SiamFly - A Novel Pre-Trained Architecture for Large-Scale Fine-Grained Butterfly Classification

- [1]Hari Theivaprakasham (cb.en.p2cen20026@cb.students.amrita.edu), [1]V Sowmya (v_sowmya@cb.amrita.edu), [2]Vinayakumar Ravi (vravi@pmu.edu.sa), [1]E A Gopalakrishnan (ea_gopalakrishnan@cb.amrita.edu) and [1]KP Soman (kp_soman@amrita.edu).

[1] Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

[2] Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia.

Abstract

Butterfly classification is a fine-grained classification problem due to highly complex features and variability between inter-species and intra-species levels. In this study, we introduce a large-scale, research-grade, fine-grained butterfly image dataset comprising 62,287 images corresponding to 686 butterfly species. We also propose SiamFly - a novel pre-trained architecture to facilitate the deep learning model to extract more relevant features pertinent to the butterfly dataset. The experimental results show that the proposed method can achieve better performance on a low-data regime and also surpasses the ImageNet-based transfer-learning performance by a larger margin. We validate our findings by achieving 88.2% top-1 accuracy on the dataset with ConvNeXt model when trained on our proposed approach.

Keywords: butterfly classification; deep learning; transfer-learning; Siamese network.

Corresponding Author:
Hari Theivaprakasham
Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.
Email: cb.en.p2cen20026@cb.students.amrita.edu

Acknowledgment

We would like to thank Mr. Pavendhan A (President of The Nature and Butterfly Society) for his kind assistance in butterfly identification.

SiamFly - A Novel Pre-Trained Architecture for Large-Scale Fine-Grained Butterfly Classification

- [1]Hari Theivaprakasham, [1]V Sowmya, [2]Vinayakumar Ravi, [1]E A Gopalakrishnan and [1]KP Soman

[1] Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

[2] Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia

Abstract

Butterfly classification is a fine-grained classification problem due to highly complex features and variability between inter-species and intra-species levels. In this study, we introduce a large-scale, research-grade, fine-grained butterfly image dataset comprising 62,287 images corresponding to 686 butterfly species. We also propose SiamFly - a novel pre-trained architecture to facilitate the deep learning model to extract more relevant features pertinent to the butterfly dataset. The experimental results show that the proposed method can achieve better performance on a low-data regime and also surpasses the ImageNet-based transfer-learning performance by a larger margin. We validate our findings by achieving 88.2% top-1 accuracy on the dataset with ConvNeXt model when trained on our proposed approach.

Keywords: butterfly classification; deep learning; transfer-learning; Siamese network.

Introduction

Butterflies are a class of insects that belong to the order Insecta and have over 20,000 species (Heppner, 2008). Butterflies play several important roles in the maintenance of biodiversity, the ecological food chain and the sustainable environment. Butterflies help to boost agricultural yields through their pollination of crops, serve as food for various organisms in the food chain, induce a sense of creativity for artists, inspire the researchers through their adaptions and bio-mimicry (Tsai et al., 2020) and even create livelihood opportunities through butterfly nature tourism. Butterflies, like other flora and fauna, act as bio-indicators due to their quick and sensitive responses to subtle habitat or climate change (Rödder et al., 2021).

However, the population of butterflies has been declining in recent decades as a result of rapid climate change, habitat destruction, and urbanization (Wepprich et al., 2019).

The identification and documentation of butterflies are the most difficult challenges in butterfly conservation. Butterflies are visually-complex feature-rich species that come in a variety of shapes, colours, textures, forms, and sexual dimorphism. Their camouflaged appearance makes identification of the butterflies more difficult for human eyes. Butterflies are known to have little intra-class or inter-class variations. For example, the butterfly species namely *Eurema hecabe* and *Eurema blanda* differ mainly in the cell markings on the upper fore-wing. Other differences on the wings are very slight and can be hardly recognizable visually. Some butterfly species exhibit a greater level of sexual dimorphism. For example, the male of *Hypolimnas misippus* is black whereas the female butterfly is orange in colour. The female of *Hypolimnas misippus* also closely resembles another species of butterfly *Danaus chrysippus*. All of the aforementioned challenges combine to make butterfly identification a highly challenging fine-grained classification problem.

Traditional methods of butterfly identification are time-consuming and expensive because they necessitate the intervention of experts and more expensive dissection or genome analysis technologies. Furthermore, skilled taxonomists are frequently in short supply, especially for lower taxonomic groups such as butterflies. To address these difficulties in identification, a sophisticated digital system that can rapidly and precisely recognize butterflies is required. The development of deep learning solutions in the domain of artificial intelligence has been found promising in various computer vision, natural language processing, data analytics, cyber-security tasks, etc (LeCun et al., 2015). Deep learning solutions also have the advantage of learning only the relevant features from a given set of data using various models, filters, and algorithms (M. et al., 2021). Because of these benefits, we will be able to leverage the use of deep learning concepts for our butterfly identification task.

Datasets and algorithms are two of the most important resources for any identification task. The butterfly image dataset forms the basis for the butterfly identification task. Table 1 summarises the butterfly image datasets that were used for butterfly classification. As reported in Table 1, the number of datasets is limited both in terms of the number of species and the number of images per species. This highly imbalanced dataset makes fine-grained recognition more difficult. We should also mention that the geographical distribution of the species in the dataset is very limited. For example, the country of India is known to be home to over 1300 different species of butterflies (Varshney & Smetacek, 2015). However, the currently available datasets are limited to only 315 species, accounting for only 24% of the total butterfly diversity

of India. As a result, there is a need for more butterfly species images to help with the fine-grained classification task.

Table 1 Summary of various butterfly image datasets used in previous studies on butterfly classification

Research Work	Dataset Name	Image Nature	Number of Images	Number of species	Geographical range	Pre-trained models available
(F. Li & Xiong, 2018)	-	Laboratory	750	50	-	No
(Lazebnik et al., 2004)	-	Laboratory	619	7	-	No
(Wang et al., 2009)	Leeds Butterfly Dataset	Laboratory	832	10	-	No
(Rodner et al., 2015)	Ecuador moth dataset and Costa Rica dataset	Laboratory	3000	675	-	No
(Nie et al., 2017)	-	Ecological	10881	82	China	No
(Carvajal et al., 2017)	-	Laboratory	2110	15	Ecuador	No
(Chang et al., 2017)	-	Ecological	14270	636	-	No
(Xie et al., 2018)	Chinese Butterfly Dataset	Ecological and Laboratory	5695	1176	-	No
(Almryad & Kutucu, 2020)	Turkey Butterfly Dataset	Ecological	44659	104	Turkey	No
(Theivaprakasham, 2021)	Indian Butterfly dataset	Ecological	34024	315	India	No
(Z. Lin, 2019)	Butterfly Specimen	Laboratory	19368	48	-	No
Ours (Proposed Dataset)	Extended Indian Butterfly Ecological Dataset	Ecological	62287	686	India	Yes

Initial works on computer-assisted butterfly identification were done by extracting relevant low-medium level wing features from images, such as shape, colour, texture, wing length, and patterns, using various filters and computer vision algorithms (Kang et al., 2012, 2014; Kaya et al., 2014; Kaya & Kayci, 2014; Lazebnik et al., 2004; F. Li & Xiong, 2018). The extracted features were then classified using Linear Classifier (LC), Support Vector Machines (SVM) or Artificial Neural Networks (ANN). However, due to the clear butterfly specimen and clean background, the majority of these works were only successfully implemented on the laboratory image dataset. Traditional methods failed to extract relevant features from real-world ecological images with highly complex backgrounds. The ecological butterfly images also have varying regions of interest in which the butterflies' pixel occupancy in an image varies greatly. Because multiple butterfly species can have similar shapes, colours, and textures,

manually engineering selected attributes may not be the best approach. As a result, the butterfly classification task can be considered fine-grained and requires an advanced methodology to achieve better performance.

Recent studies have utilized the power of deep learning models to extract the high-level features of the butterfly automatically through continuous training and updating of model weights. Convolutional Neural Network (CNN) based models are currently been used in image-based tasks to extract important features (Vinayakumar et al., 2019; Yadav et al., 2021). Several studies have used CNN models namely AlexNet (Rodner et al., 2015), VGGNet (Nie et al., 2017), ResNet (Carvajal et al., 2017), Inception-v3 (Almryad & Kutucu, 2020; Chang et al., 2017), Faster-RCNN (Xie et al., 2018; Zhao et al., 2019), dilated ResNet50 with squeeze-and-excitation (SE) and spatial attention (SA) module (Xin et al., 2020), Multi-scale Sparse Network with Cross-Attention Mechanism (CA-MSNet) (M. Li et al., 2022) for feature extraction and butterfly identification through transfer-learning. The models were evaluated based on the performance metrics namely accuracy, precision, recall and F1-score. Though the performance of these models was reported high on the test data, their trained dataset with a lesser number of images raises the question of their performance on a larger dataset. Also, their performance on a real-time system is not known due to smaller data used for training. Recently, the transformer-based model has produced state-of-the-art (SOTA) results on image classification tasks, based on the principle of attention mechanism (Ze Liu et al., 2021). The transformer models take an image as input, divide it into smaller patches, and then use an attention mechanism to learn the relevant features. In this way, the hard inductive bias of the CNN-based models can be mitigated. To date, the use of transformer-based models on image data for butterfly identification is yet to be investigated.

Most of the published studies on butterfly identification used the transfer-learning technique with ImageNet weights in their models. ImageNet Dataset (Deng et al., 2009) is a collection of over a million images that correspond to 1000 classes. The model trained on the ImageNet dataset is expected to learn generalized features and aid in the transfer-learning approach, especially on small datasets. Without the help of transfer learning, training an entire model with randomly assigned weights may take longer and the convergence may never occur. Hence, the majority of butterfly identification studies used a transfer-learning approach to fine-tune their models on the butterfly dataset, with pre-trained ImageNet weights for faster training and convergence. However, the ImageNet pre-trained model's features may not be relevant to the butterfly dataset, causing it to learn irrelevant features. Therefore, considering the high data imbalance, a technique that can discriminate the image representation in a latent space

even before training the model to the entirety will be more helpful to learn the important features.

In deep learning, Siamese Networks (SN) (Koch et al., 2015) are popularly used to learn contrastive features from a smaller dataset and have also been extensively used in face verification, signature verification, anomaly detection, etc, (Becker & Hinton, 1992; Vinayakumar et al., 2020). SN consists of two or more identical models with shared weights and configuration that accepts multiple inputs and verifies the similarity of the inputs. The verification of the input similarity is carried out by comparing the embedding features of the inputs by a similarity distance function (Cosine distance, Euclidian distance, dot-product distance, etc.,). The SN uses loss special loss function (Triplet Loss, Contrastive Loss) to discriminate the embeddings based on the similarity in the latent space. SN are known to be more robust to the data class imbalance and are known to be scalable due to their contrastive similarity learning. Recently SN are also been widely used in semi-supervised learning and self-supervised learning due to their contrastive learning approach. Self-supervised learning algorithms namely SimCLR (Chen, Kornblith, Norouzi, et al., 2020), BYOL (Grill et al., 2020), SimCLR-v2 (Chen, Kornblith, Swersky, et al., 2020), MOCO (He et al., 2019), Barlow Twins (Zbontar et al., 2021) and DINO (Caron et al., 2021) uses SN-based architectures to learn the features without the help of labeled dataset. Self-supervised learning studies show that self-supervised trained encoders can train and converge with better results even with a limited dataset. Though the SN-based approach has performed better in recent studies, SN-based pre-training has not been explored enough for fine-grained classification tasks or the fully supervised task. Hence, we propose an architecture for pre-training that is structurally similar to SN but with modifications in the last few layers to suit the full-supervised similarity classification.

Motivated by the aforementioned observations, we

- Propose a novel fine-grained large-scale butterfly dataset comprising 62,287 images corresponding to 686 butterfly species.
- Propose a novel architecture named SiamFly, which will be used for pre-training our deep learning models.
- Extensive experiments on CNN and Vision Transformer-based models to understand the effect of pre-trained weights and the limited samples of the dataset.
- Comprehensive Gradient-weighted Class Activation Mapping (GradCAM) analysis on the models and pre-trained weights to analyze the feature learnability.

The paper is structured as follows: Section 2 details the information on the newly proposed large-scale butterfly dataset. Section 3 explains the proposed SiamFly architecture in detail for similarity-based pre-training. In Section 4, we evaluate several experiments on the proposed architecture and compare their performances quantitatively and qualitatively with the ImageNet pre-trained base model. Finally, our conclusion is presented in Section 5.

The interactive working demonstration of the proposed work and the training codes can be found at https://huggingface.co/spaces/Theivaprakasham/indianbutterflyidentification.

2. Dataset

In this study, we propose a new large-scale, fine-grained, research-grade butterfly image dataset with 62,287 images corresponding to 686 butterfly species. Most of the images were collected from internet sources, namely Google Images, Bing and Flickr. Some images were collected from the iNaturalist GBIF dataset filtered over the Indian sub-continent, and the rest of the images were contributed by the first author. Then the collected images were labeled and verified by various butterfly identification experts from India. Sample images from the dataset are illustrated in Fig. 1. From Fig. 1, we can observe high variations in the species' distribution and complex backgrounds with minor inter-class and intra-class variations, which makes the dataset more challenging. The dataset also represented the species' sexual dimorphism, various seasonal and mimetic forms. The photographic similarity of some species of butterflies was also very high, which made the identification task more complex. The dataset however large, had the issue of large data imbalance (Fig. 2). As noticed from Fig. 1, the pixel area occupied by the butterflies in each image varied largely with the diverse region of interest in the image.

Fig. 1 Examples of the butterfly images from the proposed dataset showing diverse variation and the region of interest

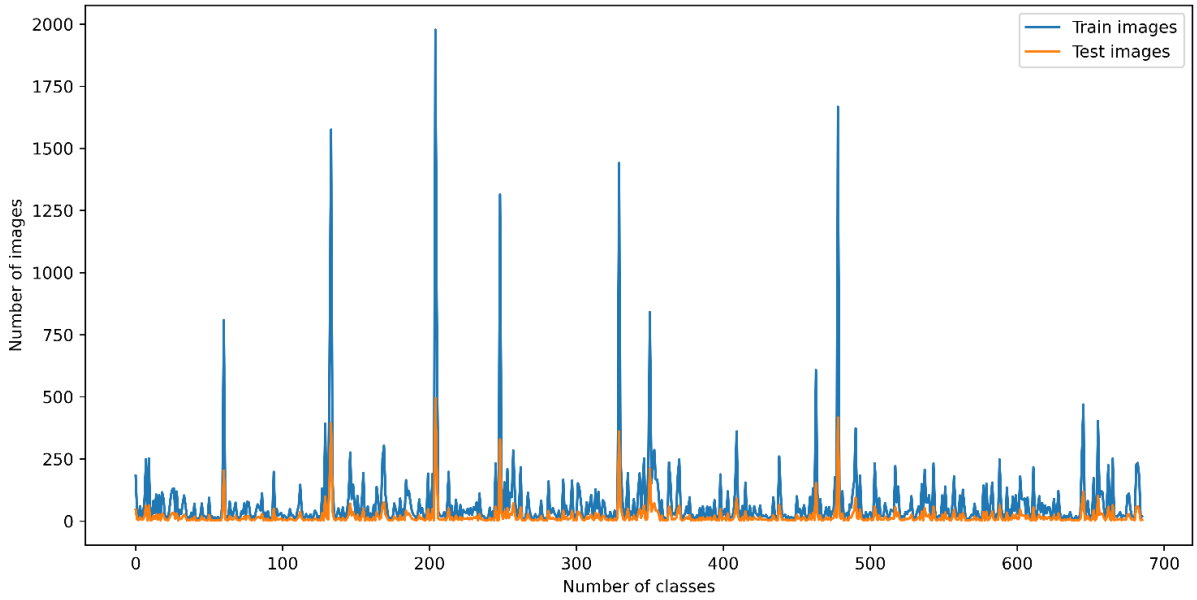

Fig. 2 Distribution of the training and testing butterfly images per species from the dataset

2.1 Data Augmentation

Data augmentation techniques help to increase the dataset and also facilitate better performance. Commonly, data augmentation is achieved in the dataset by artificially creating images through various image transforms. A regular image size of most of the images in the dataset was around 1280px X 800px. However, training the models by utilizing this higher resolution may not be feasible due to high computational complexity and model support. To satisfy the model's input size requirement, we applied random resized crop transformation that out the image at a resolution of 224px X 224px. As the butterflies are fast-moving insects, their chances of getting blurred while being photographed are quite high. Hence, we apply the Random Blur transformation with the maximum blur limit set at 70%. Butterflies are flying insects that often position themselves around flowers to extract nectar. Hence, we apply random rotation, horizontal flips and vertical flips to recreate the scenario and to make the model more generalizable and rotationally invariant. Time, weather, location, and the mode of butterfly image capture can produce images with varying brightness, colour, or contrast or with random noise. We apply random brightness and contrast and Random CutOut transformation to the butterfly image. Fig. 3 illustrates various image transformations, namely resizing with random cropping, rotation, horizontal flip, vertical flip, cut-out, brightness, contrast, and blur performed on the images in the dataset for data augmentation.

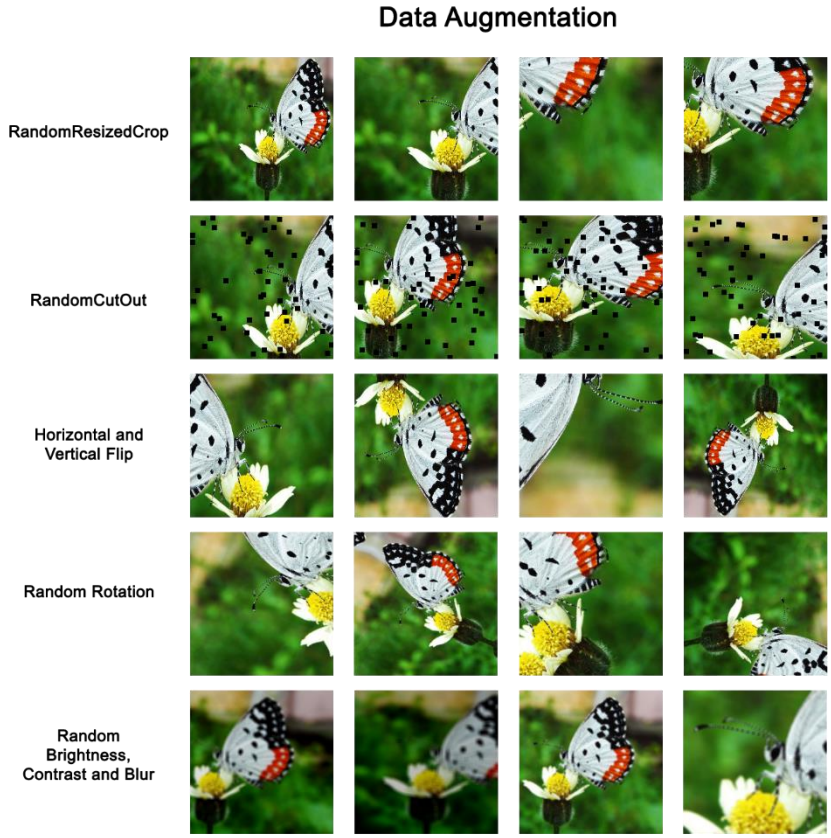

Fig. 3 Image transformation namely resizing with random cropping, rotation, horizontal flip, vertical flip, cut-out, brightness, contrast, and blur applied to the images in the dataset

3. Proposed Methodology

Conventionally, deep neural network training is carried out either using randomly initialized weights or pre-trained weights. The image dataset is forward propagated in batches to the model, the loss function is computed based on the outputs and the weights of the model are updated through back-propagation. When the model is trained with fewer images, its performance is known to be relatively lower compared to pre-trained initialized weights. Fig. 4 illustrates the conventionally used pre-training approach using ImageNet weights.

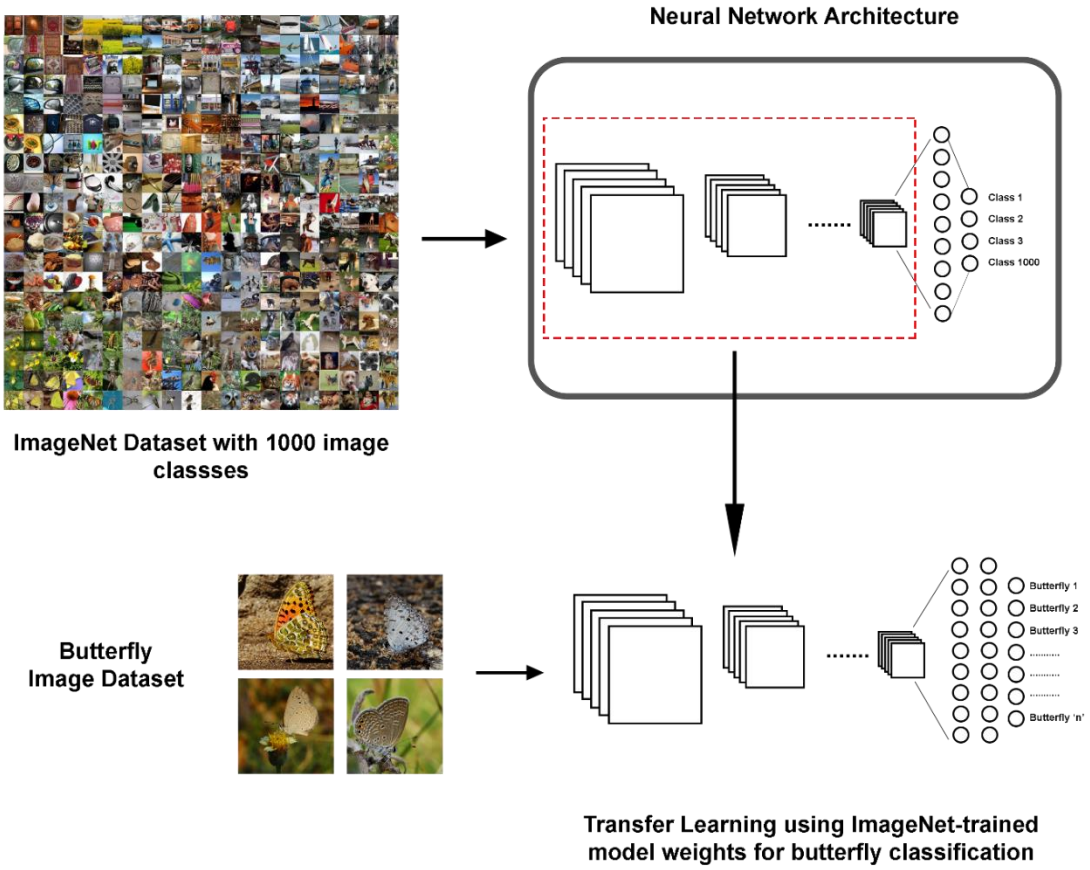

Fig. 4 Workflow of training a deep learning model using ImageNet pre-trained weights

3.1 SiamFly architecture

The proposed SiamFly architecture (Fig. 5) consists of two parallel deep learning models with shared weights and configurations. Each deep learning model's final classifier layer is removed, and the pre-final layer is connected to the global average pooling layer and the global max-pooling layer. Furthermore, the layer is connected to a linear embedding layer of size 512 neurons to obtain a dense feature representation of an image. The size 512 of the embedding layer was selected based on various trial and error experiments on various embedding sizes varying between 128, 256, 512 and 1024. We observed that the embedding size of 512 neurons had the best similarity classification accuracy. Also, another motive is to fix the number of neurons in the embedded layer is to make the architecture more portable for the addition of different models as followed in various studies (Chen, Kornblith, Norouzi, et al., 2020; Khosla et al., 2020; Zbontar et al., 2021). This modified deep-learning model is named the encoder model. The output of the parallel encoders is concatenated and mapped with a Rectified Linear Unit (ReLU) activation function. Further, the output features are then linked to a linear layer made up of two neurons, followed by a softmax layer. The softmax layer outputs the class similarity probabilities of the paired input images.

The workflow of the SiamFly architecture is as follows:

- As each encoder accepts a three-channel RGB image *x* as an input of dimensions of 224px width and 224px height, First, each image is data augmented *Augm(.)* as explained in Section 2.1 to obtain a transformed image ẋ = *Augm(x).*
- Second, the data-augmented image is passed as an input to the Encoder *Encode(.)* to obtain a dense embedding representation of 512 sized vectors *Embed = Encode(ẋ).*
- Third, the two embeddings *Embed 1 and Embed 2* from the parallel encoders are concatenated together and mapped with the ReLU activation function. Further, the 1024-sized dense representation is mapped to a linear layer *Lin(.)* which outputs a similarity probability ***z*** = *Lin(ReLU[Embed 1, Embed 2]).*
- To compute the loss for back-propagation and model weights updating, focal loss (T.-Y. Lin et al., 2017), a modified version of cross-entropy loss that is commonly used to address the issue of class imbalance, is used as a loss function. Focal loss can be calculated as per the equation (1) as follows:

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

where

$\alpha_t$is the balancing factor, $p_t$is the ground truth probability and γ is the calibration factor.

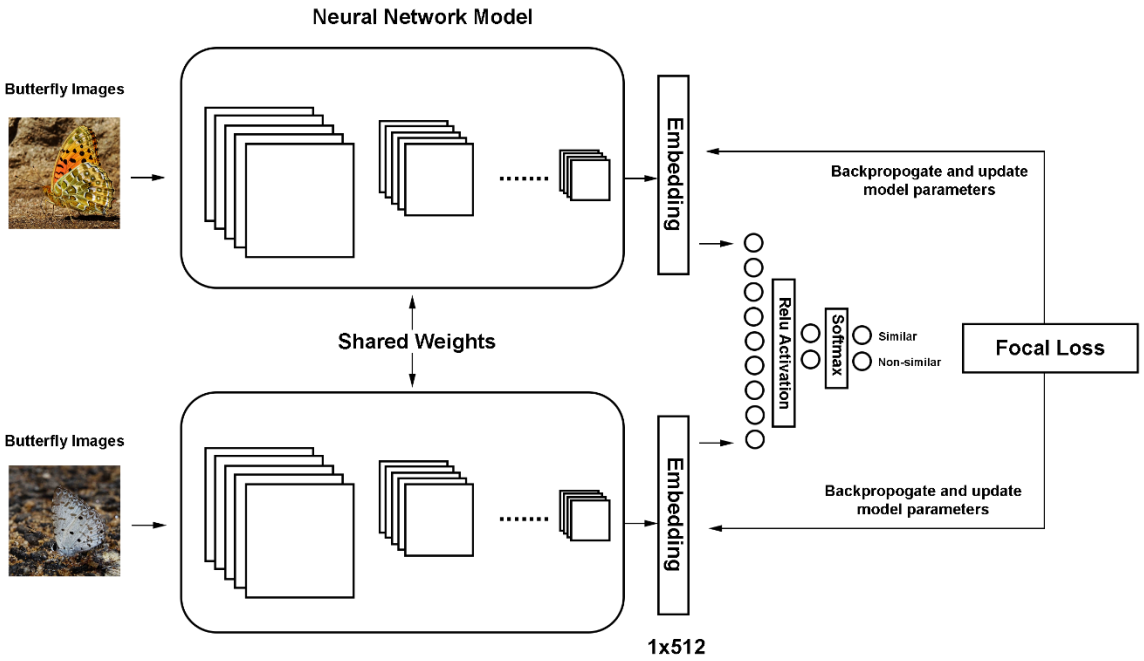

Fig. 5 Overview of the proposed SiamFly architecture that accepts two input images and trains the encoder based on the class similarity

3.2 Encoder Model

We evaluate the proposed pre-training approach using four different deep learning models namely VGG16 (Simonyan & Zisserman, 2014), ResNet50 (He et al., 2016), Swin Transformer (Ze Liu et al., 2021) and ConvNeXt (Zhuang Liu et al., 2022). VGG16 had the highest number of parameters of 137,071,086 followed by ConvNeXt with 88,251,566 parameters and Swin Transformer with 87,382,830 parameters. ResNet50 had the least number of parameters with 24,913,646. Fig. 6 illustrates the high-level overview of the architectures used as an encoder in this study. VGG16 model used a sequential connection of multiple VGG blocks consisting of two-dimensional convolutional layers, ReLU and Max-pooling. This arrangement of multiple blocks helps the model explore deep-level features from the image. However, the model suffered the problem of vanishing gradient when exploring deeper features.

The next breakthrough of deep learning was the introduction of residual networks where the output of the previous layer is fed as an input to the subsequent block through a skip connection. The ResNet model consisted of a series of ResNet blocks which were made up of three convolution layers connected by Batch normalization and ReLU activation function. The output of the first block is then concatenated with the output of the second block and continues similarly till the output layer. This arrangement helps the model retain the feature information and prevent the vanishing gradient problem to a certain extent.

Transformers-based computer vision deep learning models (Han et al., 2022) became popular due to their patch-based image partitioning and self-attention mechanisms, which enabled the model to focus on the important features rather than the irrelevant features. The Swin Transformer produced cutting-edge results on the ImageNet classification task. The Swin transformer is made up of swin blocks that include window-based multi-head self-attention, Multi-Layer Perceptron (MLP), Shifted Window-based multi-head self-attention, and the MLP layer. Before forward propagating to self-attention and MLP layers, the inputs are Layer Normalized. The output of the MLP layer is mapped to the Gaussian Error Linear Unit (GeLU) activation function.

Though the transformer-based models achieved SOTA results on the ImageNet dataset, a new deep convolutional neural network model named ConvNeXt was designed by accounting for the various configurations and architectural designs of the ResNet and Swin transformer models that led to the better performance. The ConvNeXt model consists of a convnext block that has a series of three convolutional layers. The output of the first convolutional layer is layer normalized before propagating to the second convolutional layer. The GeLU activation function is used to map the output of the second convolutional layer to the third convolutional layer before it moves on to the next layer. Similar to ResNet architecture, the output of the first

convnext block is then concatenated with the output of the second convnext block and the sequence continues similarly till the final output layer.

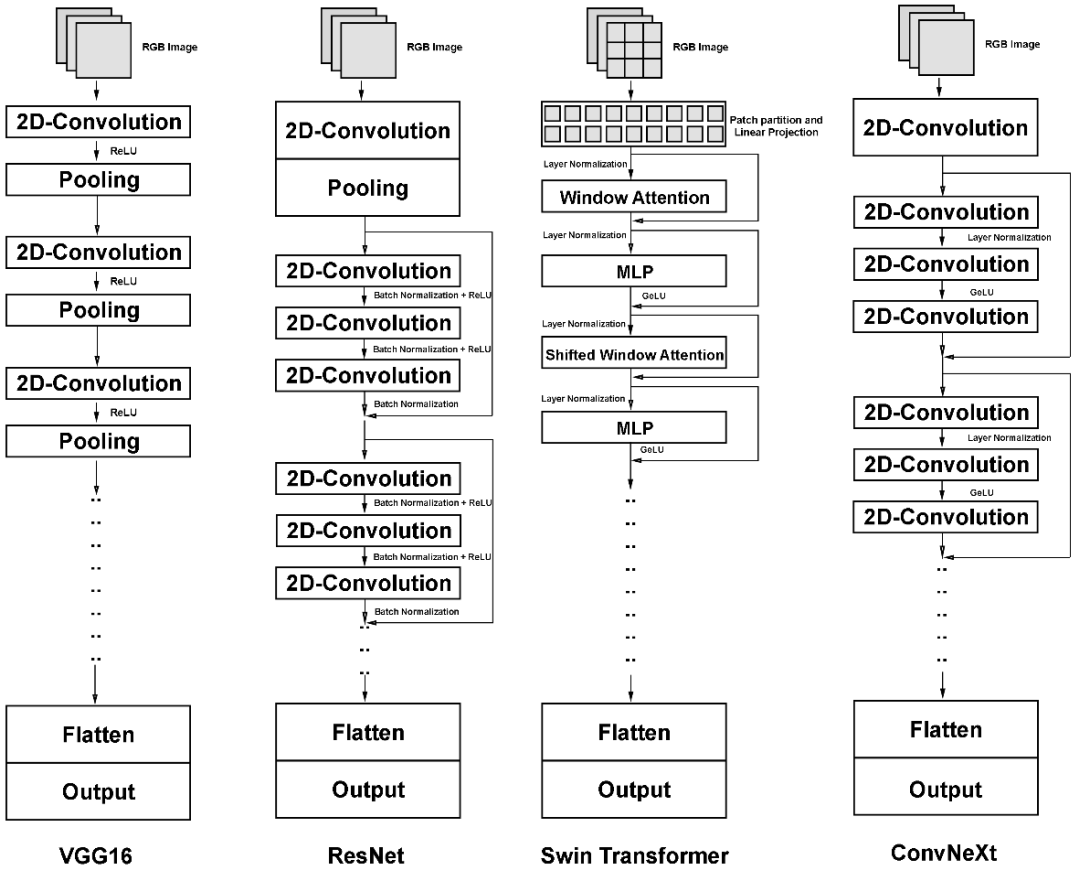

Fig. 6 High-level overview of the deep learning models namely VGG16, ResNet50, Swin Transformer and ConvNeXt

3.3 Evaluation Metrics

To assess the performance of the trained model, we have selected five metrics namely top-1 accuracy, top-3 accuracy, precision, recall and F1-score. The F1-score was purposely selected to mitigate the issue of class imbalance. The metrics can be computed from the equations as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2\,x\,\frac{Precision\,x\,Recall}{Precision + Recall}$$

where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

4. Experiments, results and analysis

In this section, we perform various experiments on the proposed dataset and methods. We also evaluate the results by comparing our proposed method with baseline methods trained with ImageNet weights. In all the experiments, we used our proposed dataset which was split into 70-10-20% comprising 44,600 training images, 4,955 validation images and 12,732 test images. As our training and testing datasets are unique and disjoint in distribution, training on the training data and prediction on the test dataset simulates real-time performance. Experiments were performed using Jarvis Cloud AI environment comprising of 4 x Nvidia A100 Graphical Processing Unit (GPU) (40GB), 32GB Random Access Memory and 64 virtual Central Processing Unit. We used Distributed Data-Parallel strategy for training our model using multiple GPUs We used python programming language and Fastai (Howard & Gugger, 2020), and PyTorch deep learning library to train our model.

The following three main experiments were evaluated as follows.

1) Similarity-based pre-training on encoder model using the SiamFly architecture on the butterfly dataset.
2) Add a linear classifier to the encoder model and train only the linear classifier with both pre-trained ImageNet and SiamFly weights.
3) Train the entire encoder with a linear classifier with both pre-trained ImageNet and SiamFly weights.

4.1 SiamFly based model pre-training

In this experiment, first, we generate random image pairs based on their species' similarities. The image pairs that correspond to the same species are labeled as similar and others are labeled as not similar. We pass each image pair as an input to the proposed SiamFly architecture and train the model using a batch size of 48, Adam optimizer function (Kingma & Ba, 2014), and cosine annealing scheduler with weight decay of $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The learning rate was chosen by following the learning rate finder (LRF) method as proposed by Smith (2017). First, we iterate one mini-batch data over varying learning rates from 10^{-6} and 10^{1} and calculate the validation loss. We select the learning rate where the validation loss was lowest. For VGG16 and ConvNext model, we selected a learning rate of 10^{-4}, Swin

Transformer with 10^{-5} and ResNet50 with 10^{-3}. We trained all the model for 50 epochs where the validation loss is saturated. Then, we evaluate on the performance on various encoder models in Table 3 based on their ability to distinguish the image pair species as similar or not. Best performing model is highlighted in the bold in Table 2.

Table 2 Test data results of similarity-based training of different encoder models using SiamFly architecture

Models	Accuracy	Precision	Recall	F1-score	Time per epoch (second)
Swin-Transformer	***0.987***	***0.987***	***0.987***	***0.987***	91
ConvNeXt	0.986	0.986	0.986	0.986	93
ResNet50	0.987	0.987	0.987	0.987	87
VGG16	0.983	0.983	0.983	0.983	135

We observe that all the four proposed models perform relatively similarly with all metrics nearly equal to 98% in the similarity classification task. Since the number of false positives and false negatives are equal, the precision and recall values remain the same. F1-score also remains the same as it is the harmonic mean of precision and recall. The time taken for training the VGG16 model was quite long due to the computational complexity of the higher number of parameters.

Fig. 7 reveals the GradCAM (Selvaraju et al., 2017) feature-map visualization of all the models on a few test images. Although the encoder models report similar evaluation results, the activated features of the encoders are fairly different. This observation elucidates the model's image understanding and representation. As seen from Fig. 8, Swin Transformer's GradCAM map features are activated on the entirety of the butterfly in the image. Whereas in other models, only a few specialized features of the butterfly are activated. However, we observe that the GradCAM activated region occurred only on the image portion pertinent to the butterfly proving the model's learnability to distinguish the background from the butterfly.

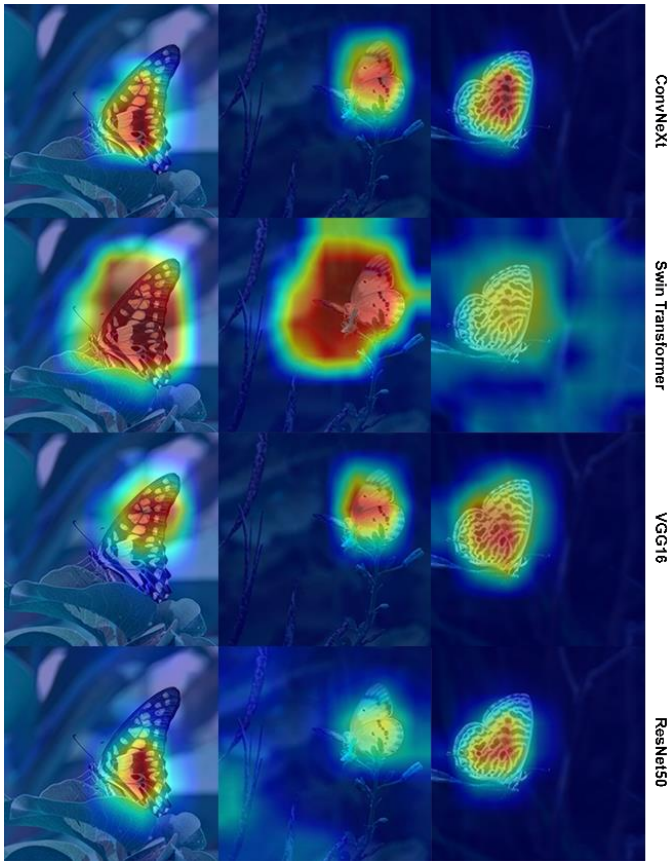

Fig. 7 GradCAM visualization of various encoder models trained using SiamFly architecture

4.2 Linear classifier evaluation

To evaluate the SiamFly trained encoder model on a downstream butterfly classification task, we added a linear classifier layer with 686 neurons to the end of the pre-trained encoder as shown in Fig. 8. We trained only the newly added linear classifier layer by freezing the weights of the SiamFly pre-trained encoder. The experiment was carried out for 50 epochs with a batch size of 64, utilizing the cross-entropy loss function, Adam optimizer and cosine annealing scheduler. The learning rate was chosen by following the LRF method. The training images were data-augmented as per transforms as mentioned in Section 2.1 before passing it to the model for training. This experiment will reveal the encoder's learnability to classify the butterflies based on the image representation. We also repeat the same experiment on the encoder model with ImageNet pre-trained weights and evaluate the performance of the model. We report the top-1 accuracy, the top-3 accuracy, precision, recall and F1-score metrics of both the SiamFly pretrained model and ImageNet pretrained weights trained over a linear classifier in Table 3.

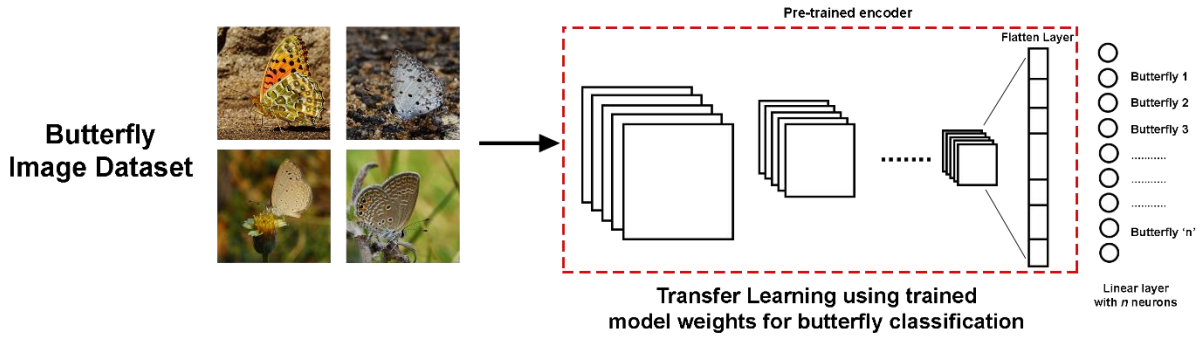

Fig. 8 Training only output linear layer of the model using pre-trained encoder weights

Table 3 Evaluation results on test data using linear classifier appended on the pretrained encoder

Pre-training Method	Model	Accuracy	Top-3 accuracy	Precision	Recall	F1-score	Time per epoch (second)
ImageNet	ResNet50	0.671	0.830	0.590	0.480	0.506	64
ImageNet	VGG16	0.590	0.785	0.426	0.374	0.359	90
ImageNet	ConvNeXt	0.714	0.870	0.570	0.500	0.511	74
ImageNet	Swin Transformer	0.781	0.926	0.646	0.598	0.574	73
SiamFly (ours)	ResNet50	0.760	0.916	0.596	0.541	0.539	64
SiamFly (ours)	VGG16	0.592	0.768	0.477	0.388	0.410	90
SiamFly (ours)	ConvNeXt	0.732	0.865	0.592	0.485	0.508	74
SiamFly (ours)	**Swin Transformer**	**0.849**	**0.948**	**0.795**	**0.714**	**0.734**	73

In Table 3, we see that all the SiamFly pre-trained encoder models out-perform the traditional transfer-learning approach of using ImageNet weights. SiamFly pre-trained Swin Transformer model had the highest top-1 accuracy, top-3 accuracy, precision, recall and F1-score. Swin transformer when trained on frozen ImageNet pre-trained weights reported a 57.4% F1-score, whereas our proposed method had a 16% increment in the performance of the same model. Similarly, ConvNeXt model showed a 2% increment and the ResNet50 model showed a 9% increment on top-1 accuracy. Except for the VGG16 model, all the other models showed performance improvement when trained using our proposed method. Hence, we prove that our proposed approach of transfer-learning on a linear classifier works significantly compared to ImageNet weights-based transfer-learning. The best-performing model is highlighted in bold in Table 3.

4.3 Complete model training with SiamFly weights and ImageNet weights

To completely understand the capability and robustness of our proposed method on the limited dataset, we trained the entire encoder model with a complete dataset and also with 30% of the training dataset utilizing both the SiamFly pre-trained weights and ImageNet weights. We use the same set of training parameters as used in Section 4.2. Table 4 reports the evaluation metrics on the experiments performed over the test dataset. We observe that when the model was trained entirely, the weights initialized by our proposed methods show a drastic improvement in the model performance. ConvNeXt model trained on the SiamFly-based methodology achieved the best results with 88.2% top-1 accuracy and 77.3% F1-score. Also, we observe that the SiamFly-based pre-training worked significantly better even with 30% of the training data. It matched the performance of the ImageNet pre-trained model trained on 100% of the trained data. We analyzed the GradCAM features maps of the fully trained model with different initial pre-trained weights on both 100% and 30% of the training data.

Table 4 Results of butterfly classification after fine-tuning the entire model using our proposed weights and the ImageNet weights

Pre-training Method	Data used from training dataset	Models	Accuracy	Top-3 Accuracy	Precision	Recall	F1-score
ImageNet	100%	Swin-Transformer	0.869	0.983	0.815	0.761	0.766
		ConvNeXt	0.862	0.981	0.823	0.746	0.764
		ResNet50	0.861	0.981	0.842	0.721	0.763
		VGG16	0.858	0.972	0.816	0.715	0.759
ImageNet	30%	Swin-Transformer	0.718	0.954	0.573	0.483	0.498
		ConvNeXt	0.813	0.966	0.703	0.613	0.634
		ResNet50	0.791	0.963	0.712	0.631	0.671
		VGG16	0.72	0.956	0.617	0.452	0.482
SiamFly	**100%**	Swin-Transformer	0.873	0.982	0.800	0.755	0.757
		ConvNeXt	**0.882**	**0.987**	**0.816**	**0.770**	**0.773**
		ResNet50	0.881	0.987	0.843	0.731	0.771
		VGG16	0.86.1	0.981	0.823	0.746	0.766
SiamFly	30%	Swin-Transformer	0.859	0.974	0.767	0.713	0.716
		ConvNeXt	0.855	0.976	0.821	0.751	0.742
		ResNet50	0.821	0.969	0.731	0.678	0.683
		VGG16	0.815	0.949	0.725	0.668	0.621

As seen in Fig. 9, the feature maps are not only activated on the butterflies but also in other regions surrounding the butterfly. This behaviour hints that the model is extracting the non-essential feature to identify the butterfly which may end up with the wrong species classification. This may be due to the data limitation in the training process as the model was trained only on 30% of the dataset. Though the activated features on the butterflies are known to increase when the data was trained with 100% of the data as shown in Fig. 10, the activation of features other than the butterflies shows the downside of using ImageNet pre-trained weights. Contrary to ImageNet-based pre-trained weights, the model trained on our proposed method shows that GradCAM activated features stay well close to the butterflies making it more robust in locating the butterflies as shown in Fig. 11 and Fig. 12. The GradCAM analysis is also directly reflected in evaluation results as tabulated in Table 4.

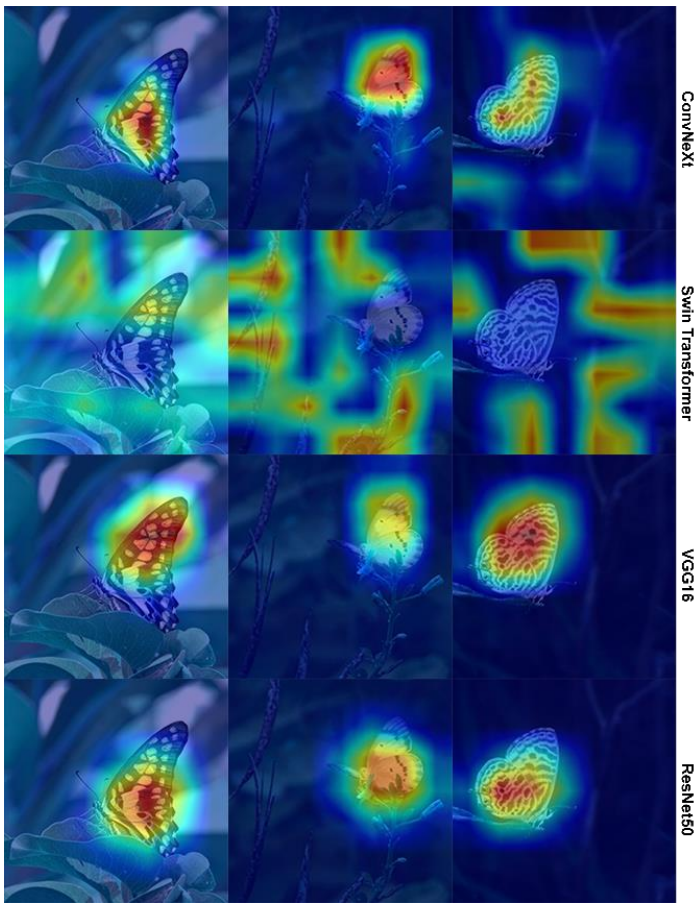

Fig. 9 GradCAM visualization of various encoder models trained using ImageNet pre-trained weights on 30% of the training dataset

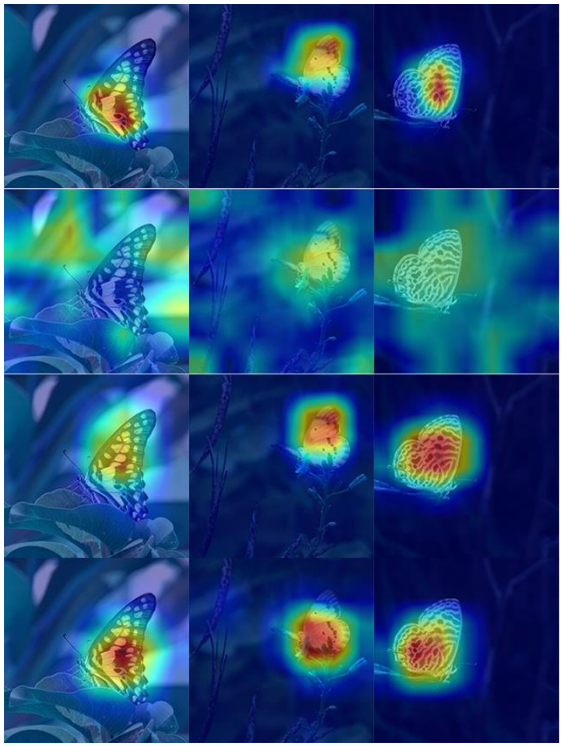

Fig. 10 GradCAM visualization of various encoder models trained using ImageNet pre-trained weights on 100% of the training dataset

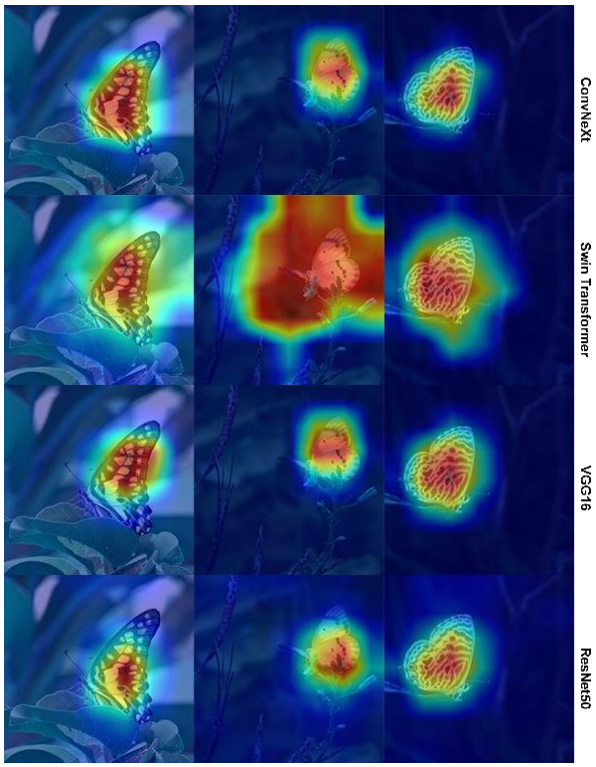

Fig. 11 GradCAM visualization of various encoder models trained using SiamFly pre-trained weights on 30% of the training dataset

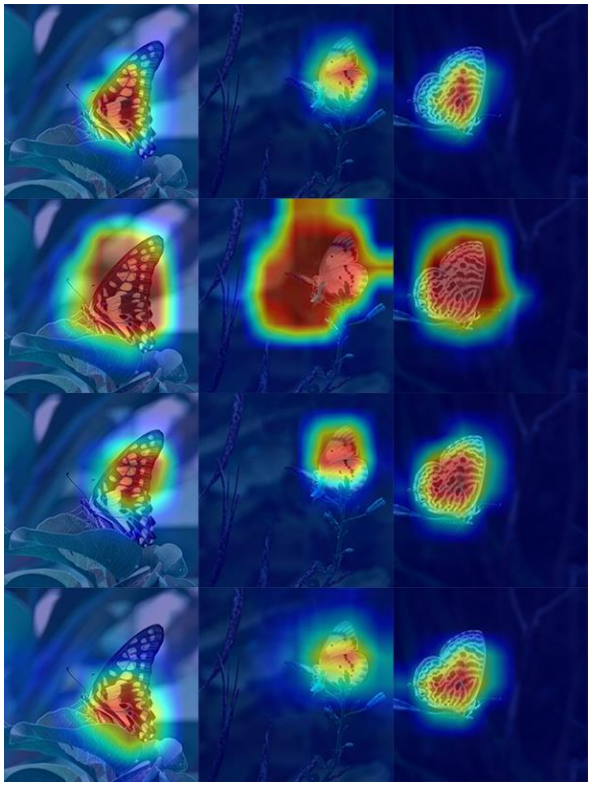

Fig. 12 GradCAM visualization of various encoder models trained using SiamFly pre-trained weights on 100% of the training dataset

4.4 Limitations and future scope of SiamFly-based pre-training

Our proposed method has shown a few important results that assist us in training a relatively better model compared to the ImageNet-based pre-trained model with fewer data. However, there are many areas for improvement. The embedding size of the encoder network is an essential hyper-parameter. The optimal choice of the size of embedding should be explored in future works. The squeeze and excitation strategy should be explored as the strategy induces an attention mechanism to extract relevant features. First, the encoder's output embeddings should be connected to a linear layer with a reduced number of parameters followed by a linear layer with an equal or higher number of parameters equal to the embedding size. A residual connection can also be introduced between the encoder's embedding and the excite squeeze output. To extract only the relevant information about the class, U-NET (Ronneberger et al., 2015) based models use this strategy for the instance and panatopic segmentation tasks. The effect of the number of bottle-neck layers after the encoders should be explored in detail as the backpropagation on a larger parameter can have a relevant effect in earning image representation. Currently, the ReLU activation function has been used in the pre-final layers of the SiamFly architecture. However, the choice of various loss functions needs to be assessed and validated.

4.5 Overall shortcomings of the butterfly classification models

Although our proposed model has many advantages, the butterfly classification failed in a few scenarios. For example, the best performing model ConvNeXt failed to identify the very closely related butterfly species belonging to genera Nacaduba, Euploea, Tirumala, Pelopidas and Tarucus. Some of the species from the above genera are difficult to identify with visual features even difficult for experts in a taxonomy without dissection. When an image of the new species other than the trained is passed as an input, the model results in a probability percentage that matches close to the trained data. However, the model is not able to classify the input as a new species. Hence, a distance-based similarity metric may be designed to measure the distance between the embeddings to discriminate the out-of-distribution data. This method can aid researchers in the discovery of new species. As the model's primary focus was on the multi-class single-label classification, it failed to detect the correct species when there were more than one butterfly species in the input image. Hence, a modification can be carried out in future work to enable multi-label detection in a single image. Our model also failed to detect the butterfly species when the butterfly's area in the image was very small. This is primarily due to the image being resized to a lower resolution of 224 px x 224 px, which reduced the finer essential features of the butterfly wings. As a result, the model may have been misclassified. In future studies, models should be trained and analyzed by cropping the images to different sizes so that all of the butterflies' details can be captured.

The previous studies (Krause et al., 2016; M. Li et al., 2021; Z. Lin et al., 2020; Nie et al., 2017; Rodner et al., 2015; Xiao et al., 2015) on fine-grained classification tasks were mostly carried out by modifying the pre-trained architecture with attention mechanisms or by ensembling several models/feature maps to reinforce the important features from the image. These modifications can sometimes make the model highly complex. Also, the earlier fine-grained studies on butterflies have reported results from only small datasets with a limited number of images and classes. The performance of these studies on a larger dataset with a highly imbalanced dataset is highly doubtful. However, from our experiments, we prove that adopting the SN-based architecture can also be used for fine-grained image classification with better performance over a largely unbalanced dataset. In future studies, a detailed ablation study can be done to compare and analyze the performance of various fine-grained classification problems.

5. Conclusion

A novel pre-training and transfer-learning technique based on the SiamFly network was proposed, which significantly improves model generalization and evaluation performance on the butterfly dataset. We introduce a large-scale, research-grade, fine-grained butterfly image

dataset comprising 62,287 images corresponding to 686 butterfly species. We demonstrate that our proposed method outperforms an ImageNet-based pre-trained linear classifier, which can aid in the transfer-learning task. We also show how the proposed technique helps the fully-supervised task learn the important features despite having limited data. We demonstrate that our proposed method outperforms an ImageNet-based pre-trained linear classifier, which can aid in the transfer-learning task. We also show how our proposed pre-training technique helps the fully-supervised task learn the important features despite having limited data. With the help of GradCAM feature maps, we visualize and analyze the learned image representations of different models. In the future, we plan to expand this project with different self-supervised learning-based approaches based on the proposed architecture and also carry out a detailed ablation study on the encoder's projection size, the number of bottle-neck layers, and various loss functions, and the effect of data-augmentation. We also will focus on introducing attention-based feature extraction for fine-grained butterfly classification.

Declarations

Funding

All the works were self-funded by the authors and no financial support was received from any organization for the submitted work.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

Almryad, A. S., & Kutucu, H. (2020). Automatic identification for field butterflies by convolutional neural networks. *Engineering Science and Technology, an International Journal*, *23*(1), 189–195. https://doi.org/10.1016/j.jestch.2020.01.006

Becker, S., & Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*(6356), 161–163. https://doi.org/10.1038/355161a0

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging Properties in Self-Supervised Vision Transformers. *CoRR*, *abs/2104.1*.

https://arxiv.org/abs/2104.14294

Carvajal, J., Romero, D., & Sappa, A. (2017). Fine-Tuning Based Deep Convolutional Networks for Lepidopterous Genus Recognition. *Lecture Notes in Computer Science*, *10125*, 467–475. https://doi.org/10.1007/978-3-319-52277-7_57

Chang, Q., Qu, H., Wu, P., & Yi, J. (2017). *Fine-Grained Butterfly and Moth Classification Using Deep Convolutional Neural Networks*. https://www.semanticscholar.org/paper/Fine-Grained-Butterfly-and-Moth-Classification-Deep-Chang-Qu/4cf2045b811c9e0807f9c94fc991566a6f5adbf4

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (Vol. 119, pp. 1597–1607). PMLR. https://proceedings.mlr.press/v119/chen20j.html

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020). Big Self-Supervised Models are Strong Semi-Supervised Learners. *CoRR*, *abs/2006.1*. https://arxiv.org/abs/2006.10029

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., & Valko, M. (2020). Bootstrap Your Own Latent: {A} New Approach to Self-Supervised Learning. *CoRR*, *abs/2006.0*. https://arxiv.org/abs/2006.07733

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., & Tao, D. (2022). A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. https://doi.org/10.1109/TPAMI.2022.3152247

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. B. (2019). Momentum Contrast for Unsupervised Visual Representation Learning. *CoRR*, *abs/1911.0*. http://arxiv.org/abs/1911.05722

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Heppner, J. B. (2008). Butterflies (Lepidoptera: Rhopalocera). In J. L. Capinera (Ed.), *Encyclopedia of Entomology* (pp. 623–626). Springer Netherlands. https://doi.org/10.1007/978-1-4020-6359-6_497

Howard, J., & Gugger, S. (2020). Fastai: A Layered API for Deep Learning. *Information*, *11*(2), 108. https://doi.org/10.3390/info11020108

Kang, S.-H., Cho, J.-H., & Lee, S.-H. (2014). Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network. *Journal of Asia-Pacific Entomology*, *17*(2), 143–149. https://doi.org/https://doi.org/10.1016/j.aspen.2013.12.004

Kang, S.-H., Jeon, W., & Lee, S.-H. (2012). Butterfly species identification by branch length similarity entropy. *Journal of Asia-Pacific Entomology*, *15*(3), 437–441. https://doi.org/10.1016/j.aspen.2012.05.005

Kaya, Y., & Kayci, L. (2014). Application of artificial neural network for automatic detection of butterfly species using color and texture features. *The Visual Computer*, *30*(1), 71–79. https://doi.org/10.1007/s00371-013-0782-8

Kaya, Y., Kayci, L., Tekin, R., & Faruk Ertuğrul, Ö. (2014). Evaluation of texture features for automatic detecting butterfly species using extreme learning machine. *Journal of Experimental & Theoretical Artificial Intelligence*, *26*(2), 267–281. https://doi.org/10.1080/0952813X.2013.861875

Kaya, Y., Kayci, L., & Uyar, M. (2015). Automatic identification of butterfly species based on local binary patterns and artificial neural network. *Applied Soft Computing*, *28*, 132–137. https://doi.org/https://doi.org/10.1016/j.asoc.2014.11.046

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised Contrastive Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 18661–18673). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. arXiv 2014. *ArXiv Preprint ArXiv:1412.6980*.

Koch, G., Zemel, R., Salakhutdinov, R., & others. (2015). Siamese neural networks for one-shot image recognition. *ICML Deep Learning Workshop*, *2*, 0.

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., & Fei-Fei, L. (2016). The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision -- ECCV 2016* (pp. 301–320). Springer International Publishing.

Lazebnik, S., Schmid, C., & Ponce, J. (2004). Semi-local Affine Parts for Object Recognition. In A. Hoppe, S. Barman, & T. Ellis (Eds.), *British Machine Vision Conference (BMVC '04)* (pp. 779–788). The British Machine Vision Association (BMVA). https://hal.inria.fr/inria-00548542

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, F., & Xiong, Y. (2018). Automatic identification of butterfly species based on HoMSC and GLCMoIB. *The Visual Computer*, *34*(11), 1525–1533. https://doi.org/10.1007/s00371-017-1426-1

Li, M., Zhou, G., Cai, W., Li, J., Li, M., He, M., Hu, Y., & Li, L. (2021). MRDA-MGFSNet: Network Based on a Multi-Rate Dilated Attention Mechanism and Multi-Granularity Feature Sharer for Image-Based Butterflies Fine-Grained Classification. *Symmetry*, *13*(8). https://doi.org/10.3390/sym13081351

Li, M., Zhou, G., Cai, W., Li, J., Li, M., He, M., Hu, Y., & Li, L. (2022). Multi-scale Sparse Network with Cross-Attention Mechanism for image-based butterflies fine-grained classification. *Applied Soft Computing*, *117*, 108419. https://doi.org/https://doi.org/10.1016/j.asoc.2022.108419

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Lin, Z. (2019). *Butterfly Specimens*. IEEE Dataport. https://doi.org/10.21227/baeg-m011

Lin, Z., Jia, J., Gao, W., & Huang, F. (2020). Fine-grained visual categorization of butterfly specimens at sub-species level via a convolutional neural network with skip-connections. *Neurocomputing*, *384*, 295–313. https://doi.org/https://doi.org/10.1016/j.neucom.2019.11.033

Liu, Ze, Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Liu, Zhuang, Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

M., G., Ravi, V., V, S., E.A, G., & K.P, S. (2021). Explainable Deep Learning-Based Approach for Multilabel Classification of Electrocardiogram. *IEEE Transactions on Engineering Management*, 1–13. https://doi.org/10.1109/TEM.2021.3104751

Nie, L., Wang, K., Fan, X., & Gao, Y. (2017). Fine-Grained Butterfly Recognition with Deep Residual Networks: A New Baseline and Benchmark. *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1–7. https://doi.org/10.1109/DICTA.2017.8227435

Rödder, D., Schmitt, T., Gros, P., Ulrich, W., & Habel, J. C. (2021). Climate change drives mountain butterflies towards the summits. *Scientific Reports*, *11*(1), 14382. https://doi.org/10.1038/s41598-021-93826-0

Rodner, E., Simon, M., Brehm, G., Pietsch, S., Wägele, J.-W., & Denzler, J. (2015). Fine-grained Recognition Datasets for Biodiversity Analysis. *CoRR*, *abs/1507.0*.

http://arxiv.org/abs/1507.00913

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*, 618–626.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv Preprint ArXiv:1409.1556*.

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. *2017 {IEEE} Winter Conference on Applications of Computer Vision, {WACV} 2017, Santa Rosa, CA, USA, March 24-31, 2017*, 464–472. https://doi.org/10.1109/WACV.2017.58

Theivaprakasham, H. (2021). Identification of Indian butterflies using Deep Convolutional Neural Network. *Journal of Asia-Pacific Entomology*, *24*(1), 329–340. https://doi.org/https://doi.org/10.1016/j.aspen.2020.11.015

Tsai, C.-C., Childers, R. A., Nan Shi, N., Ren, C., Pelaez, J. N., Bernard, G. D., Pierce, N. E., & Yu, N. (2020). Physical and behavioral adaptations to prevent overheating of the living wings of butterflies. *Nature Communications*, *11*(1), 551. https://doi.org/10.1038/s41467-020-14408-8

Varshney, R. K., & Smetacek, P. (2015). *A Synoptic Catalogue of the Butterflies of India* (R. K. V. & P. Smetacek (ed.); First). Butterfly Research Centre, Bhimtal & Indinov Publishing. https://doi.org/10.13140/RG.2.1.3966.2164

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, *7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q.-V., Padannayil, S. K., & Simran, K. (2020). A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities. *IEEE Transactions on Industry Applications*, *56*(4), 4436–4456. https://doi.org/10.1109/TIA.2020.2971952

Wang, J., Markert, K., & Everingham, M. (2009). Learning Models for Object Recognition from Natural Language Descriptions. *BMVC*.

Wepprich, T., Adrion, J. R., Ries, L., Wiedmann, J., & Haddad, N. M. (2019). Butterfly abundance declines over 20 years of systematic monitoring in Ohio, USA. *PLOS ONE*, *14*(7), 1–21. https://doi.org/10.1371/journal.pone.0216270

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., & Zhang, Z. (2015, June). The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-Grained Image Classification. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xie, J., Hou, Q., Shi, Y., Peng, L., Jing, L., Zhuang, F., Zhang, J., Tang, X., & Xu, S. (2018). The Automatic Identification of Butterfly Species. *ArXiv Preprint ArXiv:1803.06626*.

Xin, D., Chen, Y.-W., & Li, J. (2020). Fine-Grained Butterfly Classification in Ecological Images Using Squeeze-And-Excitation and Spatial Attention Modules. *Applied Sciences*, *10*(5). https://doi.org/10.3390/app10051681

Yadav, P., Menon, N., Ravi, V., & Vishvanathan, S. (2021). Lung-GANs: Unsupervised Representation Learning for Lung Disease Classification Using Chest CT and X-Ray Images. *IEEE Transactions on Engineering Management*, 1–13. https://doi.org/10.1109/TEM.2021.3103334

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *CoRR*, *abs/2103.0*. https://arxiv.org/abs/2103.03230

Zhao, R., Li, C., Ye, S., & Fang, X. (2019). Butterfly Recognition Based on Faster R-CNN. *Journal of Physics Conference Series*, *1176*, 32048. https://doi.org/10.1088/1742-6596/1176/3/032048