A Deep Learning Approach for Accurate Dog Breed Image Classification

Joey Zhang
Student# 1006750437
joeyz.zhang@mail.utoronto.ca

Ingrid Wu
Student# 1006805085
ingrid.wu@mail.utoronto.ca

Shilei (Andy) Zhao
Student# 1006867491
andyy.zhao@mail.utoronto.ca

Raymond Huynh
Student# 1007058238
r.huynh@mail.utoronto.ca

Abstract

Accurate dog breed image classification is crucial for improving the dog adoption process and enhancing veterinary care. This project proposes a deep learning model based on convolutional neural networks (CNN) for classifying images of 120 dog breeds from a large, high-resolution dataset. The model is structured with a 9-layer CNN, incorporating residual blocks, leaky ReLU activation, and batch normalization to improve performance. We trained the model using a 70-20-10 train-validation-test split and implemented data augmentation techniques to enhance its generalization. Our results show that while the model achieves strong performance on the training data, with accuracy reaching approximately 70%, there is a noticeable gap of about 20% between training and validation accuracy, indicating overfitting. The model's challenges in fine-grained classification suggest that it tended to focus on broad, high-level features like coat color and body proportions rather than subtle breed-specific traits. —-Total Pages: 10

1 Introduction

Animal shelters are critical in providing temporary homes for lost or abandoned dogs, with approximately 3.1 million dogs entering shelters annually ASPCA (2024). These shelters face the challenge of classifying and profiling dogs to streamline the adoption process. Dog breed classification is an essential tool for enhancing adoption rates, as many potential adopters search for specific breeds based on their lifestyle, preferences, or previous experience with pets. By facilitating accurate breed identification, shelters can better match dogs with the right families, potentially decreasing the time dogs spend in shelters and increasing their chances of finding permanent homes.

Dog breed classification is also beneficial for tracking lost pets and reuniting them with their owners. In many cases, breed characteristics can help identify stray dogs or missing pets. Systems that leverage breed classification can integrate with databases of lost pets, improving the odds of reunification, a significant issue as only about 26% of stray dogs are returned to their owners each year Community Care College (2023).

Breed classification is crucial in veterinary medicine, as different breeds have unique predispositions to certain health conditions. By accurately identifying a dog's breed, veterinarians can predict potential health issues and offer breed-specific preventive care. Moreover, this can help streamline patient records, reducing administrative time and allowing veterinarians to focus more on direct care.

Our goal is to develop an accurate image classification system capable of identifying various dog breeds. We aim to improve the precision of breed recognition, even in cases where dogs share similar features or exhibit variations in age and color. This project has the potential to streamline processes in shelters, veterinary clinics, and wildlife conservation efforts, ultimately improving outcomes for both animals and caretakers.

2 ILLUSTRATION

Figure 1 illustrates the basic idea of our project. The model will take in input images of various dog breeds, and produce an output with determined prediction labels.

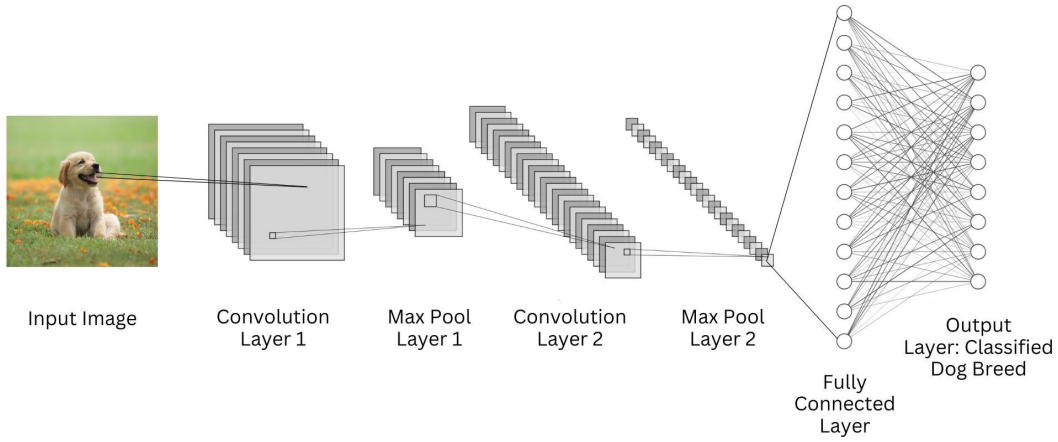


Figure 1: Illustration of Overall Model

3 BACKGROUND AND RELATED WORK

In recent years, image recognition technology has been intensively researched and improved upon. This section will explore existing implementations of image classification on different use cases, while providing key comparisons and contrasts with our approach.

Studies like Liu et al. (2012) use attention localization to focus on key features (e.g., ear shape, fur texture). While this improves accuracy, our model focuses on a general CNN, with attention layers potentially added in the future for feature distinction. Another research by Howard et al. (2017) used noisy data to simulate real-world imperfections. Our curated Stanford dataset is less noisy, but we plan to use data augmentation to improve robustness against imperfect images, similar to their approach.

The use of GAP in Varshney et al. (2021) reduces parameters while maintaining accuracy, which aligns with our goal of computational efficiency for deployment on low-power devices like those in shelters and clinics.

In Battu & Reddy Lakshmi (2023), k-means clustering was used to handle noisy labels. While our dataset is clean, we may explore clustering if new data with noisy labels is added in the future. Finally, techniques from Zhang (2021), such as multi-scale feature maps, could be applied to improve our CNN's performance by handling scale variations in dog breed images.

It is worthwhile to note that while these papers explored various different architectures such as MobileNet and VGGNet, our model will focus on a general CNN model; specific enhancements may include attention layers or modular design to improve accuracy in noisy environments. Additionally, our project will prioritize computational efficiency to allow deployment on low-power devices or systems used in shelters or clinics.

4 DATA PROCESSING

The data that we have collected comes from ImageNet and is titled "Stanford Dog breed dataset" Khosla et al. (2011). Overall, there are 120 different dog breeds where the average number of pictures per breed is 159 and the median is 171. The dataset comes in two folders, the first folder being the annotations of each image in XML format, and the second folder contains all the images.

These two folders are structured identically such that within these folders contains 120 sub-folders of each dog breed. The dataset can be freely downloaded online and comes packaged in a .zip file.

Using Google Colab, we first mount Google Drive to our Colab runtime. We also upload the .zip file of our dataset to our Google Drive. We then use the "zipfile" python library to extract the contents of that .zip file to a local directory in our runtime (e.g. /content/project/). Then, we go through every picture and its corresponding annotation file and crop the image based on its bounding box values. These cropped images are dumped in the same local directory with 120 folders for each dog breed. Afterwards, we leverage the python library "splitfolders" to split the cropped images into 3 folders for training, validation and testing on our local Colab runtime. We are using a 70-20-10 split. Lastly, we augment the images to be 300x300 and transform them into data loaders for training.

(a) Original uncropped dataset sample image

(b) Cropped and resized dataset sample image

Figure 2: Dataset sample for uncropped (left) and cropped (right) images

One key point of interest is the cropping of the images. Many of the bounding boxes and the dimensions of the original images do not follow a 1:1 aspect ratio shown in Figure 2a, which is required for our model input. Therefore, when cropping, padding must be applied to ensure a square image; the final cropped image is shown in Figure 2b. We believe padding is better than stretching the image to fill the square frame so as to not significantly distort key features of the image. However, we are aware of the trade-off padding has as the model may start to learn the features of the "black bars" attributed to the padding. Out of 20 580 total images, 12 094 images require some sort of padding. Additionally, we are cropping based on the long side, meaning if the width is longer than the height, we pad the height. If we were to crop based on the short side, this could potentially crop out key features of the dog such as its face, which we have decided against.

The 300x300 image size output comes from the dimension distribution of the cropped images, shown in Figure 3 below. The average dimension is 333x333, with the lower end being around 100x100 and the higher end being around 2500x2500. However, most of the distribution falls around 300x300. Therefore to balance fidelity and padding, we decided to go for an image size of 300x300.

For data augmentation, we leveraged the "torchvision" python library. To convert each image to 300x300, we leveraged bicubic interpolation to minimize any artifacts and based on its good reputation for interpolation. We've applied light augmentations such as reflections and normalizations, however have opted out for any rotations. This is because we are already using a bounding box, and rotations could further crop the image and we might lose key features of the dog.

For obtaining new data, the biggest thing is simply how the classes are labeled. For our current dataset, everything is neatly packaged in titled folders for each class. The annotations are purely for the bounding box values. The benefit of having the bounding box is such that it allows the model to focus more on the dog and less on surrounding objects. If we were collecting new data ourselves, we could manually crop them thus not needing to worry about cropping afterwards, and only have to ensure the structure of the folders for each dog breed class.

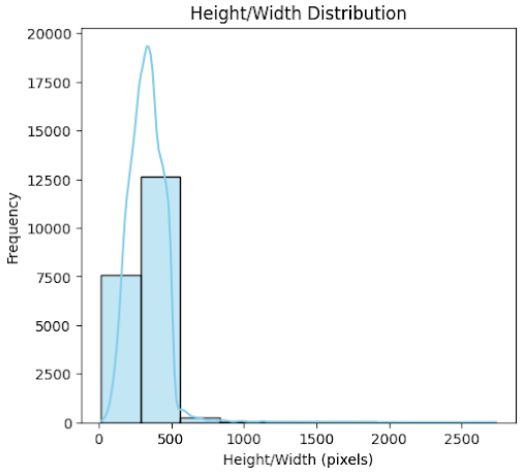


Figure 3: Dataset height and width distribution graph

5 Architecture

For our final model (RizzNet V2), we took inspiration from the 2015 paper on Deep Residual Learning from He et al. (2015) and implemented 10-layer neural network model. The initial 5x5 convolution uses padding=2, stride=2, output channel of 64, followed by a 2x2 max pool with stride=2. Subsequently, the next 8 layers are split into 4 residual blocks where the number of channels for the first block is 64, then doubles at each block up to 512. Lastly, we use global average pooling to output into a 120 fully-connected layer, corresponding to the 120 breeds of dogs for classification. Dropout=0.5 is also used after every residual block and the fully-connected layer. A diagram of this high-level overview can be seen in Figure 4a.

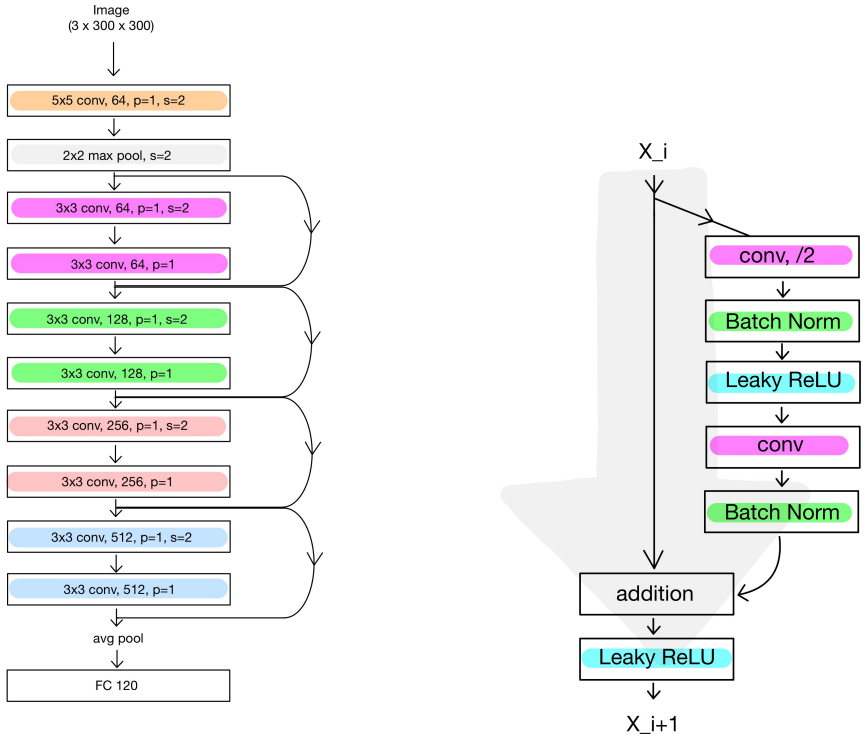
For the residual unit, as shown in Figure 4b, there are two convolutions, where both have padding=1 but the first has stride=2. This is so every time we double the number of channels, we also downsize the image resolution. Through experimentation, our final model implements post-order activation with batch normalization and leaky ReLU with a negative slope=0.2 to help with the dying ReLU problem. For the shortcut connection, it begins at the top of the residual unit, and is summed before the last leaky ReLU activation function.

6 Baseline Model

A Support Vector Machine (SVM) with a one vs one variant and a simple Artificial Neural Network (ANN) are used as baseline models to benchmark the success of our primary model. These baselines provide reference points for performance, and allow us to measure how effectively the primary model improves upon standard classification approaches.

6.1 SVM

In implementing our SVM Baseline Model, the team first converted the tensors loaded via DataLoader to numpy arrays, as this format is required for SVM fitting. We then loaded a subset (around 20%) of the training and test data and applied Principal Component Analysis (PCA) to reduce the feature dimensions to 150 components. This dimensionality reduction aimed to lower computational complexity and prevent system crashes due to high memory usage. While we recognize that fitting the SVM on the entire dataset could potentially improve accuracy, our team was constrained by the available hardware. The highest training and test accuracy the SVM model achieved were merely 18.25% and 4.65% respectively. Despite the extensive tuning of hyperparameters, including regularization constant C and kernel types (linear, RBF, and sigmoid for non-linearity), the test accuracy hovered below 5%. There are various causes for such a low accuracy rate; however, the team sus-



(a) High-level overview

(b) Residual unit

Figure 4: Our final model architecture (RizzNet V2)

pects that the primary reason behind this issue is the lack of sufficient dimensions and the presence of noisy data.

6.2 ANN

The ANN baseline model is a straightforward artificial neural network implemented using PyTorch. It consists of four fully connected layers: the first layer takes a flattened input of size 3×300×300, corresponding to an RGB image with dimensions of 300x300 pixels, and outputs 512 features. This is followed by two additional layers that reduce the feature size to 256 and then to 128, with Rectified Linear Unit (ReLU) activation functions applied after each of these layers to introduce non-linearity. The final layer outputs 120 features representing class scores for a classification task.

Compared to the SVM, the ANN baseline performed better even on the full dataset. This provides a qualitative indication that neural networks, in general, are more effective for image classification tasks, due to their capacity to capture complex patterns and nonlinear relationships within a dataset. During our testing, we noticed that the ANN architecture, while functional, struggled to achieve the same level of accuracy as the CNN. This difference arises because the ANN flattens the images into one-dimensional arrays, discarding valuable spatial information. Without spatial awareness, the ANN has difficulty distinguishing subtle patterns and features in the images, leading to less accurate predictions.

Figure 5 below shows the results for training and validation error for the ANN baseline model. The training error starts relatively high but decreases consistently as training progresses, which shows that the model is learning and improving on the training data. The training error reaches around 0.88 by the end of 30 epochs, which indicates that the model is slowly fitting the training data but is still not achieving very high accuracy (only around 12% error reduction). The validation error begins close to the training error but quickly diverges, fluctuating and showing less consistent improvement

Figure 5: ANN Training results showing signs of over-fitting

compared to the training error. The validation error ends slightly above 0.92, which is higher than the training error, indicating a noticeable performance gap between the training and validation sets. This gap suggests that the model is over-fitting to the training data and not generalizing well to unseen data. The relatively high training and validation errors suggests that the model is limited in its capacity to capture patterns in the data, which could be due to the lack of spatial feature extraction in the ANN.

7 Quantitative Results

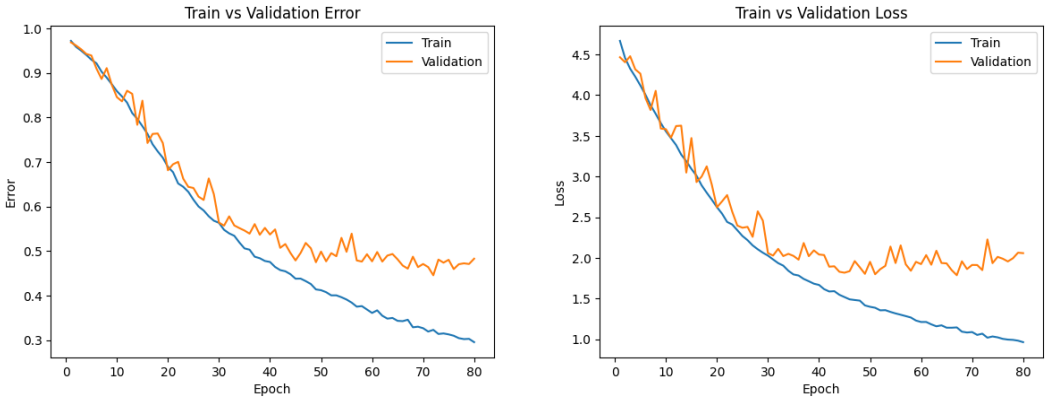
To evaluate the model's performance in classifying 120 dog breeds, we looked at key quantitative measures such as error, loss, and accuracy to understand how well the model performs both on the training and validation datasets.

The training accuracy increases steadily as the model learns, starting at around 5% (due to random initialization) and improving to approximately 70% by the final epoch. This indicates that the model is successfully recognizing and classifying the training data. The validation accuracy follows a similar pattern, beginning at about 10% and reaching around 50% at the end of training. While this shows progress, there is still a large discrepancy when compared to the training accuracy. The gap of around 20% between training and validation accuracy suggests that the model is overfitting, performing well on the training data but struggling to generalize to unseen data.

In terms of error, the training error decreases from 95% to 30%, indicating that the model is reducing its prediction mistakes effectively during training. Similarly, the validation error drops from 90% to 50%, but it plateaus after epoch 30, indicating that the model's ability to generalize to new, unseen data has stagnated. The loss depicts similar information. The training loss consistently decreases from 4.5 at the beginning of the training process to around 1.0 by epoch 80. This suggests that the model is minimizing the difference between its predictions and the actual labels. The validation loss follows a similar trend, starting at 4.5 and dropping to around 1.8 by the final epoch. While the validation loss improves initially, it shows a plateau after epoch 30, further indicating the model's struggle with generalizing beyond the training data.

8 Qualitative Results

The graphs in Figure 6 below illustrate the performance of a machine learning model by comparing training and validation error and loss over 80 epochs.



(a) Train vs. Validation Error

(b) Train vs. Validation Loss

Figure 6: Train and Validation Error (left) and Loss (right) for RizzNet V2

9 Evaluate Model on New Data

Extensive work has been done during the development of the model to aim for a high performance in generalization. However, to ensure that the results collected sufficiently reflect the model's ability to generalize, the team evaluated the model's performance with an additional 2 sets of unseen data to observe performance on new data.

The first dataset consists of unused samples from our original Stanford dataset. The set includes 2 153 new images across the 120 dog breeds from training. Using this dataset, the team saw that performance aligned closely with our collected results, with a classification accuracy of approximately 52.4%.

The second dataset, collected online under the title "The Oxford-IIIT Pet Dataset" Parkhi et al. (2012), provides a collection of both cat and dog breed photos. To ensure consistency between the model and testing data, the dataset was parsed to ensure that only breeds classified in our model were included. The final set of data used consists of 14 dog breeds with 200 images per breed. The performance of the model on this new data was unexpectedly poor, with a classification accuracy of 0.7%.

One reason for this large discrepancy may be due to the difference in the bounding boxes used for the respective datasets. While the bounding boxes of the Stanford dataset typically contain the dog's entire body, the bounding boxes from the Oxford-IIIT dataset are constrained to only the dog's head. Since the data processing workflow in our model crops images to their bounding box dimensions, this leads to a significant difference in the information available in images from each dataset. Thus, the resulting images from our new dataset may not contain enough information to make a confident guess, as the model was trained on images with more visual features. For example, breeds where their bodies are a substantial contributor to their visual classification, such as Pomeranians or chihuahuas, may be disproportionately affected in the second dataset as that information is no longer present in the image. A comparison of a sample from the two datasets cropped to their bounding box dimensions is shown in figure 7 below.

The discrepancy in accuracy may also be attributed to the percentage of visually similar breeds in the new dataset, such as boxers and Saint Bernards. The team noticed that the model was confused between breeds with similar visual features, so the effect of this behaviour could have been multiplied by the smaller range of dog breeds. As the number of breeds decreases, the likelihood of the input being a breed that has a high visual similarity with another breed may be higher. Since the number of breeds in the new dataset decreased from 120 to 14, even 1 pair of visually similar breeds would result in approximately 14% of the input data having a high likelihood of confusion, which could contribute to the low accuracy rate.

Generally, it should be expected that performance on unseen data from the same dataset should be close to our observed results, since the format of the images, metadata, and annotations match those

Figure 7: Images cropped to their bounding box dimensions from the Stanford dataset (left) and the Oxford-IIIT dataset (right)

that the model has been trained on. As our results on new data reflect that, so long as the dataset contains data in a format similar to that of the Stanford dataset, we should expect the model to perform as expected with our training data.

10 Discussion

The team undertook the ambitious challenge of multi-class dog breed classification, targeting 120 distinct breeds. While this scope provided an opportunity to test the robustness of our model, it also introduced significant complexity. A narrower focus—such as classifying 30 or 60 breeds—might have resulted in better performance, particularly given the constraints of our 10-layer neural network. The limited depth and capacity of the model likely made it difficult to effectively distinguish subtle variations between similar breeds. Despite these challenges, the project provided valuable insights into the patterns the model captured and revealed some discriminatory tendencies in how it classified breeds with overlapping characteristics.

10.1 Capturing high level traits

One of the most interesting findings was the model's ability to capture high-level features associated with traditional breed groups (e.g., Toy Dogs, Hounds, Herding Dogs) rather than the fine-grained distinctions of specific breeds (e.g., Chihuahua, Walker Hound). Instead of identifying the unique characteristics of individual breeds, the model often focused on broader patterns such as coat color, fur texture, and body proportions—traits commonly shared within the same group. This suggests that our model was better optimized for recognizing general traits rather than the nuanced details required for precise breed classification.

The team explored solutions to distinguish subtle breed-specific features, including using deeper neural networks like ResNet-18, ResNet-34, or ResNet-50 with transfer learning. This approach enables the model to capture both low-level features (e.g., edges and textures) and higher-level, breed-specific details.

The team conducted testing using ResNet-18 for feature extraction, followed by a single fully connected layer for classification. The ResNet-18 model outperformed our final model, achieving a validation accuracy of 26.8% and test accuracy of 27%. These results highlight the effectiveness of deeper architectures in addressing fine-grained image classification tasks.

Figure 8 illustrates the performance of the ResNet-18 model, showcasing the improvements in both training and test accuracy the ResNet model achieved within only 30 epochs.

10.2 Model Bias in Fine-Grained Classification

Additionally, the RizzNet v2 model exhibited notable classification biases. For instance, it consistently performed better with breeds that have distinct physical characteristics (e.g., Dalmatians vs Pugs) but struggled with visually similar breeds, such as Dobermans and Rottweilers, which share

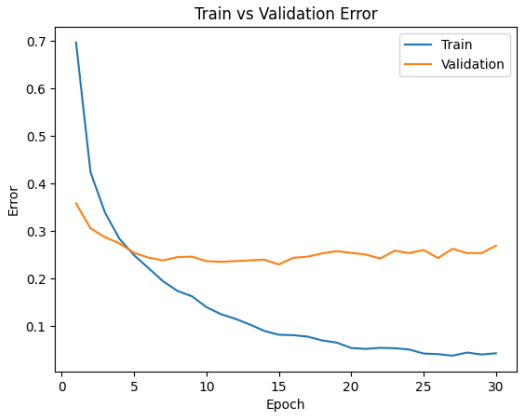


Figure 8: Train and validation error of Resnet-18 architecture over 30 epochs showing significant improvements in terms of starting and ending validation accuracy over the team's RizzNet V2

similar color patterns. In some cases, the model relied heavily on color as the primary determining factor, leading to misclassifications.

One example is the misclassification of Dobermans as Rottweilers due to their shared black-and-tan coat patterns (Figure 9 illustrates this case). Similarly, the model occasionally confused Beagles with Walker Hounds, likely because of their comparable coat coloring and markings. These observations suggest that the model tends to overemphasize visual features like color, while overlooking finer breed-specific distinctions.

(a) Rottweiler misclassified as Doberman

(b) Doberman correctly classified as Doberman

Figure 9: The model classifying a Rottweiler (left) as a Doberman (right)

11 Ethical Considerations

There are multiple ethical issues that may arise from usage of the proposed model including misclassification leading to incorrect owners, as well as bias against certain breeds. Misidentifying a dog could result in the wrong owner being contacted or, conversely, a rightful owner being denied their pet. If a dog is misclassified, veterinarians may fail to recognize breed-specific health risks that are relevant to that dog. This oversight can lead to inadequate preventive care and management.

This model may also contribute to the popularity of certain breeds for wanna-be dog owners, which may amplify existing issues currently seen in selective dog breeding such as genetic disorders and long-term health issues Menor-Campos (2024).

REFERENCES

ASPCA. Pet statistics. 2024. http://vision.stanford.edu/aditya86/ImageNetDogs/.

Thirupathi Battu and D. Sreenivasa Reddy Lakshmi. Animal image identification and classification using deep neural networks techniques. *Measurement: Sensors*, 25:100611, 2023. ISSN 2665-9174. doi: https://doi.org/10.1016/j.measen.2022.100611. URL https://www.sciencedirect.com/science/article/pii/S2665917422002458.

Community Care College. Pet facts. https://communitycarecollege.edu/veterinary-assistant/pet-facts/, 2023. Accessed: 2024-10-04.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 04 2017. doi: 10.48550/arXiv.1704.04861.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Imagenet dogs. http://vision.stanford.edu/aditya86/ImageNetDogs/, 2011. Accessed: 2024-09-28.

Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (eds.), *Computer Vision – ECCV 2012*, pp. 172–185, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33718-5.

D. J. Menor-Campos. Ethical concerns about fashionable dog breeding. *Animals : an open access journal from MDPI*, 14(5), 2024. doi: 10.3390/ani14050756.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. The oxford-iiit pet dataset [dataset], 2012. URL https://www.robots.ox.ac.uk/~vgg/data/pets/.

Akash Varshney, Abhay Katiyar, Aman Kumar Singh, and Surendra Singh Chauhan. Dog breed classification using deep learning. In *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–5, 2021. doi: 10.1109/CONIT51480.2021.9498338.

Ruihao Zhang. Classification and identification of domestic catsbased on deep learning, 2021.