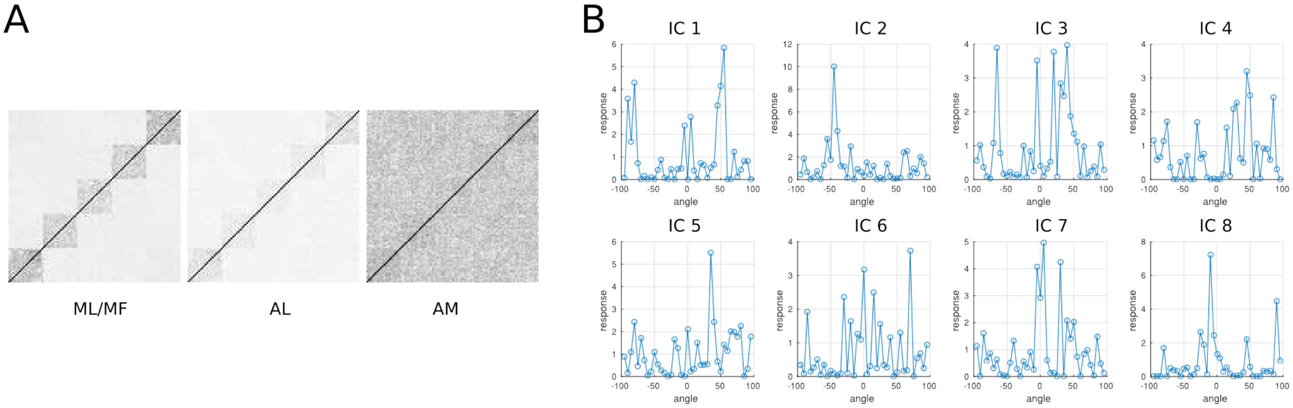

Figure S1: Related to main figures 3 and 4. Experiments with ICA. This experiment used the same pipeline as in main text (Figures 3 and 4) but with PCA replaced by ICA (including the ZCA-whitening preprocessing step [S1]). For each training identity, ICA was performed to get 39 independent component directions. Each testing image was projected to these directions. A square nonlinearity and a pooling were performed on the results. We show (A) the model population similarity matrices of different stages (similar to Figure 4A) and (B) some single cell responses in stage AL (similar to Figure 3A). Unlike PCA, the order of the independent components are arbitrary.

SUPPLEMENTAL EXPERIMENTAL PROCEDURES

Stimuli

40 face models were rendered with perspective projection. Each face was rendered (using Blender) at each orientation in 5° increments from $-95°$ to $95°$. The untextured face models were generated using Facegen (Singular Inversions). All faces appeared on a uniform gray background.

View-invariant Same-different Pair Matching Task

For each of the 5 repetitions of the same-different pair matching task, 20 template and 20 test individuals were randomly selected from the full set of 40 individuals. The template and test sets were chosen independently and were always disjoint. 50% of the 600 test pairs sampled from each testing interval depicted the same two individuals. Each testing interval was symmetric about 0° (frontal) and testing intervals were ordered by inclusion. The smallest was $[-10°, 10°]$ and the largest was $[-95°, 95°]$ (all views up to and including the left and right profile views). The classifier compared the Cosine similarity of the two zero-mean, and unit-standard deviation representations to a threshold. The threshold was integrated over to compute the area under the ROC curve (AUC). The abscissa of Figure 3 is the radius of the testing interval from which test pairs were sampled. The ordinate of Figure 3 is the mean AUC ± the standard deviation computed over the 5 repetitions of the experiment. The similarity matrix in Figure 4 was obtained by computing Pearson's linear correlation coefficient between each test sample pair. The same matrix was computed 10 times with different training/test splits and the average was reported. Same procedures were repeated for features from area MLMF, AL and AM to get corresponding matrices.

MATHEMATICAL APPENDIX

The key results in this appendix can be informally stated as follows:

- We prove than a number of learning rules, supervised and unsupervised, are equivariant with respect to the symmetries of the training data. We use this result in the case of training data consisting of images of faces for all view angles obtaining equivariance of the solutions of the learning rules with respect to the reflection group and the group of rotations. The implications that we use in the paper are

  - the solutions of all learning rules can be used as templates in the computation of an invariant signature. The algorithm consists of performing dot products of the input image with each template, transforming nonlinearly (for instance using a rectifier nonlinearity or a square) the result and then pooling over *all* templates, i.e., the solutions of the learning rule. The result is approximately invariant to rotation in depth.
  - in the case of the Oja rule we prove that the solutions are even or odd functions of the view angle; a square nonlinearity provides even functions, which are mirror-symmetric. We were not able to prove such a property for any of the other learning rules.

- in the case of the ICA rule we show empirical evidence that the solutions are neither odd nor even. This suggests that most learning rules do not lead to even or odd solutions.

This appendix is divided into three sections:

1. In section 1 we show how recent theorems on invariance under group transformations could be extended to nongroups and under which conditions. We show how an *approximately invariant* signature can be computed in this setting. In particular we analyze the case of rotation in depth and mirror symmetry transformations of bilateral symmetric objects such as faces.

2. In section 2 we describe how the group symmetry properties of the set of images to which neurons are exposed (the "unsupervised" training set) determine the symmetries of the learned weights. In particular we show how the weight symmetries gives a simple way of computing an invariant signature.

3. In section 3 we prove that the solutions of the Oja equation, given that the input vectors that are reflections of each other (like a face's view at θ degrees and its view at $-\theta$ degrees), must be odd or even.

In the following we indicate with $x \in R^d$ an image, with $w \in R^d$ a filter or neural weight and with G a locally compact group.

1 Approximate invariance for non-group transformations

In this section we analyze the problem of getting an approximately invariant signature for image transformations that do not have a group structure. In fact, clearly, not all image transformations have a group structure. However assuming that the object transformation defines a smooth manifold we have (by the theory of Lie manifolds) that locally a Lie group is defined by the generators on the tangent space. We illustrate this in a simple example. Let $x \in \mathbb{R}^d$. Let $s : \mathbb{R}^d \times \mathbb{R}^Q \to \mathbb{R}^d$ a C^∞ transformation depending on $\Theta = (\theta_1, \cdots, \theta_Q)$ parameters. For any fixed $x \in \mathbb{R}^d$ the set $M = (s(x, \Theta),\ \Theta \in \mathbb{R}^Q)$ describe a differentiable manifold. If we expand the transformation around e.g. $\vec{0}$ we have:

$$s(x, \Theta) = s(x, \vec{0}) + \sum_{i=1}^{Q} \frac{\partial s(x, \Theta)}{\partial \theta_i} \theta_i + o(\|\Theta\|^2) = x + \sum_{i=1}^{Q} \theta_i L_{\theta_i}(x) + o(\|\Theta\|^2) \tag{1}$$

where L_{θ_i} are the infinitesimal generators of the transformation in the i^{th} direction.
Therefore locally (when the term $o(\|\Theta\|^2)$ can be neglected) the associated group transformation can be expressed by exponentiation as:

$$g(\Theta) = \exp(\theta_1 L_{\theta_1} + \theta_2 L_{\theta_2} + \cdots + \theta_Q L_{\theta_Q}).$$

Note that the above expansion is valid only locally. In other words instead of a global group structure of the transformation we will have a collection of local transformations that obey a group structure. The results derived in section 2 will then say that the local learned weights will be orbits w.r.t. the local group approximating the non-group global transformation.

1.1 Invariance under rotations in depth

The 3D "views" of an object undergoing a 3D rotation are group transformations but the 2D projections of an object undergoing a 3D rotation are not group transformations. However for any fixed angle θ_0 and for small rotations the projected images approximately follow a group structure. This can be easily seen making the substitution in eq. (1) $s(x, \Theta) = P(r_\theta x)$ where P is the 2D projection. Let $\eta : \mathbb{R} \to \mathbb{R}$ be a nonlinear function, e.g., squaring or rectification. For small values of θ we have therefore that the signature:

$$\mu_w(x) = \int_{-\theta_0}^{\theta_0} d\theta \; \eta(\langle Px, Pr_\theta w\rangle)$$

or its discrete version

$$\mu_w(x) = \sum_i \eta(\langle Px, Pr_{\theta_i} w\rangle) = \sum_i \eta(\langle Px, g(\theta_i)Pw\rangle)$$

is invariant under 3D rotation of x of an angle $\bar{\theta}$ up to a factor proportional to $O(\|\bar{\theta}\|)$. Alternatively if the following property holds:

$$\langle Px, Pr_\theta w\rangle = 0 \;\; \theta > \bar{\theta} \tag{2}$$

the invariance will be exact (see [S2, S3]); this is the case e.g. when both w and x are faces.

The locality of the group structure (eq. (2)) means that we have invariance of the signature only within each local neighborhood but not over all viewpoints. A reasonable scenario could be that each local neighborhood may consist of, say, ±30 degrees (depending on the universe of distractors). Almost complete view invariance can be obtained from a single view at +30 degrees. In fact the view, together with the associated virtual view at −30 degrees because of mirror symmetry, provides invariance over −60, +60 degrees [S4].

1.2 Rotation in depth and mirror symmetry.

As explained on the previous paragraph, projected rotations in depth are not group transformations. However in the case of a bilateral symmetric objects, as we will see below, projected rotations in depth are a collection of orbits of the mirror symmetry group. Section 2 will clarify why this property is important proving that it forces the set of solutions of a variety of learning rules to be a collection of orbits w.r.t. the mirror symmetry group.

Consider e.g. a face, x, which is a bilateral symmetric object and its orbit in $3D$ w.r.t. the rotation group (here we take a subgroup of cardinality N for simplicity):

$$O_x = (r_{-\theta_N} x, \cdots, r_0 x, r_0 x, \cdots, r_{\theta_N} x).$$

where r is a rotation matrix in 3D, e.g. w.r.t. the z axis.

Projecting onto $2D$ we have

$$P(O_x) = (P(r_{-\theta_N} x), \cdots, P(r_0 x), (r_0 x), \cdots, P(r_{\theta_N} x)).$$

Note now that, due to the bilateral symmetry, the above set can be written as:

$$P(O_x) = (x_0, \cdots, x_N, Rx_0, \cdots, Rx_N).$$

where $x_n = Pr_{\theta_n} x$, $n = 0, \cdots, N$ and R is the reflection operator. The set consists of a collection of orbits w.r.t. the group $G = \{e, R\}$. This is due to the relation

$$x_n = P(r_{\theta_n} x) = Rx_{N+n} = R(Pr_{-\theta_n} x).$$

i.e. a face rotated by an angle θ_n and then projected is equal to the reflection of the same face rotated by an angle $-\theta_n$ and projected.

The reasoning generalizes to multiple faces. In summary in the specific case of bilateral symmetric objects rotating in depth, a projection onto a plane parallel to the rotation axis creates images which are transformations w.r.t. the group of reflection, thus falling in the group case described in the above paragraphs.

2 Unsupervised and supervised learning and data symmetries

In the following we show how symmetry properties on the neuronal inputs affect the learned weights. We model different unsupervised (Hebbian, Oja, Foldiak, ICA) or supervised learning (SGD) rules as dynamical systems coming from the requirement of minimization of some target function. We see how these dynamical systems are equivariant (in the sense specified below) and how equivariance determines the symmetry properties of their solutions.

This gives a simple way to generate an invariant signature by averaging over all solutions.

2.1 Equivariant dynamical systems and their solutions.

We make the general assumption that the dynamical system can be described in terms of trying to minimize a non-linear functional of the form:

$$\operatorname*{argmin}_{w \in X} \ \mathcal{L}(w,x), \quad \mathcal{L}(w,x) = h(w,x), \quad x, w \in \mathbb{R}^d \tag{3}$$

The associated dynamical system reads as:

$$\dot{w} = f(w) = \dot{h}(w,x). \tag{4}$$

A general result holds for equivariant dynamical systems. A dynamical system is called *equivariant* w.r.t. a group G if f in eq. (4) commutes with any transformation $g \in G$ i.e.

$$f(gw) = gf(w), \quad \forall g \in G. \tag{5}$$

In this case we have:

Theorem 1 *If an equivariant dynamical system has a solution w, then the whole group orbit of w will also be a set of solutions (see [S5]).*

In the following we are going to analyze different cases of updating rules for neuronal weights showing, under the hypothesis that the training set is a (scrambled) collection of the orbits i.e. we specialize the set X to be of the form:

$$X = G\mathcal{T}, \ \ \mathcal{T} \in \mathbb{R}^{d\times N}, \ \ X = \{x_1, \cdots, x_N\}, \tag{6}$$

that the dynamical system is equivariant.
We will see that the following variant of the equivariance holds for many dynamical systems:

$$f(gw, x) = gf(w, \pi_g(x)), \quad \forall g \in G, \ x \in X. \tag{7}$$

where $\pi_g(x)$ is permutation of the set X that depends on g. The derivation stands on the simple observation:

$$\langle x, gw \rangle = \langle g^{-1}x, w \rangle$$

and the hypothesis that the training set is a collection of orbits. In fact in this case

$$gX = \pi_g(X).$$

In general if the training set X is large enough the dynamical system will be equivalent to the unpermuted one due to the stability of the stochastic gradient descent method [S6] for the supervised learning case. For the unsupervised case, since the dynamical systems associated with the Oja and the ICA rules minimize statistical moments they are (being averages) independent of training data permutations. The fact that the set of solutions is a collections of orbits, $S = \bigcup_i O_i$ implies that any average operator over them is invariant. In our case the operator is the signature:

$$\mu(x) = \sum_{ij} \eta(\langle x, O_{ij} \rangle) \tag{8}$$

where O_{ij} is the element j of the orbit i and $\eta : \mathbb{R} \to \mathbb{R}$ is a non-linear function.

In the following we prove equivariance of a few learning rules.

1. **Unsupervised learning** rules [S7]:

   In the following $x \in X$ and $\alpha > 0$ and with the notation $\pi_g(x)$ we indicate the permutation of the element x in the training set X due to the transformation g.

   - **Hebbian learning**. Choosing
   $$\mathcal{L}(w,x) = \frac{\alpha}{2} y^2 \tag{9}$$
   where $y = \langle x, w \rangle$ is the neuron's response, we have the associated dynamical system is:
   $$\dot{w} = f(x,w) = \alpha \langle x, w \rangle x. \tag{10}$$
   The system is equivariant. In fact:
   $$f(x, gw) = \alpha \langle x, gw \rangle x = g\alpha \langle g^{-1}x, w \rangle g^{-1}x = g\alpha \langle \pi_g(x), w \rangle \pi_g(x) = gf(\pi_g(x), w).$$

   - **Oja learning**. Choosing
   $$\mathcal{L}(w,x) = \frac{\alpha}{2\left\|w\right\|_2} \langle x, w \rangle^2 \tag{11}$$
   we obtain by differentiation:
   $$\dot{w} = f(w,x) = \alpha \frac{y}{\left\|w\right\|_2}(x - y\frac{w}{\left\|w\right\|_2}). \tag{12}$$
   The obtained dynamical system is that of Oja's for the choice $\left\|w\right\|_2 = 1$. The system is equivariant (note that $\left\|gw\right\|_2 = \left\|w\right\|_2$). In fact:
   $$\begin{aligned} f(gw, x) &= \alpha \langle x, gw \rangle (x - \langle x, gw \rangle gw) = \alpha \langle g^{-1}x, w \rangle g(g^{-1}x - \langle g^{-1}x, w \rangle w) \\ &= \alpha \langle \pi_g(x), w \rangle g(\pi_g(x) - \langle \pi_g(x), w \rangle w) = gf(w, \pi_g(x)) \end{aligned}$$

   - **ICA**. Choosing
   $$\mathcal{L}(w,x) = \alpha \frac{\langle x, w \rangle^4}{4} + \frac{\left\|w\right\|_2^2}{2} \tag{13}$$
   we obtain the dynamical system:
   $$\dot{w} = \alpha(\langle x, w \rangle^3 x - w) \tag{14}$$
   which can be shown to extract one ICA component [S8]. The system is equivariant. In fact:
   $$f(x, gw) = \alpha(\langle x, gw \rangle^3 x - gw) = g\alpha(\langle g^{-1}x, w \rangle^3 g^{-1}x - w) = gf(\pi_g(x), w).$$

   - **Foldiak.** Choosing:
   $$\mathcal{L}(x,w) = \frac{\alpha}{2}\bar{y}^2, \quad \bar{y} = \int_{t_0}^{t} d\tau \;\; \langle w, x \rangle (\tau) \tag{15}$$
   the associated dynamical system is:
   $$\dot{w}(t) = \alpha \Big(\int_{t_0}^{t} d\tau \;\; \langle w, x \rangle (\tau) \Big) x(t) = \alpha \bar{y} x(t) \tag{16}$$
   which is the so called Foldiak updating rule. The system is equivariant. In fact:
   $$\begin{aligned} f(x, gw) &= \alpha \Big(\int_{t_0}^{t} d\tau \;\; \langle gw, x \rangle (\tau) \Big) x = g\alpha \Big(\int_{t_0}^{t} d\tau \;\; \langle w, g^{-1}x \rangle (\tau) \Big) g^{-1}x \\ &= \alpha g \bar{y}(w, \pi_g(x)) \pi_g(x) = gf(w, \pi_g(x)) \end{aligned}$$

2. **Supervised learning in deep convolutional networks**. The reasoning above can be extended to supervised problems of the form:
   $$\underset{W}{\text{argmin}} \;\; \mathcal{L}(X, \ell, W), \;\; X = (x_1, \ldots, x_N) \tag{17}$$

where $\mathcal{L}(X, \ell, W) = Loss(X, \ell, W)$. The term $Loss(X, \ell, W)$ is a function defined using the loss of representing a set of observations X, their labels ℓ, and a the set of the network weights W. The updating rule for each weight w_l is given by the backpropagation algorithm:

$$\dot{w}_l = \frac{\partial \mathcal{L}}{\partial w_l}. \tag{18}$$

If the equation above is equivariant the same results of the previous section will hold, i.e., if there exists a solution the whole orbit will be a set of solutions. In the following we analyze the case of deep networks showing that equivariance holds if the output at each layer l, o_l is covariant w.r.t. the transformation, i.e.:

$$o_l(gx) = g o_l(x), \;\; \forall\, g \in G \tag{19}$$

We analyze the case of **deep convolutional networks** with pooling layers between each convolutional layer. In this case the response at each layer is covariant w.r.t. to the input transformation: the output at layer l is of the form:

$$o_l(X, W_{l-1})(g) = \int_{gG_l} d\hat{g}\; \eta(\langle o_{l-1}(X, W_{l-2}), \hat{g} w_l\rangle) = \int_{gG_l} d\hat{g}\; \eta(o_{l-1}(X, W_{l-2}) * w_l)(\hat{g}) \tag{20}$$

i.e. it is an average of a group convolution where o_{l-1} is the output of layer $l-1$ and W_{l-1} is the collection of weights up to layer $l-1$. Using the property that the group convolution commutes with group shift i.e. $[(T_{\bar{g}} f) * h](g) = T_{\bar{g}}[f * h](g)$ we have:

$$\begin{aligned} o_l(\bar{g}X, W_{l-1})(g) &= \int_{gG_l} d\hat{g}\; \eta(\bar{g} o_{l-1}(X, W_{l-2}) * w_l)(\hat{g}) = \int_{gG_l} d\hat{g}\; \eta(o_{l-1}(X, W_{l-2}) * w_l)(\bar{g}\hat{g}) \\ &= \int_{\bar{g}gG_l} d\hat{g}\; \eta(o_{l-1}(X, W_{l-2}) * w_l)(\hat{g}) = o_l(X, W_{l-1})(\bar{g}g) = \bar{g} o_l(X, W_{l-1})(g). \end{aligned}$$

where we used the property $o_{l-1}(\bar{g}X, W) = \bar{g} o_{l-1}(X, W)$. This can be seen to hold using an inductive reasoning up to the first layer where:

$$o_2(\bar{g}x, W_1)(g) = \int_{gG_1} d\hat{g}\; \eta((\bar{g}x) * w_1)(\hat{g}) = \int_{\bar{g}gG_1} d\hat{g}\; \eta(x * w_1)(\hat{g}) = \bar{g} o_1(x, W_1)(g).$$

In the following we prove that the dynamical systems (updating rules for the weights) associated to a deep convolutional network are equivariant. We consider e.g. the square loss function (the same reasoning can be extended to many commonly used loss functions):

$$\mathcal{L}(\phi_L(X, W), \ell) = \sum_{\ell} (1 - y_\ell \phi(X, W))^2.$$

where

$$\phi_L(X, W) = \phi_L(\cdots, \phi_3(\phi_2(X, w_1), w_3), \cdots, w_l \cdots, w_L)$$

being L the layers number and ℓ is a set of labels. The associated dynamical system reads as:

$$\frac{\partial \mathcal{L}(\phi_L(X, W), \ell)}{\partial w_l} = \dot{\mathcal{L}}(\phi_L(X, W), \ell) \frac{\partial \phi_L(X, W)}{\partial w_l} = 2 \sum_{\ell} (1 - y_\ell \phi_L(X, W)) \frac{\partial \phi_L(X, W)}{\partial w_l}$$

Substituting w_l with $\bar{g} w_l$ we have, by the covariance property, that the first factor of the r.h.s. of the equation above becomes $\sum_\ell (1 - y_\ell \phi_L(\pi_{\bar{g}}(X), W))$. We are then left to prove the equivariance of the second factor.

Using the chain rule, we have:

$$\begin{aligned} \dot{w}_l &= \frac{\partial \phi_L(\cdots, \phi_3(\phi_2(x, w_1), w_2) \cdots, w_L)}{\partial w_l} \\ &= \dot{\phi}_L[o_L(W_{L-1}, x)]\; \dot{\phi}_{L-1}[o_{L-1}(W_{L-2}, x)] \;\cdots\; \dot{\phi}_l(o_{l-1}(x, W_{l-2}), w_l), \cdots, w_L) \end{aligned}$$

where $o_j(W_{l-1}, x) = \phi_j(\cdots \phi_l(o_{l+1}(x, W_{l-1}), w_l), \cdots, w_L)$, $l < j < L$, being the output at layer j.
Notice that, in the case of covariant layer outputs, we have:

$$\begin{aligned} \phi_j(\cdots \phi_{l+1}(o_l(X, W_{l-1}), \bar{g}w_l), \cdots, w_L) &= \phi_j(\cdots \phi_{l+1}(\bar{g}^{-1}o_l(X, W_{l-1}), w_l), \cdots, w_L) \\ &= \phi_j(\cdots \phi_{l+1}(o_l(\bar{g}^{-1}X, W_{l-1})), w_l), \cdots, w_L) \\ &= \phi_j(\cdots \phi_{l+1}(o_l(\pi_{\bar{g}}(X), W_{l-1})), w_l), \cdots, w_L) \end{aligned}$$

where we used the covariance property in eq. (19) and the fact that the training set is a collection of orbits w.r.t. the group G.
Finally we have:

$$\frac{\partial \mathcal{L}(X, \{w_1, \cdots, \bar{g}w_l, \cdots, w_L\}, \ell)}{\partial w_l} = \bar{g}\frac{\partial \mathcal{L}(\pi_g(X), \{w_1, \cdots, w_l, \cdots, w_L\}, \ell)}{\partial w_l}$$

where the $\bar{g}$ comes from the derivative of $\bar{g}w_l$ w.r.t. w_l.
Summarizing we have the following result

Theorem 2 *For $i = 1, \ldots, L$, let $\phi_i : \mathbb{R}^{d_i} \to R^{d_{i+1}}$ depend on a set of weights w_i. Consider a deep convolutional network with output of the form*

$$\phi_L(X, W) = \phi_L(\cdots, \phi_2(\phi_1(X, w_1), w_2), \cdots, w_l) \cdots, w_L). \tag{21}$$

and a differentiable square loss $\mathcal{L}(\phi_L(X, W), \ell)$, being ℓ a set of labels.
If X is a collection of orbits and and each ϕ_i is covariant, then the associated dynamical systems for each layer's weights' evolution in time

$$\dot{w}_l = \frac{\partial \mathcal{L}(\phi_L(X, W), \ell)}{\partial w_l}$$

are equivariant w.r.t. the group G.

3 Proof that the Oja equation's solutions are odd or even.

So far we have shown how biologically plausible learning dynamics in conjunction with appropriate training sets lead to solutions capable of supporting the computation of a view-invariant face signature (Sections 1 − 2). We showed that several different learning rules satisfied these requirements: Hebb, Oja, Foldiak, ICA, and supervised backpropagation (Section 2.1). Now we use properties specific to the Oja rule to address the question of why mirror symmetric responses arise in an intermediate step along the brain's circuit for computing view-invariant face representations.

We now use the following well-known property of Oja's learning rule: that it implements an online algorithm for principal component extraction [S9]. More specifically, we use that the Oja dynamics converge to an eigenfunction of the training set's covariance $C(X)$.

Recall from section 1.2 that in order to guarantee approximate view-invariance for bilaterally symmetric objects like faces, the training set X must consist of a collection of orbits of faces w.r.t. the reflection group $G = (e, R)$. We now show that this implies the eigenfunctions of $C(X)$ (equivalently, the principal components (PCs) of X) must be odd or even.

Under this hypothesis the covariance matrix $C(X)$ can be written as

$$C(X) = XX^\intercal = \mathcal{T}\mathcal{T}^\intercal + R\mathcal{T}\mathcal{T}^\intercal R^\intercal$$

where $\mathcal{T}$ is the set of the orbit representatives (untransformed vectors).
It is immediate to see that the above implies $[C(X), R] = 0$ (they commute). Thus $C(X)$ and R must share the same eigenfunctions. Finally, since the eigenfunctions of the reflection operator R are odd or even, this implies the eigenfunctions of $C(X)$ must also be odd or even.

Finally, we note that in the specific case of a frontal view, even basis functions (w.r.t. the zero view) are mirror symmetric.

SUPPLEMENTAL REFERENCES

S1. Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis,? Neural Netw. 10, 626–634.

S2. Anselmi, F., Leibo, J.Z., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. (2016). Unsupervised learning of invariant representations. Theor. Comput. Sci. 633, 112–121.

S3. Leibo, J. Z., Liao, Q., Anselmi, F., and Poggio, T. (2015). The Invariance Hypothesis Implies Domain-Specific Regions in Visual Cortex. PLoS Comput. Biol. 11, e1004390.

S4. Poggio, T., Vetter, T., Bulthoff, H.H. (1992). 3D Object Recognition: Symmetry and Virtual Views. AI-M-1409.

S5. Golubitsky, M., Stewart, I. (2002). The symmetry perspective: from equilibrium to chaos in phase space and physical space, (Springer Basel AG).

S6. Hardt, M., Recht, B. and Singer, Y. (2015). Train faster, generalize better: Stability of stochastic gradient descent. arXiv:1509.01240.

S7. Hassoun, M.H. (1995). Fundamentals of artificial neural networks, (MIT Press).

S8. Hyvärinen, A. Oja, E. (1998). Independent component analysis by general nonlinear hebbian-like learning rules. Signal Processing 64, 301–313.

S9. Oja, E. (1992). Principal components, minor components, and linear neural networks. Neural Netw. 5, 927–935.