VISUAL HYPERACUITY WITH MOVING SENSOR AND RECURRENT NEURAL COMPUTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamical phenomena, such as recurrent neuronal activity and perpetual motion of the eye, are typically overlooked in models of bottom-up visual perception. Recent experiments suggest that a tiny inter-saccadic eye motion ("fixational drift") enhances visual acuity beyond the limit imposed by the density of retinal photoreceptors. Here we hypothesize that such an enhancement is enabled by recurrent neuronal computations in early visual areas. Specifically, we explore a setting involving a low-resolution dynamical sensor that moves with respect to a static scene, with drift-like tiny steps. This setting mimics a dynamical eye, viewing objects in perceptually-challenging conditions. The dynamical sensory input is classified by a convolutional neural network with recurrent connectivity added to its lower layers, in analogy to recurrent connectivity in early visual areas. Applying our system to CiFAR-10 and CiFAR-100 datasets down-sampled via 8x8 sensor, we found that (i) classification accuracy, which is drastically reduced by this down-sampling, is mostly restored to its 32x32 baseline level when using a moving sensor and recurrent connectivity, (ii) in this setting, neurons in the early layers exhibit a wide repertoire of selectivity patterns, spanning the spatio-temporal selectivity space, with neurons preferring different combinations of spatial and temporal patterning, and (iii) curved sensor's trajectories improve visual acuity compared to straight trajectories, echoing recent experimental findings involving eye-tracking in challenging conditions. Our work sheds light on the possible role of recurrent connectivity in early vision as well as the roles of fixational drift and temporal-frequency selective cells in the visual system. It also proposes a solution for artificial image recognition in settings with limited resolution and multiple time samples, such as in edge AI applications.

1 INTRODUCTION

Biological vision is known to be a dynamical process. Two factors contributing to these dynamics are eye motion and recurrent neuronal connections in the brain. Our eyes move constantly with movements that, kinematically, can be divided into saccades - quick gaze shifts, and drifts - small scanning movements between saccades (often referred to as "fixational drift") (Rucci et al., 2018). These dynamical aspects of vision are reflected only partially in contemporary computer vision systems. Some works addressed large scale shifts in visual attention resembling saccades (Mnih et al., 2014). Others explored properties and benefits of recurrent top down connections (Nayebi et al., 2018), reminiscent of top-down processing in biological vision (Hochstein & Ahissar, 2002). Notably, the dynamics of low-level visual processes, occurring early in the bottom-up visual hierarchy and sensitive to the fixational drift (Snodderly et al., 2001; Ölveczky et al., 2003; Malevich et al., 2020; Hohl & Lisberger, 2011), remains largely overlooked in models of vision as well as in bio-inspired computer vision systems.

In fact, since the seminal studies by Hubel & Wiesel (1962), selectivity in primary visual cortex has been traditionally described in terms of static spatial filters (e.g., simple and complex spatial fields or Gabors of varying frequency and orientation). In convolutional neural networks (CNNs) (Krizhevsky et al., 2012), which have dominated computer vision over the last decade, features resembling the spatial filters deduced from biological studies emerge spontaneously over the course of training (Zeiler & Fergus, 2014; Lindsey et al., 2019). In some cases, remarkable correlations

were found between spatial neural representations in CNNs and those identified in the biological brain (Yamins & DiCarlo, 2016).

On the other hand, temporal dynamics, and sensitivity to temporal features, characterize visual neurons throughout the visual system, from retinal receptors and ganglion cells to thalamic and cortical neurons (Berry et al., 1997; Chichilnisky, 2001; Lee et al., 1981; Levick et al., 1972; Reinagel & Reid, 2000; Shimaoka et al., 2018). Existing evidence suggests that both eye motion (Snodderly et al., 2001; Ahissar & Arieli, 2001; Ölveczky et al., 2003; Malevich et al., 2020; Gruber et al., 2021; Hohl & Lisberger, 2011) and recurrent neuronal connectivity (Bejjanki et al., 2011; Samonds et al., 2013; Kar et al., 2019; Kubilius et al., 2019; Kietzmann et al., 2019) contribute to this temporal dynamics.

One niche where spatio-temporal computation is probably necessary is the perception of tiny objects. It is well known that the acuity of biological vision is not limited by the spatial resolution of retinal photoreceptors ("visual hyperacuity"; Westheimer (2009); Barlow (1979)). Vernier acuity, for example, is dramatically higher than might be expected from pure spatial acuity derived from the photoreceptor density in the retinal mosaic (Westheimer, 2009). Whether hyperacuity is obtained via spatial, temporal or spatio-temporal mechanisms is not yet known (Rucci et al., 2018). In any case, it is evident that the visual processing allowing hyperacuity, or perception of any tiny stimulus, should cope with the fixational drift; if it doesn't, the drift, whose amplitude is at least two orders of magnitude larger than the smallest perceivable spatial offsets, would impair acuity (Ahissar & Arieli, 2001; Rucci et al., 2018; Ratnam et al., 2017). The same drift motion could potentially improve acuity if spatio-temporal computations are employed. Such computations can be based on the emphasis of high-frequency spatial details (Rucci et al., 2007), temporal coding of spatial offsets (Ahissar & Arieli, 2001; 2012), Bayesian inference (Anderson et al., 2020), or on any other derivative of the interactions between ocular motion and the external image. Furthermore, it is reasonable to attribute such spatio-temporal computations to early visual areas which are known to exhibit faster dynamics and shorter integration windows compared to regions upstream in the visual processing chain (Gauthier et al., 2012). Indeed, it had been shown that the recurrent neuronal circuitry in early visual areas could enable countering the blurring from retinal motion (Burak et al., 2010).

Using the information available from over-sampling low-resolution images has an extensive history in computer vision as part of the field of super-resolution (Milanfar, 2017). Multi-image super-resolution (MISR) (Farsiu et al., 2004), distinguished from single-image super-resolution (Glasner et al., 2009; Dong et al., 2015), aims to reconstruct high-resolution images from a set of low-resolution ones (Arefin et al., 2020; Ge et al., 2018; Bhat et al., 2021; Li et al., 2017). An adjacent field of research, low-resolution object recognition, investigates algorithms to maximize the performance on a given task (Xi et al., 2020; Ge et al., 2018). Both fields use low-resolution images as input but differ in the goal of the training and evaluation.

In this paper we introduce a classifier that exploits spatio-temporal computations in early layers to perceive tiny images. More specifically, we trained a convolutional neural network with recurrent connectivity (Arefin et al., 2020) introduced to early layers. The network receives a sequence of low-resolution images generated via sensor motion mimicking ocular drift. We used high-resolution images to obtain a set of features that was then used to facilitate learning in a teacher-student framework (Hinton et al., 2015). The outcome is a dynamical classifier that suffers from only a small drop in accuracy when tasked with a significant decrease in spatial resolution, a decrease which substantially impairs the accuracy of a comparable static feed-forward classifier.

Using a novel generative model, we found that our dynamical classifier developed features that were sensitive mostly to spatial changes, others that were sensitive mostly to temporal changes, and a majority that exhibited sensitivity to mixed spatio-temporal patterns.

Finally, when examining the correlations between patterns of motion and accuracy of classification, we observed that curved trajectories are favorable for recognition, which is consistent with recent findings about the curvature of fixational drift trajectories in humans. (Intoy & Rucci, 2020; Gruber & Ahissar, 2020).

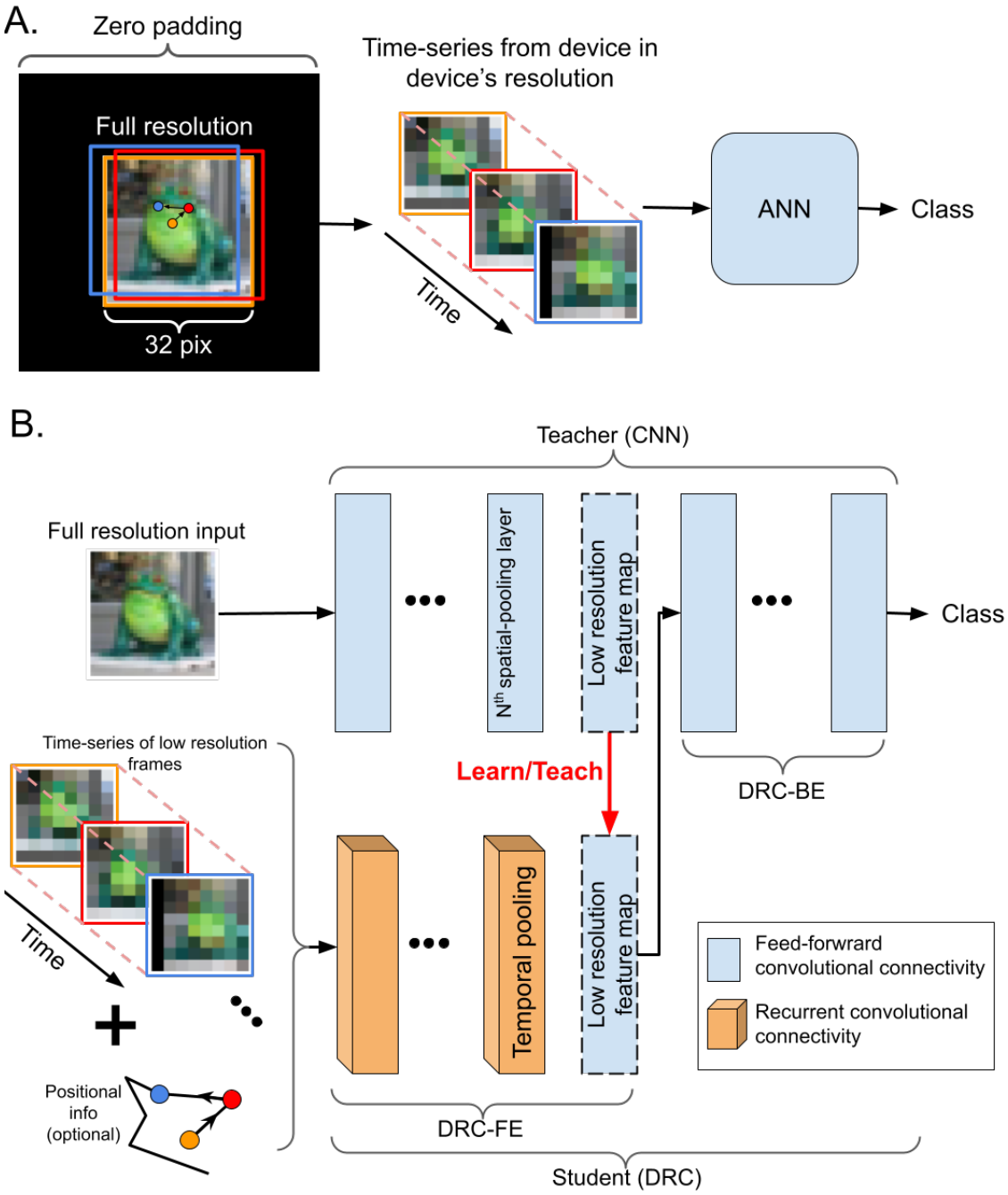


Figure 1: **Description of the the task and the system** - A. A time-series of low resolution images, simulating a sequence of frames generated by a sensor moving in a static scene, is fed to the network one-by-one following the order of acquisition, along with optional position information; the network integrates the information from the whole sequence of images and outputs a class. A trajectory composed of 3 steps (orange $\rightarrow$ red $\rightarrow$ blue) is illustrated on the full resolution image together with the corresponding 3 generated frames. B. Network architecture and training procedure: *Teacher* is a feed-forward convolutional neural network, e.g. ResNet50 or Small-net. *Student* is a multilayer recurrent convolutional neural network. At the first phase of training Student's bottom layers (DRC-FE) are trained to reproduce teacher's features. These features are extracted from the teacher at the point where the teacher's spatial resolution corresponds to the student's input resolution. The top shared layers (DRC-BE) are then fine-tuned to improve accuracy of the low resolution task. See main text for more details.

2 RESULTS

2.1 TASK AND MODELS

To create a synthetic setting reminiscent of ocular drift, we used images from popular CiFAR datasets (Krizhevsky et al., 2009), embedded in a large (200x200 pixel) *scene* padded by zeros. *Sensor* position was defined in units of pixels on the scene and its motion was modeled by a stochastic process that is discussed below. The sensor's frames were obtained by cropping a 32x32 pixels window from the scene, around the sensor position. Resolution was then reduced to 8x8 using a standard OpenCV (Bradski, 2000) function (that does not include anti-aliasing filter) with bi-cubic interpolation (Fig. 1A).

A ResNet50 (He et al., 2016) network pre-trained on Imagenet (Deng et al., 2009), which is available as a part of Keras package (Chollet, 2015), was used as a model of reference. The model was fine-tuned to one of the CiFAR datasets, reaching accuracy of 96.83 and 82.94 percent for CiFAR-10 and CiFAR 100, respectively (Table 1, Standard resolution (32x32)). In order to feed the 32x32 pix CiFAR images to the network, images were upsampled by factor of 7 to (32x7)x(32x7) pix (i.e. 224x224 pix which are the dimensions of ImageNet images).

In order to verify the generality of our conclusions we tested another more compact variant of reference CNN with 3M parameter model. This smaller network that we refer to as Small-net (Table S7,S8) receives 32x32 pix images as input, therefore, eliminating the need to upsample the CiFAR images before feeding them into the network. This smaller network also simplified the analysis of internal representations as explained below.

2.1.1 TRAINING

We applied a feature learning paradigm (Hinton et al., 2015). while using our reference network as teachers for the dynamical recurrent classifier (DRC) student.

Typical CNNs perform a series of spatial pooling operations. Max pooling layers in the reference CNNs effectively reduce spatial resolution while preserving relevant information about the underlying scene. To develop our DRC, we exploited this spatial pooling line-up. We thus took instances of trained CNNs and replaced their bottom layers with recurrent convolutional networks (Fig. 1B). Specifically, we used a stack of ConvGRU layers (Ballas et al., 2015; Van Valen et al., 2016) without spatial poolings to replace the original network all the way from the input to the point where the CNN's spatial resolution is reduced by the desired factor (Table S3). In our case, resolution was decreased by factor of 4, therefore the appropriate resolution was achieved after the second max pooling layer. At this point, considering the 2 DRC systems we have tested, the resolution of the ResNet50-based teacher is (8x7)x(8x7) (the 'x7' factor is due to the upsampling of the original 32x32 images by factor of 7 to (32x7)x(32x7)); while the Small-net-based teacher resolution is 8x8. We refer to the bottom recurrent part of the DRC as DRC-front-end (DRC-FE). For the rest of the processing stack we reuse the reference (teacher) network architecture (either ResNet50 or Small-net). We refer to this reused part of the DRC as DRC- back-end (DRC-BE) (Fig. 1B).

We trained the DRC in two steps - first the DRC-FE was trained to reproduce features of the teacher network. Here we used mean-squared loss between the teacher network and the DRC-FE (other optimization goals, such as mean absolute loss and cosine similarity resulted in very similar performance and are not shown). Positional data were concatenated with the images time series; see B.4 for further clarification. Next, the DRC-BE was fine-tuned using cross-entropy loss.

Our model was mostly implemented in the Keras package (Chollet, 2015), with the convolutional GRU layer adapted from the project of (Van Valen et al., 2016) [1].

2.2 PERFORMANCE

2.2.1 BASELINE

In order to evaluate the improvement in performance which can be attributed to the unique architecture of the DRC, we considered a few baseline solutions.

[1] Anonymized code is provided as supplementary material.

Table 1: Test accuracy [%] of various **ResNet50 baselines** and **ResNet50-DRC** in various configurations, compared to baseline results. Results are presented as mean $\pm$ std.

	CiFAR-10	**CiFAR-100**
Standard resolution (32x32)	96.83 ± 0.09	82.94 ± 0.23
Low resolution (8x8) baseline:		
Naive training	78.88 ± 0.54	54.41 ± 0.21
Naive training + teacher	78.54 ± 0.09	54.75 ± 0.14
Averaged prediction, 5 steps	83.86 ± 0.47	60.24 ± 0.22
Averaged prediction, 10 steps	83.87 ± 0.23	60.22 ± 0.25
Averaged prediction, 5 steps + teacher	**84.04** ± 0.15	**61.56** ± 0.23
ResNet+GRU, 5 steps, w/o position input	83.52 ± 0.22	59.32 ± 0.20
ResNet+GRU, 5 steps, with position input	83.94 ± 0.12	59.61 ± 0.59
ResNet+convGRU, 5 steps, w/o position input	83.73 ± 0.10	58.62 ± 0.16
ResNet+convGRU, 5 steps, with position input	83.84 ± 0.09	58.05 ± 0.24
GAN-based (Xi et al., 2020)	**88.1**	–
Low resolution (8x8) DRC:		
DRC 5 steps, *not moving*	75.87 ± 0.29	52.02 ± 0.07
DRC 5 steps, w/o position input	87.83 ± 0.38	67.23 ± 0.30
DRC 10 steps, w/o position input	90.16 ± 0.59	70.38 ± 0.67
DRC 5 steps, with position input	92.26 ± 0.19	74.23 ± 0.11
DRC 5 steps, deeper, with position input	93.45 ± 0.15	76.24 ± 0.21
DRC 10 steps, with position input	**94.77**± 0.09	**78.75**± 0.55

The accuracy of the ResNet50 reference (teacher) network when applied to a single low resolution image, was chosen as a simplistic baseline. The performance of this network, shown in Table 1 ('Naive training'), demonstrates a large degradation of accuracy in both datasets (See also Table S4 for network's architecture). To facilitate fair comparison, we also checked trained such a naive clasifier with feature learning as done in DRC. The results here did not change significantly.

As a more advanced baseline we considered an averaged prediction (AP) of a feed-forward model over the T sampled frames. Namely, the estimated probability $\hat{p}_k$ of a class k is given by $\hat{p}_k = \frac{1}{T}\sum_{t=1}^{T}\hat{p}_k^t$, where $\hat{p}_k^t$ are predictions of the above naive baseline. The situation here is similar to test time data augmentation (Perez & Wang, 2017) with sensor motion being the augmenter. Notably, the AP saturated with the number of timesteps while our full system as described below kept improving (Table 1). Here application of teacher did slightly improve the accuracy and in a case of CiFAR-100 the improvement is significant (1.34% in average).

Next, we evaluated a models where an convGRU (resp. GRU) is connected before (resp. on the top of) the last global average pooling layer of ResNet, we denote it as Resnet+convGRU (resp. ResNet+GRU) (Table S6). At their best, these models achieved accuracy lower by approximately 4% and 7% for CiFAR-10 and CiFAR-100 datasets respectively, compared to 5-step DRC w/o positional information. The fact that these models and the AP achieve approximately equal performance indicates that trainable recurrent connectivity in top layers has little benefit over simplistic integration. This is in contrast to the DRC where recurrent connectivity is implemented in the low layers. This result is not surprising since convolutional networks tend to develop invariance to small shifts (Zeiler & Fergus, 2014), such as those that DRC relies on, albeit with important caveats (Azulay & Weiss, 2018).

Finally, we refer to a recent work (Xi et al., 2020) that uses generative adversarial network to enhance feature representation in CiFAR-10 task with 8x8 resolution. This solution performs slightly better than DRC without positional information and 5 timesteps but underperforms versus the same DRC setting with 10 step. Furthermore, no results for CiFAR-100 are available in this work.

Table 2: Accuracy [%] of Small-net variants with 10 time-steps. Each column corresponds to a single realization. The version, marked by $*$ is used for further representation analysis.

	depth	**position input**	**CiFAR-10**
Standard resolution (32x32)			88.6
Low resolution (8x8) DRC	3	no	84.6*
	3	yes	85.2
	6	no	85.8
	6	yes	87.1

2.2.2 OUR MODEL

Table 1 summarizes performance of DRC on both datasets. The simplest version, with 5 time-steps and with no positional encoding outperforms the baseline solutions, including the GRU based one. It can be clearly seen that adding time steps or increasing network's depth is leveraged to higher accuracy in both datasets. The version with 10 time steps achieves accuracy which is just 2-4% inferior to the full resolution setting.

To check whether DRC advantage vs. baselines relies solely on the recurrent computation in the bottom layers, we tested it without motion. The corresponding result in Table 1 indicates that this is *not* the case.

Table 2 reports three examples of Small-nets trained on CiFAR-10. Here we see that same trends of performance hold for a shallower and more compact network architecture and for teacher trained from scratch, without transfer learning.

2.3 REPRESENTATION

To better understand how our dynamical network extracts high-level features from low resolution images, we analyzed the activation sensitivity of each of the 64 neurons of the final layer of the DRC-FE ("feature-neurons"). We started by using activation maximization with gradient ascent over the input pattern space (Zeiler & Fergus, 2014) and obtained the maximally-activating patterns (AMs) for the static teacher network (Fig. S7). Unfortunately, applying this tool to the spatio-temporal features learned by the student network failed to converge systematically (Fig. S4). We thus developed a deep generator network (DGN) (Table S9), partially inspired by (Nguyen et al., 2016), that proved capable of producing AMs with spatio-temporal patterns, while remaining consistent with the results obtained using gradient ascent in a purely spatial setting (Fig. S7).

For each feature-neuron we found a specific series of images which maximized its activity, by allowing the generator to devise unconstrained spatio-temporal patterns. As previously seen in (Zeiler & Fergus, 2014), we found that Gabor-like images maximized the activation of many feature-neurons of the teacher network, reminiscent of the sensitivity of neurons in the early visual system to similar stimuli (Carandini et al., 2005). As expected, we also found that the Gabor-like patterns in our (dynamic) students' feature-neurons were often reminiscent of those of the (static) teacher. Importantly, however, the features presented in our student network exhibited dynamics with high spatial and temporal variability, resembling visual receptive fields for drifting Gabors (Figures 2A, S8, S9).

To isolate the contribution of spatial and temporal variations to the AM of each feature-neuron we explored two settings of constrained maximization (see Fig. 2B for visualization of both constraints). Specifically, we tasked the generator with creating either purely spatial input patterns, where all frames must be identical (see middle rows in examples at Fig. 2B), or purely temporal patterns in which differences between frames were allowed but all the pixels of each frame were identical (e.g., bottom rows in Fig. 2B). We found features that were more sensitive to temporal dynamics along with others that were more sensitive to spatial dynamics, with the majority of features exhibiting mixed spatio-temporal sensitivity pattern (Fig. 2B).

Interestingly, many features exhibited spatio-temporal AMs that were substantially higher than the corresponding purely spatial and purely temporal AMs, suggesting specific coding benefits for spatio-temporal fields in our dynamic network. This finding illustrates the importance of studying spatio-temporal receptive fields in the visual system (DeAngelis et al., 1993; Rust et al., 2005).

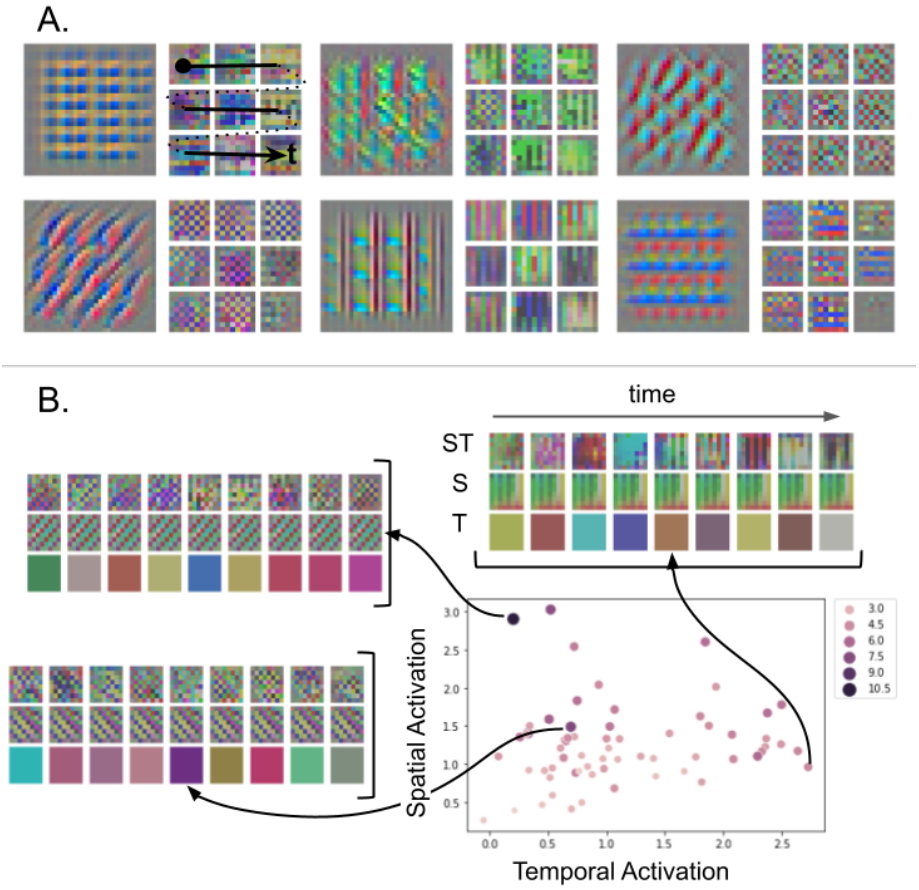


Figure 2: **Spatial and temporal receptive fields in the top recurrent layer** - A. Examples of six pairs of teacher (high-resolution, left) and student (low-resolution, right) feature AM. Note the visual resemblance between the teacher's and student's features and the dynamical nature of the student's features. The arrow illustrates time flow. B. Central plot: the values along the X- (and resp. the Y-)axis represent the activation values in constrained maximization setting when limiting the DGN to purely temporal (resp. spatial) changes. The size and the color of the dots represent the intensity of activity in the full unconstrained (spatio-temporal) maximization. Call-outs depict predominantly temporal (bottom-right dot), predominantly spatial (top-left dot), and mixed spatio-temporal (center-left dot) selectivity. Rows correspond to spatio-temporal (ST), purely spatial (S), and purely temporal (T) maximization. Columns correspond to timesteps where 8 time-steps out of 10 are shown for visual clarity.

2.4 Sensor's trajectory and its effect on performance

While ocular drift is considered a diffusive, stochastic process, recent evidence suggests that its high-level properties are controlled by the brain in stimulus or task-dependent manner. In particular, (Gruber & Ahissar, 2020; Intoy & Rucci, 2020) demonstrated that ocular drifts in human subjects exhibit more curved paths when viewing more informative regions. We thus examined the effect of our sensor trajectory on recognition accuracy. We devised a family of stochastic diffusive trajectories with controllable curvature properties. To do that, we assumed that the random distribution of angles between successive intervals is governed by von Mises distribution (Appendix B for details).

Testing our DRC with varying κ (the parameter which controls the trajectory's mean curvature; see Appendix B for details), we found that the recognition performance improved with curvature, providing a possible functional interpretation for the experimental findings of (Gruber & Ahissar, 2020; Intoy & Rucci, 2020). Figure 3 shows gradual improvement in accuracy on CiFAR-100 dataset

as κ decreases. Representative trajectories for each tested value are shown in the top panel with their corresponding accuracy presented in the bottom panel.

To distinguish between the properties of the actual trajectory with the generated sequence of points as opposed to the collection of sample points, we performed shuffling experiment. Fig. 3 shows that randomly shuffling points on the trajectory mostly affects trajectories with low curvature. For trajectories with negative kappa, shuffled trajectory visually resembles non-shuffled ones and, therefore, it is not of surprise that shuffling has little or no effect. On the other hand, for trajectories with low curvature shuffling does have a significant effect. This implies that while same information is available in both cases, the decoding is compromised in the case of a shuffled path.

To leverage the advantage of the trajectory's curvature further, we devised another family of trajectories for which curvature was explicitly enforced. We refer to these trajectories as "spirals" (Appendix for details). As can be seen in 3 the 'spiral' trajectories resulted with the best performance. The 'spiral' ensemble of trajectories is the one that is reported in Tables 1 and 2.

We used the curvature-index defined in Gruber & Ahissar (2020) (See also Appendix B) in order to compare our synthetic trajectories with real drift trajectories acquired in an experiment (Gruber & Ahissar, 2020). We found that by using κ values of -1.0 and 0.0 the generated distributions have similar characteristics (see details in Appendix B) to the ones observed in the experiment with the relevant condition- recognizing small 2d shapes. The empirical distributions in (Gruber & Ahissar, 2020) were obtained by dividing the data to periods where relevant information is projected on the retina, and periods when such stimulus is not available to the retina (compare Fig. S10 with Figure 3, 'Natural-small' curvature-index distributions in Gruber & Ahissar (2020)). These similarities demonstrate that (a) the range of kappas we are using is biologically relevant, and (b) our model can be used for exploring the mechanistic details underlying the biological control preferring such curvatures, as they demonstrate their advantage in recognition.

To conclude, we find that a curved motion of the sensor is beneficial for our DRC setting, offering a potential functional interpretation to similar kinematics observed in human vision (Gruber & Ahissar, 2020; Intoy & Rucci, 2020).

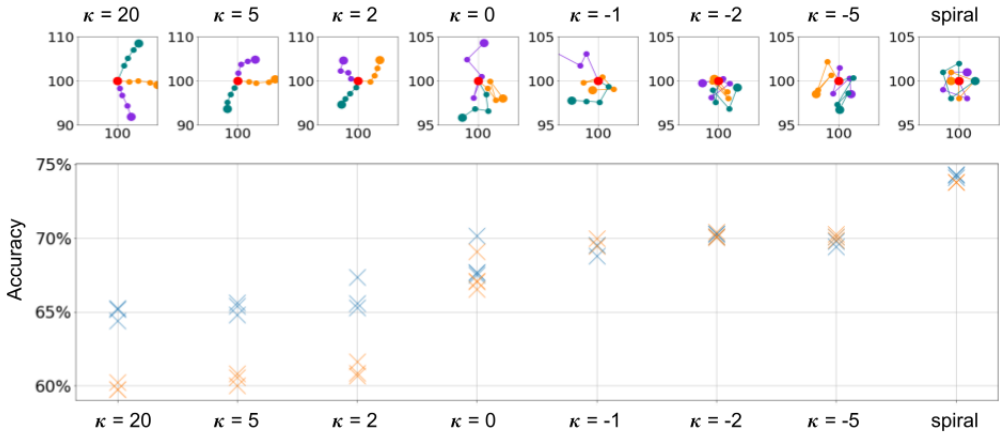


Figure 3: **Sensor's trajectory and its impact on performance.** Top: representative examples of trajectories are shown for 5-time steps long trajectories generated according to dynamics in equation 1 of gradually increasing curvature, which corersponds to decreasing κ. Spiral trajectories governed by equation 2 are also shown. Bottom: accuracy for DRC performing 5 timesteps on CiFAR 100 is plotted for each setting for original (blue) and randomly shuffled (orange) trajectories.

3 Discussion

We introduced a dynamical recurrent classifier (DRC), a system that recruits tiny motions of a sensor to compensate for low spatial resolution with temporal over-sampling. This setting is novel, although

some of its components had been used before (Kanazawa et al., 2021; Arefin et al., 2020). We introduced recurrent dynamics to the low layers of the network (the DRC-FE in Fig. 1B), followed by time averaging and by feed forward convolutional layers. This architecture is reminiscent of the biological brain: the dynamics in early visual areas is faster and the integration windows are shorter than in higher areas. Therefore, an assumption of static (or slowly varying) representation in high areas is reasonable. Furthermore, high visual areas (e.g. V4 and IT) exhibit invariance to variety of stimulus distortions (Cadieu et al., 2007; Rust & DiCarlo, 2010), which is lacking in the low areas (V1, V2). This fact is echoed by our training method which allows the student and the teacher to have different low level representations but request similarity of representations starting at the point along processing hierarchy upstream of which a common architecture is used (DRC-BE in Fig. 1B).

Moving the sensor along the image may be considered as yet another variant of test-time data augmentation. However, we find that the recurrent computation provides an extra benefit compared to the averaged prediction baseline (Table 1) corresponding to such an augmentation.

Importantly, the task of recognition from a series of low-resolution frames differs from the related task of MISR (Farsiu et al., 2004; Arefin et al., 2020). While the latter task requires high-resolution reconstruction of the input scene, the former one does not. The DRC does not need to learn all the particularities needed to reconstruct a high-resolution image; instead, it focuses on extracting the necessary features for the given task. Future work may compare the performance of the DRC with standard classifiers that use task driven super-resolution (Haris et al., 2018) as a preprocessing step. These solutions are applicable in a variety of settings such as body worn cameras, UAVs and self-driving cars with the need of performing real-time image recognition tasks, on a stream of visual input images captured from a moving vision sensor of limited quality, e.g. (Desai et al., 2015; Merenda et al., 2020; Howard et al., 2017; Jiang et al., 2018). Table 1 suggests that in a setting with large number of samples and limited resolution DRC may be competitive e.g. vs. Xi et al. (2020).

Recently, Anderson et al. (2020) performed an approximate Bayesian inference to decode features that could account for the improvement in acuity observed in the experiments of Ratnam et al. (2017). In our approach we assume that the primitives are shaped by the stimuli in full resolution – i.e. in the regime of convenience – rather than handcrafted, and are then adapted in a more challenging regime of low resolution. Furthermore the inference is performed by a trained neuronal agent as opposed to the idealistic Bayesian estimate in (Anderson et al., 2020).

A teacher network assisted our DRC in developing its latent neuronal representation. In biological terms, this would be analogous to hyperacuity being based on representations developed using regular acuity. Consistent with this analogy are the findings that (i) the development of hyperacuity in humans follows the development of regular acuity (tested using Snellen tables) (Skoczenski & Norcia, 2002; Wang et al., 2009) and (ii) recognizing the smallest Snellen optoypes, which improves with age (Wang et al., 2009), likely requires hyperacuity (Ratnam et al., 2017; Intoy & Rucci, 2020). Interestingly, there is evidence that the fixational drift contributes to the perception of the small Snellen optotypes (Ratnam et al., 2017; Intoy & Rucci, 2020).

The trajectory along which samples are taken affects the recognition accuracy, echoing experimental findings (Gruber & Ahissar, 2020; Intoy & Rucci, 2020). Notably, the sensor trajectories in this work were generated independently of the underlying scene. This is a possibly sub-optimal situation, and future work may focus on closed loop interaction between sensor trajectory and the perceived scene (Ahissar & Assa, 2016; Gruber et al., 2021). This could also shed light on the ongoing effort to identify controllable ingredients in the ocular drift motion (Ratnam et al., 2017).

The results of this work can be used when constructing specific hypotheses about the ways in which the visual system copes with tiny images. Specifically, our work suggests that the ocular drift plays a major role in such conditions, supporting earlier hypotheses and empirical findings (Marshall & Talbot, 1942; Steinman & Levinson, 1990; Ahissar & Arieli, 2001; Snodderly et al., 2001; Rucci et al., 2007; Ratnam et al., 2017; Anderson et al., 2020; Gruber & Ahissar, 2020; Gruber et al., 2021; Intoy & Rucci, 2020) and that the processing of the drift-derived spatio-temporal information requires recurrent processing in retinal, sub-cortical or cortical visual networks (Burak et al., 2010). Our results also support the inclusion of eye position signals in such processing (Burak et al., 2010); the accuracy required from such signals likely dictates that they should be derived from retinal signals (e.g., Ahissar et al. (2015)).

REFERENCES

Ehud Ahissar and Amos Arieli. Figuring space by time. *Neuron*, 32(2):185–201, 2001.

Ehud Ahissar and Amos Arieli. Seeing via miniature eye movements: a dynamic hypothesis for vision. *Frontiers in computational neuroscience*, 6:89, 2012.

Ehud Ahissar and Eldad Assa. Perception as a closed-loop convergence process. *Elife*, 5:e12830, 2016.

Ehud Ahissar, Shira Ozana, and Amos Arieli. 1-d vision: Encoding of eye movements by simple receptive fields. *Perception*, 44(8-9):986–994, 2015.

Alexander G Anderson, Kavitha Ratnam, Austin Roorda, and Bruno A Olshausen. High-acuity vision from retinal image motion. *Journal of vision*, 20(7):34–34, 2020.

Md Rifat Arefin, Vincent Michalski, Pierre-Luc St-Charles, Alfredo Kalaitzis, Sookyung Kim, Samira E Kahou, and Yoshua Bengio. Multi-image super-resolution for remote sensing using deep recurrent networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 206–207, 2020.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

H. B. Barlow. Reconstructing the visual image in space and time. *Nature*, 279(5710):189–190, May 1979. ISSN 0028-0836, 1476-4687. doi: 10.1038/279189a0. URL `http://www.nature.com/articles/279189a0`.

Vikranth R Bejjanki, Jeffrey M Beck, Zhong-Lin Lu, and Alexandre Pouget. Perceptual learning as improved probabilistic inference in early sensory areas. *Nature neuroscience*, 14(5):642–648, 2011.

Michael J Berry, David K Warland, and Markus Meister. The structure and precision of retinal spike trains. *Proceedings of the National Academy of Sciences*, 94(10):5411–5416, 1997.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9209–9218, 2021.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Yoram Burak, Uri Rokni, Markus Meister, and Haim Sompolinsky. Bayesian model of dynamic image stabilization in the visual system. *Proceedings of the National Academy of Sciences*, 107 (45):19525–19530, 2010.

Charles Cadieu, Minjoon Kouh, Anitha Pasupathy, Charles E Connor, Maximilian Riesenhuber, and Tomaso Poggio. A model of v4 shape selectivity and invariance. *Journal of neurophysiology*, 98 (3):1733–1750, 2007.

Matteo Carandini, Jonathan B Demb, Valerio Mante, David J Tolhurst, Yang Dan, Bruno A Olshausen, Jack L Gallant, and Nicole C Rust. Do we know what the early visual system does? *Journal of Neuroscience*, 25(46):10577–10597, 2005.

EJ Chichilnisky. A simple white noise analysis of neuronal light responses. *Network: computation in neural systems*, 12(2):199, 2001.

François Chollet. keras. `https://github.com/fchollet/keras`, 2015.

Gregory C DeAngelis, Izumi Ohzawa, and RD Freeman. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. ii. linearity of temporal and spatial summation. *Journal of Neurophysiology*, 69(4):1118–1135, 1993.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Soham Jayesh Desai, Mohammed Shoaib, and Arijit Raychowdhury. An ultra-low power,"always-on" camera front-end for posture detection in body worn cameras using restricted boltzman machines. *IEEE transactions on multi-scale computing systems*, 1(4):187–194, 2015.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307, 2015.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13(10):1327–1344, 2004.

Baptiste Gauthier, Evelyn Eger, Guido Hesselmann, Anne-Lise Giraud, and Andreas Kleinschmidt. Temporal tuning properties along the human ventral visual stream. *Journal of Neuroscience*, 32 (41):14433–14441, 2012.

Shiming Ge, Shengwei Zhao, Chenyu Li, and Jia Li. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing*, 28(4):2051–2062, 2018.

Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pp. 349–356. IEEE, 2009.

Liron Zipora Gruber and Ehud Ahissar. Closed loop motor-sensory dynamics in human vision. *PloS one*, 15(10):e0240660, 2020.

Liron Zipora Gruber, Shimon Ullman, and Ehud Ahissar. Oculo-retinal dynamics can explain the perception of minimal recognizable configurations. *Proceedings of the National Academy of Sciences*, 118(34), 2021.

Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Task-driven super resolution: Object detection in low-resolution images. *arXiv preprint arXiv:1803.11316*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Shaul Hochstein and Merav Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804, 2002.

Sonja S Hohl and Stephen G Lisberger. Representation of perceptually invisible image motion in extrastriate visual area mt of macaque monkeys. *Journal of Neuroscience*, 31(46):16561–16569, 2011.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154.2, January 1962. ISSN 0022-3751. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1359523/`.

Janis Intoy and Michele Rucci. Finely tuned eye movements enhance visual acuity. *Nature Communications*, 11(1), February 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-14616-2. URL `https://www.nature.com/articles/s41467-020-14616-2`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pp. 253–266, 2018.

Hiroki Kanazawa, Yuta Nakamoto, Jiaxin Zhou, and Takashi Komuro. Human detection from low-resolution video images using 3d convolutional neural network. In *Fifteenth International Conference on Quality Control by Artificial Vision*, volume 11794, pp. 117941G. International Society for Optics and Photonics, 2021.

Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B Issa, and James J DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature neuroscience*, 22(6):974–983, 2019.

Tim C Kietzmann, Courtney J Spoerer, Lynn KA Sörensen, Radoslaw M Cichy, Olaf Hauk, and Nikolaus Kriegeskorte. Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 116(43):21854–21863, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-like object recognition with high-performing shallow recurrent anns. *arXiv preprint arXiv:1909.06161*, 2019.

BB Lee, A Elepfandt, and V Virsu. Phase of responses to moving sinusoidal gratings in cells of cat retina and lateral geniculate nucleus. *Journal of Neurophysiology*, 45(5):807–817, 1981.

WR Levick, BG Cleland, MW Dubin, et al. Lateral geniculate neurons of cat: retinal inputs and physiology. *Invest Ophthalmol*, 11(5):302–311, 1972.

Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1222–1230, 2017.

Jack Lindsey, Samuel A Ocko, Surya Ganguli, and Stephane Deny. A unified theory of early visual representations from retina to cortex through anatomically constrained deep cnns. *arXiv preprint arXiv:1901.00945*, 2019.

Tatiana Malevich, Antimo Buonocore, and Ziad M Hafed. Rapid stimulus-driven modulation of slow ocular position drifts. *Elife*, 9:e57595, 2020.

WH Marshall and SA Talbot. Recent evidence for neural mechanisms in vision leading to a general theory of sensory acuity. 1942.

Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533, 2020.

Peyman Milanfar. *Super-resolution imaging*. CRC press, 2017.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014.

Aran Nayebi, Daniel Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel LK Yamins. Task-driven convolutional recurrent models of the visual system. *arXiv preprint arXiv:1807.00053*, 2018.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29:3387–3395, 2016.

Bence P Ölveczky, Stephen A Baccus, and Markus Meister. Segregation of object and background motion in the retina. *Nature*, 423(6938):401–408, 2003.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Kavitha Ratnam, Niklas Domdei, Wolf M. Harmening, and Austin Roorda. Benefits of retinal image motion at the limits of spatial vision. *Journal of Vision*, 17(1):30, January 2017. ISSN 1534-7362. doi: 10.1167/17.1.30. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5283083/.

Pamela Reinagel and R Clay Reid. Temporal coding of visual information in the thalamus. *Journal of neuroscience*, 20(14):5392–5400, 2000.

Michele Rucci, Ramon Iovin, Martina Poletti, and Fabrizio Santini. Miniature eye movements enhance fine spatial detail. *Nature*, 447(7146):852–855, 2007.

Michele Rucci, Ehud Ahissar, and David Burr. Temporal coding of visual space. *Trends in cognitive sciences*, 22(10):883–895, 2018.

Nicole C Rust and James J DiCarlo. Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, 30(39):12978–12995, 2010.

Nicole C Rust, Odelia Schwartz, J Anthony Movshon, and Eero P Simoncelli. Spatiotemporal elements of macaque v1 receptive fields. *Neuron*, 46(6):945–956, 2005.

Jason M Samonds, Brian R Potetz, Christopher W Tyler, and Tai Sing Lee. Recurrent connectivity can account for the dynamics of disparity processing in v1. *Journal of Neuroscience*, 33(7): 2934–2946, 2013.

Daisuke Shimaoka, Kenneth D Harris, and Matteo Carandini. Effects of arousal on mouse sensory cortex depend on modality. *Cell reports*, 22(12):3160–3167, 2018.

Ann M. Skoczenski and Anthony M. Norcia. Late Maturation of Visual Hyperacuity. *Psychological Science*, 13(6):537–541, November 2002. ISSN 0956-7976. doi: 10.1111/1467-9280.00494. URL https://doi.org/10.1111/1467-9280.00494. Publisher: SAGE Publications Inc.

D Max Snodderly, Igor Kagan, and Moshe Gur. Selective activation of visual cortex neurons by fixational eye movements: implications for neural coding. *Visual neuroscience*, 18(2):259–277, 2001.

Robert M Steinman and John Z Levinson. The role of eye movement in the detection of contrast and spatial detail. *Eye movements and their role in visual and cognitive processes*, 4:115–212, 1990.

David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS computational biology*, 12(11):e1005177, 2016.

Yi-Zhong Wang, Sarah E. Morale, Robert Cousins, and Eileen E. Birch. The Course of Development of Global Hyperacuity Over Lifespan. *Optometry and vision science : official publication of the American Academy of Optometry*, 86(6):695–700, June 2009. ISSN 1040-5488. doi: 10.1097/OPX.0b013e3181a7b0ff. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2733828/.

G. Westheimer. Hyperacuity. In Larry R. Squire (ed.), *Encyclopedia of Neuroscience*, pp. 45–50. Academic Press, Oxford, January 2009. ISBN 978-0-08-045046-9. doi: 10.1016/B978-008045046-9.00218-7. URL https://www.sciencedirect.com/science/article/pii/B9780080450469002187.

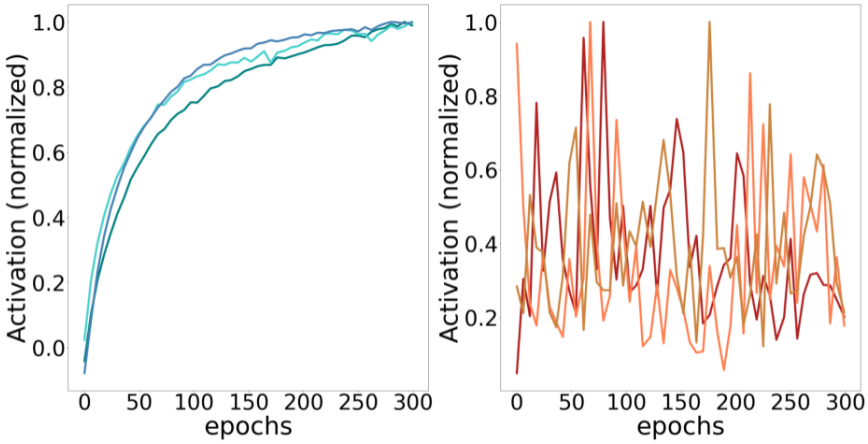
Yue Xi, Jiangbin Zheng, Wenjing Jia, Xiangjian He, Hanhui Li, Zhuqiang Ren, and Kin-Man Lam. See clearly in the distance: Representation learning gan for low resolution object recognition. *IEEE Access*, 8:53203–53214, 2020.

Daniel L. K. Yamins and James J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, March 2016. ISSN 1546-1726. doi: 10.1038/nn.4244. URL https://www.nature.com/articles/nn.4244. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 3 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Computational neuroscience;Object vision Subject_term_id: computational-neuroscience;object-vision.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

A FEATURE VISUALIZATION TRAINING PROCEDURE

As a first attempt of feature visualization training we followed the general guidelines of (Erhan et al., 2009; Nguyen et al., 2016). The code we used was based on an online available code example[2]. The activation of a feature i in a layer j of our network is denoted by $H_{ij}(\theta, x)$, where θ is the network's parameters, and x is the input. Using gradient ascent we optimize the input x to maximize the feature activation H_{ij}. Following Erhan et al. (2009) x was initialized randomly and the gradient ascent algorithm was applied modifying it pixel by pixel at each iteration. While this procedure has been shown to work with typical Convolutional Neural Network layers it failed to converge with the DRC-front-end, probably due to its higher dimensional input and complexity of the recurrent layers, (Fig. S4). Therefore, we designed a new training procedure that includes a deep generative network (DGN) which generates the synthetic input x, and uses gradient ascent to update the DGN weights using the Adam optimizer. The DGN architecture is shown in Table S9, the architecture was modified to enable the constraints we enforced over the system, i.e. using 3D,2D and 1D Transpose Convolution for the corresponding spatio-temporal, spatial-only and temporal-only activation maximization.



Supplementary Figure S4: Convergence of AM generator - on the left a plot of the DGN converging to a stable AM feature using the base small net; and on the right non-convergence of the DGN while using a deeper teacher framework. Data points were down samples and normalized for clarity

Activation maximization is usually a non-convex optimization problem, and therefore we do not expect to reach the same global maxima on each run. We applied a combination of two regularization methods, Batch-normalization over the networks layer output (Ioffe & Szegedy, 2015) and added Gaussian noise, with $\mu = 0$ and $\sigma = 0.05$ empirically chosen, to obtain repeatability of AM in an individual feature, while maintaining a significant difference between features. We averaged the results over ten different initialization of the generator to report more consistent results; see (Fig. S5 for examples of the convergence behavior of the algorithm.

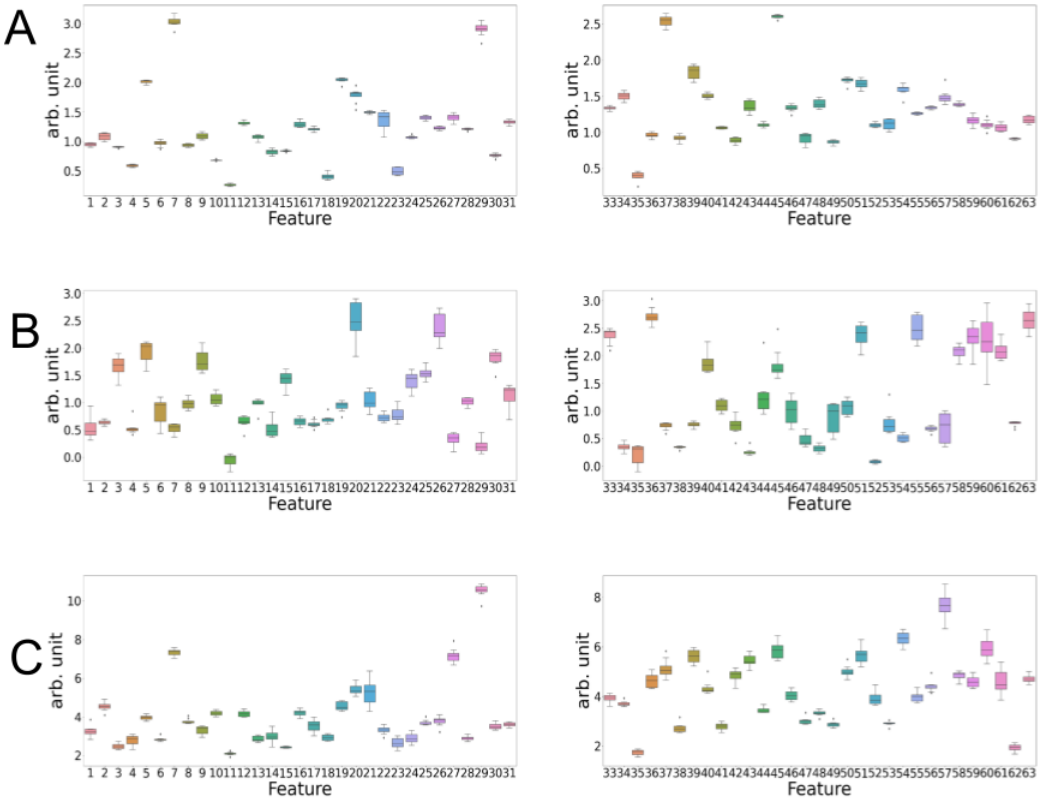
B IMPLEMENTATION OF DRC TRAJECTORIES

B.1 TRAJECTORIES WITH CONTROLLED CURVATURE

To create a family of trajectories with controllable curvature, we assumed that at each time-step the sensor location $x(t), y(t)$ is updated via polar increment $\delta r, \delta\phi$. Namely:

$$\begin{aligned} \phi(t) &= \phi(t-1) + \delta\phi(t) \\ x(t) &= x(t-1) + \delta r \cos(\phi(t)) \\ y(t) &= y(t-1) + \delta r \sin(\phi(t)) \end{aligned} \tag{1}$$

[2] $https://keras.io/examples/vision/visualizing_what_convnets_learn/$



Supplementary Figure S5: Boxplots displaying the distribution over intra-DRC AM values. The data represents six independent training trials of the generator with a single DRC-front-end. (A) boxplots of spatial only AM, we report the Coefficient of variation (CV) $CV = 0.043$, (B) temporal only AM $CV = 0.134$ (C) spatio-temporal AM $CV = 0.066$. The box represents the quarterlies of the data, and the whiskers mark the rest of the distribution. Outliers are marked in the graph. As can be seen, training the generators leads to a relatively constant local minimum.

With $\delta\phi(t)$ being i.i.d. stochastic variables drawn from a von Mises distribution with controlled parameter κ. Zero κ corresponds to uniform distribution of $\delta\phi(t)$, positive values of κ correspond to straighter trajectories and negative κ corresponds to more curved ones (here we define that for $\kappa < 0$, $\delta\phi = \pi + \delta\phi'$ with $\phi' \sim$ von Mises$(-\kappa)$).

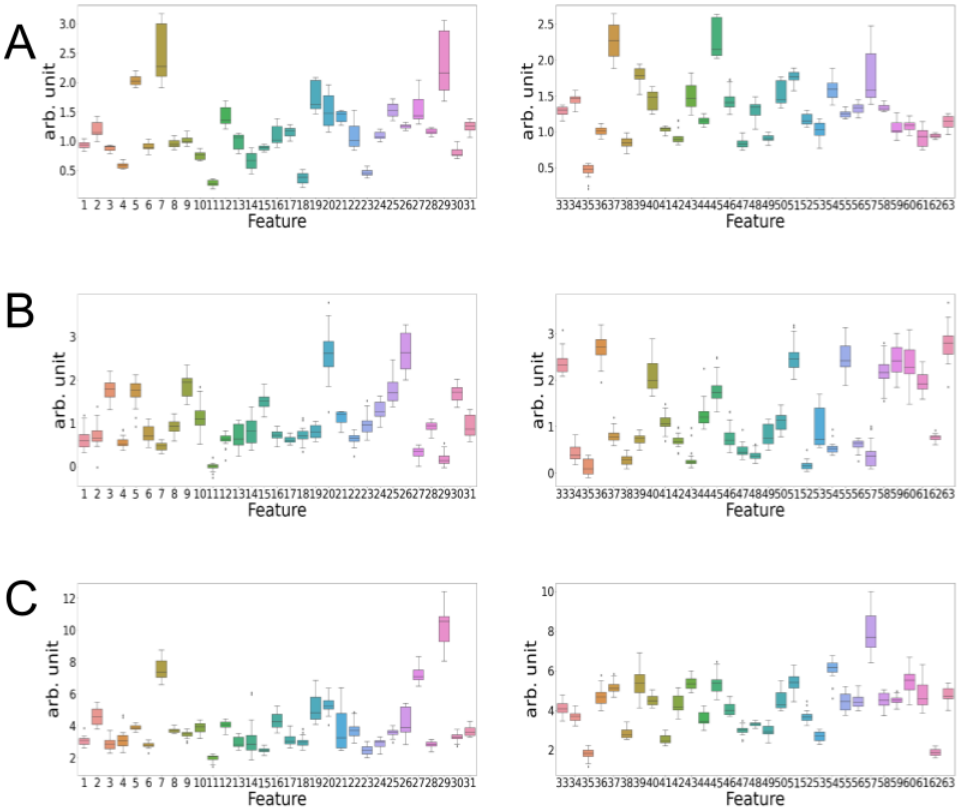
The second parameter, δr was drawn from a half-normal distribution, so that $\delta r = r_0 + |r_1|$ where $r_1 \sim \mathcal{N}(0, 1)$ and where we set $r_0 = \sqrt{2}$ to ensure that two consequent steps do not fall on the same point for any angle ϕ after rounding to integer pixel coordinates.

B.2 Spirals

Spirals were created by setting:

$$\begin{aligned} \phi(t) &= \phi(t-1) + \delta\phi(t) \\ r(t) &= r(t-1) + \delta r(t) \\ x(t) &= r(t)\cos(\phi(t)) \\ y(t) &= r(t)\sin(\phi(t)) \end{aligned} \tag{2}$$

with $\delta\phi(t) \sim \mathcal{N}(\pm\frac{\pi}{2}, \frac{\pi}{8})$, $\phi(0)$ drawn uniformly from circle, and the polarity of $\pm$ is fixed along each individual trajectory. The parameters $r(0) = 3$, $\delta r \sim \mathcal{N}(-0.1, 0.16^2)$ were picked heuristically to optimize Small-net performance as well as prevent trajectories from coinciding. Regarding coinciding and repetition of trajectories, we found that approximately 10.3K distinct trajectories



Supplementary Figure S6: Boxplots display the distribution over inter-DRC AM values, comparing different DRC trained on the same teacher. The data represent three independent DRCs, each trained on the same Small-net teacher. The box notation is as in the Fig S5. Spatial only $CV = 0.1$, temporal only $CV = 0.16$, spatio-temporal $CV = 0.104$.

were generated for a single pass over 45K large training set, making any fitting to specific trajectory unlikely.

B.3 Curvature index distributions

The same curvature index as in Gruber & Ahissar (2020) was used in order to compare our synthetic trajectories to biological drift motion. The index is defined as follows: $CI = 1 - ds/l$ where l is the length of the trajectory and ds is the final displacement (i.e. the distance between the starting and ending point of the trajectory). Using this definition we get one index per trajectory where highly curved trajectory would result with number close to (and smaller than) 1 while low curvature trajectories would result with number close to (and greater than) 0.

Comparing the curvature index distributions obtained with $\kappa = -1.0$ and $\kappa = 0.0$ (Fig. S10) with the relevant distributions from Gruber & Ahissar (2020) (Figure 3, 'Natural-small' curvature-index distributions), where distribution of curvature index with $\kappa = -1.0$ corresponds to 'border' drift trajectories distribution and distribution of curvature index with $\kappa = 0.0$ corresponds to 'non-border' drift trajectory distribution, we see similar trends: The $\kappa = 0.0$ ('non-border') distribution is shifted to the left and broader compared to the $\kappa = -1.0$ ('border') distribution. The means of the distributions are also comparable and exhibit the same trend: 0.699 ± 0.002 and 0.597 ± 0.003 for $\kappa = -1.0$ and $\kappa = 0.0$ respectively, and 0.65 ± 0.05 and 0.55 ± 0.02 for 'border' and 'non-border' conditions respectively.

Supplementary Table S3: DRC network architecture

DRC (time steps = 5)				
Layer	Number of outputs	Kernel size	Padding	Activation Function (Small-Net/ResNet)
Input image	5*8*8*3			
Input trajectory	10*2			
Broadcast trajectory	10*8*8*2			
Concat	10*8*8*5			
ConvGRU	10*8*8*32	3 * 3	same*	tanh
ConvGRU	10*8*8*64	3 * 3	same	tanh
ConvGRU	-	3 * 3	same	ELU/ReLU
AveragePooling1D	1*8*8*64	5		
LayerNormalization	-			
Back-end				
Weights Copied from Teacher				
Convolution ResNet	8*8*64	$\begin{bmatrix} 3*3,128 \\ 3*3,128 \\ 3*3,64 \end{bmatrix} \times 3$	same	ELU/ReLU
LayerNormalization	-			
Fully Connected	128			ELU/ReLU
Dropout - 0.0	-			
Fully Connected	10/100			softmax

*"same" padding refers to padding to the input image so that the input image gets fully covered by the filter and specified stride

Supplementary Table S4: Baseline ResNet50 based network architecture

Reference Net				
Layer	Number of outputs	Kernel size	Padding	Activation Function
ResNet50	14*14*2048			
Classifier				
GlobalAvrgPooling	7 * 7 * 2048	2 * 2		
Fully Connected	1024			ReLU
Fully Connected	512			ReLU
Fully Connected	10/100			softmax

B.4 Integration of positional Information

The positional information fed into the network contained the (x,y) normalized coordinates of the lower left corner of each frame relative to the center position of the original CiFAR image. It was integrated into the network by first broadcasting it to the 2d dimensions (height and width) of a single input frame (8,8,2) and then concatenating it with the frame. resulting in a (8,8,5) input for each time-step (three dimensions for RGB and two dimensions for location).

Supplementary Table S5: ResNet+GRU network architecture (applied to time series)

Control Net v2				
Layer	Number of outputs	Kernel size	Padding	Activation Function
ResNet50 (applied at each timestep)	5*14*14*2048			
Classifier				
GlobalAvrgPooling	5*7*7*2048	2 * 2		
GRU	5*1024			sigmoid-tanh
GRU	512			sigmoid-tanh
Fully Connected	10/100			softmax

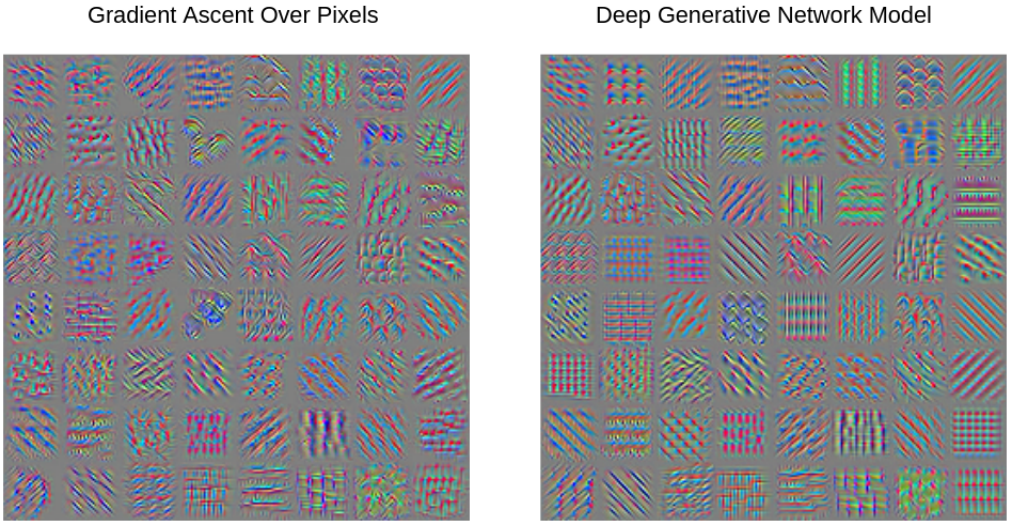
Supplementary Table S6: ResNet+ConvGRU network architecture (applied to time series)

Control Net v3				
Layer	Number of outputs	Kernel size	Padding	Activation Function
ResNet50 (applied at each timestep)	5*14*14*2048			
Classifier				
ConvGRU	5*1024			sigmoid-tanh
ConvGRU	512			sigmoid-tanh
GlobalAvrgPooling	5*7*7*2048	2 * 2		
Fully Connected	10/100			softmax

For ResNet+convGRU controls the positional information was fed similarly to the first recurrent layer, with appropriate scaling (approximately square root of the layer depth).

Supplementary Table S7: Base Small-net network architecture

Teacher				
Layer	Number of outputs	Kernel size	Padding	Activation Function
Front-end				
Input image	32*32*3			
Convolution	32*32*32	3 * 3	same	ELU
Convolution	32*32*32	3 * 3	same	ELU
MaxPooling2D	16*16*32	2 * 2		
Dropout - 0.2	-			
Convolution	16*16*64	3 * 3	same	ELU
Convolution	16*16*64	3 * 3	same	ELU
MaxPooling2D	8*8*64	2 * 2		
LayerNormalization	-			
Back-end				
Convolution ResNet	8*8*64	$\begin{bmatrix} 3*3,128 \\ 3*3,128 \\ 3*3,64 \end{bmatrix} \times 3$	same	ELU
LayerNormalization	-			
Fully Connected	128			ELU
Dropout - 0.0	-			
Fully Connected	10/100			softmax



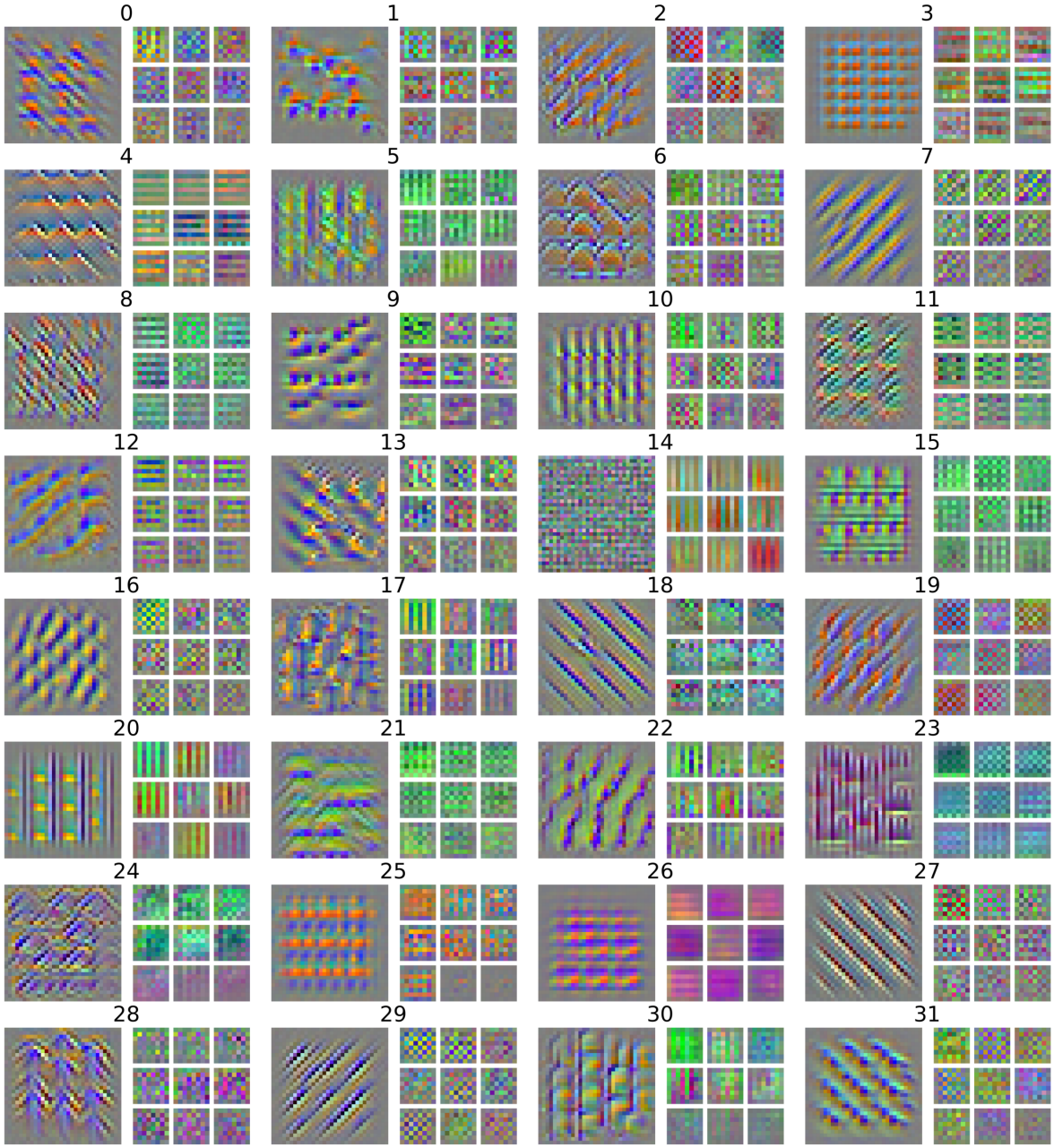
Supplementary Figure S7: Comparing results of activation maximization of the teacher and student networks. Left: results of gradient ascent over the input pixels for the teacher network. Right: output of DGN trained to generate maximizing input for the student network. The Similarity, easily observed between the two panels, confirms the validity of the DGN approach as an activation maximization methods for spatial networks.

Supplementary Table S8: Small-net v2 network architecture

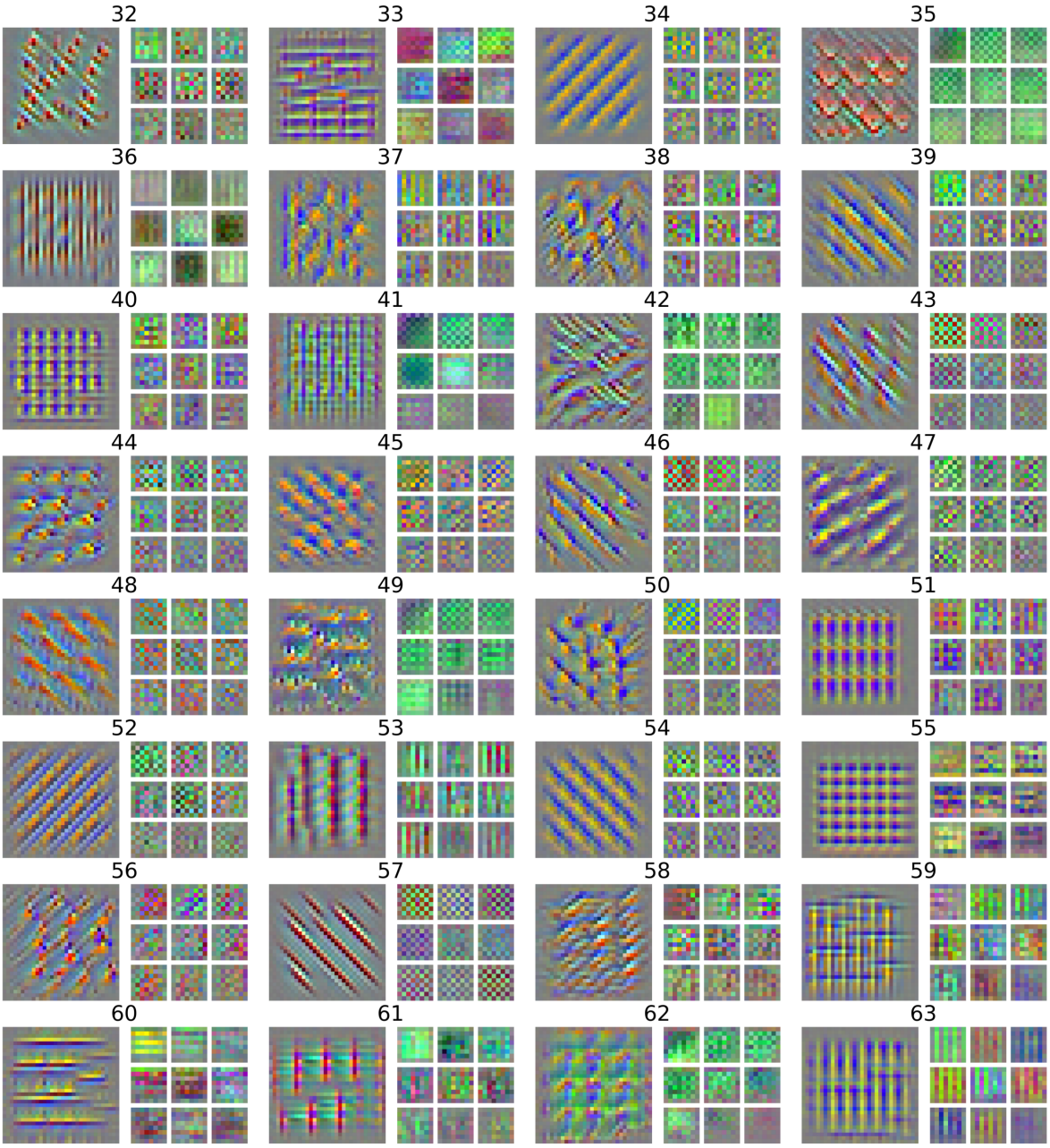
Teacher				
Layer	Number of outputs	Kernel size	Padding	Activation Function
Front-end				
Input image	32*32*3			
Convolution	32*32*32	3 * 3	same	ELU
Convolution	32*32*64	3 * 3	same	ELU
MaxPooling2D	16*16*64	2 * 2		
Dropout - 0.2	-			
Convolution ResNet	16*16*64	$\begin{bmatrix} 3*3,128 \\ 3*3,128 \\ 3*3,64 \end{bmatrix} \times 3$	same	ELU
MaxPooling2D	8*8*64	2 * 2		
LayerNormalization	-			
Back-end				
Convolution ResNet	8*8*64	$\begin{bmatrix} 3*3,128 \\ 3*3,128 \\ 3*3,64 \end{bmatrix} \times 3$	same	ELU
LayerNormalization	-			
Fully Connected	128			ELU
Dropout - 0.0	-			
Fully Connected	10/100			softmax

Supplementary Table S9: Activation Maximization DGN

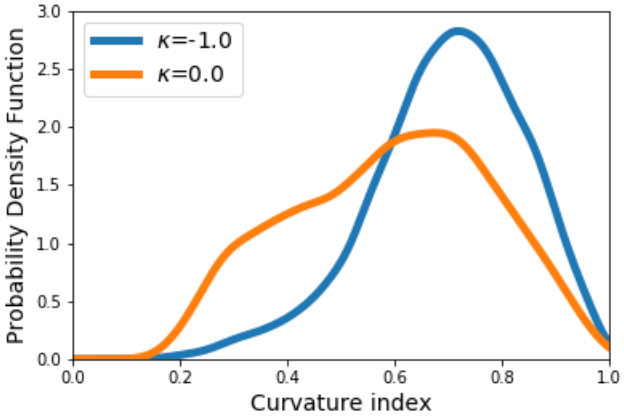
Generator					
Layer	Number of outputs	Kernel size	stride	Padding	Activation Function
input	100				
Fully Connected	128				LeakyReLU
Reshape	1*1*1*128				
Conv3DTranspose	2*2*2*128	2 * 2 * 2	2 * 2 * 2	valid	LeakyReLU
Conv3DTranspose	5*4*4*128	3 * 2 * 2	2 * 2 * 2	valid	LeakyReLU
Conv3DTranspose	10*8*8*128	2 * 2 * 2	2 * 2 * 2	valid	LeakyReLU
Conv3DTranspose	10*8*8*3	3 * 3 * 3	1 * 1 * 1	same	tanh



Supplementary Figure S8: Teacher features vs. student features. Same as Fig. 2A for all the features.



Supplementary Figure S9: Teacher features vs student features, continued. Same as Fig. 2A for all the features.



Supplementary Figure S10: **Dsitributions of curvature index.** Dsitributions of curvature index of trajectories generated with $\kappa = -1.0$ (blue) and $\kappa = 0.0$ (orange).