Tracking Emerges by Colorizing Videos

C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, K. Murphy

Contributions

In this work, the authors the author learns to colorize gray-scale videos by copying colors from a reference frame. This task will cause the tracker to internally emerge, which can be applied directly to downstream tasks without any fine-tuning. Major components of the framework are summarized as follows:

- By equipping the model with a pointing mechanism into a reference frame, an explicit representation is learnt that can be used for down streaming tasks without any additional training.
- Model is able to track any segmented regions/human pose specified in the first frame.

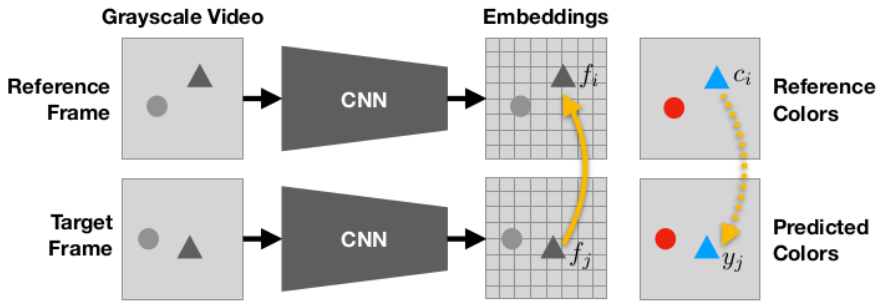

Fig. 2. Model Overview: Given gray-scale frames, the model computes low-dimensional embeddings for each location with a CNN. Using softmax similarity, the model points from the target frame into the reference frame embeddings (solid yellow arrow). The model then copies the color back into the predicted frame (dashed yellow arrow). After learning, we use the pointing mechanism as a visual tracker. Note that the model's pointer is soft, but for illustrations purposes we draw it as a single arrow.

Method

Stage 1: Model True color of pixel i and j in reference and target frame is denoted as $c_i \in R^d$, and $c_j \in R^d$ respectively. $y_j \in R^d$ model prediction is a linear combination of colors in reference frame

$$y_j = \sum_i A_{ij} c_i \tag{1}$$

where A is similarity matrix computed using low dimensional embedding of pixel i and pixel j by following equation:

$$A_{ij} = \frac{exp(f_i^T f_j)}{\sum_k exp(f_k^T f_j)} \tag{2}$$

Two objects of same color need not to have same embeddings. Model only needs to meet the point one reference pixel in order to copy color.

Stage 2: Learning & Inference Loss used is categorical cross entropy after quantizing color space into discrete categories. Similarity matrix A is computed using pair of target and reference frames for two different tasks as follows: 1) Segment tracking, there are d categories corresponding to learning/inference. Initial frame labels is one-hot vectors whereas predicted frames has soft vector representation. 2) Keypoint Tracking, Keypoints are sparse but it's converted into dense representation. A binary vector indicates whether the keypoint is present or not. The network architecture is ResNet-18 followed by a five-layer 3D CNN.

Results

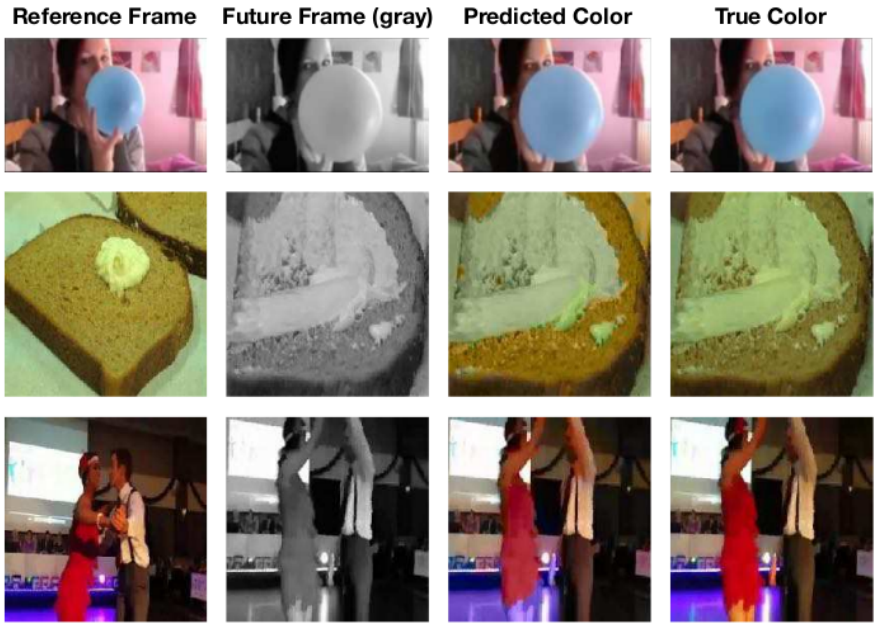

Fig. 4. Video Colorization: We show video colorization results given a colorful reference frame. Our model learns to copy colors over many challenging transformations, such as butter spreading or people dancing. Best viewed in color.

Method	Supervised?	Segment	Boundary
Identity		22.1	23.6
Single Image Colorization		4.7	5.2
Optical Flow (Coarse-to-Fine) [59]		13.0	15.1
Optical Flow (FlowNet2) [23]		26.7	25.2
Ours		34.6	32.7
Fully Supervised [47, 46]	✓	55.1	62.1

Table 1. Video Segmentation Results. We show performance on the DAVIS 2017 validation set for video segmentation. Higher numbers (which represent mean overlap) are better. We compare against several baselines that do not use any labeled data during learning. Interestingly, our model learns a strong enough tracker to outperform optical flow based methods, suggesting that the model is learning useful motion and instance features. However, we still cannot yet match heavily supervised training.

Method	PCK@.1	PCK@.2	PCK@.3	PCK@.4	PCK@.5
Identity	43.1	64.5	76.0	83.5	88.5
Optical Flow (FlowNet2) [23]	45.2	62.9	73.5	80.6	85.5
Ours	45.2	69.6	80.8	87.5	91.4

Table 2. Human Pose Tracking (no supervision): We show performance on the JHMDB validation set for tracking human pose. PCK@X is the Probability of Correct Keypoint at a threshold of X (higher numbers are better). At a strict threshold, our model tracks key-points with a similar performance as optical flow, suggesting that it is learning some motion features. At relaxed thresholds, our approach outperforms optical flow based methods, suggesting the errors caused by our model are less severe.

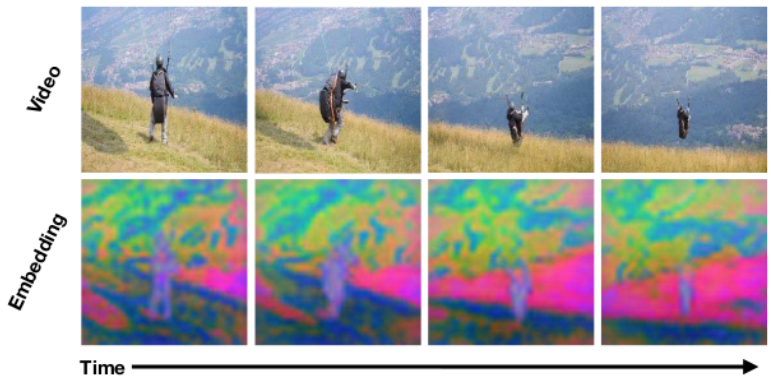

Fig. 9. Visualizing the Learned Embedding: We project the embeddings into 3 dimensions using PCA and visualize it as an RGB image. Similar colors illustrate the similarity in embedding space. Notice that the learned embeddings are stable over time even with significant deformation and viewpoint change. Best viewed in color.