


## Article

# Improving Road Safety during Nocturnal Hours by Characterizing Animal Poses Utilizing CNN-Based Analysis of Thermal Images

Derian Mowen <sup>1,\*</sup>, Yuvaraj Munian <sup>2,\*</sup>  and Miltiadis Alamaniotis <sup>2</sup>

<sup>1</sup> Department of Computer Science, Trinity University, San Antonio, TX 78212, USA

<sup>2</sup> Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA

\* Correspondence: dmowen265@gmail.com (D.M.); yuvaraj.munian@utsa.edu (Y.M.)

**Abstract:** Animal–vehicle collision is a common danger on highways, especially during nighttime driving. Its likelihood is affected not only by the low visibility during nighttime hours, but also by the unpredictability of animals’ actions when a vehicle is nearby. Extensive research has shown that the lack of visibility during nighttime hours can be addressed using thermal imaging. However, to our knowledge, little research has been undertaken on predicting animal action through an animal’s specific poses while a vehicle is moving. This paper proposes a new system that couples the use of a two-dimensional convolutional neural network (2D-CNN) and thermal image input, to determine the risk imposed by an animal in a specific pose to a passing automobile during nighttime hours. The proposed system was tested using a set of thermal images presenting real-life scenarios of animals in specific poses on the roadside and was found to classify animal poses accurately in 82% of cases. Overall, it provides a valuable basis for implementing an automotive tool to minimize animal–vehicle collisions during nighttime hours.

**Keywords:** 2D convolutional neural network; animal–vehicle collision; thermal imaging; animal action prediction; pose detection; nocturnal



**Citation:** Mowen, D.; Munian, Y.; Alamaniotis, M. Improving Road Safety during Nocturnal Hours by Characterizing Animal Poses Utilizing CNN-Based Analysis of Thermal Images. *Sustainability* **2022**, *14*, 12133. <https://doi.org/10.3390/su141912133>

Academic Editors: Sikai Chen, Ali Karimoddini, Samuel Labi and Hualiang (Harry) Teng

Received: 28 August 2022

Accepted: 22 September 2022

Published: 25 September 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

Wildlife–vehicle collision represents a costly and lethal consequence of human technology interfacing with animal environments. Wildlife to vehicle collisions are estimated to be responsible for over 35,000 yearly automobile incidents in the United States, resulting in approximately 200 human fatalities per year [1]. This in turn leads to about 4000 insurance case filings per year, with an average cost of USD 1000 per individual case. Furthermore, roadside collisions are the most prominent threat to many endangered species within the United States [1]. There have been numerous attempts to mitigate these losses through various methods, such as using electrical mats and wildlife fencing to keep animals from entering the roadway [2,3]. However, these methods have proven inefficient and ineffective; none of these proposed solutions have fully solved the issue of vehicle collisions with wildlife on the roadway.

In this study, a new artificial intelligence system was designed to mitigate the issue of wildlife–vehicle collisions during nighttime hours, aiming at minimizing financial and human loss. Research has shown that the risk of wildlife–vehicle collisions can be reduced by dynamic and transferable state prediction using machine learning [4]. The proposed system targets classification of animals’ positions using roadside thermal images of the environment, to determine the risk that an animal in a specific pose presents to a passing automobile. The system implements a two-dimensional convolutional neural network, with the data input being thermal images of roadside scenes including wildlife images of antlered animals on the roadside near to a passing automobile. It should be noted that

antlered animals were chosen for use within these images as they make up most of the fatal wildlife-vehicle collisions in the United States [1].

For our purposes, thermal images were collected, processed, and then run through the system to identify the poses. Thermal imagery is far from a recent technology, having been developed in the 1950s for application within the medical field [5], and having since found various applications in medicine, biology, ecology, police use, and military applications [6]. Images are generated by capturing light naturally emitted from heated objects [7] by using infrared technology, with the thermal images indicating heat emission differences in captured scenes. Thermal imaging has successfully addressed the issue of observing and locating animals during the night [8]. Additionally, simulation studies have shown that vehicle-mounted thermal imaging methods can significantly improve anticipatory driver control, reducing the likelihood of an automobile accident [9].

The proposed intelligent system uses a two-dimensional convolutional neural network (2D-CNN) to classify incoming thermal images in real time. Convolutional neural networks are composed of four layers, i.e., a convolutional layer, a non-linearity layer, a pooling layer, and a fully connected layer [10]. A 2-D convolutional neural network receives an image as input and then breaks down the image into features that are sent through the network layers, resulting in a specific output associated with the task at hand. The primary motivator for the use of 2D-CNNs in this work was their reported success and efficiency for image detection and classification [10]. More specifically, 2D-CNNs have demonstrated a distinct advantage in image classification tasks [11] and have been successfully applied to create novel animal detection and collision avoidance systems for enhancing driver safety [12].

## 2. Literature Review

A variety of systems implementing artificial intelligence methods have been developed to assist drivers with on-road safety. The coupling of thermal imagery and convolutional neural networks has successfully been used to identify potholes in the road [13]. That system's self-built model used a two-dimensional convolutional neural network to attain detection of potholes from thermal images, with approximately 63% accuracy. However, one disadvantage of the system was that it was only trained on images of potholes and not on images of roadside scenes. Wildlife detection using convolutional neural network methods has also been successfully applied with roadside thermal images [14]. The previously described system successfully identified wildlife within roadside scenes, using a one-dimensional convolutional neural network, with an approximate accuracy of 89%. The system showcased the advantage of detecting wildlife from roadside scenes that may or may not contain animals.

Additionally, collision-avoidance systems using convolutional neural networks have been successfully implemented [12]. The implementation of the collision-avoidance system achieved an accuracy of 82.5% for detecting cows from the roadway. The disadvantage to that particular work was the tradeoff between the cost of deployment and cow collision risk; cows account for barely 6% of fatal human collisions with animals. On the contrary, antlered animals comprise 94% of deadly crashes, limiting the usefulness of the work in [12]. In direct relation to this current article, other research [15] assessed animals' orientation from thermal images taken during nighttime hours to predict the trajectory of an animal's movement. Furthermore, animals have been shown to elicit anti-predatory responses when caught in vehicle headlights [16]. The model described in [15] was shown to be more efficient for classifying the animal pose than state-of-the-art methods including histogram of oriented gradients with support vector machine, or the boosted Har-stumps methodology. Given the success of the idea of using animal pose identification to predicting animal movement [15], the intelligent system described in the current paper builds upon this methodology utilizing animal pose classification from thermal images. The image classification was performed with a 2D-CNN by processing images of complex roadside scenes as seen by a passing automobile, as opposed to [15] where only pure animal images

roadside scenes as seen by a passing automobile, as opposed to [15] where only pure animal images were used (i.e., no roadside scenes). Furthermore, the proposed system can be used (i.e., no roadside scenes). Furthermore, the proposed system can contribute to the development of autonomous and intelligent vehicle safety methods in the broader context of smart cities.

The remainder of this article is formed as follows: the methods behind collecting the thermal image datasets are described in Section 3; the developed intelligent system is introduced in Section 4; testing results from real roadside scenes are presented in Section 5; Section 6 concludes the work and outlines the future progression of this topic.

### 3. Data Description

This section provides a description of the data used for the development of the artificial intelligence model. Furthermore, the data collection method is also described, together with the setup utilized during the images.

#### 3.1. Thermal Image System Setup

This research made use of a Forward Looking Infrared (FLIR) thermal camera. Specifically, a FLIR One Pro camera for an iOS device was used to capture the images of roadside scenes with wildlife during night hours. An example of image capture from this device is provided in Figure 1. The focus of the FLIR camera ranges from 15 cm to infinity, and the operating temperature for the FLIR system is between 0 and 35 degrees Celsius with a dynamic scene range of 120 to 220 degrees Celsius. This FLIR Pro model can capture images with a thermal sensitivity of 100 mK and is also capable of recording image and video output in the corresponding MPEG and MOV formats. It should be mentioned that the use of the FLIR system to implement infrared imaging is not directly affected by light; hence, the headlights of nearby vehicles would have no impact on the images captured (the heat generated by lights is too low to affect the images).



Figure 1. Example thermal image capture.

#### 3.2. Data Collection

This subsection explains the data collection method and the data used for input in the artificial intelligence system. Images were collected by passing an automobile with the single FLIR Pro camera mounted onto the right-hand side mirror of the vehicle. The frame rate of the FLIR One Pro camera was 8.7 Hz per second, meaning that eight or nine frames were captured in one second. Images were taken from multiple camera angles to account for the variety of poses that a wild animal may present in relation to the vehicle's direction. The variety of angles contributes to the diversity of the data and thus improves the generalization ability of our model. The images were captured within the San Antonio metropolitan area in the state of Texas, United States. The thermal data was collected in a period from November 2020 to December 2020 and comprised images captured daily between 6 p.m. and 10 p.m. for two consecutive weeks. In general, the dataset consisted of a wide variety of pictures including occluded images, low-visibility images, long-distance images, blurred images, and multi-object images.

There were a few difficulties encountered while collecting the dataset used in this research. The data collected indicated general animal inactivity due to various stances such as rain and high-density fog. Furthermore, some of the local wildlife was in a natural state of hibernation. These difficulties led to a decrease in the number of images we were able to collect for use within the dataset.

The dataset was then filtered for images to be used in the system input. Images subject to human or equipment error were removed from the dataset, resulting in around 800 images with potential to be used for the proposed approach. However, further filtering was needed to accomplish the goal of this research. Images that included multiple objects, non-determinable poses, or no wildlife were removed from the dataset. Finally, the adopted dataset consisted of 182 prints. This dataset was then balanced through random selection to generate equal amounts of data for each corresponding pose. This led to 111 images remaining to be used for testing our artificial intelligence system. An overview of the data changes throughout implementation of these methods is provided in Table 1.

There were a few difficulties encountered while collecting the dataset used in this research. The data collected indicated general animal inactivity due to various circumstances such as rain and high-density fog. Furthermore, some of the local wildlife was in a natural state of hibernation. These difficulties led to a decrease in the number of images we were able to collect for use within the dataset. The dataset was then filtered for images to be used in the system input. Images subject to human or equipment error were removed from the dataset, resulting in around 800 images with potential to be used for the proposed approach. However, further filtering was needed to accomplish the goal of this research. Images that included multiple objects, non-determinable poses, or no wildlife were removed from the dataset. Finally, the adopted dataset consisted of 182 prints. This dataset was then balanced through random selection to generate equal amounts of data for each corresponding pose. This led to 111 images remaining to be used for testing our artificial intelligence system. An overview of the data changes throughout implementation of these methods is provided in Table 1.

The dataset was then filtered for images to be used in the system input. Images subject to human or equipment error were removed from the dataset, resulting in around 800 images with potential to be used for the proposed approach. However, further filtering was needed to accomplish the goal of this research. Images that included multiple objects, non-determinable poses, or no wildlife were removed from the dataset. Finally, the adopted dataset consisted of 182 prints. This dataset was then balanced through random selection to generate equal amounts of data for each corresponding pose. This led to 111 images remaining to be used for testing our artificial intelligence system. An overview of the data changes throughout implementation of these methods is provided in Table 1.

| Description                       | No. of Images |
|-----------------------------------|---------------|
| Total Images                      | 1000          |
| No. of images after error removal | 800           |
| No. of images after filtering     | 182           |

Table 1. An overview of changes of data throughout processing.

| Description  | No. of Images |
|--|---------------|
| Images with animals lying down                       | 37            |
| Images with wildlife facing toward automobile        | 45            |
| Images with wildlife facing away from automobile     | 100           |
| No. of images after filtering                        | 182           |
| Images with animals lying down                       | 37            |
| Images with wildlife facing toward automobile        | 45            |
| Images with wildlife facing away from automobile     | 100           |
| No. of images after balancing                        | 111           |
| Total number of inputs to network after augmentation | 222           |
| Images with wildlife facing away from automobile     | 100           |
| No. of images after balancing                        | 111           |
| Total number of inputs to network after augmentation | 222           |

Determination of the image set was followed by further processing of the data. Specifically, images were cropped and then resized to a smaller scale. After the images were cropped and resized, they were converted from the RGB heatmap image to a grayscale image, as shown in Figure 2. Following these changes, the resulting dataset was formed and supplemented by mirroring the previously processed images. The function stipulated in this work added a set of extra images by mirroring the images from left orientation to right. The result of this data augmentation was the doubling the dataset for use within the proposed artificial intelligence system. This step was performed because data augmentation techniques have been successfully employed to combat input variability and to allow convolutional neural networks to perform efficient deep learning [17]. The result was 222 different inputs for the proposed system, as shown in Table 1.

Determination of the image set was followed by further processing of the data. Specifically, images were cropped and then resized to a smaller scale. After the images were cropped and resized, they were converted from the RGB heatmap image to a grayscale image, as shown in Figure 2. Following these changes, the resulting dataset was formed and supplemented by mirroring the previously processed images. The function stipulated in this work added a set of extra images by mirroring the images from left orientation to right. The result of this data augmentation was the doubling the dataset for use within the proposed artificial intelligence system. This step was performed because data augmentation techniques have been successfully employed to combat input variability and to allow convolutional neural networks to perform efficient deep learning [17]. The result was 222 different inputs for the proposed system, as shown in Table 1.



Figure 2. Image resized, cropped, and converted to grayscale.



## 4. Methodology

### 4. Methodology

The following section presents and discusses the proposed artificial intelligence system for pose identification. The subsections are organized as follows: the convolutional neural network is described in Section 4.1; the methodology behind calculating the accuracy is provided in Section 4.2; Section 4.3 provides a complete overview of the system.

#### 4.1. 2D-CNN

Convolutional neural networks are generally considered to be one of the most influential innovations in the field of artificial intelligence for image classification and object detection [18–20]. In the proposed system, a processed thermal image is provided as the input, and then the features are extracted by the convolutional neural network. The block diagram of the system is depicted in Figure 3.

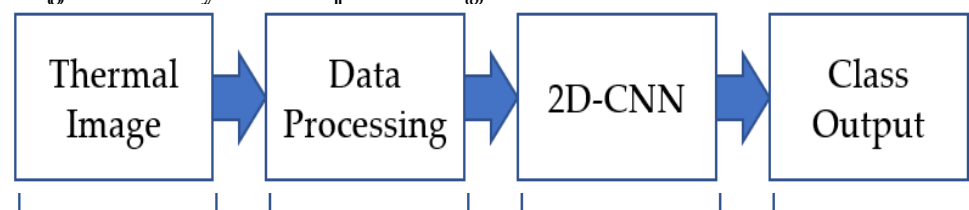


Figure 3. Generalized block diagram of the overall system.

**Figure 3.** Generalized block diagram of the overall system.

A convolutional neural network consists of multiple perceptron layers through which the extracted features of an image pass and are implicitly processed. This processing throughout the convolutional neural network is regarded as the deep learning process for image classification. Each convolutional layer extracts features that are subsequently passed to the next convolutional layer of the neural network. Many convolutional network paradigms exist, such as one-dimensional, two-dimensional, and three-dimensional paradigms [18–21]. In the proposed system, a two-dimensional convolutional network is utilized. The architecture of the CNN consists of filters, layer output, feature vector dimension, batch normalization, dropout, max-pooling, dense layers, and final production. A detailed overview of the 2D-CNN can be found in Figure 4.

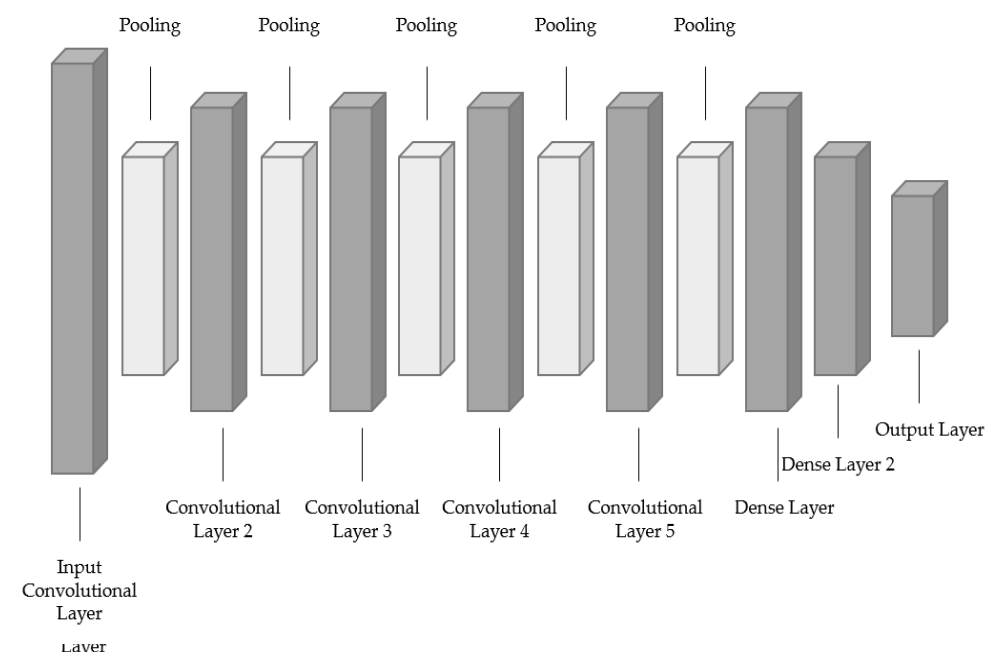


Figure 4. Overview of the 2D-CNN architecture.

**Figure 4.** Overview of the 2D-CNN architecture.

Within the architecture of the 2D-CNN itself, max pooling, batch normalization, and dropout are used. Pooling reduces the size of a feature map, which is beneficial as the

convolutional neural network is more susceptible to recognizing variance within an input [22,23]. Furthermore, without the use of pooling, the accuracy of deep learning models suffers severely. Batch normalization is a method of applying normalization to smaller batches of the layer output. This was carried out as previous research has shown that batch normalization tends to improve accuracy and speed during the training process of a convolutional neural network, by enhancing the neural network's stability [24,25]. Meanwhile, dropout is a method of breaking down a more extensive model by continuously sampling and training smaller sub-models, and has been shown to reduce model overfitting [26]. Given this reasoning, the addition of dropout was tested and implemented as its application was found to enhance model performance.

Multiple different activations were applied at different layers within the 2D-CNN. A Rectified Linear Unit (*ReLU*) activation was involved for each convolutional layer. *ReLU* is a fast-learning activation function and is one of the most successful and broadly used activation functions [27,28]. The *ReLU* function has also been shown to eliminate the issue of vanishing gradients [29]. The process works by returning the input element if it is greater than zero, or returning zero if the input variable is less than zero, as shown below:

*ReLU function:*

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (1)$$

Furthermore, the *SoftMax* activation function was applied to the output layer of the neural network. This is a function used in multi-classification models; its output is the probability of each class that is being categorized, and it returns the class with the highest likelihood. The *SoftMax* function is modeled [29] as follows:

*SoftMax function:*

$$f(x_i) = \frac{e^{x_i}}{\sum e^{x_j}} \quad (2)$$

Many different functions were applied to the model for quantifying the loss (i.e., error in classification) and to further ensure that the data could be processed throughout the model. To address the loss function, categorical cross entropy was used. Cross-entropy loss is a measure of the difference between the actual labels of an input and the predicted labels [30]. Due to the multi-classification nature of the 2D-CNN the categorical cross-entropy measure was adopted in the current research. The analytical form of the cross entropy is given below [30]:

*Cross entropy*

$$L(\hat{y}, y) = - \sum y_i \log(\hat{y}_i) \quad (3)$$

Finally, flattening was applied to the dense, fully connected layers of the CNN. Flattening is the method of converting the layers' output data to a one-dimensional output. This was adopted in our system, where we designed the layers of our network to process one-dimensional inputs.

#### 4.2. Accuracy Calculation

An explanation of the methodology behind the calculation of the network's accuracy is provided in this subsection. A confusion matrix was constructed to determine the accuracy of the model's tests. A confusion matrix is a table in which rows represent the actual classes and columns denote the predicted classes [31]. The true positive (T.P.), true negative (T.N.), false positive (F.P.), and false negative (F.N.) values from the confusion matrix were taken into account to determine the precision and accuracy of the corresponding models. With the multi-classification nature of the model, only true positives were considered in the accuracy calculation.

Given that the classification had three varying classes, each class had its corresponding actual positive value. The resulting equation for calculating the accuracy of the model is expressed as Equation (4) [32,33]:

Given that the classification had three varying classes, each class had its corresponding actual positive value. The resulting equation for calculating the accuracy of the model is expressed as Equation (4) [32,33]:

Accuracy Equation:

$$\text{Accuracy} = \frac{TP1+TP2+TP3}{\text{Total}} \quad (4)$$

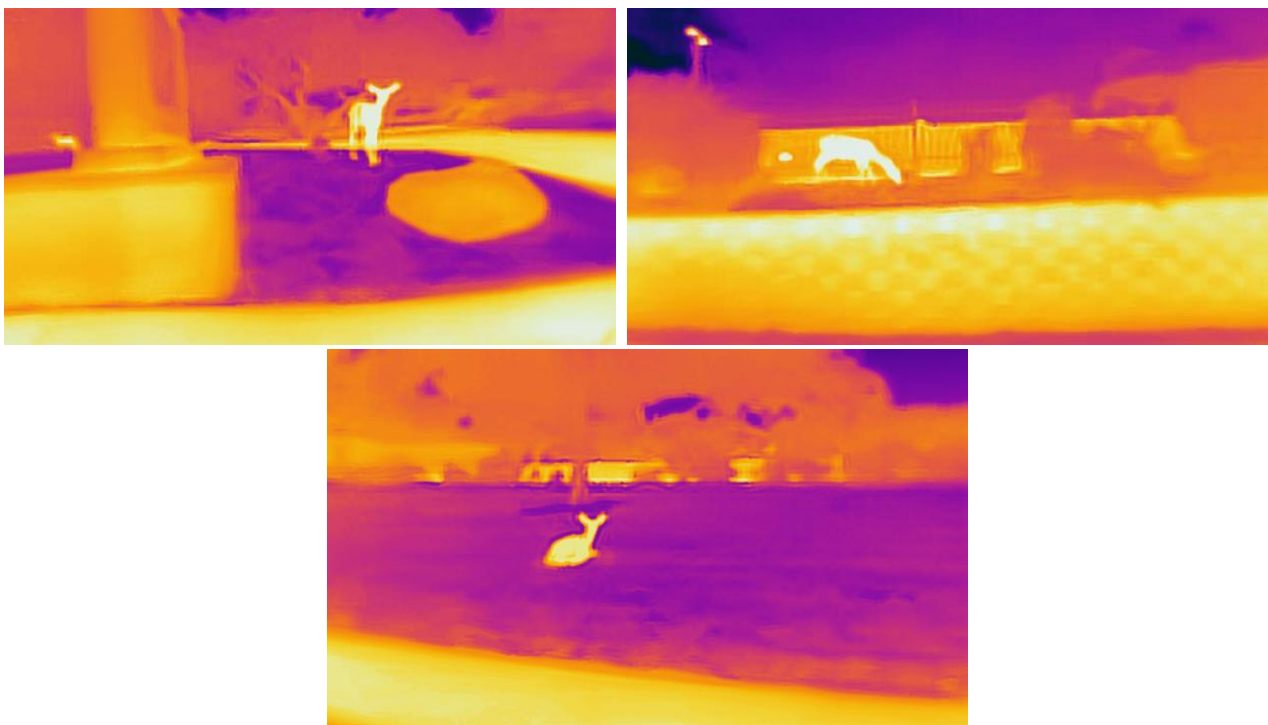
Accuracy Equation:

Overall, the confusion matrix allows the detection accuracy of the model to be calculated, and the model's misdetections to be explicitly detailed.

#### 4.3. General Overview

##### 4.3. General Overview

This section provides a concise explanation of the methodology behind the functionality of the proposed intelligent system. The system was encoded using python v3.9.7 [34], and for illustration purposes an example of the initial input to the system is given in Figure 5.



**Figure 5.** Example of thermal image input (Top left—towards the road, Top right—towards the side, bottom middle—sitting).

Initially, the thermal input images were acquired and were then labeled by being placed in labeled directories. The images then underwent conversion to grayscale, cropping, and resizing. The features were extracted, and all the components were appended into vector values. The mirrored and original vectors were combined into a single dataset, with the resulting size of the dataset being 222 input vectors for the system. The dataset was split into training and testing datasets, respectively comprising 80% and 20% of the total. Then, 20% of the training data was used for creating the validation set for our system. The created feature vectors were then input into the 2D-CNN [33], and subsequently applied for training across varying epochs, processed using the activation functions, filters, dropout, batch normalization, and max-pooling. Finally, the model's accuracy was determined through calculations using the measure formula shown in Equation (4). The list of steps using the image data that were followed to develop the system are summarized below:

1. Acquire data.
2. Categorize into folder directories.
3. Process images through resizing, cropping, and conversion to grayscale.

Initially, the thermal input images were acquired and were then labeled by being placed in labeled directories. The images then underwent conversion to grayscale, cropping, and resizing. The features were extracted, and all the components were appended into vector values. The mirrored and original vectors were combined into a single dataset, with the resulting size of the dataset being 222 input vectors for the system. The dataset was split into training and testing datasets, respectively comprising 80% and 20% of the total. Then, 20% of the training data was used for creating the validation set for our system. The created feature vectors were then input into the 2D-CNN [33], and subsequently applied for training across varying epochs, processed using the activation functions, filters, dropout, batch normalization, and max-pooling. Finally, the model's accuracy was determined through calculations using the measure formula shown in Equation (4). The list of steps using the image data that were followed to develop the system are summarized below:

4. Acquire vector values of images.
5. Mirror vectors and combine mirror dataset with the original dataset.
6. Divide the data into training and test datasets with an 80/20 split.
7. Use 20% of the training data for validation data.
8. Train the model with feature vector input to enhance performance through hyperparameters.
9. Run the model with test data and determine the accuracy of the model through the accuracy formula.

## 5. Results and Discussion

This section includes the results analysis, an outline of the solution to the problem, and a consideration of the impact of this study. The results section provides examples of inputs within each described system category. It includes a tabulation of the results, an accuracy graph, and a confusion matrix for training and testing the 2D-CNN.

### 5.1. Overall System Description

This section describes the combination of data methods and the artificial intelligence model, along with the system specifications. As mentioned in the introduction, the wildlife–vehicle collision cost consistently increases each year. To address this issue, a system to avoid automobile collisions with wildlife during the night hours is proposed in this study. Implementation of the design began with acquiring the thermal image data during nighttime hours. These collected thermal images were then processed and filtered into a dataset, with the wildlife in the thermal image labeled in their respective categories, such as facing away from the automobile, facing towards the automobile, and lying down. The images from this dataset were then turned into feature vectors that were forwarded through the 2-D CNN for training and testing the model.

The device used for the training, validation, and testing of the 2D-CNN was a Windows desktop computer with the following hardware specifications:

*Processor AMD Ryzen 9 3900x 12-Core Processor 3.79 GHz*

*Memory 16.0 GB 2666 MHz*

*Graphics NVIDIA GeForce RTX 3080 10GB*

### 5.2. CNN Model Parameters

This section is an overview of the parameters within the artificial 2D-CNN. The resulting parameters provided by the output of a line of source code execution are outlined in Figure 6. Figure 6 shows the neural network specifics acquired through usage of the model. Each row represents the corresponding layer type, including the number of convolutional layers, activation, batch normalization, flattening, max-pooling layers, dropout, and dense layers.

Every individual layer is presented together with the parameters detected in that layer. The same parameters were used in the training, validation, and testing of the 2D-CNN. It should be noted that the SoftMax function was adopted as the activation function in the output layer, as indicated at the foot of Figure 6 (i.e., above the parameter “count”). Furthermore, the figure also showcases the output shape of each layer of the neural network. The resulting trainable and non-trainable parameters are shown at the bottom of Figure 6.



| Layer (type)                                  | Output Shape          | Param # |
|---|-----------------------|---------|
| conv2d_475 (Conv2D)                           | (None, 118, 158, 255) | 2550    |
| activation_570 (Activation)                   | (None, 118, 158, 255) | 0       |
| max_pooling2d_475 (MaxPooling2D)              | (None, 59, 79, 255)   | 0       |
| conv2d_476 (Conv2D)                           | (None, 57, 77, 64)    | 146944  |
| activation_571 (Activation)                   | (None, 57, 77, 64)    | 0       |
| batch_normalization_380 (Batch Normalization) | (None, 57, 77, 64)    | 256     |
| max_pooling2d_476 (MaxPooling2D)              | (None, 28, 38, 64)    | 0       |
| conv2d_477 (Conv2D)                           | (None, 26, 36, 128)   | 73856   |
| activation_572 (Activation)                   | (None, 26, 36, 128)   | 0       |
| batch_normalization_381 (Batch Normalization) | (None, 26, 36, 128)   | 512     |
| max_pooling2d_477 (MaxPooling2D)              | (None, 13, 18, 128)   | 0       |
| conv2d_478 (Conv2D)                           | (None, 11, 16, 64)    | 73792   |
| activation_573 (Activation)                   | (None, 11, 16, 64)    | 0       |
| batch_normalization_382 (Batch Normalization) | (None, 11, 16, 64)    | 256     |
| max_pooling2d_478 (MaxPooling2D)              | (None, 5, 8, 64)      | 0       |
| conv2d_479 (Conv2D)                           | (None, 3, 6, 128)     | 73856   |
| activation_574 (Activation)                   | (None, 3, 6, 128)     | 0       |
| batch_normalization_383 (Batch Normalization) | (None, 3, 6, 128)     | 512     |
| max_pooling2d_479 (MaxPooling2D)              | (None, 1, 3, 128)     | 0       |
| flatten_95 (Flatten)                          | (None, 384)           | 0       |
| dropout_95 (Dropout)                          | (None, 384)           | 0       |
| dense_285 (Dense)                             | (None, 64)            | 24640   |
| dense_286 (Dense)                             | (None, 32)            | 2080    |
| dense_287 (Dense)                             | (None, 3)             | 99      |
| activation_575 (Activation)                   | (None, 3)             | 0       |
| Total params: 399,353                         |                       |         |
| Trainable params: 398,585                     |                       |         |
| Non-trainable params: 768                     |                       |         |

**Figure 6.** Overview of the model parameters.

### 5.3. Results Analysis

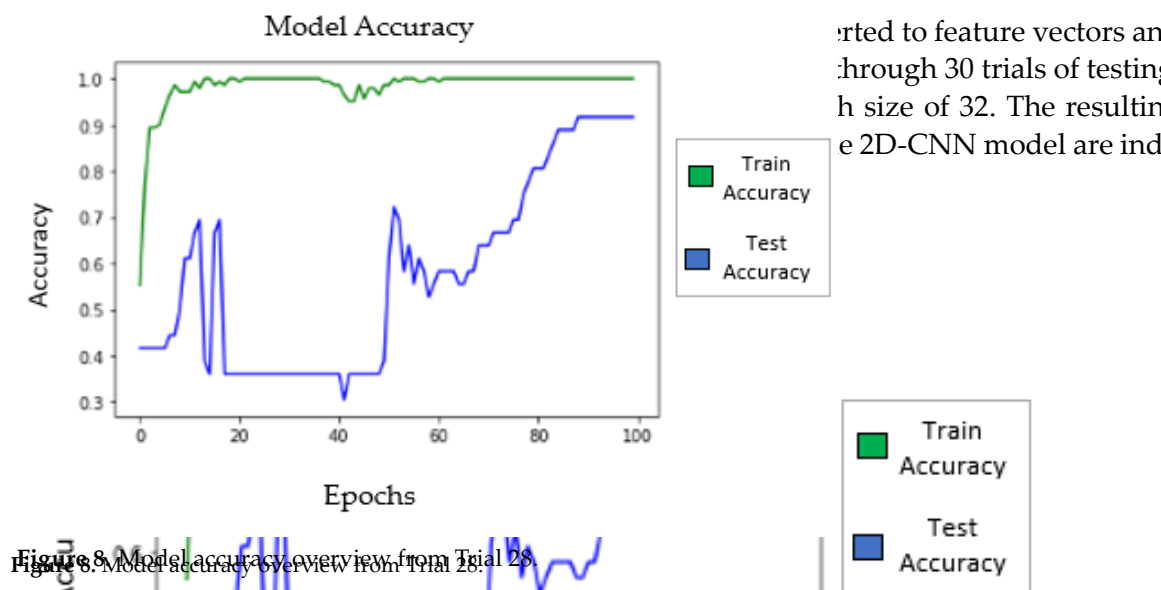
Every individual layer is presented together with the parameters detected in that layer. The results section provides an example of possible inputs to the system and a description of the outputs of the artificial intelligence model. Examples of an animal in a roadside scene from each of the respective categories of the dataset are shown in Figure 7. In the output layer, as indicated at the bottom of Figure 6 (i.e., above the parameter count). The resulting images pictured in Figure 7 were then converted to feature vectors and fed through the artificial intelligence system. The model went through 30 trials of testing, validating, and training, with 100 epochs per trial and a batch size of 32. The resulting accuracies from the training and testing dataset run through the 2D-CNN model are indicated in Figure 8.

The results section provides an example of possible inputs to the system and a description of the outputs of the artificial intelligence model. Examples of an animal in a roadside scene from each of the respective categories of the dataset are shown in Figure 7.



**Figure 7.** Examples of input images and their categories used for input in the 2D-CNN (left) Wildlife categorized in the facing away pose; (right) wildlife categorized in the facing towards pose (center) wildlife categorized in the lying down pose.

The resulting images pictured in Figure 7 were then converted to feature vectors and fed through the artificial intelligence system. The model went through 30 trials of testing, and the resulting images and their categories used for input in the 2D-CNN (left) Wildlife categorized in the facing away pose; (right) wildlife categorized in the facing towards pose (center) wildlife categorized in the lying down pose.



**Figure 8.** Model accuracy overview from Trial 28.

In Figure 8, the training and test accuracies are shown as percentage per epoch, where 1.0 stands for 100%. The model's training began from around 40% accuracy, then varied and dropped, but eventually increased to approximately 89% accuracy. The same pattern was followed throughout the 30 trials, with variations in accuracy. The representation in Table 2 presents the model's overall performance across the 30 different trials.

Table 2 presents the confusion matrix (i.e., the true positive values for each category) and the accuracy for each of the 30 different trials of the model, after 100 epochs per trial. True positive for category 1 is labeled TP1, true positive for category 2 is marked as TP2, and true positive for category 3 is labeled TP3. The total number of test inputs was 45. The figure was calculated using the formula 28 and rounded to the nearest whole number, shown in the final column. A breakdown of an individual confusion matrix from trial 30 is provided in Figure 9.

**Table 2.** Confusion matrix and accuracy per trial run of the model.

| Trial No. | Confusion Matrix                     | TP 1 | TP 2 | TP3 | Accuracy |
|-----------|--------------------------------------|------|------|-----|----------|
| 1         | [[15 3 1]<br>[ 0 15 0]<br>[ 1 0 10]] | 15   | 15   | 10  | 89%      |
| 2         | [[15 4 0]<br>[ 0 15 0]<br>[ 1 1 9]]  | 15   | 15   | 9   | 87%      |
| 3         | [[19 0 0]<br>[ 6 9 0]<br>[ 5 1 5]]   | 19   | 9    | 5   | 73%      |
| 4         | [[18 1 0]<br>[ 0 15 0]<br>[ 0 4 7]]  | 18   | 15   | 4   | 89%      |
| 5         | [[16 3 0]<br>[ 0 15 0]<br>[ 2 1 8]]  | 16   | 15   | 8   | 87%      |
| 6         | [[13 6 0]<br>[ 0 15 0]<br>[ 0 4 7]]  | 13   | 15   | 6   | 78%      |
| 7         | [[14 5 0]<br>[ 0 15 0]<br>[ 1 2 8]]  | 14   | 15   | 8   | 82%      |
| 8         | [[13 6 0]<br>[ 0 15 0]<br>[ 1 2 8]]  | 13   | 15   | 8   | 80%      |
| 9         | [[14 5 0]<br>[ 0 15 0]<br>[ 0 4 7]]  | 14   | 15   | 7   | 80%      |
| 10        | [[14 5 0]<br>[ 0 15 0]<br>[ 1 3 7]]  | 14   | 15   | 3   | 80%      |
| 11        | [[14 5 0]<br>[ 0 15 0]<br>[ 0 2 9]]  | 14   | 15   | 9   | 84%      |
| 12        | [[15 4 0]<br>[ 0 15 0]<br>[ 1 1 9]]  | 15   | 15   | 9   | 87%      |
| 13        | [[15 4 0]<br>[ 0 15 0]<br>[ 0 2 9]]  | 15   | 4    | 15  | 87%      |
| 14        | [[13 6 0]<br>[ 1 14 0]<br>[ 1 3 7]]  | 13   | 14   | 7   | 76%      |
| 15        | [[15 4 0]<br>[ 0 15 0]<br>[ 1 3 7]]  | 15   | 15   | 7   | 82%      |
| 16        | [[15 0 4]<br>[ 0 4 11]<br>[ 1 0 10]] | 15   | 4    | 10  | 64%      |
| 17        | [[15 4 0]<br>[ 0 15 0]<br>[ 1 2 8]]  | 15   | 15   | 8   | 84%      |

Table 2. Cont.

| Trial No. | Confusion Matrix                     | TP 1 | TP 2 | TP3 | Accuracy |
|-----------|--------------------------------------|------|------|-----|----------|
| 18        | [[14 5 0]<br>[ 0 15 0]<br>[ 0 1 10]] | 15   | 15   | 10  | 87%      |
| 19        | [[15 4 0]<br>[ 0 15 0]<br>[ 0 2 9]]  | 15   | 15   | 9   | 87%      |
| 20        | [[15 3 1]<br>[ 0 15 0]<br>[ 2 3 6]]  | 15   | 15   | 6   | 80%      |
| 21        | [[11 8 0]<br>[ 0 15 0]<br>[ 1 6 4]]  | 11   | 15   | 8   | 67%      |
| 22        | [[14 4 1]<br>[ 0 14 1]<br>[ 0 3 8]]  | 14   | 14   | 8   | 80%      |
| 23        | [[15 3 1]<br>[ 0 15 0]<br>[ 2 1 8]]  | 15   | 15   | 8   | 84%      |
| 24        | [[14 5 0]<br>[ 4 11 0]<br>[ 2 1 8]]  | 14   | 11   | 8   | 73%      |
| 25        | [[14 5 0]<br>[ 0 15 0]<br>[ 0 4 7]]  | 14   | 15   | 7   | 80%      |
| 26        | [[11 8 0]<br>[ 0 15 0]<br>[ 1 3 7]]  | 11   | 15   | 7   | 73%      |
| 27        | [[15 4 0]<br>[ 0 15 0]<br>[ 1 3 7]]  | 15   | 15   | 7   | 82%      |
| 28        | [[15 4 0]<br>[ 0 15 0]<br>[ 0 2 9]]  | 15   | 15   | 9   | 87%      |
| 29        | [[15 4 0]<br>[ 0 15 0]<br>[ 0 1 10]] | 15   | 15   | 10  | 89%      |
| 30        | [[15 3 1]<br>[ 1 14 0]<br>[ 0 0 11]] | 15   | 14   | 11  | 89%      |

Figure 9 presents the overall picture of an individual trial's confusion matrix. The highlighted boxes in gray represent the actual positive values for each category. For example, the exact positive count for animals classified as lying down was 15, the actual positive count for animals classified in the facing away category was 15 also, and the exact positive count for animals in the facing toward classification was 11. To determine the accuracy, the sum of these values was then calculated and divided by 45, which is the total count of the values within the confusion matrix. Overall, the accuracy of this trial was estimated to be 89%. The number of total missed classifications is represented by the sum of the values that are not considered valid positive values for the respective categories. In this case, the resulting number of misdetected values was calculated to be five.



Table 2 presents the confusion matrix (i.e., the true positive values for each category) and the accuracy for each of the 30 different trials of the model, after 100 epochs per trial. True positive for category 1 is labeled TP1, true positive for category 2 is marked as TP2, and true positive for category 3 is labeled TP3. The total number of test inputs was 45. The test accuracy was calculated using the formula and rounded to the nearest whole number, shown in the final column. A breakdown of an individual confusion matrix from trial 30 is provided in Figure 9.

|               |                | Predicted Values |             |                |
|---------------|----------------|------------------|-------------|----------------|
|               |                | Laying Down      | Facing Away | Facing Towards |
| Actual Values | Laying Down    | 15               | 3           | 1              |
|               | Facing Away    | 1                | 14          | 0              |
|               | Facing Towards | 0                | 0           | 11             |

**Figure 9.** Breakdown of Trial 30's confusion matrix.

Figure 9 presents the overall picture of an individual trial's confusion matrix. The highlighted boxes in gray represent the actual positive values for each category. For example, the exact positive count for animals classified as lying down was 15, the actual positive count for animals classified in the facing away category was 14 also, and the exact positive count for animals in the facing toward classification was 11. To determine the process used within the *training* and *test* datasets. The lower accuracy can be explained by the randomly selected dataset, which contained inadequate numbers of images in individual classes for the system to learn efficiently and identify different image classes. Furthermore, the model achieved an average of 82% accuracy based on the provided test data, while the maximum accuracy attained within the set of 30 trials was 89%. Overall, the results showcase the model's ability to successfully classify animal poses from thermal images of roadside scenes.

Overall, the deep learning model achieved maximum accuracy of 89%, confirmed by testing the data as per Table 3. The minimum accuracy for the model was 64%. Generally, the trials that performed poorly overall did so as a result of the random data selection process used within the *training* and *test* datasets. The lower accuracy can be explained by the randomly selected dataset, which contained inadequate numbers of images in individual classes for the system to learn efficiently and identify different image classes. Furthermore, the model achieved an average of 82% accuracy based on the provided test data, while the maximum accuracy attained within the set of 30 trials was 89%. Overall, the results showcase the model's ability to successfully classify animal poses from thermal images of roadside scenes.

Table 3. Summary of Model Performance for Testing Accuracy.

|         |     |
|---------|-----|
| Average | 82% |
| Maximum | 89% |
| Minimum | 64% |

## 6. Conclusions

The AI system presented in this article can be utilized to classify an animal pose from a thermal image to determine the risk an animal presents to a passing automobile during nighttime hours. The implementation of the system involves acquiring thermal images of roadside scenes containing wildlife, and processing them through an artificial intelligence model to classify the pose of the animal in the roadside image. The study used a two-dimensional convolutional neural network to organize accurately and efficiently the animals' poses from the acquired thermal images. The use of thermography within the system allowed the detection of animal poses during nighttime hours, even with limited visibility during these times. The overall methodology involved the collection of thermal images along with their processing and filtering into a novel dataset. These images were fed through the two-dimensional convolutional neural network to garner the results. Overall, the results from the developed system suggest that artificial intelligence methods can successfully determine animal poses from thermal images of wildlife in roadside scenes. The artificial intelligence method enables systems to consider whether animals threaten to enter the roadway in front of a passing automobile, based on their pose. Their pose

determines whether they have the potential to run into the vehicle, or be spurred into motion from a standing position and come into contact with the vehicle, or present no threat at all if they are lying down. Overall, the potential of a crash is greatly reduced as the determination of the possibilities of an animal's actions allows preemptive warnings about oncoming wildlife to be sent to the driver, who will be alert to mitigate the possibility of a crash. In a potential application of the system, alerts can be sent to the driver as a warning display or issued as an audio cue that the vehicle plays to make the driver aware of the potential threat posed by an animal on the side of the road [34]. Furthermore, the application of this methodology is potentially immense as the system provides a precursor to creating further automated safe nocturnal driving methods to avoid wildlife-to-vehicle collisions. This article also suggests how thermal imaging and artificial intelligence methods can be used successfully to identify nighttime driving risks.

Furthermore, the model's results showcase the accuracy of animal pose classification from roadside scenes, allowing an assessment of the amount of risk an animal presents to a passing automobile. In a specific context, engineers at the forefront of creating advanced artificial intelligence methods for automated driving could find these results helpful in their efforts to address prominent safety concerns. More generally, a warning system built into automobiles could be applied alongside this proposed system to alert the driver if an animal threatens to collide with the vehicle. In summary, the proposed approach reveals the potential of the coupling of thermal imagery and artificial intelligence methods for classifying animal poses and mitigating risks associated with nighttime driving.

A few limitations of the study allow the possibility of future improvements to the system. The first and most prominent limitation was the size of this novel dataset. The data acquisition process can be repeated to provide more data for training the deep learning model. Furthermore, multiple animals in different poses were not considered in this dataset and within the classification. This allows the research to be extended and improved upon by adding the capability to classify every individual animal within a roadside scene. Moreover, the research only considered data collected within the San Antonio, Texas area in the United States, and future work should incorporate roadside scenes from rural non-city regions. In summary, future work should be undertaken to increase the size of the dataset, to consider multiple animals in different poses, and to incorporate rural data to address the study's limitations.

**Author Contributions:** Formal analysis, D.M.; Funding acquisition, M.A.; Methodology, Y.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was part of the REU program at UTSA funded under NSF award #2051113.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available upon request.

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

## References

- Huijser, M.P.; McGowan, P.; Hardy, A.; Kociolek, A.; Clevenger, A.P.; Smith, D.; Ament, R. *Wildlife-Vehicle Collision Reduction Study: Report to Congress*; Federal Highway Administration: McLean, VA, USA, 2017.
- Seamans, T.W.; David, H.A. Evaluation of an electrified mat as a white-tailed deer (*Odocoileus virginianus*) barrier. *Int. J. Pest Manag.* **2008**, *54*, 89–94. [[CrossRef](#)]
- Stein, L. Oh, Deer! *U.S. News World Rep.* **2003**, *135*, 19.
- Pagany, R. Wildlife-vehicle collisions-Influencing factors, data collection, and research methods. *Biol. Conserv.* **2020**, *251*, 108758. [[CrossRef](#)]
- Barnes, R.B. Thermography, thermography & its clinical applications. *Ann. N. Y. Acad. Sci.* **1964**, *121*, 34–48. [[PubMed](#)]
- McCafferty, D.J. The value of infrared thermography for research on mammals: Previous applications and future directions. *Mammal. Rev.* **2007**, *37*, 207–223. [[CrossRef](#)]
- Lloyd, J.M. *Thermal Imaging Systems*; Springer Science & Business Media: Berlin, Germany, 2013.
- Cilulko, J.; Janiszewski, P.; Bogdaszewski, M.; Szczygielska, E. Infrared thermal imaging in studies of wild animals. *Eur. J. Wildl. Res.* **2013**, *59*, 17–23. [[CrossRef](#)]

9. Hollnagel, E.; Källhammer, J.E. Effects of a night vision enhancement system (NFS) on driving: Results from a simulator study. In Proceedings of the Driving Assessment Conference, Park City, UT, USA, 21–24 July 2003; Volume 2.
10. Albani, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6. [\[CrossRef\]](#)
11. Xin, M.; Wang, Y. Research on image classification model based on deep convolution neural network. *J. Image Video Proc.* **2019**, *2019*, 40. [\[CrossRef\]](#)
12. Sharma, S.U.; Shah, D.J. A practical animal detection and collision avoidance system using computer vision technique. *IEEE Access* **2016**, *5*, 347–358. [\[CrossRef\]](#)
13. Bhatia, Y.; Rai, R.; Gupta, V.; Aggarwal, N.; Akula, A. Convolutional neural networks-based potholes detection using thermal imaging. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *34*, 578–588.
14. Munian, Y.; Martinez-Molina, A.; Alamaniotis, A. Intelligent system for detecting wild animals using HOG and CNN in automobile applications. In Proceedings of the 2020 11th International Conference on Information, Intelligence, Systems, and Applications (IISA), Piraeus, Greece, 15–17 July 2020.
15. Wagner, R.; Thom, M.; Gabb, M.; Limmer, M.; Schweiger, R.; Rothmel, A. Convolutional neural networks for nighttime animal orientation estimation. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (iv), Gold Coast, QLD, Australia, 23–26 June 2013; pp. 316–321.
16. DeVault, T.L.; Seamans, T.W.; Blackwell, B.F. Frontal vehicle illumination via rear-facing lighting reduces potential for collisions with white-tailed deer. *Ecosphere* **2020**, *11*, e03187. [\[CrossRef\]](#)
17. Hernández-García, A.; König, P. Further advantages of data augmentation on convolutional neural networks. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 95–103.
18. Gomez, A.; Diez, G.; Salazar, A.; Diaz, A. Animal Identification in Low-Quality Camera-Trap Images Using Very Deep Convolutional Neural Networks and Confidence Thresholds. In Proceedings of the Advances in Visual Computing: 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, 12–14 December 2016; Springer International Publishing: New York, NY, USA, 2016; pp. 747–756.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
20. Zilkha, M.; Spanier, A. Real-time CNN-based object detection and classification for outdoor surveillance images: Daytime and thermal. In Proceedings of the Artificial Intelligence and Machine Learning in Defense Applications, Strasbourg, France, 19 September 2019; Volume 11169, p. 1116902.
21. Kim, D.; Kwon, D.S. Pedestrian detection and tracking in thermal images using shape features. In Proceedings of the 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Goyangi, Korea, 28–30 October 2015; pp. 22–25.
22. Akhtar, N.; Ragavendran, U. Interpretation of intelligence in CNN-pooling processes: A methodological survey. *Neural Comput. Appl.* **2020**, *32*, 879–898. [\[CrossRef\]](#)
23. Munian, Y.; Martinez-Molina, A.; Alamaniotis, M. Design and implementation of a nocturnal animal detection intelligent system in Automobile Applications. In *International Conference on Transportation and Development 2021—Transportation Operations Technologies and Safety*; American Society of Civil Engineers (ASCE): Reston, VA, USA, 2021; pp. 438–449. [\[CrossRef\]](#)
24. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How does batch normalization help optimization? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 2–8 December 2018.
25. Bjorck, N.; Gomes, C.P.; Selman, B.; Weinberger, K.Q. Understanding batch normalization. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 2–8 December 2018.
26. Srivastava, N. Improving Neural Networks with Dropout. Master's Thesis, The University of Toronto, Toronto, ON, Canada, 2013.
27. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for Activation Functions. *arXiv* **2017**, arXiv:1710.05941. Available online: <http://arxiv.org/abs/1710.05941> (accessed on 2 July 2022).
28. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
29. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
30. Koidl, K. *Loss Functions in Classification Tasks*; School of Computer Science and Statistic Trinity College: Dublin, Ireland, 2013.
31. Haghighi, S.; Jasemi, M.; Hessabi, S.; Zolanvari, A. PCM: Multiclass confusion matrix library in Python. *J. Open-Source Softw.* **2018**, *3*, 729. [\[CrossRef\]](#)
32. Munian, Y.; Martinez-Molina, A.; Miserlis, D.; Hernandez, H.; Alamaniotis, M. Intelligent System Utilizing HOG and CNN for Thermal Image-Based Detection of Wild Animals in Nocturnal Periods for Vehicle Safety. *Appl. Artif. Intell.* **2022**, *36*, 2031825. [\[CrossRef\]](#)
33. Munian, Y.; Martinez-Molina, A.; Alamaniotis, M. Active Advanced Arousal System to Alert and Avoid the Crepuscular Animal Based Vehicle Collision. *Intell. Decis. Technol.* **2021**, *15*, 707–720. [\[CrossRef\]](#)
34. Python 3.9.7. 2021. Available online: <https://www.python.org/downloads/release/python-397/> (accessed on 25 June 2022).