Spatio-temporal convolutional neural networks explain human neural representations of action recognition

Andrea Tacchetti¹, Leyla Isik¹, and Tomaso Poggio

Center for Brains, Minds, and Machines, MIT

Abstract
Recognizing the actions of others from complex visual scenes is an essential task for humans. Here we investigate the computational mechanisms that support action recognition in the human visual system. We use a novel dataset of well-controlled naturalistic videos of five actions performed by five actors at five viewpoint and extend a class of biologically inspired hierarchical computational models of object recognition to recognize actions from videos. We explore a number of variations within the class of convolutional neural networks and assess classification accuracy on a viewpoint invariant action recognition task as well as a correlation measure with magnetoencephalography (MEG) human brain recordings acquired using the same stimuli. We show that feed-forward spatio-temporal convolutional neural networks solve the task of invariant action recognition and account for the majority of the explainable variance in the neural data. Furthermore, we show that model features that improve performance on viewpoint invariant action recognition lead to a model representation that better matches the representation encoded by neural data. These results advance the idea that robustness to complex transformations, such as 3D viewpoint invariance, is a specific goal of visual processing in the human brain.

Introduction
Humans’ ability to recognize actions of others is a crucial aspect of visual perception. We are incredibly skilled at recognizing what other people are doing. The degree of accuracy to which we can finely discern actions of others is largely unaffected by transformations that, while substantially changing the visual appearance of a given scene, do not change the semantics of what we observe. Here we investigate the computational and algorithmic level aspects of computing a representation of complex

¹ These authors contributed equally to this work
visual input that supports our ability to recognize actions robustly to complex transformations like 3D rotation.

A number of computer vision approaches have been proposed to define and study the recognition of actions; borrowing from the established taxonomy [1], we mean by action the middle-ground between action primitives (e.g. raise the left foot and move it forward) and activities (e.g. playing basketball). Actions are possibly repeated sequences of primitives like walking or running.

Recent work comparing neural data and computational models, mainly convolutional neural networks, using representational similarity analysis (RSA) [2] has provided computational accounts of the visual representations underlying invariant object recognition. This line work has revealed that optimizing the performance of convolutional neural networks on simple discrimination tasks (e.g. object recognition) results in models that produce representations that match neural data in humans and monkeys [3]–[5]. Here we compare neural representations, measured with MEG recordings from [6], and computational models to gain insight on how we recognize actions from videos.

We put two main hypotheses to the test, the first is that the algorithmic level aspects of the neural computations underlying human action recognition can be modeled by feed-forward neural networks that are convolutional in space and time [7], [8]. The second hypothesis is that building representations that are robust to complex transformations is the computational level goal that of networks [9], [10].

Results

**Novel invariant action recognition dataset**

To study the effect of changes in view and actor on action recognition, we filmed a dataset of five actors performing five different actions (drink, eat, jump, run and walk) on a treadmill from five different views (0, 45, 90, 135, and 180 degrees from the front of the actor/treadmill; the treadmill rather than the camera was rotated in place to acquire from different viewpoints) [Figure 1]. The dataset was filmed on a fixed, constant background. To avoid low-level object/action confounds (e.g. the action “drink” being classified as the only videos with water bottle in the scene) and guarantee that the main sources of variation of visual appearance are due to actions, actors and viewpoint, the actors held the same objects (an apple and a water bottle) in each video, regardless of the action
they performed. This controlled design allows us to test hypotheses on the computational mechanisms underlying invariant recognition in the human visual system without having to settle for a synthetic dataset. Each action-actor-view combination was filmed for at least 52-seconds. The videos were cut into two-second clips that each included at least one cycle of each action, and started at random points in the cycle (for example, a jump may start mid air or on the ground). The dataset includes 26 two-second clips for each actor, action, and view, for a total of 3250 video clips. This dataset allows testing of actor and view invariant action recognition, with few low-level confounds.

**Figure 1: Novel action recognition stimulus set.** The stimulus set consists of five actors, performing five actions (drink, eat, jump, run and walk), at a fixed position in the visual field (while on a treadmill) and on a fixed background across five different views (0, 45, 90, 135, and 180 degrees). To avoid low-level confounds, the actors held the same objects in each hand (a water bottle and an apple), regardless of the action performed.

**Recognizing actions with a biologically-inspired hierarchical model**

In order to test the hypothesis that Spatio-temporal Convolutional Neural Networks provide an algorithmic explanation of how the brain quickly computes a representation for actions we implement variants of these models and train them to recognize action from videos. We use two different training regimes, with and without back-propagation,
to investigate the individual roles of architecture and templates in these models. These systems are extensions of classical models of visual cortex, convolutional neural networks, which have successfully explained object recognition from static images [11]–[13], to stimuli that extend in time. Spatio-temporal convolutional neural networks are hierarchical: an input video goes through a layer of computation and the output of this layer serves as input to the next layer, the sequence of layers is inspired by Hubel and Wiesel's findings in primary visual cortex, and is constructed by alternating layers of simple cells, which perform template matching or convolution, and complex cells, which perform max pooling [14]. Qualitatively, these models work by detecting the presence (or lack thereof) of a certain video segment (a template) in the input stimulus. The exact position in space and time of the detected template is discarded by the pooling mechanism and only the information about its presence is passed on to the next layer². The specific models that we present here consist of two simple-complex layer pairs denoted S1, C1, S2 and C2 [Figure 3a]. The two models differ in the way that their layer weights are learned and help us investigate the individual roles of network architecture and template weights.

² Models templates extend both in space and time in these models so a scrambled stimulus would elicit a substantially different response from a temporally coherent one as the template matching stage would be disrupted.
Figure 2: Computational model for action recognition – fixed templates. An input video is convolved with moving Gabor-like templates at the S1 layer. At the C1 layer, a local max pooling is applied across position and time. At the S2 layer, previous layer’s outputs are convolved with templates sampled from a sample set of videos disjoint from the test set. Videos in the sample set go through the S1 and C1 layers before being used for sampling. At the final layer a global max across positions and views is computed.

![Computational model for action recognition – fixed templates](image)

Figure 3: Pooling Strategies. Invariance is obtained by enforcing structure in the wiring between simple and complex cells at the S2-C2 pooling stage. C2 units pool over all S2 units whose templates come from videos containing a particular actor performing a particular action across different views. We compare this experimental model [top] to an unstructured control model [bottom], which contains the same S2 templates, but where each C2 cell pools over a random, unstructured set of S2 cell templates.

The role of network architectures in action recognition

The first model class is designed to assess the quality of different neural architectures with a fixed set of biologically inspired templates. These models have hard-coded S1 templates (moving Gabor-like stimuli, with both a spatial and temporal component, that model the receptive fields found in primate V1 and MT [15–17]) and S2 templates that
are sampled from a dataset denoted template-set (see methods section). In order to produce a response that is invariant to rotation in depth, the model’s top complex cell units (C2) pool over all templates containing patches of videos of a single actor performing a specific action recorded at different viewpoints. This “pooling across channels” mechanism detects the presence of a certain template (e.g. the torso of someone running) regardless of its 3D pose to produce a pose invariant signature. Both theoretical insights and experimental evidence have suggested how this wiring across views might be learned in development [18]–[21]. We compare this structured model to a traditional convolutional model that does not pool over channels, and to an unstructured control model, which contains the same templates, but where action is not taken into account in the pooling scheme and instead each C2 cell pools over a random, unstructured set of S2 cell templates [Figure 3b].

The role of template weights in action recognition
The second model class is similar in its basic structure to the model described above but its S1 and S2 templates are learned from the template dataset (see methods section) using back-propagation [22] (unlike the first model which has fixed templates). We can use the comparison of the two models to test the effect of learning templates. Again we compare different pooling architectures: specifically, a model that pools over C2 channels, to traditional convolutional neural networks with no additional pooling.

Model performance on invariant action recognition task
We test the performance of each model by training and testing a machine learning classifier to recognize actions based on the model response to entire videos. The machine learning classifier is trained on responses to videos at one of two viewpoints, 0 or 90 degrees, and tested on responses to videos either at the same viewpoint (within view) or at the opposite one (across view), this experimental design lets us test the model’s ability to produce a signature that is useful to recognize actions and generalize across views. We find that top layer responses from all models can discriminate actions equally well when a classifier is trained and tested on model responses to videos at the same viewpoint. Learning templates and pooling over C2 channels both improve invariant recognition performance. Importantly, a motion energy model (C1 layer of the model described below) cannot distinguish action invariant to view, showing that these
results are not due to low level dataset confounds. [Figure 4]. These results first show that models with templates learned via back-propagation perform best on our invariant action discrimination task. The second, insight is that the simple pooling across channels described above improves invariant action decoding performance. Particularly in the first model class, pooling across view works significantly better than just pooling across random templates. This result suggests that non-affine transformations can be dealt with in the same manner as scale and position are with convolution and pooling.

**Figure 4: Performance on simple and invariant action recognition.** Mean accuracy of a classifier trained to recognize actions based on models output. The classifier is trained and tested on the same view (‘within-view’ condition), or trained on one view (0 degrees or 90 degrees) and tested on second view (‘across view’ condition). HMAX-
models employ fixed templates and help us compare different architectures. The Structured model employs structured pooling over channels as described in Figure 3b, top, and the Unstructured model employs random pooling over channels as described in Figure 3b, bottom. HMAX-Convolutional models do not feature pooling over channel. We test a convolutional model that has the same number of S2 templates (many S2) as its non-convolutional counterparts and a convolutional model that has the same number of C2 units as its non-convolutional counterpart (Few S2). Backprop- models have S1 and S2 templates learned with back-propagation. Again we report model architectures with and without pooling across channels (Channel Pooling) and compare it to convolutional models with same number of S2 units (Many S2) or the same number of C2 units (Few S2) as their non-convolutional counterpart. Error bars indicated standard error across model runs [see supplementary information]. Horizontal line indicates chance performance (20%). Brackets indicate a statistically significant difference with p<0.05.

Comparing MEG data and model representations

We quantitatively measure how well the pattern of neural responses are matched by the computational models we introduced above by using representational similarity analysis [2], [23] to compare the video representation encoded in MEG brain recordings and the model responses. We compare each of the models described above (see also methods section for details) to MEG from subjects viewing the same action recognition dataset [6]. We again investigate the role of model architecture and templates individually by comparing models with different architectures and fixed or learned templates.

Eight subjects viewed two views (0 and 90 degrees) from the dataset described above and were instructed to recognize which of the five actions was performed in each video while their neural activity was recorded with a MEG scanner. We average raw MEG data from each sensor across a 10ms window sliding by 5ms and across stimulus repetitions within each of the eight subjects. For each subject, video and time-point the MEG representation is encoded in a 306 dimensional vector with each entry being associated with a single sensor. We only consider recordings at the peak action recognition decoding time reported in the original paper (420ms after stimulus onset). See [24] [6] for more details.

We compute representational similarity matrices for each model at the C1 and C2 layers and compare it with the similarity matrix computed from MEG recordings. We observe that the C2 units of the highest performing model with pooling over channels and templates learned through back-propagation produce a representation that matches that read out from the MEG recordings significantly better than any other model.
Furthermore, among the models with templates that are not learned through backpropagation, the model featuring structured pooling across 3D viewpoints matches the neural data significantly better than its unstructured and convolutional counterparts [Figure 4B]. Overall, traits that lead to better performance, especially in the viewpoint invariant action recognition task – learning templates from data, pooling across channels – also lead to a better match to neural data despite the fact that these architectures and template changes were not designed to match the MEG data, only to optimize performance on the invariant action recognition task.

![Figure 5: MDS analysis. We report here the representational similarity between the MEG recordings and each stage of the models described in Figure 4. The representational similarity is measured at the peak of MEG decoding accuracy for the](image-url)
action recognition task. The bottom black line indicates the chance level as measured by a permutation test (worst case across conditions and time). The top black line indicates the noise ceiling measured as the best agreement between two human subjects and their mean (best case across subjects). Brackets indicate a statistically significant difference with $p<0.05$.

Discussion
This work furthers our understanding of how our brain recognizes the actions of others in complex visual scenes in three aspects. First we show that convolutional neural networks can recognize actions invariant to actor and view (non-affine transformations) and offer a compelling algorithmic level model of the neural mechanisms underlying fast and invariant action representations in visual cortex. Our computational models, particularly those that pool over top layer channels to build view-point invariance, produce robust representations for actions that closely match those encoded in human neural data. The results presented here show that simple-complex cell architectures [9], are sufficient to explain fast invariant action recognition across video stimuli with complex transformations. This suggests that no special neural circuitry is required to achieve robustness to transformations that go beyond affine. Furthermore our results indicate that a discriminative feedback signal is sufficient to learn the appropriate templates and that learning a full generative model is unnecessary for recognition. A competing hypothesis to the last point is that humans learn complete generative model of 3D object to be able to solve recognition tasks robustly [25].

Second, we show that architecture and template modifications that improve model performance on the invariant action recognition task, specifically, learning templates with back-propagation and pooling over channels to build viewpoint invariance, also improve how well the model's representation matches neural data. This suggests that robustness to complex transformations is one of the computational goals of visual cortex. This finding is in agreement with what has been shown for objects in static images [2], [4], [23], our extension of these methods to videos allows us to investigate the role of invariance to non-affine transformation in human action recognition.

Lastly, we present a novel, well-controlled dataset to study the transformations of actor and viewpoint, in an isolated and parameterized manner [26] without settling for computer generated avatars.
Conclusions
This work shows that, at the algorithmic level, a simple spatio-temporal convolutional neural network can support neural representations for actor- and view-invariant action recognition. Model features that improve classification accuracy for invariant action recognition produce representations that better match those encoded in MEG brain recordings suggesting that computing representations for actions that are invariant to complex transformations such as viewpoint and actor is a specific computational goal of visual cortex. Our work expands methods to match and compare neural data and computational models representations to visual stimuli that extend in time, and shows that for actions of others, like for inanimate objects, computational models that perform better at a recognition task more closely match neural data. This line of work has shown that with extremely weak computational constraints and by optimizing simple task oriented metrics one can obtain accurate models of visual cortex. Our findings strengthen this claim by broadening the scope of these methods to include stimuli that extend in time, human MEG data and by specifically analyzing the separate roles of templates and network architecture.

Acknowledgements
This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. We thank the McGovern Institute for Brain Research at MIT for supporting this research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. We would like to thank Patrick Winston, Gabriel Kreiman, Martin Giese, Charles Jennings, Heuihan Jhuang, and Cheson Tan for their feedback on this work.

Materials and methods
Action recognition dataset
We filmed a dataset of five actors performing five actions (run, walk, jump, eat and drink) from five views (0, 45, 90, 135, and 180 degrees from the front) on a treadmill in front of a fixed background. By using a treadmill we avoided having actors move in and out of frame during the video. To avoid low-level object confounds, the actors held a water
bottle and an apple in each hand, regardless of the action they performed. Each action was filmed for 52 seconds, and then cut into 26 two-second clips at 30 fps.

Models with fixed templates
Models are composed of 4 layers. The input video is scaled down, preserving the aspect ratio, to 128x76px. Models were implemented with CNS: Cortical Network Simulator, a GPU library to declare Convolutional Architectures [13]. For this class of models, a total of three scaled replicas of each video are run through the model in parallel; the scaling is by a factor of 1/2. The first layer is composed of a grid of simple cells placed 1px apart (no sub-sampling), the templates for these units are Gabor receptive fields that move in space while they change phase (as described in previous studies on the receptive fields of V1 and MT cells [15], [17], [27]). Cells have spatially square receptive field of size 7, 9 and 11px, extend for 3, 4 and 5 frames and compute the dot product between the input and their template. The Gabor filters in each receptive field move exclusively in the direction orthogonal to the spatial modulation at 3 speeds, linearly distributed between 4/3 and 4 pixels per frame. The second layer (C1) is a grid of complex cells that compute the maximum of their afferent simple cells. Cells are placed 2 units apart in both spatial dimensions (spatial subsampling by a factor of 2) and every unit in the time dimension (no time subsampling). Complex cells at the C1 level have spatial receptive fields of 4 simple cells and span 2 scales with one scale overlap, bringing the number of scaled replicas in the model from 3 to 2. The third layer (S2) is composed of a grid of simple cells that compute the dot product between their input and a stored template. The templates at this level are sampled randomly from a sample dataset that has no overlap with the test set; we sample 512 different templates, uniformly distributed across classes and across videos within each class. The cells span 9, 17 and 25 units in space and 3, 7 and 11 units in time. The fourth layer, C2, is composed of complex units that compute the maximum of their inputs; C2 cells pool across all positions and scales. The wiring between simple and complex cells at the C2 layer is described by an adjacency matrix with each column corresponding to a complex cell; each column is then a list of indices for the simple cells that the complex cells pools over. In the convolutional model, each template is treated individually and only pooling across space and time is implemented at the C2 layers. In the remaining two models, where pooling over channel is implemented we have two cases. In the structured pooling model each column of the
adjacency matrix, indexes cells with templates samples from videos featuring a single actor performing a single action. In scrambled pooling model, the rows of this matrix are scrambled and therefore the columns (i.e. indices of simple cells pooled together by a single complex cell) have no semantic common thread. S2 template sizes are always pooled independently from one another. The output of the C2 layers is concatenated over time and cells and serves as input to a supervised machine learning classifier.

Models with templates learned through back-propagation
The second class of models was written using the Torch nn and cunn packages and their Volumetric modules. The models are identical to the ones described above except for three differences: templates are learned through back-propagation, there is no scale replication of input nor templates. Using shorthand notation the architecture is as follows: S1: Volumetric Convolution(kernel: 9x9x9, stride: 2x2x2), C1: Volumetric Pooling (kernel: 4x4x1, stride: 2x2x1), S2: Volumetric Convolution(kernel: 17x17x3, stride 1x1x2) and C2: Volumetric Pooling (kernel: 13x13x1, stride 1x1x1). We use an Adam solver [28] to perform back-propagation and interleave each layer in the model with a Volumetric Batch Normalization layer [29] which we observe dramatically improves the learning rate in the first few iterations. We use mini-batches of 10 videos to estimate the gradient. The networks that pool over channels while learning templates are implemented with a standard Max-Out network layer [30].

Video pre-processing and classification
We used non-causal temporal median filtering background subtraction for all videos [31]. All classification experiments for the model were carried out using the Gaussian Kernel Regularized Least Squares classification pipeline available in the GURLS package [32]. Both the kernel bandwidth and the regularization parameter were chosen using leave-one-out cross validation.

Action recognition experiments
Model experiments are divided in three steps: sampling or learning templates from a template set in order to populate the model’s units, computing the model’s response to a set of training and test videos and lastly training and testing a classifier on these
responses to report its accuracy. For each experiment we make sure that the test set has no overlap with neither the template nor the training sets.

When the templates are learned from data, we allow the back-propagation algorithm to proceed for 10 full passes over the template set. In these cases, the loss that the network minimizes is a Negative Log-likelihood classification loss. Before starting the optimization on the sample set we endow the network with a linear classifier for the target class “action” composed of two hidden fully connected layers with 256 and 128 hidden units respectively and a logarithmic soft-max on the output of the last layer. These layers were then removed at the end of training and only the model output was considered.

When the templates were sampled from the template set, the sampling probability was uniform across classes and uniform across videos and space-time position within each class.

For all models the template set contained 2600 videos of four out of the five actors performing all five actions and at all five views. The training set for this experiment was composed of 600 videos of four of the five actors performing all five actions at either the frontal or side viewpoints. The test set was composed of 150 videos of the fifth actor performing all five actions at either the frontal or side viewpoint. We only used either one of the viewpoints to train or test so as to verify the ability of the model to recognize actions within the same view and to generalize across views. This sample/train/test split was repeated five times, using each actor for testing once and re-sampling the S2 templates each time.

Representational Similarities analysis

Representational Similarity Analysis for representational similarity was introduced in [2] as a tool to measure the degree of agreement between two representations of the same data that are not directly comparable. We applied the method as described in the original paper to our data. We computed a pairwise dissimilarity matrix for the model representation (see below for conditions) of each of 50 unique videos. We generated the pairwise dissimilarity matrix induced by the raw sensor outputs (averaged over 10 showings, within subject), for each time point and each of eight different subjects and then averaged across the eight subjects. For each time point we bootstrapped 50 times a subsample of 30 rows and columns from the dissimilarity matrices of both MEG and
model. We then computed the Spearman correlation coefficient between the lower triangular portion of this matrix and reported this number as the measure of agreement.

We performed the same analysis but scrambled the rows and columns of the model dissimilarity matrix 100 times and recorded the best agreement and used it as reference for chance.

Finally we used the same technique to assess the degree of agreement across human subjects. We used the method described above to compare each individual subject to the subjects’ mean dissimilarity matrix.
References

[1] T. B. Moeslund and E. Granum, “A Survey of Computer Vision-Based Human Motion Capture,” *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231–268, 2001.

[2] N. Kriegeskorte, M. Mur, and P. Bandettini, “Representational similarity analysis – connecting the branches of systems neuroscience,” 2008.

[3] N. Kriegeskorte, M. Mur, D. A. Ruff, R. Kiani, J. Bodurka, H. Esteky, K. Tanaka, and P. A. Bandettini, “Matching Categorical Object Representations in Inferior Temporal Cortex of Man and Monkey,” *Neuron*, vol. 60, no. 6, pp. 1126–1141, 2008.

[4] D. L. K. Yamins and J. J. DiCarlo, “Using goal-driven deep learning models to understand sensory cortex,” *Nat. Neurosci.*, vol. 19, no. 3, pp. 356–365, 2016.

[5] K. N. Kay, T. Naselaris, R. J. Prenger, and J. L. Gallant, “Identifying natural images from human brain activity.,” *Nature*, vol. 452, no. 7185, pp. 352–5, 2008.

[6] L. Isik, A. Tacchetti, and T. Poggio, “Fast , invariant representation for human action in the visual system,” *arXiv Prepr. arXiv*, 2016.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[8] H. Jhuang, “A biologically inspired system for action recognition,” no. 7879304, p. 58, 2008.

[9] F. Anselmi, J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio, “Unsupervised learning of invariant representations,” *Theor. Comput. Sci.*, no. 0, p. -., 2015.

[10] J. Z. Leibo, J. Mutch, and T. Poggio, “How can cells in the anterior medial face patch be viewpoint invariant?,” *MIT-CSAIL-TR-2010-057, CBCL-293; Present. COSYNE 2011, Salt Lake City*, 2011.

[11] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[12] T. Serre, A. Oliva, and T. Poggio, “A feedforward architecture accounts for rapid categorization,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 15, pp. 6424–6429, 2007.

[13] J. Mutch, U. Knoblich, and T. Poggio, “CNS: a GPU-based framework for simulating cortically-organized networks,” *MIT-CSAIL-TR*, vol. 2010–013, no. 286, 2010.

[14] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *J. Physiol.*, 1962.

[15] E. H. Adelson and J. R. Bergen, “Spatiotemporal energy models for the perception of motion,” *J. Opt. Soc. Am. Image Sci. Vis.*, vol. 2, no. 2, pp. 284–299, 1985.

[16] N. C. Rust, O. Schwartz, J. A. Movshon, and E. P. Simoncelli, “Spatiotemporal
elements of macaque v1 receptive fields,” *Neuron*, vol. 46, no. 6, pp. 945–956, 2005.

[17] N. C. Rust, V. Mante, E. P. Simoncelli, and J. A. Movshon, “How MT cells analyze the motion of visual patterns.,” *Nat. Neurosci.*, vol. 9, pp. 1421–1431, 2006.

[18] P. Földiák, “Learning Invariance from Transformation Sequences,” *Neural Comput.*, vol. 3, pp. 194–200, 1991.

[19] L. Wiskott and T. J. Sejnowski, “Slow feature analysis: unsupervised learning of invariances.,” *Neural Comput.*, vol. 14, no. 4, pp. 715–770, 2002.

[20] S. M. Stringer, G. Perry, E. T. Rolls, and J. H. Proske, “Learning invariant object recognition in the visual system with continuous transformations,” *Biol. Cybern.*, vol. 94, no. 2, pp. 128–142, 2006.

[21] G. Wallis and H. H. Bulthoff, “Effects of temporal association on recognition memory.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 98, no. 8, pp. 4800–4804, 2001.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, no. 4. pp. 541–551, 1989.

[23] S. M. Khaligh-Razavi and N. Kriegeskorte, “Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation,” *PLoS Comput. Biol.*, vol. 10, no. 11, 2014.

[24] L. Isik, E. M. Meyers, J. Z. Leibo, and T. Poggio, “The dynamics of invariant object recognition in the human visual system.,” *J. Neurophysiol.*, vol. 111, no. 1, pp. 91–102, 2014.

[25] I. Yildirim, T. Kulkarni, W. Freiwald, and J. Tenenbaum, “Efficient analysis-by-synthesis in vision: A computational framework, behavioral tests, and comparison with neural representations,” *37th Annu. Conf. Cogn. Sci. Soc.*, no. April, pp. 1–5, 2015.

[26] A. Tacchetti and L. Isik, “Invariant action recognition dataset,” 2016. [Online]. Available: http://web.mit.edu/atacchet/www/dataset.

[27] E. P. Simoncelli and D. J. Heeger, “A model of neuronal responses in visual area MT.,” *Vision Res.*, vol. 38, pp. 743–761, 1998.

[28] D. P. Kingma and J. L. Ba, “Adam: a Method for Stochastic Optimization,” *Int. Conf. Learn. Represent.*, pp. 1–13, 2015.

[29] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *arXiv:1502.03167*, pp. 1–11, 2015.

[30] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout Networks,” *arXiv Prepr.*, pp. 1319–1327, 2013.

[31] M. Piccardi, “Background subtraction techniques: a review,” *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, vol. 4, pp. 3099–3104, 2004.

[32] A. Tacchetti, P. K. Mallapragada, L. Rosasco, and M. Santoro, “Gurls: A least squares library for supervised learning,” *J. Mach. Learn. Res.*, vol. 14, pp. 3201–3205, 2013.