Convolutional Neural Network for dog cardiomegaly assessment

Dushyanth Reddy Pisati
Yeshiva University
dpisati@mail.yu.edu

Abstract

Cardiomegaly is a serious heart disease in dogs that can lead to life-threatening complications if not detected early. Timely detection plays a crucial role in improving treatment outcomes. In this study, we introduce a custom convolutional neural network (CNN) model designed to classify canine heart conditions into three categories: small, normal, and large [1]. The model consists of four convolutional layers and three fully connected layers, trained on a well-structured dataset comprising training, validation, and test sets. In the preprocessing step, I performed resizing and normalization of the images to standardize the input data and ensure consistency. Resizing ensures that all images are of uniform dimensions, while normalization scales pixel values to a range that improves model training stability. Our approach achieves a 70.75% classification accuracy on the test dataset, demonstrating its potential as a reliable tool for automated cardiomegaly assessment. This model is simple, lightweight, and uses cross-entropy loss and the Adam optimizer in the training process, which ensures fast convergence and efficient learning. By integrating deep learning into veterinary diagnostics, this research aims to enhance early detection methods, ultimately supporting more effective clinical decision-making and better patient care for dogs.

1. Introduction

In dogs, cardiomegaly, or heart enlargement, is a strong indicator of underlying cardiac disorders such dilated cardiomyopathy (DCM) and mitral valve disease (MVD) [2]. Cardiomegaly must be identified early in order to avoid serious consequences including heart failure and sudden cardiac arrest. A common method for determining heart size is the Vertebral Heart Score (VHS) [5], which compares the length of the vertebrae in thoracic radiography to the size of the heart. There is a need for more effective and precise diagnostic techniques because, although being a standardized and systematic approach, VHS is still time-consuming, subjective, and vulnerable to inter-observer variability.

With the advancement of deep learning in medical imaging, convolutional neural networks (CNNs) have emerged as a transformative tool. These AI-powered models provide automatic, precise, and consistent evaluations, making them highly effective for various medical imaging applications. CNNs have already shown promise in veterinary medicine, such as in hip dysplasia classification [4], and have been successfully utilized in human radiology and dermatology. When applied to canine cardiology, deep learning algorithms can analyze thoracic X-rays to detect cardiomegaly, eliminating the need for manual VHS computations and minimizing human interpretation errors.

By incorporating AI technologies into veterinary clinics, veterinarians can greatly improve early intervention efforts, resulting in faster and more accurate diagnosis. This method not only improves diagnosis accuracy, but it also boosts workflow efficiency and allows doctors to make more informed judgments [6]. Finally, using AI-driven solutions in canine cardiology can improve healthcare outcomes for dogs by ensuring they receive early and appropriate therapy.

2. Related Work

Thoracic radiographs are frequently used to diagnose cardiac conditions in dogs, with the traditional VHS score calculated by measuring the short and long axes of the heart and dividing the total by the vertebral length. Jeong and Sung [3] developed a new deep learning-based radiographic metric, the ”adjusted heart volume index” (aHVI), which calculates heart size in dogs using retrospective data to assist in diagnosing heart diseases. However, these CNN models do not directly account for the calculation of VHS. Li and Zhang [1] introduced a regressive vision transformer (RVT) model for classifying canine cardiomegaly, which extends beyond radiograph X-rays and can also be applied to other medical imaging modalities, such as CT scans and ultrasound. This model can potentially be adapted to detect human cardiomegaly as well, using various diagnostic technologies.

3. Methods

3.1. Datasets

The data set used in this study comprises thoracic radiographs of dog hearts classified into three groups: normal, small, and large, to aid in the diagnosis of canine cardiomegaly. It is separated into 1,400 images for training, 200 images for validating and 400 images for testing to ensure an even learning and evaluation procedure. The dataset contains photos of differing dimensions and resolutions, representing various kinds and sizes of dogs, which improves the model's capacity to generalize across different scenarios. To improve model performance and generalization, data augmentation techniques like rotation, flipping, and contrast modifications were used, as well as scaling and resizing to standardize input dimensions.

3.2. Data Preprocessing

- **Resizing**: All images were resized to **224×224 pixels** to maintain uniform input dimensions.
- **Grayscale Conversion**: The images were converted to **grayscale** to reduce computational complexity.
- **Data Augmentation**: Applied **rotation** and **horizontal flipping** to improve the variability of the data set and improve the generalization of the model.
- **Normalization**: Used **Min-Max scaling** to normalize pixel values from **0-255 to 0-1**, ensuring stable training and faster convergence.

3.3. Model Architecture

The model architecture for classifying dog cardiomegaly includes the following layers: Convolutional layers, batch normalization, ReLU activation, Max pooling, Fully connected layers, and the output layer with 3 neurons for classification.

3.3.1 Convolutional Layers

The model begins with multiple convolutional layers using a 3×3 kernel (stride = 1, padding = 1), progressively increasing the output channels from 1 to 128. This allows the model to extract more complex features from the canine heart images as it progresses through the layers.

3.3.2 Batch Normalization

Batch normalization is applied after each convolutional layer to stabilize and accelerate the training process, improving convergence and helping the model generalize better for the canine cardiomegaly classification task.

3.3.3 Activation Function (ReLU)

ReLU activation introduces non-linearity into the network, enabling the model to learn complex patterns in the data. It also helps avoid the problem of the vanishing gradient and improves learning efficiency during backpropagation.

3.3.4 Pooling Layers

Max pooling layers with a kernel 2×2 and stride = 2 reduce the spatial dimensions of feature maps while retaining essential information, thus improving the model's ability to generalize and reducing computational load.

3.3.5 Fully Connected Layers

After the convolutional and pooling layers, the feature maps are flattened and passed through two fully connected layers with 256 and 128 neurons, respectively. ReLU activation is applied, along with a dropout rate of $p = 0.5$ to prevent overfitting and improve classification performance.

3.3.6 Output Layer

The final output layer consists of 3 neurons, corresponding to the three classification classes. This layer produces the predicted class probabilities based on the features extracted from the previous layers.

3.3.7 Summary of Architecture

The complete model architecture used for canine cardiomegaly classification is summarized below. Each layer plays a role in extracting and transforming features from the input images, eventually producing class predictions through the final output layer.

Layer	Filters / Units	Output Shape
Conv1	32	224×224×32
BN1 + ReLU + MaxPool	-	112×112×32
Conv2	64	112×112×64
BN2 + ReLU + MaxPool	-	56×56×64
Conv3	128	56×56×128
BN3 + ReLU + MaxPool	-	28×28×128
Conv4	128	28×28×128
BN4 + ReLU + MaxPool	-	14×14×128
Flatten	-	25088
FC1 + ReLU + Dropout	256	256
FC2 + ReLU + Dropout	128	128
Output	3	3

Table 3.3.7: Custom CNN architecture used for canine cardiomegaly classification.

4. Training and Hyperparameters

The training process was designed to ensure efficient learning and generalization. The model was trained with the following configuration:

- **Batch Size:** A batch size of 64 was used for both training and validation phases. Shuffling was enabled for training data to promote randomness, while it was disabled for validation to ensure consistent evaluation.

- **Epochs:** The model was trained for 100 epochs to ensure sufficient exposure to the training data.

- **Loss Function:** Cross-Entropy Loss (`nn.CrossEntropyLoss()`) was used to measure the prediction error for multi-class classification. It is defined as:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

  where C is the number of classes, y_i is the ground truth label (one-hot encoded), and $\hat{y}_i$ is the predicted probability for class i.

- **Optimizer:** The Adam optimizer (`optim.Adam`) was used to update model parameters, with a learning rate of 0.001.

- **Weight Decay:** A regularization parameter of 1×10^{-4} was applied to prevent overfitting by penalizing large weights.

These hyperparameters were selected to balance convergence speed and generalization performance during training.

5. Validation and Testing

5.1. Validation Strategy

During training, the model was evaluated on a separate validation set at the end of each epoch using average loss and accuracy to monitor training process. The best-performing models were saved based on the highest validation accuracy and lowest validation loss, along with the final model at epoch 100. This approach helped prevent overfitting and ensured reliable performance on unseen data. Training and validation metrics were recorded throughout for performance tracking and visualization.

5.2. Testing and Evaluation

For testing, the model that achieved the highest validation accuracy was used predict class labels for unseen test images. Each image in the test dataset was passed through the model individually, and the predicted class (the one with the highest softmax probability) was recorded. The test predictions were saved in a CSV file that contains the filename and the corresponding predicted class.

The resulting CSV file was then evaluated using an official tool provided in Zhang's GitHub repository, which computes the final classification accuracy by comparing the predicted labels with the ground truth.

6. Results and Analysis

6.1. Performance Metrics

The custom CNN model achieved a test accuracy of 70.75%, demonstrating competitive performance for the classification task. During training, the model progressively improved over 100 epochs, with consistent improvements in both training and validation metrics. The best validation accuracy recorded was 76%, and the lowest validation loss observed was 0.6554. A summary of the key metrics is provided in Table 1.

Metric	**Value**
Test Accuracy	70.75%
Best Validation Accuracy	76%
Best Validation Loss	0.6554
Final Training Loss	0.1919
Final Validation Loss	1.2554

Table 1: Metrics of the CNN Model

6.2. Accuracy Trends

Figure 1 shows the training and validation accuracy across epochs. The model started with an initial accuracy of around 40% and gradually improved as training progressed. While the training accuracy reached over 91.98%, the validation accuracy stabilized around 73% toward the final epochs. The gap between training and validation performance remained relatively small for some epochs; however, the validation accuracy began to decrease afterward. The model was selected at the point where the validation accuracy had stabilized.

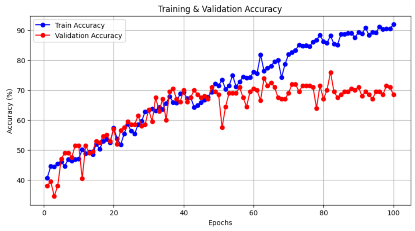

Figure 1: Training and Validation Accuracy over 100 Epochs

6.3. Loss Trends

The loss curves shown in Figure 2 illustrate the decline in both training and validation loss throughout the training process. Starting from higher values, the training loss steadily decreased to 0.19 by the end. The validation loss, while showing some fluctuations, generally trended downward initially but began to rise toward the later epochs, which may indicate the onset of overfitting.

Figure 2: Training and Validation Loss over 100 Epochs

7. Discussion

The custom CNN model developed in this study achieved a test accuracy of 70.75%, which is close to the commonly accepted benchmark of 75% for this task. Although slightly below the threshold, the model demonstrates strong learning capability, especially considering its relatively simple architecture. When compared to deeper architectures like VGG16, which typically achieve slightly higher accuracy (e.g., around 74–75%), the performance gap remains modest.

One key advantage of the proposed model is its efficiency. VGG16, with its 16 layers and large number of parameters, generally requires more computational resources and longer training times. In contrast, the custom model is lightweight, faster to train, and less prone to overfitting due to having fewer parameters. This makes it a more practical choice in environments with limited GPU memory or time constraints.

While the model's performance is already competitive, there is still room for improvement. Techniques such as learning rate scheduling, data augmentation, and regularization could be explored to further enhance accuracy. Overall, the results show that a carefully designed shallow CNN can offer a good balance between performance and efficiency, and can serve as a strong alternative to larger pretrained models in resource-constrained settings.

References

[1] Jialu Li and Youshan Zhang. *Regressive Vision Transformer for Dog Cardiomegaly Assessment*. Scientific Reports, 14(1):1539, 2024. 1

[2] VCA Animal Hospitals. *Heart Disease in Dogs*. Available at: `https://vcahospitals.com/know-your-pet/heart-disease-in-dogs`. Accessed March 25, 2025. 1

[3] Yeojin Jeong and Joohon Sung. *An Automated Deep Learning Method and Novel Cardiac Index to Detect Canine Cardiomegaly from Simple Radiography*. Scientific Reports, 12(1):14494, 2022. 1

[4] Leonie Barrett *et al*. *Deep Learning for Radiographic Canine Hip Dysplasia Classification*. Veterinary Radiology and Ultrasound, 61(5):554–562, 2020. 1

[5] Mengni Zhang, Kai Zhang, Deying Yu, Qianru Xie, Binlong Liu, Dacan Chen, Dongxing Xv, Zhiwei Li, and Chaofei Liu. *Computerized Assisted Evaluation System for Canine Cardiomegaly via Key Points Detection with Deep Learning*. Preventive Veterinary Medicine, 193:105399, 2021. 1

[6] Jun-Young Oh, In-Gyu Lee, Young-Min Go, Euijong Lee, and Ji-Hoon Jeong. *Leveraging Image Classification and Semantic Segmentation for Robust Cardiomegaly Diagnosis in Pet*. J. Pap. Korean Soc. Inf. Technol, 21:143–152, 2023. 1