Identification of Snake Species in Sri Lanka Using Convolutional Neural Networks

SB Abayaratne[1], WMKS Ilmini[1] and TGI Fernando[2]

[1]Deparment of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Rathmalana, Sri Lanka
[2]Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka

{saviniabayaratne@gmail.com, kalaniilmini@kdu.ac.lk, tgi@sjp.ac.lk}

Abstract. Snake bites in Sri Lanka cause death to nearly 100 people annually. Administering the appropriate anti-venom treatment for snake bite victims gets delayed causing complications as a result of the inability of people to identify the snake. Incorrect identification of snakes also causes threats to the existence of harmless snakes threatening the biodiversity of Sri Lanka. As a remedial measure to these problems, the first automatic snake identification from a given image using convolutional neural networks (CNN) is described in this study using 2000 images from each of six snake species found in Sri Lanka to train five CNN models. Four of the models were developed using the pre-trained architectures InceptionV3, VGG16, ResNet50 and MobileNet using transfer learning while the fifth model was developed from scratch. This study revealed that MobileNet with transfer learning yielding an accuracy of 90.5% is the most suitable model for automatic snake identification.

Keywords: convolutional neural networks, snakes, automatic snake identification, transfer learning, MobileNet

1 Introduction

Sri Lanka is home to about 100 types of snakes [1], [2]. Six of these snakes are considered as deadly venomous namely, Saw Scaled Viper, Ceylon Krait, Cobra, Common Indian Krait, Russell's Viper, Merrem's Hump-nosed Pit Viper with the latter four being responsible for most of the fatalities that take place due to snakebites in Sri Lanka [3]. Figure 1 shows images of five of the most venomous snakes (excluding the Sri Lankan Krait) and the most common non-venomous snake, the Rat Snake. For administering proper treatment for snakebites, it is essential to identify the snake type. In addition to this, misidentification of the snake type also leads to the slaughter of harmless snakes, which poses a threat to the biodiversity of the environment.

To ascertain the ability of people to correctly identify the snake type when an image is available, a survey was conducted through a questionnaire, and the results showed that a majority out of 233 respondents were unable to recognize the type of snake correctly. The key finding of the survey is illustrated in the bar chart of Figure

2. The mean score obtained for the questionnaire was 6.93 out of a total of 14. As a solution to minimize misidentification of the snake type, this paper describes the application of computer vision with deep learning for snake identification. Objective of this study is to minimize the harm caused to humans as well as snakes due to misidentification of snake types. To achieve the objective this work presents Convolutional Neural Network (CNN) based models developed to identify the type of snake when an image is given as an input to the model.

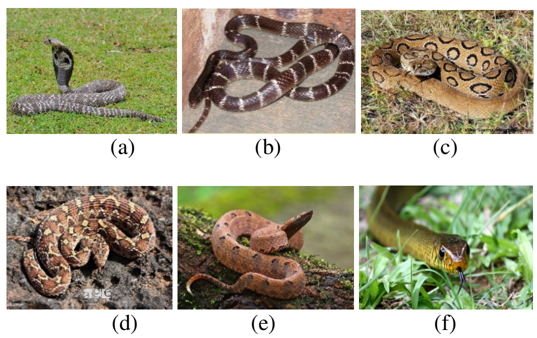

Fig. 1. Five of the most venomous snakes and the most common non-venomous snake in Sri Lanka. *(a)Cobra (b)Common Krait (c)Russell's Viper, (d)Saw Scaled Viper (e)Hump-nosed Pit Viper. (f)Rat Snake*

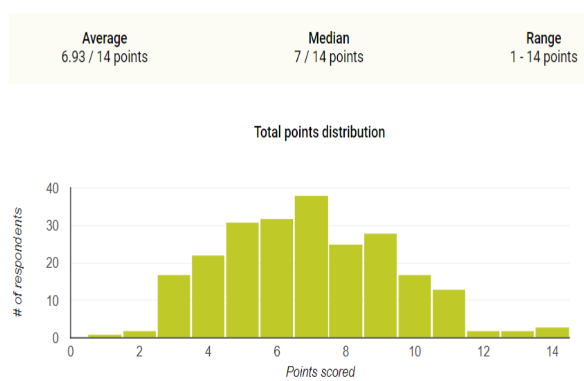

Fig. 2. Total Points Distribution of the Survey.

In Section 2 of this paper, a literature review on human, object or animal identification systems using CNN as well as other methods is presented. This is followed by a description of the methodology used in the study in Section 3 which includes a description of CNN and transfer learning methods, how transfer learning was adopted for this work and a model built from scratch to be compared with pre-trained models. An analysis of the results is given in Section 4 and Section 5 gives a discussion of the results. The conclusion of the study is given in Section 6.

2. Literature Survey

Various studies have been conducted for classifying different types of classes including images of sceneries, wild animals, farm animals, birds, marine species, fruits, vehicles, objects as well as age and gender of humans. These studies have employed various methods to achieve their aims and objectives. The afore-mentioned methods include image processing techniques, Radiofrequency Identification technology (RFID), Support Vector Machines (SVM), Neural Networks (NN), k-nearest neighbours' algorithm (KNN) as well as the proposed method in this proposal, Convolutional Neural Networks (CNN). All these classification methods have been implemented to reduce the cost incurred and the time consumed when it is done manually.

A study directly addressing the problem domain of this proposal has been conducted by A.P. James and others where thirteen probabilistic graphical models and 12 attribute enhancing methods have been used to identify the most relevant features in snakes for classification purposes [4]. The samples have contained images of 6 types of snakes found in India. Out of initially chosen 38 features, the results have shown that a total of 15 taxonomy features are sufficient to classify the type of the snake. The study has achieved the highest accuracy rate of 87.5% for IBk classifier [5] in identifying the required 15 features of the snake.

RFID technology has been used for identification of animals in farms mainly to identify and monitor cattle [6]. The RFID tags which have been placed on the cattle have not only been useful in detecting the animal but also in monitoring its health. Stating an accuracy rate of 75%, this study has hinted the use of NNs for future farm animal detection systems. A text message alert system integrated with RFID for identifying farm animals has been proposed by V.M. Anu and others in their Literature Review regarding RFID for farm animal identification [7]. As a solution for identifying many objects concurrently, a much more enhanced RFID method has been employed by H Vogt where around 30 tags were concurrently identified with an accuracy of 96% [8].

To identify the standard quality of tomatoes, a method using 5 image processing algorithms for correct texture, texture homogeneity, shape, stem and injury free identification, which has included operations such as morphology, has resulted in an accuracy of over 90% in correct texture, 80% in identifying defect-free tomatoes [9]. The colour of the tomatoes was identified using Mean Standard Deviation method, Slide Block method and Quad Tree method. Employing Sobel edge detection and evaluating the histograms obtained the homogeneity of colour for each tomato was detected. Similarly, the shape was also detected. A rule-based identification system has been implemented for all categories. All the image processing portions of the study has been implemented using MATLAB. Image processing techniques have also been employed in size identification of fruits and vegetables such as tomatoes and lemons [10]. In that particular study a software (ImageJ) which offers capabilities in calculation of means and detection of edges has been used.

SVM-based bird image identification system has resulted in an accuracy of over 98% in testing [11]. Bringing out a solution to the fact that images of birds are of various sizes as well as angles, this study has classified images of two kinds of bird species. Prior to classifying, image processing techniques such as edge detection using Sobel Operator and morphology have been applied as pre-processing steps. A similar system which has also used six image processing techniques as well as SVM to identify three types of animals, namely tiger, dog and cat has achieved respective highest accuracies of 94%, 93% and 93% for the edge histogram descriptors pre-processing technique [12].

An animal classification comparison has been carried out by a system using KNN and probabilistic NN (PNN) where the analysis of the image for segmenting has been done using a graph-cut method [13]. For a chosen 25 categories of animals with 64 block images, an accuracy of about 52% has been recorded for 70% portion of training when using PNN. In contrast, the KNN has shown an accuracy of about 60% for the same type of block images pointing out their recommendation of KNN.

NN approaches in classification and identification has been widely adapted by many researchers. ANN-based systems showcase efficiency as pre-processing of images are done separately from the network. Such a system, which has used ANN for the identification of gender with the use of features recognized in the face, while the extraction of features has been carried out using Viola Jones algorithm [14] been proposed by A Jaswante and others where an accuracy of about 98% has been recorded [15]. The efficiency of the proposed system has been pointed out by the researchers in this study stating that it will be of great use for a real time system because of its efficiency. Moving a step further, a gender identification system which also identifies age has been discussed in the work carried out by T Kalansuriya and others with the use of images of faces of people [16]. This system has used image processing techniques for pre-processing and feature extraction stages while the classification has been done using ANNs. Four age classes have been categorized to train and test the system for faces of both Asian and non-Asian continents. The human ability to identify the age and gender of a person has been quantitively compared with the system's ability for the same. Though gender identification is 100% in humans, the age identification is of an accuracy rate of about 67% while the proposed system has achieved accuracies of about 85% and 74% respectively for the same. Another ANN-based gender classification system which has used kinematic data of eight walking movement features to identify the gender of children has used algorithmically produced data as well as originally obtained data [17]. A comparison of the two datasets that have been trained using ANN has shown an accuracy increase of up to 86% when algorithmically produced data was also included.

Another type of NN known as CNN is rapidly moving forward to claim its spot in machine learning and classification problems. The proposed solution for the problem identified in this proposal is also based on CNNs and therefore CNN-based identification systems have been of primary focus in this literature review. The prominent work of the classification of ImageNet using a 5 layered CNN has included good quality images of 1000 categories where error rates of about 37% and 17% have been stated for the top most identification and 5 top most identifications [18]. With the use of many GPUs and max pooling layers in the CNN, this work is considered as ground breaking. Another general classification of images using CNN which has included the identification of images with faces against images without faces, images of buildings, images of sites of agriculture, images of highly populated urban areas and images of forests as well as images with sceneries such as images with beaches, gardens, streets, roads and battle sites against each other [19]. The best results were obtained for the identification of images with faces against images without faces which has been recorded as about 92% accurate on testing data while the lowest has been recorded as about 51%, again on testing data.

The previously discussed topic of age and gender classification has been tackled using CNN as well by G Levi and others, where the proposed system has surpassed the previous work that had been done at that time regarding gender and age classification by recording accuracies of 86.8% for gender identification and 50.7% for exact age identification and 84.7% for 1 category off age identification [20].

CNN-based systems have been proposed in identifying marine animals. One such study has been carried out to identify two types of fish where CNN is integrated with a set of hand-designed images [21]. With the use of DeCaf framework, the researchers have stated the overall error rate to be of about 1.38% when only CNN is used while 1.08% is the recorded error rate when both images are used together. In contrast to recognizing a specific specie type, a method for recognizing a specific individual animal has been proposed in a Minke Whale recognition study [22]. Using the stable unique colour pattern of each Minke Whale, a model has been developed to identify each individual whale using CNNs. An accuracy of 93% has been achieved which has been stated as a higher rate than that of a Gorilla recognition study.

The use of CNN has been very popular among studies based on wild life monitoring. Employing the well-known method of camera traps which are based on motions for collecting images, these studies have shown promising results further cementing CNN as an accurate approach for classification. One such study has stated comparative accuracies of three different CNN architectures employed in their work [23]. In the study the highest accuracy rate of 96.60% for animal detection has been observed for VGG-16 CNN architecture while those for recognition of the three most frequently found animals and the six most frequently found animals were shown to be 90.40% and 83.93% respectively for ResNet-50 CNN architecture. A CNN based wild animal identification system has shown a quantitative comparison between the system the work has proposed and the Bag-of-Words (BoW) model where segmentation has been carried out using graph-cuts [24]. Though the accuracy of this proposed system is higher than that of the BoW method, the accuracy rate has been recorded as about 38% which is comparatively a very low value for a CNN-based system. As the reasons for obtaining such low accuracies, the insufficiency in proper data and the small number of layers in the CNN can be pointed out. Yet another study which is very much similar to the former study discussed under animal classification using CNN, in which 48 types of species have been selected for identification, counting and for elaborating on their characteristics [25]. Out of the nine architectures that have been used to select the best achieving architecture, ResNet-152 has been identified as the best performing one. A whopping 95% accuracy rate has been recorded for the identification being in the top-5 while a 63% accuracy rate has been obtained for counting the animals.

All the papers reviewed are based on classification problems and 16 papers out of the 20 papers reviewed were either animal, human, marine species or bird classifications and only one paper was based on snake classification and even that was done using probabilistic graphical models. 17 of these research studies were classifications using images. Out of the 8-research work carried out using CNN, only 2 have accuracies below 80% which can be considered as successful. Only 2 research studies out of the 20 reviewed had been carried out in Sri Lanka and even those two are not on snake classification.

This review reveals that out of all automatic identification methods, CNN-based methods yield the most accurate results. Although ANN methods can also identify snake types, the accuracies are relatively less than that achieved through CNN methods and the pre-processing of the images is done inside the CNN classifier itself unlike in ANN. The only paper which discusses the classification of snakes describes the identification of the snake features that are most relevant rather than to classify them. The use of RFID is not practical when it comes to identifying the snake type since placing the RFID tag on a snake is impossible. The easiest way to identify snakes which differ from species to species by subtle features is by visual perception. When this is not possible, identification has to be made through an image. Classification of images using CNN gives higher accuracies compared to other machine learning approaches since a CNN learns to identify fine features through the training process.

3 Methodology

Initially, images of snakes were collected by taking photographs and videos of snake types described in Section 1 at the Dehiwela Zoological Gardens and from Google Images. The videos were augmented to obtain a large number of images. Altogether 250 images of each snake type were retrieved for processing. This was followed by removing as much of the background as possible by cropping the images. In order to obtain 2,000 images for each class, the mirror images were first obtained giving 500 images. Next the 90^0, 180^0 and 270^0 rotations of these 500 images were taken, giving a total of 2,000 images for each class.

Out of the 2,000 images in each class, 1,200 images were utilized for training the CNN models while 400 images were used for validating the model. The remaining 400 images were retained for testing the model. The selection was done adhering to the 60%, 20% and 20% image percentages for training, validation and testing respectively.

Training and validation datasets were input into the network to proceed with training. First all images in the training dataset were input into the network in batches of size 32 (32 images) and at the end of each epoch the validation dataset is used to validate the learning where adjustments are automatically made to the weights by the network after convolving the images before going through the next iteration.

Pre-trained models; InceptionV3 [26], VGG16 [27], ResNet50 [28], MobileNet [29] using ImageNet dataset as well as a model developed from scratch were used to train the models for the required classification of the 6 snake types employed in this study. Transfer learning and fine tuning were used with the pre-trained models while the sequential model offered by Keras [30] was used to train the model built from scratch. Training of all the models was done using TensorFlow [31] as the backend machine learning library while Keras, a python deep learning library, was used for the frontend.

Initially, the training was carried out using the original 250 images in each class. Since this number turned out to be insufficient, the sets of 2000 augmented images from each class were used for training. In order to improve the accuracy obtained, the cropped images were used for training after trying the raw images. To further improve the classification accuracy, the cropped images which were of varying dimensions, were inserted on a black square background of 200 x 200 pixels. This is expected to prevent image distortion since the pre-trained models scale an input image of any dimensions into a pre-defined size which depends on the type of pre-trained model. The model was trained using an Intel Core i7, 7th Generation CPU with a RAM of 8 GB.

After the completion of the training of the models, the testing dataset was used to evaluate the models. This dataset contained images that were not used in the training and validation processes. Therefore, the results provided at the end of the evaluation process depict an accuracy which can be considered to be very close to the actual accuracy that can be expected from the model when it is used in somewhat real scenarios.

3.1 Convolutional Neural Networks

Computer vision with deep learning has made significant advances with Convolutional Neural Networks (CNN). A CNN is an algorithm based on Deep Learning which can assign weights to different features/parts of an input image using suitable filters with very little pre-processing compared to other classification algorithms. CNN architecture resembles the connectivity pattern of neurons in the human brain and has been inspired by the layout of the visual cortex. CNN uses filters to segment images for easy processing without losing important features. By convolving the input image with a filter, a convolution layer is obtained for extracting low-level features such as sharp edges and colours. This layer with extracted features is reduced in size by a pooling (max pooling or average pooling) layer to reduce the computational power required for data processing. These two layers comprise one layer of the CNN. Based on the complexity of the image, a desired number of such layers can be incorporated into the network to train the model to further extract low-level features. This is followed by flattening the output in order to feed it to a regular fully-connected layer to classify images. Next, a fully-connected layer is added to the system to learn high-level features through nonlinear combinations of high-level

features computed by the convolution layer. This image is then flattened into a column vector and given as an input to a feed-forward neural network. Backpropagation is then applied to every iteration of the training process. The model acquires the ability to discern dominant and low-level features of images to classify the images through a succession of epochs [32].
In this research, the Softmax activation function is employed to classify the images as there are more than two classes (> 2 snake types) discussed in this study. The CNN architectures InceptionV3, VGGNet, ResNet and MobileNet have been developed.

3.2 Model built from Scratch

Using the sequential model given in Keras, a model was built from scratch. This model architecture was adjusted until a high accuracy without over-fitting was achieved. The final model which gave the highest accuracies out of the models built from scratch included four layers. Three inner layers were added one by one which included Conv2D layers, MaxPooling2D layers and Relu activation function. The final layer included a Dense layer with size 64 and Relu activation function, a BatchNormalization layer, a Dropout layer with a 0.4 Dropout ratio and a final Dense layer of size 6 with Softmax activation function.

3.3 Transfer Learning and Fine Tuning

Transfer learning is a method that utilizes already acquired knowledge for solving a new problem that resembles the solved problem where models which are trained previously using ImageNet dataset are used to train the model to identify images. As the first step, a base model of the pre-trained model (the model excluding the top most/last layer of the model) is compiled with the appropriate optimizers, loss function and metrics while having all the remaining layers set to *non-trainable*. A new last layer (classification layer) is added next, which is specific to the problem that is being addressed as the number of classes to which the classification is done differs from problem to problem. After transfer learning is applied to the model of the problem being addressed, the last layers of the network are fine tuned to refine the model for the particular problem. As mentioned in sub section 3.1, the first few layers of the model are used to identify the most basic types of patterns such as edges and boundaries. With the use of the initial layers of the pre-trained models such basic types of features can be identified without having to train the layers all over again. Hence, they are set to *non-trainable* when fine tuning the model. The knowledge acquired from the pre-trained model is used to identify these primary features.

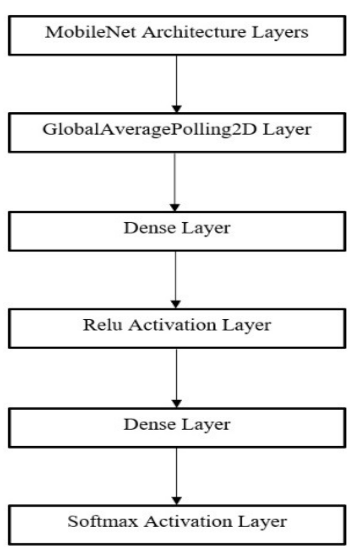

Fig. 3. Network architecture of the model developed using MobileNet.

The layers in the latter part of the network are set to *trainable* to enable the network to learn more advanced features of the images which are specific to the problem being dealt with but are different from the fine features of the originally trained models. Again, the model is compiled with an appropriate optimizer specifying the learning rate (which is another method of fine tuning) and specifying the loss function and metrics.

In this study as already mentioned, MobileNet pre-trained model was used for transfer learning and fine tuning. Setting up all layers except the final layer of the model as *non-trainable*, the model was compiled using Adam optimizer with categorical cross-entropy as the loss function (since there were more than two classes in this problem) and accuracy as the metric to evaluate the model. A new layer which includes a GlobalAveragePooling2D layer, a 128 sized Dense layer with Relu activation function and finally a Dense layer with size 6 (number of snake types to be classified) and the activation function Softamx was added to the previously compiled pre-trained model for transfer learning. After training the final layer of the model, it was fine-tuned using the training and validation datasets given as inputs to the network. In fine tuning, the first 20 layers of the model (taken after transfer learning) were frozen (weights set to *non-trainable*) and the remaining layers were set to *trainable* (to refine the model according to the problem addressed in this study). The model was compiled using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.0001 and a momentum of 0.9. The categorical cross-entropy loss function was again used with the accuracy metrics.
The network architecture of the final model which was accepted as the best model for this study is shown in Figure 3.

4 Data Analysis

A summary of the outcome of each of the models employed is given in Table 1 along with the number of epochs, training, validation and testing accuracies and losses. Most of the overfitting models were not tested using the test dataset as an overfitting model is not an accurate model.

The models given in Table 1 were trained and validated with different numbers of epochs. For the pretrained models, a few epochs (3-5) were sufficient to obtain high accuracies. For the new model developed from scratch, 35, 100, 200, 250 and 300 epochs were used for training and testing in 4 attempts in order to find an optimal number of epochs to achieve a significant accuracy and low loss rate. The table also contains data regarding the number of images each class had while training. Some details of these runs are given in the next section. After training and testing, the model was executed through a prediction code that predicted the snake type of a given image. The output also provides the probabilities that the given image had of being any of the six snake types predicted by the model.

As Table 1, attempt 17, shows, the highest accuracy using transfer learning was achieved for the MobileNet pre-trained model. Even though the attempts 12-14 of models built from scratch in Table 1 have obtained higher training and testing accuracies without getting overfitted, predictions were inaccurate whereas transfer learning obtained from the MobileNet pre-trained model along with the proper pre-processing techniques yielded correct predictions.

Table 1. Summary of the details of the models trained.
**IV3: Inception V3, **Built: Model built from scratch, ***RN: ResNet, ****MN: MobileNet*

Attempt	Model	Images per Class	No. of Epochs	Overfitting	Accuracy (%)		
					Training	Validation	Testing
1	IV3*	265	10	Yes	95.7	75.9	-
2	VGG16	265	10	Yes	88.1	64	-
3	Built**	265	10	Yes	52.1	45.3	-
4	Built	265	100	Yes	85.9	48.6	-
5	Built	350	100	Yes	90.3	54.7	-
6	Built	1000	300	Yes	79.8	74.7	-
7	Built	1000	300	Yes	94.2	74	-
8	Built	2000	300	Yes	96.3	90	-
9	Built	2000	100	Yes	92.6	84.3	85.8
10	Built	2000	100	Yes	92.6	86.5	88
11	Built	2000	35	No	84.5	79.8	80.6
12	Built	2000	100	No	92.5	89.8	91
13	Built	2000	200	No	93.9	93	93.8
14	Built	2000	250	No	96.5	95.1	96.3
15	IV3	2000	4	Yes	82.6	70.2	-
16	RN50***	2000	2	Yes	89.63	67.43	-
17	MN****	2000	3	No	91.7	89.9	90.5

The training and validation accuracy and loss graphs as functions of the number of epochs obtained for attempt 5, 8 and 17 stated in Table 1 can be seen in Figures 4.1, 4.2, 5.1, 5.2, 6.1 and 6.2 respectively.

Figures 4.1 and 4.2 show a significant overfitting in both accuracy and loss graphs for the attempt 5 in Table 1 where after a few epochs, the validation accuracy falls far behind the training accuracy while the validation loss is far higher than the training loss.

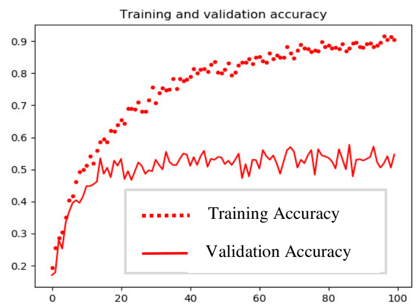

Fig. 4.1. Graphs of training and validation accuracies for attempt 5 of Table 1.

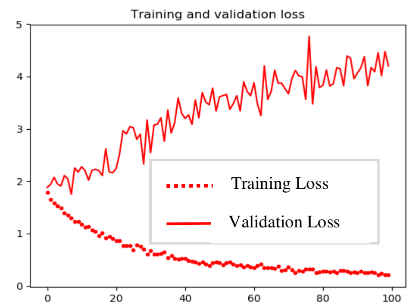

Fig. 4.2. Graphs of training and validation losses for the attempt 5 of Table 1.

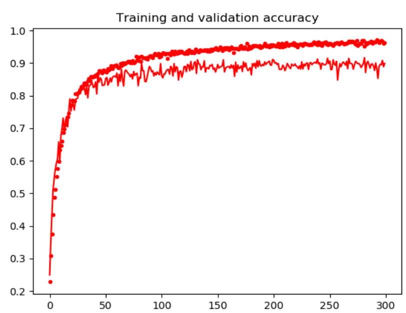

Fig. 5.1. Graphs of training and validation accuracies for attempt 8 of Table 1.

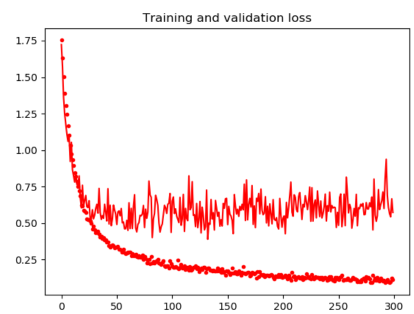

Fig. 5.2. Graphs of training and validation losses for attempt 8 of Table 1.

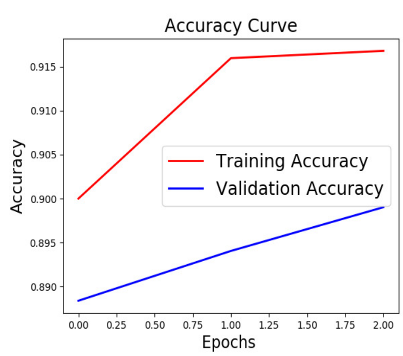

Fig. 6.1. Graphs of training and validation accuracies for attempt 17 of Table 1.

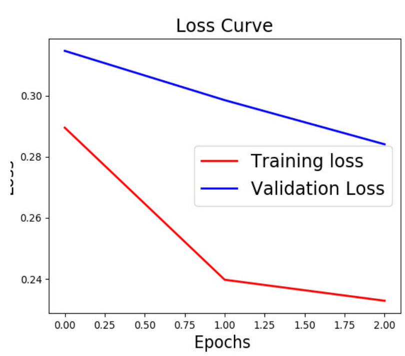

Fig. 6.2. Graphs of training and validation losses for attempt 17 of Table 1.

Figure 5.2 depicts a significant overfitting in the loss graphs for the attempt 8 with the validation loss getting much larger than the training loss unlike the accuracy graphs of it where the gap between training and validation accuracies is less than 10% as shown in Figure 5.1. Figure 6.1 shows that the gap between validation and training accuracies is about 2% and Figure 6.2 shows that the difference between training and validation losses is less than 0.06 for the final model obtained using the MobileNet pre-trained network.

The evaluation process carried out for the trained models using the testing dataset resulted in testing accuracies as shown under the testing accuracy in Table1. Some models were not evaluated as they gave low accuracies for training and validation at the end of the training process. The model developed using the MobileNet pre-trained network showed a testing accuracy of over 90%.

A prediction is illustrated in Figures 7.1 and 7.2. The snake in Figure 7.1 is a hump-nosed pit viper and the correct prediction given for it with an accuracy of 92.44% by the model trained in the attempt 17 of Table 1 (the model trained using the MobileNet pre-trained model) is shown in Figure 7.2.

Fig. 7.1. Image of a hump-nosed pit viper used for prediction.

```
Use tf.where in 2.0, which has the same broadcast rule as np.where
PIL image size (224, 224)
numpy array size (224, 224, 3)
image batch size (1, 224, 224, 3)
[[5.2515894e-05 1.8204021e-04 9.2435616e-01 3.4879163e-04 2.6991718e-02
  4.8068769e-02]]
[2]
Hump-nosed Pit Viper
```

Fig. 7.2. Image illustrating the predictions for Figure 7.1.

5 Discussion

As shown in Figure 4.1, the MobileNet, pre-trained model has reached a training accuracy rate exceeding 90% after 3 epochs for fine tuning. The newly developed model required nearly 250 runs to reach around the same testing accuracy rate. This difference can be explained by the large number of layers present in the architectures of the pre-trained models whereas the model developed from scratch included only 4 inner layers.

As shown in Table 1, in the early stages of training, the dataset contained only a fewer number of images per class. After using various augmentation techniques and increasing the number of images per class, better accuracies could be obtained. In most cases as shown in Figure 4.1, it is evident that overfitting of the models occurred much earlier in the training process when the dataset was comparatively smaller. After the attempt 5 of Table 1 the images in the dataset were cropped to remove the unnecessary background details that were present in most of the images. In the dataset used in attempts 14 – 17 of the Table 1 the images were padded to give each of the images a square shape which would not get distorted when giving them as inputs into the network for training as all the models (both pre-trained and built from scratch) require an image size specified before training to have all the images resized when the training process begins. Hence, before training the model, the input layer rescales each image according to the pre-specified image size without distorting the image. Disproportionately scaled images could lead to incorrect learning.

It was possible to significantly reduce the overfitting in the models built from scratch, after lowering the drop out percentage to 0.4 from 0.5 and adding a batch normalization layer right after the first Dense layer. Even though those models gave good training and testing accuracies, they yielded incorrect predictions. When the MobileNet pre-trained model was used with the correct pre-processing technique (pre-process input method defined for MobileNet) the model gave accurate predictions. Attempts 15 and 16 of Table 1 were made to terminate midway through the training as the model started to overfit. This was done to save time since it requires a long time even to run a single epoch for transfer learning with pre-trained models as they are quite complex in their network architecture with over 200 layers in some cases.

MobileNet was selected with the purpose of developing a mobile application which contains the trained model. A mobile application will be rather useful in accurately identifying snake types. MobileNet is the most suitable architecture for the purpose since it is light-weighted and therefore, can be easily supported in mobile phones in contrast to other models. Furthermore, the impact on the accuracy of the model when the size of the trained model is reduced to be compatible with mobile development purposes is extremely low when using MobileNet compared to other models.

Images of the Sri Lankan Krait given as the input were predicted as the Common Krait as expected due to the similarity between the two snakes. It was difficult to find a large number of images of the Sri Lankan Krait which is an endemic species which compelled the elimination of the Sri Lankan Krait from the study. In most cases, the prediction probability for the correctly identified snake was high.

6 Conclusion

No work on automatic snake identification methods using CNN has been reported in the literature. This research fills that gap and reveals that CNN methods are suitable in identifying the subtle differences in the appearance of different snake types. Usually identification of the snake type is done using blood samples of the victim and with the introduction of the method proposed in this work, the type of the snake can be identified instantaneously without having to wait for a blood sample to be tested for Enzyme Immunoassay (EIA) [33] and also whenever a person encounters a snake he or she will be aware of the type of the snake that he is dealing with and therefore, will be more cautious and will refrain from trampling. The results obtained using the MobileNet pre-trained model with transfer learning to identify the six snake types in this work, provide a high accuracy rate of 90.5% illustrating that this model is suitable for snake classification. Since transfer learning offers the ability to use pre-trained models with correctly specified pre-processing techniques it can be identified as the better method to train the CNN model that is developed in this study.

Deploying the model as a mobile application would be of use to people in identifying snakes and thereby aiding in administering appropriate treatment in cases of snakebites to reduce the morbidity and mortality rates of snake bite victims. It can also prevent harming innocent snake types. Therefore, using the light-weighted MobileNet pre-trained model with the use of transfer learning can be identified as the best method in trying to develop a CNN model that has the ability to classify the six types of snakes specified in this study.

7 Future Work

This study can be extended to include other snake types found in Sri Lanka. The accuracy of the network can be further improved with the increment of images per each class. A mobile application that uses the developed model can be very useful in serving the purpose of this study. The mobile app to be developed could be further modified to show details regarding each of the identified snake type and indicate any first-aid that needs to be carried out in case of an emergency. For it to be extremely useful in a case of snakebite, the mobile application can be further enhanced to give directions to reach the nearest government hospital for prompt treatment.

Acknowledgement

The authors would like to sincerely thank Mr. Sumedha Abeysinghe, Curator, Herpetarium, Dehiwela Zoological Garden, for facilitating taking images of snakes and Mr. Shavin Fernando for taking videos and photos of the snakes.

References

1. Das, I., de Silva, A.: Snakebite and its Management. In: SNAKES AND OTHER REPTILES OF SRI LANKA. p. 15. New Holland Publishers (2005).
2. Sunday Times: Snake bites - Are we paying enough attention?, https://www.pressreader.com/, (2017).
3. Malik, D.: Snakebite: Prevention and First Aid, http://slma.lk/wp-content/uploads/2017/11/sbc_fa_english.pdf, (2013).
4. James, A.P., Mathews, B., Sugathan, S., Raveendran, D.K.: Discriminative histogram taxonomy features for snake species identification. Human-centric Computing and Information Sciences. 4, (2014). https://doi.org/10.1186/s13673-014-0003-0.
5. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based Learning Algorithms. Machine Learning. 6, 37–66 (1991). https://doi.org/10.1023/A:1022689900470.
6. Eradus, W.J., Jansen, M.B.: Animal identification and monitoring. Computers and Electronics in Agriculture. 24, 91–98 (1999). https://doi.org/10.1016/S0168-1699(99)00039-3.
7. Anu, V.M., Deepika, M.I., Gladance, L.M.: Animal identification and data management using RFID technology. In: International Confernce on Innovation Information in Computing Technologies. pp. 1–6. IEEE, Chennai, India (2015). https://doi.org/10.1109/ICIICT.2015.7396069.
8. Vogt, H.: Efficient Object Identification with Passive RFID Tags. In: Mattern, F. and Naghshineh, M. (eds.) Pervasive Computing. pp. 98–113. Springer Berlin Heidelberg, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-45866-2_9.
9. Laykin, S., Alchanatis, V., Fallik, E., Edan, Y.: IMAGE–PROCESSING ALGORITHMS FOR TOMATO CLASSIFICATION. Transactions of the ASAE. 45, (2002). https://doi.org/10.13031/2013.8838.
10. Lino, A.C.L., Sanches, J., Fabbro, I.M.D.: Image processing techniques for lemons and tomatoes classification. Bragantia. 67, 785–789 (2008). https://doi.org/10.1590/S0006-87052008000300029.
11. Roslan, R., Nazery, N.A., Jamil, N., Hamzah, R.: Color-based bird image classification using Support Vector Machine. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). pp. 1–5. IEEE, Nagoya (2017). https://doi.org/10.1109/GCCE.2017.8229492.
12. Shalika, A.W.D.U., Seneviratne, L.: Animal Classification System Based on Image Processing & Support Vector Machine. Journal of Computer and Communications. 04, 12 (2016). https://doi.org/10.4236/jcc.2016.41002.
13. Kumar, Y.H.S., Manohar, N., Chethan, H.K.: Animal Classification System: A Block Based Approach. Procedia Computer Science. 45, 336–343 (2015). https://doi.org/10.1016/j.procs.2015.03.156.
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. pp. I-511-I–518. IEEE Comput. Soc, Kauai, HI, USA (2001). https://doi.org/10.1109/CVPR.2001.990517.
15. Jaswante, A., Khan, A., Gour, B.: Gender Classification Technique Based on Facial Features using Neural Network. 4, 5 (2013).

16. Kalansuriya, T.R., Dharmaratne, A.T.: Neural Network based Age and Gender Classification for Facial Images. ICTer. 7, (2014).
17. Zakaria, N.K., Jailani, R., Tahir, N.M.: Application of ANN in Gait Features of Children for Gender Classification. Procedia Computer Science. 76, 235–242 (2015). https://doi.org/10.1016/j.procs.2015.12.348.
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM. 60, 84–90 (2017). https://doi.org/10.1145/3065386.
19. Jaswal, D., V, S., Soman, K.P.: Image Classification Using Convolutional Neural Networks. International Journal of Scientific and Engineering Research. 5, 1661–1668 (2014). https://doi.org/10.14299/ijser.2014.06.002.
20. Levi, G., Hassncer, T.: Age and gender classification using convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 34–42. IEEE, Boston, MA, USA (2015). https://doi.org/10.1109/CVPRW.2015.7301352.
21. Cao, Z., Principe, J.C., Ouyang, B., Dalgleish, F., Vuorenkoski, A.: Marine animal classification using combined CNN and hand-designed image features. In: OCEANS 2015 - MTS/IEEE Washington. pp. 1–6. IEEE, Washington, DC (2015). https://doi.org/10.23919/OCEANS.2015.7404375.
22. Konovalov, D.A., Hillcoat, S., Williams, G., Birtles, R.A., Gardiner, N., Curnock, M.I.: Individual Minke Whale Recognition Using Deep Learning Convolutional Neural Networks. Journal of Geoscience and Environment Protection. 06, 25–36 (2018). https://doi.org/10.4236/gep.2018.65003.
23. Nguyen, H., Maclagan, S.J., Nguyen, T.D., Nguyen, T., Flemons, P., Andrews, K., Ritchie, E.G., Phung, D.: Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 40–49. IEEE, Tokyo, Japan (2017). https://doi.org/10.1109/DSAA.2017.31.
24. Chen, G., Han, T.X., He, Z., Kays, R., Forrester, T.: Deep convolutional neural network based species recognition for wild animal monitoring. In: 2014 IEEE International Conference on Image Processing (ICIP). pp. 858–862. IEEE, Paris, France (2014). https://doi.org/10.1109/ICIP.2014.7025172.
25. Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., Clune, J.: Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. PNAS. 115, E5716–E5725 (2018). https://doi.org/10.1073/pnas.1719367115.
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826. IEEE, Las Vegas, NV, USA (2016). https://doi.org/10.1109/CVPR.2016.308.
27. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs]. (2014).
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. IEEE, Las Vegas, NV, USA (2016). https://doi.org/10.1109/CVPR.2016.90.
29. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs]. (2017).
30. Home - Keras Documentation, https://keras.io/, last accessed 2019/09/22.
31. TensorFlow, https://www.tensorflow.org/, last accessed 2019/09/22.

32. Saha, S.: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, last accessed 2019/05/23.
33. Enzyme Immunoassays (EIA) and Enzyme-Linked Immunosorbent Assays (ELISA), https://bio.libretexts.org/Bookshelves/Microbiology/Book%3A_Microbiology_(OpenStax)/20%3A_Laboratory_Analysis_of_the_Immune_Response/20.4%3A_Enzyme_Immunoassays_(EIA)_and_Enzyme-Linked_Immunosorbent_Assays_(ELISA), last accessed 2019/11/12.