A Convolutional Neural Network Approach for Classifying Leukocoria

Ryan Henning*, Pablo Rivas-Perea*, Bryan Shaw† and Greg Hamerly*
*Department of Computer Science
†Department of Chemistry
Baylor University
Waco, TX, USA
Contact Info: http://leuko.net/

***Abstract*—We use Convolutional Neural Networks to detect leukocoria, or white-eye reflections, in recreational photography. Leukocoria is the most prominent symptom of retinoblastoma, a solid-tumor cancer of the eye that occurs most often in young children. We trained several networks for the task, using training images downloaded from Flickr. We achieved low error rates (<3%) for classification of eye images into three classes: normal, leukocoric, and pseudo-leukocoric. We also provide a method for tuning the outputs of a trained network to match desired true-positive/false-positive rates.**

***Keywords*-machine learning; retinoblastoma; leukocoria**

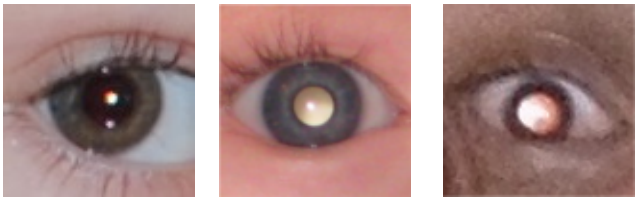
I. Introduction

Retinoblastoma (Rb) is the most common ocular malignancy in children, occurring in 1 in 18,000 to 30,000 live births worldwide [1]. A child with Rb can develop one or more tumors in one or both eyes. Treatment of Rb can include external beam radiation and photocoagulation. Early detection prevents surgical removal of the eye [1].

The most common symptom of Rb is a white reflection emitted from the retina of the eye through the pupil. It is the reflection of light off the tumor which causes the white color. This symptom is called *leukocoria* (white pupillary reflection), and is present in 60% of the reported cases in the United States [1]. Researchers at Baylor University characterised leukocoria in recreational photography, and they concluded that the intensity of the symptom is an indicator of the cancer's stage [2]. An example of leukocoria compared to a "normal eye" can be seen in Figures 1a and 1b. Leukocoria does not always indicate Rb, but it also indicates several other ocular diseases including Coats disease and cataracts.

While physicians screen for leukocoria, parents often detect it first (using photography), but are not aware of its connection to Rb. Automated detection methods do not yet exist. Automated leukocoria detection would be of high value, as it would lead to earlier diagnosis of this cancer in children, thereby increasing the survival rate and providing better quality of life for those who survive.

The task is difficult partly due to images taken with bright LED flashes, such as images taken by the Apple iPhone. Such flashes cause a very bright, slightly diffuse surface-level white reflection in the eye region. These surface-level white reflections resemble leukocoria that arises from ocular tumors, but they are not created by a tumor, and they do not come from inside the pupil. We call this visual effect *pseudo-leukocoria*. See Figure 1c for an example.



(a) normal (b) leukocoric (c) pseudo-leukocoric

Figure 1: Examples of each type of cropped eye image.

With the increasing popularity of mobile-device photography, pseudo-leukocoria is fairly common. For example, the Apple iPhone is currently the most popular camera on Flickr. Additionally, by our estimation, the bright LED flash fires in approximately 12% of the photos taken with the iPhone on Flickr. Therefore, a feasible solution for detecting leukocoria must not only distinguish leukocoric eyes from normal eyes, but it must also be able to distinguish true pupillary leukocoria from surface-level pseudo-leukocoria.

We demonstrate that it is possible to detect leukocoria in raw digital images of cropped eyes by training convolutional neural networks (CNNs). Our solution can accurately distinguish between three classes of eye images: normal, true leukocoria (pupillary), and pseudo-leukocoric (surface-level). Our classification program is available at http://leuko.net as well.

II. Data

Our data come from two sources: recreational photographs contributed by families of children with Rb (including photos of people who don't have Rb) and recreational photographs we gathered from Flickr. We analyzed these images and extracted by hand three types of cropped eye images: normal (437 eye images), leukocoric (222), and pseudo-leukocoric (173). The normal eyes came from both data sources, the leukocoric eyes came from the images of children with Rb, and the pseudo-leukocoric eyes came from

Table I: Networks trained and evaluated in this paper. Networks 1-15 were trained using momentum with parameter values $a = 1.0$ and $m = 0.9$. Networks 16-20 were trained with RMSPROP with parameter value $a = 0.1$. Please see the text for a description of the columns.

id	Convolutional Layers	Fully-connected Layers	Layer Types	# Free Parameters	Error Rate ± Std. Error
1	—	6-3	h-s	28,827	$6.37 \pm 1.29\%$
2	—	12-3	h-s	57,651	$5.89 \pm 1.19\%$
3	—	25-3	h-s	120,103	$6.49 \pm 1.18\%$
4	—	50-3	h-s	240,203	$6.73 \pm 1.13\%$
5	—	100-3	h-s	480,403	$6.97 \pm 1.12\%$
6	7	5-3	h-h-s	11,898	$5.05 \pm 0.88\%$
7	14	5-3	h-h-s	23,767	$5.05 \pm 0.98\%$
8	21	5-3	h-h-s	35,639	$5.17 \pm 0.97\%$
9	21	10-3	h-h-s	69,679	$5.41 \pm 1.02\%$
10	21	15-3	h-h-s	103,719	$5.29 \pm 0.91\%$
11	7-7	5-3	h-h-h-s	3,502	$3.97 \pm 0.83\%$
12	14-14	5-3	h-h-h-s	9,431	$4.09 \pm 0.76\%$
13	21-21	5-3	h-h-h-s	17,810	$3.73 \pm 0.69\%$
14	21-21	10-3	h-h-h-s	22,975	$4.21 \pm 0.66\%$
15	21-21	15-3	h-h-h-s	28,140	$4.33 \pm 0.88\%$
16	7-7-7	5-3	h-h-h-h-s	3,334	$2.40 \pm 0.74\%$
17	14-14-14	5-3	h-h-h-h-s	11,545	$2.88 \pm 0.63\%$
18	21-21-21	5-3	h-h-h-h-s	24,656	$3.00 \pm 0.83\%$
19	21-21-21	10-3	h-h-h-h-s	25,621	$2.88 \pm 0.61\%$
20	21-21-21	15-3	h-h-h-h-s	26,586	$3.73 \pm 0.64\%$

Flickr. Since each image source contributes to two image classes, an algorithm trained on this dataset cannot fully rely on camera characteristics to achieve accurate classification.

We used the Flickr API to download several thousand images licensed as CC-SA or CC-NC. We filtered using Exif data, keeping only images that were taken by the iPhone 4/4s/5 where the LED flash was reported as "fired". From those images we cropped out normal and pseudo-leukocoric eyes. See http://leuko.net/cnn2014/data/ for attributions.

Ground-truth classification of each eye image was performed by the authors of this paper. The set of leukocoric eye images and part of the set of normal eye images overlaps the dataset used in [2]. We assumed that the set images from Flickr do not demonstrate true leukocoria, as true leukocoria is rare while bright white-eye reflection in images taken by an iPhone using the flash is quite common.

III. Methods

A. Neural networks for classification

There are many modifications and adaptations to popular fully-connected feed-forward neural networks. One adaptation is the convolutional neural network (CNN), which is well-suited for image-processing tasks [3]. CNNs have been shown to significantly outperform many other state-of-the-art algorithms for classification of images [4].

CNNs differ from traditional neural networks by using a more biased structure akin to human visual processing. Specifically, the architecture is tailored in two ways. First, each neuron in a convolutional layer only receives input from a small patch of the entire image. This captures the idea of *receptive fields* (i.e. each neuron receives input from its own receptive field). Second, each neuron in a convolutional layer is replicated across the layer's input such that each replica has a different receptive field and the union of the receptive fields covers the entire input. This captures the idea of *filters* (i.e. each neuron in a convolutional layer acts as a translation invariant filter of the input it receives).

The first layer of a CNN learns filters similar in concept to the features that are learned by RISA [5], which has been used in the analysis of tumor signatures and outperforms the best known expert-designed feature detector for that task [6].

To use our training images as input to a CNN, we need only to scale each image such that they all have the same dimensions; we do not perform any other pre-processing. In this paper, we scale each image to 40×40 pixels using bilinear interpolation, and we use full color (RGB) as input.

To use a CNN for classification of our dataset into three classes, we use a three-neuron fully-connected layer as the output of the network. The output layer nodes are tied together as a soft-max group so that the outputs can be interpreted as probabilities, one for each of the possible image classifications. The predicted class is the one with the highest associated probability.

Because our training set is relatively small, we employ ten-fold cross-validation for estimating the generalization error of each network. We accumulate the performance on each fold in the results that follow.

B. ROC tuning

In this application, it is important to control false positive predictions so that a user will take seriously predictions of leukocoria. To that end, we develop a method of tuning the outputs of a trained network to manipulate its receiver operator characteristics (ROC).

Due to the soft-max group used in the output layer, the outputs sum to one. Naming those outputs x, y, and z, we create three positive constants a, b, and c which also sum to one. Thus, $(x + a)/2 + (y + b)/2 + (z + c)/2 = 1$.

We adapt the outputs of the network to be the values $x' = (x+a)/2$, $y' = (y+b)/2$, and $z' = (z+c)/2$. These output values still represent probabilities of each classification, but include a bias. The predicted class is still the one with the highest associated probability.

This method can be generalized to n outputs, requiring $n-1$ constant assignments. For $n = 2$ (binary classification), this is equivalent to the practice of setting a single threshold.

For tuning ROC true-/false-positive rates, we add the same constant to the *normal* and *pseudo-leukocoric* outputs. This allows us to set a single threshold value on the output.

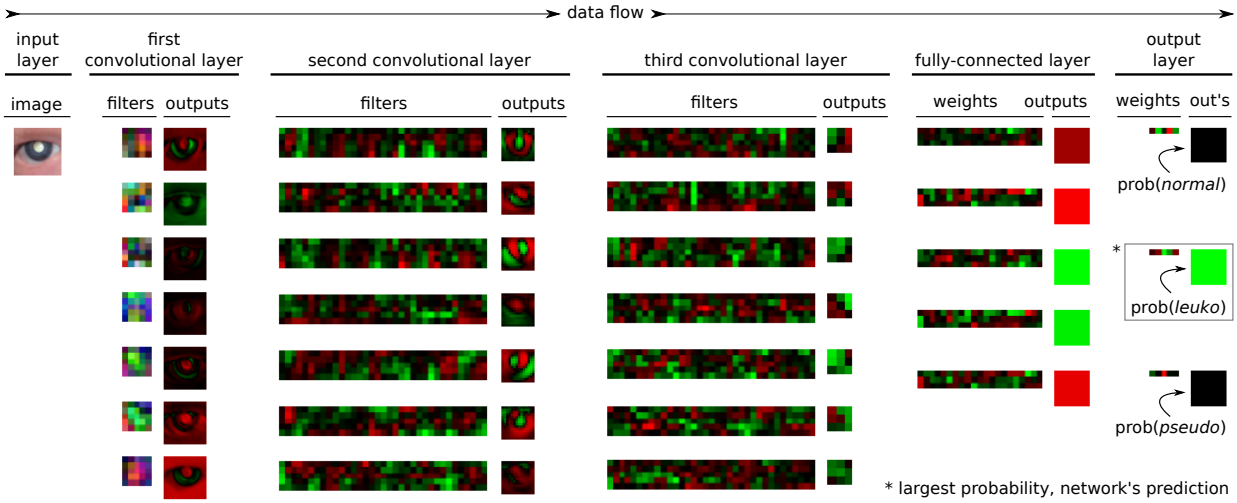


Figure 2: Visualization of passing a leukocoric image through network #16 (after training). Each set of filters and weights are static values (after training), while the outputs shown are for this particular input. The filters in layer 1 operate in RGB color, thus they are shown in RGB color. All other filters, weights, and outputs are shown in the red-green scale (red denotes negative values, green positive values, and intensity denotes magnitude). Max-pooling in convolutional layers one and two are not shown here. The output is correct on this input.

IV. Experiments and Results

We initially trained traditional three-layer fully-connected artificial neural networks. Such neural networks can be considered CNNs which lack convolutional layers. We then trained CNNs of varying architecture and depth in order to demonstrate their effects on performance. Each network has some number of convolutional layers followed by exactly two fully-connected layers, where the second of the fully connected layers (the output layer) always has exactly three neurons. See Table I for a description of each trained network and for the error rate achieved by each network. Table I is organized as follows:

- The "Convolutional Layers" column denotes the number of trainable filters in each layer, where left-to-right indicates lower-to-upper layers. In this paper, all convolutional layer neurons have a receptive field size of 5×5 pixels. Each first- and second-level convolutional layer is followed by a 2×2-pixel max-pooling layer to decrease network dimension.
- The "Fully-connected Layers" column denotes the number of hidden neurons in each fully-connected layer. All fully-connected layers are above all convolutional layers. Note that in all networks the last fully-connected layer has three neurons, representing the probability of each possible classification.
- The "Layer Types" column denotes the squashing (activation) function used by the neurons in each layer ('h' for hyperbolic-tangent, 's' for soft-max).
- The "# Free Parameters" column denotes the number of independently trainable parameters in the network. This is calculated by summing the number of independent weights in each layer of the network. Note that not every weight in a convolutional layer is independent—replicated neurons have tied weights.
- The "Error Rate" column denotes the error rate of the learner on the accumulated folds. Classification at this stage is done with no ROC turning.

The first 15 networks were trained using the *momentum* optimizer for updating the weights during training, a commonly used optimizer for neural networks [7]. Using momentum on the remaining (deeper) networks would require a very long training time, so we used a new optimizer, *RMSPROP*, designed for quickly training deep networks [8].

The network with the lowest error rate was network #16, which also has the least number of free parameters. Figure 2 shows a visualization of network #16. Note that most filters in the first layer do not respond to the iris; however, filters 5 and 6 of the first layer, which are activated by green pixels surrounded by darker pixels, are responding to the pupil. In the second layer, the network continues focusing on the pupil and now appears to ignore the surrounding skin, as indicated by the outputs having bright green centers. The remaining layers continue transforming the input into more abstract features until we get a representation of probabilities in the output layer. The prediction in the figure is correct.

Figure 3 shows a visual confusion matrix over the accumulated training folds, evaluated by network #16. Figure 4 demonstrates our method of tuning the ROC of network #16 to decrease the number of false positives at the expense of a few true positives, as described by Section III-B.

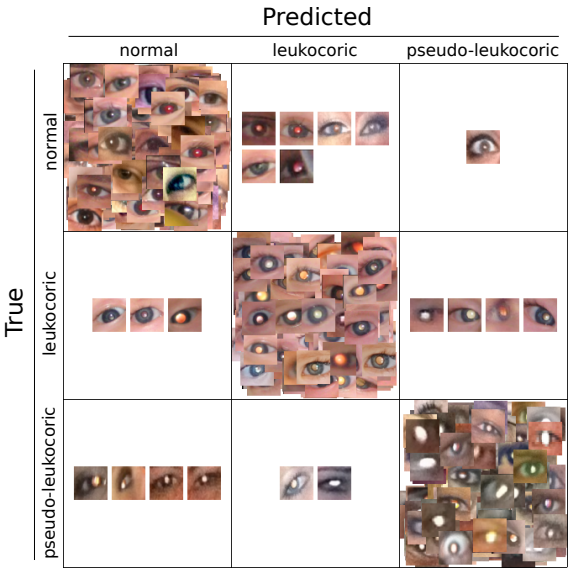


Figure 3: Visual confusion matrix over the accumulated training folds, evaluated by network #16.

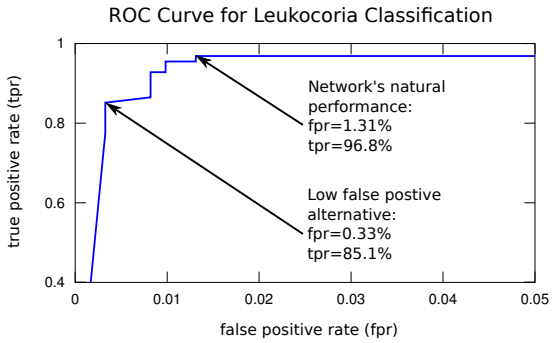


Figure 4: ROC tuning on network #16.

V. Conclusion

CNNs are a good tool for identifying leukocoria in recreational photography. With a moderate amount of data, no preprocessing (other than rescaling), and a relatively small network we are able to achieve excellent accuracy at this task, using a purely learning process.

CNNs produced much better results than did the traditional three-layer fully-connected neural networks. These better results are due to the biased architecture of the CNN, biased by the researcher toward a certain way of solving the classification problem (i.e. local connectivity and replicated neurons to produce trainable filters). Such a biased architecture still allows for a fully trainable system, requiring no hand-coded feature extractors.

For our task, where the training set is not large, we achieved better results from a small capacity network (i.e. one with few free parameters). Networks with large capacity are most appropriate when a lot of training data is available. We expect that we could achieve better results using such a network if we had more data.

ROC tuning proved useful, allowing the same network to be used in different contexts with different thresholds. It's likely preferable to have few false positives in recreational use, while more are acceptable in clinical settings where a professional can immediately validate predictions.

VI. Future Work

We are working on improving prediction accuracy on eye images that are poorly cropped. To achieve this, we will broaden our dataset by duplicating each example at differing scales, translations, and rotations. We will also investigate how performance improves when the images are pre-processed (using image registration and normalization).

The long-term goal of our research is to build a system for detecting leukocoric eyes in raw, full-scale recreational photographs. Such a system could be installed in recreational cameras and offer warnings when leukocoria is detected.

Acknowledgments

Thanks to Dr. Erich Baker for providing data support.

References

[1] D. Abramson, A. Schefler, I. Dunkel, B. McCormick, and K. W. Dolphin, "Pediatric ophthalmic oncology: Ocular diseases." in *Holland-Frei Cancer Medicine. 6th edition.*, 2003.

[2] A. Abdolvahabi, B. W. Taylor, R. L. Holden, E. V. Shaw, A. Kentsis, C. Rodriguez-Galindo, S. Mukai, and B. F. Shaw, "Colorimetric and longitudinal analysis of leukocoria in recreational photographs of children with retinoblastoma," *PLoS ONE*, vol. 8(10), 2013.

[3] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, 1995.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[5] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3361–3368.

[6] Q. V. Le, J. Han, J. W. Gray, P. T. Spellman, A. Borowsky, and B. Parvin, "Learning invariant features of tumor signatures," in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*. IEEE, 2012, pp. 302–305.

[7] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[8] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.