Video Understanding

Asim Kadav

Researcher, NEC Labs America

Topics in Deep Learning: Methods and Biomedical Applications
Spring 2020
MOTIVATION: WIDESPREAD VIDEO-BASED APPLICATIONS

- Video organization
- Retail intelligence
- AR/VR
- Public safety
- Home safety
- Robotic manipulation
Some Video Understanding Tasks and Datasets

- **Segmentation**
  - YouTube-VOS
  - COCO
  - PoseTrack’17

- **Skeleton**
  - VOT’18

- **Detection/Tracking**
  - Youtube 8M
  - Kinetics
  - Charades

- **Classification**
  - AVA
  - Charades

- **Localization**
  - ActivityNet Captions
  - Charades-STA

- **Language**
  - TALL

**Key challenges:**

1) Understand the higher-level spatio-temporal concept in the overall video snippet
2) Understand temporal motion of objects to reduce inaccuracies due to occlusion, background clutter, lighting conditions as objects
WHAT IS A VIDEO?

Sequence of frames representing temporal motion
Each second represented by multiple frames (FPS)
Each color pixel represented by 3 channels (R, G, B)
Fine-grained spatial relationship
Short and long temporal relationships

Example: Video from kinetics dataset
10 seconds x 720p (1280x720)

Raw space necessary:
3 (Channels) x 8 bits per channel x 1280 x 720 x 10 seconds * 15 FPS = 395 MB

After compression
~ 5 MB (H.264)
• High encoding cost but supported by most modern processors
• But ML algorithms operate on raw frames (~395 MB every 10s)
• Action recognition (video classification) is the most well studied video understanding task
• Most interesting videos (and complex motion) are based around human actions

What spatio-temporal features does the model need to learn?

Spatial
• Human pose
• Background
• Interacting object

Temporal
• Objects in motion
• Motion tracking
• Odometry

Others
• Audio
• Action length
• Specific combination of all of the above
PRE-DEEP LEARNING APPROACHES — SVM BASED

Recognizing Human Actions: A Local SVM Approach
Schuldt et. al. (2004)
Optical flow computes a motion field that gives:

1. Motion field of overall scene
2. Object tracking
3. Visual odometry
Dense trajectories and motion boundary descriptors for action recognition, 
International Journal of Computer Vision, H Wang et. al. 2013
Action recognition with improved trajectories, Wang et. al. ICCV 2013
EX. VIDEO CLASSIFICATION TASK (UCF-11)

Detect human actions in video classification instead of objects in image classification.
for f in frames:

224x224x3x1

Video frames

INPUT 32x32

C1: 6 f. maps
28x28

S2: 6 f. maps
14x14

C3: 16 f. maps
10x10

S4: 16 f. maps
5x5

C5: layer
120

O/P 10

F6: layer
84

Convolutions

Subsampling

Convolutions

Subsampling

Fully connected

Tennis forehand

67% on UCF-101 dataset

Video frames

CNN

224x224x3x15

Pool/Concat intermediate features for all frames

Tennis forehand
What is the problem here if we just use RGB features? Or even use RGB+flow features?
Two Stream Networks: Fusing RGB and Flow Scores

Two Stream Networks for Action Recognition in Videos. Simonyan et. al. NIPS 2014

<table>
<thead>
<tr>
<th>Model</th>
<th>Spatial Stream ConvNet</th>
<th>Temporal Stream ConvNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-stream model (fusion by averaging)</td>
<td>73.0%</td>
<td>40.5%</td>
</tr>
<tr>
<td>Two-stream model (fusion by SVM)</td>
<td>86.9%</td>
<td>58.0%</td>
</tr>
<tr>
<td>Two-stream model (fusion by SVM)</td>
<td>88.0%</td>
<td>59.4%</td>
</tr>
</tbody>
</table>
USING 2D CNN FEATURES WITH LSTM

Video frames → CNN → Intermediate Features → Sequence model → Tennis forehand

<table>
<thead>
<tr>
<th>Model</th>
<th>Single Input Type</th>
<th>Weighted Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGB</td>
<td>Flow</td>
</tr>
<tr>
<td>Single frame</td>
<td>67.37</td>
<td>74.37</td>
</tr>
<tr>
<td>LRCN-fc6</td>
<td>68.20</td>
<td>77.28</td>
</tr>
</tbody>
</table>

TABLE 1

Activity recognition: Comparing single frame models to LRCN networks for activity recognition on the UCF101 [25] dataset, with RGB and flow inputs. Average values across all three splits are shown. LRCN consistently and strongly outperforms a model based on predictions from the underlying convolutional network architecture alone.

Long-term recurrent CNNs for Visual Recognition and Description, Donahue et. al., CVPR 2015
3D CONVOLUTION NETWORKS

- Convolution in time and space domain (e.g. 5x5xT filters)
- Huge increase in parameters (e.g. UCF-101 2D -> 3D, 5M -> 33M params), C3D is 39.5 GFlop (as compared to resnext 8GFlop)
- Slowly learns time and space relationships through depth of the network
- 2D -> pooling/concat instead bring the temporal information all at once

3D Convolution Neural Networks for Human Action Recognition, Ji et. al. ICML 2010

Learning station-temporal features with 3D convolutional networks. Tran et. al., 2015
• Uses 3-D convolution (C3D) features from attended intermediate layers with LSTM

• Used to solve the video captioning task, but the intermediate features can be used for any video understanding task
ACTION RECOGNITION DATASETS
EARLY DATASETS: UCF-101 & HMDB-51

- **UCF 101**: 101 classes, 7 sec videos, 13K videos
- Large and commonly used dataset until 2017
- Youtube videos: variety of camera angles (first person, ego-centric, TV), illuminations, background, pose etc.
• HMDB 51: 51 classes, 4sec videos, 5K videos from movies
• J-HMDB dataset (21 classes from HMDB relying w/ joint information)
• YouTube videos: 1M
• 487 classes
• Fine-grained sports classes
• Pre-training on sports 1M and fine-tuning on UCF-101 generally improves performance
KINETICS

The Kinetics Human Action Video Dataset. Kay et. al. arXiv, 2017

- 10s clips
- Every clip is from a different YouTube video
- For each action, huge variety in people, viewpoint, execution

<table>
<thead>
<tr>
<th>Year</th>
<th>Actions</th>
<th>Clips per class</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017</td>
<td>400</td>
<td>400-1000</td>
<td>300k</td>
</tr>
<tr>
<td>2018</td>
<td>600</td>
<td>600-1000</td>
<td>500k</td>
</tr>
</tbody>
</table>

- Person Actions (Singular): e.g. waving, blinking, running, jumping
- Person-Person Actions: e.g. hugging, kissing, shaking hands
- Person-Object Actions: e.g. opening door, mowing lawn, washing dishes

The Kinetics Human Action Video Dataset. Kay et. al. arXiv, 2017
Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? Hara et. al., CVPR 2018
AVA: ATOMIC VISUAL ACTIONS
# SOMETHING SOMETHING DATASET

## 20BN-SOMETHING-SOMETHING-DATASET

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of videos</td>
<td>220,847</td>
</tr>
<tr>
<td>Training Set</td>
<td>168,913</td>
</tr>
<tr>
<td>Validation Set</td>
<td>24,777</td>
</tr>
<tr>
<td>Test Set (w/o labels)</td>
<td>27,157</td>
</tr>
<tr>
<td>Labels</td>
<td>174</td>
</tr>
<tr>
<td>Putting something on a surface</td>
<td>4,081</td>
</tr>
<tr>
<td>Moving something up</td>
<td>3,750</td>
</tr>
<tr>
<td>Covering something with something</td>
<td>3,530</td>
</tr>
<tr>
<td>Pushing something from left to right</td>
<td>3,442</td>
</tr>
<tr>
<td>Moving something down</td>
<td>3,242</td>
</tr>
<tr>
<td>Pushing something from right to left</td>
<td>3,195</td>
</tr>
<tr>
<td>Uncovering something</td>
<td>3,004</td>
</tr>
<tr>
<td>Taking one of many similar things on the table</td>
<td>2,969</td>
</tr>
<tr>
<td>Turning something upside down</td>
<td>2,943</td>
</tr>
<tr>
<td>Tearing something into two pieces</td>
<td>2,849</td>
</tr>
<tr>
<td>Putting something into something</td>
<td>2,703</td>
</tr>
</tbody>
</table>
Video classification datasets often suffer from visual bias (scene, objects) and difficulties in learning temporal relationships (long and very short temporal relationships)

E.g. 1) Eating watermelon involves a watermelon and with lack of other watermelon actions in the dataset, model infers “eating watermelon” when it sees a visually similar object to a watermelon

2) Short actions like ‘slapping’ are very short, as compared to median length of other actions

How do we design a dataset to include spatial and temporal understanding?
CATER DATASET

- Synthetic video dataset built over CLEVR (Johnson et al., 2017)

CATER: A diagnostic dataset for compositional actions & temporal reasoning. Giridhar et al., ICLR 2020

Atomic action recognition (13 classes)

Compositional action recognition (301 classes)

Localization (36 classes, 6x6 grid)
COIN DATASET

- 11,827 videos, 180 tasks in 12 domains
- Domains: nursing & caring, vehicles, leisure & performance, gadgets, electric appliances, household items, science & craft, plants & fruits, snacks & drinks dishes, sports, and housework
- Tasks include: replace a bulb, install a ceiling fan (domain: electric appliance)
- Steps "remove the lampshade", "take out the old bulb", "install the new bulb" and "install the lampshade" are associated with the tasks "replace a bulb".

COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis, Tang et. al. arXiv 2019
**COIN DATASET**

- 11,827 videos, 180 tasks in 12 domains
- Domains: nursing & caring, vehicles, leisure & performance, gadgets, electric appliances, household items, science & craft, plants & fruits, snacks & drinks dishes, sports, and housework
- Tasks include: replace a bulb, install a ceiling fan (domain: electric appliance)
- Steps “remove the lampshade”, “take out the old bulb”, “install the new bulb” and “install the lampshade” are associated with the tasks “replace a bulb”.

Table 1. Comparisons of existing instructional video datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Duration</th>
<th>Samples</th>
<th>Segments</th>
<th>Type of Task</th>
<th>Video Source</th>
<th>Hierarchical</th>
<th>Classes</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPIII [35]</td>
<td>9h,48m</td>
<td>44</td>
<td>5,609</td>
<td>cooking activities</td>
<td>self-collected</td>
<td>✓</td>
<td>-</td>
<td>2012</td>
</tr>
<tr>
<td>YouCook [14]</td>
<td>2h,20m</td>
<td>88</td>
<td>-</td>
<td>cooking activities</td>
<td>YouTube</td>
<td>✓</td>
<td>-</td>
<td>2013</td>
</tr>
<tr>
<td>S0Salads [40]</td>
<td>5h,20m</td>
<td>50</td>
<td>966</td>
<td>cooking activities</td>
<td>self-collected</td>
<td>✓</td>
<td>-</td>
<td>2013</td>
</tr>
<tr>
<td>Breakfast [28]</td>
<td>77h</td>
<td>1,989</td>
<td>8,456</td>
<td>cooking activities</td>
<td>self-collected</td>
<td>✓</td>
<td>10</td>
<td>2014</td>
</tr>
<tr>
<td>“5 tasks” [10]</td>
<td>5h</td>
<td>150</td>
<td>-</td>
<td>comprehensive tasks</td>
<td>YouTube</td>
<td>✓</td>
<td>5</td>
<td>2016</td>
</tr>
<tr>
<td>Ikea-FA [41]</td>
<td>3h,50m</td>
<td>101</td>
<td>1,911</td>
<td>assembling furniture</td>
<td>self-collected</td>
<td>✓</td>
<td>-</td>
<td>2017</td>
</tr>
<tr>
<td>YouCook8 [52]</td>
<td>176h</td>
<td>2,000</td>
<td>13,829</td>
<td>cooking activities</td>
<td>YouTube</td>
<td>✓</td>
<td>89</td>
<td>2018</td>
</tr>
<tr>
<td><strong>COIN (Ours)</strong></td>
<td>476h,30m</td>
<td>11,827</td>
<td>46,354</td>
<td>comprehensive tasks</td>
<td>YouTube</td>
<td>✓</td>
<td>180</td>
<td></td>
</tr>
</tbody>
</table>

**COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis,**
Tang et. al. arXiv 2019
PROGRESSION OF THE FIELD THROUGH DATASETS

- Datasets have generally grown larger (demands of the deep learning models being used)
- Models have grown diverse from simple movies scenes to various view points, camera, lighting (mostly from youtube)
- Addition of fine-grained spatial actions (Kinetics, Sports-1M); also attempt to capture complex temporal relationships (Sth-sth, CATER)
- Many language grounded tasks beyond captioning

Table from CATER, Giridhar et. al., ICLR 2020
Motivation: Human actions involve complex interactions between the scene objects.

How do we learn complex interactions between scene elements?

Idea: Train an object detector to extract regions and learn their interaction across spatial and temporal domain.
LEARNING PAIR-WISE INTERACTIONS CAN BE EXPENSIVE

• For each pair of objects, mean pool the vectors and train an MLP
• Computation grows quickly with the number of projects
• Adding more objects in a single vector drops accuracy quickly
• Cannot learn higher-order interactions
• Need to do this across frames
USING SELF-ATTENTION TO SELECT SALIENT OBJECTS

Interactions/relationships:

\[ RN(O) = f_\phi \left( \sum_{i,j} f_\theta(o_i, o_j) \right) \]

Concatenation [1]:

\[ f_\theta(o_i, o_j) = W_{f_\theta}^T (o_i \parallel o_j) \]

Dot-product:

\[ f_\theta(o_i, o_j) = \theta(o_i)^T \phi(o_j) \]
\[ \rightarrow O^T W_{f_\theta}^T W_\phi O \]

Higher-order interactions:
- Interactions over groups of inter-related objects
- Covers pair-wise or triplet object relationships as a special case

Goal:
- Detect inter-object relationships
- Objects with significant relationships are selected
- Groups of selected object relationships are concatenated.

Higher-Order Interaction


SINET LEARNS INTERACTION BETWEEN SCENE ELEMENTS (ROIs)

SINET obtains a global video representation via the Scale Dot-Product Attention and a fine-grained representation (over objects) via recurrent higher-order interaction (HOI) module. The latter selects groups of objects with inter-relationships via an attention mechanism, and encodes the attended object features with LSTM. The coarse and fine-grained representations are concatenated for final prediction.
LEARNING HIGHER-ORDER INTERACTIONS WITH SELF-ATTENTION

Goal: Learn higher-order interactions between arbitrary (learnt) subgroups of objects

- Introduce learnable parameters via MLP to address domain shift problem
- Attentive selection with image context to co-attend with overall context
- This is combined with all previous interactions to generate a probability distribution over all objects using self-attention


Objects \( O_t : o_{1,t}, o_{n,t} \)

Image context \( v_{c,t} \)

MLP \( g_{\theta_1} \)

MLP \( g_{\theta_2} \)

MLP \( g_{\theta_3} \)

Attentive Selection

LSTM Cell

\( k = 3 \)
Figure 1. Higher-order object interactions are progressively detected based on selected lower-order interrelationships. ROIs with the same color (weighted r, g, b) indicating there exist inter-object relationships, e.g. eggs in the same bowl, hand breaks egg, and bowl on top of campfire (interaction within the same color). Groups of inter-relationships then jointly model higher-order object interaction of the scene (interaction between different colors). Bottom: ROIs are highlighted with their attention weights for higher-order interactions. The model further reasons the interactions through time and predicts cooking on campfire and cooking egg. Images are generated from SINet (best viewed in color).
QUALITATIVE RESULTS CONTINUED..

Example: All images show a horse and a person. But actions are very different and difficult to distinguish for other methods.
Table 2. Comparison of pairwise (or triplet) object interaction with the proposed higher-order object interaction with dot-product attentive selection method on Kinetics. The maximum number of objects is set to be 15. FLOP is calculated per video. For details on calculating FLOP, please refer to Sec. 7.5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
<th>FLOP ($e^9$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj (mean-pooling)</td>
<td>73.1</td>
<td>90.8</td>
<td>1.9</td>
</tr>
<tr>
<td>Obj pairs (mean-pooling)</td>
<td>73.4</td>
<td>90.8</td>
<td>18.3</td>
</tr>
<tr>
<td>Obj triplet (mean-pooling)</td>
<td>72.9</td>
<td>90.7</td>
<td>77.0</td>
</tr>
<tr>
<td>SINet ($K = 1$)</td>
<td>73.9</td>
<td>91.3</td>
<td>2.7</td>
</tr>
<tr>
<td>SINet ($K = 2$)</td>
<td>74.2</td>
<td>91.5</td>
<td>5.3</td>
</tr>
<tr>
<td>SINet ($K = 3$)</td>
<td>74.2</td>
<td>91.7</td>
<td>8.0</td>
</tr>
</tbody>
</table>

Overall efficiency

I3D: 216 GFlops

SINET: 69 (53+8+8) GFlops
USING TRANSFORMERS INSTEAD OF SINET-HOI

• Object features -> Position encodings -> Type encodings (object and frame) -> Transformer (higher-order learning)
• Large increase in memory requirements
• Performs comparably to SINET (0.1 % lower)
MAKING ACTION RECOGNITION PRACTICAL

- **Subsample videos**: Sample videos at 1-5 FPS

- **Reduce computation along the temporal dimension**: Most modern benchmarks heavily rely on spatial information

- **Use parallel operations**: Transformers and convolution blocks can execute in parallel but the former have huge memory costs

- **On-device processing**: Suppress dead-frames; Use motion or audio to trigger processing

- **Limit to RGB modality**: Most Activity-Net contests are won using combination of optical-flow, audio, and skeletal modalities but recently RGB-only approaches have been competitive
• Compositional Methods: Understand videos in a compositional, spatio-temporal format.

• Using keypoints beyond pose: We have lot of experience modeling key point modality. Can we use these tools to solve other problems beyond pose estimation?

• Self-supervised understanding: Finite labelled data in the world; How do we use video data to generate its own labels?

COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis, Tang et. al. arXiv 2019

CornerNet: Detecting Objects as Paired Keypoints. Law et. al. ECCV 2018

15 Keypoints is all you need. Snower et. al, CVPR 2020

Shuffle and Learn: Mishra et. al. 2016
UNDERSTANDING ACTIONS FROM KEYPOINTS

2-D motion perception, Gunnar Johansson. 1971
EXAMPLE VIDEO UNDERSTANDING TASK: POSE TRACKING

15 Keypoints is all you need. Snower et. al, CVPR 2020
HOW TO SOLVE POSE TRACKING TASK?

Keypoint estimation
- Optical Flow, GCN, Transformer (KeyTrack)
- Use temporal information to augment missed/poor quality detections

Temporal Matching
- Match to ID from one of previous N frames

Assign IDs
- Learn temporal pose warping using transformer

15 Keypoints is all you need.
Snower et. al, CVPR 2020
We have a set of $\mathcal{K}$ keypoints which we wish to track for a video with $\mathcal{T}$ frames, s.t. $t \in \mathcal{T}$.

The $MOTA^k$ for each keypoint $k \in \mathcal{K}$ is:

$$1 - \frac{\sum_t (FN_t^k + FP_t^k + IDSW_t^k)}{\sum_t GT_t^k}$$

Our final MOTA is the average of all $MOTA^k$:

$$\frac{\sum_k MOTA^k}{|\mathcal{K}|}$$
### PoseTrack 2018 ECCV Challenge Val Set

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>Extra Data</th>
<th>AP&lt;sup&gt;T&lt;/sup&gt;</th>
<th>AP</th>
<th>FPS</th>
<th>MOTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>KeyTrack (ours)</td>
<td>×</td>
<td>74.3</td>
<td>81.6</td>
<td>1.0</td>
<td>66.6</td>
</tr>
<tr>
<td>2.</td>
<td>MIPAL [27]</td>
<td>×</td>
<td>74.6</td>
<td>-</td>
<td>-</td>
<td>65.7</td>
</tr>
<tr>
<td>3.</td>
<td>LightTrack (offline) [37]</td>
<td>×</td>
<td>71.2</td>
<td>77.3</td>
<td>E</td>
<td>64.9</td>
</tr>
<tr>
<td>4.</td>
<td>LightTrack (online) [37]</td>
<td>×</td>
<td>72.4</td>
<td>77.2</td>
<td>0.7</td>
<td>64.6</td>
</tr>
<tr>
<td>5.</td>
<td>Miracle [61]</td>
<td>✓</td>
<td>-</td>
<td>80.9</td>
<td>E</td>
<td>64.0</td>
</tr>
<tr>
<td>6.</td>
<td>OpenSVAI [38]</td>
<td>×</td>
<td>69.7</td>
<td>76.3</td>
<td>-</td>
<td>62.4</td>
</tr>
<tr>
<td>7.</td>
<td>STAF [40]</td>
<td>✓</td>
<td>70.4</td>
<td>-</td>
<td>3</td>
<td>60.9</td>
</tr>
<tr>
<td>8.</td>
<td>MDPN [22]</td>
<td>✓</td>
<td>71.7</td>
<td>75.0</td>
<td>E</td>
<td>50.6</td>
</tr>
</tbody>
</table>

### PoseTrack 2017 Test Set Leaderboard

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>Extra Data</th>
<th>AP&lt;sup&gt;T&lt;/sup&gt;</th>
<th>AP</th>
<th>FPS</th>
<th>MOTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>KeyTrack (ours)</td>
<td>×</td>
<td>74.0</td>
<td>80.9</td>
<td>-</td>
<td>61.2</td>
</tr>
<tr>
<td>2.</td>
<td>POINet [42]</td>
<td>×</td>
<td>72.5</td>
<td>-</td>
<td>-</td>
<td>58.4</td>
</tr>
<tr>
<td>3.</td>
<td>LightTrack [37]</td>
<td>×</td>
<td>66.7</td>
<td>75.0</td>
<td>0.2</td>
<td>58.0</td>
</tr>
<tr>
<td>4.</td>
<td>HRNet [47]</td>
<td>×</td>
<td>68.8</td>
<td>75.0</td>
<td>0.2</td>
<td>57.9</td>
</tr>
<tr>
<td>5.</td>
<td>FlowTrack [57]</td>
<td>×</td>
<td>74.6</td>
<td>75.0</td>
<td>0.2</td>
<td>57.8</td>
</tr>
<tr>
<td>6.</td>
<td>MIPAL [27]</td>
<td>×</td>
<td>68.8</td>
<td>75.0</td>
<td>0.2</td>
<td>54.5</td>
</tr>
<tr>
<td>7.</td>
<td>STAF [1]</td>
<td>✓</td>
<td>70.3</td>
<td>75.0</td>
<td>2</td>
<td>53.8</td>
</tr>
<tr>
<td>8.</td>
<td>JointFlow [17]</td>
<td>×</td>
<td>63.6</td>
<td>75.0</td>
<td>0.2</td>
<td>53.1</td>
</tr>
</tbody>
</table>

---

**Figure 5.** Top scores on the PoseTrack leaderboards. E indicates an ensemble of detectors is used, and results in the method being offline. A check indicates external training data is used beyond COCO and PoseTrack. A "-" indicates the information has not been made publicly available. FPS calculations for JointFlow and FlowTrack are taken from [62]. HRNet FPS is approximated from FlowTrack since the methods are very similar. The AP column has the best AP score. AP<sup>T</sup> is the AP score after tracking post-processing.

---

**Figure 6.** Qualitative results of KeyTrack, on the PoseTrack 18 Validation Set (top row) and PoseTrack 17 Test Set (bottom row). Used to optimize MOTA, rather than AP.

---

**Efficiency:**

Our tracking approach is efficient, not reliant on optical flow or RGB data. When processing an image at our optimal resolution, 24x18, we reduce the GFLOPS required by optical flow, which processes images at full size, from 52.7 to 0.1. [37]'s GCN does not capture higher-order interactions over keypoints and can be more efficient than our network with local convolutions. However, this translates to a "1ms improvement in GPU runtime. In fact, with other optimizations, our tracking pipeline has a faster end-to-end runtime than [37] by 30%, shown in 4.4. We have the fastest FPS of Top-down models. Bottom-up models such as STAF, are more efficient but have poor accuracy. Also, we do not rely on optical flow to improve bounding box propagation as [57, 47] do, instead we use TOKS. This contributes to our 5x FPS improvement over [57, 47].

---

**5. Analysis**

### 5.1. Tracking Pipeline

**Varying Tokenization Schemes and Transformer Hyper-parameters**

We examine the benefits of each embedding. As evident in Table 3, Segment embeddings are crucial because they enable the network to distinguish between the poses being matched. Token embeddings give the network information about the orientation of a pose and help it interpret keypoints which are in close spatial proximity; i.e. keypoints that have the same or similar position embedding. We also train a model that uses the relative keypoint distance from the Pose center rather than the absolute distance of the keypoint in the entire image. We find that match accuracy deteriorates with this embedding. This is likely because many people perform the same activity, such as running, in the PoseTrack dataset, leading to them having nearly identical poses. We vary the number of transformer blocks, the hidden size in the transformer block, and number of heads. Decreasing the number of transformer blocks and hidden size hurts performance, while increasing the number of heads too greatly hurts performance. Results are in Table 7.

**Number of Timesteps and Other Factors**

We find that reducing the number of timesteps, adversely effects the MOTA score. It drops up to 0.3 points with a single timestep.
SELF-SUPERVISED METHODS FOR VIDEO UNDERSTANDING

- **Track moving objects:** Wang et. al. 2015: Track patches with motion over a small temporal window => Learns temporal motion of objects

- **Shuffle and Learn:** Mishra et. al. 2016: Validate frame order by shuffling frames => Learns temporal order of whole scene

- **Colorizing videos:** Vonderick et. al. 2018: Given two nearby frames, one in color and another in grey scale, the task is to copy colors from one frame to another nearby frame

Self-supervised approaches. Slides from Lecun, 2019

Example: Use these methods for generating disentangled representation for video generation

S3VAE: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation, CVPR 2020.
SUMMARY

• Video has numerous applications in modern applications such as AR/VR, retail etc.

• Understanding video and generating a good representation is complex and computationally intensive

• Numerous opportunities with new datasets, tasks and compute platforms
QUESTIONS