Fish-Vista: A Multi-Purpose Dataset for Understanding & Identification of Traits from Images

Supplementary Material

Appendix

A. Code and Dataset

We provide the dataset, along with instructions to download the images and metadata files in the following public HuggingFace repository: https://huggingface.co/datasets/imageomics/fish-vista. All necessary code is provided in the following GitHub repository: https://github.com/Imageomics/Fish-Vista

B. Licensing Information

The source images in our dataset come with various licenses, mostly within the Creative Commons family. We provide license and citation information, including the source institution for each image, in our metadata CSV files, which is available in our HuggingFace repository. Additionally, we attribute each image to the original FishAIR URL from which it was downloaded.

A small subset of our images (approximately 1k) from IDigBio are licensed under CC-BY-ND, which prohibits us from distributing processed versions of these images. Therefore, we do not publish these $\approx 1,000$ images in the repository. Instead, we provide the URLs for downloading the original images and a processing script that can be applied to obtain the processed versions we use. Detailed instructions are provided in the HuggingFace repository.

Our dataset is licensed under CC-BY-NC 4.0. However, as mentioned earlier, individual images within our dataset have different licenses, which are specified in our metadata CSV files. We provide the licenses of the original sources so that anyone using our dataset can adhere to the licensing requirements of the individual images.

C. Further Details of Data Processing Pipeline

In this section, we provide further details of the data processing pipeline that we use to obtain the images in Fish-Vista.

C.1. Examples of Raw Museum Images

As mentioned in Section 3.3, the raw images obtained from the FishAIR repository exhibit a range of noisy artifacts. We observe that images of museum specimens predominantly include rulers and tags (Figure 8, Raw Image). Some images contain radiographic images (Figure 9, first row), while others include hand-written notes with no fish images (Figure 9, second row).

C.2. Quality Metadata-based Filtering (Step 2)

In this step, we leverage quality metadata provided in CSV files by Fish-AIR, containing manually annotated information about the image quality of museum fish specimens. The metadata include the following fields (among others):

- *allPartsVisible*: A boolean variable indicating whether all parts of the fish specimen are visible or not.
- *partsMissing*: A boolean indicating whether any parts of the specimen are missing or not.
- *specimenView*: A categorical variable specifying the view of the specimen (e.g., 'top view', 'bottom view', 'side view', 'complicated view').

At the time of this study, quality metadata were available for 29,075 GLIN images and 1,435 iDigBio images used in our dataset. No quality metadata were available for the Morphbank images.

We filtered out images based on the following criteria:

1. Images where *allPartsVisible* is *False* (see Figure 7, top row).
2. Images where *partsMissing* is *True* (see Figure 7, middle row). For example, the first image shows a specimen missing its head, and the second image is missing its tail.
3. Images labeled with a *specimenView* of 'complicated view'. Manual inspection revealed that these images do not adequately display the visual traits that we need to analyze (see Figure 7, bottom row).

As a result of this filtering process, we discarded 4,467 images from GLIN and 301 images from iDigBio.

C.3. Detecting and Cropping Fish Bounding Boxes (Step 4)

We use Grounding DINO to detect and extract tight bounding boxes around fish specimens in the images. This step ensures that images without any fish specimens are excluded. Additionally, this process removes undesired visual elements, such as rulers and tags, which could otherwise introduce noise and detract machine learning models from focusing on the visual traits of the specimens. Additionally, museum images often feature multiple fishes within a single frame. To facilitate the study of visual traits, it is essential to ensure that each image contains only a single fish specimen. By detecting and cropping individual fishes, we achieve this objective, resulting in a dataset of individual fish images.

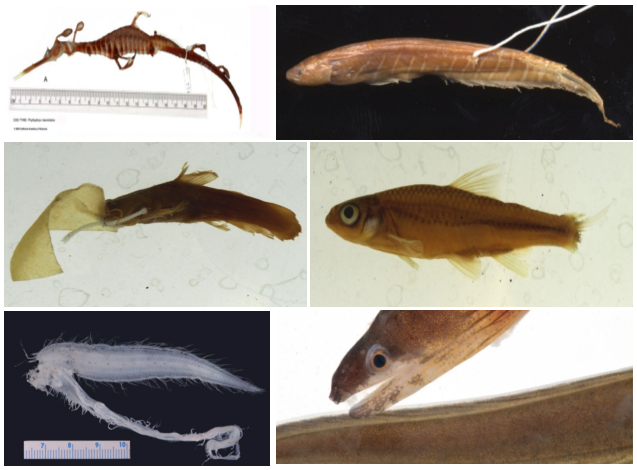

Figure 7. Examples of images filtered during the metadata filtering step of the processing pipeline.

Grounding DINO implementation details: Grounding DINO uses a textual prompt to detect bounding boxes in an image. For our use case, we find that using the prompt "fish" results in good detection of fish specimens from museum images. A *box_threshold* of 0.4 is set for initial detection, but only bounding boxes with a confidence score of 0.5 or higher are retained. We avoid setting a higher confidence threshold to minimize exclusion of valid fish images.

How good is Grounding DINO on detecting fish from museum images? In order to validate the use of Grounding DINO, we manually inspected ≈ 500 randomly chosen images and observed no inaccuracies in bounding box detection. We show some examples in Figure 8, second column. For quantitative evaluation, we obtained 311 GLIN museum fish images from [13], which contains manually annotated bounding boxes of fishes. We obtained an mIOU of 90.1%, which shows that our bounding boxes are tight.

Following the detection process, we discard 2,062 images where no fish specimens were detected. Figure 9 shows a few examples of the discarded images. Since individual images may contain multiple fishes, our cropping approach results in the addition of 12,320 bounding boxes to the dataset, corresponding to individual fishes. To maintain a minimum quality standard, we further filter out bounding boxes with height and width smaller than 224 pixels, ensuring that very low-resolution images are excluded from our dataset. This step results in the removal of 422 bounding boxes from the dataset.

C.4. Removing Background using SAM (Step 5)

Why do we remove backgrounds? Museum collection images often feature artificial backgrounds, which can introduce unintended biases into trained models. For instance, if a particular species is consistently associated with a black

Figure 8. Examples of raw images from Fish-AIR (first column), crops generated by Grounding DINO (middle column) and background removed images by SAM (last column)

background, while other species lack such backgrounds, the classifier may learn to distinguish backgrounds rather than focusing on the visual traits of the specimens. To mitigate this bias and create a controlled experimental environment, we remove backgrounds from all images.

We use the Segment Anything Model (SAM) to remove backgrounds from the fish images. Specifically, we use the bounding boxes from Step 4 (Grounding DINO) as prompts to SAM to detect foregrounds from images. We then replace the backgrounds with white color, while cropping into the segmented fish. We use default hyperparameters suggested in the SAM repository and the ViT-H SAM model.

Background removal using SAM also serves as a filtering step in our pipeline, operating in two key ways. First, SAM may detect no foreground in an image. In such cases, we discard those images. Second, we apply an explicit filtering condition: if SAM detects more than one foreground component, the image is discarded. This strict condition addresses two issues. Multiple detected components may indicate the presence of undesired elements, such as rulers or

Figure 9. Examples of noisy images that are discarded by the fish detection step of Grounding DINO.

tags, that are either on or in close proximity to the fish body, and therefore falls within the bounding box. Alternatively, it may indicate that the fish specimen is broken or disconnected, suggesting deformation that we aim to exclude from the dataset.

This filtering step removes approximately 12,000 images, resulting in a final dataset of 100,300 images spanning 10,681 species.

How good is SAM on segmenting whole fishes and the background? In order to validate the use of SAM, we manually inspected ≈ 500 randomly chosen images and observed no inaccuracies in the background removal. We show some examples in Figure 8, third column. For quantitative evaluation, we utilized manual segmentation annotations for 492 whole fish images sourced from GLIN, provided by [13]. Using these annotations as ground truth, SAM achieved an mIoU of 90.8%, demonstrating its capability to accurately segment fish specimens. These results confirm the suitability of SAM for background removal in our dataset.

D. Manual Filtering of Species

After completing the data processing steps detailed in Section 3.3 and Appendix C, we obtain complete fish images free from noisy artifacts and with uniform backgrounds. However, further filtering is required to remove images that may not adequately exhibit visual traits. This issue can arise for several reasons, including when fish specimens are photographed in views where traits are obscured or when specimens are deformed due to prolonged preservation in museum conditions.

To address this, we perform a manual inspection of the remaining images. Our inspection follows a rule-of-thumb: an image is deemed low quality if any of the key traits, such as the eye, tail, or head, are not visible, or if fewer than two fins are visible. Examples of filtered images from this process are shown in Figure 10. These examples clearly demonstrate the absence of visual traits, justifying their removal to maintain dataset quality.

Given the labor-intensive nature of manual inspection, filtering every image in the dataset is infeasible. Instead, we randomly sample 15% of images per species for manual inspection. If more than half of the sampled images for a species meet the criteria for being filtered out, we infer that most images of that species are of low quality and discard the entire species from the dataset. We chose 15% to match the proportion that would later be allocated for test sets. We set a minimum threshold of 7 images per species for discard decisions, to ensure we observe adequate samples from very small classes (species with images ≤ 50). This means, we look at increasingly higher proportions of images for very minority classes. This is because noise in these classes can have greater adverse effects on training and evaluation.

We discover that this approach only discards species with fewer images per species, which are more prone to containing a significant proportion of low-quality images. In total, we discard 420 species, comprising a total of 4,886 images, during this manual filtering step. Also, this filtering is applied only to obtain the classification and identification datasets but not the segmentation dataset, as the segmentation dataset is entirely manually annotated.

Why does filtering entire species make sense for the classification and identification datasets? For species classification, retaining species with predominantly low-quality images would not be ideal, since we would be training and evaluating models on noisy images. Recall that to create the identification dataset, images are mapped to species-level labels. Species with mostly poor-quality images lacking visible traits would therefore result in invalid mappings – that is, we would be mapping presence of traits to images which do not exhibit any visual trait. Species-level filtering ensures our classification and identification dataset remains focused on high-quality data for these tasks.

While some low-quality images may still remain in the dataset, we expect that the majority of images in our dataset are of good quality that can be used for training machine learning models. To guarantee clean evaluation, every image in the test sets are manually inspected. Noisy images

Figure 10. Examples of images that do not demonstrate visual traits. We consider such images to be of bad quality during our manual inspection and filtering.

are discarded during this process, as detailed in Appendix F, ensuring that the test sets remain free of noisy images and do not negatively impact the evaluation of model performance.

E. Manual Annotation for Segmentation

12 annotators used the Computer Vision Annotation Tool (CVAT) [2] to annotate nine traits in a subset of processed images: eye, head, barbel, dorsal fin, adipose fin, pectoral fin, pelvic fin, anal fin, and caudal fin. We provide additional examples of the annotations in Figure 11. These traits were chosen due to their well-defined physical boundaries which can be accurately segmented using CVAT.

We prioritized images containing specimens oriented in lateral view over specimens in top or bottom-view for consistency and to maximize the visibility of traits. Images of damaged or degraded specimens were excluded (similar to those shown in Figure 10), as were images with poor resolution. We also omitted images of specimens that are difficult to photograph in standard lateral view, such as elongated species or those prone to curling when preserved (e.g., eels).

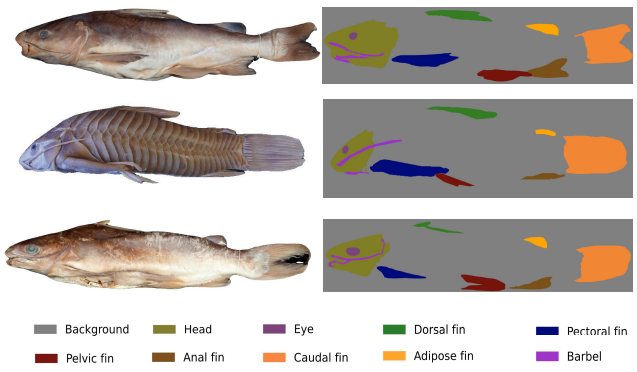

Figure 11. Examples of annotated visual trait segmentations for the nine different traits from the Segmentation dataset.

F. Additional Dataset Details

F.1. Classification Dataset

To create the classification dataset, we further filter the dataset by retaining only species with at least 4 images per species for the classification dataset. This ensures that the classification dataset includes a minimum of 2 images for training, 1 for testing, and 1 for validation for every species. Following this step, and the manual test-set filtering step described below, the final classification dataset consists of 56,360 images spanning 1,758 species.

In order to create the train, test, and validation split, we perform a stratified split of 75%, 15% and 10% respectively. We set a minimum threshold of 1 image in the test set and 1 image in the validation set for cases where the splitting would result in no images being set out for the test and validation splits. This results in a training set of 39,800 images, a validation set of 6,779 images and an *initial test set* of 10,830 images.

Manual filtering of the initial test set: In order to ensure that the models are tested on a perfectly clean dataset, we manually inspect every image in the initial test set to obtain our final test set for classification. We follow the same manual inspection guidelines as discussed in Appendix D. We discard 1,049 images that do not have traits visible, either owing to deformity, or because of the view of the image. This is less than 10% of the initial test set images, which demonstrates the effectiveness of our prior data processing steps. We obtain a *final test set* of 9,781 images.

Statistics across the four categorizations of species: majority, neutral, minority and ultra-rare, along with an overview of the long-tailed distribution in Figure 3 - Species Classification.

F.2. Identification Dataset

The trait identification dataset is designed to achieve three key objectives: detecting the presence or absence of four traits—adipose fin, pelvic fin, barbel, and multiple dorsal fins; evaluating model performance on out-of-distribution (OOD) test sets; and assessing whether traits can be localized in images using coarse-grained weak labels by predicting their presence/absence. The initial dataset, consisting of 52,982 images from 682 species, is divided into four splits: training, validation, in-species test, and leave-out-species test sets. We reserve 51 species (1,935 images) for the leave-out test set, ensuring sufficient variation in the presence and absence of all four traits for robust OOD evaluation. The remaining images are split into training (75%), in-species test (15%), and validation (10%) sets stratified by the unique combination of the four traits. The training set comprises 38,038 images from 628 species, the validation set includes 5,238 images from 451 species, and the in-species test set contains 7,771 images from 450 species,

all of which overlap with the species in the training set to provide an in-distribution evaluation set. To ensure high-quality evaluation, we manually inspect the test sets to remove noisy or low-quality samples, following the same process used for the classification dataset.

We construct a manual-annotation dataset comprising presence/absence annotations for four traits across 1,281 images spanning 1,075 species. This dataset is derived from a subset of the manually annotated segmentation dataset, carefully curated to ensure no overlap with the other four identification datasets and no species overlap with the training set. For this dataset, the presence/absence of traits such as the adipose fin, pelvic fin, and barbel is straightforward to infer from segmentation annotations, as the presence of corresponding pixel labels directly indicates the traits' presence. However, this approach cannot be applied to images with multiple dorsal fins, as all dorsal fins share the same pixel label in the segmentation annotations. To address this, we used help from expert biologists in our team to manually inspect these images and annotate the presence/absence of multiple dorsal fins.

Incorporating the *manual-annotation* dataset brings the total identification dataset to 54,263 images spanning 1,754 species. The key statistics of the identification dataset are illustrated in Figure 3 - Trait Identification. In the figure, the presence percentage for each trait represents the proportion of images in which the trait is present, highlighting the highly imbalanced distribution of each trait in our dataset.

Remark: Note that the number of species in each data split does not necessarily sum up to the total number of species in FV-Id. This is because there would be overlapping species across the training, validation, and in-species test sets. While the *leave-out-species* and *manual-annotation* sets are constructed to have no species overlap with the training set, they still share some species with the validation set. This means that a subset of the species in the validation set are not in the training set. This setup enables hyperparameter tuning and model selection based on performance on both seen and unseen species in the validation set.

F.3. Segmentation Dataset

	Head	Eye	Dorsal	Pectoral	Pelvic	Anal	Caudal	Adipose	Barbel
Presence (%)	100	100	100	100	95.91	100	100	10.67	7.58
Mean Area (%)	6.97	0.72	4.57	2.43	0.96	2.47	5.04	0.38	0.42

Table 5. The proportion of images where each of the nine traits are present, and the average area they occupy per image

For the segmentation dataset, we create a split of *70%-25%-5%* split for train-test-validation sets respectively, stratified by the unique combination of the nine traits. We obtain a training set of 4,312 images, test set of 1,504 images and validation set of 316 images. Key statistics are shown in Figure 3. In the figure, the 'Presence in Images (%)' indicates the proportion of images for which a trait is present, and the 'Mean Area (%)' indicates the average proportion of pixels that a trait covers. The complete table of Presence (%) and Mean Area (%) for all 9 traits are provided in Table 5. We can see the high imbalance associated with the trait presences, as well as the very small area covered by some of the traits, particularly the eye, barbel and the adipose fin.

G. Experiment Details

We provide implementation details, including training details, for all our experiments in this section. Note that, we report results for all trained models on the best validation checkpoints.

G.1. Classification Experiments

Hyperparameters: For all the CNN-based backbone models reported in Table 2, we use hyperparameters following suggestions of training routines for imbalanced image datasets provided in [7] and [16]. We use the SGD optimizer with a base learning rate of 0.1, with a linear warmup of the learning rate for 5 epochs. We also employ cosine annealing decay for the learning rate. We use a weight decay parameter of 2e-4. We train all CNN-based models for 100 epochs, since we observe that these models converge well within this limit. We employ early stopping with a patience of 10 epochs that goes into effect after training for the first 50 epochs.

For all the transformer-based (ViT) backbone models, we use hyper-parameter suggestions from [18]. We use Adam optimizer with base learning rate of 3e-4, and a linear warm-up of the learning rate for the first 50 epochs. We set the weight decay parameter to be 0.1. We train the transformer-based models for 150 epochs, since we observe that these models take longer to converge than the CNN-based models. For all classification experiments, we use a batch size of 128. We also employ cosine annealing decay for the learning rate. All of our models are pretrained on the ImageNet-1k [3] dataset, unless explicitly tagged with *22k*, in which case we start with ImageNet-22k [3] pretrained weights. We employ early stopping with a patience of 10 epochs that goes into effect after training for the first 100 epochs.

Image augmentations: For species classification and trait identification, it is essential to maintain the aspect ratio of the various parts of the input images. Therefore, we pad all images along the shorter edge to make both sides of the image the same length (square padding). We then resize the image to the required resolution while maintaining the aspect-ratio according to the model that we use. We resize the image to the resolution expected by each model, which is 224×224 pixels in most cases. We calculate the

mean and standard deviation of our training set and normalize every input image accordingly. During training, we randomly augment the images with the following operations: rotations between 0 and 180 degrees, adjusting the sharpness, changing the contrast, and performing horizontal and vertical flips.

Loss function: For CNN models, ViT models and linear probing of foundation models, we use the standard cross entropy (CE) loss with the objective of predicting the correct class through empirical risk minimization.

G.1.1. Details of Zero-Shot Classification Experiments

For CLIP and Bio-CLIP Zero-Shot (ZS), we use textual prompt ensembling using the same set of 80 prompts provided by OpenAI in the original CLIP paper [14].

G.1.2. Details of Fine-grained Categorization Methods

We experiment using two FGVC methods – INTR and TransFG (see Table 2). We use their default settings of hyper-parameters and follow the implementations as provided in the original repositories.

G.1.3. Details of Imbalanced Methods

Class-balanced re-weighting (CB-RW): CB-RW [1] is a re-weighting strategy that assigns weights to each class based on the inverse of the effective number of samples in the class. The effective number is defined as a function of the number of samples in class k, denoted as N_k, and a hyperparameter β. The weight for class k is given by:

$$w_k = \frac{1 - \beta^{N_k}}{1 - \beta} \tag{1}$$

For our experiments, we set $\beta = 0.9999$. This weighting scheme ensures that underrepresented classes receive higher weights, addressing the class imbalance problem.

Focal Loss: Focal loss [8] down-weights the loss for well-classified examples, thus reducing their impact during training. By applying a modulation factor to the standard cross-entropy loss, focal loss ensures the model concentrates on difficult, underrepresented classes. In our implementation, we use $\gamma = 2$ for the loss modulation factor.

G.2. Identification Experiments

Model Details of Figure 6: EffNetV2 refers to EfficientNet-v2 [20], CNext-B refers to ConvNext-Base [11], Swin-B refers to Swin-Base [10], MaxViT refers to MaxViT-T [22]; Q2L-R34 refers to Query2Label [9] with Resnet-34 backbone, trained with a single attention head (SH); Q2L-Swin refers to Query2Label with Swin Transformer backbone, trained with 4 attention heads (that is, multiple heads or MH).

Hyperparameters: For all the backbone models used in trait identification, we use the Adam optimizer with weight decay of 0.1. We train every model for 150 epochs.

Model	F1	Major Acc.	Neutral Acc.	Minor Acc.	Ultra-R Acc.
VGG-19[17]	49.7	93.5	83.0	74.2	45.9
Resnet-34 [5]	35.6	89.9	68.4	60.9	30.7
Inception-v3 [19]	40.2	90.0	77.7	67.7	34.5
ResNext-50 [25]	44.4	91.4	78.3	69.8	39.1
MobileNet-v3 [6]	40.1	86.0	74.4	65.5	34.0
RegNet-y [15]	43.7	89.8	77.4	68.5	38.5
EfficientNet-v2 [20]	34.3	89.0	75.0	62.3	28.5
ConvNext-B [11]	49.5	89.6	81.8	73.1	44.9
ViT-B-16 [4]	48.3	88.7	82.3	73.3	43.4
ViT-B-32 [4]	45.2	86.9	75.8	66.6	41.8
DEiT-distilled-s [21]	46.2	91.7	76.8	72.3	40.8
Swin-B-22k [10]	55.1	92.6	86.2	79.6	50.4
CVT-13 [24]	49.3	92.0	83.3	73.5	44.7
MobileViT-xs [12]	49.0	92.2	85.9	74.1	43.7
MobileViT-v2 [12]	42.7	91.4	80.8	66.8	37.6
MaxViT-t [22]	57.8	94.4	86.7	81.4	53.9
PVT-v2 [23]	51.0	92.0	83.4	75.7	45.8

Table 6. Comparison of the classification performance (in %) of different mainstream CNN-based and vision transformer-based backbones. Results are color-coded as Best, Second best, Worst, Second worst.

For CNN-based models, we use a maximum learning rate of 1e-4 with a linear warm-up for 5 epochs. For transformer-based models, we use a maximum learning rate of 3e-4 with a linear warm-up of 50 epochs. We use cosine annealing learning rate decay. We train with a batch size of 128.

For the Query2Label [9] models, we use the default set of hyperparameters used in the original paper. We use the Adam optimizer with weight decay coefficient of 1e-2. We use a learning rate of 1e-4 with cosine annealing. In the Query2Label transformer, we use 1 encoder layer and 2 decoder layers. We vary the number of heads between 1 and 4. We use Resnet34 and SWIN-base backbones, pretrained on the ImageNet-22k dataset

Image augmentations: We use the same augmentations for the identification experiments that we use for classification, described in Section G.1.

Loss function: Binary cross entropy loss is used to train the models, since we have a multi-label classification objective. In order to account for the imbalance demonstrated by each trait, we use the weighted binary cross entropy loss for all models except Query2Label. For each trait, the loss for minority labels is scaled by a factor Γ_{scale}, where $\Gamma_{scale} = \frac{N_{major}}{N_{minor}}$ and N_{major}, N_{minor} are the number of majority labels and minority labels for each trait, respectively. Query2Label uses the assymmetric loss as part of their implementation, and we use the default implementation presented in the original paper.

G.2.1. Attention Maps from Query2Label

We visualize the attention maps shown in Figure 5 following the method described in the original Query2Label paper.

For multi-head models, we take the mean of the multiple attention maps. The attention maps are interpolated to the original image size. This allows us to compute the mIoU with the ground-truth segmentation maps on the *manual-annotation* test set, as shown in Table 3. Since the model is trained on squared images, we ensure that the attention maps are interpolated according to the resized input image.

G.3. Segmentation Experiments

Hyperparameters: For the semantic segmentation methods listed in the first section of Table 4 (PSPNet to Semantic FPN), we use the implementation provided in the Segmentation Models Pytorch (SMP) library. For our experiments, we used the Adam optimizer with a learning rate of 2e-4. The learning rate was scheduled using a cosine annealing learning rate scheduler, with a minimum learning rate of 1e-5. The models were trained with a batch size of 32 for up to 100 epochs. Early stopping was employed with a patience of 10 epochs.

For the instance segmentation methods listed in the second section of Table 4 (Mask2Former and YOLOv8), we used a learning rate of 2.5e-4 and a batch size of 4.

Augmentations: We used the *albumentations* library in pytorch for training data augmentations. The augmentation pipeline includes horizontal flipping with a probability of 0.5 and shift-scale-rotate transformations that allow scaling up to 50%, rotating within a limit of 0 degrees, and shifting up to 10%, applied with a probability of 1. The images were resized to a maximum size of 320 pixels while maintaining the aspect ratio, and padding was added as needed to ensure a size of 320 $\times$ 320 pixels. Padding used a constant border mode with a white background. Gaussian noise was added to images with a probability of 0.2, and perspective transformations were applied with a probability of 0.5. To enhance brightness and contrast variations, one of the following augmentations was randomly applied with a probability of 0.9: CLAHE (Contrast Limited Adaptive Histogram Equalization), random brightness and contrast adjustment, or gamma adjustment. The pipeline also included blurring effects, where one of the following was applied with a probability of 0.9: sharpening, Gaussian blur, or motion blur, each with a blur limit of 3. Furthermore, to introduce color variations, one of the following was randomly applied with a probability of 0.9: hue-saturation adjustment or additional brightness and contrast adjustment. This comprehensive augmentation strategy was adapted from default recommendations in the SMP library.

Loss Functions: We trained all models in Table 4 (except YOLOv8) with cross-entropy loss and dice loss for segmentation, weighed equally. We used the default loss implementation for the YOLOv8 model.

G.3.1. Molmo-SAM Implementation Details

For the zero-shot segmentation method combining Molmo and SAM, we provide Molmo with images in the FV-Segmentation test set and prompt it using the text: "Point me to the $< trait >$ of the fish." The placeholder $< trait >$ is replaced with one of the nine trait names listed in Figure 11. For the caudal fin, $< trait >$ is replaced with "caudal fin or tail" to account for the non-scientific terminology, as the caudal fin is commonly referred to as the tail.

Molmo outputs numeric points corresponding to the traits, if detected. Using these points, we prompt SAM-v2 to generate nine binary segmentation masks for each image, where each mask corresponds to one of the nine traits. These binary masks are then merged into a single segmentation map labeled with the different traits. In cases where traits overlap, we resolve the conflict using a predefined priority order (low to high): Head, Eye, Dorsal Fin, Pectoral Fin, Pelvic Fin, Anal Fin, Caudal Fin, Adipose Fin, Barbel. Traits with higher priority are assigned overlapping pixels. This priority order is determined based on the physical arrangement of traits and their segmentation difficulty. For instance, Eye is prioritized over Head since the eye is always within the head, while Adipose Fin and Barbel are given the highest priority as they are the most challenging traits to segment.

In our implementation, we use the 'allenai/Molmo-7B-D-0924' variant of Molmo. We enable greedy decoding (we set temperature to 0), to prevent varying outputs. We use the 'sam2.1_hiera_large' variant of SAM-v2, and use default configuration provided in the SAM-v2 repository.

Exploring alternative methods to enhance the performance of the Molmo+SAM pipeline is an interesting direction for future work but is beyond the scope of this paper.

H. Additional Experiments and Results

H.1. Classification

In addition to the classification results provided in the main paper in Table 2, we provide a comprehensive evaluation of mainstream vision backbones in Table 6. We use the same implementation details described in Appendix G.1.

H.2. Identification

H.2.1. Comprehensive Benchmarking

A comprehensive benchmarking was conducted for trait identification, evaluating each model on the three evaluation sets: the *in-species test set* (Table 8), the *leave-out-species test set* (Table 9), and the *manual-annotation test set* (Table 10). In terms of the metrics, we report the mean average precision (mAP), the average precision for each of the four traits, the macro-averaged F1 score at a 0.5 threshold,

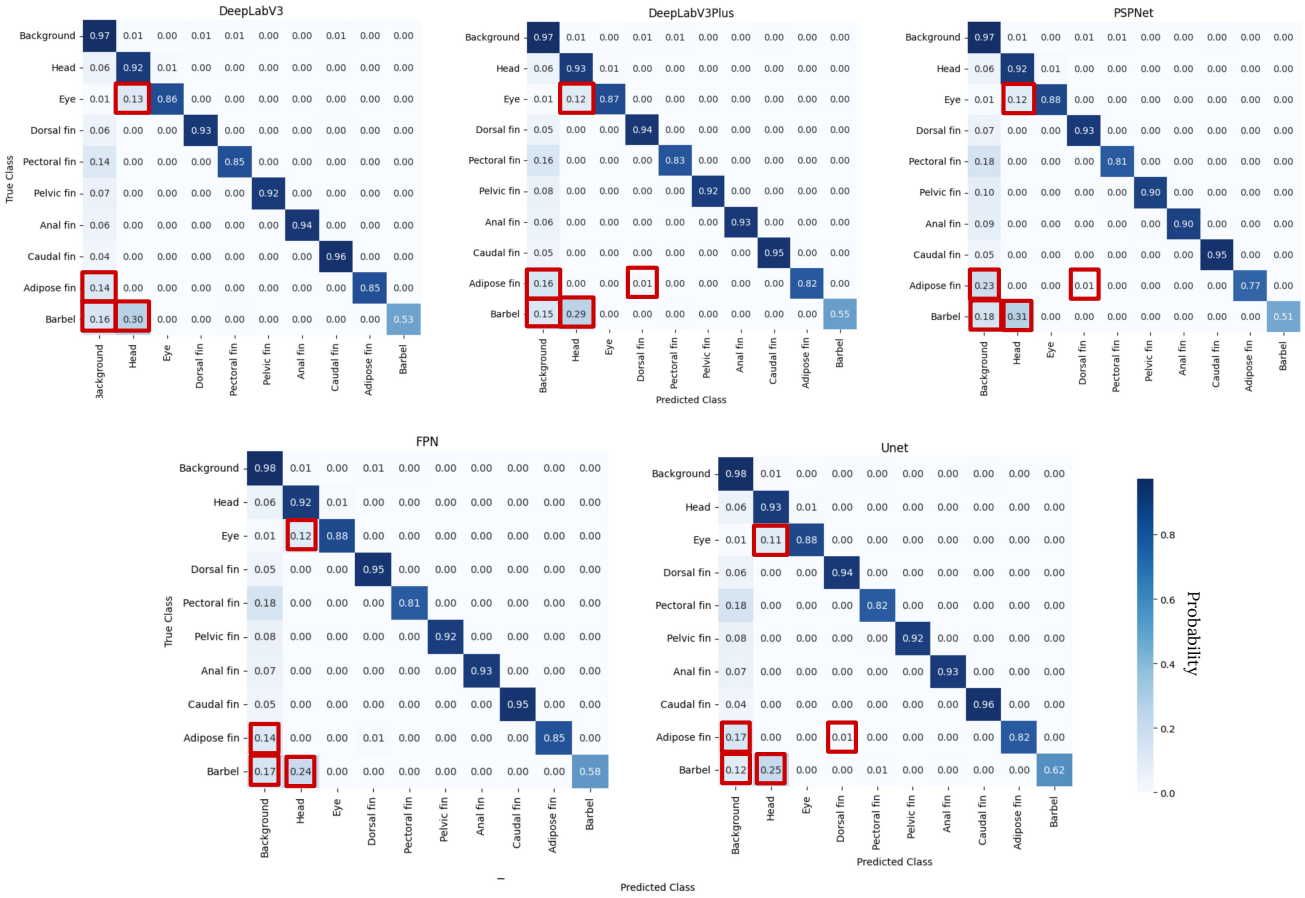

Figure 12. Confusion matrix for the five semantic segmentation models, with cells of interest highlighted in red frames.

and the macro-averaged F1 score at the optimal threshold. For each model, the optimal threshold is determined from the precision-recall curve of the validation set. Adip, Pelv, Barb and Dors refer to each of the 4 individual traits for trait identification – adipose, pelvic, barbel and multiple dorsal fins. We use the same implementation details described in Appendix G.2.

We observe similar results as the main paper – models obtain high performance on the *in-species test set*, with progressively lower performance on the *leave-out-species test set* and the *manual-annotation test set*. Computing the optimal threshold for F1 score generally improves performance over the default 0.5 threshold, which is expected given the imbalanced nature of our traits. We observe that all variants of the Query2Label model – Resnet34 (R34) and SWIN backbones, each trained with either a single head (SH) or four heads (Multiple Head, MH) – consistently outperforms other models. As mentioned in the main paper, all models face significant challenges in achieving decent performance on the *manual-annotation* set. This underscores the difficulty state-of-the-art models encounter in robustly generalizing to predict the fine-grained visual traits.

H.2.2. Query2Label Loss Ablation

As mentioned in Appendix H.2, the Query2Label results that we present are with the original implementation with asymmetric loss (ASL), while all other backbones are trained with the weighted BCE loss. For comparison, we train a Q2L-SWIN-MH model (Query2Label with SWIN backbone and 4 attention heads) with the weighted BCE (W-BCE) loss, and evaluate it on the *manual-annotation* dataset for FV-Id, shown in Table 7. We observe that W-BCE performs slightly worse than ASL loss (58.22%vs 58.27% mAP), but Q2L still outperforms other backbones from Table 10.

H.3. Segmentation

We present the confusion matrices for the five semantic segmentation models used in our experiments in Figure 12, with cells of interest highlighted in red. A consistent pattern emerges across all models: the adipose fin, when misclassified, is often segmented as background, likely due to its rarity in the dataset. In other instances, it is misclassified

Loss	mAP	F1 @ optimal threshold			
		Adip	Pelv	Barb	Dors
ASL	58.27	73.52	71.46	77.68	75.53
W-**BCE**	58.22	72.27	70.34	76.06	78.50

Table 7. Comparison of ASL vs W-BCE loss for Query2Label. We observe slight performance difference between the two losses.

as the dorsal fin, which can be attributed to their close spatial proximity, and their subtle difference in appearance, as shown in Figure 11.

Similarly, the barbel is frequently misclassified as background or as the head. This behavior can be explained by the barbel's rarity, its typical appearance over the head region, and its small area in images. On the other hand, the eye, which is also small and located on the head, is misclassified much less than the barbel. This can be attributed to the fact that the eye is consistently present in our dataset (i.e., it is not a rare trait). However, the eye is often segmented as part of the head due to their close association.

This analysis highlights key challenges in segmentation: small and rare traits are more likely to be segmented as background, traits that are spatially adjacent, or overlaid on other traits, or appear similar to other traits are prone to being misclassified as the other trait. These findings highlight the need for segmentation methods to handle rare, small, spatially nearby and fine-grained traits effectively.

Model	Average Precision					F1@0.5				F1@optimal threshold			
	mAP	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors
VGG-19	87.06	95.09	61.85	96.21	95.07	92.61	68.28	94.62	93.17	93.69	80.85	94.7	93.28
ResNet-18	87.85	96.56	65.96	95.16	93.73	91.26	78.95	93.7	91.16	94.47	79.01	94.07	92.18
ResNet-34	91.77	95.71	80.73	95.53	95.1	94.42	72.4	94.64	92.96	94.41	88.84	94.72	92.78
Inception-v3	77.0	93.08	52.4	86.03	76.47	87.45	77.67	88.93	83.53	93.16	78.48	88.69	84.95
ResNext-50	91.44	98.53	74.64	97.03	95.54	96.07	86.56	96.32	94.09	96.9	85.23	96.45	94.99
MobileNet-v3	90.49	96.34	73.79	96.33	95.51	94.38	83.73	95.19	93.69	94.58	84.26	95.14	94.73
RegNet-Y	89.51	95.54	73.06	95.45	94.0	94.62	82.28	94.86	94.12	94.69	81.6	94.77	94.04
EfficientNet-v2	95.96	99.7	86.41	98.43	99.3	98.13	90.69	97.29	98.04	98.48	92.22	97.23	98.28
ConvNext-Base	97.46	99.54	92.33	98.67	99.32	98.62	92.31	98.28	98.46	98.62	95.93	98.19	98.45
ViT-B-16	86.95	92.93	67.7	93.29	93.89	91.51	66.75	92.41	92.57	91.71	81.23	92.14	92.68
ViT-B-32	81.44	89.79	59.81	89.18	86.97	88.13	64.63	88.48	86.57	90.08	79.07	89.46	87.91
DEiT-distilled	93.74	97.61	82.85	97.02	97.48	95.5	76.11	94.93	96.12	95.72	86.88	95.76	96.47
SWIN-B	92.02	96.64	76.98	97.35	97.11	94.99	82.12	95.69	95.46	95.28	86.78	95.76	95.76
SWIN-B-22k	95.21	98.05	86.94	97.55	98.28	95.95	86.37	96.07	96.47	96.12	91.29	96.13	96.62
CVT-13	83.61	93.22	49.56	96.21	95.43	92.92	71.72	94.47	94.08	93.37	78.1	94.47	94.61
Mobile-ViT-xs	75.12	90.57	25.66	93.78	90.5	91.94	65.17	93.84	91.21	92.7	66.94	94.25	90.96
Mobile-ViT-v2	86.3	95.21	62.3	95.29	92.4	92.81	70.38	94.72	91.66	93.28	78.25	94.38	92.32
MaxViT-t	95.49	98.33	86.44	98.88	98.3	96.87	87.71	97.91	97.77	97.06	91.99	97.74	97.68
PVT-v2	96.48	98.84	90.85	97.75	98.49	97.61	84.15	96.94	97.58	97.62	91.63	97.15	97.92
Q2L (R34 SH) [9]	97.78	99.47	93.91	98.47	99.26	97.65	97.36	97.35	98.27	97.9	95.1	97.73	98.45
Q2L (R34 MH)	97.22	99.4	91.92	98.38	99.17	98.05	96.65	97.43	98.39	98.36	96.03	97.61	98.62
Q2L (Swin SH)	98.32	99.81	94.94	99.0	99.53	99.09	97.36	97.82	99.19	98.93	95.66	98.03	99.06
Q2L (Swin MH)	98.32	99.82	94.91	98.94	99.61	98.73	98.04	98.38	99.02	98.93	96.82	98.35	99.23

Table 8. Trait identification results on mainstream visual models using the **in-species test set**. Results are color-coded as Best, Second best, Worst, Second worst.

Model	Average Precision					F1@0.5				F1@optimal threshold			
	mAP	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors
VGG-19	49.97	68.42	43.36	65.47	22.62	79.93	57.35	79.55	65.03	81.18	67.52	81.05	63.5
ResNet-18	49.27	81.46	23.89	80.68	11.06	79.16	66.84	83.77	54.31	85.16	64.85	84.0	55.42
ResNet-34	52.14	79.8	37.24	75.45	16.09	86.67	58.07	82.87	55.76	86.67	69.73	82.87	55.73
Inception-v3	28.93	67.01	6.33	27.27	15.13	73.56	55.49	63.14	57.37	76.53	49.61	61.65	61.51
ResNext-50	53.0	91.16	1.42	72.86	46.56	93.28	48.29	81.23	73.4	82.29	48.71	80.91	72.85
MobileNet-v3	53.36	89.93	21.18	60.63	41.68	89.58	60.13	77.15	71.08	90.16	61.08	74.65	73.2
RegNet-y	34.67	67.04	12.23	45.17	14.24	80.03	54.01	69.27	57.17	75.4	56.86	71.16	56.88
EfficientNet-v2	83.86	95.45	88.1	83.97	67.91	95.15	84.59	86.76	79.31	92.76	87.07	82.62	79.43
ConvNext-Base	79.6	96.87	73.18	85.14	63.21	96.24	84.32	88.3	74.24	96.24	77.98	88.3	78.71
ViT-B-16	47.27	58.54	40.13	67.6	22.82	69.78	66.66	70.2	62.41	69.27	65.89	76.86	61.87
ViT-B-32	35.29	47.21	14.49	50.29	29.17	69.3	56.96	69.79	64.87	62.59	56.1	64.72	63.66
DEiT-distilled	61.74	65.19	43.15	82.55	56.07	80.14	63.26	85.71	76.08	73.5	70.35	86.78	75.87
SWIN-B	60.22	81.1	38.66	79.96	41.15	87.02	71.63	86.15	69.24	82.11	59.69	86.87	68.83
SWIN-B-22k	68.18	89.0	59.52	80.03	44.18	92.04	75.96	85.03	57.88	91.5	74.77	83.88	58.25
CVT-13	28.94	55.04	1.84	48.15	10.73	77.28	48.23	72.95	52.77	69.61	47.13	72.95	54.55
Mobile-ViT-xs	34.23	55.65	13.99	58.08	9.2	77.36	56.12	75.46	52.03	76.35	54.39	75.17	52.67
Mobile-ViT-v2	33.76	52.62	3.74	64.05	14.63	73.59	46.78	79.37	54.76	70.19	51.98	77.4	53.75
MaxViT-t	75.42	87.18	81.3	76.03	57.15	88.61	69.04	79.62	75.05	88.21	83.98	84.27	75.63
PVT-v2	60.72	83.91	9.26	89.3	60.42	80.42	56.72	88.41	78.74	79.24	49.62	90.49	76.61
Q2L (R34 SH)	79.86	92.93	67.54	89.14	69.82	88.54	85.56	86.14	70.02	92.79	88.51	90.09	78.74
Q2L (R34 MH)	74.64	92.5	50.37	85.21	70.47	91.04	78.74	86.84	70.83	93.48	78.74	89.0	82.07
Q2L (SWIN SH)	88.41	98.61	93.06	97.3	64.65	97.84	92.75	96.08	74.5	97.42	93.98	95.22	75.43
Q2L (SWIN MH)	88.23	99.17	96.39	97.62	59.76	97.49	99.04	95.84	73.98	97.84	98.12	95.84	76.69

Table 9. Trait identification results on the **leave-out-species test set**. Results are color-coded as Best, Second best, Worst, Second worst.

Model	Average Precision					F1@0.5				F1@optimal threshold			
	mAP	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors	Adip	Pelv	Barb	Dors
VGG-19	38.48	49.36	23.15	35.79	45.61	64.89	58.48	62.53	64.09	70.12	59.19	66.48	65.87
ResNet-18	39.61	46.24	18.42	42.02	51.74	58.86	57.21	68.76	59.42	69.07	57.69	70.89	66.68
ResNet-34	45.53	58.78	25.66	38.66	59.02	75.17	61.16	66.45	70.64	74.81	56.85	66.9	68.02
Inception-v3	30.07	48.07	14.22	22.66	35.32	58.86	49.07	58.26	53.36	72.76	48.09	60.23	55.66
ResNext-50	43.25	57.02	19.56	36.62	59.79	70.12	52.37	63.25	62.53	73.58	52.37	62.67	68.89
MobileNet-v3	41.99	46.18	27.96	30.44	63.36	69.87	57.63	61.03	70.68	70.5	56.75	61.72	71.85
RegNet-y	38.1	43.45	23.14	31.93	53.87	66.74	54.87	64.19	66.65	70.37	52.25	66.74	66.56
EfficientNet-v2	55.36	63.96	30.75	55.08	71.66	74.03	60.84	73.13	75.42	76.35	58.07	75.28	78.06
ConvNext-Base	53.96	61.27	38.19	47.55	68.84	73.39	69.59	66.03	78.68	73.5	67.52	68.63	77.02
ViT-B-16	37.63	37.69	26.92	36.72	49.2	64.72	58.76	66.35	66.57	65.74	60.47	66.61	67.57
ViT-B-32	33.7	33.43	24.34	30.47	46.54	58.11	60.19	60.33	63.91	62.28	56.96	63.14	65.76
DEiT-distilled	40.69	40.36	32.82	29.5	60.08	61.85	64.24	58.77	69.82	67.34	61.97	64.11	70.63
SWIN-B	44.68	40.67	35.6	36.36	66.07	64.16	62.32	67.73	74.7	68.0	60.16	68.14	76.93
SWIN-B-22k	51.53	55.55	32.2	50.54	67.82	72.51	65.99	73.78	75.45	72.21	63.0	71.46	74.37
CVT-13	34.76	35.87	24.28	30.76	48.13	58.53	53.3	59.69	63.48	61.84	57.93	59.69	67.91
Mobile-ViT-xs	30.6	39.08	13.7	23.99	45.62	65.27	54.82	62.68	64.64	69.42	52.95	61.58	67.46
Mobile-ViT-v2	34.89	44.5	19.84	28.45	46.78	67.61	57.85	63.99	63.17	68.33	54.95	65.63	65.83
MaxViT-t	49.67	50.34	33.71	51.55	63.07	69.21	65.67	74.5	72.52	68.97	63.33	72.03	71.53
PVT-v2	46.42	44.53	27.7	47.75	65.71	62.68	62.42	71.54	73.56	64.08	56.96	71.45	75.24
Q2L (R34 SH)	55.84	61.67	36.8	50.44	74.43	76.16	67.35	73.61	80.07	75.23	69.25	72.76	79.63
Q2L (R34 MH)	53.98	56.6	40.04	45.91	73.36	75.42	67.54	72.57	78.9	70.93	67.88	72.18	77.8
Q2L (SWIN SH)	59.39	59.18	41.65	63.46	73.28	75.74	70.83	79.16	78.69	74.8	70.54	77.78	78.44
Q2L (SWIN MH)	58.27	57.97	42.71	60.56	71.83	74.55	70.83	77.89	78.18	73.52	71.46	77.68	75.53

Table 10. Trait identification results on the challenging **manual-annotation test set**. All models struggle to identify traits on the diverse set of species contained within the manual-annotation set. Results are color-coded as Best, Second best, Worst, Second worst.

References for Appendix

[1] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 6

[2] CVAT.ai Corporation. Computer vision annotation tool (cvat) (v2.4.3), 2023. 4

[3] 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*, pages 248–255, 2009. 5

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[6] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 6

[7] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020. 5

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[9] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*, 2021. 6, 10

[10] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 6

[11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 6

[12] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 6

[13] Joel Pepper, Jane Greenberg, Yasin Bakiş, Xiaojun Wang, Henry Bart, and David Breen. Automatic metadata generation for fish specimen image collections. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 31–40. IEEE, 2021. 2, 3

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[15] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 6

[16] Ravid Shwartz-Ziv, Micah Goldblum, Yucen Li, C Bayan Bruss, and Andrew G Wilson. Simplifying neural network training under class imbalance. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[18] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 5

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[20] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 6

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 6

[22] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pages 459–479. Springer, 2022. 6

[23] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 6

[24] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021. 6

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6