Robust Visual Voice Activity Detection Using Long Short-Term Memory Recurrent Neural Network

Zaw Htet Aung and Panrasee Ritthipravat(✉)

Artificial Intelligence in Medicine Laboratory,
Department of Biomedical Engineering, Mahidol University,
25/25 Puttamonthon 4, Salaya 73170, Nakhon Pathom, Thailand
z.zawhtet.a@ieee.org, panrasee.rit@mahidol.ac.th

Abstract. Many traditional visual voice activity detection systems utilize features extracted from mouth region images which are sensitive to noisy observations of the visual domain. In addition, hyperparameters of the feature extraction process modulating the desired compromise between robustness, efficiency, and accuracy of the algorithm are difficult to be determined. Therefore, a visual voice activity detection algorithm which only utilizes simple lip shape information as features and a Long Short-Term Memory recurrent neural network (LSTM-RNN) as a classifier is proposed. Face detection is performed by structural SVM based on histogram of oriented gradient (HOG) features. Detected face template is used to initialize a kernelized correlation filter tracker. Facial landmark coordinates are then extracted from the tracked face. Centroid distance function is applied to the geometrically normalized landmarks surrounding the outer and inner lip contours. Finally, discriminative (LSTM-RNN) and generative (Hidden Markov Model) methods are used to model the temporal lip shape sequences during speech and non-speech intervals and their classification performances are compared. Experimental results show that the proposed algorithm using LSTM-RNN can achieve a classification rate of 98 % in labeling speech and non-speech periods. It is robust and efficient for realtime applications.

Keywords: Visual voice activity detection · Long short-term memory · Recurrent neural network · Supervised sequence classification

1 Introduction

Voice activity detection plays a significant role in enabling natural and spontaneous Human-Computer and Human-Robot Interaction applications. However, voice activity detection based solely on audio modality faces various challenges in real world environments. The fact that visual information is complimentary to acoustic speech signal has been well founded in the literature. The influence of visual features on the perception of speech has been demonstrated in [1]. It is

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 380–391, 2016.
DOI: 10.1007/978-3-319-29451-3_31

well-known in the literature as the McGurk effect. Also, if there is noise corruption in the environment, the availability of lip movement data grants a person an extra 4–6 dB of noise tolerance compared to audio data alone [2].

1.1 Related Works

Many works in the area of visual voice activity detection utilize mouth region image intensity based features. Siatras et al. [3] used variations in the amount and intensity of mouth region pixels as cues for voice activity detection. A case-specific threshold was needed to identify relevant pixels. Ahmad et al. [4] presented a method where changes in the mean intensity values of mouth area during speech and silence periods were modeled by Gaussian Mixture Models. These methods rely solely on image intensities which make them unsuitable for adverse lighting conditions. In [5], Song et al. described a method based on chaos inspired similarity measure to mitigate changes in lighting conditions. However, their classification model depends on a number of predefined thresholds which are not easily generalizable. From the previously described approaches, they do not consider natural lip movements during non-speaking intervals. Aubrey et al. [6] attempted to model both speech and non-speech movements by Hidden Markov Models (HMM), using optical flow vectors computed from consecutive mouth area frames as observations. Additionally, similar approach based on optical flow based mouth image energy feature and bi-level HMM was proposed by Tiawongsombat et al. [7]. Most techniques of the optical flow require appropriate illumination condition and that the displacement of pixels between frames are not abrupt [8]. Moreover, Hidden Markov Models are not suitable for modeling long data sequences because of the independence assumption where each hidden state can depend only on the immediate preceding one.

2 Proposed Method

The technique proposed in this paper only utilizes simple lip shape information extracted from the video sequences containing the speaker's face. First, the speaker's face is detected by histogram of oriented gradients (HOG) based detector [9]. Secondly, the detected face is used as a template to initialize the tracking algorithm [11] and then, in each subsequent frame, the facial landmark coordinates of the tracked face are located using the pretrained shape predictor [9]. Next, the lip shape vectors are geometrically normalized and the centroid distance function is applied to the latter, obtaining the lip shape features. Temporal variations of lip shape during speech and silence periods including non-trivial head motions are modeled using Long Short-Term Memory (LSTM) recurrent neural network. Figure 1 illustrates the overall approach. Details of each step are described as follows.

2.1 Face Detection

Face detection is performed on the initial frames until a face is detected by using the pre-trained sliding window detector provided by [9] which utilizes the

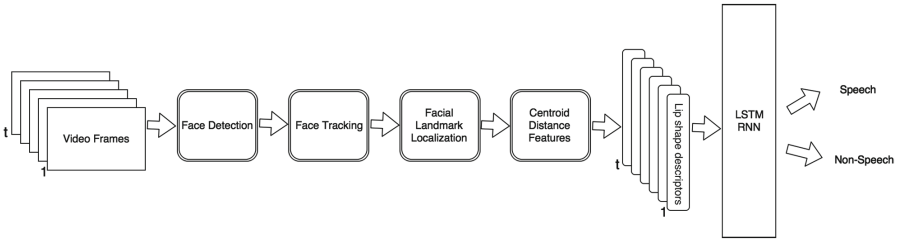

Fig. 1. Proposed approach

structural SVM based classifier trained on histogram of oriented gradient (HOG) features. It has a substantially lower false detections than Viola Jones face detector [10].

2.2 Face Tracking

Kernelized Correlation Filter tracker [11] is then applied. It is an online discriminative tracking algorithm. The object of interest can be continuously tracked while adapting the tracking model to incorporate new information about the former. The KCF tracker exploits the performance and computational efficiency afforded by utilizing cyclically translated samples and by performing corresponding kernel correlations in the Fourier domain. The bounding rectangle returned from the face detector is used to extract the face template to initialize the KCF tracker. Given a base template x of size $(m \times n)$ and the Gaussian-shaped regression targets y of size $(m \times n)$, the dual coefficients $\hat{\alpha}$ to solve the kernelized ridge regression in Fourier Domain is given by

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{1}$$

where

$\hat{\alpha}$ = DFT (Discrete Fourier Transform) of dual coefficients
$\hat{y}$ = DFT of Gaussian-shaped regression targets
$\hat{k}^{xx}$ = DFT of kernel autocorrelation vector
λ = Regularization term

In the subsequent frames, the kernel crosscorrelation $(\hat{k}^{xz})$ between cyclically shifted versions of base sample x and new candidate patch z can be computed and an $(m$ x $n)$ map of responses given by the following equation is obtained. $\odot$ represents element-wise multiplication.

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{2}$$

By calculating the location of the maximum response in the real part of the inverse Discrete Fourier transform of $\hat{f}(z)$ $(Re(\mathcal{F}^{-1}(\hat{f}(z)))$, the location of the

tracked face in the current frame can be found. The tracking model is then adapted to the newly found face patch x_n in frame n to reflect the changes.

$$\hat{\alpha}_n = (1-\eta) * \hat{\alpha}_{n-1} + \eta * (\frac{\hat{y}_{n-1}}{\hat{k}^{x_n x_n} + \lambda}) \tag{3}$$

$$x_n = (1-\eta) * x_{n-1} + \eta * x_n \tag{4}$$

Here η is the adaptation parameter, $\hat{\alpha}_n$ and x_n are the Fourier transform of dual coefficients and new interpolated base template respectively.

2.3 Landmark Localization and Geometric Normalization

To localize the facial landmarks in the tracked face, an implementation of the method described in [12] is utilized. It has been trained on an IBUG 300-W face landmark dataset to predict the location of 68 facial landmarks in realtime. Here it is defined that the $c_i \in R^2$ be the i^{th} x and y coordinate of lip contour in a face image F. The shape vector $C = (c_1^T, c_2^T, c_3^T, ..., c_{20}^T)$ represents the 20 coordinates of inner and outer lip contours in F. Next, the scale, orientation and translation components of the detected lip landmarks have to be normalized. In this step, it is necessary to find the 2×3 affine transformation matrix $\mathbf{A}$ which maps the vector C onto the coordinate frame with the size of 128×96 as shown in Fig. 2. $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix} \alpha & \beta & [(1-\alpha) * c_{x\,center} - \beta * c_{y\,center}] \\ -\beta & \alpha & [\beta * c_{x\,center} + (1-\alpha) * c_{y\,center}] \end{bmatrix} \tag{5}$$

where

$\alpha = \text{scale} * \cos\theta$
$\beta = \text{scale} * \sin\theta$
$c_{x\,center}$ = x coordinate of mouth center
$c_{y\,center}$ = y coordinate of mouth center
θ = angle of rotation around c_x and c_y

To get θ and scale, the following equations are computed.

$$\theta = \arctan(\frac{d_y}{d_x}) * \frac{180}{\theta} \tag{6}$$

$$scale = \frac{\gamma * 128}{||c_{lmc} - c_{rmc}||^2} \tag{7}$$

where d_y and d_x are the differences between y and x coordinates of left c_{lmc} and right c_{rmc} mouth corners respectively. γ is the scaling factor.

Finally, the geometrically normalized lip shape vector C_{norm} can be obtained by

$$C_{norm} = \mathbf{A} \begin{bmatrix} C \\ 1 \end{bmatrix}^{\mathbf{T}} \tag{8}$$

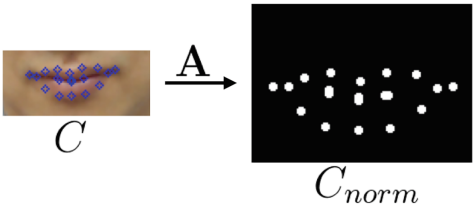

Fig. 2. Lip shape normalization

2.4 Centroid Distance Features

In this step, the centroid distance function [13] (CDF) is applied to the normalized outer and inner lip contour points in C_{norm}. CDF measures the distances between outer and inner lip boundary points and their respective centroids.

$$d^O{}_n = \sqrt{(x^O{}_n - x_c^O)^2 + (y^O{}_n - y_c^O)^2} \tag{9}$$

$$d^I{}_m = \sqrt{(x^I{}_m - x_c^I)^2 + (y^I{}_m - y_c^I)^2} \tag{10}$$

The resulting 20 dimensional feature vector $d_i = [d_n^O \; d_m^I]$ conveys the lip shape information during speech and silence frames.

2.5 LSTM Recurrent Neural Network

Recurrent neural networks (RNNs) belong to a family of supervised learning techniques and they have major advantages over feed forward neural networks and support vector machines in which recurrent neural networks can efficiently capture time dynamics and handle long-range time-dependencies. Moreover, unlike Hidden Markov models (HMM), they do not inherit the flaws of independence assumption where the current state can only depend on a limited number of previous ones. RNNs can model a probability distribution over an arbitrarily long sequence, $P(x_1, x_2, ..., x_T)$, without simplifications necessary to make HMMs mathematically and computationally tractable. At time step t, a recurrent neural network retains a state $\boldsymbol{s}_t$ which encodes the information regarding the entire sequence of previous inputs $(x_1, x_2, ..., x_t)$. Therefore, an RNN can be trained to learn a function f such that

$$\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}, \boldsymbol{x}_t) \tag{11}$$

For a simple recurrent neural network, the Eq. 11 becomes

$$\boldsymbol{s}_t = \phi(W^{ss}\boldsymbol{s}_{t-1} + W^{sx}\boldsymbol{x}_t) \tag{12}$$

where ϕ is either a logistic sigmoid or hyperbolic tangent nonlinearity. W_{ss} represents the state-to-state connections and W_{sx} represents input-to-hidden state

connections. However, simple RNNs cannot effectively learn long-range temporal and non-temporal dependencies since the backpropagated errors may either decay (vanishing gradients) or grow (exploding gradients) across several time steps.

Long Short-Term Memory Recurrent Neural Networks [14,15] are an extension of the original RNNs that address the major short comings of the latter. It achieves this by replacing the traditional hidden recurrent nodes with "LSTM cells" that ensure constant error propagation across time steps. Hence, they are readily suitable to capture the dynamics of lip movements over long temporal scales. An LSTM cell contains a special node (c) with a self-recurrent connection and an input node (I) while the flow of information to and from the cell is controlled by input (i) and output (o) gates. The forget gate (f) determines the persistence of the state of the special memory node. The equations governing the forward propagation mechanisms through an LSTM layer are as follows:

The input node $\boldsymbol{I}$ receives the previous state information of the network $\boldsymbol{s}_{t-1}$ and the current input $\boldsymbol{x}_t$. Then, a squashing function ϕ is applied on the affine transformation of its inputs.

$$\boldsymbol{I}_t = \phi(W^{Ix}\boldsymbol{x}_t + W^{Is}\boldsymbol{s}_{t-1} + \boldsymbol{b}^I) \tag{13}$$

The input and forget gates has the same inputs but the sigmoid nonlinearity σ is used as a gating function to produce a value between 0 and 1. Once learned, the input and forget gates determine how much of new information is allowed into the memory node and how much of previously memorized content should be discarded.

$$\boldsymbol{i}_t = \sigma(W^{ix}\boldsymbol{x}_t + W^{is}\boldsymbol{s}_{t-1} + \boldsymbol{b}^i) \tag{14}$$

$$\boldsymbol{f}_t = \sigma(W^{fx}\boldsymbol{x}_t + W^{fs}\boldsymbol{s}_{t-1} + \boldsymbol{b}^f) \tag{15}$$

Based on the activation values of the input and forget gates, the new internal state $\boldsymbol{c}_t$ of the cell is computed as a weighted sum of the new input information $\boldsymbol{I}_t$ and past internal state $\boldsymbol{c}_{t-1}$. $\odot$ represents element-wise multiplication.

$$\boldsymbol{c}_t = \boldsymbol{I}_t \odot \boldsymbol{i}_t + \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t \tag{16}$$

The output gate will modulate the extent to which the new state of the LSTM cell will be exposed to the rest of the network. $\boldsymbol{b}^I$, $\boldsymbol{b}^i$, $\boldsymbol{b}^o$, $\boldsymbol{b}^f$ are bias vectors.

$$\boldsymbol{o}_t = \phi(W^{ox}\boldsymbol{x}_t + W^{os}\boldsymbol{s}_{t-1} + \boldsymbol{b}^o) \tag{17}$$

Lastly, the hidden state of the LSTM network is updated according to the following equation.

$$\boldsymbol{s}_t = \boldsymbol{c}_t \odot \boldsymbol{o}_t \tag{18}$$

Single LSTM unit and the overall network architecture can be seen in Figs. 3 and 4 respectively. The HMM-based classification scheme is presented in the next section.

2.6 Hidden Markov Model

In this section, a brief description of the Hidden Markov Model (HMM) which is used to model the time varying lip shape vectors is provided. Hidden Markov Models are probabilistic generative models widely used in a variety of sequence generation and classification tasks. An HMM is parameterized by the initial state distribution π, the state transition matrix A, and the observation model O. Given that there are M hidden states and N dimensional feature vectors, the state variable $X_t \in \{i, j | i, j \in 1, ..., M\}$ and the observation variable $Y_t \in R^N$ at time t can be defined. Then, it follows that

$$\pi(i) = P(X_1 = i) \tag{19}$$

$$A(i, j) = P(X_t = j | X_{t-1} = i) \tag{20}$$

$$O(i) = P(Y_t | X_t = i) \tag{21}$$

From a series of T observations generated by a process c, ${Y^c}_{t=1:T}$, π^c, A^c, and O^c for the model M^c can be estimated using the well-known Baum-Welch [16] algorithm. The classification task using trained HMMs, $M^{c=speech}$ and $M^{c=silence}$, can therefore be formulated using log likelihood of the models.

$$M^{c*} = \underset{c}{\operatorname{argmax}}\, P(Y | M^c) P(M^c) \tag{22}$$

$P(Y^c | M^c)$ is the likelihood defined as the probability of the observed data given the model M^c and $P(M^c)$ is the prior for the model which can be omitted as it is assumed to be uniform.

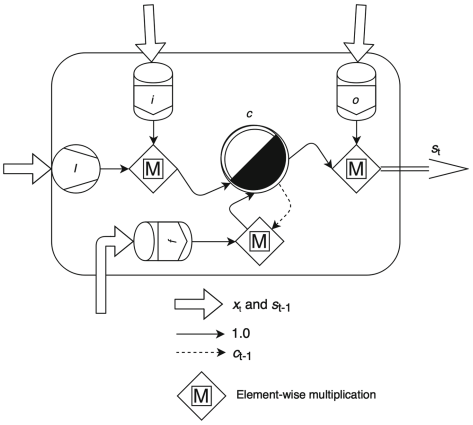

Fig. 3. An LSTM memory cell

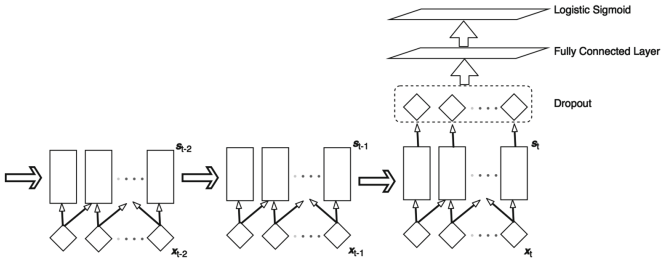

Fig. 4. LSTM network architecture

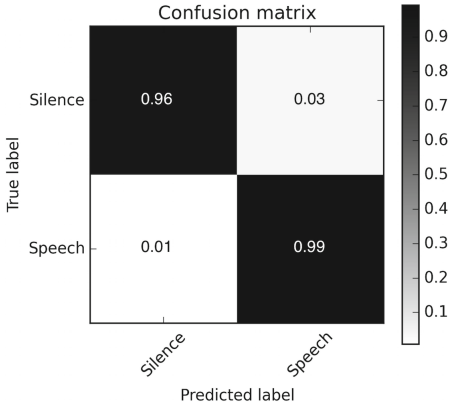

Fig. 5. Confusion matrix for LSTM classification

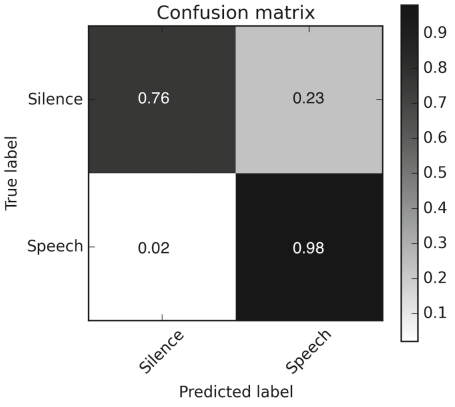

Fig. 6. Confusion matrix for HMM classification

3 Experiment Settings

3.1 Dataset

For evaluating the performance of the proposed approach, visual data collection system was set up as follows. A standard webcam with a resolution of 640 x 480 capturing at 30 frames per second was connected to a laptop running the feature extraction algorithm including face detection and tracking. A total of six subjects were asked to sit in front of the webcam and instructed to perform speech and non-speech lip movements. During the non-speech movement period, in addition to stationary lips, the subjects were instructed to perform typical behaviors such as smiling, laughing, shaking and nodding head, etc. for approximately 5 min. For data collection during speech, each subject was asked to read out loud a collection of Thai words and articles for about 5 min. Therefore, 20 dimensional centroid distance features were collected in realtime from approximately 230000 frames. This process was carried out under uncontrolled common lighting conditions.

3.2 Network Architecture

In this experiment, the temporal evolutions of lip shapes during speaking and non-speaking states are modeled with LSTM recurrent neural network containing one hidden layer of LSTM units. Hence, the input layer of the network receives a 20 dimensional feature vector at each time step. The number of LSTM units in the hidden layer is 100 which is experimentally found to be optimal for the current problem. A dropout layer [17] which randomly sets a certain portion of its inputs to zero with probability P ($P = 0.5$) is also added to reduce over-fitting of the network. The final layer is a fully connected layer with a sigmoid activation function which outputs a value between 0 an 1. Then, the binary cross-entropy between the target and network output values is evaluated. The network is trained using RMSProp [18] algorithm with the initial learning rate set to 0.001.

3.3 HMM-based Classifier

HMMs with Gaussian mixture observation model are trained using the same lip shape sequences as described in the previous section. 20 dimensional feature vectors, spanning a time window of 60 frames are used as training sequences to estimate the parameters of the speech and silence HMMs. The classification scheme described in Eq. 22 is employed to classify the video frame sequences into either speech or silence.

4 Experimental Results

In the first part of the experiment, the performance of the LSTM network and HMM-based classifiers is evaluated on the dataset which contains lip shape features extracted from all subjects. Two HMMs are trained on 75 % of the data

Fig. 7. An example of the program running in realtime

Table 1. Classification rate

	Classification Rate
CDF + LSTM	**98 %**
CDF + HMM	**87 %**

Table 2. Classification accuracies for different subjects using LSTM

Subject	1	2	3	4	5	6
1	-	0.83	0.75	0.95	0.89	0.95
2	0.95	-	0.85	0.96	0.80	0.98
3	0.73	0.78	-	0.98	0.96	0.98
4	0.92	0.79	0.89	-	0.96	0.99
5	0.63	0.71	0.90	0.97	-	0.91
6	0.75	0.80	0.86	0.97	0.90	-

while the remaining 25 % are used for testing. For LSTM-based classifier, the dataset is divided into training, validation and test sets containing 50 To investigate the effectiveness and generalizability of the proposed approach, the two models are trained solely on the centroid distance features extracted from each person and their classification performances are assessed on the datasets of every other subjects. The results of the second experiment are shown in Tables 2 and 3. Even though the LSTM network is trained only on a fraction of the whole dataset, it is still able to classify with high accuracy in most cases whereas the HMM-based classifier shows consistently lower performance. Also, the proposed approach using LSTM network is tested in realtime on a laptop with Intel Core i5 processor and 30 fps webcam with a resolution of 640x480, while performing forward pass on the network every 60 frames. Given that the user's face

Table 3. Classification accuracies for different subjects using HMM

Subject	1	2	3	4	5	6
1	-	0.56	0.47	0.53	0.46	0.55
2	0.52	-	0.54	0.59	0.83	0.69
3	0.52	0.80	-	0.88	0.88	0.86
4	0.53	0.84	0.80	-	0.86	0.90
5	0.56	0.75	0.86	0.96	-	0.97
6	0.63	0.77	0.78	0.94	0.90	-

is detected and tracked correctly, the algorithm can efficiently and accurately classify speech and complex non-speech lip shape sequences (Fig. 7).

5 Conclusion

A novel method for visual voice activity detection with integrated face tracking framework has been proposed. Since the features utilized by this method are simple and purely geometric, the efficiency and robustness of the algorithm is greatly increased. Another contribution of this paper is the use of Long Short-Term Memory neural network to model long range temporal evolutions of lip shapes during periods of speech and non-speech. To the best of the authors' knowledge, this is the first time that a temporal connectionist model is applied to visual voice activity detection. Moreover, the performances of and LSTM recurrent neural network and the classical HMM on visual voice activity detection task, using the proposed features are compared. Experimental results show that the trained network can achieve classification rate of above 98 % and also it is demonstrated that the generalization performance of the proposed approach using LSTM network is better than using HMM.

Acknowledgements. This research was supported by National Science and Technology Development Agency (NSTDA) and National Research Council of Thailand (NRCT).

References

1. McGurk, H., MacDonald, J.: Hearing lips and seeing voices. Nature **264**, 746–748 (1976)
2. Summerfield, Q.: Lipreading and audio-visual speech perception. Philos. Trans. R. Soc. Lond. B Biol. Sci. **335**(1273), 71–78 (1992)
3. Siatras, S., Nikolaidis, N., Krinidis, M., Pitas, I.: Visual lip activity detection and speaker detection using mouth region intensities. IEEE Trans. Circ. Syst. Video Technol. **19**(1), 133–137 (2009)

4. Ahmad, R., Raza, S.P., Malik, H.: Visual speech detection using an unsupervised learning framework. In: 12th International Conference on Machine Learning and Applications (ICMLA), vol. 2, pp. 525–528. 4–7 Dec 2013
5. Song, T., Lee, K., Ko, H.: Visual voice activity detection via chaos based lip motion measure robust under illumination changes. IEEE Trans. Consum. Electron. **60**(2), 251–257 (2014)
6. Aubrey, A.J., Hicks, Y.A., Chambers, J.A.: Visual voice activity detection with optical flow. IET Image Process. **4**(6), 463–472 (2010)
7. Tiawongsombat, P., Jeong, M.-H., Yun, J.-S., You, B.-J., Oh, S.-R.: Robust visual speakingness detection using bi-level HMM. Pattern Recogn. **45**(2), 783–793 (2012). ISSN 0031–3203
8. Roth, S., Lewis, J.P., Sun, D., Black, M.J.: Learning optical flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
9. King, D.E.: Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**, 1755–1758 (2009)
10. King, D.E.: Max-Margin Object Detection. CoRR abs/1502.00046 (2015)
11. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
12. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 1867–1874. IEEE Computer Society (2014)
13. Mingqiang, Y., Kidiyo, K., Joseph, R.: A Survey of Shape Feature Extraction Techniques. In: Yin, P.Y. (ed.) Pattern Recognition Techniques, Technology and Applications, InTech (2008). doi:10.5772/6237, ISBN: 978-953-7619-24-4
14. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (2000). PubMed PMID: 11032042
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
16. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
18. Dauphin, Y.N., de Vries, H., Chung, J., Bengio, Y.: RMSProp and equilibrated adaptive learning rates for non-convex optimization, CoRR abs/1502.04390 (2015)