Decoding movement direction from cortical microelectrode recordings using an LSTM-based neural network

Brian Premchand[1], Kyaw Kyar Toe[1], Chuanchu Wang[1], Shoeb Shaikh[1], Camilo Libedinsky[2], Kai Keng Ang[1], and Rosa Q. So[1]

1. Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore
2. Singapore Institute for Neurotechnology, National University of Singapore (NUS), Singapore

***Abstract*— Brain-machine interfaces (BMIs) allow individuals to communicate with computers using neural signals, and Kalman Filter (KF) are prevailingly used to decode movement directions from these neural signals. In this paper, we implemented a multi-layer long short-term memory (LSTM)-based artificial neural network (ANN) for decoding BMI neural signals. We collected motor cortical neural signals from a non-human primate (NHP), implanted with microelectrode array (MEA) while performing a directional joystick task. Next, we compared the LSTM model in decoding the joystick trajectories from the neural signals against the prevailing KF model. The results showed that the LSTM model yielded significantly improved decoding accuracy measured by mean correlation coefficient (0.84, $p < 10^{-7}$) than the KF model (0.72). In addition, using a principal component analysis (PCA)-based dimensionality reduction technique yielded slightly deteriorated accuracies for both the LSTM (0.80) and KF (0.70) models, but greatly reduced the computational complexity. The results showed that the LSTM decoding model holds promise to improve decoding in BMIs for paralyzed humans.**

Keywords—brain-computer interface, BCI, brain-machine interface, BMI, asynchronous decoding, recurrent neural network, LSTM, principal component analysis, PCA

I. Introduction

Recordings from multi-electrode arrays (MEAs) implanted in the motor cortex have been used to drive brain-machine interfaces (BMIs) in many human and non-human primate (NHP) experiments [1]–[4]. Such BMIs have potential benefits to humans with tetraplegia, who would benefit from communication methods that do not require physical movement, and prefer BMIs with a high degree of accuracy [5], [6].

Kalman filters (KF) and variants thereof are frequently used to decode trajectories from neural data [7]–[10]. However, they suffer from several limitations. KFs are essentially linear in nature, while variants of KFs such as the extended KF and unscented KF can only attempt to approximate non-linear functions [11]. On the other hand, brain activity in the motor cortex during task execution is observed to be highly non-linear in nature [12]. In addition, KFs function best with real-time feedback (e.g. when controlling land/air/space vehicles), but this is not possible when a tetraplegic patient with no actual motor output uses a BMI for communication. Thus, we implemented a decoding method based on a multi-layer artificial neural network (ANN) with a long short-term memory (LSTM) layer [13]. ANNs with non-linear activation functions are inherently able to perform non-linear computations, and generally do not depend on real-time feedback, so we hypothesized that an LSTM model would outperform a KF model. Also, ANN-based models including single-cell LSTMs and neuromorphic decoders have successfully been used to decode directional movements from motor cortical signals [14], [15].

In addition, for BMIs to be implemented in human patients, computational complexity is a potential issue. Paralyzed patients prefer the convenience of wireless and compact BMIs [5], but more complex machine learning algorithms tend to consume more power, take more time to train and decode, and/or generate more heat [16]. To reduce the complexity of our decoding models, we used principal component analysis (PCA), a widely used dimension reduction technique that has been successfully applied to LSTM inputs [17], [18]. We tested how well the aforementioned LSTM and KF models performed when using PCA-reduced input data for training and testing models, in lieu of the full dataset.

II. Methods

A. Recording from motor cortex during a joystick task

The experimental procedures described in this paper involving an animal model were approved by, and conducted in compliance with the standards of, the Agri-Food and Veterinary Authority of Singapore and the Singapore Health Services Institutional Animal Care and Use Committee. A electrode array was implanted in the motor cortex of an adult male macaque (*Macaca fascicularis*), and recordings performed as previously described [19], [20].

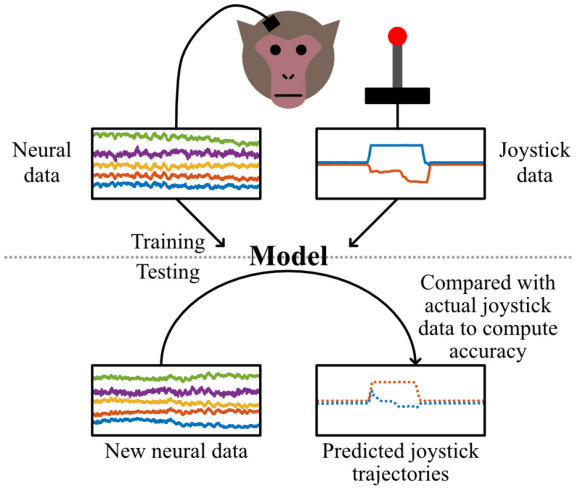

Figure 1 – Overview of model training and testing

During recording sessions, the NHP held a joystick with its left forelimb while it was head-fixed in front of a screen. At the start of each trial, an audio cue was delivered. A cursor appeared in the center of the screen, and a visual cue appeared at 1 of 8 points positioned radially around the cursor. The X and Y coordinates of the joystick were translated into X and Y cursor velocity on the screen. The NHP was trained to the joystick to move the cursor towards the target, with the task succeeding when the cursor reached the target. A reward of fruit juice was delivered upon the successful completion of each trial. Neural activity was continuously sampled at 30 kHz on 100 channels throughout the recording session. For this paper, we used data recorded from three training days within one week. For each day, there were 7-8 recording blocks, with each block containing 16-40 trials. Unsuccessful trials were excluded, then successful trials were stacked end-to-end and treated as a continuous time series for model training and testing.

B. Decoding movement using machine learning models

Our models and analyses were coded by the authors, and run on MATLAB R2018b with the Deep Learning Toolbox Version 12.0.

Inactive channels were first excluded from the dataset, to prevent them from contaminating subsequent calculations [8]. The common average referencing (CAR) technique was applied to reduce cross-channel noise [21]. Signals from each channel were band-pass filtered between 300 to 3000 Hz, then action potentials (i.e. neural spikes) detected using a threshold-based algorithm as previously described [19]. Neural activity in each channel was quantified by measuring the number of spikes in sliding 500 ms windows, updated every 100 ms. Joystick data, in terms of X and Y vectors, was also resampled to temporally align with the neural signals. Spike data for each channel was then normalized (i.e. converted to standard score) before subsequent calculations.

For the KF model, a KF was trained to derive joystick direction from neural activity as previously described [22]. For the LSTM model, the architecture is described below.

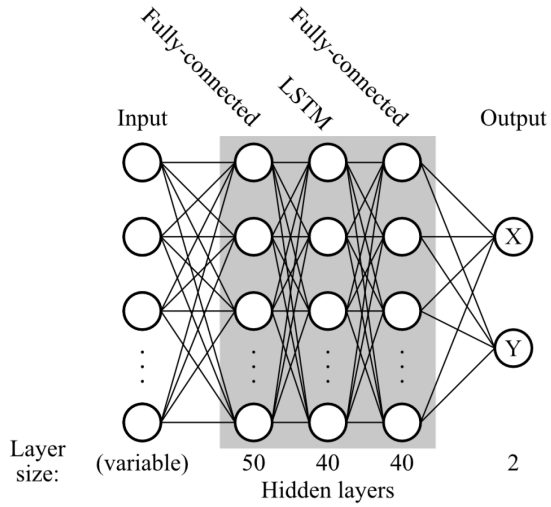

Figure 2 – Architecture of the LSTM-based neural network model

The LSTM model used 3 hidden layers, with two fully-connected layers and an LSTM layer, followed by a 2-unit regression layer representing the joystick X and Y values. Every hidden layer was followed by a leaky rectified linear unit layer, and every layer except the output layer was followed by a dropout layer with 0.25 probability, to help reduce overfitting [23]. The Adam optimizer was used for training the model [24], with number of epochs was set to 200, learning rate set at 0.01, and L2 regularization (also to reduce overfitting) set at 0.001.

To quantify model accuracy, the correlation coefficient between actual joystick position and decoded joystick position was calculated separately for the x- and y-axes. The mean of these two numbers was taken as the final measure of accuracy. Leave-one-out cross-validation was performed on each block of trials, so each block (23 in total) was tested exactly once. Because motor cortical signals are highly non-stationary across different days [25], [26], models were trained and tested only on blocks recorded on the same day. During testing, the neural data was also normalized, with the mean and standard deviation of each channel assumed to be the same as in the training dataset.

C. PCA-based dimensionality reduction

For the PCA-reduced training and testing sets, PCA was performed on the training set, and the top 20 principal components used in lieu of actual data. For testing, the loadings matrix from the training dataset was applied to the testing dataset, and again, the top 20 principal components used in lieu of actual data.

III. Results

A. Joystick trajectories decoded from KF and LSTM models

We found that the decoded X and Y joystick positions (i.e. cursor velocity) matched actual positions far more closely when using the LSTM model than when using the KF model.

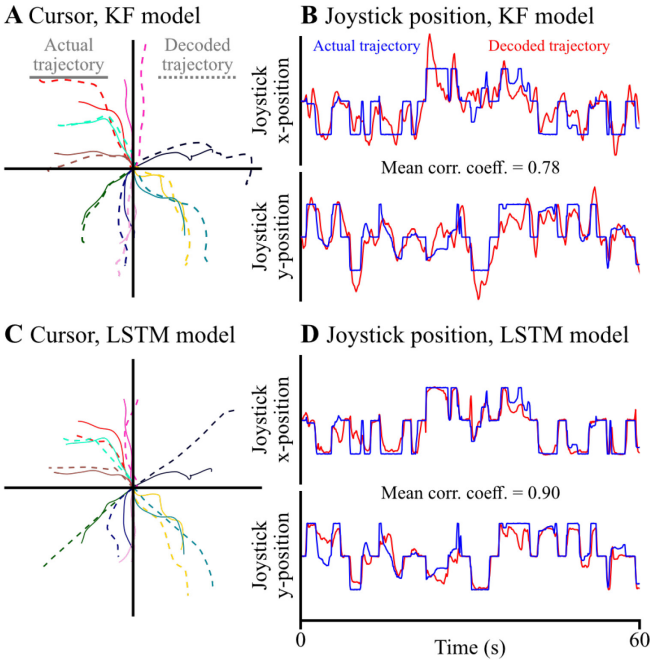

Figure 3 – **A)** Decoding joystick movements using the KF model. On the left are the recorded cursor trajectories of 10 successful actions from a block of trials (solid lines), overlaid with the decoded cursor trajectories of the same actions (dashed lines). **B**) Time series of recorded (blue) and decoded (red) joystick positions on the x- and y-axes, over 60s. **C**) and **D**) Neural decoding of the same 10 actions as in **A** and **B**, but using the LSTM model used instead of the KF model.

In the graphs illustrated above, the decoded joystick positions produced by the LSTM model tended to closely follow actual joystick positions, while the KF model tended to deal less effectively with abrupt changes in joystick position, frequently overshooting and undershooting the actual joystick position.

B. Variance explained by principal components

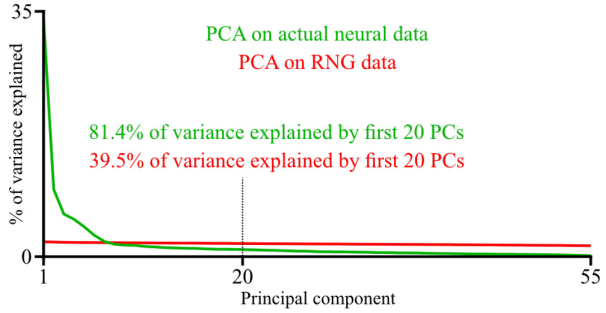

Figure 4 – Percentage of variance explained by each principal component, in a representative set of 8 trial blocks with 55 active channels. The green line illustrates how most of the variance was captured in the first few principal components. The red line shows an identical analysis on normalized output from a random number generator, i.e. how it would look like if the input data was completely random.

PCA tended to capture most of the variance in the data within the first few principal components. We found that on average, the top 20 principal components explained 84 ± 4 % (mean ± standard deviation in this document) of the variance present in model input.

C. Cross-validation of model accuracy

Next, we compared how accurate the KF and LSTM models were, as well as their performance after reducing the dimensionality of their inputs using PCA.

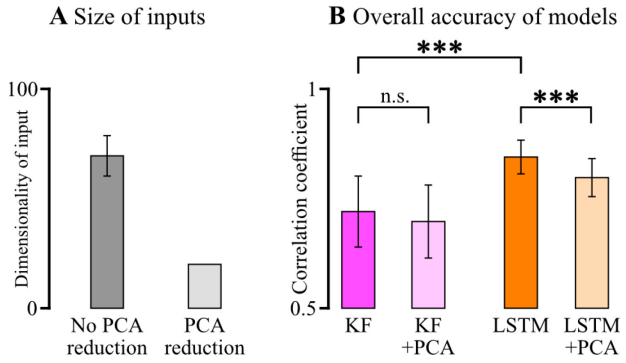

Figure 5 – **A)** The full models had 69 ± 9 input dimensions on average, whereas the PCA-reduced models all had only 20 input dimensions. **B)** Accuracy of decoding across all training blocks using the KF model, the LSTM model, and their PCA-reduced variants. The LSTM model (0.84 ± 0.04) performed significantly better than the KF model (0.72 ± 0.08) (two-tailed two-sample *t*-test for all tests in this figure, $p < 10^{-7}$). If PCA reduction was used, the LSTM model slightly but significantly decreased in performance to 0.80 ± 0.04 ($p < 10^{-3}$). The PCA-reduced KF model (0.70 ± 0.08) did not perform significantly differently when compared with the KF model ($p = 0.35$).

Across the 23 blocks of trials, our LSTM model greatly outperformed the KF model, and produced predictions of joystick position with consistently higher accuracy.

The number of channels used for the full models varied between 55 and 78, as a different number of inactive channels were detected during different recording sessions. After PCA reduction, only the top 20 principal components were used for model training and testing.

PCA reduction allowed us to cut the dimensionality of the model input by more than two thirds, while retaining most of the variance in the not significantly changing the performance of the KF model, and producing only modest decreases in performance for the LSTM model.

IV. Discussion

We have described a multi-layer LSTM-model that outperforms the more commonly used KF, when decoding neural spike data from an NHP using a joystick. Similar models may ultimately be used in BMIs designed for tetraplegic human patients, helping them to communicate and thereby improving their quality living.

There are many possible reasons why a multi-layer LSTM-based model would outperform a KF-based model. One would be that a multi-layer ANN is able to sequentially extract features from the underlying data, while the basic KF only performs 1 step of computations during decoding. ANNs are also able to perform non-linear computations, while the KF is limited to linear transforms. Perhaps most importantly, neural activity in the brain exhibits hysteresis, that is, the current state is dependent on past states [27]. KFs only "remember" the previous state, while LSTMs were specifically designed to extract information from past states. It is likely that this is the reason why the KF model tended to "overshoot" and "undershoot" abrupt changes in joystick position (Figure 3), while the LSTM model tracked changes more closely.

One drawback of using ANNs such as the LSTM model is the high degree of computational complexity involved during model training. Training a KF model took 1.6 s (all times stated here are estimates using MATLAB's inbuilt tic/toc functions), while for the same training set, the LSTM model took 81.5 s. This is due to the iterative nature of gradient descent when training neural networks; the LSTM model was configured to train itself on the input data 200 times, while the KF model was trained once. Nonetheless, both the KF and LSTM model took 0.2 s to decode the same input data; ANNs are not particularly slow during the decoding step.

The potential problems caused by high computational complexity inspired us to look into dimensionality reduction. We hypothesized that PCA would be an appropriate

technique, as signals from electrodes implanted in the motor cortex (and many regions of the brain) tend to be correlated, with different electrodes often producing similar signals [28]. Therefore, it is not entirely surprising that when transformed by PCA-based dimensionality reduction, the principal components were highly skewed, with most of the variance captured in the first few principal components. As far as the authors are aware, PCA-reduction is not frequently used in BMIs. Nonetheless, it appears to be a functional dimensionality reduction method, and will be useful in cases where it is important to reduce the size of model inputs, such as when using wireless BMIs where bandwidth is limited.

The results of this study also hint at other possible approaches for analyzing and studying BMIs. Our multi-layer LSTM model performed better than a single-layer LSTM model (results not included here). This suggests that the additional layers were able to sequentially extract features from the inputs, and associate them with the outputs. What are these features, exactly, and what is each layer of the ANN doing? The problem of explaining and interpreting ANN models is a pressing issue in machine learning today [29]. The application of tools such as visualizers and explainers may help us to understand the fine details of how our LSTM model works. A greater understanding of how ANN-based BMIs work may not only aid the rational design of more effective BMIs, but may also elucidate how biological neural networks in the brain encode, process, and produce motor signals.

REFERENCES

[1] D. M. Brandman *et al.*, "Rapid calibration of an intracortical brain–computer interface for people with tetraplegia," *J. Neural Eng.*, vol. 15, no. 2, p. 026007, 2018, doi: 10.1088/1741-2552/aa9ee7.

[2] R. So *et al.*, "Neural representations of movement intentions during brain-controlled self-motion," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2015, pp. 228–231, doi: 10.1109/NER.2015.7146601.

[3] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct Cortical Control of 3D Neuroprosthetic Devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, Jun. 2002, doi: 10.1126/science.1070291.

[4] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, Jun. 2008, doi: 10.1038/nature06996.

[5] C. H. Blabe, V. Gilja, C. A. Chestek, K. V. Shenoy, K. D. Anderson, and J. M. Henderson, "Assessment of brain-machine interfaces from the perspective of people with paralysis," *J Neural Eng*, vol. 12, no. 4, p. 043002, Aug. 2015, doi: 10.1088/1741-2560/12/4/043002.

[6] J. E. Huggins, A. A. Moinuddin, A. E. Chiodo, and P. A. Wren, "What would brain-computer interface users want: opinions and priorities of potential users with spinal cord injury," *Arch Phys Med Rehabil*, vol. 96, no. 3 Suppl, pp. S38-45.e1–5, Mar. 2015, doi: 10.1016/j.apmr.2014.05.028.

[7] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. L. Nicolelis, "Unscented Kalman Filter for Brain-Machine Interfaces," *PLOS ONE*, vol. 4, no. 7, p. e6243, Jul. 2009, doi: 10.1371/journal.pone.0006243.

[8] B. Premchand, K. K. Toe, C. C. Wang, C. Libedinsky, K. K. Ang, and R. Q. So, "Rapid Detection of Inactive Channels during Multi-unit Intracranial Recordings," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, Chicago, IL, USA, 2019, pp. 1–4, doi: 10.1109/BHI.2019.8834654.

[9] A. K. Vaskov *et al.*, "Cortical Decoding of Individual Finger Group Motions Using ReFIT Kalman Filter," *Front Neurosci*, vol. 12, p. 751, 2018, doi: 10.3389/fnins.2018.00751.

[10] W. Wu and N. G. Hatsopoulos, "Real-time decoding of nonstationary neural activity in motor cortex," *IEEE Trans Neural Syst Rehabil Eng*, vol. 16, no. 3, pp. 213–222, Jun. 2008, doi: 10.1109/TNSRE.2008.922679.

[11] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory Applications*, vol. 4, no. 8, pp. 1303–1318, Aug. 2010, doi: 10.1049/iet-cta.2009.0032.

[12] V. A. Maksimenko *et al.*, "Nonlinear analysis of brain activity, associated with motor action and motor imaginary in untrained subjects," *Nonlinear Dyn*, vol. 91, no. 4, pp. 2803–2817, Mar. 2018, doi: 10.1007/s11071-018-4047-y.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[14] T. Hosman *et al.*, "BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, San Francisco, CA, USA, 2019, pp. 1066–1071, doi: 10.1109/NER.2019.8717140.

[15] S. Shaikh, R. So, T. Sibindi, C. Libedinsky, and A. Basu, "Towards Intelligent Intra-cortical BMI (i2BMI): Low-power Neuromorphic Decoders that outperform Kalman Filters," Bioengineering, preprint, Sep. 2019.

[16] K. Zotos, A. Litke, E. Chatzigeorgiou, S. Nikolaidis, and G. Stephanides, "Energy complexity of software in embedded systems," *ACIT - Automation, Control, and Applications*, 2005.

[17] T. Gao and Y. Chai, "Improving Stock Closing Price Prediction Using Recurrent Neural Network and Technical Indicators," *Neural Comput*, vol. 30, no. 10, pp. 2833–2854, 2018, doi: 10.1162/neco_a_01124.

[18] L. H. Nguyen and S. Holmes, "Ten quick tips for effective dimensionality reduction," *PLOS Computational Biology*, vol. 15, no. 6, p. e1006907, Jun. 2019, doi: 10.1371/journal.pcbi.1006907.

[19] C. Libedinsky *et al.*, "Independent Mobility Achieved through a Wireless Brain-Machine Interface," *PLoS One*, vol. 11, no. 11, Nov. 2016, doi: 10.1371/journal.pone.0165773.

[20] H. Yang, C. Libedinsky, C. Guan, K. K. Ang, and R. Q. So, "Boosting performance in brain-machine interface by classifier-level fusion based on accumulative training models from multi-day data," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 1922–1925, doi: 10.1109/EMBC.2017.8037224.

[21] K. A. Ludwig, R. M. Miriani, N. B. Langhals, M. D. Joseph, D. J. Anderson, and D. R. Kipke, "Using a Common Average Reference to Improve Cortical Neuron Recordings From Microelectrode Arrays," *J Neurophysiol*, vol. 101, no. 3, pp. 1679–1689, Mar. 2009, doi: 10.1152/jn.90989.2008.

[22] V. Gilja *et al.*, "A High-Performance Neural Prosthesis Enabled by Control Algorithm Design," *Nat Neurosci*, vol. 15, no. 12, pp. 1752–1757, Dec. 2012, doi: 10.1038/nn.3265.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580 [cs]*, Jul. 2012.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014.

[25] B. Jarosiewicz *et al.*, "Retrospectively supervised click decoder calibration for self-calibrating point-and-click brain-computer interfaces," *J Physiol Paris*, vol. 110, no. 4 Pt A, pp. 382–391, Nov. 2016, doi: 10.1016/j.jphysparis.2017.03.001.

[26] S. Shaikh, R. So, T. Sibindi, C. Libedinsky, and A. Basu, "Sparse Ensemble Machine Learning to Improve Robustness of Long-term Decoding in iBMIs," *IEEE Trans Neural Syst Rehabil Eng*, Dec. 2019, doi: 10.1109/TNSRE.2019.2962708.

[27] H. Kim, J.-Y. Moon, G. A. Mashour, and U. Lee, "Mechanisms of hysteresis in human brain networks during transitions of consciousness and unconsciousness: Theoretical principles and empirical evidence," *PLoS Comput Biol*, vol. 14, no. 8, Aug. 2018, doi: 10.1371/journal.pcbi.1006424.

[28] S. Perel *et al.*, "Single-unit activity, threshold crossings, and local field potentials in motor cortex differentially encode reach kinematics," *J Neurophysiol*, vol. 114, no. 3, pp. 1500–1512, Sep. 2015, doi: 10.1152/jn.00293.2014.

[29] V. Buhrmester, D. Münch, and M. Arens, "Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey," *arXiv:1911.12116 [cs]*, Nov. 2019.