# Conditional Random Fields Incorporate Convolutional Neural Networks for Human Eye Sclera Semantic Segmentation

Russel Mesbah rassoul@cs.otago.ac.nz
Brendan McCane mccane@cs.otago.ac.nz
Steven Mills steven@cs.otago.ac.nz
Department of Computer Science, University of Otago

## Abstract

*Sclera segmentation as an ocular biometric has been of an interest in a variety of security and medical applications. The current approaches mostly rely on handcrafted features which make the generalisation of the learnt hypothesis challenging encountering images taken from various angles, and in different visible light spectrums. Convolutional Neural Networks (CNNs) are capable of extracting the corresponding features automatically. Despite the fact that CNNs showed a remarkable performance in a variety of image semantic segmentations, the output can be noisy and less accurate particularly in object boundaries. To address this issue, we have used Conditional Random Fields (CRFs) to regulate the CNN outputs. The results of applying this technique to sclera segmentation dataset (SSERBC 2017) are comparable with the state of the art solutions.*

## 1. Introduction

Biometrics generally refers to the study of automatic authentication (verification or recognition) of individuals based on their measurable biological unique information such as facial, fingerprint, iris, retinal, signature, and voice characteristics. Sclera segmentation as a newly emerged ocular biometric has an important role to play in identification scheme since the random pattern of the sclera vessels is highly unlikely to be the same for two individuals [3].

Moreover, segmentation of the sclera region helps to improve iris recognition accuracy under different lighting conditions and eye gazes. Despite the fact that the vessels pattern is steady during our lifetime, there are other sclera characteristics which are more correlated with age and diseases [12].

The sclera is a white tissue surrounding the iris (Fig. 1). The main difficulty in sclera segmentation arises from the inclusion of eyelids and eyelashes in the sclera region and the noticeable effect of lighting conditions on the appearance of different ocular areas in the image. There are also other factors that may affect the segmentation accuracy e.g. different skin colours, medical conditions, the level of alcohol intoxication, age, and so on [3].

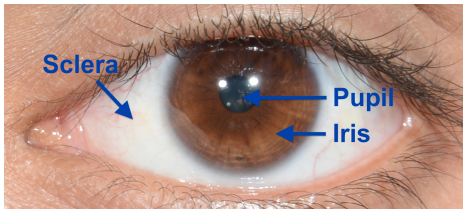



Figure 1. Human eye structure

## 2. Related Works

Research devoted to the automatic segmentation of the sclera from other human ocular regions mostly relies on modelling different parts of the eye based on their shape and colour space characteristics.

In [7] the authors proposed to employ a time-adaptive active contour method in which iris localisation helps to detect the two corners of the eye and enhance the sclera boundary detection accuracy. Similarly, in [15], [16] an active contour based on a time-adaptive self-organising map helps to reduce the sclera's inner boundary detection error. In [19] the input image was down-sampled and mapped from RGB to HSV colour space. The authors employed iris and eyelid detection techniques to enhance iris boundary detection accuracy prior to up-sampling the final segmented image.

The submitted methods in the Sclera Segmentation Benchmarking Competition (SSBC 2015) [4] tackled the problem using different multistage algorithms. The first participating team down-sampled the image, applied a nonlinear digital filtering to each colour map prior to using marker-controlled watershed segmentation method.

The second proposed algorithm was based on k-means. An elliptic model for the sclera and iris regions were employed to enhance the segmentation accuracy in which the

size of the ellipse was approximated using a polygon fitting algorithm. The iris pixels were eliminated from the set of chosen points based on their intensity characteristics.

In order to maintain the high contrast between the sclera and other ocular areas, the third participating team proposed to apply a shifting technique to the intensity histogram of the image. They employed Otsu's method to nominate the potential sclera pixels. Then a morphological operation was used to enhance the segmentation accuracy.

The last participating team in SSBC 2015 competition used three different types of features including statistical measures, Zernike moments, and HoG-style descriptors. These extracted features were employed by a set of simple classifiers where the corresponding probability outputs were fed into a neural network classifier for inferring the right label for the pixel of interest. The technique is reported to be robust against noise and showing stability encountering different gaze directions.

Submitted works to the segmentation scheme of the Sclera Segmentation and Recognition Benchmarking Competition (SSRBC 2016) are published in [5]. The first proposed algorithm was based on a new Robust Spatial Kernel Fuzzy C-means (RSKFCM) method in which the Gaussian kernel function was used to estimate the centroid of the clusters.

Similarly, the second team used an unsupervised method based on the colour space characteristics in which the bigger is the cluster region the higher is the assigned score to the corresponding cluster. The Otsu's binarisation technique was applied to the clusters with the highest and the next-to-highest scores. According to [5], the technique succeeded in filling most of the holes in the sclera cluster (sclera should not include any holes).

Although the approaches described in this section show a good performance in sclera segmentation, they highly rely on hand-crafted features. It necessitates a great deal of effort alongside a profound knowledge about the particular field of interest. To address this issue, we propose to employ Convolutional Neural Network (CNN) based techniques as CNN's automatic feature extraction precludes the need for having engineered features.

## 3. CRF+CNN for Semantic Segmentation

Convolutional neural networks are known by their weight-sharing and downsampling characteristics which bring about CNN's location and scale invariance [13]. These universal function approximators are based on a hierarchy of convolutional layers, each followed by a downsampling function.

At each layer, a set of sliding learnable convolutional filters projects the extracted patterns onto the feature maps. The preceding downsampling function (usually max pooling operation) reduces the resolution of the feature maps. It helps to improve the network's robustness to noise and small variations.

The output of each layer is fed into the next layer. Moving up through this hierarchy, high-level low-resolution abstractions are extracted. At the apex of the pyramid of abstractions, using a classifier, single or multiple concepts can be inferred.

The trained architecture can be applied to a subset of the input image (sampling window) to guess the right label for the pixel of interest (usually the central pixel in the sampling window). In order to segment the entire image, the sampling window should be shifted across all the pixels.

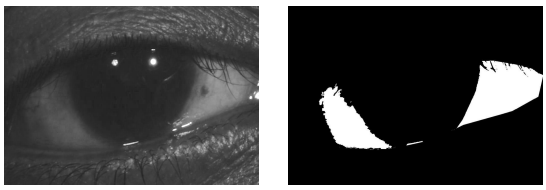
Despite the fact that CNNs show remarkable performance in a variety of image segmentation problems, the presence of noise is noticeable in the output (Fig. 6). Moreover, the correct pixel-wise classification rate in object boundaries is usually less than other regions [9]. Employing Conditional Random Fields (CRFs) as a post-processing stage can be a solution to this problem as CNNs are potentially unable to model the interactions between neighbouring output pixels and CRFs can model these dependencies directly [8].

Since factorising a function over a large set of variables is practically unattainable, there is always a demand to find a representation based on the product of local functions acting over smaller subsets. CRFs can factorise the probability distribution over a large collection of pixels based on the product of local functions defined over a set of features. The performance, consequently, highly relies on the selection of these features [17] so that the combination of CNN and CRF can decrease the complexity of the feature engineering process.

Some approaches propose to integrate CNN and CRF in a single training phase [14, 18] while other methods use CRF as a post-processing technique [1, 2, 6, 10, 11]. As per our knowledge, the combination of convolutional neural networks and conditional random fields has not been used for sclera semantic segmentation.

## 4. Dataset

We have used the segmentation dataset of the Sclera and Eye Recognition Benchmarking Competition (SSERBC 2017) in our experiments. The dataset includes ocular images of 30 participants taken with different eye gazes (looking straight, up, down, left, and right). To meet our technical requirements, we have randomly chosen one image for each participants. All the 30 images are converted to greyscale and resized to $700 \times 1000$ pixels using quadratic interpolation. Fig. 2 shows a sample human eye and the corresponding labelled image from the dataset.



(a) (b)

Figure 2. A grayscale resized SSERBC sample: (a) sample human eye ; (b) the corresponding labelled image in which sclera is shown in white against black background.

## 5. Method

We apply a CRF-based post-processing approach to our convolutional neural networks architecture in which the output of the trained CNN can be used as one of the features in the conditional random fields design. Other CRF features are either based on the input information or rely on the interactions between the neighbouring output pixels or are affected by both factors.

### 5.1. Convolutional Neural Network Design

Our CNN consists of three main units each having a convolutional layer followed by a MaxPooling layer (Fig. 3). Eight $3 \times 3$ convolutional filters are used at the first unit. The MaxPooling layer reduces the dimensions of the produced feature map by a factor of two. As the input frame is a $32 \times 32$ greyscale image, the output of the first unit is an $8 \times 16 \times 16$ tensor (tensor refers to a matrix-like type of variable being used in the context of deep learning).

At the second unit, 64 $3 \times 3$ convolutional filters are employed to map the extracted patterns onto a $64 \times 16 \times 16$ feature map (feature map is a tensor produced by applying convolutional filters to another tensor). Similarly to the first layer, a MaxPooling layer down samples the feature map and produces a $64 \times 8 \times 8$ tensor. The tensor is fed into the last unit including 256 $3 \times 3$ convolutional filters and a MaxPooling layer generates an $256 \times 4 \times 4$ output tensor.

A fully connected architecture (FC) is employed to infer the right label from the output tensor for the pixel of interest. The FC consists of 4096 and 256 nodes in the first and second layers, respectively. The last layer has only one node associated with the hypothesised class. The hyperbolic tangent is used as non-linear activation function at all layers.

### 5.2. Convolutional Neural Networks Training

We performed 30 trials, in each one image is excluded and the rest of the dataset is used for training. There are 78000 samples in each training set randomly cropped from the ocular images using a $32 \times 32$ window size. The corresponding label is the class associated with the pixel at coordinate $(16, 16)$. The label is either 1 or -1 representing sclera or background, respectively. At runtime, a $32 \times 32$ window is shifted across the image with the step size of one pixel. Therefore, to segment the entire image, 646624 classification tasks are to be performed.

We have used automatic Stochastic Gradient Descent (SGD) algorithm to optimise the training performance. We employ a fixed number of training epochs to avoid overfitting. This was determined by observation and is the same for all trials. We have chosen 5 epochs for all the training trials.

### 5.3. Conditional Random Fields: Feature Design

The energy function of our CRF is based on four different types of features including the label-observation, input-observation, edge-observation, and CNN-dependent (model combination) features.

#### 5.3.1 Label-Observation Features

The probability of occurrence for each class at the pixel of interest can be obtained simply by counting the number of pixels associated with the particular class against the total number of pixels in the ground truth images. We represent the likelihood of occurrence for class $\mathcal{L}_k$ at pixel $x_i$ as $\mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k)$ where $k$ is the class index. The corresponding CRF feature is

$$\vartheta_c(x_i) = \sum_{k=1}^{n} 1\{\mathcal{L}_{x_i} = \mathcal{L}_k\}\, \mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k) \qquad (1)$$

where $\mathcal{L}_{x_i}$ is the pixel's assigned label, $n$ is the total number of classes and

$$1\{True\} = 1\ ,\ 1\{False\} = 0 \qquad (2)$$

Table 1 shows $\mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k)$ for the two classes in the sclera segmentation dataset.

| $\mathcal{L}_{x_i} = \mathcal{L}_k : k \in \{1, 2\}$ | $\mathcal{P}$ |
|---|---|
| Sclera | 0.238 |
| Background | 0.762 |

Table 1. The distribution of classes in sclera segmentation dataset

Similarly, as the interactions between first order neighbouring output pixels should be taken into the consideration, we have defined $\mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k \mid \mathcal{L}_{x_j} = \mathcal{L}_m)$ as the probability of having pixel $x_i$ with label $\mathcal{L}_k$ at the centre conditioned to having pixel $x_j$ with label $\mathcal{L}_m$ ($k$ and $m$ are the arbitrary class indices) in the neighbourhood (Table 2). The associated CRF feature is

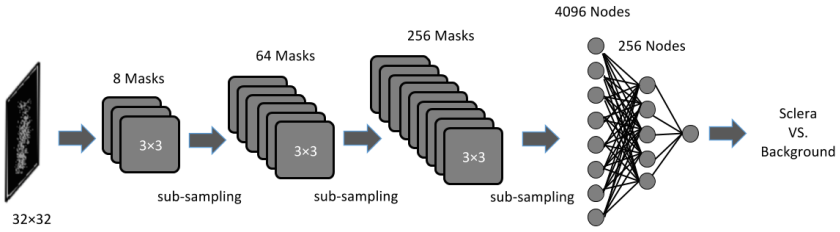



Figure 3. CNN architecture

| $\mathcal{L}_{x_i} = \mathcal{L}_k \mid \mathcal{L}_{x_j} = \mathcal{L}_m : k, m \in \{1, 2\}$ | $\mathcal{P}$ |
|---|---|
| $\mathcal{L}_{x_i} = Sclera \mid \mathcal{L}_{x_j} = Sclera$ | 0.999 |
| $\mathcal{L}_{x_i} = Background \mid \mathcal{L}_{x_j} = Sclera$ | 0.001 |
| $\mathcal{L}_{x_i} = Sclera \mid \mathcal{L}_{x_j} = Background$ | 0.006 |
| $\mathcal{L}_{x_i} = Background \mid \mathcal{L}_{x_j} = Background$ | 0.994 |

Table 2. The probability of having pixel $x_i$ with label $\mathcal{L}_k$ at the centre conditioned to having pixel $x_j$ with label $\mathcal{L}_m$ in the neighbourhood.

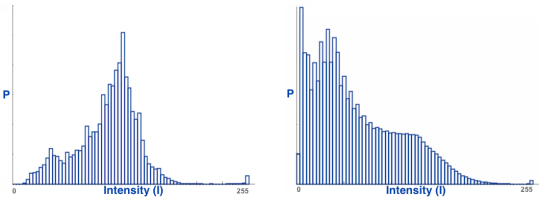
$$\vartheta_1(x_i, x_j) = \sum_{m=1}^{n} \sum_{k=1}^{n} 1\{\mathcal{L}_{x_i} = \mathcal{L}_k, \mathcal{L}_{x_j} = \mathcal{L}_m\} \; \mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k \mid \mathcal{L}_{x_j} = \mathcal{L}_m) \tag{3}$$

#### 5.3.2 Input-Dependent Features

As the intensity value has an important role to play in sclera segmentation, we model the likelihood of assigning each label to $x_i$ conditioned to the pixel's intensity value as below:

$$\psi_{in}(x_i) = \sum_{k=1}^{n} 1\{\mathcal{L}_{x_i} = \mathcal{L}_k\} \, \mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k \mid \mathcal{I}_{x_i}) \tag{4}$$

where $\mathcal{I}_{x_i}$ is the intensity value associated with the pixel $x_i$. Fig. 4 shows the $\mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_k \mid \mathcal{I}_{x_i})$ values for the two different classes.




(a) (b)

Figure 4. The likelihood of assigning $\mathcal{L}_k$ to $x_i$ conditioned to the pixel's intensity value: (a) $\mathcal{P}(\mathcal{L}_{x_i} = Sclera \mid \mathcal{I}_{x_i})$ ; (b) $\mathcal{P}(\mathcal{L}_{x_i} = Background \mid \mathcal{I}_{x_i})$

#### 5.3.3 Edge-Observation Features

This feature encourages the algorithm to select different labels at the edge points. The higher is the difference between two neighbouring pixel's intensity values, the more likely are the corresponding assigned labels to be different (Eq. 4).

$$\psi_{edge}(x_i, x_j) = 1\{\mathcal{L}_{x_i} \neq \mathcal{L}_{x_j}\} \parallel \frac{\mathcal{I}_{x_i} - \mathcal{I}_{x_j}}{255} \parallel^2 \tag{5}$$

#### 5.3.4 Features as Model Combination

CNNs show a remarkable performance in segmentation tasks by modelling the interactions between all the input pixels in the sampling frame and the corresponding assigned label. CNN's confusion matrix includes the likelihood of correct classification for a pixel of interest. Eq. 6 defines $\phi_{cnn}$ as the likelihood of choosing each label for $x_i$ conditioned to the CNN's output at the pixel of interest. The corresponding values can be calculated based on the CNN's confusion matrix (Table 3).

$$\phi_{cnn}(x_i) = \sum_{m=1}^{n} \sum_{k=1}^{n} 1\{\mathcal{L}_{x_i} = \mathcal{L}_m, \mathcal{L}_{x_i^{cnn}} = \mathcal{L}_k\} \; \mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_m \mid \mathcal{L}_{x_i^{cnn}} = \mathcal{L}_k) \tag{6}$$

where $\mathcal{L}_{x_i^{cnn}}$ is the CNN's output at $x_i$.

| $\mathcal{P}(\mathcal{L}_{x_i} = \mathcal{L}_m \mid \mathcal{L}_{x_i^{cnn}} = \mathcal{L}_k) : k, m \in \{1, 2\}$ | $\mathcal{P}$ |
|---|---|
| $\mathcal{L}_{x_i} = Sclera \mid \mathcal{L}_{x_i^{cnn}} = Sclera$ | 0.513 |
| $\mathcal{L}_{x_i} = Background \mid \mathcal{L}_{x_i^{cnn}} = Sclera$ | 0.487 |
| $\mathcal{L}_{x_i} = Sclera \mid \mathcal{L}_{x_i^{cnn}} = Background$ | 0.122 |
| $\mathcal{L}_{x_i} = Background \mid \mathcal{L}_{x_i^{cnn}} = Background$ | 0.878 |

Table 3. The likelihood of choosing $\mathcal{L}_m$ for $x_i$ conditioned to the CNN's output at the pixel of interest.

### 5.4. Simulated Annealing

To optimise CRF's performance, we adopt Simulated Annealing (SA) algorithm as it can find inexpensive solutions in large state space. Rather than starting the search with a randomised state, we have used the CNN outputs as the initial state. We generate random labels for the pixels and the algorithm always accepts the label which decreases the energy (cost function). SA accepts the bad moves respectively to a temperature-based probability distribution called Boltzmann distribution. The energy function can be defined as

$$E_{\xi_i} = \vartheta_c(x_i) + \sum_j \vartheta_1(x_i, x_j) + \psi_{in}(x_i) + \sum_j \psi_{edge}(x_i, x_j) + \phi_{cnn}(x_i) \quad (7)$$

where $\xi_i$ is the state of the pixel at $x_i$. The corresponding Boltzmann distribution is

$$\mathcal{P}(\xi_i) = \frac{e^{(-E_{\xi_i}/T)}}{Z(T)} \quad (8)$$

where $T$ represents the temperature-based probability distribution and $Z(T)$ is a normalisation constant calculated as

$$Z(T) = \sum_{\xi_i'} e^{(-E_{\xi_i'}/T)} \quad (9)$$

———— Our Simulated Annealing Algorithm ————

$\mathcal{L}_{x_i} \leftarrow \mathcal{L}_{x_i^{cnn}}$
$E_{old} \leftarrow E_{\xi_i}$
$T \leftarrow T_{init}$
$Iteration \leftarrow 0$
**while** $Iteration < 100000$ **do**
  $T \leftarrow T(Iteration)$
  $\mathcal{L}_x^{new} \leftarrow \mathcal{L}_k \in \{set\ of\ all\ labels\}$
  $E_{new} \leftarrow E_\xi^{new}$
  **if** $\mathcal{P}(E_{new} - E_{old}) > Random()$ **then**
    $\mathcal{L}_{x_i} \leftarrow \mathcal{L}_x^{new}$
    $E_{old} \leftarrow E_{new}$
  **end if**
  $Iteration \leftarrow Iteration + 1$
**end while**

———————————————————————————

In practice, cooling schedule plays an important role in the convergence of SA algorithm, so that the commonly used multistep logarithmic temperature decrease is employed (Fig. 5). We initialise the temperature where 50% of bad moves are accepted. Our SA includes 100000 iterations and in the quenching step ($T\ min$), the algorithm does not accept any bad moves.

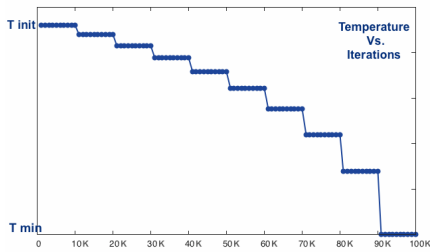



Figure 5. Multistep logarithmic temperature decrease: vertical axis represents the temperature values while the horizontal axis corresponds to the number of iterations.

### 5.5. Experimental Setup

For all the experiments we have used Torch7 library (www.torch.ch) and an iMac machine with a 3.2 GHz Core i5 quad-core processor. The CNN training time for each trial was roughly 35 minutes and the segmentation of the entire image using trained CNN module took more than 57 minutes. Our CRF algorithm required nearly 82 minutes to perform 100000 iterations over the CNN output.

## 6. Results & Discussion

We compared our results with the the first and second ranked solutions presented in sclera segmentation and recognition benchmarking competition (SSRBC 2016). As it can be seen in Table 4, the combination of CRF and CNN outperforms the state of the art solutions in pixel-wise correct classification rate.

| Method | Accuracy |
|---|---|
| SSRBC: team 1 | 82.2% |
| SSRBC: team 2 | 80.5% |
| CNN | 81.1% |
| CRF+CNN | **83.2%** |

Table 4. Comparison of the pixel-wise correct classification rates

Table 5 demonstrates the precision and recall for CNN and the combination technique. The associated standard deviations across the trials are 2.4% and 3.2% for the two methods, respectively. We have applied the standard t-test to the proposed solutions to evaluate the significance of the improvement in accuracy and the corresponding $p$ value is 0.0059 ($< 0.05$).

| Method | Precision | Recall |
|---|---|---|
| CNN | 51.3% | 48.7% |
| CRF+CNN | 54.7% | 74.3% |

Table 5. Precision and recall for CNN and CRF+CNN techniques

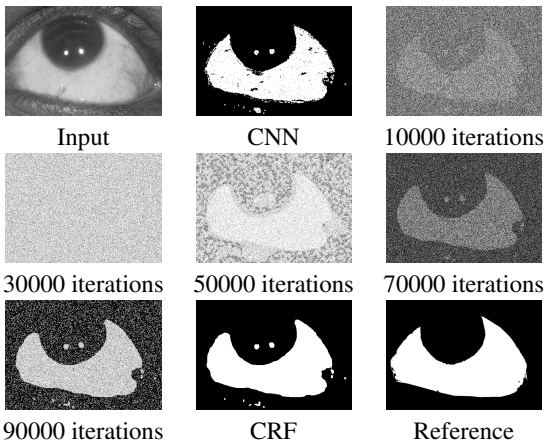



Figure 6. An example of good segmentation


Figure 7. An example of bad segmentation

## 7. Conclusion

Although CNN's correct classification rate is comparable to the state of the art solutions, the segmentation is noticeably noisy at some pixels. CRF can optimise CNN's performance by homogenising the regions and enhancing the accuracy at many pixels including object boundaries. In addition, the incorporation technique produces qualitatively more plausible results than convolutional neural networks.

While CNN archives 81.1% correct classification rate, incorporation of CNN and CRF outperformed the state of the art solutions by labelling 83.2% of the pixels correctly. The statistical t-test shows enhancement in pixel-wise accuracy using the incorporation of CNN and CRF techniques versus CNN.

Despite the improvement in accuracy, the noticeable computation time can be considered as a drawback of using this approach since it makes real-time image segmentation unattainable.

## References

[1] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *CVPR*, 2015.

[2] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015.

[3] A. Das, U. Pal, M. Blumenstein, and M. A. F. Ballester. Sclera recognition - a survey. *Proceedings - 2nd IAPR Asian Conference on Pattern Recognition*, 2013.

[4] A. Das, U. Pal, M. A. Ferrer, and M. Blumenstein. Ssbc 2015: Sclera segmentation benchmarking competition. *IEEE 7th International Conference on Biometrics Theory, Applications and Systems*, 2015.

[5] A. Das, U. Pal, M. A. Ferrer, and M. Blumenstein. Ssbc 2015: Sclera segmentation benchmarking competition. *International Conference on Biometrics*, 2016.

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8), 2013.

[7] M. H. Khosravi and R. Safabakhsh. Human eye sclera detection and tracking using a modified time-adaptive self-organizing map. *Pattern Recognition*, 41:2571–2593, 2008.

[8] B. Liu and Y. Wen. Cnn and crf for semantic image segmentation. University of Toronto `http://www.cs.toronto.edu`.

[9] R. Mesbah, B. McCane, and S. Mills. Deep convolutional encoder-decoder for myelin and axon segmentation. *IVCNZ*, 2016.

[10] G. Papandreou, L. Chen, I. Kokkinos, K. Murphy, and A. L. Yuille. Weakly and semi-supervised learning of a dcnn for semantic image segmentation. *ICCV*, 2015.

[11] X. Qi, J. Shi, S. Liu, R. Liao, and J. Jia. Semantic segmentation with object clique potentials. *ICCV*, 2015.

[12] M. Sarala, Vijayanand, and M. Malathy. Effective application for detection of diseases from sclera image. *Int. J. Advanced Networking and Applications*, 8(4), 2016.

[13] J. Schmidhuber. Deep learning in neural networks: An overview. *arXiv preprint arXiv:1404.7828*, 61:1–66, 2014.

[14] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv:1503.02351*, 2015.

[15] H. Shah-Hosseini and R. Safabakhsh. Human eye sclera detection and tracking using a modified time-adaptive self-organizing map. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41:271–282, 2003.

[16] H. Shah-Hosseini and R. Safabakhsh. A tasom-based algorithm for active contour modeling. *Pattern Recognition Letters*, 24:1361–1373, 2003.

[17] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. University of Massachusetts `www.cs.umass.edu`.

[18] S. Zheng, S. Jayasumana, B. Romera-Paredes, and V. Vineet. Conditional random fields as recurrent neural networks. *ICCV*, 2015.

[19] Z. Zhou, E. Y. Du, and N. L. Thomas. A comprehensive approach for sclera image quality measure. *11th International Conference Control, Automation, Robotics and Vision Singapore*, 2010.