Playback detection using machine learning with spectrogram features approach

Jerzy Dembski, Jacek Rumiński
Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, ul. Narutowicza 11/12, 80-952 Gdańsk,Poland
Email: dembski@ue.eti.pg.gda.pl, jacek.ruminski@pg.gda.pl

***Abstract*—This paper presents 2D image processing approach to playback detection in automatic speaker verification (ASV) systems using spectrograms as speech signal representation. Three feature extraction and classification methods: histograms of oriented gradients (HOG) with support vector machines (SVM), HAAR wavelets with AdaBoost classifier and deep convolutional neural networks (CNN) were compared on different data partitions in respect of speakers or playback devices: for instance with different speakers in training and test subsets. The playback detection systems were trained and tested on two speech datasets S_1 and S_2 manufactured independently by two different institutions. The test error for both datasets oscillates about the level of 1% for HOG+SVM and even below it for CNN in bigger S_1 base. In cross validation scenario in which one base was used for training and second base for the test the results were very poor what suggests that the information relevant for playback detection appeared in each base in different way.**

1. Introduction

In the recent years the mobile banking applications or any telephone-based services have become more and more popular. One of the quickest and most convenient way of security ensuring is automatic speaker verification (ASV) by voice. There are two kinds of ASV systems: active systems which prompt a user a random sentence to repeat or demand from a user to answer an additional random question, and passive systems which are more convenient for users but less safe because of higher possibilities of effective spoofing attacks. There are four identified spoofing methods: impersonation, voice conversion, speech synthesis and playback or replay. Playback is one of the easiest and most effective spoofing attack types against ASV systems. An authentic voice is recorded and reproduced using mobile devices or more specialized equipment.

There are two groups of playback detection methods: based on speech signals similarity measurement and based on distinguishable features between the authentic and playback signals.

The first group of methods [1], [2], [3] requires text-dependent ASV version in which the user is obligated to speak fixed passphrase, for instance "Log me into my account". Speech signal from phone is compared to earlier recorded signals from the dataset. In practice all signals are processed to special form before comparing to avoid time and memory excessive complexity. The playback attack is detected if current signal is too similar to one of the signals stored in the database. This approach needs the assumption that each user utterance of the passphrase text is recorded and stored in the database. Generally this assumption is in opposition to other kinds of attacks like impersonation, speech synthesis and voice conversion which need spoofing detection based rather on differences between signals. Another drawback of this group of methods is due to the fact that the signal may be corrupted by channel noise or by intentional action of the impostor and can be classified as authentic one.

In the second group of playback detection methods the characteristic features are searched to distinguish the playback from authentic signals. In [4] a high-pass filter was used to disclose low frequency differences which is the authentic signal and which is the playback one. Cepstral coefficients and additional statistical features were used for playback detection by support vector machines (SVM) classifier with radial basis function (RBF) kernel. In [5] the SVM classifier with RBF kernel was used also with a bit different feature set customized to the assumption that a far field microphone is usually used in playback attack which cause some phenomena like noise and reverberation level increasing which distinguishes playback from the authentic signal. In [6] another kind of low frequency noise was observed and utilized for playback detection. The pop-noise arises when the air from the lungs hits a microphone and is present only in an authentic signal because of usual weak quality of loudspeakers which skip low frequencies. Such an authentic signal fingerprint can be detected depend on weak playback equipment quality and the assumption that the authentic signal is not filtered by a mobile device. Another amplitude decreasing effect in low frequencies region in spectrogram image of playback signal was observed and described in [7]. Two types of deep neural networks (DNN) were compared to several classical methods of feature extraction and classification in playback detection system.

As was arising in literature, different playback attack conditions and different kinds of attack cause different signal distortions. The effective countermeasure in each situation requires special method of feature extraction but all of the methods will be ineffective if new, previously unknown kind

of playback attack. It suggests to search an universal set of features by machine learning methods. In [8] the local binary patterns (LBP) and local ternary patterns (LTP) based on spectrograms with SVM as classifier were compared to classical methods based on mel frequency cepstral coefficients (MFCC). Moreover, two datasets were used for training and testing purposes. One of the datasets VL-Bio created by VoiceLab company was used also in this work as the S_2 dataset.

In this work HOG features as another kind of feature set based on spectrograms was compared to classical graphical pattern detection approach HAAR+AdaBoost known from face detection task [9] and more modern deep convolutional neural networks approach (CNN) which can exploit raw spectrograms to find internal representation of optimal features.

2. Feature extraction and classification methods

In this section the 3 recently most popular methods of image feature extraction and classification: HAAR+AdaBoost, HOG+SVM, Convolutional Neural Networks were described.

2.1. HOG+SVM

Histograms of Oriented Gradients (HOG) are successfully used in pattern matching tasks like stereography matching or searching by images [10]. One of the most famous application was human silhouette detection system which was robust for occlusions and a great number of body parts configurations [11]. The gradient direction Ω is calculated by equation:

$$\Omega = \arctan \frac{G_y(x,y)}{G_x(x,y)}, \tag{1}$$

where $G_x(x,y)$ and $G_y(x,y)$ are gradient values in vertical and horizontal directions in surrounding of the pixel (x,y) calculated by equations:

$$\begin{aligned} G_x(x,y) &= I(x+1,y) - I(x-1,y) \\ G_y(x,y) &= I(x,y+1) - I(x,y-1), \end{aligned} \tag{2}$$

where $I(p,q)$ - pixel (p,q) intensity. In the case of usage of 360° range gradient direction as in this work the absolute part of Ω should be increased by π if $G_x(x,y) < 0$

Each histogram is created as a vector of N bars as was presented in Fig. 1. Each bar corresponds to one fixed gradient direction and is calculated as a sum of weights for all pixels which belong to particular L_x on L_y window. The nonzero weight value is possible when the fixed direction is the nearest at left or right side to gradient direction as was shown in Fig. 1 b. The weight value for fixed direction i is calculated from equation: $w_i = (1 - \frac{\beta_i}{\gamma})G(x,y)$, where β_i is an angle between fixed direction i and gradient direction, γ is an angle between two neighboring fixed directions, $G(x,y) = (G_x^2(x,y) + G_y^2(x,y))^{0.5}$ is the gradient value at the point (x,y). The histogram adjustment is repeated

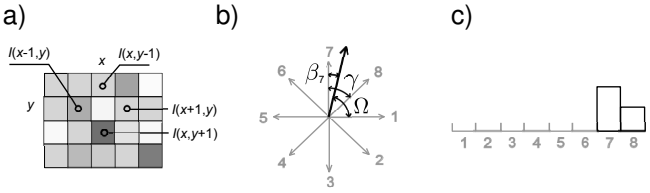

Figure 1. The HOG histogram adjustment by example (x,y) pixel gradient direction: a) neighboring pixel intensities used in calculation, b) fixed directions and gradient direction, c) histogram adjustment vector.

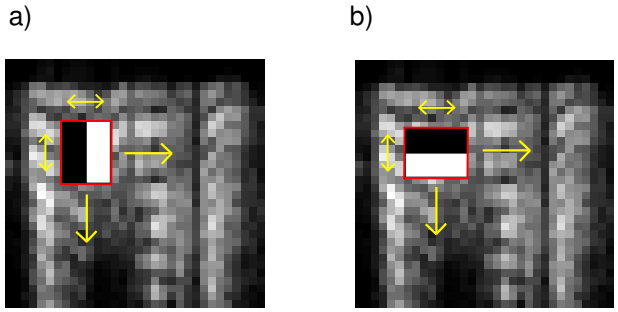

Figure 2. Example HAAR features: the difference of mean pixel intensities between rectangular regions with a) vertical, b) horizontal neighboring orientation.

for each pixel in L_x on L_y window. The window is moved in vertical and horizontal direction with a certain step.

In this work some HOG parameter values were selected "by hand" and in the best configuration the window size $L_x = L_y = 3$, the window shift step equals 1 pixel, the number of bars $N = 8$, $\gamma = 45^\circ$ is due to 360° range of gradient direction. For 32x32 images obtained from spectrograms (see Section 3) the number of features is equal $(32 - Lx + 1)(32 - Ly + 1) * 8 = 30 * 30 * 8 = 7200$.

The histogram features vectors are treated as an input to SVM classifier with linear kernel function [12]. The open library LibSVM [13] was used for training and testing purposes.

2.2. HAAR+Adaboost

The main idea of AdaBoost classifier training algorithm [14] is to build strong classifier with a sequence of the most simplest classifiers named weak classifiers. Each weak classifier in the sequence is chosen from huge number depend on sum of weighted error in a way first of all to correctly classify the most difficult training examples for the current sequence. In this way the generalization is quite good but classification process is not so computationally complex which is important in mobile devices and obtained also by cascade version and integral images in the case of the composite of Haar features with AdaBoost classifier [15].

Haar rectangular features shown in Fig. 2 are created by feature window scaling and shifting in vertical and horizontal direction.

The additional advantage is due to easy float control of false negative and false positive error which is important in the situation when blocking of an intruder is more important than allowing a legal user pass and on the contrary in opposite situation. The main drawbacks of this method are great training complexity and that AdaBoost classifier can't cope with statistically dependent features which is not a problem in SVM or CNN classifiers.

2.3. Convolutional Neural Networks (CNN)

The CNN also known as a particular kind of deep neural networks (DNN) approach is applied to image recognition tasks due to usage of adaptive 2D filters in early layers. In this work CNN was used only with S_1 base because S_2 base do not contain enough high number of training examples.

After some trials the suitable net architecture and training parameters for 32x32 spectrogram images were chosen. The network implemented in Tensorflow library is made up with 4 layers. First two layers are convolutional with respectively 20 7x7 and 40 5x5 filters with ReLU activation function. Thanks to maxout 2x2 technique both layers reduce image size twice, so the output of second layer corresponds to 40 8x8 images. Each of $40 * 8 * 8 = 2560$ outputs is connected to 500 neurons in fully connected third layer with ReLU activation function. The last layer contains 2 output softmax neurons: one for the playback and one for legitimate trial recognition. The adaptive moment optimization was used during learning process with cross entropy error function.

3. Preliminary data processing

The datasets used in experiments are considerably different. The S_1 dataset contains only one short sentence spoken by many speakers many times in noisy environment e.g. with many additional sounds in the background. Playbacks were synchronized with original recordings so there is no need to shift time domain sample sequences in any direction which is not true in S_2 dataset where a special synchronization algorithm must be applied. Such synchronization is easy in text-dependent approach [3] where a classified signal is matched to the reference signal, provided that only one password sentence is admissible which is not true in the dataset S_2 where many different sentences can be used as a password. In accordance to this the only way to make signals comparable is to recognize the beginning of the utterance. In this work it is implemented by the quotient of mean signal amplitude absolute value in fixed range (300 samples in this work) to mean signal amplitude absolute value in whole range of signal. The new starting point of the signal is fixed if this quotient is equal or greater than 0.75. Both numerical values are chosen by several trial and error attempts.

The spectrograms were done using 512 length Hamming window with 32 shift. Each spectrogram was reduced by choosing frequency range (the stripe number) and 4 times resolution reduction in time domain to 32x32 image. In the final processing step each spectrogram is normalized to have the same fixed mean amplitude absolute value in both S_1 and S_2 datasets.

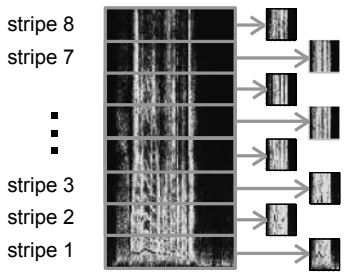

Figure 3. Spectrogram reformulation by partition into stripes and horizontal resolution reduction.

4. Experiments and results

Two datasets of positive and negative examples S_1 and S_2 were used in experiments.

S_1 dataset contains near 4000 wave recordings with one original utterance spoken by several different speakers many times and 5 playback wave files for each original recording which was played by 5 different mobile devices. S_2 dataset contains 120 original recordings spoken by 5 speakers and 4 playback wave files played by 4 devices. Both sets were created independently in different conditions by different institutions. The S_1 dataset use 8 kHz sampling frequency while S_2 dataset is available in 8 kHz and 16 kHz versions.

First challenge was to find the best method of feature extraction and the best classification algorithm. In the situation that there is no any obvious knowledge about it we decided to try three approaches which seems to be the most proper for voice signals classification. In all of them the spectrograms are used as first-stage feature representation. Due to high size of spectrogram with appropriate resolution the preliminary choice of the most relevant region is important to avoid excessive computational complexity. The 128x128 spectrogram was divided into 8 strips in respect of frequency range in vertical direction as is showed in Figure 3. The horizontal resolution of each stripe was reduced 4 times to obtain 32x32 image. Table 1 presents the test error for two different methods of spectrogram feature extraction and classification. The error is presented by three values: false rejection rate (FRR) which is a number of authentic recordings classified as playback ones divided by a number of authentic recordings, false acceptance rate (FAR) which is a number of playback recordings classified as authentic ones divided by a number of playback recordings and error rate (ER) which is computed simply as a number of false classified examples divided by a number of all examples. The distinctive best results of both methods were obtained using stripe 1 which corresponds to low frequency range in the dataset S_2. The results for Haar features and AdaBoost classifier are considerably worse than for HOG+SVM but

TABLE 1. PERCENTAGE CLASSIFICATION TEST ERROR FOR S_2 DATASET WITH 16 KHZ SAMPLING FREQUENCY WITH RANDOM PARTITION - ABOUT A HALF OF 640 EXAMPLES FOR CLASSIFIER TRAINING AND A HALF FOR TEST IN RESPECT OF FREQUENCY RANGE SPECTROGRAM STRIPES. THE STRIPE 1 RESPONDS TO LOWEST FREQUENCY RANGE

	HOG+SVM			HAAR+Adaboost		
	FRR	FAR	ER	FRR	FAR	ER
stripe 1	1.25	0.0	0.3	8.8	3.8	5.0
stripe 2	30.0	6.3	12.2	37.5	23.3	26.9
stripe 3	40.0	8.3	16.3	30.0	26.3	27.2
stripe 4	40.0	15.4	21.6	32.5	17.5	21.3
stripe 5	55.0	7.5	19.4	37.5	25.4	28.4
stripe 6	56.2	10.4	21.9	52.5	27.1	33.4
stripe 7	43.8	9.6	18.1	42.5	29.2	32.5
stripe 8	51.3	10.4	20.6	37.5	21.7	25.6

TABLE 2. PERCENTAGE CLASSIFICATION TEST ERROR FOR S_1 DATASET WITH RANDOM PARTITION - ABOUT HALF OF 24000 EXAMPLES FOR CLASSIFIER TRAINING AND HALF FOR TEST IN RESPECT OF FREQUENCY RANGE SPECTROGRAM STRIPES

	FRR	FAR	ER
HOG+SVM			
stripe 1	0.97	0.04	0.19
stripe 2	10.8	1.62	3.12
stripe 3	19.0	3.50	6.08
stripe 4	25.9	5.20	8.65
HAAR+Adaboost			
stripe 1	0.97	0.76	0.80
stripe 2	2.52	4.10	2.74
stripe 3	23.6	9.62	11.9
stripe 4	10.9	6.69	7.37
CNN			
stripe 1	0.20	0.08	0.10
stripe 2	3.73	0.07	0.70
stripe 3	8.00	0.65	1.90
stripe 4	10.8	1.05	2.66

the AdaBoost classifier with Haar features is much quicker what is important for mobile devices.

In the next series of experiments the bigger dataset S_1 is used to confirm the usefulness of low frequency region of spectrogram and to compare three methods of feature extraction and classification. As is shown in Table 2 the results for dataset S_1 are considerably better than those for dataset S_2 probably due to higher number of examples and an another base preparation as was specified in Section 3. The low frequencies also seems to be highly more important for playback detection purposes than other frequency regions in spectrograms. The deep convolutional neural networks are significantly better then HOG+SVM and HAAR+AdaBoost maybe on account of enough high number of training examples which overcomes possible overfitting despite of lack of regularization like dropout technique.

In the next experiment the (HOG+SVM) method was used in a more demanding assumption that the system must cope with new devices and other speakers than in training. Figure 4 presents the partition of test and training subset which models such demand. The results presented in Table 3 are a bit worse than those in the experiment with random

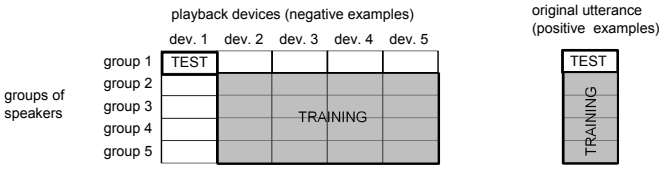

Figure 4. The choice of the test set in the assumption of original device and a person in test set.

TABLE 3. PERCENTAGE HOG+SVM CLASSIFICATION TEST ERROR RATE (ER) FOR STRIPE 1 IN S_1 DATASET FOR EACH DEVICE AND EACH GROUP OF SPEAKERS

	dev. 1	dev. 2	dev. 3	dev. 4	dev. 5
group 1	1.78	0.76	0.64	0.76	0.64
group 2	1.97	0.63	0.70	1.08	0.70
group 3	1.78	0.89	0.76	0.76	0.70
group 4	2.22	1.33	1.21	1.40	1.14
group 5	2.99	1.84	1.40	1.72	1.40

partition. The average error rate ER = 1.25%, FRR = 1.70%, FAR = 0.79%.

In the most demanding assumption, the test error is evaluated on the different dataset than used for training. For instance the S_1 dataset can be used for training and S_2 for the test. Results presented in Table 4 are very poor taking into account that a trivial classifier e.g. which classifies always to the most numerous class provides 16.7% of general error for the S_1 dataset and 25% for the S_2 dataset. So what is the reason that results are quite good when examples for training and test are from one base while in the level of trivial classifier in opposite case? The greater clarity can be provided if we compare spectrograms of original and playback recordings for both bases. In Figure 5 it can be observed that high amplitudes in lower parts of images which correspond to low frequencies of playback spectrograms are extended in compare to the original one. That characteristic "echo effect" doesn't appear in the case of playback recordings in the dataset S_2 what is shown in Figure 6. Moreover, in the dataset S_2 the lowest frequencies are strongly dumped in playback whereas other frequencies are not considerably corrupted apart some pepper noise. It can explain why in S_2 dataset good results can be obtained only using stripe 1 while in dataset S_1 more stripes are valuable especially in lower part of spectrograms. It can be probably due to different recording conditions.

In the last experiment both bases were used for training

TABLE 4. PERCENTAGE TEST ERROR FOR DIFFERENT DATASETS FOR TRAINING AND TESTING

	FRR	FAR	ER
S_1 for training, S_2 for test			
HOG+SVM	91.7	0.97	23.7
HAAR+AdaBoost	52.5	50.0	50.6
CNN	82.5	51.2	59.1
S_2 for training, S_1 for test			
HOG+SVM	63.9	11.6	20.3

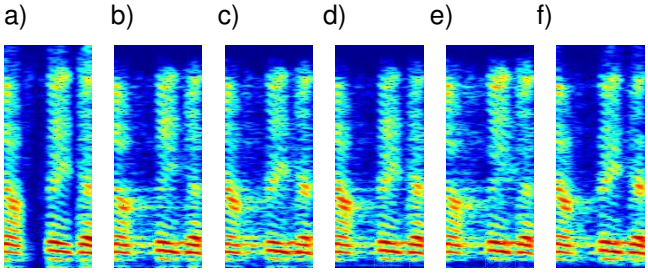

Figure 5. The spectrograms of one example utterance for S_1 dataset: a) original recording, b-f) playbacks replayed by 5 different mobile devices.

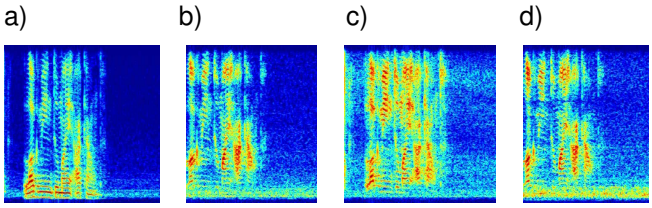

Figure 6. The spectrograms of one example utterance for S_2 dataset: a) original recording, b-d) playbacks replayed by 3 different mobile devices

and test purposes to ask a question if mixed recording conditions considerably worsen generalization. The results presented in Table 5 give a positive answer. Generalization in the dataset S_1 is slightly worse in compare to first row in Table 2 and much worse in the dataset S_2 but it is due to 40 times greater number of examples in the dataset S_1 because after reducing the number of training examples from the dataset S_1 the test error in the dataset S_2 is also reduced.

5. Conclusions

In this work the spectrogram speech signal representation was used to playback detection. Due to the 2D spectrogram image representation three graphical pattern detection methods were applied. Two datasets S_1 and S_2 were used in experiments and if each one is used individually for training and the test, the results will be quite well especially when a deep convolutional neural network (CNN) or HOG+SVM approach is used as it is shown in Tables 1 and 2. If one dataset is used for training and the other for the test, the results will be very poor as it is shown in Table 4 what suggests that features found during learning are extremely different for both datasets. Good results can be obtained when both datasets are mixed for training and the test purposes what is depicted in Table 5.

TABLE 5. PERCENTAGE TEST ERROR FOR MIXED TRAINING SET CONTAINING EXAMPLES FROM S_1 AND S_2 DATASETS USING STRIPE 1 SPECTROGRAMS

	S_1			S_2		
	FRR	FAR	ER	FRR	FAR	ER
HOG+SVM	1.06	0.56	0.81	36.3	5.00	12.8
CNN	1.21	0.01	0.21	29.1	1.23	8.51

It leads to conclusion that an effective playback detection system learning process requires a representative dataset which contains examples which reflect different playback recording conditions like different kinds of playback attack, microphone and loudspeakers positions and quality, kinds of channel noises, background noises and acoustic properties of different rooms or places.

The second conclusion is due to the lowest frequency spectrogram region (stripe 1 in Tables 1 and 2) which seemed to be the most important for playback detection taking into account currently used devices. It was confirmed by results reported in [4], [5], [6] although with different interpretations.

References

[1] Z. Wu, S. Gao, E.S. Cling and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification", *Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC)*, 2014.

[2] W. Shang and M. Stevenson, "A playback attack detector for speaker verification systems", *Communications, Control and Signal Processing ISCCSP 2008*, 3rd International Symposium on, March 2008, pp. 11441149, 2008.

[3] J. Gałka, M, Grzywacz and R. Samborski, "Playback attack detection for text-dependent speaker verification over telephone channels", *Speech Communication*, Volume 67, pp. 143–153, 2015.

[4] Z.F. Wang, G. Wei and Q.H. He, "Channel pattern noise based playback attack detection algorithm for speaker recognition", *Proceedings International Conference on Machine Learning and Cybernetics (ICMLC 2011)*, Vol. 4, IEEE, Guilin, China, pp. 17081713, 2011.

[5] J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems", *Proceedings IEEE International Carnahan Conference on Security Technology (ICCST 2011)*, IEEE, Barcelona, Spain, 2011.

[6] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen and T. Matsui, "Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification", *Proceedings Interspeech*, ISCA, Dresden, Germany, pp. 239243, 2015.

[7] D. Luo, H. Wu and J. Huang, "Audio recapture detection using deep learning", *Proceedings IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP 2015)*, IEEE, Chengdu, China, pp. 478482, 2015.

[8] M. Smiatacz, "Playback attack detection: the search for the ultimate set of antispoof features", *Advances in Intelligent Systems and Computing*, 2017, accepted for printing.

[9] M. Jones and P. Viola, "Face recognition using boosted local features", *Technical Report MERL-TR-2003-25*, Mitsubishi Electric Research Laboratory, 2003.

[10] D. Lowe, "Object recognition from local scale-invariant features", *Proceedings of International Conference on Computer Vision*, 1999.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, 2005.

[12] C. Cortes,V. Vapnik, "Support-vector networks" *Machine Learning* 20 (3), 273-297, 1995.

[13] C. Chang and C. Lin, "LIBSVM : a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[14] R.E. Schapire and Y. Freund, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods", *The Annals of Statistics*, v. 26(5), 1651–1686, 1998.

[15] M. Jones and P. Viola, "Robust Real-Time Face Detection", M. Jones, *International Journal of Computer Vision*, 57(2), pp. 137–154, 2004.