Weak Supervision based Multi-Object Tracking

Alex Ruiz
University of Puerto Rico at Bayamon
alex.ruiz3@upr.edu

Jyoti Kini
University of Central Florida
jyoti.kini@knights.ucf.edu

Abstract

In this work, we designed a neural network architecture called the Weak-Supervision Based Similarity Model. The proposed model is designed as a Multiple Object Tracking (MOT) system where it finds dense key-points correspondences between a set of images to further generate tracklets per object. It has been implemented using the Neighbourhood Consensus Networks and we have incorporated a history-based cascading capability to account for occlusions and to reduce identity switches. We have evaluated the proposed model with the MOT17 dataset.

1. Introduction

Object tracking is one of the most researched and most studied problem in the field of Computer Vision, and it is the backbone for surveillance applications and for molecular observations. Many approaches like the GMMCP [1], Deep Afnity Network [5], Learning Video Representations [2], and others were motivated on solving the data association problem in which suffers from identity switches, occlusions, abrupt motion, and many more in the everyday basis.

In this work, we are taking a different approach on solving the data association problem by proposing a neural network architecture called the Weak-Supervision Based Similarity Model, a pixel-level MOT system in which finds dense key-points correspondences between a set of images so that we can extract relevant pixel-to-pixel matches to further generate tracklets per object. The tracklets are based on all possible pairwise matches between pixels of a pair of images and retain the tracklets with a tracklet-history based cascading capabilities.

2. Related Work

GMMCP Tracker. In this work, the authors have proposed a graph-based MOT framework. It takes a k-partite graph as an input. Then, finds K cliques (tracks) by selecting K nodes (tracklets) from every clusters, a batch of frames [1]. Occlusion management is handled by adding dummy nodes. They build the output from low-level tracklets, then mid-level tracklets and, finally, getting the full trajectories.

Figure 1. Example of the proposed model finding relevant key-point matches (a) to further get final trajectories (b).

Deep Afnity Network. Unlike pixel-to-pixel correspondence, this end-to-end trainable network learns the affinity, characteristics, of object between a pair of video images by evaluating all possible permutations from extracted feature images to further associate objects at different frames for tracking [5].

Learning Video Representations. This paper helps to get rich features representation with temporal context, time related context, and implies that if we pass a set of frames, the resultant output of the network should be good representation of individual frames as well as the temporal relation among each other. This network can find correspondences between a set of images while viewing the input video tensor as a point cloud of features, then it finds k nearest neighbors as potential correspondence based on semantic similarity in multiple frames [2].

These are some of the proposed MOT systems from different papers and are helpful for us to understand the various ways of tackling the data association problem. In our work, we intend to utilize the Neighbourhood Consensus Networks [4] and the Self Attention Generative Adversar-

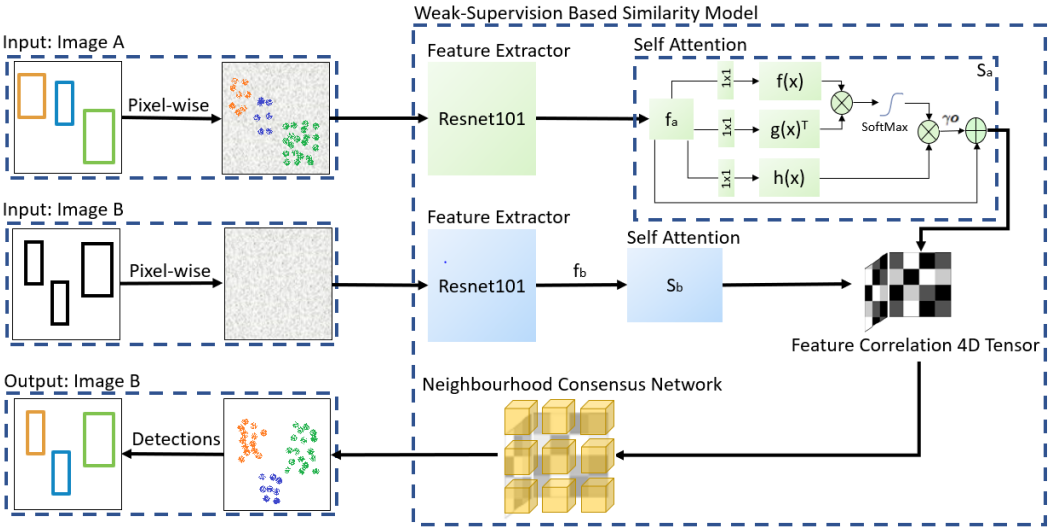

Figure 2. Weak-Supervision Based Similarity Model.

ial Networks [6] research papers to accomplish our goal of designing a MOT system.

3. Proposed Approach

In this section, we are going to explain the different components that are included in the proposed model. It follows the methodology of the Neighbourhood Consensus Networks [4] in which the pixel-to-pixel matches are extracted between a pair of input images. We have arranged from their model to accommodate our need of designing a MOT system is to get matches on all the frames and retain tracklets based on cascade matching. Our model comprises of a series of components whereby includes the following: Feature Extractor, Self Attention Module, Feature Correlation, and Neighbourhood Consensus Networks. We demonstrate the model in Figure 2.

Our model begins with the Feature Extractor component, implemented as a Resnet101 architecture that uses 2-D convolutions by, independently, extracting dense feature maps f_a and f_b from a pair of input images I_a and I_b, respectively. Then, we want to be able to learn the relationship between pixels and all other regions of the image, retaining long-range dependencies [6]. That is why we incorporated the Self Attention Module from [6], S_a and S_b, so that we can fed the extracted feature maps to model enhanced features. This module was not previously implemented in [4].

The computation, storage and the finding of relevant keypoint correspondence from all pairwise feature matches motivates the approach of using 4D convolutions. The Feature Correlation component computes the cosine similarity between a pair of feature maps and stored into a similarity matrix, a 4D correlation map tensor. Now, to extract pixel-to-pixel correspondences, we must mention that the great majority of the information from this tensor corresponds to inconsistent matches. Hence, we use the final component in which is the Neighbourhood Consensus Networks, a custom 4D convolutional architecture, in order to eliminate those matches.

4. Results

4.1. Dataset

We evaluated our proposed model using the MOT17 dataset in which, similar to the dataset used in [3], comprises of 14 video sequences with crowded scenarios, camera motions, varying viewpoints, challenging weather conditions and balanced distribution of crowd density across training and the test-set Figure 3. Additionally, the dataset provides object detections using existing detectors - Deformable Part-Based Model (DPM), Faster Region-based Convolutional Neural Network (FRCNN), and Scale-Dependent Pooling (SDP). Each datapoint in the CSV detection file is in the format: frame number, identity number, bounding box coordinates left, top, width, height, confidence score, class and visibility.

4.2. Weakly-supervised Loss

For a training pair of images I_a and I_b, the weakly-supervised loss is computed as below:

$$\mathcal{L}(I_a, I_b) = -y(S^{-a} + S^{-b}) \tag{1}$$

where S^{-a} and S^{-b} represent the mean matching scores, also positive pairs are labeled with y=1 and negative pairs with y=-1.