What You See Is What You Transform: Foveated Spatial Transformers as a bio-inspired attention mechanism

Ghassan Dabane
Polytech Marseille
Aix Marseille Univ
Marseille, France
ghassan.dabane@etu.univ-amu.fr

Laurent U Perrinet
Institut de Neurosciences de la Timone
Aix Marseille Univ, CNRS
Marseille, France
laurent.perrinet@univ-amu.fr

Emmanuel Daucé
Institut de Neurosciences de la Timone
Centrale Marseille, CNRS
Marseille, France
emmanuel.dauce@centrale-marseille.fr

***Abstract*—Decoding the semantic content of images is nowadays dominated by the use of deep convolutional neural networks (DCNNs). However, their generalization capability is still undermined by the small translation invariance of their max-pooling layers. Taking inspiration from biological vision, we develop here a new methodology for translation-invariant processing with DCNNs. We build upon a recent model that implements two key biological mechanisms: foveated vision and the separation of the visual processing into a "what" and a "where" pathways. Alongside such foveal vision, we demonstrate the capability of a foveated spatial transformer to learn both pathways in an end-to-end fashion, without any spatial labelling whatsoever. Our results pave the way towards a new class of spatial visual transformers, implementing the principles of active (saccadic) vision over large visual displays.**

***Index Terms*—Neural Models of Perception, Visual system, Attention, Bioinspired and Biomorphic Systems, Brain-inspired cognitive architectures.**

I. Introduction

SINCE the emergence of AlexNet; the winner of the 2012 ILSVRC image classification competition [2], computer vision has been dominated by the use of deep convolutional neural networks (DCNNs) [3] to capture the semantic content of images. Nowadays, most classifiers are capable of surpassing human level performance on specific visual categorization challenges [4]. From object recognition [5], [6] and natural language processing tasks [7], [8], to lymph node metastasis detection [9] and diagnostic radiology in patient care [10], there is no questioning the breadth of their applications throughout various fields. Thanks to the massive sharing of weights in convolutional layers inside their architecture, DCNNs keep the number of parameters to be learned relatively small, which facilitates the abstraction of complex feature spaces. Although DCNNs provide an exceptionally powerful set of architectures for computer vision, they still lack one very important property of a visual processing system: spatial invariance, that is, the ability to separate the object's pose and position from its identity, i.e., its texture and shape. In practice, small invariance to features in the visual space can be achieved by local max-pooling layers embedded within the architecture, but it remains limited in scope due to the small spatial support (e.g., 2×2 pixels), leaving unsolved the invariance to large transformations in input data [11].

On the one hand, the spatial distribution of objects in space can be addressed, in the general case, from the perspective of inverse graphics [12], the subjective shape and position of a given object being the result of a certain number of affine transformations (shifts, rotations, ...) operated in the physical world over object templates. Learning such transformations together with the object identity (that is learning the inverse transformation to operate on the data) is thus expected to implement a wider range of spatial invariance, providing more parsimonious and interpretable classifiers by aligning the visual input to a limited number of object templates. To this end, Spatial Transformers Networks (STN) were introduced [13], a fully differentiable module that can be inserted inside a DCNN at any depth giving it the ability to learn how to actively manipulate and transform input feature maps spatially without any extra supervision (e.g. pose annotation) added to the process, allowing the network to only select relevant regions of the image (attention mechanism). Learning is also performed in an end-to-end fashion with standard backpropagation without modifying the optimization hyper-parameters. State-of-the-art results were achieved on several benchmarks giving DCNNs invariance to several classes of spatial transformations, most notably affine transformations, i.e., scaling, rotation, translation, shear, and reflection.

On the other hand, leveraging spatial transformations comes with a cost: the processing of a transformation itself, implemented in a conventional deep convolutional neural network. Stacking a transformer over a classifier provides a more efficient use of the visual data, at the cost of twice more parameters to tune. The tuning of parameters being the principal cost and concern when time and computational resources are limited, one should consider more parsimonious approaches to inverse transformations. Finding such transformations from

This work was supported by the CNRS and French ANR AgileNeuRobot [grant no ANR-20-CE23-002]. The POLO library was developed by L Perrinet, E Daucé and J Gachot [1].

message passing and node to node agreement was for instance proposed in [14]. Key-Query based visual attention, such as the one proposed in Visual Transformer [15], [16], was also proposed as a solution to effectively implement spatial routing agreement, by analogy with the sequential agreement found in text processing.

Coming from a different line of research, biologically-inspired image processing has converged towards a quite similar description of the main processes taking place in the brain when animals have to detect objects of interest in a visual scene. For the same reason that the Neocognitron [17]; the predecessor of modern DCNNs, was inspired by the discovery of simple and complex cells in the primary visual cortex in mammals [18], the need for architectures that are inspired from biological underlying principles is growing [19]. Indeed, the human visual processing system is still considered unrivalled when it comes to speed of detection and computational efficiency. Bio-inspired artificial visions systems thus concentrate on the most salient aspects of biological visual processing, that is the use of non-spatially homogeneous visual sensors (foveated retina) and the use of eye saccades to shift attention toward different parts of the visual scene. Considering foveated inputs implies, at first hand, the *compression* of the visual data through a center-surround log-polar grid representation, as is the case of the foveated vision in mammals [20]. Next key aspect of natural vision is the use of high-speed eye movements [21], that is shifting the fovea that concentrates most of the photoreceptors, towards specific spatial positions to improve the decoding [22]. Finally, the processing of visual information that is found in mammals is anatomically separated in two pathways, namely the dorsal pathway responsible for the localization, and the ventral pathway responsible for object recognition [23]. Next comes the question: how to effectively implement such principles? And, more importantly, why and how such a specific visual processing is optimal with regards to the physical and ecological constraints found in the natural world?

One recent paradigm for the application of these principles is the artificial What/Where model [24] that combines each previous aspect in a trainable deep convolutional architecture. This model works in a sequential way. The foveated visual input is processed through a first neural network layer, referred to as the "Where" module, in order to determine the optimal viewpoint upon which the agent shall fixate its center of gaze. Next, after moving the eye towards this new position, a second neural network, named the "What" module, will oversee classifying a small region around the center (mimicking the "fovea") to detect the object contained within it. One key result here is the considerable information gain provided by iterating only one saccade over the visual field [25], which allows to link saccade selection to the more general active inference principles [26], [27]. As a side result, the log-polar compression, placed at the entry level of this architecture, provides a complexity (processing time) that is sub-linear with regards to the number of pixels, which is unprecedented with regards to classical computer vision that is still considered

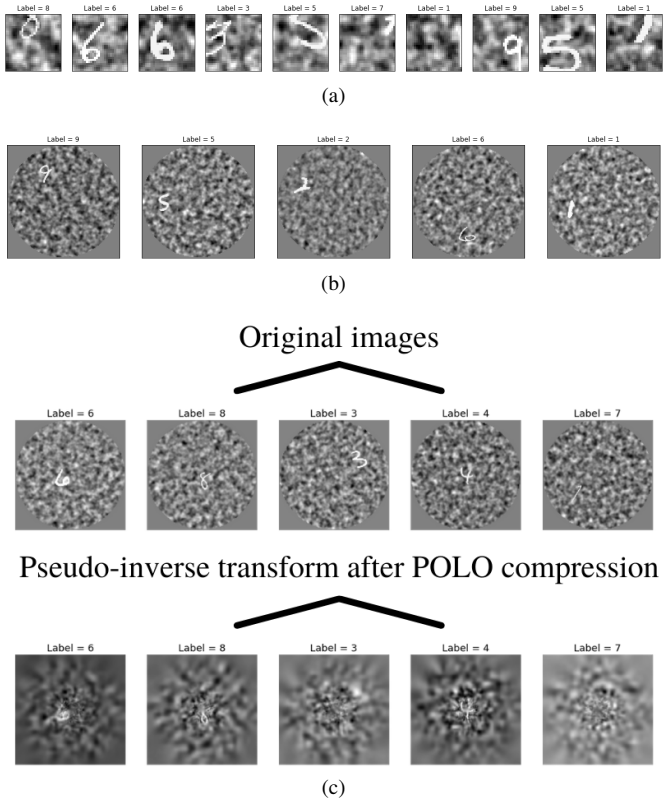

Fig. 1: The datasets that were used for the visual search task. (a) The 28×28 pixel Noisy Shifted MNIST dataset. (b) the 128×128 pixels Noisy Shifted MNIST dataset. (c) Visualizing the Polar-Logarithmic (POLO) version of the 128×128 Noisy dataset by using the pseudo-inverse transform, images are compressed up to 95%

linear in the number of pixels. This decrease in the computational load can be seen as the main reason why such principles are so largely adopted in biology. From a practical standpoint however, the training of each module was done separately, over spatially-shifted targets with a noisy background, providing an explicit spatial labelling of the targets during the training phase. This supervised training, used as a shorthand, renders the method impracticable to implement over real world natural images, assuming that explicit spatial labelling is not provided in general.

To overcome this limitation, we thus developed a new architecture that consists of training a spatial transformer as a replacement for the original "Where" module, and progressively implements the different properties required, namely log-polar compression, saccade selection and foveal downsampling. We were able to demonstrate in a step-by-step fashion that the full What/Where processing pipeline, including the log-polar foveal magnification, saccade selection and foveal processing, can be trained in an end-to-end fashion, *i.e., without supervision of the spatial transformation*.

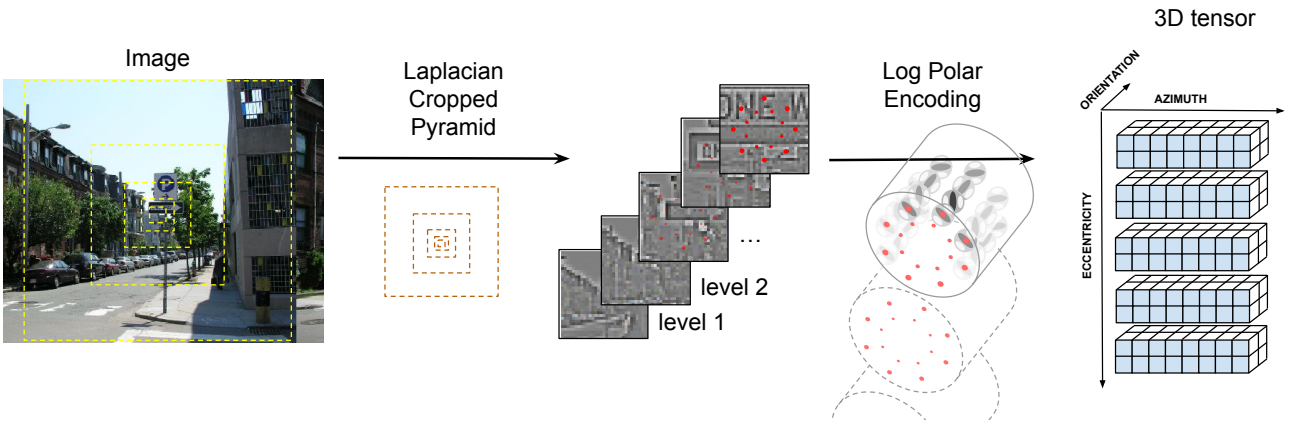

Fig. 2: POLO [1] LogPolar encoding pipeline

II. Materials and Methods

The task at hand is a simple environment where the agent must localize and identify a random handwritten digit inside a big cluttered noisy image, similar to the one described in the original What/Where model [24]: a random handwritten digit is placed inside a screen with added clutter and noise, and the agent's mission is to classify the digit; as in determining its label. However, the digit is placed in a random position and the difficulty of the task will be modified according to two parameters, the eccentricity; the digit's distance from the center point of the image, and the contrast, or the digit's visibility relative to the background. The larger the eccentricity or the lower the contrast, the harder the task. Training datasets are prepared, and networks are implemented in Python, using the high-performance deep learning framework "PyTorch" [28]. All networks are trained on a GTX 1660 Ti GPU, and results are visualized and organized within Jupyter Notebooks using Python's scientific plotting libraries NumPy [29] and Matplotlib [30]. The source code is available at https://github.com/dabane-ghassan/int-lab-book

A. Datasets

The MNIST database [6] is used for this task. It consists of a set of 70000 grayscale images of handwritten digits of size 28×28 split between 60000 training examples and 10000 validation examples. The input data is made of grayscale images containing a MNIST digit placed randomly over a noisy background. Moreover, the digit's contrast varies randomly between 70% and 30%. For the purpose of this application, three variants are prepared. In a first variant, said the "foveal" 28×28 dataset, the input images keep their original size, but a synthetic random texture is added in the background [24], [31] (see Fig. 1a for some examples) and, importantly, the digits are randomly shifted away from the center. Depending on the shift and the contrast, the classification difficulty ranges from "easy" toward "extremely hard" or even impossible when the digit is only partly visible at the border. In a second one, said the 128×128 "full visual field", the MNIST digits are placed randomly over a similar random texture, but the image is much larger (128×128 pixels), and includes a circular mask (of radius 64) (Fig. 1b). In that case, the digit's eccentricity can vary between 0 and 40 pixels, forcing the digit to fit entirely inside the circular mask.

Last, a compressed "LogPolar visual field" dataset, mimics the log-polar encoding of the mammalian retina over the full visual field. The principles of the log-Polar encoding pipeline is presented on Fig. 2. For computational efficiency reasons, the encoding of the image is split in two phases. The image is first recoded with a Laplace pyramid [32], and cropped at the different resolution levels in order to only keep a series of 32×32 or 64×64 snippets of Laplacian coefficients, for the K different levels considered [33], [34]. Then a bank of Log-Gabor filters, radially disposed on a log-polar grid, serves to linearly transform the snippet images into log-gabor coefficients. The resulting coefficients are then stacked in a 3D tensor, organized spatially with their log-polar coordinates, and 8 orientations for the depth. In our case, the original image size is $128 \times 128 = 16384$. The number of levels varies from 3 to 6 depending on the compression rate. Two banks are considered in our experiments. A first bank has 3 levels, each level providing 16 azimuthal, 2 radial and 8 orientation coordinates, making a total of $3 \times 2 \times 16 \times 8 = 768$ predisposed filters, providing a compression rate of approximately 95% $(1 - (768/16384))$. A second keeps the two first precedented levels and adds 2 additional "high-resolution" levels based on 64×64 snippets, having 32 azimuthal, 4 radial and 8 orientation coordinates, making a total of $2 \times 2 \times 16 \times 8 + 2 \times 4 \times 32 \times 8 = 2560$ filters, providing a compression rate of approximately 85% $(1 - (2560/16384))$. It is also worth mentioning that the original What/Where model has a compression rate of about 83% [24], which can be helpful to test and benchmark Spatial Transformers on roughly the same compression rate and also on a higher one. In order to visualize the compressed version of the dataset, it remains possible to represent it in the visual space using the pseudo-inverse of the transform (Fig. 1c). Finally, the radial organization of the log-Gabor filters makes it possible to represent the obtained coefficients on a bi-dimensional grid, using log-polar

TABLE I: Different architectures and their parameters

Parameter \ Network	STN_28×28	STN_128×128	ATN	POLO_ATN	convPOLO_ATN
Dataset	28×28 Noisy	128×128 Noisy	128×128 Noisy	128×128 Noisy + Compressed ($85/95\%$)	128×128 Noisy + Compressed (85%)
Localization Network	2 CN* Layers, 2 FC** Layers	4 CN* Layers, 2 FC** Layers	4 CN* Layers, 2 FC** Layers	Only 2 FC** Layers	2 CN* Layers, 2 FC** Layers
Grid generator	28×28	128×128	28×28 (DS***)	28×28 (DS***)	28×28 (DS***)
Output size	6	6	3	2	2
Transformation types	Affine	Affine	Attention (scaling, translations)	Fixed Attention (translations)	Fixed Attention (translations)
Epochs trained	160	110	110	110	270
Learning rate	0.01	0.01	0.01	0.005	0.005
Learning rate decay	None	0.1 every 30 epochs	0.5 every 10 epochs	0.5 every 10 epochs	0.1 every 10 epochs

*Convolutional layer, **Fully-connected layer, ***Downsampling

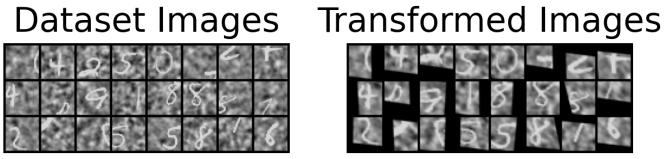

Fig. 3: The STN_28x28 Network. Examples of spatially transformed input feature maps with the network, when the input image (from the 28×28 Noisy dataset) is presented.

coordinates instead of cartesian ones (the azimuth on one axis and the log-eccentricity on the other), making the log-polar data amenable for a bi-dimensional convolutional processing (see fig. 2).

B. Networks

To natively compare the performance of the What/Where model with a Spatial Transformer Network (STN), i.e., a Spatial Transformer augmented DCNN classifier, four different STN architectures are created from a similar computational graph. All networks are composed of two main modules, namely a spatial transformer module, the localization network mainly, whose output is a set of spatial transformation coordinates, implementing the group of affine transformation over the visual field (that is translation, scaling, rotation, shear, etc.). Implemented in PyTorch, a fully differentiable grid sampler allows to apply the transformation over the pixel input, feeding the second module (see Fig. 4). This second module, said the classifier, uses the "LeNet" architecture [6] as a backbone classifier for digit recognition, similar to the one used in the original What network [24]. This module has two 5×5 convolutional layers (stride 1, no padding) interleaved with 2×2 max-pooling layers, followed by two fully connected layers that lead to a 10-way classifier.

Next, four distinct localization modules are considered here. They correspond to both an incremental complexity of the models as well as an incremental difficulty of the localization/classification task. The first one; **The STN_28x28**, serves to test the robustness of a Spatial Transformer on a small generic dataset from the original task (e.g., by classifying only a 28×28 pixels image). It also serves as a comparison with the What module of the original What/Where Network.

The other three localization modules are concerned with the processing of the full visual field. Each of them evolves from the previous one in an incremental way, presenting an important supplementary feature, that is the foveal downsampling, and the log-polar compression at the input, finally implementing all the features of the original What/Where model.

- A first vanilla STN is parametrized to detect all types of affine transformations; scaling, rotation, and translation, **the STN_128x128**.
- The second one; **The ATN (Attention-only spatial Transformer Network)**, is restricted only for attention, i.e., scaling and translation, and will introduce a downsampling mechanism of the image by passing from 128×128 pixels to 28×28 pixels in the grid sampler inside its transformer module. This combination of a visual shift followed by a downsampling contains both the principles of a gaze shift and the the selection of the foveal part of the visual field for further processing.
- Finally, the last model; **The POLO_ATN (POlar-LOgarithmic Attention-only spatial Transformer Network)**, is similar to the previous one, except that it is set to detect only translations (fixed attention), and uses as input the coefficients of the Log-Polar transformation of the original image. This latter network is tested on the different Log-Polar compression configurations, the **POLO_ATN_85%** and the **POLO_ATN_95%** for a compression rate of 85% and 95%, respectively, this high compression rate gives the possibility to use an only fully-connected network inside the localization module within the spatial transformer. Last, in order to check whether the visual processing architecture of the localization network inside the STN plays a role in performance, a convolutional counter part is tested, the **convPOLO_ATN_85%**, with two convolutional layers followed by two fully-connected layers, only using a compression rate of 85%.

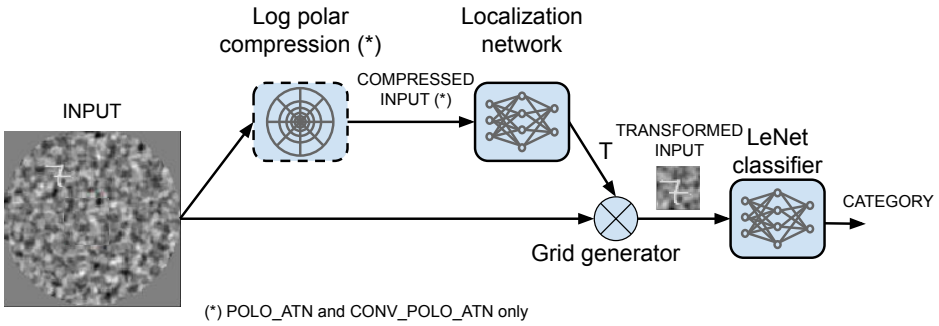

Fig. 4: Computational graph of a Foveated Spatial Transformer. The image is first compressed to its Log-Polar counterpart using a bank of filters, the compressed feature vector is then passed to the localization network that takes charge of determining the translation over the two axes. After this, the fixed attention matrix is built and used by the downsampled grid generator to hightlight the region of interest with its coordinates, i.e., the digit. Finally, the downsampled and attention-restricted feature vector is passed to the classification network.

In all of our convolutional architectures, the first convolutional layer has 20 filters and the second one has 50 filters, except the STN_128x128 which has 100 filters in its second convolutional layer; this choice was made because this network is the only architecture that operates on a full 128×128 image for classification.

A curriculum learning training scheme [35] is used to train the networks, meaning that at the beginning of training, only small eccentricities with a fixed high contrast are used, then incrementally making the task harder throughout epochs. It is worth mentioning that all networks place a 2×2 max-pooling layer subsequent to every convolutional layer and use rectified linear (ReLU) non-linearities. For more information concerning the four architectures and their training, see Table I.

III. Results

A. Foveal transformations

The central accuracy is defined as the performance of the network when the digit's eccentricity is set to 0, the general accuracy is when the digit's shift can vary up to 15 pixels. After training, the STN_28x28 was able to achieve a central accuracy of 88% and a general accuracy of 43% on this dataset. Then, to see how the transformer operates on input images, some examples of feature vectors are transformed with the Spatial Transformer module of the STN_28x28 and represented next to their original counterparts (see Fig. 3), we can see that the Spatial Transformer is going to crop relevant parts of the image and center them, this happens before feeding the feature vector to the classification network.

B. Full-field transformations

In order to investigate the attentional mechanism and the inner workings of each of the three full-field architectures, dataset images with different varying eccentricity values (a maximum of 40 pixels) and different varying contrasts were transformed using the trained spatial transformer modules (Fig. 5). First, and in the case of STN_128x128, we can observe that the Spatial Transformer is going to center the digit by creating another warped 128×128 pixels version of the original feature map (see Fig. 5a), even when the eccentricity is at its maximum and the task becomes harder, it is capable of centering the region of interest. Second, for the ATN, we can see that the transformer is capable of attending to the digit and centering it on the small 28×28 grid that will be fed later to the classification network (see Fig. 5b), in the same manner as the STN_128x128, and even when the contrast is fixed to 0.3 and the digit is barely visible, the ATN network will be able to localize the digit inside the 128×128 screen. Finally, and the conv_POLO_ATN_85 was tested on the hardest setup for this particular dataset, with targets placed at random between $0-40$ pixels away from the center, and a contrast randomly set in the $30-70\%$ range. For the majority of cases, the network was able to bound the digit inside its sampler, centering it perfectly in some cases and close calling its position for the remaining, and sometimes totally missing it out (see Fig. 5c).

C. Model benchmarking

It is worth highlighting that in the case of log-polar compression, the visual information is mostly conserved around the center of sight and at the fovea, and strongly compressed at the periphery. This makes the visual targets more difficult to detect in the latter, which leads the network to make errors, doing a greater proportion of missed displacements, and lowering the overall classification accuracy. The three architectures; STN_128x128, ATN and POLO_ATN, were then benchmarked on eccentricities ranging from 0 to 40, on each of the three different following contrasts; 0.7, 0.5 and 0.3. Classification accuracies are represented alongside the performance of the baseline What/Where model on the same dataset parameters (see Fig. 6).

When considering the baseline What/Where model, as it is reported in [24], the effect of the log-polar compression is reflected in a decreasing classification accuracy with regards to the eccentricity. The leftmost value corresponds to the

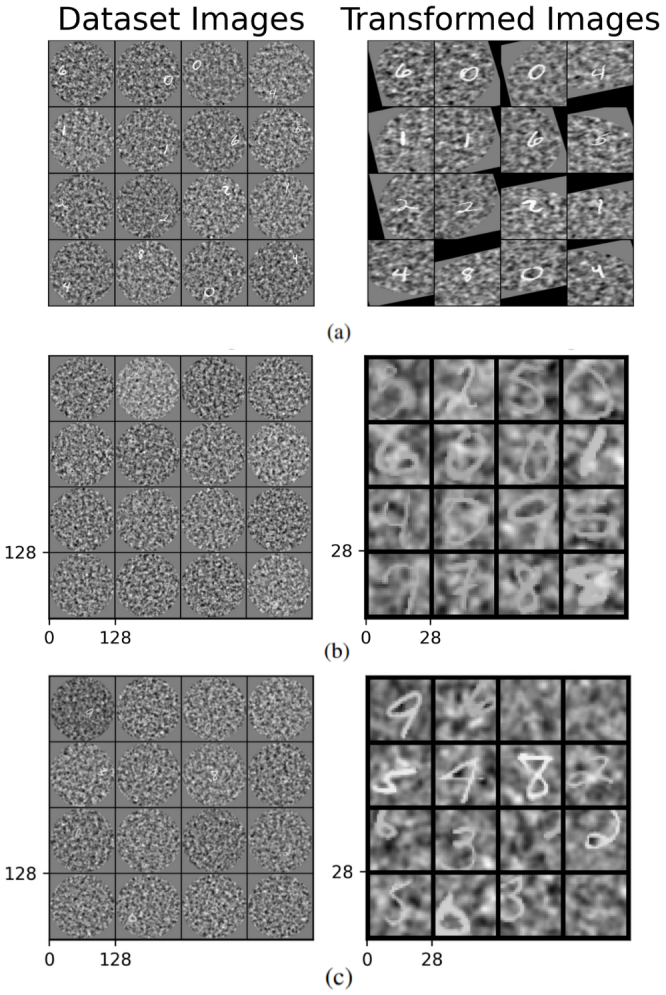

Fig. 5: Examples of some of the spatial transformations that were learned, The 128×128 dataset images before and after passing the Spatial Transformer architecture. (a) Transformations applied by the STN_128x128 network, the digit shift's is set to 40 pixels and the contrast is set to 70%. (b) Transformations applied by the ATN network, dataset configuration is at its hardest, digit's shift is set to the maximum amount allowed which is 40 pixels and the digit's contrast is set to 30%. (c) Transformations applied by the convPOLO_ATN network, digit's shift and contrast vary randomly between $0-40$ pixels and $30-70\%$, respectively.

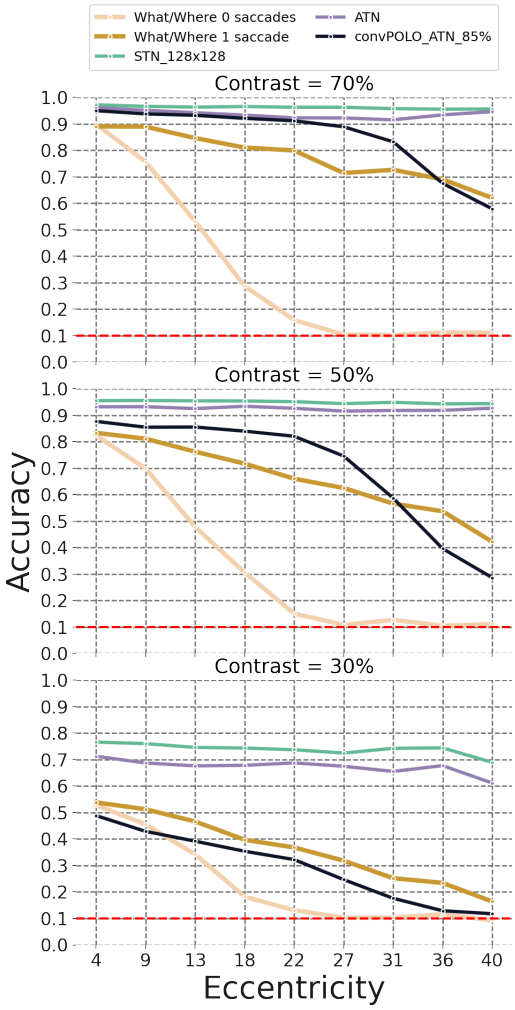

Fig. 6: Benchmark comparison between the three Spatial Transformer architectures (STN_128x128, ATN and convPOLO_ATN) and the What/Where model on the 128×128 Noisy MNIST dataset, classification accuracy as a function of the digit's eccentricity and contrast, the baseline performance is the What/Where 0 saccades which corresponds to a normal LeNet classifier that was trained and tested on the dataset without any architectural modification.

"central accuracy", that is the classification rate when the target is at the center. Then, the "0 saccade" curve reflects the mere foveal classification obtained with varying target eccentricities, without transformation. There, the values above 0.1 reflect the baseline shift invariance of the "What" classifier. Then, the "1 saccade" curve reflects the intervention of the localization network (the "Where" module), that is displacing the fovea toward the putative target, and then classifying the foveal image. The lower the decrease (with respect to the center), the higher the radial distance at which targets can be detected. The contribution of the localization network can be measured quantitatively in terms of information gain, that is the difference in classification rate before and after the eye saccade [25].

Considering first the STN_128x128 and the ATN architectures, higher overall accuracies on all eccentricities and contrasts are observed compared to the What/Where model (with 1 saccade). This is expected as the two models take advantage of the full visual information. For these two networks, small to no difference in performance is observed between contrasts 0.7 and 0.5, followed by a decrease for a contrast of 0.3. Another important feature that can be observed from these two architectures is that eccentricity does not affect the classification rate, i.e., no matter how far the digit is, the

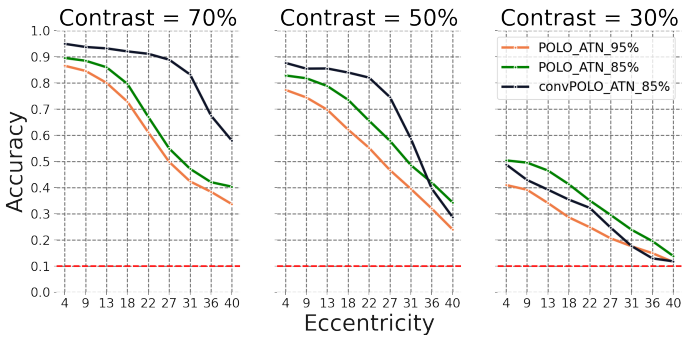

Fig. 7: Benchmark between the three POLO_ATN architectures (POLO_ATN_85%, POLO_ATN_95%, convPOLO_ATN_85%) on the 128×128 Noisy MNIST dataset, accuracy as a function of eccentricity and contrast.

network will be able to classify it, which is not the case for the remaining architectures that use Log-Polar compressed coordinates (the What/Where model and POLO_ATNs).

Jumping on to the convPOLO_ATN_85 architecture, the classification accuracy, as expected, now tends to decrease in proportion with the eccentricity. This decreasing performance with the target eccentricity here clearly outperforms the What/Where model for contrasts of 70% and 50%, showing a higher accuracy over a wider range of eccentricity, providing a remarkable stable performance up to an eccentricity of 30 pixels, covering about 75% of the total visual field with only 15% of the visual information. A more linear decrease is observed in the low contrast case (30%), with slightly lower classification rates than the original What/Where model. Interestingly, the difference is stable along the eccentricity range, showing that the accuracy deficit may be imputable to the final classifier rather than the localization module, which indicates that it may possibly be improved with increasing training time.

To measure the effect of compression and convolutional processing on the architecture, the convPOLO_ATN_85% architecture is compared with the POLO_ATN_85% and POLO_ATN_95% models in Fig. 7. Considering the effect of the log-polar compression first, a small but significant drop in classification is observed for the higher compression (95%) with regards to the lower compression (85%). Interestingly, the same drop is observed across the full eccentricity range, keeping the same qualitative decrease. The difference is more pronounced, when comparing the convolutional network with the non-convolutional ones. The convolutional network clearly outperforms the two others in the high-medium contrast case, and closely compares with the others in the low contrast case. Considering that both convPOLO_ATN_85% and POLO_ATN_85% process the same visual information, this brightly illustrates the advantage of exploiting the 2D structure of the log-polar grid to put a greater constraint (inductive bias) on information processing to improve learning.

IV. Discussion

Our step-by-step investigation of the properties of the spatial transformer networks over fully-resolved and then log-polar compressed visual inputs has provided a way to fill the gap between the original vanilla STNs and the more biologically relevant visual processing architectures, reaching and even surpassing the performance of the original What/Where model *in the absence of any spatial labelling*.

- On the one hand, STN_128x128 and ATN perform exceptionally well on this dataset and largely outperform their counterpart. However, they are more computationally costly as they process the full 128×128 image instead of the log-polar compressed version. It should be emphasized that although the ATN architecture limits the number of transformations to attention only and introduces a downsampling mechanism, the difference of performance with the STN_128x128 is considered minimal, meaning that this architecture should be privileged when thinking in terms of localizing the object in visual space without log-polar compression.
- On the other hand, the POLO_ATN architecture, which implements the log-polar compression of the input, shows a similar decrease in performance with the eccentricity than its close counterpart, the What/Where model. Still it was able to surpass the What/Where model and to gain a stable performance for lower to mid eccentricities. All this inaugurates the POLO_ATN as a viable candidate for implementing object localization in more general visual search setups.

Three key results finally emerge from this series of experiments. The first and the main outcome is that we demonstrated how to leverage the properties of spatial transformers to get rid of the spatial labelling constraint that was inherent to the original What/Where model. On contrary, our specially-modified Spatial Transformer Networks follow the classical Deep Learning paradigm, that is being totally differentiable, and learning in an end-to-end fashion how to map each input to its appropriate linear spatial transformation in an unsupervised manner and solely based on a classification criterion during training. This is at the condition, in our specific setup, to use a curriculum learning scheme that progressively increases the difficulty of the task. The tuning of this curriculum training was done by hand and requires quite a few care and expertise to get to the final result.

A second result is the capability of spatial transformers to deal with non-linear spatial deformations of their visual inputs to process attention shift over the full visual field. This distortion invariance of STNs shown in our simulations, makes it possible to consider a new class of visual processing setups that can track objects in a *sub-linear* fashion rather than analyzing all the pixels of the image. This comes with a strong reduction of the number of parameters of the localization network, that should make it possible to scale-up the method to size-realistic input images. This is a generic property of the log-polar encoding (with a number of coefficients scaling like

the log of the initial data), combined with attentional shift [25]. The notion of "Foveated" Spatial Transformers finally comes to light (see Fig. 4); wholly based on specially modified attention-only spatial transformers [13], they integrate the biological realism and the computational efficiency of a Log-Polar based artificial vision system alongside the easiness of learning of spatial transformers of different translations in objects inside images, without any annotation added to the training procedure.

Last, we demonstrate that convolutions matter! In the case of a spatially structured log-polar encoding, the convolutional layers that follow the organization of a spatially-homotopic transformation, improve the learning of the spatial transformer. This is connected to the conservation of the spatial topographic organization of the retina throughout multiple layers in visual and visuo-motor processing in the brain. This is at odds with recent trends claiming that DCNNs are not necessarily important for optimizing image classification tasks; examples are the Vision Transformer architecture [15] as well as MLP-mixer [36], that only use either a linear self-attention mechanism [37] or even only fully-connected layers, respectively.

Finally, extending the visual search task to more elaborate setups that can handle natural images is clearly required for scaling up our "Foveated Spatial Transformers" towards real world applications. For that, in addition to VGG-19 style convolutional backbones [5], non-convolutional Vision Transformer/MLP-Mixer architectures should also be considered for deeper comparison. Last, our proposed architecture is only evaluated over one translational movement toward one target object inside an image. Extending the operational range to multiple saccades and multiple objects is a rather straightforward task and should provide in the future additional insight about natural vision and the natural processing of complex visual scenes.

REFERENCES

[1] L. Perrinet, E. Daucé, and J. Gachot. https://github.com/dabane-ghassan/int-lab-book/tree/main/src/POLO, 2021.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," tech. rep., 2012.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," may 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," tech. rep., 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, dec.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[7] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of 52nd Annual Meeting of the ACL 2014*, vol. 1, pp. 655–665, apr 2014.

[8] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *EMNLP 2014: Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, aug 2014.

[9] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Journal of the American Medical Association*, vol. 318, pp. 2199–2210, dec 2017.

[10] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," aug 2018.

[11] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," *International Journal of Computer Vision*, vol. 127, pp. 456–476, nov 2014.

[12] G. Hinton, A. Krizhevsky, N. Jaitly, T. Tieleman, and Y. Tang, "Does the Brain do Inverse Graphics?,"

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," *Advances in Neural Information Processing Systems*, vol. 2015-Janua, pp. 2017–2025, 2015.

[14] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules,"

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, *et al.*, "an image is worth 16x16 words: Transformers for image recognition at scale," tech. rep.

[16] R. Vanrullen and A. Alamia, "GAttANet: Global attention agreement for convolutional neural networks,"

[17] K. Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," tech. rep., 1980.

[18] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, pp. 574–591, oct 1959.

[19] J. Herault, "Biologically inspired computer vision : fundamentals and applications," 2015.

[20] D. L. Sparks and I. S. Nelson, "Sensory and motor maps in the mammalian superior colliculus," aug 1987.

[21] H. Kirchner and S. J. Thorpe, "Ultra-rapid object detection with saccadic eye movements: Visual processing speed revisited," *Vision Research*, vol. 46, pp. 1762–1776, may 2006.

[22] E. Daucé, "Active fovea-based vision through computationally-effective model-based prediction," *Frontiers in neurorobotics*, vol. 12, p. 76, 2018.

[23] M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," 1983.

[24] E. Daucé, P. Albiges, and L. U. Perrinet, "A dual foveal-peripheral visual processing model implements efficient saccade selection," *Journal of Vision*, vol. 20, no. 8, pp. 1–20, 2020.

[25] E. Daucé and L. Perrinet, "Visual search as active inference," in *Communications in Computer and Information Science*, vol. 1326, sep 2020.

[26] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, J. O'Doherty, and G. Pezzulo, "Active inference and learning," *Neuroscience & Biobehavioral Reviews*, vol. 68, pp. 862–879, sep 2016.

[27] K. Friston, R. A. Adams, L. Perrinet, and M. Breakspear, "Perceptions as hypotheses: Saccades as experiments," *Frontiers in Psychology*, vol. 3, no. MAY, p. 151, 2012.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv*, 2019.

[29] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, *et al.*, "Array programming with NumPy," sep 2020.

[30] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[31] P. S. Leon, I. Vanzetta, G. S. Masson, and L. U. Perrinet, "Motion clouds: Model-based stimulus synthesis of natural-like random textures for the study of motion perception," *Journal of Neurophysiology*, vol. 107, pp. 3217–3226, jun 2012.

[32] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," in *Readings in computer vision*, pp. 671–679, Elsevier, 1987.

[33] P. Kortum and W. S. Geisler, "Implementation of a foveated image coding system for image bandwidth reduction," in *Human Vision and Electronic Imaging*, vol. 2657, pp. 350–360, SPIE, 1996.

[34] N. J. Butko and J. R. Movellan, "Infomax control of eye movements," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 91–107, 2010.

[35] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ACM International Conference Proceeding Series*, 2009.

[36] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, *et al.*, "MLP-Mixer: An all-MLP Architecture for Vision," 2021.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, jun 2017.