Article
Image-Based Arabian Camel Breed Classification Using Transfer Learning on CNNs

Sultan Alfarhood 1*, Atheer Alrayeh 1, Mejdl Safran 1, Meshal Alfarhood 1 and Dunren Che 2

1 Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia; atheeralrayeh@gmail.com (A.A.); mejdl@ksu.edu.sa (M.S.); malf@ksu.edu.sa (M.A.)
2 School of Computing, Southern Illinois University, Carbondale, IL 62901, USA; dche@cs.siu.edu
* Correspondence: sultanf@ksu.edu.sa

Abstract: Image-based Arabian camel breed classification is an important task for various practical applications, such as breeding management, genetic improvement, conservation, and traceability. However, it is a challenging task due to the lack of standardized criteria and methods, the high similarity among breeds, and the limited availability of data and resources. In this paper, we propose an approach to tackle this challenge by using convolutional neural networks (CNNs) and transfer learning to classify images of six different Arabian camel breeds: Waddeh, Majaheem, Homor, Sofor, Shaele, and Shageh. To achieve this, we created, preprocessed, and annotated a novel dataset of 1073 camel images. We then pre-trained CNNs as feature extractors and fine-tuned them on our new dataset. We evaluated several popular CNN architectures with diverse characteristics such as InceptionV3, NASNetLarge, PNASNet-5-Large, MobileNetV3-Large, and EfficientNetV2 (small, medium, and large variants), and we found that NASNetLarge achieves the best test accuracy of 85.80% on our proposed dataset. Finally, we integrated the best-performing CNN architecture, NASNetLarge, into a mobile application for further validation and actual use in real-world scenarios.

Keywords: transfer learning; convolutional neural networks; image classification; camel; InceptionV3; NASNetLarge; PNASNet; MobileNetV3; EfficientNetV2

1. Introduction
Camels are remarkable animals for their ability to adapt to harsh environmental conditions. They can survive for long periods without drinking water thanks to their specialized physiology and behavior. They can also tolerate high temperatures and solar radiation as well as cold and windy climates. Therefore, camels are a vital resource for desert dwellers who have developed rich knowledge regarding the breeds and characteristics of camels. However, most people lack the expertise to recognize the different breeds of camels, especially Arabian camels, which may appear similar but have distinct features and colors. Camels are large, hump-backed and iso-toed ungulates belonging to the mammalian genus Camelus of the family Camelidae. There are two species of camels: the Arabian camel (Camelus dromedarius) and the Bactrian camel (Camelus bactrianus). This work focuses on the Arabian camel, which has several breeds that mostly vary in color. One of the most widely accepted classifications of Arabian camel breeds is based on the King AbdulAziz Camel Beauty Competition, also known as Mezayen, which divides the participating Arabian camels into six categories: Majaheem (black camels), Waddeh (white camels), Shageh (off-white camels with some redness), Sofor (yellow camels with black humps), Homor (red-brown camels), and Shaele (pure yellow camels) [1,2]. Figure 1 shows an example of each breed. In terms of color, certain categories of Arabian camel breeds, such as Majaheem and Waddeh, are easily distinguishable. However, other categories, including Shageh, Sofor, Homor, and Shaele, exhibit a similar appearance and coloration, which can lead to misidentification.
Figure 1. The six Arabian camel breeds are: (A) Waddeh, (B) Majaheem, (C) Homor, (D) Sofor, (E) Shaele, and (F) Shageh.

Image-based Arabian camel breed classification is a challenging problem that has not been adequately addressed in the literature. Unlike other animal breeds such as birds and dogs, which have been extensively studied using image classification techniques, Arabian camel breeds have distinctive and subtle features that require more sophisticated methods to recognize and distinguish them. In this paper, we propose a novel approach for image-based Arabian camel breed classification using transfer learning on convolutional neural networks (CNNs). Transfer learning is a powerful technique that enables the reuse of knowledge learned from one domain to another domain with different data distributions. CNNs are state-of-the-art deep learning models that can automatically learn hierarchical representations from raw image data and achieve high performance on various computer vision tasks, such as image classification, object detection, face recognition, and more [3]. Although there are several publicly available image-based datasets for animal breed classification, such as the Stanford Dogs dataset [4], Oxford IIIT-Pet Dataset [5], and Tsinghua Dogs Dataset [6], there is currently no dataset specifically designed for the classification of Arabian camel breeds. This presents an opportunity for researchers to create a new dataset to address this gap in the literature and advance the field of animal breed classification. In our work, we adapt pre-trained models originally trained on the ImageNet dataset to a new and smaller dataset of Arabian camel images. By doing so, we aim to learn features specific to camels and use them to classify the images into different camel breeds. This novel dataset of Arabian camel images is collected from various sources and annotated with their corresponding breeds by subject matter experts. We evaluate several pre-trained well-known CNN architectures with diverse characteristics on our dataset, including InceptionV3 [7], NASNetLarge [8], PNASNet-5-Large [9], MobileNetV3-Large [10], and EfficientNetV2 (small, medium, and large variants) [11].

Our proposed approach to image-based Arabian camel breed classification brings several benefits. Firstly, by using transfer learning, we can leverage the knowledge learned from pre-trained CNNs on another large datasets such as ImageNet to improve the classification accuracy of our model. This is particularly useful when dealing with a learning problem with limited training data, which is often the case for niche classification problems such as Arabian camel breed classification. Secondly, through further training (fine-tuning) of pre-trained CNNs on a domain-specific dataset, namely our Arabian camel dataset, our
approach is able to effectively capture the distinctive and subtle features of different Arabian camel breeds and improve our system’s classification accuracy. Finally, our approach is scalable and can be easily extended to classify other animal breeds or even other types of objects after fine-tuning of our model on a domain-specific niche dataset.

The main contributions of our work presented in this paper are as follows:

- We introduce a new image dataset for Arabian camel breed classification, consisting of 1073 images of 6 different breeds of Arabian camels collected from various sources. The breeds are: Majaheem, Homor, Shaele, Shageh, Sofor, and Waddeh. To the best of our knowledge, this is the first and only dataset of its kind;
- We propose a machine learning application framework for image-based camel breed classification based on transfer learning. We use pre-trained CNNs as feature extractors, fine-tune them on our new dataset, and retain only the best performing CNN in our final application system;
- We evaluate our model on our new dataset and compare it with several baselines and state-of-the-art methods. We show that our model achieves high accuracy and good generalization across different breeds and image conditions;
- We integrate our selected best-performing CNN into a mobile application for further validation and actual use in a real-world scenario.

The rest of this paper is structured as follows. Section 2 gives an overview of previous related works that utilize machine learning to solve animal breed classification. Section 3 provides the details of the proposed solution for Arabian camel breed classification. Section 4 presents the implementation details and our experimental results. Finally, Section 5 concludes the paper and points out the directions of our future work.

2. Literature Review

The classification of animal breeds is an important task in the field of animal research and many related applications. With the advancement of technology, machine learning techniques have been increasingly applied to this task, leading to promising results. In particular, the use of image-based classification and transfer learning on CNNs have shown great potential for accurately classifying and predicting animal breeds. The following sections provide an overview of previous research on machine learning techniques for animal classification and detection, with a focus on image-based classification and transfer learning on CNNs.

2.1. ML-Based Approaches for Animal Classification and Detection

Nowadays, CNN has become an immensely popular learning method used for image classification problems [12]. There have been several attempts to automatically classify animals based on input images into distinct categories using CNN. Alsaaedi and El Abbadi [13] proposed a novel model based on deep convolutional neural networks to detect and categorize two classes of vertebrate animals: mammals and reptiles. Their proposed algorithm was trained on 4000 images and later tested using 1200 images. They achieved an accuracy of 97.5% for their system’s prediction of target objects. Additionally, Trnovszky et al. [14] proposed a CNN model to classify input images into categories of fox, wolf, bear, hog, or deer and compared their proposed CNN model with well-known algorithms for image recognition, feature extraction, and image classification, including PCA, LDA, SVM, and LBPH. They obtained the best animal recognition accuracy of 98% using their proposed CNN. Furthermore, Zhou et al. [15] developed an algorithm to categorize images of dogs and cats by investigating two approaches to address their targeted problem. Their first approach is a traditional pattern recognition model by which they trained the classification model from some human-crafted features, like color, Dense-SIFT, and a combination of the features. Their second approach uses deep convolutional neural networks to learn the features of images by training neural networks and SVMs for classification. They have conducted different experiments to evaluate the performance of the two approaches on a test dataset. The second approach outperforms the first approach with an accuracy of
Moreover, Zeng [16] tried to classify similar animal images by applying a simple CNN. He focused on binary classification for snub-nosed monkeys and normal monkeys which are considered hard to distinguish manually in real-time. He used a database constructed by a Python crawler that consists of 1000 images for each class, split as two subsets of 800 and 200 images, used as the training set and the test set. His model achieved an accuracy of 96% after many attempts of optimization in terms of hyperparameters and model structures. In addition, Lin [17] trained a deep learning model to distinguish between images of cats and dogs. He first trained a VGG model from scratch and then used transfer learning to further improve the accuracy of the model. His objective is to study household animals’ demeanor and body language to find out if these animals are sick and offer necessary help to the sick animals in time. Through transfer learning the model’s accuracy is increased from 80% to over 95%. Also, Khandale and Ramdev [18] proposed a mammal classification system based on the Inception-v3 CNN architecture to classify mammals into lions, tigers, elephants, giraffes, and monkeys. They utilized transfer learning to fine-tune their model. Their model gained a classification accuracy of approximately 95% which is higher than other methods proposed for the same classification task.

2.2. ML-Based Approaches for Classifying Other Animal Properties

CNN is a popular tool for tackling automatic age estimation and gender classification problems based on images. Zamansky et al. [19] combined CNNs for feature extraction with classifiers based on the extracted features to predict the ages of dogs based on their images. They tried various combinations based on two CNN architectures, SqueezeNet and Inception V3, with six different famous classifiers, including kNN, SVM, Logistic Regression, Naive Bayes, and Random Forest. However, SqueezeNet with Naive Baye achieved 3.05 MAE which is the best score compared to their other combinations. Furthermore, the study proposed by Wang et al. [20] was the first one of its kind to solve the panda gender classification problem based on their face images by using various CNN architectures, including VGG11 with BN and different depth ResNet such as ResNet18, ResNet34, and ResNet50. Their model aims to learn features from the panda face image dataset and the results have proven that panda face does contain gender-specific information; male and female pandas carry distinct features on their faces. Based on their experiments, the normalized ResNet18 achieved the best accuracy among others with 77.2%.

2.3. ML-Based Approaches for Animal Breeds Classification

Different studies in the literature used various CNN models to predict the breeds of different animals (e.g., dogs, horses, buffalo, goats, and sheep) using image data. Varshney et al. [21] employed the transfer learning approach on two neural network architectures, VGG16 and Inception V3, to classify dogs into breeds based on their images. Their results show that Inception V3 works better than VGG16 on dog breed prediction with an accuracy of 85%. In addition, Raduly et al. [22] also tried to solve the dog classification problem based on a dog’s images using the transfer learning approach on two different CNN architectures, NASNet-A mobile and Inception Resnet V2. Their results show that NASNet-A generates 10% less accuracy than deep Inception Resnet V2, indicating that Inception Resnet V2 performed the best in their experiments with an accuracy of 90.69%. Additionally, Borwarginn et al. [23] proposed a method that also applies the transfer learning approach by retraining three existing pre-trained CNNs, MobileNetV2, InceptionV3, and NASNet, to recognize dog breeds based on their images. Their proposed method was evaluated based on the three CNNs with various augmentation settings and the results show that the NASNet model with augmented data achieves the best accuracy of 89.92%. Furthermore, Mulligan et al. [24] attempted to identify the breed of dogs from input images. They have used a variety of techniques to categorize the images in a dataset, which includes 120 distinct dog breeds, originally taken from Kaggle. They investigated a CNN model under different settings to build their model but ended up adopting Xception and a multilayered Perceptron architectures as their prediction approach. They tried to achieve
a higher accuracy rate and reduced error rate by modifying the parameters. They found that their best model had a log-loss of 9.5954 and a balanced accuracy of 54.80%. Moreover, Gupta et al. [25] attempted to tackle another classification task with 120 classes representing diverse dog breeds. They adopted various augmentation techniques such as flipping, zooming, moving, and rescaling. They applied deep CNN and transfer learning in their approach to improve classification accuracy. More specifically, they configured the ResNet-101 architecture as their feature extractor and applied transfer learning as the training method using a softmax activation function in the classifier. Their experiments showed that ResNet-101 achieved the highest accuracy of 71.63% among other models. In addition, Shi et al. [26] used convolutional neural networks to distinguish between 120 different dog breeds. They employed transfer learning because their data was not sufficiently large to prevent overfitting. The authors carried out dog breed identification using four separate techniques, each with a unique training model, i.e., ResNet18, VGG16, DenseNet161, and AlexNet. They also made some adjustments to the optimization methods based on their models to improve the identification accuracy. They found that the DenseNet model was the best and adopted it as their main model which achieved an accuracy of 85.14%. Another related work by Ayanzadeh and Vahidnia [27] used a pre-trained CNN model to extract the breed feature from a dog breed dataset. The authors compared the results of multiple pre-trained models, including DenseNet-121, ResNet50, DenseNet169, and GoogleNet which were all pre-trained on ImageNet. Then, they applied data augmentation and fine-tuned the models to improve the breed identification accuracy of the models. They discovered that the ResNet50 model can generate 89.66% test accuracy after fine-tuning, which is the best among all the models they tested with.

Atabay’s study [28] applies transfer learning on well-known deep CNN classifiers, including VGG architectures, InceptionV3, ResNet50, and Xception for horse breed recognition using horse images. The study indicated that the VGG architecture would be the best choice for training if given a limited training time and limited hardware facilities, while ResNet50 generates the highest accuracy of 95.90%; however, it takes a long time to converge during the training phase. De la Cal et al. [29] conducted a similar study by carrying out an exhaustive analysis of five pre-trained CNN models, including VGG16, VGG19, Resnet50, InceptionV3, and Xception to detect horse breeds using two horse images datasets. Furthermore, Fu et al. [30] proposed a novel method using transfer learning of pre-trained deep CNN architectures including MobilenetV2, Mobilenet, Xception, VGG16, and VGG19 to solve the same task based on a horse images dataset. Their results showed that transfer learning on the MobilenetV2 architecture achieved the best accuracy of 89.34%, which confirms the best approach to horse breed classification compared to other deep CNNs given a small dataset.

Pan et al. [31] presented a computer-vision-based recognition system to categorize buffalo breeds. Their suggested framework combines self-activated convolutional neural networks with self-transfer learning. Additionally, rich feature vectors are obtained by further transferring the feature maps that were generated from the CNN. To classify the feature vectors, they used various machine learning (ML) classifiers. Their suggested framework is tested on various buffalo breeds. They reached a maximum accuracy of 93% using SVM and 85% using other recent learning models including Fine-KNN, Medium-KNN, Coarse-KNN, LP-Boost, Total-Boost, and Bag-Ensemble.

For classifying sheep into different breeds based on their images, Salama et al. [32] tried to find the appropriate design for CNNs by experimenting with two methodologies. The first applies Bayesian optimization to set the parameters for a convolutional neural network automatically, whereas the second uses AlexNet, a pre-trained CNN model. Their hybrid approach of using a CNN with Bayesian optimization achieved an accuracy of 98%, while their second approach of using pre-trained AlexNet achieved an accuracy of 97.5%. Agrawal et al. [33] proposed an ensemble model using the ResNet50 and VGG16 CNN architectures to predict sheep breeds. Five cutting-edge transfer learning models, namely ResNet50, VGG16, VGG19, InceptionV3, and Xception, were compared to their proposed
ensemble model. Their ensemble model performed the best and achieves 97.32% accuracy according to their experiments. In addition, Kaushik et al. [34] used the Inception-v3 CNN architecture to recognize and classify goat images into six different breeds. As training such a large network end-to-end is very time-consuming, the authors instead used transfer learning to overcome this problem. The result showed that their modified Inception-v3 correctly classified the animals in 93.33% of the cases with more than 95% confidence.

2.4. Discussion

This paper focuses on image-based classification of Arabian camel breeds, which is a unique and challenging issue compared to the above reviewed related works. To the best of our knowledge, this is the first study that addresses the problem of image-based Arabian camel breed classification. In this work, a new image dataset of six different Arabian camel breeds is introduced and various pre-trained CNN architectures are fine-tuned on the new dataset to fit the specific classification problem. Furthermore, the proposed model is integrated into a mobile application to validate its performance in real-world scenarios. This poses an additional computational challenge due to the limited resources of mobile devices in terms of memory and processing power.

3. Proposed Approach

In this section, we present our proposed approach for image-based Arabian camel breed classification by describing the overall process of our approach in the first section. Then, we introduce the Arabian camel breed dataset that we used for our experiments and explain how we performed data preprocessing and dataset partitioning. Next, we describe the convolutional neural network models that we adopted for our classification task and explain how we used transfer learning to leverage the pre-trained weights of the best performing (per our evaluation) state-of-the-art model. Finally, we present the evaluation metrics that we used to measure the performance of our selected baseline models on the test data.

3.1. Overall Pipeline

Our overall approach to image-based Arabian camel breed classification is best illustrated as a pipeline framework as shown in Figure 2. The pipeline consists of several stages as follows: (1) Arabian camel dataset creation: we collect and annotate a diverse dataset of Arabian camel images with their corresponding breeds; (2) data pre-processing: we apply various techniques to enhance the quality and diversity of the image dataset, such as cropping, resizing, and normalization; (3) data partitioning: we split the dataset into three subsets for training, validation, and testing, ensuring that each subset has a balanced distribution of camel breeds; (4) CNN model training with transfer learning: we experiment with well-known CNN models pre-trained on the ImageNet dataset and fine-tuned on our Arabian camel dataset to learn camel-specific features and we classify camel images into six breeds; (5) CNN model evaluation: we evaluate the performance of the fine-tuned CNN models on the test subset using various metrics, such as accuracy, precision, recall, F1-score, and more; (6) CNN model integration: we integrate the best-performing, pre-trained, and fine-tuned CNN model into a web service that takes camel images from users using our mobile app and predicts the camels’ breeds in real time.

3.2. Arabian Camel Breed Dataset

To the best of our knowledge, there is no existing image-based dataset for Arabian camel breed classification that can be used to train our model. Therefore, we constructed a novel dataset for this task from scratch by collecting and annotating images of different breeds of Arabian camels from various sources. The images were labeled with six distinct Arabian camel breeds: Waddeh, Majaheem, Homor, Sofor, Shaele, and Shageh. We obtained 1073 images in total with approximately 180 images per breed. We also consulted three
experts in the field to review the dataset and remove any ambiguous or mislabeled images. The images were organized into six separate folders corresponding to the six breeds.

![Proposed process of the image-based Arabian camel breed classification.](image)

**Figure 2.** Proposed process of the image-based Arabian camel breed classification.

### 3.2.1. Data Preprocessing

The initial dataset required some preprocessing steps before being applied to model training. The images of the Arabian camel breeds were obtained from different sources of varying sizes which could affect the performance of the CNN models. Therefore, we resized and cropped the images to make them suitable for training. Moreover, we applied some preprocessing techniques to enhance the quality of the images. We first removed any duplicate images from the folders to avoid redundancy. Then, we cropped the images to eliminate irrelevant parts such as text or people that could interfere with the classification. Table 1 summarizes the details of the Arabian camel breed dataset and shows the balanced distribution of images across different classes after preprocessing.

**Table 1.** Details of images in the Arabian camel breed dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
<th>Total Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waddeh</td>
<td>125</td>
<td>27</td>
<td>27</td>
<td>179</td>
</tr>
<tr>
<td>Majaheem</td>
<td>125</td>
<td>28</td>
<td>27</td>
<td>180</td>
</tr>
<tr>
<td>Homor</td>
<td>125</td>
<td>28</td>
<td>27</td>
<td>180</td>
</tr>
<tr>
<td>Sofor</td>
<td>121</td>
<td>26</td>
<td>27</td>
<td>174</td>
</tr>
<tr>
<td>Shaele</td>
<td>125</td>
<td>28</td>
<td>27</td>
<td>180</td>
</tr>
<tr>
<td>Shageh</td>
<td>125</td>
<td>28</td>
<td>27</td>
<td>180</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>746</strong></td>
<td><strong>165</strong></td>
<td><strong>162</strong></td>
<td><strong>1073</strong></td>
</tr>
</tbody>
</table>

### 3.2.2. Dataset Partitioning

We split the dataset of the Arabian camel breeds into three subsets: training, testing and validation. The dataset consists of 1073 images of Arabian camels which we allocated to three subsets as follows: 70% for training (746 images), 15% for validation (165 images), and 15% for testing (162 images). We used stratified random sampling to ensure a balanced distribution of labels across all the subsets. Table 1 depicts the label distribution for each subset.

### 3.3. Convolutional Neural Network Model

We used transfer learning on one of the state-of-the-art convolutional neural network architectures. Transfer learning is a technique that allows us to use pre-trained CNNs
trained on a large and general dataset, such as ImageNet, and fine-tune them on our specific dataset such as our Arabian camel dataset. This can save us time and computational resources and improve the performance of our model. We evaluated several widely used CNN architectures, namely InceptionV3, NASNet-Large, PNASNet-5-Large, MobileNetV3-Large, and EfficientNetV2 (small, medium, and large variants). The selection of these CNN architectures was based on several factors. Firstly, we considered the design principles and characteristics of each architecture, such as the depth, width, and skip connections, to ensure that we have a diverse set of models to evaluate. Secondly, we chose architectures that have shown high performance on the ImageNet dataset as this indicates their ability to learn useful features from image data. Thirdly, we also took into account the computational resources required to fine-tune them on our dataset. Finally, all of the selected CNN architectures are publicly available on the TensorFlow Hub [35]. Table 2 summarizes the main characteristics of these CNN architectures. Based on our evaluation we then selected the best performing CNN architecture as the base model for our prediction model to identify Arabian camel breeds.

Table 2. CNN architectures’ details.

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3</td>
<td>2015</td>
<td>23.8 million</td>
</tr>
<tr>
<td>NASNet-Large</td>
<td>2018</td>
<td>88.9 million</td>
</tr>
<tr>
<td>PNASNet-5-Large</td>
<td>2018</td>
<td>86.1 million</td>
</tr>
<tr>
<td>MobileNetV3-Large</td>
<td>2019</td>
<td>5.4 million</td>
</tr>
<tr>
<td>EfficientNetV2-S</td>
<td>2021</td>
<td>22 million</td>
</tr>
<tr>
<td>EfficientNetV2-M</td>
<td>2021</td>
<td>54 million</td>
</tr>
<tr>
<td>EfficientNetV2-L</td>
<td>2021</td>
<td>121 million</td>
</tr>
</tbody>
</table>

3.4. Evaluation Metrics

We focused on the performance aspect of the machine learning models to evaluate them and decide our final base model. Performance refers to how well the models can classify the input images of Arabian camels based on their breeds, which is the goal of this work. We used a set of evaluation criteria to measure and compare the performance of various selected models. The evaluation criteria are based on comparing the predictions made by the models on a given dataset of Arabian camel images with true class labels verified by domain experts. The evaluation metrics that we used are as follows:

Accuracy: This metric measures how often a classifier predicts correctly. It can be calculated mathematically using Equation (1).

\[
\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number of Predictions}} \tag{1}
\]

Precision: This metric measures how many of the positive predictions made by a classifier are actually positive. To compute precision, we use Equation (2).

\[
\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}
\]

Recall: This metric measures how many of the actual positive instances are predicted correctly by a classifier. To calculate the recall score, we use Equation (3).

\[
\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}
\]
F1-score: This metric combines precision and recall into a single score. The F1-score is computed using Equation (4).

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}
\]

AUC: AUC stands for area under the ROC curve. The ROC curve shows the performance of a binary classifier as a function of its cut-off threshold. The ROC curve plots the true positive rate (Recall) against the false positive rate (1 − Precision) for different values of the threshold. In our multi-class classification problem, we applied the one-vs-all method to adapt AUC. In this approach, each class is treated as the positive class and all other classes are treated as negative. The AUC is then calculated for each class and averaged to obtain an overall value.

3.5. CamelBreeds App

To utilize the result of our study in real-world scenarios helping users identify Arabian camel breeds, we incorporated our proposed model within an Android mobile application called the CamelBreeds App (as illustrated in Figure 3a). The app allows individuals to take and upload a picture of an Arabian camel using their mobile phones. The app can instantly process the image and make prediction of the camel breed (as depicted in Figure 3b).

Figure 3. CamelBreeds App Screenshots.

4. Experimentation

In this section, we present the results of our experiments on image-based Arabian camel breed classification using transfer learning on CNNs. We first describe the experimental setup, including the hardware and software specifications, the hyperparameters, and the training procedure. Then, we report the results of the experiments that we conducted to evaluate and compare selected CNN models.

4.1. Experimental Setup

To implement the convolutional neural networks, we used the very well-known framework of Google, TensorFlow [36], and the Python language. TensorFlow is a Google open-source framework to run machine learning, deep learning, and other statistical and pre-
dictive analytics workloads. Python is an easy-to-learn programming language that has an elegant and dynamic typing and interpreted nature that makes it an ideal language for scripting and rapid application development in many areas on most platforms. The environment that was used to run the experiments is Google Colab [37]. We used the TensorFlow 2.9.0 library to establish artificial neural networks and the NVIDIA T4 16GB GPU to speed up training. All pre-trained models utilized in this experiment were retrieved from the TensorFlow Hub [35].

4.2. Pre-Trained CNNs and Hyperparameter Settings

The CNN architectures utilized in this project were pre-trained using the ImageNet dataset which contains millions of images from a thousand classes. There are images of Arabian camels and other similar animals in the ImageNet dataset; however, it does not include breeds of the Arabian camel.

The CNN framework we propose in this work for Arabian camel breed prediction consists of four layers: an input layer, a hub layer, a dropout layer, and a dense layer. The input layer takes images as an input. The hub layer is a pre-trained model retrieved from the TensorFlow Hub as the base model in our application framework. The dropout layer randomly drops out 20% of the units to prevent overfitting. Dropout is actually a method for regularizing neural networks and involves randomly changing some features to 0 during the forward pass [38]. The dense layer outputs a vector of a length equal to the number of classes, with a L2 regularization of 0.0001.

The experimented CNNs’ hyper-parameters are configured as follows. A stochastic gradient descent (SGD) with a learning rate of 0.005 and a momentum 0.9 is used in all experiments as the optimization method because it produced the best results when compared to other optimizers we experimented with such as Adam, Adamax, Adadelta, Nadam, Adagrad, RMSprop, and Ftrl. Moreover, the batch size is set to 16 but for large models we set it to 8; the learnable parameters are initialized using Keras’ default values. The loss function is the categorical crossentropy with logits and a label smoothing of 0.1. The metric used to evaluate the model is the accuracy.

All models (each incorporating a different base model) are trained for 200 epochs. However, we used an early stopping callback function that monitors the validation loss and stops the training if it does not improve for 10 consecutive epochs.

4.3. Experimental Results

We evaluated and compared the performance of seven different CNN models that served as alternative base models in our project and were pre-trained on ImageNet and fine-tuned on our Arabian camel breed dataset. The seven base models are InceptionV3, NASNet-Large, PNASNet-5-Large, MobileNetV3-Large, and EfficientNetV2 (small, medium, and large variants). We evaluated these alternative base models in terms of accuracy, F1 score, and AUC on the test dataset, as well as the training and inference time.

Table 3 shows the evaluation results of the models on the test dataset. We can see that NASNet-Large achieved the highest accuracy (85.80%) and F1 score (86%) on the test dataset, followed by EfficientNetV2-L with an accuracy of 83.95% and an F1 score of 84%. MobileNetV3-Large and InceptionV3 performed the worst on the test dataset, with accuracies of 77.78% and 79.63% and F1 scores of 77% and 79%, respectively. This experimental result suggests that NASNet-Large and EfficientNetV2-L are more suitable for camel breed classification than the other investigated models.

To further analyze the performance of the models, we plotted the training accuracy and loss curves for all models on both training and validation datasets. Figure 4 shows the accuracy and loss progress for each model over the epochs. These results are consistent with the confusion metrics shown in Figure 5. The results of the confusion matrices indicate that all the models are able to accurately distinguish between the Majaheem and Waddeh classes, which can be attributed to their distinctive colors. However, the majority of the
models exhibit a higher error rate when distinguishing between the Shageh, Sofor, Homor, and Shaele classes, likely due to their similarities in color.

**Table 3.** Comparison of performance metrics of the CNN models on the test dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3</td>
<td>79.63</td>
<td>79</td>
<td>0.95</td>
</tr>
<tr>
<td>NASNet-Large</td>
<td>85.80</td>
<td>86</td>
<td>0.98</td>
</tr>
<tr>
<td>PNASNet-5-Large</td>
<td>81.48</td>
<td>81</td>
<td>0.95</td>
</tr>
<tr>
<td>MobileNetV3-Large</td>
<td>77.78</td>
<td>77</td>
<td>0.94</td>
</tr>
<tr>
<td>EfficientNetV2-S</td>
<td>82.10</td>
<td>81</td>
<td>0.95</td>
</tr>
<tr>
<td>EfficientNetV2-M</td>
<td>83.34</td>
<td>83</td>
<td>0.96</td>
</tr>
<tr>
<td>EfficientNetV2-L</td>
<td>83.95</td>
<td>84</td>
<td>0.96</td>
</tr>
</tbody>
</table>

**Figure 4.** The training accuracy and loss curves for all models on both training and validation datasets.

NASNet-Large’s superior performance may be attributed to its larger number of parameters and depth compared to the other models. This allows it to learn more complex and discriminative features from the images. Additionally, NASNet-Large was generated by a neural architecture search algorithm and optimized for both accuracy and computational efficiency, making it more suitable for image classification tasks. On the other hand, InceptionV3 and MobileNetV3-Large’s inferior performance may be due to their smaller number of parameters and depths compared to the other models, which limits their capacity to learn from the images. Furthermore, InceptionV3 and MobileNetV3-Large were designed for different purposes than image classification, such as reducing computational cost and latency, which may compromise their accuracy.

**Table 4** shows the comparison of the models in terms of the number of epochs, training time, and inference time. We can see that EfficientNetV2-L took the longest time to train followed by NASNet-Large. MobileNetV3-Large has the shortest training time followed by InceptionV3. In terms of the inference time, MobileNetV3-Large is the fastest, followed by PNASNet-5-Large. InceptionV3 is the slowest followed by EfficientNetV2-L.
Figure 5. Confusion Matrix of All Models Using the Test Dataset.
Table 4. Comparison of the epochs, training time, and inference time of the CNN models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Epochs</th>
<th>Training Time (s)</th>
<th>Inference Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3</td>
<td>19</td>
<td>666.01</td>
<td>1</td>
</tr>
<tr>
<td>NASNet-Large</td>
<td>46</td>
<td>4805.74</td>
<td>0.97</td>
</tr>
<tr>
<td>PNASNet-5-Large</td>
<td>42</td>
<td>4180.67</td>
<td>0.69</td>
</tr>
<tr>
<td>MobileNetV3-Large</td>
<td>51</td>
<td>407.75</td>
<td>0.61</td>
</tr>
<tr>
<td>EfficientNetV2-S</td>
<td>47</td>
<td>1604.88</td>
<td>0.72</td>
</tr>
<tr>
<td>EfficientNetV2-M</td>
<td>41</td>
<td>3636.33</td>
<td>0.93</td>
</tr>
<tr>
<td>EfficientNetV2-L</td>
<td>34</td>
<td>5762.35</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Among the six models, NASNet-Large has the best trade-off between accuracy and inference time. However, it also requires more computational resources and time to train than the other models. Therefore, depending on the application scenario and resource constraints, other models such as EfficientNetV2-M or MobileNetV3-Large might be preferred.

The ROC curve of our best model, NASNet-Large, shown in Figure 6, demonstrates its ability to achieve a good balance between sensitivity and specificity for the different classes. This indicates that the model is able to accurately classify the different breeds of Arabian camels with a low rate of false positives and false negatives. The area under the ROC curve (AUC) is a commonly used measure of a model’s performance, with a value of 1 indicating perfect classification and a value of 0.5 indicating random classification. Our best model achieved an AUC value close to 1, indicating its high accuracy in classifying the different breeds of Arabian camels.

![ROC Curve](image)

Figure 6. ROC Curve of the NASNetLarge model.
4.4. Data Augmentation Discussion

To increase the diversity and size of the dataset and prevent overfitting, we experimented with data augmentation techniques that modify the data to generate additional images or create images that can improve the robustness of the training model without collecting new data. However, we did not observe any significant improvement in the performance of the convolutional neural network model after applying data augmentation. The data augmentation techniques that we experimented with include random rotation, random flipping, rescaling, and random contrast adjustment.

- Random rotation: This technique randomly rotates the images by a certain angle within a specified range. We expected this to introduce some variations in the orientation of the images and make the model more invariant to rotation, but it did not have any noticeable effect on the accuracy or loss;
- Random flipping: This technique randomly flips the images horizontally or vertically (only horizontally in our experiment). We expected this to introduce some variations in the symmetry of the images and make the model more invariant to flipping but it did not have any noticeable effect on the accuracy or loss;
- Rescaling: This technique randomly rescales the images by a certain factor within a specified range. We expected this to introduce some variations in the size of the images and make the model more invariant to scaling but it did not have any noticeable effect on the accuracy or loss;
- Random contrast adjustment: This technique randomly adjusts the contrast of the images by a certain factor within a specified range. We expected this to introduce some variations in the brightness of the images and make the model more invariant to contrast but it did not have any noticeable effect on the accuracy or loss.

The four aspects of data augmentation we carried out did not show a noticeable effect on improving the performance of our system as expected. One possible explanation is that these augmentation techniques did not sufficiently alternate the distinctive features of camel breeds in the augmented images. In other words, the performed augmentation operations very well reserved the distinctive features of camel breeds in the augmented images. This seems to indicate that we need to apply more aggressive augmentation techniques in order to render sufficient alternation to the distinctive features of camel breeds in the augmented training dataset, with the hope of being able to noticeably improve our system’s prediction accuracy and robustness.

5. Conclusions and Future Work

Our work reported in this article aims to provide a starting point in Arabian camel breed image classification. We started by gathering and annotating a dataset of Arabian camel breed images which consists of 1073 images of 6 Arabian camel breeds: Waddeh, Majaheem, Homor, Sofor, Shaele, and Shageh. We proposed a CNN model for Arabian camel breed classification and prediction based on extensive empirical studies on popular pre-trained CNN models and selected the best-performing model, i.e., NASNet-Large, as the basis in our model. We pre-trained and experimented with seven popular CNN architectures including InceptionV3, NASNet-Large, PNASNet-5-Large, MobileNetV3-Large, and EfficientNetV2 (with three variants: small, medium, and large) on our Arabian camel breed dataset using transfer learning to achieve the best classification accuracy; NASNet-Large turned out to perform the best on our curated Arabian camel dataset with an accuracy of 85.80% and an F1 score of 86%. Furthermore, we integrated our proposed model into a mobile application for ease of access, further validation, and practical application in real-world scenarios. As future work, we plan to enhance our curated Arabian camel dataset in terms of both scale (number of images) and quality as well as expand the scope of our research by incorporating object detection and seeking better robustness and higher accuracy of camel classification. Additionally, we plan to experiment with a variety of more aggressive data augmentation techniques and new state-of-the-art CNN architectures to further improve the performance of our model. Based on the encouraging result we have
achieved with Arabian camel classification, we plan also to extend the capabilities of our model and system to classify Arabian camels along other properties, such as age, gender, disease, and degree of beauty.

Author Contributions: Conceptualization, S.A.; Methodology, S.A.; Data curation, S.A. and A.A.; Software, S.A.; Validation, S.A., M.S., M.A. and D.C.; Formal analysis, S.A.; Investigation, S.A., M.S. and M.A.; Resources, S.A.; Writing—original draft, S.A. and A.A.; Writing—review and editing, S.A., A.A., M.S. and M.A. and D.C.; Visualization, S.A.; Supervision, S.A.; Project administration, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deputyship for Research and Innovation “Ministry of Education” in Saudi Arabia (IFKSUOR3-031-1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research and Innovation “Ministry of Education” in Saudi Arabia for funding this research (IFKSUOR3–031-1).

Conflicts of Interest: The authors declare no conflict of interest.

References


30. Fu, Y.; Huang, X.; Li, Y. Horse Breed Classification Based on Transfer Learning. In Proceedings of the 4th International Conference on Advances in Image Processing, Chengdu, China, 13–15 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; Volume PartF169255, pp. 42–47. [CrossRef]

31. Pan, Y.; Jin, H.; Gao, J.; Rauf, H.T. Identification of Buffalo Breeds Using Self-Activated-Based Improved Convolutional Neural Networks. Agriculture 2022, 12, 1386. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.