Eye corners tracking for head movement estimation

Agostina J. Larrazabal
Research institute for signals, systems and computational intelligence , sinc(i)
FICH-UNL/CONICET
3000 Santa Fe, Argentina
alarrazabal@sinc.unl.edu.ar

Cecilia E. García Cena
Centre for Robotics and Automation
UPM-CSIC
28006 Madrid, Spain
cecilia.garcia@upm.es

César E. Martínez
Research institute for signals, systems and computational intelligence , sinc(i)
FICH-UNL/CONICET
3000 Santa Fe, Argentina
cmartinez@sinc.unl.edu.ar

***Abstract*—Recently, video-oculographic gaze tracking has begun to be used in the diagnosis of a wide variety of neurological diseases, such as Parkinson and Alzheimer. For this application, the so-called feature-based methods are used, more precisely, 2D regression-based methods. They use geometrically derived eye features from high-resolution eye images captured by zooming into the user's eyes. The main weakness of these methods is that the head of the user must remain motionless to avoid estimation errors. In some patients, some involuntary movements cannot be avoided and it is necessary to measure them. In this paper, we tackle the measurement of head position as a way to improve the gaze tracking on these precision demanding medical applications. As a first stage, we propose to obtain the eye corners coordinates as a reference point, since they are the most stable points in front of the eyeball and eyelids movements. The problem was handled as a regression problem using a coarse-to-fine cascaded convolutional neural network in order to accurately regress the coordinates of the eye corner. Particularly, with the aim of achieving high precision we cascade two levels of convolutional networks. Finally, we added temporal information to increase accuracy and decrease computation time. The accuracy of the estimation was calculated from the mean square error between the predictions and the ground truth. Subjective performance was also evaluated through video inspection. In both cases, satisfactory results were obtained.**

***Index Terms*—Landmark Tracking, Convolutional Neural Networks, Head Movements**

I. Introduction

Studies have shown that activity related to eye movements is observed in cortical and subcortical areas, which are directly and indirectly connected with several neural systems. They interact with each other to control the suitable performance of the ocular and ocular-cephalic movements. As a result, a broad spectrum of oculomotor alterations are usually observed in the presence of neurodegenerative motor disorders. An accurate and detailed analysis of eye movements becomes a unique opportunity to detect the presence of different injuries in the nervous center. These injuries involve a wide variety of neurological diseases including parkinsonian syndromes [1], amyotrophic lateral sclerosis [2], Huntingtons disease [3] Alzheimer disease [4] and minimal hepatic encephalopathy [5], among others. In recent years, the video-oculography (VOG) has become the most widely used eye-movements assessment method, as it is the only one considered non-invasive; also it allows for easy coordination of test design and stimuli provision that make it possible to automatically analyse the data [6], [7].

In order to be able to measure alterations in eye movements, the measurement of them must be performed with high precision and accuracy. That is why methods that use extracted eye features, such as pupil center or eye reflections, called feature-based methods, are the most popular approaches to gaze estimation in these kinds of applications. Feature-based methods, more precisely 2D regression-based methods, use geometrically derived eye features from high-resolution eye images captured by zooming in on the user's eyes. Then, they find a mapping function from the 2D feature space to gaze directions or the computer screen coordinates. These techniques are widely used and achieve really good results, but they have one major issue: the head of the person must remain motionless, otherwise there will be large errors between the actual and estimated directions. In order to avoid these errors, head restraint systems are often used. In spite of the fact that the head movements are restricted, in people with certain neurological diseases, it is possible to observe some involuntary movements. Being able to measure these head movements would be very important not only to correct the gaze estimation errors but also because these measurements seem to be another indicators of the presence or progress of certain diseases.

As of today, despite its importance for clinical diagnosis, there are not many studies about gaze estimation techniques that considers both the head and eye movements in the detailed conditions. We have also not found any research work aimed at detecting short involuntary head movements for future analysis. In this paper we tackle the measurement of the head position with the goal of being able to register small head movements only from videos of the patient's eye. With this aim, as a first stage we develop a method for estimating the eye corners coordinates. Since the eye corners are the most stable points in front of the eyeball and eyelids movements, the changes in their position may be used as a reference for the head movements estimation.

Facial landmark detection is a topic of much interest these days. Different algorithms aim to automatically identify the locations of the facial key points on facial images or videos for different applications. Most of them are designed for landmarks detection throughout the entire face [8], [9] and

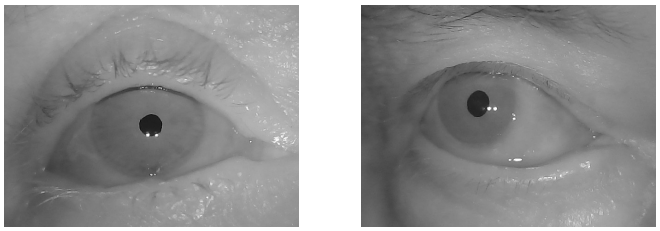
therefore require the whole face to be present in the image, regardless of variations in position, angle or illumination. Others are intended to detect the eye's corners exclusively by extracting the eye region from a given face image, mainly for low cost gaze tracking systems [10], [11]. In such circumstances, the extracted region is usually small in comparison to the image size and therefore the eye has a poor resolution. Contrary to this, in the studied application, the eyes are recorded directly at very high resolution. Therefore, their appearance and the their corners shape change significantly through the subjects or through small eye movements. This is why the goal of regressing the exact position of the corner becomes more challenging.

In recent time, there is a trend to shift from traditional methods to deep learning based methods, more specifically Convolutional Neural Network (CNN) for the task of facial landmark detection and tracking [12]–[14]. In this work, the problem is handled as a regression problem and a deep CNN is proposed to regress the landmark positions from the image appearance. Particularly, with the aim of achieving high precision, we cascade two levels of convolutional neural networks to make a coarse-to-fine prediction of the eye corners position in each frame.

Another point to consider, is the fact that many works address the landmark detection task in the same way for both static images and videos. Thus, when detection is performed in each individual frame, information from preceding frames is not used. Instead, the location of facial landmarks in preceding frames could be used to make it easier to find the facial landmarks in the current frame. One way to add temporal information consists of *tracking* the landmark region instead of detecting it each time. The problem with the direct application of a conventional tracker is that they are sensitive to appearance changes. At high image resolutions, when the person blinks the eye's corner region appearance changes completely and the tracker get lost. In these cases it is really complex to recover the region-of-interest (ROI) tracked and when is recovered, the position is not exactly the same, decreasing the accuracy. Alternatively, in this work we propose a mix approach: during the first frames a tracker is initialized with the two level networks prediction, afterwards the first network is no longer used. Instead, the corner patch is tracked along the video and the fine estimation is obtained with the second network from these patches. This approach showed to provide a fast estimation and a more accurate measurement. The latter is accomplished since when the eye's appearance does not change, the second level network receive the same patch and give a similar landmark estimation.

Finally, we can sum up the main contributions of this paper as follows.

- We created a specific database for this type of videos.
- We designed a coarse-to-fine cascaded convolutional neural network in order to accurately regress the coordinates of the eye corner.
- We added temporal information to increase accuracy and decrease computation time.



(a) (b)

Fig. 1: Eye images from different patients.

II. Data

Unlike the large number of facial landmark datasets available [15]–[17], there are no public datasets for this kind of application. Therefore, it was necessary to generate a customized database by selecting eye images with accurate eye related landmark labels for training and testing, and annotate them with ground truth. Eighteen videos were recorded from eighteen different patients using the Oscann device [18]. An infrared camera and an image resolution of 640x480 pixels were used. As these clinical studies are carry out on either of the two eyes and the choice is made by the doctor, the videos were recorded from both eyes. Therefore it was decided to mirror some of them in order to make them all look the same. Figure 1 shows an example of eye images from different patients.

Then, fifteen frames per video were carefully selected. Due to the large size of the eye in the image, its shape varies greatly depending on the person and in the presence of eye movements or blinks. As a consequence and in order to have a wider range of training data, those frames in which the eye position and the eyeball direction differ as much as possible were chosen. Each frame was identified with the patient number, so that we can split the data in training and validation sets without blending the patient. Since the eye's corner are kept relatively stable again the eye deformations, its accurate detection is critical to estimate the head movements only from the videos. To this end, the ground truth was carefully generated by hand for each eye's corner.

III. Methodology

A. Pre-processing

One of the main challenges faced in implementing this landmark regression cascade is the limited amount of available data. Deep learning methods have outperformed state of the art in various tasks but, due to the large number of parameters to be learned in the model, they require large amount of data. One way to add more data to the training process without the need to collect it is to perform *data augmentation*. It consists of a series of methods applied to each image in a random way during training, in order to train the networks in a robust manner avoiding overfitting at the same time. The individual transformations considered are:

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
A. Larrazabal, C. García Cena & C. E. Martínez; "Eye corners tracking for head movement estimation"
IEEE International Work Conference on Bioinspired Intelligence (IWOBI 2019), 2019.

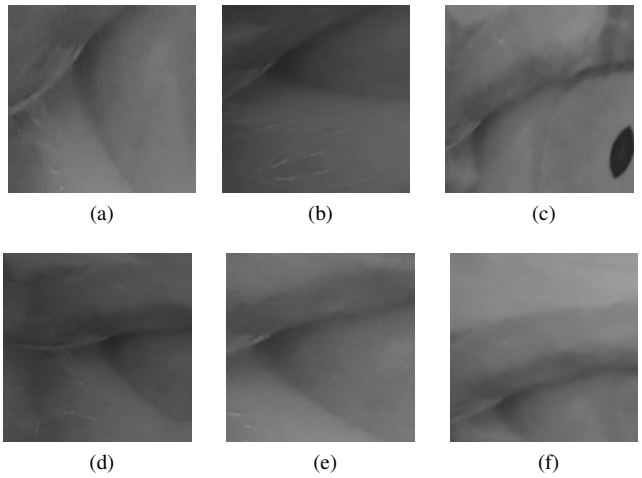


Fig. 2: Different appearance obtained from data augmentation techniques.

- Image rotations by an angle θ between $\pm 40°$ about the centre.
- Gaussian blurring with a 5×5 kernel.
- Change in brightness level.
- Changes in the image shape applying an affine transformation with random matrices.
- Elastic deformations [19].

For training the second level network, different ROIs were obtained from the original training images using the eye's corner ground truth as the reference point. To make the network more robust to the varying position of the eye corner, in each frame the ROI was cropped with a random offset from the centered reference point. In this way, all possible corner positions could be learned from the network. In addition, the ROIs were cropped with random width and height, and then interpolated to the pre-established size. This also varies the eye corner aspect and adds new data to the training process. After that, the data augmentation techniques previously explained were performed to these patches. Figure 2 shows six different patches generated by applying these transformations to a single frame. It should be noted that both, the appearance and the position of the corner, vary considerably. Finally, the intensity of the transformed images was normalized between 0 and 1 to be used as network inputs.

In the augmentation process, the landmarks coordinates were mapped to their new positions. Lastly, these ground truth coordinates lx and ly were normalized between -1 and 1 using the following equations:

$$lxn = \frac{lx - 0.5 \cdot Iw}{0.5 \cdot Iw} \quad lyn = \frac{ly - 0.5 \cdot Ih}{0.5 \cdot Ih} \tag{1}$$

where Iw and Ih are the image (or patch) width and height, respectively.

B. Convolutional Neural Network Structure

For landmark regression, we cascade two levels of convolutional neural networks to make a coarse-to-fine prediction.

TABLE I: Network architecture

	First level network		Second level network	
Block	**Layer (type)**	**Filter size**	**Layer (type)**	**Filter size**
Input	eye image (640x480)		image patch (140x140)	
Conv 1	convolutional convolutional Max pooling	3x3(32) 3x3(32)	convolutional convolutional Max pooling	3x3(32) 3x3(32)
Conv 2	convolutional convolutional Max pooling	3x3(64) 3x3(64)	convolutional convolutional Max pooling	3x3(64) 3x3(64)
Conv 3	convolutional convolutional Max pooling	3x3(128) 3x3(128)	convolutional convolutional Max pooling	3x3(128) 3x3(128)
Conv 4	convolutional convolutional Max pooling	3x3(128) 3x3(128)	convolutional convolutional Max pooling	3x3(128) 3x3(128)
Conv 5	convolutional convolutional Max pooling	3x3(256) 3x3(256)		
Conv 6	convolutional convolutional Max pooling	3x3(256) 3x3(256)		
Fully connected	flatten dense dense dense	 600 300 2	flatten dense dense dense	 600 300 2

The first level network makes an initial prediction of the eye corner landmarks. The second level network receives the image patch cropped from the original image centered in the first prediction, and implement a local refinement of the landmark coordinate. Since not all videos show the complete eye, the network was designed to estimate only the coordinates of one eye corner at a time. After preliminary experiments, the selected network architecture detailed in Table I was used. The ReLu activation function was applied after each convolutional layer and after the first two dense layer. In the last layer, a linear activation function was used.

C. Training

The first and second networks were trained separately. During training, the coarse regression network received the 640x480px augmented images as inputs and the refinement network received the 140x140px patches generated as it was detailed above. In some videos the eye does not appear complete within the image. This is why only sixteen of the patients could be used to train the inner corner model and seventeen could be used to train the outer corner model. The following steps were shared by both CNN.

1) Cross-Validation: An eigth-fold cross-validation [20] approach was used in this study. The data were split into eight folders depending on the patient number, i.e. in each iteration two patients were left for testing and the rest was used to train the neural network. For the outer corner models, as we had an odd number of patients, three of them were used for validation

in the first run. The partition was designed in this way to test the generalization ability of the network to correctly regress the position of the eye corner on previously unseen patients. The models were then compared using the average and the standard deviation of the eight folds.

2) Network parameters: The network was trained to minimise the mean square error (mse) between the landmarks labels and predictions applying the Adam Optimizer. To select the training parameters, a grid was designed in which different learning rates, batch sizes, patch sizes and dropout rates were tested. The best performance was achieve with learning rate: 0.0001; batch size: 16; patch size for the second network: 140x140px.

3) Testing: For testing, the original 15 frames of each patient were used following the validation scheme. To this end, the first network made the coarse prediction and, centered in its prediction, the patch was cropped and it was used as input for the second network which regressed the final coordinate. This prediction was transformed to pixel value by means of:

$$\begin{aligned} lxp &= lxn \cdot 0.5 \cdot Pw + 0.5 \cdot Pw + l \\ lyp &= lyn \cdot 0.5 \cdot Ph + 0.5 \cdot Ph + t \end{aligned} \tag{2}$$

being l and t the right and top patch positions and (lxp, lyp) the eye corner coordinate in pixels. For each model, the mean error and the standard deviation were calculated among all patients.

In addition, a subjective analysis with videos was carried out. Due to the laboriousness of annotating hundreds of data, it is really difficult to have an objective performance evaluation over the large videos. Furthermore, some aspects such as the tremor of the estimated point around the corner are complex to evaluate only by means of these metrics, but can be easily evaluated by inspection.

4) Tracking: The tracker used to follow the patch throughout the video was the Kernelized Correlation Filter (KCF) from OpenCV3 [21]. To initialize the tracker, the two-step network prediction was applied to the first frame and the ROI centered in its prediction was selected. After that, the first part of the network was no longer used and the input patch to the second network was update with this tracker along the videos.

IV. Results

In Figure 3 it can be seen a diagram of the entire process applied as an example to a particular frame. It is possible to observe how the patch is cropped on test time and how the second network improve the first network prediction. Figure 4 shows the mean error and standard deviation of the predictions given by the first and second networks on the complete dataset. The results show separately the predictions made on the inner and the outer corner of the eye. It can be seen that the second network greatly improves the results of the first one achieving a significantly better prediction.

Despite the fact that the mean accuracy obtained was quite good, and it improved even more by adding the tracker, the jiterring did not disappear completely. For this reason, we decided to add an off-line filter that smooths the prediction over time [22]. This feature was subjectively evaluated and it was found that the filter significantly improves this undesirable effect. These qualitative results for difficult validation videos can be seen in the videos of the following link: *https://agostinal.github.io/Corner-detector-project*. It is important to point out that for this application it is not necessary to work online since the final purpose is to use the predictions to post-process an eye tracking algorithm.

Finally, the tracking instance was also evaluated subjectively throughout the videos, finding that it is robust to blinking and rapid eye movements. Furthermore, the computation time was measured during each video both applying the cascade to each frame and following the tracking approach where only the fine-convolutional neural network is applied to each frame. The analysis was made in CPU as the algorithm it is thought to be used in a doctor office. It was found that applying patch tracking, the processing time was reduced 4 times. While on average the double cascade lasts 300 ms per frame, applying the tracker reduces the time per frame to 70 ms. This becomes very significant if it is considered that the videos are taken at 100 fps.

V. Conclusion

A model for eye corner coordinate estimation was constructed from scratch. The model was designed for a particular application for which it was necessary to collect and annotate the complete database. A coarse-to-fine cascaded convolutional neural network was implemented for static-frames landmark regression. Temporal information was added both from the patch tracking along the videos and from the implementation of a temporal filter to smooth the estimation. The experimental results confirmed the efficacy of the proposed method in different videos. With these promising results, we expect to provide a valuable information to be able to measure changes in the head position during clinical routine trials in patients.

Acknowledgments

The authors gratefully acknowledge NVIDIA Corporation with the donation of the Titan Xp GPU used for this research, and the support of UNL (CAID-PIC-50420150100098LI) and ANPCyT (PICT 2016-0651).

References

[1] Elena Pretegiani and Lance M Optican. Eye Movements in Parkinson's Disease and Inherited Parkinsonian Syndromes. *Frontiers in Neurology*, 8:592, 2017.

[2] Colette Donaghy, Matthew J Thurtell, Erik P Pioro, J Mark Gibson, and R John Leigh. Eye movements in amyotrophic lateral sclerosis and its mimics: a review with illustrative cases. *Journal of Neurology, Neurosurgery & Psychiatry*, 82(1):110–116, 2011.

[3] Stephen L Hicks, Matthieu PA Robert, Charlotte VP Golding, Sarah J Tabrizi, and Christopher Kennard. Oculomotor deficits indicate the progression of Huntington's disease. In *Progress in brain research*, volume 171, pages 555–558. Elsevier, 2008.

[4] Gerardo Fernández, Pablo Mandolesi, Nora P Rotstein, Oscar Colombo, Osvaldo Agamennoni, and Luis E Politi. Eye movement alterations during reading in patients with early alzheimer disease. *Investigative ophthalmology & visual science*, 54(13):8345–8352, 2013.

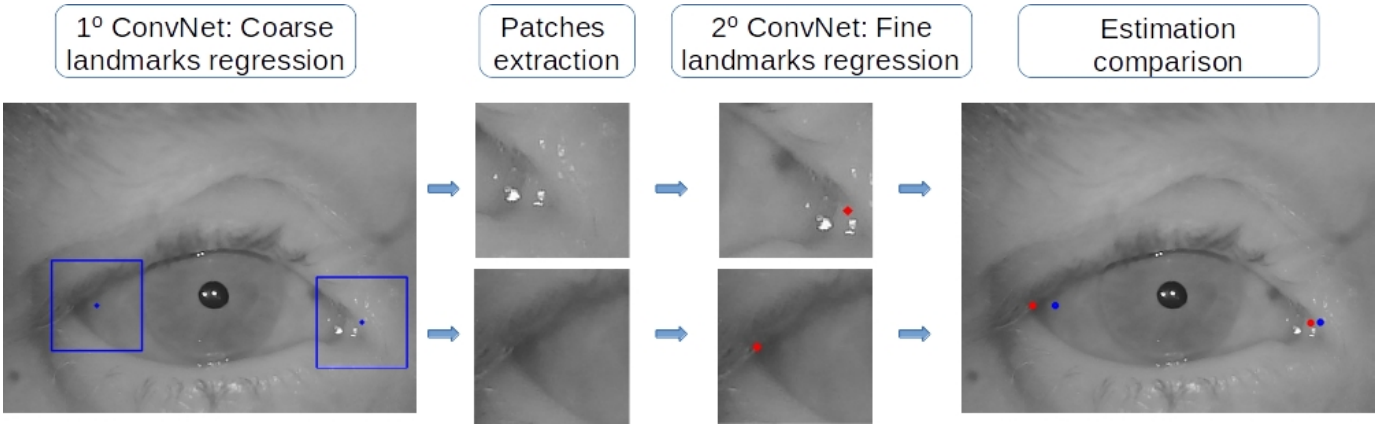


Fig. 3: Eye corner coordinate estimation from the first level network (blue) and second level network (red)

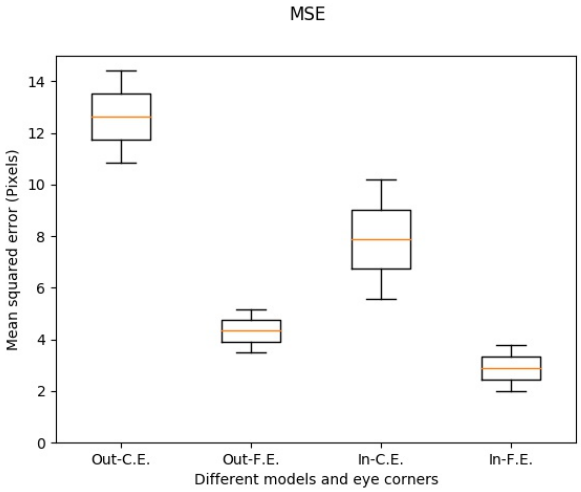


Fig. 4: MSE and standard deviation obtained from the first (C-E) and the second (F-E) networks.

[5] Sara Montagnese, Harriet M Gordon, Clive Jackson, Justine Smith, Patrizia Tognella, Nutan Jethwa, R Michael Sherratt, and Marsha Y Morgan. Disruption of smooth pursuit eye movements in cirrhosis: relationship to hepatic encephalopathy and its treatment. *Hepatology*, 42(4):772–781, 2005.

[6] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2010.

[7] Anuradha Kar and Peter Corcoran. A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access*, 5:16495–16519, 2017.

[8] Yue Wu and Qiang Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, pages 1–28, 2017.

[9] Chao Gou, Yue Wu, Kang Wang, Kunfeng Wang, Fei-Yue Wang, and Qiang Ji. A joint cascaded framework for simultaneous eye detection and eye state estimation. *Pattern Recognition*, 67:23–31, 2017.

[10] Jose Javier Bengoechea, Juan J Cerrolaza, Arantxa Villanueva, and Rafael Cabeza. Evaluation of accurate eye corner detection methods for gaze estimation. *Journal of Eye Movement Research*, 7(3), 2014.

[11] Yiu-ming Cheung and Qinmu Peng. Eye gaze tracking with a web camera in a desktop environment. *IEEE Transactions on Human-Machine Systems*, 45(4):419–430, 2015.

[12] Haoqiang Fan and Erjin Zhou. Approaching human level facial landmark localization by deep learning. *Image and Vision Computing*, 47:27–35, 2016.

[13] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[14] Yue Wu, Tal Hassner, KangGeon Kim, Gerard Medioni, and Prem Natarajan. Facial landmark detection with tweaked convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3067–3074, 2018.

[15] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2144–2151. IEEE, 2011.

[16] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.

[17] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 896–903, 2013.

[18] Erik Hernández, Santiago Hernández, David Molina, Rafael Acebrón, and Cecilia E García Cena. Oscann: Technical characterization of a novel gaze tracking analyzer. *Sensors*, 18(2):522, 2018.

[19] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *null*, page 958. IEEE, 2003.

[20] Richard O Duda, Peter E Hart, David G Stork, et al. Pattern classification. 2nd. *Edition. New York*, 55, 2001.

[21] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[22] Damien Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational statistics & data analysis*, 54(4):1167–1178, 2010.