# Decoding the Deep: Exploring class hierarchies of deep representations using multiresolution matrix factorization

Vamsi K. Ithapu
University of Wisconsin-Madison
http://pages.cs.wisc.edu/~vamsi/projects/incmmf.html

## Abstract

*The necessity of depth in efficient neural network learning has led to a family of designs referred to as very deep networks (e.g., GoogLeNet has 22 layers). As the depth increases even further, the need for appropriate tools to explore the space of hidden representations becomes paramount. For instance, beyond the gain in generalization, one may be interested in checking the change in class compositions as additional layers are added. Classical PCA or eigen-spectrum based global approaches do not model the complex inter-class relationships. In this work, we propose a novel decomposition referred to as multiresolution matrix factorization that models hierarchical and compositional structure in symmetric matrices. This new decomposition efficiently infers semantic relationships among deep representations of multiple classes, even when they are not explicitly trained to do so. We show that the proposed factorization is a valuable tool in understanding the landscape of hidden representations, in adapting existing architectures for new tasks and also for designing new architectures using interpretable, human-releatable, class-by-class relationships that we hope the network to learn.*

## 1. Introduction

The ability of a feature to succinctly represent the presence of an object/scene is, at least, in part, governed by the relationship of the learned representations across multiple object classes/categories. Cross-covariate contextual dependencies have been shown to improve the performance, for instance, in object tracking and recognition in vision [37, 29] and medical applications [13] (a motivating aspect of adversarial learning [23]). This then poses an interesting question – *Do the semantic relationships learned by the deep representations associate with those seen by humans?* Or in other words, can the relationships between different tasks (or categories) inferred by the highest layer representations 'drive' the design of the network itself (e.g., to add an additional layer or to remove an existing one)? For instance, can such models infer that cats are closer to dogs than they are to bears; or that bread goes well with butter/cream rather than, say, salsa. Invariably, addressing these questions amounts to learning hierarchical and categorical relationships in the class-covariance of hidden representations. Using classical techniques may not easily reveal interesting, human-relateable trends as recently shown by [26]. Note that the problem here is not about constructing learning models that *model* the semantics, like those using statistical relational learning [25] or rule-based methods [6, 18]. Instead, we are asking whether one can *decode* the semantics of representations from an *arbitrary* (trained) network.

**Related Work:** The problem of understanding the landscape of representations learned by machine learning models is not entirely new, and interpreting learning models has a long history [19, 22, 17]. For instance, some approaches construct the learning model to be parsable to begin with, like decision sets [32] or decision lists [19]. Alternatively, the learned features are inverted appropriately to visualize what the learning model sees (e.g. HOGgles [31]). Several recent studies have approached the problem of *interpreting of deep representations* from alternate view-points [21]. The seminal papers on deep networks [10, 9, 34] argue that, intuitively and empirically, higher layer representations learn abstract relationships between objects in a given image. However, it is unclear whether we can "define" what these abstract relationships (that the network should learn) are supposed to be. In [28, 35, 36], the authors visualize the representation classes using supervised learning methods, and extra semantic information is provided during training time to improve interpretability [7]. [28, 8] have addressed similar aspects for deep representations by visualizing image classification and detection models, and there is recent interest in designing tools for visualizing what the network perceives when predicting a test label [35]. As shown in [1], the contextual images that a deep network (even with good detection power) desires to see may not even correspond to real-world scenarios.

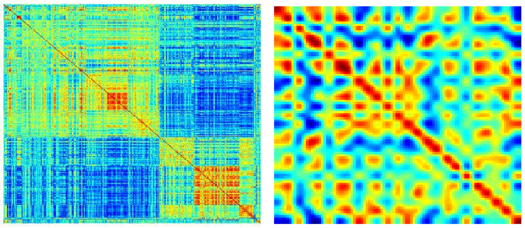


Figure 1. Example category (or class) covariances from AlexNet and VGG-S (of a few ImageNet classes).

### 1.1. Overview

The contribution of this work is a general framework, and an exploratory tool, that models the hierarchical relationships between several classes (or categories, tasks) of any given arbitrary (trained) deep network. To concretize the argument above, consider the representations of some ImageNet classes learned by AlexNet [16] and VGG-S [5]. Figure 1 shows the class covariance matrices (i.e., each entry is covariance computed using multiple instances from two classes) computed using the final hidden layer representations of the two networks respectively. Clearly, there is rich hierarchical, compositional, blocks-of-blocks type structure, and directly accounting for this parsimonious structure will reveal the hierarchical relationships between different class-specific representations. Invoking the de facto constructs like sparsity, low-rank or a decaying eigen-spectrum cannot account for such localized cluster-like structures. Devising such block-structured kernels was the original motivation for several local low-rank and hierarchical factorizations [27, 4, 20, 38]. However these methods are mainly heuristic, sensitive to the choice of several hyper-parameters involved (like the choice of local patch size, the local-rank etc.), require making hard-partitioning of data into clusters, non-trivial to compute for large dimensions and mostly do not necessarily capture the global and local structure equally well.

We instead take a more natural approach that follows directly from the *generative structure* of the matrix. Consider a symmetric matrix $\mathbf{C} \in \mathbb{R}^{m \times m}$ (like those from Figure 1), and let us say that an oracle provides the strongest possible $k < m$ number of rows/columns (classes here). In principle, one can decorrelate (diagonalize) these $k$ rows to reveal the hidden dependance of the remaining classes on the $k$ de-correlated set of classes. This procedure can then be repeated in a systematic fashion – the only unknowns being the choice of $k$, and which rows/columns to diagonalize within each iteration. One can see that this procedure relates to decomposing a higher-order rotation matrix into composition of $k^{th}$ order rotations. Unlike PCA which decomposes $\mathbf{C}$ as $\mathbf{Q}^T \mathbf{\Lambda} \mathbf{Q}$ where $\mathbf{Q}$ is an orthogonal matrix or sparse PCA (sPCA) [39] which imposes sparsity on the columns of $\mathbf{Q}$, the proposed *multi-scale factorization* computes a sequence of (carefully chosen) rotations $\mathbf{Q}^1, \mathbf{Q}^2, \ldots \mathbf{Q}^L$ to factorize $\mathbf{C}$ in the form

$$\mathbf{C} = (\mathbf{Q}^1)^T (\mathbf{Q}^2)^T \ldots (\mathbf{Q}^L)^T \mathbf{\Lambda} \mathbf{Q}^L \ldots \mathbf{Q}^2 \mathbf{Q}^1,$$

$\mathbf{Q}^\ell$s are sparse $k^{\text{th}}$-order rotations (orthogonal matrices that are the identity except for at most $k$ of their rows/columns), leading to a hierarchical tree-like matrix organization (and if $L = 1$, this is sPCA). This decomposition is referred to as multi-resolution matrix factorization (MMF). In [14], the authors show that this factorization is the first Mallat-style wavelet decomposition on the class of symmetric matrices. It has been shown to be an efficient compression tool [30] and a preconditioner [14]. Clearly this multi-scale factorization involves searching a combinatorial space of row/column indices (i.e., choosing which $k$ candidates suit the best), thereby restricting the order of the rotations to be small (typically, $\leq 3$). Not allowing higher order rotations restricts the richness of the allowable block structure, resulting in a hierarchical decomposition that is "too localized" to be sensible or informative (reverting back to the issues with sPCA and other block low-rank approximations).

Until recently randomized heuristics have been proposed to handle large $k$ and large matrices [15]. In [11], the authors have proposed an incremental version of MMF by observing that many of the computations involved in the exhaustive procedures ([30, 15]) may be by-passed or approximated to a high fidelity. In Sections 2 and 3, we briefly describe this incremental procedure, and in turn show that the generative structure of a symmetric matrix can be represented via an object referred to as *MMF graph*. We show that this MMF graph is a vital tool in answering the questions posed at the beginning of this Section (expanding upon the observations made in [11]). The exhaustive evaluations presented in Section 4 use VGG-S network's deep representations corresponding to multiple ImageNet classes and attributes. The code-base of the proposed factorization is made open-source to further encourage the readers to explore interesting interpretability questions about deep representations, beyond what will be presented in this work.

## 2. Multiresolution Matrix Factorization

**Notation:** We begin with some notation. Matrices are bold upper case, vectors are bold lower case and scalars are lower case. $[m] := \{1, \ldots, m\}$ for any $m \in \mathbb{N}$. Given a matrix $\mathbf{C} \in \mathbb{R}^{m \times m}$ and two set of indices $\mathcal{S}_1 = \{r_1, \ldots r_k\}$ and $\mathcal{S}_2 = \{c_1, \ldots c_p\}$, $\mathbf{C}_{\mathcal{S}_1, \mathcal{S}_2}$ will denote the block of $\mathbf{C}$ cut out by the rows $\mathcal{S}_1$ and columns $\mathcal{S}_2$. $\mathbf{C}_{:,i}$ is the $i^{th}$ column of $\mathbf{C}$. $\mathbf{I}_m$ is the $m$-dimensional identity. $SO(m)$ is the group of $m$ dimensional orthogonal matrices with unit determinant. $\mathcal{R}^m_{\mathcal{S}}$ is the set of $m$-dimensional symmetric matrices which are diagonal except for their $\mathcal{S} \times \mathcal{S}$ block ($\mathcal{S}$–core-diagonal matrices).

Multiresolution matrix factorization (MMF), introduced in [14, 15], retains the locality properties of sPCA while also capturing the global interactions provided by the many variants of PCA, by applying not one, but multiple *sparse* rotation matrices to $\mathbf{C}$ in sequence. We have the following.

**Definition.** Given an appropriate class $\mathcal{O} \subseteq SO(m)$ of sparse rotation matrices, a depth parameter $L \in \mathbb{N}$ and a sequence of integers $m = d_0 \geq d_1 \geq \ldots \geq d_L \geq 1$, the **multi-resolution matrix factorization (MMF)** of a symmetric matrix $\mathbf{C} \in \mathbb{R}^{m \times m}$ is a factorization of the form

$$\mathcal{M}(\mathbf{C}) := \overline{\mathbf{Q}}^T \mathbf{\Lambda} \overline{\mathbf{Q}} \quad \text{with} \quad \overline{\mathbf{Q}} = \mathbf{Q}^L \ldots \mathbf{Q}^2 \mathbf{Q}^1, \tag{1}$$

where $\mathbf{Q}^\ell \in \mathcal{O}$ and $\mathbf{Q}^\ell_{[m]\setminus\mathcal{S}_{\ell-1},[m]\setminus\mathcal{S}_{\ell-1}} = \mathbf{I}_{m-d_\ell}$ for some nested sequence of sets $[m] = \mathcal{S}_0 \supseteq \mathcal{S}_1 \supseteq \ldots \supseteq \mathcal{S}_L$ with $|\mathcal{S}_\ell| = d_\ell$ and $\mathbf{\Lambda} \in \mathcal{R}^m_{\mathcal{S}_L}$.

$\mathcal{S}_{\ell-1}$ is referred to as the 'active set' at the $\ell^{th}$ level, since $\mathbf{Q}^\ell$ is identity outside $[m] \setminus \mathcal{S}_{\ell-1}$. The nesting of the $\mathcal{S}_\ell$s implies that after applying $\mathbf{Q}^\ell$ at some level $\ell$, $\mathcal{S}_{\ell-1} \setminus \mathcal{S}_\ell$ rows/columns are removed from the active set, and are not operated on subsequently. This active set trimming is done at all $L$ levels, leading to a nested subspace interpretation for the sequence of compressions $\mathbf{C}^\ell = \mathbf{Q}^\ell \mathbf{C}^{\ell-1} (\mathbf{Q}^\ell)^T$ ($\mathbf{C}^0 = \mathbf{C}$ and $\mathbf{\Lambda} = \mathbf{C}^L$). In fact, [14] has shown that, for a general class of symmetric matrices, MMF from Definition 2 entails a Mallat style multiresolution analysis (MRA) [24]. Observe that depending on the choice of $\mathbf{Q}^\ell$, only a few dimensions of $\mathbf{C}^{\ell-1}$ are forced to interact, and so the composition of rotations is hypothesized to extract subtle or softer notions of structure in $\mathbf{C}$.

Since multiresolution is represented as matrix factorization here (see (1)), the $\mathcal{S}_{\ell-1} \setminus \mathcal{S}_\ell$ columns of $\overline{\mathbf{Q}}$ correspond to "wavelets". While $d_1, d_2, \ldots$ can be any monotonically decreasing sequence, we restrict ourselves to the simplest case of $d_\ell = m - \ell$. Within this setting, the number of levels $L$ is at most $m - k + 1$, and each level contributes a single wavelet. Given $\mathcal{S}_1, \mathcal{S}_2, \ldots$ and $\mathcal{O}$, the matrix factorization of (1) reduces to determining the $\mathbf{Q}^\ell$ rotations and the residual $\mathbf{\Lambda}$, which is usually done by minimizing the squared Frobenius norm error

$$\min_{\mathbf{Q}^\ell \in \mathcal{O}, \mathbf{\Lambda} \in \mathcal{R}^m_{\mathcal{S}_L}} \quad \|\mathbf{C} - \mathcal{M}(\mathbf{C})\|^2_{\text{Frob}}. \tag{2}$$

The above objective can be decomposed as a sum of contributions from each of the $L$ different levels (see Proposition 1, [14]), which suggests computing the factorization in a greedy manner as $\mathbf{C} = \mathbf{C}^0 \mapsto \mathbf{C}^1 \mapsto \mathbf{C}^2 \mapsto \ldots \mapsto \mathbf{\Lambda}$. This error decomposition is what drives much of the intuition behind our algorithms.

After $\ell - 1$ levels, $\mathbf{C}^{\ell-1}$ is the compression and $\mathcal{S}_{\ell-1}$ is the active set. In the simplest case of $\mathcal{O}$ being the class of so-called $k$–point rotations (rotations which affect at most $k$ coordinates) and $d_\ell = m - \ell$, at level $\ell$ the algorithm needs to determine three things: (a) the $k$–tuple $t^\ell$ of rows/columns involved in the rotation, (b) the nontrivial part $\mathbf{O} := \mathbf{Q}^\ell_{t^\ell,t^\ell}$ of the rotation matrix, and (c) $s^\ell$, the index of the row/column that is subsequently designated a wavelet and removed from the active set. Without loss of generality, let $s^\ell$ be the last element of $t^\ell$. Then the contribution of level $\ell$ to the squared Frobenius norm error (2) is

$$\begin{aligned} \mathcal{E}(\mathbf{C}^{\ell-1}; \mathbf{O}^\ell; t^\ell, s) &= 2 \sum_{i=1}^{k-1} [\mathbf{O} \mathbf{C}^{\ell-1}_{t^\ell,t^\ell} \mathbf{O}^T]^2_{k,i} \\ &+ 2[\mathbf{O}\mathbf{B}\mathbf{B}^T\mathbf{O}^T]_{k,k} \quad \text{where} \quad \mathbf{B} = \mathbf{C}^{\ell-1}_{t^\ell, \mathcal{S}_{\ell-1}\setminus t^\ell}, \end{aligned} \tag{3}$$

and, in the definition of $\mathbf{B}$, $t^\ell$ is treated as a set. The factorization then works by minimizing this quantity in a greedy fashion, i.e.,

$$\begin{aligned} &\mathbf{Q}^\ell, t^\ell, s^\ell \leftarrow \operatorname*{argmin}_{\mathbf{O},t,s} \mathcal{E}(\mathbf{C}^{\ell-1}; \mathbf{O}; t, s) \\ &\mathcal{S}_\ell \leftarrow \mathcal{S}_{\ell-1} \setminus s^\ell \quad ; \quad \mathbf{C}^\ell = \mathbf{Q}^\ell \mathbf{C}^{\ell-1} (\mathbf{Q}^\ell)^T. \end{aligned} \tag{4}$$

## 3. Incremental MMF

We now motivate our algorithm using (3) and (4). Solving (2) amounts to estimating the $L$ different $k$-tuples $t^1, \ldots, t^L$ sequentially. At each level, the selection of the best $k$-tuple is clearly combinatorial, making the exact MMF computation (i.e., explicitly minimizing (2)) very costly even for $k = 3$ or 4 (this has been independently observed in [30]). As discussed in Section 1, higher order MMFs (with large $k$) are nevertheless inevitable for allowing arbitrary interactions among dimensions (see supplement for a detailed study), and our proposed incremental procedure exploits some interesting properties of the factorization error and other redundancies in $k$-tuple computation. The core of our proposal is the following setup.

### 3.1. Overview

Let $\tilde{\mathbf{C}} \in \mathbb{R}^{(m+1)\times(m+1)}$ be the extension of $\mathbf{C}$ by a *single* new column $\mathbf{w} = [\mathbf{u}^T, v]^T$, which manipulates $\mathbf{C}$ as:

$$\tilde{\mathbf{C}} = \left[ \begin{array}{c|c} \mathbf{C} & \mathbf{u} \\ \hline \mathbf{u}^T & v \end{array} \right]. \tag{5}$$

The goal is to compute $\mathcal{M}(\tilde{\mathbf{C}})$. Since $\mathbf{C}$ and $\tilde{\mathbf{C}}$ share all but one row/column (see (5)), if we have access to $\mathcal{M}(\mathbf{C})$, one should, in principle, be able to modify $\mathbf{C}$'s underlying sequence of rotations to construct $\mathcal{M}(\tilde{\mathbf{C}})$. This *avoids* having to recompute everything for $\tilde{\mathbf{C}}$ from scratch, i.e., performing the greedy decompositions from (4) on the entire $\tilde{\mathbf{C}}$.

The hypothesis for manipulating $\mathcal{M}(\mathbf{C})$ to compute $\mathcal{M}(\tilde{\mathbf{C}})$ comes from the precise computations involved in the factorization. Recall (3) and the discussion leading up to the expression. At level $\ell + 1$, the factorization picks the 'best' candidate rows/columns from $\mathbf{C}^\ell$ that correlate the most with each other, so that the resulting diagonalization induces the smallest possible off-diagonal error over the rest of the active set. The components contributing towards this error are driven by the inner products $(\mathbf{C}^\ell_{:,i})^T\mathbf{C}^\ell_{:,j}$ for some columns $i$ and $j$. In some sense, the largest such correlated rows/columns get picked up, and adding one new entry to $\mathbf{C}^\ell_{:,i}$ may not change the *range* of these correlations. Extending this intuition across all levels, we argue that

$$\underset{i,j}{\operatorname{argmax}}\, \tilde{\mathbf{C}}^T_{:,i}\tilde{\mathbf{C}}_{:,j} \approx \underset{i,j}{\operatorname{argmax}}\, \mathbf{C}^T_{:,i}\mathbf{C}_{:,j}. \tag{6}$$

Hence, the $k$-tuples computed from $\mathbf{C}$'s factorization are reasonably good candidates even after introducing $\mathbf{w}$. To better formalize this idea, and in the process present our algorithm, we parameterize the output structure of $\mathcal{M}(\mathbf{C})$ in terms of the sequence of rotations and the wavelets.

### 3.2. The graph structure of $\mathcal{M}(\mathbf{C})$

If one has access to the sequence of $k$-tuples $t^1, \ldots, t^L$ involved in the rotations and the corresponding wavelet indices $(s^1, \ldots, s^L)$, then the factorization is straightforward to compute i.e., there is no greedy search anymore. Recall that by definition $s^\ell \in t^\ell$ and $s^\ell \notin \mathcal{S}_\ell$ (see (4)). To that end, for a given $\mathcal{O}$ and $L$, $\mathcal{M}(\mathbf{C})$ can be 'equivalently' represented using a depth $L$ *MMF graph* $\mathcal{G}(\mathbf{C})$. Each level of this graph shows the $k$-tuple $t^\ell$ involved in the rotation, and the corresponding wavelet $s^\ell$ i.e., $\mathcal{G}(\mathbf{C}) := \{t^\ell, s^\ell\}_1^L$. Interpreting the factorization in this way is notationally convenient for presenting the algorithm. More importantly, such an interpretation is central for visualizing hierarchical dependencies among dimensions of $\mathbf{C}$, and will be further discussed in Section 4. An example of such a $3^{rd}$ order MMF graph constructed from a $5 \times 5$ matrix is shown in Figure 2(a) (the rows/columns are color coded for better visualization). At level $\ell = 1$, $s_1$, $s_2$ and $s_3$ are diagonalized while designating the rotated $s_1$ as the wavelet (the black ellipse). The dashed arrow from $s_1$ indicates that it is not involved in future rotations. This process repeats for $\ell = 2$ and 3, and as shown by the color-coding of different compositions, MMF gradually teases out *higher-order* correlations that can only be revealed after composing the rows/columns at one or more scales (levels here). Figure 2(b) shows the *visualization* of the resulting MMF graph – each ellipse is a level and the black lines simply indicate interactions. To avoid clutter as $\ell$ increases, lines are only shown from higher level categories ($s_4$ and $s_5$ here) to lower level ellipses (and not $s_1$, $s_2$ and $s_3$).

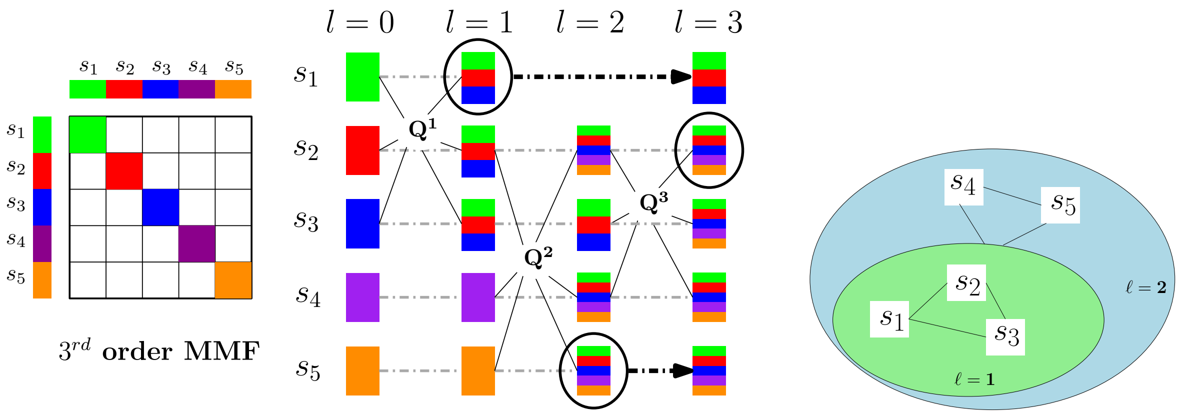
For notational convenience, we denote the MMF graphs of $\mathbf{C}$ and $\tilde{\mathbf{C}}$ as $\mathcal{G} := \{t^\ell, s^\ell\}_1^L$ and $\tilde{\mathcal{G}} := \{\tilde{t}^\ell, \tilde{s}^\ell\}_1^{L+1}$. Recall that $\tilde{\mathcal{G}}$ will have one more level than $\mathcal{G}$ since the row/column $\mathbf{w}$, indexed $m+1$ in $\tilde{\mathbf{C}}$, is being added (see (5)). The goal is to estimate $\tilde{\mathcal{G}}$ without recomputing all the $k$-tuples using the greedy procedure from (4). This translates to *inserting* the new index $m+1$ into the $t^\ell$s and modifying $s^\ell$s accordingly. Following the discussion from Section 3.1, incremental MMF argues that inserting this one new element into the graph will not result in global changes in its topology. Clearly, in the pathological case, $\mathcal{G}$ may change arbitrarily, but as argued earlier (see discussion about (6)) the chance of this happening for non-random matrices with reasonably large $k$ is small. The core operation then is to compare the *new* $k$-tuples resulting from the addition of $\mathbf{w}$ to the best ones from $[m]^k$ provided via $\mathcal{G}$. If the newer $k$-tuple gives better error (see (3)), then it will *knock out* an existing $k$-tuple. This constructive insertion and knock-out procedure is the incremental MMF.

### 3.3. The Incremental MMF

The basis for this incremental procedure is that one has access to $\mathcal{G}$ (i.e., MMF on $\mathbf{C}$). We first present the algorithm assuming that this "initialization" is provided, and revisit this aspect shortly. The procedure starts by setting $\tilde{t}^\ell = t^\ell$ and $\tilde{s}^\ell = s^\ell$ for $\ell = 1, \ldots, L$. Let $\mathcal{I}$ be the set of elements (indices) that needs to be inserted into $\mathcal{G}$. At the start (the first level) $\mathcal{I} = \{m+1\}$ corresponding to $\mathbf{w}$. Let $\tilde{t}^1 = \{p_1, \ldots, p_k\}$. The new $k$-tuples that account for inserting entries of $\mathcal{I}$ are $\{m+1\} \cup t^1 \setminus p_i$ $(i = 1, \ldots, k)$. These new $k$ candidates are the probable alternatives for the existing $\tilde{t}^1$. Once the best among these $k+1$ candidates is chosen, an existing $p_i$ from $\tilde{t}^1$ may be knocked out.

If $\tilde{s}^1$ gets knocked out, then $\mathcal{I} = \{\tilde{s}^1\}$ for future levels. This follows from MMF construction, where wavelets at $\ell^{th}$ level are not involved in later levels. Since $\tilde{s}^1$ is knocked out, it is the new inserting element according to $\mathcal{G}$. On the other hand, if one of the $k-1$ scaling functions is knocked out, $\mathcal{I}$ is not updated. This simple process is repeated sequentially from $\ell = 1$ to $L$. At $L+1$, there are no estimates for $\tilde{t}^{L+1}$ and $\tilde{s}^{L+1}$, and so, the procedure simply selects the best $k$-tuple from the remaining active set $\tilde{\mathcal{S}}_L$. This insertion and knock-out procedure is for the setting from (5) where one extra row/column is added to a given MMF, and clearly, the incremental procedure can be repeated as more and more rows/columns are added, thereby estimating the factorization for large (possibly dense) matrices.

The overall procedure will then have two components: an initialization on some randomly chosen small block (of size $\tilde{m} \times \tilde{m}$) of the entire matrix $\mathbf{C}$; followed by insertion of the remaining $m - \tilde{m}$ rows/columns in a streaming fashion (similar to $\mathbf{w}$ from (5)). The initialization entails computing a batch-wise MMF on this small block ($\tilde{m} \geq k$). Note that whenever $\tilde{m}$ is reasonably small ($< 10$ or so), one can exhaustively compute the MMF using the classical algorithms from [14, 15]. Although there are approximate faster procedures as well for computing this initialization. For brevity, the precise algorithms and the different types of initializa-




(a) **Example Construction** (b) **MMF graph Visualization**

Figure 2. (a) An example $5 \times 5$ matrix, and its $3^{rd}$ order MMF graph (better in color). $\mathbf{Q}^1$, $\mathbf{Q}^2$ and $\mathbf{Q}^3$ are the rotations. $s_1$, $s_5$ and $s_2$ are wavelets (marked as black ellipses) at $l = 1$, 2 and 3 respectively. The arrows imply that a wavelet is not involved in future rotations. (b) The corresponding MMF graph visualization (used in Section 4).

tions that one can explore, are not presented below. We instead direct the readers to Algorithms 1 and 2 from [11], where we also show exhaustive evidence that this incremental procedure scales efficiently for very large matrices (while recovering good factorization, in terms of the MMF loss from (2)), compared to using the batch-wise scheme on the entire matrix.

## 4. Experiments

We demonstrate the ideas presented above by constructing MMF graphs, and interpreting them, using task-specific representations learned by VGG-S network [5]. Multiple different ImageNet classes and attributes are used. The evaluation setup involves computing MMFs on class covariance matrices. Let $m$ be the number of classes/attributes we are interested in, and $\mathbf{C} \in \mathbb{R}^{m \times m}$ be the class covariance matrix. For each class $i$, let $p_1, \ldots, p_{|i|}$ denote the indices of the data instances belonging to class $i$ (For large classes at most a randomly chosen 2k instances are used). Each entry of $\mathbf{C}$ is

$$\mathbf{C}_{i,j} = \frac{1}{|i||j|} \sum_{\hat{i}=p_1}^{p_i} \sum_{\hat{j}=q_1}^{q_j} \mathbf{d}_{\hat{i}}^T \mathbf{d}_{\hat{j}} \tag{7}$$

where $\mathbf{d}_{\hat{i}}$ is the hidden representation of $\hat{i}^{th}$ data instance coming from some hidden layer of the given deep network. MMF graphs would then be constructed on $\mathbf{C}$ as described in Sections 3.2 and 3.3.

### 4.1. Decoding the deep

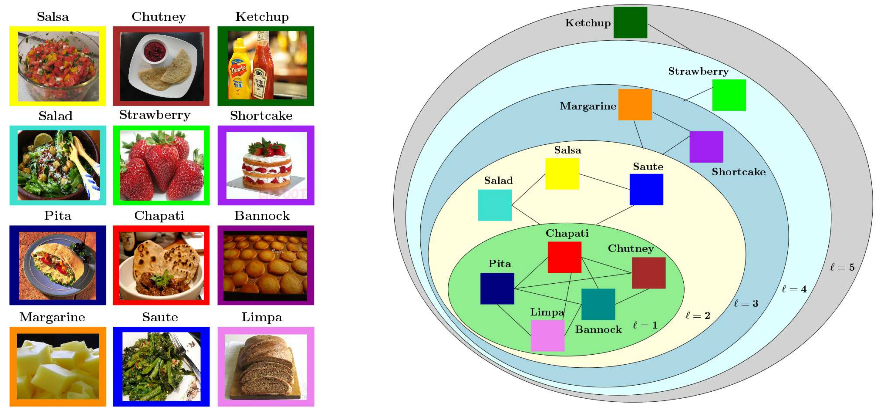
To precisely walk through the power of MMF graphs, consider the last hidden layer (FC7, that feeds into softmax) representations from a VGG-S network [5] corresponding to 12 different ImageNet classes, shown in Figure 3(a). Figure 3(b) visualizes a $5^{th}$ order MMF graph learned on this class covariance matrix.

**The semantics of breads and sides.** The $5^{th}$ order MMF says that the five categories – *pita*, *limpa*, *chapati*, *chutney* and *bannock* – are most representative of the localized structure in the covariance. Observe that these are four different flour-based main courses, and a side *chutney* that shared strongest context with the images of *chapati* in the training data (similar to the body building and dumbell images from [1]). MMF then picks *salad*, *salsa* and *saute* representations' at the $2^{nd}$ level, claiming that they relate the strongest to the *composition* of breads and *chutney* from the previous level (see visualization in Figure 3(b)). Observe that these are in fact the sides offered/served with bread. Although VGG-S was *not* trained to predict these relations, according to MMF, the representations are inherently learning them anyway – a fascinating aspect of deep networks i.e., they are seeing what humans may infer about these classes.

**Any dressing? What are my dessert options?** Let us move to the $3^{rd}$ level in Figure 3(b). *margarine* is a cheese based dressing. *shortcake* is dessert-type meal made from *strawberry* (which shows up at $4^{th}$ level) and bread (the composition from previous levels). That is the full course. The last level corresponds to *ketchup*, which is an outlier, distinct from the rest of the 10 classes – a typical order of dishes involving the chosen breads and sides does not include hot sauce or ketchup. Although *shortcake* is made up of strawberries, "conditioned" on the $1^{st}$ and $2^{nd}$ level dependencies, it is less useful in summarizing the covariance structure. An interesting summary of this hierarchy from Figure 3(b) is – an order of *pita* with side *ketchup* or *strawberries* is atypical in the data seen by these networks.

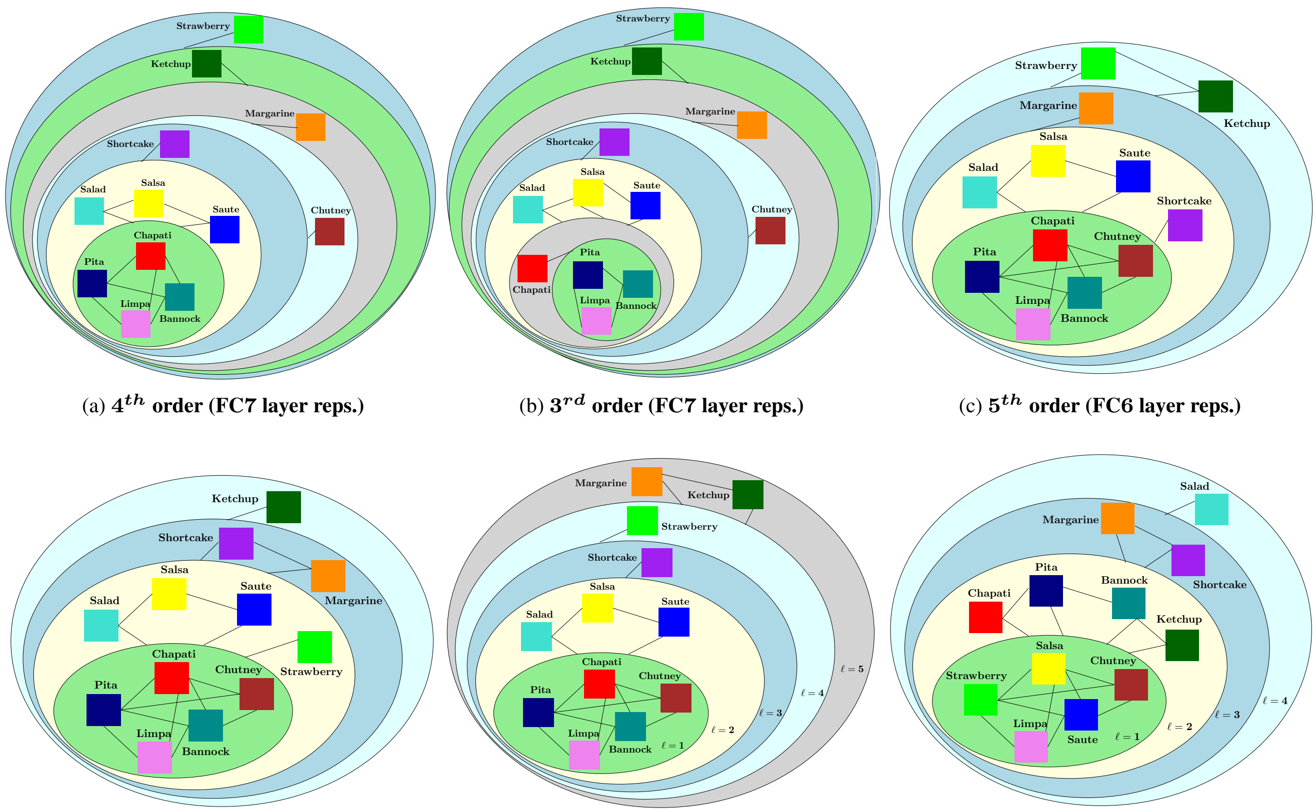
### 4.2. Are we reading tea leaves?

The networks are *not* trained to learn the hierarchy of categories (unlike the alternate works from [33, 7, 2]) – the




(a) **12 classes** (b) **$5^{th}$ order (FC7 layer reps.)**

Figure 3. **Hierarchy and Compositions of VGG-S [5] representations inferred by $5^{th}$ order MMF.** (a) The **12** classes, (b) Hierarchical structure.




(a) **$4^{th}$ order (FC7 layer reps.)** (b) **$3^{rd}$ order (FC7 layer reps.)** (c) **$5^{th}$ order (FC6 layer reps.)**

(d) **$5^{th}$ order (conv5 layer reps.)** (e) **$5^{th}$ order (conv3 layer reps.)** (f) **$5^{th}$ order (Pixel reps.)**

Figure 4. **Hierarchy and Compositions of VGG-S [5] representations inferred by MMF.** (a,b) Hierarchical structure from $4^{th}$ and $3^{rd}$ order MMF, (c-f) the structure from a $5^{th}$ order MMF on other VGG-S layers.

task was object/class detection. Hence, the relationships are completely a by-product of the power of deep networks to learn contextual information, *and* the ability of MMF to model these compositions by uncovering the structure in the covariance matrix. Nevertheless it is reasonable to ask if this description is meaningful since the semantics drawn above are subjective. We provide explanations below.

**Choice of the order:** The first aspect that one may ask is if the compositions are sensitive/stable to the order $k$ – a critical hyperparameter of MMF. Figure 4(a,b) uses a $4^{th}$ and $3^{rd}$ order MMF respectively, and the resulting hierarchies are similar to that from Figure 3(b). Specifically, the differ-

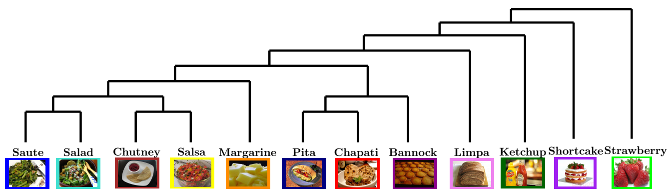



Figure 5. Agglomerative clustering (Dendrogram) constructed on VGG-S FC7 representations.

ent breads and sides show up early, and the most distinct categories (*strawberry* and *ketchup*) appear at the higher levels.

**Changing the hidden layer:** If the class relationships from Figures 3 and 4(a,b) are non-spurious, then similar trends should be implied by MMF's on different (higher) layers of VGG-S. Figure 4(c–e) shows the compositions from the FC6, $5^{th}$ and $3^{rd}$ convolutional layer representations of VGG-S for the 12 classes in Figure 3(a). The strongest compositions, the 8 classes from $\ell = 1$ and 2, are already picked up half-way thorough the VGG-S, providing further evidence that the compositional structure implied by MMF is data-driven. We further discuss this in Section 4.3. Overall, the MMF graphs from Figures 3 and 4 show many of the summaries that a human may infer about the 12 classes in Figure 3(a).

**Baseline comparison:** We compared MMF's class-compositions to the hierarchical clusters obtained from agglomerative clustering of representations shown in Figure 5. The categories like *bannok* and *chapati*, or *saute* and *salad* are closer to each other here. However, the compositional relationships (like the dependency of *chutney*/*salsa*/*salad* on several breads, or the disparity of *ketchup* from others) shown in Figures 3(b) and 4, especially in the $FC7$ and $FC6$ layers, are not apparent in these dendrogram outputs.

## 4.3. The flow of MMF graphs: A tool for designing deep networks

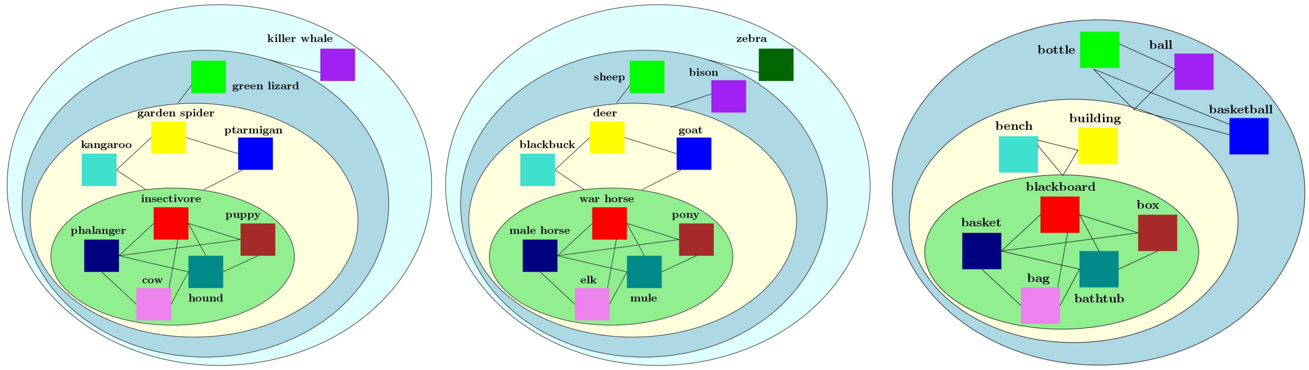
Figure 4(f) shows the compositions from the $5^{th}$ order MMF on the *input* (pixel-level) data. Pixel-level features are non-informative (the success of deep learning is a direct evidence for this), and clearly, the classes whose RGB values/signals correlate are at $l = 0$ in Figure 3(f). Most importantly, comparing Figure 3(b) to Figures 4(c–e), which are from the FC6, $5^{th}$ and $3^{rd}$ convolutional layer respectively, we see that $l = 1$ and 2 have the same compositions. This says that the strongest possible representative classes may be learned in lower layers of the network. Visualizations like Figure 3(b) followed by Figures 4(c–f) are a powerful exploratory tool in decoding, and possibly, designing deep networks [12] – providing solutions to our original motivation of interpretability of deep representations. We here point out few such aspects.

Firstly, visualizations like these can be constructed for all the layers of the network leading to a trajectory of class compositions. This is driven by the saturation of the compositions – if the class compositions from last few layers do not change, then the additional layer may be removed, thereby reducing the parameters to be trained (and in turn improving the optimization). The saturation at $l = 1, 2$ in Figure 3(b) and Figure 4(c,d) is one such example. If these 8 classes are a priority, then the predictions of the VGG-S' $3^{rd}$ convolutional layer may already be *good enough*. On the other hand, if there is large variance among the compositions, especially in lower MMF graph levels, then the lower layers of the deep network may not have efficiently accounted for the strongest representative classes. This implies that the hidden layer lengths or activations for lower network layers need to be modified. A smaller variance however may imply that adding additional layer may be beneficial. Such constructs can be tested across other layers and architectures (see supplement for MMFs from AlexNet, VGG-S and other networks).

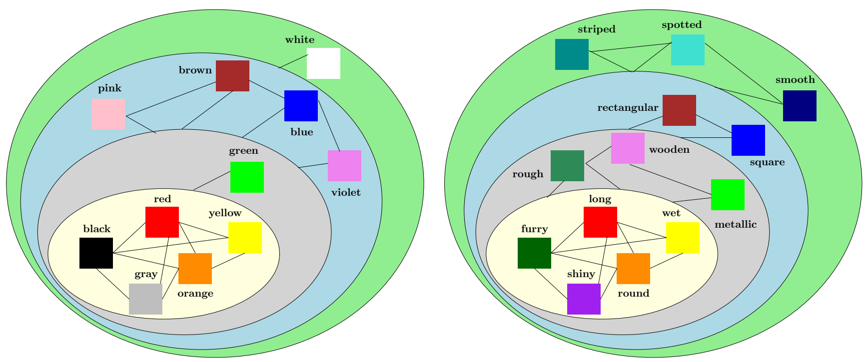
### 4.3.1 Alternate Classes

To further explore the interpretations of MMF graphs on ImageNet representations, we constructed $\mathbf{5}^{th}$ order MMFs on other families of classes from ImageNet as shown in Figure 6. Figure 6(a) contains $\mathbf{10}$ different versatile set of classes i.e., the contextual information (like background, typical sets of objects that are found etc.) for *cow* or *hound* may be drastically different from *kangaroo* or *ptarmigan*. Clearly, the most distinct classes – *green lizard*, *killer whale* are resolved at highest levels. The composition at first level involves those classes which share context – there is most certainly grass or trees on the bark in these images. Given the first level, Figure 6(a) shows that *kangaroo* rather than *green lizard* is most related to this first level composition – implying a clear sense of hierarchy in learned representations. The categories in Figure 6(b) that MMF picked for first level compositions are highly correlated to begin with, both with respect to the texture/pixel-values as well as the background (most show up with grassy or landscape background). It is reasonable to say that human perception would imply that *blackbuck*, *deer*, *goat* are most related to the first level compositions – which were composed at the next level. *zebra* clearly is the most distinct one, implying that many compositions of the first and second level categories may not really infer what constitutes a *zebra*.

In another example, $\mathbf{8}$ of the $\mathbf{12}$ classes from Figure 6(c) are sports accessories/objects and the rest (*pot*, *couch*, *sheet*, *toilet seat*) are more of household-type categories. This was done to see what the MMF will do, if it is forced to pick hierarchy among arbitrary classes. Clearly, *pot* shows up at last level. The first level compositions are $\mathbf{5}$ of the sports related classes. Interestingly *couch* is closer in composi-




(a) **Animal Classes (FC7 layer reps.)** (b) **Correlated Classes (FC7 layer reps.)** (c) **Another Example (FC7 layer reps.)**




(d) **color attributes (FC7 layer reps.)** (e) **shape attributes (FC7 layer reps.)**

Figure 6. **Hierarchy and Compositions of VGG-S [5] reps inferred by $5^{th}$ order MMF.** (a–c) Alternate classes, (d,e) color and shape attributes.

tion to the first level than a *roller coaster* or *sail*, which may be because the context of four of the first level classes is that they are 'indoors'. *sail* and *roller coaster* have sky and water as background respectively, making them distinct enough from the first level composition, and pushing *couch* or *toilet seat* instead.

### 4.3.2 Colors and Shape Attributes

The VGG-S network used in the evaluations has been trained on the ImageNet categories. If the networks are as powerful as we hope them to be, then this class-wise trained network should capture interesting, human-releatable, compositions for meta-class attributes like *color* or *shape*. Figure 6(d,e) show the MMF graphs from FC7 representations for these *color* and *shape* attributes. This most distinctive of the colors (like *red*, *yellow*) are picked up at lowest level. Bright colors like *pink* and *violet* which are in fact compositions of basis RGB colors are at higher levels of the MMF graph. Similarly, visually the most evident shapes like *furry* and *shiny* form lower level composition, while most subtle shapes like *smooth* and *spotted* are at the last level.

Apart from visualizing deep representations, these evaluations show that MMF graphs are vital exploratory tools for category/scene understanding from unlabeled representations in transfer and multi-domain learning [3]. This is because, by comparing the MMF graph *prior* to inserting the new unlabeled instance to the one *after* insertion, one can infer whether the new instance contains non-trivial information that cannot be expressed as a composition of existing categories.

## 5. Conclusions

Bridging matrix decomposition and harmonic analysis, we proposed a multi-resolution matrix factorization scheme that models symmetric matrices. We showed that this factorization uncovers the class compositions learned in higher layers of deep networks. Beyond addressing whether semantics of deep representations are similar to those inferred by the humans, we showed that this tool can also help designing and debugging/adapting an existing network (e.g., checking the goodness of lower layers). The proposed tool is made open-source for further exploration.

**Acknowledgments:** The author thanks Risi Kondor and Vikas Singh. The work is supported by NIH AG021155, EB022883, AG040396, NSF CAREER 1252725, NSF AI117924 and 1320344/1320755.

## References


[1] Inceptionism: Going deeper into neural networks. 2015. 1, 5

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 5

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 8

[4] S. Chandrasekaran, M. Gu, and W. Lyons. A fast adaptive solver for hierarchically semiseparable representations. *Calcolo*, 42(3-4):171–185, 2005. 2

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 2, 5, 6, 8

[6] K. A. DeJong and W. M. Spears. Learning concept classification rules using genetic algorithms. Technical report, DTIC Document, 1990. 1

[7] Y. Dong, H. Su, J. Zhu, and B. Zhang. Improving interpretability of deep neural networks with semantic information. *arXiv preprint arXiv:1703.04096*, 2017. 1, 5

[8] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. *arXiv preprint arXiv:1506.02753*, 2015. 1

[9] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010. 1

[10] D. Erhan, A. Courville, and Y. Bengio. Understanding representations learned in deep architectures. *Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep*, 1355, 2010. 1

[11] V. K. Ithapu, R. Kondor, S. C. Johnson, and V. Singh. The incremental multiresolution matrix factorization algorithm. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 2, 5

[12] V. K. Ithapu, S. N. Ravi, and V. Singh. On architectural choices in deep learning: From network structure to gradient convergence and parameter estimation. *arXiv preprint arXiv:1702.08670*, 2017. 7

[13] V. K. Ithapu, V. Singh, O. C. Okonkwo, et al. Imaging-based enrichment criteria using deep learning algorithms for efficient clinical trials in mild cognitive impairment. *Alzheimer's & Dementia*, 11(12):1489–1499, 2015. 1

[14] R. Kondor, N. Teneva, and V. Garg. Multiresolution matrix factorization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1620–1628, 2014. 2, 3, 4

[15] R. Kondor, N. Teneva, and P. K. Mudrakarta. Parallel mmf: a multiresolution approach to matrix computation. *arXiv preprint arXiv:1507.04396*, 2015. 2, 3, 4

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[17] H. Lakkaraju and J. Leskovec. Confusions over time: An interpretable bayesian model to characterize trends in decision making. In *Advances in Neural Information Processing Systems*, pages 3261–3269, 2016. 1

[18] H. Lakkaraju and C. Rudin. Learning cost-effective and interpretable treatment regimes. In *Artificial Intelligence and Statistics*, pages 166–175, 2017. 1

[19] B. Letham, C. Rudin, T. H. McCormick, D. Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015. 1

[20] Y. Liang, M.-F. F. Balcan, V. Kanchanapally, and D. Woodruff. Improved distributed principal component analysis. In *Advances in Neural Information Processing Systems*, pages 3113–3121, 2014. 2

[21] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016. 1

[22] Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158. ACM, 2012. 1

[23] D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. ACM, 2005. 1

[24] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989. 3

[25] R. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, volume 328, page 334, 1999. 1

[26] J. C. Peterson, J. T. Abbott, and T. L. Griffiths. Adapting deep network features to capture psychological representations. *arXiv preprint arXiv:1608.02164*, 2016. 1

[27] B. Savas, I. S. Dhillon, et al. Clustered low rank approximation of graphs in information science applications. In *SDM*, pages 164–175. SIAM, 2011. 2

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[29] Y. Su and F. Jurie. Improving image classification using semantic attributes. *International journal of computer vision*, 100(1):59–77, 2012. 1

[30] N. Teneva, P. K. Mudrakarta, and R. Kondor. Multiresolution matrix compression. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1441–1449, 2016. 2, 3

[31] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2013. 1

[32] T. Wang, C. Rudin, F. Doshi, Y. Liu, E. Klampfl, and P. MacNeille. Bayesian ors of ands for interpretable classification with application to context aware recommender systems, 2015. 1

[33] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2740–2748, 2015. 5

[34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 1

[35] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 1

[36] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[37] T. Zhang, B. Ghanem, and N. Ahuja. Robust multi-object tracking via cross-domain contextual information for sports video analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 985–988. IEEE, 2012. 1

[38] Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592–617, 2013. 2

[39] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006. 2