Dog Emotion Classification with Transfer Learning

Xi Du, Wei Shan, Yuewei Wang, Tianrui Ye, and Jeff Zhuo
It's Ok To use an anonymized version of this report as an example of a great project for future classes.

Abstract

Sentiment analysis and Transfer learning are two popular fields in machine learning. However, the combination of them was rarely seen, moreover researches on animal sentiment analysis. Applying transfer learning techniques to classify dogs' emotion became the objective of this project. Most of the methods applied yielded a better result than the traditional machine learning model. Dog lovers could benefit from this project by better understanding their dogs' emotions and treating them in a more appropriate way.

1 Introduction

Dogs are known for being one of the most loyal and affectionate companions to humans. As dog lovers, we always want to understand our furry friends better and take care of them in the best possible way. One way to do that is by analyzing their emotions and behavior. In recent years, deep learning and transfer learning techniques have been widely used in image-based sentiment analysis and emotion classification tasks.

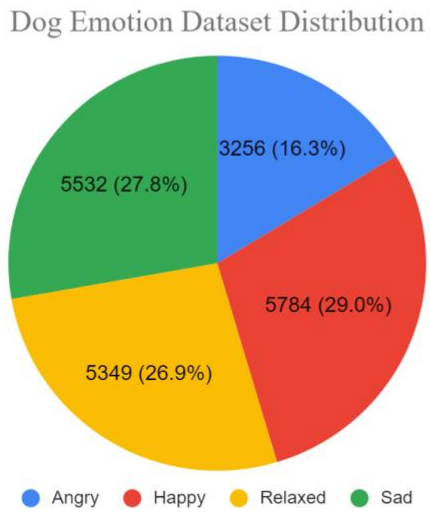
Several studies have explored the use of deep learning models for sentiment analysis, including image-based sentiment analysis, pretrained models, and transfer learning. For instance, Lei Zhang, Shuai Wang, & Bing Liu (2018) conducted a survey on deep learning for sentiment analysis. Mittal, Sharma, & Joshi (2018) proposed a deep learning-based approach for image sentiment analysis. Cetinic, Lipic, & Grgic (2019) used deep learning models to analyze beauty, sentiment, and remembrance of art. Additionally, pretrained models such as Deep Residual Learning for Image Recognition (He et al., 2015) and Very Deep Convolutional Networks for Large-Scale Image Recognition (Simonyan & Zisserman, 2015) have been used for various image classification tasks.

Our motivation for this study is our love for dogs, and we want to understand their emotions better. The objective of this study is to develop a transfer learning model for dog emotion classification. The dataset we obtained consists of four classes: angry, happy, relaxed, and sad. The proposed model can help dog owners better understand their dogs' emotions, leading to improved care and well-being.

2 Data

For our dog emotion classification task, our team will use a combined dataset from Kaggle (Dataset1 & Dataset2). This dataset consists of a total of 19, 921 dog images. Each image was labeled with one of the four emotion classes: angry, happy, relaxed, sad. There are about 5500 images for each emotion class, but the angry class only has about 3000 images. In consideration of our goal was to compare transfer learning techniques and imbalance was not extreme. Our team decided to move forward with the dataset unchanged.

[Figure 1: Distribution of Dog Emotion



Dataset]

[Figure 2: Images Classes: Angry, Happy, Relaxed, Sad]
Figure 2: Images from different classes taken from various angles and environment

3 Transfer Learning Methodologies

Using knowledge learned in one model to help the performance of another model is the core of transfer learning. Conventionally, transfer learning has three constraints: first, the task completed by the two models should be similar; second, the first model was trained on a much bigger dataset than the second model(Y.Zhang, A Survey).

[Figure 3: Transfer Learning]

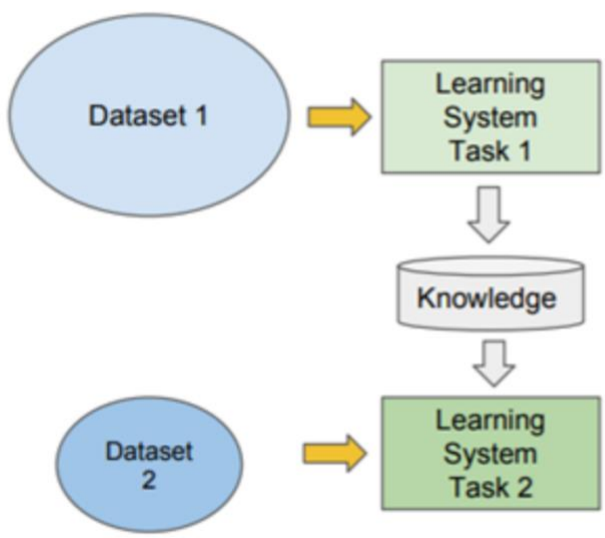


Figure 3: The knowledge in the learning system completing task 1 is transferred into the learning system completing task2. The first learning system was trained on a larger dataset than the dataset that the second learning system was trained on.

Projecting the constraints to this project, firstly, we need to find image classification pretrained models that are trained on a dataset with a size much larger than 10000, which is roughly the size of our dataset. Then, more transfer learning techniques could be applied to increase the performance of the model.

3.1 Pretrained Model Selection

Based on the constraints mentioned above, Residual Network (ResNet), VGG, AlexNet, MobileNet, and GoogLeNet were selected. They were trained by professional research groups on a large and well-maintained dataset, ImageNet. However, applying transfer learning techniques on the five models at the same time is time-consuming, one best model was proposed to be selected for implementing the transfer learning techniques. Because they are all convolutional neural networks(CNN) and their structures could be related, ResNet will be used to demonstrate the pretrained model selection process.

3.1.1 Pretrained Model Selection Process

All parameters before the final fully connected layer of the models were freezed, and the final fully connected layers were modified from output size of 1000 to 4, because ImageNet has 1000 classes and the dataset for this project has 4.

[Figure 4: Pretrained Model Selection Process on ResNet]

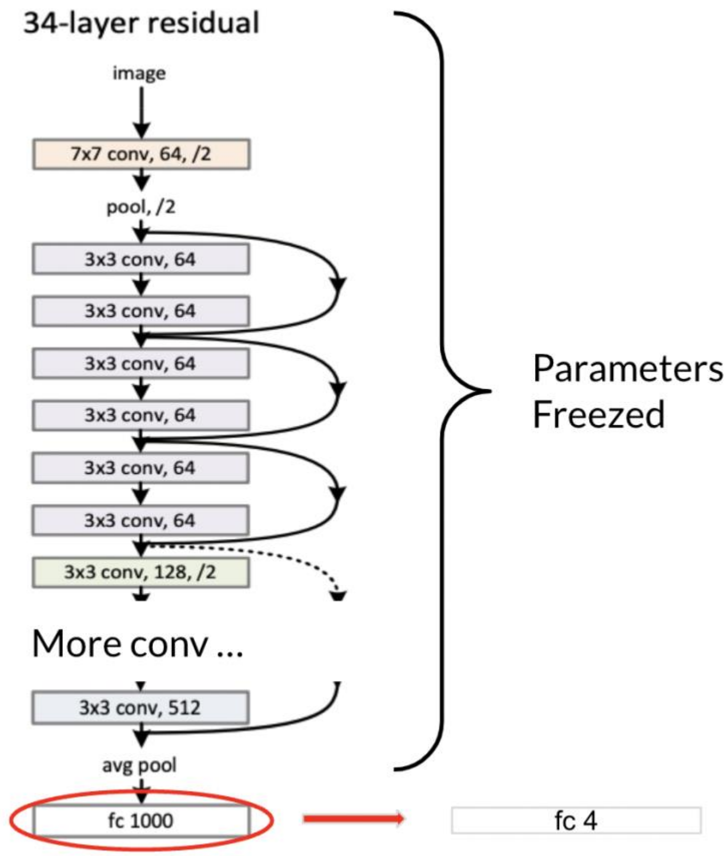


Figure 4: Convolutional layers were abbreviated as conv and fully connected layers are abbreviated as fc. The output size is indicated by the number at the end of each layer block. All parameters before the final fully connected layer are freezed. The output size is changed from1000 to 4 for the last layer.

After all five models were loaded as desired, they were trained by our dataset, classifying four dog emotions.

3.1.2 Pretrained Model Selection Result

10 epochs of training were performed and training accuracy was obtained to compare the performance of the models.

[Table 1: Training Set Accuracies of Pretrained Models]

Model	ResNet18	ResNet50	VGG16	AlexNet	MobileNet	GoogLeNet
Training Accuracy	44%	54%	58%	47%	51%	46%

Table 1: The first row shows the name of the pretrained model and the training accuracy is presented in the second row. Some of the models have different versions with different numbers of layers. For such models, the number of layers is indicated at the end of the name.

VGG16 yielded the best result. However, VGG16 is the largest model among all the other models. Therefore, VGG16 also yielded the longest training time. Therefore, the second best model ResNet50 is chosen as its accuracy was only 7% lower than VGG16 with much shorter training time.

Moreover, the 54% training accuracy was set as a benchmark for this project in order to evaluate whether the techniques shown later were effective.

3.2 Advanced Techniques

Having the selected pretrained model, ResNet50, the objective turned out to further increase the model's accuracy. Four transfer learning techniques were applied: Classifier Modification, Multi-task Learning, Fine-tuning, and Unsupervised Domain Adaptation.

3.2.1 Classifier Modification

In the simple transfer task done for pretrained model selection , the classifier was a linear fully connected layer. Alternatively, the last fully connected layer was replaced by the following four classifiers individually: Naive Bayes Classifier (LDA/ QDA), KNN, SVM, and Random Forest.

[Figure 5: Alternative Classifier: Naive Bayes Classifier (LDA/QDA)]

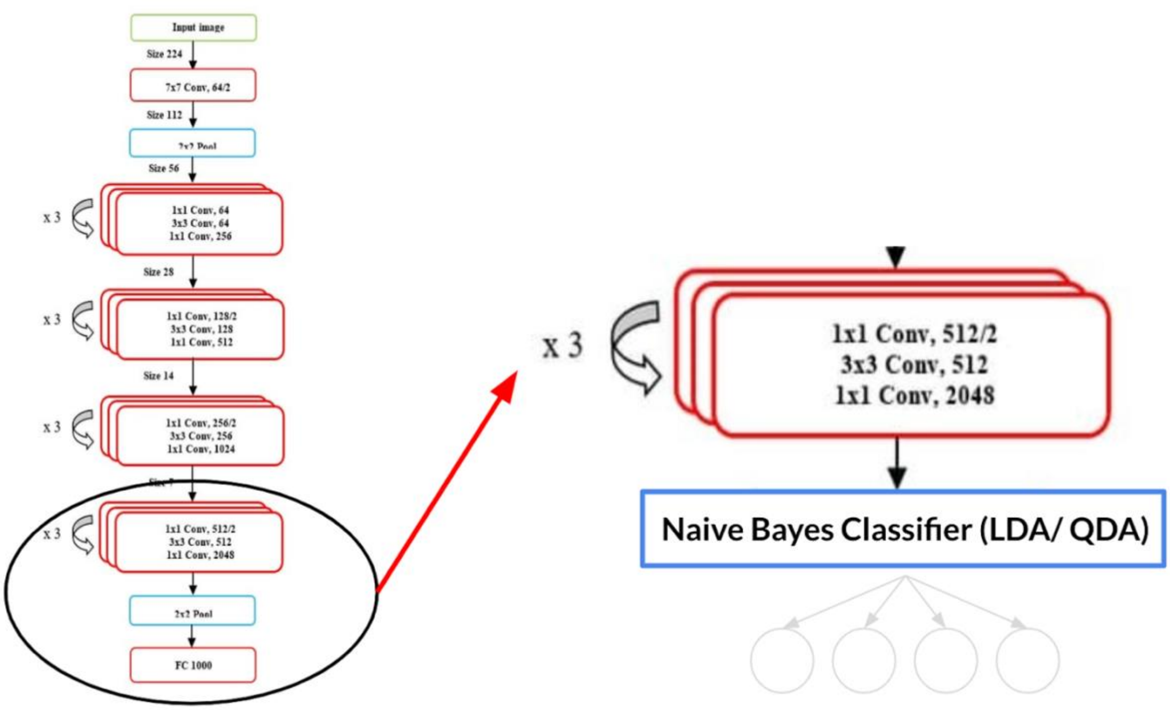


Figure 5: The original ResNet model is shown on the left. The part modified is circled and demonstrated on the right. The last convolutional layer block's output is fed into the Naive Bayes Classifier, and outputs four elements as the conditional probability of four classes.

3.2.2 Multi-task Learning

The second approach we adopted to further increase our accuracy is multi-task learning. Two related tasks could be trained in one model with shared early layers in order to obtain high accuracy(Y.Zhang, A Survey).

[Figure 6: Single Task Learning and Multi Task Learning]

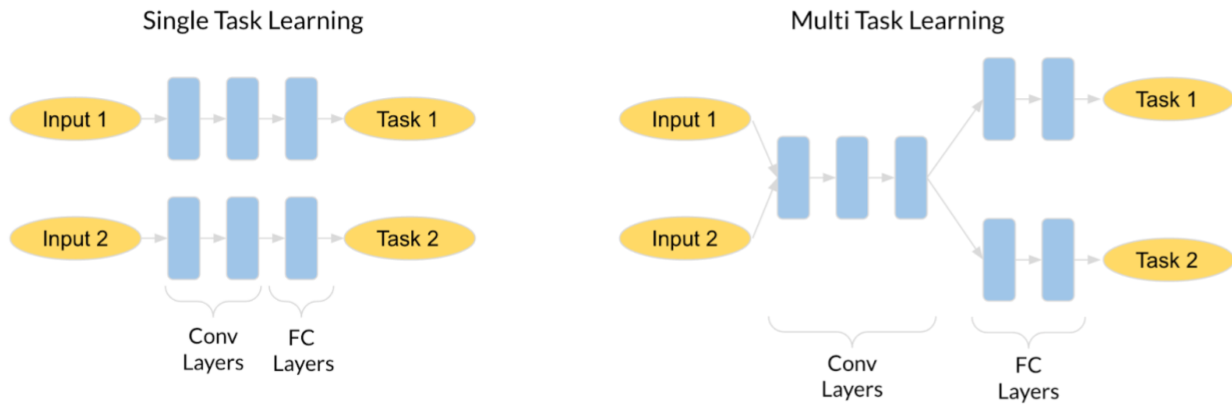


Figure 6: The traditional way of machine learning is shown in the left figure. Models were built for individual tasks. The multi task learning model is shown on the right. Convolutional layers are shared by the two different but related tasks and deviated in the fully connected layers in the end to complete different tasks.

Only one task is introduced in our project, which is to classify dogs' emotions into four classes. Therefore, another “dummy” task is created in order to fit the multi task learning model.

[Figure 7: Multi Task Learning Model]

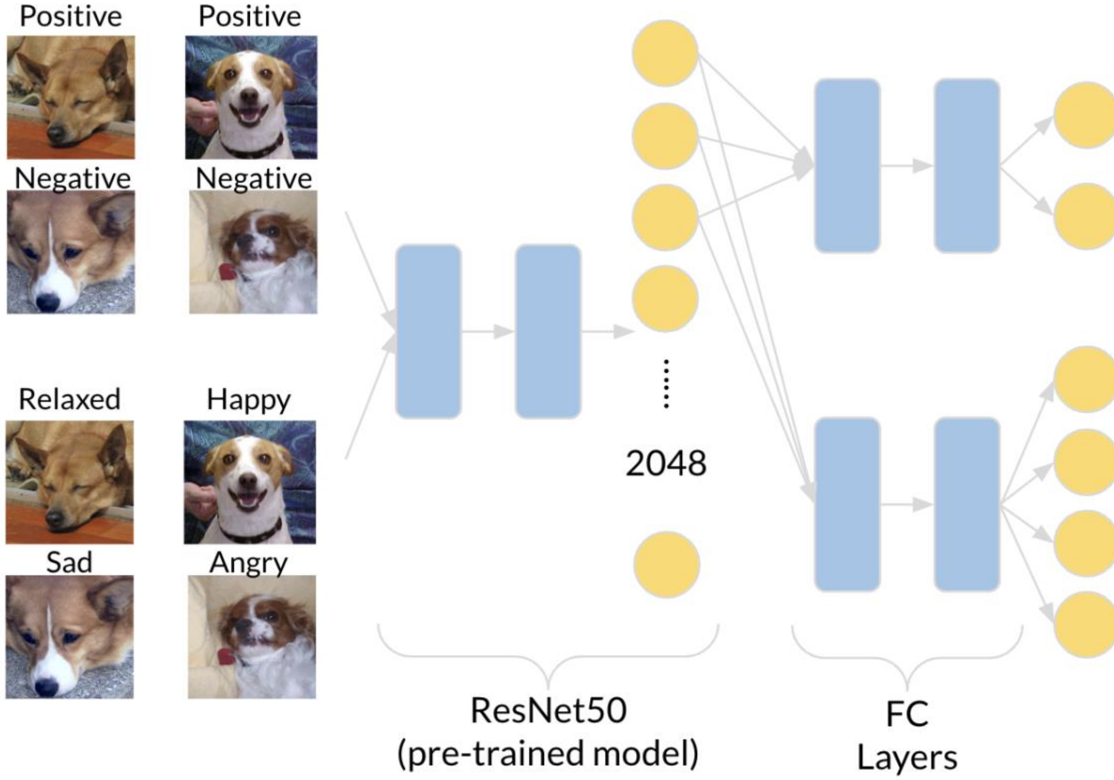


Figure 7: The “dumpy” task designed is classifying dogs emotion into two classes: positive and negative. The second task is the original task. The ResNet50’s convolutional layers is transferred into the model, with all weights kept. The 2048 features extracted are forwarded into two blocks of fully connected layers, with first block outputting two classes’ probabilities and second block outputting four classes’ probabilities.

The two tasks are considered as related because: first, they share a similar dataset; second, the classes in the second task are subclasses of the classes in the first task.
Relaxed Positive, HappyPositive, SadNegative, AngryNegative

3.2.3 Fine-tuning

Another popular transfer learning technique is called Fine-tuning. The main idea behind Fine-tuning is the realization that different models and datasets share similar low-level features (J.Yosinski, 2014). That means with a given pretrain model, it could achieve high classification accuracy on a new classification task while training only less than half of the layers. Models trained with the fine-tuning technique would require less computation and fewer data to achieve high classification accuracy.

[Figure 8: Fine-tuning with ResNet50]
Figure 8: Top shows model with fine-tuning applied; bottom shows pretrain model with no transfer learning technique applied.

This idea was applied to the ResNet50 model to improve its classification accuracy. In addition to applying the fine-tuning technique, the weights were initialized using the weights from the original pretrain model. According to Yosinski's study, transfer learning models initialized with weights from its pretrain counterpart would converge faster and lead to better overall accuracy (2014). The model resulted with 9.3% higher training and 4.3% higher testing accuracy than our benchmark score trained only on the classification layer.

3.2.4 Unsupervised Domain Adaptation

The last method in the research is Unsupervised Domain Adaptation. It is difficult and time-consuming to collect high-quality labeled data for emotional classification of dogs, especially as it involves subjective human judgments. Therefore, the effectiveness of traditional supervised learning methods may be limited as the resulting model may not generalize well to new, unseen data. To address this issue, unsupervised domain adaptation is used by adapting a model for the target domain using unlabeled data from the source domain. The basic idea is to learn a feature representation from the source domain that can be used in the target domain. This is achieved by training a model on the source domain and then fine-tuning it on a small amount of labeled data from the target domain. (Yaroslav, 2014)

The general process of unsupervised domain adaptation includes the following steps:

1. Feature extraction: Feature extraction is performed on the source domain data to extract a feature representation that can be used in the target domain. Popular feature extraction techniques include autoencoders, generative adversarial networks (GANs), and deep neural networks.
2. Domain adaptation: Domain adaptation is performed by adapting the model from the source domain to the target domain. This can be done using transfer learning techniques such as fine-tuning or by using domain adaptation methods such as adversarial domain adaptation.
3. Classification: Finally, the adapted model is used for classification on the target domain data.

In the case of dog emotion classification, unsupervised domain adaptation can be used to address the differences between different dog breeds or shooting scenes, which may cause variations in the emotional expression of dogs. By learning a transferable feature representation from the source domain and adapting the model to the target domain, the emotions of dogs in the target domain can be effectively classified, even with limited labeled data.

For feature extraction, the pretrained ResNet50 model was used, and the fully connected layer was removed to extract the 2048-dimensional feature representation for each image. A domain discriminator network was then trained to distinguish between the source and target domain features. The domain discriminator network is a binary classifier that takes the ResNet50 features as input and outputs a probability indicating whether the features come from the source or target domain. The domain discriminator network was trained using adversarial learning to encourage the feature representation to be domain-invariant.

To perform unsupervised domain adaptation, the domain discriminator network (DDN) algorithm was used, which involved training the feature extractor and domain discriminator network simultaneously. The objective function consisted of three terms: the classification loss, the domain adversarial loss, and the domain confusion loss. The classification loss was used to minimize the error of the emotion classification task on the labeled target domain data. The domain adversarial loss was used to maximize the error of the domain discriminator network in distinguishing between the source and target domain features. The domain confusion loss was used to minimize the error of the domain discriminator network in distinguishing between the source and target domain features.

During training, a batch size of 64 and a learning rate of 0.001 were used for the feature extractor and domain discriminator network. A learning rate of 0.1 was used for the classification layer, and it was fine-tuned on the labeled target domain data. The model was trained for 50 epochs using stochastic gradient descent with momentum.

4 Results

Four different transfer learning techniques were applied, and their results are shown in four subsections below.

4.1 Classifier Modification

The training and testing accuracies of models with different classifiers is shown below.

[Table 2: Classifier Modification Accuracy]

Model	FC	LDA	QDA	KNN	SVM	Random Forest
Training Accuracy	54%	79%	82%	48%	79%	61%
Testing Accuracy	53%	31%	35%	20%	59%	27%

Table 2: The first row shows the alternative classifier used. The training accuracy and testing accuracy is shown in the second and third row correspondingly.

Except KNN, all the classifiers had higher training accuracies than the benchmark(accuracy of FC), which is 54%. However, all classifiers (except FC) showed a sharp decrease in the testing accuracy, also known as overfitting issue. Among all, KNN performed the worst as its testing accuracy is lower than 25%, which is the expected accuracy of random guessing. SVM performed the best with a second best training accuracy and best testing accuracy.

4.2 Multi-task Learning

Unfortunately, the multi-task learning model didn’t yield satisfactory results. The best accuracy obtained so far is 35% for both training and testing accuracy. The result was better than random guessing but still much lower than the benchmark set earlier.

It’s still not clear what’s the cause of the issue because it’s very unlikely that the multi-task learning model will yield a lower training accuracy than the single-task model. However, a few possibilities had been excluded. Firstly, the code was likely to be correct. Codes written by myself, and codes on the Internet written by tutorial makers and researchers were all implemented on this task and yielded similar results. Second, the issue was most likely irrelevant to epoch size and hyperparameters. The maximum epoch size used is 50 which took almost a day to complete training. However, the accuracy was fluctuating around a low percentage. Also, a lot of learning rates from 0.0001 and 0.1 were tested and none of them showed a significant difference from the other. It's hypothesized that the cause of the problem could be the design of the second "dummy" task. However, experiments haven't been done to prove or disprove this statement.

4.3 Fine-tuning

Overall, the ResNet50 model results in an increase in both training and testing accuracy. There is also a 5% increase in test-accuracy loss after fine-tuning was applied. This phenomenon aligns with fine-tuning ideology and the test-accuracy loss is not extreme. Therefore, there is no major concern for overfitting for the ResNet50 model.

[Table 3: Fine-tuning with ResNet50]

Method	Training on FC Layer	Training on FC + Stage 4 CV Layers	Accuracy Increase
Training Accuracy	54.5%	63.8%	9.3%
Testing Accuracy	53.3%	57.6%	4.3%
Accuracy Difference	-1.2%	-6.2%	

4.4 Unsupervised Domain Adaptation

A training accuracy of 62.8% and a testing accuracy of 53% were achieved in the dog emotion classification task. Although the accuracy was not very high, it was 10% higher accuracy compared to the original pretrained model. The model was able to learn transferable features from the source domain data and adapt them to the target domain, which demonstrates the effectiveness of unsupervised domain adaptation in addressing the domain shift problem.

Several reasons could explain the relatively low testing accuracy of the UDA model. Firstly, it can be attributed to the dataset not being big enough, which may have limited the performance of the model. Secondly, the feature extraction and domain adaptation techniques used in the UDA model may not have been optimal, and more advanced techniques may lead to better results. Finally, the emotional expressions of dogs can be affected by various factors such as breed, age, and context, which may increase the difficulty of the classification task.

5 Discussion

The results of the four techniques implemented showed that the simplest method that changed the classifier to SVM produced the highest accuracy. It's possible that a simpler model could perform better than a complicated model on some simple tasks. However, this statement was made arbitrarily. Some issues remained unfixed regarding this project due to the time limit.

5.1 Data Quality

The data was obtained on Kaggle and weren't well-maintained. Therefore, a lot of data labeling is ambiguous. Dogs' emotions were decided by humans subjectively.

[Figure 9: Ambiguous Dog Emotion Labeled Angry]

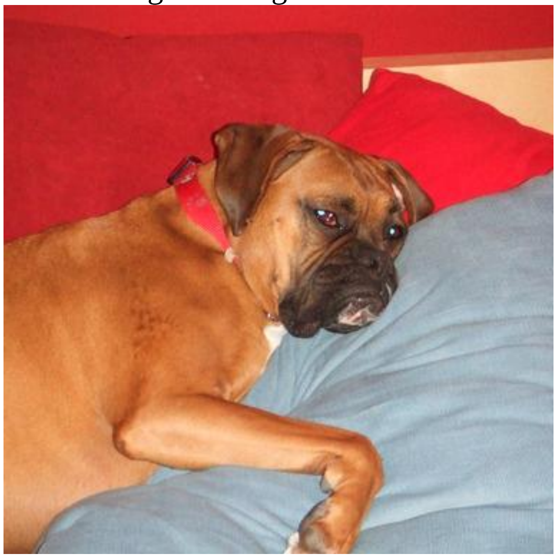


Figure 9: This picture was labeled angry but could also be labeled as sad.

Moreover, dogs sometimes weren't the majority of the picture or multiple dogs were shown in one picture.

[Figure 10: More Bad Data Examples]

Figure 10: The left picture shows an example of a dog that isn't the majority of the picture. The right picture shows two dogs in one picture.

Most of the data existing in our world isn't well-maintained. How to clean those subjectively labeled data and train the model under a not well-maintained dataset could be a topic for the future experiment.

5.2 Intermixed Transfer Learning

All the transfer learning techniques except multi task learning yielded accuracy higher than the benchmark. Will implementing them all together in one model produce a much better result? Combining some/all of the four techniques shown above could be another potential future experiment.

7 Conclusion

In conclusion, this study aimed to develop a deep learning model for dog emotion classification using transfer learning techniques. Our motivation for this study was our love for dogs, and we wanted to understand their emotions better to provide them with better care.

We first introduced five popular pretrained models. We then selected the ResNet50 model and applied different transfer learning techniques, including classifier modification, multi-task learning, fine-tuning, and domain adaptation.

Our results showed that all the transfer learning techniques improved the training accuracy of the ResNet50 model. Specifically, we observed a 9.8% increase in training accuracy. However, the testing accuracy was 53%, indicating that there is room for improvement in our model.

As a takeaway, this study demonstrated the effectiveness of transfer learning techniques for dog emotion classification. These techniques can be applied to other image classification tasks to improve the model's accuracy. By better understanding our dogs' emotions, we can provide them with better care and improve their overall well-being.

References

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, & Hartwig Adam. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Cetinic, E., Lipic, T., & Grgic, S. (2019). A Deep Learning Perspective on Beauty, Sentiment, and Remembrance of Art. IEEE Acce

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, & Andrew Rabinovich. (2014). Going Deeper with Convolutions.ss, 7, 73694-73710.

Jason Yosinski, Jeff Clune, Yoshua Bengio, & Hod Lipson. (2014). How transferable are features in deep neural networks?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015). Deep Residual Learning for Image Recognition.

Karen Simonyan, & Andrew Zisserman. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks.

Lei Zhang, Shuai Wang, & Bing Liu. (2018). Deep Learning for Sentiment Analysis : A Survey.

Mittal, N., Sharma, D., & Joshi, M. (2018). Image Sentiment Analysis Using Deep Learning. In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI) (pp. 684-687).

Y. Zhang and Q. Yang. A Survey on Multi-Task Learning.

Yaroslav Ganin, & Victor Lempitsky. (2015). Unsupervised Domain Adaptation by Backpropagation.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, & Victor Lempitsky. (2016). Domain-Adversarial Training of Neural Networks.

Team Distribution

Xi Du: Classifier Modification and Transfer Learning Methodologies
Wei Shan: Multi-task Learning and Discussion
Yuewei Wang: Abstract, Introduction and Conclusion
Tianrui Ye: Pretrain Model Selection and Unsupervised Domain Adaptation
Jeff Zhuo: Data and Fine-tuning