Constructive sampling for patch-based embedding

Moshe Salhov, Guy Wolf, Amit Bermanis and Amir Averbuch
School of Computer Science, Tel Aviv University
Tel Aviv 69978, Israel

***Abstract*—To process high-dimensional big data, we assume that sufficiently small patches (or neighborhoods) of the data are approximately linear. These patches represent the tangent spaces of an underlying manifold structure from which we assume the data is sampled. We use these tangent spaces to extend the *scalar* relations that are used by many kernel methods to *matrix* relations, which encompass multidimensional similarities between local neighborhoods in the data. The incorporation of these matrix relations improves the utilization of kernel-based data analysis methodologies. However, they also result in a larger kernel and a higher computational cost of its spectral decomposition. We propose a dictionary construction that approximates the oversized kernel in this case and its associated patch-to-tensor embedding. The performance of the proposed dictionary construction is demonstrated on a super-kernel example that utilizes the Diffusion Maps methodology together with linear-projection operators between tangent spaces in the manifold.**

I. Introduction

Recent methods for massive high dimensional data analysis utilize a manifold structure on which data points are assumed to lie. This manifold is immersed (or submersed) in an ambient space that is defined by observable parameters. Kernel methods such as Diffusion Maps (DM) [1] have provided good results in analyzing such massive high dimensional data. These methods are based on the spectral decomposition of a kernel designed to incorporate scalar similarities between data points. The resulting embedding of the data points into an Euclidean space preserves the qualities represented by the designed kernel.

Recently, DM was extended in several different ways to handle the orientation in local tangent spaces [2]–[4]. The relation between two patches is described by a matrix instead of a scalar value. The resulting kernel captures enriched similarities between local structures in the underlying manifold. These enriched similarities can be used to analyze local areas around data points instead of analyzing their specific locations.

The discussed enrichments increase considerably the kernel size, which is a limiting factor in the applicability of kernel methods to real problems. Considerable efforts have been invested for example in [5], [6] and others in approximating the spectral decomposition operator to become computationally feasible. In this paper, we combine the patch-based embedding from [3], [4] with the dictionary construction approach in [5] to approximate the spectral decomposition of a non-scalar kernel that utilizes the underlying patch structure inside the ambient space.

II. Problem setup

Let $\mathcal{M}$ be a d dimensional manifold that lies in the ambient space $\mathbb{R}^m$, where $d \ll m$, and let $M \subseteq \mathbb{R}^m$ be a set of n points sampled from it. Each point $x \in M$ has a d-dimensional tangent space $T_x(\mathcal{M})$, which is a subspace of $\mathbb{R}^m$. Let $O_x \in \mathbb{R}^{m\times d}$, $x \in M$, be a matrix whose columns $o_x^1, \ldots, o_x^d \in \mathbb{R}^m$ form an orthonormal basis of $T_x(\mathcal{M})$. If the manifold is densely sampled, $T_x(\mathcal{M})$ can be approximated by a small enough patch $N(x) \subseteq M$ around $x \in M$. We will assume that $o_x^1, \ldots, o_x^d$ are the principal directions of $N(x)$ and vectors in $T_x(\mathcal{M})$ are expressed according to this basis.

A. Diffusion Maps

The original diffusion maps method [1] is based on defining an isotropic kernel K as $k(x,y) \triangleq e^{-\frac{\|x-y\|}{\varepsilon}}$, for every $x, y \in \mathcal{M}$, where ε is a meta-parameter of the algorithm. This kernel represents the affinities between points on the manifold. The kernel is normalized with the degrees $q(x) \triangleq \int_{y\in\mathcal{M}} k(x,y)$, $x \in \mathcal{M}$ to produce a stochastic transition operator P, with $p(x,y) = \frac{k(x,y)}{q(x)}$, which defines a Markov process (i.e., a diffusion process) over the manifold $\mathcal{M}$. Its symmetric conjugate A, where $a(x,y) = \sqrt{q(x)}p(x,y)\frac{1}{\sqrt{q(y)}} = \frac{k(x,y)}{\sqrt{q(x)q(y)}}$, defines the diffusion affinities between data-points. Spectral analysis of the diffusion affinity kernel A yields the eigenvalues $1 = \sigma_0 \geq \sigma_1 \geq \ldots$ and their corresponding eigenvectors $\psi_0, \psi_1, \ldots$, which are used to construct the desired map that embeds each data point $x \in \mathcal{M}$ onto the point $\Psi(x) = (\sigma_i \psi_i(x))_{i=0}^{\delta}$ for a sufficiently small δ, which is the dimension of the embedded space and depends on the decay of the spectrum of A.

III. Super-kernel

For $x, y \in M$, let $O_{xy} = O_x^T O_y \in \mathbb{R}^{d\times d}$, where O_x and O_y represent bases of the tangent spaces $T_x(\mathcal{M})$ and $T_y(\mathcal{M})$, respectively. The matrix O_{xy} represents a linear-projection between these tangent spaces, and, in some sense, the similarity between them. We will refer to it as a tangent similarity matrix. We use the diffusion affinity kernel A and the tangent similarity matrices O_{xy} to define the following *super-kernel*:

Definition 1. *A super-kernel is a block matrix $G \in \mathbb{R}^{nd\times nd}$ with $n \times n$ blocks and each block in it is a $d \times d$ matrix. Each block $G_{xy} \in \mathbb{R}^{d\times d}$ of a* Linear-Projection Diffusion Super-kernel *is defined as $G_{xy} \triangleq a(x,y)O_{xy} = a(x,y)O_x^T O_y$, $x, y \in M$ and represents the affinity between the patches $N(x)$ and $N(y)$.*

We will use spectral decomposition for analyzing a super-kernel G, and utilize it to embed the patches $N(x)$ of the manifold (for $x \in M$) into a tensor space. Let $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_\ell|$ be the ℓ most significant eigenvalues of G and let $\phi_1, \phi_2, \ldots, \phi_\ell$ be their corresponding eigenvectors. According to the spectral theorem, if ℓ is greater than the numerical rank of G, then $G \approx \sum_{i=1}^{\ell} \lambda_i \phi_i \phi_i^T$, where the eigenvectors are treated as column vectors.

Each eigenvector ϕ_i, $i = 1, \ldots, \ell$, is a vector of length nd. We denote each of its elements as $\phi_i(o_x^j)$ where $x \in M$ and $j = 1, \ldots, d$. An eigenvector ϕ_i can also be regarded as a vector of n sections, each of which is a vector of length d that corresponds to a point $x \in M$ on the manifold. To express this notion we use the notation $\varphi_i^j(x) = \phi_i(o_x^j)$ (for $x \in M, i = 1, \ldots, \ell, j = 1, \ldots, d$). Thus, the section in ϕ_i, which corresponds to $x \in M$, is the vector $(\varphi_i^1(x), \ldots, \varphi_i^d(x))^T$.

We use the eigenvalues and eigenvectors of G to construct a spectral map whose definition is similar to the standard (i.e., classic) diffusion map: $\Phi(o_x^j) = (\lambda_1 \phi_1(o_x^j), \ldots, \lambda_\ell \phi_\ell(o_x^j))^T$. By using this construction, we get nd vectors of length ℓ. Each $x \in M$ corresponds to d of these vectors, i.e., $\Phi(o_x^j)$, $j = 1, \ldots, d$. We use these vectors to construct the tensor $\mathcal{T}_x \in \mathbb{R}^\ell \otimes \mathbb{R}^d$ for each $x \in M$, whose coordinates are $[\mathcal{T}_x]_{ij} = \lambda_i \varphi_i^j(x)$, $x \in M, i = 1, \ldots, \ell, j = 1, \ldots, d$. Each tensor $\mathcal{T}_x$ represents an embedding of the patch $N(x)$, $x \in M$, into the tensor space $\mathbb{R}^\ell \otimes \mathbb{R}^d$.

A. Mathematical properties

1) Spectral properties: The linear-projection operators, which define the tangent similarity matrices by a LPD super-kernel, express some important properties of the manifold structure, e.g., curvatures between patches and differences in orientation. While there might be other ways to construct a super-kernel that expresses these properties, LPD super-kernels do have an important property, which is given by the following theorem:

Theorem 1. *A LPD super-kernel G is positive definite and its operator norm satisfies $\|G\| \leq 1$.*

Proof. Theorem 3.1 from [3] shows that linear-projection super-kernels have a non-negative spectrum that is bounded from above by the spectral norm of the used scalar affinities. Following the footsteps of that proof in our case, with the diffusion affinity kernel, which is positive definite and whose spectral norm is one, yields the result in the theorem. □

The patch-to-tensor embedding that is achieved by the LPD super-kernel is defined by the spectral analysis of this super-kernel. Therefore, the spectral properties of this super-kernel, which are shown in Theorem 1, are crucial for the patch-based data analysis that utilizes this embedding.

2) Embedded distances: The classical diffusion map provides an embedded space in which the Euclidean distance between data points is equal to a diffusion distance in the original ambient space. This diffusion distance measures the distance between two diffusion "bumps" $a(x, \cdot)$ and $a(y, \cdot)$, each of which is a row in the symmetric diffusion kernel that defines the diffusion map. From a technical point of view, this relation means that the Euclidean distance between two arbitrary points in the range of a diffusion map is equal to the Euclidean distances between the corresponding rows of its symmetric diffusion kernel. The following theorem (whose proof appears in [3]) shows a similar property of the LPD-based patch-to-tensor embedding:

Theorem 2. *Let $x, y \in M$ be two points on the manifold and let $\mathcal{T}_x$ and $\mathcal{T}_y$ be their embedded tensors, then $\|\mathcal{T}_x - \mathcal{T}_y\|_F^2 = \sum_{z \in M} \sum_{j=1}^{d} \|(a(x,z)O_x^T - a(y,z)O_y^T)o_z^j\|^2$, where the tensors are treated as matrices (i.e., their coordinate matrices) when computing the Frobenius distance between them.*

The vectors o_z^j in Theorem 2 are unit vectors that form an orthonormal basis of the tangent space $T_x(\mathcal{M})$ at the point $z \in M$. For each point $z \in M$, the matrix $[a(x,z)O_x^T - a(y,z)O_y^T]$ is applied to each of these unit vectors and the squared lengths of the resulting vectors are summed. These terms can be seen as extensions of the terms $(a(x,z) - a(y,z))$ of the original diffusion distance, which only consider the differences between scalar affinities. Further explanations about the meaning of the extended diffusion distance can be found in [3].

IV. Out-of-Sample Extension for Vector Fields

The presented patch-to-tensor embedding is based on the spectral analysis of a large super-kernel G. In order to approximate this spectral decomposition, we will use a dictionary (i.e., a set of representatives) and extend its results (using an out-of-sample extension) to the entire dataset. This extension method can also be utilized to extend this decomposition either from the dictionary or from the dataset to new data points. The super-kernel G can be regarded as an operator on tangent vector fields of the manifold $\mathcal{M}$ restricted to a dataset M. Therefore, the spectral decomposition of G consists of eigenvector fields that span the range of G. Hence, an out-of-sample extension of the eigenvector fields is equivalent to the out-of-sample extension of vector fields in the range of G.

Out-of-sample extension of vector fields assumes an a priori knowledge of a set of data points M and a corresponding vector field where each vector lies on the respective local tangent space. Consider a tangent vector field $\vec{v} : M \to \mathbb{R}^d$ such that $\vec{v}(x) \in T_x(\mathcal{M})$ for all $x \in M$. Then, the given data points are used to construct the super-kernel G. Since G is positive definite (see Theorem 1), it is also invertible and its range consists of all these vector fields.

The out-of-sample extension of a new data point under the PTE settings aims to find the new corresponding vector in the local tangent space of the new point. The extension coefficients $\vec{u}$ are designed to minimize $\|G\vec{u} - \vec{v}\|_2$ over the given set of training data points. These coefficients, which minimize the l_2 norm, are computed by using the inverse of G such that $\vec{u} = G^{-1}\vec{v}$.

The coefficient vector $\vec{u}$ can be interpreted as a vector field

$\vec{u} : M \to \mathbb{R}^d$ over the set of training points or, equivalently,

$$\vec{v}(x) = \sum_{y \in M} G_{(x,y)} \vec{u}(y),\ x \in M, \tag{1}$$

where $\vec{u}(y)$, $y \in M$, are considered as the coefficients of the vector field $\vec{v}$ according to the super-kernel G. Consider a new data point $x' \in \mathcal{M} \backslash M$ with the matrix $O_{x'}$ whose columns $o_{x'}^1, \ldots, o_{x'}^d$ form an orthonormal basis for the tangent space $T_{x'}(\mathcal{M})$. We can extend the vector field to a new data point x' by setting the value $\vec{v}(x')$ to be

$$\vec{v}(x') \triangleq \sum_{y \in M} \tilde{G}_{(x',y)} \vec{u}(y), \tag{2}$$

where $\tilde{G}_{(x',y)} = \bar{p}(x', y) O_{x'}^T O_y$, $y \in M$, are the non-scalar affinity blocks between the new data point and the data points in the dataset. The extension in Eq. 2 is consistent with the values $\vec{v}(x)$, $x \in M$, in Eq. 1.

While the new affinity blocks in Eq. 2 are not known in advance as part of the super-kernel, they are easily computed for any new data point. This approximation only considers values of the vector field $\vec{u}$ for the data points in M, which can be computed in advance by using the pseudo inverse of the super-kernel G. This computation is not complicated, but it is beyond the scope of this paper since it is not essential for the presented dictionary construction. Therefore, this provides a feasible out-of-sample extension of a vector field, which is similar to the methods shown in [7], [8] for the scalar case.

The extension in Eq. 2 can be interpreted geometrically by separately considering the projections and the scalar weights in the affinity blocks of the super-kernel. First, the extension projects the coefficient vector field $\vec{u}$ from the manifold M to the tangent space $T_{x'}(\mathcal{M})$ of the new data point x'. This projection expresses the coefficient vectors in local terms of the manifold around x'. Then, the value of the vector field $\vec{v}$ at x' is computed by using a weighted sum of the projected coefficient vectors on the tangent space $T_{x'}(\mathcal{M})$.

V. Constructive Patch Sampling

According to Lemma 3.3 in [3], the sum in Eq. 1 can be rephrased in terms of the embedded tensors $x \in M$ to be

$$\vec{v}(x) = \sum_{y \in M} \mathcal{T}_x^T \mathcal{T}_y \vec{u}(y). \tag{3}$$

However, due to linear dependencies between the embedded tensors, this sum may contain redundant elements. Indeed, if $\mathcal{T}_z = \sum_{z \neq y \in M} c_y^z \mathcal{T}_y$ for some scalar coefficients $c_y^z \in \mathbb{R}$, $z \neq y \in M$, then Eq. 3 becomes $\vec{v}(x) = \sum_{z \neq y \in M} \mathcal{T}_x^T \mathcal{T}_y (\vec{u}(y) + c_y^z \vec{u}(z))$. This enables us to eliminate the redundant tensors and by applying an iterative approach, we obtain a small subset linearly independent tensors that are sufficient for computing Eqs. 1 and 2.

Similarly, we can use matrix coefficients instead of scalar ones to incorporate reacher relations between tensors. Therefore, $\mathcal{T}_z$ is tensorialy dependent in $\{\mathcal{T}_y\}_{z \neq y \in M}$ if $\mathcal{T}_z = \sum_{z \neq y \in M} \mathcal{T}_y C_y^z$ for some matrix coefficients $C_y^z \in \mathbb{R}^{d \times d}$, $z \neq y \in M$. This dependency expresses more redundancies than the standard linear dependency. As a result, we obtain a sparser set of tensorialy independent tensors that enables us to efficiently compute Eqs. 1 and 2. This set of representative tensors constitutes a dictionary that compactly represents the embedded tensor space.

A. Dictionary Construction

We use an iterative approach to construct the described dictionary by a sequential scan of the data points in M. In the first iteration, we define the scanned set $X_1 = \{x_1\}$ and the dictionary $D_1 = \{x_1\}$. At each iteration $s = 2, \ldots, n$, we have a new data point x_s, the scanned set $X_{s-1} = \{x_1, \ldots, x_{s-1}\}$ from the previous iteration and the dictionary D_{s-1} that represents X_{s-1}. The dictionary D_{s-1} is in fact a subset of η_{s-1} data points from X_{s-1} that are sufficient to represent its embedded tensors. We define the scanned set $X_s = X_{s-1} \cup \{x_s\}$. Our goal is to define the dictionary D_s of X_s, based on the dictionary D_{s-1} with the new data point x_s. To do this, a dependency criterion has to be established. If this criterion is satisfied, then the dictionary remains the same such that $D_s = D_{s-1}$. Otherwise, it is updated to include the new data point $D_s = D_{s-1} \cup \{x_s\}$.

We use a dependency criterion that is similar to the approximated linear dependency (ALD) criterion from [5]. The ALD measures the distance between vector candidates and the span by the dictionary vectors. In our case, we want to approximate the tensorial dependency of $\mathcal{T}_{x_s}$ on the tensors in the dictionary D_{s-1}. Therefore, we define the distance of $\mathcal{T}_{x_s}$ from the dictionary D_{s-1} as $\delta_s \triangleq \min_{C_1, \ldots, C_{\eta_{s-1}}} \left\| \sum_{j=1}^{\eta_{s-1}} \mathcal{T}_{y_j} C_j - \mathcal{T}_{x_s} \right\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm, and $C_1, \ldots, C_{\eta_{s-1}} \in \mathbb{R}^{d \times d}$ are matrix coefficients. The approximated tensorial dependency (ATD) criterion is defined as $\delta_s \leq \mu$, for some accuracy threshold $\mu > 0$. If the ATD criterion is satisfied, then the tensor $\mathcal{T}_{x_s}$ can be approximated by the dictionary D_{s-1}, using the matrix coefficients $C_1^s, \ldots, C_{\eta_{s-1}}^s$ of δ_s. Otherwise, the dictionary has to be updated by adding x_s to it. Lemma 3 (whose proof appears in [9]) shows that δ_s and the dictionary-based approximation can be expressed in terms of the super-kernel and without requiring knowledge of the embedded tensors the embedded tensors.

Lemma 3. *Let $\hat{G}_{s-1} \in \mathbb{R}^{d\eta_{s-1} \times d\eta_{s-1}}$ be the super-kernel of the data points in D_{s-1}, and let $H_s \in \mathbb{R}^{d\eta_{s-1} \times d}$ be a $\eta_{s-1} \times 1$ block matrix whose j-th $d \times d$ block is $G_{(y_j, x_s)}$, $j = 1, \ldots, \eta_{s-1}$. Then, the optimal matrix coefficients in δ_s are the η_{s-1} blocks, of size $d \times d$, in $\hat{G}_{s-1}^{-1} H_s$. The achieved δ_s satisfies $\delta_s = \mathrm{tr}[G_{(x_s, x_s)} - H_s^T \hat{G}_{s-1}^{-1} H_s]$.*

Essentially, this lemma eliminates the need for prior knowledge of the embedded tensors during the dictionary construction. At each iteration s, the criterion $\delta_s < \mu$ is considered. Based on this condition, we decide whether to add x_s to the dictionary or just approximate its tensor. The threshold μ is given in advance as a meta-parameter and δ_s can be computed by using Lemma 3. Therefore, the dictionary construction process only requires knowledge of a relatively limited number

of super-kernel blocks, which is determined by the size of the dictionary and not by the size of the dataset.

VI. Example: MNIST Handwritten Digit Classification using Patch-based Analysis

The patch-based methodology provides a general framework that can be utilized to a wide spectrum of data analysis tasks such as clustering, classification, anomaly detection and related manifold learning tasks. In this section, we demonstrate its utilization of the task of MNIST Handwritten digit classification. This experiment was done utilizing an of-the-shelve computer with a $I7-2600$ quad core CPU and a $16GB$ of DDR3 memory.

The MNIST database of handwritten digits [10] (available from http://yann.lecun.com/exdb/mnist/) consists of a training set of $60,000$ examples and a test set of $10,000$ examples. Each digit example is given as a grey levels 28×28 image. The digit images were centered by computing the center of mass of the pixels, and a translation operation was preformed to position this point at the center of the 28×28 field. MNIST is a subset of a larger set available from NIST. Many machine learning methods have been tested on this data set, hence the recognition performance is highly competitive. Currently, convolutional networks show a state-of-the-art recognition accuracy with an error of 0.23% [11]. For our purpose, the MNIST dataset provides a dataset of $70,000$ data points of very high dimensional measurements of size 728 pixels per a measured digit. In our experiments, we used the images as is.

The dictionary approximated patch-based embedding was utilized to embed the MNIST dataset of $70,000$ examples by the following steps. First, in each data point we identified the 150 nearest neighbors and computed the corresponding local PCA. For each local tangent space, we kept the 3 significant eigenvectors. Secondly, the diffusion affinities were computed with $\varepsilon = 105$ (see Section II-A), which is the Euclidean distance mean of all pairwise data points. The proposed dictionary construction with ATD threshold $\mu = 0.0001$ identified 93 important patches and their corresponding local tangent spaces. Finally, the approximated tensors were constructed utilizing $\ell = 30$. The labeling of each test data-point was estimated using the label of the nearest training data-point, where the pairwise distance was computed as the Frobenius norm of the difference between the corresponding embedded tensors. The resulting labeling error of the patch-based recognition method is 5.8%. Table I compares the computational costs of the straightforward implementation of the PTE algorithm from [3] and the presented dictionary-based algorithm on the MNIST dataset.

Size	SVD Cost - Full G	Dict. Size	SVD Cost - Approx. G
$70,000$	$O\left(70,000^3 \times 3^3\right)$	93	$O\left(70,000 \times 77,841\right)$

TABLE I
Computational cost of the SVD step in the dictionary approximated PTE (*SVD Cost - Approx. G*) vs. the full SVD of the super-kernel (*SVD Cost - Approx. G*) of the NIST dataset.

Although we are not far away from the state-of-the-art in digit recognition, the proposed method has the following advantages: 1. It shows that patch processing can be practically utilized for recognition and data analysis tasks. 2. Big high-dimensional datasets can be processed on "cheap" hardware such as in our case where the algorithm ran on less than $1000\$$ worth of hardware.

VII. Conclusions

The proposed construction in the paper extends the dictionary construction in [5] by using the LPD super-kernel from [3], [4]. This is done by an efficient dictionary-based construction that assumes the data is sampled from an underlying manifold while utilizing the non-scalar relations between manifold patches instead of considering individual data-points. The constructed dictionary contains patches from the underlying manifold, which are represented by the embedded tensors from [3], instead of individual data points. Therefore, it encompasses multidimensional similarities between local areas of the data. The patch-based dictionary reduces the computational costs of the spectral analysis in comparison to the PTE [3], hence, it enables us to apply this patch processing approach for datasets that were impractical to process and embed before.

Acknowledgments

This research was supported by the Israel Science Foundation (Grant No. 1041/10) and the Eshkol Fellowship from the Israeli Ministry of Science & Technology.

References

[1] R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[2] A. Singer and H. Wu, "Vector diffusion maps and the connection laplacian," *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012.

[3] M. Salhov, G. Wolf, and A. Averbuch, "Patch-to-tensor embedding," *Applied and Computational Harmonic Analysis*, vol. 33, no. 2, pp. 182 – 203, 2012.

[4] G. Wolf and A. Averbuch, "Linear-projection diffusion on smooth Euclidean submanifolds," *Applied and Computational Harmonic Analysis*, vol. 34, pp. 1 – 14, 2013.

[5] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2275 – 2285, aug. 2004.

[6] C. Baker, *The Numerical Treatment of Integral Equations*. Oxford: Clarendon Press, 1977.

[7] R. Coifman and S. Lafon, "Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 31–52, 2006.

[8] A. Bermanis, A. Averbuch, and R. Coifman, "Multiscale data sampling and function extension," *Applied and Computational Harmonic Analysis*, vol. 34, pp. 182 – 203, 2013.

[9] M. Salhov, A. Bermanis, G. Wolf, and A. Averbuch, "Approximate patch-to-tensor embedding via dictionary construction," *Submitted*, 2012.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[11] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *CVPR*, 2012, pp. 3642–3649.