2022

Towards an Active Foveated Approach to Computer Vision

Dario Dematties
University of Buenos Aires

Silvio Rizzi
Argonne National Laboratory

George K. Thiruvathukal
Loyola University Chicago, gkt@cs.luc.edu

Alejandro Javier Wainselboim
University of Buenos Aires

ISSN 2007-9737

Towards an Active Foveated Approach to Computer Vision

Dario Dematties[1], Silvio Rizzi[2], George K. Thiruvathukal[3], Alejandro Wainselboim[4]

[1] Northwestern University,
Northwestern Argonne Institute of Science and Engineering,
United States

[2] Argonne Leadership Computer Facility,
Argonne National Laboratory,
United States

[3] Loyola University Chicago,
Computer Science Department,
United States

[4] Instituto de Ciencias Humanas, Sociales y Ambientales,
CONICET Mendoza Technological Scientific,
Argentina

dario.dematties@northwestern.edu, srizzi@alcf.anl.gov

Abstract. In this paper, a series of experimental methods are presented explaining a new approach towards active foveated Computer Vision (CV). This is a collaborative effort between researchers at CONICET Mendoza Technological Scientific Center from Argentina, Argonne National Laboratory (ANL), and Loyola University Chicago from the US. The aim is to advance new CV approaches more in line with those found in biological agents in order to bring novel solutions to the main problems faced by current CV applications. Basically this work enhance Self-supervised (SS) learning, incorporating foveated vision plus saccadic behavior in order to improve training and computational efficiency without reducing performance significantly. This paper includes a compendium of methods' explanations, and since this is a work that is currently in progress, only preliminary results are provided. We also make our code fully available.[1]

Keywords. Foveated computer vision, saccadic behavior, reinforcement learning, self-supervised learning, General-Purpose Graphics Processing Units (GPGPUs).

1 Introduction

1.1 About the Collaboration

We begin by highlighting some aspects of international collaboration. This work is an extended version of a talk presented in the Americas HPC Collaboration Workshop, part of the CARLA 2021 Latin America High Performance Computing Conference.

Our scope aligns particularly well with the aims of the workshop, especially "partnerships formed between researchers and entities across the Americas, from Patagonia to Alaska" [2].

[1] https://github.com/dariodematties/Multimodal-Active-AI

[2] See http://carla2021.org/callforworkshops

We are a geographically-distributed team of investigators hailing from research and educational institutions in Argentina and the United States. We collaborate by leveraging leadership supercomputing resources in our research, and state-of-the-art tools for remote collaboration, which are discussed below.

Throughout years of successful collaboration, we have advised and graduated a doctoral student (Co-author, Dematties) and published in prestigious journals [7, 6, 8]. As part of his graduate education, Dr. Dematties attended the Argonne Training Program for Extreme-Scale Computing (ATPESC), where he acquired invaluable experience with common tools used in High Performance Computing. Readers interested in knowing more about the program are invited to visit https://extremecomputingtraining.anl.gov/

This experience allowed Dr. Dematties to port his software infrastructure to supercomputers, leveraging hybrid OpenMP+MPI parallelism. A Director's Discretionary allocation was granted at the Argonne Leadership Computing Facility, providing the foundation to perform large-scale computational experiments.

This collaboration makes extensive use of tools such as Zoom videoconferencing, GitHub for collaborative development, and Zenodo for publishing datasets and results.

Our work continues well past Dr. Dematties earning his Ph.D. We would especially like to mention our participation in the CyberColombia 2020 conference, where we presented a tutorial at the HPC Summer School, which covered the science behind bio-inspired models, working with supercomputers, and software engineering. See https://figshare.com/articles/presentation/Towards_High-End_Scalability_on_Bio-Inspired_Computational_Models/12762260 for the tutorial materials.

1.2 About the Research

The difficulty linked to CV comes from its hardware limitations as well as its data set shortages for training. In some cases, CV applications could depend on near real-time video processing, demanding Artificial Intelligence (AI) solutions on edge computing devices [3] which appear as the only way to overcome the latency limitations of centralized computing.

Fitting CV models on edge devices is not an easy task, given the complexity of such models. In other applications–such as in 3D medical imaging–the computational demands could be prohibitively expensive and the data sets collection could require extremely skillful staff resulting in prevalent scarcity.

Facing such challenges requires new ideas in this area. For instance, the exorbitant demands on labeled data sets could be alleviated using new SS strategies while the excessive computational demands imposed by these algorithms could be reduced utilizing inspiration from visual systems found in biological agents.

We humans as well as other higher mammals do not sense visual information as we perceive images. The retina, a specific organ located in the posterior hemisphere of the eye ball has the function of transforming rays of light entering the eye into electric signals which are later processed by the brain [25].

Yet, the perception of an image is not only a matter of the information coming from outside. We also affect our visual field perception with the architecture of our visual system and our behavior.

From an architectural point of view, our retina samples the visual field with a very high resolution in a tiny portion called fovea and with very low resolution in the periphery of such a structure (Fig. 1). From a behavioral point of view, our saccadic behavior determines where, when and how long we fixates. This significantly affects the way in which we sense and perceive the world around us.

Evidently, foveated vision reduces computational (metabolic) cost, since it is not necessary for the brain to process all the scene at high resolution. Yet, this strategy brings an undeniable cost in a lost of information. What remains is only an appropriate saccadic behavior in order to make information

[3] Edge computing is a distributed computing paradigm that brings computation and data storage closer to the sources of data. This is expected to improve response times and save bandwidth. Source: https://en.wikipedia.org/wiki/Edge_computing

processing more efficient for reproduction and survival in certain niche.

In this manuscript we report our current endeavor towards solving some of the major challenges faced by CV by means of active foveated strategies. Basically, the complete system depicted in Fig. 2 aims to palliate data sets scarcity and the prohibitive computational demands found in CV models.

We first developed a foveated system which pre-processes a batch of images. This foveated system is based on the physiology of the visual system and not on psychological aspects of vision as is the case in other works [1, 11]. We addressed foveated vision utilizing its properties as a natural augmentation approach for self-supervised learning.

Fig. 2 A shows how we advanced a strategy utilized in SimCLR [4], where a new approach to contrastive self-supervised learning algorithm is proposed.

A network is taught to discriminate images disregarding several augmentations–*i.e* crop-resize, Gaussian blur, Gaussian Noise, Color distortions, flipping, rotations, cutouts, etc. In our work we propose that such augmentations could be obtained in a more biologically inspired strategy by means of our foveated system.

A similar rationale is conducted in [11]. Basically we implemented a SimCLR like algorithm in which we teach a Residual Neural Network (ResNet) architecture to distinguish foveated fixations that come from the same image from those coming from different images.

We also incorporated additional augmentations such as color distortion, crop and resize, Gaussian noise and flipping. We tested the learned representations by means of a linear classifier–as is the standard protocol used in SimCLR.

Fig. 2 B shows how we also incorporated a transformer architecture utilizing our pre-trained ResNet network as a backbone. To that end we adapted an architecture developed by Facebook AI Research Group called DEtection TRansformer (DETR) [3]. In DETR a new method is developed that conceives object detection as a direct–end-to-end–prediction problem.

Transformers are usually employed in Seq2Seq modelling approaches especially in language models. In DETR, such an architecture is used encoding an image pixel by pixel–instead of word by word as in Natural Language Processing (NLP). In our case we adapted the original architecture eliminating several losses concerning detection, keeping only losses concerned with classification.

We also adapted the positional encoding mechanisms of the network and instead of encoding the position of every component from the ResNet backbone output we encoded the position of each fixation from our pre-trained ResNet backbone.

Self-attention has a quadratic complexity and in a patch by patch scenario–as is the case in image Transformers–this situation represents a great obstacle in the implementation of this kind of architectures when trying to process high resolution images. We address this by changing the strategy of giving each patch a position.

We instead give each fixation a position in the network. The number of fixations will be considerable smaller than the number of patches in an image. This is a huge advantage in computational load terms, especially when the use transformers brings to the scene a complexity of n^2 where n is the length of the sequence.

Finally, as shown in Fig. 2 C, we incorporated a RL mechanism in our model with the aim of learning an effective saccadic behavior. Basically we trained a DQN, which treated the dynamic of the Transformer classifier as the environment.

The observation of the state of the environment was the output from our foveated system, the actions taken by our network were the coordinates of the next fixation which gave rise to the next state from our foveator. Finally the classification performance from DETR was taken as the reward in this RL scenario.

ISSN 2007-9737

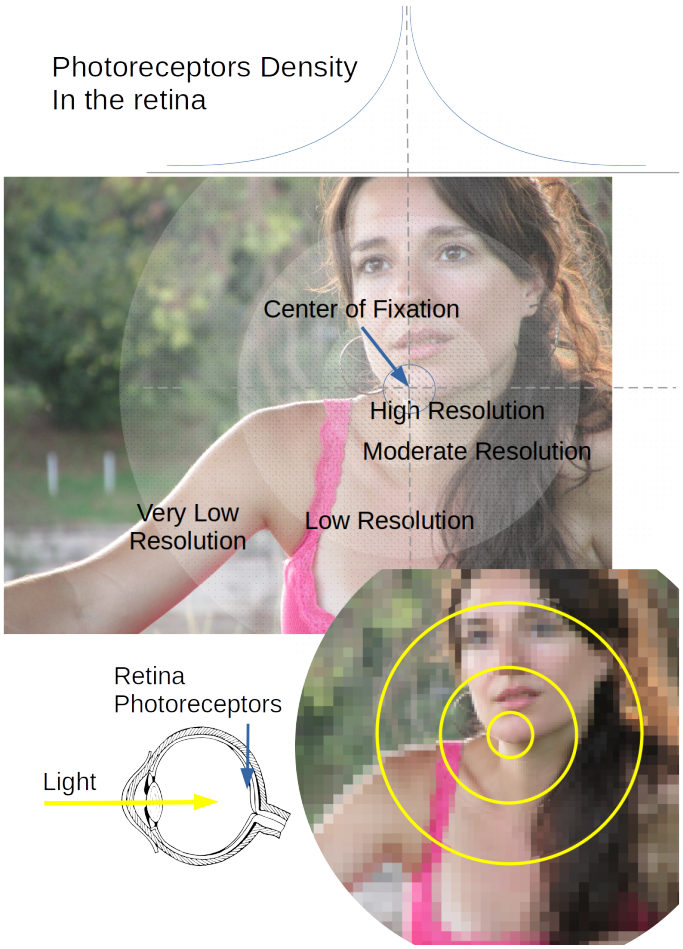

Fig. 1. Foveation in biological agents is a phenomenon in which the density of photoreceptors located on the eye's retina varies in a way that there is acutely more density near the fovea, while such a density decreases drastically in the fovea's vicinity. The fovea, is a small fraction of the retina which corresponds to the center of fixation in the sight. The consequence is that the perceptual detail regarded by the agent varies across the image according to the current fixation point, which confers the highest resolution region of the image to the center of the eye's retina, (*i.e.* the fovea)

2 Related Work

In a recent work [5], without using foveation, but in lines with saving computational effort and retaining fine details in CV tasks, a method based on a differentiable Top-K operator to select the most relevant parts of high resolution images was introduced.

In regards to foveation, in [13], a foveated vision system was introduced for face reconstruction algorithms.

In [9] a foveated model to provide *clutter* measures was introduced[4]. In regards to object detection, in [1] a foveated object detector was introduced.

Later, in [10] NeuroFovea was proposed as a model to generate visual metamerism in images [5]. In [18], it was investigated quantitatively how

[4]Clutter perception is the typically negative visual perception effect that emerges from the disordered organization of an excessive number of objects in a visual scene

[5]Metamers are stimuli that are physically distinct but that are perceived to be the same by a human observer

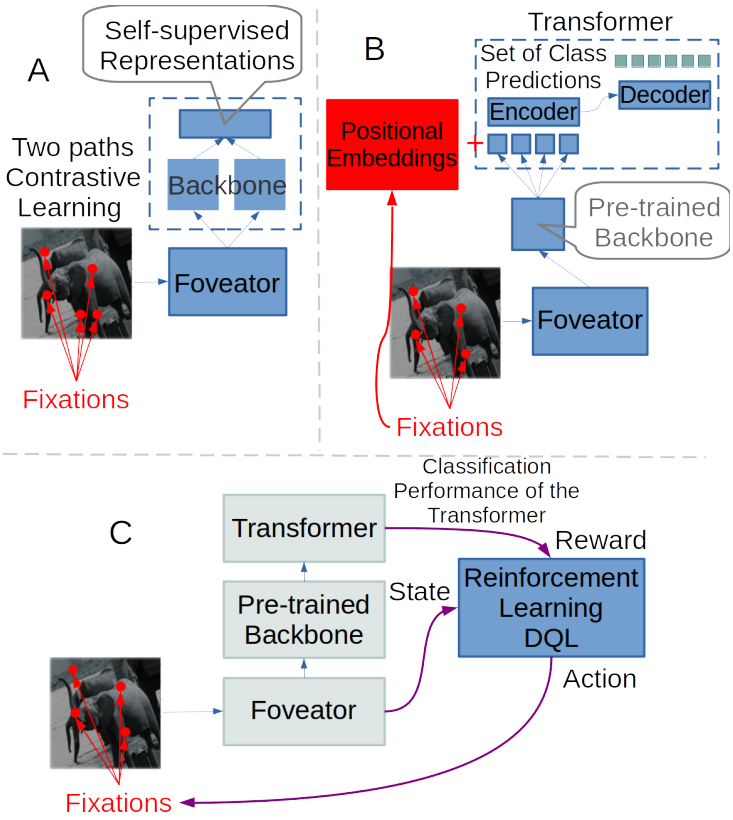

Fig. 2. Global strategic scheme to solve CV problems such as the labeled data sets scarcity and the high computational complexity demanded by the models implementation. (A) Self-supervised contrastive learning approach using foveated vision as an additional biologically-inspired augmentation strategy. With this strategy we aim to mitigate labeled data sets scarcity. (B) A transformer architecture processing a sequence of outputs from a pre-trained backbone which process foveated fixations. Positional embeddings are determined by individual fixations and not by image patches which saves great computing power demands from the transformer architecture perspective. (C) A Reinforcement Learning (RL) architecture–a Deep Q Network (DQN)–is added to the architecture to learn the saccadic behavior which is supposed to generate more effective fixation coordinates in order to increment the classification performance from the transformer

detection, recognition and processing speed in a Convolutional Neural Network (CNN) were affected by reducing image size using a foveated transformation.

In [20] images were compressed in videos applying foveation, gradually reducing the resolution in the periphery.

Afterwards images were reconstructed utilizing generative adversarial approaches. In [11] it was found that a CNN trained on foveated inputs with texture-like encoding on the peripheral information has similar scene classification performance to a matched resource CNN without foveated inputs.

Finally in [19] a foveated Transformer model was proposed.

None of the previous research analysed foveation utilizing computational hypotheses based on a developmental approach.

As is the case for this study, foveation has been applied following a developmental appeal from a SS learning strategy, passing through a Seq2Seq scheme–with random fixations–to finally end up employing a RL policy, learning the saccadic behavior of the agent.

ISSN 2007-9737

3 Computational Hypotheses

3.1 Image Foveator

In Fig. 3 we have sketched the foveation process. First of all, we resize all the images in the batch to the same specific size (640 x 640). Then we apply a series of successive crop and resize operations. All crop operations are resized to a resolution of 30 x 30 pixels. In the first row in Fig. 3, there is not crop operation, we only resize the complete 640 x 640 span to a 30 x 30 pixels resolution. Successive crop operations are illustrated with corresponding yellow squares. Such operations reduce the complete 640 x 640 span to 400 x 400, 240 x 240, 100 x 100 and finally to 30 x 30 pixels.

We use NVIDIA DALI library for the process of foveation[6]. This library is utilized for data loading and pre-processing accelerating Deep Learning (DL) workflows.

Even though the foveation operation is–in itself–an augmentation operation, we also apply additional augmentations to the batch. The pipelines also apply operations of random resize plus crop before foveation. The random area in the cropping operation ranges from 10% to 100% of the span. The location of the cropping operation is also random.

The final size of the image after the random crop plus resize is 640 x 640 pixels. Flipping, Gaussian noise, color distortion and grid mask operations are also applied randomly. Both, flipping as well as color augmentations are applied with a 50% chance. When applied, color augmentation has several components such as brightness, contrast, hue and saturation which are also generated randomly. Grid mask is applied with a 10% chance. When applied, this has two components which are generated randomly too (*i.e.* ratio, tile). The ratio indicates the quotient between occluded and free space in the image, whilst the tile sets the size of the grid inside the image.

[6]https://docs.nvidia.com/deeplearning/dali/user-guide/docs/

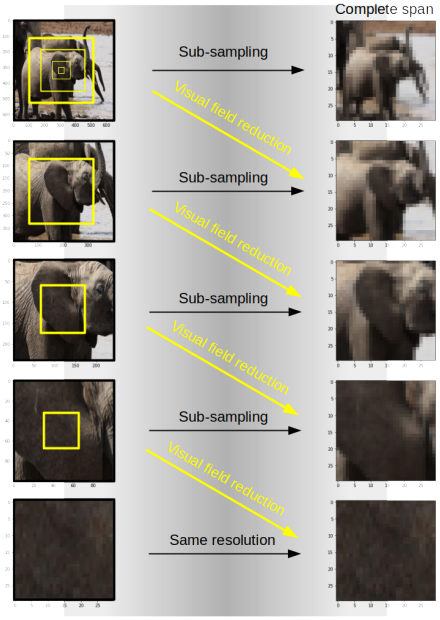

Fig. 3. Foveator. To simulate the foveation process found in some biological agents such as some mammals, we generate 5 spans from the same image around a fixation point. All the spans have the same number of pixels (30 x 30) but some of them sub-sample the complete original image while others sub-sample smaller regions of it. The smaller the region spanned the better the resolution captured by the (30 x 30) span. On the left hand side column we show the spans extracted from the original portions of the image. On the right hand side column we show the 30 x 30 spans returned by the foveator. From top to bottom rows we have bigger spans with less resolution to smaller spans with higher resolution. Yellow squares show the following span proportion respecting the previous span

3.2 Self-Supervised Approach

SS methods do not rely on human created labels but on intrinsic characteristics immersed in the statistical structure of data sets. Yet, SS learning does not only confers advantages from saving large and expensive labeled data sets. It delivers much richer representations which are not constrained to loss functions supported only by human provided labels, but by diverse features hidden in the statistical structure of the data sets. Human provided labels are instead subjective and limited. The pre-training phases in SS learning

make the networks to acquire relevant information through loss functions based on pretext tasks.

In vision, pretext tasks are very diverse. In this work we will concentrate our attention in Contrastive Learning (CL), specifically, in the research conducted by T. Chen et al. [4]. In CL the pretext task consists on teaching a model to classify different augmented versions of an image as coming from the same image and to discriminate such augmentations when coming from different images. In this way the model learns image features with invariant properties to the different augmentations applied. The hypothesis is that the more augmentations one adds to the inputs, the more robust are the features acquired.

In CL, a batch of images is augmented applying diverse augmentations such as crop and resize, flipping, color distortions, rotations, cutouts, Gaussian noise, Gaussian blur and filtering among others. Generally a batch of images is augmented twice, providing two augmented versions of the batch. Afterwards a Neural Network (NN) is trained to maximize agreement between representations produced by augmentations coming from the same image and minimize such an agreement between augmentations coming from different images.

In [4] the NN is composed by two stages–*i.e.* $f(\cdot)$ and $g(\cdot)$. $f(\cdot)$ is called the base encoder network, while $g(\cdot)$ is called the projection head. After training, the projection head is removed using only the output from $f(\cdot)$ for any downstream task.

We approached this same strategy utilizing augmentations coming from our foveated system. We humans perceive visual information as static and well defined scenes even when we do several saccades per second. This means that at each second our retina is receiving information from very different versions of the same perceived scene. In some way, our visual system is considering such different representations provided by different fixations as coming from the same source of information. As a result we see a static scene.

We hypothesize that our propioceptive system, in tandem with our saccades, inform our visual system that we are watching at the same image. Maybe the best way to survive and reproduce is that–under such circumstances–our visual system learned [7] to produce a representation that we perceive as a static and well defined scene. Moving our gaze to another location is reported by our propioceptive system and the information aproaching our retina could be considered as coming from a different source.

Inspired by this biological rationale we applied the method developed in [4] but producing the augmented images by means of different fixations coming from our foveator. We only used 4 of the 5 spans showed in Fig. 3. We discarded the first complete span and used instead only the spans from 2 to 5. We also implemented $f(\cdot)$ utilizing a ResNet 50 and $g(\cdot)$ by means of a MultiLayer perceptron (MLP) as in the original implementation.

Additionally we incorporated further augmentations in advance to the foveation process, for instance, we added crop and resize, color distortion, Gaussian noise and grid mask. Such additional augmentation improved representations considerably. Some aspects related to the quality of the representations will be addressed in the following section.

3.3 Supervised Approaches

3.3.1 Linear Evaluation

With the aim of evaluating the learned representations in $f(\cdot)$, we followed the linear evaluation protocol utilized in [4]. To that end we trained a linear classifier on top of the frozen base network $f(\cdot)$, and then tested the accuracy of the linear classifier using it as a proxy for representation quality.

In our case we generated n fixations from our foveator and passed them through our frozen base network $f(\cdot)$. We then collected the n outputs from $f(\cdot)$ and merged then in a unique vector which we used as input for the linear classifier. The n fixations were produced randomly, *i.e.* no pattern was followed to cover the image in any conceivable way with the fixation locations.

[7] When we talk about *learn* here we mean phylogenetic and ontogenetic processes

3.3.2 Processing Sequences of Fixations with a Transformer Architecture

Attention mechanisms–predominantly self–attention–came to the scene playing a more important role in deep feature representation in CV. This strategy captures long-range dependencies within a single sample [29].

Nevertheless, self-attention has a quadratic complexity which could make its implementation difficult, especially in high resolution images with maybe millions of pixels. With the aim of circumventing this problem, alternative architectures are implemented for substituting self-attention [14, 23, 28].

In our approach we propose a different strategy. Instead of encoding positions for each pixel–or patch–in an image, we encode positions for each fixation in the visual field.

The number of fixations executed by our foveator will–logically–tend to be considerably smaller than the number of pixels–or maybe patches–found in an image. For instance, for humans only two fixations suffice to recognize faces [16].

We used the learned representations in our base network $f(\cdot)$ fine-tuning it by means of the architecture showed in Fig. 2 B. We fed a Transformer with a series of learned representations from a sequence of fixations. We used the main organization and strategy introduced by N. Carion et al. [3].

We used a random number of fixations which ranged from 2 to 9. We also used 10 prediction queries which ended up being 10 image class predictions which in a way *voted* for the different classes in imagenet.

3.4 Reinforcement Learning for the Acquisition of Saccadic Behavior

In the system shown in Fig. 2 B, not only the number but also the locations of the successive fixations in an image were chosen at random.

Yet the behavioral patterns found in saccades of biological systems are far from random [26, 22, 27, 2]. Which are the optimization mechanisms behind the oculomotor behavior emergence in biological systems? Compelling research shows that there are links between the dopaminergic reward system and the saccadic behavior of some mammals [21, 15, 17]. Hence, RL in saccadic behavior is amply supported by these data.

Thus, in Fig. 2 C we show the application of RL to our model. As can be seen in the figure, we use a Deep Q Learning strategy to optimize the saccadic behavior of the system to achieve better performance [24].

We use a ResNet-50 architecture which takes the model in Fig. 2 B as the part of the environment that produces rewards in response to changes in its states.

4 Implementation

Regarding the High Performance Computing (HPC) system We used ThetaGPU, which is an extension of Theta supercomputer at the Argonne Leadership Computing Facility (ALCF). ThetaGPU is composed of 24 NVIDIA DGX A100 nodes.

Each DGX A100 node comprises eight NVIDIA A100 Tensor Core Graphical Processing Units (GPUs) that provide 320 gigabytes of GPU memory for running Machine Learning (ML) workflows.

Regarding ML framework and model parallelization All the implementations have been done utilizing the Pytorch ML framework. DistributedDataParallel from Pytorch and mpi4py were used to manage dataset parallel processing. Basically we distributed 8 Message Passing Interface (MPI) processes in 8 GPUs.

MPI manages the communication among processes. DistributedDataParallel strategy–on the other hand–is to replicate the whole model in each MPI process.

Each model replica processes a different part of the dataset. Gradients computed during the backward pass in each model replica are communicated, averaged and used to conduct weight adjustments in each model.

In regards to dataset and pre-processing The dataset used in this work is imagenet ILSVRC 2012. The foveator is implemented using NVIDIA DALI library. With NVIDIA DALI we load, decode, foveate and augment the images from the dataset. DALI takes care of splitting the dataset in different shards in each epoch. Therefore, each network replica in each MPI rank takes care of a part of the dataset which corresponds to such a process in a given epoch. From one epoch to another, the assignment of shards to specific processes rotates in order to provide variation to the training process.

Regarding software compatibility To run our ML workflow, we used Singularity containerization. Singularity is a container system specifically designed for HPC that allowed us to define our own environment making our work portable and reproducible on any HPC that supports it. Therefore we proceeded to install all the necessary software in a singularity container and afterward we could use such a container to run our models in ThetaGPU.

5 Preliminary Results

The preliminary results of our experiments are shown in Table 1. Here we show results of contrastive accuracy while training the base network $f(\cdot)$, the linear evaluation of the frozen base network, the classification performance of the transformer architecture and finally the performance of the same transformer when successive fixations are guided by a DQN.

To train the base network we used a mini batch size of 128 images in 8 GPUs (*i.e.* batch size of 1024 images). This pre-training phase took 300 epochs, with *adam* optimizer, with a linear decaying *learning rate* schedule and with 5 *warm-up* epochs. The base network utilized was a ResNet 50. With a global batch size of 1024 images we end up with 2048 augmented fixations, each fixation has one positive example and 2046 negative examples in the CL approach.

For the linear evaluation we used the base network (*i.e.* our pre-trained ResNet 50) with its weights frozen and added a linear classifier at the top. We used 5 fixations for each image, without any augmentation. We applied a mini batch size of 512 images in 8 GPUs (*i.e.* batch size of 4096 images). The total number of epochs was 500 with 5 warm-up epochs. We used the same learning rate schedule used for the CL task.

For the Transformer training process we used the base network as a pre-trained backbone fine-tuning its weights with reduced learning-rate. We used a random number of fixations for each image which ranged from 2 to 9. We applied a mini batch of 64 images in 8 GPUs (*i.e.* batch size of 512 images). The total number of epochs was 68 without learning-rate scaling schedule.

6 Conclusion and Future Work

In this paper we compiled a series of methods aimed to find solutions to CV challenges by bio-inspired strategies. Our focus is in the application of foveation and saccadic behavior–in a developmental fashion–to achieve data set and computational savings in CV tasks without diminishing performance significantly. Although foveation provides notorious computational savings in the information processing flow, during experimentation we noticed that it also compromised performance in downstream CV tasks considerably (see Tab. 1).

One important aspect that could be causing this decline is the fact of considering only one positional location per fixation in section 3.3.2. Inside each fixation, there exist much more information corresponding to the complete foveation. From a tiny fraction of the visual field to almost its complete range, one foveation spans almost the entire image from low resolution wide spans to higher resolution acute spans at the center of fixation (Fig. 3).

Incorporating such information to the processing flow of the Transformer in the system could drastically improve the model's performance. Hence, a suitable strategy to follow is the one implemented by Jonnalagadda et al. [19]. In such a model, 11 Transformer blocks (0 to 10) process single fixations. Each of these blocks uses positional embeddings inside each foveation. That is, each foveation comprise all the positional information.

Table 1. Performance in contrastive learning when training the base network $f(\cdot)$, in the linear evaluation of the frozen base network and finally in the classification using a Transformer architecture

Top-1 Acc.	Contrastive acc.	Linear evaluation	Transformer class.	DQL class.
All augmentations	0.7696	0.2093	0.0937	0.0561
Without grid mask	0.8601	0.2502	0.1289	——

Then, the last Transformer block in this set (Transformer block number 10) provides its attention weights to chose the coordinates of the best next fixation location. A final Transformer block (Transformer 11) collects the successive outputs, corresponding to each fixation (as a sequence of fixations). The output from Transformer block 11 is used to classify the image. As we can see, in this model, positional information is managed inside each fixation (foveation).

In our model instead, we collapse all the information corresponding to one fixation in one position and use the positional information corresponding to the centers of the fixations. Our strategy provides an enormous computational saving regarding the quadratic complexity concerning Transformers but it could also be the source in the lost of information that is producing a sharp performance decline in our system. Future applications will take into account this issue, incorporating in some way positional information inside each fixation.

In regards to the acquisition of the saccadic behavior proposed in section 3.4, the RL mechanism "sees" the classifier introduced in section 3.3.2 as its environment. The RL algorithm receives the successive outputs from the foveator as the states of the environment and the reward is the classification performance of the Seq2Seq classifier. As expected, the actions produced by the system control the next fixation coordinates.

The problem that instantaneously arise in the approach is that the classifier–which is considered as the environment by the RL system–is highly dynamic. The classifier's behavior changes continuously as a result of its training. This circumstance makes extremely difficult for the RL algorithm to learn the environment behavior and can in this way "catch" a good policy at time of generating future fixations.

Several possible solution strategies arise in such regard, one is to alternatively freeze the classifier (the environment) and the RL mechanism as training proceeds. We could freeze the RL algorithm during one epoch and the classifier during the next one or maybe use several frozen epochs alternatively to promote stabilization in each algorithm.

In our case, we train the two networks together. In the first epoch we give a preliminary training to the Seq2Seq classifier using random fixation coordinates. In this first epoch, data is accumulated in a memory which collects information regarding states, actions, next states and rewards. This memory is used in the meantime to train the RL algorithm. At each batch the RL algorithm is trained at random, consuming the memory.

Next, in the subsequent epochs the training process of the classifier continues but the sequence of fixations is chosen by following actions accordingly to an epsilon greedy policy from the DQN. Briefly, sometimes we use our DQN to choose the action, and sometimes we just sample one randomly. The probability of choosing a random action starts high at the beginning and decays exponentially towards as training proceeds epoch by epoch.

This strategy is not returning good results either (Tab. 1). The dynamic character of the classifier makes us think in the application of more sophisticated RL strategies. Unexpected perturbations or unseen situations in RL scenarios cause proficient but specialized policies to fail at test time. Here our main problem is that the learning process in RL requires a huge number of trials every time the environment is modified. Animals instead learn new tasks in only a few trials, exploiting their prior knowledge about the world. We need a RL system that could adapt quickly to the changes produced in the classifier (our environment in this case). Several groups have tackled such a challenge [12].

In the next steps of our research we will inspire our model in the work produced in [19], incorporating positional information inside foveation in some way. There are many ways to do that, but we have to try to find the best way given the semantic behind images and foveation. In a second stage we will inspire our RL strategy trying to incorporate meta-RL to adapt rapidly the changes of the environment (our Seq2Seq classifier) [12].

7 Conclusions

CV as a sub-field of ML is a specific discipline in which humans prepare machines, making them able to autonomously do some of the visual task we do routinely. While humans and animals can naturally solve some of the more challenging CV tasks for machines, the possibility that machines provide in terms of scalability is peerless by biological agents in general. Therefore, it is paramount to provide machines with such a natural ability to solve routine CV problems at scale.

Yet, one of the far-reaching challenges in CV is our inability to understand the human visual system, which we think is cardinal for this endeavour. In this paper we report a series of steps devoted to solve some of the major challenges faced by CV–*i.e.* data set scarcity and algorithmic high computational demands–precisely, proposing a compendium of methodologies inspired in the visual system found in biological agents.

We fused SS learning with active foveated vision with the aim of palliating excessive data set and computational demands. We also noticed that the implementation of such methods seriously degrade performance in simple CV tasks. In this paper we propose alternative solutions to be implemented in future editions of this research.

Acknowledgments

This work used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy, Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

1. **Akbas, E., Eckstein, M. P. (2017).** Object detection through search with a foveated visual system. PLOS Computational Biology, Vol. 13, No. 10, pp. e1005743. DOI: 10.1371/journal.pcbi.1005743.
2. **Alahyane, N., Lemoine-Lardennois, C., Tailhefer, C., Collins, T., Fagard, J., Doré-Mazars, K. (2016).** Development and learning of saccadic eye movements in 7- to 42-month-old children. Journal of Vision, Vol. 16, No. 1, pp. 6–6. DOI: 10.1167/16.1.6.
3. **Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020).** End-to-end object detection with transformers. European Conference on Computer Vision, Springer, Vol. 12346, pp. 213–229. DOI: 10.1007/978-3-030-58452-8_13.
4. **Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020).** A simple framework for contrastive learning of visual representations. International conference on machine learning, PMLR, pp. 1597–1607.
5. **Cordonnier, J. B., Mahendran, A., Dosovitskiy, A., Weissenborn, D., Uszkoreit, J., Unterthiner, T. (2021).** Differentiable patch selection for image recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2351–2360.
6. **Dematties, D., Rizzi, S., Thiruvathukal, G. K., Pérez, M. D., Wainselboim, A., Zanutto, B. S. (2020).** A computational theory for the emergence of grammatical categories in cortical dynamics. Frontiers in Neural Circuits, Vol. 14, pp. 12. DOI: 10.3389/fncir.2020.00012.
7. **Dematties, D., Rizzi, S., Thiruvathukal, G. K., Wainselboim, A., Zanutto, B. S. (2019).** Phonetic acquisition in cortical dynamics, a computational approach. PLoS ONE, Vol. 14, No. 6, pp. e0217966. DOI: 10.1371/journal.pone.0217966.
8. **Dematties, D., Thiruvathukal, G. K., Rizzi, S., Wainselboim, A., Zanutto, B. S. (2020).**

Towards high-end scalability on biologically-inspired computational models. Parallel Computing: Technology Trends, IOS Press, Vol. 36, pp. 497 – 506. DOI: 10.3233/APC200077.

9. **Deza, A., Eckstein, M. P. (2016).** Can peripheral representations improve clutter metrics on complex scenes?. Advances in Neural Information Processing Systems, Vol. 29.
10. **Deza, A., Jonnalagadda, A., Eckstein, M. (2018).** Towards metamerism via foveated style transfer. arXiv preprint arXiv:1705.10041. DOI: 10.48550/arXiv.1705.10041.
11. **Deza, A., Konkle, T. (2021).** Emergent properties of foveated perceptual systems. arXiv preprint arXiv:2006.07991. DOI: 10.48550/arXiv.2006.07991.
12. **Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., Abbeel, P. (2016).** RLˆ2: Fast reinforcement learning via slow reinforcement learning. arXiv preprint arXiv:1611.02779.
13. **Fang, F., Ma, Z., Qing, L., Miao, J., Chen, X., Gao, W. (2008).** Face reconstruction using fixation positions and foveated imaging. 8th IEEE International Conference on Automatic Face & Gesture Recognition, pp. 1–6. DOI: 10.1109/AFGR.2008.4813393.
14. **Guo, M. H., Liu, Z. N., Mu, T. J., Hu, S. M. (2021).** Beyond self-attention: External attention using two linear layers for visual tasks. arXiv preprint arXiv:2105.02358.
15. **Hikosaka, O., Nakamura, K., Nakahara, H. (2006).** Basal ganglia orient eyes to reward. Journal of Neurophysiology, Vol. 95, No. 2, pp. 567–584. DOI: 10.1152/jn.00458.2005.
16. **Hsiao, J. H. W., Cottrell, G. (2008).** Two fixations suffice in face recognition. Psychological Science, Vol. 19, No. 10, pp. 998–1006. DOI: 10.1111/j.1467-9280.2008.02191.x.
17. **Ikeda, T., Hikosaka, O. (2003).** Reward-dependent gain and bias of visual responses in primate superior colliculus. Neuron, Vol. 39, No. 4, pp. 693–700. DOI: 10.1016/S0896-6273(03)00464-1.
18. **Jaramillo-Avila, U., Anderson, S. R. (2019).** Foveated image processing for faster object detection and recognition in embedded systems using deep convolutional neural networks. Conference on Biomimetic and Biohybrid Systems, Springer, Vol. 11556, pp. 193–204. DOI: 10.1007/978-3-030-24741-6_17.
19. **Jonnalagadda, A., Wang, W., Eckstein, M. P. (2021).** Foveater: Foveated transformer for image classification. arXiv preprint arXiv:2105.14173. DOI: 10.48550/arXiv.2105.14173.
20. **Kaplanyan, A. S., Sochenov, A., Leimkühler, T., Okunev, M., Goodall, T., Rufo, G. (2019).** Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. ACM Transactions on Graphics (TOG), Vol. 38, No. 6, pp. 1–13. DOI: 10.1145/3355089.3356557.
21. **Kato, M., Miyashita, N., Hikosaka, O., Matsumura, M., Usui, S., Kori, A. (1995).** Eye movements in monkeys with local dopamine depletion in the caudate nucleus. I. Deficits in spontaneous saccades. Journal of Neuroscience, Vol. 15, No. 1, pp. 912–927. DOI: 10.1523/JNEUROSCI.15-01-00912.1995.
22. **Kowler, E. (2011).** Eye movements: The past 25 years. Vision Research, Vol. 51, No. 13, pp. 1457–1483. DOI: 10.1016/j.visres.2010.12.014.
23. **Melas-Kyriazi, L. (2021).** Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. arXiv preprint arXiv:2105.02723. DOI: 10.48550/arXiv.2105.02723.
24. **Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. (2013).** Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
25. **Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W. C., Matia, A.-S. L., Mcnamara, J. O., Williams, S. M. (2019).** Neurosciences,

chapter 11. De Boeck Supérieur, 6 edition, pp. 259–282.

26. **Ross-Sheehy, S., Reynolds, E., Eschman, B. (2020).** Evidence for Attentional Phenotypes in Infancy and Their Role in Visual Cognitive Performance. Brain Sciences, Vol. 10, No. 9, pp. E605. DOI: 10.3390/brainsci10090605.
27. **Spotorno, S., Malcolm, G. L., Tatler, B. W. (2014).** How context information and target information guide the eyes from the first epoch of search in real-world scenes. Journal of Vision, Vol. 14, No. 2, pp. 7–7. DOI: 10.1167/14.2.7.
28. **Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. (2021).** Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems, Vol. 34, pp. 24261–24272.
29. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. Advances in Neural Information Processing Systems, Vol. 30.

Article received on 07/05/2022; accepted on 18/09/2022.
Corresponding author is Dario Dematties.