

Saccadic Vision for Fine-Grained Visual Classification

Johann Schmidt
Artificial Intelligence Lab
Otto-von-Guericke University Magdeburg
Germany
johann.schmidt@ovgu.de

Joachim Denzler
Computer Vision Group
Friedrich Schiller University Jena
Germany
joachim.denzler@uni-jena.de

Sebastian Stober
Artificial Intelligence Lab
Otto-von-Guericke University Magdeburg
Germany
stober@ovgu.de

Paul Bodesheim
Computer Vision Group
Friedrich Schiller University Jena
Germany
paul.bodesheim@uni-jena.de

Abstract

Fine-grained visual classification (FGVC) requires distinguishing between visually similar categories through subtle, localized features — a task that remains challenging due to high intra-class variability and limited inter-class differences. Existing part-based methods often rely on complex localization networks that learn mappings from pixel to sample space, requiring a deep understanding of image content while limiting feature utility for downstream tasks. In addition, sampled points frequently suffer from high spatial redundancy, making it difficult to quantify the optimal number of required parts. Inspired by human saccadic vision, we propose a two-stage process that first extracts peripheral features (coarse view) and generates a sample map, from which fixation patches are sampled and encoded in parallel using a weight-shared encoder. We employ contextualized selective attention to weigh the impact of each fixation patch before fusing peripheral and focus representations. To prevent spatial collapse — a common issue in part-based methods — we utilize non-maximum suppression during fixation sampling to eliminate redundancy. Comprehensive evaluation on standard FGVC benchmarks (CUB-200-2011, NABirds, Food-101 and Stanford-Dogs) and challenging insect datasets (EU-Moths, Ecuador-Moths and AMI-Moths) demonstrates that our method achieves comparable performance to state-of-the-art approaches while consistently outperforming our baseline encoder.

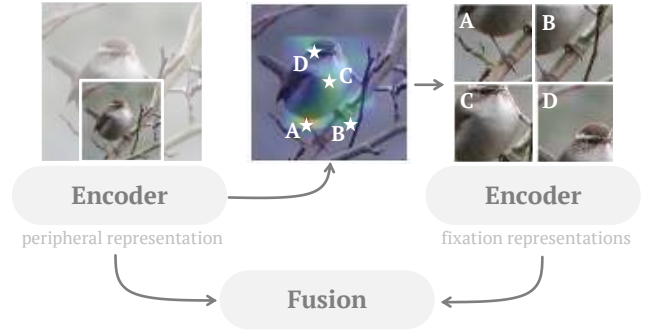


Figure 1. The *saccader* extends the usual forward prediction process by sampling fixation points from a priority map generated by the encoder, extracting fixation patches at these positions, and calling the encoder again with these refined inputs.

1. Introduction

Fine-grained visual classification (FGVC) has emerged as a critical area of research within computer vision, aiming to distinguish between visually very similar categories, such as different bird species [50, 55], dog breeds [28] or insects [26, 30, 43]. Unlike traditional image classification tasks, where inter-class differences are often pronounced, FGVC focuses on identifying subtle, local patterns and nuanced variations across categories. This level of granularity has wide-ranging applications, like biodiversity monitoring [30], where precise categorization is crucial.

Despite its significance, FGVC remains a challenging problem due to several inherent complexities. First, labeling fine-grained categories often requires domain expertise. Data is sparse, introducing a high risk of overfitting and rendering regularization essential [4, 29, 37]. Second, fine-

grained categories often exhibit high intra-class variability caused by changes in pose, lighting, occlusion, and background clutter. Lastly, inter-class differences are typically subtle, often localized to specific regions, such as the shape of a beak or the texture of a petal, making these distinctions difficult to capture without specialized feature extraction mechanisms. Models need to process high-resolution signals to extract visual details that would otherwise be lost.

Part-based and part-sampling models [22, 53] tackle these issues by extracting and encoding visual parts of the source image.¹ These methods typically employ specialized localization networks that learn mappings from pixel space to a sampling space, based on which the parts are sampled [22, 30]. This is related to object detection [15] and spatial transformers [25, 45], all suffering from the required complexity of such mapping. Learning a mapping from the high-dimensional pixel space to a sampling space requires the network to understand image content before identifying relevant parts. However, the obtained salient features are often not reused for downstream prediction; instead, new feature representations are extracted again from the sampled parts, leading to redundant computation. Furthermore, typically two more models are employed: one designed to capture contextual information from the source image and another to extract fine-grained details from individual parts [22, 30, 63].

However, real-world scenes present significant variability challenges. Target objects appear at dramatically different scales — the same bird might sit prominently in the foreground or appear barely visible in the background. Parts can be occluded to varying degrees, sometimes leaving only a single part visible. This variability forces the contextual encoder to adaptively extract part-level features anyway. Consequently, all three models learn fundamentally similar features. This leads to stacked loss functions, where each encoder optimizes its own log-likelihood [63] to converge to satisfying results.

Recently, more and more works shifted towards Vision Transformers (ViT) [12] as backbones. Here, parts are learned implicitly by a soft [53] or a hard [5] token selection. As tokens are patches from the source image, we refer to these part representations as *multi-patch parts* (MPPs) compared to *single-patch parts* (SPPs). SPPs [22, 30, 64] are restricted to rectangular form factors, while MPPs are more flexible. SPPs are usually very few (often around 4 parts [60]), MPPs are significantly more (often around 64 [53]). This comes from the use of non-maximum suppression (NMS) in SPP-based approaches. In transformers, MPPs are intended to carry information to the next layer, while SPPs are extracted once from a mature feature map.

¹Part-based methods [22] learn unambiguous semantic parts, while part-sampling approaches [53] model are less strict, focusing on salient regions. This work uses ideas from both fields.

This renders MPPs very noisy, where most of the contained parts are usually non-interpretable.

While these methods achieve promising results, they still suffer from noise in MPPs and inefficiencies in SPP extraction. To overcome these limitations, we take inspiration from human visual perception. Humans developed very efficient and effective visual perception, which relies heavily on visual search mechanisms to efficiently process and interpret complex scenes. Unlike uniform scanning, the human visual system employs a dynamic strategy of rapid eye movements, known as saccades [39, 61], interspersed with brief fixations [24]. These fixations are concentrated on the fovea, the central region of the retina that occupies only 1% of the visual field but provides the highest acuity [13, 27]. The remaining 99% comprises peripheral vision, which, despite its lower resolution, plays a crucial role in guiding attention [57]. This search process is not random but guided by various sources of pre-attentive information, which coalesce to form a spatial *priority map* [56]. This also allows capturing a virtual high-resolution image of the entire field of view by multiple low-resolution windows [2].

Our Contributions Most prior work in fine-grained visual classification (FGCV) either extracts parts from down-scaled feature maps, losing high-frequency details, or processes full-resolution images, which is memory- and compute-intensive. Inspired by human vision, we reinterpret single-patch parts (SPPs) as freely steerable fixation patches on the high-resolution image and fuse them with peripheral patches using a weight-tied multi-patch part (MPP) encoder. This allows us to encode only downsampled views while steering attention over the full high-resolution canvas, capturing fine-grained details efficiently. Our core contributions are:

- A biologically-inspired *saccader* framework that simulates peripheral vision through peripheral patch encoding and foveal attention² via fixation patch sampling.
- We bypass the need for a localisation network by sampling from an aggregated high-level feature map of the encoder (i.e., the priority map).
- We propose an effective non-maximum suppression algorithm that avoids the elimination of redundant fixation points (compared to common IoU-based approaches).
- A single-encoder architecture that processes both peripheral and fixation views, eliminating the need for separate contextual and part-based encoders, which renders the framework resolution-agnostic and a wide range of backbones to be used off-the-shelf.
- Contextualised selective attention for dynamic fixation patch weighting, enabling adaptive influence adjustment and view dropping based on relevance.

²Foveal refers to the fovea, the central retinal region responsible for sharp vision [39].

A high-level overview is illustrated in Figure 1.

2. Related Work

Existing approaches to FGVC have made considerable strides through the use of part-based feature localization. The benefits of these approaches are the interpretability and faithfulness of results, as predictions are formed based on the extracted parts. As aforementioned, we distinguish two forms of part sampling: (i) Single-Patch Part (SPP) methods, where parts are rectangular crops from a source image, and (ii) Multi-Patch Part (MPP) methods, where parts comprise multiple patches from the source image.

Extracting Single-Patch Parts Extracting patches from source images is fundamental in object detection [15], which can be approached via single-stage or two-stage methods. Two-stage detectors [14, 15, 19, 41] achieve high accuracy by processing individual region proposals but incur significant computational overhead. In contrast, single-stage detectors [34, 40] streamline the process by combining proposal and prediction in a single forward pass, greatly improving efficiency. In FGVC, similar techniques can be used to extract object parts [22]. However, due to the lack of ground-truth part annotations in FGVC datasets, part extraction is implicitly trained by conditioning class predictions on the extracted parts. Part-Stacked CNNs [22] adopt a single-stage framework to simultaneously generate region proposals and integrate their encodings for classification. A critical aspect of these methods lies in encoding the contextual information, which can be achieved using either high-resolution images, as demonstrated in [30], or downsampled images, as employed in [22]. As previously noted, the encoders in such architectures often learn highly redundant features. This redundancy is partially addressed in [60] by employing a weight-tying mechanism between the context and part encoders. To effectively fuse part representations, Xu et al. [60] used multi-scale cross-attention between parts. Alternatively, Sikdar et al. [48] proposed leveraging a graph attention network to explicitly model the spatial relationships between parts. Dongliang et al. [5] learns multi-grained parts using top-down spatial attention. Another significant challenge in these multi-step architectures concerns the gradient flow. To mitigate this issue, Zhang et al. [63] optimized log-likelihoods at the level of individual encoders, ensuring a consistent and robust learning signal is maintained throughout the pipeline.

Extracting Multi-Patch Parts When encoding an image, the obtained feature maps can be interpreted as hierarchies of part maps (potentially sparsified by thresholds). Although this can be done for ConvNets as well, it is more common for vision transformers (ViTs) [12, 49]. This might

be due to the fact that transformers selectively attend globally to relevant image regions, leading to sparse part maps. To remove background noise, improve efficiency, and increase scarcity further, token sparsification (token dropout or pruning) is usually used. This process corresponds to the localization process in SPP methods. TransFG [17] uses accumulated attention scores for token dropout. DynamicViT’s [38] dynamic token pruning creates spatially coherent token selections that inherently form part maps. A halting probability is introduced in A-ViTs [62], enabling adaptive token selection where high-probability tokens correspond to discriminative parts. In ACC-ViTs [65], the high-attention tokens from different classes are mixed, effectively isolating class-specific part regions by excluding background noise. In FAL-ViTs [23], coarse-to-fine token masking is used to create hierarchical part representations, progressing from global object structure to fine-grained details. PIMs [6] use a graph neural network to model the intra- and inter-stage relationships between the top-k tokens (preserved tokens). HERBS [7] retains only top-k spatial features at different network stages, using higher temperature distributions in early layers to encourage exploration while providing guidance.

3. Methodology

Let $(\mathbf{x}, y) \sim \mathcal{D}$ represent a sampled tuple from a dataset $\mathcal{D}$, where $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ is an image and $y \in \mathbb{N}$ is its corresponding class label. Here, $C \in \mathbb{N} \setminus \{0\}$ is the number of channels, usually $C = 3$ for RGB images, and $H, W \in \mathbb{N} \setminus \{0\}$ the spatial dimensions of the image. We work with high-resolution signals, $H, W \geq 512\text{px}$. The premise of your approach is to maintain the high-resolution signal during the forward pass to extract patches from it. Our proposed saccadic vision algorithm comprises a shared encoding step and a fusion step of peripheral and focus views. The process is illustrated in Figure 2 and discussed in the following.

3.1. Multi-Granular MPP-Encoder

Features from a downsampled version of $\mathbf{x}$, called $\mathbf{x}_{\text{per}} \in \mathbb{R}^{H' \times W' \times C}$ (short for peripheral) with $H' \ll H$ and $W' \ll W$ are extracted to provide contextual information. This is done by a backbone encoder ϕ (e.g., a Swin-Transformer [35]), which extracts from $\mathbf{x}_{\text{per}}$ a stack of feature maps. More precisely, we extract feature maps at $S \in \mathbb{N} \setminus \{0\}$ (usually $S = 4$) different encoding stages. Hence, $\phi : \mathbb{R}^{H' \times W' \times C} \mapsto \{\mathbb{R}^{H_s W_s \times C_s}\}_{s \in S}$ with $C_s, H_s, W_s \in \mathbb{N} \setminus \{0\}$ being the encoder-specific channel and spatial dimensions of the feature maps at each stage, respectively.³ To remove visual clutter, we define $\mathcal{F} := \phi(\mathbf{x}_{\text{per}})$, where

³The backbone can either be a Convolution-based or a Transformer-based encoder. For consistency, feature maps are represented as spatial token sets $HW \times C$ instead of $H \times W \times C$.

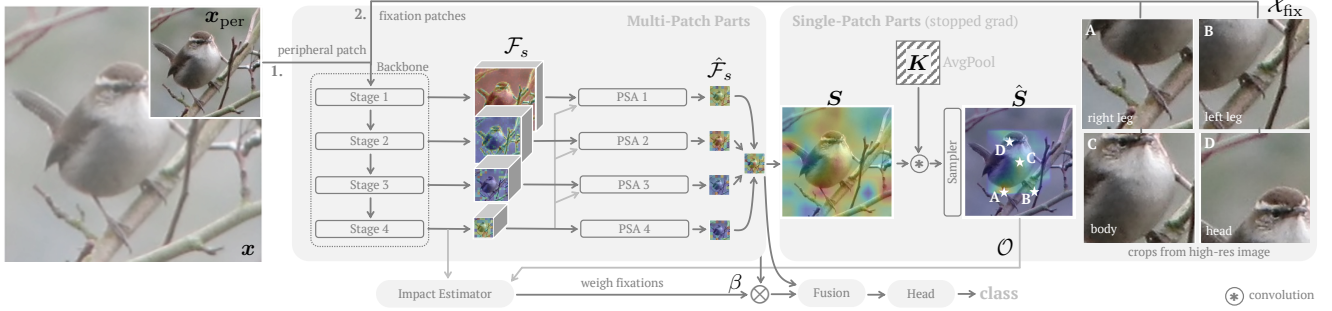


Figure 2. A multi-stage backbone is used to encode a downsampled (peripheral) view x_{per} of a query image x . At each stage s (here four stages are shown) the intermediate feature maps $\mathcal{F}_s$ are extracted and fed into part sampling attention (PSA) blocks [53]. The resulting feature maps $\hat{\mathcal{F}}_s$ are fused and upsampled to the size of x . The resulting map S is further refined to form the priority map $\hat{S}$ from which N fixation points are sampled. At these fixation points $\mathcal{O}$ fixation patches are cropped from the source image x . The resulting fixation patches are encoded again. The locations and the feature maps $\mathcal{F}_s$ are used to compute impact weights β of each fixation point. Finally, the representations are fused to obtain the logit scores used to solve FGVC downstream task.

$\mathcal{F}_s \in \mathbb{R}^{H_s W_s \times C_s}$ are the feature maps of the stage s . To refine the features in $\mathcal{F}$, we use a MPP-encoder.

In this work, we use the Multi-Granularity Part-Sampling (MPSA) mechanism proposed by Wang et al. [53], which we re-introduce in the following two paragraphs for clarity. This mechanism comprises three encoding stages for the feature maps $\mathcal{F}$ to extract MPPs.

Part Sampling Attention A spatial distribution over the feature maps is learned to down-weight background features and extract P_s MPPs mapping $\mathbb{R}^{H_s W_s \times C_s}$ to $\mathbb{R}^{P_s \times C_s}$ by

$$\mathcal{P}_s = \left[\text{softmax}_{H,W}(\sigma_s(\mathcal{F}_s) + \mathbf{B}_s) \right]^\top \mathcal{F}_s, \quad (1)$$

where $\sigma_s : \mathbb{R}^{H_s W_s \times C_s} \mapsto \mathbb{R}^{H_s W_s \times P_s}$ is a chain of LayerNorm [1], linear layer and non-linearity (we use GeLUs [20]). This function learns a compression from the feature space C_s of the backbone to a part space P_s with $P_s \ll C_s$ of the MPSA pipeline. $\mathbf{B}_s \in \mathbb{R}^{H_s W_s \times P_s}$ is a spatial bias tensor, which learns frequent position maps of the target objects. The subscripts of the softmax operator indicate the dimensions over which it is applied. This performs a spatial weighing per feature map, focusing on different spatial regions (parts). Cross-attention between the original feature maps $\mathcal{F}$ and part maps $\mathcal{P}_s$ encodes the correlation between contextual information and MPPs (by pairwise similarities). To avoid visual clutter, the Multi-Head Cross-Attention is shown here only for a single head:

$$\mathbf{A}_s = \frac{\mathbf{Q}_s \mathbf{K}_s^\top}{\sqrt{C_s}} + \bar{\mathbf{B}}_s \quad (2)$$

$$\text{and } \bar{\mathcal{F}}_s = \text{softmax}_{P_s}(\mathbf{A}_s) \mathbf{V}_s, \quad (3)$$

where the queries are the linear projections of the last feature maps $\mathbf{Q}_s := \mathcal{F}_s \mathbf{W}_s^Q \in \mathbb{R}^{C_s \times H_s W_s}$. The keys and

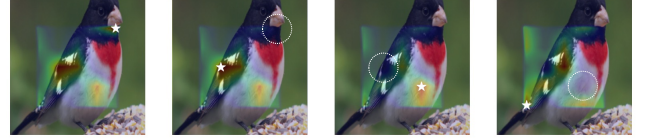


Figure 3. *Priority Progression*: Suppressing previously sampled locations reduces proximity clustering of focus points.

values are the linear projections of the part maps $\mathbf{K}_s := \mathcal{P}_s \mathbf{W}_s^K \in \mathbb{R}^{C_s \times H_s W_s}$ and $\mathbf{V}_s := \mathcal{P}_s \mathbf{W}_s^V \in \mathbb{R}^{C_s \times P_s}$, respectively. $\bar{\mathbf{B}}_s \in \mathbb{R}^{C_s \times H_s W_s \times P_s}$ is another position bias, which allows the model to decide which parts are relevant. The attention scores $\mathbf{A}_s \in \mathbb{R}^{C_s \times H_s W_s \times P_s}$ hold information about how much each part is present at each spatial location. This is used to construct a spatial scalar field $\mathbb{R}^{H_s W_s}$ by

$$S_s = \frac{1}{C_s} \sum_{C_s} \left(\text{softmax}_{H_s W_s}(\mathbf{A}_s) \right) p_s(\mathcal{P}_s), \quad (4)$$

where $p_s : \mathbb{R}^{P_s \times C_s} \rightarrow [0, 1]^{P_s}$ is a Squeeze-and-Excite [21] obtaining part-importance vectors (see [53] for details). As the name suggests, the part-importance scores how valuable each part is (learning to down-weight background parts).⁴ The attention scores are used to bias the part maps $\bar{\mathcal{F}}_s \in \mathbb{R}^{H_s W_s \times C_s}$ such that

$$\hat{\mathcal{F}}_s = \sigma_s(\bar{\mathcal{F}}_s + \alpha_s S_s), \quad (5)$$

where $\sigma_s : \mathbb{R}^{H_s W_s \times C_s} \rightarrow \mathbb{R}^{H_s W_s \times C_s}$ is a linear layer.

Multi-Granularity Fusion Each feature map $\hat{\mathcal{F}}_s$ is linearly scaled by a factor $\gamma_s \in \mathbb{R}$, where these scaling factors serve as learnable attention weights that dynamically balance the contribution of features from different scales in the

⁴This is scaled down by a fixed weight $\alpha_s = 0.1, \forall s$ for stability.

feature pyramid [33]. This allows the model to adaptively emphasize the most informative scales for a given input. The feature maps are aggregated along the stage dimension S , such that

$$\mathbf{S} = \sum_s \gamma_s \mathcal{S}_s \quad \text{and normalized} \quad \mathbf{S} \leftarrow \frac{\mathbf{S}}{\sum_{H,W} \mathbf{S}}. \quad (6)$$

This spatial pooling maps to $\mathbb{R}^{H_S W_S}$. The stack of S feature maps is concatenated along the embedding dimension and passed through a non-linearity σ (again, we use GeLU) to obtain a logit vector $\mathbf{z} \in \mathbb{R}^{\sum_s C_s \times H_S W_S}$, such that

$$\mathbf{z} = \sigma \left(\text{concat} \left(\hat{\mathcal{F}}_s, \forall s \in S \right) \right)^\top \mathbf{S}. \quad (7)$$

The concatenation operation preserves the distinct characteristics of each scale level, while the multiplication with $\mathbf{S}$ acts as a global context modulation mechanism.

3.2. SPP-Extraction and Fusion

Fixation Sampling As argued in [53], $\mathbf{S}$ is intended to have high activation at salient spatial positions. Therefore, it is reasonable to leverage $\mathbf{S}$ as a priority map to sample fixation points. $\mathbf{S}$ is upsampled by bilinear interpolation to the resolution of the source image $\mathbf{x}$, which allows for continuous fixation locations. As we use fixed-sized fixation windows⁵, we constrain the priority map to a center region, ensuring that windows do not reach out of the pixel grid of the source image. We achieve this using 2D average pooling with a $[H', W']$ pooling kernel (with stride one). We refer to the resulting priority map as $\hat{\mathbf{S}} \in [0, 1]^{H' \times W'}$. Examples are illustrated in Figure 4. Interpreting $\hat{\mathbf{S}}$ as a spatial probability distribution allows us to draw samples

$$\mathcal{O} = \left\{ \mathbf{o} + \left[\frac{H'}{2}, \frac{W'}{2} \right] \mid \mathbf{o} \sim_N \hat{\mathbf{S}} \right\}, \quad (8)$$

with $|\mathcal{O}| = N$ and the fixed bias maps location coordinates back to high-resolution coordinate space $[0, W] \times [0, H]$. This number of fixations can vary, so that we can set a specific value during training ($N_{\text{train}} \in \mathbb{N}$) and during testing ($N_{\text{test}} \in \mathbb{N}$) even varying per sample to maximize flexibility.⁶ At each fixation point, a patch is extracted, resulting in a set $\mathcal{X}_{\text{fix}} := \{\mathbf{x}_{\text{fix}}^{(1)}, \mathbf{x}_{\text{fix}}^{(2)}, \dots, \mathbf{x}_{\text{fix}}^{(N)}\}$. Because most of the sampling space has low, but non-zero probability, the distribution becomes heavy-tailed and still draws background samples. We counteract this by using a low-temperature softmax to concentrate probability on salient regions.

⁵This eases parallel processing of fixations, which otherwise would require padding or warping, which are both non-optimal w.r.t. memory and distortions, respectively.

⁶We found $N_{\text{train}} = N_{\text{test}} = 4$ to work well for the benchmarks used in this work (see Sec. 4).

Samples are drawn sequentially from the priority map $\hat{\mathbf{S}}$ resulting in a *Priority Progression* as shown in Figure 3. To prevent redundant patch sampling in nearby locations, we apply non-maximum suppression (NMS) during sampling.⁷ After sampling a location $\mathbf{o}$, we down-weight surrounding areas using a 2D multilateral Gaussian penalty kernel. The sampling algorithm is detailed in Algorithm 1.

Peripheral and Fixation Fusion We apply the same MPP-Encoder (see Sec. 3.1) to the extracted fixation patches $\mathcal{X}_{\text{fix}}$. These fixation representations are pooled and fused with the peripheral representation $\mathbf{z}$. Some target objects in the dataset might fill out most of the canvas, rendering the focus views unnecessary. To cope with these cases, we introduce a global impact factor $\alpha \in [0, 1]$ to weigh the importance of fixations in the fusion with the peripheral representation. This weight is learned from the early high-level feature maps of the backbone, such that

$$\alpha = \phi(\text{GAP}(\mathcal{F}_S)), \quad (9)$$

where GAP denotes global average pooling and $\mathcal{F}_S$ are the feature maps from the last encoding stage of the backbone. $\phi : \mathbb{R}^C \mapsto [0, 1]$ is a shallow non-linear fully-connected encoder compressing C channels to a scalar. Now, another issue is that focus views might contain views without many salient features (background crops). Hence, we introduce another impact weighing, but this time on the fixation representations. We leverage $\mathcal{F}_S$ again to parameterise a low-temperature Boltzmann distribution over the fixations,

$$\beta = \text{softmax}_{n \in N} \left[\frac{1}{\tau} (\phi(\text{GAP}(\mathbf{M}(\mathbf{o}_n) \odot \mathcal{F}_S))) \right]. \quad (10)$$

The Hadamard product is denoted by $\odot$, which is used to mask the feature maps by a location-parameterized spatial scalar field $\mathbf{M}_s(\mathbf{o}_n) \in [0, 1]^{H_S \times W_S}$. The mask is a 2D multivariate Gaussian with a fixed variance centered at $\mathbf{o}_n$. The idea is to preserve features around the fixation point while down-scaling features elsewhere. Finally, this allows us to compute the final representations by

$$\mathbf{z} = \sigma(\mathbf{z}_{\text{per}} + \alpha \mathbf{z}_{\text{fix}}) \quad \text{with} \quad \mathbf{z}_{\text{fix}} = \frac{1}{N} \sum_n \beta_n \mathbf{z}_{\text{fix}}^n, \quad (11)$$

where σ is a linear layer. This representation encodes the final logits, which are used to parameterize the downstream class distribution.

Training Procedure and Losses As shown in Figure 2, the gradients are stopped during fixation point sampling

⁷Since gradients are stopped during sampling, other non-differentiable sampling strategies could be substituted.

Algorithm 1: Fixation Sampler.

Only one forward pass is shown, while in practice a batch of sample maps are processed in parallel.

Require: $\hat{\mathbf{S}} \in \mathbb{R}^{H \times W}$, $N \in \mathbb{N}$
Ensure: Fixations $\mathcal{O} \in \mathbb{R}^{N \times 2}$

- 1: Initialize $\mathcal{C} \leftarrow \emptyset$
- 2: **for** $n = 1$ to N **do**
- 3: **Obtain Fixation Point** $\mathbf{o} \sim \hat{\mathbf{S}}$
- 4: $\mathbf{i} \sim \text{softmax}(\hat{\mathbf{S}}/\tau)$ // $\tau = 0.1$
- 5: $\mathbf{o} \leftarrow [H \bmod \mathbf{i}_0, W \bmod \mathbf{i}_1]$
- 6: **Non-Maximum Suppression**
- 7: $d \leftarrow \|\hat{\mathbf{S}} - \mathbf{o}\|_2$
- 8: $K \leftarrow \exp(-d/(2\sigma^2))$ // $\sigma = 50$
- 9: $\hat{\mathbf{S}} \leftarrow \hat{\mathbf{S}} \odot (1 - \lambda K)$ // $\lambda = 0.95$
- 10: $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathbf{o}\}$
- 11: **end for**
- 12: **return** $\mathcal{O}$

and patch extraction, but otherwise flow through the entire pipeline. During training, the vanilla negative log-likelihood (NLL) on the peripheral representation is minimized, along with a confidence-integrated NLL on the aggregated fixation representation. During training the following loss is minimized

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{NLL}}(\mathbf{z}_{\text{per}}) + \lambda_2 \mathcal{L}_{\text{Conf-NLL}}(\mathbf{z}_{\text{fix}}), \quad (12)$$

where $\lambda_1, \lambda_2 \in \mathbb{N}$.⁸ In our implementation, we used the GradCAM loss as in [53] and additional regularisation terms (we only used weight decay), but omit it here for visual clarity. Similar to the variance-weighted confidence-integrated loss [47], but using the global impact α as an explicit confidence estimate, we used

$$\mathcal{L}_{\text{Conf-NLL}} = \alpha p(\mathbf{z}_{\text{fix}}) + (1 - \alpha) p_{\mathcal{U}} + \lambda \log(\alpha). \quad (13)$$

The higher the confidence (global impact α) for an input image, the higher the influence of the parameterised distribution $p(\mathbf{z}_{\text{fix}})$. For low confidence predictions, the influence of the constant uniform distribution $p_{\mathcal{U}}$ rises, which would result in high NLL penalty. This pushes the model to increase its confidence. Additionally, $\log(\alpha)$ is minimised to amplify this behaviour.⁹

Removing Part-wise Translation Biases In essence, our saccading mechanism learns to remove the translation bias of visual parts of the target object. Instead of regressing an affine transformation matrix [25] or translation vector [16], we use $\hat{\mathbf{S}}$ to sample locations $\mathcal{O}$ which correspond to linear translation vectors. This is much easier to learn as otherwise

⁸We used $\lambda_1 = \lambda_2 = 0.5$ in our experiments.

⁹This is controlled by the hyper-parameter λ , which we set to 0.1.

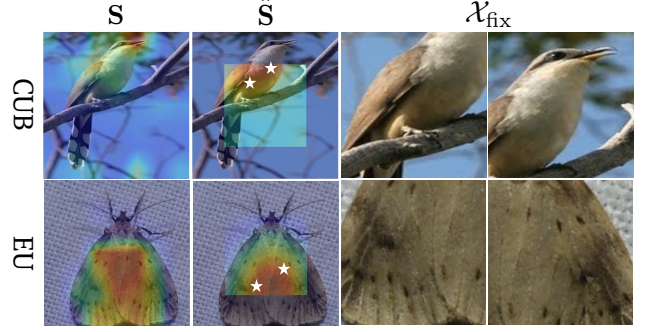


Figure 4. Examples of the raw priority maps $\mathbf{S}$, the optimized priority maps $\hat{\mathbf{S}}$ and sampled fixation patches $\mathcal{X}_{\text{fix}}$.

the regressor must learn to relate its prediction (the bias vector) to the underlying coordinate grid. Furthermore, we use the low-dimensional latent map $\hat{\mathbf{S}}$ to ease learning instead of predicting from the original pixel space [25], which is challenging and unstable. Instead of stacking multiple localization networks [32] (high memory cost) or sampling from the same map [46] (high runtime cost), we use a single network and manipulate the priority map instead (see the NMS in Algorithm 1). Our fixation points $\mathcal{O}$ are translation vectors by which the coordinate grid of $\mathbf{x}$ is shifted. At these focus locations, a window of size $[H', W']$ is sampled following [25]. This results in the following affine transformation matrix: $\begin{pmatrix} 1+(H'/H) & 0 & \mathbf{o}_0 \\ 0 & 1+(W'/W) & \mathbf{o}_1 \end{pmatrix}$.

4. Experiments

We follow the training recipe from [53] with adaptations detailed in our publicly available source code. All our models are pretrained on ImageNet-1k (INet1k) [9], while some benchmarks are pretrained on ImageNet-21k (INet21k) [42] or iNaturalist (iNat) [51]. Models are trained for 100 epochs with early stopping using mini-batches. We employ 512px source images and extract 224px peripheral and fixation patches via bilinear interpolation.

Insect FGVC Datasets We evaluate on four insect datasets with minimal background noise: EU [30] (1650 images, 200 species), ECU [43] (1445 images, 675 species) with significant center bias, and AMI-GBIF [26] subsets¹⁰ for North America (NA) (854k images, 2405 species) and Central America (CA) (71k images, 547 species) with a significantly higher variability and noise rate. Samples are shown in Figure 7. EU and ECU feature clean backgrounds ideal for evaluating fixation mechanisms without background interference.

¹⁰Due to several link corruptions in the original AMI dataset, we only included two out of three subsets. We also filtered the subsets to at least comprise 10 samples per class.

Table 1. Average Top-1 Test Accuracy on EU and CUB.

Backbone	Param.	Input	EU	CUB
ResNeXt50	63.2M	224	94.8	83.0
ViT-B16	103.6M	224	86.4	88.7
Swin-B	99.8M	224	97.7	91.8

Standard FGVC Datasets We evaluate on four established FGVC benchmarks with increasing background complexity: CUB200-2011 [55] (12k images, 200 bird species), NABirds [50] (49k images, 555 species), Stanford-Dogs [28] (21k images, 120 breeds), and Food101 [3] (101k images, 101 categories). Samples are shown in Figure 8. These datasets progress from moderate natural backgrounds to highly complex real-world imaging conditions.

4.1. Ablation Study

Plug-In Module for any Backbone We evaluate our method’s compatibility across different backbone architectures. Table 1 shows results for ResNeXt50 [59], ViT-B16 [12], and Swin-B Transformer [35]. Our encoder architecture [53] is backbone-agnostic, supporting both modern convolutional [18, 36, 59] and transformer-based architectures [12, 35, 49]. We use backbone-specific learning rates: 0.003 for ViT, 0.01 for ResNet, and 0.008 for Swin Transformer, with extended warm-up (15 epochs) for ViT compared to others (5 epochs). Given the Swin Transformer’s superior performance, we adopt it for all subsequent experiments.

Number of Fixation Points We investigate the impact of varying fixation sample numbers during training and testing on EU (Figure 5).¹¹ The optimal configuration is $N_{\text{train}} = N_{\text{test}} = 4$, aligning with findings by Xu et al. [60] for optimal part numbers in FGVC benchmarks like CUB200 [55]. Performance degrades significantly with $N_{\text{train}} = 8$ and lower N_{test} values, suggesting excessive fixations are counterproductive. We reserve detailed analysis for future work.

Runtime and Complexity Our method exhibits linear sampling complexity $\mathcal{O}(N)$ as shown in Algorithm 1, with weighted averaging (Equation (11)) scaling linearly with fixation count. However, empirical runtime measurements (Figure 6) reveal slight exponential growth with increasing N_{test} . We attribute this discrepancy to memory hierarchy effects—as N_{test} increases, working sets may exceed cache capacity, causing increased memory access latency and computational overhead from data movement and synchronization that compound nonlinearly.

¹¹With $N_{\text{train}} \geq 2$, as otherwise Equation (11) collapses.

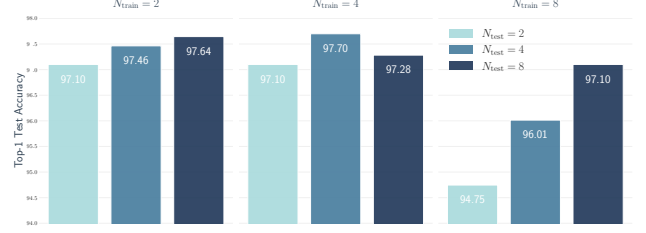


Figure 5. Average Top-1 Test Accuracy on EU with different number of fixation patches during training and testing.

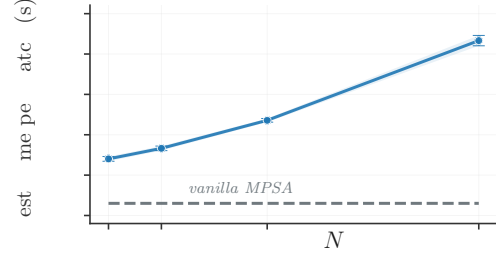


Figure 6. Average walltime in seconds (with confidence interval) over 32 test batches (EU) on an NVIDIA A40.

4.2. FGVC Benchmarks

Insect Benchmarks We evaluate our method on all four insect benchmarks using learning rate 0.008, 5-epoch warm-up, 0.001 weight decay, 10% label smoothing, 60% part drop, 20% classification head dropout, 10% attention dropout, and 32 attention heads. For datasets with significant center bias (ECU), we disable position bias $\mathbf{B}_s$ in Equation (1). To prevent overfitting, we apply random affine transformations and AutoAugment [8] beyond standard pre-processing [53]. As shown in Tab. 2, our method outperforms baselines on three of four benchmarks and achieves comparable performance on the fourth. The performance gains are most pronounced on clean background datasets (EU and ECU), where fixation sampling can operate without background interference. On the more variable AMI subsets with higher noise levels, our saccadic approach maintains competitive performance but shows less consistent improvements due to the increased complexity of distinguishing relevant fixation targets from background distractors.

Standard Benchmarks We compare against vanilla MPSA on four challenging FGVC datasets: CUB200-2011 [55], NABirds [50], Stanford-Dogs [28], and Food101 [3]. We use the training protocols provided by Wang et al. [53]. These datasets provide comprehensive evaluation across diverse recognition scenarios with varying background complexity and real-world noise. Table 3 compares our model against state-of-the-art baselines, noting that these leverage

Table 2. Average top-1 test accuracy on FGVC insect benchmarks.

Method	Backbone	Pretrain	Input	ECU	EU	NA	CA
SPPs-Cond. [31]	Inc.V3	INet1k	299	-	91.50	-	-
SPPs-Cond. [31]	Inc.V3	iNat	299	-	93.13	-	-
-	Swin-B	INet1k	224	76.19	97.06	96.32	98.34
MPSA	Swin-B	INet1k	224	52.08	96.51	98.02	98.69
Saccadic MPSA	Swin-B	INet1k	224	76.93	97.70	97.89	99.04

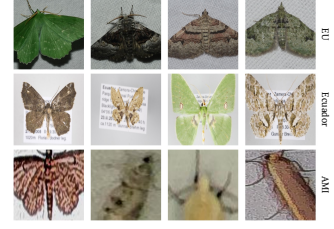


Figure 7. Data samples.

Table 3. Average top-1 test accuracy on standard FGVC benchmarks.

Method	Backbone	Pretrain	Input	CUB	NABirds	Dogs	Food
ACC-ViT [65]	ViT-B16	INet1k	448	91.8	91.4	92.9	-
FAL-ViT [23]	ViT-B16	INet21k	448	91.7	91.1	91.1	91.8
TransFG [17]	ViT-B16	INet21k	448	91.7	90.8	92.3	-
MPSA [53]	Swin-B	INet1k	384	92.8	92.5	95.4	-
FIDO [11]	Inc.V3	iNat	299	90.9	89.3	75.7	-
MPSA [53]	Swin-B	INet1k	224	91.6	90.2	94.1	93.7
Saccadic MPSA	Swin-B	INet1k	224	91.8	90.8	94.5	94.0

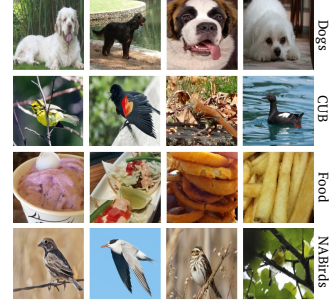


Figure 8. Data samples.

various backbones and input sizes that impact performance. Our method achieves the best performance among all 224px input methods, though higher-resolution baselines maintain advantages due to increased input detail. Our saccadic vision extension consistently improved the performance of the vanilla MPSA across all benchmarks.

Train- and Test-Time Augmentation Our dynamic fixation sampling creates natural augmentation during training as priority maps evolve, exposing the model to new fine-grained perspectives that regularize learning. This extends beyond conventional data augmentation (DA) and test-time adaptation (TTA) [52, 54, 58] through contextual weighting (Equation (11)) rather than simple signal averaging with random sampling. Table 4 demonstrates that our contextual sampling and pooling strategy outperforms both individual and combined traditional augmentation approaches, confirming the benefit of our principled attention mechanism over random patch selection.

5. Conclusion

We introduce the *saccader* framework, a biologically-inspired approach to fine-grained visual classification (FGVC) that mimics human saccadic vision through peripheral-foveal attention. Our method eliminates complex localization networks and reduces redundancy via a

Table 4. Comparison of average top-1 test accuracies on EU between our Saccader, (training-time) data augmentation (DA), test-time adaption (TTA), and a combination of both. Both DA and TTA commonly use random sampling of target object patches and signal averaging.

Method	Sampler	Pool	N_{train}	N_{test}	Acc
Vanilla	-	-	0	0	96.0
DA	Random	Avg	4	0	97.1 (+1.1)
TTA	Random	Avg	0	4	96.0 (+0.0)
DA+TTA	Random	Avg	4	4	96.4 (+0.4)
Saccader	Algorithm 1		4	4	97.7 (+1.7)

single weight-shared encoder. The saccader framework offers multiple key advantages: (1) sampling fixation points directly from high-level features bypasses localization network overhead, (2) our non-maximum suppression algorithm prevents spatial collapse while maintaining patch diversity, and (3) the weight-shared architecture reduces parameters while enabling resolution-agnostic processing across backbone architectures. Evaluation across eight FGVC benchmarks shows consistent improvements over baselines, with particularly strong performance on insect classification.

Limitations and Future Work While our saccader framework demonstrates promising results, several limitations merit further investigation. Our current implementation employs fixed scale factors in the affine transformation, constraining the model’s capacity for adaptive magnification across regions of varying significance. Future research should investigate dynamic scale regression mechanisms conditioned on input characteristics. Incorporating affine adaptations of extracted fixation patches, such as rotation canonicalization [44], presents compelling opportunities for enhancement. Although designed as a model-agnostic framework, our evaluation predominantly examines MPSA-based [53] MPP-encoders. The weight-shared encoder architecture generates priority maps for fixation sampling, establishing a foundation for iterative location refinement. Information-theoretic view selection principles [10] can be used to model sequential saccadic processes, where initial fixations guide subsequent sampling. The saccadic vision framework represents a promising direction to enhance contemporary computer vision architectures.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, 2016. 4
- [2] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 1991. 2
- [3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014. 7
- [4] Julia Böhlke, Dimitri Korsch, Paul Bodesheim, and Joachim Denzler. Lightweight filtering of noisy web data: Augmenting fine-grained datasets with selected internet images. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021. 1
- [5] Dongliang Chang, Yixiao Zheng, Zhanyu Ma, Ruoyi Du, and Kongming Liang. Fine-grained visual classification via simultaneously learning of multi-regional multi-grained features. In *ArXiv*, 2021. 2, 3
- [6] Po-Yung Chou, Chu-Hsing Lin, and Wen chung Kao. A novel plug-in module for fine-grained visual classification. *ArXiv*, 2022. 3
- [7] Po-Yung Chou, Yu-Yung Kao, and Cheng-Hung Lin. Fine-grained visual classification with high-temperature refinement and background suppression. *ArXiv*, 2023. 3
- [8] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6
- [10] Joachim Denzler and C. Brown. Information-theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 9
- [11] Joachim Denzler Dimitri Korsch, Maha Shadaydeh. Simplified concrete dropout - improving the generation of attribution masks for fine-grained classification. In *International Journal of Computer Vision*, 2025. 8
- [12] 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. *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 7
- [13] Lex Fridman, Benedikt Jenik, Shaiyan Keshvari, Bryan Reimer, Christoph Zetzsche, and Ruth Rosenholtz. Sideeye: A generative neural network based simulator of human peripheral vision. *ArXiv*, 2017. 2
- [14] Ross Girshick. Fast r-cnn. *International Conference on Computer Vision (ICCV)*, 2015. 3
- [15] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 3
- [16] Ethan William Albert Harris, Mahesan Niranjan, and Jonathon Hare. Foveated convolutions: improving spatial transformer networks by modelling the retina. In *NeurIPS Workshop on Shared Visual Representations in Human and Machine Intelligence*, 2019. 6
- [17] Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Loddon Yuille. Transfg: A transformer architecture for fine-grained recognition. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021. 3, 8
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 7
- [19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017. 3
- [20] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *ArXiv*, 2023. 4
- [21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [22] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3
- [23] Yueting Huang, Zhenzhe Hechen, Mingliang Zhou, Zhengguo Li, and Sam Kwong. An attention-locating algorithm for eliminating background effects in fine-grained visual classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 3, 8
- [24] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2001. 2

- [25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. 2015. [2](#), [6](#)
- [26] Aditya Jain, Fagner Cunha, Michael James Bunsen, Juan Sebastián Cañas, Léonard Pasi, Nathan Pinoy, Flemming Helsing, JoAnne Russo, Marc Botham, Michael Sabourin, Jonathan Fréchet, Alexandre Anctil, Yacksecari Lopez, Eduardo Navarro, Filonila Perez Pimentel, Ana Cecilia Zamora, José Alejandro Ramirez Silva, Jonathan Gagnon, Tom August, Kim Bjerger, Alba Gomez Segura, Marc Bélisle, Yves Basset, Kent P. McFarland, David Roy, Toke Thomas Høye, Maxim Larrivée, and David Rolnick. Insect identification in the wild: The ami dataset. In *European Conference on Computer Vision (ECCV)*, 2024. [1](#), [6](#)
- [27] V. Javier Traver and Alexandre Bernardino. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 2010. [2](#)
- [28] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011. [1](#), [7](#)
- [29] Sungnyun Kim, Sangmin Bae, and Se-Young Yun. Core-set sampling from open-set for fine-grained self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [1](#)
- [30] Dimitri Korsch, Paul Bodesheim, and Joachim Denzler. Deep learning pipeline for automated visual moth monitoring: Insect localization and species classification. *INFORMATIK 2021*, 2021. [1](#), [2](#), [3](#), [6](#)
- [31] Dimitri Korsch, Paul Bodesheim, Gunnar Brehm, and Joachim Denzler. Automated visual monitoring of nocturnal insects with light-based camera traps. In *CVPR Workshop on Fine-grained Visual Classification*, 2022. [8](#)
- [32] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2252–2260, 2017. [6](#)
- [33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. [5](#)
- [34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016. [3](#)
- [35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021. [3](#), [7](#)
- [36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [7](#)
- [37] Eyal Michaeli and Ohad Fried. Advancing fine-grained classification by structure and subject preserving augmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2025. [1](#)
- [38] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [3](#)
- [39] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 1998. [2](#)
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [3](#)
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. [3](#)
- [42] Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik. Imagenet-21k pretraining for the masses. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. [6](#)
- [43] Erik Rodner, Marcel Simon, Gunnar Brehm, Stephanie Pietsch, Johann Wolfgang Wägele, and Joachim Denzler. Fine-grained recognition datasets for biodiversity analysis. *ArXiv*, 2015. [1](#), [6](#)
- [44] Johann Schmidt and Sebastian Stober. Learning continuous rotation canonicalization with radial beam sampling. *ArXiv*, 2023. [9](#)
- [45] Johann Schmidt and Sebastian Stober. Geometrically constrained and token-based probabilistic spatial transformers. 2025. [2](#)
- [46] Pola Schwöbel, Frederik Rahbæk Warburg, Martin Jørgensen, Kristoffer Hougaard Madsen, and Søren Hauberg. Probabilistic spatial transformer networks. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022. [6](#)
- [47] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [6](#)
- [48] Arindam Sikdar, Yonghuai Liu, Siddhardha Kedarisetty, Yitian Zhao, Amr Ahmed, and Ardhendu Behera. Interweaving insights: High-order feature interaction for fine-grained visual recognition. *International Journal on Computer Vision*, 2024. [3](#)
- [49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers: distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021. [3](#), [7](#)
- [50] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [1](#), [7](#)
- [51] Grant Van Horn, Oisín Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and

- Serge Belongie. The inaturalist species classification and detection dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6
- [52] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021. 8
- [53] Jiahui Wang, Qin Xu, Bo Jiang, Bin Luo, and Jinhui Tang. Multi-granularity part sampling attention for fine-grained visual classification. *IEEE Transactions on Image Processing*, 2024. 2, 4, 5, 6, 7, 8, 9
- [54] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8
- [55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical report, 2010. 1, 7
- [56] Jeremy Wolfe. Guided search 6.0: An updated model of visual search. *Psychonomic bulletin and review*, 28, 2021. 2
- [57] Jeremy M. Wolfe and Todd Steven Horowitz. Five factors that guide attention in visual search. *Nature Human Behaviour*, 2017. 2
- [58] Yanan Wu, Zhixiang Chi, Yang Wang, Konstantinos N Plataniotis, and Songhe Feng. Test-time domain adaptation by learning domain-aware batch normalization. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2024. 8
- [59] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 7
- [60] Qin Xu, Sitong Li, Jiahui Wang, Bo Jiang, and Jinhui Tang. Context-semantic quality awareness network for fine-grained visual categorization. *ArXiv*, 2024. 2, 3, 7
- [61] A. L. Yarbus. *Eye Movements and Vision*. 1967. 2
- [62] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [63] Fan Zhang, Meng Li, Guisheng Zhai, and Yizhao Liu. Multi-branch and multi-scale attention learning for fine-grained visual categorization. In *International Conference on Multi-Media Modeling (MMM)*, 2021. 2, 3
- [64] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *International Conference on Computer Vision (ICCV)*, 2021. 2
- [65] Zi-Chao Zhang, Zhen-Duo Chen, Yongxin Wang, Xin Luo, and Xin-Shun Xu. A vision transformer for fine-grained classification by reducing noise and enhancing discriminative information. *Pattern Recognition*, 2024. 3, 8