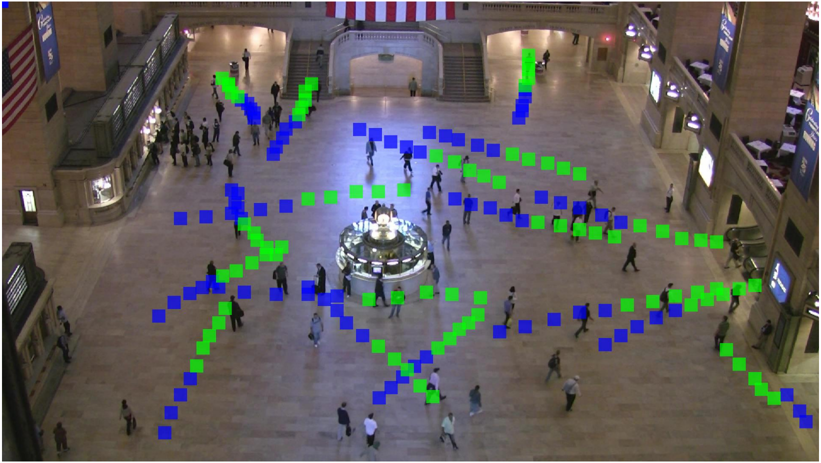
Predicting Human Behavior from Public Cameras with Convolutional Neural Networks

Ondrej Jariabka
Supervisor: prof. Ing. Igor Farkaš, Dr.

Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava

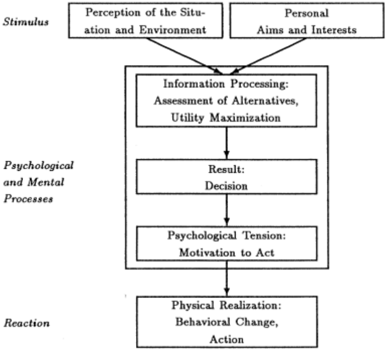
June 12, 2018

Aim



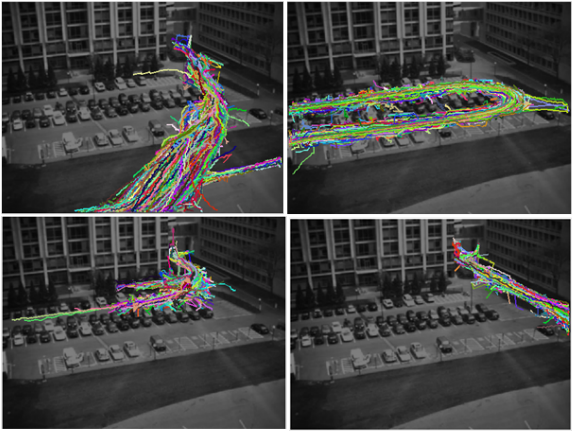
Prediction of pedestrian trajectories through the scene [Yi et al., 2016]

Related Work I.



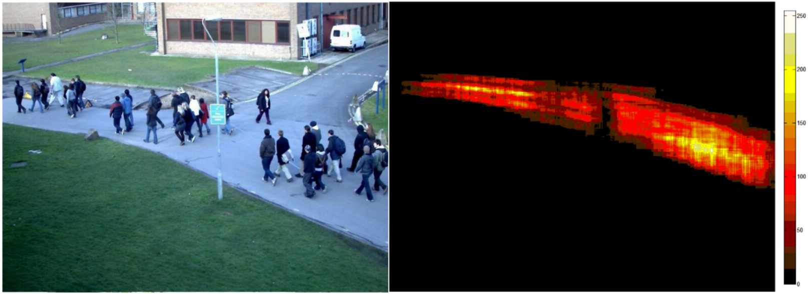
Graph of decision making of individual pedestrian based on "social force" model [Helbing and Molnar, 1995]

Related Work II.



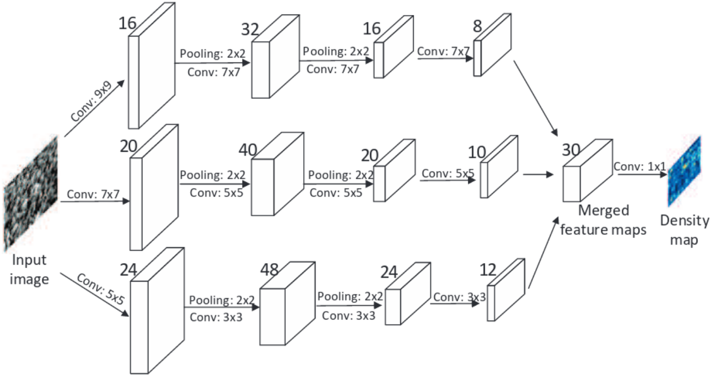
Cluster trajectories used model by Dual-HDP [Wang et al., 2009]

Related Work III.



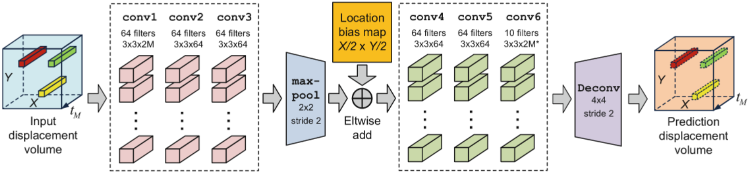
Constructed flow map of the scene by multiview face detector [Xing et al., 2011]

Related Work IV.



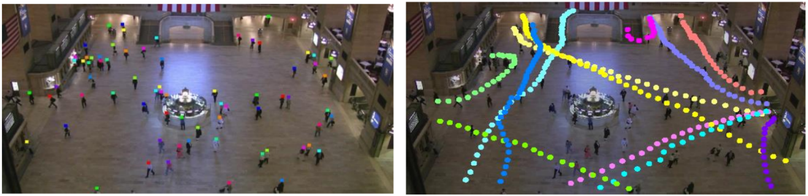
Multicolumn Convolutional Neural Network [Zhang et al., 2016]

Related Work V.



Behavioral Convolutional Neural Network [Yi et al., 2016]

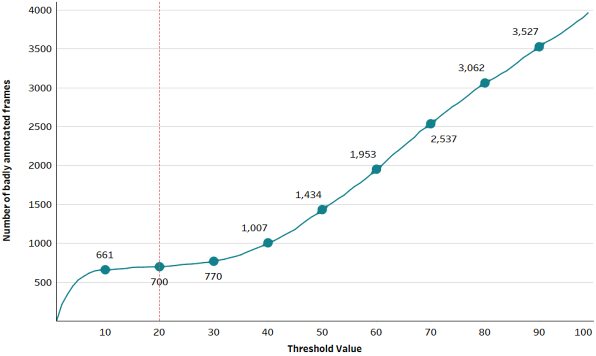
Dataset I.



A one-hour surveillance video with the exact walking paths of 12,684 pedestrians [Yi et al., 2016]

Dataset II.

- Full HD images
- Badly annotations
  - Remove non-annotated sequences
  - Select threshold on number of annotations

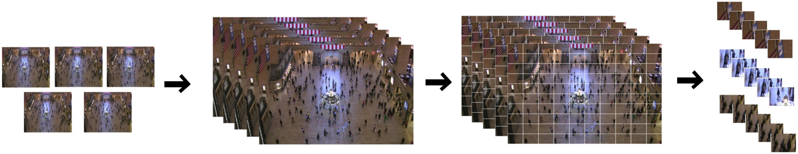


Current approach

- Behavioral-CNN
- Uses pedestrian encoding
  - Need to correctly extract position of pedestrian in test time
  - Construction of volume of displacement vectors for each pedestrian
  - Very sparse input and output
  - Needs special learning scheme
- Predicting same sparse volume
  - Needs a lot of computational power
  - Needs a lot of space

Our approach I.

- No special encoding
- Split the image (9×8 patches)



Process of segmenting input into smaller patches.

Our approach II.

- No special encoding
- Split the image (9×8 patches)
- Predict pedestrian mask
  - Predict entire patch (a)
  - Classify each pixel (b)
- Pedestrian representations (6×6 pixels)

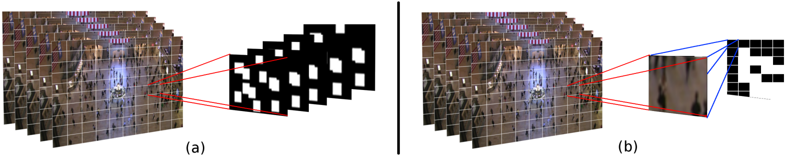


Illustration of our prediction schemes

Our approach III.

- Simple convolutional encoder
  - simple linear architecture

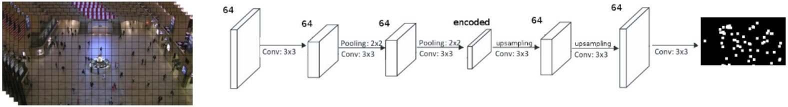


Illustration simple encoder architecture

Our approach IV.

- Multi-column convolutional encoder
  - multiple layers extracting low level features
  - each simultaneous layer with different filter size

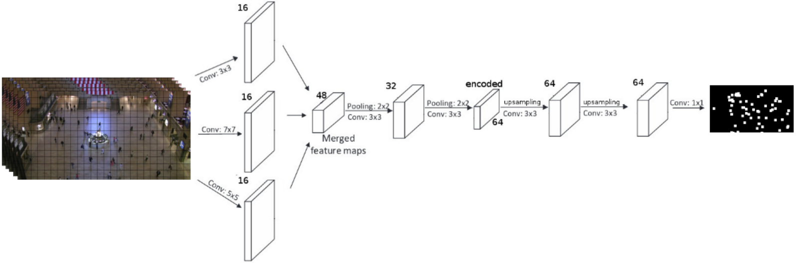


Illustration column encoder architecture

Results I.

- Use **MSE** to make results comparable to [Yi et al., 2016]
- Post processing step to extract pedestrian locations
  - Threshold pedestrian mask
  - Find contours
  - Pick top and left most pixel as pedestrian representation

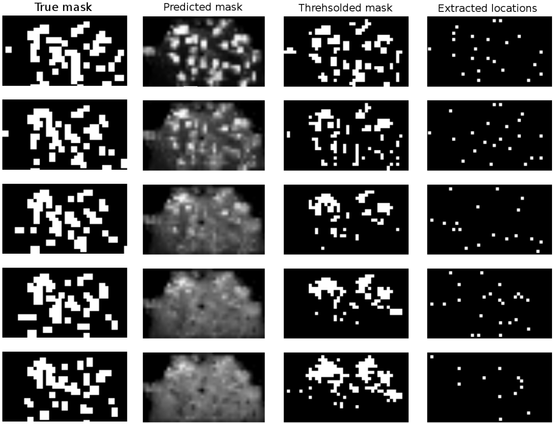
Posprocessing step needed to extract exact pedestrian coordinates

Results II.

architecture	1 frame	2 frame	3 frame	4 frame	5 frame
simple encoder	5.805	13.112	13.899	14.010	15.472
column encoder	3.946	11.766	12.101	12.002	12.621
simple encoder + sparsity	4.900	13.761	14.189	14.510	15.706
column encoder + sparsity	3.401	11.102	**11.336**	11.919	12.803
simple encoder + penalization	4.605	12.779	13.860	14.217	15.873
column encoder + penalization	4.070	11.690	12.093	12.370	12.110
column classifier	3.245	11.042	11.711	**11.893**	12.829
column encoder ensemble	3.103	11.983	12.622	12.891	13.209
column classifier ensemble	**3.015**	**10.814**	*11.519*	*11.978*	**12.780**

Final *MSE* for various types of tested model architectures predicting 1 to 5 frames ahead

Results III.



Sample prediction of our best model. Each row represents one step ahead in prediction

Conclusion

- Prediction of one and two frames ahead
- Need of post-processing step
  - Find good threshold
  - Extract pedestrian coordinates from blobs
- Find good splitting windows size
- Pedestrian representation
  - Merging of pedestrians

Future work

- Find better representation of pedestrian
  - Autoencoder
  - CNN to extract coordinates
- Recurrent Neural Network to reduce sparsity
  - Predict coordinates
  - Time series prediction

Thank you for your attention!

Section methodology

- Heavily based on Deep Learning Book [Bengio et al., 2014] and Stanford cs231n course notes [Li et al.,]

Section methodology

- Heavily based on Deep Learning Book [Bengio et al., 2014] and Stanford cs231n course notes [Li et al.,]
- *The main purpose of a neural network is to approximate some arbitrary function* f'
- Each layer defined as function which we stack to approximate given function

Section methodology

- Heavily based on Deep Learning Book [Bengio et al., 2014] and Stanford cs231n course notes [Li et al.,]
- *The main purpose of a neural network is to approximate some arbitrary function f'*
- Each layer defined as function which we stack to approximate given function
- $y = f(\vec{w}x)$, where w are weights...
- $\vec{y} = \hat{f}(W^T x + \vec{b})$, where W are layer weights... b represents the bias vector...

Rosenblatt neuron

$$output = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \tag{1}$$

Equation 3.1 - Rosenblatt neuron

$$output = \begin{cases} 1 & \text{if } \sum_j w_j x_j + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Next equation 3.2 - redefinition of previous function

Filling empty annotations

- Same as constructing special encoding in test time
- Head-and-Shoulders or face detectors
    - Problem with occlusion
- Pixel-wise segmentation
- Special features with various machine learning algorithms
- Neural networks

LeNet [LeCun et al., 1990]

- Do not describe history
- Various examples of CNNs in context of behavioral modeling

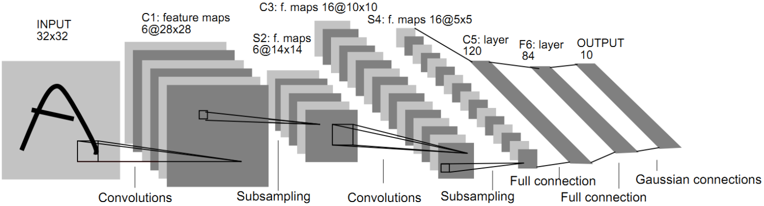


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Hand-written digit recognition with a back-propagation network [LeCun et al., 1990]

Pixel representation

- The current annotation is on the head
- Imperfect edges

MSE as define in [Yi et al., 2016]

$$MSE = \frac{1}{NM'} \sum_{i}^{N} \sum_{j}^{M'} ||I_i^j - I_i'^j||_2 \times 100\%$$

where N is the number of samples, M' is the number of predicted frames, I is the volume containing normalized annotated positions of each pedestrian and I' is predicted volume of normalized pedestrian locations.

Bengio, Y., Laufer, E., Alain, G., and Yosinski, J. (2014).
Deep generative stochastic networks trainable by backprop.
In *International Conference on Machine Learning*, pages 226–234.

Helbing, D. and Molnar, P. (1995).
Social force model for pedestrian dynamics.
Physical Review E, 51(5):4282.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990).
Handwritten digit recognition with a back-propagation network.
In *Advances in neural information processing systems*, pages 396–404.

Li, F.-F., Karpathy, A., and Johnson, J.
Cs231n: Convolutional neural networks for visual recognition 2016.

Wang, X., Ma, X., and Grimson, W. E. L. (2009).
Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(3):539–555.

Xing, J., Ai, H., Liu, L., and Lao, S. (2011).
Robust crowd counting using detection flow.
In *2011 18th IEEE International Conference on Image Processing*, pages 2061–2064. IEEE.

Yi, S., Li, H., and Wang, X. (2016).
Pedestrian behavior understanding and prediction with deep neural networks.
In *European Conference on Computer Vision*, pages 263–279. Springer.

Zhang, Y., Zhou, D., Chen, S., Gao, S., and Ma, Y. (2016).

Single-image crowd counting via multi-column convolutional neural network.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597.