Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Bornean Orangutan Nest Classification using Image Enhancement with Convolutional Neural Network and Kernel Multi Support Vector Machine Classifier

Amanda Aiza Amran[1], Chin Kim On[1,*], Samsul Ariffin Abdul Karim[1], Lai Po Hung[1], Chai Soo See[2], Donna Simon[3], Munirah Rossdy[4], Chi Jing[5]

[1] Faculty of Computing and Informatics, Creative Advanced Machine Intelligence (CAMI) Research Centre, Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
[2] Faculty Computer Science and Information Technology, 94300 Kota Samarahan, Sarawak, Malaysia
[3] Orangutan Conservation, WWF-Malaysia, 88000 Kota Kinabalu, Sabah, Malaysia
[4] Faculty of Computer and Mathematical Science, Universiti Teknologi MARA, 88997 Kota Kinabalu, Sabah, Malaysia
[5] Hebei University of Engineering, Handan, Hebei 056038, China

ARTICLE INFO

Article history:
Received 8 December 2023
Received in revised form 11 May 2024
Accepted 15 June 2024
Available online 31 July 2024

Keywords:

Bornean Orangutan classification; convolutional neural networks; image classification; image enhancement; support vector machine

ABSTRACT

Preserving wildlife habitats is crucial in mitigating climate change. Species like orangutans and monkeys contribute to fruiting and planting in forests. The World Wide Fund Sabah Malaysia faces challenges in manually identifying and classifying orangutan nests for studying their behaviour and conserving their habitats. To address this, we propose automating the classification of captured images using machine learning algorithms. This research involves three key components: image processing, feature extraction, and image classification. Our proposed image processing includes several steps, such as image pre-processing and enhancement techniques like local contrast enhancement, sharpening, intensity adjustment, histogram equalization, and colour thresholding. We applied four different Convolutional Neural Networks (CNNs) to extract and identify orangutan nests' features. Subsequently, we utilize Support Vector Machine (SVM) for image classification. The results reveal that the Inception Residual Network Version 2 (ResNet-v2) achieves the best performance. This architecture is then combined with a kernel SVM to classify Bornean orangutan nests. Our approach demonstrates impressive results, boasting an accuracy of 96.60%, an F1-score of 96.60%, a precision of 96.59%, and a recall of 96.58%. These metrics underscore the high accuracy and effectiveness of our proposed methodology for classifying Bornean orangutan nests. By reducing the need for extensive human intervention in image analysis, our method presents a valuable tool for conservationists and researchers committed to studying and safeguarding these endangered orangutans and their habitats. In future work, we aim to develop orangutan nest detector, contributing to wildlife conservation research.

[*] *Corresponding author.*
E-mail address: kimonchin@ums.edu.my

https://doi.org/10.37934/araset.49.2.187204

1. Introduction

The International Union for Conservation of Nature (IUCN) has listed the Bornean orangutan, or Pongo Pygmaeus as one of the critically endangered animals [1]. Wildlife researchers and Non-Government Organizations (NGOs) such as the World Wide Fund (WWF) have been making many efforts to estimate the population for several years using spotting the nest in the forest [2,3] in order to monitor their population. Simon *et al.,* [3] have been using aerial imagery to spot the Bornean orangutan nest from the top view. The aerial imagery captured using drones and Unmanned Aerial Vehicles (UAVs) is high resolution, but small objects are very blurry, particularly for the Bornean orangutan nest, due to various challenges. The challenges of viewing Bornean orangutan nests from the top view using aerial imagery are foliage cover, camouflage and blend-in, nest concealment, nest size and scale, and nest degradation [4]. Thus, a technique is required to improve the quality of the images before useful information can be extracted from the sources. Image processing plays critical role and it is a fundamental step of image recognition research such as object detection [5], remote sensing [6] and object classification [7]. Surprisingly, there is no research conducted thus far to recognize Bornean orangutan nests using machine learning algorithms.

In this work, the objectives are to design an image processing algorithm to improve the quality of the data source captured using UAV, to extract the orangutan's nests features using several Convolutional Neural Networks (CNNs), and lastly to train and test the Bornean orangutan nest using Support Vector Machine (SVM). In image classification, there are many important steps, which include image labelling and image segmentation. For this work, we propose image classification, which includes edge threshold, sharpening, intensity adjustment and colour thresholding L*a*b*. To test the effectiveness of image classification, we extract the images' features using several CNNs and then classified the results using Kernel Multi SVM. Lastly, the results are compared and discussed. The subsequent sections of this article are organized as follows: Section 2 discusses related research, Section 3 elaborates on the methodology, Section 4 presents the experimental results, Section 5 covers the discussions, and Section 6 concludes the paper.

2. Literature Review
2.1 Wildlife Image Processing

Aerial photography is a powerful tool in various domains, including environmental monitoring [8] and wildlife analysis [9]. However, aerial images often suffer from challenges such as low contrast [10], noise [11], and inconsistent lighting conditions [12], which can hinder the identification and analysis of important features like animal nests. To overcome these limitations, researchers have explored various image processing techniques specifically tailored for aerial photography [13].

The quality of wildlife images significantly affects subsequent processing tasks [14]. Recent advancements in camera technology have allowed researchers to capture high-resolution images with increased accuracy and detail [15]. The use of high precision equipment, such as camera traps [16] and UAVs [17], has enabled efficient and non-intrusive data collection. Researchers have also explored techniques to automate image acquisition using motion-triggered cameras [18] and sensor networks [19], to capture images in a continuous and unobtrusive manner. Pre-processing techniques are applied to enhance the quality of wildlife images and improve subsequent processing steps. Noise reduction [20], contrast adjustment [21], and colour correction [22] algorithms are commonly used to improve image quality. Segmentation techniques are crucial for separating wildlife objects from complex backgrounds. Various approaches, including thresholding [23], edge detection [24], and clustering algorithms [25], have been applied to identify and extract wildlife regions accurately. Additionally, advanced techniques like deep learning-based segmentation [26] have shown promising results in segmenting wildlife objects with high precision and recall rates.

Once wildlife objects are segmented, object recognition techniques are applied to identify and classify species. Traditional approaches relied on handcrafted features and machine learning algorithms, such as SVM [27] and Random Forests (RF) [28]. However, with the advent of deep learning, CNNs have demonstrated remarkable performance in species classification tasks [29]. By training on large annotated datasets, CNN models can learn discriminative features [30] and generalise well to unseen wildlife images [31], enabling accurate and automated species identification [32].

2.2 Convolutional Neural Network (CNN) as Feature Extractor

Feature extraction plays a critical role in machine learning tasks, enabling the transformation of raw data into meaningful representations that capture relevant information [33]. CNNs have emerged as powerful tools in computer vision, demonstrating remarkable capabilities in feature extraction from various types of data [34].

The evolution of CNNs is characterised by significant milestones that have shaped their architecture and performance. The pioneering work of Lecun *et al.,* [35] introduced the LeNet architecture, which demonstrated the efficacy of CNNs in character recognition. Subsequent advancements, such as the development of ResNet [36], have significantly improved the depth and capacity of CNNs, enabling them to extract more complex and abstract features. The introduction of transfer learning further revolutionised the field by leveraging pre-trained models and enabling the transfer of learned features to new tasks, even with limited labelled data.

The architecture of CNNs is designed to extract hierarchical features from input data by utilizing filters in convolutional layers to conduct localized operations, capturing spatial details and acquiring knowledge about spatial structures in a hierarchical manner [37]. Following the convolutional layers, there are pooling layers that down sampling the data, promoting spatial invariance and decreasing computational complexity [38]. Non-linear activation functions introduce non-linearity to the network, enabling the extraction of more complex features [39]. Finally, fully connected layers are responsible for classification or regression tasks, making predictions based on the extracted features.

CNNs have achieved remarkable success in image classification tasks. CNN architecture such as Inception-ResNet [40], have pushed the boundaries of image recognition accuracy. These networks leverage deep architectures, skip connections, and parallel operations to extract discriminative features from images. Additionally, transfer learning approaches have been widely used, enabling the adaptation of pre-trained models to new tasks, even with limited labelled data. The combination of deep architectures, transfer learning, and large-scale datasets has propelled CNNs to achieve state-of-the-art performance in image classification.

2.3 Kernel Support Vector Machines (SVMs) as Classifier

Kernel SVMs have emerged as a potent machine learning method for performing classification tasks across diverse domains [41]. Kernel SVM is a supervised learning algorithm that performs binary classification by finding an optimal hyperplane that maximally separates data points belonging to different classes [42]. Unlike linear SVM, kernel SVM allows for non-linear decision boundaries by transforming the input data into a high-dimensional feature space using a kernel function [43]. The decision boundary is then computed in this transformed feature space, enabling the classification of complex and non-linear patterns.

The choice of kernel function plays a crucial role in the performance of kernel SVM. Various kernel functions, such as linear, polynomial, Radial Basis Function (RBF), and sigmoid, have been proposed and extensively studied. The linear kernel is suitable for linearly separable data, while polynomial and RBF kernels [44] effectively handle non-linear data. The sigmoid kernel is often used for similarity-based learning tasks [45]. Researchers have explored the impact of different kernel functions on classification accuracy, computational efficiency, and generalisation ability, leading to advancements in kernel function selection and design [46].

Kernel SVM involves the selection of optimal model parameters, including the regularisation parameter (C) and the kernel-specific parameters (such as the degree of polynomial kernel or the gamma parameter for RBF kernel) [47]. Various techniques have been proposed for model selection and parameter tuning, including cross-validation, grid search, and genetic algorithms [48]. These approaches aim to balance model complexity and generalisation, ensuring the SVM achieves optimal performance on unseen data. Feature selection and dimensionality reduction techniques play a vital role in enhancing the efficiency and accuracy of kernel SVM [49]. Feature selection methods aim to identify the most informative features for classification, thereby reducing the input space's dimensionality and improving the model's generalisation ability [50]. Several studies have explored integrating these techniques with kernel SVM to enhance classification performance.

3. Methodology

3.1 Data Source and Sampling

Collecting orangutan nest images is challenging and costly. Therefore, we collaborated with WWF Malaysia, Sabah branch, for the Bornean orangutan nest detection project. The researchers from WWF Malaysia have shared their dataset, which consists of 746 photos captured using high-spec drones. The drone images include Bornean orangutan nests, dead trees (brown trees), branches, buildings, automobiles, and other trees. Dead trees and Bornean orangutan nests having similar characteristics, nearly identical colour and roughly identical shape. Consequently, identifying orangutan nests without expert assistance from WWF is very challenging. Due to limited Bornean orangutan’s nest, data augmentation was applied. Data augmentation is a technique commonly used in machine learning and computer vision to increase the size and diversity of a training dataset artificially. It involves applying various transformations or modifications to the existing data samples, resulting in new, slightly altered versions of the original data.

For sampling, we have created a dataset of 5,000 images by cropping and augmented the UAV images. This dataset includes 1,000 images of Bornean orangutan nests, 1,000 images of brown trees and branches, 1,000 images of buildings, 1,000 images of automobiles, and 1,000 images of trees without any nests. The data is then sent for image enhancement before features are extracted. Data is trained and tested later using raw data, raw data without background information, and enhanced data without background information for each CNN architecture.

The decision to compare vehicles and structures to orangutan nests rather than bird nests, which may resemble gorilla nests more, was made for a number of reasons, including structural dissimilarity to make clear distinctions. Structures such as cars and buildings may differ significantly from orangutan nests. This difference can aid in creating a classification model that is more reliable. The uniqueness of the structural elements may make it easier to distinguish between orangutan nests and artificial constructions. Furthermore, more difficult classification scenarios could arise from using cars and structures as comparisons. Differentiating between artificial structures like automobiles and houses and natural habitats like orangutan nests could demonstrate the model's ability to distinguish between disparate elements.

3.2 Image Enhancement Frameworks

As depicted in Figure 1, the acquired images undergo an initial augmentation process. This step is to generate additional images, enhancing the extraction capabilities and overall robustness of the training model. Next, these images go through local contrast enhancement to further refine their quality. Local contrast enhancement aims to improve the visibility and definition of edges in an image by adjusting the contrast local as seen in Eq. (1). By considering the local contrast characteristics, this technique enhances not only the edges but also other details and textures in the image.

$$E_{PV} = I + (I - L)\, E_F \tag{1}$$

$E_{PV}$is Enhanced Pixel Value, and it represents the adjusted pixel value after local contrast enhancement. The I is the input which is the original image or data that will undergo the process. L is the Local Mean Value which represents the mean value of the pixels within a local neighbourhood centred around the pixel being processed. It measures the difference in pixel values between neighbouring pixels within a local neighbourhood. E_F is the Enhancement Factor which determines the magnitude of contrast enhancement applied to a pixel. It is typically a positive scalar value greater than 1. In this process, is the edge threshold value where the constant number 0.5 is being set.

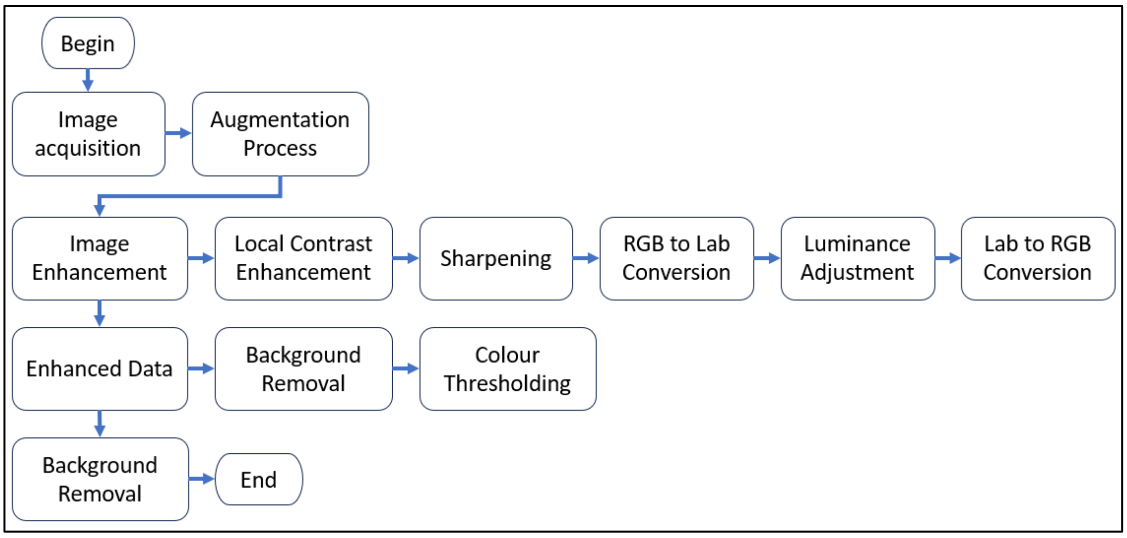


Fig. 1. The proposed image enhancement framework

Then, the images are applied for sharpening process to further enhance the details and edges in the image refer to Eq. (2). Sharpening is used to enhance an image's edges and fine details, making them appear more pronounced and visually appealing.

$$S = E_{PV} + (\, E_{PV} + B)A \tag{2}$$

S is the sharpened image, which represents the resulting image after applying the sharpening operation. E_{PV} is a locally contrasted image that has been improved from the previous process. Here, it acts as an input image to run the process of sharpening. B is the blurred image that represents a blurred version of the original image obtained by applying a blurring filter. A is the amount that determines the strength of the sharpening effect. It is typically a positive scalar value. This is the function where we put the amount of sharpening, radius, and threshold.

After sharpening the images, the sharpened image is converted from RGB to L*a*b*. The conversion is to enable colour-based analysis and adjustments. It involves manipulating the colour values to compensate for variations in lighting conditions, colour cast, or inaccuracies introduced during image capture, digitisation, or display. The resulting Lab image will have separate channels for the L, a, and b components, representing the luminance, green-red chromaticity, and blue-yellow chromaticity. This can be seen from Eq. (3) to Eq. (8).

$$CSI = RGBconvertLAB(S) \tag{3}$$

$$L = 116 \cdot f(Y/Y_{ref} - 16) \tag{4}$$

$$a = 500 \cdot [f(X/X_{ref} - f(Y/Y_{ref})] \tag{5}$$

$$b = 200 \cdot [f(Y/Y_{ref}) - f(Z/Z_{ref})] \tag{6}$$

L, a, and b are the Lab colour components. X, Y, and Z are tristimulus values of the original RGB image. Tristimulus refers to a concept used in colour science and colourimetry to describe the three independent quantities that are used to specify a colour sensation or stimulus in a three-dimensional colour space. Tristimulus refers to the three independent quantities used to specify a colour sensation or stimulus in a three-dimensional colour space, typically represented as X, Y, and Z values.

X_{ref}, Y_{ref}, and Z_{ref} are tristimulus values of the original RGB image. Tristimulus refers to a concept used in colour science and colourimetry to describe the three independent quantities that are used to specify a colour sensation or stimulus in a three-dimensional colour space. For the Luminance (L) equation in Eq. (4), the luminance component represents the image's brightness. Y is the tristimulus value of the image's luminance from the RGB colour space. Chromaticity (a) calculation in Eq. (5) is a component that represents the green-red chromaticity. X is the tristimulus value of the image's red channel from the RGB colour space. X_{ref} is the reference white point tristimulus value. Y_{ref} is the reference white point tristimulus value. Then, the resulting L*a*b* colour image is stored in a variable CSI. For now, the L is the main component that is needed to be focused on. Next, maximum luminosity is assigned to the value of 100. The maximum luminosity value is used as a scaling factor or reference point to normalise the luminance values within a specific range. The resulting values are typically mapped to a normalised range between 0 and 1 by dividing the luminance values by the maximum luminosity value. The reason why the max luminosity is 100 is to provide visually appealing results. t extracts the luminance values from the L*a*b* image's first channel (index 1) using the indexing operation shadowRGB (:,:,1). The extracted luminance values are then divided by the maximum luminosity value. All of these can be seen in Eq. (7).

In Eq. (7) and Eq. (8), $f(t)$ is a non-linear function defined as:

$$f(t) = t^{\frac{1}{3}} \ for\ t > \left(\frac{6}{29}\right)^3 \tag{7}$$

$$f(t) = \frac{t}{(3(6/29)^2)} + \frac{4}{29}\ ,\ otherwise \tag{8}$$

$f(t)$ is a non-linear function that transforms the tristimulus value to a perceptually uniform scale.
$f(t)$ is the same non-linear function as mentioned above.

$$Luminate\_Image = CSI(:,:,1)/max\_luminosity \tag{9}$$

The purpose of dividing the luminance values by the maximum luminosity is to scale the luminance values within a specific range. The luminance values are mapped to a normalised range between 0 and 1 by dividing by the maximum luminosity. This normalisation process is for contrast enhancement of the object that must be trained. The purpose of *L**a*b* also is to separate the colour information from the brightness information and create a colour space that is more perceptually uniform. This allows for more effective colour-based analysis and manipulation of the image.

Furthermore, the process of adjusting luminance, histogram equalization, and converting back to the RGB colour space is applied for enhancement and restoration. The purpose of adjusting luminance is for fine-tuning the luminance and achieving the desired visual enhancement in the RGB image. By modifying the luminance values, we can emphasise or de-emphasise certain image features, highlight details, or improve overall visual quality.

Then, we enhance the contrast and improve the overall visual appearance of an image using histogram equalization. It achieves this by redistributing the pixel intensity values of the image's histogram, effectively stretching the dynamic range to utilise the full available range of intensities. The first step in histogram equalisation is to compute the histogram of the input image as can be seen in Eq. (10).

$$H(i) = n(i) / (M * N) \tag{10}$$

H(i) represents the normalised histogram value at intensity level *i*, *n(i)* represents the number of pixels with intensity *i*, and (*M*, *N*) represents the dimensions of the image.

Next, the Cumulative Distribution Function (CDF) is computed. The *CDF* represents the accumulated probability of encountering a pixel with an intensity value up to a certain point. To ensure that the *CDF* spans the full range of intensity values (0 to 255 in an 8-bit grayscale image), the *CDF* is normalised by dividing each value by the total number of pixels in the image. The cumulative distribution function represents the cumulative sum of histogram values up to a certain intensity level, as seen in Eq. (11).

$$CDF(i) = \Sigma\, H(j) \text{ for } j = 0 \text{ to } I \tag{11}$$

where *CDF(i)* represents the cumulative distribution function value at intensity level *i*.

The next step is to transform the intensity values of the input image using the computed CDF. The transformed intensity value is denoted as *T(x, y)*, as seen in Eq. (12).

$$T(x, y) = CDF(I(x, y)) * L \tag{12}$$

Where *L* represents the maximum possible intensity level (e.g., 255 for an 8-bit image), the *CDF(I(x, y))* represents the value of the cumulative distribution function at the intensity level of the pixel *I(x, y)*.

Finally, the transformed image is normalised to the range *[0, L]* to obtain the final result, as seen in Eq. (13).

$$T\_norm(x, y) = (T(x, y) - min(T)) / (max(T) - min(T)) * L \tag{13}$$

where *T_norm(x, y)* represents the normalised transformed image, *min(T)* represents the minimum intensity value in the transformed image, and *max(T)* represents the maximum intensity value in the transformed image.

The resulting *T_norm(x, y)* represents the histogram-equalised image with enhanced contrast. Next, the equalised image is created by assigning the new intensity values to the corresponding pixels based on the mapping obtained in the previous step. Lastly, the input images are converted back to the RGB format and background removal is applied before objects are cropped.

3.3 Convolutional Neural Networks

CNNs draw inspiration from the visual cortex's organization and comprise layers: convolutional, pooling, fully connected, and activation layers. They learn hierarchical features from raw pixel values to interpret visual patterns. Convolutional layers extract local features, pooling layers preserve vital features, fully connected layers enable high-level representations and predictions, and activation functions like ReLU introduce non-linearity. CNNs excel in recognizing complex visual patterns through backpropagation during training.

In this study, we compare the performance of CNN-based ReLU with ResNet architectures (ResNet-18, ResNet-50, ResNet-101) and Inception ResNet-V2 for feature extraction. We specifically chose a three-layer CNN-based ReLU due to its characteristics and performance in computer vision tasks. These deep CNN architectures, including the three-layer version, effectively capture intricate features, allowing for accurate and discriminative representations.

CNNs offer several advantages over other feature extractors. One key benefit lies in their proficiency in local feature extraction, where convolutional layers enable effective identification of edges, textures, and forms within images. The second advantage is the ability of CNNs to extract hierarchical features, progressively learning more sophisticated representations of input images. Additionally, CNNs exhibit the third benefit of transferring learned features from pre-trained models, such as those trained on extensive datasets like ImageNet, to smaller datasets through fine-tuning. This feature proves particularly valuable when dealing with limited wildlife image databases, addressing challenges posed by small sample sizes. While architectures like Recurrent Neural Networks (RNNs) or Transformers may excel in sequential or language-related tasks due to their comparative efficiency, they might not be as well-suited for image-based tasks. This is because they prioritize temporal relationships or global context over local features within images. In contrast, CNNs prove more practical for processing extensive amounts of image-based wildlife data, showcasing computational efficiency in handling both the complexity and calculations associated with image data.

3.3.1 The ResNet-18

ResNet-18, a variant of the ResNet architecture introduced by Kaiming He *et al.,* [36], is widely adopted for image classification tasks. It comprises 18 layers, including convolutional, batch normalization, ReLU activation, max-pooling, and a fully connected layer for classification, organized sequentially. What sets ResNet-18 apart is its use of residual connections (skip connections), allowing information to flow directly from early layers to later ones, mitigating the vanishing gradient problem and facilitating training of deeper networks. Serving as a feature extractor, ResNet-18 transforms an input image into hierarchical feature maps, capturing both low-level and high-level information. Its pre-trained models are commonly used for transfer learning, leveraging learned weights to fine-tune on smaller datasets or specific tasks, ensuring efficient and accurate training even with limited data. Figure 2 shows the ResNet-18 architecture.

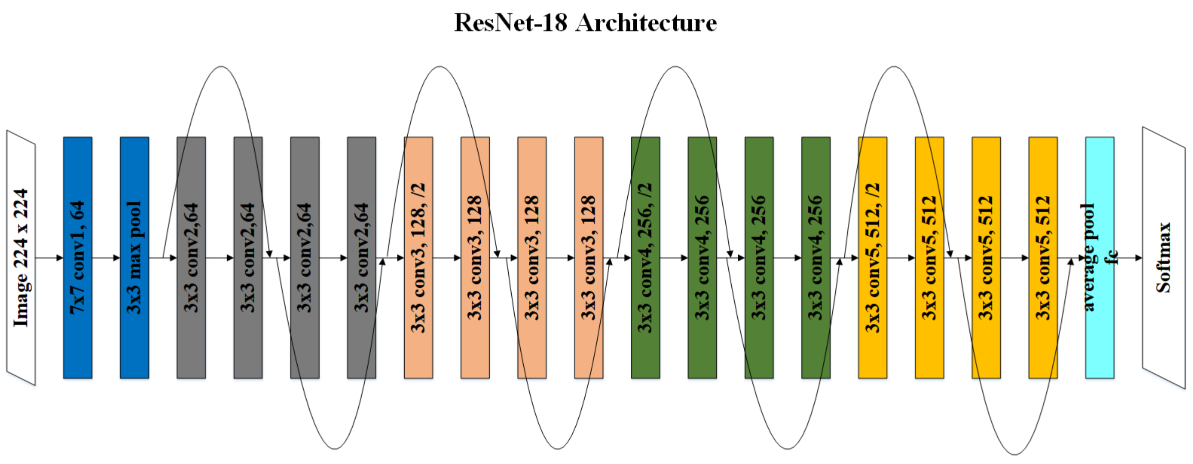


Fig. 2. ResNet-18 architecture

3.3.2 The ResNet-50

ResNet-50, with its 50 layers, is a deeper and more complex architecture compared to ResNet-18. It retains the same structure of residual connections and building blocks but incorporates additional layers to enhance its representational capacity. These building blocks include basic ones with two stacked 3x3 convolutional layers and more intricate bottleneck blocks, which use 1x1, 3x3, and 1x1 convolutions to reduce computational cost while maintaining representational power. Like ResNet-18, ResNet-50 employs down sampling via stride convolutions or max-pooling layers to decrease feature map dimensions progressively and increase channel numbers at specific stages in the network. The ResNet-50 architecture is shown in Figure 3 below.

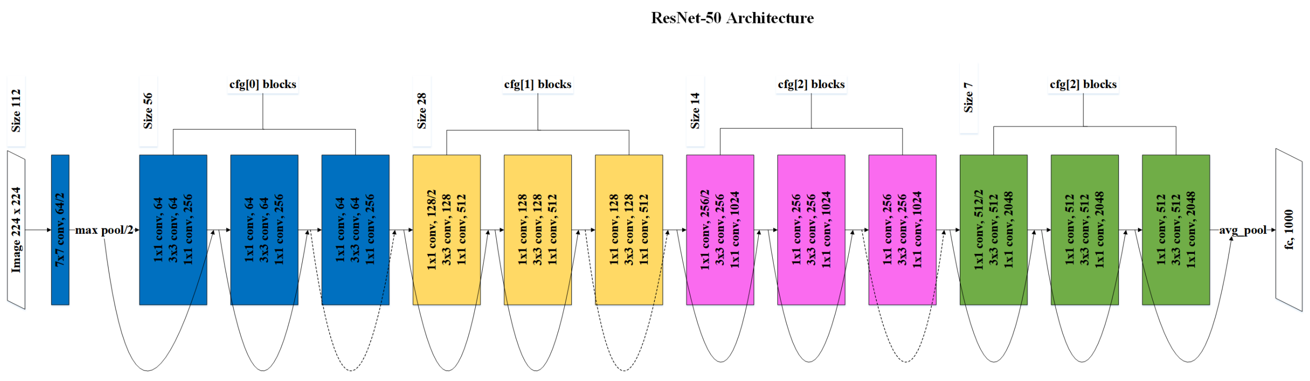

ResNet-50 Architecture
Size 112
Size 56
cfg[0] blocks
Size 28
cfg[1] blocks
Size 14
cfg[2] blocks
Size 7
cfg[2] blocks
Image 224 x 224
7x7 conv, 64/2
max pool/2
1x1 conv, 64 3x3 conv, 64 1x1 conv, 256
1x1 conv, 64 3x3 conv, 64 1x1 conv, 256
1x1 conv, 64 3x3 conv, 64 1x1 conv, 256
1x1 conv, 128/2 3x3 conv, 128 1x1 conv, 512
1x1 conv, 128 3x3 conv, 128 1x1 conv, 512
1x1 conv, 128 3x3 conv, 128 1x1 conv, 512
1x1 conv, 256/2 3x3 conv, 256 1x1 conv, 1024
1x1 conv, 256 3x3 conv, 256 1x1 conv, 1024
1x1 conv, 256 3x3 conv, 256 1x1 conv, 1024
1x1 conv, 512/2 3x3 conv, 512 1x1 conv, 2048
1x1 conv, 512 3x3 conv, 512 1x1 conv, 2048
1x1 conv, 512 3x3 conv, 512 1x1 conv, 2048
avg_pool
fc, 1000

Fig. 3. ResNet-50 architecture

3.3.3 The ResNet-101

ResNet-101, with its 101 layers, represents an even deeper and more potent iteration of the ResNet model, engineered to capture intricate and abstract image features. It retains the foundational structure of ResNet models, incorporating residual connections and building blocks, but extends its complexity for enhanced representational capacity. Employing residual connections, also known as skip connections, ResNet-101 facilitates direct information flow from earlier layers to later ones, mitigating the vanishing gradient problem and enabling effective training of exceptionally deep neural networks. Similar to other ResNet variants, it combines basic blocks, featuring two stacked

3x3 convolutional layers, and bottleneck blocks, which optimize computation by using 1x1, 3x3, and 1x1 convolutions. Serving as a feature extractor, ResNet-101 progressively extracts hierarchical representations from input images, encompassing both fine-grained details and high-level semantic information. These representations prove valuable in diverse tasks, including image classification, object detection, and image segmentation. Figure 4 shows the ResNet-101 architecture.

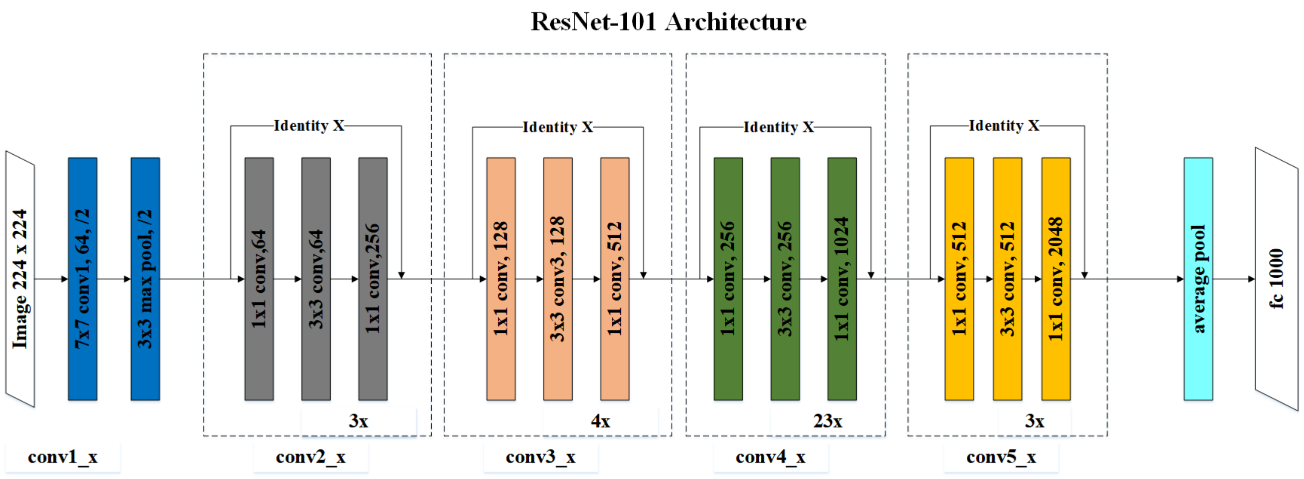


Fig. 4. ResNet-101 architecture

3.3.4 Inception ResNet-V2

Inception ResNet-V2 is an advanced CNN architecture that merges the strengths of the Inception module and residual connections. Building upon the Inception architecture, known for its effectiveness in capturing multi-scale features using parallel convolutional operations, it also incorporates residual connections inspired by the ResNet architecture to facilitate deep network training. The Inception module utilizes 1x1, 3x3, and 5x5 convolutions and max-pooling to capture features at various spatial scales, enabling effective handling of objects of different sizes. At the heart of Inception ResNet-V2 are the Inception-ResNet modules, which combine the Inception module with residual connections, enhancing feature extraction and gradient flow during training. To aid training and address the vanishing gradient issue, Inception ResNet-V2 includes auxiliary classifiers at intermediate network stages, introducing supplementary supervision signals and gradients to facilitate network training. The ResNet-V2 architecture is shown in Figure 5.

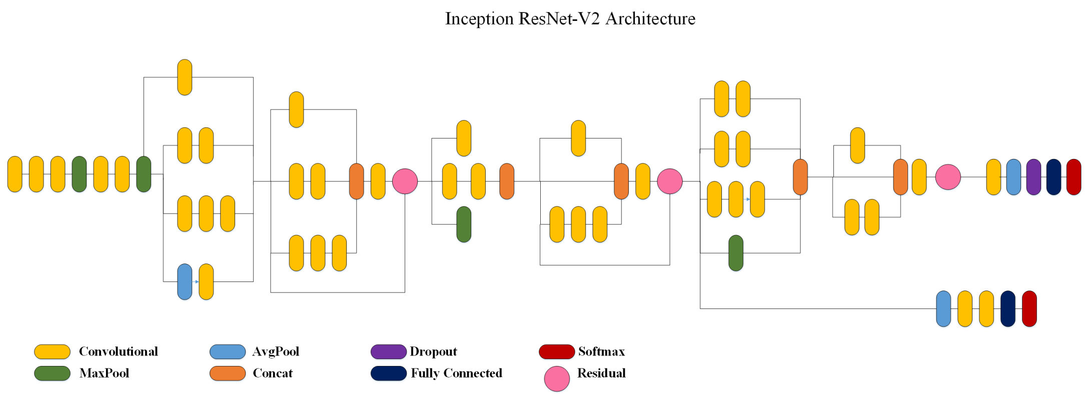


Fig. 5. Inception ResNet-V2 architecture

3.3.5 Experimental setup for CNNs

The total training dataset is divided into training, validation, and testing. Specifically, the training dataset comprises 70% of the total data, 20% for the validation dataset, and the testing dataset covers the remaining 10%. To initiate the training process, an initial learning rate of 0.01 is chosen. This selection prioritizes the stability of the training process, mitigating the risk of instability or divergence that can occur with higher learning rates like 0.1. It also aligns with best practices for fine-tuning pre-trained CNN models, where smaller learning rates are preferred to make precise adjustments to pre-learned features. Additionally, the lower learning rate promotes smoother convergence, reduces the risk of overshooting optimal solutions, and ensures consistent progress during training. During training, the data is shuffled, and the process continues for a maximum of 100 epochs or until convergence, whichever is reached earlier. Performance evaluations of the model are conducted at 30-epoch intervals. These settings adhere to the recommendations outlined in [36].

3.4 Kernel Multiclass Support Vector Machine (SVM)

SVM, a supervised machine learning algorithm, excels in classification and regression tasks by finding the optimal decision boundary or hyperplane that separates data points into classes or predicts continuous outcomes. SVM is renowned for its efficacy in handling high-dimensional data and its versatility in addressing both linear and non-linear classification challenges. In this study, we used multiclass SVM equations, utilizing the Error-Correcting Output Codes (ECOC) method. ECOC's core concept involves decomposing the original multiclass problem into a series of binary classification tasks, each tackled with a two-class SVM.

In this research, the SVM serves as the classifier, as illustrated in Figure 6 below. Initially, the input images undergo an enhancement process detailed in Section 3.2. Subsequently, feature extraction is carried out utilizing ResNet18, ResNet50, ResNet10, and Inception ResNet-V2. The CNN model that demonstrates superior performance is then fed into the Kernel SVM for classification.

Default settings were used for parameter configuration in the Kernel Multiclass SVM implementation. The choice to use default configurations was based on the recognition of the parameters' sensitivity as well as the lack of specialized knowledge and resources for thorough parameter adjustment. It is acknowledged that while default parameters provide a basic framework for implementing the model, they may not fully maximize performance for our particular dataset. This consideration draws attention to the possible influence on the model's capacity to identify fine-grained decision boundaries or capture complex data patterns.

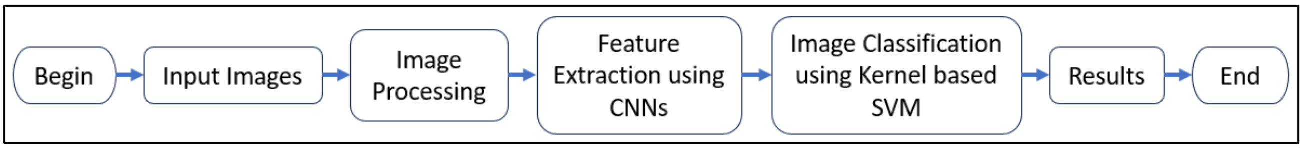
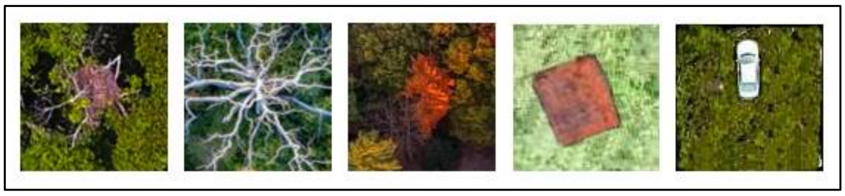
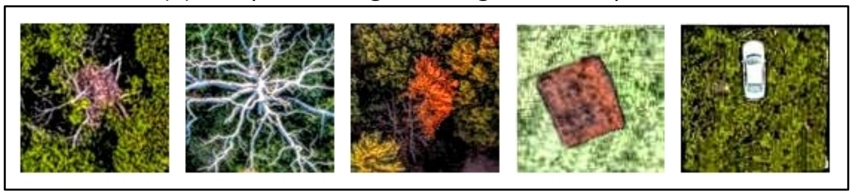

Begin
Input Images
Image Processing
Feature Extraction using CNNs
Image Classification using Kernel based SVM
Results
End

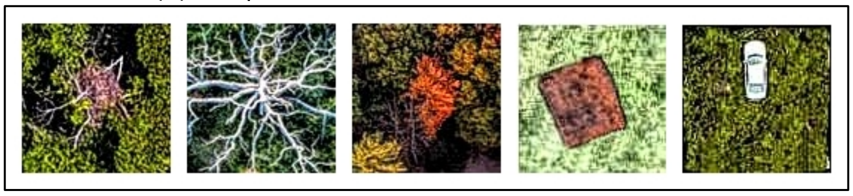
Fig. 6. The proposed hybrid CNN-SVM classification framework

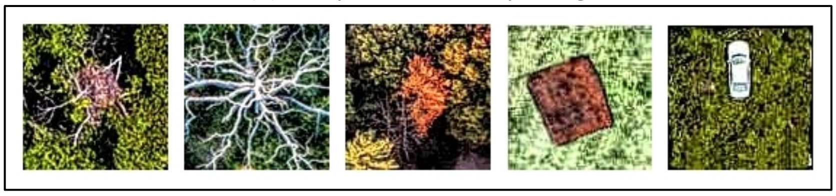
4. Results
4.1 Image Processing Results

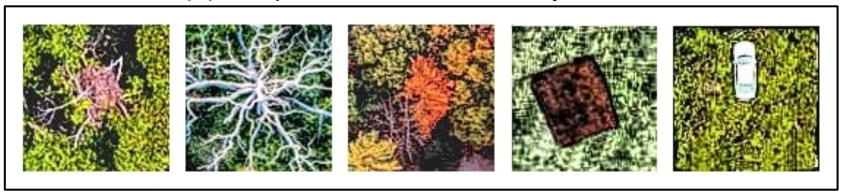
Figure 7 below shows the samples of UAV. Figure 7(a) shows the original images of orangutan nests, tree branches, other trees, buildings, and a car. In Figure 7(b), the enhanced quality of the samples after applying local contrast enhancement. Moving on to Figure 7(c), it shows further refinement of the samples, with an emphasis on enhancing the edges and fine details, resulting in a more pronounced and visually appealing appearance. Figure 7(d) proofs that the luminance adjustments algorithm has highlighted the image features, and overall visual quality. Figure 7(e) reveals the samples of images after undergoing histogram equalization, further enhancing their visual appeal. Finally, in Figure 7(f), it shows the samples after background removal.


(a) Samples of original images from top view


(b) Samples after local contrast enhancement


(c) Samples after sharpening


(d) Samples after luminance adjustment


(e) Samples after histogram equalization

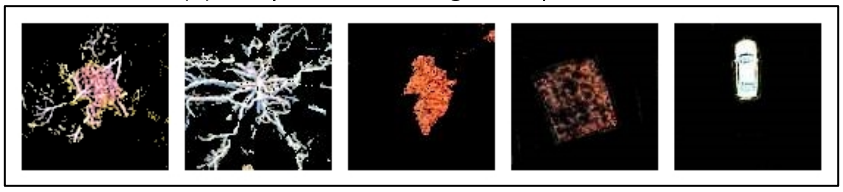

(f) Samples after background removal

Fig. 7. Samples of image processing results

4.2 The Comparison Results of CNNs Models

Table 1 summarizes model performance evaluated with metrics including accuracy, precision, recall, and F1-Score across three datasets: original images without backgrounds, enhanced images with backgrounds, and enhanced images without backgrounds. Initial training and testing on original images without backgrounds showcased Inception ResNet-V2 + Kernel SVM as the top performer, with ResNet-101 + Kernel SVM slightly trailing both ResNet-50 + Kernel SVM and Inception ResNet-V2 + Kernel SVM. Table 1 also presents some surprising results obtained when the models were trained and tested with enhanced images including backgrounds.

The results indicate that the ResNet-18 + Kernel SVM model outperformed both the ResNet-50 + Kernel SVM and ResNet-101 + Kernel SVM models. Moreover, the ResNet-18 + Kernel SVM model performed as well as the Inception ResNet-V2 + Kernel SVM model. In theory, both ResNet-50 + Kernel SVM and ResNet-101 + Kernel SVM should have outperformed the ResNet-18 + Kernel SVM, given their ability to capture deeper features.

Finally, the performance of the proposed models was compared using enhanced images without backgrounds. The results indicated that ResNet-18 + Kernel SVM was slightly better than both ResNet-50 + Kernel SVM and ResNet-101 + Kernel SVM. However, it performed just slightly worse than Inception ResNet-V2 + Kernel SVM. Additionally, ResNet-50 + Kernel SVM showed slightly better performance than ResNet-101 + Kernel SVM.

Table 1
Comparison of CNNs models + Kernel SVM using three different datasets

	Performance Metrics	ResNet-18 + Kernel SVM	ResNet-50 + Kernel SVM	ResNet-101 + Kernel SVM	Inception ResNet-V2 + Kernel SVM
Original images without Background	Accuracy	0.9120	0.9320	0.9240	0.9380
	Precision	0.9145	0.9337	0.9242	0.9425
	Recall	0.9120	0.9320	0.9240	0.9380
	F-1 Score	0.9132	0.9328	0.9242	0.9402
	Training Time	5 min 59 sec	19 min 53 sec	50 min 14 sec	63 min 20 sec
Enhanced images including background	Accuracy	0.9220	0.8840	0.8600	0.9220
	Precision	0.9235	0.8849	0.8618	0.9218
	Recall	0.9220	0.8840	0.8600	0.9220
	F-1 Score	0.9228	0.8845	0.8609	0.9219
	Training Time	6 min 3 sec	20 min 21 sec	45 min 52 sec	56 min 48 sec
Enhanced images without background	Accuracy	0.9620	0.9580	0.9460	0.9660
	Precision	0.9621	0.9583	0.9470	0.9659
	Recall	0.9620	0.9580	0.9460	0.9658
	F-1 Score	0.9621	0.9582	0.9465	0.9660
	Training Time	5 min 54 sec	20 min 28 sec	46 min 1 sec	57 min 2 sec

In terms of training time, the experimental results consistently demonstrated that ResNet-18 + Kernel SVM outperformed the other models in terms of speed. This can be attributed to ResNet-18 having fewer layers and parameters, resulting in reduced computational intensity during training. On the other hand, the slowest training time among the proposed models was observed for Inception ResNet-V2 + Kernel SVM. This was expected due to the complexity of its architecture. However, it's worth noting that the difference in training time between Inception ResNet-V2 + Kernel SVM and ResNet-18 + Kernel SVM was relatively small, with Inception ResNet-V2 + Kernel SVM being only a few minutes slower on average in the training results.

When considering the utilization of original images for training, validation, and testing across the proposed CNN architectures, it becomes evident that their performances, including metrics such as accuracy, recall, precision, and F1-scores, tend to be quite low. This phenomenon appears because the models trained on these images end up encompassing not only the objects of interest but also their backgrounds. Similarly, when working with enhanced images that retains the background, the training process tends to yield lower model performances. This occurs because the enhanced images contain additional values, including background information. However, a notable observation is that using original images without backgrounds results in a slight improvement in the training of each model. Consequently, this underscores the significance of using image enhancement techniques and background removal, facilitated by the Lab colour thresholding algorithm during image pre-processing. These steps are crucial to ensure that the model exclusively learns the designated target objects within the data.

Figure 8 displays the confusion matrix for the Inception ResNet-V2 + Kernel SVM model. The findings reveal that there was a single instance where a branch was misclassified as a building, eight cases where buildings were misclassified as cars, six occurrences where cars were incorrectly classified as buildings, one instance where a nest was wrongly labelled as a branch, and one more case where another tree was inaccurately classified as a branch.

branch	99 19.8%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	99.0% 1.0%
building	0 0.0%	93 18.6%	8 1.6%	0 0.0%	0 0.0%	92.1% 7.9%
car	0 0.0%	6 1.2%	91 18.2%	0 0.0%	0 0.0%	93.8% 6.2%
nest	1 0.2%	0 0.0%	0 0.0%	100 20.0%	0 0.0%	99.0% 1.0%
other trees	0 0.0%	0 0.0%	1 0.2%	0 0.0%	89 17.8%	99.0% 10.1%
	99.0% 1.0%	93.0% 7.0%	91.0% 9.0%	100.0% 0.0%	100.0% 0.0%	96.6% 3.4%
	branch	building	car	nest	other trees	

Targeted Class

Fig. 8. Confusion matrix for inception ResNet-V2 + Kernel SVM for Bornean orangutan nest classification

5. Discussions

In summary, the study's results shed light on various aspects of model performance for orangutan nest classification. In the initial assessment, Inception ResNet-V2 + Kernel SVM excelled with original images, showcasing its deep architecture's effectiveness. Surprisingly, ResNet-18 + Kernel SVM outperformed more complex models when tested on enhanced images with backgrounds, emphasizing the need to consider dataset-specific characteristics. However, in scenarios with enhanced images without backgrounds, Inception ResNet-V2 + Kernel SVM led the pack, underscoring the influence of image pre-processing techniques. Notably, ResNet-18 + Kernel SVM consistently outperformed other models in training speed due to its shallower architecture, while Inception ResNet-V2 + Kernel SVM took longer to train but with only a slight difference, suggesting its complex architecture doesn't hinder training efficiency significantly. The study also highlights the impact of image characteristics on model performance, where original images without backgrounds yielded the best results. This underscores the importance of proper image enhancement techniques and background removal during pre-processing. In conclusion, the research underscores the significance of thoughtful model selection, considering dataset nuances, and image pre-processing in wildlife image classification tasks. The discussion on confusion matrix insights and the trade-offs between model complexity and training time enriches our understanding, offering valuable insights for future studies in orangutan nest recognition.

6. Conclusions

This work introduces an innovative Bornean orangutan nest classification algorithm, harnessing the power of machine learning to identify nests from UAV-captured images. To elevate the quality of the input imagery, we have devised a novel image processing pipeline involving techniques such as edge thresholding, sharpening, intensity adjustment, histogram equalization, and colour thresholding within the L*a*b* colour space. Subsequently, we explored the efficacy of various CNN models as feature extractors, identifying Inception ResNet-V2 as the standout performer based on key metrics such as accuracy, precision, recall, F1-score, and proven from the confusion matrix. As a conclusion of our efforts, we have introduced a hybrid model, Inception ResNet-V2 + Kernel SVM, for the purpose of classifying Bornean orangutan nests and other objects. These advancements in wildlife image processing, feature extraction via CNNs, and classification with kernel SVMs collectively contribute to the domain's continued progress and hold potential for numerous applications in environmental and wildlife research. In the future, our goal includes the creation of a Bornean orangutan nest detecting technology to aid WWF researchers in the identification and localisation of these critical orangutan habitats.

Acknowledgement
This work was supported in part by the Universiti Malaysia Sabah under SDG02-2020.

References

[1] Ancrenaz, Marc, Melvin Gumal, Andrew J. Marshall, Erik Meijaard, Serge A. Wich, and Simon Husson. "Pongo pygmaeus (Bornean Orangutan)." (2016).

[2] Pandong, Joshua, Melvin Gumal, Lukmann Alen, Ailyn Sidu, Sylvia Ng, and Lian Pin Koh. "Population estimates of Bornean orang-utans using Bayesian analysis at the greater Batang Ai-Lanjak-Entimau landscape in Sarawak, Malaysia." *Scientific Reports* 8, no. 1 (2018): 15672. https://doi.org/10.1038/s41598-018-33872-3

[3] Simon, Donna, Glyn Davies, and Marc Ancrenaz. "Changes to Sabah's orangutan population in recent times: 2002–2017." *PloS One* 14, no. 7 (2019): e0218819. https://doi.org/10.1371/journal.pone.0218819

[4] Milne, Sol, Julien GA Martin, Glen Reynolds, Charles S. Vairappan, Eleanor M. Slade, Jedediah F. Brodie, Serge A. Wich, Nicola Williamson, and David FRP Burslem. "Drivers of bornean orangutan distribution across a multiple-use tropical landscape." *Remote Sensing* 13, no. 3 (2021): 458. https://doi.org/10.3390/rs13030458

[5] Devi, Ms RS Sandhya, VR Vijay Kumar, and P. Sivakumar. "A review of image classification and object detection on machine learning and deep learning techniques." In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1-8. IEEE, 2021. https://doi.org/10.1109/ICECA52323.2021.9676141

[6] Li, Haodong. "An overview on remote sensing image classification methods with a focus on support vector machine." In *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, pp. 50-56. IEEE, 2021. https://doi.org/10.1109/CONF-SPML54095.2021.00019

[7] Turay, Tolga, and Tanya Vladimirova. "Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey." *IEEE Access* 10 (2022): 14076-14119. https://doi.org/10.1109/ACCESS.2022.3147495

[8] Chen, Xiwen, Bryce Hopkins, Hao Wang, Leo O'Neill, Fatemeh Afghah, Abolfazl Razi, Peter Fulé, Janice Coen, Eric Rowell, and Adam Watts. "Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset." *IEEE Access* 10 (2022): 121301-121317. https://doi.org/10.1109/ACCESS.2022.3222805

[9] Qian, Yifei, Grant RW Humphries, Philip N. Trathan, Andrew Lowther, and Carl R. Donovan. "Counting animals in aerial images with a density map estimation model." *Ecology and Evolution* 13, no. 4 (2023): e9903. https://doi.org/10.1002/ece3.9903

[10] Ding, Jian, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie *et al.,* "Object detection in aerial images: A large-scale benchmark and challenges." *IEEE transactions on pattern analysis and machine intelligence* 44, no. 11 (2021): 7778-7796. https://doi.org/10.1109/TPAMI.2021.3117983

[11] Khan, Adnan, Muhammad Uzair Khattak, and Khaled Dawoud. "Object detection in aerial images: A case study on performance improvement." In *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*, pp. 1-9. IEEE, 2022. https://doi.org/10.1109/ICAIoT57170.2022.10121898

[12] Burdziakowski, Pawel, and Katarzyna Bobkowska. "UAV photogrammetry under poor lighting conditions—Accuracy considerations." *Sensors* 21, no. 10 (2021): 3531. https://doi.org/10.3390/s21103531

[13] Yurchuk, Iryna, Vladyslav Kovdrya, and Lolita Bilyanska. "Segmentation of Digital Images of Aerial Photography." In *2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, pp. 258-261. IEEE, 2019. https://doi.org/10.1109/APUAVD47061.2019.8943841

[14] Shaw, Meghan N., William T. Borrie, Emily M. McLeod, and Kelly K. Miller. "Wildlife photos on social media: A quantitative content analysis of conservation organisations' Instagram images." *Animals* 12, no. 14 (2022): 1787. https://doi.org/10.3390/ani12141787

[15] Yavari, Somayeh, Mohammad Javad Valadan Zoej, Mahmod Reza Sahebi, and Mehdi Mokhtarzade. "Accuracy improvement of high resolution satellite image georeferencing using an optimized line-based rational function model." *International journal of remote sensing* 39, no. 6 (2018): 1655-1670. https://doi.org/10.1080/01431161.2017.1410294

[16] Leorna, Scott, and Todd Brinkman. "Human vs. machine: Detecting wildlife in camera trap images." *Ecological Informatics* 72 (2022): 101876. https://doi.org/10.1016/j.ecoinf.2022.101876

[17] Petso, Tinao, Rodrigo S. Jamisola Jr, Dimane Mpoeleng, Emily Bennitt, and Wazha Mmereki. "Automatic animal identification from drone camera based on point pattern analysis of herd behaviour." *Ecological Informatics* 66 (2021): 101485. https://doi.org/10.1016/j.ecoinf.2021.101485

[18] Nazir, Sajid, and Muhammad Kaleem. "Advances in image acquisition and processing technologies transforming animal ecological studies." *Ecological Informatics* 61 (2021): 101212. https://doi.org/10.1016/j.ecoinf.2021.101212

[19] Norouzzadeh, Mohammad Sadegh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S. Palmer, Craig Packer, and Jeff Clune. "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning." *Proceedings of the National Academy of Sciences* 115, no. 25 (2018): E5716-E5725. https://doi.org/10.1073/pnas.1719367115

[20] Zhang, Luming, Guifeng Wang, Ming Chen, Fuji Ren, and Ling Shao. "An enhanced noise-tolerant hashing for drone object detection." *Pattern Recognition* 143 (2023): 109762. https://doi.org/10.1016/j.patcog.2023.109762

[21] Liu, Cheng-Chien. "Adaptive Contrast Enhancement of Optical Imagery Based on Level of Detail (LOD)." *Remote Sensing* 12, no. 10 (2020): 1555. https://doi.org/10.3390/rs12101555

[22] Huang, Mengxing, Jinjin Ye, Shenghan Zhu, Yang Chen, Yuanyuan Wu, Di Wu, Siling Feng, and Feng Shu. "An Underwater Image Color Correction Algorithm Based on Underwater Scene Prior and Residual Network." In *International Conference on Artificial Intelligence and Security*, pp. 129-139. Cham: Springer International Publishing, 2022. https://doi.org/10.1007/978-3-031-06788-4_11

[23] Zhao, Songwei, Pengjun Wang, Ali Asghar Heidari, Xuehua Zhao, and Huiling Chen. "Boosted crow search algorithm for handling multi-threshold image problems with application to X-ray images of COVID-19." *Expert Systems with Applications* 213 (2023): 119095. https://doi.org/10.1016/j.eswa.2022.119095

[24] Feng, Wenzhao, Junguo Zhang, Chunhe Hu, Yuan Wang, Qiumin Xiang, and Hao Yan. "A novel saliency detection method for wild animal monitoring images with WMSN." *Journal of Sensors* 2018 (2018). https://doi.org/10.1155/2018/3238140

[25] Zhang, Mingyu, Fei Gao, Wuping Yang, and Haoran Zhang. "Wildlife object detection method applying segmentation gradient flow and feature dimensionality reduction." *Electronics* 12, no. 2 (2023): 377. https://doi.org/10.3390/electronics12020377

[26] Jessel, Hadass R., Lior Aharoni, Sol Efroni, and Ido Bachelet. "A modeling algorithm for exploring the architecture and construction of bird nests." *Scientific Reports* 9, no. 1 (2019): 14772. https://doi.org/10.1038/s41598-019-51478-1

[27] Radhakrishnan, Saieshwar, and R. Ramanathan. "A support vector machine with Gabor features for animal intrusion detection in agriculture fields." *Procedia computer science* 143 (2018): 493-501. https://doi.org/10.1016/j.procs.2018.10.422

[28] Valavi, Roozbeh, Jane Elith, José J. Lahoz-Monfort, and Gurutzeta Guillera-Arroita. "Modelling species presence-only data with random forests." *Ecography* 44, no. 12 (2021): 1731-1742. https://doi.org/10.1111/ecog.05615
[29] Chen, Leiyu, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. "Review of image classification algorithms based on convolutional neural networks." *Remote Sensing* 13, no. 22 (2021): 4712. https://doi.org/10.3390/rs13224712
[30] Schlagenhauf, Tobias, Yiwen Lin, and Benjamin Noack. "Discriminative feature learning through feature distance loss." *Machine Vision and Applications* 34, no. 2 (2023): 25. https://doi.org/10.1007/s00138-023-01379-1
[31] Faizal, Sahil, and Sanjay Sundaresan. "Wild Animal Classifier Using CNN." *arXiv preprint arXiv:2210.07973* (2022).
[32] Favorskaya, Margarita, and Andrey Pakhirka. "Animal species recognition in the wildlife based on muzzle and shape features using joint CNN." *Procedia Computer Science* 159 (2019): 933-942. https://doi.org/10.1016/j.procs.2019.09.260
[33] Benyahia, Samia, Boudjelal Meftah, and Olivier Lézoray. "Multi-features extraction based on deep learning for skin lesion classification." *Tissue and Cell* 74 (2022): 101701. https://doi.org/10.1016/j.tice.2021.101701
[34] Barburiceanu, Stefania, Serban Meza, Bogdan Orza, Raul Malutan, and Romulus Terebes. "Convolutional neural networks for texture feature extraction. Applications to leaf disease classification in precision agriculture." *IEEE Access* 9 (2021): 160085-160103. https://doi.org/10.1109/ACCESS.2021.3131002
[35] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324. https://doi.org/10.1109/5.726791
[36] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016. https://doi.org/10.1109/CVPR.2016.90
[37] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9 (2018): 611-629. https://doi.org/10.1007/s13244-018-0639-9
[38] Zafar, Afia, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. "A comparison of pooling methods for convolutional neural networks." *Applied Sciences* 12, no. 17 (2022): 8643. https://doi.org/10.3390/app12178643
[39] Kulathunga, Nalinda, Nishath Rajiv Ranasinghe, Daniel Vrinceanu, Zackary Kinsman, Lei Huang, and Yunjiao Wang. "Effects of nonlinearity and network architecture on the performance of supervised neural networks." *Algorithms* 14, no. 2 (2021): 51. https://doi.org/10.3390/a14020051
[40] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1. 2017. https://doi.org/10.1609/aaai.v31i1.11231
[41] Cervantes, Jair, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. "A comprehensive survey on support vector machine classification: Applications, challenges and trends." *Neurocomputing* 408 (2020): 189-215. https://doi.org/10.1016/j.neucom.2019.10.118
[42] Awad, Mariette, Rahul Khanna, Mariette Awad, and Rahul Khanna. "Support vector machines for classification." *Efficient learning machines: Theories, concepts, and applications for engineers and system designers* (2015): 39-66. https://doi.org/10.1007/978-1-4302-5990-9_3
[43] Yalsavar, Maryam, Paknoosh Karimaghaee, Akbar Sheikh-Akbari, Mohammad-Hassan Khooban, Jamshid Dehmeshki, and Salah Al-Majeed. "Kernel parameter optimization for support vector machine based on sliding mode control." *IEEE Access* 10 (2022): 17003-17017. https://doi.org/10.1109/ACCESS.2022.3150001
[44] Patle, Arti, and Deepak Singh Chouhan. "SVM kernel functions for classification." In *2013 International conference on advances in technology and engineering (ICATE)*, pp. 1-9. IEEE, 2013. https://doi.org/10.1109/ICAdTE.2013.6524743
[45] Jiang, Han-Jing, Zhu-Hong You, and Yu-An Huang. "Predicting drug− disease associations via sigmoid kernel-based convolutional neural networks." *Journal of translational medicine* 17 (2019): 1-11. https://doi.org/10.1186/s12967-019-2127-5
[46] Achirul Nanda, Muhammad, Kudang Boro Seminar, Dodi Nandika, and Akhiruddin Maddu. "A comparison study of kernel functions in the support vector machine and its application for termite detection." *Information* 9, no. 1 (2018): 5. https://doi.org/10.3390/info9010005
[47] Han, Shunjie, Cao Qubo, and Han Meng. "Parameter selection in SVM with RBF kernel function." In *World Automation Congress 2012*, pp. 1-4. IEEE, 2012.
[48] Elgeldawi, Enas, Awny Sayed, Ahmed R. Galal, and Alaa M. Zaki. "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis." In *Informatics*, vol. 8, no. 4, p. 79. MDPI, 2021. https://doi.org/10.3390/informatics8040079

[49] Alhassan, Afnan M., and Wan Mohd Nazmee Wan Zainon. "Review of feature selection, dimensionality reduction and classification for chronic disease diagnosis." *IEEE Access* 9 (2021): 87310-87317. https://doi.org/10.1109/ACCESS.2021.3088613

[50] Khaire, Utkarsh Mahadeo, and R. Dhanalakshmi. "Stability of feature selection algorithm: A review." *Journal of King Saud University-Computer and Information Sciences* 34, no. 4 (2022): 1060-1073. https://doi.org/10.1016/j.jksuci.2019.06.012