IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

Learning Human Perceptual Hysteresis via Deep CNN-RNN Networks

Jason Hwang

Cherry Creek High School, 9300 East Union Avenue, Greenwood Village, CO, 80111, USA; jhwang112358@gmail.com

ABSTRACT: For many years, neuroscientists have been interested in modeling human perceptual hysteresis. However, previous models were hard to tune and could not generalize across different datasets. To address this, we propose a supervised convolutional neural network - recurrent neural network (CNN-RNN) that learns hysteresis directly from data instead of separate simulation models. To control the hysteresis effect, we also introduce a regularization term that penalizes changes in low-confidence classes. We show that our CNN-RNN model successfully mimics human perceptual hysteresis across a large number of examples. Our model is expected to decrease radiologists' hysteresis on disease screens by finding a sequence of images that has no perceptual hysteresis.

KEYWORDS: Behavioral and Social Studies, Cognitive Psychology, Perceptual Hysteresis, Deep Learning, CNN-RNN.

Introduction

Various artificial intelligence (AI) techniques have been developed, outperforming handcrafted techniques in many applications. We have been particularly interested in AI that embodies human characteristics. Human perception is one of the areas we have looked into, and its hysteresis property is what we focus on in this paper. Human perception has evolved to maintain its stability and consistency. This is especially apparent when humans perceive ambiguous images. Over many years, scientists have documented how humans classify images based on what they saw previously. For example, van Rooji *et al*. found that people's perception of a sequence of ambiguous Necker cubes, whose ambiguity decreased or increased, was strongly influenced by the order in which they saw the sequence.[1] Figure 1, for example, illustrates this phenomenon. Depending on the sequence, i.e., from left to right or from right to left, human perception shows hysteresis. The degree of such hysteresis also depends on the image details. Many previous studies have tried to model this behavior computationally using competitive attractor networks, spiking neuron networks, slow dynamic systems, etc.[2,3] However, these models are hard to tune and don't learn the underlying dynamics from data. Modern deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are much easier to tune and learn features directly from data. In this paper, we propose a new approach to modeling human perceptual hysteresis using CNNs and RNNs.

Human perception maintains only one perspective of an ambiguous image at any time.[4] This has been hard to replicate using current modeling methods. Hence, we propose a new low-confidence regularization term that tries to remove the ambiguity during the class transition. In this way, the hysteresis effect can be controlled, which is valuable for creating a universal model that adapts to the radiologists' perception.

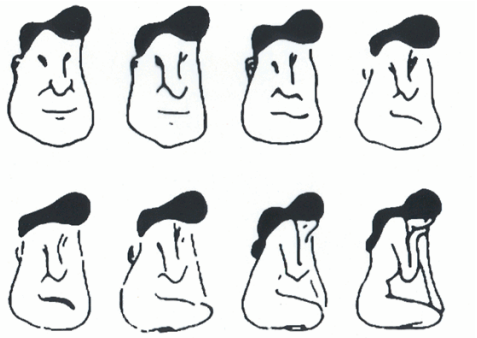

Figure 1: Illustration of human perceptual hysteresis.[5] A sequence of images morphs a man's face into a seated girl. If a person classifies the images from top left to bottom right, the initial perception of a man's face might cause them to misclassify later images of a seated girl as a man's face due to prior perception influencing the classification of subsequent images.

Methods

Convolutional neural networks (CNNs) are commonly used for image recognition. Compared to a traditional neural network, CNNs include convolutional and pooling layers. Pooling layers perform spatial downsampling, which decreases computational cost and increases the model's generalization ability. Convolutional layers apply a shared set of parameters across all parts of the image. This allows the model to efficiently learn location-independent features of an image, such as edges and textures.

A major limitation of traditional CNNs, such as VGG and AlexNet, is their reduced ability to scale with more layers.[6] While adding more layers to a neural network tends to increase performance on training data, this has generally been false for very deep neural networks. To address this issue, He *et al*. proposed residual networks (ResNets), which incorporate residual blocks. Residual blocks include a skip connection that adds the input activation to the last layer's pre-activation before applying the final activation. These skip connections facilitate

DOI: 10.36838/IJHSR86.69

identity mappings, enabling faster backpropagation and improved performance. Our model incorporates ResNets to learn spatial features from the image data.

To model human perceptual hysteresis, a ResNet by itself is insufficient. ResNets are designed to extract complex spatial features of images and are not optimized for capturing the long-range dependencies of sequential data. In other words, the ResNets alone would produce the same inference output of an image, no matter what the sequence of the test images is. In contrast, recurrent neural networks (RNNs) are well-suited for analyzing sequence data. Hence, our model processes the features extracted by the ResNet with an RNN to take advantage of each architectures' strengths.

For analyzing sequence data such as videos or voice recordings, recurrent neural networks (RNNs) are widely used. RNNs enable the network to retain information from previous inputs when processing the current input. This is crucial because earlier information provides context for interpreting later data. For example, when analyzing text, the word "Teddy" could either refer to President Teddy Roosevelt or a Teddy bear. In a sentence, remembering the previous word "President" would help solve this ambiguity. Hence, RNNs are suitable for sequence recognition.

However, traditional RNNs have major limitations, including difficulty remembering information over long sequences and the vanishing gradient problem. The long short-term memory network (LSTM) helps alleviate these issues.[7] Instead of only remembering information from the previous input, LSTMs utilize gates and memory cells to determine what previous information should be kept and forgotten over the entire sequence. This helps the network optimize parameters faster and increase its performance.

Our complete architecture consists of a pretrained ResNet50 feature extractor followed by an LSTM layer. Our ResNet50 feature extractor was pretrained with its last layer replaced by one neuron for binary classification. The training data was provided from the IllusionBench dataset.[8]

For training the entire ResNet50-LSTM network, we froze all layers of the pretrained ResNet50 feature extractor, removed the last fully connected and average pooling layers, and added an adaptive pooling layer. Hence, the output of the feature extractor matched the input size of the LSTM. The individual images from the input sequence are first propagated through the feature extractor and then trained by the LSTM. Since the output data is binary, the binary cross-entropy loss was chosen. The ADAM optimizer was used for training the LSTM. Figure 2 illustrates the proposed model schematically.

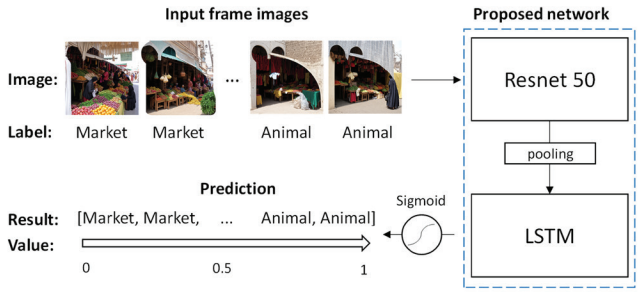

Figure 2: Overall workflow of the proposed method. Labeled input frames are processed by a supervised network composed of ResNet50 and an LSTM. The network's raw outputs are converted into values between 0 and 1 using a sigmoid function, with animal images assigned a value of 1 and non-animal images a value of 0.

In obtaining our data, we preprocessed images from the IllusionBench dataset. Only the animal-market and animal-village subsets were used. In total, there are 1068 training images, 228 cross-validation images, and 228 testing images. To preprocess our training and validation data, we took j consecutive market images and 8-j consecutive animal images and concatenated them in order. j is a random number between 2 and 6. Our testing dataset instead concatenated 4 consecutive animal images and 4 consecutive market images. We resized all images to 512 x 512, converted them to a float32 numpy array, and reshaped the sequence for PyTorch syntax. We chose the PyTorch library because the ResNet50 and LSTM frameworks are preserved in the library. The default hyperparameters of the ResNet50 and LSTM were used. We deliberately decided to use differently formatted sequence data for training/validation and testing. Since shuffle is off to maintain the structure of the sequence, a fixed number of images for each class could allow the model to rely on sequence indices to predict class labels (e.g., a fixed training sequence similar to the test sequence could result in the model always predicting [1 1 1 1 0 0 0 0]). Hence, varying the number of images of each class forces the model to learn the visual dynamics in the sequence.

Humans tend to perceive only one interpretation of an ambiguous image at a given moment.[4] To extract this behavior into our model, we introduce a low confidence regularization term that penalizes low confidence class transitions. This idea is related to entropy minimization, first suggested by Grandvalet and Bengio.[9] Entropy minimization increases the performance of binary classification models by pushing the decision boundary into low-density regions of the input space. Specifically, our regularization term singles out probabilities that induce class transition. Using 0.5 as the threshold, the model incurs a penalty with a Gaussian distribution centered at 0.5. The standard deviation chosen for the Gaussian distribution is 0.1. The unimodal distribution of the penalty ensures that there is a large loss for a class change of low confidence, which is defined as a probability close to 0.5. For example, if the output probabilities were [1, 1, 1, 1, 1, 1, 0.4, 0], the class transition at the probability of 0.4 will induce a large penalty. This regularization is excluded for the loss of the first and last images of the sequence. The low-confidence regularization term is defined as follows.

We first convert the model's output probability $p_{b,t} \in [0,1]$ into a hard decision $\hat{p}_{b,t} \in \{0,1\}$ using a fixed threshold $\tau = 0.5$:

$$\hat{p}_{b,t} = \begin{cases} 1, & \text{if } p_{b,t} \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

A temporal flip occurs when two consecutive thresholded predictions disagree:

$$\text{flip}_{b,t} = \begin{cases} 1, & \text{if } \hat{p}_{b,t} \neq \hat{p}_{b,t+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

For each adjacent pair, we compute the average confidence:

$$a_{b,t} = \frac{p_{b,t} + p_{b,t+1}}{2}. \tag{3}$$

We penalize flips that occur when the average confidence is near the threshold, using a Gaussian penalty centered at $\mu = 0.5$:

$$g_{b,t} = \exp\left(-\frac{(a_{b,t} - \mu)^2}{2\sigma^2}\right). \tag{4}$$

The penalty for timestep t of batch element b is:

$$\text{penalty}_{b,t} = \text{flip}_{b,t} \cdot g_{b,t}. \tag{5}$$

Finally, the loss is computed as the mean penalty across all batch elements and all valid temporal positions:

$$\mathcal{L} = \frac{1}{B(T-1)} \sum_{b=1}^{B} \sum_{t=0}^{T-2} \text{penalty}_{b,t} \tag{6}$$

$$= \frac{1}{B(T-1)} \sum_{b=1}^{B} \sum_{t=0}^{T-2} \text{flip}_{b,t} \cdot \exp\left(-\frac{\left(\frac{p_{b,t}+p_{b,t+1}}{2} - \mu\right)^2}{2\sigma^2}\right). \tag{7}$$

By encouraging confident transitions, the model becomes more sensitive to changes in the ground truth label due to a more distinct decision boundary. Therefore, the model is less likely to maintain the previous classification when the ground truth label changes. We hypothesize that applying the low-confidence regularization will reduce the hysteresis effect.

To evaluate model performance, we examined how its predicted probabilities changed over time relative to the ground truth. A model was considered to successfully reproduce human perceptual hysteresis if its class transitions showed a delay, relative to the ground truth, that matches the delay observed in human non-hysteresis and hysteresis classification patterns (Figure 3). A representative output from our test set is shown in Figure 4. The class transition from Animal to Market is delayed by 2 images due to perceptual hysteresis. The three models that we tested were ResNet50, ResNet50-LSTM, and ResNet50-LSTM with a low confidence penalty.

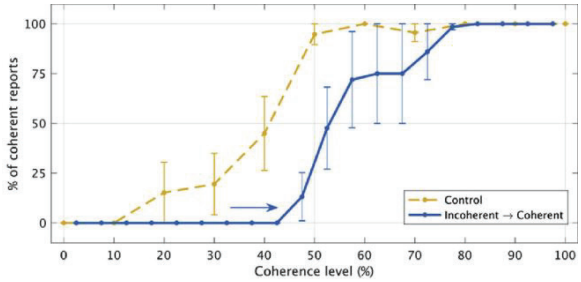

Figure 3: Illustration of human perceptual hysteresis.[10] The yellow line shows results under no hysteresis. The blue line shows classifications under consecutive stimuli, which induces hysteresis as shown by the delayed transition from incoherent to coherent classifications relative to the yellow line. The independent variable: coherence level describes the verticality of the dots' movement in a stimulus, with high levels inducing vertical movement and low levels inducing horizontal movement. Subjects predict whether the current stimuli are coherent (dots moving vertically) or incoherent (dots moving horizontally). Taking the percentage of a subject classifying the stimuli as coherent gives the dependent variable.

Figure 4: Example results of the proposed method. The model incorrectly classifies later Market images as Animal images because it saw Animal images first. This demonstrates that the network reproduces perceptual hysteresis, exhibiting prediction biases that depend on the sequence of preceding frames.

Results and Discussion

Results:

Using our evaluation metric, the ResNet50-LSTM model without the low-confidence penalty demonstrated a significant transition delay for time-reversed sequences. The ResNet50-LSTM model with the low confidence penalty exhibited a decreased hysteresis effect, as shown by the much steeper transition from Animal to Market. The ResNet50 model without the LSTM showed no hysteresis effect. Figure 5 illustrates our results. However, all models exhibited high standard deviation. Tables 1, 2, and 3 show the test results for each model. Figure 6 shows the training and validation loss graph.

Table 1: Test results for ResNet50-LSTM without confidence loss

Sequence number	Probability by frame							
1	0.6587	0.9974	0.9987	0.9993	0.9976	0.9361	0.3873	0.0358
2	0.6320	0.2662	0.1364	0.9668	0.1391	0.0038	0.0016	0.0009
3	0.9969	0.9990	0.9994	0.9993	0.9953	0.9203	0.9405	0.1682
4	0.8246	0.9955	0.9987	0.9993	0.9903	0.7429	0.1502	0.0249
5	0.9971	0.9991	0.9994	0.9995	0.9989	0.9947	0.9944	0.9656
6	0.9969	0.9992	0.9993	0.9993	0.9975	0.9886	0.7644	0.2174
7	0.9151	0.9973	0.9988	0.9991	0.9951	0.9244	0.9880	0.7372
8	0.9917	0.9987	0.9992	0.9993	0.9988	0.9924	0.9925	0.7651
9	0.1968	0.9971	0.9980	0.9992	0.9881	0.6069	0.2467	0.0187
10	0.9066	0.9979	0.9992	0.9991	0.9848	0.6876	0.6970	0.1125
11	0.9930	0.9990	0.9989	0.9993	0.9979	0.9948	0.8864	0.8014
12	0.9931	0.9988	0.9983	0.9993	0.9969	0.9085	0.9610	0.8966
13	0.9953	0.9989	0.9992	0.9994	0.9968	0.9492	0.8475	0.6320
14	0.9959	0.9989	0.9993	0.9994	0.9994	0.9991	0.9945	0.9524

Table 2: Test results for ResNet50-LSTM with confidence loss

Sequence number	Probability by frame							
1	0.0546	0.9545	0.9877	0.9913	0.9845	0.3052	0.0431	0.0203
2	0.3485	0.0088	0.6143	0.9775	0.8326	0.0656	0.0304	0.0124
3	0.9279	0.7462	0.9331	0.9775	0.6191	0.0627	0.5441	0.0245
4	0.0791	0.0060	0.5669	0.9442	0.1518	0.0163	0.0273	0.0074
5	0.9756	0.9891	0.9908	0.9906	0.9830	0.8039	0.8859	0.4226
6	0.9723	0.9899	0.9920	0.9923	0.9015	0.9244	0.4596	0.1951
7	0.9089	0.9825	0.9873	0.9894	0.9811	0.7676	0.9036	0.5493
8	0.7863	0.9205	0.9652	0.9826	0.8346	0.2072	0.0949	0.0208
9	0.9315	0.9842	0.9804	0.9876	0.9132	0.7778	0.4422	0.0669
10	0.3055	0.9649	0.9842	0.9898	0.9214	0.3008	0.6480	0.4266
11	0.1504	0.9166	0.9559	0.9820	0.7734	0.8355	0.1274	0.0375
12	0.4230	0.9458	0.3874	0.9771	0.7853	0.0847	0.1275	0.0258
13	0.8578	0.9794	0.9894	0.9869	0.8948	0.4482	0.3713	0.3720
14	0.9575	0.9883	0.9918	0.9898	0.9693	0.9466	0.3470	0.0714

DOI: 10.36838/IJHSR86.69

Table 3: Test results for ResNet50

Sequence number	Probability by frame							
1	0.944	1.000	1.000	1.000	0.005	0.000	0.000	0.000
2	0.970	0.990	0.989	1.000	0.000	0.000	0.000	0.000
3	1.000	1.000	0.988	1.000	0.002	0.000	0.013	0.000
4	0.766	0.993	1.000	0.996	0.000	0.000	0.000	0.000
5	1.000	0.999	0.902	0.996	0.000	0.000	0.651	0.011
6	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
7	0.977	1.000	1.000	1.000	0.000	0.000	0.459	0.000
8	1.000	1.000	1.000	1.000	0.001	0.000	0.000	0.000
9	0.917	1.000	0.994	1.000	0.003	0.000	0.000	0.000
10	0.907	1.000	1.000	1.000	0.000	0.000	0.947	0.000
11	0.998	0.998	0.990	1.000	0.000	0.176	0.000	0.000
12	0.998	0.995	0.014	0.999	0.000	0.000	0.002	0.000
13	0.997	1.000	1.000	1.000	0.000	0.000	0.008	0.000
14	1.000	1.000	1.000	1.000	0.984	0.267	0.000	0.000

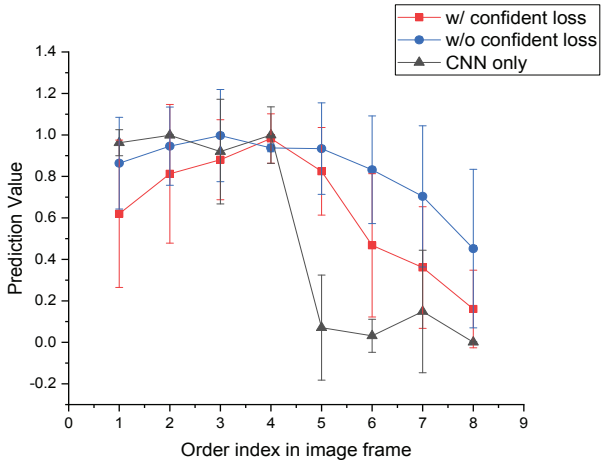

Figure 5: Mean prediction value (probability) curves across the image sequence. The curve without the confidence penalty exhibits pronounced hysteresis bias, while the curve with the low-confidence penalty shows less hysteresis. The error bars represent the standard deviation at each index.

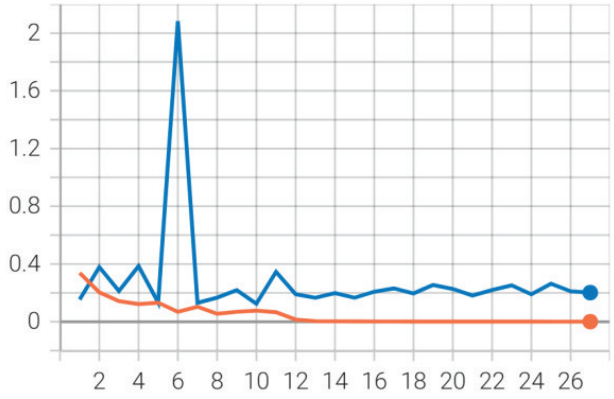

Figure 6: Training loss graph of the ResNet50-LSTM without confidence loss. The training loss and validation loss are shown by the orange line and blue line, respectively.

Discussion:

The results demonstrated that human perceptual hysteresis can be modeled using a combination of CNN and LSTM architectures. The delayed class transitions observed by our two ResNet-LSTM models resemble the tendency to repeat previous choices found in behavioral patterns of human perception. This suggests that memory-based algorithms such as LSTMs can compensate for the lack of temporal context in CNNs.

Furthermore, our results confirm that the low-confidence regularization term decreases the hysteresis effect by increasing the emphasis on the ground truth. This allows the model to generalize across varying strengths of the hysteresis effect.

Although we have successfully demonstrated that the model exhibits perceptual hysteresis on time-reversed sequences, this study has a limitation in that the model's dependency on the image details has not been explored. The gradual change of image details in Figure 1 significantly induces a hysteresis effect of human perception, while our dataset has image transitions clear to the human observer that reduce the hysteresis effect in Figure 4.

The relatively high standard deviation exhibited by ResNet50-LSTM models seems to be consistent with patterns observed in human perceptual hysteresis. However, because Sayal *et al.*'s design uses 11 discrete stimulus levels while the response at each level is binary, the variance at several mid-level stimuli is inherently high.[10] At ambiguous stimulus levels where participants respond approximately equally with 0 or 1, the data are maximally variable. Thus, the large standard deviations in their results mainly reflect the experimental structure, especially the ambiguous stimulus levels where $p \approx 0.5$, and should not be interpreted as a consequence of perceptual hysteresis. This conclusion is consistent with Schwiedrzik *et al.*'s findings.[11] They argue that perceptual hysteresis stabilizes perception by introducing a bias toward previous stimuli, which doesn't increase variance. Since hysteresis acts by shifting the prior rather than increasing sensory noise, it influences the mean perceptual response without increasing variability.

From further analysis, the high standard deviation exhibited by both our ResNet50-LSTM models is likely a result of how hysteresis varies across different stimuli. According to Pasccui *et al.*, uncertainty and the stability of attention can strongly influence perceptual hysteresis in humans.[12] Since there is research that supports how algorithms can model the effects of uncertainty and stability of attention,[13,14] the high standard deviation during testing might be attributed to the variation of uncertainty and features extracted across individual sequences. Uncertainty introduced by higher frequency information in our images can lead to more hysteresis.[12] Since feature extraction across each image sequence is subject to distinct positional and structural inconsistencies, the stability of attention across space and/or time may be broken, leading to decreased hysteresis. To verify that the discrepancies in hysteresis observed in some sequences are a result of these factors, future data analysis is required. One possible approach is to compare our models to spiking neural networks and slow dynamic systems, both of which are known to accurately model perceptual hysteresis.[3]

Another limitation shown in both ResNet50-LSTM architectures is the elevated loss during initial classifications. This is likely attributable to the use of zero padding for the initial boundary condition, a problem accentuated by our short sequence lengths.[15] With zero padding, the LSTM hidden states lack sufficient temporal context, resulting in poor feature integration and increased error for initial classifications. A po-

tential solution involves repetition padding, which requires the model to process the initial frame twice to better initialize the hidden state parameters and stabilize early predictions.

The effects of a regularization hyperparameter that controls the strength of the low-confidence penalty are an additional research direction that requires more in-depth analysis.

These findings contribute to the broader effort in understanding and replicating human-like behaviors. For example, a CNN-LSTM model proposed by Qin *et al*. successfully modeled the hysteresis behavior of congested traffic flow.[16] Similarly, Wang *et al*. demonstrated the improved performance of LSTMs compared to basic neural networks for modeling mechanical hysteretic behavior in tendon-actuated continuum robots.[17] These studies, along with our findings, suggest that LSTMs are well-suited for modeling history-dependent behaviors characteristic of human hysteresis.

One particular example where our study can provide direct assistance is sorting image data sequences for disease screening in a clinical radiology setting. Radiologists typically review several images per day for screening, and the perceptual hysteresis, if it occurs, may affect their screening performance. The proposed network model can go through the entire image set repeatedly in search of a sequence that does not incur perceptual hysteresis.

■ Conclusion

Our findings show that human perceptual hysteresis can be modeled using a CNN-RNN. With the proposed regularization term that penalizes low-confidence decisions, our CNN-RNN model can effectively replicate human perceptual hysteresis across varying strengths of the effect. Although the accuracy of the hysteresis effect needs to be further evaluated and initial frame errors detracts its relevancy, it is still expected to be useful when implementing AI-assisted applications that address human perceptual hysteresis.

■ Acknowledgments

I would like to express my gratitude to Drs. Seungryong Cho and Sungho Yun at Korea Advanced Institute of Science and Technology (KAIST) for guiding me and providing ideas during the early research stage.

■ References

1. van Rooij, M. M. J. W.; Atmanspacher, H.; Kornmeier, J. *Hysteresis in Processing of Perceptual Ambiguity on Three Different Timescales; Proc. Annu. Meet. Cogn. Sci. Soc*. **2016**, *38*. https://escholarship.org/uc/item/62s8w33t.
2. Bonaiuto, J. J.; Berker, A.; Bestmann, S. Response Repetition Biases in Human Perceptual Decisions Are Explained by Activity Decay in Competitive Attractor Models. *eLife* **2016**, *5*, e20047. https://doi.org/10.7554/eLife.20047.
3. You, H.; Meng, Y.; Huan, D.; Wang, D.-H. The Neural Dynamics for Hysteresis in Visual Perception. *Neurocomputing* **2011**, *74* (17), 3502–3508. https://doi.org/10.1016/j.neucom.2011.06.004.
4. Meilikhov, E. Z.; Farzetdinova, R. M. Bistable Perception of Am-biguous Images: Simple Arrhenius Model. *Cogn. Neurodyn.* **2019**, *13* (6), 613–621. https://doi.org/10.1007/s11571-019-09554-9.
5. Fisher, G. H. *Percept. Psychophys*. **1967**, *2* (9), 421–422.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*. **2016**, 770–778. https://doi.org/10.1109/CVPR.2016.90.
7. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput*. **1997**, *9* (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
8. Hemmat, A.; Davies, A.; Lamb, T. A.; Yuan, J.; Torr, P.; Khakzar, A.; Pinto, F. IllusionBench dataset. *Hugging Face Datasets*, arshiahem-mat/IllusionBench; CC BYNC 4.0; **2024**; https://huggingface.co/datasets/arshiahemmat/IllusionBench.
9. Grandvalet, Y.; Bengio, Y. Semi-Supervised Learning by En-tropy Minimization. *Adv. Neural Inf. Process. Syst.* **2005**, *17*, 529–536. https://proceedings.neurips.cc/paper_files/paper/2004/file/96f2b50b5d3613adf90c27049b2a888c7-Paper.pdf.
10. Sayal, A.; Sousa, T.; Duarte, J. V.; Costa, G. N.; Martins, R.; Cas-telo-Branco, M. Identification of Competing Neural Mechanisms Underlying Positive and Negative Perceptual Hysteresis in the Human Visual System. *NeuroImage* **2020**, *221*, 117153. https://doi. org/10.1016/j.neuroimage.2020.117153.
11. Schwiedrzik, C. M.; Ruff, C. C.; Lazar, A.; Leitner, F. C.; Singer, W.; Melloni, L. Untangling Perceptual Memory: Hysteresis and Adaptation Map into Separate Cortical Networks. *Cereb. Cortex* **2014**, *24* (5), 1152–1164. https://doi.org/10.1093/cercor/bhs396.
12. Pascucci, D.; Tanrikulu, Ö. D.; Ozkirli, A.; Houborg, C.; Ceylan, G.; Zerr, P.; Rafiei, M.; Kristjánsson, Á. Serial Dependence in Visual Perception: A Review. *J. Vision* **2023**, *23* (1), 9.
13. Kutschireiter, A.; Basnak, M. A.; Wilson, R. I.; Drugowitsch, J. Bayesian Inference in Ring Attractor Networks. *Proc. Natl. Acad. Sci. U.S.A.* **2023**, *120* (9), e2210622120.
14. Seeholzer, A.; Deger, M.; Gerstner, W. Stability of Working Memory in Continuous Attractor Networks under the Control of Short-Term Plasticity. *PLoS Comput. Biol.* **2019**, *15* (4), e1006928.
15. Zimmermann, H.-G.; Tietz, C.; Grothmann, R. Forecasting with Recurrent Neural Networks: 12 Tricks. In *Neural Networks*; **2012**.
16. Qin, P.; Li, H.; Li, Z.; Guan, W.; He, Y. A CNN-LSTM Car-Following Model Considering Generalization Ability. *Sensors* **2023**, *23* (2), 660. https://doi.org/10.3390/s23020660.
17. Wang, Y.; McCandless, M.; Donder, A.; Pittiglio, G.; Moradkhani, B.; Chitalia, Y.; Dupont, P. E. Using Neural Networks to Model Hysteretic Kinematics in TendonActuated Continuum Robots. *arXiv* **2024**, *2404.07168*. https://doi.org/10.48550/arXiv.2404.07168.

■ Authors

Jason Hwang is a dedicated researcher with a strong interest in applications of AI to real-world problems. His academic aspirations include pursuing majors in Electrical and Computer Engineering (ECE) and Neuroscience. He is currently seeking admission to institutions with rigorous programs in AI and ECE.

DOI: 10.36838/IJHSR86.69