Dog Breed Classification Using Machine Learning

Karolayne Isabel Gaona Chong
Earlham College
Richmond, Indiana, USA
kigaona20@earlham.edu

ABSTRACT

Image classification is an important problem task in computer vision. The field of Machine Learning in Computer Vision is growing due to technological developments by improving system performances by learning from experience via computational methods. Convolutional Neural Networks (CNNs) are important elements for feature extraction in image classification. This project attempts to classify images of dogs into their breed categories. Three selected CNNs architectures were used and merged into a single model, applying data augmentation techniques to improve the robustness of the model.

KEYWORDS

Convolutional neural networks (CNNs), Data augmentation, Stanford Dog Dataset, Image classification, ResNet, NASNet, InceptionV3, Testing, Trainig

ACM Reference Format:

Karolayne Isabel Gaona Chong. 2024. Dog Breed Classification Using Machine Learning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

There are multiple applications for Machine Learning, and classification problems are one of them [11]. The field of Machine Learning in Computer Vision is growing. Technological developments have impacted the field to gain increased momentum [13]. Computer vision is a subfield of artificial intelligence that aims for computers to perform tasks and understand input information as humans from images [15]. Machine learning improves system performances by learning from experience via computational methods [19]. The main task of machine learning is to develop learning algorithms that build models from data by feeding the learning algorithm with experience data. These algorithms may make predictions on new observations [19]. The process that the system uses to learn from data is iterative. As the model continues to be exposed to more data, it will capture more information and learn from it [13]. The motivation for this project is to learn how to apply classification tools for dog breed classification through digital photographs. There are multiple efforts in researching the animal recognition field. Dog breed recognition can be crucial in providing proper training and health training [2]. In daily cases, dog breed recognition is done by humans; however, some dog breeds may be challenging to recognize due to their physical similarities. The current papers that research dog breed classification uses the methodology and results of fine-tuning Convolutional Neural Networks (CNNs) for two different CNNs architectures using a large dog images dataset [14]. CNNs are becoming more popular for image classification [13]. A CNN is a feed-forward Artificial Neural Network and deep learning class that can take images as input. CNNs have at least one fully connected layer followed by the desired number of fully convolutional layers as a standard multilayer network [1]. The layers are input, hidden, and output layers. In a basic form, images are constructed as matrices of pixels. The pixel values are given as input to an input layer with weights and biases. The output layer is a fully connected layer to classify the images with their belonging class. Finally, the hidden layer could be convolutional, pooling, or fully connected [15]. Although CNNs are powerful architectures with excellent features extraction capabilities, they still have explanation limitations if compared to manually extracted statistical features such as decisions trees or k-Nearest Neighbours, etc [7]. The core idea of this project is to aim improvement accuracy and efficiency of dog breed identification from digital photographs by merging three pre-trained CNNs models, specifically ResNet, NASNet, and InceptionV3, on the Stanford Dogs Dataset. While implementing data augmentation techniques to contribute to the robustness of the model.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 BACKGROUND AND RELATED WORK

Before machine learning became more popular became so popular, several traditional techniques were used for image classification problems. These techniques relied on handcrafted features and heuristics to recognize patterns in images, Feature Extraction, Image Segmentation, Handcrafted Neural Networks, etc. The research of Borwarnginn et al. addresses some of the challenges of a few traditional approaches that attempted to solve the problem of dog breed classification using methods such as Coarse to fine classification [8][2]. Traditional techniques used local descriptors to attempt to find similarities between images. After local descriptors were extracted, they were put together in multidimensional features space to create words to describe them visually. However, for image classification problems, it seems like machine learning outperforms traditional techniques due to its ability to automatically extract representative features and patterns from each image [12].

3 METHODOLOGY

I used the Stanford Dog Dataset [10], which has 20,580 images which was organized in 120 classes. Each class represents a different dog breed. This dataset was built using pictures and annotations from ImageNet. The dataset was separated into training data with

Figure 1: Example of Stanford Dog Dataset images

Figure 2: Distribution of images in dataset

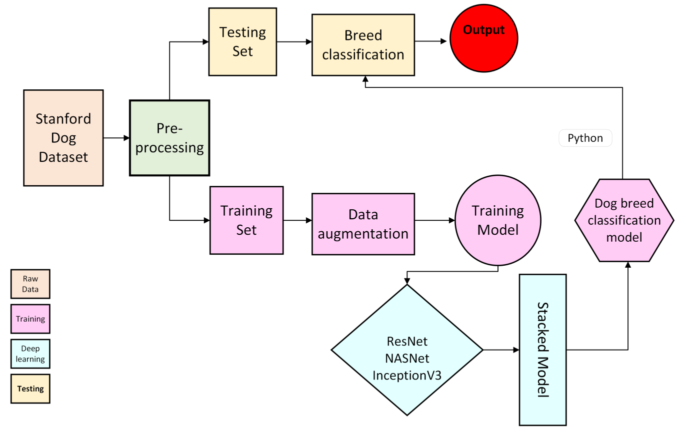

Figure 3: Data Architecture Diagram

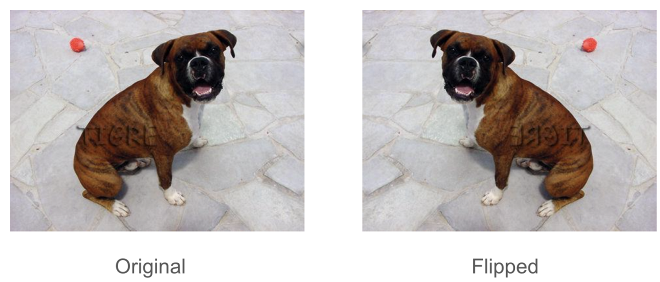

Figure 4: Example of an image from the Stanford Dog Dataset being flipped horizontally

12,000 images and testing data with 8,580 images. I choose the Stanford Dog Dataset due to its large number of high quality images of dogs from different breeds, providing a diverse dataset for training and testing the machine learning models. In this machine learning model implemented for image classification, the output of the training phase is a trained neural network model capable of classifying dog breeds from input images. The model was trained to predict the probability distribution over the different dog breed classes for a given input image. The model also evaluates the trained model during the testing phase. Similarly to [2], this model architecture consists of three main phases: data preparation, training, and testing.

3.1 Data Preparation

There are important keys for the preprocessing part to process and prepare the images to be fed to the deep learning model. Here is what happened during this process:

- The images were resized to a set size. It made sure that all images had the same dimensions, which was necessary to feeding them into the neural network.
- Each pixel value in the images was normalized to a value between 0 and 1. In this project, normalization was achieved by dividing the pixel values by 255, which was the maximum pixel value for an 8-bit color channel. Normalization helps in stabilizing the training process and improving convergence [6].

3.1.1 Data Augmentation: Data augmentation technique was applied to the training data. The images were horizontal flipped. This technique may increase the diversity of the training data without requiring additional labeled samples. This technique could helps prevent overfitting and improves the model's ability to generalize the unseen data.

3.1.2 Preprocessing Functions: Three preprocessing functions were defined. One function for each pre-trained CNN model. These functions performed specific preprocessing steps needed by the requirements of each model. These steps include resizing the images to the expected input dimensions of each model and applied mean subtraction and scaling for preprocessing operations.

3.2 Model Training

This model underwent training on a large dataset, analyzing dog images to get the ability to identify various dog breeds. The initial training phase involved understanding traditional attributes [2]. As the model go deeper into the dataset, it attain a holistic understanding involving identifying entire objects, such as dogs, and parts of them, such as tails or ears, as well as the relationships between these elements. A natural characteristic of these model lies in its autonomous learning capability, requiring no explicit instructions for feature identification. The necessity for manual specification of these parts is obviated, as the model autonomously understands relevant features through a process denoted as feature learning. This methodology emulates the model's training to recognize dog breeds adeptly, alleviating the need for exhaustive manual intervention.

3.2.1 Model Architecture: A stacked model architecture was used to gather all the outputs and learned information from the three pre-trained CNNs. The stacked model architecture combined features extracted from ResNet50, NASNet, and InceptionV3. The pre-trained CNNs were used as feature extractors, gathering representations of features from the input images. The stacked model may have improved the strengths of the three CNN architectures. Each CNN was able to capture different elements of the input images, leading to a more comprehensive representation of features. By combining features from multiple CNNs, the stacked model aimed to improve the overall classification performance and robustness. The used CNNs are parametrized by the size and the number of the maps, kernel sizes, skipping factors, and the connection table [3]. They used initialization parameters such as batch size, number of epochs, learning rate, etc. The batch size determined the number of samples for the training phase of the CNNs. Then, the CNNs processed all the training data but incremented only by the batch size [5]. In this project, the batch size was 32. The training process may be more efficient if the batch size was higher than 1. An epoch is one complete run through the whole training process; the number of epochs may vary depending on the task, the dataset, etc. In this project, the training process went through 30 epochs.

3.2.2 Feature Extraction: ResNet50, NASNet, and InceptionV3 are state-of-the-art CNN architectures that have been pre-trained on large-scale image dataset. Using State-of-the-art in classification problems can be a gradient booster. It gives a prediction model as an ensemble of weak prediction models that make very few assumptions about the data. These pre-trained CNNs have learned to extract meaningful hierarchical representations of features from input images through layers of convolutional and pooling operations. These features captured different levels of abstraction, from low-level edges and textures to high-level object parts and concepts. Rather than training these CNNs from scratch, which would require a large amount of labeled data and computational resources, the pre-trained models were used as feature extractors. The learned weights and architectures of these models were preserved, and only the top layers (fully connected layers) were replaced or modified for the specific classification task. During the training process of the stacked model, the input images are passed through each pre-trained CNN, and the features extracted from different layers of the CNNs were concatenated or combined to form a unified feature representation for classification. This approach allows the stacked model to benefit from all the representations learned by the pre-trained CNNs, capturing diverse features from the input images. The selection of ResNet50V2, NASNetLarge, and InceptionV3 for this project was based on several factors: architectural characteristics, performance in image classification tasks, and suitability for transfer learning. They all are models with multiple layers, allowing to capture complex features from input images. Each of these CNN architectures have documentation in previous research on performance in image classification tasks. Finally, each of these CNNs models have strength in different areas, so they can capture a broader range of visual features and patterns from the input images.

- *ResNet*: Residual Neural Network is a deep learning model in which the weight layers learn residual functions with reference to the layer inputs. A Residual Network is a network with skip connections that perform identity mappings, merged with the layer outputs by addition [4].
- *NASNet*: is an architectural building block on a small database and then transfer the block to a larger database. The NASNet model achieves state-of-the-art results with smaller model size and lower complexity (FLOPs) [18].
- *InceptionV3*: is a convolutional neural network for working with image analysis and object detection. Inception helps with the classification of objects [16].

Table 1: Architecture for CNNs

Layer Name	Output Shape
input_1	[(None, 331, 331,3)]
lambda	(None, 331, 331,3)
lambda_1	(None, 331, 331,3)
lambda_2	(None, 331, 331,3)
resnet50v2	(None, 2048)
NASNet	(None, 4032)
inception_v3	(None, 2048)
concatenate_6	(None, 8128)
dense	(None, 512)
dense_1	(None, 256)
dense_2	(None, 120)

3.2.3 Transfer Learning. Transfer learning focuses on storing information gained while solving one problem and applying it to a different but related problem. According to Raduly et al., transfer learning helps improve the performance of models without starting from scratch. In this project, transfer learning was applied when using CNNs as feature extractors in the stacked model architecture. The CNNs were initialized with pre-trained weights; the models already know about generic visual features, such as edges, textures, and object parts, which those parts are essential for classification tasks. Fine-tuning was also applied to the training data with learned weights and biases.

3.3 Model Testing

The trained stacked model was evaluated on a separate dataset to assess its performance in classifying dog breeds. The testing dataset consists of images that the model has not been exposed to during training. This ensures an unbiased evaluation of the model's performance on unseen data. The performance of the model was evaluated using evaluation metrics. The trained stacked model was evaluated on the testing dataset using an evaluation method, which computes the specified metrics on the test data. The model's predictions on the test images were compared against the ground truth labels to calculate the evaluation metrics.

4 DESIGN AND IMPLEMENTATION

To build this project, it was necessary to have an environment that supports the needed libraries. The Stanford Dog Dataset was obtained as a folder with 120 subfolders, each with hundreds of JPG images. The dataset was read into a Python script using the TensorFlow library's ImageDataGenerator class. The training and testing models were incorporated into a Python script using TensorFlow, NumPy, and Matplotlib.

5 RESULTS

5.1 Training Evaluation

The model's training and validation accuracy were monitored and plotted over the epochs. The training loss was of 0.62 and the training accuracy was of 0.81. From the plotted graphs, I was able to obtain the trend of accuracy improvement during training. Higher accuracy indicates better performance in correctly classifying images into their respective dog breeds. Similarly, the training and validation loss was monitored and plotted over the epochs. Lower loss values indicate better convergence and model performance. A decrease in loss over epochs indicates that the model was effectively learning the patterns in the training data.

- The training loss decreases consistently over the epochs, indicating that the model was learning to minimize the error of the predicted and the actual labels.
- The training accuracy increases rapidly, meaning that the model was improving in correctly classifying the training data.

5.2 Testing Evaluation

After training, the model was evaluated on the test data to assess its performance on unseen data. The testing accuracy and loss provide insights into how well the model generalizes to new, unseen images. Higher accuracy and lower loss on the test set indicate better generalization ability. The model seems to have achieved a 50.64 on test loss while a 0.007 on test accuracy.

- The testing loss changes and was considerably higher than the training loss, indicating potential overfitting.
- The testing accuracy was low and did not improve over the epochs. This may indicate that the model did not classify the unseen data well.

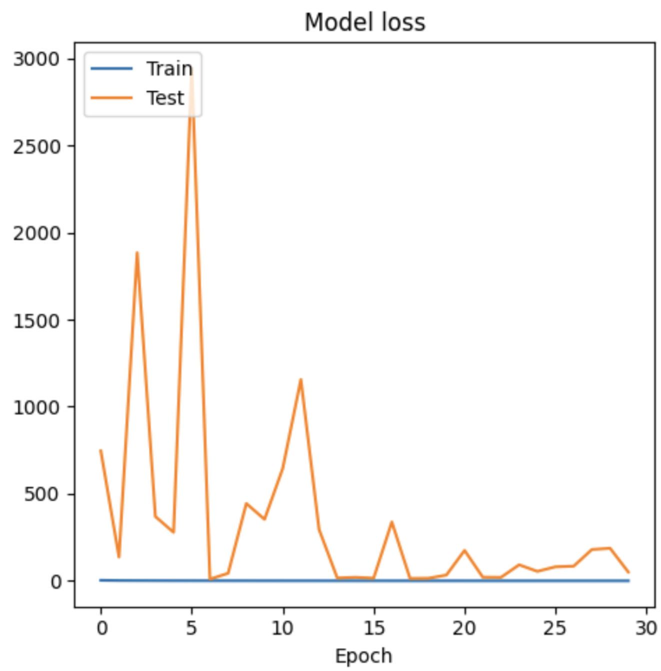

Figure 5: Train loss

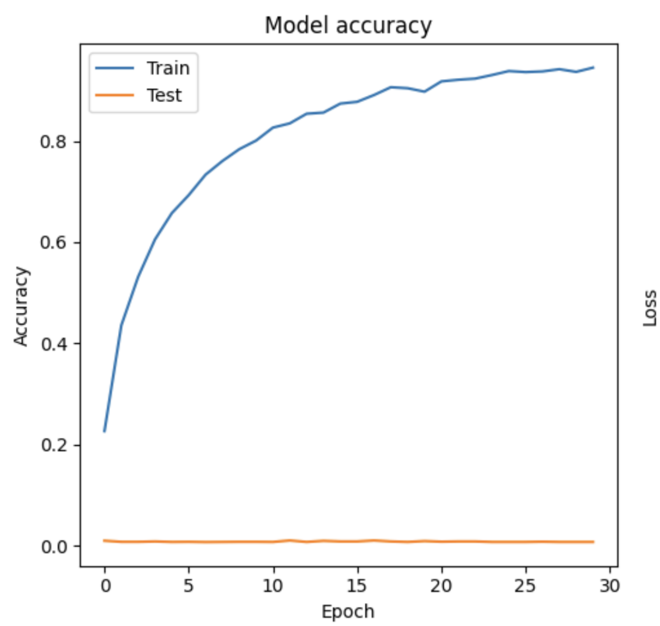

Figure 6: Train accuracy

Figure 7: Test loss and accuracy

5.3 Metrics scores

The performance of the model was evaluated using evaluation metrics. Standard metrics for multi-class classification tasks like dog breed classification include accuracy, precision, recall, and F1-score. Accuracy measures the overall accuracy of the model. Precision measures the proportion of correctly predicted positive samples among the instances predicted as positive. Recall is a classification performance metric for measuring how well a model can identify all positive samples out of the total number of positive samples. The F1-score is a classification performance metric in predicting a particular class [17]. As seen in Figure 9, the model achieved perfect performance across all metrics. This information suggests that the model learned the patterns from the training data. However, since the testing results were not satisfactory, there could have been a few issues this project encountered, such as overfitting.

Table 2: Model Performance Over Epochs

Metric	Value
Accuracy	1.0
Precision	1.0
Recall	1.0
F1-score	1.0

5.3.1 Overfitting: In machine learning, overfitting happens when an algorithm fits too closely or exactly to its training data. This can lead to a model that is not able to make predictions out of any data expect its own training data [9]. This model performed well during the training phase and did pretty poorly during the testing phase. They also showed good metrics during the testing phase but had poor testing results. This may suggest an overfitting problem. Because the models I chose for this project can be too complex, overfitting was an issue this project could encounter. Despite the effort to apply techniques to reduce overfitting, such as data augmentation, the project seemed to fall into overfitting due to the model's ability to memorize patterns specific to the training data rather than learning general features to be able to recognize them in unseen data.

6 CONCLUSION

While the stacked model architecture was a combination from features from pre-trained CNNs like ResNet50, NASNet, and InceptionV3, it showed promising results during training. However, the model struggled to classify well on unseen test data for dog breed classification. The high training accuracy but low testing accuracy, together with the significant gap between training and testing loss, indicate the model likely overfitted to the training data. Despite techniques like transfer learning and data augmentation, the complex CNN architectures may have memorized patterns specific to the training set rather than learning generalizable features for robust classification on new images.

7 FUTURE WORK

Further analysis can be implemented in the testing phase to find the root of the poor accuracy. Further experimentation can also be implemented by using other combinations of CNN models, data augmentation, and fine-tuning techniques to improve performance on this specific dataset. Explore hyperparameters such as batch size, learning rate, and dropout rate to find the optimal configuration for the model. Finally, I am looking to apply this model and implement it into an interactive user mobile application where users can take a real-time picture of a dog and get the dog's breed's name and further information about the breed.

8 ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Charlie Peck, Dr. David Barbella, and Porter Libby for their invaluable guidance, support, and constructive feedback throughout all stages of this project.

REFERENCES

[1] Durga Bhavani, Mir Habeebullah Shah Quadri, and Y Ram Reddy. 2019. Dog Breed Identification Using Convolutional Neural Networks on Android. *CVR Journal of Science and Technology* 17, 1 (2019), 62–66.
[2] Punyanuch Borwarnginn, Worapan Kusakunniran, Sarattha Karnjanapreechakorn, and Kittikhun Thongkanchorn. 2021. Knowing your dog breed: Identifying a dog breed with deep learning. *International Journal of Automation and Computing* 18 (2021), 45–54.
[3] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. 2011. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*. Citeseer.
[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[5] Md Anwar Hossain and Md Shahriar Alam Sajib. 2019. Classification of image using convolutional neural network (CNN). *Global Journal of Computer Science and Technology* 19, D2 (2019), 13–18.
[6] Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. 2023. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE transactions on pattern analysis and machine intelligence* 45, 8 (2023), 10173–10196.
[7] Javier Huertas-Tato, Alejandro Martín, Julian Fierrez, and David Camacho. 2022. Fusing CNNs and statistical indicators to improve image classification. *Information Fusion* 79 (2022), 174–187.

[8] Yuqi Huo, Yao Lu, Yulei Niu, Zhiwu Lu, and Ji-Rong Wen. 2019. Coarse-to-fine grained classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1033–1036.
[9] IBM. [n. d.]. *Overfitting*. IBM. https://www.ibm.com/topics/overfitting
[10] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. 2024. Stanford Dog Dataset. http://vision.stanford.edu/aditya86/ImageNetDogs/
[11] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* 160, 1 (2007), 3–24.
[12] Òscar Lorente, Ian Riera, and Aditya Rana. 2021. Image classification with classic and deep learning techniques. *arXiv preprint arXiv:2105.04895* (2021).
[13] Kaitlyn Mulligan and Pablo Rivas. 2019. Dog breed identification with a neural network over learned representations from the xception cnn architecture. In *21st International conference on artificial intelligence (ICAI 2019)*.
[14] Zalán Ráduly, Csaba Sulyok, Zsolt Vadászi, and Attila Zölde. 2018. Dog breed identification using deep learning. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 000271–000276.
[15] Muthukrishnan Ramprasath, M Vijay Anand, and Shanmugasundaram Hariharan. 2018. Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics* 119, 17 (2018), 1307–1319.
[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.
[17] Nattakan Towpunwong, Napa Sae-Bae, et al. 2023. Dog Breed Classification and Identification Using Convolutional Neural Networks. (2023).
[18] Sik-Ho Tsang. 2019. *Review: NASNet — Neural Architecture Search Network (Image Classification)*. https://sh-tsang.medium.com/review-nasnet-neural-architecture-search-network-image-classification-23139ea0425d December,2023.
[19] Zhi-Hua Zhou. 2021. *Machine learning*. Springer Nature.