TESA: Tensor Element Self-Attention via Matricization

Francesca Babiloni¹, Ioannis Marras¹, Gregory Slabaugh¹, Stefanos Zafeiriou²

¹Huawei Noah’s Ark Lab  ²Imperial College London

Abstract

Representation learning is a fundamental part of modern computer vision, where abstract representations of data are encoded as tensors optimized to solve problems like image segmentation and inpainting. Recently, self-attention in the form of a Non-Local Block has emerged as a powerful technique to enrich features, by capturing complex interdependencies in feature tensors. However, standard self-attention approaches leverage only spatial relationships, drawing similarities between vectors and overlooking correlations between channels. In this paper, we introduce a new method, called Tensor Element Self-Attention (TESA) that generalizes such work to capture interdependencies along all dimensions of the tensor using matricization. An order R tensor produces R results, one for each dimension. The results are then fused to produce an enriched output which encapsulates similarity among tensor elements. Additionally, we analyze self-attention mathematically, providing new perspectives on how it adjusts the singular values of the input feature tensor. With these new insights, we present experimental results demonstrating how TESA can benefit diverse problems including classification and instance segmentation. By simply adding a TESA module to existing networks, we substantially improve competitive baselines and set new state-of-the-art results for image inpainting on CelebA and low light raw-to-rgb image translation on SID.

1. Introduction

Deep Convolutional Neural Networks (DCNNs) represent the state-of-the-art method in a variety of computer vision problems but, in their standard implementation, they are limited to compute only local regions of the input. This innate characteristic makes long-range dependencies, which are a key aspect in a variety of tasks, hard to capture without the use of circumventing techniques. The use of deeper stacks of convolutional layers, for instance, increases neurons’ receptive fields [31] at the cost of optimization difficulties [20] and higher complexity [38]. Recently, more sophisticated layers (e.g. Non-Local Blocks) have been proposed, which directly leverage these interdependencies as a means to enrich the intermediate CNN representations [41, 47, 39, 49, 28]. These blocks have been proven useful, beating competitive baselines in video action recognition, classification and instance segmentation. At the same time,
the majority of these methods try to estimate only spatio-
temporal correlations among positions of the input tensor
[49, 41, 28] or overlook its complex topology [47, 3]. In this
paper, we build upon the aforementioned line of research
and extend its scope to the goal of mining tensor element
interactions from the input. Our three main contributions
can be summarized as follows:

- We propose a new self-attention block (TESA) able to
  leverage correlations in all possible directions of the
  input tensor to take advantage of channel information
  without losing the topology of the input tensor. Instead
  of completely flattening the elements of the input in
  a single vector and facing intractable complexity, we
  propose to use tensor matricizations as a way to extract
  complex interactions (Figure 1).

- We provide a statistical interpretation of the proposed
  family of non-local blocks. In particular, we demon-
strate that our block can be seen as an operator act-
ing as a regulariser of the spectrum (i.e. the variance)
of the various matricizations of the feature tensor. We
prove from a theoretical and empirical perspective how
TESA adjusts the relative importance of the singular
values. This is achieved implicitly without the need
to compute an expensive singular value decomposition
(SVD) in a direct way.

- We demonstrate the power of TESA in a battery of het-
erogeneous computer vision tasks. Our method shows
a consistent improvement in large-scale classification,
detection, instance segmentation and puzzle-solving.
It also achieves state-of-the-art performance in the two
dense image-to-image translation problems of inpainting
and short-exposure-raw to long-exposure-rgb.

2. Related work

Self-similarites The concept of similarity among
image parts or video frames is pivotal in many computer
vision applications. Therefore, a long-lasting trend in
the community has been understanding how to properly define
and exploit self-similarity. The idea of relating features to
each other (i.e. CNN’s channels or classical descriptors)
has inspired various pooling methods [23, 27, 12, 6, 25, 4]
where correlation is used as a higher order representation
for the image and fed to a classifier. Simultaneously,
a complementary line of research proposed techniques
to relate an image part to its context using both classi-
cation and has recently received interest in various applications
of machine learning. In machine translation, self-attention
vectors assess how strongly each element attends (i.e.
is correlated) to all the others and estimate the target as
the sum of all elements in a sentence, weighted by their
attention values [2, 40]. Variants of self-attention have
been used in computer vision to solve a variety of problems
ranging from inpainting [45] to zero-shot learning [44] and
visual question answering [35, 37]. Noticeable examples
can be found in classification, where it has been used to
estimate attention-masks for intermediate CNN features
[32] or to learn re-calibration of features given global
channels descriptors [19]. Recently, the Non-Local Block
has been proposed as a plug-and-play extension to existing
architectures. The purpose of this block is to enrich
features using spatial-temporal interaction, considering all
position at once [41, 49, 39] or a single position and its
neighborhood [28]. This formulation inspired new deep
learning architectures [48, 11, 7] and has been extended
in recent works to the scope of integrating with the input
compact global descriptor of feature maps [47, 3, 43].

3. Method

In this section, we introduce the notation used through-
out the paper, give an overview of the concept of self-
attention and describe in detail the proposed method. At
first, we study the spatial version of our non-local block.
Next, we generalize to the scope of capturing more complex
tensor interdependencies. Finally, we relate our method to
other existing non-local blocks.

3.1. Notation

In the rest of the paper, we adopt the notation of Kolda
et al. in [22]. Tensors are denoted using calligraphic let-
ters (e.g. \(\mathcal{X}\)) and matrices by bold upper-case letters (e.g.
\(\mathbf{X}\)). The \(i^{th}\) row of a matrix \(\mathbf{X}\) is a vector denoted
using lower-case bold letters as \(\mathbf{x}_i\). The order \(N\) of a ten-
sor corresponds to the number of its dimensions and can
be also called mode. A mode-\(n\)-fiber of a tensor is the
vector obtained by fixing all indices of \(\mathcal{X}\) except for the
\(n^{th}\) dimension and can be seen as a generalization of ma-
trix’s rows and columns. The mode-\(n\)-matricization of a
tensor \(\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}\) is a case of matricization de-
noted as \(\mathbf{X}_{(n)}\) and arranges its mode-\(n\)-fibers to be the
columns of the resulting matrix. More formally, the ten-
sor elements \((i_1, i_2, \ldots, i_N)\) are rearranged into the matrix
element \((i_n, j)\) where \(j = 1 + \sum_{k=1, k\neq n}^{N} (i_k - 1)J_k\) and
\(J_k = \prod_{m=1, m\neq n}^{k-1} I_m\).

3.2. Overview of self-attention

Given an input matrix \(\mathbf{X}\), an attention mechanism
weights \(\mathbf{X}\) with an attention matrix \(\mathbf{A}\) to highlight the rel-
evant parts of the input. Different ways of computing \(\mathbf{A}\) en-
tail different variants of attention mechanisms. This paper
focusing on self-attention, where weights are only a function of input $X$. In particular, we consider a pairwise function $f$, which can be used to capture interdependencies between each $x_i$ and every $x_j$. A self-attention block is a variation of a residual block [18] which sums the output of a self-attention mechanism to the original input $X$. The output $Y$ of the self-attention block is expressed as follows:

$$Z = X + AX = X + f(X, X).$$  (1)

### 3.3. Capturing spatial correlation

Let a $3^{rd}$ order tensor $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ be the feature map output of one of the layers of a CNN. Let $\mathcal{X}$ be rearranged, using its mode-$c$ matricization, as a $X_{(c)} \in \mathbb{R}^{WH \times C}$ where each spatial position is described by its $C$ features. Let’s assume $X_{(c)}$ is mean normalized. In the linear version of the proposed block, we choose as attention matrix the covariance $X_{(c)}X_{(c)}^\top \in \mathbb{R}^{WH \times HW}$, which expresses the correlations between each $i^{th}$ position and every $j^{th}$ position. Thus, the output of the spatial self-attention block using this mechanism can be written as follows:

$$Z = \alpha_cX_{(c)} + \beta_cX_{(c)}X_{(c)}X_{(c)}$$  (2)

where $\alpha_c$ and $\beta_c$ are learnable scalars modulating the contribution of each term. In Eq. 2, the global covariance term modulates the feature’s representation with spatial similarities. The residual term, together with the two learnable scalars allows the implicit regularization of the spectrum via a polynomial function. In the following, we draw a connection between the input and the output of the self-attention block, omitting the subscripts to simplify the notation. The matrix $X$ and its positive, semi-definite covariance matrix have the following singular-value and eigen decomposition:

$$X = U\Sigma V^\top, \quad XX^\top = Q\Lambda Q^\top = U(\Sigma^2)U^\top$$  (3)

where $Q = U$ is the eigenvectors matrix, $V^\top$ and $U$ are the right and left singular vectors, $\Lambda$ is the eigenvalue matrix and $\Sigma$ its corresponding singular value diagonal matrix. Notably, $\Lambda = \Sigma^2$. Thus, the $\beta$ parameter learns to modulate the contribution of the following term:

$$XX^\top X = U\Sigma^2 U^\top U\Sigma V^\top = U\Sigma^3 V^\top$$  (4)

From the above, it is evident that using the proposed self-attention block changes the spectrum of $X$ as

$$\alpha X + \beta(XX^\top)X = U(\alpha \Sigma + \beta \Sigma^3)V^\top$$  (5)

Hence, the self-attention block described in Equation (2) learns the coefficients of a polynomial function of the singular values, without operating on the input’s orthogonal vectors $U$ and $V^\top$ and it can be seen as an operator that modifies the singular values of the input matrix $X$ without the need of a direct and expensive SVD computation. Only two learnable parameters, $\alpha$ and $\beta$, are used for this purpose. Since $\alpha$ and $\beta$ can be either positive or negative, the method performs an algebraic sum of two functions and, therefore, has the flexibility to regularise the spectrum by performing shrinkage or whitening.

#### 3.4. Capturing tensor elements interdependencies

In the previous subsection, the choice of unfolding the tensor as a matrix $X_{(c)} \in \mathbb{R}^{WH \times C}$ drives the focus of the attention mechanism to capture only spatial similarities. In the following, we introduce a generalization that leverages both spatial and channel-based correlations, while keeping intact the module’s effect on the spectrum. As depicted in Figure 1, the proposed generalisation represents the feature tensor $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ using its three mode matricizations $X_{(c)}$, $X_{(h)}$ and $X_{(w)}$, each embedded in a different subspace via a weight matrix $W$, followed by a non-linear function $\sigma$:

$$Y_{(n)} = \phi(X_{(n)}) = \sigma(X_{(n)} W_{(n)}) \quad n \in \{C, H, W\}.$$  (6)

In our implementation, $\sigma$ is a ReLU activation function [33] and the weight matrices $W_{(c)} \in \mathbb{R}^{C \times C}$, $W_{(h)} \in \mathbb{R}^{H \times H}$, $W_{(w)} \in \mathbb{R}^{W \times W}$ correspond to $1 \times 1$ convolutions in the tensor space over each respective dimension.

Then, a self-attention block is applied independently to each $Y_{(n)}$ and the three contributions are reshaped and combined via summation to generate the final output $Z$.

$$Z = \sum_n^{C,H,W} \Psi_{(n)}(\alpha_n Y_{(n)} + \beta_n Y_{(n)} Y_{(n)}^\top Y_{(n)}),$$  (7)

where $\Psi_{(n)}$ is a reshape function which rearranges the matrices as tensors of dimension $H \times W \times C$. In the above equation, each embedded matricization represents a different point of view on the input tensor: $Y_{(c)}$ accounts for spatial interactions, $Y_{(w)}$ for interactions between rows and channel activations and $Y_{(k)}$ for interactions between columns and channels. Our method processes each $Y_{(n)}$ in its own space, modulating its representation with a self-attention block as described in Eq. 2. Thus, it is not limited to capture only correlations among positions but is also capable of capturing correlations across channels. In order to be considered simultaneously, the three contributions are fused in tensor space (i.e. overlaid in the same coordinate-space). The fusion through summation ensures i) the same dimensionality of input and output and ii) equal contribution for each term. As depicted in Figure 1, although the output of each self-attention block encapsulates similarity between pairs of vectors, their summation allows the
method to directly relate tensor elements to each other. We call our method Tensor Element Self-Attention or TESA.

3.5. Relation with Other Self-Attention Blocks

In this section, we connect TESA with other self-attention works. The non-local block [41] and its variants [49, 39, 28] explore the introduction of non-local information in a neural network and can be framed as self-attention methods investigating spatial correlation. In the formulation that is closest to ours, the non-local block applies three learnable weights matrices ($W_θ$, $W_φ$ and $W_γ$) on the same input $X$ [41]. The first two matrices are in charge of extracting spatial long-range dependencies using a dot-product similarity, while the third one embeds the input.

Given $X \in \mathbb{R}^{WH \times C}$, the output $Z$ of the original non-local block is: $Z = X + \text{softmax}(XW_θ(W_φ^T X^T)XW_γ)$. Our goal is to generalize the self-attention mechanism to more complex interactions without overlearning channel information. Therefore, our block embeds each tensor mode separately and aims to extract different correlations from each embedding. Recent works propose to leverage channel information by estimating a scalar global description of each embedding [43] or tensor feature maps [47]. In the case closest to ours in [47], the method divides the input in $G$ separate groups $X_i \in \mathbb{R}^{WH \times C}$ and extracts a global representation ($\mathbb{R}^1$) for each of them. On the contrary, we tackle the problem from a complementary perspective, with a formulation that focuses on the explicit computation of tensor elements’ pairwise correlation and provides interpretability regarding the self-attention effect on the features.

4. Illustrative experiment

One of the goals of self-attention mechanisms is to equip a model with the capacity to reason about the whole input representation at one glance. We first test this property in a controlled scenario, designing a new “puzzle MNIST” experiment. We used the MNIST dataset and a four-layer fully convolutional encoder-decoder architecture. To test the ability of our self-attention method to make use of available but scattered information, we attempted the reconstruction of an image given its shuffled version. To obtain an input puzzle, each image is split into 16 tiles of equal size. These tiles are then randomly rotated and mirrored before being stitched back together.

Input and output samples can be seen in Figure 2e and 2a, respectively. Between the encoder and the decoder part of the network, the self-attention module integrates information about positions or tensor self-similarity.

4.1. Capturing spatial correlations

We start by analyzing the spatial self-attention block as formulated in Equation 2. To highlight the effect of self-attention in the latent space, we compared a model trained without any attention ($\alpha = 1, \beta = 0$) and two variants of our block: one where $\alpha$ and $\beta$ are fixed to be equal to unity and the other where they are treated as learnable scalars. The first row of Figure 2 shows a qualitative overview of our comparisons. The baseline is limited to process the input locally and performs worse than models trained with self-attention. The second row shows comparisons on the empirical distribution of singular values for the puzzle MNIST test set. Given a sample of the test-set, we extracted features before and after the self-attention block, returning for each image two matrices $X_{in}$ and $X_{out}$. As explained in Section 3, the left and right singular vectors are left untouched by the method. Consequently, the relation between input and output can be computed using only the $\alpha$ and $\beta$ parameters and the effect of self-attention can be captured plotting the singular value spectrum of input and output side-by-side. Figures 2g and 2h show the singular values of the input $X_{in}$ (in blue) and the singular values of the output $X_{out}$ (in white) plotted in descending order. The red bars depict the prediction for $\Sigma_{out}$ obtained using Equation 5. Comparing input and output in each plot shows how the attention block shrinks the spectrum of the input, automatically choosing which information (i.e. components) is highlighted and which is suppressed to simplify the subsequent decoding task. Moreover, it shows how the direct SVD computation matches closely the theoretical prediction. The comparison between the two plots shows how the possibility to learn the spectrum transformation (Figure 2h) retains more expressive components compared to the fixed contribution of self-attention and input (Figure 2g).

For example, in Figure 2h the drop between the first and second singular value is substantially smaller (30% drop) than what occurs in the case of $\alpha$ and $\beta$ fixed to one (60% drop).

4.2. Capturing tensor elements interdependencies

The same logic can be used to analyze the generalized case of Equation 7. As a first step, we extended the linear spatial case to consider channel based interdependencies. This case is equivalent to substitute $Y_n$ with $X_n$ in Equation 7. This allows a comparison in the same latent space for input ($X$) and output ($Z$) tensors, and gives the possibility to inspect directly their matricization’s spectrum. In $3b$, $3c$, $3d$ plots, each mode matricization ($H$, $C$, $W$) is treated separately, showing the comparison between input and output of each self-attention. The figures depict how self-attention produces a shrinkage effect on all mode matricizations. The possibility to correlate channels with rows and columns patterns modifies the role of $X_{(c)}$. Its spectrum is drastically reduced to have only two meaningful components, accounting for more than 99% of the whole variance. Figure 3a shows sample outputs for our method, which is not limited to spatial similarity but can leverage multiple views on the original tensor. It produces considerably sharper outputs
Figure 2: **Spatial Self-Attention Overview**. The first row shows reconstructed digits. The baseline with no attention is outperformed using spatial-correlation. The best quality is achieved by $\alpha$ and $\beta$ learnable scalars. Features’ singular values show how self-attention drives the first principal components to account for the majority of the variance in the matrix. Computing the output spectrum empirically (white bars) or using Equation 5 (red bars) yields very close results. Blue plots differ due to the different embeddings learned by the architectures.

Figure 3: **Tensor Element Self Attention Overview**. The qualitative comparisons showcase the benefits of our method, which makes use of channel information to produce more defined output.

when compared with previous cases.

To discuss the case described in Equation 7, we have to extend the analysis to consider the embedded mode matricizations, $Y_c$, $Y_h$, $Y_w$. In this case, the input $X$ and $Z$ tensors live in different subspaces, due to the learnable parameters of the projection matrices $W_c$, $W_h$, $W_w$. Each mode matricization is embedded separately, but each self-attention operates directly on its input without any additional transformation. Thus, the input/output pairs of each self-attention still share the same orthogonal vectors and their spectrum can be still compared and used to highlight the impact of the self-attention module on each latent space. In the next section, we will report this effect on different problems and datasets. On ”shuffle MNIST”, the use of embeddings produce shrinking trend and qualitative output similar to those reported in Figure 3.

### 5. Experiments

We evaluated TESA on a series of computer vision problems, ranging from dense image-to-image translation to detection. This section starts by presenting results on two dense tasks based on an encoder-decoder architecture, where the self-attention block is used to enrich the encoded features representation. Then, our analysis is extended to the case of a ResNet architecture used for classification and as the backbone for instance segmentation.

#### 5.1. Short exposure Raw to Long exposure rgb

Initially, we address the task of reconstructing a high-quality long-exposure rgb image given a noisy short-exposure sensor raw image captured in low-light conditions. In digital photography, an Image Signal-processing Pipeline (ISP) transforms raw data collected by an image sensor into
The qualitative comparison shows how the method of [5] can be improved by the use of self-attention. Our method is able to recover cleaner patterns and generate outputs without strong color artifacts. The second row shows the singular value plots of the input/output pair for each mode matricization. The trend depicts the whitening effect that the self-attention has on the input spectrum. Images visible better zoomed in electronically.

The Sony camera uses a Bayer sensor pattern to capture a single raw frame with short exposure. Simultaneously, the camera shot a reference rgb image increasing the exposure factor of 100 or 300 times used as ground-truth by the network. Table 1b reports the reference metrics PSNR and SSIM obtained by the different methods. New state-of-the-art performance is achieved by powering the architecture with our self-attention block. Qualitatively, our method shows better details and color recovery when compared with the competitors (Fig. 4). Figures 4e, 4f, 4g depict the effect of TESA on the singular values. In this case, the input spectrum of $Y_h$, $Y_w$, and $Y_c$ is whitened; note that the input (dark blue bars) singular values fall off quickly (e.g. exponentially), whereas self-attention with TESA rebalance their intensities and produces an output (white bars) where the singular values fall off more gradually (e.g. approximately linearly).

5.2. Inpainting - CelebA

Image inpainting requires missing pixels in the input image to be filled in. An inpainting algorithm hallucinates the missing image pixels and blends them in with the surrounding regions in a coherent manner, producing a real-
Figure 5: **Inpainting Qualitative Comparison.** The Partial Convolution baseline (Pconv) creates blurry and artificial outputs that are partially improved by the use of self-attention (e.g. PConv + NL implementing the non-local block). Our method (right), leverages similarities across multiple dimensions and produces consistent colors and realistic details.

### Table 1: Quantitative Comparisons: Unet for inpainting and Raw-to-rgb.

Reconstruction metrics for the inpainting and short-exposure-raw to long-exposure-rgb tasks. Experiments employ different variants of the Unet architecture: 1 attention block is added for raw-to-rgb and 3 attention blocks for inpainting. State-of-the-art performance can be achieved by using our method. Asterisks ‘*’ indicate results produced using software provided by the authors.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>MS-SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PConv [29]</td>
<td>25.36</td>
<td>0.877</td>
<td>0.928</td>
</tr>
<tr>
<td>+Wang [41]</td>
<td>25.41</td>
<td>0.881</td>
<td>0.928</td>
</tr>
<tr>
<td>+Yue [47]</td>
<td>25.66</td>
<td>0.888</td>
<td>0.931</td>
</tr>
<tr>
<td>+Ours</td>
<td>26.00</td>
<td>0.895</td>
<td>0.936</td>
</tr>
</tbody>
</table>

(a) Inpainting - CelebA. (Random and Center Crop Evaluation)

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID [5]</td>
<td>28.88</td>
<td>0.787</td>
</tr>
<tr>
<td>+Wang [41]</td>
<td>28.57</td>
<td>0.884</td>
</tr>
<tr>
<td>+Yue [47]</td>
<td>29.54</td>
<td>0.888</td>
</tr>
<tr>
<td>+Ours</td>
<td>29.79</td>
<td>0.891</td>
</tr>
</tbody>
</table>

(b) Raw-to-rgb - SID Sony

The CelebA dataset [30], which consists of more than 202K samples, was used in our experiments. The training data were generated by randomly cropping a 128x128 patch from each training sample (a quarter of the input image). Table 1a reports PSNR, SSIM, MS-SSIM for the methods. TESA achieved the best results for all the evaluated metrics and generates convincing images with rich details and reduced artifacts (e.g. more defined wrinkles).

### 5.3. Instance Segmentation - MS-COCO

The task of image instance segmentation requires the detection and segmentation of each item in the input image, differentiating among instances. It outputs a per-pixel mask that identifies both the category and the instance for each object. The baseline model for these experiments is the two stages Mask R-CNN [16]. The first Region Proposal stage (RPN) uses a network that serves as “attention” for the entire pipeline: it takes an image as input and outputs a set of rectangular object proposals. We tested the capacity of the self-attention block to enrich the representation of RPN features. Following the implementation of related work, we added a self-attention block right before the last residual block of the ResNet50 feature extractor, reducing the channel dimension while embedding the modes’ matricizations. To bring back the overall output to the original channel dimension, we used one extra convolution and a weighted global skip connection. We compared against the original implementation of [16], trained end-to-end, and its non-local block extension [41, 49]. Please refer to the original papers and code.

---

4 https://github.com/NVIDIA/partialconv

5 https://github.com/latentgnn/LatentGNN-V1-PyTorch
Table 2: Quantitative Comparisons: ResNet. Performance metrics for the task of classification on the Imagenet dataset [34] and object detection and instance segmentation on COCO [26]. Results are based on ResNet50 and Mask R-CNN with a ResNet50-FPN backbone. Both use one single attention block. Asterisks ‘*’ indicate results obtained using software provided by the authors.235. Yue et al. did not converge during our training and is not reported in table (b).
References

[1] Deepisp: Toward learning an end-to-end image processing pipeline. 4326

[29] Guolin Liu, Fitzum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for


