HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Data organisation in video surveillance systems using deep learning

Anastasiia D. Sokolova
Email: adsokolova96@mail.ru

Outline

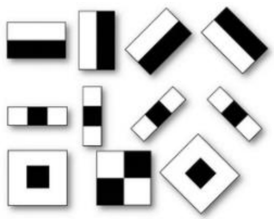
- Image recognition and faces ordering problem
- Proposed two-stage approach of organizing information in video surveillance systems
- Experimental results in face recognition
- Concluding comments and future plans

Relevance

- Automatic recognition of objects in the field of public safety
- Grouping of video data for statistics
- Face verification as a part of the identity authentication procedure

Face detection

- Haar cascades

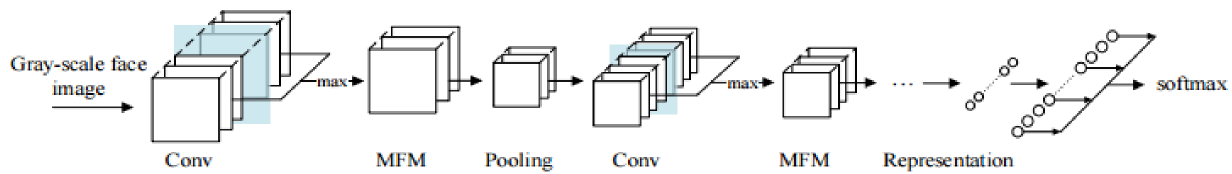


- Tensorflow

Feature extraction

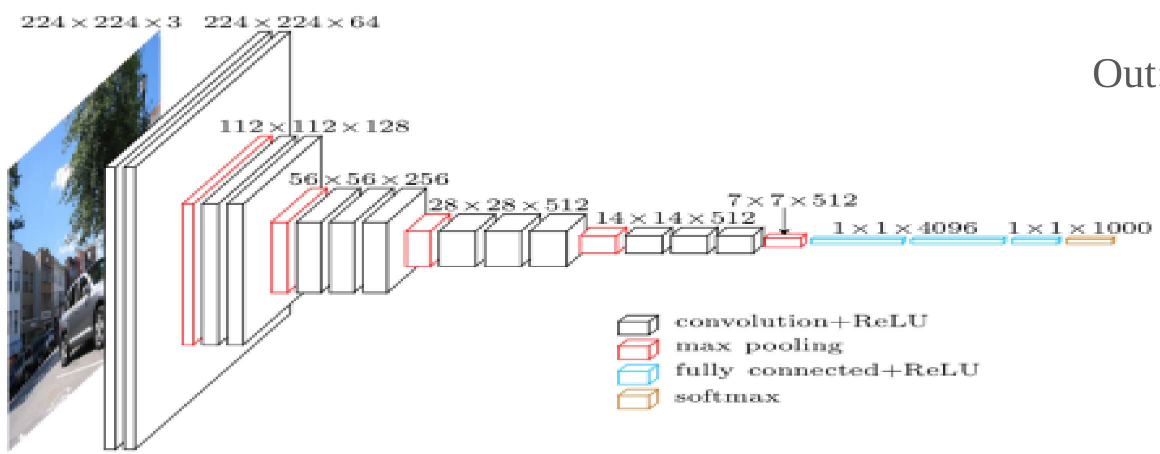
- Lightened CNN (version C)

Size: 119 Mb
Out: vector of 256 elements

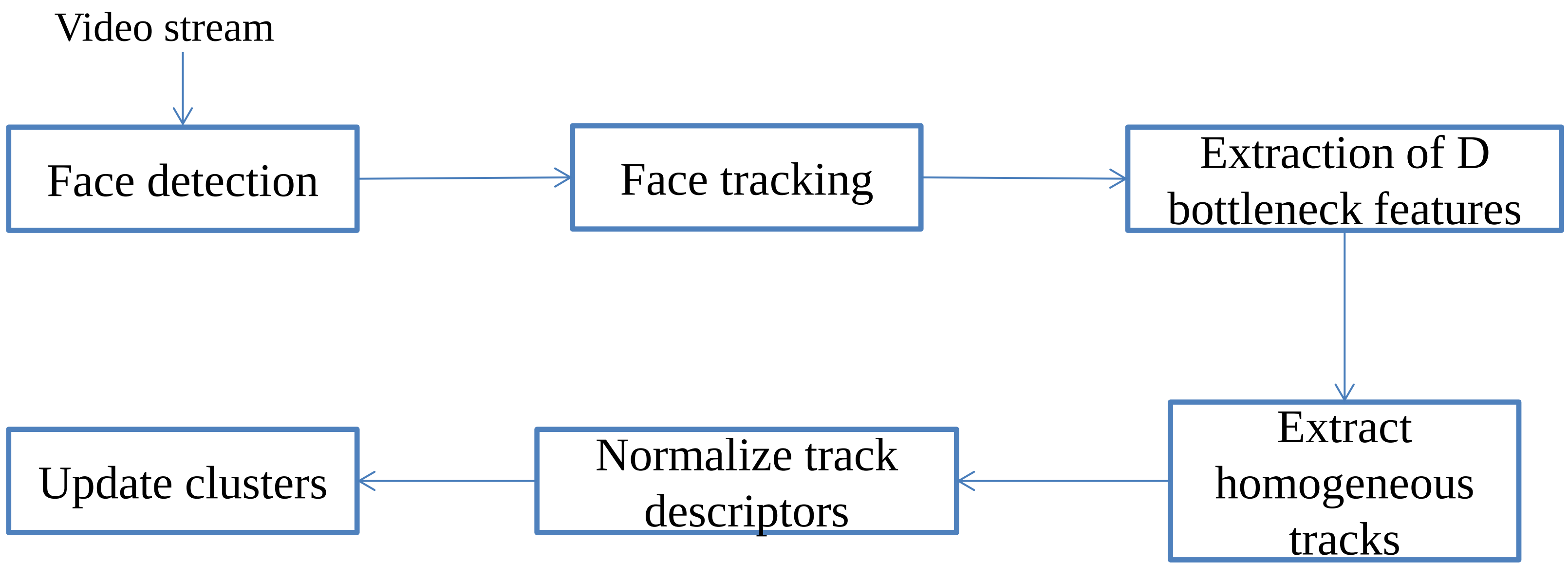


- VGGNet

Size: 553 Mb
Out: vector of 4096 elements



Proposed approach



Algorithms

- Computation of average track features

$$\rho\big(X(m_1), X(m_2)\big) = \rho\big(\bar{\mathbf{x}}(m_1), \bar{\mathbf{x}}(m_2)\big), \quad \bar{\mathbf{x}}(m_i) = \frac{1}{\Delta t(m_i)} \sum_{t'=t_1(m_i)}^{t_2(m_i)} \mathbf{x}(t) \qquad (1)$$

- Pair-wise distance between all frames:

$$\rho\big(X(m_1), X(m_2)\big) = \frac{1}{\Delta t(m_1)\Delta t(m_2)} \sum_{t=t_1(m_1)}^{t_2(m_1)} \sum_{t'=t_1(m_2)}^{t_2(m_2)} \rho\big(\mathbf{x}(t), \mathbf{x}(t')\big) \qquad (2)$$

- Distance between medoids of tracks:

$$\rho\big(X(m_1), X(m_2)\big) = \rho\big(\mathbf{x}^*(m_1), \mathbf{x}^*(m_2)\big), \mathbf{x}^*(m_i) = \operatorname*{argmin}_{\mathbf{x}(t), t\in[t_1(m_i), t_2(m_i)]} \sum_{t'=t_1(m_i)}^{t_2(m_i)} \rho\big(\mathbf{x}(t), \mathbf{x}(t')\big) \qquad (3)$$

- Distance between median features of tracks:

$$\rho\big(X(m_1), X(m_2)\big) = \rho\big(\mathbf{x}'(m_1), \mathbf{x}'(m_2)\big) \qquad (4)$$

Datasets

- LFW (Labeled Faces in the Wild)
  - 1680 people
  - 13000 images
  - 1-10 frames
- YTF (YouTube Faces)
  - 1595 people
  - 3425 videos
  - 48-6070 frames

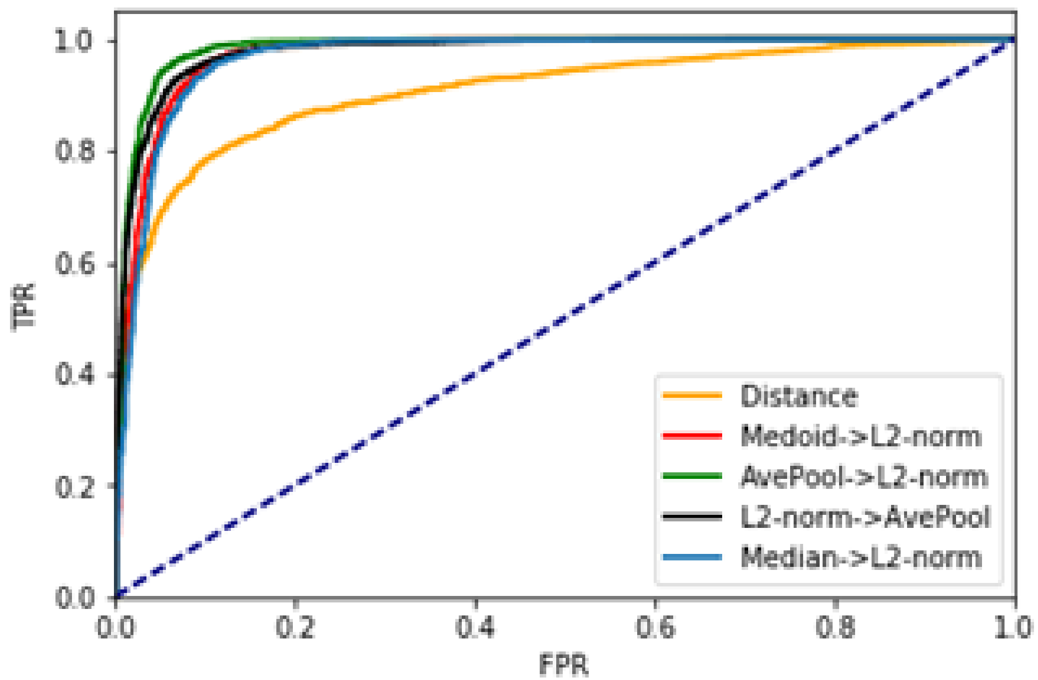
Homogeneous segmentation

0.4285

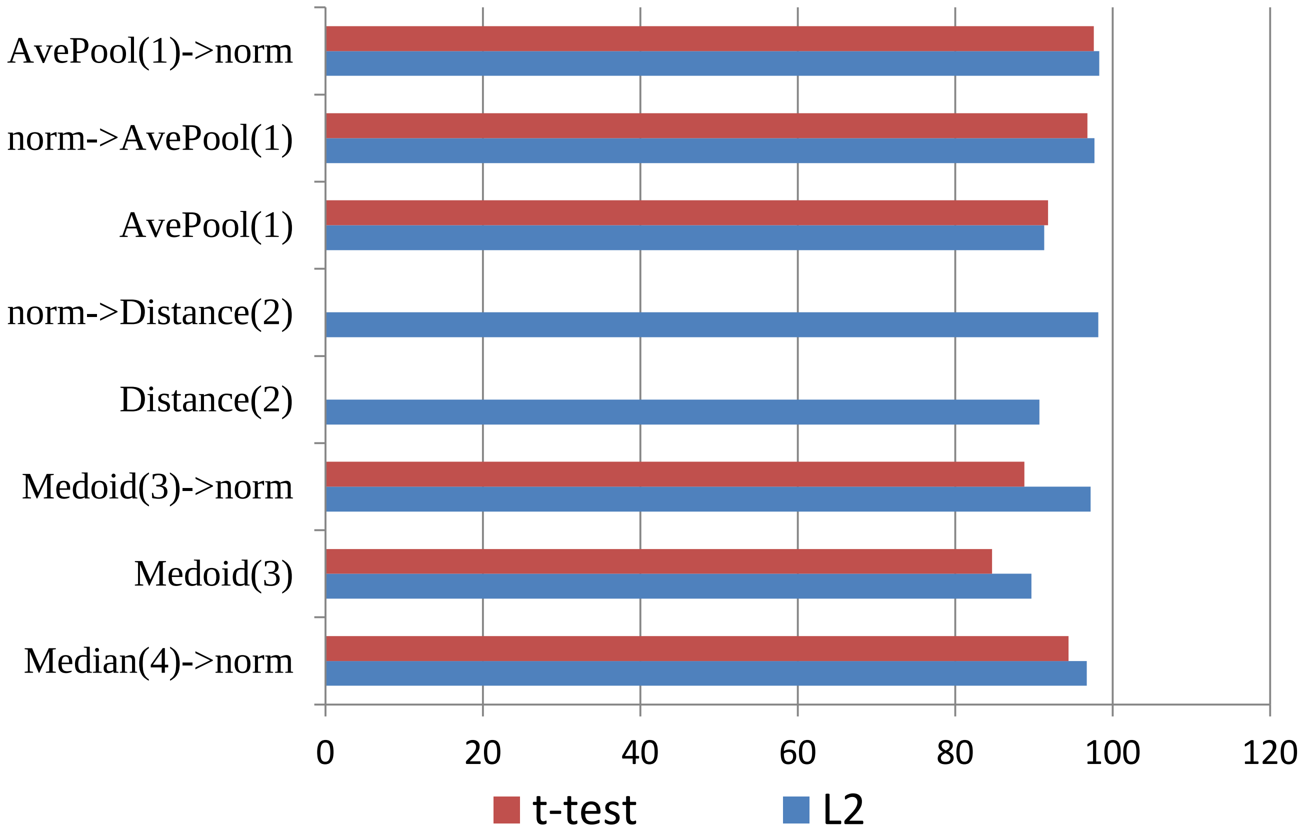
0.521

0.8934

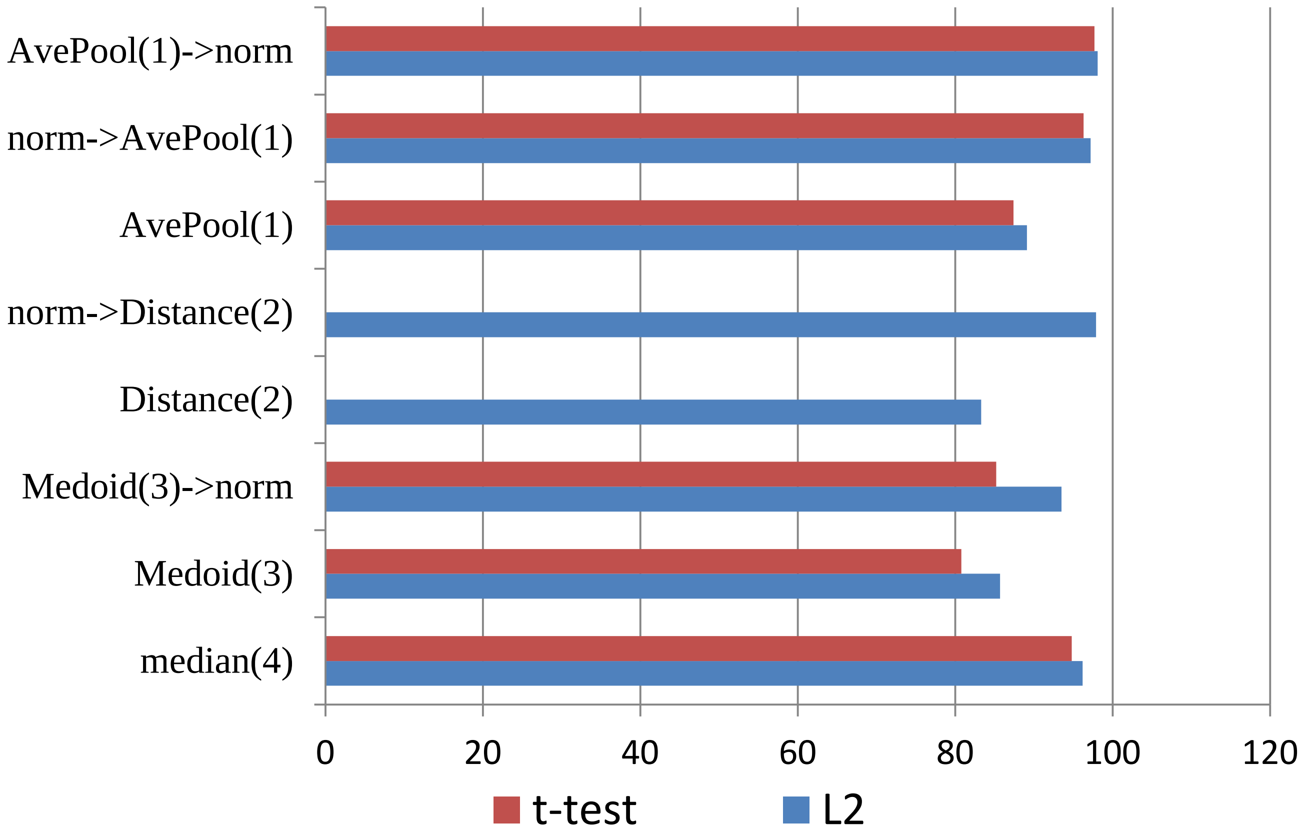
Roc-curves(Lightened CNN)



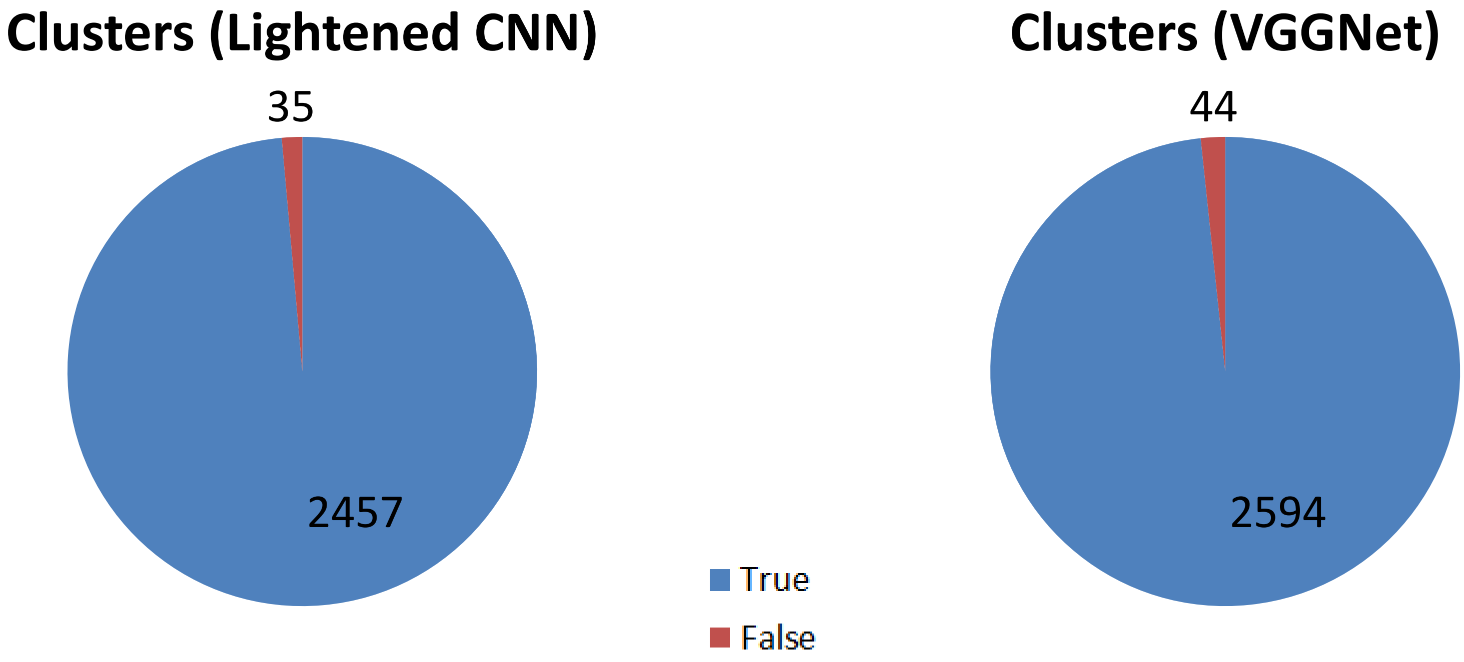
Area under curve (Lightened CNN)



Area under curve (VGGNet)



Clustering results



threshold = 0.0569

Conclusion

- Our algorithm is based on the ways to efficiently compute the dissimilarity of video tracks by using rather simple aggregation techniques
- The most accurate and computationally cheap technique involves the L_2-normalization of average unnormalized features of individual frames

Future work

- Research more sophisticated distances between video tracks, e.g., metric learning or statistical homogeneity testing
- Usage other clustering methods, e.g., approximate nearest neighbor search

Thank you for attention!