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

Ghassan Dabane[1], Laurent U Perrinet[1], Emmanuel Daucé[1,2]
[1]Institut de Neurosciences de la Timone (UMR 7289); Aix Marseille Univ, CNRS; Marseille, France
[2]Ecole Centrale de Marseille, France

Convolutional Neural Networks have been considered the go-to option for object recognition in computer vision for the last couple of years. However, their invariance to object's translations is still deemed as a weak point and remains limited to small translations only via their max-pooling layers. One bio-inspired approach considers the What/Where pathway separation in Mammals to overcome this limitation. This approach works as a nature-inspired attention mechanism, another classical approach of which is Spatial Transformers. These allow an adaptive end-to-end learning of different classes of spatial transformations throughout training. In this work, we overview Spatial Transformers as an attention-only mechanism and compare them with the What/Where model. We show that the use of attention-restricted or "Foveated" Spatial Transformer Networks, coupled alongside a curriculum learning training scheme and an efficient log-polar visual space entry, provides better performance when compared to the What/Where model, all this without the need for any extra supervision whatsoever.

***Index Terms*—visual system and development, biologically inspired feature extraction, spatial transformers, What/Where bio-inspired vision model, visual attention, curriculum learning.**

I. Introduction

Since the emergence of AlexNet; the winner of the 2012 ILSVRC image classification competition [1], computer vision has been dominated by the use of deep convolutional neural networks (DCNNs) [2] to capture the semantic content of images. Today, some classifiers are even able to surpass human level performance on some visual categorization challenges [3]. From object recognition [4], [5] and natural language processing tasks [6], [7], to lymph node metastasis detection [8] and diagnostic radiology in patient care [9], 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 DCNNs invariant to large transformations in input data [10]. To deal with this constraint, on one hand, Spatial Transformers Networks (STN) were introduced [11], 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, and for the same reason that the Neocognitron [12]; the predecessor of modern DCNNs, was inspired by the discovery of simple and complex cells in the primary visual cortex in mammals [13], the need for architectures that are inspired from biological underlying principles is growing [14]. Indeed, the human visual processing system is still considered unrivalled when it comes to speed of detection and computational efficiency. In this paper, we focus on the fact that object recognition is done at ease using high-speed eye movements [15]. One recent paradigm for a biologically-inspired computer vision application of that principle is the artificial What/Where model [16]. This model mimics the anatomical separation of the corticocortical pathways processing visual information that is found in mammals. In that model, the ventral and dorsolateral pathways are responsible for object vision and spatial localization, respectively [17]. Captured within an active inference framework [18]–[20], this model works in a sequential way. A first and key aspect of this artificial visual processing setup is the compression of the visual data through a center-surround log-polar grid representation; as is the case of the foveated vision in mammals [21]. This foveated visual input is processed through the "Where" module to determine the optimal viewpoint upon which the agent shall fixate its center of gaze. After moving the eye toward this new position, the "What" module will oversee classifying a small region around the center (the "fovea") to detect the object contained within it. Thanks to the log-polar compression placed at the entry level of this architecture, complexity (processing time) is sub-linear with regards to the number of pixels, as opposed to classical computer vision where it is still considered linear in the number of pixels. This provides a decrease of the computational load on the network while, at the same time, im-

plementing a biologically-inspired attention-driven mechanism for computer vision to help solve the visual search task. This bio-inspired "dual pathway" is consistent with more recent trends in visual processing, that is the routing of the visual data through linear (affine) transformations layers through Spatial Transformer Networks (STN). In particular, the possibility to backpropagate the gradients through the visual transformations layers may allow to overtake the less data-efficient actor-critic principles used in the original model, where each pathway was separately trained. In this work, we thus explore and benchmark the visual spatial transformer paradigm against the latter bio-inspired attentional What/Where model. We 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*. 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.

II. Materials and Methods

The visual search task is exactly similar to that described in the experimental setup for the original What/Where model [16]: A random handwritten digit will be placed inside a screen with added clutter and noise, and the agent's mission will be to classify the digit; as in determining its label. However, the digit will be 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 and 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" [22]. All networks are trained on a GTX 1660 Ti GPU, and results are visualized and organized within Jupyter Notebooks [23] using Python's scientific plotting libraries NumPy [24] and Matplotlib [25]. The source code is available at https://github.com/dabane-ghassan/int-lab-book

A. Datasets

The MNIST database [5] 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. For the purpose of this application, three variants are prepared, the 28×28 Noisy dataset, where images keep their original size and are mixed with synthetic textures of random noise in the background [16], [26] (see Fig. 1a for some examples), the second one; the 128×128 Noisy dataset, where MNIST images are embedded randomly inside a larger 128×128 image that contains a circular mask (of radius 64) of random textures (Fig. 1b), and finally, the compressed 128×128 noisy dataset, in which we use two banks of filters disposed on a log-polar grid to linearly transform the original feature map of size $128 * 128 = 16384$ into a compressed form.

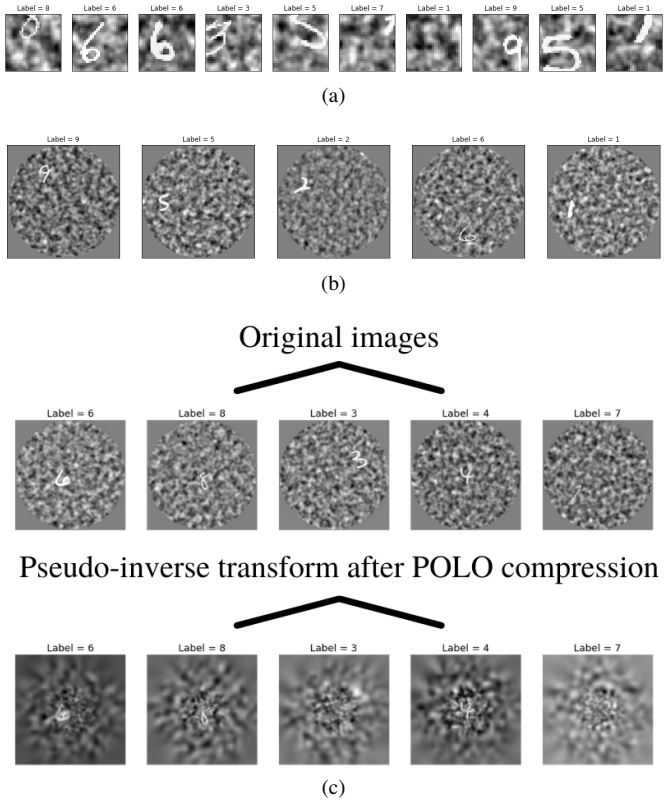


Fig. 1. The datasets that were used for the visual search task. (a) The 28×28 pixel Noisy Shifted MNIST dataset. (b) the 128×128 pixels Noisy Shifted MNIST dataset. (c) Visualizing the Polar-Logarithmic (POLO) version of the 128×128 Noisy dataset by using the pseudo-inverse transform, images are compressed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

The first bank has 768 predisposed filters and the second one has 2560, this provides a compression rate of approximately 95% $(1-768/16384)$ and 85% $(1-2560/16384)$, respectively. It is also worth mentioning that the original What/Where model has a compression rate of about 83% [16], 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). The digit's eccentricity can vary between 0 and 40 pixels in all the datasets, forcing the digit to fit entirely inside the circular mask in the case of the 128×128 image and sometimes making the task impossible in the case of a 28×28 image. The digit's contrast varies randomly in a uniform fashion between 70% and 30%.

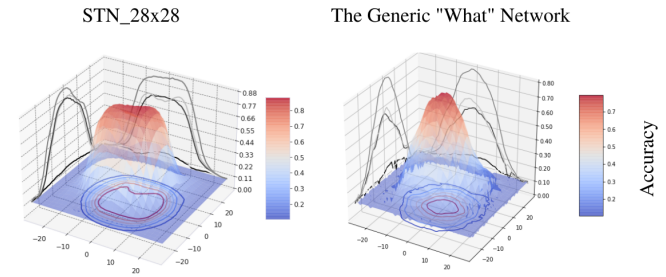
B. Networks

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. The first one; **The STN_28x28**, serves only as a comparison with the What module of the What/Where Network to test the robustness of a Spatial Transformer on a small generic dataset from the original task

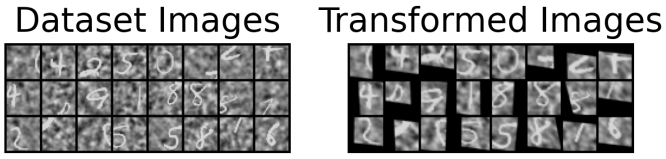
(e.g., by classifying only a 28×28 pixels image), the other three compare to the full What/Where architecture and each one of them presents an important feature, one vanilla STN is parametrized to detect all types of affine transformations; scaling, rotation, and translation, **the STN_128x128**. The 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 bothe the principles of a gaze shift and the selection of the selection of the foveal part of the visual field for further processing. Finally, the last network; **The POLO_ATN (POlar-LOgarithmic Attention-only spatial Transformer Network)**, is similar to the previous one, axcept 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 will be 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 only a fully-connected network inside the localization module within the spatial transformer. In order to check whether the visual processing architecture plays a role in performance, one last variant is created with a normal DCNN as a localization network; **convPOLO_ATN_85**. Furthermore, and in order to provide a robust comparison, all networks use the "LeNet" architecture [5] as a backbone classifier for digit recognition, the exact same one used for the What network [16]. This architecture 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. In all of our four spatial transformer architectures, the first convolutional layer has 20 filters and the second one has 50 filters, except the STN_128x128 where it 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 [27] 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.

	STN_28x28	STN_128x128	ATN	POLO ATN	convPOLO ATN
Dataset	28x28 Noisy	128x128 Noisy	128x128 Noisy	128x128 Noisy + Compressed (95%)	128x128 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	28x28	128x128	28x28 (DS***)	28x28 (DS)	28x28 (DS)
Output	6	6	3	2	2
Types of transformations	Affine	Affine	Attention (scale, translations)	Fixed attention (translations)	Fixed attention (translations)
Epochs trained	160	110	110	110	270
Base learning rate	0.01	0.01	0.01	0.005	0.005
Learning rate decay	X	1/10 every 30 epochs	1/2 every 10 epochs	1/2 every 10 epochs	1/10 every 10 epochs

TABLE I
DIFFERENT NETWORK ARCHITECTURES AND PARAMETERS



(a)



(b)

Fig. 2. The STN_28x28 Network. (a) Accuracy map comparison between the STN_28x28 (left) and the original generic What network (right), classification accuracy is represented on the vertical axis and calculated for a grid of shift values of size 55×55. (b) Examples of spatially transformed input feature maps with the network, when the input image (from the 28×28 Noisy dataset) is presented, the spatial transformer module will warp it, and then it will feed it to the classification part of the network.

III. RESULTS

A. Spatial Transformer Network Vs. The Generic "What" pathway

After training, the STN_28x28 was able to achieve a central accuracy of 88% and a general accuracy of 43% on this dataset, compared to 84% and 34% from the Generic "What" Pathway [16], 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. Overall, this suggests that we have an improvement in the accuracy of the classifier. Next, digit's coordinates are fixed according to a grid of 55×55 pixels (for a 28×28

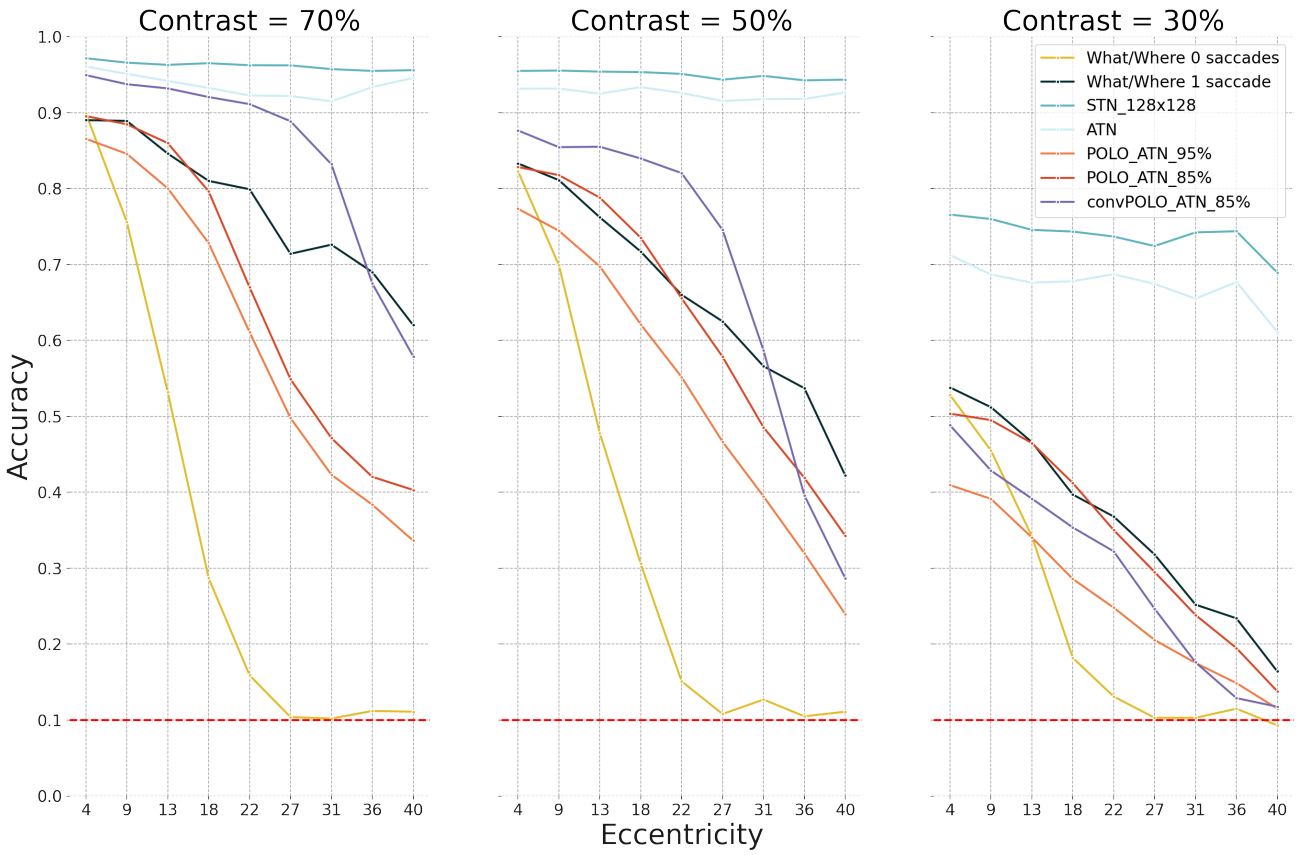


Fig. 3. Benchmark comparison between the three Spatial Transformer architectures (STN_128x128, ATN and POLO_ATN) and the What/Where model on the 128×128 Noisy MNIST dataset, classification accuracy as a function of the digit's eccentricity and contrast, the baseline performance is the What/Where 0 saccades which corresponds to a normal LeNet classifier that was trained and tested on the dataset without any architectural modification, three variants of the POLO_ATN were tested with different compression levels (95% or 85%) and with different localization network architectures (fully-connected only or convolutional).

pixels image) and the accuracy value that corresponds to every possible position on this grid is measured, defining an accuracy map, like described for the What pathway [16]. This accuracy map is calculated for the STN_28x28 and is represented on Fig. 2a, next to the What network's accuracy map. Upon first glance, we can distinguish that the accuracy map of the new STN_28x28 has a different shape, with higher accuracies ($>$0.8) occupying a bigger region on the grid when compared to the What network, then accuracy decreases sharply for a few pixels getting to the baseline (0.1), in contrast with the What network where accuracy decreases more slowly. Lastly, and for both networks, the baseline is reached starting from an eccentricity of 15 pixels, this is considered normal knowing that the digit will be entirely fitted outside the image in this case, and thus out of reach for classification.

Finally, and 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. 2b), 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. Spatial Transformer Networks Vs. The What/Where model

The three architectures; STN_128x128, ATN and POLO_ATN, were 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 What/Where model on the same dataset parameters (see Fig. 3). Generally speaking, we can observe that the STN_128x128 and the ATN architectures have higher overall accuracies on all eccentricities and contrasts compared to the What/Where model (with 1 saccade). Next, and 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 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 POLO_ATN architecture, It is worth

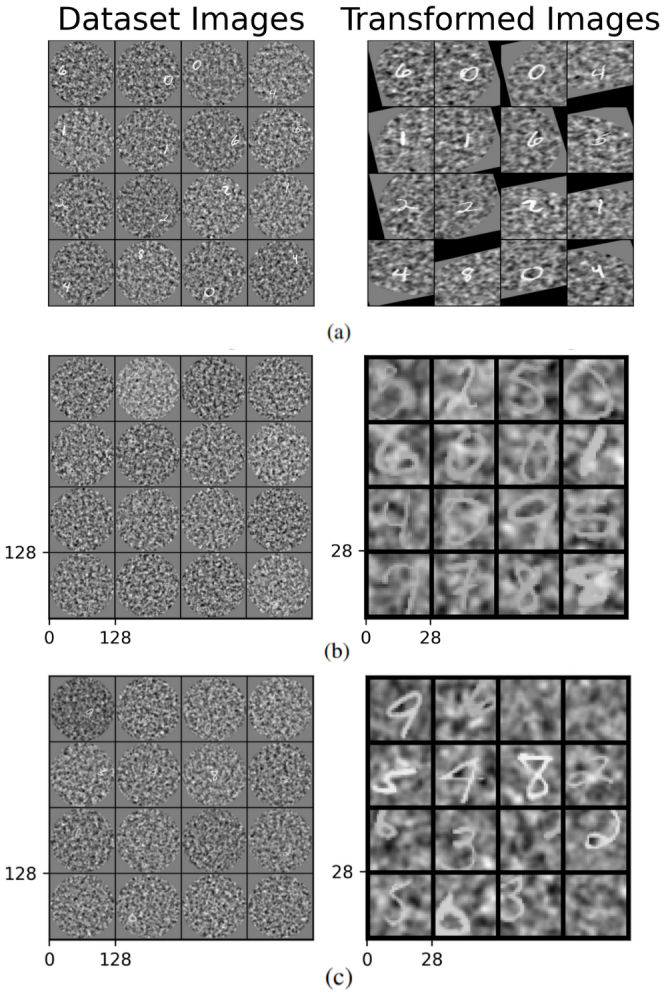


Fig. 4. Examples of some of the spatial transformations that were learned, The 128×128 dataset images before and after passing the Spatial Transformer architecture. (a) Transformations applied by the STN_128x128 network, the digit shift's is set to 40 pixels and the contrast is set to 70%. (b) Transformations 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.

noting that all the 3 networks manifest in particular the same tendency as the What/Where model while operating on larger and larger eccentricities; the classification accuracy tends to decrease the more the digit is far away from the center. Furthermore, we can see observe that POLO_ATNs that use a fully-connected only localization network, e.g., POLO_ATN_95 and POLO_ATN_85, match the What/Where Network's performance when the digit is close to the center, but underperforming it with higher eccentricities, all contrasts alike. Finally, the convPOLO_ATN_85 that uses a DCNN for its localization module, outperforms the What/Where model for contrasts 0.7 and 0.5, providing a remarkable stable performance up to an eccentricity of 22 pixels despite the Log-Polar compression rate. In order to investigate the attentional mechanism and the inner workings of each of the three proposed architectures, dataset images with different varying eccentricity values (a maximum of 40 pixels) and different varying contrasts were transformed using the trained spatial transformer module. 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. 4a), 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. 4b), 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 to test the POLO_ATN architecture, the conv_POLO_ATN_85 was tested on the hardest setup for this particular dataset, a totally random image, taken between $0 - 40$ pixels and $30 - 70\%$ for the eccentricity and the contrast, respectively. For the majority of cases that were taken, 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. 4c).

IV. Discussion

When comparing the vanilla "What" Network to the STN_28x28; both using a LeNet architecture for classification, we have demonstrated the effectiveness of using a Spatial Transformer module on the classification rate for this dataset, a significant improvement is obtained in the accuracy map giving the classifier robust spatial invariance to affine transformations in feature maps, which explains higher accuracies over translated digits. Although placing this module at the beginning of the "What" network for future applications should yield its benefits in performance, it should be noted that it accumulates a certain computational complexity on the architecture overall (by adding more layers). However, this problem can be solved by sharing parameters between the classification network and the module itself [11]. Next, and although STN_128x128 and ATN perform exceptionally well on this dataset and outperform their counterpart; the What/Where model, 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 with foveated vision like in mammals. Regarding the POLO_ATN architecture, the loss of information in the peripheral zone for log-polar coordinates explains the decrease in performance with the eccentricity, this is similar to the What/Where model and to all architectures that use Log-Polar entry coordinates. Furthermore, a small difference in

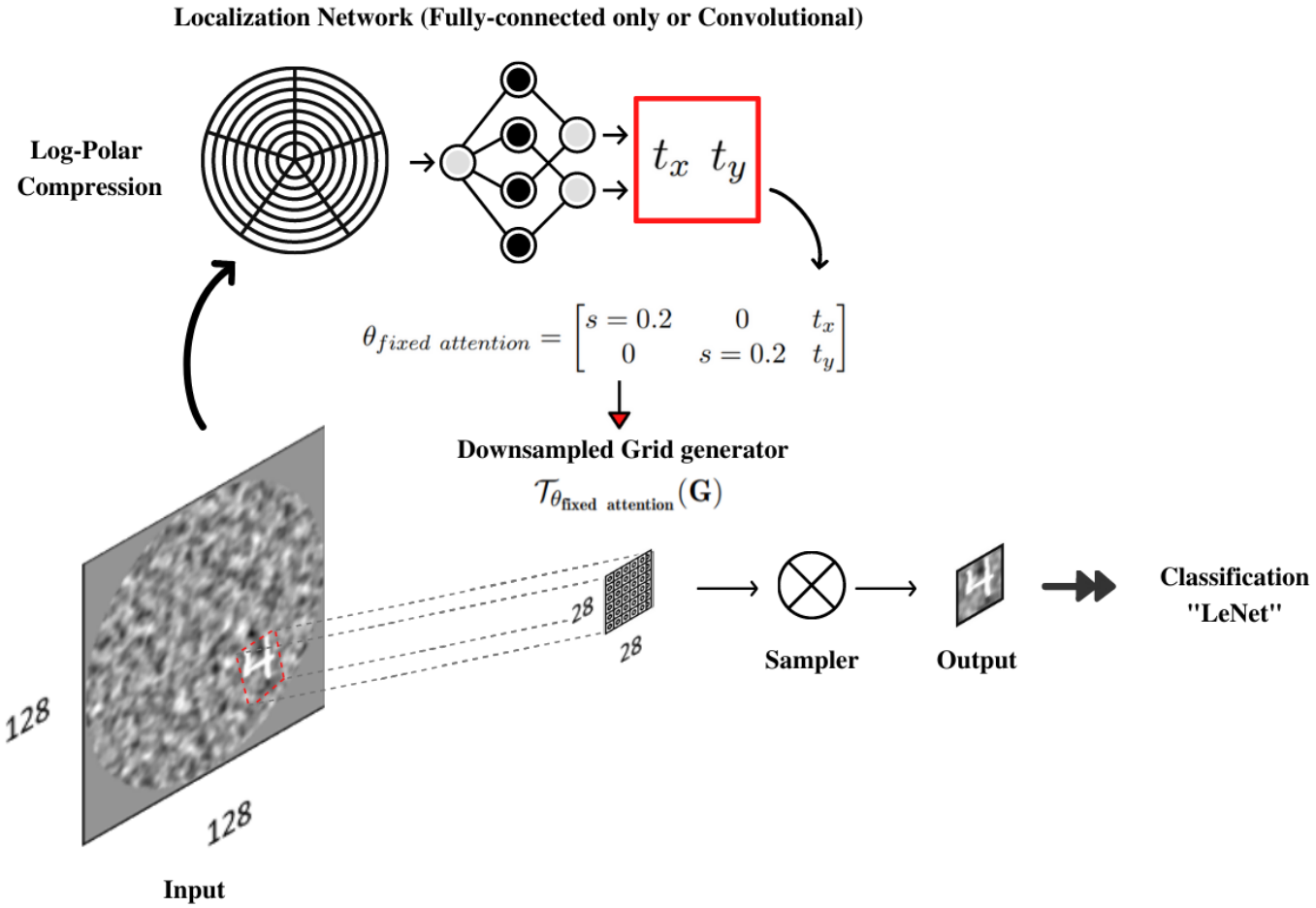


Fig. 5. The computational graph of a Foveated Spatial Transformer, the image will first be compressed to its Log-Polar counterpart using a bank of filters, the compressed feature vector will then be passed to the localization network that will take charge of determining the translation over the two axes, after this, the fixed attention matrix will be built and will be 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 will be passed to the classification network.

performance was observed when the Log-Polar compression rate was lowered from 95% to 85%, but the major improvement in accuracy came from changing the architecture of the localization network. When using a DCNN classifier inside the Spatial Transformer, the POLO_ATN architecture was able to surpass the What/Where model and to gain a stable performance for lower to mid eccentricities, even when the loss of information in the Log-Polar visual space is considered, all this inaugurates the POLO_ATN as a viable candidate for a "Where" network that localizes objects in this particular visual search setup. Finally, the difference between the two paradigms should be considered, on one hand, the What/Where model uses an actor/critic framework to determine a focal accuracy-seeking policy while training the agent, and works on the dataset in a sequential manner, i.e., scanning the visual environment, followed by the best action (a saccade) that maximizes its information gain. On the other hand, Spatial Transformer Network follows the classical Deep Learning paradigm, totally differentiable, they learn end-to-end 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 does indeed make learning the task at hand easier but adds more hyper-parameters to be controlled to the learning process.

Taking into account the architecture of POLO_ATN, the notion of "Foveated" Spatial Transformers comes to light (see Fig. 5); wholly based on specially modified attention-only spatial transformers [11], they integrate the biological realism and the computational efficiency of a Log-Polar based artificial vision system like the recent What/Where Model [16], [18], alongside the easiness of learning of spatial transformers of different translations in objects inside images, all this happens during classification without any annotation added to the training procedure.

V. Perspectives

Very recent advancements in the field have shown that DCNNs are not necessarily important for optimizing image classification tasks; the new Vision Transformer architecture [28] as well as MLP-mixer [29], use the natural language processing's self-attention mechanism [30] or yet an only

multi-layer perceptron architecture [31], respectively. State-of-the-art results are achieved without using any convolutional layers whatsoever. For further exploration of a universal robust visual attention mechanism, we find that it is worth exploring the potential of a Vision Transformer/MLP-Mixer architecture alongside Spatial Transformers for further future applications. Furthermore, the proposed architecture is only capable of doing one translational movement per object inside an image; one biological saccade, we argue that extending its capability to multiple saccades may yield an improvement in performance and a more biologically-realistic system. Finally, extending the visual search task to more elaborate setups that can handle natural images like VGG-19 [4] remains a necessity for real world applications, and can measure the effectiveness of using "Foveated Spatial Transformers" on large-scale images.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Tech. Rep., 2012. [Online]. Available: https://doi.org/10.1145/3065386

[2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," pp. 436–444, may 2015. [Online]. Available: https://www.nature.com/articles/nature14539

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

[4] 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*, vol. 2016-December. IEEE Computer Society, dec 2016, pp. 770–778. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, vol. 1. Association for Computational Linguistics (ACL), apr 2014, pp. 655–665. [Online]. Available: www.nal.co

[7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751, aug 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[8] B. E. Bejnordi, M. Veta, P. J. Van Diest, [...], and R. Venâncio, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *JAMA - Journal of the American Medical Association*, vol. 318, no. 22, pp. 2199–2210, dec 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29234806/https://pubmed.ncbi.nlm.nih.gov/29234806/?dopt=Abstract

[9] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," pp. 611–629, aug 2018. [Online]. Available: https://doi.org/10.1007/s13244-018-0639-9

[10] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 456–476, nov 2014. [Online]. Available: http://arxiv.org/abs/1411.5908

[11] 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. [Online]. Available: http://arxiv.org/abs/1506.02025

[12] K. Fukushima, "Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," Tech. Rep., 1980. [Online]. Available: https://doi.org/10.1007/BF00344251

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, oct 1959. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/

[14] G. Cristóbal, L. Perrinet and K. Keil, Matthais S, "Biologically inspired computer vision: fundamentals and applications" *John Wiley & Sons* 2015

[15] H. Kirchner and S. J. Thorpe, "Ultra-rapid object detection with saccadic eye movements: Visual processing speed revisited," *Vision Research*, vol. 46, no. 11, pp. 1762–1776, may 2006.

[16] 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. [Online]. Available: https://doi.org/10.1167/jov.20.8.22.

[17] M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," pp. 414–417, 1983.

[18] E. Daucé and L. Perrinet, "Visual search as active inference," in *Communications in Computer and Information Science*, vol. 1326. Springer Science and Business Media Deutschland GmbH, sep 2020, pp. 165–178. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-64919-7_17

[19] K. Friston, R. Adams, L. Perrinet, and M. Breakspear, "Perceptions as hypotheses: saccades as experiments," *Front Psychology* (3), pp. 151, 2012.

[20] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, J. O'Doherty, and G. Pezzulo, "Active inference and learning," pp. 862–879, sep 2016.

[21] D. L. Sparks and I. S. Nelson, "Sensory and motor maps in the mammalian superior colliculus," pp. 312–317, aug 1987.

[22] A. Paszke, S. Gross, F. Massa, [...], and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv*, 2019. [Online]. Available: http://arxiv.org/abs/1912.01703

[23] T. Kluyver, B. Ragan-Kelley, F. Pérez, [...], and C. Willing, "Jupyter Notebooks—a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*. IOS Press BV, 2016, pp. 87–90.

[24] C. R. Harris, K. J. Millman, S. J. van der Walt, [...], and T. E. Oliphant, "Array programming with NumPy," pp. 357–362, sep 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[25] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4160265/

[26] 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, no. 11, pp. 3217–3226, jun 2012. [Online]. Available: https://doi.org/10.1152/jn.00737.2011

[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ACM International Conference Proceeding Series*, vol. 382, 2009.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, [...], and N. Houlsby, "An image is worth 16X16 words: Transformers for Image Recognition at Scale," Tech. Rep., 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[29] I. Tolstikhin, N. Houlsby, A. Kolesnikov, [...], and A. Dosovitskiy, "MLP-Mixer: An all-MLP Architecture for Vision," may 2021. [Online]. Available: http://arxiv.org/abs/2105.01601

[30] A. Vaswani, N. Shazeer, N. Parmar, [...], and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 2017-December. Neural information processing systems foundation, jun 2017, pp. 5999–6009. [Online]. Available: https://arxiv.org/abs/1706.03762v5

[31] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, nov 1958.