

Article

DISubNet: Depthwise Separable Inception Subnetwork for Pig Treatment Classification Using Thermal Data

Savina Jassica Colaco ¹, Jung Hwan Kim ¹, Alwin Poulose ², Suresh Neethirajan ³
and Dong Seog Han ^{1,*}

¹ School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; savinacolaco@knu.ac.kr (S.J.C.); jkim267@knu.ac.kr (J.H.K.)

² School of Data Science, Indian Institute of Science Education and Research (IISER), Thiruvananthapuram 695551, India; alwinpoulosepalatty@iisertvm.ac.in

³ Farmworx Wageningen Institute, 6706 JS Wageningen, The Netherlands; sneethir@gmail.com

* Correspondence: dshan@knu.ac.kr; Tel.: +82-53-950-6609

Simple Summary: Thermal imaging is gaining popularity in poultry, swine, and dairy animal husbandry for detecting disease and distress. In this study, we present a depthwise separable inception subnetwork (DISubNet) for classifying pig treatments, offering two versions: DISubNetV1 and DISubNetV2. These lightweight models are compared to other deep learning models used for image classification. A forward-looking infrared (FLIR) camera captures thermal data for model training. Experimental results show the proposed models outperform others in classifying pig treatments using thermal images, achieving 99.96–99.98% accuracy with fewer parameters, potentially improving animal welfare and promoting sustainable production.

Abstract: Thermal imaging is increasingly used in poultry, swine, and dairy animal husbandry to detect disease and distress. In intensive pig production systems, early detection of health and welfare issues is crucial for timely intervention. Using thermal imaging for pig treatment classification can improve animal welfare and promote sustainable pig production. In this paper, we present a depthwise separable inception subnetwork (DISubNet), a lightweight model for classifying four pig treatments. Based on the modified model architecture, we propose two DISubNet versions: DISubNetV1 and DISubNetV2. Our proposed models are compared to other deep learning models commonly employed for image classification. The thermal dataset captured by a forward-looking infrared (FLIR) camera is used to train these models. The experimental results demonstrate that the proposed models for thermal images of various pig treatments outperform other models. In addition, both proposed models achieve approximately 99.96–99.98% classification accuracy with fewer parameters.

Keywords: animal welfare; depthwise separable layer; image classification; inception; thermal data



Citation: Colaco, S.J.; Kim, J.H.; Poulose, A.; Neethirajan, S.; Han, D.S. DISubNet: Depthwise Separable Inception Subnetwork for Pig Treatment Classification Using Thermal Data. *Animals* **2023**, *13*, 1184. <https://doi.org/10.3390/ani13071184>

Academic Editors: Irenilza de Alencar Nääs, Danilo Florentino Pereira and Saman Abdanan Mehdizadeh

Received: 26 January 2023

Revised: 19 March 2023

Accepted: 23 March 2023

Published: 28 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past few years, the number of applications for image classification has significantly increased. The goal of image classification is to determine the class to which a target object belongs. Classification is required whenever an object is assigned to a specific group or class based on the characteristics associated with that object. Image classification has many applications, including medical image analysis, human and animal face recognition, and monitoring and classifying animal behaviour [1]. It can be difficult to distinguish an object in an image if it is obscured by background clutter, noise, poor image quality, or other factors. Furthermore, the visible spectrum has limitations, such as lighting conditions and shadows, that could be overcome by thermal imaging. Thermal imaging is a non-destructive testing method that can be utilized to determine the surface temperature of objects. Thermal imaging is increasingly utilized in animal welfare to increase farm

production efficiency. Calves [2], poultry [3], and pig production [4] have been evaluated for animal welfare using thermal imaging. In addition, it is used to identify the temperature increase in pigs to predict their health [5].

In computer vision, animal classification using thermal images has been a crucial field of study. Continuous automatic systems for animal welfare typically provide information by collecting raw data and identifying key features through deep learning techniques. Farmers were better able to understand specific animal needs, such as welfare [6–8], and reproductive efficiency [9,10], with the aid of this method. The problem with automatic systems is that they use all nearby natural objects to represent animals in a scene, rather than just the animals themselves. In addition, animals can be viewed from various perspectives, scales, and shapes, as well as under different lighting conditions. However, this could be resolved using thermal images for animal classification. Thermal images capture heat emitted by animals, and these data can be used to identify patterns and detect abnormalities that are not visible to the naked eye. The use of thermal imaging in livestock applications has the potential to improve animal welfare, increase productivity, and reduce the environmental impact of livestock production. Ongoing research and development in this field will likely result in even more advanced applications in the near future. Thermal imaging can be used in livestock to detect indications of illness or injury. Changes in body temperature, for example, can suggest the existence of a fever, which is a typical sign of many disorders [11]. Thermal imaging can be used to detect estrus, which indicate when a female animal is in heat [12]. Furthermore, this information can be used to improve breeding programs and increase reproductive efficiency. Stress detection in livestock using elevated body temperature or changes in respiratory patterns for animal welfare [13]. Thermal imaging can be used to monitor individual animal growth as well as environmental factors such as temperature and humidity in livestock facilities [14]. Since there are many different animal classes, each with a complex intra-class variability and inter-class similarity, methods for classifying human faces have been developed with high accuracy, but those methods are incredibly inaccurate for classifying animal faces [6]. With each approach having advantages and disadvantages, researchers have tried various approaches to address these issues.

Convolutional neural network (CNN)-based classification techniques have, however, drawn a lot of attention in recent years. Deep learning methods involve representation learning and have multiple levels of representation [1]. Each of the modules that make up these algorithms transforms the representation at one level into a representation at a higher, more fundamental level while still being relatively simple but non-linear. As a result, a combination of these transformations can be used to learn quite complex functions. The higher-level representation enlarges aspects of data input that are important for distinguishing and suppressing irrelevant changes in animal classification tasks. The benefits of deep learning techniques have been successfully demonstrated in numerous applications where the input values are characterized by high dimensionality, enormous quantities, and highly structured data [15]. Deep learning techniques have a good performance and are, therefore, widely used in animal classification. Additionally, they have been widely used with thermal images [16]. Deep learning tools are incredibly helpful in image classification because the structure of the image is made up of millions of pixels that can be aligned into distinct objects [17]. The development of deep learning models has practical implications for pig farm management, allowing farmers to make data-driven decisions that improve pig health, welfare, and productivity. The advancement of neural models has greatly enhanced our ability to predict and manage various aspects of pig farming. Deep learning can help farmers optimize feeding programs and predict growth rates by analyzing large datasets of pig growth and feed intake [18]. Deep learning techniques can help predict disease outbreaks early by finding patterns in pig behavior and health data. These data can be utilized to develop early warning systems and guide disease management strategies [19]. They may be used to predict temperature and humidity levels in pig barns, which can assist farmers in maintaining optimal environmental conditions for

pig growth and health [20]. Deep learning models can enable farmers to identify breeding pairs that are likely to generate high-quality offspring with desirable traits by examining large datasets of genetic and phenotypic data [21].

To achieve greater accuracy, the general trend has been to create deeper and more complex networks [22]. These improvements in accuracy are necessarily making networks less effective in terms of size and speed. The recognition tasks in many real-world applications, including robotics, self-driving cars, and augmented reality, must be completed promptly on a platform with constrained computational resources [23]. To solve this issue, scaling CNN can improve accuracy while keeping the model lightweight and efficient. We propose a lightweight model that employs depthwise convolution layers and inception modules to reduce computational load while increasing accuracy with fewer parameters. We use thermal images instead of standard RGB images to overcome varying lighting and background conditions.

The main contributions of the paper are as follows:

1. We propose a depthwise separable inception subnetwork (DISubNet), a lightweight model for pig treatment classifications that consist of depthwise separable layers and an inception module.
2. We propose two versions of DISubNet: DISubNetV1 and DISubNetV2. The models are modified based on the concatenation of depthwise layers and inception modules.
3. Experiments are carried out on the pig image thermal dataset collected from the FLIR camera. The collected dataset consists of four pig treatment categories, such as isolation after feeding (IAF), isolation before feeding (IBF), paired after feeding (PAF), and paired before feeding (PBF).
4. Detailed experiments are conducted on both versions of DISubNet models with other image classification models using various evaluation metrics.

The rest of the paper is organized as follows: Section 2 provides the related works on image classification. The proposed models are explained in detail in Section 3. Section 4 provides details about the experiment. Section 5 contains the results of the experiments and their discussion. Finally, we conclude in Section 6.

2. Related Work

2.1. Image Classification Methods

Deep learning methods are commonly used in image classification tasks. The image classification process begins with the input image and ends with a classified result based on the class. The same principle applies to animal classification. The CNN-based animal classification system can be divided into three phases: pre-processing, feature learning, and classification. Firstly, to maximize the impact of factors that influence the animal classification algorithm, the input image undergoes a rescaling and image augmentation process in the pre-processing stage [24]. Second, in the feature learning step, the convolution algorithm is used to calculate the features of the input image. Finally, in the classification step, a predictive model is constructed using the features from the training data [25]. These predictive models estimate their class labels by comparing learned features from training data with test data or validation data [26]. The output classes are specific, and the user can discover the precise name of the class based on the prediction ratio. Animal image classification has previously been carried out using a variety of conventional classifiers, including support vector machine (SVM) [27,28] random forest (RF) [29,30], and decision tree (DT) [31–33]. In various settings, the use of an ensemble has grown in popularity. An ensemble is a supervised learning strategy that uses multiple models to boost the performance of a single model [7]. Recent research has mainly used deep learning techniques due to the promising results it has demonstrated in challenging computer vision tasks. In their work on animal species identification, Villa et al. [34] used the AlexNet [35], VGGNet [36], GoogLeNet [37], and ResNets [23] to analyze images of animals taken with a digital camera and an infrared sensor. The wildlife detector [38] was provided as a CNN model that trains a multi-class classifier while also learning a binary classification with two classes:

animal and non-animal. There are a few popular methods to divide and categorize animals in camera-trap images [39]. Animal recognition methods such as robust layer principal component analysis for segmentation, CNN for feature extraction, the least absolute shrinkage and selection operator (LASSO) for characteristics, and the SVM for classification of mammalian genera have been used in the Colombian forest [40]. As a classification model, ResNet50, ResNet101, ResNet152, GoogLeNet, and MixtureNet, which are all frequently used CNN models, were utilized [40]. CNNs have great potential in agriculture and livestock contexts for improving animal health and welfare, as well as for increasing efficiency and productivity on farms. As machine learning and computer vision technologies continue to advance, we can expect to see more innovative applications of CNN in the agricultural industry. CNNs can be trained to recognize individual animals, such as pigs or cows, based on their facial features or body markings [41]. This can be useful for tracking animal health and growth over time. CNNs can also be used to analyze animal behavior, such as monitoring pig or cow facial expressions to detect signs of pain or distress [42]. Tools such as ChickTrack use CNNs to track chicken activity levels, which can help farmers to monitor the health and welfare of their birds [43]. CNNs can help to automatically record and manage animals using different sensor technologies [44].

2.2. Model Design and Efficiency

For the past few years, researchers have been working on fine-tuning deep neural architectures to achieve the best possible balance between accuracy and performance. Small and effective neural networks are becoming increasingly popular in animal welfare [45,46]. Both compressing pre-trained networks and training small networks directly fall under the broad categories of many different approaches. There have been significant advancements over early designs such as AlexNet, VGGNet, GoogLeNet, and ResNet thanks to both manual architecture search and training algorithm improvements. In recent years, there has been significant progress in algorithmic architecture exploration, including hyperparameter optimization [47] network pruning [48] and connectivity learning [49]. As seen in ShuffleNet [50] or the addition of sparsity, much work has also gone into changing the connectivity structure of the internal convolutional blocks. Another advantage of deep learning is creating distributed representations that generalise newly learned characteristics and those observed during training. As a result, each of these representations can help model similar representations in other domains [49]. However, it is important to note that deep learning models are frequently complex models that involve the use of a large number of computational resources. Therefore, the goal of this paper is to design the model structure for the classification of pig treatments using thermal images with a focus on the need for smaller and more effective models.

3. Materials and Methods

In this section, we describe the various models used in the experiments, including LeNet5 [51], AlexNet, VGGNet, Xception [52], CNN-LeakyReLU [53], CNN-inception, and the proposed DISubNet model. These models are compared for the classification of the pig treatments.

3.1. Image Classification Models

One of the first pre-trained models is LeNet5, which recognises handwritten and machine-printed characters. The main reason that the model is popular is due to its straightforward structure. It is an image classification multi-layer convolution neural network which is made up of five layers that have learnable parameters. This network comprises three sets of convolutional layers, followed by a combination of average pooling layers and two fully connected hidden layers [51]. The images are classified using a softmax classifier. AlexNet won the Imagenet large-scale visual recognition challenge in 2012. The network depth in this model was increased when compared to the LeNet5 network. It has eight layers with learnable parameters. The model has five layers, the first of which

is a max-pooling combination, followed by three fully connected layers [35]. The layers use rectified linear unit activation (ReLU) as their activation function, which speeds up the training process. Dropout layers are also used in the model to avoid overfitting. The final layer employs softmax as its activation function. So, as we progress deeper into the architecture, the number of filters grows. As a result, it extracts more features as we progress deeper into the architecture. Furthermore, the filter size is decreasing, indicating that the initial filter was larger and that as we progress, the filter size is decreasing, resulting in a decrease in the feature map shape. The University of Oxford's visual geometry group (VGGNet) [36] created a deep convolutional neural network, which is widely used in computer vision fields. It comprises VGG-16 or VGG-19, which refer to the 16 and 19 convolutional layers, respectively. Xception employs depthwise separable convolutions [52]. It was developed by researchers at Google. They interpreted inception modules in CNN as an intermediate step between conventional convolution and the depthwise separable convolution in which a depthwise convolution is followed by a pointwise convolution.

3.2. Modified CNN Models

The CNN model with LeakyReLU [53] is a straightforward sequential model consisting of several convolutional layers and a batch normalization layer. Following the convolutional layers is LeakyReLU, which is based on ReLU but has a small slope for negative values rather than a flat slope. To reduce the spatial dimension of the feature map, max pooling is applied after each even convolution layer. The convolution layer has a filter size of 3×3 and a pooling size of 2×2 across all layers. Figure 1 depicts the CNN-LeakyReLU model structure.

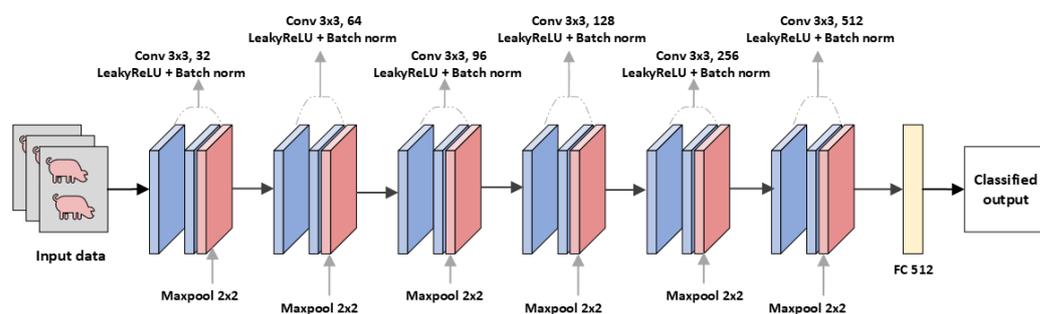


Figure 1. CNN-LeakyReLU: Convolution neural network with LeakyReLU and batch normalization.

Similar to the CNN-leakyReLU model structure, the model consists of convolutional layers and batch normalization layers. The max pooling is followed after every even convolution layer. Convolutional layers are made up of 3×3 filters in each layer. After every two convolution layers, max pooling with a 2×2 filter is applied to reduce the spatial dimension of the feature map. Figure 2 shows a representation of CNN-inception. To extract features, the model is further modified with a tunable inception module [37] consisting of filters such as 1×1 , 3×3 , and dilation filters. Dilated filters increase the area covered by the input image without pooling. The goal is to extract more information from each convolution operation's output. The different feature extraction from filters aids in focusing on different parts of images to detect complex patterns. In addition, the inception module includes a skip connection for identity mapping. The class scores will be processed by the fully connected layer, resulting in a volume in size, where each of the four numbers corresponds to a class score. The filters used in the inception module are more specifically shown in Figure 3.

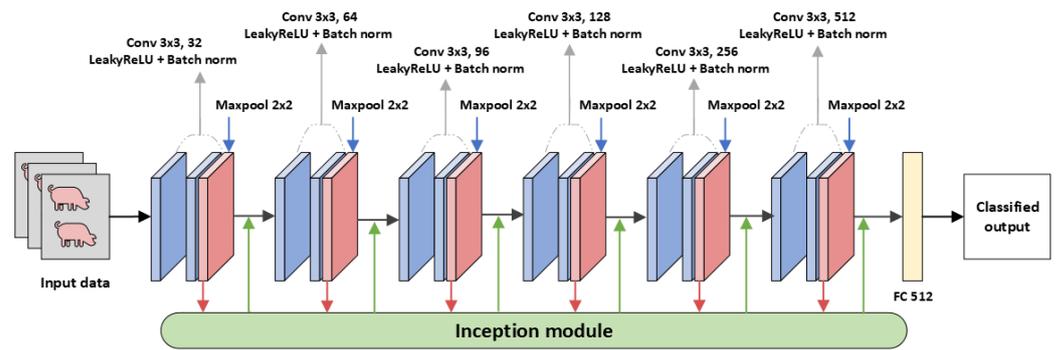


Figure 2. CNN-Inception: Convolution neural network with Inception module.

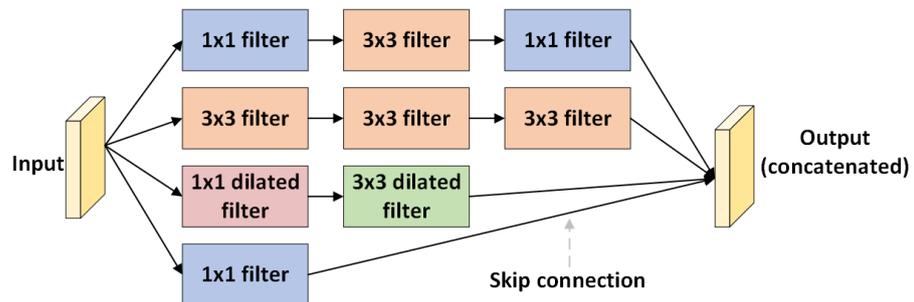


Figure 3. Inception module with different filters for extraction.

3.3. Proposed Model for Pig Treatment Classification

In comparison to large convolutional neural networks such as LeNet5, AlexNet, and VGGNet, DISubNet aims to make all of these networks smaller with fewer parameters while maintaining the same level of accuracy or even improving model generalization using fewer parameters. Larger networks are more prone to overfitting and raise the computation complexity. CNNs can also benefit from the extraction of features at different scales. Therefore, we propose DISubNet comprising of two subnetworks with alternating depthwise separable convolution layers [54] and an inception module. Additionally, we propose two DISubNet versions, DISubNetV1 and DISubNetV2. Figures 4 and 5 provide detailed information about the DISubNetV1 and DISubNetV2 models.

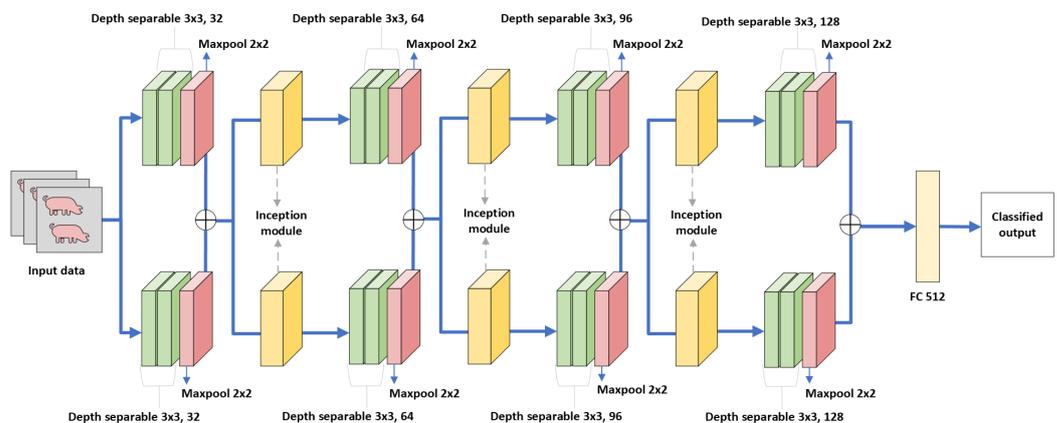


Figure 4. DISubNetV1: Depthwise separable convolution with inception module subnetwork. The output from depthwise layers is concatenated in this model and fed into the inception module for further extraction.

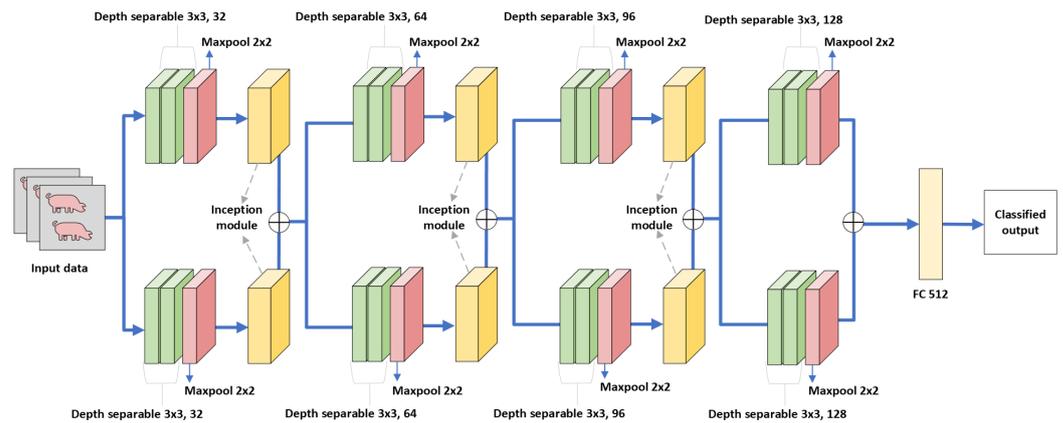


Figure 5. DISubNetV2: Depthwise separable convolution with inception module subnetwork. In this model, the inception modules are concatenated and fed as input to subsequent layers.

The depthwise separable convolution layers from both subnetworks are concatenated in the DISubNetV1. The concatenated output from both subnetworks is fed into the inception module. In the DISubNetV2 model, we concatenate inception modules from both subnetworks and feed them as input to the depthwise layers.

Depthwise separable convolutions, also known as separable convolutions, are one approach. It separates the channel and spatial convolutions normally combined in convolutional layers. The number of output channels equals the number of input channels because we apply one convolutional filter to each output channel. We then apply a pointwise convolutional layer after the depthwise convolutional layer. A pointwise convolutional layer is a convolutional layer with a 1×1 kernel. A 1×1 kernel is used to use non-linearity. A ReLU activation function is applied after each layer of a neural network. The inception module follows the same structure as the CNN-inception model. The inception modules from both subnetworks are concatenated and become inputs to the subsequent layers. Figure 6 illustrates a comparison of depthwise convolution layers and standard convolution layers.

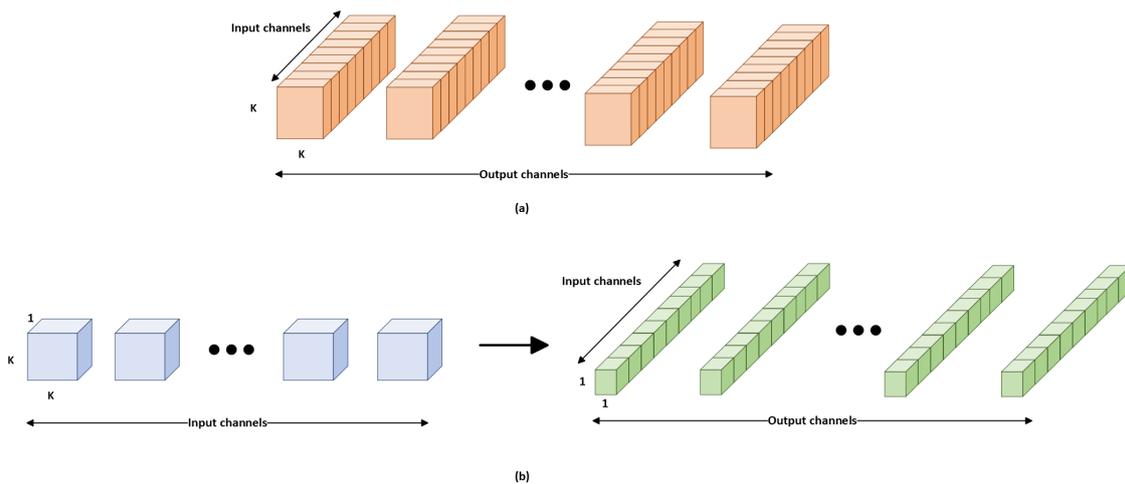


Figure 6. Comparison of depthwise separable and standard convolution layers. K refers to the kernel size. (a) Standard convolution filter, (b) Depthwise convolution followed by pointwise convolution.

The DISubNet models can regularize our model by reducing the number of parameters and the number of computations required during training or inference. Additionally, the model takes advantage of the inception module’s capacity to extract features from input data at different scales by employing different convolutional filter sizes. DISubNet models use computing resources efficiently with minimal increase in computation load.

4. Experimental Setup

4.1. Dataset

The data were collected by Wageningen University and Research using a FLIR camera. The FLIR T1020 with a standard 28-degree lens and FLIR Thermal Studio was used to acquire the thermal videos. Thermal videos of different pig treatments are included in the dataset. For simplicity, we extract the images from the video and convert them to grayscale with 62,800 images in total. The pigs were filmed in pairs and separated before and after feeding as shown in Figure 7, resulting in four treatment groups: isolation after feeding (IAF), isolation before feeding (IBF), paired after feeding (PAF), and paired before feeding (PBF).

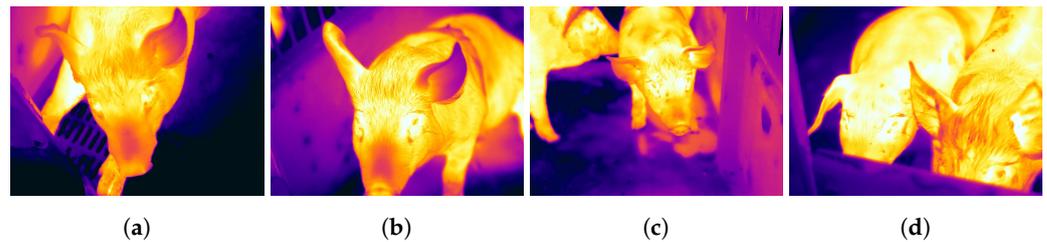


Figure 7. Thermal images of different pig treatments [53]. (a) IAF, (b) IBF, (c) PAF, and (d) PBF.

The pigs were classified into four treatment groups to assess animal welfare during physical separation and transport using a thermal camera. These labels represent the four different pig treatments as well as the experiment's required classified output. The thermal images of the IAF and IBF contained single pigs. The images in the PAF and PBF contain multiple pigs. Arousal in pigs is manipulated by delayed feeding due to short-term food restriction. Delaying feeding often increases the rate of eating, indicating higher arousal. Restrictive feeding tends to enhance aggression in pigs, which may result in adversarial social behavior when dealing with other pigs in the pen. To be able to build solutions and animal welfare monitoring systems for overcoming aggression and tail biting, it is crucial to analyze the impact of feeding intervals and pen mate manipulation behavior. The abnormal behavior of the pigs may be related to the redirection of the pig's exploratory behavior, such as the ability to engage with the pen mate whether maintained in groups or in isolation. Hence these four treatments namely IAF, IBF, PAF, and PBF were chosen to understand the effect of feeding intervals and access to socializing conditions on the behaviour of pigs. The entire dataset is divided into 60, 20, and 20 ratios for train, test, and validation data, respectively. As a result, the training data have 37,680 images, and the test data have 25,120 thermal images.

4.2. Implementation Details

The experiment uses images resized to 112×112 resolution. The models were trained using the Keras framework with a batch size of 32 and epochs of 100. All models have been trained on the Nvidia GeForce RTX 2070 SUPER GPU. For network training, the Adam optimization [55] method is used, which is an effective stochastic optimization that only requires first-order gradients and needs less memory. It combines the benefits of two common methods: AdaGrad [56], which works well with sparse gradients, and RMSProp [57], which works well in non-stationary and online settings. Instead of stochastic gradient descent, Adam is used to iteratively update network weights based on training data. The Adam technique is used to optimize the model at various learning rates, such as 10^{-2} , 10^{-3} , and 10^{-4} .

4.3. Loss Function

The categorical cross-entropy loss is also called softmax loss. It is closely related to the softmax function because categorical cross-entropy loss almost exclusively affects

networks with a softmax layer at the output. The categorical cross-entropy loss is only employed in multi-class classification tasks where each sample precisely belongs to one of the C classes. Each sample is given a ground truth label, an integer value between 0 and $C - 1$. A one-hot encoded vector of size C with a value for the correct class and zeroes everywhere can represent the label. The cross-entropy algorithm takes two discrete probability distributions as input and produces a single real-valued number indicating the correlation of both probability distributions. The categorical cross-entropy loss function is represented as,

$$E_{\text{loss}}(y,s) = - \sum_{i=1}^C y_i \log(s_i) \quad (1)$$

where C denotes the number of distinct classes and i denotes the i -th element of the vector. The one-hot encoded label is fed into y , and the probabilities generated by the softmax layer are placed in s . The lower the cross-entropy, the closer the two probability distributions are to one another.

4.4. Activation Function

ReLU is a non-linear activation function with output zero if the input x is less than zero and output equivalent to the input if the input is greater than zero. Hence, the ReLU function takes the maximum value of x . It has more advantages than the sigmoid function, which has more backpropagation errors. ReLU could be represented as

$$f(x) = \max(x, 0) \quad (2)$$

However, there are a few drawbacks to ReLU, including the fact that it is not zero-centred and is not differentiable at zero. Another issue that the ReLU faces is the dying ReLU problem in which some ReLU neurons essentially die for all inputs and remain inactive regardless of input, resulting in no gradient flow and affecting performance. As a result, we use LeakyReLU in experiments where there is a small negative slope so that instead of not firing at all for large gradients, the neurons do output some value, making the layer much more optimized. LeakyReLU is represented as

$$f(x) = \max(0.1x, x) \quad (3)$$

4.5. Evaluation Metrics

The accuracy, loss, F1 score, precision, recall, and number of parameters are used to compare the various models. The accuracy of the validation data measures how often the classifier predicts correctly. The precision metric explains how many of the correctly predicted cases were positive. It is useful in situations where false positives are more serious than false negatives. Recall describes how many of the actual positive cases the model correctly predicted. It is useful when false negatives are more concerning than false positives. The F1 score is derived from precision and recall metrics. It is also used to balance precision and recall when dealing with uneven dataset distribution. The evaluation metrics for the model are described as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1}_{\text{score}} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (7)$$

where TP , TN , FP , and FN represent true positive, true negative, false positive, and false negative, respectively. The confusion matrix is a popular performance metric for classification problems with two or more classes as output.

5. Results and Discussion

5.1. Model Comparison

We evaluated and visualized our results using an accuracy, loss, and confusion matrix. For our experiment, we have modified the LeNet5 for input data of 112×112 . The network consists of two sets of convolution layers followed by max pooling. The filter size for the convolution layer is 5×5 with stride 1, and the pooling size is 2×2 . There are 500 neurons in the hidden layers. The activation function used in this model is the ReLU activation function. With 19.6 M parameters, LeNet5 has an accuracy of 99.9%. After a certain epoch, the model converges but slightly overfits the model. With a learning rate of 10^{-3} , the LeNet5 was able to close the generalization gap with a 0.006 error. LeNet5 is limited by the availability of computing resources because processing higher-resolution images require larger and more convolutional layers, which are difficult to implement. Figure 8a,b show the accuracy and loss plot of the LeNet5 with slight overfitting at the beginning of the training.

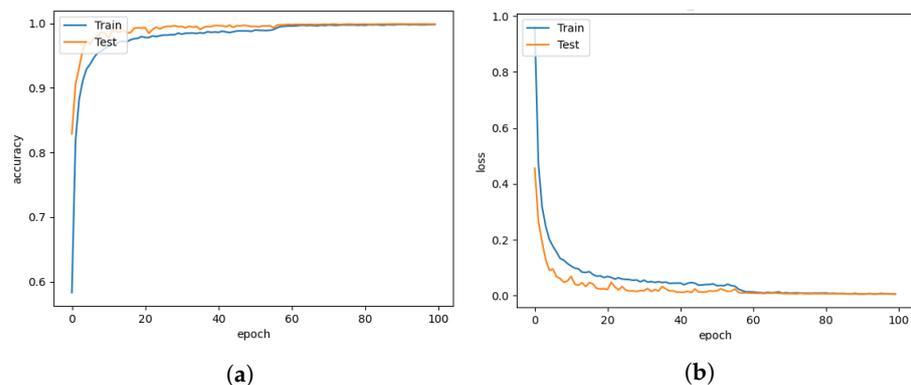


Figure 8. LeNet5 for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

The AlexNet model is slightly modified to use 4 convolutional layers instead of 5 for a 112×112 input size. The convolutional layers employ 11×11 , 5×5 , and 3×3 filter sizes. As a result of the varying convolution filter sizes, the network can learn various spatial patterns at different scales. The max pooling is applied with the size of 3×3 with stride 2. Despite having an accuracy of 90.22% with many parameters, AlexNet has several misclassified images. In comparison to LeNet5, AlexNet has 23.3 M parameters because of the addition of layers. As a result, AlexNet is not only a large model but also highly prone to overfitting. With a 0.27 error value, AlexNet has more errors than LeNet5. Figure 9a,b shows that AlexNet shows an accuracy plot and a loss plot.

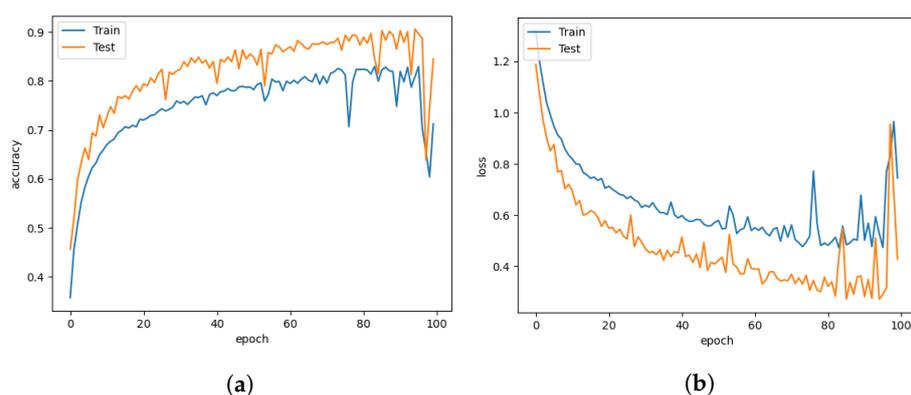


Figure 9. AlexNet for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

In this paper, we compare the 16-layer VGG-16 model with other models. VGG-19 was excluded from the experiment because it has a 55 M number of parameters. The convolutional layers are followed by single max pool layers. The layers use a 3×3 kernel size for a minimal receptive field. These are followed by the ReLU unit, which reduces training time compared to AlexNet. The number of depth layers has increased, and the hyperparameter tuning process has been simplified using only 3×3 filters. Consequently, increasing the depth of the model structure could enhance generalizability. Additionally, a larger receptive field might be offered. The number of parameters might be decreased by using a smaller filter size. Due to a large convergence gap between train and test data, VGGNet performed worse than other models. Figure 10a,b show that the VGGNet has a smoother learning curve than AlexNet. The model had an accuracy of 85.43% with 17 M parameters. Since the data are not evenly distributed, the VGGNet overfits similarly to AlexNet. With a 0.416 error, the VGGNet has a higher loss value than the AlexNet.

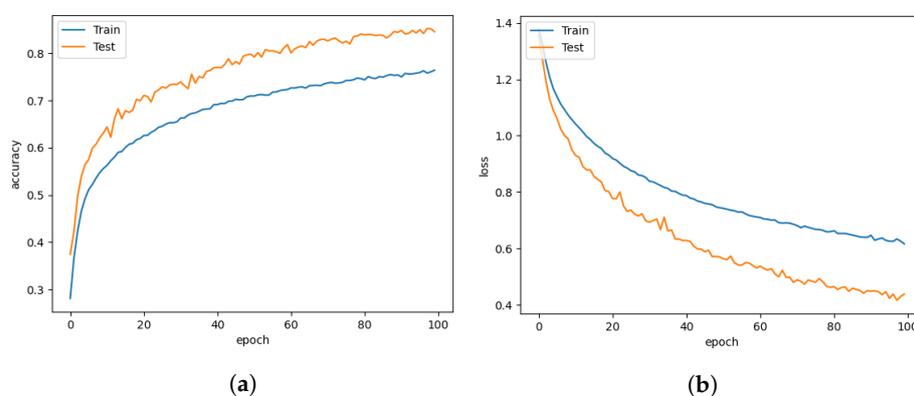


Figure 10. VGGNet for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

The Xception model emphasizes the inception hypothesis. Hence, this model is known as the Xception model. Xception provides an architecture that consists of depthwise separable convolution blocks and maxpooling, all of which are connected using shortcuts similar to ResNet implementations. The distinguishing characteristic of Xception is that the depthwise Convolution is not followed by the pointwise convolution; instead, the sequence is inverted. The 1×1 convolutions capture the correlations between channels. Regular 3×3 or 5×5 convolutions capture the spatial correlations within each channel. Hence, 1×1 is applied to each channel, followed by 3×3 to each output. It is similar to substituting depthwise separable convolutions for the inception module. Xception model has the accuracy of 99.95% with 20 M parameters. According to the accuracy and loss plots of the Xception model presented in Figure 11a,b, depthwise separable convolutions reduce overfitting compared to AlexNet and VGGNet. The Xception model has classification accuracy similar to DISubNet V1 and V2 but requires more parameters and a larger model size.

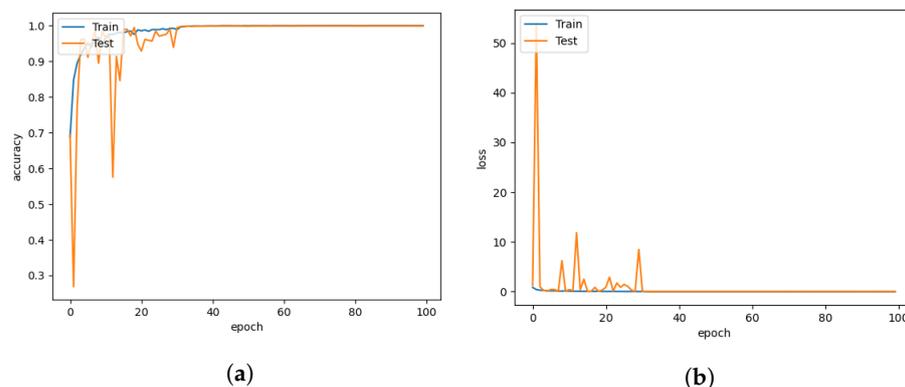


Figure 11. Xception for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

The confusion matrix in Figure 12 shows that the LeNet5 model classifies the paired before feeding treatment class more accurately than the other classes. When compared to other classes, the AlexNet model performs best at classifying isolation before feeding, followed by the class paired before feeding. Among the image classification methods, the VGGNet model illustrates the highly misclassified pig treatments. Furthermore, Xception performs a more accurate classification of pig treatments than LeNet5.

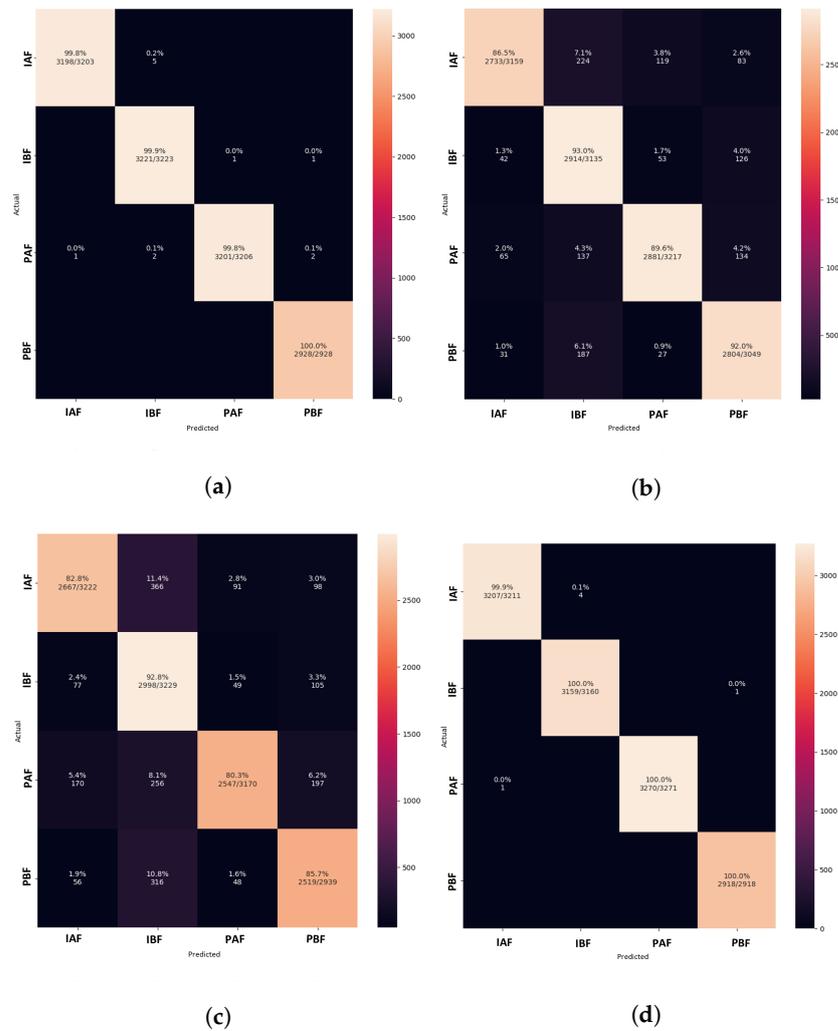


Figure 12. Confusion matrix of image classification models for learning rate = 0.001. (a) LeNet5. (b) AlexNet. (c) VGGNet. (d) Xception.

In comparison to Lenet5, which uses 19.6 M parameters, the CNN-leakyReLU achieves an accuracy of 99.14% with 7.2 M parameters. Figure 13a,b demonstrate CNN-leakyReLU with more fluctuations in the learning curve at the beginning of the training. The model fluctuated during training due to the uneven data distribution, but it converged successfully after a certain number of epochs. With a 0.097 error, it displays a higher loss value than LeNet5. An L2 regularizer is used to lessen the overfitting of the proposed model. The confusion matrix shown in Figure 14a indicates that most pig treatment classes were also categorized with higher performance.

The CNN-inception model makes use of the ability of the inception module to focus on different parts of images to find patterns that can be associated with classification labels. Working with different filters to capture the level of abstraction is possible with the inception. As a result, they are not limited to using a single filter size in a single image block, which is then concatenated and passed onto the next layer. After each max pooling, the inception module is added. When the dataset is trained with the CNN-Inception

model, it captures better patterns. It thus achieves 99.97% accuracy with a slightly higher number of parameters (i.e., 7.4 M) than CNN-LeakyReLU. Figure 15a,b demonstrate that the CNN-inception model has a better learning and convergence curve than the other models.

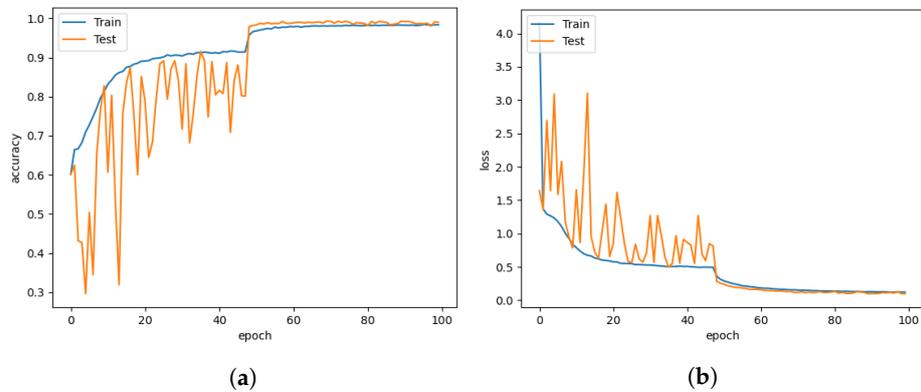


Figure 13. CNN-LeakyReLU for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

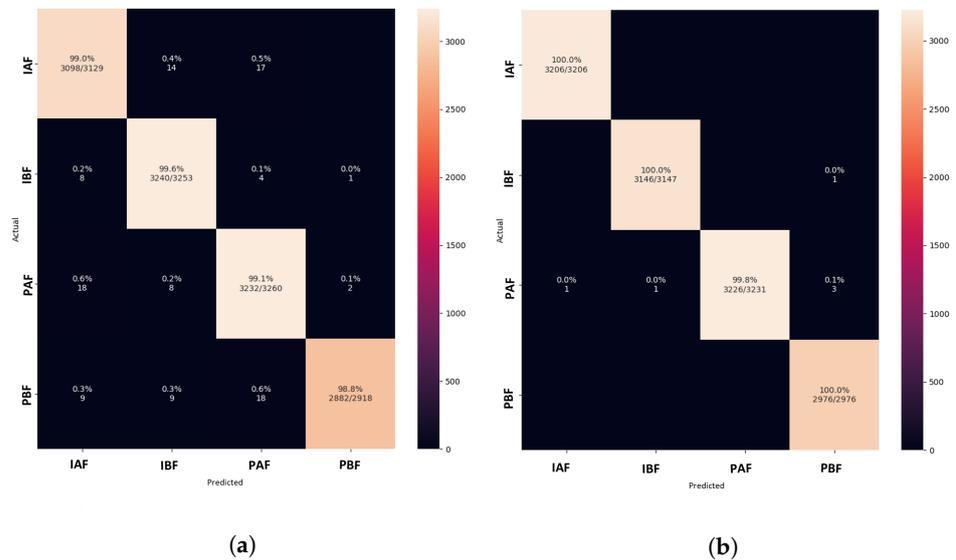


Figure 14. Confusion matrix of modified CNN models for learning rate = 0.001. (a) CNN-LeakyReLU. (b) CNN-Inception.

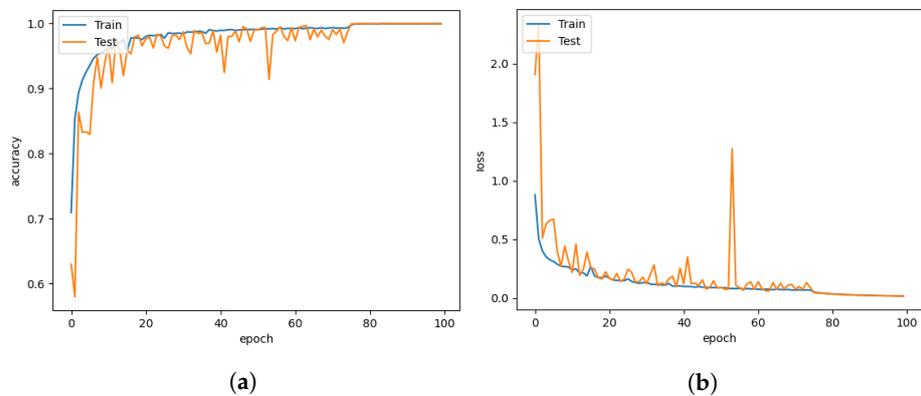


Figure 15. CNN-Inception for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

In the model, the filters are slid over the entire image, and the dot product of the image and filter values are calculated. The number of filters produces the same number of feature maps as the number of filters, which becomes the parameter for the model to be learned. Deep neural networks that are highly efficient must be large. A neural

network had to have several more layers and units within these layers to be considered large. Multi-scale convolutional layers may also be able to learn more. However, large networks are prone to overfitting, and chaining multiple convolutional operations together raises the computational cost of the network [51]. In this case, the inception module is more advantageous. When compared to CNN-LeakyReLU, the model achieves a lower loss of 0.017. As a result, for use in any application, a trade-off between the number of parameters and accuracy could be considered. The CNN-Inception model correctly classifies three treatment categories, as shown by the confusion matrix in Figure 14b.

The DISubNet model, which employs depthwise separable convolution layers, has significantly fewer parameters and a slightly lower train time per epoch. A normal convolutional layer differs from a depthwise convolution where the depthwise convolution applies the convolution along only one spatial dimension (i.e., channel), whereas a normal convolution applies the convolution across all spatial dimensions or channels at each step. Depthwise separable convolutions are more likely to perform more effectively on deeper models that may have an overfitting problem and on layers with larger kernels because there is a greater decrease in parameters and computations that would offset the high computation cost of performing two convolutions instead of one. Non-linear layers broaden the model's possibilities, making a deep network superior to a wide network. We use a 1×1 kernel and add an activation layer after it to increase the number of non-linear layers without significantly increasing the number of parameters and computations. This adds a layer of depth to the network. Based on the model structure, our proposed model has two versions: DISubNetV1 and DISubNetV2. Depthwise convolution layers from both subnetworks are concatenated to form the DISubNetV1. Because the depthwise layers are close to the input, it extracts low-level features and concatenates features from both subnetworks to provide more information to the inception module. This version of the model achieves 99.96% accuracy, which is higher than all other models except CNN-Inception. In Figure 16a,b, the accuracy and loss plots of DISubNetV1 exhibit better convergence and fewer fluctuations. The DISubNetV2 concatenates inception modules rather than depthwise layers. At the beginning of the model, the input from different subnetworks goes through different levels of abstraction with different filters. As a result, it enables in obtaining more features when concatenated and provides better classification output. Regarding accuracy, the DISubNetV2 outperformed all other models with a score of 99.98% on thermal data. Although there are a few more fluctuations in the accuracy and loss of DISubNetV2 in Figure 17a,b, there is a better learning curve over the course of training. Even though DISubNetV2 has 0.002 more errors than DISubNetV1, it can still be used as a straightforward model with 4.5 M parameters.

In comparison with other models, the confusion matrix of both proposed versions in Figure 18a,b shows correctly classified pig treatment classes. As a result, the model outperforms other models trained on thermal data from pig treatments.

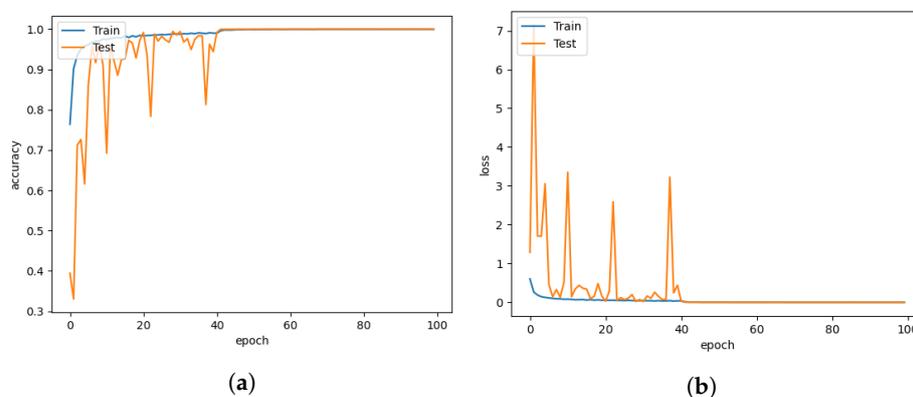


Figure 16. DISubNetV1 for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

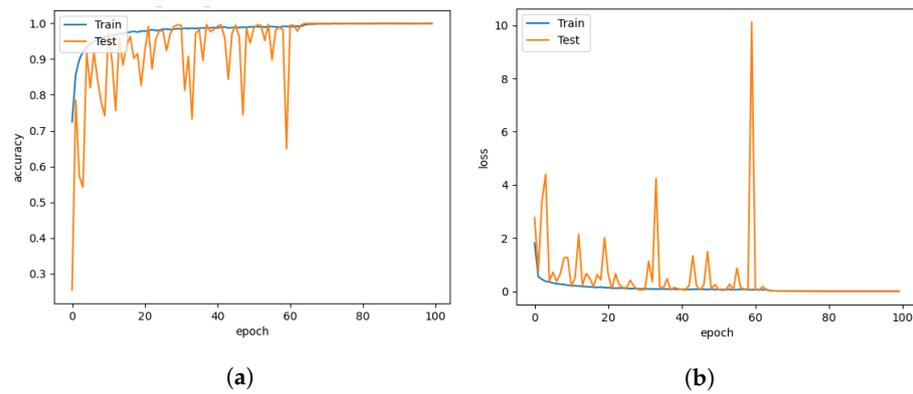


Figure 17. DISubNetV2 for learning rate = 0.001. (a) Model accuracy. (b) Model loss.

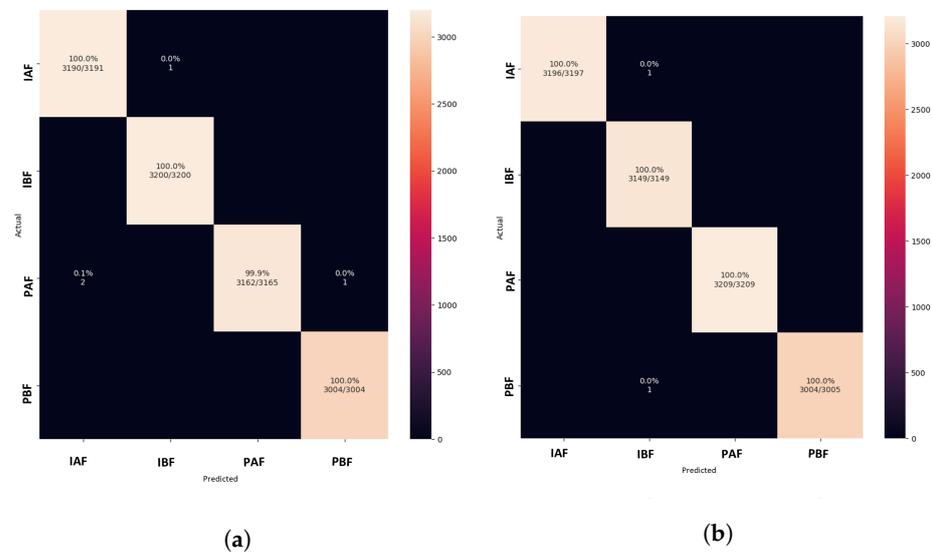


Figure 18. Confusion matrix of proposed DISubNet models for learning rate = 0.001. (a) DISubNetV1. (b) DISubNetV2.

Table 1 summarizes our results for learning rate = 0.001. The proposed model DISubNet models, DISubNetV1 and DISubNetV2, provides increased accuracy compared to all the models for pig treatment classification.

Table 1. Performance comparison of different models with learning rate = 0.001.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Loss
LeNet5	99.9045	99.9045	99.9045	99.9045	0.0061
AlexNet	90.2229	90.2581	90.2581	90.2229	0.2716
VGGNet	85.4379	86.3148	85.5091	85.4379	0.4164
Xception	99.9522	99.9522	99.9522	99.9522	0.0043
CNN-LeakyReLU	99.1401	99.1426	99.1401	99.1403	0.0976
CNN-inception	99.9761	99.9761	99.9761	99.9761	0.0179
DISubNetV1	99.9682	99.9682	99.9682	99.9682	0.0014
DISubNetV2	99.9841	99.9841	99.9841	99.9841	0.0036

5.2. Comparison with Different Learning Rates

Our proposed models were trained at various learning rates, including 10^{-2} , 10^{-3} , and 10^{-4} . Table 2 summarizes the experiment and includes evaluation metrics such as accuracy, precision, recall, and F1 score.

Table 2. Comparison of all models with different learning rates.

Model	Learning Rate	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
LeNet5	0.01	25.3025	6.4022	25.3025	10.2188
	0.0001	99.9124	99.9125	99.9124	99.9124
AlexNet	0.01	24.8885	6.1944	24.8885	9.9199
	0.0001	99.9602	99.9602	99.9602	99.9602
VGGNet	0.01	25.6528	6.5806	25.6528	10.4744
	0.0001	99.9840	99.9840	99.9840	99.9840
Xception	0.01	99.9682	99.9682	99.9682	99.9682
	0.0001	99.9920	99.9920	99.9920	99.9920
CNN-LeakyReLU	0.01	25.8996	6.7079	25.8996	10.6560
	0.0001	99.9601	99.9601	99.9601	99.9601
CNN-inception	0.01	37.7388	50.2383	37.7388	32.1467
	0.0001	99.9681	99.9681	99.9681	99.9681
DISubNetV1	0.01	25.1433	6.3218	25.1433	10.1034
	0.0001	99.9682	99.9682	99.9682	99.9682
DISubNetV2	0.01	25.2229	6.3619	25.2229	10.1610
	0.0001	99.9920	99.9920	99.9920	99.9920

All models perform better with lower learning rates, such as 10^{-3} and 10^{-4} . Furthermore, for the learning rate of 10^{-4} , our proposed models outperformed all other models with improved accuracy in the range of 99.96–99.99%. It also clearly shows that at higher learning rates, all models have an accuracy of less than 40% excluding the Xception model. With a learning rate of 10^{-2} , Xception outperforms all other models with an accuracy of 99.96%. However, the proposed model is smaller in compared to number of parameters. Though VGGNet has a similar accuracy of 99.98% to DISubNetV2, it is a relatively large model with 17.7 M parameters, particularly in comparison to DISubNetV2 which has 4.5 M parameters. The models are unable to converge well when the learning rate is 10^{-2} , which may be caused by a smaller validation data sample or an uneven distribution of data. Since the dataset for paired before feeding data contains few samples, all models exhibit high learning fluctuations without increasing the accuracy. On the other hand, performance improves when the learning rate is reduced. Therefore, it is obvious that lowering the learning rate when training these models will result in better performance. In a few instances, the unbalanced dataset makes it challenging to learn the model for each batch, producing a high loss value.

5.3. Comparison with Number of Parameters and Model Size

In comparison to other models, our proposed models, DISubNetV1 and DISubNetV2, provide few parameters. The number of parameters typically rises when CNN models are expanded, potentially leading to a deeper model. However, this might impact the accuracy gain caused by the vanishing gradient. The depthwise convolution layer model requires fewer parameters and is more accurate. Table 3 compares all models in terms of parameter count and model size (in MB). With 4.5 M parameters, our suggested model yields a size of 53.7 MB.

It is advantageous to have lightweight models in applications that run on mobile devices. Mobile-based deep learning applications have the potential to revolutionize pig farming by providing farmers with real-time data and insights that can help them optimize their operations and improve animal welfare. With the use of mobile-based deep learning applications, farmers can identify each pig in their herd and track their growth and health. This information can be used to monitor individual pig performance and to identify and address any health issues early on. Deep learning models can be trained to analyze pig

behavior, such as eating and drinking patterns, activity levels, and social interactions. This information can be used to identify any abnormal behavior, which could be a sign of stress, illness, or other problems. With the use of mobile-based deep learning applications, farmers can use predictive analytics to forecast the growth rate of their pigs, identify potential health problems early, and optimize their feeding and breeding strategies. By monitoring the individual behavior and performance of pigs, farmers can optimize their resource allocation, such as feed and water, and minimize waste. The use of mobile-based deep learning applications can help farmers save time and money by automating data collection and analysis, reducing the need for manual labor, and increasing efficiency.

Table 3. Model parameters and size comparison.

Model	Number of Parameters	Model Size
LeNet5	19,628,074	224 MB
AlexNet	23,392,580	267 MB
VGGNet	17,075,396	195 MB
Xception	20,991,980	240 MB
CNN-LeakyReLU	7,255,332	83.2 MB
CNN-inception	7,419,812	85.8 MB
DISubNetV1	4,591,574	53.7 MB
DISubNetV2	4,591,574	53.7 MB

5.4. Importance of Pig Treatment Classification in Animal Welfare

Pig treatment classification can be applied to many aspects of farming and animal care. The goal of the model is to create a framework for a decision support system for predictive analytics that can be used to identify changes in pig behaviour in response to environmental perturbations such as shifts in playtime, feeding interval time, and rest time. Isolated pigs develop behavioral stress reactions. Pigs that are completely isolated continue to display behavioural signs of stress, whereas pigs that are partially isolated (contact through a fence) eventually display fewer behavioral signs of stress [58]. Researchers working with animals can use these data to advocate for better treatment of animals. Future monitoring and treatment could benefit from using a non-invasive thermal camera to record the skin's surface temperature. In veterinary medicine, thermal imaging is used to help diagnose diseases and to detect (early) signs of pain or stress in animals. Thermal imaging can also detect postoperative inflammation and changes in blood flow to the surgical site. Therefore, thermal images are a useful tool for identifying issues that may impact animal welfare.

6. Conclusions

This paper proposed the DISubNetV1 and DISubNetV2 models, which are made up of depthwise convolution layers and inception modules for classifying pig treatments. Various evaluation metrics are used to compare the proposed model to LeNet5, AlexNet, VGGNet, Xception, CNN-LeakyReLU, and CNN-inception models. The versions differ in terms of the concatenation of the layers in the subnetworks. Based on thermal data, the models classify four pig treatment categories. The proposed model outperforms all other models with fewer parameters and higher accuracy. Although the model improves accuracy, it misclassifies one of the paired before-feeding classes. It also shows fluctuations in learning due to the uneven distribution of the data. In the future, we plan to use this research for other applications such as emotion recognition to provide better information based on the features learned in the pig treatment classification. Since only thermal images were used, we intend to use videos instead. In addition, the conversion of thermal scores to grayscale may have resulted in the loss of some features. Therefore, future work on the model must target the feature loss to improve its accuracy.

Author Contributions: Writing—original draft, S.J.C., J.H.K., A.P. and D.S.H.; Writing—review and editing, S.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2021R1A6A1A03043144) and the BK21 Four project funded by the Ministry of Education, Korea (4199990113966).

Institutional Review Board Statement: The data used in this paper were graciously provided by Professor Neethirajan’s research team and the data belongs to another experiment. These animals were used for another research study approved by the CCD and IVD of the Netherlands. The safety and health department of CARUS (Wageningen University and Research, Wageningen, The Netherlands) approved any additional non-invasive handling for this study. The approval number is 20210521ADP.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the authors upon reasonable request.

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

References

1. Xing, F.; Xie, Y.; Su, H.; Liu, F.; Yang, L. Deep learning in microscopy image analysis: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 4550–4568. [[CrossRef](#)] [[PubMed](#)]
2. Stewart, M.; Webster, J.; Schaefer, A.; Cook, N.; Scott, S. Infrared thermography as a non-invasive tool to study animal welfare. *Anim. Welf.* **2005**, *14*, 319–325. [[CrossRef](#)]
3. Cangar, Ö.; Aerts, J.-M.; Buyse, J.; Berckmans, D. Quantification of the spatial distribution of surface temperatures of broilers. *Poult. Sci.* **2008**, *87*, 2493–2499. [[CrossRef](#)] [[PubMed](#)]
4. Warriss, P.; Pope, S.; Brown, S.; Wilkins, L.; Knowles, T. Estimating the body temperature of groups of pigs by thermal imaging. *Vet Rec.* **2006**, *158*, 331–334. [[CrossRef](#)]
5. Weschenfelder, A.V.; Saucier, L.; Maldague, X.; Rocha, L.M.; Schaefer, A.L.; Faucitano, L. Use of infrared ocular thermography to assess physiological conditions of pigs prior to slaughter and predict pork quality variation. *Meat Sci.* **2013**, *95*, 616–620. [[CrossRef](#)]
6. Rodriguez-Baena, D.S.; Gomez-Vela, F.A.; García-Torres, M.; Divina, F.; Barranco, C.D.; Daz-Diaz, N.; Jimenez, M.; Montalvo, G. Identifying livestock behavior patterns based on accelerometer dataset. *J. Comput. Sci.* **2020**, *41*, 101076. [[CrossRef](#)]
7. Dutta, R.; Smith, D.; Rawnsley, R.; Bishop-Hurley, G.; Hills, J.; Timms, G.; Henry, D. Dynamic cattle behavioural classification using supervised ensemble classifiers. *Comput. Electron. Agric.* **2015**, *111*, 18–28. [[CrossRef](#)]
8. Phung Cong Phi, K.; Nguyen Thi, K.; Nguyen Dinh, C.; Tran Duc, N.; Tran Duc, T. Classification of cow’s behaviors based on 3-DoF accelerations from cow’s movements. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 1656–1662.
9. Becciolini, V.; Ponzetta, M.P. Inferring behaviour of grazing livestock: Opportunities from GPS telemetry and activity sensors applied to animal husbandry. *Eng. Rural Dev.* **2018**, *17*, 192–198.
10. Decandia, M.; Giovanetti, V.; Acciaro, M.; Mameli, M.; Molle, G.; Cabiddu, A.; Manca, C.; Cossu, R.; Serra, M.; Rassu, S. Monitoring grazing behaviour of Sarda cattle using an accelerometer device. In *Grassland Resources for Extensive Farming Systems in Marginal Lands: Major Drivers and Future Scenarios, Proceedings of the 19th Symposium of the European Grassland Federation, Alghero, Italy, 7–10 May 2017*; Istituto Sistema Produzione Animale Ambiente Mediterraneo: Sassari, Italy, 2017; Volume 22, p. 143.
11. McManus, R.; Boden, L.A.; Weir, W.; Viora, L.; Barker, R.; Kim, Y.; McBride, P.; Yang, S. Thermography for disease detection in livestock: A scoping review. *Front. Vet. Sci.* **2022**, *1163*, 1163. [[CrossRef](#)]
12. Sykes, D.; Couvillion, J.; Cromiak, A.; Bowers, S.; Schenck, E.; Crenshaw, M.; Ryan, P. The use of digital infrared thermal imaging to detect estrus in gilts. *Theriogenology* **2012**, *78*, 147–152. [[CrossRef](#)] [[PubMed](#)]
13. Pacheco, V.M.; de Sousa, R.V.; da Silva Rodrigues, A.V.; de Souza Sardinha, E.J.; Martello, L.S. Thermal imaging combined with predictive machine learning based model for the development of thermal stress level classifiers. *Livest. Sci.* **2020**, *241*, 104244. [[CrossRef](#)]
14. Arulmozhi, E.; Basak, J.K.; Sihalath, T.; Park, J.; Kim, H.T.; Moon, B.E. Machine learning-based microclimate model for indoor air temperature and relative humidity prediction in a swine building. *Animals* **2021**, *11*, 222. [[CrossRef](#)] [[PubMed](#)]
15. Oliveira, D.A.B.; Pereira, L.G.R.; Bresolin, T.; Ferreira, R.E.P.; Dorea, J.R.R. A review of deep learning algorithms for computer vision systems in livestock. *Livest. Sci.* **2021**, *253*, 104700. [[CrossRef](#)]
16. Boileau, A.; Farish, M.; Turner, S.P.; Camerlink, I. Infrared thermography of agonistic behaviour in pigs. *Physiol. Behav.* **2019**, *210*, 112637. [[CrossRef](#)]
17. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 2818–2826.

18. Alameer, A.; Kyriazakis, I.; Dalton, H.A.; Miller, A.L.; Bacardit, J. Automatic recognition of feeding and foraging behaviour in pigs using deep learning. *Biosyst. Eng.* **2020**, *197*, 91–104. [[CrossRef](#)]
19. Alameer, A.; Kyriazakis, I.; Bacardit, J. Automated recognition of postures and drinking behaviour for the detection of compromised health in pigs. *Sci. Rep.* **2020**, *10*, 1–15. [[CrossRef](#)]
20. Peng, S.; Zhu, J.; Liu, Z.; Hu, B.; Wang, M.; Pu, S. Prediction of Ammonia Concentration in a Pig House Based on Machine Learning Models and Environmental Parameters. *Animals* **2022**, *13*, 165. [[CrossRef](#)]
21. Tusell, L.; Bergsma, R.; Gilbert, H.; Gianola, D.; Piles, M. Machine learning prediction of crossbred pig feed efficiency and growth rate from single nucleotide polymorphisms. *Front. Genet.* **2020**, *11*, 567818. [[CrossRef](#)]
22. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Trnovszky, T.; Kamencay, P.; Orjesek, R.; Benco, M.; Sykora, P. Animal recognition system based on convolutional neural network. *Adv. Electr. Electron. Eng.* **2017**, *15*, 517–525. [[CrossRef](#)]
25. Kernel (Image Processing). Available online: [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)) (accessed on 23 September 2019).
26. Mouloudi, S.; Rahmanpanah, H.; Burvill, C.; Davies, H.M. Prediction of displacement in the equine third metacarpal bone using a neural network prediction algorithm. *Biocybern. Biomed. Eng.* **2020**, *40*, 849–863. [[CrossRef](#)]
27. Hamilton, A.W.; Davison, C.; Tachtatzis, C.; Andonovic, I.; Michie, C.; Ferguson, H.J.; Somerville, L.; Jonsson, N.N. Identification of the rumination in cattle using support vector machines with motion-sensitive bolus sensors. *Sensors* **2019**, *19*, 1165. [[CrossRef](#)] [[PubMed](#)]
28. Rahman, A.; Smith, D.; Hills, J.; Bishop-Hurley, G.; Henry, D.; Rawnsley, R. A comparison of autoencoder and statistical features for cattle behaviour classification. In Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2954–2960.
29. Rahman, A.; Smith, D.; Little, B.; Ingham, A.; Greenwood, P.; Bishop-Hurley, G. Cattle behaviour classification from collar, halter, and ear tag sensors. *Inf. Process. Agric.* **2018**, *5*, 124–133. [[CrossRef](#)]
30. Riaboff, L.; Poggi, S.; Madouasse, A.; Couvreur, S.; Aubin, S.; Bédère, N.; Goumand, E.; Chauvin, A.; Plantier, G. Development of a methodological framework for a robust prediction of the main behaviours of dairy cows using a combination of machine learning algorithms on accelerometer data. *Comput. Electron. Agric.* **2020**, *169*, 105179. [[CrossRef](#)]
31. Benaissa, S.; Tuytens, F.A.; Plets, D.; Cattrysse, H.; Martens, L.; Vandaele, L.; Joseph, W.; Sonck, B. Classification of ingestive-related cow behaviours using RumiWatch halter and neck-mounted accelerometers. *Appl. Anim. Behav. Sci.* **2019**, *211*, 9–16. [[CrossRef](#)]
32. Williams, L.R.; Bishop-Hurley, G.J.; Anderson, A.E.; Swain, D.L. Application of accelerometers to record drinking behaviour of beef cattle. *Anim. Prod. Sci.* **2017**, *59*, 122–132. [[CrossRef](#)]
33. Smith, D.; Little, B.; Greenwood, P.I.; Valencia, P.; Rahman, A.; Ingham, A.; Bishop-Hurley, G.; Shahriar, M.S.; Hellicar, A. A study of sensor derived features in cattle behaviour classification models. In Proceedings of IEEE SENSORS, Busan, Republic of Korea, 1–4 November 2015; pp. 1–4.
34. Villa, A.G.; Salazar, A.; Vargas, F. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Ecol. Inform.* **2017**, *41*, 24–32. [[CrossRef](#)]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
37. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
38. 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 Proceedings of IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 40–49.
39. Brown, G.V.; Warrell, D.A. Venomous bites and stings in the tropical world. *Med. J. Aust.* **1993**, *159*, 773–779. [[CrossRef](#)]
40. Giraldo-Zuluaga, J.-H.; Salazar, A.; Gomez, A.; Diaz-Pulido, A. Automatic recognition of mammal genera on camera-trap images using multi-layer robust principal component analysis and mixture neural networks. *arXiv* **2017**, arXiv:1705.02727.
41. Neethirajan, S. The role of sensors, big data and machine learning in modern animal farming. *Sens.-Bio-Sens. Res.* **2020**, *29*, 100367. [[CrossRef](#)]
42. Neethirajan, S. Happy cow or thinking pig? Wur wolf—facial coding platform for measuring emotions in farm animals. *AI* **2021**, *2*, 342–354. [[CrossRef](#)]
43. Neethirajan, S. ChickTrack—a quantitative tracking tool for measuring chicken activity. *Measurement* **2022**, *191*, 110819. [[CrossRef](#)]
44. Heuvel, H.v.d.; Graat, L.; Youssef, A.; Neethirajan, S. Quantifying the Effect of an Acute Stressor in Laying Hens using Thermographic Imaging and Vocalizations. *bioRxiv* **2022**. [[CrossRef](#)]

45. Jin, J.; Dundar, A.; Culurciello, E. Flattened convolutional neural networks for feedforward acceleration. *arXiv* **2014**, arXiv:1412.5474.
46. Wang, M.; Liu, B.; Foroosh, H. Factorized convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 545–553.
47. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
48. Hassibi, B.; Stork, D. Second order derivatives for network pruning: Optimal brain surgeon. *Adv. Neural. Inf. Process. Syst.* **1992**, *5*, 164–171.
49. Ahmed, K.; Torresani, L. Connectivity learning in multi-branch networks. *arXiv* **2017**, arXiv:1709.09582.
50. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. In Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
51. LeCun, Y. LeNet-5, Convolutional Neural Networks. Available online: <http://yann.lecun.com/exdb/lenet> (accessed on 26 January 2023).
52. Chollet, F. In Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
53. Colaco, S.J.; Kim, J.H.; Poulouse, A.; Van, Z.S.; Neethirajan, S.; Han, D.S. Pig Treatment Classification on Thermal Image Data using Deep Learning. In Proceedings of 13th IEEE International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5–8 July 2022; pp. 8–11.
54. 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* **2017**, arXiv:1704.04861.
55. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
56. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
57. Tieleman, T.; Hinton, G. *Lecture 6.5-rmsprop*, Coursera: *Neural Networks for Machine Learning*; Technical Report; University of Toronto: Toronto, ON, Canada, 2012; Volume 6.
58. Brandt, P.; Rousing, T.; Herskin, M.; Olsen, E.; Aaslyng, M. Development of an index for the assessment of welfare of finishing pigs from farm to slaughter based on expert opinion. *Livest. Sci.* **2017**, *198*, 65–71. [[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.