FOVEATER: FOVEATED TRANSFORMER FOR IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many animals and humans process the visual field with varying spatial resolution (foveated vision) and use peripheral processing to make eye movements and point the fovea to acquire high-resolution information about objects of interest. This architecture results in computationally efficient rapid scene exploration. Recent progress in vision Transformers has brought about new alternatives to the traditionally convolution-reliant computer vision systems. However, the Transformer models do not explicitly model the foveated properties of the visual system nor the interaction between eye movements and the classification task. We propose foveated Transformer (FoveaTer) model, which uses pooling regions and eye movements to perform object classification tasks using a vision Transformer architecture. Our proposed model pools the image features using squared pooling regions, an approximation to the biologically-inspired foveated architecture, and uses the pooled features as an input to a Transformer Network. It decides on subsequent fixation locations based on the attention assigned by the Transformer to various locations from previous and present fixations. The model uses a confidence threshold to stop scene exploration, dynamically allocating more fixation/computational resources to more challenging images. After reaching the stopping criterion, the model makes the final object category decision. We construct a Foveated model using our proposed approach and compare it against a Full-resolution model, which does not contain any pooling. On the ImageNet-100 dataset, our Foveated model achieves the accuracy of the full-resolution model using only 35% transformer computations and 73% overall computations. Finally, we demonstrate our model's robustness against adversarial attacks, where it outperforms the full-resolution model.

1 INTRODUCTION

Many mammals, including humans, have evolved a locus (the fovea) in the visual sensory array with increased spatial fidelity and use head and eye movements (Land, 2012; Marshall et al., 2014) to orient such locus to regions and objects of interest. The system design allows visual-sensing organisms to accomplish two objectives: fast target detection crucial for survival and savings in the computational cost. Computational savings are accomplished by limiting the number of units with high computational costs (i.e., higher spatial resolution processing) to the fovea's small spatial region. Fast target detection is achieved by distributing the remaining computational power across a much larger area in the periphery, with a lower spatial resolution with increasing distance from the fovea. Critical to the design is an efficient algorithm to guide through eye movements the high-resolution fovea to regions of interest using the low-resolution periphery (Hayhoe & Ballard, 2005; Strasburger et al., 2011; Ludwig et al., 2014) and allow optimizing the target detection and scene classification. Various computational models were proposed to model the search using foveated visual system (Yamamoto et al., 1996; Prince et al., 2005).

Computer vision has evolved from using hand-crafted features to data-driven features in modern CNNs. Due to their computational limitations, the objectives of the computer vision systems align well with those of human visual system: to optimize visual detection and recognition with an efficient computational and metabolic footprint. Approaches towards saving computational power can be seen; for example, computer vision systems evolved from using sliding windows to RCNN's (Gir-

shick et al., 2014) use of selective search and Faster-RCNN's (Ren et al., 2015) use of Region Proposal Network (RPN).

A system that mimics the human vision by processing the scene with a foveated system and rational eye movements has also been proposed. This approach to exploring the scene can be seen in models like RAM (Mnih et al., 2014) for recognizing handwritten single-digits or detecting objects (Akbas & Eckstein, 2017) where they sequentially process the image and decide what to process next by using the peripheral information. These foveated models reach that approaches that of full-resolution models but using a fraction of the computations. Foveated systems have also shown to result in more robustness (Luo et al., 2015; Deza et al., 2019; Deza & Konkle, 2020; Kiritani & Ono, 2020; Vuyyuru et al., 2020) against adversarial attacks.

There has been a recent innovation in computer vision of using Transformers (Touvron et al., 2020; Dosovitskiy et al., 2020) for object classification tasks which departs from the traditional over-reliance on convolutions. Even after replacing the convolutions with attention modules and multilayer perceptrons, visual Transformers (Dosovitskiy et al., 2020; Touvron et al., 2020) achieve close to state-of-the-art performance on the ImageNet dataset and provide better robustness against adversarial attacks (Shao et al., 2021).

Due to the flattened architecture of the transformers, it is easier for multi-resolution features to share the same feature channels. Transformers (Vaswani et al., 2017) have the added benefit of self-attention using which locations far apart in the image will interact with each other.No papers have evaluated the additional potential gains of incorporating a foveated architecture into transformers.

Here, we evaluate the effect of a foveated architecture and sequential eye movements on a state of the art transformer model sitting on a convolutional backbone. We compare the foveated transformer relative to the full-resolution model in terms of classification accuracy and robustness to adversarial attacks. Furthermore, we extend previous papers by investigating the separate contributions to robustness related to the foveated processing (Luo et al., 2015; Deza et al., 2019; Deza & Konkle, 2020; Kiritani & Ono, 2020; Vuyyuru et al., 2020) and that of the process of fixating images at different locations.

To this end, we perform an object classification task using multiple fixations, moving foveal attention across different parts of the image, and using only a limited portion of the image information at each fixation, thereby reducing the input to the transformer by many folds. The model decides on subsequent fixation locations using the learned self-attention weights and fixation locations of all the fixations until the current step. Finally, the model integrates information across fixations using average-pooling to make the final classification decision.

2 Related work

Transformers have achieved great success in the field of Natural Language Processing since their introduction by Vaswani et al. (2017) for machine translation. Recently, the application of Transformer models in computer vision has seen tremendous success. Vision Transformer (ViT) model introduced by Dosovitskiy et al. (2020) achieved remarkable performance on ImageNet (Deng et al., 2009) by using additional data from JFT 300M (Sun et al., 2017) private dataset. Subsequently, the DeiT model (Touvron et al., 2020) introduced knowledge transfer concepts in transformers to leverage the learning from existing models. Using augmentation and knowledge transfer, the DeiT model achieved close to state-of-the-art performance using training data from the ImageNet dataset alone.

Sequential processing provides three main advantages in computer vision. Larochelle & Hinton (2010) proposed a model based on the Boltzmann machine that uses foveal glimpses and can make eye movements. First, it can limit the amount of information to be processed at a given instant to be constant, i.e., the ability to keep computations constant irrespective of the input image size. Second, sequential models can help model human eye movement strategies and help transfer that information to build better computer vision systems. RAM (Mnih et al., 2014) introduced a sequential model capable of making a sequence of movements across the image to integrate information before classification. In addition, the hard-attention mechanism, implemented using reinforcement learning, was used to predict the sequence fixation locations. Ba and Minh (Ba et al., 2015) extended these ideas to recognize multiple objects in the images on a dataset constructed using MNIST. Third, sequential processing requires fewer parameters than a model using full-resolution image input. Other

models (Xu et al., 2015) have proposed image captioning models based on both hard-attention and soft-attention.

Computational models of categorization and eye movements have been proposed for rapid categorization in terms of low-level properties such as spatial envelopes (Oliva & Torralba, 2001) and texture summary statistics (Rosenholtz et al., 2012). Saliency-based models (Koch & Ullman, 1987; Itti et al., 1998; Itti & Koch, 2000) traditionally tried to model eye movements by identifying bottom-up properties in the image that will capture attention. Torralba et.al (Torralba et al., 2006) showed how saliency could be combined with contextual information to guide eye movements. These models do not explicitly incorporate foveated vision. Akbas (Akbas & Eckstein, 2017) implemented a biologically-inspired foveated architecture (Freeman & Simoncelli, 2011) with a deformable parts model to build a foveated object detector which accuracy was close to a full-resolution model but using a fraction of the computations (Akbas & Eckstein, 2017).

FoveaTer combines an approximation to biologically-inspired foveated architecture with a Vision Transformer Network. We apply our model to real-world images from the ImageNet dataset for the task of image classification. We pool the input to the transformer using the pooling architecture described in the following sections based on the fixation location, which reduces the number of inputs to the transformer. The subsequent fixation location is given by a confidence map, constructed using the attention weights from the last transformer block. Feature vectors corresponding to the class token from different fixations are averaged using an average-pooling layer followed by a fully connected layer, resulting in the final classification decision.

A novel aspect of the proposed work is that the model also learns that all images are not equally difficult to classify adapting the exploration eye movements to different image classes and thus varying computational resources used to classify different images successfully. The model implements this idea using a confidence threshold to restrict the scene exploration to the necessary fixations to classify the image.

Also novel is an evaluation the adversarial robustness of our model to understand the separate contributions of the foveated architecture and that of sequential fixations towards defense against adversarial attacks. We use the fast gradient sign method (FGSM (Goodfellow et al., 2014)), which is an adversarial attack that computes the adversarial image by backpropagating the gradients once, and the projected gradient descent method (Kurakin et al., 2017; Madry et al., 2018), which iteratively computes the adversarial image, as our adversarial attacks.

3 MODEL

DeiT-Tiny (Touvron et al., 2020): The DeiT-Tiny architecture begins with a convolution embedding layer that transforms the $[3, 224, 224]$ input image into a $[192, 14, 14]$ representation that is input into a series of twelve transformer blocks, each sized for a 192-dimensional embedding.

Full-resolution model: Our Full-resolution model refers to a hybrid adaptation, transformer network with convolutional backbone, of the DeiT-Tiny architecture using the same number of parameters. This hybrid model replaces the single convolutional layer of the DeiT-Tiny model with a convolutional backbone. The convolutional backbone is composed of first two stages of the ResNet-18 (He et al., 2016) architecture, which transforms the input from $[3, 224, 224]$ to $[128, 28, 28]$, followed by three convolutional layers to acheive a final feature map of size $[192, 14, 14]$. The first convolutional layer transforms the input feature dimensionality from 128 to 192 using kernels of size 3 and stride 1, resulting in $[192, 28, 28]$. The second convolutional layer downscales the features using a kernel of size 3 and a stride of 2, resulting in features of size $[192, 14, 14]$. The third convolutional layer uses a kernel size of 1 and stride of 1, retaining the size of the feature map. The number of transformer blocks is reduced from 12 to 9 to match the number of parameters with the DeiT-Tiny model.

Foveated model: Extending the hybrid architecture described above, our Foveated model analyzes a pooled-version of the features from the convolution backbone at each time step and arrives at the final category decision after making the required number of time steps to satisfy the decision criterion. We show the architecture of our model in Figure 1. The input image is first passed through the convolutional backbone described above, resulting in size $[192, 14, 14]$. Then, we perform fixation dependent average-pooling, using the "Foveation module" (see next section), on the features from

the convolutional backbone resulting in features of size $[192, 29]$, where the fixation refers to one of the 196 possible spatial locations in the input feature map. Under this non-uniform average-pooling model, locations closer to the fixation location use smaller neighborhoods for pooling than locations far from the fixation location. Pooled features of size $[192, 29]$ along with the class token are passed through the nine transformer blocks, where class token refers to a learnable vector of size 192 values. Since the transformer blocks retain the dimensionality of the input, the output of the transformer blocks has the same size of $[192, 30]$. We use the self-attention weights, corresponding to the class token feature vector from the last transformer block, in the "Accumulator module" (see next section) to predict the following fixation location. Feature vector corresponding to the class token is given as input to the average-pooling layer, averaging across such vectors received from all the fixations. Finally, the classification layer transforms the average-pooled feature vector into a logits vector.

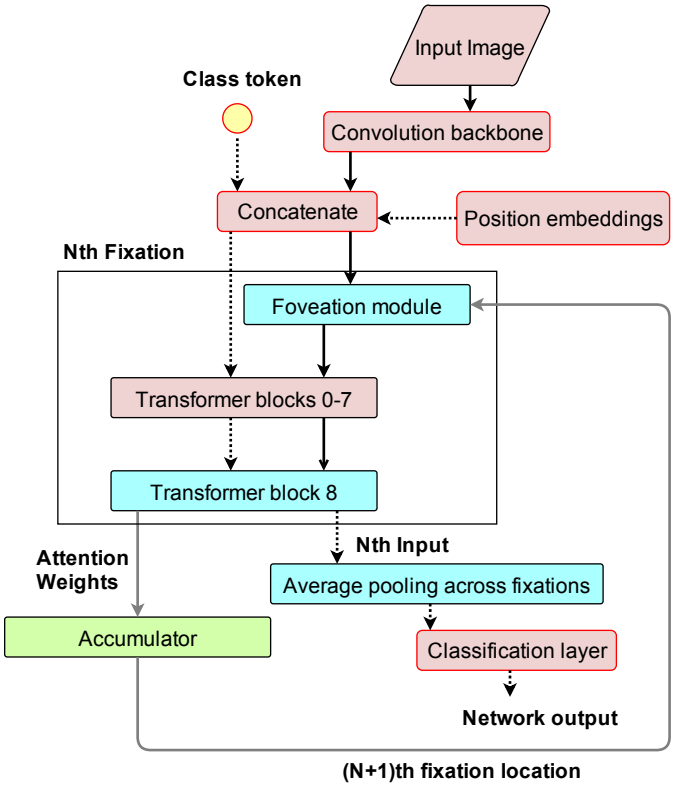


Figure 1: **FoveaTer architecture:** Solid black arrows denote the flow of image-related features. The foveation module performs fixation-dependent pooling, on the full-resolution features ($[192, 14, 14]$) output of the convolution backbone, resulting in pooled feature vectors ($[192, 29]$). During training, a random location in the image serves as the initial fixation. *Accumulator* uses the attention weights from the last transformer block to predict the next fixation location. The Average-pooling layer takes the feature vector corresponding to the class token as input from each past fixation, followed by a final classification layer.

3.1 FOVEATION MODULE

Inspired from the radial-polar pooling regions proposed in Freeman & Simoncelli (2011), we use squared pooling regions in this paper for computational speed-up. Each image in a mini-batch has a corresponding fixation location, where the fixation location represents the center of the visual field shown in Figure 2, thereby allowing us to align the input image/feature map with the visual field. After aligning the input feature map with the visual field, features falling within a pooling region are average-pooled, and the resultant pooled vector represents that pooling region. Highlighted regions in Figure 2 correspond to the centers of the pooling regions, 49 blocks.

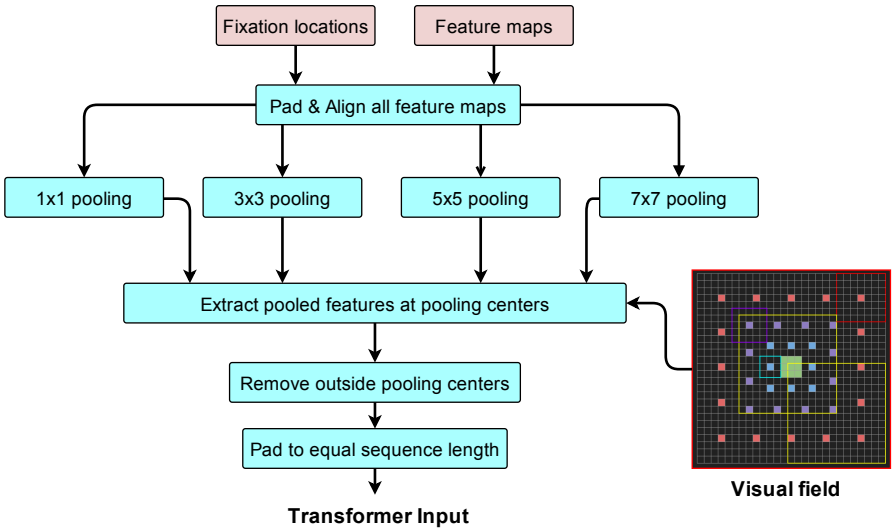


Figure 2: **Foveation module:** Inputs to the module are aligned to the visual field and are zero-padded to have the same size as the visual field. We pool the input features according to the pre-defined pooling neighborhoods in the visual field, resulting in one feature vector for each of the 49 pooling regions in the visual field, colored blocks in the visual field figure correspond to the centers of these 49 pooling regions. After ignoring the pooling centers falling in the padded region instead of on the original image, the remaining pooled features are zero-padded to a constant sequence length of 29.

We use pooling regions with receptive field sizes 1×1, 3×3, 5×5, 7×7 blocks on a feature map of size 14×14 blocks, which is the output of the convolution backbone. Each block corresponds to a $[16, 16]$ pixel region in the input image of width and height 224. The visual field is of size 27×27 blocks. Central green 3×3 block represents the high-resolution Fovea, where there is no average-pooling. The next ring of pooling regions, where the center of the pooling region is blue, has a receptive field of 3×3 which translates to average-pooling of 9 feature vectors to generate the representative feature vector for that pooling region. Similarly, the rings of violet and red-colored pooling centers have receptive fields of 5×5 and 7×7 blocks, respectively. Yellow squares are of size 14×14 blocks, i.e., equal to the feature map of the image, thereby illustrating the size of the image size to the visual field, which is approximately one-quarter.

We ignore the pooling regions falling outside the image, and the output of the foveation module is zero-padded to maintain the same length across all images present in a batch since different fixation locations give rise to an unequal number of active pooling regions. We use 29 as the feature length as it is the maximum possible number of active pooling regions. The feature vector corresponding to each pooling region is computed as follows,

$$P_i = (1/M) \sum_{j=0}^{M-1} E_{ij} \tag{1}$$

Where $i = \{1, \ldots, 49\}$ corresponds to the pooling region index, E_{ij} is the embedded feature vector present in that pooling region, $M = \{1, 9, 25, 49\}$ corresponds to number of features inside the pooling region and P_i is the mean feature vector corresponding to that pooling region.

3.2 GUIDED FIXATION SEQUENCE PREDICTION:

Self-attention weights from the last transformer block are collected to get each pooling region's contribution to the feature vector corresponding to the class token. A confidence map is constructed based on the fixation point location by putting these weights back on a 14×14 map at the corresponding pooling region's location, where 14×14 corresponds to the size of the input feature map. After each fixation, we degrade the attention weights in the confidence map at a rate of 50%. Inhibition of return (IOR) (Dukewich & Klein, 2015) refers to a tendency in human observers not to attend at previously attended or fixated regions. We constructed an IOR map by applying an IOR of 3×3 blocks around a fixation location and degraded it by 50% after each fixation. After subtracting the IOR map from the confidence map, the following fixation location is extracted based on the location of the maximum in the confidence map.

Loss function: We use Cross-entropy for computing the classification loss. Loss from all fixations is incorporated to get the mini-batch loss, as shown below.

$$loss = \sum_{i=1}^{N} \sum_{j} y_j \log(p(y_j)) + (1 - y_j) \log(1 - p(y_j)) \tag{2}$$

Where i corresponds to the fixatiothe n index, $N = 5$ for the Foveated model & $N = 1$ for Full-resolution model, i.e., single-pass, y corresponds to the target label, j corresponds to the class index, and p correspond to the predicted probability.

4 RESULTS

We train the model weights from scratch and no pre-trainng. The Foveated model is trained for five fixations to have a computational complexity lower than the Full-resolution model. In the following sub-sections, we compare the performance and computational complexity of the Foveated model performance against the Full-resolution model. In addition, we compare the adversarial robustness of the Foveated and Full-resolution models.

4.1 IMPLEMENTATION DETAILS

Training: We train all the 5.5M parameters of the model. Using initial learning of $1e-5$ and a minimum learning rate of $1e-6$, we train both Foveated and Full-resolution models for 300 epochs. We use AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2019) optimizer with a decay of $1e-8$ and a cosine learning rate schedule. We use the hyperparameters and augmentation settings from DeiT implementation (Touvron et al., 2020). We use ImageNet-100 (Tian et al., 2020) and ImageNet (Deng et al., 2009) datasets for the results shown in the following sub-sections. For ImageNet-100, We use the train validation split provided by the ImageNet and report results on the validation set. We use GeForce GTX 1080 Ti for training and testing purposes. Both Foveated and Full-resolution models take 25 hrs and 21 hrs respectively to train for 300 epochs on three GPUs. The Foveated model made five fixations on each input image during training, with the initial fixation starting at a random location. Due to our model's flexibility, the same model can make any desired number of fixations.

Dynamic-stop of Fixation Exploration: Due to various factors such as occlusion, camera angle, brightness, the difficulty of making a classification decision varies across images. To achieve higher computational efficiency in our Foveated model during inference, we stop exploring the images with fixations when the predicted class with the highest probability reaches a pre-defined threshold corresponding to that class. The threshold equals the mean probability of all the correct predictions belonging to that class in the training dataset.

Images/Sec: We report the number of inferences completed by the GPU during a one-second time interval to compare the computational complexity of different models during inference time.

Initial fixation for the Foveated model: The input feature map to the *Foveation module* has a spatial size of 14×14, and thus consider any one of these 196 fixation locations. During training, a random location is selected as the initial fixation point, and the model guides subsequent fixations. During inference, for the results presented in Table 1 and Table 2, the initial fixation is set at the center of the image to remove any variation in results arising from the randomness in the initial fixation location.

4.2 TOP-1 ACCURACY:

The full-resolution model allows the long-range full-resolution interactions between the feature vectors, whereas the Foveation model serially explores the image allowing only a limited number of interactions between pooled feature vectors at each given fixation. Using the guided fixation prediction method described in the previous section, the Foveated model makes four guided fixations after the initial random fixation. Sample fixations made by the model are shown in Figure 3a and average number of fixations made by the model per class are shown in Figure 3b.

Table 1: **Performance on ImageNet100 (Tian et al., 2020) dataset:** We find that the Foveated model outperforms the Full-resolution model both in terms of performance and computational complexity. Pure-Trans refers to the model with a single convolutional layer, and Hybrid refers to the models with convolutional backbone at the start of the transformer as described in the previous sections. For evaluation, the initial fixation for the Foveated model is chosen at the center of the image to remove effects due to variations of the initial fixation location.

Model	Type	Params	PC	Conv Flops	Trans Flops	Total Flops
Deit-Tiny	Pure-Trans	5.5M	76.3	0.029	1.046	1.07
Full-resolution **(Ours)**	Hybrid	5.3M	84.58	1.245	0.697	2.03
Foveated **(Ours)**	Hybrid	5.3M	84.66	1.245	0.531	1.84
Foveated with Dynamic-Stop **(Ours)**	Hybrid	5.3M	**84.68**	1.245	**0.245**	1.49

Table 2: **Performance on ImageNet dataset:** We compare our models against the state-of-the-art models with a similar number of parameters using three metrics: **Top-1 accuracy**, **Image throughput**, and **Adversarial robustness**. For reporting adversarial robustness, ϵ is set at 0.01, and the adversarial attack is computed for the transformer front-end only; therefore, we do not report comparisons to the convolutional models. **DS:**Dynamic-Stop. Throughput for the dynamic-stop model is computed as a weighted combination of throughputs of the Foveated model with a constant number of fixations, i.e., 1 to 5 fixations.

Model	Type	Params	Images/sec	Top-1	Top-1-Adv (ϵ = 0.01)	
					FGSM	PGD
ResNet-18 (He et al., 2016)	Conv	11.7M	1505	69.8	—	—
EfficientNet-B0 (Tan & Le, 2019)	Conv	5M	720	77.1	—	—
Deit-Ti (Touvron et al., 2020)	Pure-Trans	5.7M	796	72.2	46.53	36.5
Deit-Ti (distilled)	Trans-ConvT	6M	796	74.5	—	—
Full-resolution **(Ours)**	Hybrid	5.5M	652	73.6	39.37	28.98
Foveated **(Ours)**	Hybrid	5.5M	637	70.2	**59.24**	**57.67**
Foveated-DS **(Ours)**	Hybrid	5.5M	960	69.9	—	—

4.2.1 RESULTS ON IMAGENET-100

Due to the hybrid nature of the models proposed, incorporating a CNN backbone and a transformer network, the Foveated and Full-resolution models outperform the DeiT-Tiny model, as shown in Table 1. The Foveated model's accuracy is slightly above the Full-resolution model but critically has lower computational cost. Critically, the last row shows that with the addition of the dynamic-stop, the FoveaTer model can achieve close to 65% computational savings in transformer flops and 26% savings in terms of overall flops while retaining the performance achieved by the Full-resolution model.

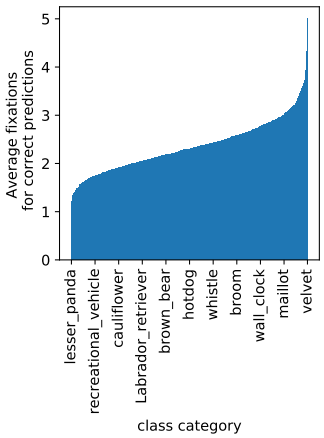
4.2.2 RESULTS ON IMAGENET

We present the results on the ImageNet dataset in Table 2. The Foveated model achieves an accuracy above the ResNet-18 model. For the dynamic-stop, we first compute the number of images falling in each of the five fixation bins and throughputs of the Foveated models with one to five fixations. The throughput of the Dynamic-stop model is computed as the weighted average of the throughputs of individual fixation models using the bin count as the weights. Using the dynamic-stop, the throughput of the the Foveated model is improved ($637 \rightarrow 960$), as shown in the last row.

4.3 ROBUSTNESS AGAINST ADVERSARIAL ATTACKS

We demonstrate the robustness of our foveated system in this section. We consider the two adversarial attacks, Fast Gradient Sign Method (FGSM Goodfellow et al. (2014)) and Projected Gradient Descent (PGD Kurakin et al. (2017); Madry et al. (2018)), to compare the robustness of Foveated and Full-resolution models. We use Cleverhans library (Papernot et al., 2018) for implementing the adversarial attacks.

(a) **Fixation sequences:** A maximum of five fixations are made for each image with an adaptive termination of the fixation exploration (dynamic-stop). The initial random fixation is followed by four guided fixations. Each subsequent fixation center is represented with a bigger red circle. Blue arrows represent the transition from one fixation to the next. Green and Red bars on top correspond to correct and wrong predictions respectively.



(b) **Average fixations per class:** Average number of fixations used by each class are plotted on y-axis. Some of the class names are displayed on the x-axis.

Figure 3: Fixation sequences and Average fixations per class

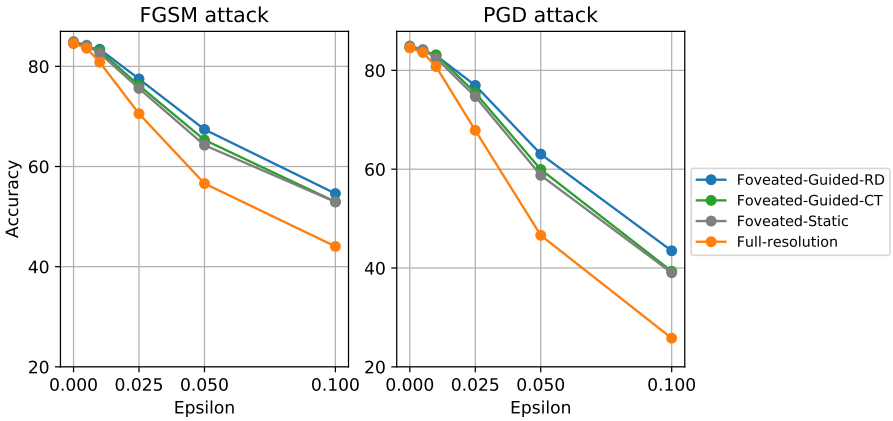


Figure 4: **Adversarial robustness:** Comparison of Foveated model (with guided fixations and two different initial fixations, *RD* - initial random fixation. *CT* - initial fixation at the image center), a foveated model that makes the same fixation sequence for every image (Foveated model with static fixations), and the Full-resolution model. Epsilon represents the strength of the attack. As the strength of the attack increases the accuracy of the Foveated model with guided fixations accuracy is greater than the Foveated model with static fixations while the Full-resolution model has the lowest accuracy. Raw values are available in Table 3 in Appendix.

To compute the adversarial robustness of the model, we separated the network into a convolutional backbone and a Transformer front-end. The adversarial attack is computed only for the Transformer front-end, i.e., features from the convolutional backbone as network input and Transformer front-end as the model. Since our model only affects the Transformer part of the network, this disassociates the interaction of the adversarial attack with the convolutional backend, which does not interact with the foveation module. For this reason, purely convolutional networks are not considered. We compare only the DeiT-Tiny model with Full-resolution and Foveated models. Epsilon corresponds to the strength of the attack.

Overall, Foveated model outperforms both DeiT-Tiny and Full-resolution models in terms of adversarial robustness.

4.3.1 RESULTS ON IMAGENET-100

Figure 4 shows that performance of the Full-resolution model (84.58) and Foveated model (84.62) are similar for zero epsilon value, but as the strength of the attack increases, the Foveated model outperforms the Full-resolution model. The robustness of the Foveated model can be either due to the pooling operations that are part of the model or our model's inherent ability to dynamically decide fixation paths, thereby minimizing the effect of the attack by automatically following a different path from that of the adversarial model.

To isolate effect of foveation from the influence of fixation exploration, We compare various configurations of the Foveated model. For the foveated guided fixations model, we explored a version with a random initial fixation and another version with a constant initial fixation at the center of the image. The model then dynamically selected the subsequent four fixations, thereby allowing the possibility of different fixation paths for the adversarial image computation and model evaluation. The model with the random initial fixation results in more variability in subsequent fixations. Critically, we also used a pre-defined fixation sequence (center initial fixation followed by four fixations at the center of each quadrant), thereby ensuring that the adversarial image computation and the model evaluation that follows will use the same fixation locations. We refer to this model as the Foveated static.

Results in Figure 4, show that the Foveated guided model with an initial random fixation outperforms the static fixation model, suggesting that the variability in fixation sequences contributes towards adversarial robustness. The comparison to the Foveated Guided model with the fixed initial fixation at the center of the image shows that a stochastic component to the initiation can be important to vary the fixation sequence and result in greater robustness.

Finally, we illustrate the contributions of the foveated nature of the model by comparing the Foveated model with the five pre-defined fixations (*grey curve, Foveated static*) to the Full-resolution model (*orange curve*). This pre-defined fixation path Foveated model will also be the desirable solution when the computational time of a serial fixation model is a bottleneck, as all the pre-defined fixations can run in parallel. Figure 4, show that the Foveated model with static fixations outperforms the Full-resolution model illustrating the contributions of the foveated nature(Luo et al., 2015; Deza et al., 2019; Deza & Konkle, 2020; Kiritani & Ono, 2020; Vuyyuru et al., 2020) of the model to robustness to adversarial attacks.

4.3.2 RESULTS ON IMAGENET

We show the results of adversarial attacks in the last two columns of Table 2. Due to architectural changes from the DeiT-Tiny model, the Full-resolution model (*FGSM*-39.4*; PGD*-29) exhibits lower adversarial robustness compared to Deit-Tiny model (*FGSM*-46.5*; PGD*-36.5). On the other hand, Foveated model (*FGSM*-59.2*; PGD*-57.7), which shares architectural similarity with the Full-resolution model, exhibits higher adversarial robustness than both DeiT-Tiny and Full-resolution architectures.

5 CONCLUSION

We provided a comprehensive framework for using foveal processing and fixation exploration on a vision transformer architecture for image classification. The proposed architecture introduces a way to limit computations required to process an image by flexibly adjusting the required number of fixations, allowing larger batch sizes during training, providing robustness to adversarial attacks, and giving us a model that can allocate computational resources based on the difficulty of an image. Our findings reveal that variability in fixation sequences to images can contribute over foveated processing to robustness to adversarial attaches. In conclusion, we leveraged the most recent Vision Transformer architecture and combined it with ideas from foveated vision to come up with a model which has multiple knobs in terms of the number of fixations to be made and limits on the computations done so that the end-user will have the flexibility to fine-tune depending on their needs.

REFERENCES

Emre Akbas and Miguel P. Eckstein. Object detection through search with a foveated visual system. *PLOS Computational Biology*, 13(10):1–28, 10 2017. doi: 10.1371/journal.pcbi.1005743. URL https://doi.org/10.1371/journal.pcbi.1005743.

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.7755.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Arturo Deza and Talia Konkle. Emergent properties of foveated perceptual systems. *ArXiv*, abs/2006.07991, 2020.

Arturo Deza, Aditya Jonnalagadda, and Miguel P. Eckstein. Towards metamerism via foveated style transfer. *ArXiv*, abs/1705.10041, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kristie R. Dukewich and Raymond M. Klein. Inhibition of return: A phenomenon in search of a definition and a theoretical framework. *Attention, Perception, & Psychophysics*, 77(5):1647–1658, Jul 2015. ISSN 1943-393X. doi: 10.3758/s13414-015-0835-3. URL https://doi.org/10.3758/s13414-015-0835-3.

Jeremy Freeman and Eero P. Simoncelli. Metamers of the ventral stream. *Nature neuroscience*, 14: 1195 – 1201, 2011.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014. doi: 10.1109/CVPR.2014.81.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv 1412.6572*, 12 2014.

Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188—194, April 2005. ISSN 1364-6613. doi: 10.1016/j.tics.2005.02.009. URL https://doi.org/10.1016/j.tics.2005.02.009.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. doi: 10.1109/34.730558.

Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10):1489–1506, 2000. ISSN 0042-6989. doi: https://doi.org/10.1016/S0042-6989(99)00163-7. URL https://www.sciencedirect.com/science/article/pii/S0042698999001637.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Taro Kiritani and Koji Ono. Recurrent attention model with log-polar mapping is robust against adversarial attacks. *ArXiv*, abs/2002.05388, 2020.

Christof Koch and Shimon Ullman. *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*, pp. 115–141. Springer Netherlands, Dordrecht, 1987. ISBN 978-94-009-3833-5. doi: 10.1007/978-94-009-3833-5_5. URL https://doi.org/10.1007/978-94-009-3833-5_5.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ArXiv*, abs/1607.02533, 2017.

M. Land. Oculomotor behaviour in vertebrates and invertebrates. *The Oxford Handbook of Eye Movements*, 01 2012. doi: 10.1093/oxfordhb/9780199539789.013.0001.

Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/677e09724f0e2df9b6c000b75b5da10d-Paper.pdf.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Casimir J. H. Ludwig, J. Rhys Davies, and Miguel P. Eckstein. Foveal analysis and peripheral selection during active visual sampling. *Proceedings of the National Academy of Sciences*, 111 (2):E291–E299, 2014. ISSN 0027-8424. doi: 10.1073/pnas.1313553111. URL https://www.pnas.org/content/111/2/E291.

Yan Luo, Xavier Boix, Gemma Roig, Tomaso A. Poggio, and Qi Zhao. Foveation-based mechanisms alleviate adversarial examples. *ArXiv*, abs/1511.06292, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

N Marshall, M. Land, and T Cronin. Shrimps that pay attention: Saccadic eye movements in stomatopod crustaceans. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 369:20130042, 02 2014. doi: 10.1098/rstb.2013.0042.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2204–2212, Cambridge, MA, USA, 2014. MIT Press.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001. ISSN 1573-1405. doi: 10.1023/A:1011139631724. URL https://doi.org/10.1023/A:1011139631724.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

Simon Prince, James H. Elder, Yuqian Hou, Mikhail Sizintsev, and Yevgen Olevskiy. Statistical cue integration for foveated wide-field surveillance. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:603–610 vol. 2, 2005.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

Ruth Rosenholtz, Jie Huang, and Krista Ehinger. Rethinking the role of top-down attention in vision: Effects attributable to a lossy representation in peripheral vision. *Frontiers in Psychology*, 3:13, 2012. ISSN 1664-1078. doi: 10.3389/fpsyg.2012.00013. URL https://www.frontiersin.org/article/10.3389/fpsyg.2012.00013.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *ArXiv*, abs/2103.15670, 2021.

Hans Strasburger, Ingo Rentschler, and Martin Jüttner. Peripheral vision and pattern recognition: A review. *Journal of vision*, 11:13, 05 2011. doi: 10.1167/11.5.13.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852, 2017. doi: 10.1109/ICCV.2017.97.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019. URL http://proceedings.mlr.press/v97/tan19a.html.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 776–794, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58621-8.

Antonio Torralba, Aude Oliva, Monica Castelhano, and John Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological review*, 113:766–86, 11 2006. doi: 10.1037/0033-295X.113.4.766.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Manish Reddy Vuyyuru, Andrzej Banburski, Nishka Pant, and Tomaso Poggio. Biologically inspired mechanisms for adversarial robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2135–2146. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/17256f049f1e3fede17c7a313f7657f4-Paper.pdf.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/xuc15.html.

H. Yamamoto, Y. Yeshurun, and M. Levine. An active foveated vision system: Attentional mechanisms and scan path covergence measures. *Comput. Vis. Image Underst.*, 63:50–65, 1996.

Table 3: Adversarial robustness on ImageNet100: dataset. **Foveated-S-RD:** Foveated model with guided fixations and random initial fixation. **Foveated-P:** Foveated model with pre-defined fixations with parallel execution capability. **Foveated-S-CT:** Foveated model with guided fixations and image center as initial fixation.

	DeiT-Tiny		**Full-resolution**		**Foveated-S-RD**		**Foveated-P**		**Foveated-S-CT**	
Epsilon (ϵ)	**FGSM**	**PGD**	**FGSM**	**PGD**	**FGSM**	**PGD**	**FGSM**	**PGD**	**FGSM**	**PGD**
0	76.34	76.34	84.58	84.58	84.60	84.64	84.94	84.94	84.62	84.62
0.005	69.82	69.26	83.62	83.60	83.82	83.98	84.22	84.22	84.0	83.94
0.01	58.62	55.10	80.82	80.78	83.42	83.02	82.62	82.50	83.12	83.14
0.025	35.90	23.74	70.54	67.86	77.48	76.96	75.58	74.70	76.18	75.46
00.05	23.56	10.14	56.6	46.62	67.40	63.06	64.26	58.76	65.30	59.94
0.1	15.42	6.58	44.04	25.80	54.60	43.48	52.92	39.04	52.96	39.34
0.2	10.24	5.56	35.08	8.38	46.26	25.18	44.80	21.46	44.54	21.28

6 APPENDIX