Volume XXVI 2023
ISSUE no.2
MBNA Publishing House Constanta 2023

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Face extraction and clustering with Viola - Jones object detection framework and T-SNE dimensionality reduction

To cite this article: S. Milutinovici, C. Răcuciu and I. Priescu, Scientific Bulletin of Naval Academy, Vol. XXVI 2023, pg. 100-107.

Submitted: 24.04.2023
Revised: 05.08.2023
Accepted: 29.08.2023

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-23-I2-012

Face extraction and clustering with Viola - Jones object detection framework and T-SNE dimensionality reduction

Milutinovici Sorin, Răcuciu Ciprian, Priescu Iustin

Titu Maiorescu University, Faculty of Informatics,
Military Technical Academy, Doctoral School,
Titu Maiorescu University, Faculty of Informatics,

Milutinovici Sorin, sorin.milutinovici@prof.utm.ro

Abstract. We investigate the possibility to use Viola-Jones [1] object detection framework through a multi-model approach to build a face extraction pipeline that will be used in video appearance tagging. Although deep convolutional neural networks have surpassed previous algorithms in performance [2], Haar Cascades needs much lower memory than CNN, does not require specialized hardware, and has lower storage requirements. Most videos will show the same face more than once, at least a few close-ups that are full frontal and well lit. We need an efficient system that will extract the best appearances. This study shows the pre-trained model selection, the fine-tuning of run-time parameters and the test. After selection of models for faces, eyes, mouths and noses and testing the right runtime parameters we were able to establish a procedure that will avoid any false positives and will produce a set of well defined faces.tart your abstract here…

1. Introduction

Video monitoring, especially TV news monitoring has a few specific challenges, when it comes to face detection and recognition. There are usually well known faces (TV employees, regular participants) and unknown faces. The well-known faces can be part of a trained face-recognition system. But, the list of people that will not appear on a regular basis is very fluid and the tagging of their faces has to be done by a human operator. Sometimes a person will be added to the trained recognition system, sometimes it is just a matter of recording the presence at any given moment. The human operator needs to be presented with very good examples of faces, and in a TV show those can usually be easily found. What is needed is a system that can extract the best version of a person's face, not all the appearances.

Haar Cascade models are fast [3] and require low hardware [4]. It is sometimes used together with more complex methods (such as Convolutional Neural Networks - e.g. [2]). Along with many face detection trained models that are readily available, eye recognition, nose recognition and mouth recognition have also been published. A multi-model approach may be exactly what we need - by keeping only the faces that have also tested positive for eyes, nose and mouth we may be able to isolate the best frames from a video.

2. Viola-Jones Object Detection Framework (Haar Cascades)

Viola-Jones object detection framework was one of the first successful methods used in face detection. Many consumer devices that have face-detection capabilities are using it due to the speed, efficiency and relatively low memory consumption of the algorithm. The main principle of Viola - Jones object detection method is to isolate a reduced set of image features that can classify together an image as a positive detection [1]. Each feature has a very small chance of detecting the target, some of them being slightly better than others. The training process identifies those features that can, together, form a strong classifier.

The features extracted for each image are called Haar-like features, due to their similarity with the wavelet sequence proposed in 1909 by Alfréd Haar [5]. Each feature has one or more "black" areas and one or more "white" areas and will result in a number that represents the difference between the sum of pixel intensities in the "black" areas and the number of pixel intensities in the "white" area [Fig. 1]. of the current designations.

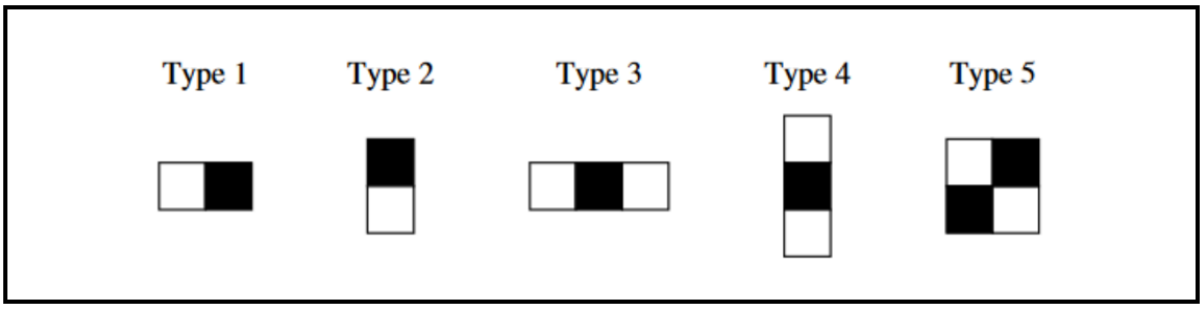

Fig. 1 Different types of Haar patterns.

O Leinhart and Maydt [6] added rotated Haar-like features with 10% - 12% increase in false positive rate (in face detection tests). The rotated Haar-like features are available in the OpenCV (Open Source Computer Vision Library) implementation and a few models used in our tests are using them (M1, M6, M7) [Table 1].

During training, all Haar-like features are calculated for every target image [7] using a fixed size target. Many face detection models are using a 24 x 24 target size which yields a number of over 160.000 possible Haar-like features (of all shapes and sizes) to be calculated for every training sample. Normally, this will be a very time-consuming operation but Viola - Jones used a clever data structure called "integral image" that simplified the computations. Integral image is an application of Crow's summed area table [8]. It involves building an alternative matrix, for every training sample, in which every cell (x,y) contains the sum of all cell values above and to the left from the original matrix (x', y'), including the row and column of the cell itself [Equation 1].

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y') \qquad (1)$$

This data structure allows the sum of pixels from arbitrary large areas to be calculated in constant time.

The training process proceeds then to select only a few of these weak classifiers using a modified version of AdaBoost algorithm [9]. Usually, a few thousand features will remain and those will be used during detection. The object detection process will scan the detector many times on the same image – every time with a new size. Since most of these windows will not contain the target object and only a few of them will, the Viola-Jones algorithm is not built to find objects but, instead, to discard

non-objects as fast as possible. Therefore, a set of strong classifiers is called by Viola-Jones an attentional cascade, because more attention (computing power) is directed towards non-negative regions of the image.

All in all, the Viola - Jones object detection framework, also known as "Haar cascades" is a very efficient, fast and low on resource demands in the detection phase. The quality of the detection itself is highly dependent on the target object type - face detection works extremely well, others less so [10].

3. Testing process

3.1. Test data

We prepared a dataset of 2000 images, of which 1000 are positives (they contain one or more faces, noses, eyes and mouths) and 1000 are negatives (no faces). The positive images are a subset of the Flickr Face HQ dataset [11], resized to 1000 pixels on the larger edge and the negative images are mainly taken from the COCO database [12]. The positive images were annotated with rectangles for each feature present (faces, eyes, noses and mouths).
For video testing, we prepared 20 video materials, of various length, consisting of panel interviews, news broadcasts or video podcasts.

3.2. Model selection

We used the OpenCV implementation of Haar cascade. OpenCV offers a set of pre-trained models for face and eye detection. Other models are available - we selected models published by Castrillon et al [10] for nose and mouth detection. A list of models can be found in Table 1. At this stage, we will try to avoid training our own models - the models available are well tested and perform reasonably well.

TABLE I. LIST OF PRE TRAINED HAAR MODELS USED IN TESTING

ID	Model name	Target object	Distribution	Published by
M1	Frontal eye detector with better eyeglasses handling	eye	OpenCV (eye_tree_eyeglasses.xml)	Shameem Hameed [13]
M2	Frontal eye detector	eye	OpenCV (eye.xml)	Shameem Hameed [13]
M3	Frontal face detector	face	OpenCV (frontalface.xml)	Rainer Lienhart [14]
M4	Alternative frontal face detector	face	OpenCV (frontalface_alt_tree.xml)	Rainer Lienhart [14]
M5	Default frontal face detector	face	OpenCV (frontalface_default.xml)	Rainer Lienhart [14]
M6	Left eye detector	eye	OpenCV (lefteye_2splits.xml)	Shiqi Yu [13]
M7	Right eye detector	eye	OpenCV (righteye_2splits.xml)	Shiqi Yu [13]
M8	Mouth detector	mouth	Modesto Castrillon-Santana (mcs_mouth.xml)	Castrillon-Santana [10]
M9	Nose detector	nose	Modesto Castrillon-Santana (mcs_nose.xml)	Castrillon-Santana [10]

Due to the nature of the detection process (sliding window), for each detected object (true positive or false positive) the classifier will output more than one bounding box. Typically, a true positive will have more adjacent bounding boxes. To avoid false positives, the OpenCV detector uses a neighborhood approach through a very important parameter called minNeighbors. minNeighbors determines the number of detected neighbors required to pass as a detected object bounding box.

If a low value of minNeighbors is used, the result will contain false positives. If a large value of minNeighbors is used, the result will contain false negatives. The value of minNeighbors depends loosely on the complexity of the structure detected - a face that has many more features will require a lower minNeighbors than an eye. In a normal image, with many objects, there are many things that may "look like an eye" - that will yield a few positive detection.

Another parameter that is somewhat related to the total number of "detection boxes" - true or false - is the scaleFactor. As the detection process slides the detector over the image, it will increase the detector size by scaleFactor after every complete pass. A very small scaleFactor will trigger many more passes of the detector and will increase the detection time. At the same time, it will yield more neighbors, because detector windows with very close sizes will yield similar results.

To ensure the best results, each model was tested on the same dataset with various minNeighbors values. We kept the scaleFactor fixed at 1.025 (meaning that each increase will make the detector window 2.5% larger.). A more fine value was deemed unsuitable because of the significant increase in detecting time. The tests follow the value of F1 score [Equation 2], calculated by comparing the detection results with the annotations of the dataset (tp = true positives, fp = false positives, fn = false negatives).

$$F1 = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \qquad (2)$$

As we increase the minNeighbors, we expect the F1 score to increase (due to decreasing false negatives) and then to start decreasing (when minNeighbors is too large and we start to lose positives). This is indeed the case for mouths, noses and eyes [Fig. 2, Fig. 3 and Fig. 4]. The face detectors are relatively stable [Fig. 5].

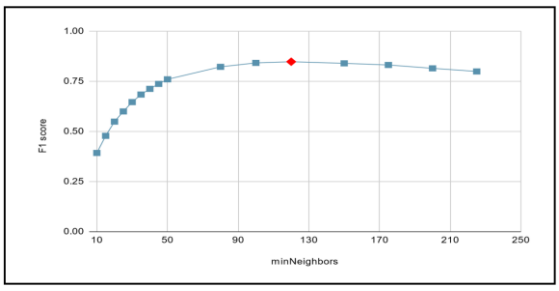

Fig. 2. M9 Noses model - max F1 score is 0.8480

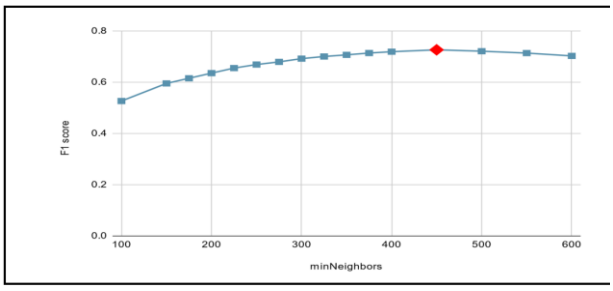

Fig. 3. M8 Mouth model - max F1 score is 0.72677

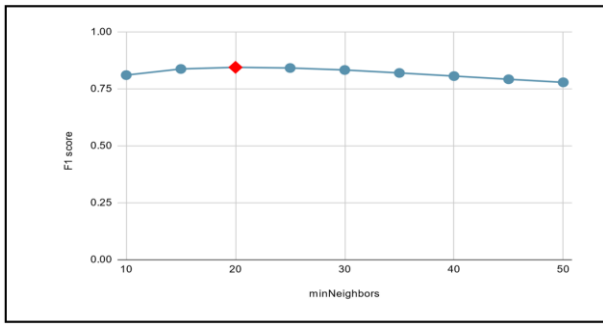

Fig. 4. M1 Frontal eye detector - max F1 is 0.8449

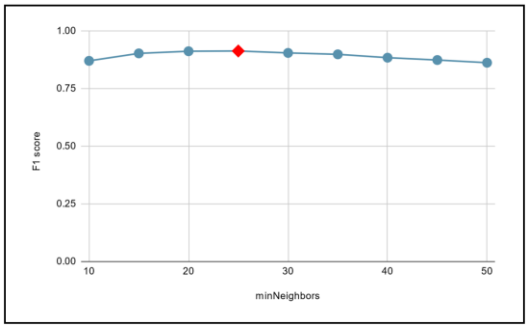

Fig. 5. M3 Frontal face detector - max F1 is 0.9133

The mouth model needs by far the largest minNeighbors parameter - 450, followed by the nose model at 120. M2 frontal eye detector reaches its maximum F1 score at 100 minNeighbors, while the rest of the models require values between 10 and 50. In the case of M6 and M7 the F1 scores may be lower than deserved, since the testing process did not consider the left/right eye preference. A visual inspection of detection results did not reveal enough ability to discriminate between right/left eye to consider using two models instead of one for eye detecting.

TABLE II. HIGHEST SCORES WITH CORRESPONDING MINNEIGHBORS VALUES

ID	Model	minNeighbors	F1 score
M1	Frontal eye detector with better eyeglasses handling	20	0.8449
M2	Frontal eye detector	100	0.6692
M3	Frontal face detector	25	0.9133
M4	Alternative frontal face detector	10	0.6192
M5	Default frontal face detector	50	0.8867
M6	Left eye detector	20	0.6833
M7	Right eye detector	25	0.6475
M8	Mouth detector	450	0.7257
M9	Nose detector	120	0.848

We selected the models with the best F1 scores for the image extraction procedure: M3 for face detection, M1 for eye detection, M8 for mouth detection and M9 for nose detection.

We tested the face extraction model on video streams containing news broadcasts, interviews or video podcasts. The full workflow is as follows:

- The video file is split into individual frames.
- If there are very similar consecutive frames those will be discarded, using structural similarity. We use a score of 0.75 to 0.90 to discard similar frames.
- We extract perfect and complete faces (according to the definitions in Table 3) from every frame.

The distribution of various types of faces in video materials is different.

4. Image testing results

The image test dataset contains a total of 1519 faces (some photos have more than one face). The face detector model was able to detect 1224 of them, out of which 28 false positives. We split the resulted images in the following categories: a) „**Perfect faces**" (detected faces that contain two eyes, one nose and one mouth inside the detected face bounding box) [Fig. 6], b) „**Complete faces**" (detected faces that contains at least an eye, at least a mouth and at least a nose inside the detected face bounding), c) „**Silent faces**" (detected faces that contain eyes and noses inside the detected face bounding) and d) „**Discarded faces**" (any other combination). All false positives of the face detector were in the discarded faces group.

TABLE III. DISTRIBUTION OF DETECTED FACES IN GROUPS

Group	Nr. of faces extracted	Percent from total faces
Perfect faces	441	36% of total faces
Complete faces	299	24% of total faces
Silent faces	169	14% of total faces
Discarded faces	315	26% of total faces

A few of the extracted faces from the „Perfect Faces" group can be seen in [Fig. 6]. Visual inspection of all the samples shows that the model can identify faces of various shapes, ages and faces with glasses. The most unstable of all models is (as expected) M8, the mouth model. A significant number of faces extracted in the "Complete faces" group are actually faces with additional mouths detected, usually on the eyes.

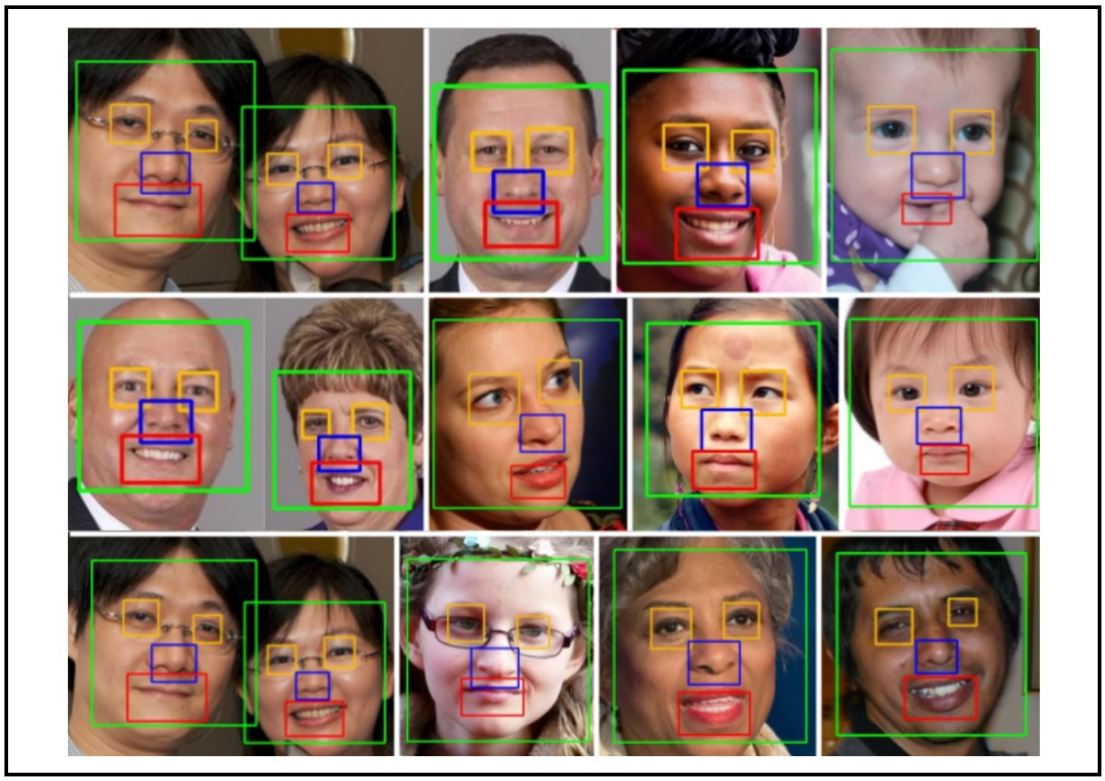

Fig. 6. Examples of extracted "perfect faces"

5. Video testing results

Since in all processed video clips the people were speaking, the mouth model was much less effective with a lot of false positives. Extracting complete and perfect faces with a 0.90 structural similarity retention coefficient will yield from 3% of the frames to 33%.

TABLE IV. DISTRIBUTION OF DETECTED FACES TYPES IN VIDEO

Group	Average video percentage
Perfect faces	6.78
Complete faces	29.5
Silent faces	1.39
Discarded faces	69.33

6. Clustering of faces extracted from video.

Final goal of this project is to be able to present to a human operator a very succinct report of people that were identified in the videos. We do not aim for automatic face recognition since this is easily achieved by the human operator. But we hope for a correct grouping of the identified faces.

After extracting all perfect and complete faces we investigated whether various image features (SIFT key points, edges, and many others.) could be used to cluster the faces in groups that match the faces of the people. The most successful attempt was the simplest one, based on pixel intensities:

- Resize each face to 60/60 pixels.
- Convert to LAB color space.
- Extract the intensities of every pixel's L channel in a vector of 3600 parameters.
- Apply a dimensionality reducing algorithm (T-SNE [15] – scikit implementation) to obtain a two-dimensional representation of each image.
- Apply a density-based clustering algorithm (DBSCAN [16] – scikit implementation) to separate each cluster of images.

Our tests have shown that the separation efficiency of this method is very good - if we manually select the best parameters for DBSCAN algorithm we could in all cases separate the perfect and complete faces into correct groups (such as in Fig. 8).

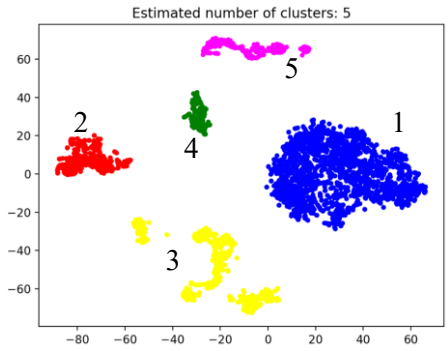

Fig 7 Too many clusters: 5,4 and 3 are in fact the same face.

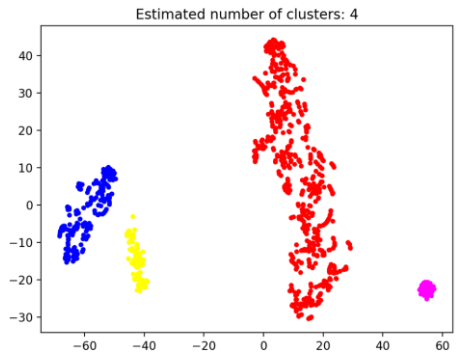

Fig. 8. Perfect number of clusters – each corresponds to a group of faces belonging to the same person.

Automatically selecting DBSCAN parameters is possible (based on the extent of T-SNE coordinates space). When using automatic parameters, we run the model for a few possible combinations, and we select the parameters that will yield the largest number of clusters with minimum amount of noise points (i.e., points that have hot been associated to a cluster by DBSCAN).

7. Conclusions and future developments.

A multi-model approach for face extraction in this case is possible. The detection process works relatively well, especially if we allow for less than perfect faces (that is two eyes, one nose and one mouth). There are still problems with people with sunglasses or normal glasses that have powerful reflections (the latter case may be solved in video) .

The possible increase of scaleFactor may be interesting, since it will lead to faster detection time. Due to its connection to minNeighbors, more tests will have to be run to determine how far we can go with it. The stability of the face detector models makes it a good candidate to start experimenting with increasing scaleFactors - it will not impact minNeighbors much. The mouth and nose model may need to remain at small scaleFactors because the target objects are relatively small in most images.

The video testing has proved that is at least possible to obtain clusters of images that belongs to the same faces when working with well-lit, well-prepared video interviews. Even when working with automatic DBSCAN parameters, the clusters are well separated so that they can be subsequently joined by an operator.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[2] R. Andrie Asmara, M. Ridwan and G. Budiprasetyo, "Haar Cascade and Convolutional Neural Network Face Detection in Client-Side for Cloud Computing Face Recognition," 2021 International Conference on Electrical and Information Technology (IEIT), 2021, pp. 1-5, doi: 10.1109/IEIT53149.2021.9587388.

[3] Guennouni, Souhail & Ali, Ahaitouf & Mansouri, Anass. "A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms" Modelling and Simulation in Engineering, 2015, pp. 1-8. doi: 10.1155/2015/948960.

[4] S. Guennouni, A. Ahaitouf and A. Mansouri, "Multiple object detection using OpenCV on an embedded platform," 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377, doi: 10.1109/CIST.2014.7016649.

[5] Haar, A. "Zur Theorie der orthogonalen Funktionensysteme" Math. Ann. 69, 331–371, 1910. doi: 10.1007/BF01456326

[6] Lienhart, Rainer and Jochen Maydt. “An extended set of Haar-like features for rapid object detection.” Proceedings. International Conference on Image Processing 1, 2002: I-I.

[7] Jensen, Ole Helvig. “Implementing the Viola-Jones Face Detection Algorithm.”, 2008.

[8] Crow, Franklin C.. “Summed-area tables for texture mapping.” Proceedings of the 11th annual conference on Computer graphics and interactive techniques. 1984, vol. 18, n. 3, pp. 207-212

[9] Schapire, Robert E. and Yoram Singer. “Improved Boosting Algorithms using Confidence-Rated Predictions.” COLT, 1998.

[10] Castrillón Santana, Modesto & Deniz, Oscar & Hernández-Sosa, Daniel & Lorenzo-Navarro, Javier. "A comparison of face and facial feature detectors based on the Viola–Jones general object detection framework" Machine Vision and Applications, 2011, vol. 22. pp. 481-494. doi: 10.1007/s00138-010-0250-7.

[11] Karras, Tero & Laine, Samuli & Aila, Timo. "A Style-Based Generator Architecture for Generative Adversarial Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 4396-4405. doi: 10.1109/CVPR.2019.00453.

[12] Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick. “Microsoft COCO: Common Objects in Context.” ECCV., 2014.

[13] The Open Computer Vision Library Source Code, https://github.com/opencv/opencv

[14] Lienhart, Rainer & Kuranov, Alexander & Pisarevsky, Vadim. "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection", Proceedings of the 25th Pattern Recognition Symposium; 10-12 September 2003, vol. 2781, pp. 297-304. doi: 10.1007/978-3-540-45243-0_39.

[15] van der Maaten, Laurens & Hinton, Geoffrey. (2008). Viualizing data using t-SNE. Journal of Machine Learning Research. 9. 2579-2605.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.