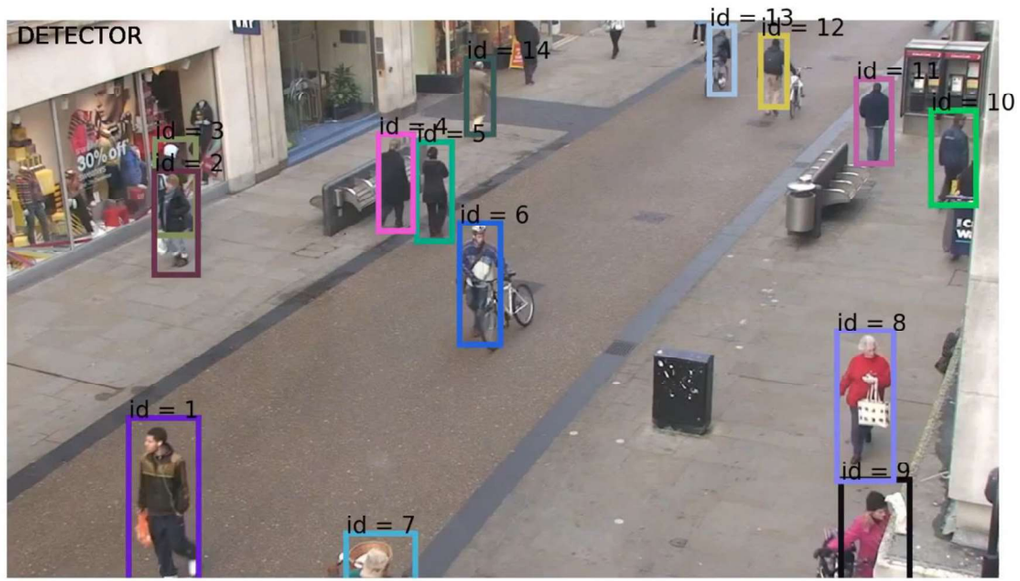
Project scope

- Review a recently published DL based Multi-Object Tracking method.
- Implement it and perform an analysis of its performance.
- Experiment by changing the depth of its architecture.
- Apply some DL method that differentiates its pipeline.



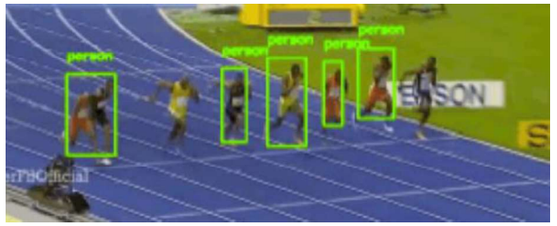
Problems in Multi-Object tracking

I. Detect objects
II. Regress bounding boxes of targets between detections
III. Associate detections to targets
IV. Manage tracked objects (tracklets) by birth, temporary storage and termination.

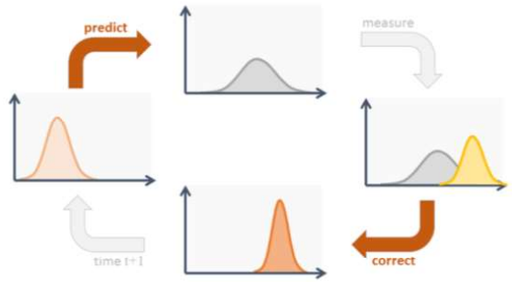
Recent approaches tackling each problem

- Object Detection in Videos with Tubelet Proposal Networks, CVPR 2017
- Tracking without bells and whistles, ICCV 2019
- Graph Neural Based End-to-end Data Association Framework for Online Multiple-Object Tracking, Arxiv 2019
- Online Multi-Target Tracking Using Recurrent Neural Networks, AAAI 2017

Simple Online and Realtime Tracking, CVPR 2016



I. Detect objects using object detector FASTER RCNN
   I. object detection considers only a single frame and no object IDs, only classes



II. Estimation model-Kalman Filter
   I. “u” horizontal and “υ” vertical center of target
   II. s scale and r aspect ratio $\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$
   III. dot variables are a linear constant velocity model with r constant
   IV. prediction is done using the velocity model and correction using the new detections

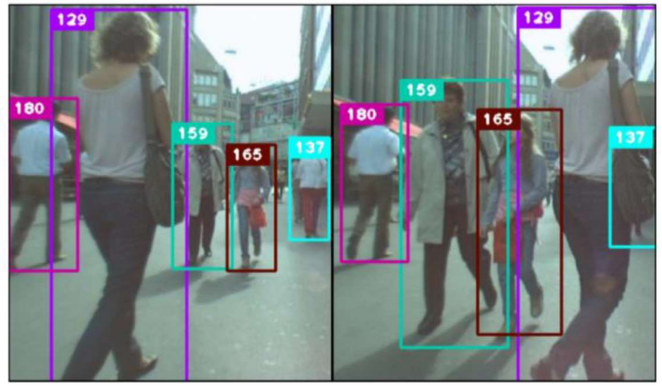
III. Associate detections to tracklets using cost matrix
   I. Assignment cost matrix is computed using the intersection over union (IOU) distance between each detection and existing target boxes
   II. The Hungarian algorithm is used to solve optimally the minimum cost assignment problem.

DEEP Simple Online and Realtime Tracking, ICIP 2017

I. Uses FASTER RCNN again for detection

II. Uses optical flow for, between detections, tracking

III. Uses Kalman Filter for target prediction and correction



IV. Uses a deep association metric to be used for finding correspondences
   I. It is basically a distance metric, Mahalanobis distance
   II. Two metric distances are used:
      I. First one is only distance in pixels between detection and tracklet
      II. Second is distance in appearance space, r_j represents the appearance descriptor
      III. The last 100 frames of apperence descriptor are kept and used for matching for each tracklet
      IV. Those appearance descriptors are extracted from target boxes and detections

$$d^{(1)}(i,j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i)$$

$$d^{(2)}(i,j) = \min\{1 - \boldsymbol{r}_j^{\mathrm{T}} \boldsymbol{r}_k^{(i)} \mid \boldsymbol{r}_k^{(i)} \in \mathcal{R}_i\}$$

$$c_{i,j} = \lambda\, d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j)$$

Deep Cosine Metric Learning for Person Re-Identification, WACV 2018

I. A CNN architecture is used for feature extraction of the bounding boxes.

II. This is trained on a person re-identification dataset to learn to distinguish between same and different people.

   I. Market 1501 that contains 1,261 identities and over 1,100,000 images.

III. They use a modified sofmax classifier called cosine softmax.

$$p(y=k \mid \boldsymbol{r}) = \frac{\exp\left({\boldsymbol{w}_k}^{\mathrm{T}}\boldsymbol{r} + b_k\right)}{\sum_{n=1}^{C}\exp\left({\boldsymbol{w}_n}^{\mathrm{T}}\boldsymbol{r} + b_n\right)} \longrightarrow p(y=k \mid \boldsymbol{r}) = \frac{\exp\left(\kappa \cdot \tilde{\boldsymbol{w}}_k^{\mathrm{T}}\boldsymbol{r}\right)}{\sum_{n=1}^{C}\exp\left(\kappa \cdot \tilde{\boldsymbol{w}}_n^{\mathrm{T}}\boldsymbol{r}\right)}$$

IV. This classifier takes positive and negative examples and tries to minimize the cross entropy loss for classification. Also indirectly the triplet loss is minimized.

$$\mathcal{L}(\mathcal{D}) = -\sum_{i=1}^{N}\sum_{k=1}^{C} \mathbb{1}_{y_i=k} \cdot \log\, p(y_i = k \mid \boldsymbol{r}_i)$$

$$\mathcal{L}_{\mathrm{t}}(\boldsymbol{r}_a, \boldsymbol{r}_p, \boldsymbol{r}_n) = \left\{\|\boldsymbol{r}_a - \boldsymbol{r}_n\|_2 - \|\boldsymbol{r}_a - \boldsymbol{r}_p\|_2 + m\right\}$$

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

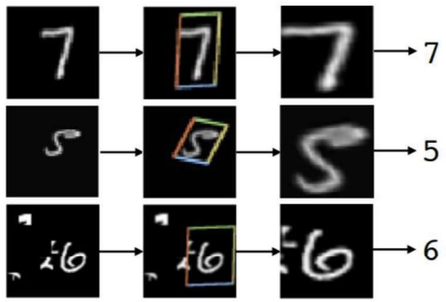
Approach

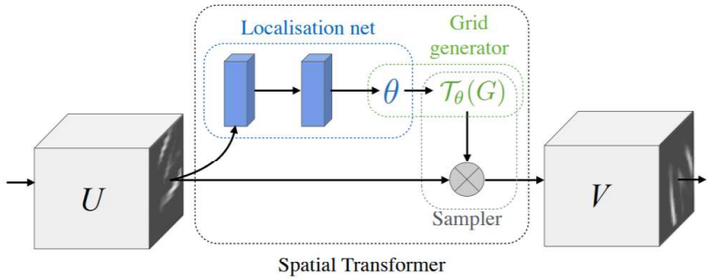
I. Detection:
   I. Multiple detectors will be tested and compared given those available at the MOT17 dataset. DPM, Faster-RCNN, and SDP.

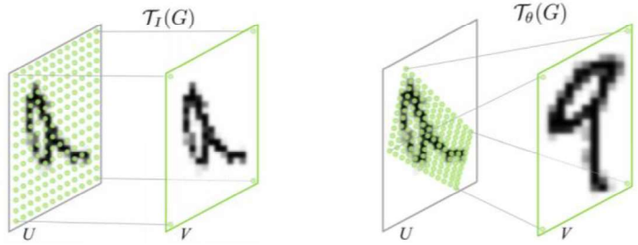
II. Feature appearance extraction:
   I. Structure of the network for re-identification will be changed to shallower and deeper architectures.
   II. The structure will be changed to allow for an attention mechanism to capture more of the details of each detection box
   III. Results will be compared using the Market1501 dataset.

Approach-Improving feature appearance

I. Spatial Transformer Networks, 2016







- a spatial transformer can be used for tasks requiring an attention mechanism
- can be trained purely with backpropagation without reinforcement learning

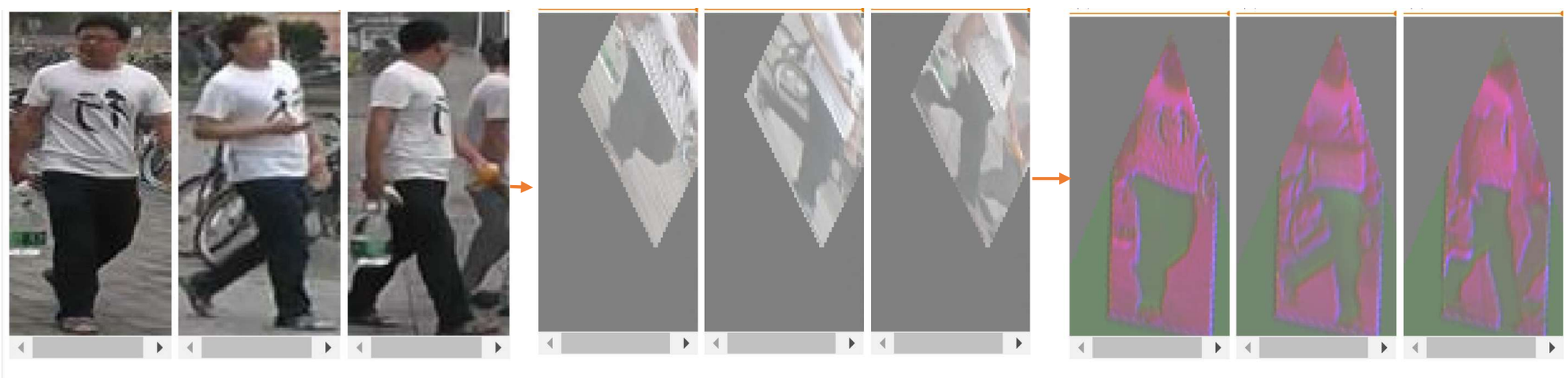
- the input feature map U is passed to a localization network which regresses the transformation parameters θ
- the regular spatial grid G over V is transformed to the sampling grid T_θ(G), which is applied to U

Approach-Improving feature appearance

- Visualizing feature maps of attention network variants

- Attention-1

- Attention-12



Approach-Improving feature appearance

- Visualizing feature maps of attention network variants

- Attention-123