- learns a representation that encodes long term temporal dependencies between appearance, motion and interaction cues
- the network eventually provides similarity scores between all targets and detections that are used to arrange these into a bipartite graph that can be subjected to assignment by the Hungarian algorithm;
- one RNN is used for each cue and outputs from the 3 RNNs are combined by another RNN which outputs a feature vector that in turn is converted into a similarity score;
- using LSTM to encode long term dependencies means that the similarity score can take into accout a sequence of target patches instead of just the previous one;
- Appearance cue:
    - each of the previous t target patches are passed through a CNN which produces a 500-D feature vectors for each;
        - pre trained 16 layer VGGNet with its last FC layer replaced by a 500 sized one;
        - trained using softmax binary classifier
    - all of these (?) are passed through the LSTM that produces an H-D feature vector
    - detection patch is also passed through the CNN and its H-D output feature vector (?) is concatenated with the LSTM output of the last step;
    - the 2H-D concatenated feature vector is passed through an FC layer that finally produces the 500-D feature vector used as the appearance input to the O-RNN
- Motion Cue:
    - the target velocity in t frames passed through LSTM to get H-D feature vector
        - target velocity defined as the difference in the x, y coordinates of the bounding box center between the current and previous frames
    - detection velocity passed though FC layer to get another H-D vector that is concatenated with the LSTM one
    - 2H-D vector passed through another FC to get 500-D vector that becomes motion input to the O-RNN;
    - soft margin classifier used here too;
- Interaction Cue:
    - a binary occupancy grid is created for each target and represented as a vector
    - each cell of the grid is 1 if at least one of the target's neighbors is present in it
    - the grids of targets in last t frames are passed through LSTM to get H-D vector
    - grid of detection is passed through FC to get another H-D vector that is concatenated with the last one to get a 2H-D vector
    - this is passed through an FC layer that outputs a 500-D vector that becomes the interaction input to the O-RNN
    - softmax classifier is used here too;
- Two stage training:
    - the three RNNs are trained separately to output probability of the detection belonging to the trajectory
        - softmax classifier and cross entropy loss used for training
    - joint end to end training of the three RNNs with O-RNN is performed by concatenating their feature vectors (?) and using it as input to the O-RNN
        - the H-D state vector of the last hidden layer of O-RNN is passed through an FC layer to produce another feature vector (?) that encodes the long term dependencies between the cues
        - this O-RNN is also trained (?) to output a similarity score between the target and detection based on their feature vector
    - both training stages use MOT15 and MOT16 datasets