Cross-Family U-Net Landmark Heatmap Regression For Butterfly Wings -- Context and Generalization by Grouping

Auteur : Akkawi, Jad

Promoteur(s) : Geurts, Pierre; Marée, Raphaël

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil en science des données, à finalité spécialisée

Année académique : 2024-2025

URI/URL : http://hdl.handle.net/2268.2/23219

University of Liège

School of Engineering and Computer Science

Cross-Family U-Net Landmark Heatmap Regression For Butterfly Wings

Context and Generalization by Grouping

Author

Jad Akkawi

Supervisors

Pr. Pierre Geurts

Pr. Raphaël Marée

Pr. Vincent Débat

Master's thesis completed in order to obtain the degree of Master of Science in Engineering and Data Science

Academic Year 2024-2025

Acknowledgments

I would like to sincerely thank my supervisors, Professors Pierre GEURTS *and Raphaël* MARÉE*, and Vincent* DÉBAT *for their precious help, advice, and guidance throughout my work. Their expertise and insightful discussions have been invaluable to this research.*

I also extend my heartfelt gratitude to Maëlle STEYER *for her unwavering support and motivation throughout this journey, especially as I balanced this work alongside a full-time engineering job.*

I acknowledge the use of AI tools including Grammarly, SciSpace, and Claude to enhance writing clarity, literature searches, and technical implementation guidance, while all research, development, and analysis remain my original work.

This project has been both a challenge and an enriching experience, and I am thankful for the opportunity to undertake it.

Abstract

University of Liège

School of Engineering and Computer Science

Cross-Family U-Net Landmark Heatmap Regression For Butterfly Wings
Context and Generalization by Grouping

Jad Akkawi

Supervised by Pr. Pierre Geurts and Pr. Raphaël Marée and Pr. Vincent Débat

Academic Year 2024-2025

This thesis presents a novel approach to automated landmark detection on butterfly wings using deep learning techniques, addressing the challenge of cross-family generalization in morphometric analysis. Accurate landmark detection is essential for studying morphological variations in butterflies, but traditional manual annotation is time-consuming and impedes large-scale research. We propose several improvements to U-Net-based landmark detection through anatomically-informed grouping strategies and enhanced preprocessing. By integrating YOLOv8 for butterfly detection and cropping before landmark prediction, we significantly improve input quality. We explore multiple model configurations, comparing single-landmark versus multi-landmark channel approaches, different loss functions (MSE and FBCE), and varying input resolutions (256×256 and 512×512). Our experiments on specimens from the Papilionidae family demonstrate that anatomically-guided multi-landmark grouping achieves convergence four times faster than conventional approaches while maintaining comparable accuracy. Higher resolution models (512×512) show substantially improved precision when evaluated at original image scale. Most importantly, our cross-family generalization experiments reveal that models trained on the combined Papilionidae and Morpho datasets with strategic landmark grouping and index matching successfully adapt to varying numbers of landmarks across butterfly families. These findings advance morphometric analysis capabilities in Lepidopterology and demonstrate the value of incorporating domain knowledge into neural network architecture design for biological feature detection tasks.

Keywords: *Butterfly morphometrics; landmark detection; U-Net architecture; heatmap regression; anatomical landmark grouping; cross-family generalization; YOLOv8; deep learning; wing venation patterns.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

1.1.1 Landmark Detection in Butterflies

Landmark detection is essential for analyzing morphological variations across species. Scientists can quantify shape differences by identifying landmarks to study biological form, evolutionary relationships, and functional morphology. In butterflies, wing morphology is particularly important, influencing key behaviors such as migration efficiency, mating strategies, feeding behavior, oviposition, and predator avoidance. Accurate and consistent landmark detection is, therefore, crucial for understanding these ecological and evolutionary dynamics.

1.1.2 Manual Preparation of Butterfly Specimens for Imaging

Large-scale landmark-based studies, such as those conducted in natural history museums, require high-throughput imaging techniques to analyze extensive specimen collections. For instance, the Muséum National d'Histoire Naturelle de Paris houses over one million butterfly specimens, providing a vast dataset for morphometric research.

Specimen preparation follows a standardized workflow to ensure imaging accuracy. Butterflies are stored in archival boxes to prevent degradation. When retrieved, they are handled carefully to preserve delicate structures. Proper positioning on a neutral background with fully extended and symmetrical wings is essential for consistency in image acquisition. High-resolution cameras capture anatomical features under controlled lighting conditions, and researchers manually annotate landmarks by marking homologous points on the wings and body using an annotation software for images. Each specific landmark is indexed so that all points are stored in the same order for every butterfly specimen. However, this process is meticulous, time-consuming, and inherently limited in scalability.

Given the vast number of specimens in museum collections, manual annotation presents a significant bottleneck in morphometric research. The labor-intensive nature of specimen preparation, imaging, and annotation underscores the urgent need for automated methods that can accelerate landmark detection while maintaining accuracy, robustness, and reproducibility.

1.1.3 Automatic annotation of Butterfly Images with Deep Learning

Automated landmark detection refers to the computational process of identifying specific anatomical points on biological specimens without human intervention. Traditional computer vision techniques (like edge detection and feature extraction) were initially used for this task, but have been largely surpassed by machine learning approaches, particularly deep learning models that employ convolutional neural networks (CNNs). These deep learning methods learn hierarchical feature representations directly from training data, enabling them to identify landmarks across varied specimens despite differences in size, coloration, and morphology. By automating this traditionally manual task, researchers can rapidly analyze large collections of specimens. A review of existing deep learning methods for automatic annotation is presented in Section 2.2.

1.2 Problem Specification

The latest work on butterfly wing landmark detection by Marganne et al. [7] explored various techniques for automatically detecting landmark coordinates. The most promising approach employed a U-Net encoder-decoder architecture, where each landmark was assigned a separate output channel, represented as a regressed heatmap. The model was trained on butterfly species from the Morpho genus, achieving promising results. However, its performance degraded significantly when applied to butterflies from different families.

This failure stems from landmark variability, in number and distribution, across species, driven by evolutionary adaptations. Landmarks, typically located at vein intersections or along structural frames, may shift, disappear, or split into multiple points depending on wing morphology. For instance, a single landmark in one species may correspond to a triangular arrangement of three points in another due to differences in venation complexity.

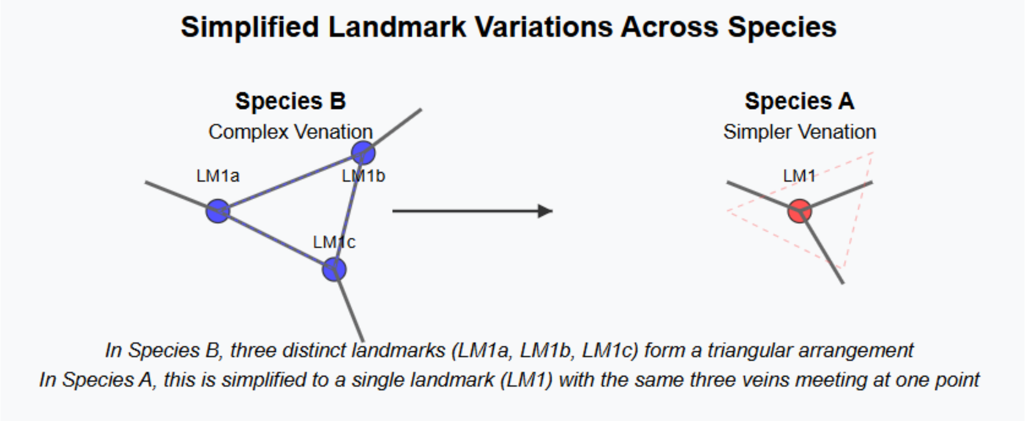

Simplified Landmark Variations Across Species
Species B
Complex Venation
LM1a
LM1b
LM1c
Species A
Simpler Venation
LM1
In Species B, three distinct landmarks (LM1a, LM1b, LM1c) form a triangular arrangement
In Species A, this is simplified to a single landmark (LM1) with the same three veins meeting at one point

Figure 1.1: *Comparative wing venation showing triangular landmark arrangement in Species B reduced to a single point in Species A. The Image is created by a generative AI tool, it's a compiled SVG code, after many instructions and corrections based on a sketch provided by Professor Vincent Débat.*

Current deep learning models struggle to generalize across diverse butterfly families, requiring extensive re-training when applied to new datasets. Addressing challenges like the one in figure 1.1 requires a more adaptable approach capable of capturing contextual relationships between landmarks rather than relying on static, predefined points.

1.3 Objective

Building upon the work of Marganne et al. [7] done on the Morpho genus, this study aims to enhance butterfly landmark detection by improving both processing and model architecture. First, we integrate a YOLO v8 model into the preprocessing phase to detect and crop butterflies before resizing the images to the model's input size, ensuring a more accurate representation of butterflies at the pixel level.

Next, we explore U-Net's capacity to capture landmark correlations by grouping landmarks connected by veins into subsets and assigning each subset to a dedicated output heatmap channel. This encourages the model to attend to wing venation patterns during feature extraction, leveraging regional context to improve robustness against high inter-species variability.

The dataset for this study consists of butterfly specimens from the Papilionidae family, which exhibit greater shape, size, and pattern diversity, as well as a higher number of landmarks compared to the Morpho genus. This increased variability makes it a suitable testbed for assessing model generalization.

In post-processing, we transform heatmaps to landmarks for each channel, averaging the coordinates of landmarks that exist in more than one channel. We also develop spatial metrics to compare model outputs better.

Finally, we compare this structured landmark grouping approach to the classical U-Net architecture, which assigns one channel per landmark. Additionally, we explore training a model on a combined dataset of both butterfly families to evaluate its ability to predict a varying number of landmarks, thereby improving generalization across species.

1.4 Data

1.4.1 MNHN

The Muséum national d'Histoire naturelle in Paris is a renowned institution dedicated to natural history research and preservation. Established in 1793, it encompasses various scientific disciplines such as botany, zoology, geology, and paleontology. The museum houses vast collections of millions of specimens vital for educational and research endeavors. Its galleries and research centers attract scientists and tourists from around the globe, establishing it as a hub for scientific inquiry and public engagement. The MNHN's commitment to biodiversity and conservation emphasizes its importance as a leading institution in the natural sciences.

1.4.2 MNHN's Papilionidae Butterfly Dataset

The dataset used in this study comprises **2,655** high-resolution images (2848×4288 pixels) from MNHN's collection, provided by Professor Vincent Débat and belonging to the Papilionidae family. Members of this family can be found on every continent except Antarctica, yet most are tropical. This dataset is particularly valuable for landmark detection research due to its taxonomic breadth within the Papilionidae family (commonly known as swallowtails), significant morphological diversity, and standardized imaging conditions. Each butterfly specimen was photographed from dorsal and ventral views, resulting in approximately 1,327 unique butterfly specimens.

Taxonomic Composition

The dataset exhibits remarkable taxonomic diversity, containing:

- **31** unique genera within Papilionidae
- **374** distinct species and subspecies
- Coverage of all three subfamilies within Papilionidae:
    - Papilioninae (dominant, including genera such as *Papilio*, *Graphium*, and *Troides*)
    - Parnassiinae (represented by *Parnassius*, *Archon*, and others)
    - Baroniinae (represented by *Baronia* with 3 specimens)

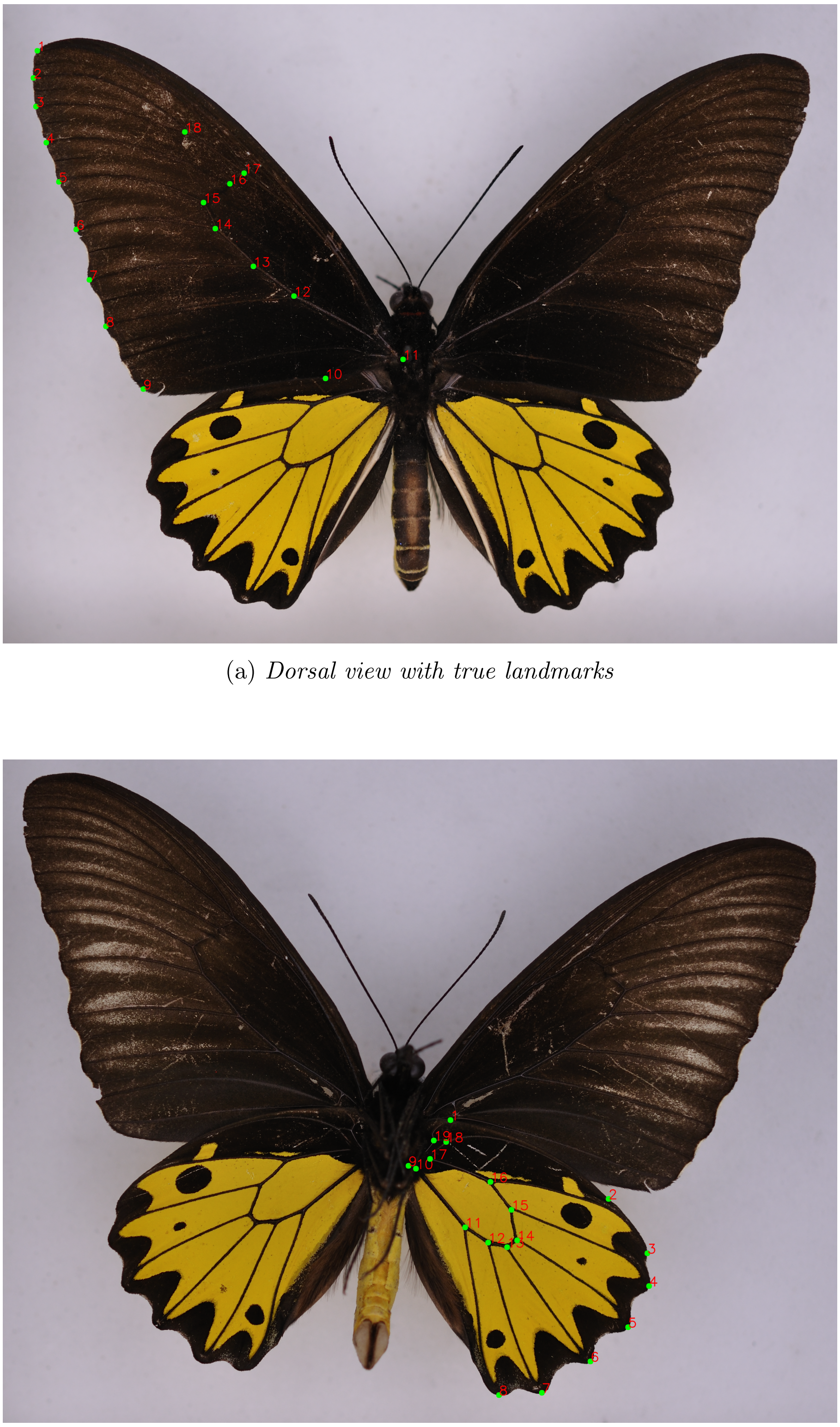
Table 1.1: *Distribution of butterfly genera in the dataset*

Genus	**Count (%)**
Papilio	616 (23.2%)
Parnassius	150 (5.6%)
Graphium	119 (4.5%)
Parides	84 (3.2%)
Troides	64 (2.4%)
Ornithoptera	52 (2.0%)
Protographium	23 (0.9%)
Byasa	22 (0.8%)
Battus	20 (0.8%)
Allancastria	16 (0.6%)
Pachliopta	16 (0.6%)
Atrophaneura	14 (0.5%)
Protesilaus	14 (0.5%)
Mimoides	12 (0.5%)
Luehdorfia	12 (0.5%)
Archon	8 (0.3%)
Iphiclides	8 (0.3%)
Euryades	8 (0.3%)
Trogonoptera	8 (0.3%)
Zerynthia	8 (0.3%)
Losaria	7 (0.3%)
Lamproptera	7 (0.3%)
Meandrusa	6 (0.2%)
Bhutanitis	5 (0.2%)
Hypermnestra	4 (0.2%)
Eurytides	4 (0.2%)
Pharmacophagus	4 (0.2%)
Cressida	4 (0.2%)
Sericinus	4 (0.2%)
Teinopalpus	4 (0.2%)
Baronia	3 (0.1%)

Landmark Annotation

The dataset includes detailed landmark annotations, carefully curated to account for specimen quality:

- Some Specimens have damaged wings, so all specimens were annotated only on their best-preserved side
- Landmarks were organized into separate annotation files (TPS format) for each wing position:
  - HL: Hind Left wing - HR: Hind Right wing
  - FL: Front Left wing - FR: Front Right wing
- Two landmark categories were identified :
  - **True landmarks**: Anatomically homologous points located at vein intersections or vein terminations (18 in Front wings and 19 in Hind wings, see Figure 1.2)
  - **Semi-landmarks**: Series of points delineating wing edges and major structural curves (multiple curves with varying point densities, see Figure 1.3)
- Each annotation includes scale factors for converting pixel measurements to physical dimensions



(a) *Dorsal view with true landmarks*

(b) *Ventral view with true landmarks*

Figure 1.2: *Examples of true landmarks on butterfly wings showing anatomically homologous points located at vein intersections and terminations*

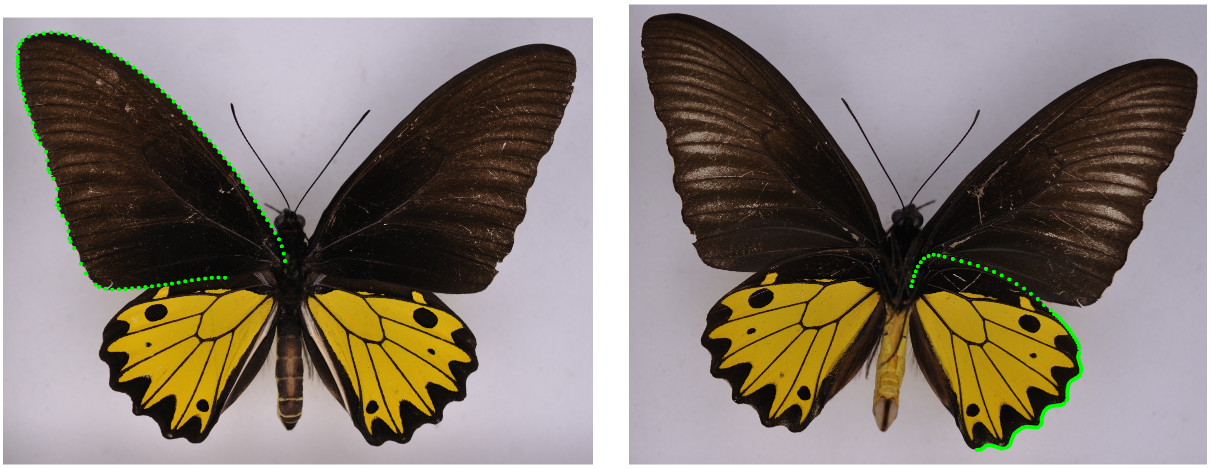


Figure 1.3: *Examples of Semi-Landmarks for Front and Hind Wings and their labels*

Morphological Variability

A significant challenge in this dataset is the high degree of morphological diversity (Figure 1.4). Specimens exhibit substantial variation in:

- **Wing shape:** From the elongated hindwings of *Papilio* to the rounded wings of *Parnassius*
- **Size:** Wingspan ranging significantly.
- **Wing patterns:** From relatively uniform coloration to complex patterning with eyespots, bands, and patches
- **Venation patterns:** Varying configurations of wing veins affecting landmark positions

Focus on True Landmarks

This morphological diversity creates a challenging test bed for our landmark detection algorithms, which must maintain accuracy while generalizing across significant anatomical variations. While the dataset contains both true landmarks and semi-landmarks, our study focuses specifically on the detection of true landmarks. This decision was made because (1) semi-landmarks are numerous and simply delineate the outer wing edges, making them less informative for morphological analysis, and (2) true landmarks represent anatomically homologous points with greater biological significance at vein intersections and terminations. By concentrating our efforts on the more challenging task of robust and accurate true landmark detection, we address the core problem in butterfly morphometrics while developing algorithms that can handle high taxonomic and morphological variability.

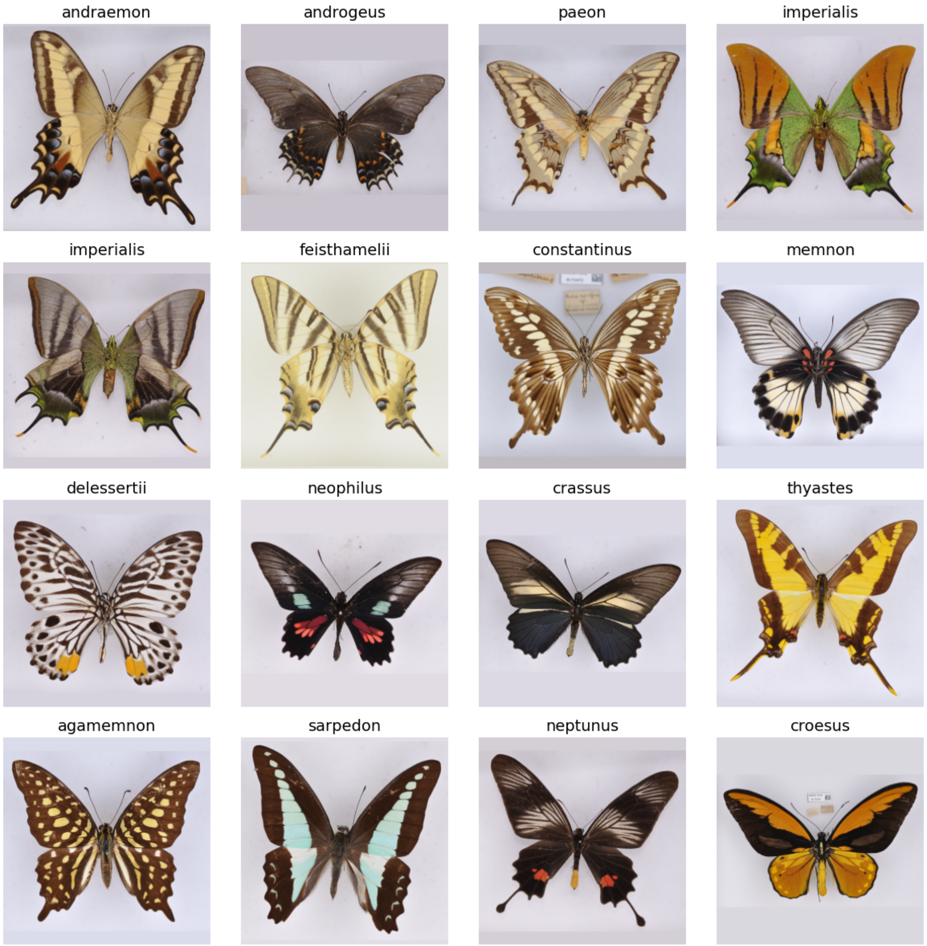


Figure 1.4: *Morphological diversity in the Papilionidae family showing significant variations in wing shape, coloration patterns, and venation structures. Specimens include multiple species (labeled above each specimen) from different genera within Papilionidae, with the top two rows displaying ventral views and the bottom two rows showing dorsal views. Note: Butterflies are not presented at the same scale, further highlighting the size variability that challenges automated landmark detection systems.*

1.4.3 MNHN's Morpho Butterfly Dataset

In addition to the Papilionidae dataset, this study also incorporates a second dataset of Morpho genus butterflies, previously analyzed by Marganne et al. [7]. This Morpho dataset comprises 945 individuals (1,841 annotated images) across 28 species, featuring different landmark distributions; 14 true landmarks for hind wings compared to 19 in Papilionidae. The Morpho specimens exhibit less morphological variability, but they still present challenges, namely color pattern variation and substantial variance in landmark positions, even within the same species. The annotated images (933 dorsal views and 908 ventral views) contain 44 landmarks for dorsal views (18 true landmarks, 26 semi-landmarks) and 29 landmarks for ventral views (14 true landmarks, 15 semi-landmarks). By combining both datasets, our research explores model performance with varying numbers of landmarks and across different taxonomic groups, providing insights into the generalization capabilities of landmark detection algorithms across butterfly families.

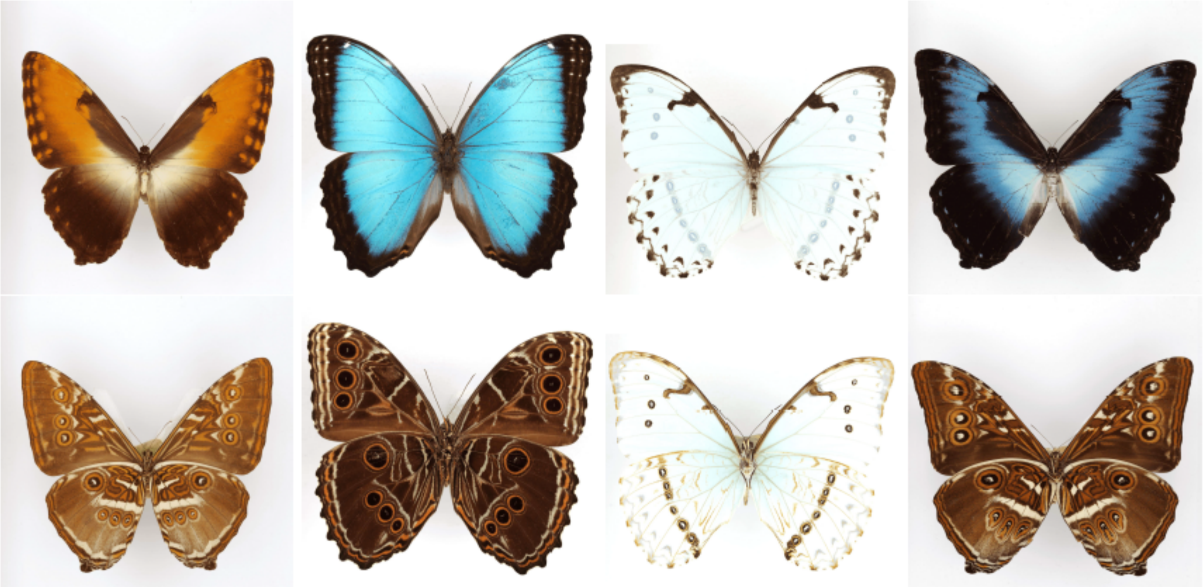
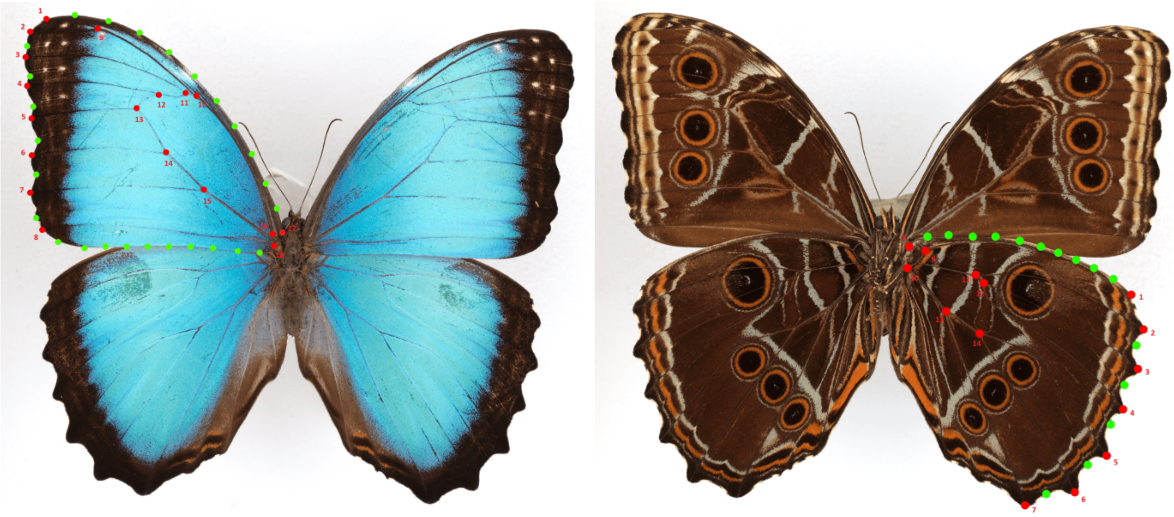


Figure 1.5: *"Sample of individuals from the Morpho genus (top: dorsal view; bottom: ventral view; from left to right: M. hecuba, M. granadensis, M. polyphemus and M. cisseis; all males. NB: specimens not at the same scale)"* [7]

We can see here the difference in landmark number and distribution of the Morpho genus when compared with Figure 1.2. The Morpho butterfly shown below has fewer landmarks on its hind wing, whereas Papilionidae specimens typically have additional landmarks at vein intersections throughout the hind wing surface. Since both butterfly families have the same number of true landmarks on their front wings, we will focus our study on the hind wings only.



(a) Dorsal view

(b) Ventral view

Figure 1.6: *"Example of annotations (Red: true landmarks, Green: semi-landmarks)"* [7]

1.4.4 Cytomine Research Platform

Our research utilized Cytomine (research.cytomine.be), an open-source collaborative image analysis platform initially developed at the University of Liège in 2010 [32]. This platform is significant for our work as it represents an ongoing collaboration between our university and the Cytomine research team at ULiège's Montefiore Institute. Cytomine describes itself as *"an* *open-source rich internet application for collaborative analysis of multi-gigapixel images"* [33], enabling precise landmark annotation on high-resolution images and providing a Python API for efficient data retrieval and manipulation. The platform was instrumental in our workflow for two key reasons: it allowed for detailed examination of wing venation patterns through its advanced visualization tools and facilitated seamless access to both our Papilionidae dataset and Marganne's existing Morpho dataset.

Chapter 2

Related Work

2.1 Landmark Detection in Biology

Landmark detection is a widely used technique in biological sciences for analyzing morphological variations, particularly in evolutionary biology, functional morphology, and taxonomic classification. The primary goal is to identify homologous points across specimens, allowing for comparative shape analysis [15]. This method is frequently used in fields such as geometric morphometrics, where the spatial configuration of biological structures is studied in a statistically robust manner [14].

In butterflies, landmark-based morphometric analysis has been extensively applied to study wing shape, flight dynamics, and evolutionary adaptations. Landmark-based analysis was used to investigate how butterfly wings vary in shape across different populations[16]. Other research highlights the importance of landmark selection in butterfly wing morphometrics and its relevance in evolutionary studies[17], or discusses how natural and sexual selection drive the evolution of butterfly wing morphology [20]. Geometric morphometrics were also crucial in examining environmental effects on butterfly wing shape variation[18].

2.2 Deep Learning Architectures for Landmark Detection

Recent advancements in deep learning have significantly enhanced landmark detection in biological images, providing automated solutions that aim to improve upon traditional manual annotation methods. Yet challenges with accuracy and handling outliers persist, as well as with generalization, making continuous refinement necessary, especially where annotated data is limited and requires high expertise. Several deep learning approaches have shown promise in addressing these challenges, although their effectiveness can vary widely depending on the specific application and data quality.

2.2.1 Convolutional Neural Networks

CNNs are fundamental in landmark detection due to their exceptional feature extraction capabilities. These networks process images hierarchically, identifying spatial patterns. For instance, Zhang *et al.* (2017) [22] demonstrated a two-stage deep learning framework where The first-stage CNN model takes millions of local 3D image patches as input and predicts their 3D displacements to multiple anatomical landmarks, effectively learning patch-to-landmark associations. The second-stage CNN model refines this by using a fully convolutional network (FCN) that treats the entire image as input, leveraging the learned weights from the first stage to jointly predict the displacements of grid-sampled patches. This significantly improves accuracy in scenarios where training data is scarce.

2.2.2 Heatmap-based Approaches

Unlike regression-based methods that predict landmark coordinates directly, heatmap-based approaches represent a spatial probability distribution around points of interest. This technique has gained prominence in landmark detection due to its ability to preserve spatial relationships and facilitate a mapping from image space to image space. Given the success of these methods, our work contributes to this evolving subfield of landmark localization.

Kumar et al. [8] conducted a comparative study on heatmap-based regression versus direct coordinate regression across various CNN architectures and imaging modalities. Their results demonstrate that heatmap-based U-Net architectures outperform direct regression, particularly when paired with exponential heatmap generation functions.

Capturing Landmark Correlations and Context

One of the major challenges in anatomical landmark detection is incorporating the spatial correlation between landmarks and leveraging local and global context. These correlations are often biologically driven, such as spatial dependencies between vertebrae in spinal images or the intersections of veins on insect wings. Several studies have introduced novel approaches to model these relationships:

- **Spatial Feature-Based Correlation Modeling**: Ham et al. [10] proposed a spatial-configuration-feature-based network that integrates anatomical correlations into the model. Their U-Net architecture generates heatmaps, which are then transformed into spatial feature vectors using the differentiable "soft-argmax" function. A correlation map is derived from these vectors and incorporated into the loss function to enhance localization accuracy in Hand X-rays.

- **Attention-Driven Landmark Detection**: Zhong et al. [11] introduced a two-stage U-Net framework leveraging attention mechanisms to capture inter-landmark dependencies. The first stage generates global heatmaps, which inform attention maps that guide the second stage to refine high-resolution heatmap patches. This method has proven effective in cephalometric X-ray images. Similarly, a cascade U-Net framework [12] combines global U-Net regression with CNN-based displacement refinement to further enhance precision.

- **Transformer-Based Landmark Detection**: Kasturi et al. [23] developed a Transformer-based landmark detection model, integrating long-range dependencies with encoder-decoder architectures reminiscent of U-Net. By combining CNN feature extraction with self-attention mechanisms, their approach effectively captures global spatial relationships in chest X-ray images.

Generalizing to a Varying Number of Landmarks

Most landmark detection models assume a fixed number of keypoints, limiting their ability to generalize across datasets with varying anatomical or structural features. Stern et al. [21] address this challenge by proposing a single-stage heatmap-based U-Net architecture that detects an arbitrary number of landmarks using a single foreground heatmap, instead of separate channels for each keypoint. Their approach is designed to dynamically detect a varying number of sutures in endoscopic images of mitral valve repair, applying thresholding and center-of-mass extraction directly to the heatmap output to localize keypoints adaptively.

Chen et al. [19] propose a complementary two-stage framework that combines 3D Faster R-CNN with multi-scale U-Net for craniomaxillofacial landmark detection. Their method addresses landmark variability caused by anatomical deformities and imaging limitations by treating landmark detection as an object detection problem in the first stage, followed by heatmap-based refinement in the second stage, enabling simultaneous detection of varying numbers of anatomical landmarks in CBCT images without requiring fixed landmark configurations across subjects.

Applications and DL research on insect wings

Deep learning has been applied to insect landmark detection, particularly in wing morphometrics. One study on tsetse fly wings [13]used a two-step approach: the first step applies ResNet-based CNN classifiers to distinguish complete from incomplete wings, and the second step employs a CNN for direct coordinate regression and a UNet++ segmentation model which is a version of UNet with nested skip connections that enhances gradient flow and captures finer details, the Dice loss was used to handle class imbalance.

Marganne et al.[7] explored automatic landmark annotation on Morpho butterfly wings using three approaches: computer vision, machine learning, and deep learning. The computer vision method, which relied on edge detection and concave hull computations, proved ineffective. The machine learning approach used Extremely Randomized Trees to predict landmarks separately and performed well on smaller datasets with a very reduced number of species. The deep learning approach employed a U-Net model to generate heatmaps representing landmark probability distributions, with exponential and Gaussian functions defining landmark influence. Deep learning outperformed machine learning on larger datasets, and was integrated into the Cytomine [25] platform for practical use by biologists.

Chapter 3

Theoretical Background

3.1 Fundamental Concepts in Deep Learning for Computer Vision

Neural Networks

Neural networks are system architectures made up of interconnected units called neurons, arranged in layers that process input data to produce desired outputs.

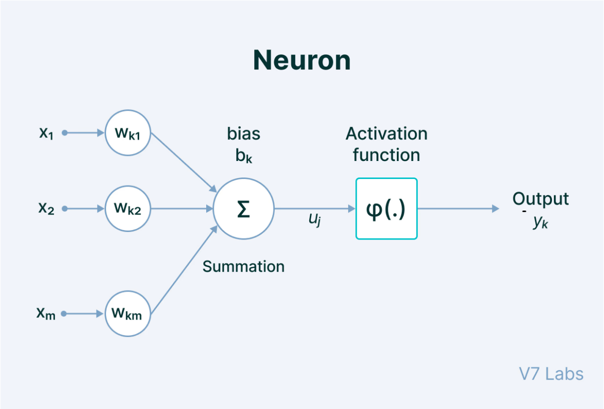


Figure 3.1: *Schematic representation of a neuronal unit processing an input vector, where each input element* $\boldsymbol{x_i}$ *is weighted by corresponding parameters* $\boldsymbol{w_{k,i}}$ *and summed with a bias term. The resulting linear combination is transformed by an activation function, producing either an intermediate feature representation for subsequent layers or a final output prediction.*[27]

In the forward propagation phase, data flows from the input layer through one or more hidden layers to the output layer. Here, each neuron calculates a weighted sum of its inputs followed by a non-linear activation function, leading to predictions based on the current parameters of the network.

Next, in the backward propagation phase, known as backpropagation, the gradient of the loss function is calculated concerning each weight using the chain rule of calculus, sending error signals backward through the network. These gradients are then used to update the network weights according to the gradient descent rule:

$$w_{t+1} = w_t - \eta \frac{\partial L}{\partial w_t} \tag{3.1}$$

where w_t represents the weight at iteration t, η is the learning rate, and $\frac{\partial L}{\partial w_t}$ is the gradient of the loss function with respect to the weight.

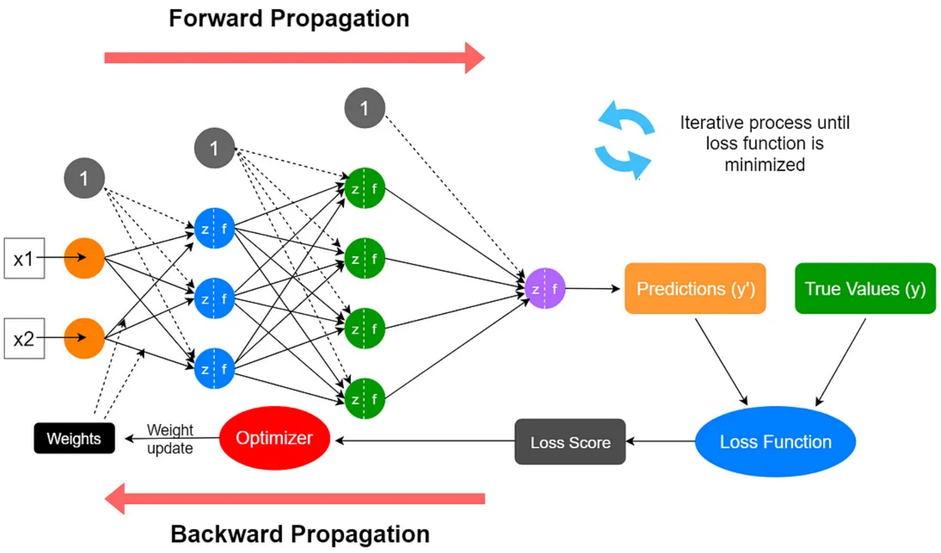


Figure 3.2: *Schematic representation of the neural network training cycle: forward propagation generates predictions, followed by loss computation against ground truth. Next, backpropagation occurs, along with weight updates by an optimization algorithm. This iterative process continues until convergence.*[26]

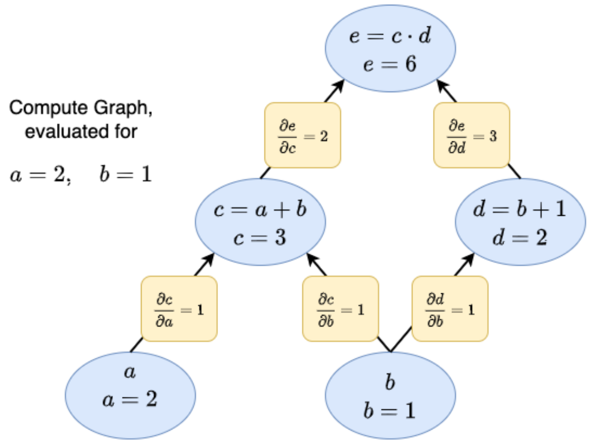

Compute Graph, evaluated for
$a = 2, \quad b = 1$
$e = c \cdot d$
$e = 6$
$\frac{\partial e}{\partial c} = 2$
$\frac{\partial e}{\partial d} = 3$
$c = a + b$
$c = 3$
$d = b + 1$
$d = 2$
$\frac{\partial c}{\partial a} = 1$
$\frac{\partial c}{\partial b} = 1$
$\frac{\partial d}{\partial b} = 1$
a
$a = 2$
b
$b = 1$

Figure 3.3: *Computational graph depicting a sequence of interconnected neuronal operations, with explicit calculation of partial derivatives at each node illustrating the gradient flow during backpropagation.* [28]

This dual-directional flow of information allows neural networks to iteratively refine their parameters with gradient-based optimization methods, thereby reducing the gap between predictions and actual outcomes.

The layers examined in the following sections (convolutional, transposed convolutional, and max pooling) serve as specialized components. Together with suitable activation functions, these elements help build the core structure of contemporary computer vision networks, like U-Net.

Convolutional layers

Convolutional layers are essential in CNNs and are the main feature extractors in U-Net(see 3.4). They slide a filter across the input map, performing element-wise multiplication with local regions and summing the results to generate output activations. This enables hierarchical feature extraction, allowing the network to capture spatial patterns at different scales.

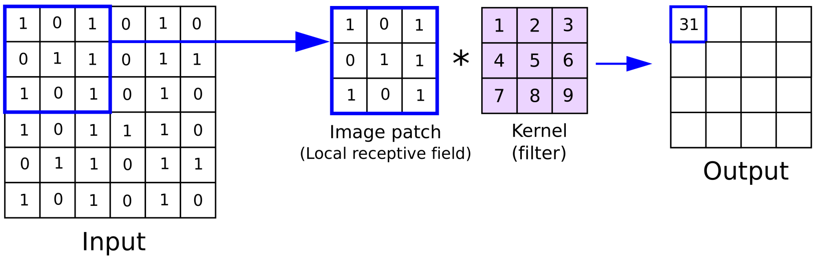


Figure 3.4: *Schematic representation of a cross-correlation operation applied to a single-channel 6×6 input using a 3×3 convolutional kernel. The resulting output dimensions indicate a stride parameter of 1 and no padding, demonstrating the spatial reduction characteristic of standard convolution operations.*[29]

A key feature of convolutional layers is parameter sharing, as the same filter weights are applied across the input, improving efficiency and preserving spatial relationships. Moreover, these layers utilize local connectivity, connecting each output neuron to a small input region (receptive field), which captures localized patterns. This design also provides translation invariance, helping the network recognize features regardless of their position in the image.

Transpose Convolutional layers

Transpose convolutional layers, or deconvolutional layers, increase the spatial resolution of feature maps, thereby reversing the standard convolution process. They achieve upsampling by expanding the input feature map, often by inserting zeros between elements, then applying a learnable convolutional filter. This process reconstructs spatial details lost during downsampling.

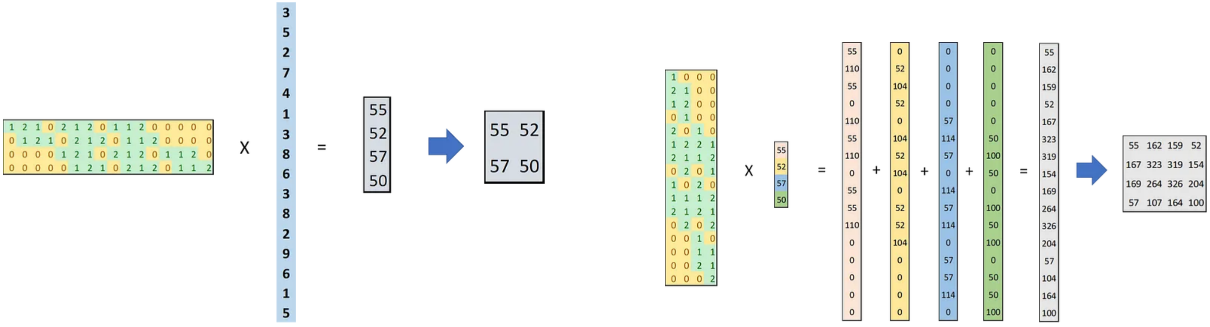


Figure 3.5: Illustration of convolution and transposed convolution as matrix multiplications. *Left: Cross-correlation represented as matrix multiplication between a zero-padded* 4×16 *convolution matrix and a* 16×1 *input vector. Right: Transposed convolution (deconvolution) represented as matrix multiplication between a zero-padded* 16×4 *matrix and a* 4×1 *input vector.* (Adapted from [34])

Transpose convolutions are crucial in the decoder of architectures like U-Net, helping recover spatial resolution for high-resolution feature maps in pixel-wise prediction tasks, such as heatmap regression.

Max Pooling layers

Max pooling is a non-parametric spatial downsampling operation commonly employed in convolutional neural networks to reduce the dimensionality of feature maps, thus decreasing computational complexity in subsequent layers. The core principle of max pooling involves selecting the maximum activation value from each local region of the input feature map, thereby preserving the most prominent features while discarding less significant spatial details.

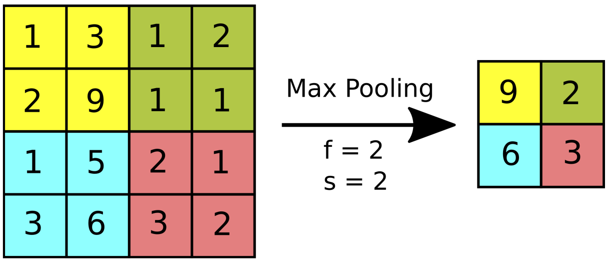


Figure 3.6: *Illustration of a max pooling operation with a filter size of 2×2 (f* = 2*) and stride of 2 (s* = 2*), transforming a 4×4 input feature map into a 2×2 output feature map by selecting the maximum value within each receptive field.*[29]

This systematic reduction allows deeper layers to process increasingly larger receptive fields, enabling hierarchical feature extraction.

3.1.1 Activation Functions

Activation functions are crucial in neural networks, providing non-linear transformations to neuron outputs. Mathematically, this is represented as:

$$y = f(Wx + b) \tag{3.2}$$

where W is the weight matrix, x is the input vector, b is the bias, and f is the activation function. They introduce non-linearity in the computational graph, allowing complex function approximations, supported by the Universal Approximation Theorem. This theorem states that a single hidden-layer neural network can approximate continuous functions on compact subsets of $\mathbb{R}^n$ with certain activation functions.

A notable example is the Rectified Linear Unit, or ReLU for short, activation function[37], defined as:

$$f(x) = \max(0, x) \tag{3.3}$$

which retains positive inputs while setting negative inputs to zero. Unlike sigmoidal activations [38], ReLU is computationally efficient and helps mitigate vanishing gradient issues.

3.1.2 RMSProp Optimization Algorithm

Root Mean Square Propagation, or RMSprop for short, is an adaptive learning rate optimization algorithm that improves upon classical gradient descent by automatically adjusting the learning rate for each parameter based on the historical magnitude of gradients [24]. Unlike standard gradient descent which uses a fixed learning rate for all parameters, RMSprop scales the learning rate inversely proportional to the root mean square of recent gradients. The RMSprop update rules are given by:

$$v_t = \beta v_{t-1} + (1-\beta)\left(\frac{\partial L}{\partial w_t}\right)^2 \tag{3.4}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t + \epsilon}}\frac{\partial L}{\partial w_t} \tag{3.5}$$

where v_t maintains an exponential moving average of squared gradients, β is the decay rate (typically 0.9), η is the base learning rate, and ϵ provides numerical stability. This adaptive mechanism allows parameters with large gradients to have smaller effective learning rates and parameters with small gradients to have larger effective learning rates.

3.1.3 Multi-Scale Feature Extraction and Reconstruction

In modern neural network architectures for computer vision, the synergistic interaction between convolutional layers, transposed convolutional layers, max pooling, and activation functions establishes a sophisticated downsampling-upsampling pipeline.

The downsampling path progressively reduces spatial dimensions while increasing feature depth, enabling the network to develop hierarchical representations with enhanced semantic richness. Conversely, the upsampling path facilitates spatial expansion, reconstructing higher-resolution feature maps from these compact, semantically rich representations.

When augmented with *skip connections* (as in U-Net) or *lateral connections* as in Feature Pyramid Networks, this integrated pipeline effectively combines multi-scale features. Skip connections in U-Net directly concatenate feature maps from encoder to decoder layers, while FPN uses lateral connections to merge semantically strong deep features with spatially precise shallow features (see Figure 3.7).

3.2 YOLO Architecture for Object Detection

YOLO [4] is a single-stage object detection framework that treats detection as a regression problem. Unlike traditional two-stage detectors, YOLO predicts bounding boxes and class probabilities directly from full images in a single forward pass, enabling efficient real-time detection.

In our preprocessing pipeline, we specifically employ YOLOv8 [5], one of the most recent iterations of this architecture. While a comprehensive analysis of this model's technical advancements is beyond the scope of this thesis, its architecture leverages several of the fundamental concepts discussed earlier. Unlike traditional CNNs, these object detection variants avoid max pooling operations in favor of strided convolutions for downsampling, which helps maintain spatial information crucial for localization tasks. YOLOv8 implements a Feature Pyramid Network approach (see Figure 3.7), which creates multi-scale feature representations by connecting semantically strong deep features with spatially precise shallow features, addressing the multi-scale nature of object detection.

The general YOLO approach divides input images into a grid system, with each cell responsible for detecting objects centered within it. This design, combined with YOLOv8's architectural advantages, provides robust butterfly detection across varied backgrounds, orientations, and lighting conditions - essential for standardizing inputs to our landmark detection pipeline.

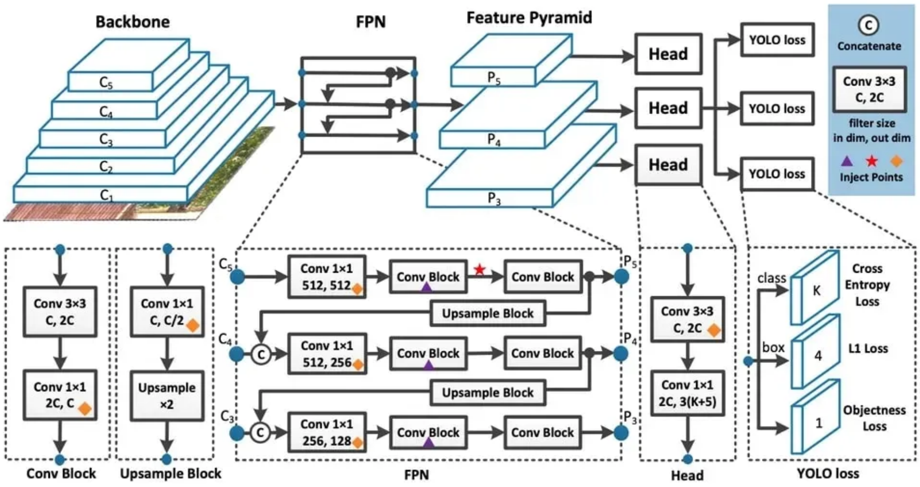


Figure 3.7: *Overview of YOLOv8 Architecture.*[30]

3.3 Open Image Dataset

The Open Images Dataset [1] is a large-scale image dataset designed for machine learning and computer vision tasks. It contains over 9 million images, with annotations for more than 600 object classes. For our project, we employed a YOLOv8 model pre-trained on this dataset, leveraging its butterfly class annotations. The Open Images Dataset provides bounding box annotations, hierarchical class labels, and segmentation masks, which enabled the model to learn robust butterfly detection. The diversity in the dataset, including variations in species, poses, and environmental conditions (see Figure 3.9), ensures that the resulting detection model generalizes effectively to real-world scenarios. This pre-trained model serves as the initial preprocessing step in our pipeline, automatically detecting and cropping butterfly specimens before they are passed to our landmark detection architecture.

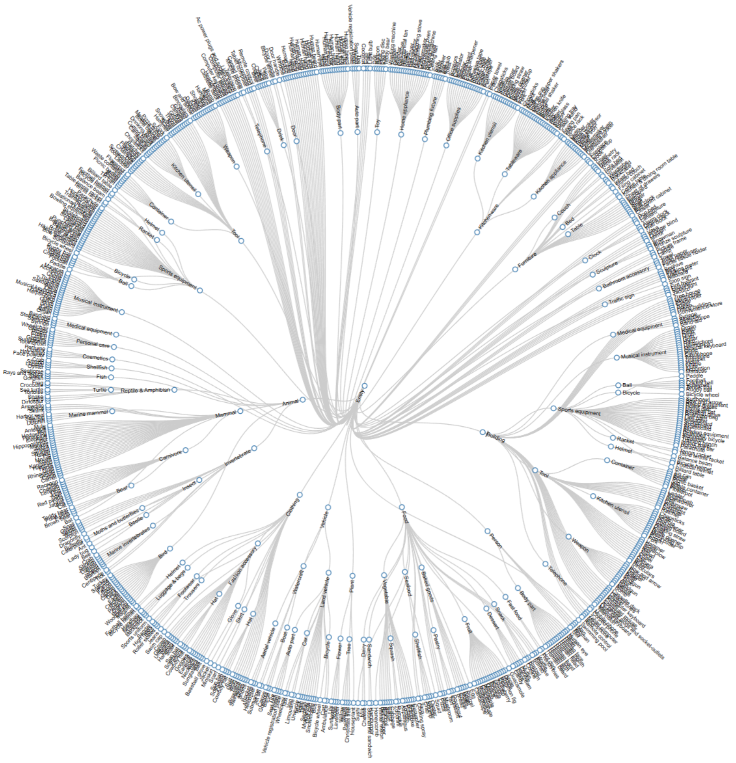


Figure 3.8: *Hierarchical organization of the 600 object classes in the Open Images Dataset. This visualization demonstrates the taxonomic structure of the dataset, with 'Moths and butterflies' represented as a distinct category, as well as 'Butterfly' as its subcategory, within the broader classification system* [2]

Positive image-level: Insect, Invertebrate, Plant, Butterfly, Moths and butterflies, Animal
ID: 2bf20dfbe1f9c4fd | Download overlay
Source URL: Full Resolution, Thumbnail
Title: Butterfly tree 2
Author: Hafiz Issadeen (License)

Positive image-level: Insect, Invertebrate, Flower, Butterfly, Moths and butterflies, Animal
Negative image-level: Plant
ID: 4ac039dfb171791d | Download overlay
Source URL: Full Resolution, Thumbnail
Title: Florida white butterfly
Author: U.S. Fish and Wildlife Service Headquarters (License)

Positive image-level: Insect, Invertebrate, Plant, Butterfly, Moths and butterflies, Animal
Negative image-level: Flower
ID: 7c321c74665a9f7f | Download overlay
Source URL: Full Resolution, Thumbnail
Title: Walking Highgate & Hampstead Heath
Author: surprise truck (License)

Positive image-level: Poster, Insect, Invertebrate, Clothing, Butterfly, Moths and butterflies
Negative image-level: Animal
ID: c81e77e65187bc7d | Download overlay
Source URL: Full Resolution, Thumbnail
Title: B. Claudia's Picture
Author: Peter Morville (License)

Figure 3.9: *Representative examples of butterfly specimens with bounding box annotations from the Open Images Dataset. These annotations were used to train the pre-trained YOLOv8 model we employed in our preprocessing stage* [3]

3.4 U-Net Architecture

The U-Net architecture is a convolutional neural network designed for image segmentation tasks, particularly effective in applications requiring precise pixel-level segmentation, such as medical imaging and biological studies. It comprises two main components: the contracting path (encoder) and the expanding path (decoder).

Contracting Path

The contracting path performs hierarchical feature extraction through a sequence of convolutional layers followed by ReLU activations and max-pooling operations. This path reduces the spatial dimensions of the input image while increasing the depth of feature maps, thus capturing high-level semantic information. In our butterfly landmark detection context, the encoder learns to identify relevant wing structures and patterns that inform landmark positions.

Expanding Path

The expanding path reconstructs the spatial resolution of the segmented image while maintaining the semantic information extracted by the encoder. They differ by using transposed convolutions to progressively increase the resolution. As explained earlier in Section 3.1.3, skip connections are employed to concatenate feature maps from the encoder to corresponding decoder layers, integrating high-resolution spatial information with abstract features.

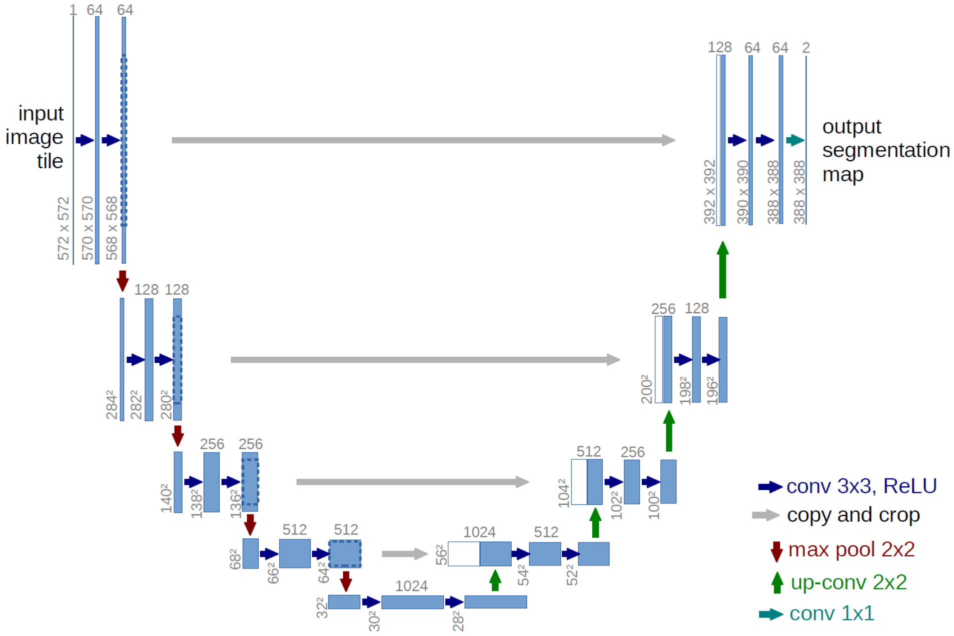


Figure 3.10: *Overview of U-Net Architecture showing the distinctive U-shaped design with contracting path (left), expanding path (right), and skip connections (horizontal arrows).* [6].

Advantages of U-Net for Landmark Detection

U-Net offers several advantages for landmark detection tasks. Its skip connections preserve spatial information during downsampling and upsampling. The architecture's design enables effective training with limited datasets through its efficient parameter sharing and feature reuse between encoder and decoder paths. U-Net's multi-channel output capability naturally accommodates heatmap regression approaches, where each landmark can be represented as a probability distribution in a separate channel.

3.5 Loss Functions

3.5.1 Mean Squared Error Loss in Heatmap Regression

Definition and Formula

MSE Loss is widely used for quantifying the error in regression tasks. It calculates the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual values. Mathematically, it is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{3.6}$$

where n represents the number of pixels in a heatmap, $\hat{y}_i$ denotes the predicted value at pixel i, and y_i is the true value at pixel i.

Application to Heatmap Regression

In heatmap regression for butterfly landmark detection, the model outputs a spatial probability map for each landmark, where intensity values represent the likelihood of a landmark's presence. The predicted heatmap $\hat{y}$ aims to mirror the ground truth heatmap y, which typically has a Gaussian distribution centered at the landmark's true location.

MSE Loss is particularly suitable for this task because it penalizes larger spatial deviations more severely, ensuring precise localization of wing landmarks such as vein intersections and boundary points.

3.5.2 Focal Loss and Its Application to Continuous Heatmap Regression

Standard Focal Loss

Focal Loss [9] is an adaptation of Binary Cross-Entropy Loss and was originally developed to address class imbalance in object detection tasks. For landmark detection on butterfly wings, where background pixels far outnumber landmark regions, this imbalance is particularly pronounced. The standard form is defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{3.7}$$

where p_t represents the model's estimated probability for the correct class:

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{if } y = 0 \end{cases} \tag{3.8}$$

This notation allows the loss function to be expressed concisely for both positive and negative examples. The α_t parameter is a balancing factor that weights the contribution of different classes, helping to address the severe class imbalance between landmark and background pixels. The γ parameter (focusing parameter) reduces the loss contribution from well-classified examples (high p_t values), forcing the model to focus training on difficult cases where landmarks are harder to detect accurately.

Focal Loss for Continuous Heatmap Regression

In our butterfly landmark detection task, the exponential decay function used to generate target heatmaps creates smooth probability distributions centered at landmark locations, with values that gradually decrease with distance from the landmark center.

For continuous labels where $y \in [0, 1]$, the focal loss formulation is extended to handle non-binary targets. The probability term p_t becomes:

$$p_t = y \cdot p + (1 - y) \cdot (1 - p) \tag{3.9}$$

which measures how well the prediction aligns with the continuous ground truth, and the alpha weighting becomes:

$$\alpha_t = y \cdot \alpha + (1 - y) \cdot (1 - \alpha) \tag{3.10}$$

providing smooth interpolation between class weights based on the continuous label value.

This continuous formulation maintains the key properties of focal loss—emphasizing difficult predictions and addressing class imbalance—while accommodating the smooth spatial probability distributions characteristic of heatmap regression.

Implementation Note: We employ TensorFlow's `BinaryFocalCrossentropy` implementation with parameters $\alpha = 0.25$, $\gamma = 2.0$, which naturally supports continuous labels and is well-suited for our heatmap regression task.

Chapter 4

Methods And Protocols

4.1 Train, Validation, and Test Sets

Exploring different variants of the U-Net architecture and comparing between them requires a methodical approach. Namely, since the data set is small and with high variability, a fair comparison starts with fixing the training, validation and testing datasets to be used for all models.

A division of the dataset into 80%, 10%, and 10% subsets for training, validation, and testing respectively, was adopted. Randomness in the subsets is assured with a fixed randomness seed and having the seed fixed helps to compare all the models against a unique train/test/val dataset.

During training, the 80% subset is augmented, see the section 4.3 for more details.

After each training epoch, a model checkpoint is saved if the model's performance of the validation set is improved.

When training is over we use the test set to evaluate the model, visually as well as using custom metrics (see Section 4.7) to assess different quality aspects of the output.

4.2 Prepocessing

The original butterfly images from the MNHN collection present several challenges for landmark detection. Each specimen is photographed on a white surface, which is placed over grid paper along with measurement labels and identification tags. These layered background elements, while useful for museum documentation, create substantial noise for the landmark detection task. Moreover, in many cases, the butterfly specimen itself occupies less than half of the total image pixels, especially for smaller species.

Simply resizing these full images to the model's input dimensions (256×256 or 512×512 pixels) would result in the actual butterfly specimen being represented by a tiny number of pixels, making accurate landmark detection very difficult. This problem is particularly acute for smaller specimens, where critical wing venation details might be reduced to just a few pixels after resizing.

4.2.1 Butterfly Detection and Cropping

The model used for butterfly detection is a YOLOv8 model trained by Ultralytics [31] on the "Open Image Dataset"[1] that contains a whole category of butterfly images. The model used is the Medium sized 'YOLOv8m' with 26.2 million parameters.

First, we utilize this pre-trained YOLOv8 model to detect and localize butterfly specimens within each image. Once detected, we extract a tight crop around the butterfly, expanding the bounding box by 5% on each side to ensure no anatomical features are truncated, for instance, wing tails. This step effectively removes the unnecessary background elements while preserving the complete butterfly morphology.

Interestingly, our preprocessing approach evolved from earlier experiments with a COCO-trained YOLOv8 model that lacked specific butterfly training. For those experiments, we found that converting images to grayscale improved detection consistency by encouraging the model to associate butterflies with structurally similar classes like kites or umbrellas. We maintained this grayscale preprocessing in our current pipeline despite using the more specialized model.

The landmark coordinates are appropriately transformed to match the new cropped reference frame using:

$$x_{new} = x_{original} - x_{min}$$

$$y_{new} = y_{original} - y_{min}$$

Where (x_{min}, y_{min}) represents the top-left corner of the cropped bounding box.

The YOLOv8 model may sometimes detect the butterfly within a single bounding box, while at other times, we find a bounding box for the main body and additional bounding boxes for the wings. In these cases, we proceed by choosing the most top left of all top lefts and the most bottom right point of all bottom right. Additionally, some butterflies are positioned close to the edge, so the extra 5% needs to be bound by the edge of the image.

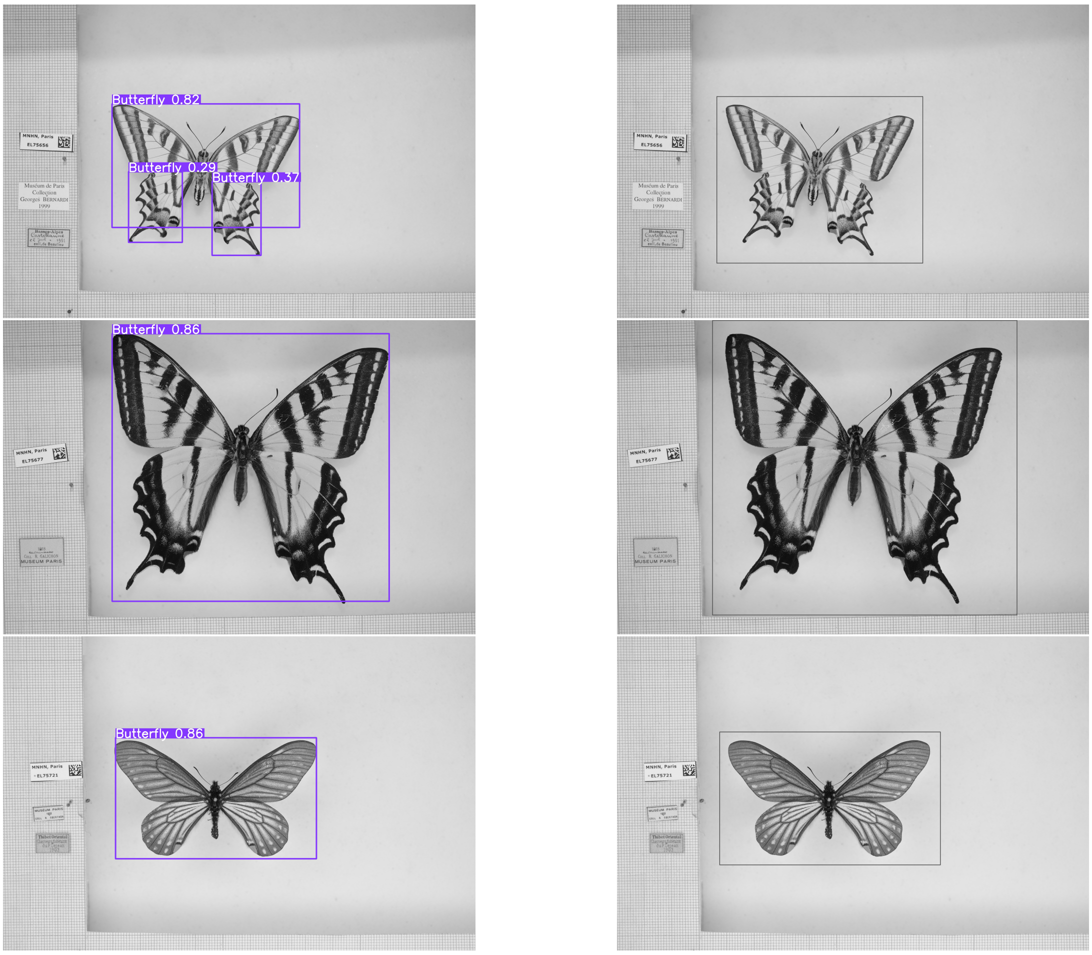


Figure 4.1: *Three examples of butterfly detection and bounding box generation by a YOLOv8 model trained by Ultralytics on the "Open Image Dataset". Left column: Original images with detection boundaries overlaid. Right column: Resulting crops with 5% expanded margins. Note that the images shown are in grayscale, due to a preprocessing technique carried over from our earlier experiments with a COCO-trained YOLOv8 model that lacked butterfly-specific training.*

4.2.2 Dimension Normalization

After cropping, we normalize the dimensions to create a uniform input for our model. Rather than simple resizing, which could distort the butterfly's proportions, we implement a dimension-preserving approach:

1. We identify the largest dimension of the cropped image (width or height)
2. This dimension is scaled to match our target size (256 or 512 pixels)
3. The smaller dimension is scaled proportionally, maintaining the original aspect ratio
4. Any remaining space is filled with padding to reach the target square dimensions; the padding signal value per channel is the average of the values found on the corresponding edge signals.

This approach ensures the butterfly maintains its natural proportions while maximizing its pixel representation within the input dimensions. The landmark coordinates are scaled accordingly using the same transformation factors.

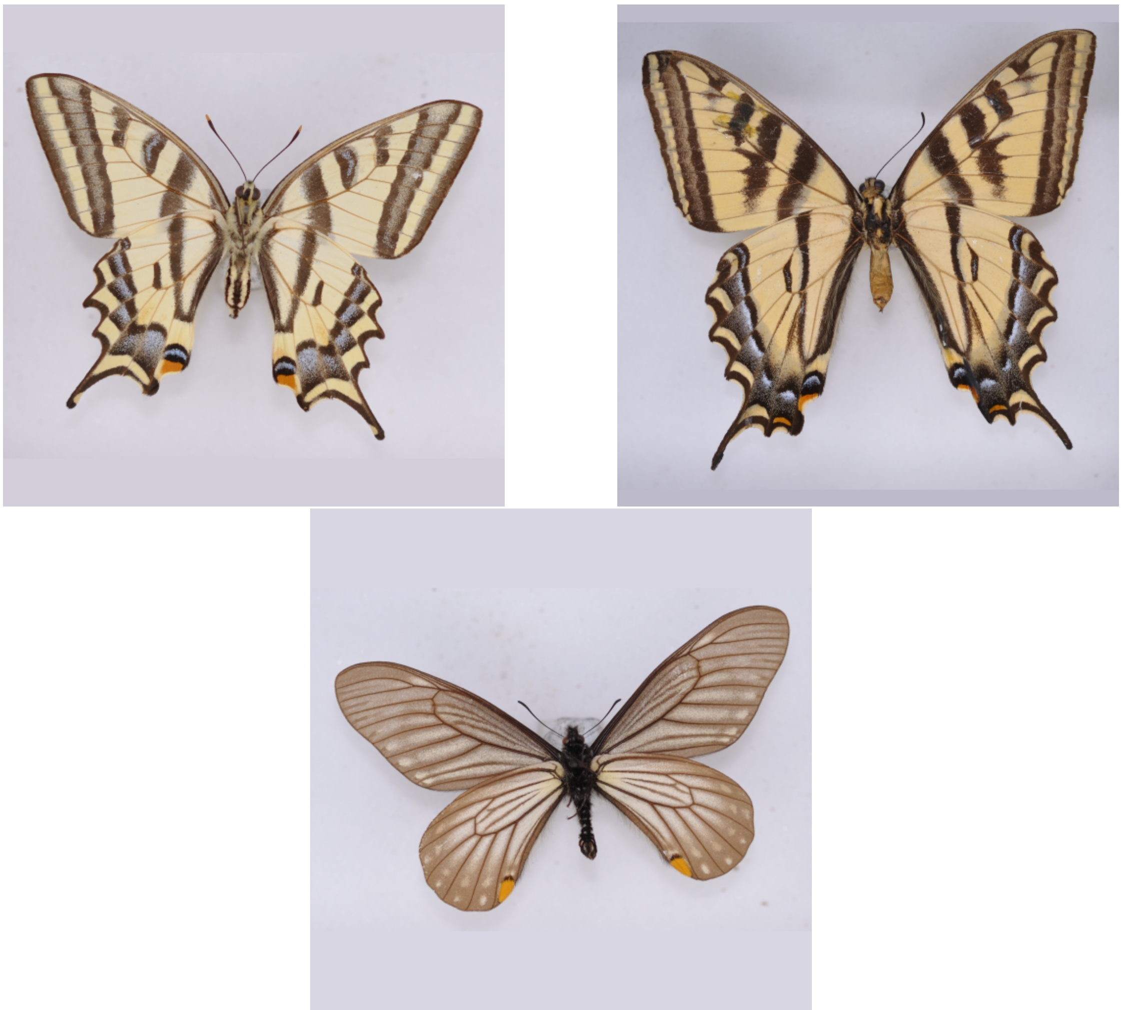


Figure 4.2: *Examples of resized butterfly specimens prepared for model input (512×512 pixels). Notice how each specimen maintains its natural proportions while maximizing its representation within the square input dimensions. Left-to-right, top-to-bottom: (1) A horizontally-oriented specimen with padding at top and bottom, (2) A specimen requiring minimal padding due to near-square proportions, and (3) An extremely horizontally-oriented specimen.*

For example, if the cropped butterfly image is 800×600 pixels and our target size is 256×256:

- The width (800px) is the larger dimension and is scaled to 256px
- The height is proportionally scaled to 192px
- 32px of padding is added to the top and bottom (distributing the difference of 64px)

After preprocessing, the butterfly specimens occupy the majority of the input image pixels, allowing the model to focus on the relevant anatomical features for landmark detection.

4.3 Image augmentation

During training, the 80% subset is augmented by applying the following alterations at certain probability of occurrence:

1. **Random Brightness and Contrast:**
    - Randomly adjusts the brightness and contrast of the image to simulate different lighting conditions.
2. **Affine Transformation:**
    - **Scaling:** Uniformly scales the image between 80% and 100% of its original size ($0.8 \rightarrow 1.0$).
    - **Translation:** Shifts the image vertically by up to 10 pixels ($0 \rightarrow 10$).
    - **Rotation:** Rotates the image by up to ±10°.
    - **Shear:** Applies a shear transformation between 0° and 10°.
    - **Interpolation:** Nearest-neighbor interpolation is used.
    - **Mode:** Out-of-bound pixels are filled using the "replicate" mode.

    The augmentation parameters were specifically selected to reflect the controlled imaging environment at MNHN. In this setting, butterfly specimens are meticulously positioned on white backgrounds as straight as possible, although minor alignment errors may occur, justifying the limited ±10° rotation. The camera is positioned perpendicular to the specimen with minimal deviation (reflected in the 0-10° shear range). Since butterflies are already cropped using the YOLOv8 preprocessing pipeline, large translations would risk moving landmarks outside the frame, hence the modest 0-10 pixel translation limit. These carefully constrained augmentation parameters simulate the realistic variability in specimen positioning while preserving the anatomical integrity essential for landmark detection.
3. **Keypoint Handling:**
    - Keypoints are defined in (y, x) format.
    - Keypoints are not removed even if they move out of the visible frame
      However, as explained in the previous paragraph, most transformations are set so that both butterfly and landmarks stay within the bounds of the image.

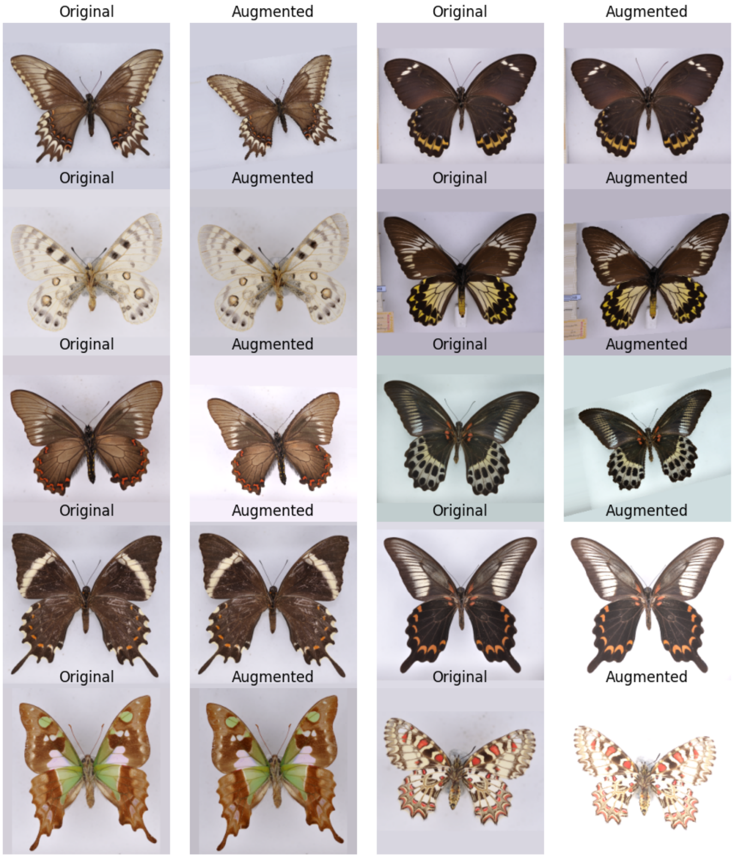


Figure 4.3: *Comparison between original butterfly specimens (left column, labeled 'Original') and their corresponding augmented versions (right column, labeled 'Augmented'). The 5×4 grid demonstrates various augmentation techniques applied during training, including brightness/-contrast adjustments, minor rotations (±10°), scaling (80-100%), translations (0-10 pixels), and shear transformations (0-10°).*

4.4 Heatmap Generation

The generation of target heatmaps for training is a critical aspect of our methodology and varies depending on the U-Net model variant. As outlined in our research objectives, we explored different approaches to leverage the correlation between anatomically connected landmarks, particularly those joined by wing venation patterns.

4.4.1 Landmark Grouping Strategies

In addition to assigning each landmark to its own output channel (the conventional approach), we implemented various grouping strategies where multiple landmarks share output channels. This design encourages the model to learn the structural relationships between landmarks connected by wing veins, potentially improving generalization across butterfly species with varying wing morphologies.

For example, one grouping configuration we tested was (also see in Table 4.3):

```
[[0, 18], [1, 16, 17], [2, 15], [3, 14], [4, 13],
 [5, 12], [6, 11], [7, 9, 10], [8]]
```

In this configuration, landmarks that are anatomically related through wing venation patterns are grouped together. For instance, landmarks 2 and 15, which define opposite ends of a primary wing vein, are assigned to the same output channel, encouraging the model to recognize their spatial relationship.

We also explored more complex grouping strategies with overlapping assignments, where certain landmarks appear in multiple output channels. This redundancy provides the model with multiple perspectives on the same landmark within different anatomical contexts, in an effort to achieve robust landmark detection and avoid outliers.

4.4.2 Heatmap Construction

After image preprocessing and augmentation, we generate target heatmaps for each batch:

1. For each defined landmark group, we create a separate output channel
2. Within each channel, we place heat spots at the corresponding landmark coordinates
3. The intensity distribution around each landmark follows an exponential decay function

Following Kumar et al. [8], we implemented the double exponential (Laplace) distribution function rather than Gaussian distribution for generating the heatmaps. Their comparative analysis demonstrated that double exponential decay functions yield superior landmark localization accuracy in similar detection tasks. This distribution creates sharper peaks with steeper gradients, facilitating more precise local maxima identification using maxpooling operations during landmark detection. Simultaneously, its smooth tail provides sufficient gradient information for the U-Net architecture to build confidence in heatmap regression tasks.

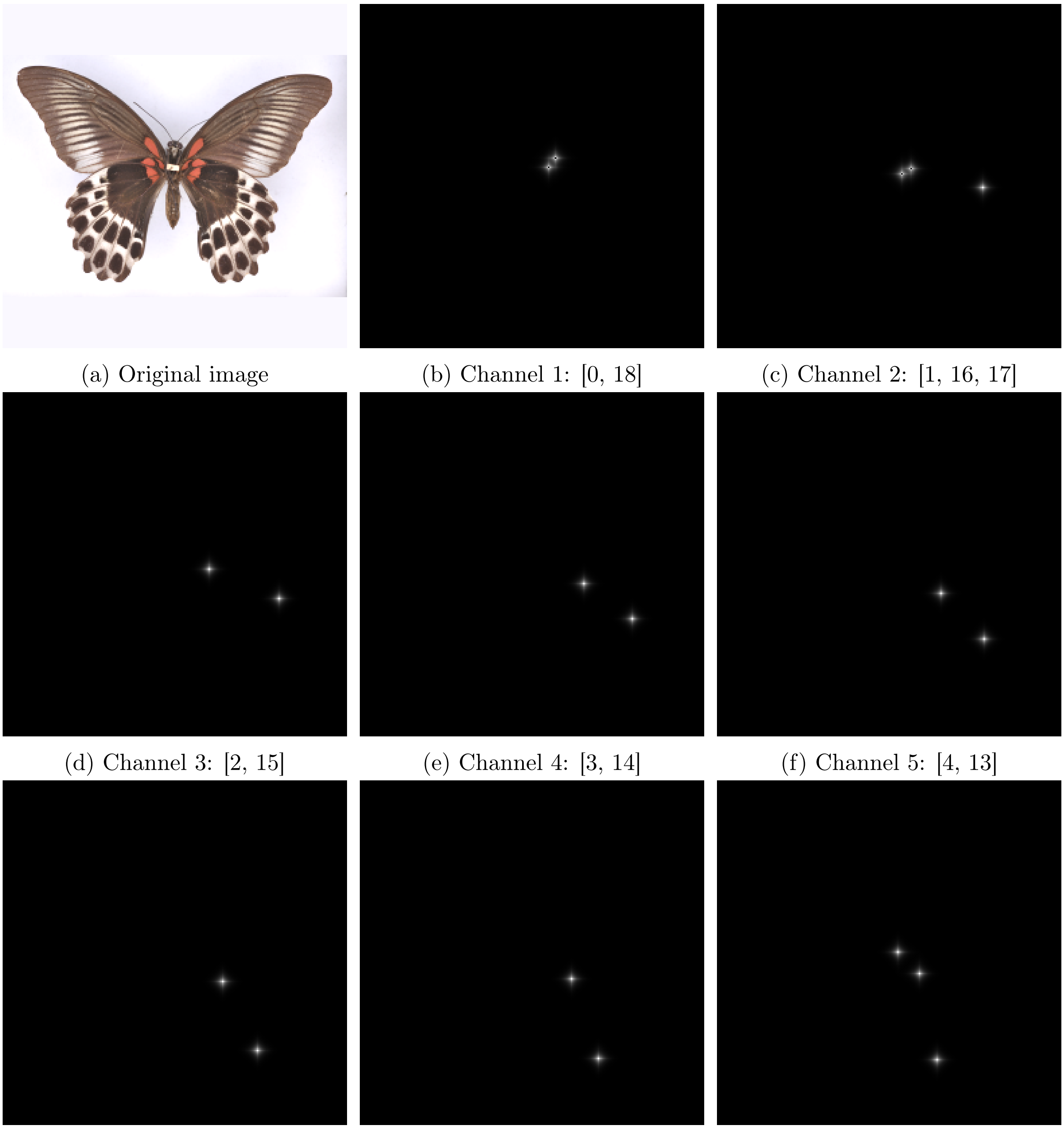
Mathematically, for a landmark at position (μ_x, μ_y), the heatmap intensity at any pixel (x, y) is defined as:

$$FE(x, y) = A \cdot \exp\left(-\frac{\log(2)}{2\sigma}\left(|x - \mu_x| + |y - \mu_y|\right)\right)$$

Where A represents the peak amplitude and σ controls the spread of the distribution. Increasing σ widens the heat region around each landmark, creating a more gradual falloff from the peak. This approach ensures that closely positioned landmarks maintain distinct peaks in the combined heatmap, preserving the localization information for each individual landmark.

When multiple landmarks are assigned to the same channel, their exponential distributions are combined by summing the values at each pixel:

$$H_{\text{channel}}(x, y) = \sum_{i \in \text{landmarks}} H_i(x, y)$$



(a) Original image

(b) Channel 1: [0, 18]

(c) Channel 2: [1, 16, 17]

(d) Channel 3: [2, 15]

(e) Channel 4: [3, 14]

(f) Channel 5: [4, 13]

(g) Channel 6: [5, 12]

(h) Channel 7: [6, 11]

(i) Channel 8: [7, 9, 10]

Figure 4.4: *Visualization of generated heatmaps for a butterfly image using the landmark grouping strategy. The top-left figure shows the original preprocessed image, while the remaining subfigures display eight of the nine output channels, each containing heatmaps for anatomically related landmarks, as indicated by their indices.*

4.5 Interpretation of Heatmap Outputs and Post-Processing

4.5.1 From Heatmaps to Landmark Coordinates

The trained U-Net model produces output heatmap channels where each pixel's intensity represents the confidence of a landmark existing at that location. These heatmaps typically contain several characteristics:

1. Background regions with low signal (values close to zero)
2. "Heat" regions (high signal) centered around predicted landmark locations
3. A faint butterfly silhouette or residual signal that persists due to skip connections between the input image channels and output channels

Within each heat region, a local intensity maximum forms, with the pixel having the highest value representing the most likely position of the corresponding landmark. These observations guide our post-processing approach for extracting precise landmark coordinates from the continuous heatmap representations.

4.5.2 Post-Processing Algorithm

To convert both the ground truth and the predicted heatmap outputs into discrete landmark coordinates and evaluate the model's performance, we implemented the following algorithm:

1. Let N be the number of ground truth landmarks.
2. Let G be the grouping matrix adopted for the model.
3. Let V be the vector of ground truth landmark coordinates of size N.
4. For each output channel i:
   (a) Identify the indices of landmarks assigned to this channel as $G[i]$.
   (b) Let B be the number of landmarks in $G[i]$ (i.e., $B = |G[i]|$).
   (c) Detect all local intensity maxima in the heatmap channel.
   (d) Select the B highest local maxima, representing the regions with highest confidence.
   (e) Match each selected local maximum coordinate to the nearest ground truth landmark found in $G[i]$, starting with the landmark of smallest index and proceeding sequentially.
   (f) Store these matched coordinates in a prediction vector V_{pred}, initialized with $(0, 0)$ at all positions except the indices in $G[i]$.
5. Apply this process to all channels defined in G.
6. For landmarks that appear in multiple channels (in overlapping grouping strategies), average their coordinates across all instances to obtain a final prediction.

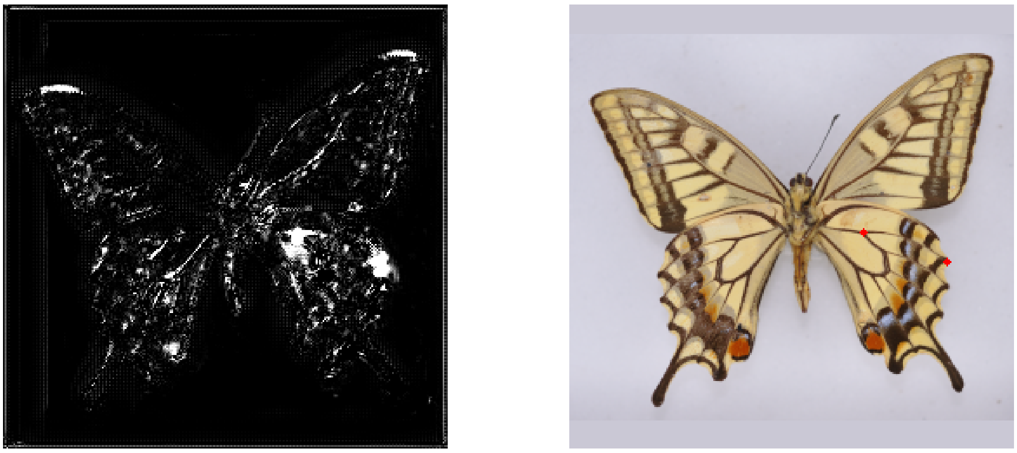


Figure 4.5: Channel output for landmarks [2, 15]

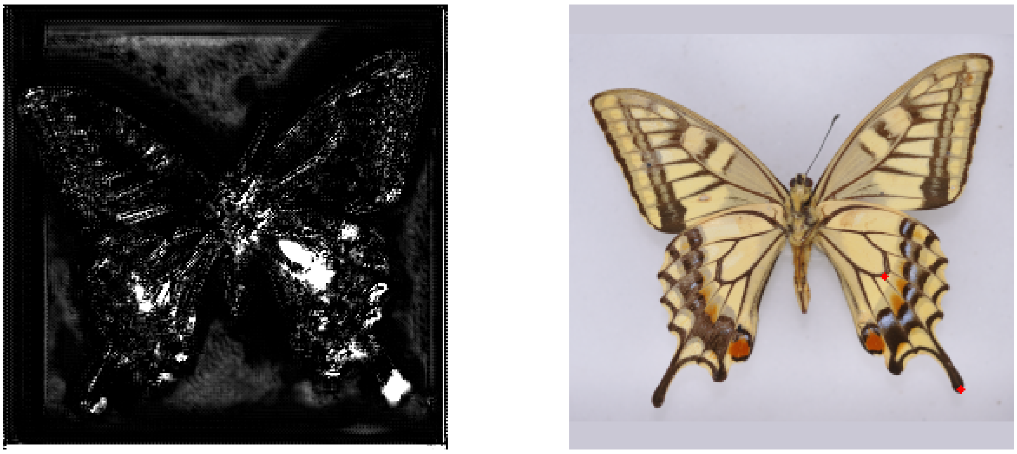


Figure 4.6: Channel output for landmarks [4, 13]

Figure 4.7: *Examples of output heatmap channels and their conversion to predicted landmarks. The leftmost images show sample output channels, where brighter regions indicate potential landmark locations. Note that for visualization purposes, the heatmap values (originally 0-1) are denormalized to 0-255, which exaggerates both the apparent "heat" in less relevant areas and the butterfly silhouette that persists due to skip connections. Despite this visual amplification, the post-processing algorithm correctly identifies the true local maxima and extracts the landmark coordinates shown in the rightmost image.*

This approach ensures that the number of predicted landmarks matches the ground truth, facilitating direct comparison while maintaining consistency through a sequential matching process that starts with the lowest landmark index. For landmarks appearing in multiple output channels (in overlapping grouping strategies), averaging their positions produces a single consensus prediction that reduces the impact of outliers. Importantly, the methodology inherently penalizes output channels that produce either fewer distinct peaks than expected landmarks or spurious high-intensity regions far from true landmark locations.

4.6 Different Models Tested

To evaluate the most effective approach for butterfly wing landmark detection, we explored multiple model configurations, varying input/output dimensions, loss functions, and landmark grouping strategies. For all experiments described in this section, we focused exclusively on the 19 true landmarks of the hind wings, although our dataset contains both true and semi-landmarks for both forward and hind wings. This narrowed experimental scope allowed for more controlled comparison between different model architectures and training strategies.

Our experimentation followed an iterative process, beginning with basic approaches and progressively refining our methodology based on initial findings.

4.6.1 Exploration Process and Configuration Rationale

We began by implementing the most basic grouping approach with all landmarks mapped to a single output channel. This proved problematic as heat zones for nearby landmarks overlapped, creating connected regions that compromised the precision of local maximum detection.

On the opposite end of the spectrum, we implemented a model with one output channel per landmark, following the approach established by Marganne et al. [7].

These limits led us to explore a middle ground: assigning multiple landmarks to each channel based on anatomical relationships. We hypothesized that grouping landmarks connected by wing veins would encourage the model to learn structural correlations while reducing computational requirements.

We also investigated the potential benefits of overlapping assignments, where the same landmark can appear in multiple channels with different contextual landmarks. Additionally, we tested different loss functions and resolution scales to optimize performance.

4.6.2 Model Configurations

Based on our exploratory process, we systematically evaluated five distinct model configurations, summarized in Table 4.1.

Table 4.1: Overview of Model Configurations Tested

Input/Output Size	Loss Function	Landmarks per Channel	Output Channels	Grouping Strategy
256×256	MSE	1 (baseline)	19	One-to-one mapping (conventional approach)
256×256	MSE	Multiple (no overlap)	9	Anatomically related landmarks grouped together
256×256	FBCE	Multiple (no overlap)	9	Same grouping as above, different loss function
256×256	MSE	Multiple (with overlap)	18	Landmarks appear in multiple channels with context
512×512	MSE	Multiple (no overlap)	9	Higher resolution version of second configuration

4.6.3 Configuration Details

Baseline Model (256×256, One Landmark per Channel)

The baseline model follows the conventional approach where each landmark is assigned its own output channel. This results in 19 output channels, one for each landmark. We used MSE loss, which is standard for heatmap regression tasks. This configuration serves as our reference point for evaluating the proposed grouping strategies.

Table 4.2: Landmark mapping for single-landmark model (19 channels)

Channel	**Landmark**	**Channel**	**Landmark**
1	[0]	11	[10]
2	[1]	12	[11]
3	[2]	13	[12]
4	[3]	14	[13]
5	[4]	15	[14]
6	[5]	16	[15]
7	[6]	17	[16]
8	[7]	18	[17]
9	[8]	19	[18]
10	[9]		

Anatomical Grouping (256×256, No Overlap)

This configuration tests our hypothesis that grouping anatomically related landmarks encourages the model to learn meaningful structural relationships. We reduced the number of output channels to 9 by grouping landmarks connected by wing veins. Each landmark appears in exactly one output channel.

Table 4.3: Landmark mapping for multi-landmark model without overlap (9 channels)

Channel	**Landmarks**
1	[0, 18]
2	[1, 16, 17]
3	[2, 15]
4	[3, 14]
5	[4, 13]
6	[5, 12]
7	[6, 11]
8	[7, 9, 10]
9	[8]

Focal Binary Cross-Entropy (256×256, No Overlap)

This variant uses the same grouping strategy as above but replaces MSE with focal Binary Cross-Entropy loss. As explained in Section 3.5.2, FBCE is designed to address class imbalance and to focus more on hard-to-classify examples, which may better handle the sparse nature of landmark heatmaps where positive pixels (landmarks) are vastly outnumbered by negative pixels (background).

Overlapping Grouping (256×256, With Overlap)

This configuration explores the potential benefits of redundancy by allowing landmarks to appear in multiple output channels. This provides the model with different anatomical contexts for the same landmark, potentially improving robustness through ensemble-like prediction averaging.

Table 4.4: Landmark grouping matrix for the overlapping model (18 channels)

Channel	Landmarks	Channel	Landmarks
1	[0, 1, 2, 17]	10	[7, 9, 10]
2	[1, 2, 3, 15]	11	[0, 18]
3	[2, 3, 4, 14]	12	[8]
4	[3, 4, 5, 13]	13	[1, 16, 17]
5	[4, 5, 6, 12]	14	[8, 11, 14]
6	[5, 6, 7, 11]	15	[9, 10, 14]
7	[0, 1, 18]	16	[16, 15]
8	[6, 7, 10]	17	[9, 12]
9	[8, 10, 15]	18	[9, 13]

Higher Resolution (512×512, No Overlap)

This model tests whether increased spatial resolution improves landmark localization accuracy. By doubling the input and output dimensions to 512×512 pixels, the model can potentially capture finer anatomical details. We maintained the same non-overlapping grouping strategy as the second configuration to isolate the effect of resolution.

4.6.4 Training Protocol

All models were trained using the same dataset split, data augmentation strategy, and preprocessing pipeline described in previous sections. We used the RMSprop optimizer with an initial learning rate of $1e-3$ and reduced it by a factor of 0.1 when validation loss plateaued for 10 epochs. In other times, we used a manual control of the learning rate, depending on the observed behavior of the training and validation loss plot. We implemented checkpointing to save the best performing model based on validation metrics. Training continued until 100 epochs have passed, or no improvement was observed in validation performance for 15-20 consecutive epochs, at which point we would stop training. In some cases, we would later resume training from the best checkpoint to explore further improvement possibilities.

The computational infrastructure for training included both local and cloud resources. For the 256×256 models, we utilized a laptop equipped with an AMD Ryzen 9 7940HS processor and NVIDIA GeForce RTX 4070 Laptop GPU with approximately 5.5 GB of memory available for computations. The more memory-intensive 512×512 models were trained on Paperspace cloud computing platform using GPUs with 16GB memory (A4000, P5000). All implementations were developed using TensorFlow with GPU acceleration. Experiment tracking was managed through Weights & Biases (wandb).

The Cytomine platform played a beneficial role in our research pipeline, providing specialized tools for analyzing high-resolution butterfly images and landmark annotations. We leveraged its visualization capabilities for examining morphological details that would be difficult to discern in standard image viewers. Additionally, Cytomine housed the previous Marganne project dataset, which facilitated the integration of Morpho genus specimens with our Papilionidae dataset for the combined-data experiment (see Section 5.7), enabling seamless data management across both butterfly families.

Each model was evaluated using the post-processing and the custom metrics described in Section 4.7 to ensure fair comparison across different configurations.

4.7 Design and Implementation of Custom Metrics for Landmark Localization

4.7.1 Briefly

Generic loss functions such as MSE or BCE provide a global measure of model performance but lack spatial interpretability, which is crucial in tasks such as facial or biological landmark detection and medical imaging. To address this, we designed and implemented a set of custom metrics in ***TensorFlow/Keras*** that measure prediction errors in various spatial contexts.

4.7.2 Implementation Considerations

Despite implementing these metrics in TensorFlow/Keras using tensor operations exclusively, their integration into the training pipeline introduced significant computational overhead and memory requirements. Attempts to optimize performance by limiting metric calculation to the validation subset (10% of total images) and reducing evaluation frequency to every fourth epoch after the initial 20 epochs proved insufficient to resolve these constraints. The original intention was to observe whether the spatial error between ground truth and predicted landmarks continued to decrease during later training epochs when the model had approached convergence but validation loss (MSE/FBCE) still showed minor improvements. Due to these computational limitations, these metrics were ultimately applied to the testing dataset after model training completion rather than during the iterative training process.

4.7.3 Motivation

The development of these custom metrics was motivated by several analytical requirements:

It's important to translate the differences in signal intensity between predicted and actual heatmaps into clear spatial errors, which helps in understanding performance effectively. We also need to make it easier to compare different models with varying input and output resolutions, like 256×256 and 512×512, by examining landmark mismatches in their original butterfly dimensions. Additionally, we should consider image scale factors, such as the pixel-to-millimeter ratio, when assessing prediction accuracy, especially in morphometric studies. Using normalization techniques is valuable to allow meaningful comparisons of spatial errors across specimens of various sizes. Finally, tracking the error distributions for each landmark across the dataset can provide analytical benefits, enabling us to refine models and identify any configurations that might need additional attention.

4.7.4 Implementations

Distance Pixel

Purpose: Measures Euclidean distances in **pixel space** at model output resolution, serving as a baseline metric.

Methodology: The methodology involves two steps. First, coordinates of the converted ground truth and predicted heatmaps are taken in. Second, the Euclidean distance between the predicted and true landmark coordinates is computed.

Original Distance Pixel

Purpose: Measures Euclidean distance between predicted and ground-truth landmarks in pixels at original image size.

Methodology: First, the method takes in the ground truth and predicted landmarks coordinates at the downscaled model output size, along with original image dimensions (Yolo V8 bounding box dimensions). Then, it converts the coordinates to the original output size using the appropriate transformations. Finally, it calculates the distances between the new ground truth landmark coordinates and predicted landmark coordinates.

Original Distance Millimeter

Purpose: Measures Euclidean distance between predicted and ground-truth landmarks in **millimeters** at original image size.

Methodology: This method takes in distances between ground truth and predicted landmarks calculated at the original image size. It converts these distances to millimeters using each image's scale factor.

Original Distance Normalized

Purpose: This process normalizes landmark distances relative to the dimensions of images, which ensures that comparisons can be made across varying sizes of butterflies.

Methodology: The first step involves normalizing both the ground truth and predicted landmarks. Following this, the Euclidean distance between these normalized coordinates is computed.

4.7.5 Return properties of each metric

1. Maintain per-landmark tracking of errors.
2. Store intermediate results dynamically.
3. Compute:
    - *Average per-landmark distance* for targeted analysis.
    - *Overall mean distance* as a model performance indicator.
    - *Stacked distances for visualizing their distributions.*

By integrating these custom metrics, we enhance the interpretability and reliability of deep learning models in landmark localization tasks.

Chapter 5

Results and Analysis

This chapter presents the experimental results and analysis of our butterfly landmark detection models. We conducted five major experiments to evaluate different aspects of our approach: landmark grouping strategies, loss functions, input resolution effects, cross-species generalization capabilities, and visualization of prediction accuracy. Each experiment provides insights into optimizing landmark detection across diverse butterfly specimens.

5.1 Landmark Grouping Strategies

Our initial experiments evaluated different landmark grouping strategies and their impact on model performance. We compared the baseline approach (one landmark per channel) with various anatomically-guided grouping strategies, examining both training efficiency and prediction accuracy.

5.1.1 Single-landmark (19 channels) vs. Multi-landmark (9 channels)

Training Convergence

Figure 5.1 illustrates the significant difference in learning dynamics between the single-landmark (Table 4.2) and multiple-landmark (Table 4.3) models.

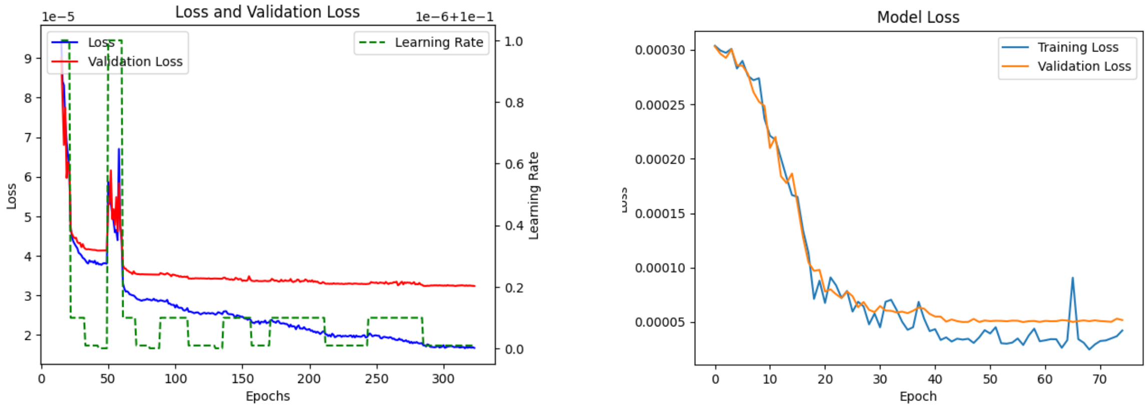


Figure 5.1: *Training convergence comparison between single-landmark model (19 channels, left) and multiple-landmark model with no overlap (9 channels, right).*

The multiple-landmark model in Figure 5.1 demonstrates significantly faster convergence, reaching stable loss values in approximately 45 epochs, while the single-landmark model continues to show gradual improvement even after 200 epochs. This indicates how anatomical grouping of landmarks substantially reduces training time by enabling more efficient learning of wing structure relationships.

This indicates that a more concise model generalized well on unseen specimens from the same butterfly family, much faster, all while maintaining comparable performance. This fourfold reduction in training time illustrates the efficiency gained by incorporating domain knowledge into the model architecture through strategic landmark grouping.

Prediction Accuracy Comparison

Despite the substantial difference in training efficiency, both models achieved comparable landmark localization accuracy, as shown in Figure 5.2. For the rest of the plots, kindly refer to Section 6 of the appendix.

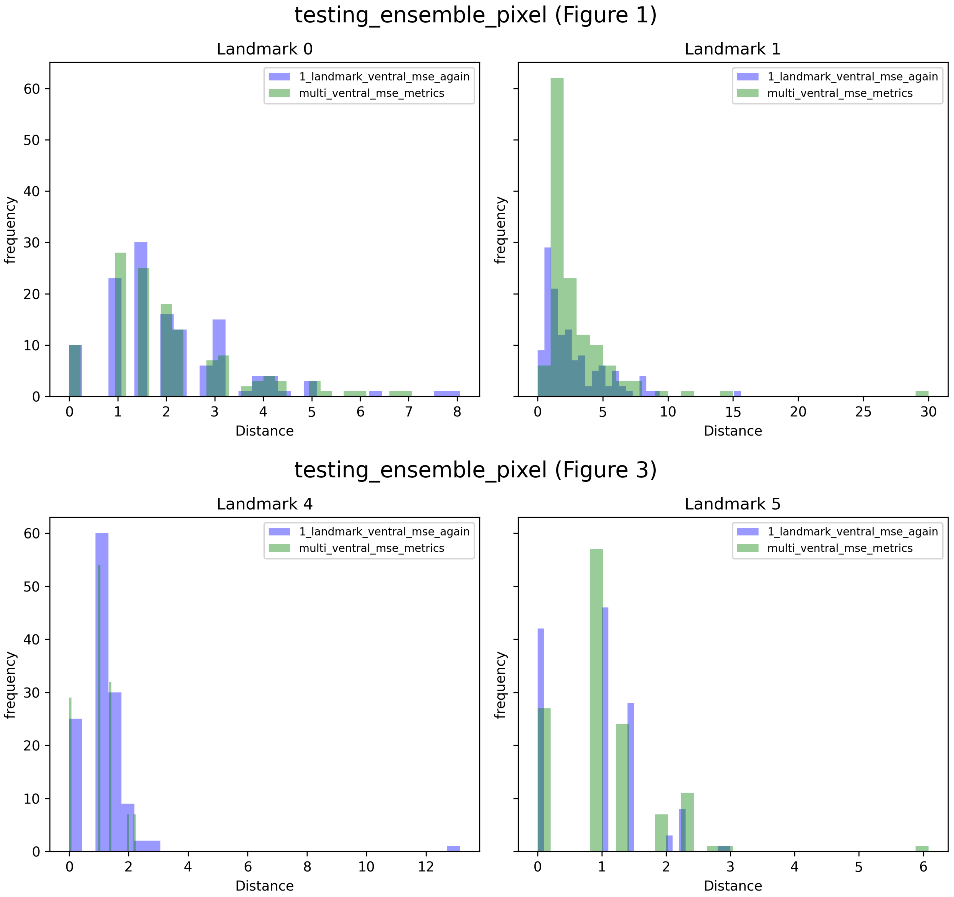


Figure 5.2: *Comparison between multi-landmark (9 channels) model and single-landmark (19 channels) model error distributions for landmarks 0, 1, 4, and 5. For the remaining landmarks, see Section 6 in Appendix. Both models show similar error patterns measured in pixels at 256×256 output resolution, though the multi-landmark model occasionally exhibits more outliers.*

The histograms display the 'distance pixel' metric distributions for selected landmarks, with the single-landmark model shown in blue and the multi-landmark model in green. Both distributions generally follow a Poisson-like pattern with most predictions falling within a small error range (1-3 pixels at 256×256 resolution).

Detailed analysis reveals that the multi-landmark model's error distributions typically exhibit longer tails, indicating slightly more frequent outlying predictions. This suggests that while the anatomically-grouped 9-channel model converges substantially faster, this efficiency comes with a minor trade-off in prediction consistency. The single-landmark model, with each output channel

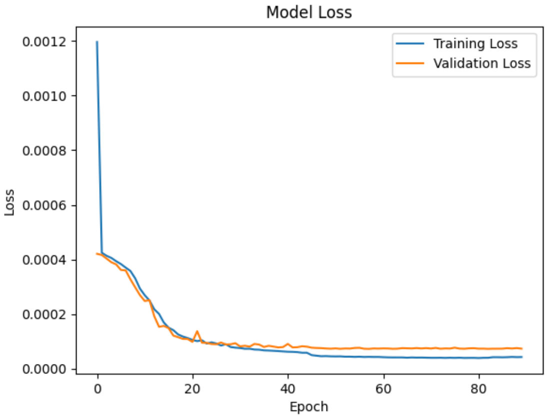


Figure 5.3: *Training convergence for the ensemble (overlapping) model, showing comparable speed to the non-overlapping multi-landmark model (approximately 45 epochs). This was contrary to our expectation that the overlapping model would require significantly more training epochs due to its increased complexity.*

specialized for a specific landmark, demonstrates marginally more robust localization with fewer extreme outliers across the testing dataset.

These results indicate that researchers can make an informed choice between computational efficiency and marginal improvements in outlier reduction, depending on their specific application requirements. For applications requiring rapid model development or deployment on resource-constrained systems, the multi-landmark approach offers substantial advantages with minimal accuracy trade-offs.

5.1.2 Multi-landmark Models: Non-overlapping vs. Overlapping Configurations

We further investigated whether allowing landmarks to appear in multiple output channels could improve prediction accuracy through ensemble averaging. The overlapping model used 18 output channels with a complex landmark grouping strategy (see Table 4.4), resulting in landmarks appearing in multiple channels, with the frequency distribution as follows:

Table 5.1: Frequency of landmark appearances across output channels

Landmark	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Channel count	3	4	3	3	3	3	3	3	3	4	4	2	2	2	3	3	2	2	2

Each landmark appeared across multiple anatomically relevant groupings. For example, landmark 1 appeared in four different output channels alongside various combinations of landmarks 0, 2, 16, and 17, which are all located in proximity on the wing.

Training Convergence

The overlapping model's training convergence rate was comparable to the non-overlapping configuration, despite the increased complexity of the prediction task. This suggests that the anatomical relationships encoded in the overlapping grouping strategy may have provided structural constraints that facilitated efficient optimization during training.

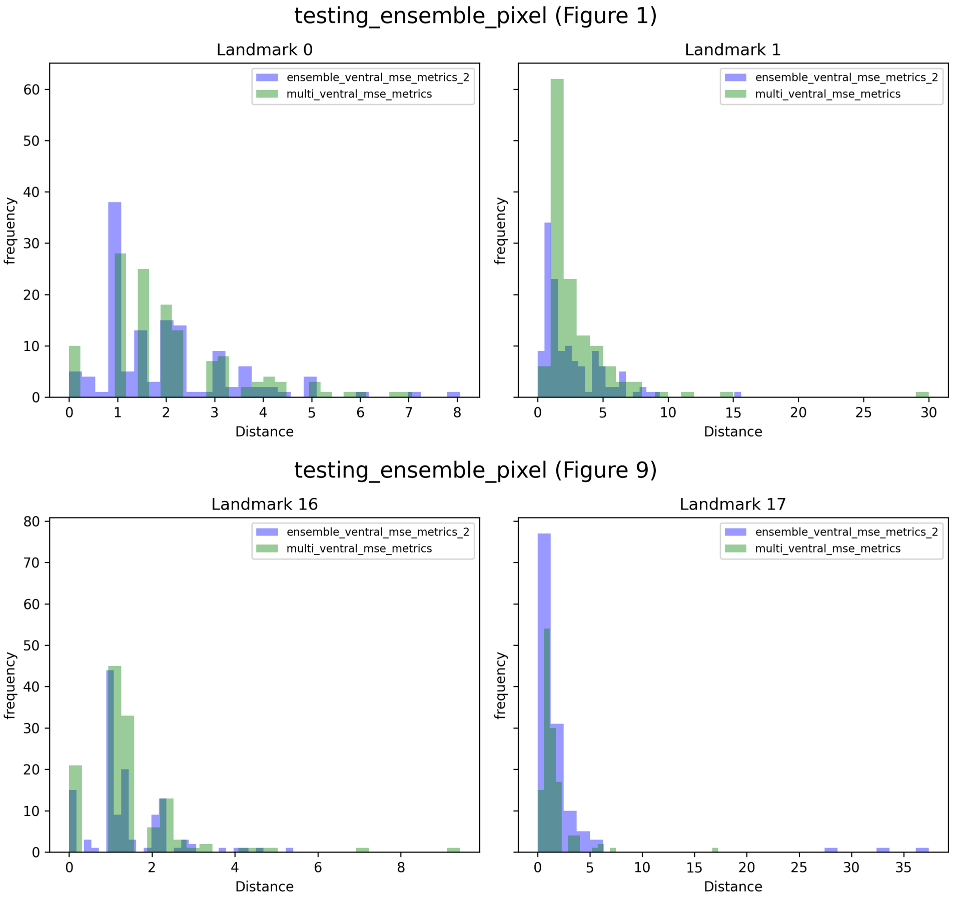


Figure 5.4: *Comparison between multi-landmark without overlap (9 channels) model and multi-landmark with overlap (18 channels) model error distributions for landmarks 0, 1, 16, and 17. Both models show similar error patterns measured in pixels at 256×256 output resolution.*

Prediction Accuracy Comparison

Contrary to our hypothesis, the overlapping model did not demonstrate improved accuracy over the non-overlapping configuration. Error distributions remained similar between both models, with outliers present in both but affecting different landmarks. The overlapping model showed more frequent outliers for certain landmarks, as evident in the complete set of error distribution plots in Section 6 of the appendix.

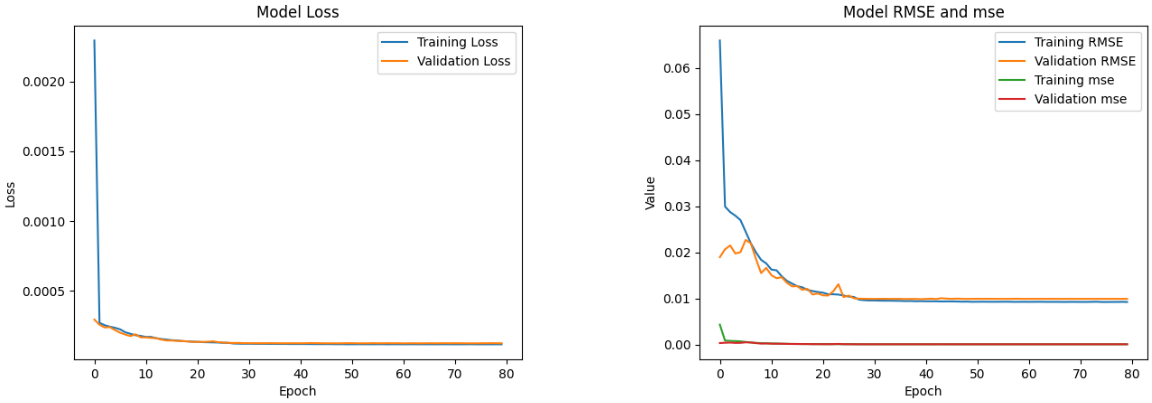
This unexpected result can be attributed to a fundamental limitation of our ensemble approach: when certain output channels fail to generate a clear heat zone for a landmark, they essentially assign default coordinates (typically at the origin), which substantially skews the averaged position. A high-quality prediction in one channel can be compromised by poor predictions in other channels containing the same landmark.

This finding suggests that while landmark prediction averaging has theoretical benefits, its practical implementation requires more sophisticated mechanisms for outlier detection and rejection. The non-overlapping configuration ultimately provides a better balance between model complexity and prediction accuracy, maintaining comparable performance with a more computationally efficient architecture.

5.2 Impact of Loss Functions on Landmark Detection Accuracy

Our second experiment examined the effect of different loss functions on model performance while maintaining a consistent architecture and landmark grouping strategy. We selected the 9-channel non-overlapping landmark configuration for its efficient convergence and compared MSE against FBCE loss.

Training Convergence



(a) FBCE loss

(b) MSE and RMSE metrics

Figure 5.5: *Training and validation loss plots for the model using FBCE loss. Both the FBCE loss (left) and the recorded RMSE metric (right) show convergence after approximately 30 epochs, demonstrating slightly faster convergence than the MSE-based model.*

As shown in Figure 5.5, the FBCE model demonstrated stable training with convergence after approximately 30 epochs, slightly faster than the MSE model's 45-epoch convergence. We next evaluated the prediction accuracy of both approaches using our distance-based metrics.

Prediction Accuracy Comparison

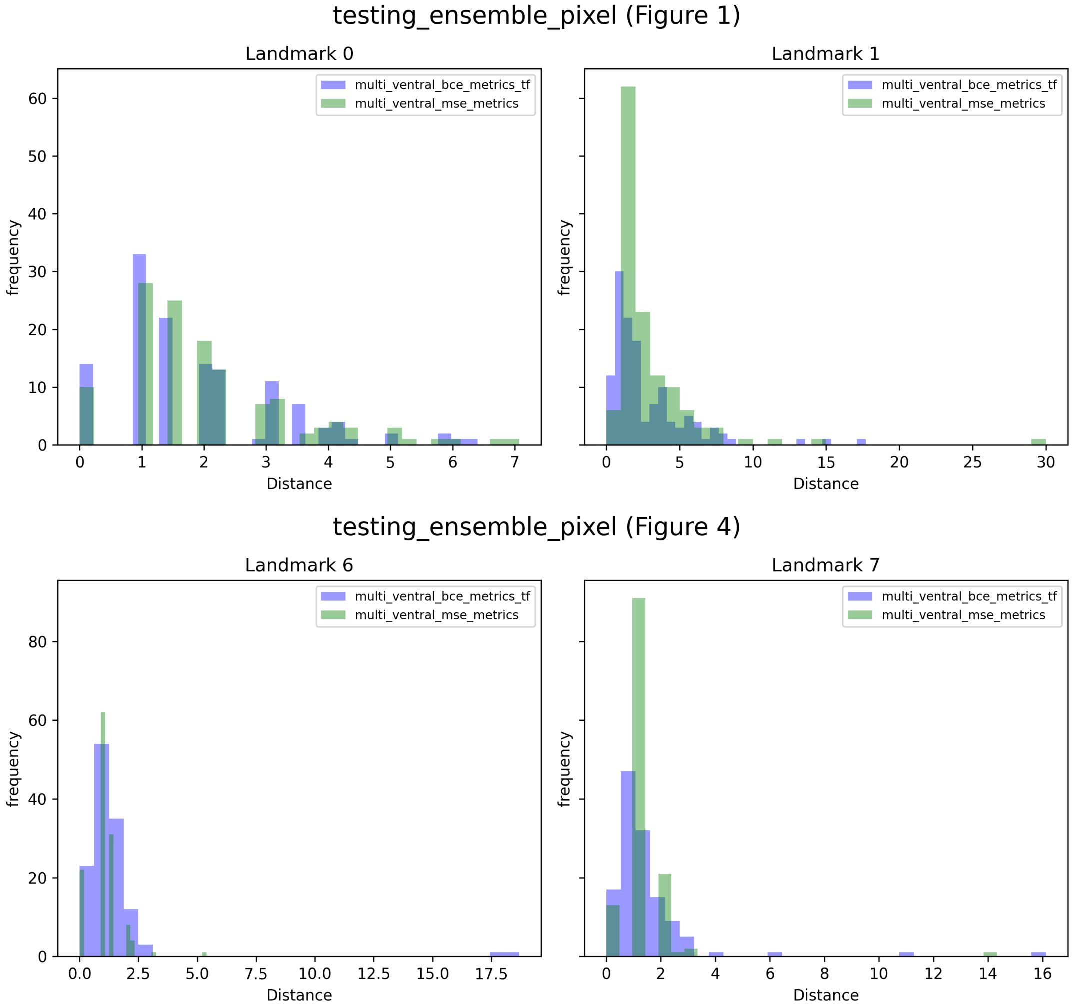


Figure 5.6: *Comparison between error distributions for models trained with MSE loss function (blue) versus FBCE loss function (green). Both models use the same 9-channel architecture with 256×256 pixel input/output resolution.*

Table 5.2 demonstrates that for identical model architecture and landmark grouping, both loss functions produced comparable results with selective advantages. The highlighted cells (green) indicate which loss function performed better for each landmark. Overall, the MSE model performed better on 10 landmarks, while the FBCE model was superior for 9 landmarks.

The error distributions in Figure 5.6 reveal that both models produce occasional outliers, though the FBCE model exhibits more pronounced outliers for some landmarks. Table 5.2 shows two landmarks with a standard deviation greater than two pixels for MSE and four for FBCE.

It is worth noting that FBCE was originally designed to address class imbalance problems and to focus on uncertain predictions. However, in the context of landmark detection, high pixel signal values do not necessarily indicate prediction certainty but rather serve as localization signals to prioritize certain pixels over others in the 2D channel space. This fundamental difference may explain why FBCE did not consistently outperform MSE across all landmarks despite its theoretical advantages for imbalanced data.

For future work, we recommend exploring a weighted BCE loss that specifically addresses class imbalance without the focusing effect that may be detrimental to spatial localization tasks. That would mean assigning the value of zero to the focusing parameter γ in Formula 3.7.

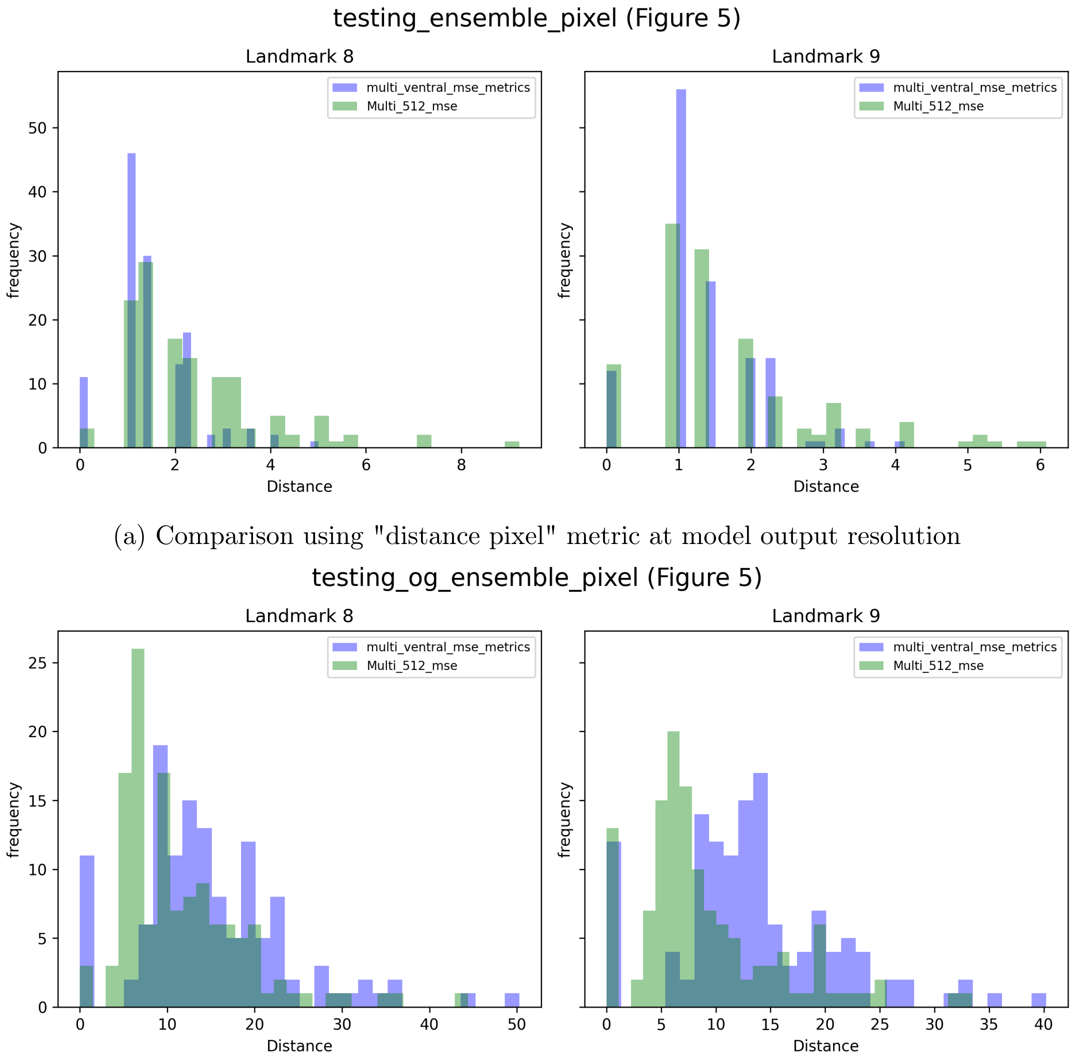
Table 5.2: *Landmark prediction error comparison between Multi-landmark models with MSE (left column) and FBCE (right column) loss functions*

Landmark	**Multi-landmark MSE**		**Multi-landmark FBCE**	
	Mean	Std Dev	Mean	Std Dev
0	2.0882	1.4292	1.9418	1.3910
1	2.7414	3.2891	2.7429	2.7695
2	1.1668	0.8278	1.2765	0.8527
3	0.9878	0.6597	0.9998	0.6651
4	0.9993	0.6360	1.0666	0.6363
5	1.0965	0.8208	1.0777	0.7735
6	1.0801	0.7071	1.3450	2.1987
7	1.3405	1.3035	1.5122	1.7785
8	1.5020	0.8803	1.4368	0.8568
9	1.3575	0.7537	1.2737	0.8340
10	1.2614	1.0371	1.2803	0.9707
11	1.2519	0.9227	1.2419	0.9023
12	1.1177	0.9101	1.1552	0.7868
13	1.1347	0.8208	1.2340	0.9383
14	1.1227	1.0169	1.3574	1.3818
15	1.2584	0.9633	1.5168	1.2976
16	1.4004	1.2483	1.3986	1.1160
17	1.5718	1.8179	2.8489	9.5981
18	2.1604	7.4219	1.9978	4.5223

5.3 Effect of Input/Output Resolution on Landmark Localization Accuracy

Our third experiment investigated whether increased spatial resolution improves landmark localization accuracy. We compared two models with identical architectures (9-channel, non-overlapping landmarks) and loss function (MSE) but different input/output dimensions: 256×256 pixels versus 512×512 pixels. The decision to use the 9-channel architecture was made to reduce model complexity, as the increased input/output dimensions significantly expand the number of learnable parameters, making larger models more challenging to train. Training the higher-resolution model required substantially greater computational resources and training time compared to the 256×256 configuration.

Prediction Accuracy Comparison



(a) Comparison using "distance pixel" metric at model output resolution

(b) Comparison using "original distance pixel" metric at original image resolution

Figure 5.7: *Comparison between models with identical architecture (9 channels, MSE loss) but different input/output resolutions.*

Figure 5.7 presents the comparative error distributions between the 256×256 and 512×512 resolution models for landmarks 8 and 9 (0-based). Panel (a) displays the error distributions at each model's native output resolution, revealing comparable central tendencies but with notable differences in distribution tails. The higher-resolution model exhibits more pronounced tail regions, indicating greater variability in certain predictions despite similar median performance. This pattern changes dramatically in panel (b), which shows the error distributions after transformation to original image dimensions. Here, a systematic leftward shift of the 512×512 model's distribution is evident, demonstrating substantially reduced error magnitudes across most landmarks when assessed at the original scale.

We then calculate the mean and standard deviation of all landmark errors for the two models to view detection quality across all the landmarks.

When using the "distance pixel" metric, which measures error at each model's native output resolution (Table 5.3), the 256×256 model appears superior, with lower mean errors for 17 of 19 landmarks. However, this comparison is misleading because it fails to account for the difference

Table 5.3: Landmark prediction error comparison between 256×256 and 512×512 models using the "distance pixel" metric

Landmark	256x256		512x512	
	Mean	Std Dev	Mean	Std Dev
0	2.0882	1.4292	3.0657	2.4169
1	2.7414	3.2891	6.6836	13.3850
2	1.1668	0.8278	1.5947	1.1612
3	0.9878	0.6597	1.5792	2.7888
4	0.9993	0.6360	1.3954	0.8061
5	1.0965	0.8208	1.3601	0.8323
6	1.0801	0.7071	2.0978	8.8156
7	1.3405	1.3035	2.9294	12.9033
8	1.5020	0.8803	2.3382	1.4935
9	1.3575	0.7537	1.7618	1.2501
10	1.2614	1.0371	1.2368	0.8006
11	1.2519	0.9227	1.3199	0.9161
12	1.1177	0.9101	1.3745	1.3929
13	1.1347	0.8208	1.3371	0.9578
14	1.1227	1.0169	1.6059	1.6689
15	1.2584	0.9633	1.3601	1.3876
16	1.4004	1.2483	1.7885	1.2831
17	1.5718	1.8179	1.6697	1.5709
18	2.1604	7.4219	1.3961	1.4563

Table 5.4: Landmark prediction error comparison between 256×256 and 512×512 models using the "original distance pixel" metric

Landmark	256×256		512x512	
	Mean	Std Dev	Mean	Std Dev
0	20.6085	13.5850	15.1666	11.6218
1	27.8828	31.9305	34.4720	76.8118
2	11.8522	9.0598	8.1456	6.1353
3	10.0022	7.0670	7.9356	13.3517
4	9.9468	6.6765	6.9150	4.1759
5	10.9973	8.2856	6.8560	4.3505
6	10.8918	7.5347	11.0210	50.1221
7	13.5190	13.3180	14.1003	57.4792
8	14.8894	8.7649	11.5353	7.1127
9	13.6248	7.5005	8.9022	6.4323
10	12.4462	10.0497	6.1118	3.8175
11	12.6306	9.3652	6.7290	5.1125
12	11.3593	9.3215	6.8596	6.0228
13	11.3517	8.7905	6.6162	4.1614
14	11.1593	9.7672	8.0070	7.7009
15	12.6157	9.7538	6.7643	5.8740
16	13.8449	11.7642	8.9065	6.2219
17	15.7671	17.9855	8.4209	8.5139
18	22.0717	86.3424	6.9270	6.7092

in spatial resolution.

When errors are transformed back to the original butterfly dimensions using the "original distance pixel" metric (Table 5.4), the results change dramatically. The 512×512 model demonstrates substantially lower mean errors for 16 of 19 landmarks. This finding confirms that higher input/output resolution yields more precise landmark localization when measured at the original image scale.

However, it is essential to note that both models still produce outliers for several landmarks. The increased accuracy observed in the 512×512 model primarily affects well-detected landmarks and does not necessarily translate to improved robustness against catastrophic prediction failures. As evident in the standard deviation values in Table 5.4, the higher-resolution model occasionally produces extreme outliers despite its generally superior performance, particularly for landmarks 1, 6, and 7 (0-based). This suggests that while spatial resolution improves overall precision, it does not fundamentally solve the challenge of occasional severe mispredictions.

5.4 Tabular Comparison of Experimental Model Test Results

This section presents a comprehensive performance comparison of all experimental models using the original distance pixel metric applied to the test dataset. Each row represents an individual butterfly specimen, with performance values color-coded to facilitate visual interpretation. The complete results are distributed across three tables: Part 1 is presented in the main text, while Parts 2 and 3 are provided in Section 6.1 of the appendix for reference.

The tabular overview reveals distinct performance hierarchies among the tested configurations. The 512×512 resolution model demonstrates superior performance across most specimens, though it exhibits occasional extreme outliers that significantly impact its reliability. Conversely, the multi-landmark model with FBCE loss function consistently shows the poorest performance. The remaining models exhibit intermediate performance levels, ranked in ascending order of effectiveness: Ensemble MSE, Multi-landmark MSE, and Single-landmark MSE.

A detailed comparative analysis focusing on 24 common test specimens is presented in Section 5.8.1, where these models are evaluated alongside a generalized model trained on the combined multi-family dataset. This subset analysis provides deeper insights into model behavior patterns and cross-family generalization capabilities.

Image Name	1-LM MSE	Multi MSE	Ensemble MSE	Multi FBCE	Multi MSE 512
EL75574V	367.05	307.26	275.83	391.98	220.22
EL75639V	170.95	183.95	218.92	178.48	95.59
EL43143V	214.57	218.81	273.08	237.68	142.45
AC-C0082V	310.96	227.39	311.99	439.23	170.06
AC-C0064V	369.04	323.09	394.45	323.12	185.62
AC-C0124V	255.54	555.82	231.48	298.18	372.71
EL43132V	367.05	347.80	381.54	436.19	253.06
EL75446V	280.21	488.81	610.42	480.57	393.25
EL72568V	519.84	484.98	503.67	520.67	444.55
AC-C0045V	267.12	267.95	184.45	289.70	142.41
EL72586V	189.02	190.87	167.92	207.64	100.59
AC-C0345V	223.59	221.28	168.37	211.31	154.22
EL72506V	119.33	139.43	136.40	134.10	104.49
AC-C0079V	264.84	269.57	261.37	230.28	207.12
EL75508V	290.42	283.92	305.12	334.41	138.23
EL72443V	320.70	268.42	271.74	249.11	190.70
EL43180V	276.94	506.46	254.25	326.04	354.47
EL72505V	185.09	175.82	158.90	162.73	102.41
EL75317V	202.83	202.80	198.87	274.06	169.65
EL43165V	208.56	219.92	193.43	253.72	164.00
EL72417V	599.37	453.85	515.97	512.74	294.34
EL72481V	145.76	151.59	172.90	159.84	134.81
EL75547V	277.41	255.59	289.14	317.99	167.86
EL75354V	238.76	222.13	247.69	232.78	148.98
EL75706V	190.67	244.17	197.88	211.08	189.77
EL75545V	260.22	249.60	269.91	282.38	177.12
EL75703V	172.74	183.36	188.93	175.31	118.98
EL75581V	368.01	318.97	327.27	594.59	244.37
AC-C0365V	163.95	193.02	167.17	182.81	108.42
EL43190V	311.76	358.33	363.30	494.66	725.22
EL43193V	422.20	353.80	461.27	341.04	186.83
AC-C0358V	133.10	110.22	106.99	75.86	79.08
EL75609V	289.84	341.33	311.54	376.21	232.80
EL75401V	225.10	158.40	553.26	487.74	137.15
EL72740V	214.99	238.55	246.88	264.94	176.57
AC-C0320V	212.17	290.80	289.12	199.43	402.01
EL75301V	143.86	174.60	163.75	164.32	104.64
AC-C0326V	195.35	214.12	203.22	224.62	198.88
AC-C0066V	423.21	468.87	518.74	534.47	299.38
EL75531V	379.54	324.70	360.45	347.15	266.59
AC-C0311V	334.77	299.92	281.72	958.03	159.06
EL75314V	276.86	380.25	343.78	300.63	159.13
EL75709V	220.92	236.39	215.74	198.05	198.24

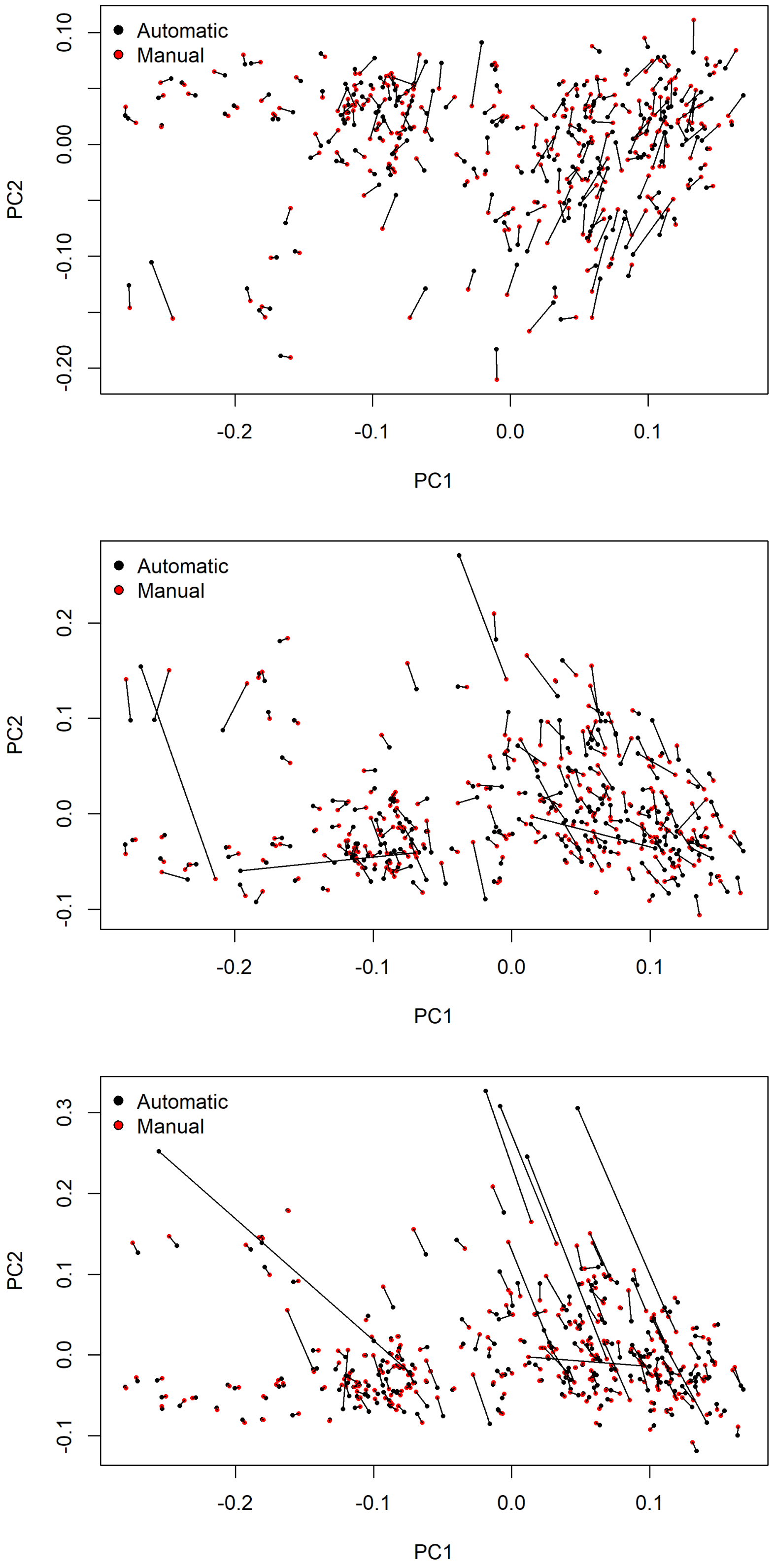
Table 5.5: *Cumulative landmark localization error per test butterfly specimen across all experimental models using the original distance pixel metric. (Part 1)*

5.5 Multivariate Analysis of Landmark Prediction Patterns

To gain deeper insights into how our models capture the overall geometric patterns of butterfly wing landmarks, we conducted a multivariate analysis using Principal Component Analysis on both ground truth and predicted landmark coordinates; Professor Vincent Débat provided the template for the analysis.

Our approach involved using the trained models to predict landmark coordinates for both validation and testing datasets, then stacking the ground truth and predictions as rows. This created a matrix of 516 rows (129 validation + 129 testing specimens × 2 for ground truth and predictions) and 38 columns (19 landmarks × 2 coordinates). Before dimensional reduction, we performed a Generalized Procrustes Analysis to superimpose the landmark configurations, removing differences in size, position, and orientation to focus analysis purely on shape variation.

We then applied PCA to this matrix and plotted the data points on the first two eigenvectors, which capture the principal axes of variation in the dataset. Two points in this space represent each specimen, one for ground truth and one for prediction, connected by a segment that visualizes the prediction error in the reduced dimensional space.



Single-landmark model
(19 channels)
(256×256 resolution)

Multi-landmark model
(9 channels)
(256×256 resolution)

Multi-landmark model
(9 channels)
(512×512 resolution)

Figure 5.8: *2D plots in the first principal plane of PCA analysis for each model configuration, showing the distribution of predicted coordinates (red points) relative to ground truth landmarks (black points). The visualizations compare the single-landmark model, multi-landmark model at 256×256 resolution, and multi-landmark model at 512×512 resolution.*

The PCA visualizations in Figure 5.8 show how predicted landmark configurations (black points) relate to ground truth configurations (red points) in the space defined by the first two principal components. These components typically capture the most significant sources of geometric variation in the dataset.

Several key observations can be made from these visualizations:

1. All three models capture the overall landmark configuration pattern, with predictions generally clustered near their corresponding ground truth positions in the principal component space.

2. The high-resolution 512×512 model demonstrates the most tightly clustered predictions around ground truth values, which indicates a superior preservation of the overall geometric relationships among landmarks. However, this model also presents several data points with the most extreme outlying predictions.

3. The single-landmark model shows more consistent preservation of landmark relationships compared to the 256×256 and the 512×512 multi-landmark models, with few extremely long segments connecting automatic to manual points.

4. The directionality of the segments connecting ground truth to predicted points warrants further investigation. Consistent directional patterns across multiple specimens would indicate systematic bias in specific landmark predictions. Future work could employ formal segment angle analysis to identify which landmarks contribute most significantly to prediction errors and potentially reveal underlying structural biases in the model's feature extraction capabilities. For example, one could investigate the correlation of these angles with landmarks 1, 6, and 7 (0-based) for the 512×512 model as described in 5.3

These multivariate patterns reinforce our previous findings: the high-resolution model provides the most accurate geometric representation, while the single-landmark model offers better consistency in maintaining anatomical relationships.

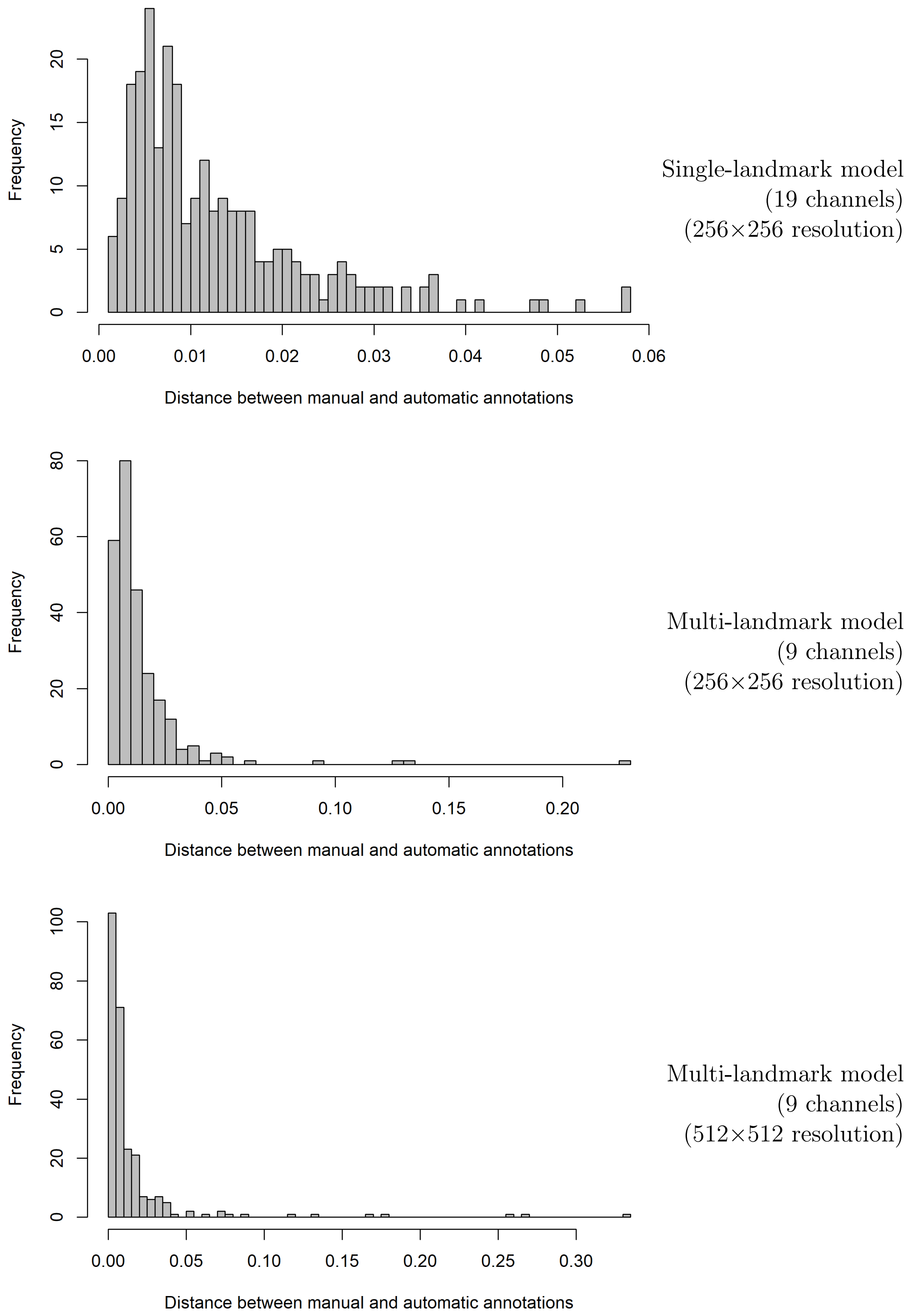


Figure 5.9: *Histograms showing the distribution of Euclidean distances between predicted and ground truth landmark configurations in PCA space. The distributions demonstrate how each model performs in maintaining overall geometric relationships between landmarks.*

The histograms in Figure 5.9 quantify the Euclidean distances between predicted and ground truth landmark configurations in the reduced PCA space. The 512×512 model exhibits a distribution with the smallest mean distance and narrowest spread, confirming its superior ability to maintain overall wing geometry. The single-landmark model shows a slightly broader distribution but with fewer extreme outliers compared to the 256×256 multi-landmark model, which displays the widest spread of distances.

These multivariate analyses demonstrate that our evaluation of model performance should consider not only the accuracy of individual landmark predictions but also how well the models preserve the geometric relationships between landmarks.

5.6 Cross-species Application: Testing Generalization to Morpho Genus

To investigate generalization capabilities across butterfly families, we applied our Papilionidae-trained models to specimens from the Morpho genus, which exhibit substantially different wing morphology, venation patterns, and landmark configurations.

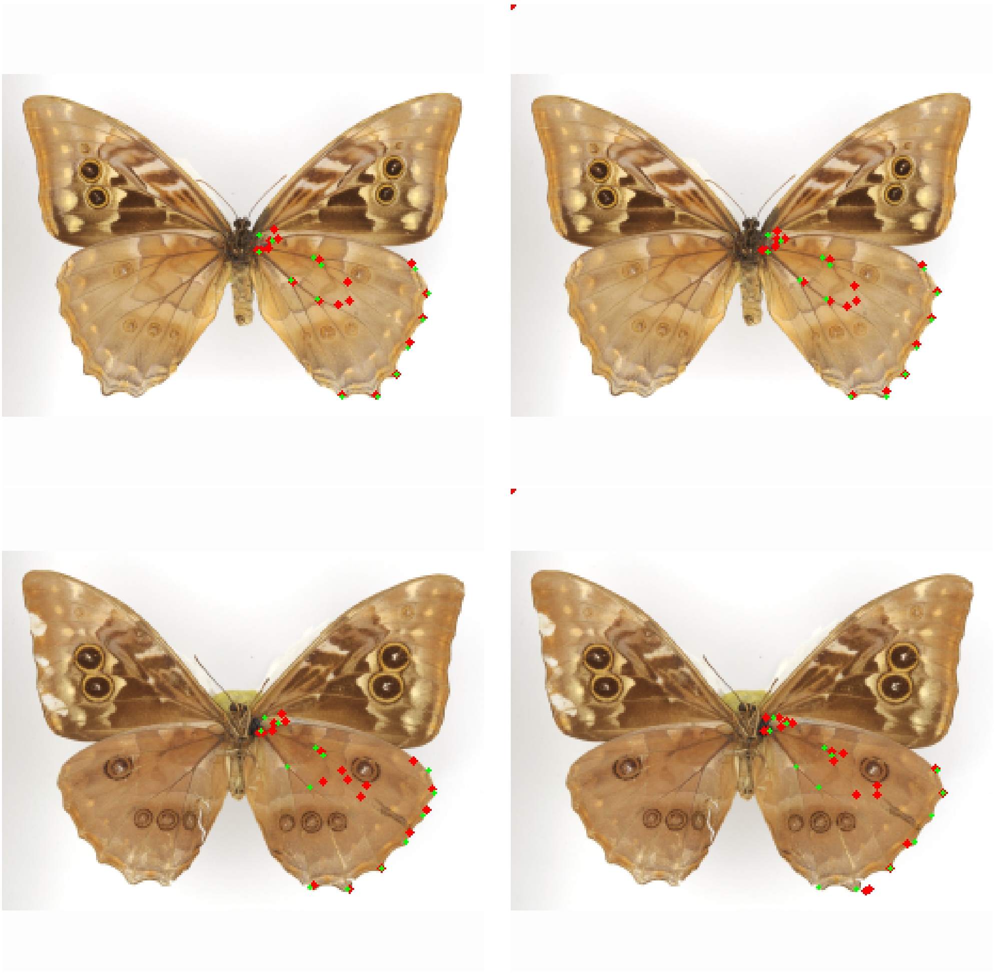


Figure 5.10: *Cross-species testing results showing the application of Papilionidae-trained models to Morpho genus specimens. Left column: Single-landmark model predictions; Right column: Multi-landmark model predictions. Both models struggled to generalize effectively across families, with significant localization errors on most landmarks.*

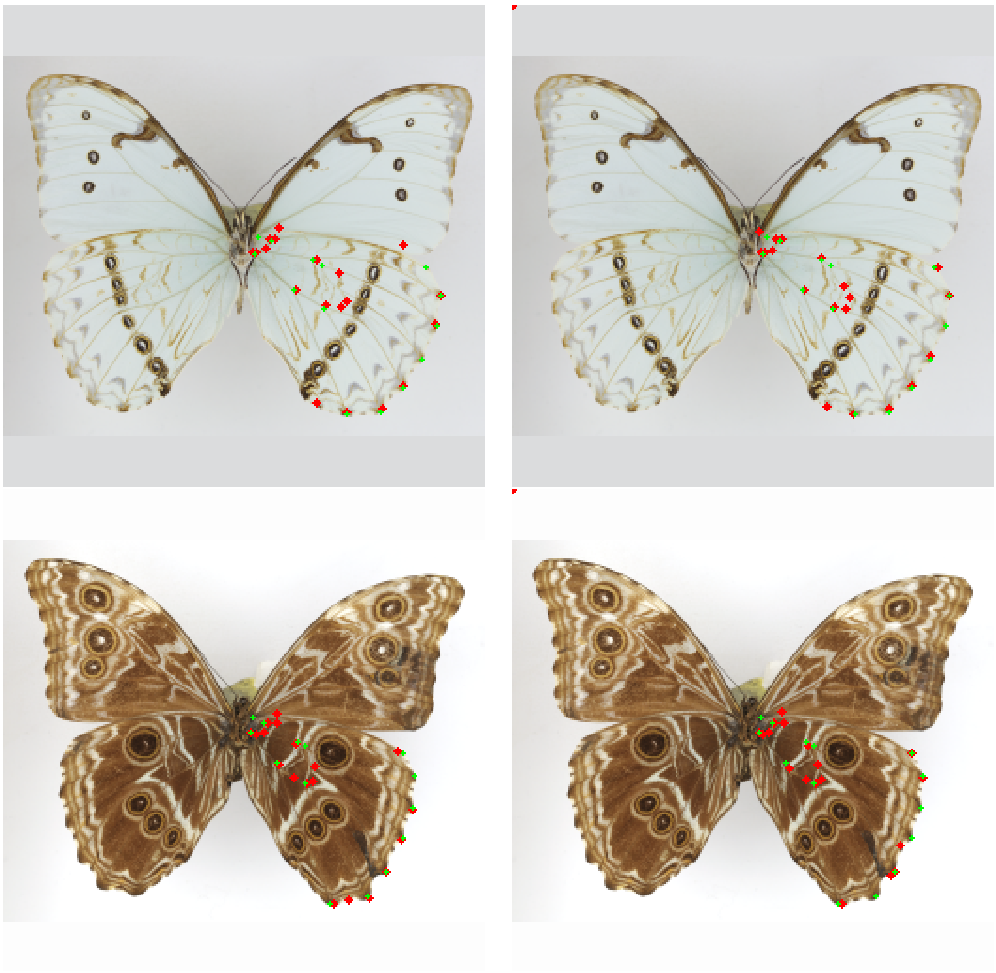


Figure 5.11: *Additional examples of cross-species testing on Morpho specimens. Both models demonstrate limited transferability, with some landmarks detected while most others are completely misplaced.*

The visual results in Figures 5.10 and 5.11 reveal significant limitations in cross-family generalization. Both the single-landmark and multi-landmark models failed to accurately localize most landmarks on Morpho specimens. Certain landmarks that correspond to relatively conserved morphological features across families (such as some wing margin points) showed better, though still imprecise, localization. However, landmarks associated with family-specific venation patterns were systematically misplaced.

These results align with observations made by Professor Vincent Débat, who applied the models developed by Marganne et al. [7] (trained on Morpho genus) to other butterfly families and found poor generalization. The fundamental differences in wing morphology, venation patterns, and landmark distributions between Papilionidae and Morpho specimens present significant challenges for direct cross-family application.

This experiment underscores the need for either family-specific models or testing if and how one of our previous model would be capable of accommodating taxonomic variation in morphological features.

5.7 Integrated Cross-family Model: Training on Combined Datasets

Our final experiment investigated whether a unified model could effectively detect landmarks across different butterfly families. We combined datasets from both Papilionidae and Morpho genus specimens, creating a dataset of 2126 ventral images of butterflies with variable landmark

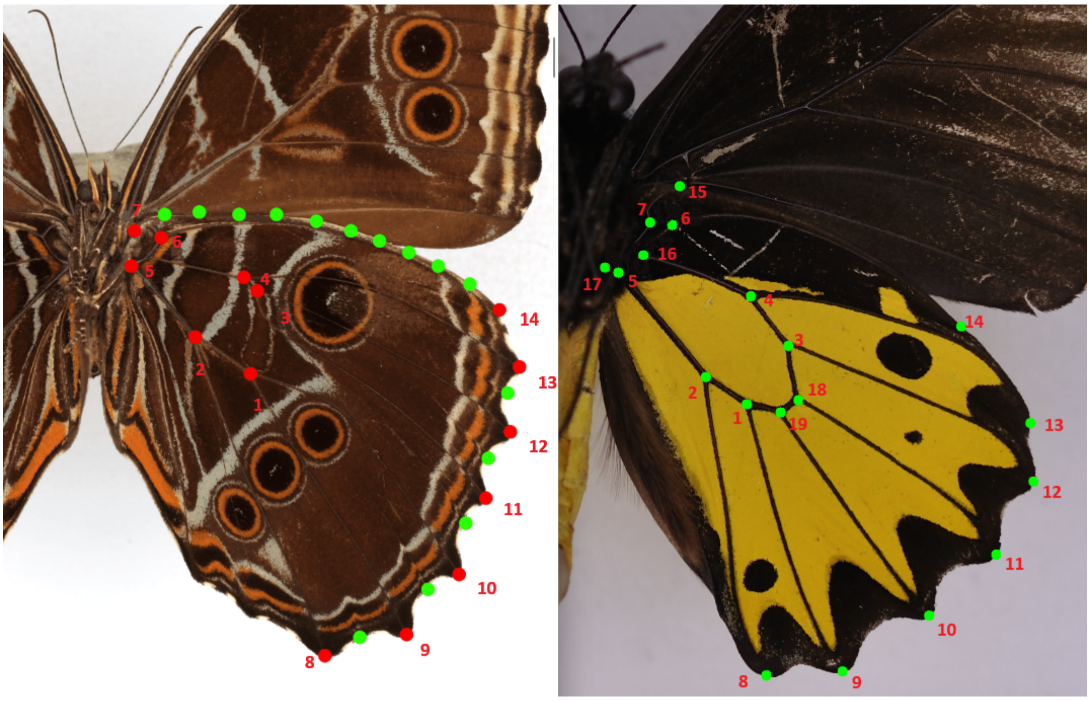


Figure 5.12: *Anatomical landmark mapping between butterfly families. Left: Morpho genus landmark indices (1-14). Right: Papilionidae family landmark indices (1-19). The first 14 landmarks were matched based on homologous anatomical structures, while landmarks 15-19 are exclusive to Papilionidae.*

configurations split into 1289 images from the 19-landmarks Papilionidae and 837 images from the 14-landmarks Morpho. The training, validation, and testing sets are composed respectively of 1700, 213, and 213 ventral images.

5.7.1 Dataset Integration Methodology

We performed careful anatomical mapping between the 14 landmarks on Morpho genus specimens and their corresponding homologous structures in Papilionidae. For the 5 landmarks unique to Papilionidae, we implemented a null-coordinate strategy during training:

- Papilionidae landmark permutation: [12, 11, 15, 16, 10, 18, 19, 8, 7, 6, 5, 4, 3, 2, 1, 17, 9, 14, 13] (based on indices from Figure 1.2)
- Morpho genus mapping with null-coordinates: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 0, 0, 0, 0] (based on indices found on Cytomine Research and not that of Marganne et al.'s report as in 1.6.)

For Morpho specimens, landmarks with no corresponding match in Papilionidae (indicated as "0" in the mapping) were assigned null coordinates (0,0) during training. When generating heatmaps from these coordinates, no heat zones were produced for these padded positions, effectively creating a model that could adapt to variable numbers of landmarks depending on the butterfly family.

5.7.2 Single-landmark Channel Model Performance

As shown in Figure 5.13, the combined model achieved convergence after approximately 50 epochs, significantly faster than the 200+ epochs typically required for single-landmark-per-

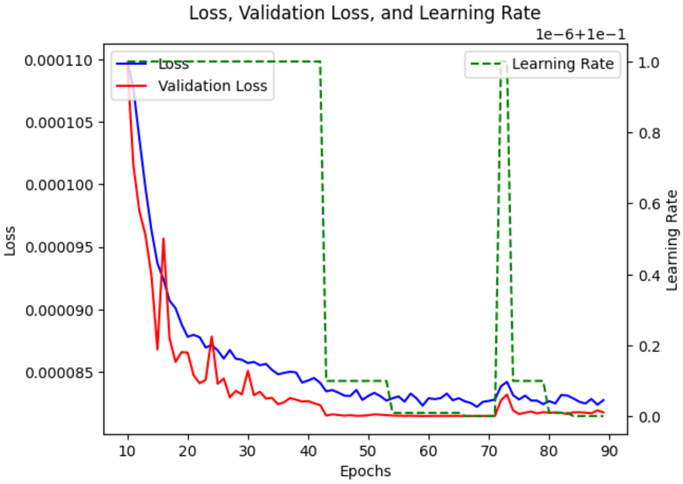


Figure 5.13: *Training and validation MSE for the single-landmark model (19 channels) trained on the integrated dataset containing both Papilionidae and Morpho genus specimens. The model converged after approximately 50 epochs.*

channel models on homogeneous datasets. This efficiency is unexpected and probably foreshadows an issue in the results.

Analysis of the error distributions in Figure 5.14 reveals a distinctive pattern: landmarks 8-14 (1-based), which represent genuinely homologous structures shared between both butterfly families, demonstrate notably accurate predictions. In contrast, the model performed poorly on two categories of landmarks: those that were matched between families despite having somewhat different anatomical contexts (landmarks 1-7) and those unique to Papilionidae (landmarks 15-19).

The model appears to have effectively learned to predict no landmarks in the last 5 channels for Morpho specimens, as evidenced by the high frequency of zero error in these channels when tested on Morpho specimens. This indicates successful adaptation to the variable landmark count across families.

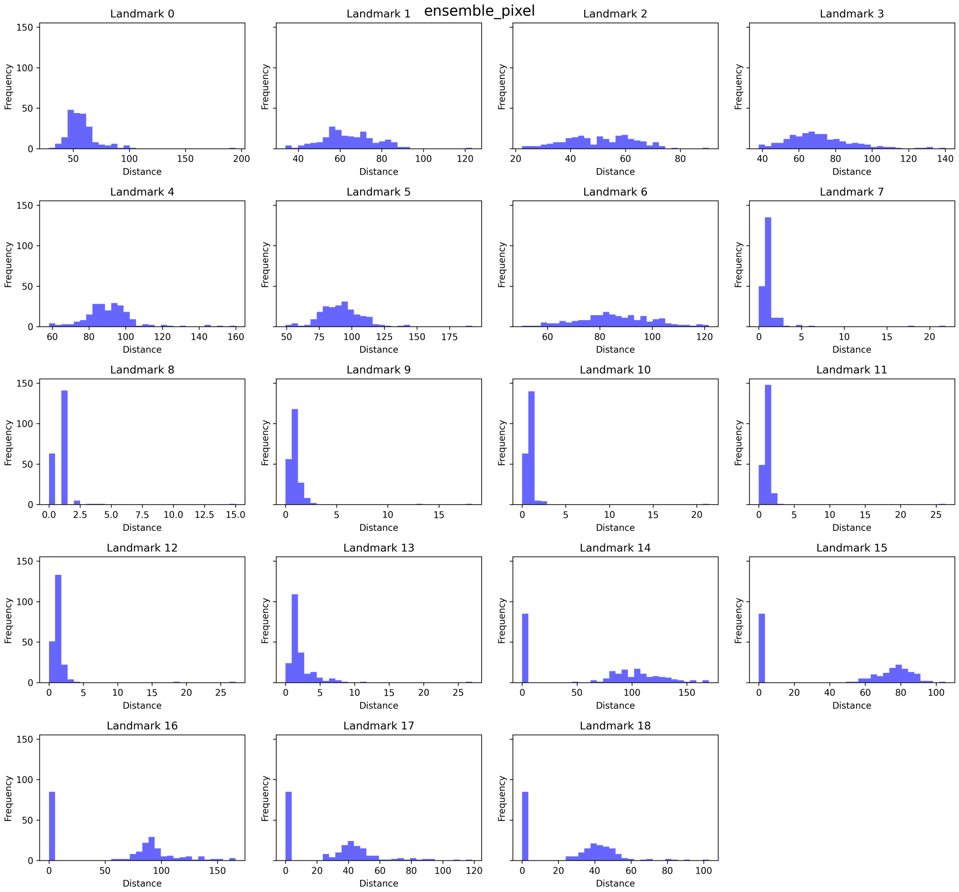


Figure 5.14: *Distribution of prediction errors across landmarks for the combined model. Landmarks 7-13 (corresponding to homologous structures between both families) show substantially lower error distances compared to other landmarks, indicating uneven learning across different landmark types.*

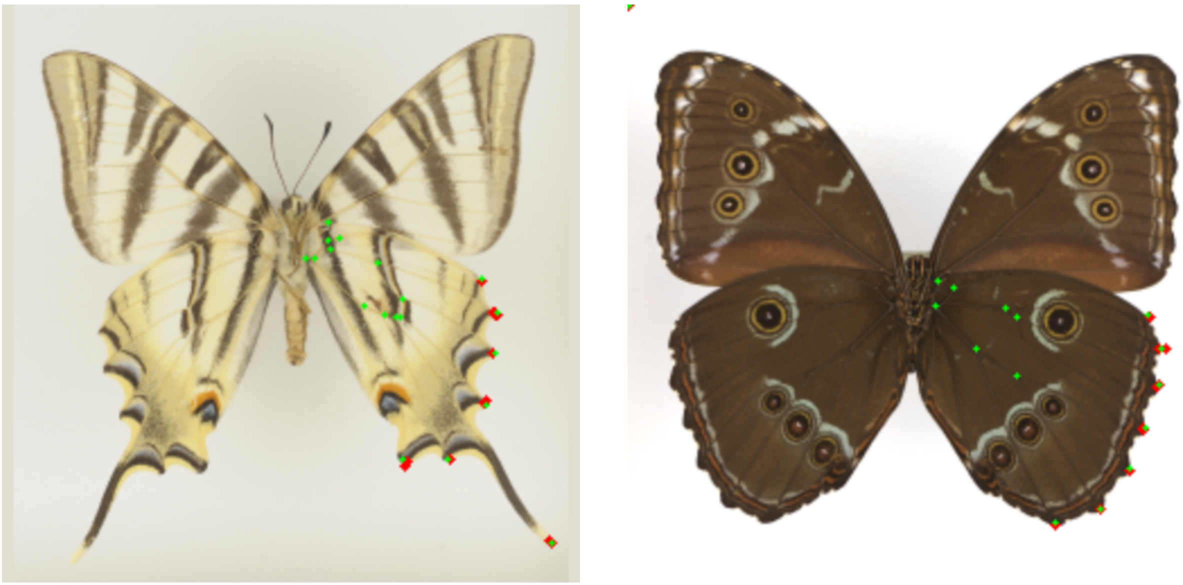


Figure 5.15: *Predictions from the combined 19-channel model. Left: Papilionidae specimen with 19 landmarks. Right: Morpho specimen with 14 landmarks. The model shows variable accuracy depending on landmark position, failing completely for all landmarks that are not on the wing's outer edge.*

5.7.3 Multi-landmark Channel Model Performance

Building on insights from previous experiments, we implemented a multi-landmark configuration with strategic grouping focused on anatomically correlated landmarks:

Table 5.6: Landmark grouping for cross-species Multi-landmark model (13 channels)

Channel	Landmark	Channel	Landmark
1	[2, 8]	8	[5]
2	[1, 9]	9	[6]
3	[10, 19]	10	[7]
4	[11, 18]	11	[15]
5	[3, 12]	12	[16]
6	[4, 13]	13	[17]
7	[14]		

This approach reduced the number of output channels from 19 to 13, decreasing model complexity while maintaining the ability to capture important anatomical correlations.

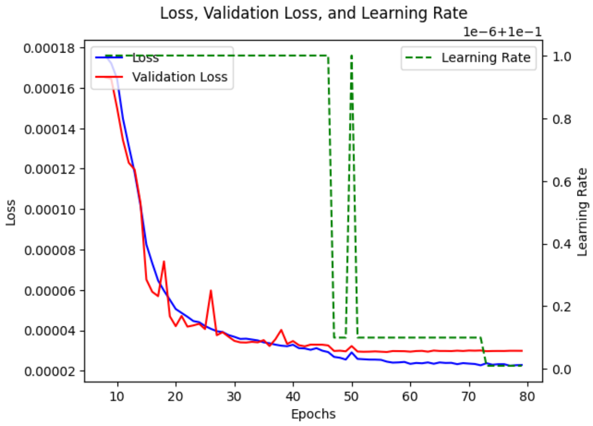


Figure 5.16: *Training and validation loss for the multi-landmark combined model (13 channels). Convergence was achieved after approximately 50 epochs, comparable to the single-landmark combined model despite the reduced channel count.*

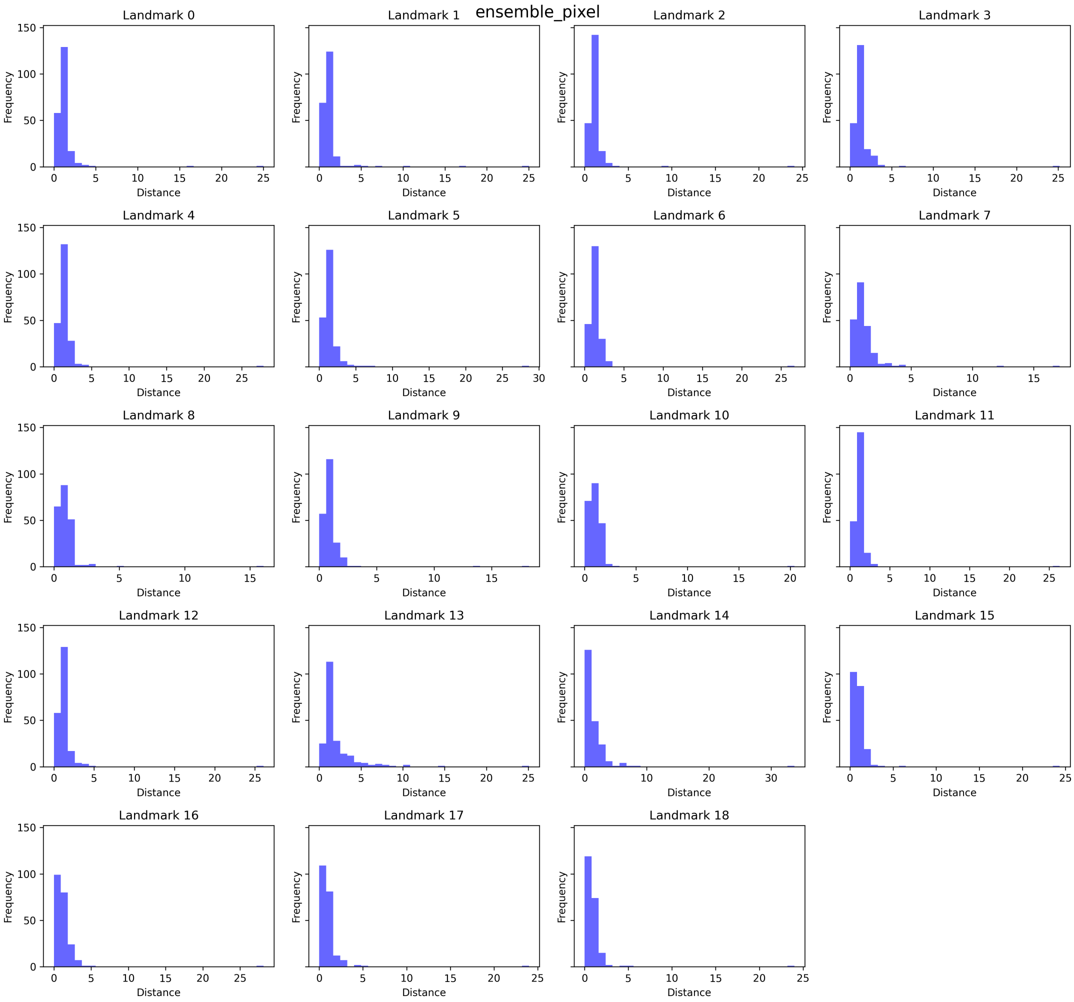


Figure 5.17: *Error distribution for the multi-landmark combined model across all 19 landmarks. The model demonstrates significantly improved performance across all landmarks compared to the single-landmark combined model, with fewer outliers and more consistent error distributions.*

The multi-landmark combined model exhibited superior performance when compared to the single-landmark approach. The strategic grouping enabled the model to capture essential correlations between landmarks, while the reduced model size facilitated a more efficient learning process. This configuration achieved consistent accuracy across all landmarks for both butterfly families, with significantly low error rates. The mean value for localization error is around 1 or 2 pixels for most landmark indices, and the spread is well below 5 for all landmarks. We can also state that the 25-pixel outlying distance recorded in all landmarks originates from an image with erroneous ground truth coordinates. It can be asserted that the model has effectively learned to implicitly classify the family of butterflies and predict the appropriate number of landmarks accordingly. Additionally, some credit may be attributed to the increased dataset size, which undoubtedly contributed to the model's enhanced generalization capabilities.

The success of this approach demonstrates that anatomically-informed landmark grouping can enhance cross-family generalization by helping the model focus on fundamental structural relationships shared across taxa, while still accommodating family-specific variations in landmark count and distribution.

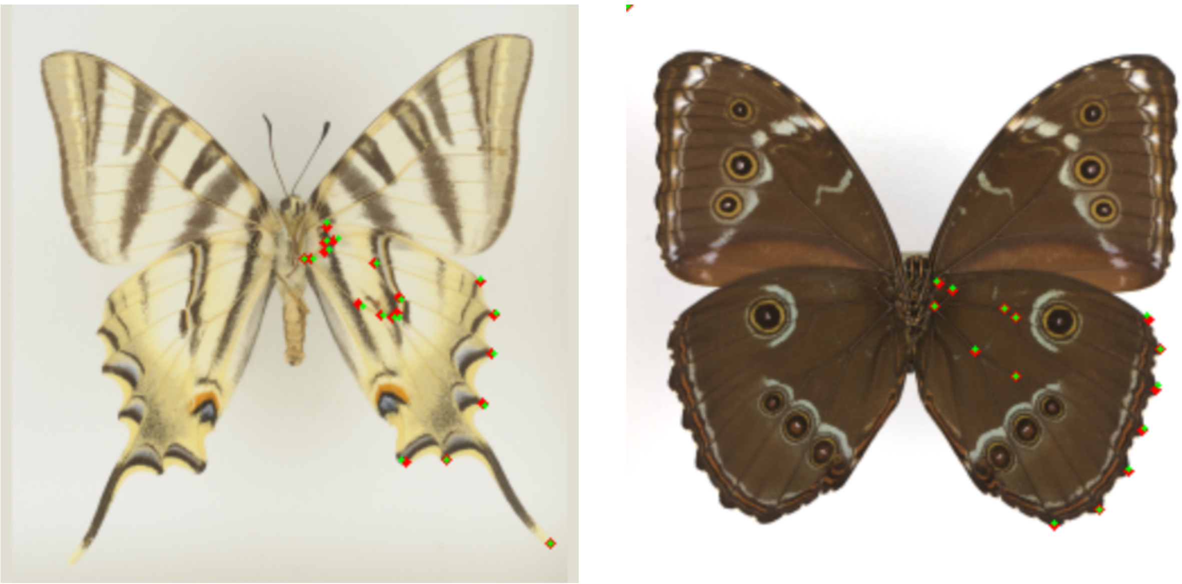


Figure 5.18: *Predictions from the combined 13-channel model. Left: Papilionidae family specimen with 19 landmarks. Right: Morpho genus specimen with 14 landmarks. The model shows very satisfying results in both cases.*

5.8 Comparative Model Performance Analysis

This section presents a comprehensive comparison of model performance across 24 test images that were common to both the Papilionidae dataset and the combined multi-family dataset. The analysis employs both quantitative metrics and qualitative visual assessment to evaluate landmark detection accuracy across different architectural configurations.

5.8.1 Quantitative Performance Comparison

Image Name	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
AC-C0052	277.40	262.82	232.21	268.44	152.07	224.73
AC-C0066	423.21	468.87	518.74	534.47	299.38	472.49
AC-C0129	166.53	197.46	190.90	284.41	121.46	199.21
AC-C0185	228.74	217.68	546.32	295.03	175.01	226.17
AC-C0236	147.69	156.70	163.18	161.52	109.83	177.36
AC-C0246	148.39	163.26	160.81	182.77	100.69	143.11
AC-C0247	246.75	305.69	277.06	314.85	115.04	275.95
AC-C0268	178.13	190.53	160.02	149.72	102.26	184.69
AC-C0311	334.77	299.92	281.72	958.03	159.06	260.76
AC-C0320	212.17	290.80	289.12	199.43	402.01	198.38
EL38900	190.02	161.56	134.83	169.74	97.08	158.28
EL43113	196.05	250.48	205.76	240.07	151.90	248.34
EL43145	305.49	277.11	280.04	257.82	151.18	232.69
EL43172	191.09	148.67	186.74	156.47	117.09	182.91
EL43229	256.53	285.76	284.83	706.34	144.85	268.27
EL72560	197.17	195.05	164.82	217.69	132.26	207.15
EL72597	268.56	263.47	217.09	288.42	148.66	237.81
EL72724	258.84	240.37	259.37	239.40	179.87	242.33
EL75301	143.86	174.60	163.75	164.32	104.64	158.70
EL75531	379.54	324.70	360.45	347.15	266.59	322.93
EL75581	368.01	318.97	327.27	594.59	244.37	280.78
EL75599	253.83	280.17	252.59	277.12	149.76	280.95

Table 5.7: Cumulative landmark localization error per test butterfly specimen across all experimental models, plus the 'combined' model, using the original distance pixel metric

The color-coded performance matrix enables systematic evaluation of model effectiveness across test specimens. Values are normalized using a color gradient where green indicates best performance (lowest error), red represents poorest performance (highest error), and intermediate colors (yellow, orange) reflect moderate accuracy levels.

Model Configurations:

- Model 1: Single-landmark (1-LM) MSE 256×256
- Model 2: Multi-landmark MSE 256×256
- Model 3: Ensemble MSE 256×256
- Model 4: Multi-landmark FBCE 256×256
- Model 5: Multi-landmark MSE 512×512
- Model 6: Multi-landmark Combined Dataset MSE 256×256

Vertical Analysis (Model Performance): Model 5 (Multi MSE 512×512) demonstrates superior overall performance, achieving most accurate predictions (green) for 23 of 24 test specimens with only one outlier. Conversely, Model 4 (Multi FBCE) exhibits the poorest performance with predominantly high error rates across specimens, achieving the best prediction for only one image.

Models 1, 2, and 3 demonstrate comparable intermediate performance levels. The single-landmark model (Model 1) tends toward the superior end of this range, while the ensemble model (Model 3) and multi-landmark model (Model 2) show slightly higher error rates with equivalent numbers of predictions in red.

Model 6 (Combined Dataset) displays balanced performance distribution across error categories, with approximately 4-5 predictions in each performance tier, indicating consistent but moderate accuracy across diverse taxonomic specimens.

Horizontal Analysis (Specimen-Specific Performance): Certain specimens demonstrate consistent performance patterns across models. Specimens AC-C0311 see Fig. 5.19 and EL75581 achieve decent predictions across all configurations except Model 4. Specimen AC-C0185 presents challenges for Model 3 specifically. Images EL75599 and EL72724 show comparable moderate performance across most models, excluding the superior 512×512 configuration.

Several specimens (EL38900, EL43172, AC-C0052, see Fig. 5.21, EL75531, see Fig. 5.20) exhibit variable performance across models. Notably, specimen AC-C0066 presents significant challenges for all of the model configurations, suggesting presence of features misleading the landmark localization for this particular image.

5.8.2 Visual Assessment of Landmark Localization

The following visual comparisons illustrate model performance on representative test specimens. Red circles indicate predicted landmarks, while ground truth annotations appear as green circles. Visualization parameters vary between models: some models employ 1-pixel radius markers while others use a 2-pixel radius for the red predicted circles for improved visibility when predictions overlap with ground truth positions. Refer to Figure 1.2b for landmark indices.

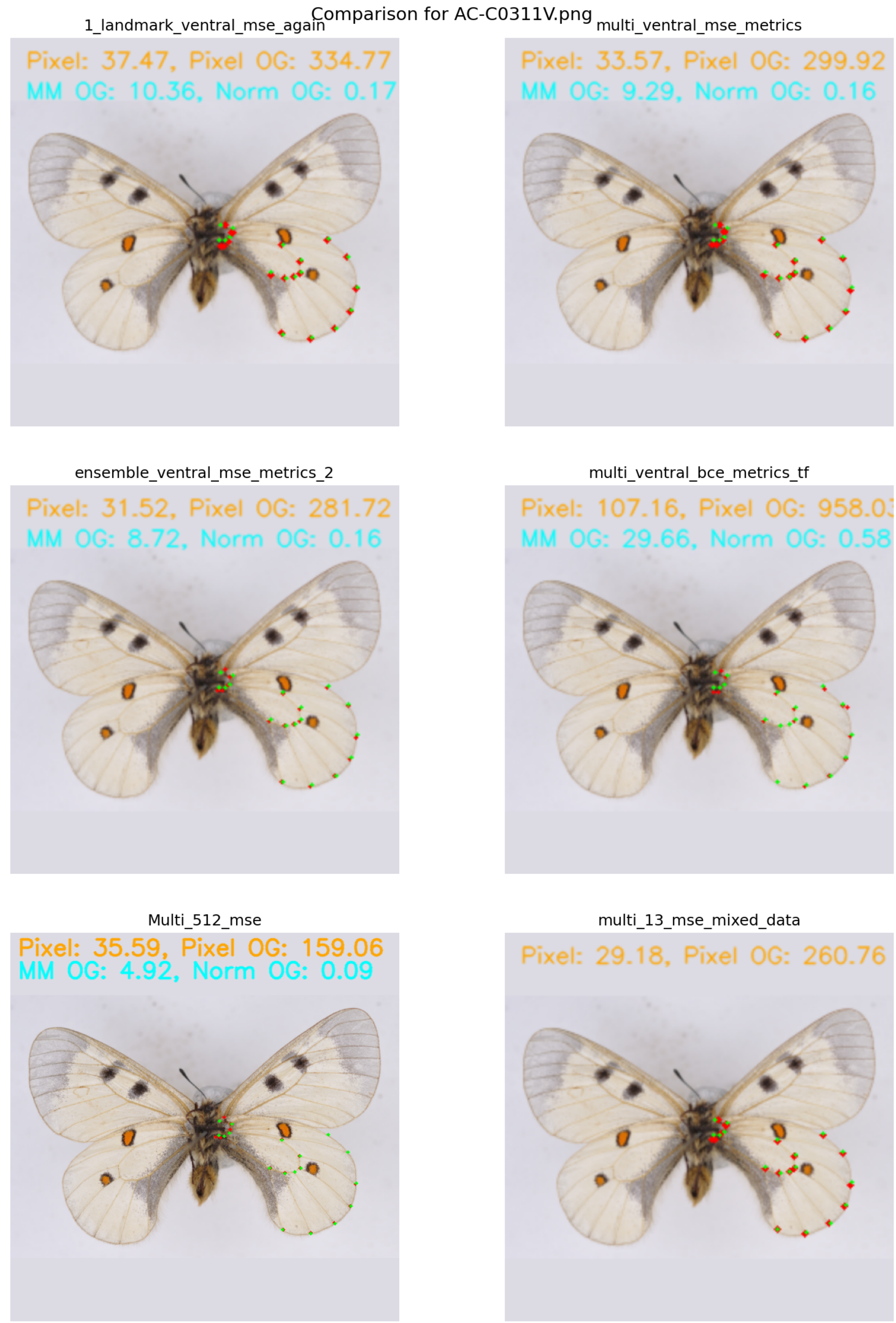

Comparison for AC-C0311V.png
1_landmark_ventral_mse_again
Pixel: 37.47, Pixel OG: 334.77
MM OG: 10.36, Norm OG: 0.17
multi_ventral_mse_metrics
Pixel: 33.57, Pixel OG: 299.92
MM OG: 9.29, Norm OG: 0.16
ensemble_ventral_mse_metrics_2
Pixel: 31.52, Pixel OG: 281.72
MM OG: 8.72, Norm OG: 0.16
multi_ventral_bce_metrics_tf
Pixel: 107.16, Pixel OG: 958.0
MM OG: 29.66, Norm OG: 0.58
Multi_512_mse
Pixel: 35.59, Pixel OG: 159.06
MM OG: 4.92, Norm OG: 0.09
multi_13_mse_mixed_data
Pixel: 29.18, Pixel OG: 260.76

Figure 5.19: *Accurate landmark detection across all model configurations for specimen AC-C0311. The FBCE model fails to localize 2 landmarks at the center of the butterfly's wing.*

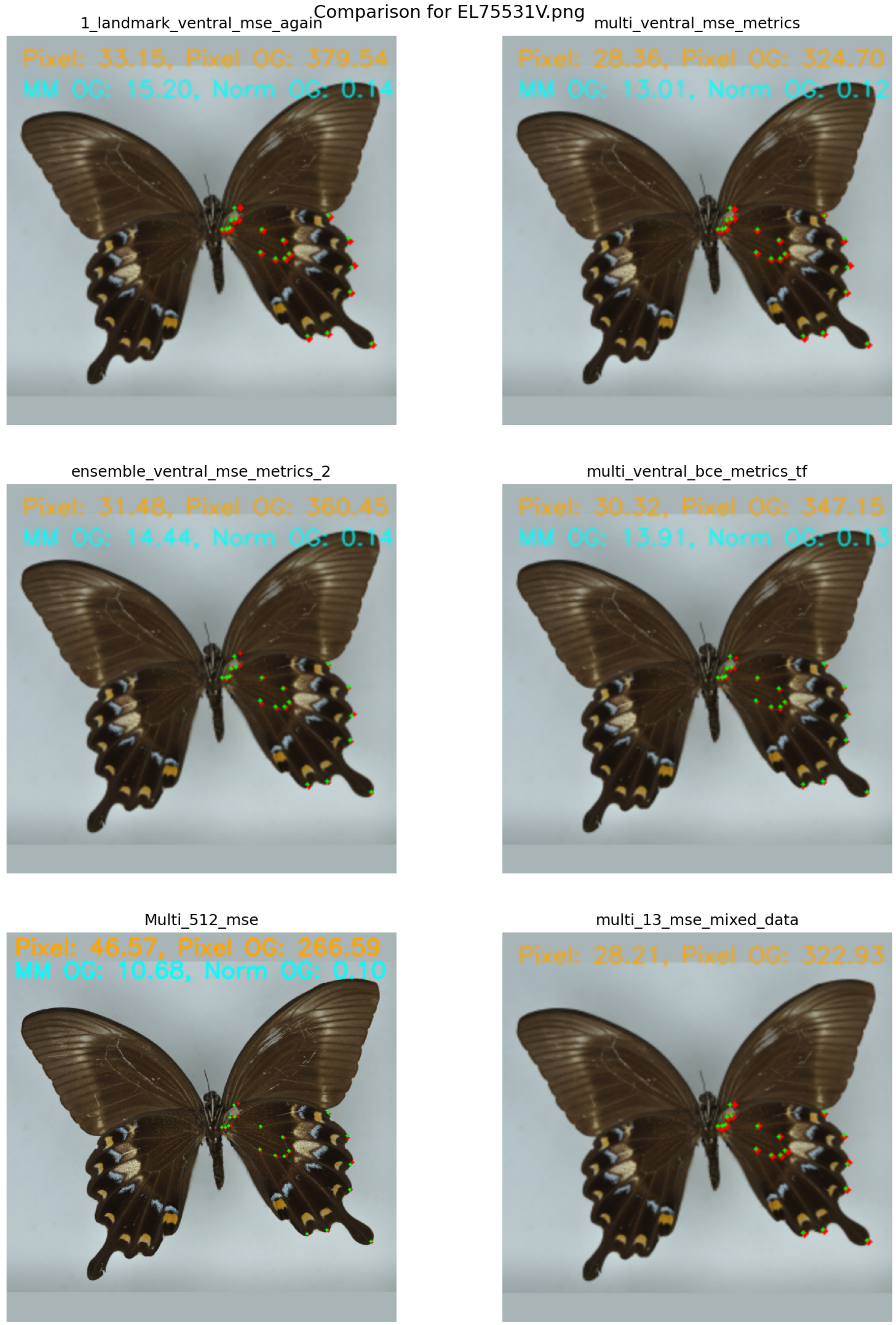

Comparison for EL75531V.png
1_landmark_ventral_mse_again
Pixel: 33.15, Pixel OG: 379.54
MM OG: 15.20, Norm OG: 0.14
multi_ventral_mse_metrics
Pixel: 28.36, Pixel OG: 324.70
MM OG: 13.01, Norm OG: 0.12
ensemble_ventral_mse_metrics_2
Pixel: 31.48, Pixel OG: 360.45
MM OG: 14.44, Norm OG: 0.14
multi_ventral_bce_metrics_tf
Pixel: 30.32, Pixel OG: 347.15
MM OG: 13.91, Norm OG: 0.13
Multi_512_mse
Pixel: 46.57, Pixel OG: 266.59
MM OG: 10.68, Norm OG: 0.10
multi_13_mse_mixed_data
Pixel: 28.21, Pixel OG: 322.93

Figure 5.20: *Systematic challenges in landmark 1 localization across all model configurations for specimen EL75531. (See Fig. 1.2b for landmark indices)*

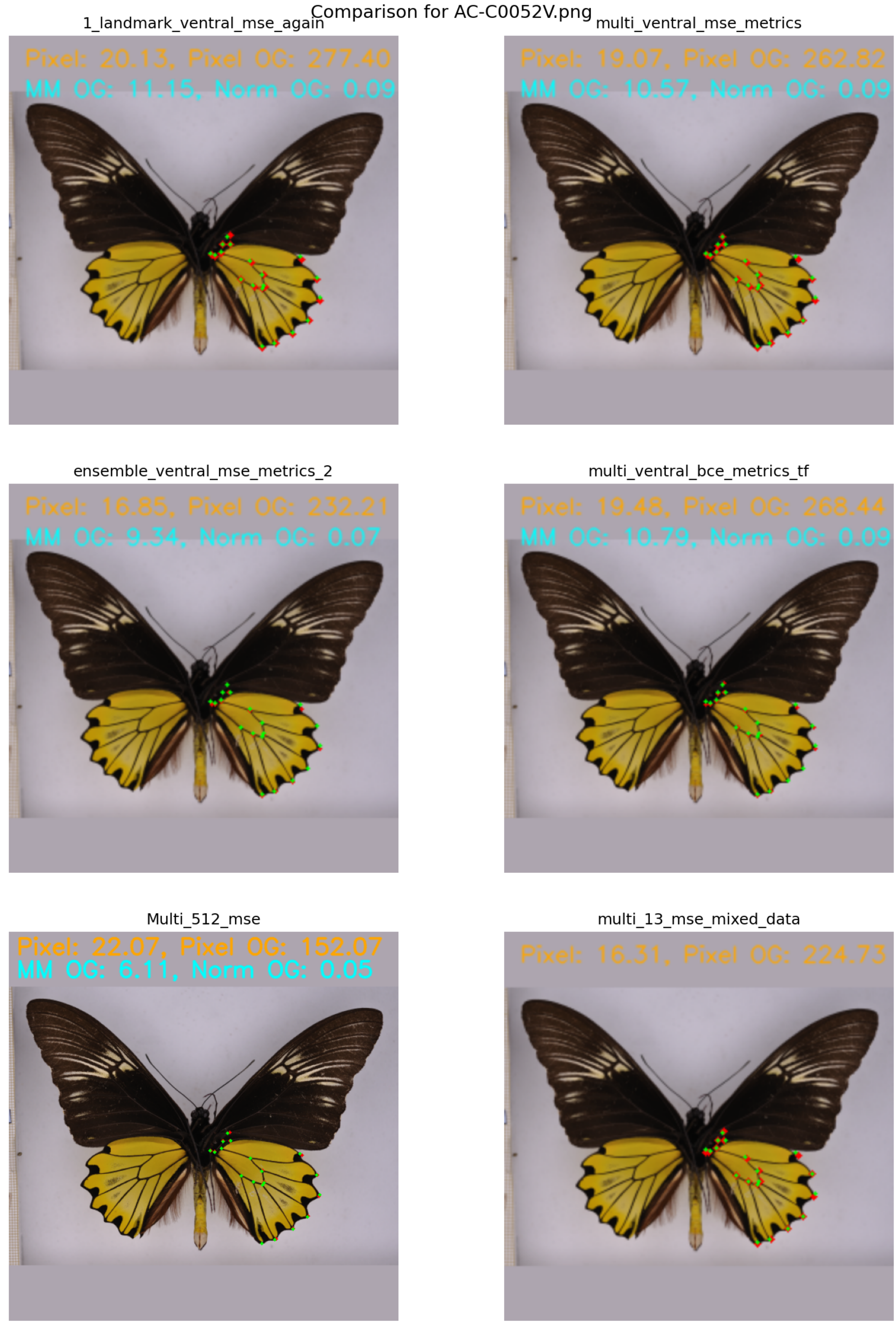

Comparison for AC-C0052V.png
1_landmark_ventral_mse_again
Pixel: 20.13, Pixel OG: 277.40
MM OG: 11.15, Norm OG: 0.09
multi_ventral_mse_metrics
Pixel: 19.07, Pixel OG: 262.82
MM OG: 10.57, Norm OG: 0.09
ensemble_ventral_mse_metrics_2
Pixel: 16.85, Pixel OG: 232.21
MM OG: 9.34, Norm OG: 0.07
multi_ventral_bce_metrics_tf
Pixel: 19.48, Pixel OG: 268.44
MM OG: 10.79, Norm OG: 0.09
Multi_512_mse
Pixel: 22.07, Pixel OG: 152.07
MM OG: 6.11, Norm OG: 0.05
multi_13_mse_mixed_data
Pixel: 16.31, Pixel OG: 224.73

Figure 5.21: *Universal detection difficulties for landmark 2 across all model architectures in specimen AC-C0052. This suggests that misleading features are being extracted by the learned feature extractors of the models, leading the models to believe that the landmark is farther to the left (See Fig. 1.2b for landmark indices).*

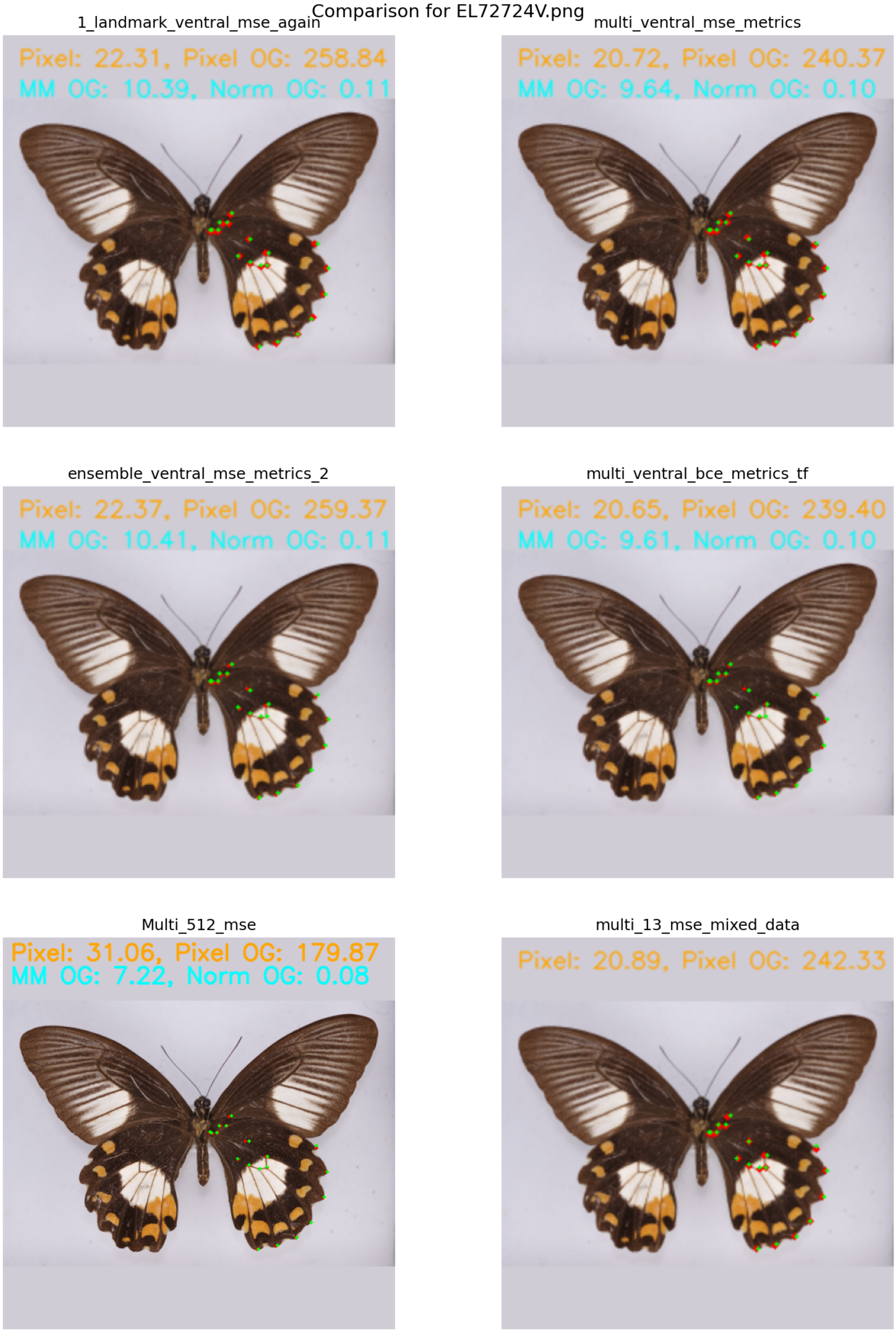
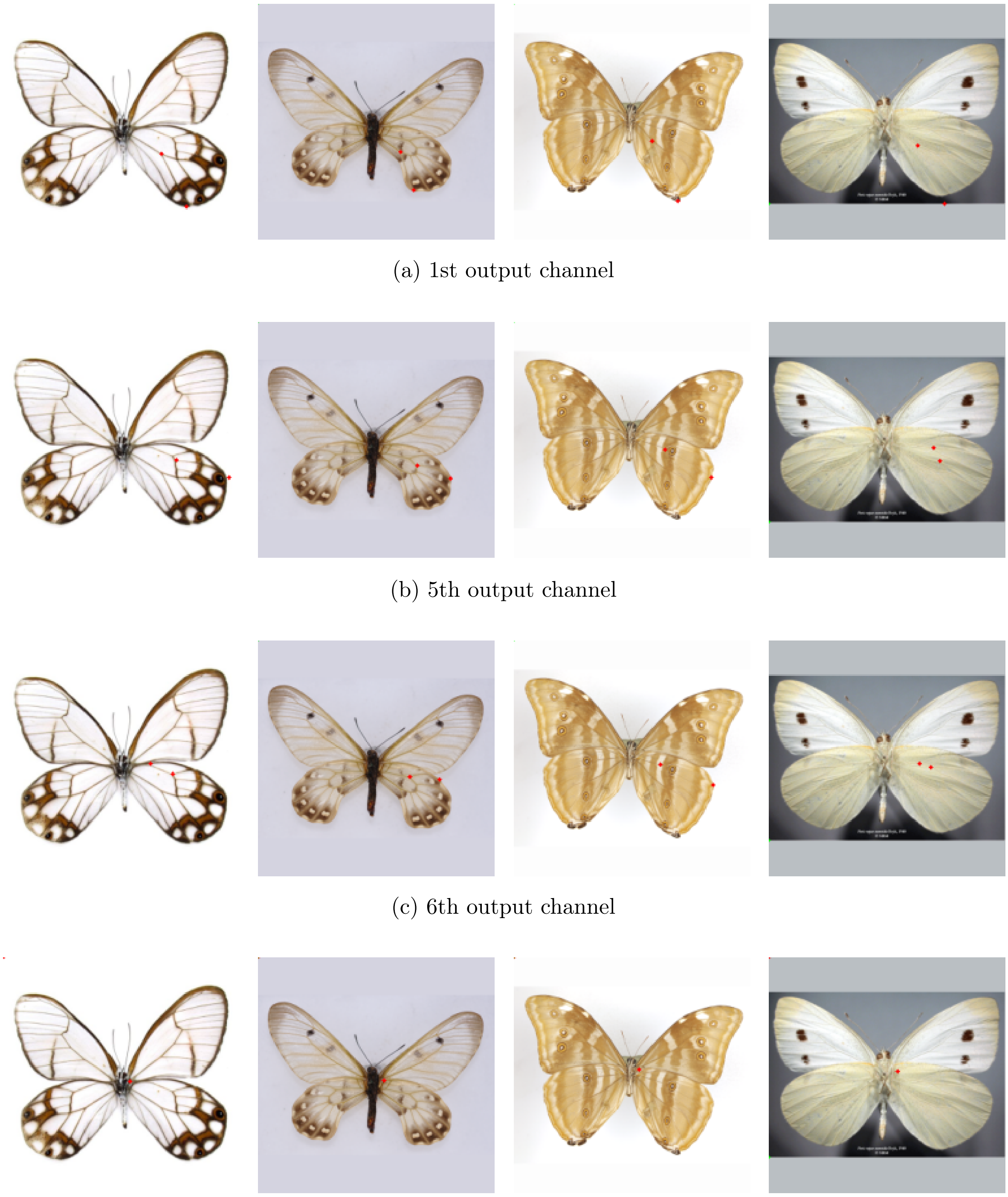

Comparison for EL72724V.png
1_landmark_ventral_mse_again
Pixel: 22.31, Pixel OG: 258.84
MM OG: 10.39, Norm OG: 0.11
multi_ventral_mse_metrics
Pixel: 20.72, Pixel OG: 240.37
MM OG: 9.64, Norm OG: 0.10
ensemble_ventral_mse_metrics_2
Pixel: 22.37, Pixel OG: 259.37
MM OG: 10.41, Norm OG: 0.11
multi_ventral_bce_metrics_tf
Pixel: 20.65, Pixel OG: 239.40
MM OG: 9.61, Norm OG: 0.10
Multi_512_mse
Pixel: 31.06, Pixel OG: 179.87
MM OG: 7.22, Norm OG: 0.08
multi_13_mse_mixed_data
Pixel: 20.89, Pixel OG: 242.33

Figure 5.22: *Comparable intermediate performance across model configurations for specimen EL72724, with consistent challenges in landmark 16 localization. (See Fig. 1.2b for landmark indices)*

5.8.3 Generalisation test for a random butterfly

We search the internet for butterflies with different number of landmarks and venation patterns. We two butterflies that are from different families of the two families that we have in our data set (Papillionidae and Morpho), but of similar chose similar shapes and color patterns.



(a) 1st output channel

(b) 5th output channel

(c) 6th output channel

(d) 10th output channel

Figure 5.23: *Cross-family generalization results showing output channels for different butterfly species. Left to right in each subfigure: Haetera hypaesia [35], Papilionidae, Morpho, Pieris rapae [36]. For channel grouping matrix see 5.6*

We can see in Fig.5.23, Panel (a) shows good localization, with inaccuracy in the prediction for the Pieris Rapae due to the dark background, as the model was trained exclusively on white background specimens. Thus it makes sense to the model to place the landmark on a line edging a white background as it did. Panel (b) shows a decline in the generalization accuracy for the Haetera specimen, and for the Pieris butterfly, a completely skewed detection of the edge landmark in this channel. Panel (c) shows a skewed detection in both Haetera and Pieris specimens; still, for the Haetera butterfly, the model tries to find 2 points connected by a vein and with relatively the same inclination angle. Panel (d) shows a good prediction of a landmark close to the body of the butterfly for all specimens.

5.8.4 Proposed Multi-Family Landmark Detection Framework

Building upon the insights gained from our butterfly landmark detection studies, we propose a comprehensive framework that integrates biological domain expertise with computational efficiency to enable scalable morphometric analysis across multiple taxonomic families.

Framework Architecture

The proposed system centers on a Cytomine Research application where biologists establish universal landmark indexing across butterfly families through strategic coordinate mapping. This approach enables researchers to train unified models on combined datasets while accommodating varying landmark configurations across taxa.

The framework operates through the following key components:

Universal Landmark Mapping: Families are systematically organized by landmark count, with smaller sets treated as subsets of the most comprehensive landmark configuration. Intermediate families integrate seamlessly within this hierarchical structure, with careful attention to anatomical correspondence between homologous structures.

Anatomically-Informed Grouping: Critical morphological features, such as the venation triangulation illustrated in Figure 1.1, are assigned to dedicated output channels. The correlation between biologist-determined mappings and anatomical groupings ensures biological validity of the computational approach.

Automated Processing Pipeline: Each specimen undergoes YOLOv8-based detection and cropping for standardized input representation, followed by coordinate transformation to maintain spatial relationships. The system utilizes a 512×512 pixel U-Net architecture based on the anatomical groupings, generating exponential decay heatmaps for each landmark group.

Implementation Workflow

The framework integrates with Cytomine's RESTful API to systematically extract annotated images where biologists have applied universal landmark tags. The system automatically prepares datasets with appropriate coordinate padding corresponding to the established indexing scheme. Training proceeds iteratively, allowing biologists to refine grouping strategies based on model performance until satisfactory results are achieved.

Recommendations for Future Development

For researchers continuing this work, we recommend several technical enhancements:

- **Noise Reduction:** Implement threshold-based filtering to eliminate artifacts when combining multiple families, particularly when null detections are treated as valid predictions for absent landmarks.

- **Architectural Enhancement:** Consider incorporating foundation models (e.g., EfficientNet-B0) as encoder backbones within the U-Net architecture to leverage pre-trained feature representations.

This integration of biological insight with computational efficiency represents a crucial advancement toward scalable, robust morphometric analysis tools for natural history collections worldwide.

Chapter 6

Conclusion

This research has demonstrated noticeable advancements in automated landmark detection for butterfly wings, focusing on cross-family generalization and computational efficiency through anatomically-informed landmark grouping strategies.

Our preprocessing pipeline incorporating YOLOv8 for butterfly detection and cropping represents an incremental improvement that helps maximize the specimen's representation in the input image. More substantially, when applied to a single butterfly family, the multi-landmark approach achieved convergence approximately four times faster than conventional single-landmark models while maintaining comparable overall accuracy, with only a minor trade-off in occasional outlier predictions.

Contrary to our expectations, implementing overlapping landmark assignments did not reduce outliers or increase accuracy. The theoretical benefits of ensemble averaging were undermined by inconsistent heat zone generation across channels, suggesting that non-overlapping configurations provide a better balance between model complexity and prediction accuracy. Higher input/output dimensions (512×512) improved landmark localization precision at the original image scale, though occasional outliers persisted across all configurations.

While direct application of models trained on one butterfly family to another yielded poor results, our combined dataset approach with anatomically-guided landmark grouping successfully accommodated variable landmark configurations across different taxonomic groups. Notably, in these cross-family applications, the multi-landmark approach clearly outperformed the single-landmark configuration, demonstrating that anatomical grouping strategies particularly enhance the model's ability to capture structural relationships shared across taxa.

Future improvements could include automated approaches to landmark grouping optimization, more sophisticated ensemble methods for outlier rejection, and testing with additional butterfly families representing minimum, maximum, and intermediate landmark counts. This could potentially enable the model to develop a more nuanced understanding of when certain landmarks should be present or absent, achieving true cross-family generalization. Additional research directions include exploring transformer architectures for better capturing long-range dependencies and developing attention mechanisms tailored to wing venation patterns.

In conclusion, this work demonstrates that incorporating domain knowledge into neural network architecture through anatomically-informed grouping strategies significantly improves both computational efficiency and cross-family generalization in butterfly landmark detection, advancing morphometric analysis capabilities in entomological research.

Appendix

TensorFlow Model Architecture Summary

This appendix presents the detailed layer-by-layer architecture of the combined dataset Multi-landmark U-Net model used in this research, as generated by TensorFlow's `model.summary()` function.

```
--------------------------------------------------------------------------------------------------
 Layer (type)                   Output Shape            Param #         Connected to

==================================================================================================
 Input (InputLayer)             [(None, 256, 256, 3)]   0                []


 Block1_Conv1 (Conv2D)          (None, 256, 256, 64)    1792             ['Input[0][0]']


 Block1_Conv2 (Conv2D)          (None, 256, 256, 64)    36928            ['Block1_Conv1[0][0]']


 Block1_Pool1 (MaxPooling2D)    (None, 128, 128, 64)    0                ['Block1_Conv2[0][0]']



 Block2_Conv1 (Conv2D)          (None, 128, 128, 128)   73856            ['Block1_Pool1[0][0]']


 Block2_Conv2 (Conv2D)          (None, 128, 128, 128)   147584           ['Block2_Conv1[0][0]']


 Block2_Pool1 (MaxPooling2D)    (None, 64, 64, 128)     0                ['Block2_Conv2[0][0]']



 Block3_Conv1 (Conv2D)          (None, 64, 64, 256)     295168           ['Block2_Pool1[0][0]']


 Block3_Conv2 (Conv2D)          (None, 64, 64, 256)     590080           ['Block3_Conv1[0][0]']


 Block3_Pool1 (MaxPooling2D)    (None, 32, 32, 256)     0                ['Block3_Conv2[0][0]']



 Block4_Conv1 (Conv2D)          (None, 32, 32, 512)     1180160          ['Block3_Pool1[0][0]']


 Block4_Conv2 (Conv2D)          (None, 32, 32, 512)     2359808          ['Block4_Conv1[0][0]']


 Block4_Pool1 (MaxPooling2D)    (None, 16, 16, 512)     0                ['Block4_Conv2[0][0]']



 Block5_Conv1 (Conv2D)          (None, 16, 16, 1024)    4719616          ['Block4_Pool1[0][0]']


 Block5_Conv2 (Conv2D)          (None, 16, 16, 1024)    9438208          ['Block5_Conv1[0][0]']



 Block6_ConvT1 (Conv2DTranspose)  (None, 32, 32, 512)   2097664          ['Block5_Conv2[0][0]']


 Block6_Concat1 (Concatenate)   (None, 32, 32, 1024)    0                ['Block6_ConvT1[0][0]',

                                                                         'Block4_Conv2[0][0]']

 Block6_Conv1 (Conv2D)          (None, 32, 32, 512)     4719104          ['Block6_Concat1[0][0]']


 Block6_Conv2 (Conv2D)          (None, 32, 32, 512)     2359808          ['Block6_Conv1[0][0]']



 Block7_ConvT1 (Conv2DTranspose)  (None, 64, 64, 256)   524544           ['Block6_Conv2[0][0]']


 Block7_Concat1 (Concatenate)   (None, 64, 64, 512)     0                ['Block7_ConvT1[0][0]',

                                                                         'Block3_Conv2[0][0]']

 Block7_Conv1 (Conv2D)          (None, 64, 64, 256)     1179904          ['Block7_Concat1[0][0]']


 Block7_Conv2 (Conv2D)          (None, 64, 64, 256)     590080           ['Block7_Conv1[0][0]']




 Block8_ConvT1 (Conv2DTranspose)  (None, 128, 128, 128) 131200           ['Block7_Conv2[0][0]']


 Block8_Concat1 (Concatenate)   (None, 128, 128, 256)   0                ['Block8_ConvT1[0][0]',
                                                                         'Block2_Conv2[0][0]']

 Block8_Conv1 (Conv2D)          (None, 128, 128, 128)   295040           ['Block8_Concat1[0][0]']


 Block8_Conv2 (Conv2D)          (None, 128, 128, 128)   147584           ['Block8_Conv1[0][0]']




 Block9_ConvT1 (Conv2DTranspose)  (None, 256, 256, 64)  32832            ['Block8_Conv2[0][0]']


 Block9_Concat1 (Concatenate)   (None, 256, 256, 128)   0                ['Block9_ConvT1[0][0]',
                                                                         'Block1_Conv2[0][0]']

 Block9_Conv1 (Conv2D)          (None, 256, 256, 64)    73792            ['Block9_Concat1[0][0]']


 Block9_Conv2 (Conv2D)          (None, 256, 256, 64)    36928            ['Block9_Conv1[0][0]']




 output (Conv2D)                (None, 256, 256, 13)    845              ['Block9_Conv2[0][0]']



 reshape (Reshape)              (None, 851968, 1)       0               ['output[0][0]']



==================================================================================================
Total params: 31,032,525
Trainable params: 31,032,525
Non-trainable params: 0
__________________________________________________________________________________________________
```

Error Distribution overlay between 1-LM and Multi-LM models with 256 MSE

This section is complementary to Section 5.1.1.

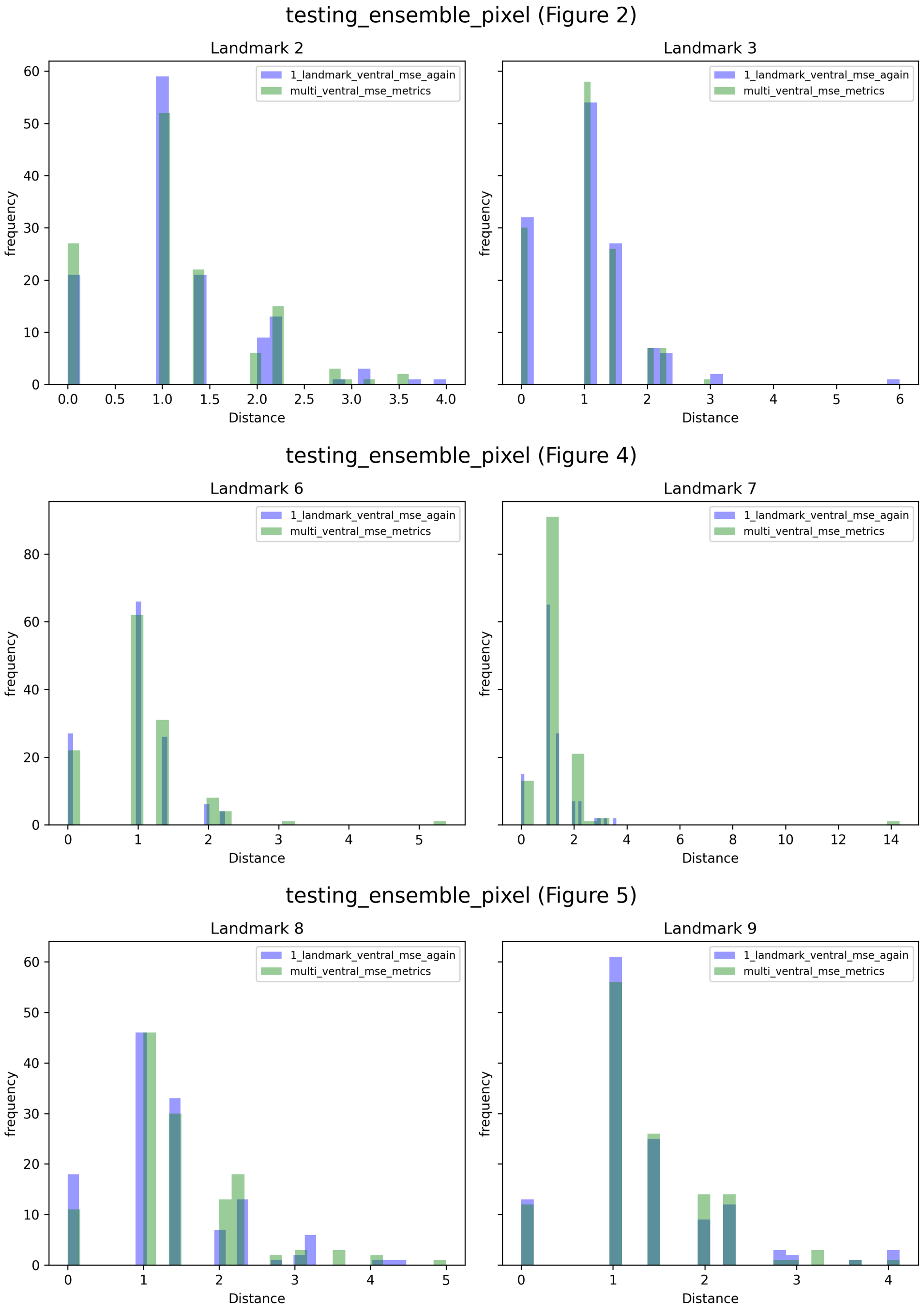


Figure 6.1: *Comparison between multi-landmark (9 channels) model and single-landmark (19 channels) model error distributions for landmarks 2, 3, 6, 7, 8, and 9. (0-indexed)*

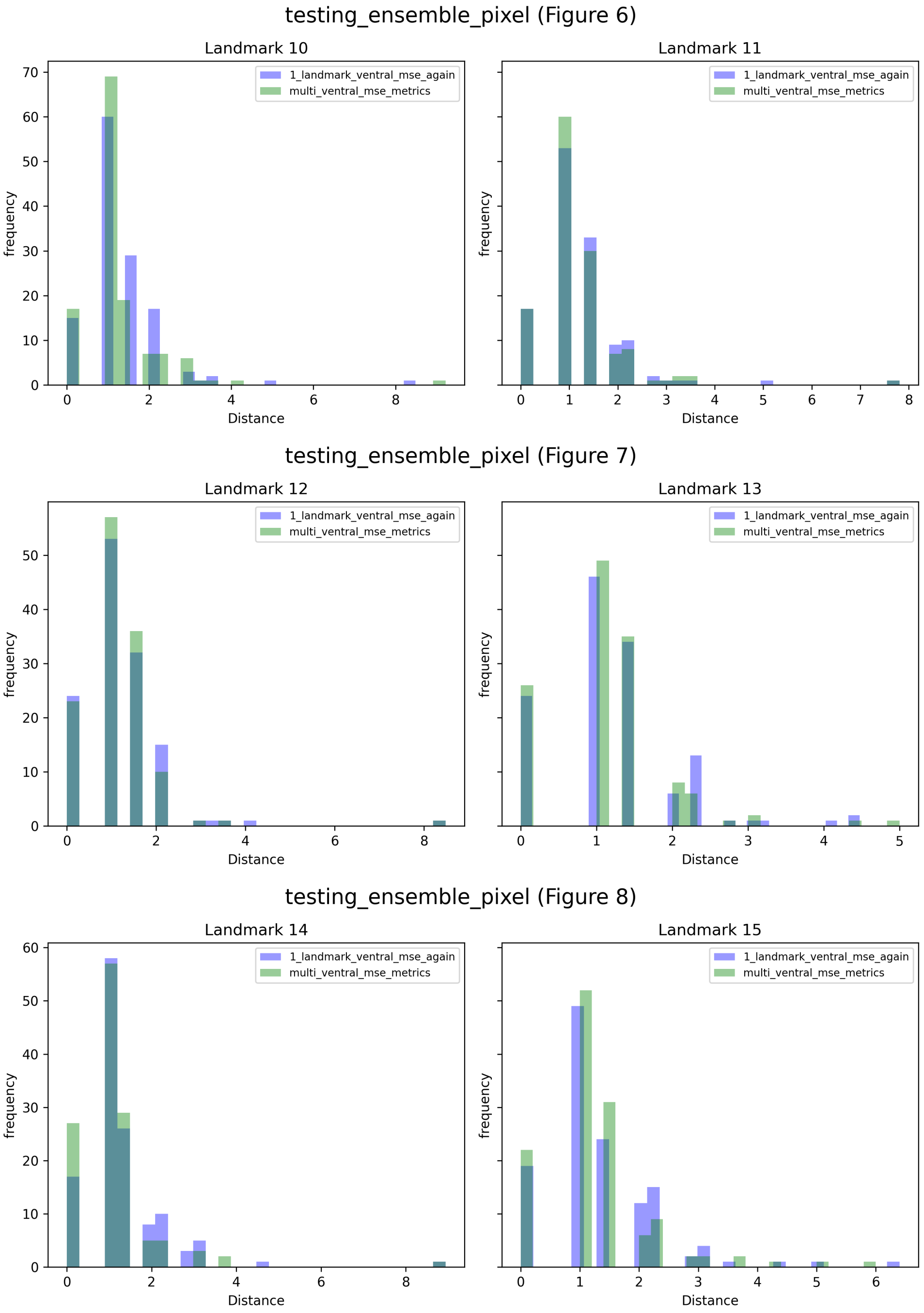


Figure 6.2: *Comparison between multi-landmark (9 channels) model and single-landmark (19 channels) model error distributions for landmarks 10, 11, 12, 13, 14, 15. (0-indexed)*

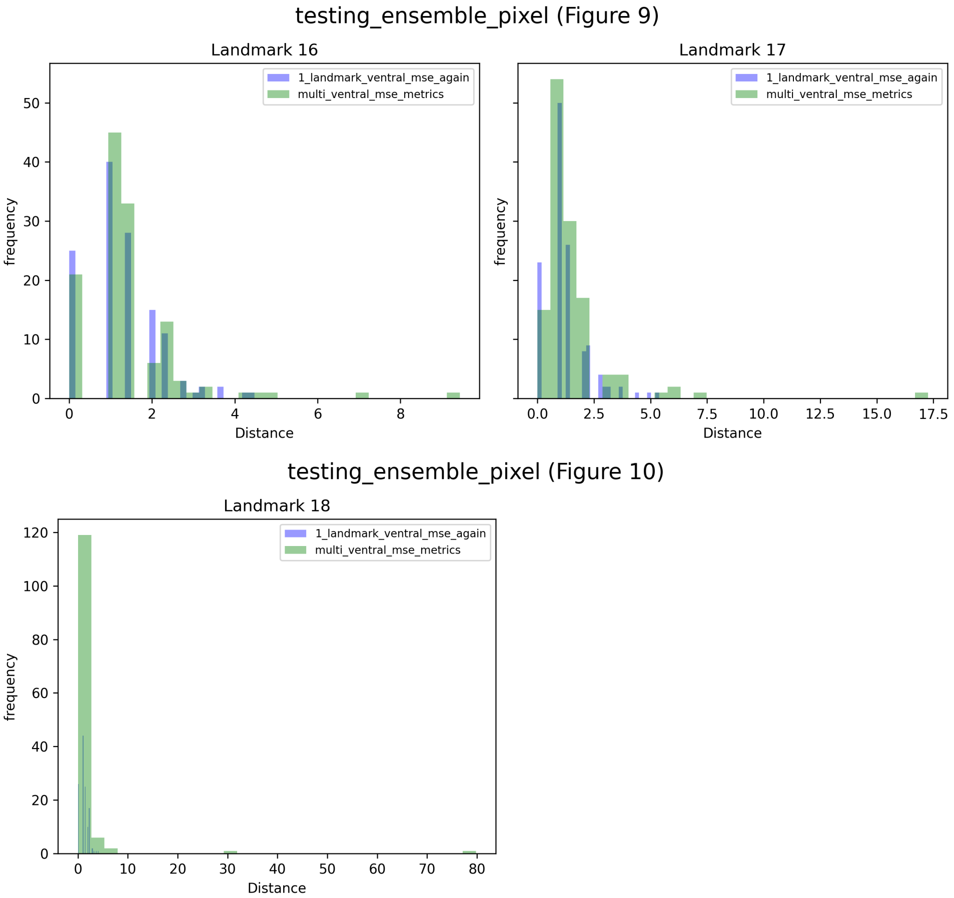


Figure 6.3: *Comparison between multi-landmark (9 channels) model and single-landmark (19 channels) model error distributions for landmarks 16, 17, and 18. (0-indexed)*

Error Distribution overlay between Multi-LM and Multi-Overlap-LM models with 256 MSE

This section is complementary to Section 5.1.2

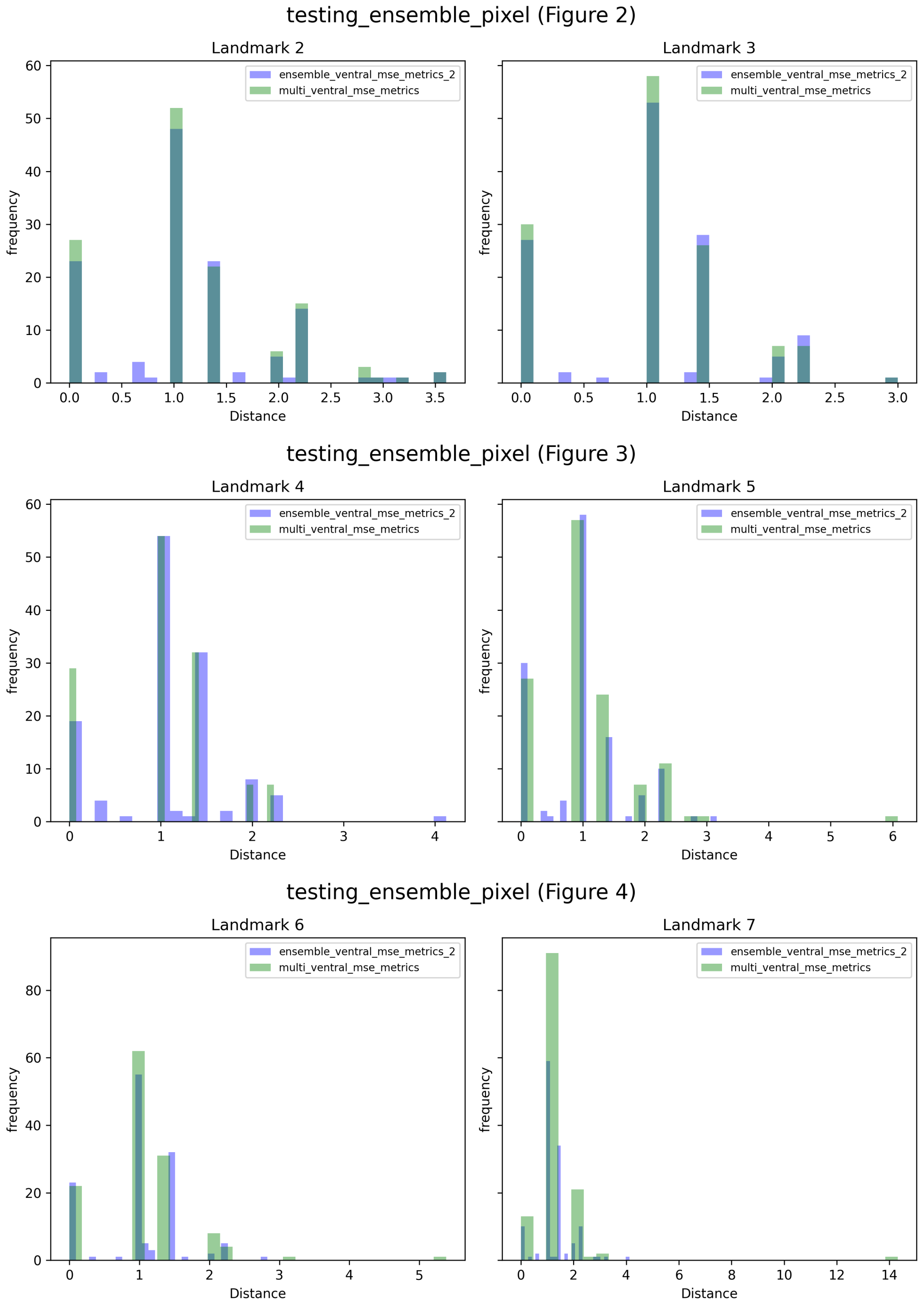


Figure 6.4: *Comparison between multi-landmark (9 channels) model and multi-landmark with overlap (18 channels) model error distributions for landmarks 2, 3, 4, 5, 6, and 7. (0-indexed)*

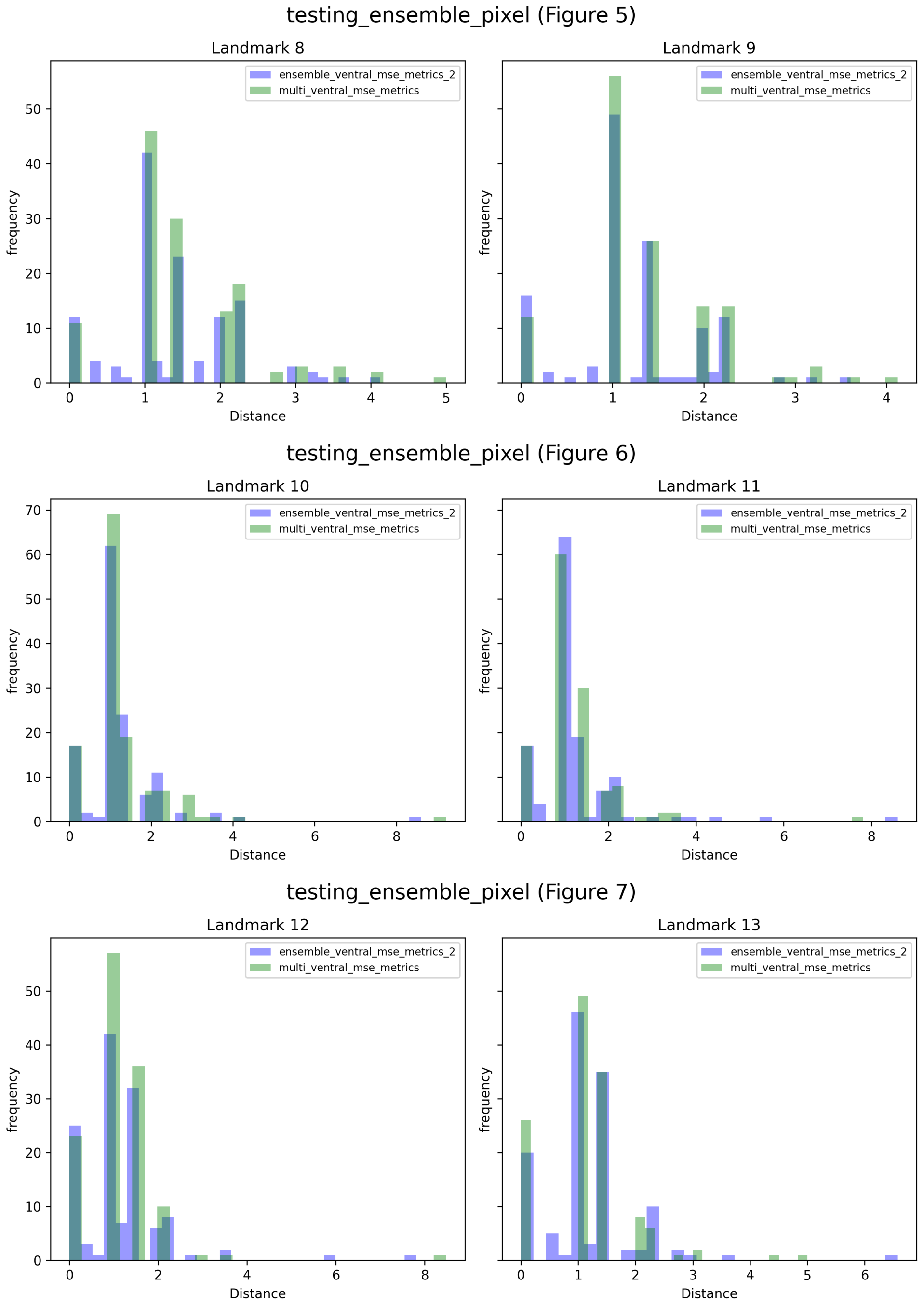


Figure 6.5: *Comparison between multi-landmark (9 channels) model and multi-landmark with overlap (18 channels) model error distributions for landmarks 8, 9, 10, 12, 12, 13. (0-indexed)*

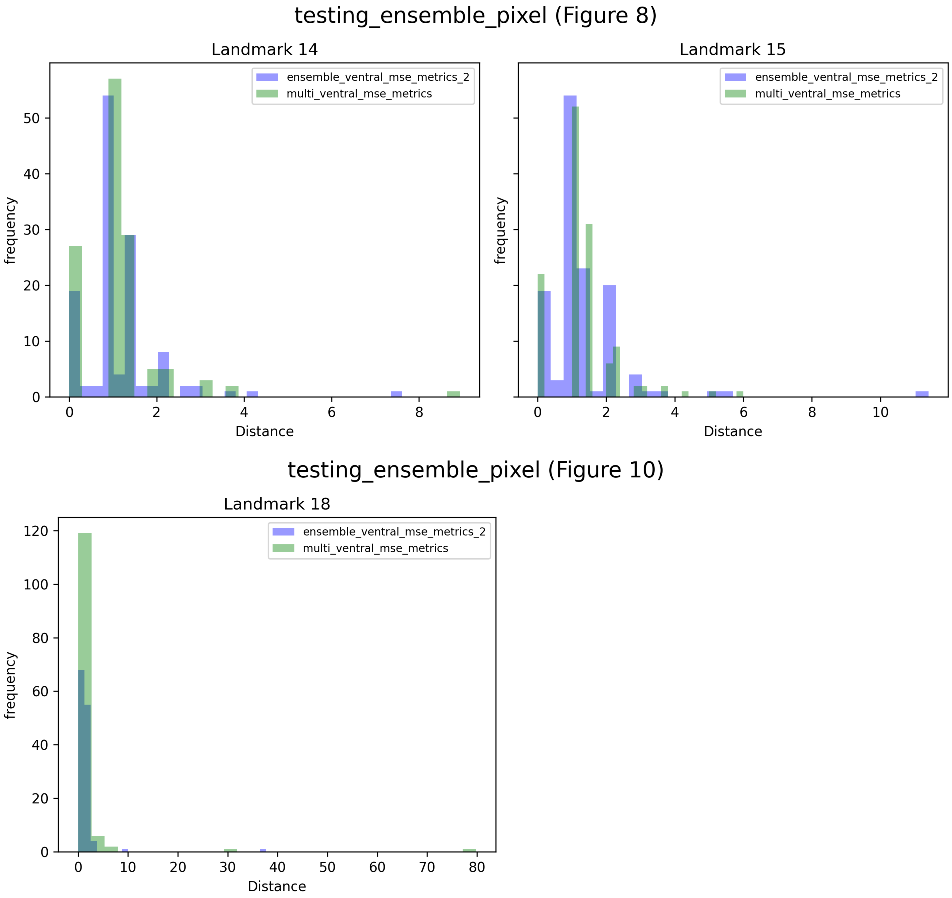


Figure 6.6: *Comparison between multi-landmark (9 channels) model and multi-landmark with overlap (18 channels) model error distributions for landmarks 14, 15, and 18. (0-indexed)*

6.1 Rest of Tabular Comparison of Experimental Model Test Results

This section is complementary to Section 5.4.

Image Name	1-LM MSE	Multi MSE	Ensemble MSE	Multi FBCE	Multi MSE 512
AC-C0419V	259.58	241.82	246.74	245.22	137.67
EL72557V	328.57	245.83	243.12	194.49	144.61
AC-C0077V	271.85	292.48	324.57	329.79	226.33
EL75426V	137.46	159.01	208.20	153.68	98.14
AC-C0102V	392.76	320.07	334.46	315.17	206.89
EL75342V	220.60	207.26	178.67	235.66	136.31
AC-C0279V	235.33	214.12	225.27	519.86	182.76
AC-C0024V	191.13	210.24	189.04	221.04	154.77
EL43239V	414.73	1474.26	952.75	699.87	304.50
EL75723V	251.79	260.17	240.96	232.94	165.69
AC-C0052V	277.40	262.82	232.21	268.44	152.07
AC-C0347V	236.84	200.60	218.84	231.58	126.13
AC-C0044V	260.15	260.66	316.07	388.95	160.10
EL72576V	224.21	222.34	214.65	188.58	103.80
EL75395V	195.32	221.10	208.90	164.09	159.37
EL72566V	300.56	294.18	224.27	255.87	182.02
AC-C0360V	202.75	182.06	157.65	182.22	113.14
EL72560V	197.17	195.05	164.82	217.69	132.26
EL43240V	244.68	210.02	233.57	251.29	122.24
EL72730V	211.02	186.55	166.05	151.29	122.90
EL72710V	157.05	174.35	139.64	204.89	103.57
EL75397V	155.44	162.41	138.55	149.69	172.40
EL72442V	347.26	293.17	304.38	313.11	224.00
AC-C0211V	196.77	263.65	139.71	160.94	100.17
AC-C0377V	276.97	300.04	297.02	284.95	195.17
AC-C0246V	148.39	163.26	160.81	182.77	100.69
EL43204V	318.67	295.59	269.84	290.93	258.75
EL75654V	185.21	192.61	164.59	224.68	137.46
AC-C0429V	199.56	155.17	154.74	189.66	108.38
EL72597V	268.56	263.47	217.09	288.42	148.66
EL43172V	191.09	148.67	186.74	156.47	117.09
EL72717V	203.73	174.46	178.96	178.03	118.78
AC-C0185V	228.74	217.68	546.32	295.03	175.01
AC-C0247V	246.75	305.69	277.06	314.85	115.04
AC-C0033V	261.70	315.41	321.35	237.52	209.82
EL75678V	256.20	304.02	304.45	257.09	143.51
EL72641V	248.11	265.81	281.51	324.54	211.84
EL72581V	195.19	231.55	233.38	223.57	154.31
EL43181V	233.41	187.78	209.79	198.50	128.26
AC-C0225V	100.49	143.50	201.69	115.91	102.02
EL43229V	256.53	285.76	284.83	706.34	144.85
EL75526V	254.63	137.53	177.80	199.69	146.36
EL72527V	220.78	269.71	329.86	279.70	205.59

Table 6.1: *Cumulative landmark localization error per test butterfly specimen across all experimental models using the original distance pixel metric. (Part 2)*

Image Name	1-LM MSE	Multi MSE	Ensemble MSE	Multi FBCE	Multi MSE 512
AC-C0223V	326.78	290.32	320.75	329.13	182.08
AC-C0226V	166.10	130.34	125.99	117.10	74.89
EL75335V	168.24	168.24	220.24	212.30	138.42
AC-C0388V	211.34	177.43	208.65	206.39	96.22
AC-C0322V	270.53	247.18	288.45	263.36	199.94
AC-C0268V	178.13	190.53	160.02	149.72	102.26
EL72434V	622.50	506.61	539.17	516.09	268.57
EL75441V	249.49	286.43	242.77	311.39	215.87
EL43225V	531.89	538.20	519.52	500.88	208.22
EL43113V	196.05	250.48	205.76	240.07	151.90
EL75733V	186.29	207.15	198.70	167.86	141.12
AC-C0193V	185.92	205.34	201.49	189.01	158.27
AC-C0214V	219.68	427.76	240.90	947.19	177.10
EL72724V	258.84	240.37	259.37	239.40	179.87
EL72412V	559.77	562.98	526.71	571.60	270.90
EL72569V	225.25	268.63	263.01	280.67	175.88
AC-C0059V	230.82	230.90	231.53	208.93	174.27
EL75425V	181.09	159.07	156.14	442.94	76.84
AC-C0385V	219.25	268.66	199.85	222.94	174.38
AC-C0330V	426.96	251.19	206.33	258.28	224.31
EL75407V	226.23	272.86	231.23	255.95	130.03
EL75657V	199.60	220.78	216.02	216.97	127.44
AC-C0236V	147.69	156.70	163.18	161.52	109.83
EL43213V	233.78	232.07	234.79	252.21	140.72
EL72439V	197.17	308.55	237.30	256.15	641.00
AC-C0050V	335.33	282.43	253.28	384.88	167.28
EL72731V	225.33	178.28	215.65	223.47	141.79
EL75630V	226.09	240.08	174.36	231.88	172.30
EL72673V	351.35	257.79	288.89	304.72	163.89
AC-C0087V	339.90	279.80	365.40	331.14	844.68
EL75348V	342.49	332.53	245.45	296.61	864.84
EL43108V	379.96	364.00	376.52	391.37	262.73
AC-C0199V	240.98	257.35	203.02	216.85	151.56
EL38900V	190.02	161.56	134.83	169.74	97.08
EL75382V	303.33	255.62	246.06	254.59	302.29
EL75599V	253.83	280.17	252.59	277.12	149.76
EL72582V	199.93	247.64	238.11	254.54	141.70
EL43145V	305.49	277.11	280.04	257.82	151.18
EL43249V	211.92	194.77	228.96	204.55	148.53
EL43134V	189.23	219.93	239.35	270.09	167.09
EL75518V	225.90	209.94	187.97	224.89	143.54
EL75553V	211.97	248.36	264.51	189.34	159.02
AC-C0129V	166.53	197.46	190.90	284.41	121.46

Table 6.2: *Cumulative landmark localization error per test butterfly specimen across all experimental models using the original distance pixel metric. (Part 3)*

Bibliography

[1] Kuznetsova Alina, et al. "The Open Images Dataset V4." International Journal of Computer Vision (IJCV), 2020.

[2] Hierarchy for the 600 boxable classes `https://storage.googleapis.com/openimages/2018_04/bbox_labels_600_hierarchy_visualizer/circle.html`

[3] Open Image Dataset Visualizer `https://storage.googleapis.com/openimages/web/visualizer/index.html`

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[5] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, In dia, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533619.

[6] Olaf Ronneberger, Philipp Fischer, Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation" arXiv preprint arXiv:1505.04597, 2015

[7] Marganne Louis, et al. "Deep-Butterflies : Automatic Landmark Detection" `http://hdl.handle.net/2268.2/14509`

[8] Kumar, N. et al. (2023). Empirical Evaluation of Deep Learning Approaches for Landmark Detection in Fish Bioimages. In: Karlinsky, L., Michaeli, T., Nishino, K. (eds) Computer Vision – ECCV 2022 Workshops. ECCV 2022. Lecture Notes in Computer Science, vol 13804. Springer, Cham. `https://doi.org/10.1007/978-3-031-25069-9_31`

[9] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. `https://doi.org/10.1109/ICCV.2017.324`

[10] Ham, G.-S., & Oh, K. (2023). Learning Spatial Configuration Feature for Landmark Localization in Hand X-rays. Electronics, 12(19), 4038. `https://doi.org/10.3390/electronics12194038`

[11] Zhong, Z., Li, J., Zhang, Z., Jiao, Z., & Gao, X. (2019). An Attention-Guided Deep Regression Model for Landmark Detection in Cephalograms. `https://doi.org/10.1007/978-3-030-32226-7_60`

[12] Tan, Z., Duan, Y., Wu, Z., Feng, J., & Zhou, J. (2019). A Cascade Regression Model for Anatomical Landmark Detection (pp. 43–51). Springer, Cham. `https://doi.org/10.1007/978-3-030-39074-7_5`

[13] Geldenhuys DS, Josias S, Brink W, Makhubele M, Hui C, Landi P, et al. (2023) Deep learning approaches to landmark detection in tsetse wing images. PLoS Comput Biol 19(6):e1011194. https://doi.org/10.1371/journal.pcbi.1011194

[14] Mitteroecker, P., Gunz, P. "Advances in Geometric Morphometrics. Evol Biol 36, 235–247 (2009)." https://doi.org/10.1007/s11692-009-9055-x

[15] Benjamin J. Pomidor, Matt Dean, "GPSA2: combining landmark-free and landmark-based methods in geometric morphometrics." https://doi.org/10.1101/2024.08.03.604701

[16] Berns, A. (2014). A geometric morphometric analysis of wing shape variation in monarch butterflies Danaus plexippus. Deep Blue https://deepblue.lib.umich.edu/bitstream/handle/2027.42/107757/bernsa.pdf

[17] Breuker, C.J., Gibbs, M., Van Dongen, S., & Merckx, T. (2010). The use of geometric morphometrics in studying butterfly wings in an evolutionary ecological context. Springer. http://publicationslist.org.s3.amazonaws.com/data/thomas.merckx/ref-27/Ch%2012%20-Breuker%20et%20al%20%202010.pdf

[18] Xu, S.Q., Bai, Y., Ma, L.B., & Wang, G.H. (2015). A geometric morphometric study of the wing shapes of Pieris rapae (Lepidoptera: Pieridae) from the Qinling Mountains and adjacent regions. Florida Entomologist, 98(1), 128–135. https://bioone.org/journals/florida-entomologist/volume-98/issue-1/024.098.0128/A-Geometric-Morphometric-Study-of-the-Wing-Shapes-of-Pieris/10.1653/024.098.0128.full

[19] Chen, X., Lian, C., Deng, H.H., Kuang, T., & Shen, D. (2021). Fast and Accurate Craniomaxillofacial Landmark Detection via 3D Faster R-CNN. IEEE Transactions on Medical Imaging. DOI: 10.1109/TMI.2021.3099509 https://ieeexplore.ieee.org/abstract/document/9494574

[20] Le Roy, C., Debat, V., & Llaurens, V. (2019). Adaptive evolution of butterfly wing shape: From morphology to behavior. Biological Reviews, 94(4), 1261–1281.

[21] Stern, A. et al. (2021). Heatmap-based 2D Landmark Detection with a Varying Number of Landmarks. In: Palm, C., Deserno, T.M., Handels, H., Maier, A., Maier-Hein, K., Tolxdorff, T. (eds) Bildverarbeitung für die Medizin 2021. Informatik aktuell. Springer Vieweg, Wiesbaden. .https://doi.org/10.1007/978-3-658-33198-6_7

[22] Zhang, Jun et al. “Detecting Anatomical Landmarks From Limited Medical Imaging Data Using Two-Stage Task-Oriented Deep Neural Networks.” IEEE Transactions on Image Processing 26 (2017): 4753-4764.

[23] Kasturi, A., Vosoughi, A., Hadjiyski, N., Stockmaster, L., Sehnert, W., & Wismüller, A. (2024). Anatomical landmark detection in chest x-ray images using transformer-based networks.https://doi.org/10.1117/12.3006881

[24] Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural Networks for Machine Learning, Lecture 6a: Overview of mini-batch gradient descent. *Coursera.*

[25] Interact with Cytomine. https://doc.uliege.cytomine.org/dev-guide/api/#arestful-application.

[26] Sankalp Salve, A Beginner’s Guide to Neural Networks: Forward and Backward Propagation Explained. https://medium.com/@xsankalp13/a-beginners-guide-to-neural-networks-forward-and-backward-propagation-explained-a814666c7

[27] Pragati Baheti, The Essential Guide to Neural Network Architectures. https://www.v7labs.com/blog/neural-network-architectures-guide

[28] Johannes Maucher, Computational Graphs and Gradient Descent Learning in Neural Networks. https://maucher.pages.mi.hdm-stuttgart.de/artificial-intelligence/00_Computational_Graphs.html

[29] Anh H. Reynolds, Convolutional Neural Networks (CNNs) https://anhreynolds.com/blogs/cnn.html

[30] What is YOLOv8? Exploring its Cutting-Edge Features https://yolov8.org/what-is-yolov8/

[31] Open Images V7 Pretrained Models https://docs.ultralytics.com/datasets/detect/open-images-v7/

[32] Raphaël Marée, Loïc Rollus, Benjamin Stévens, Renaud Hoyoux, Gilles Louppe, Rémy Vandaele, Jean-Michel Begon, Philipp Kainz, Pierre Geurts, and Louis Wehenkel. "Collaborative analysis of multi-gigapixel imaging data using Cytomine." Bioinformatics 32.9 (2016), pp. 1395–1401.

[33] Cytomine: Open-source rich internet application for collaborative analysis of multi-gigapixel images. https://cytomine.be/.

[34] Kuan Wei, Understand Transposed Convolutions. https://medium.com/data-science/understand-transposed-convolutions-and-build-your-own-transposed-convolution-layer-from-s

[35] Real Glasswing butterfly framed taxidermy - Haetera hypaesia https://ru.pinterest.com/pin/155303888057763323/

[36] Bryk, F. 1940, Die von Prof. Dr. Lundblad Gesammelten Grosssmetterlinge der Iberischen Halbinsel. - Arkiv för Zoologi Band 32 A nr. 22, Almqvist & Wiksell Boktryckeri, Uppsala. 1 - 36 pp. Mit 7 Tafeln. http://www2.nrm.se/en/lep_nrm/r/pieris_rapaemannidia.html

[37] Leshno M., Ya. Lin V., Pinkus A., Schocken S.,(1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function https://doi.org/10.1016/S0893-6080(05)80131-5

[38] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4), 303-314. DOI: 10.1007/BF02551274 https://doi.org/10.1007/BF02551274