CLRGaze: Contrastive Learning of Representations for Eye Movement Signals

Louise Gillian C. Bautista
Department of Computer Science
University of the Philippines
Quezon City, Philippines
lcbautista1@up.edu.ph

Prospero C. Naval, Jr.
Department of Computer Science
University of the Philippines
Quezon City, Philippines
pcnaval@dcs.upd.edu.ph

***Abstract*—Eye movements are intricate and dynamic biosignals that contain a wealth of cognitive information about the subject. However, these are ambiguous signals and therefore require meticulous feature engineering to be used by machine learning algorithms. We instead propose to learn feature vectors of eye movements in a self-supervised manner. We adopt a contrastive learning approach and propose a set of data transformations that encourage a deep neural network to discern salient and granular gaze patterns. This paper presents a novel experiment utilizing six eye-tracking data sets despite different data specifications and experimental conditions. We assess the learned features on biometric tasks with only a linear classifier, achieving 84.6% accuracy on a mixed dataset, and up to 97.3% accuracy on a single dataset. Our work advances the state of machine learning for eye movements and provides insights into a general representation learning method not only for eye movements but also for similar biosignals.**

***Index Terms*—deep representation learning, contrastive learning, time-series, eye movements, convolutional neural network**

I. Introduction

Eye movements have long been used in Cognitive Science to understand how people think and perceive [1]–[4]. For example, longer fixations can indicate information processing [2], rapid saccades may imply a viewer's excitement [5], and microsaccades may indicate higher cognitive load [3]. Beyond cognitive signals, researchers have also found idiosyncratic patterns in eye movements, spurring research into eye movement biometrics [6]–[8].

Due to the breadth of information that can be extracted from gaze behavior, there is a wide array of studies on eye movements for use in domains such as education, safety, and healthcare [2], [4]. However, eye movements can be difficult to process because of factors that affect gaze behavior including the stimuli used, tasks given, and eye-tracker specifications. Researchers then have to carefully select their methodologies to emphasize the patterns specific to their use-case. For example, formulas for extracting features have to be tuned [9], and areas-of-interests have to be manually defined [4], [10]. However, these methods may not generalize well to other data sets and use-cases because of their dependence on the stimuli used, prior beliefs, and data specifications (e.g. sample length, sampling frequency).

We aim to do away with this arduous task of handcrafting features and manually selecting representation methods. We instead automatically encode eye movements into feature vectors that accurately capture the salient and granular characteristics of gaze behavior. Any new eye movement sample can be mapped to this vector which can then be used for downstream tasks. The goal is to (1) encode high-resolution data such that even the minute movements are accounted for and (2) make it easier to apply machine learning algorithms to eye movement data for real-world applications.

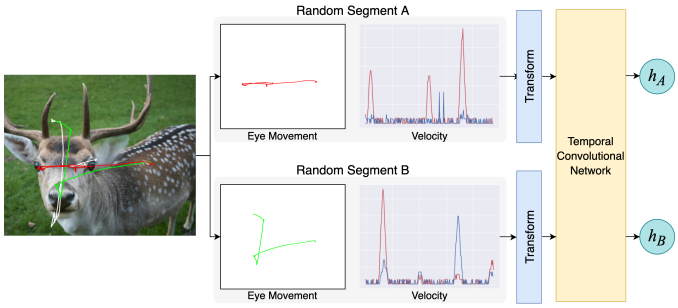


Fig. 1. Overview of CLRGaze. For any eye movement sample, random segments are taken and treated as velocity signals. These undergo transformations and are fed into an encoder as part of a batch. The encoder is then trained to discriminate segments that originated from the same signal, in effect learning meaningful abstract representations of eye movements.

We improve upon previous work [11] by taking a self-supervised contrastive learning approach, where a convolutional neural network (CNN) learns abstract representations of the data by being exposed to a large number of similar (positive) and dissimilar (negative) examples. We follow SimCLR [12], a contrastive learning framework for images. In this framework, representations are learned by comparing and contrasting different views (i.e. transformations) of images.

We port this methodology to the signal domain, specifically applying it to eye movements (Fig. 1). We propose a set of signal cropping and transformations that encourage the network to discern important patterns in eye movements. To assure the efficacy of our proposed methodology, we conduct our experiment on six eye-tracking data sets. The joint data set consists of 45,755 eye movement trials from 143 subjects. After training the contrastive CNN on the eye movements, we evaluate the learned representations through inter and intra-dataset biometric tasks with only a linear classifier. Despite

having different specifications, our method works well across these data sets, achieving 84.6% accuracy across all samples and up to 97.3% accuracy on a single data set. We also show that our model generalizes to unseen samples.

Our contributions are as follows: (1) we apply contrastive representation learning to eye movement signals, (2) we propose a set of signal data transformations to aid contrastive learning, (3) we demonstrate the validity of our method by conducting experiments on six data sets, and (4) we achieve superior accuracies and establish new baselines on biometric tasks.

II. Methodology

A. Contrastive Representation Learning

Our methodology is largely based on SimCLR [12]. For any image $x \in \mathbb{R}^{d_x}$ in a mini-batch of size N, two different random image augmentations are performed (e.g. cropping, blur, color distortion) to obtain two new sub-images x_a and x_b that form a positive pair. This doubles the mini-batch size to $2N$, where each image now has one positive example and $2(N-1)$ negative examples.

A CNN f encodes this mini-batch $\{x_k\} \in \mathbb{R}^{d_x}$ to their representations $\{h_k\}$ in a learned latent space $\mathbb{R}^{d_h}$. The representations are further mapped by a nonlinear projection head g to a small feature vector $\{z_k\} \in \mathbb{R}^{d_z}$ with which their intra-batch similarities are calculated. The encoder f and projection head g are jointly optimized such that the similarity between positive pairs are maximized while that of negative pairs are minimized. Specifically, they minimize the normalized temperature-scaled cross entropy loss (NT-Xent). For any data point i and j in the mini-batch, NT-Xent is computed as follows:

$$\ell_{i,j} = -log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} exp(sim(z_i, z_k)/\tau)} \quad (1)$$

where sim is the cosine similarity between two projections $sim(z_i, z_j) = z_i^T z_j / \|z_i\| \|z_j\|$, $\mathbb{1}$ is an indicator function evaluating to 1 when the data point is not being compared to itself, and τ is a temperature parameter that scales the similarity values. By training on larger batches and many iterations, the network is exposed to more positive and negative examples. Thus, allowing it to form richer abstractions about the data.

B. CLRGaze

We take inspiration from SimCLR and apply this framework to eye movement signals. We port their methodology to the signal or 1D time-series domain by performing a set of data transformations analogous to augmentation techniques for images or 2-D data.

A sample signal $x \in \mathbb{R}^{(2,T)}$ is a time-series with arbitrary length T and 2 channels corresponding to the x and y plane Each time step is an estimated position of a viewer's gaze on a screen. To produce a positive pair (x_a, x_b) from x, we must obtain two different views or segments $x_a, x_b \in \mathbb{R}^{(2,T')}$

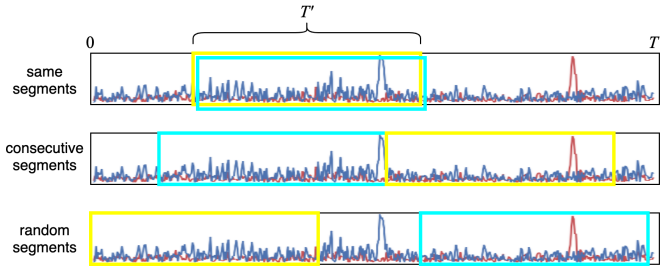


Fig. 2. Cropping methods applied to each signal to form a positive pair.

where $T' \leq T$ is a fixed input length. We do this by randomly selecting any of the three cropping methods visualized in Figure 2.

Given T' and a randomly selected time point t where $t \leq T' \leq T$, the methods are as follows: In (1) *Same*, the two segments are identical ($x_a = x_b = x_{t:t+T'}$). In (2) *Consecutive*, the two segments are consecutive portions of the signal, i.e. x_a immediately proceeds *or* precedes x_b (e.g. $x_a = x_{T'-t:t}$ and $x_b = x_{t:t+T'}$). In (3) *Random*, the two segments come from any random portion of the signal (e.g. $x_a = x_{t_1:t_1+T'}$ and $x_b = x_{t_2:t_2+T'}$). After cropping, we separately apply, with uniform probability, one out of nine transformations listed in Table I to both segments. The transformations alter or destroy the signal encouraging the network to find unique patterns and be robust to noise.

Our encoder f is a six-layer temporal convolutional network (TCN) [14] with residual and squeeze-and-excitation blocks [15], [16] detailed and visualized in Figure 3. Following SimCLR, our projection head is a multilayer perceptron (MLP) with one ReLU-activated hidden layer.

III. Data

We utilize six public data sets, having different specifications such as viewer demographics, experimental conditions,

TABLE I
THE SET OF SIGNAL TRANSFORMATIONS USED IN CLRGAZE. FOR EACH CROPPED SEGMENT, WE RANDOMLY CHOOSE ONE WITH UNIFORM PROBABILITY (11.11%) AND APPLY TO THE SIGNAL.

Transformation	Description
1. Identity	No transformation.
2. Dropout	Randomly zero out 20% of time points in the signal in both x and y dimensions.
3. Chunk Drop.	Zero out a 20% chunk of the signal in both x and y dimensions.
4. Alternate Drop.	Alternately zero out time points in both x and y dimensions.
5. Channel Drop.	Zero out either the x or y dimension.
6. Gaussian Noise	Apply additive noise sampled from $\mathcal{N}(0, 0.5)$
7. Drop. & Noise	Randomly zero out 20% of time points and apply additive noise sampled from $\mathcal{N}(0, 0.5)$.
8. Chunk Copy	Replace a 20% chunk of the signal with a different 20% chunk within the same signal.
9. Chunk Swap	Swap two disjoint 20% chunks of the signal.

Chunk transforms were partially inspired by [13].

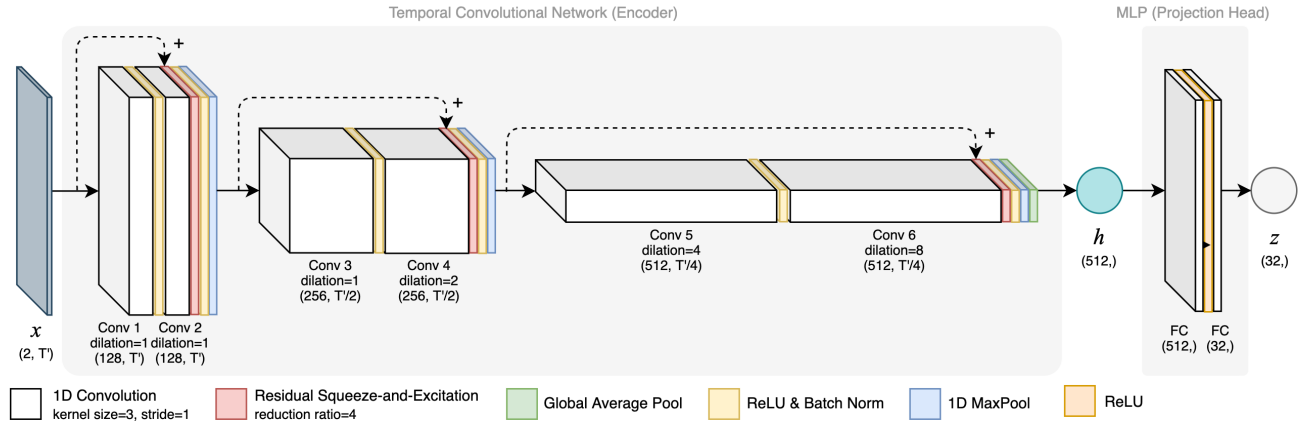


Fig. 3. Overview of the TCN encoder and the MLP projection head used in CLRGaze, with 2,147,072 parameters. A mini-batch $\{x_k\}$ of transformed segments $x \in \mathbb{R}^{(2,T')}$ passes through this network. NT-XEnt loss is calculated on z, while the final representation h is taken as the output of the Global Average Pooling (GAP) layer.

TABLE II

THE SIX DATA SETS USED JOINTLY TO LEARN EYE MOVEMENT REPRESENTATIONS. *Hz*: SAMPLING FREQUENCY OF THE EYE-TRACKER, E.G. 500 HZ = 500 TIME POINTS RECORDED PER SECOND. *Time (s)*: TIME IN SECONDS SPENT BY THE VIEWERS LOOKING AT THE STIMULI.

Data Set	Stimuli	Tasks	Eye-Tracker	Hz	Time (s)	Viewers	Samples
EMVIC [7]	normalized face images	free-viewing	Jazz-Novo	1000	2.5 (ave)	34	1430
FIFA [17]	indoor, outdoor scenes	free-viewing, search	SR Research Eye-Link	1000	2	8	3200
ETRA [3], [18]	everyday scenes, puzzles	free-viewing, search	SR Research Eye-Link II	500	45	8	960
MIT-LR [19]	outdoor scenes, pink noise	free-viewing	ETL 400 ISCAN	240	3	64	12,352
MIT-LTP [20]	outdoor scenes, portraits	free-viewing	ETL 400 ISCAN	240	3	15	15,045
MIT-Search [21]	outdoor scenes	search	ISCAN RK-464	240	1.2 (ave)	14	12,768
Total						**143**	**45,755**

and equipment (listed in Table II). We stress that this further makes representation learning a non-trivial task due to their inherent variances. Nevertheless, we chose to experiment on this scale as this is necessary to evaluate the usability and generalizability of our method. Note that our work is limited to eye movements obtained from viewing static images. Eye movements obtained during reading, watching videos, or "in-the-wild" are out of scope.

To enforce some uniformity in our joint data set, we scale the coordinates such that the viewers' one degree of visual angle corresponds to 35 pixels (35px/dva). We work at a sampling frequency of 500 Hz. We downsample 1000Hz to 500Hz by dropping every other time point, and we upsample 240Hz to 500Hz by cubic interpolation. We then opted to work with velocity signals as these were shown to be more meaningful [11]. Note it is still possible to use our methodology on position signals if preferred.

IV. EXPERIMENTS

A. Training

We train two networks: **CG**, which is trained on all available data and **CG-3**, a model trained only on EMVIC, FIFA, and ETRA data sets. We do this to have a fair comparison with the velocity models of GazeMAE (GM_v) [11], a previous work on eye movement representation learning using deep convolutional autoencoders. CG and CG-3 employ all cropping methods and data transformations, and have the same network architecture described in Section II-B and Figure 3.

Our inputs are 1-second segments or 500 time points ($T' = 500$). Our encoder f then maps a sample signal $x \in \mathbb{R}^{(2,500)}$ to its representation $h \in \mathbb{R}^{512}$. We train our networks to minimize the NT-Xent Loss (Eq. 1) with $\tau = 0.3$, learning rate=5e-4, batch size=1000 and Adam optimizer [22]. CG-3 is trained for 100 epochs while CG is trained for 800. For a training set of 45,755 samples, this results to 45 batches per epoch. We train for 800 epochs or 36,000 iterations. We notice that performance does not improve beyond this. The networks are implemented with PyTorch 1.7 [23] and trained on an NVIDIA RTX 2080Ti GPU. We compute with automatic mixed precision (AMP) to enable larger batch sizes and faster training time. All parameters are chosen empirically, and a random seed was set for all experiments. Our code will be made available at https://github.com/chipbautista/clrgaze.

B. Evaluation on Downstream Tasks

We use the trained network to encode eye movements into feature vectors. We now input full-length samples $x \in \mathbb{R}^{(2,T)}$ instead of fixed-length segments used for training. The GAP layer allows our encoder to handle arbitrary lengths, making it a more practical approach. The feature vectors are then

used as inputs to a linear classifier, which is a standard evaluation method for representation learning [24]. In our case, our classifier is a linear Support Vector Machine (SVM) implemented through scikit-learn library [25].

Limited by the available data labels, we opt to evaluate the learned representations by classifying the viewers based on their eye movements. This is also known as eye movement biometrics, a growing research area that, if done with traditional feature extraction methods, requires extensive knowledge on eye movements [6]. When possible, we compare our results with other works that have conducted the same tasks.

TABLE III
ACCURACIES ACHIEVED ON BIOMETRIC TASKS USING THE LEARNED REPRESENTATIONS AS INPUT TO A LINEAR SVM CLASSIFIER.

	Others	GM_v [11]	**CG-3** (ours)	**CG** (ours)
EMVIC-Train	86.0 [8]	86.8	**94.2**	92.7
EMVIC-Test	81.5 [8] 82.3** 86.4**	87.8	**94.5**	94.3
EMVIC,ETRA, FIFA	-	79.8	**96.6**	96.5
ETRA	-	-	**96.0**	95.0
FIFA	-	-	97.0	**97.3**
MIT-LR	-	-	60.6*	**82.9**
MIT-LTP	-	-	74.0*	**90.5**
MIT-Search	-	-	62.9*	**73.2**
All	-	-	69.5*	**84.6**

* data set not used in training
** mentioned in [8] but no citation was found

From Table III, it is shown that CLRGaze outperforms the previous works in all tasks. We believe that this boost can be attributed to our methodology. Recall that the contrastive CNN has to correctly classify if two segments originated from the same eye movement signal. To do so, it has to extract the patterns that are present throughout the signal, which are patterns that are likely to be idiosyncratic or unique to the viewer. Applying random cropping and chunk transformations to the eye movements further encouraged the CNN to extract these global information patterns. This concept is related to slow feature analysis and contrastive predictive coding [26], [27].

Also, notice that substantially lower accuracies were achieved for the MIT data sets, which may indicate that 240Hz eye-trackers cannot capture granular information needed for biometric tasks.

C. Effect of cropping methods and data transformations

Next, we train more models with the same parameters, changing only the composition of data transformations. We evaluate on the Biometrics (All) task since we deem this the most difficult. From Table IV, it is shown that the choice of cropping methods and data transformations largely impacts accuracy. While we present only the results for Biometrics (All) for simplicity, note that we observed the same trend when evaluated on other tasks.

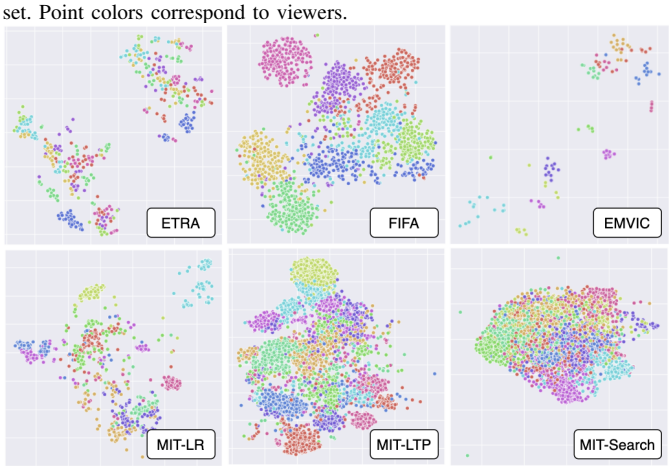
TABLE IV
ACCURACIES ACHIEVED ON THE BIOMETRICS (ALL) TASK BY MODELS TRAINED WITH VARYING CROPPING METHODS AND DATA TRANSFORMATIONS.

	Biometrics (All)
Cropping Methods (refer to Figure 2)	
Same	75.3
Consecutive	79.3
Random	84.1
Consec, Same	81.4
Random, Same	84.9
Random, Consec	83.2
Transformations (refer to Table I)	
None (#1 only)	78.9
Dropout (#1-5)	82.7
Dropout, Noise (#1-7)	83.9
Full model (**CG**)	84.6

D. CLRGaze generalizes to unseen samples

Finally, we train a model with the same parameters but on a viewer-stratified split (22,877 training and 22878 validation samples). In Figure 4, we plot the representations of the validation set, using the viewers as the labels. Eye movements of a subject lie close together in the representation space, suggesting that our model can represent unseen samples sufficiently.

Fig. 4. t-SNE [28] plots of the representations for validation samples, by data set. Point colors correspond to viewers.



V. CONCLUSION

We take on a contrastive learning approach based on SimCLR [12] to learn representations of eye movement signals. To port this methodology to the signal domain, we propose a set of data transformations that encourage a contrastive CNN to extract meaningful patterns from signals. We apply this methodology to six eye-tracking data sets despite varying specifications. The learned representations are evaluated with biometric tasks and a linear classifier, achieving high accuracies and outperforming previous works. Lastly, we show that the model can handle unseen samples well. This work presents a medium-scale experiment that advances eye movements-based deep learning applications.

REFERENCES

[1] P. König, N. Wilming, T. Kietzmann, J. Ossandón, S. Onat, B. Ehinger, R. Gameiro, and K. Kaspar, "Eye movements as a window to cognitive processes," *Journal of Eye Movement Research*, vol. 9, no. 5, Dec. 2016. [Online]. Available: https://bop.unibe.ch/JEMR/article/view/3383

[2] A. T. Duchowski, *Eye Tracking Methodology*. Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-57883-5

[3] J. Otero-Millan, X. G. Troncoso, S. L. Macknik, I. Serrano-Pedraza, and S. Martinez-Conde, "Saccades and microsaccades during visual fixation, exploration, and search: Foundations for a common saccadic generator," *Journal of Vision*, vol. 8, no. 14, pp. 21–21, Dec. 2008. [Online]. Available: https://doi.org/10.1167/8.14.21

[4] A. Coutrot, J. H. Hsiao, and A. B. Chan, "Scanpath modeling and classification with hidden markov models," *Behavior Research Methods*, vol. 50, no. 1, pp. 362–379, Apr. 2017. [Online]. Available: https://doi.org/10.3758/s13428-017-0876-8

[5] L. L. D. Stasi, A. Catena, J. J. Cañas, S. L. Macknik, and S. Martinez-Conde, "Saccadic velocity as an arousal index in naturalistic tasks," *Neuroscience & Biobehavioral Reviews*, vol. 37, no. 5, pp. 968–975, Jun. 2013. [Online]. Available: https://doi.org/10.1016/j.neubiorev.2013.03.011

[6] I. Rigas and O. V. Komogortsev, "Current research in eye movement biometrics: An analysis based on BioEye 2015 competition," *Image and Vision Computing*, vol. 58, pp. 129–141, Feb. 2017. [Online]. Available: https://doi.org/10.1016/j.imavis.2016.03.014

[7] P. Kasprowski and K. Harezlak, "The second eye movements verification and identification competition," in *IEEE International Joint Conference on Biometrics*. IEEE, Sep. 2014. [Online]. Available: https://doi.org/10.1109/btas.2014.6996285

[8] S. Mukhopadhyay and S. Nandi, "LPiTrack: Eye movement pattern recognition algorithm and application to biometric identification," *Machine Learning*, vol. 107, no. 2, pp. 313–331, Jun. 2017. [Online]. Available: https://doi.org/10.1007/s10994-017-5649-1

[9] I. Rigas, L. Friedman, and O. Komogortsev, "Study of an extensive set of eye movement features: Extraction methods and statistical analysis," *Journal of Eye Movement Research*, vol. 11, no. 1, Mar. 2018. [Online]. Available: https://bop.unibe.ch/JEMR/article/view/3795

[10] N. C. Anderson, F. Anderson, A. Kingstone, and W. F. Bischof, "A comparison of scanpath comparison methods," *Behavior Research Methods*, vol. 47, no. 4, pp. 1377–1392, Dec. 2014. [Online]. Available: https://doi.org/10.3758/s13428-014-0550-3

[11] L. G. C. Bautista and P. C. N. Jr., "Gazemae: General representations of eye movements using a micro-macro autoencoder," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[13] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[14] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016. [Online]. Available: https://doi.org/10.1109/cvpr.2016.90

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018. [Online]. Available: https://doi.org/10.1109/cvpr.2018.00745

[17] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in neural information processing systems*, 2008, pp. 241–248.

[18] M. B. McCamy, J. Otero-Millan, L. L. D. Stasi, S. L. Macknik, and S. Martinez-Conde, "Highly informative natural scene regions increase microsaccade production during visual scanning," *Journal of Neuroscience*, vol. 34, no. 8, pp. 2956–2966, Feb. 2014. [Online]. Available: https://doi.org/10.1523/jneurosci.4448-13.2014

[19] T. Judd, F. Durand, and A. Torralba, "Fixations on low resolution images," *Journal of Vision*, vol. 10, no. 7, pp. 142–142, Aug. 2010. [Online]. Available: https://doi.org/10.1167/10.7.142

[20] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[21] K. A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva, "Modelling search for people in 900 scenes: A combined source model of eye guidance," *Visual Cognition*, vol. 17, no. 6-7, pp. 945–978, Aug. 2009. [Online]. Available: https://doi.org/10.1080/13506280902834720

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013. [Online]. Available: https://doi.org/10.1109/tpami.2013.50

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation*, vol. 14, no. 4, pp. 715–770, Apr. 2002. [Online]. Available: https://doi.org/10.1162/089976602317318938

[27] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.