Fast Non-Local Neural Networks with Spectral Residual Learning

Lu Chi, Guiyu Tian, Yadong Mu*
Peking University, P.R. China

Lingxi Xie, Qi Tian
Huawei Noah's Ark Lab, P.R. China

ABSTRACT

Effectively modeling long-range spatial correlation is crucial in context-sensitive visual computing tasks, such as human pose estimation and video classification. Enlarging receptive field is popularly adopted in building such non-local deep networks. However, current solutions, including dilation convolution or self-attention based operators, mostly suffer from either low computational efficacy or insufficient receptive field. This paper proposes spectral residual learning (SRL), a novel network architectural design for achieving fully global receptive field. A neural block that implements SRL has three key components: a local-to-global transform that projects some ordinary local features into a spectral domain, compiled operations in the spectral domain, and a global-to-local transform that converts all data back to the original local format. We show its equivalence to conducting residual learning in some spectral domain and carefully re-formulate a variety of neural layers into their spectral forms, such as ReLU or convolutions. The benefits of SRL is three-fold: first, all operations have global receptive field, namely any update affects all image positions. This can extract richer context information in various vision tasks; Secondly, the local-to-global / global-to-local transforms in SRL are defined by bi-linear unitary matrices, which is both computation and parameter economic; Lastly, SRL is a generic formulation, here instantiated by Fourier transform and real orthogonal matrix. We conduct comprehensive evaluations on two challenging tasks, including human pose estimation from images and video classification. All experiments clearly show performance improvement by large margins in comparison with conventional non-local network designs.

CCS CONCEPTS

• **Computing methodologies → Computer vision tasks**; **Neural networks**.

KEYWORDS

Deep neural networks, spectral transform, residual learning, human pose estimation, video classification

ACM Reference Format:
Lu Chi, Guiyu Tian, Yadong Mu and Lingxi Xie, Qi Tian. 2019. Fast Non-Local Neural Networks with Spectral Residual Learning. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3343031.3351029

1 INTRODUCTION

The revolutionary success of deep convolutional neural networks (CNN) starts from semantic image classification [32] and quickly influences many other research areas, including computer vision and multimedia analysis [5, 35, 45]. The top performances in a variety of computer vision tasks have been re-calibrated by CNN based deep models. Though many engineering techniques (*e.g.*, ReLU, batch normalization) are recently developed, a large body of modern architectural design of CNN models is still grounded on the classic convolutional layer which can date back to Hubel and Wiesel's Nobel prize-winning research about cat's visual cortex in 1962 or earlier. Typically, a convolutional window slides across the entire image / feature channels with some pre-specified local receptive field. The convolutional kernels are shared by all image positions, thus highly parameter-economic. Through gradient back-propagation, a convolutional kernel can effectively learn different levels of discriminative visual patterns (such as low-level edges, blobs, textures, mid-level object parts, or high-level objects), depending on its depth in the neural networks.

Recent development has witnessed the importance of modeling long-range spatial or temporal dependencies in many context-sensitive visual computing tasks [13, 33, 60, 69]. For example, a good image segmentation model shall concurrently consider pixel-wise confidence and mutual semantic compatibility among pixels, as demonstrated by the empirical effectiveness of DeepLab [6]. In video classification [42, 55, 58], spatio-temporal convolutions across multiple consecutive frames are crucial for capturing temporal dynamics of semantic actions. Naturally, long-range dependencies between distant positions can be modelled by large receptive field. A widely-held belief by practitioners is that, a sufficiently deep network does not desire to explicitly use large receptive field for convolutions, since stacking many convolutions can progressively enlarge the spatial extent when calculating a neuron at high-level neural layers, such as the 3 × 3-sized kernels adopted by VGG-Net [49] and ResNet [19]. One can also use dilated [6] or deformable convolutions [12], or recently-proposed sophisticated non-local kernels, exemplified by self-attentive operator [60].

However, all aforementioned non-local convolutional schemes suffer from low efficacy issue. For simple solutions like dilation or deformable convolutions, they are still essentially local, indirectly connecting distant positions. To convey some message to far locations, multiple layers are desired to be stacked. This implies

[*] Corresponding author. Email: myd@pku.edu.cn.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351029

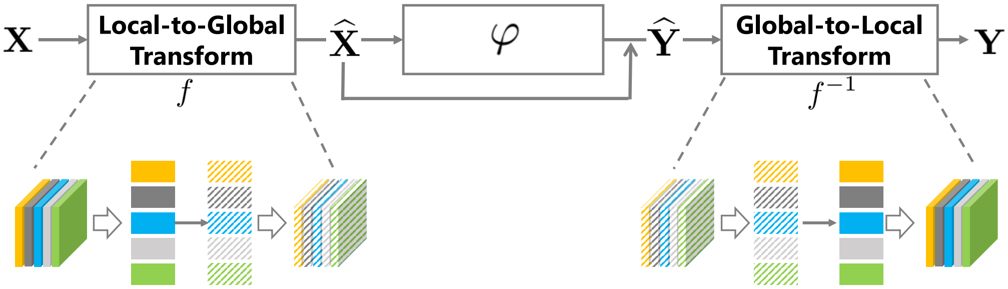

Figure 1: Computational pipeline of our proposed *spectral residual learning* (SRL). Note that both f and f^{-1} are performed in a channel-wise manner for saving computations. $\widehat{X}$ is fed to $\widehat{Y}$ via a skip connection to ensure the residual learning. Some feature maps are pattern-filled to emphasize that they are in the spectral domain. More details regarding the functions f, f^{-1}, φ are explained in the main text.

that multiple-hop propagation will take place before a message reaches its destination. Despite of fewer parameters to be learned, such scheme is arguably less efficient. In addition, large-depth networks are often more difficult to be optimized, and tend to generate more high-level neurons that are not suitable for visual localization-oriented tasks [16]. The non-local kernels in [60] naively connects all positions in a feature map. To determine the new value at a position, it calculates this position's affinity scores to all other positions and then performs a weighted average of features at all positions. This is easily verified to require a tremendous quadratic time complexity with respect to the count of all image positions. The authors thus propose to sub-sample across both spatial and temporal dimensions for reducing complexity.

This work addresses the issue of large receptive field from a rarely-explored aspect. Our proposed network architectural design, which we call *spectral residual learning* (SRL), accomplishes a fully global receptive field for visual context modeling with high efficacy. A conceptual diagram is illustrated in Figure 1. As seen, the proposed SRL is comprised of three critical components: a local-to-global transform that projects an input feature map into some spectral domain, a compilation of neural layers (*i.e.*, the function denoted by φ in Figure 1) that operate in the spectral domain, and a global-to-local transform that de-correlates all data back into the original local format. Importantly, updating an element in the spectral domain globally affects all others. State differently, all operators in the spectral domain enforce a full-image receptive field. In this work, a number of spectral operators are carefully defined, including spectral ReLU for frequency-sensitive filtering and spectral 1×1 convolutions.

Main contributions of this work are briefly summarized as below:

1. The proposed SRL models long-range dependencies in a more effective way than previous methods, through spectrally computing interactions between any two positions, regardless of their positional distance. To our best knowledge, SRL is the first work of its kind that explores the theory of spectral transform for obtaining full-image receptive field.

2. SRL enjoys several technical advantages: First, it is a generic framework, instantiated by various local-to-global / global-to-local transforms and spectral operators. A neural block that implements SRL can be inserted into many ordinary deep network's pipeline (*e.g.*, VGG-16 [49], DenseNet [20] or ResNet [19]). One can design an SRL instance in a task-specific fashion; Secondly, SRL requires significantly lower complexity and fewer parameters than competing methods, to achieve a similar level of performance. In particular, both cross-domain transforms can be defined and efficiently implemented by some bi-linear unitary transforms, either fixed or learnable by gradient back-propagation.

3. In experiments, we instantiate f in SRL with Fourier transform and real orthogonal transform respectively, φ with spectral ReLU etc., and demonstrate its effectiveness on two challenging tasks: human pose estimation and semantic video classification. Both tasks are supposedly improved by contextual information. Comprehensive quantitative evaluations show that SRL consistently improves base models by significant margins. Further ablation studies investigate the effect of several key factors in using an SRL block, including the location to insert SRL blocks in a base network and the count of SRL blocks.

2 RELATED WORK

We summarize research works that are relevant to either non-local networks or two demonstrative visual computing tasks.

Non-Local Neural Blocks: Wang *et al.* [60] devised non-local blocks to capture long-range dependencies, inspired by the self-attentive mechanism [56]. The proposed non-local operation is shown to be strongly beneficial in video classification [60], semantic image segmentation [15, 21], generative adversarial networks (GAN) [68], and recognizing fine-grained objects and actions [67].

Non-local blocks capture the global information by linking two arbitrary positions. For videos, the links will span at both spatial and temporal dimensions. Such dense links imply tremendous computational cost, which incurs a series of follow-up research. For example, the work in [9] manipulates the order of matrix multiplication, improving the efficacy under many special scenarios. Huang *et al.* [21] propose to stack two criss-cross attention modules, in order to condense the contextual information of surrounding pixels on the criss-cross path.

Non-local blocks and all of above variants are formulated on the Q-K-V attention mechanism [56], which typically admits a quadratic time complexity with respect to the position number. In contrast, the proposed SRL is formulated more efficiently in some unitary transform induced spectral domain.

Human keypoint detection: or known as human pose estimation. Traditional solutions combine local observations from body parts and spatial dependencies between them. Most of these methods use either tree-structured graphical models [3, 4, 13, 43] or non-tree models [26, 33, 47, 61].

Recent years have observed an explosive emergence of deep network based methods in this task [37, 39, 62, 65]. Among them, DeepPose [52] is the first to utilize CNN to tackle pose estimation, which devised a cascaded CNN specially for the single person case. Recent solutions to multi-person pose estimation can be roughly casted into two categories: bottom-up [22, 24, 40] or top-down approach [17, 24, 41, 51]. The former detects individual body joints and then groups them into persons. The latter approach relies on a good human detector for localizing all persons in the image, and multiple routines for single-person pose estimation are called to tackle each detected person.

Video classification: newly-constructed massive video datasets, such as [1, 18, 29], have greatly spurred the research of video classification. Karpathy *et al.* [27] introduced a first 3D convolutional neural networks for this task that far exceeded traditional methods [38, 57]. To find better initialization for the 3D convolutional kernels, a number of research works [25, 53, 54] inflate 2D convolution into 3D convolution, directly borrowing parameters from pretrained models on ImageNet [46]. This strategy proves to lead to faster convergence in most cases. Another line of research aims to prune the huge parameters in 3D CNNs, such that they can strike a compromise in accuracy and efficacy. Examples include factorizing 3D convolutional filters into separate spatial / temporal components [42, 54, 66], or mixing 3D convolution with 2D convolution in a same neural network [58, 66, 70].

We also witness recent research enthusiasm in globally modeling spatio-temporal dependency in videos. Regarding the temporal dimension in videos, one can find methods that utilize LSTM [63], global memory [64] or simply average pooling [59]. The work in [36] adopts an attentive mechanism along the temporal dimension, aggregating local features to generate a powerful global representation for video classification. For the spatial dimension in videos, Xie *et al.* [66] place a feature gating module after specific convolutional layers to weigh the global features in each channel in an adaptive, data-dependent way. All these methods reach the consensus that spatio-temporal context is of crucial importance for effectively classifying videos.

3 SPECTRAL RESIDUAL LEARNING

3.1 Formulation

Figure 1 illustrates the computational pipeline of SRL. Let $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ be the input tensor to an SRL block, and $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$ be the output tensor. The mechanism of SRL can be compactly described as:

$$\mathbf{Y} = f^{-1}\left(\varphi(f(\mathbf{X})) + f(\mathbf{X})\right) = f^{-1}\left(\varphi(f(\mathbf{X}))\right) + \mathbf{X}, \tag{1}$$

where f and f^{-1} denote the local-to-global transform / global-to-local transform in Figure 1, respectively. $\varphi : \mathbb{R}^{H\times W\times C} \mapsto \mathbb{R}^{H\times W\times C}$ represents a user-defined sub-network that operates on the globalized tensor $f(\mathbf{X})$.

The computation of an SRL block starts from feeding $\mathbf{X}$ into the local-to-global transform function f. To parameterize f, we introduce two unitary matrices[1] $\mathbf{P} \in \mathbb{C}^{H\times H}$ and $\mathbf{Q} \in \mathbb{C}^{W\times W}$. Let $concat()$ be an operator that concatenates a number of arrays along specific dimension, and $\mathbf{X}_c \in \mathbb{R}^{H\times W}$ be some single feature channel of $\mathbf{X}$ with index c. The global-to-local / local-to-global transforms are computed in a channel-wise fashion as below for efficacy consideration:

$$f(\mathbf{X}) = concat\left(\mathbf{P}\mathbf{X}_1\mathbf{Q}, \ldots, \mathbf{P}\mathbf{X}_C\mathbf{Q}\right), \tag{2}$$

$$f^{-1}(\widehat{\mathbf{Y}}) = concat\left(\mathbf{P}^*\widehat{\mathbf{Y}}_1\mathbf{Q}^*, \ldots, \mathbf{P}^*\widehat{\mathbf{Y}}_C\mathbf{Q}^*\right), \tag{3}$$

where * denotes a conjugate matrix. The global-to-local is parameter-economic since all feature channels share the same parameters $\mathbf{P}$ and $\mathbf{Q}$. The entries in $f(\mathbf{X})$ is now semi-global since $\mathbf{P}, \mathbf{Q}$ collaboratively perform information fusion in the entire $H \times W$-sized spatial space. State differently, each entry in $f(\mathbf{X})$ becomes correlated with all other entries in the same feature channel of $\mathbf{X}$.

Importantly, a simple derivation of Eqn. (1) shows $\varphi(f(\mathbf{X})) = f(\mathbf{Y} - \mathbf{X})$. It implies that φ is actually designed to approximate the residual $\mathbf{Y} - \mathbf{X}$ in some global domain induced by f. In practice, φ is often task-dependent. It encodes our key expectation about the way that information globally exchanges across the input tensor $\mathbf{X}$.

3.2 Instantiations of f

Here we describe two concrete examples of f, with the parameters of the first one fixed and the second learnable via gradient back-propagation.

Fast Fourier Transform Matrix (FFT): Fourier transform is a widely-adopted tool in the domains of signal processing, image analysis etc. It converts a function of signals into the frequencies that make it up, in a reversible way. In fact, discrete Fourier transform (DFT) can be accomplished as a complex matrix multiplication. For example, the one-dimensional DFT transforms an array of complex numbers $\mathbf{x} = \{x_0, x_1, \ldots, x_{N-1}\}$ to another sequence $\mathbf{z} = \{z_0, z_1, \ldots, z_{N-1}\}$, according to

$$z_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi k}{N}n}, k = 0, 1 \ldots N-1. \tag{4}$$

This can be re-formulated into a matrix form $\mathbf{z} = \mathbf{F}\mathbf{x}$ with

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} /N,$$

where $\omega = e^{-2\pi j/N}$ and $\mathbf{F}$ proves to be a unitary matrix.

Let $\mathcal{F}_{2d}(\cdot)$ be the function of 2-D FFT and $\mathcal{F}_{2d}^{-1}$ be the inverse. Given a 2-D array $\mathbf{X}_c \in \mathbb{R}^{H\times W}$, $\mathcal{F}_{2d}(\cdot)$ can be implemented by transforming all the rows of $\mathbf{X}_c$ and then transforming all the columns of the resulting matrix [44], namely:

$$\mathcal{F}_{2d}(\mathbf{X}_c) = \mathbf{F}_H \times \mathbf{X}_c \times \mathbf{F}_W,\ \mathcal{F}_{2d}^{-1}(\widehat{\mathbf{Y}}_c) = \mathbf{F}_H^* \times \widehat{\mathbf{Y}}_c \times \mathbf{F}_W^*, \tag{5}$$

where $F_H \in \mathbb{C}^{H\times H}$, $\mathbf{F}_W \in \mathbb{C}^{W\times W}$ are properly-sized Fourier matrices. $\mathcal{F}_{2d}(\mathbf{X}_c), \mathcal{F}_{2d}^{-1}$ can be accelerated via fast routines such as the

[1] A complex square matrix $\mathbf{U}$ is unitary if its conjugate transpose $\mathbf{U}^*$ is also its inverse, namely $\mathbf{U}^*\mathbf{U} = \mathbf{U}\mathbf{U}^* = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

Cooley-Tukey algorithm [11]. Clearly, this perfectly fits our SRL formulation in Equations (2) and (3).

Remark: FFT operates on complex numbers. To ensure compatibility with other neural layers, we simply abondon the imaginary part of $\mathbf{Y}$ after the global-to-local transform. In addition, we would also emphasize that FFT is not the only instance of its type. Many orthogonal transforms, including discrete cosine transform (DCT) [2], Walsh-Hadamard transform (WHT) [14], Haar transform, and Slant transform (ST), can be also used to instantiate f. This work focuses on Fourier transform for demonstration.

Fully Learnable Orthogonal Matrix (LO): A key difference between FFT-oriented instantiation with others is the non-learnable property of $\mathbf{F}_H, \mathbf{F}_W$ in Eqn. (5), which are thus not involved in the gradient back-propagation of deep networks where they embed. It is a favorable property when a low-parameter model is required. Nonetheless, we have also explored other choice of f. One of them is to initialize $\mathbf{P}, \mathbf{Q}$ in Eqn. (2) to be real orthogonal matrix, namely $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}_{H\times H}$ and $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{W\times W}$, where $\mathbf{I}$ denotes the identity matrix. Take $\mathbf{P}$ for instance, since it appears at both f and f^{-1} as parameters, the gradient of $\mathbf{P}$ can be described as:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{P}} &= \frac{\partial \mathcal{L}}{\partial f}\frac{\partial f}{\partial \mathbf{P}} + \frac{\partial \mathcal{L}}{\partial f^{-1}}\frac{\partial f^{-1}}{\partial \mathbf{P}} \\ &= \sum_c \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_c}\mathbf{Q}^T\mathbf{X}_c^T + \sum_c \frac{\partial \mathcal{L}}{\partial \mathbf{Y}_c}\mathbf{Q}\widehat{\mathbf{Y}}_c^T,\end{aligned}$$

where $\mathcal{L}$ denotes the objective. Unfortunately this inevitably leads to a non-symmetric $\mathbf{P}$. To ensure an orthogonal matrix, we conduct a post-processing after SGD. QR decomposition is applied to both of updated $\mathbf{P}, \mathbf{Q}$. For example, $\mathbf{P} = \widetilde{\mathbf{P}}\mathbf{R}$, where $\widetilde{\mathbf{P}}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix. We simply let $\mathbf{P} \leftarrow \widetilde{\mathbf{P}}$. Likewise $\mathbf{Q}$ is also re-orthogonalized. In practice, $\mathbf{P}, \mathbf{Q}$ are initialized by drawing from a normal distribution.

3.3 Instantiations of φ

Recall that φ is conducted on globalized $\widehat{\mathbf{X}}$. In this section, whenever needed we use the prefix *spectral* to emphasize that all operations are conducted in a spectral domain. The specification of φ is often task-dependent. In practice, we use an SRL block to wrap specific instantiation and insert it into existing neural networks.

Figure 2 shows a typical instance inspired by the bottle-neck block in ResNet [19]. It is comprised of the following spectral layers:

Spectral Convolution: according to the convolution theorem [28], for Fourier-related transforms, convolution in one domain equals point-wise multiplication in the other domain. This implies that any convolution in φ incurs a global update. For the SRL block in Figure 2, a pair of 1×1 convolutions are adopted for channel reduction and promotion respectively. We have experimented with spectral convolution with larger kernels than 1×1 that corresponds to more complex point-wise update in $\mathbf{X}$, yet do not observe any non-trivial performance improvement. When the channel number is large, standard group convolution [50] is utilized for expediting the computation. We empirically set the group number to 2 when f is *LO* and 8 for the FFT instance.

For video data, the intermediate feature map is 4-D tensor of size $H \times W \times C \times T$ (where C, T are counts of channels / stacked video frames respectively), rather than $H \times W \times C$ in images. In this case, inspired by P3D [42], we approximate complex spatio-temporal convolutions in videos into a 1×1 convolution along feature channels (denoted as $conv_c$), followed by another 1×1 convolution along the temporal dimension (denoted as $conv_t$). The design is shown in Figure 3. The transition between two heterogeneous 1×1 convolutions are efficiently implemented by tensor transpose. When applying the SRL block in Figure 2 into video data, one only needs to replace these two spatial 1×1 convolutions by the sequential operations of $conv_c + conv_t$.

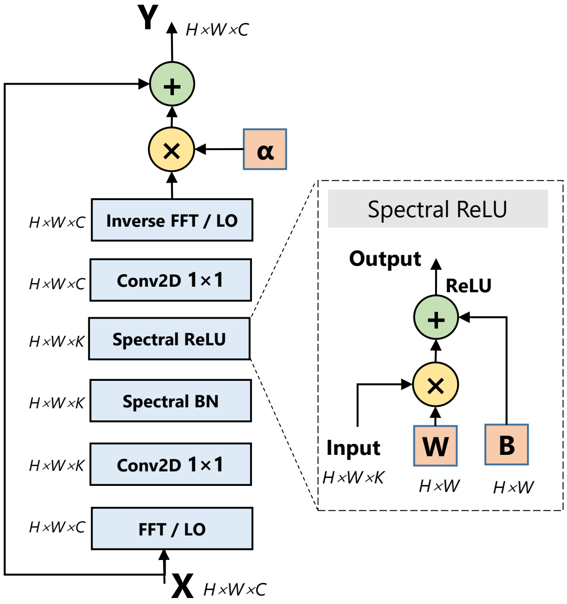

Figure 2: An instance of φ adopted in human pose estimation.

Spectral Batch Normalization (BN): The computations of spectral BN is identical to the ordinary version, yet operating on spectral frequencies and with each update affect all image positions.

Spectral ReLU: For a globalized tensor $\mathbf{Z} \in \mathbb{R}^{H\times W\times K}$, spectral ReLU is controlled by learnable parameters $\mathbf{W}, \mathbf{B} \in \mathbb{R}^{H\times W}$:

$$\mathbf{Z}_{i,j,k} \leftarrow \max\left(W_{i,j} \cdot \mathbf{Z}_{i,j,k} + \mathbf{B}_{i,j},\ 0\right), \tag{6}$$

where $(i, j, k) \in [1 \ldots H] \times [1 \ldots W] \times [1 \ldots K]$. It essentially does the job of frequency-sensitive filtering.

Parameter α in Figure 2 is introduced to balance the residual block and original $\mathbf{X}$ as defined in Eqn. (1). $\alpha = 0$ initializes an identity mapping.

3.4 Complexity analysis

Table 1 compares the computational complexities of different methods that implement non-local receptive fields. Our two representative variants with f being FFT or real orthogonal matrices are denoted by SRL-FFT, SRL-LO respectively. All estimations here take $H \times W \times C$-sized tensor as the input. The complexity analysis of video data is omitted due to the space limit.

As seen in Table 1, the influential work in [60] is highly sensitive to feature map resolution. A^2-Net [9] proposes a sophisticated technique for accelerating matrix multiplication, which reduces the dependence on the spatial resolution. CGNL [67] and CCNet [21]

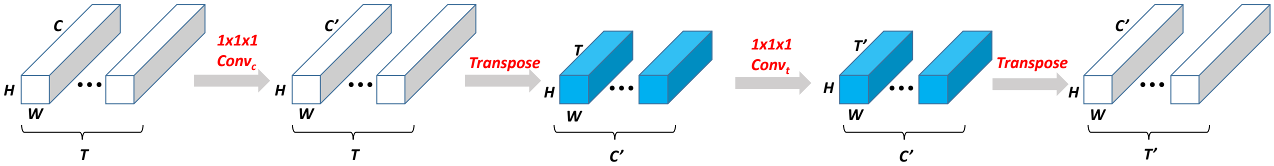

Figure 3: Extending image oriented 1×1 spectral convolution to the video data. $H \times W$, C, T denote the resolution of feature maps, feature channel number and stacked video frames. $conv_c$, $conv_t$ operate on the channels / temporal dimension, respectively.

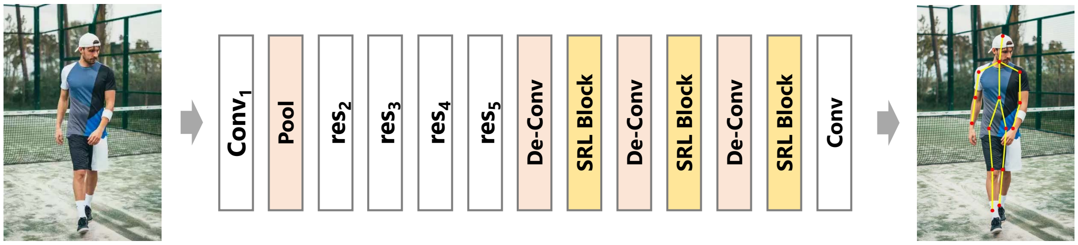

Figure 4: Network architecture adopted in human keypoint detection. This illustration inserts three SRL blocks, separately after each of the de-convolutional layers.

Method	Time Complexity
Non-Local (NL) [60]	$O(CH^2W^2)$
A^2 [9]	$O(C^2HW)$
CGNL [67]	$O(CHW(P+1))$
RCCA [21]	$O(CHW(H+W))$
SRL-LO	$O(CHW(H+W))$
SRL-FFT	$O(CHW\log(HW))$

Table 1: Computational complexities of different global operations, where C, H, W represent the counts of feature channel, height and width respectively. P is the order of Taylor series [67].

further reduce the quadratic complexity with respect to C into linear. Our proposed SRL-LO is as light-weight as previous state-of-the-art, and our parameter-free (for f, f^{-1}) variant SRL-FFT strikes the best time complexity among all methods in current literature.

4 EXPERIMENTS

To demonstrate the usefulness of SRL blocks, we apply the idea to solve two visual tasks: human keypoint detection and video classification. The evaluations involve two instances of f, f^{-1}: SRL-LO and SRL-FFT respectively. The major goal is to contrast various proposals for non-local deep networks, including our proposed SRL and those methods mentioned in Table 1. We adopt the original implementations by authors (if publicly available) or re-implement these models in PyTorch. Extensive ablative studies are also conducted to investigate key factors in the proposed method.

For fairly comparing complexity among different models, we adopt both GFLOPs and parameter number as the key metrics. Following the practice in [30], conventional models with local receptive field only consider standard types of neural layers, including convolution / linear / batch normalization etc. Additional computations in non-local methods (such as tremendous matrix multiplication introduced in non-local networks (NL) [60]) are taken into account when calculating GFLOPs.

4.1 COCO Keypoint Detection

The COCO dataset [34] contains about 250K person instances from 200K images. Each person is labeled with 17 keypoints, each of which defines a joint of human body. All models are trained on the train2017 set of 57K images, and evaluated on the val2017 set of about 5K images.

Network: We adopt the design of SimpleBaseline [65] to construct the base model where SRL bottleneck blocks in Figure 2 (with $K = C/4$) are inserted. The network structure is shown in Figure 4. It adopts pre-trained ResNet [19] on ImageNet [46] as image encoder. The original decoder consists of three de-convolutional layers (stride: 1/2, kernel size: 4×4 and filters: 256). Each is accompanied with batch normalization [23] and ReLU activation [32]. A 1×1 convolutional layer is appended to the topmost to predict 17 heatmaps, each of which corresponds to a unique key point. Our proposed SRL blocks are inserted rightly after each deconvolutional layer for contextual modeling.

Model Optimization and Evaluation: We follow typical top-down keypoint detection paradigm and use the same person detector to [65] (AP score on COCO val2017 dataset is 56.4). During the training phase, following [10, 65], we expand the ground-truth human box to a fixed aspect ratio (*e.g.*, height : width = 4 : 3), crop the image region and resize it into some fixed resolution (we use 256×192, 384×288 and 512×384). Standard data augmentations including random rotation ($[-40^\circ, 40^\circ]$), random scale ($\pm 30\%$) and flipping etc. are adopted. Models are trained by Adam optimizer [31] on an 8-GPU machine. The mini-batch on each GPU is set to 32, and maximum epoch is 210. When evaluating the model on a testing image, we calculate and average heatmaps for both the original and flipped images. In addition, directly treating local optimum as human keypoints often brings imprecise localization. We adjust keypoint' location by applying a quarter offset in the direction from the highest response to the second highest.

Method	Backbone	Input Size	GFLOPs	# Params(M)	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass [37]	8-stage Hourglass	256 × 192	14.3	25.1	66.9	-	-	-	-	-
CPN [10]	ResNet-50	256 × 192	6.2	27.0	68.6	-	-	-	-	-
CPN + OHKM [10]	ResNet-50	256 × 192	6.2	27.0	69.4	-	-	-	-	-
CPN + OHKM [10]	ResNet-50	384 × 288	-	-	71.6	-	-	-	-	-
SimpleBaseline [65]	ResNet-50	256 × 192	8.99	34.00	70.4	88.6	78.3	67.1	77.2	76.3
		384 × 288	20.23		72.2	89.3	78.9	68.1	79.7	77.6
		512 × 384	35.97		71.7	88.8	77.7	67.2	79.7	77.1
	ResNet-101	256 × 192	12.38	52.99	71.4	89.3	79.3	68.1	78.1	77.1
		384 × 288	27.85		73.6	89.6	80.3	69.9	81.1	79.1
		512 × 384	49.52		74.2	89.7	80.7	70.2	81.8	79.5
	ResNet-152	256 × 192	15.76	68.64	72.0	89.3	79.8	68.7	78.9	77.8
		384 × 288	35.47		74.3	89.6	81.1	70.5	81.6	79.7
		512 × 384	63.06		74.9	89.8	81.3	70.8	82.5	80.0
SRL-FFT	ResNet-50	256 × 192	9.08	34.10	70.9	89.1	78.5	67.4	77.9	76.8
		384 × 288	20.43		73.3	89.5	80.0	69.4	80.6	78.6
		512 × 384	36.32		73.8	89.7	80.3	69.6	81.5	79.0
	ResNet-101	256 × 192	12.46	53.09	71.8	89.3	79.6	68.4	78.7	77.6
		384 × 288	28.05		74.3	90.1	81.3	70.5	81.5	79.7
		512 × 384	49.86		74.9	89.9	81.6	71.1	82.3	80.1
	ResNet-152	256 × 192	15.85	68.74	72.1	89.5	79.7	68.8	79.1	78.0
		384 × 288	35.67		74.6	89.7	81.7	70.8	81.9	80.1
		512 × 384	63.41		75.3	90.2	81.7	71.5	82.6	80.4

Table 2: Comparisons on the COCO validation set for human keypoint detection. See the main text for more detailed descriptions.

Evaluation metric: All models are evaluated based on *object keypoint similarity* (OKS) (see http://cocodataset.org/#keypoints-eval). Several metrics are reported, including AP^{50} (AP at OKS = 0.50), AP^{75} (OKS = 0.75), AP (the mean of AP scores at OKS = $0.50, 0.55, \ldots, 0.90, 0.95$; AP^{M} for medium objects, AP^{L} for large objects, and AR at OKS = $0.50, 0.55, \ldots, 0.90, 0.95$.

Performance Analysis: We comprehensively investigate keypoint detection performance by varying network backbones or input image's spatial resolutions. Table 2 presents the experimental results of SRL-FFT, and Table 3 summarizes the information of several competing methods. From these tables, two observations can be drawn:

1). SRL is particularly effective for shallow backbone and large resolution: After inserting three SRL blocks, the improvement using ResNet-50 is the most significant among all three backbones, and SRL provides least help to ResNet-152. This is consistent to our intuition: deeper networks implicitly enlarge their receptive fields by stacking more convolutional layers, which makes the contribution of non-local blocks increasingly marginal.

Large spatial resolution for input images can notably elevate the performance in most of these experiments. Interestingly, the gains of SRL blocks are also most salient under large resolutions. For example, using ResNet-50, SRL brings an absolute improvement of 0.5% (for input size 256 × 192), 1.1% (for input size 384 × 288), and 2.1% (for input size 512 × 384) respectively. This is indeed consistent with prior observation about backbone's depth. Either making the network shallower or increasing input's spatial resolution can relatively reduce the receptive field of convolutional layers, leaving room for non-local blocks.

input size	Non-Local Block	GFLOPs(Δ)	# Params((MΔ))	AP
256 × 192	-	0	0	70.4
	NL [60]	2.90	0.39	70.6
	A^2 [9]	0.28	0.20	70.5
	CGNL [67]	0.39	0.31	70.6
	RCCA [21]	1.20	0.92	70.6
	SRL-FFT	**0.09**	**0.10**	**70.9**
384 × 288	-	0	0	72.2
	NL [60]	13.26	0.39	72.8
	A^2 [9]	0.62	0.20	72.8
	CGNL [67]	0.88	0.31	72.4
	RCCA [21]	2.76	0.92	72.8
	SRL-FFT	**0.20**	**0.10**	**73.3**

Table 3: Comparisons with other non-local blocks on COCO human keypoint detection. The backbone of all methods is ResNet-50.

2). SRL blocks strikes a better balance between accuracy and efficacy: Deeper backbone is more powerful in extracting discriminative features and thus favored in practice. From Table 2, SRL blocks + shallower backbone proves to achieve comparable performance with deeper models. For example, under an input size of 384 × 288, ResNet-101 + SRL-FFT can perform as excellent as ResNet-152 using only 80% FLOPs and 77% parameters of the latter. It also can perform better than the same backbone with larger input size 512 × 384, saving about 43% FLOPs of the latter.

We also re-produce a variety of competing non-local blocks and evaluate them on COCO's validation set. The quantities are found in Table 3. For fair comparison, none of aforementioned

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
baseline	72.2	89.3	78.9	68.1	79.7	77.6
baseline*	66.3	87.1	71.4	61.3	74.6	72.0
dilated conv	72.3	**89.4**	79.1	68.1	80.1	77.6
+3 NL [60]	72.7	88.9	79.1	68.9	79.9	77.9
+3 SRL-FFT	73.4	89.3	79.8	69.5	80.7	78.8
+4 SRL-FFT	**73.7**	**89.4**	**80.3**	**69.7**	**81.2**	**78.9**

Table 4: On COCO keypoint dataset, additional investigation on receptive field by creating new baseline with fewer downsampling operations (and thus smaller receptive field). The backbone of all methods is ResNet-50 and the input size is 384×288**.**

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
-	72.2	89.3	78.9	68.1	79.7	77.6
+1 @ d1	72.7	89.2	79.7	68.7	80.0	78.1
+1 @ d2	72.6	89.3	79.5	68.7	79.8	78.0
+1 @ d3	72.6	**89.6**	79.4	68.6	80.1	78.0
+2 @ d1 / d2	72.9	89.2	79.9	**69.5**	79.9	**78.6**
+3 @ d1 / d2 / d3	**73.3**	89.5	**80.0**	69.4	**80.6**	**78.6**

Table 5: Performance of human keypoint detection by varying the inserting positions and counts of SRL blocks. "d" represents deconvolutional layers.

Backbone	input size	baseline	SRL-LO	SRL-FFT
ResNet-50	256×192	70.4	**71.0**	70.9
	384×288	72.2	72.8	**73.3**
ResNet-101	256×192	71.4	71.7	**71.8**
ResNet-152	256×192	72.0	**72.3**	72.1

Table 6: Comparisons between two instantiations of f**.**

methods applies sub-sampling trick for acceleration, and all insert three blocks at the same positions as SRL does. Compared with those state-of-the-art competitors, our proposed SRL blocks is more light-weighted, using fewer parameters and FLOPs to achieve top performances.

Effect of Receptive Field: Additional experiments are designed for further investigating receptive field. In specific, for the network depicted in Figure 4, we set the stride of down-sampling layer to 1 in res_4 and res_5 and only preserve one deconvolutional layer in the decoder. This increases spatial resolution of intermediate feature maps and manually creates a backbone (denoted as baseline*) with shorter receptive field for comparative study. The experimental results reported in Table 4 show a significant performance drop of baseline*, caused by the reduction of receptive field.

To reveal the critical role of receptive field in human pose estimation, Table 4 includes the average precision of adding three SRL blocks into encoder (right after the last block of res_3, res_4 and res_5) of baseline*, or adding a fourth SRL block to the decoder, which are denoted by "+3 SRL-FFT", "+4 SRL-FFT" respectively.

To make a comprehensive study, we also add dilated convolution [7, 8, 21] in res_4 and res_5, same to the treatment in [7], and add three non-local blocks [60] to baseline*. A close look of Table 4 clearly proves better effectiveness of SRL blocks, in comparison with all above-mentioned baselines.

stage	ResNet-18	output size
$conv_1$	1×7×7, 64, stride (1,2,2)	$8 \times 112 \times 112$
$pool_1$	1×3×3 max, stride (1,2,2)	$8 \times 56 \times 56$
res_2	$\begin{bmatrix} 1\times3\times3, 64 \\ 1\times3\times3, 64 \end{bmatrix} \times 2$	$8 \times 56 \times 56$
$pool_2$	2×1×1 max, stride (2,1,1)	$4 \times 56 \times 56$
res_3	$\begin{bmatrix} 1\times3\times3, 128 \\ 1\times3\times3, 128 \end{bmatrix} \times 2$	$4 \times 28 \times 28$
res_4	$\begin{bmatrix} 1\times3\times3, 256 \\ 1\times3\times3, 256 \end{bmatrix} \times 2$	$4 \times 14 \times 14$
res_5	$\begin{bmatrix} 1\times3\times3, 512 \\ 1\times3\times3, 512 \end{bmatrix} \times 2$	$4 \times 7 \times 7$
global average pool, fc		$1 \times 1 \times 1$

Table 7: ResNet-18 C2D model for video classification. The kernel size and output maps are represented in the format of $T \times H \times W$**, typically with the number of channels following. The input size is** $8 \times 224 \times 224$ **with 8 stacked frames.**

Ablation Study: For ResNet-50 with input size 384×288, Table 5 shows that even adding a single SRL block can bring a noticeable improvement over the baseline. More SRL blocks continue to improve in a consistent manner, but the gain of adding new blocks quickly diminishes. Moreover, inserting an SRL block at shallower layer of the backbone is relatively more effective (+1 @ d1 > +1 @ d2 > +1 @ d3), in terms of the strict criterion AP^{75}. Tables 6 compares different instantiations of f, yet shows no obvious winner between LO or FFT based transform. Considering the additional parameters of LO, FFT is thus more favored in practice.

4.2 Video Classification

Kinetics [29] is a large-scale trimmed video dataset which contains more than 300-K video clips in total. Each clip has a duration of around 10 seconds. The dataset covers 400 human-centric classes and each has at least 400 video clips. We crawl and trim videos using officially-provided urls, obtaining 232,679 video for training and 19,169 for validation. The original test set is omitted due to the lack of ground-truth labels.

Base Network: We use C2D [60] network as the backbone for all experiments, whose architecture (with ResNet-18) is shown in Table 7. In C2D, all convolution operations are 2D (*i.e.*, the kernel size at the temporal dimension is always 1). Temporal information exchanging is accomplished via temporal pooling. As a result, the model can be initialized from pre-trained parameters on ImageNet. Standard cross-entropy loss is used to guide video classification.

Model Optimization and Evaluation: All models are trained via SGD with a mini-batch of 128 video snippets. To resist overfitting, we adopt several standard engineering tricks, including data augmentation (random horizontal flipping, random cropping and scale jittering [59] etc.), weight decaying with a rate of 5^{-4} and a dropout ratio of 0.5. On a private GPU cluster with 8 NVIDIA 1080TI, training such a model often requires about 2 days before full convergence.

The performance metric is standard multi-class classification accuracy. Following common practice, both top-1 and top-5 accuracies are reported for comparison. When evaluating a model on

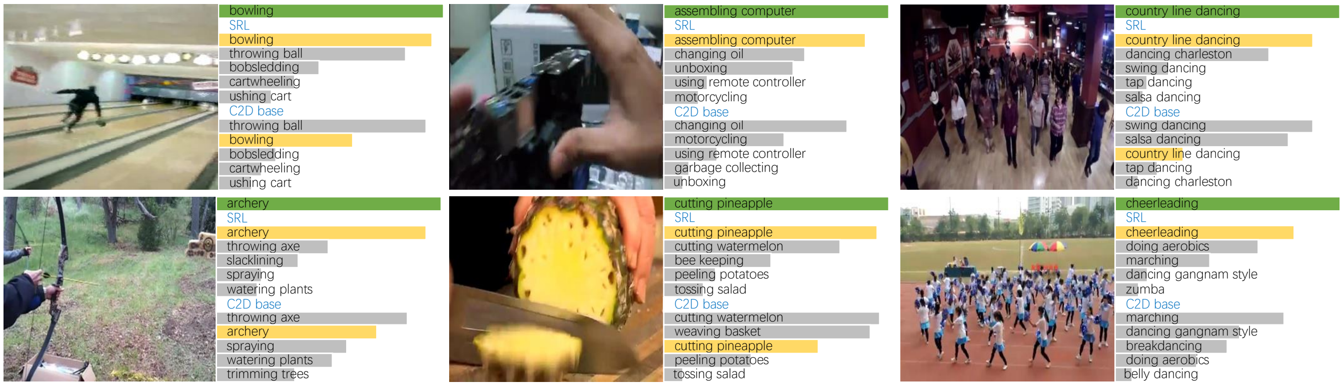

Figure 5: Illustration of how SRL benefits video classification. Ground-truth labels are displayed in green. Top-5 best guesses by SRL and C2D base model are shown. We depict three major cases that require large spatio-temporal receptive field. Left column: rapid object motions; Mid column: closeness to the camera; Right column: crowd motions.

Method	Top-1 Acc.	Top-5 Acc.	# Params(KΔ)	GFLOPs(Δ)
C2D base	62.75	83.82	0	0
NL [60]	64.48	85.48	321.50	1.23
SRL-LO	**65.02**	**85.60**	86.89	0.21
SRL-FFT	64.65	85.52	**86.06**	**0.08**

Table 8: Investigation of different instantiations of SRL blocks using C2D base network for the video classification task.

+N Blocks	Top-1	Top-5
C2D base	62.75	83.82
+1 block	64.23	85.16
+2 blocks	**64.79**	**85.71**
+4 blocks	64.65	85.52

φ operation	Top-1	Top-5
C2D base	62.75	83.82
channel-only	64.13	**85.18**
time-only	63.80	85.08
channel+time	**64.23**	85.16

Table 9: Effect of key factors in SRL. Left: Different number of SRL blocks are added. Right: Different φ in SRL blocks.

some testing videos, we randomly sample 25 clips and average their classification scores, same to the setting in [48, 59].

Instantiations: Table 8 compares different types of SRL blocks added to the C2D baseline. For all models, an individual non-local block is inserted after each residual block of res$_3$ and res$_4$, totaling 4 blocks. On the one hand, the proposed SRL blocks achieve superior classification accuracies in comparison with state-of-the-art non-local video classification model [60], using significantly fewer network parameters and FLOPs. In particular, SRL-FFT has only 0.08 extra GFLOPs (6.5% of NL) and 86.06K parameters (26.8% of NL). On the other hand, SRL-LO is slightly better yet much slower. We constrain all the rest experiments with SRL-FFT considering limited GPU resources.

Ablation Study: The left panel in Table 9 shows the effect of different number of SRL blocks. We contrast the insertion of 1 block (after res$_4$), 2 blocks (after res$_3$ and res$_4$) and 4 blocks (after every residual block in res$_3$ and res$_4$). As seen, a single non-local block can bring a significant improvement over the baseline. More blocks improve with marginal benefit.

Method	#Frames	Top-1 Accuracy	Top-5 Accuracy
I3D [5]	64	71.1	89.3
ARTNet [58]	16	70.7	89.3
S3D [66]	64	72.2	90.6
R(2+1)D [54]	-	72.0	90.0
C2D base	8	70.5	89.4
C2D base + SRL-FFT	8	71.9	90.3
C2D base + SRL-FFT*	8	**72.7**	**90.9**

Table 10: Comparisons with state-of-the-art methods on the Kinetics video benchmark. This study adopts C2D-ResNet50 as base model. SRL-FFT* represents a variant without using group convolution (thus there are more parameters in φ).

Figure 3 elaborates on separable spatio-temporal operation of SRL block for video data. Table 9 shows the results of applying feature channel-only (i.e., $conv_c$) or time-only operation (i.e., $conv_t$). It indicates that either of these two kinds of operations can bring a significant improvement since both have grasped global information from feature maps. They are also complementary to each other since concurrent use proves slight performance gain.

Comparisons with State-of-the-art Methods: Table 10 summarizes results of some state-of-the-art methods. Our budgeted GPU resources only allow a mini-batch of 8 clips per GPU, totaling 64 clips on an 8-GPU machine. Nevertheless, our method surpasses many state-of-the-art methods which have more stacked video frames. Choosing φ with more parameters (such as SRL-FFT* in Table 10) can bring further improvement. Figure 5 provides concrete examples for visually understanding SRL blocks.

5 CONCLUSION

This work proposes SRL, a novel non-local method for learning under global receptive field. The key idea is to convert data into some spectral domain via efficient bilinear unitary transforms. We carefully design several spectral operators and empirically validate SRL on human pose estimation and video classification. Strong evidence is observed to demonstrate its effectiveness.

Acknowledgments: This work is supported by Beijing Municipal Commission of Science and Technology under Grant Z181100008918005, National Natural Science Foundation of China (NSFC) under Grant 61772037.

REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. *CoRR* abs/1609.08675 (2016).

[2] N. Ahmed, T. Natarajan, and K. R. Rao. 1974. Discrete Cosine Transform. *IEEE Trans. Computers* 23, 1 (1974), 90–93.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2009. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*. 1014–1021.

[4] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2010. Monocular 3D pose estimation and tracking by detection. In *CVPR*. 623–630.

[5] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*. 4724–4733.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 834–848.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR* abs/1706.05587 (2017).

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*.

[9] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. 2018. Aˆ2-Nets: Double Attention Networks. In *NIPS*. 350–359.

[10] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *CVPR*. 7103–7112.

[11] James Cooley and John Tukey. 1965. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comp.* 19 (1965), 297–301.

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable Convolutional Networks. In *ICCV*.

[13] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2005. Pictorial Structures for Object Recognition. *IJCV* 61, 1 (2005), 55–79.

[14] B. J. Fino and V. R. Algazi. 1976. Unified Matrix Treatment of the Fast Walsh-Hadamard Transform. *IEEE Trans. Comput.* 25, 11 (1976), 1142–1146.

[15] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. 2019. Dual Attention Network for Scene Segmentation. (2019).

[16] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*. 580–587.

[17] Georgia Gkioxari, Bharath Hariharan, Ross B. Girshick, and Jitendra Malik. 2014. Using k-Poselets for Detecting People and Localizing Their Keypoints. In *CVPR*. 3582–3589.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, Vol. 1. 3.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[20] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR*.

[21] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. 2018. CCNet: Criss-Cross Attention for Semantic Segmentation. *arXiv preprint arXiv:1811.11721* (2018).

[22] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model. In *ECCV*. 34–50.

[23] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*. 448–456.

[24] Umar Iqbal and Juergen Gall. 2016. Multi-person Pose Estimation with Local Joint-to-Person Associations. In *ECCV*. 627–642.

[25] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2013), 221–231.

[26] Leonid Karlinsky and Shimon Ullman. 2012. Using Linking Features in Learning Non-parametric Part Models. In *ECCV*. 326–339.

[27] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *CVPR*. 1725–1732.

[28] Yitzhak Katznelson. 1976. *An Introduction to Harmonic Analysis.* Dover.

[29] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017).

[30] Dong Liu Ke Sun, Bin Xiao and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*.

[31] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

[33] Xiangyang Lan and Daniel P. Huttenlocher. 2005. Beyond Trees: Common-Factor Models for 2D Human Pose Recovery. In *ICCV*. 470–477.

[34] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*. 740–755.

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.

[36] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. 2018. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*. 7834–7843.

[37] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*. 483–499.

[38] Juan Carlos Niebles, Chih-Wei Chen, and Fei-Fei Li. 2010. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In *ECCV*. 392–405.

[39] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. 2014. Multi-source Deep Learning for Human Pose Estimation. In *CVPR*. 2337–2344.

[40] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. 2016. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *CVPR*. 4929–4937.

[41] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. 2012. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*. 3178–3185.

[42] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In *ICCV*. 5534–5542.

[43] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. 2005. Strike a Pose: Tracking People by Finding Stylized Poses. In *CVPR*. 271–278.

[44] P. A. Regalia and S. K. Mitra. 1989. Kronecker Products, Unitary Matrices, and Signal Processing Applications. *SIAM Rev.* 31, 4 (1989), 586–613.

[45] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149.

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115, 3 (2015), 211–252.

[47] Leonid Sigal and Michael J. Black. 2006. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In *CVPR*. 2041–2048.

[48] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*. 568–576.

[49] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[50] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. 2018. IGCV3: Interleaved Low-Rank Group Convolutions for Efficient Deep Neural Networks. In *BMVC*. 101.

[51] Min Sun and Silvio Savarese. 2011. Articulated part-based model for joint object detection and pose estimation. In *ICCV*. 723–730.

[52] Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *CVPR*. 1653–1660.

[53] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*. 4489–4497.

[54] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*. 6450–6459.

[55] Gül Varol, Ivan Laptev, and Cordelia Schmid. 2018. Long-Term Temporal Convolutions for Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 6 (2018), 1510–1517.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 6000–6010.

[57] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. 2009. Evaluation of Local Spatio-temporal Features for Action Recognition. In *BMVC*. 1–11.

[58] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. 2018. Appearance-and-Relation Networks for Video Classification. In *CVPR*.

[59] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*. 20–36.

[60] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local Neural Networks. *CVPR* (2018).

[61] Yang Wang and Greg Mori. 2008. Multiple Tree Models for Occlusion and Spatial Constraints in Human Pose Estimation. In *ECCV*. 710–724.

[62] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *CVPR*. 4724–4732.

[63] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. 2015. Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. In *ACM Multimedia*. 461–470.

[64] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. 2018. AdaFrame: Adaptive Frame Selection for Fast Video Recognition. *CoRR* abs/1811.12432 (2018).

[65] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple Baselines for Human Pose Estimation and Tracking. In *ECCV*. 472–487.

[66] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2017. Rethinking Spatiotemporal Feature Learning For Video Understanding. *CoRR* abs/1712.04851 (2017).

[67] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. 2018. Compact Generalized Non-local Network. In *NIPS*. 6511–6520.

[68] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. 2018. Self-Attention Generative Adversarial Networks. *CoRR* abs/1805.08318 (2018).

[69] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid Scene Parsing Network. In *CVPR*. 6230–6239.

[70] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. 2018. ECO: Efficient Convolutional Network for Online Video Understanding. In *ECCV*.