraints are not proven to be satisfied by the edge classification approach but empirically stated to be so about 98% of the time

Cleaning is done using batches off 8 graphs with each one containing 15 claims

sampled at 6 FPS for fix camera and 9 FPS or moving camera sequences

ResNet50 layers are not trained

Tracktor is used as a sort of pre-processing method by building the graphs on its output

all the usual heuristics and tricks in the training as well as in the conversion of the classifier output into trajectories including some sort of bilinear interpolation to fill in gaps in trajectories