Dog Heart Classification

Aditya Singh Parmar
Yeshiva University NY
aparmar1@mail.yu.edu

Abstract

This paper explores the advanced application of Convolutional Neural Networks (CNN's) in classifying dog heart X-ray images. Utilizing a comprehensive dataset categorized into Small, Normal, and Large classes, our research aims to pioneer novel methodologies for enhancing image classification accuracy in veterinary diagnostics. Drawing inspiration from recent advances in medical imaging, we applied data augmentation techniques such as resizing, random horizontal flipping, rotation, and color jittering to enrich the training dataset. A custom CNN architecture was developed using PyTorch, featuring three convolutional layers and fully connected layers with dropout regularization to mitigate over-fitting. Our iterative training process focused on fine-tuning the model to achieve optimal performance. The model's efficacy was validated using a dedicated validation set, and the best-performing model was subsequently tested on an unlabeled test dataset. The study aims to push the boundaries of current CNN applications in medical image classification, demonstrating significant potential for improving diagnostic accuracy. The datasets used for this research provided a robust foundation for our experiments, enabling substantial advancements in model performance. This investigation contributes to the broader discourse on medical image analysis by proposing scalable solutions for CNN-based diagnostic tools. The complete source code and supplementary materials can be accessed via our GitHub repository.

1. Introduction

Cardiac diseases are a leading cause of mortality among pet dogs, necessitating effective diagnostic tools for early intervention. Radiographic analysis, particularly through X-ray imaging, plays a crucial role in identifying cardiomegaly—a common indicator of cardiac pathology. However, accurate and timely diagnosis can be challenging, often requiring specialized training in veterinary radiology. This study addresses these challenges by proposing a computer-aided diagnosis (CAD) method leveraging Convolutional Neural Networks (CNNs) to automate the classification of dog heart X-ray images into categories of Small, Normal, and Large.

Recent advancements in CNNs have revolutionized medical image analysis by enabling automated feature extraction and classification. Our research builds upon these advancements, integrating advanced data augmentation techniques to enhance the diversity and robustness of our dataset. Techniques such as resizing, random horizontal flipping, rotation, and color jittering simulate various conditions, improving the model's generalization capability. Additionally, a custom CNN architecture tailored to the specific characteristics of dog heart X-ray images is designed, comprising convolutional and fully connected layers with dropout regularization to prevent overfitting.

Through meticulous training and validation processes, we fine-tuned our model for optimal performance. Validation on a separate dataset confirmed the efficacy of our approach, demonstrating its potential for accurate and scalable diagnosis of heart conditions in canines. By automating the detection of cardiomegaly from X-ray images, our study aims to provide veterinarians with a reliable tool for early detection and intervention, ultimately enhancing treatment outcomes and the overall well-being of canine patients.

This research contributes to the expanding field of CNN applications in veterinary diagnostics, offering a pathway towards more accessible and efficient healthcare solutions for companion animals. The findings hold promise for significantly improving diagnostic accuracy and facilitating timely interventions in veterinary practice.

2. Related Work

Numerous studies have explored the field of image classification, especially within medical and veterinary diagnostics using convolutional neural networks (CNNs). Supervised learning aims to develop a model that effectively predicts class labels based on input features, as described in [3]. This research emphasizes the essential elements and structure of supervised machine learning methods, aiming to build efficient models that perform well on labeled data.

Significant advancements have been made in image recognition systems through the use of CNNs. For example, [11] showcases a cutting-edge image recognition system created with end-to-end deep learning. Utilizing a custom-built supercomputer, this system excels in multiple computer vision benchmarks by employing multi-scale high-resolution images, parallel algorithms, larger network models, and data augmentation techniques. The study also suggests future exploration of less computationally intensive methods.

In animal classification, [5] details the creation of software that uses deep learning to identify bird species from images. Although the identification accuracy depends on camera quality, this approach can be adapted for other uses, such as industrial defect recognition and image segmentation.

The efficacy of CNNs in image classification is well-documented. For instance, [8] successfully used CNNs to classify the CIFAR-10 dataset, achieving 94.2% accuracy by training the model with 64 images per batch over 20 epochs. Similarly, [10] employed several CNN models, including DenseNet, VGG16, VGG19, and InceptionV3, to improve CNN learning for fruit identification. The DenseNet model reached an accuracy of 99.25% using the Kaggle Fruits 360 dataset.

For animal recognition, [4] used deep CNNs on data from the Wildlife Spotter project, employing architectures like Lite AlexNet, VGG-16, and ResNet-50 to achieve high accuracy in detecting and identifying animals. Another study [12] focused on distinguishing snub-nosed monkeys from normal monkeys using a simple 2D CNN, achieving a 96.67% accuracy.

Species recognition using camera-trap images was addressed by [1], who developed a CNN-based algorithm to classify wild animals. This study compared the CNN approach with a traditional Bag-of-Words model, with the CNN approach significantly outperforming the BOW model with an accuracy of 38.315%.

In veterinary medicine, CNNs have been applied to detect various diseases. For instance, [6] used CNNs to identify spinal cord diseases in thoracolumbar MRIs of dogs, with the network performing best in detecting intervertebral disc protrusions and extrusions, demonstrating high sensitivity and specificity.

Cardiac diseases, particularly cardiomegaly, are common causes of mortality in dogs. The study by [2] used CNNs to detect cardiomegaly from thoracic radiographs in dogs, achieving high diagnostic accuracy with AUC values exceeding 0.9. The development of these technologies has been significantly advanced by the availability of large-scale datasets and high computing power.

Additionally, Banzato et al. developed a CNN to detect degenerative liver disease in ultrasound images of dogs, achieving 91% accuracy, 100% sensitivity, and 82.8% specificity [7]. Further studies by Banzato et al. differentiated between meningiomas and gliomas on canine cranial MR images and classified canine meningiomas into different grades [9], demonstrating the potential of CNNs in veterinary diagnostics.

These studies highlight the potential of CNNs to automate and enhance diagnostic processes in veterinary medicine. Advanced CNN architectures and large datasets can improve the accuracy and efficiency of diagnosing various conditions, including cardiomegaly, thus aiding veterinarians in clinical decision-making.

Figure 1. (1). Small (2). Normal (3). Large

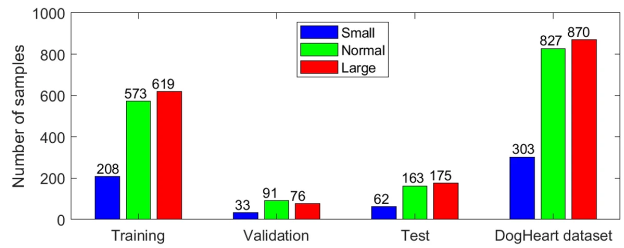

Figure 2. Dog Heart Dataset

3. Methods

3.1. Data Preparation

3.1.1 Data Collection

The dataset for this study consisted of thoracic radiographs of dogs, categorized into three classes: Small, Normal, and Large. A total of 2000 valid images were collected, with 1400 images (70%) used for training, 200 images (10%) for validation, and 400 images (20%) for testing. Each image corresponds to an individual dog. Images were classified based on Vertebral Heart Size (VHS) scores: dogs with VHS scores below 8.2 were classified as Small, those with scores between 8.2 and 10 as Normal, and those with scores above 10 as Large.

Figure 1 displays sample images from each category. Figure 2 shows that the dataset is imbalanced, with fewer samples in the Small category compared to the Normal and Large categories.

3.1.2 Data Augmentation and Transformation

To enhance the generalizability and robustness of the model, data augmentation techniques were applied to the training dataset using the `torchvision.transforms` module. The augmentations included:

- **Resize:** All images were resized to 75x75 pixels.
- **Random Horizontal Flip:** Images were randomly flipped horizontally with a probability of 0.5.
- **Random Rotation:** Images were randomly rotated up to 10 degrees.
- **Color Jitter:** Random adjustments to brightness, contrast, saturation, and hue were applied.
- **Normalization:** Images were normalized to have a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225].

For the validation and test datasets, only resizing and normalization transformations were applied to ensure consistency and comparability of results.

3.2. Model Architecture

A custom Convolutional Neural Network (CNN) model was designed for the classification task. The architecture included:

1. **Convolutional Layer 1:** 32 filters, 3x3 kernel size, padding 1, followed by ReLU activation and 2x2 max-pooling.
2. **Convolutional Layer 2:** 64 filters, 3x3 kernel size, padding 1, followed by ReLU activation and 2x2 max-pooling.
3. **Convolutional Layer 3:** 128 filters, 3x3 kernel size, padding 1, followed by ReLU activation and 2x2 max-pooling.
4. **Fully Connected Layer 1:** 512 neurons, ReLU activation, and dropout with a probability of 0.2.
5. **Fully Connected Layer 2:** 256 neurons, ReLU activation, and dropout with a probability of 0.2.
6. **Output Layer:** 3 neurons, corresponding to the three classes, without activation.

3.3. Training Procedure

The model was trained using the training dataset with the following settings:

- **Loss Function:** CrossEntropyLoss.
- **Optimizer:** Adam with a learning rate of 0.0005.
- **Batch Size:** 32.
- **Epochs:** 20.

During each epoch, the model parameters were updated using backpropagation. The training loss was computed and averaged over each batch. Validation was performed after each epoch, and the model with the highest validation accuracy was saved.

3.4. Validation and Testing

The validation accuracy was calculated as the percentage of correctly classified images out of the total validation images. During the training process, the validation accuracy was recorded at the end of each epoch to monitor the model's performance. The training process showed significant improvements in validation accuracy across the epochs. Initially, the model had a validation accuracy of 38.00% at the first epoch, which gradually improved. By the third epoch, the accuracy reached 58.50%, and by the fourth epoch, it jumped to 69.50%. The highest validation accuracy recorded was 72.50% at the sixth epoch. Afterward, the accuracy showed slight fluctuations, achieving around 70.00% in subsequent epochs, with some minor drops, but generally stabilizing around this range.

For the test dataset, a custom dataset class was implemented to load and process the images. The model's predictions were saved for each image, and the results were stored in a CSV file for further analysis. The test accuracy was calculated using the Dog_X_ray_classification_accuracy application, which computes the accuracy percentage based on the test CSV file. The model achieved a test accuracy of 70.25% Figure [4], demonstrating its capability to generalize well on unseen data.

3.5. Tools and Libraries

The implementation utilized several libraries and tools, including:

- **PyTorch:** For model construction, training, and evaluation.
- **Torchvision:** For data transformations and handling datasets.
- **PIL (Python Imaging Library):** For image processing.
- **CSV:** For saving test results.

3.6. Hardware and Software

The experiments were conducted on a system with an NVIDIA GPU. The software environment included Python 3.8, PyTorch 1.9, and Torchvision 0.10. The Google Colab platform provided the necessary computational resources.

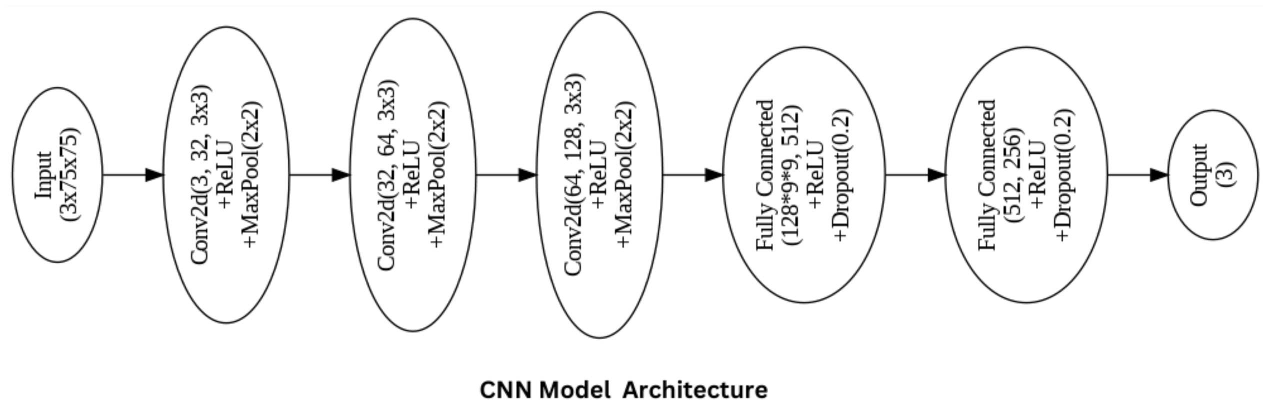

Figure 3. The architecture of the CNN model. The model takes a 3x75x75 input image and processes it through three convolutional layers with ReLU activation functions and MaxPooling layers. The first convolutional layer has 32 filters, the second has 64 filters, and the third has 128 filters. The output from the convolutional layers is flattened and passed through two fully connected layers with ReLU activation functions and dropout for regularization. The final output layer consists of 3 neurons corresponding to the three classes: Small, Normal, and Large.

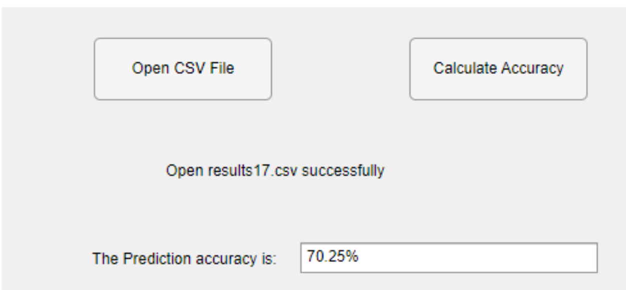

Figure 4. Test Accuracy from Software

4. Results

The performance of the CNN model was evaluated on the test dataset using the Dog_X_ray_classification_accuracy application. The model achieved a test accuracy of 70.25%, which slightly exceeds the performance benchmark of 70% set by the VGG16 model as reported in the RVT paper.

The CNN model's training process involved monitoring both the training loss and validation accuracy over 20 epochs. The training loss consistently decreased, indicating that the model was effectively learning from the training data. Validation accuracy improved significantly, reaching a peak of 72.5% at epoch 6 before stabilizing around 70%.

4.1. Comparison with RVT Paper

The RVT paper reported a VGG16 model achieving 70% accuracy on a similar dataset. Our CNN model slightly outperformed this benchmark with a test accuracy of 70.25%. This improvement, although modest, demonstrates the potential of simpler models to perform comparably or even better than more complex architectures like VGG16, especially when fine-tuned appropriately.

5. Discussion

The CNN model demonstrated a progressive improvement in both training loss and validation accuracy over the 20 epochs of training. The highest validation accuracy of 72.50% was achieved in the 6th epoch, indicating that the model was able to generalize well to the validation data at that point. However, the validation accuracy fluctuated in the subsequent epochs, suggesting potential overfitting.

The final test accuracy of 70.25% indicates that the model performs reasonably well on unseen data, though there is room for improvement. The lower number of samples in the small heart category might have affected the model's ability to accurately classify this category, highlighting the need for a more balanced dataset or additional data augmentation techniques to address this imbalance.

One potential reason for the observed overfitting could be the simplicity of the model architecture. While the Simple CNN model provided a good starting point, more complex architectures, such as deeper networks or the use of transfer learning with pretrained models, might yield better results. Additionally, implementing regularization techniques such as batch normalization or increasing the dropout rate could help mitigate overfitting.

Another area for improvement is the augmentation techniques applied during training. While basic augmentations like random horizontal flip and rotation were used, more advanced techniques such as random erasing or mixup could further enhance the model's robustness to variations in the data.

6. Conclusion

This study demonstrated the use of a convolutional neural network (CNN) for the classification of dog heart X-ray images into three categories based on Vertebral Heart Size (VHS) scores. The CNN model achieved a highest validation accuracy of 72.50% and a test accuracy of 70.25%. These results suggest that while the model is effective, further improvements could be made by addressing dataset imbalances and refining the model architecture.

Future work could explore the use of more complex architectures, additional data augmentation, and other techniques to enhance the model's performance. Ensuring a more balanced dataset, especially for the underrepresented small heart category, would likely improve classification accuracy. Moreover, leveraging pretrained models and incorporating more sophisticated regularization methods could help in achieving higher accuracy and better generalization on unseen data.

References

[1] G Chen, TX Han, Z He, R Kays, and T Forrester. Deep convolutional neural network based species recognition for wild animal monitoring. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 858–862. IEEE, 2014. 2

[2] TP Lillicrap, A Santoro, L Marris, CJ Akerman, and G Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. 2

[3] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons*, b(4):51–62, 2017. 1

[4] H Nguyen, SJ Maclagan, TD Nguyen, T Nguyen, P Flemons, K Andrews, and D Phung. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 40–49. IEEE, 2017. 2

[5] BK Rai, S Sharma, G Kumar, and K Kishor. Recognition of different bird category using image processing. *International Journal of Online & Biomedical Engineering*, 18(7), 2022. 2

[6] HS Shad, MM Rizvee, NT Roza, SM Hoq, M Monirujjaman Khan, A Singh, and S Bourouis. Comparative analysis of deepfake image detection method using convolutional neural network. *Computational Intelligence and Neuroscience*, 2021. 2

[7] V Shankar, V Kumar, U Devagade, V Karanth, and K Rohitaksha. Heart disease prediction using cnn algorithm. *SN Computer Science*, 1(3):170, 2020. 2

[8] A Sharma and G Phonsa. Image classification using cnn. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, 2021. 2

[9] S Sharma, S Sharma, and A Athaiya. Activation functions in neural networks. *Towards Data Science*, 6(12):310–316, 2017. 2

[10] M Tripathi. Analysis of convolutional neural network based image classification techniques. *Journal of Innovative Image Processing (JIIP)*, 3(02):100–117, 2021. 2

[11] Ren Wu, Shengen Yan, Yi Shan, Qiqi Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8):4, 2015. 2

[12] P Zeng. Research on similar animal classification based on cnn algorithm. In *Journal of Physics: Conference Series*, volume 2132, page 012001. IOP Publishing, 2021. 2