BEER LABEL CLASSIFICATION AND ADAPTIVE STRATEGIES FOR WORKING WITH SMALL DATA

Gordon Irving
March 14, 2021

Keywords:

Small Dataset, Transfer Learning, Feature Extraction, Convolutional Neural Networks,

Abstract

Small datasets are the reality in many real-world applications. This presents a serious challenge to the practicability of convolutional neural networks, which require large data sources to properly train. This paper investigates methods of transfer learning for use on small-scale datasets in computer vision applications. The dataset used here consists of 400 images of beer labels divided evenly across two representational classes. Two key features of this dataset make it difficult for convolutional neural networks to work with; its small size, and its intra-class representational diversity (two members of the same class may be stylistically very different). Because small data require compact, well regularized models to avoid overfitting (Chollet, 2018, 130), this dataset constrains classifier size. However, intra-class diversity and the presence of subtle class features demands higher level feature representation. Experimentation begins by establishing the relative difficulty of this dataset, where simple convolutional neural networks capable of 82% or better on the well-known cat-dog classification problem show results no better than 55-60% here. Next, through data augmentation, feature extraction and fine-tuning techniques in transfer learning, we construct several models capable of classification accuracy up to 80%, even on this very challenging dataset.

Introduction

This paper evaluates the results of several transfer learning methods, including feature extraction and fine tuning with data augmentation, on a challenging dataset of small size with high intra-class diversity and subtle class features. Using 400 images of different beer labels

handpicked for the purpose of presenting significant challenges to traditional CNN classification methods, we establish a baseline total accuracy of only 55-60% for a basic convolutional model. The ability of this admittedly simple model to classify images from the well-known cat-dog classification problem at 82% accuracy or better underscores the relative difficulty of our current dataset.

Next, we demonstrate how accuracy of up to 80% is achievable using data augmentation and transfer learning techniques including feature extraction and fine-tuning. Pretrained models including VGG16, Xception, and MobileNet are all evaluated based on performance with this dataset. This paper concludes with a discussion of key differences between each of these models and an attempt is made to uncover essential differences in feature representation relevant to the current dataset.

Literature Review

The recent literature is replete with work on the topic of adapting CNN architectures to operate in diverse circumstances with minimal data. One popular method of grappling with scarce data is to harness generative adversarial networks for the purpose of producing new observations with which to pad the training data. This approach has shown to be useful in credit card fraud detection where the extreme imbalance of classes between fraudulent and non-fraudulent transactions creates the need for a great deal of observations (Fiore, De Santis, Perla, Zanetti, and Palmieri, 2019). It has proven vital in certain facial recognition tasks as well, where even databases of 60,000 images are considered relatively small and can be difficult to train on without augmentation from artificially generated images (Saez, Ming, and Hartnett, 2021). Although recent work has shown promise for more data efficient GAN training (Zhao, Lie, Lin

Han, and Zhu, 2020) GANs remain impractical for datasets with only a few hundred examples and high levels of intra-class diversity.

Transfer learning, the art of repurposing models which have been pre-trained on abundant data sources to suit new applications in data-exiguous circumstances via fine-tuning or feature extraction, is another popular approach to grappling with the problem of minimal data. This method has shown to be successful in such disparate areas as battery capacity estimation, where training on minimal examples is computationally desirable (Yihuan, Kang, Xuan, Yanxia, and Zhang, 2021) and tumor classification, where the availability of public datasets may be minimal (Kim et.al, 2020). Transfer learning methods have also been successful while training on minimal examples in the identification of rice plant disease (Chen, Nanehkaran, Zhang, and Zeb, 2021) and many other applications.

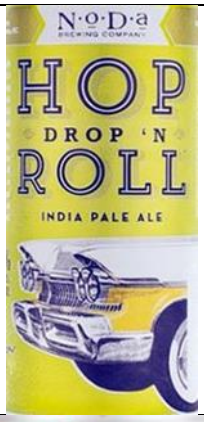
The advantages to leveraging higher level feature extraction from models trained on ImageNet or other databases is well established, and there now exists an abundance of pre-trained models to choose from, each with distinct advantages. The speed, efficiency, and portability of MobileNet, for example, makes it a desirable transfer learning candidate for such tasks as object detection in autonomous vehicles (Carranza-Garcia, Torres-Mateo, Lara-Benitez, and Garcia-Gutierrez, 2021). Bulkier models, such as ResNet50 and VGG16 provide value as the basis for feature extraction in applications where processing costs are less of a concern, such as detecting signs of bleeding in the digestive tract using endoscopic images (Caroppo, Leone, and Siciliano, 2021). Whatever the application, subtle differences between models can lead to big differences in classification performance. When engaging in transfer learning for the purpose of solving small data classification problems, understanding these differences is vital to success.

Data

Beer label images provide a rich and diverse data source for the project at hand. This dataset was collected via Google Image search and contains 400 images of beer can labels divided equally into representations of animate entities and representations of inanimate entities. Image selection was done with the specific intention of developing a challenging dataset with significant intra-class diversity. Where possible, similar styles were paired across classes. This was done with the intention of minimizing potential for stylistic elements inherent in certain brands from becoming identifiable as differentiators of class. Image size and quality are variable by design.

Here the animate class includes both realistic and cartoon-like depictions of human beings, animals, and mythical creatures. Only those creatures with identifiable facial features are included (no shellfish, for example). Some labels included in the animate class depict creatures which are common to the class (such as people and dogs). Others in the animate class are unique instances which represent the sole example of their species in the dataset (only one image of a turtle, for example). Among the images representing the inanimate class, a mix of landscapes, buildings, vehicles, and geometric patterns can be found. Figure 1.1 below illustrates a few examples from each class.

Figure 1.1

Animate Class	Inanimate Class
	
	

Methods

To evaluate the performance of transfer learning methods on this dataset, we compare three pre-trained convolutional models; Xception, MobileNet, and VGG16, run end to end on an augmented data generator which stretches, rotates, and inverts images in order to maximize the very small amount of image data available. In each case, we leverage the higher-level representations generated by these more complex, pre-trained models via feature extraction and fine-tuning. These features feed forward into an untrained classifier consisting of two hidden layers with dropout, and binary output, which sits on top of the convolutional base of each pretrained model.

Contrary to common advice regarding fine-tuning, which suggests training the classifier prior to unfreezing the convolutional base (Chollet, 2018, 150) we show preferable results when unfreezing the convolutional bases of transfer learning models and permitting them to train along with the classifier. This method runs the risk of destroying the higher-level features generated by each convolutional base but provides excellent results on this dataset when paired with a very low learning rate.

To avoid overfitting, which happens quickly on a dataset this size, training epochs have been limited to just 10 during fine tuning, followed by five more epochs which run while the convolutional bases are completely frozen. Training takes place on 320 examples (160 examples of each class) with testing being done on the remaining 80 (40 examples in each class). Because the training and test datasets are divided equally into animate and inanimate classes, total classification accuracy is an appropriate measure of model performance and is used to evaluate model success in this case. Finally, results for each model are compared side by side. Results of feature extraction and fine-tuning are discussed and an attempt is made to illuminate why some model architectures are better suited to this particular dataset.

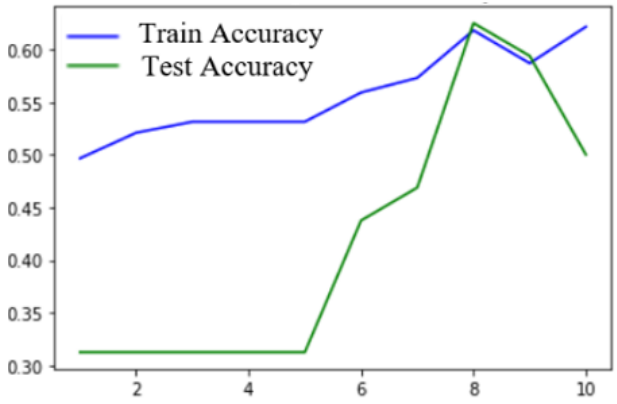
Results

All model architectures presented in this section ran a minimum of five separate experiments on the training and test data. The results presented here are representative of the median test accuracy outcomes for each model except the basic CNN, for which the trial run with the best test accuracy results is presented. As can be seen in figure 1.2, the basic convolutional neural network reached peak accuracy near 62% before promptly crashing. All trials for this basic model exceeding 10 epochs led to significant overfit. No other trial run for this model exceeded the 62% accuracy reached during the eighth epoch of the trial presented

here. Most trials for this model topped out in the 55 – 59% range and typically this took place at or near the eighth epoch with test results subsequently crashing due to overfit.

Figure 1.2

Baseline CNN Classification Accuracy

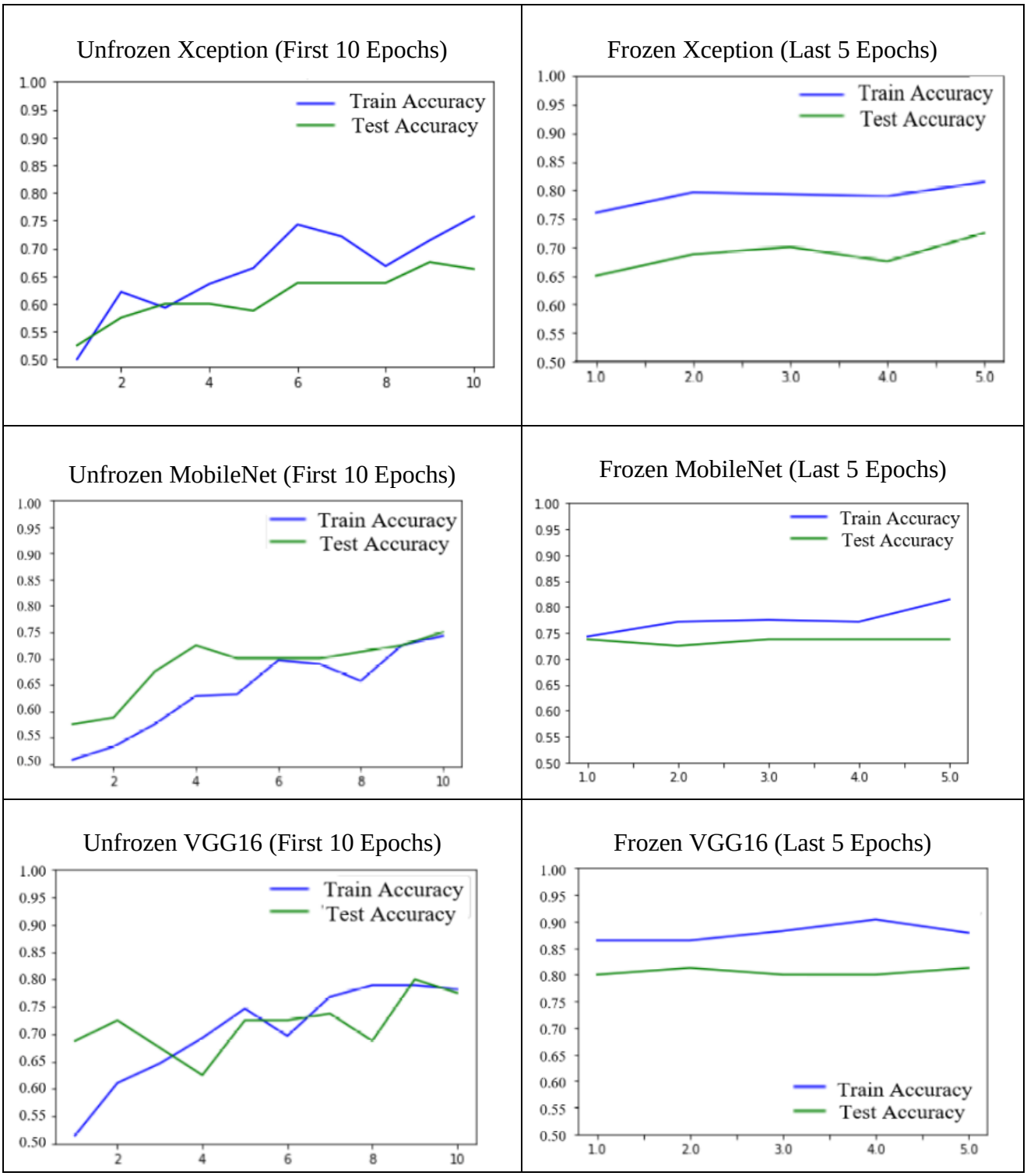


Moving on to the results of other models, please note that the scale used in the image above is unique. This scale adjustment for the basic model was done in order to fully capture the range of results. Every model from here forward uses a uniform scale between 0.5 and 1.0 for accuracy.

Figure 1.3 below displays results for each of our three transfer learning models (Xception, MobileNet, and VGG16). This figure also separates results for the first ten epochs from the last five. This has been done in order to distinguish the progression of each model while fully trainable (the first ten epochs) and while trainable only for the classifier portion with convolutional base completely frozen (the final five epochs). What permits this strategy to work well is the low learning rate (1e-05, or 0.000001). Training with unfrozen convolutional bases here permits the flexibility necessary for each pre-trained model to adapt to the idiosyncricies of this highly diverse dataset. The low learning rate and minimal number of training epochs ensure

that this fine-tuning process does not destroy the feature representations which make these pretrained convolutional bases desirable in the first place.

Figure 1.3



The first model, which employs an Xception CNN base, routinely reaches 65% accuracy during the first ten epochs and tops out at or near 70% during the final five. The MobileNet based model does a bit better, reaching 70-73% test accuracy during the first ten epochs and topping out near 75% during the final five. The last model, which uses the VGG16 convolutional base, is the best of the three. It routinely reaches 80% accuracy within the first ten epochs, occasionally reaching close to 83% in the final five as the classifier learns more about the feature representations it recieves from the base.

Analysis and Interpretation

Our VGG16 based model significantly outshines the competition by reaching and sustaining 80% classification accuracy on the total dataset. Figure 1.4 below shows three images from our test set as well as how each model classifies them, with green indicating correct classification and yellow incorrect. Among the three models, VGG16 correctly classifies two images, with MobileNet and Xception each correctly guessing only one of the three.

Figure 1.4

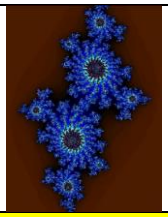
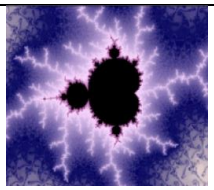
Image			
VGG16 Classification	Animate	Inanimate	Animate
MobileNet Classification	Inanimate	Animate	Inanimate
Xception Classification	Inanimate	Inanimate	Animate

What is notable about these images is how each one resists classification slightly differently. The first image is an illustration of a squid. The eye on this creature is small and hard to pick out, making it a challenging image if the classifying model relies on the finding of a face

in order to identify the image as belonging to the animate class. Only the VGG16 model accurately classifies this image. The second image is an illustration of a video game controller, the shape of which gives the general impression of eyes. VGG16 and Xception both correctly identify this as being inanimate, but the shape appears to fool MobileNet. Finally, the third image is of a Mandlebrot fractal. Only MobileNet correctly identifies this image as inanimate, which brings us to a key finding regarding shape complexity and the role fractal dimensions play in the distinction between animate and inanimate.

VGG16 is the only model to correctly identify both the cartoon squid as animate and the video game controller as inanimate, but it incorrectly classifies the Mandelbrot fractal. One possible explanation for this is the difference in complexity between organic structures (many of which are living) and manufactured structures. The relative simplicity of straight lines in human manufactured things makes them distinct from patterns occurring in nature. Neural networks are capable of leveraging fractal dimension in such tasks as histological classification (Roberto, Lumini, Neves, and Nascimento, 2021) and it is reasonable to assume that shape complexity or fractal dimension could be playing a role here too. To explore how fractal dimension might be playing a role in the success of our VGG16 classifier, we turn to images from outside the current dataset. Here again, green indicates correct classification and yellow incorrect.

Figure 1.5

Image				
VGG16 Classification	Animate	Animate	Animate	Animate
MobileNet Classification	Inanimate	Inanimate	Inanimate	Inanimate
Xception Classification	Animate	Animate	Inanimate	Inanimate

The first two images are fractals: the Julia set, and Mandelbrot set, respectively. Our VGG16 and Xception based models both classify these as belonging to the animate class, which is inaccurate. The second two images are of microscopic organisms. Only VGG16 correctly identifies these as animate entities, likely due to the complexity of their shape. It is worth noting that Xception performs the worst on these images, classifying the fractal sets as animate and the micro-organisms as inanimate. MobileNet sees nothing animate in any of the images. Figure 1.6 below illustrates how these classifiers perform on a different set of images. This one consists of human manufactured items; toy robots and cars with identifiable faces.

Figure 1.6

Image				
VGG16 Classification	Inanimate	Inanimate	Animate	Inanimate
MobileNet Classification	Animate	Animate	Inanimate	Inanimate
Xception Classification	Animate	Animate	Animate	Animate

Because the true problem these classifiers were trained to solve was whether the images were *representations* of animate or inanimate things, (not whether they were alive in a strict sense) the correct classification of these images is open to interpretation. In the case of the toy robots, they roughly imitate human form. In the case of the cars, there is an obvious representation of a face. Correctness is not of interest here. What is of interest is *how* each model sees these images. VGG16 sees only one animate image out of the four and it is the image with

fewer straight lines. The boxy looking robots do not register for VGG16 as animate. Both the MobileNet and Xception based models see the robots as animate. Xception sees the same in the cars, but here MobileNet draws a distinction that the cars are inanimate. Clearly the key for our best model, the VGG16 based model, is shape complexity. Even when a face is visible, such as with the toy robots in figure 1.6, the straight lines and boxy shape make our VGG16 based model classify the image as a representation of an inanimate object.

Conclusions

When working with datasets of only a few hundred images and high intra-class diversity, untrained CNN models are untenable. However, transfer learning and data augmentation techniques serve as powerful tools for working with small data, even when class distinctions are abstract and intra-class diversity is high. Choosing the proper pre-trained model for feature extraction and transfer learning purposes is vital. Not all models will be equally well suited to every task.

The nature of the classification problem examined here was complex. Identifying an abstract element of representation (animate or inanimate) among a small set of images with high intra-class diversity proved to be insurmountable for our untrained convolutional neural network. Through transfer learning and data augmentation, we were able to reach 80% classification accuracy on a consistent basis. Critical to reaching this level of accuracy was the ability of our VGG16 based model to identify the relationship between shape complexity and class. However, this left our VGG16 based model vulnerable to misclassifying images with high levels of complexity, such as the Mandelbrot and Julia sets. Had this set of images contained more representations of naturally occurring inanimate objects, such as lakes, rivers, and trees, which present more complex shapes, we likely have seen very different results. In this instance

however, the consistent themes present in our dataset made the VGG16 based model noticeably more successful than the competition.

Directions for Future Work

One method left unexplored in this paper is that of using an ensemble classifier. As the results show, each of the three models goes about classifying images in a different manner. It is conceivable that the vulnerability of our VGG16 based model to misclassifying fractal images as animate entities could be overcome by including the input of one or both of the other two models. Likewise, the tendency of the Xception based model to misclassify images in which faces and eyes appear to exist on inanimate things, could be overcome by including the input of one or both of the other two models. An ensemble method would likely improve overall classification accuracy.

Reference List

Caroppo, Andrea, Alessandro Leone, and Pietro Siciliano. 2021. "Deep Transfer Learning Approaches for Bleeding Detection in Endoscopy Images." Computerized Medical Imaging & Graphics 88 (March): N.PAG. doi:10.1016/j.compmedimag.2020.101852.

Carranza-García, Manuel, Jesús Torres-Mateo, Pedro Lara-Benítez, and Jorge García-Gutiérrez. 2021. "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data." Remote Sensing 13 (1): 89. doi:10.3390/rs13010089.

Chollet, Francois. 2018. *Deep Learning With Python*. New York: Manning Publications Co.

Chen, Junde, Defu Zhang, Yaser A Nanehkaran, and Dele Li. 2020. "Detection of Rice Plant Diseases Based on Deep Transfer Learning." Journal of the Science of Food & Agriculture 100 (7): 3246–56. doi:10.1002/jsfa.10365.

DeepLearningAI. 2017. "Transfer Learning (C3W2L07)." YouTube Video, 11:18, August 25, 2017. https://www.youtube.com/watch?v=yofjFQddwHE&ab_channel=DeepLearningAI

Fiore, Ugo, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. 2019. "Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection." Information Sciences 479 (April): 448–55. doi:10.1016/j.ins.2017.12.030.

Kim, Young-Gon, Sungchul Kim, Cristina Eunbee Cho, In Hye Song, Hee Jin Lee, Soomin Ahn, So Yeon Park, Gyungyub Gong, and Namkug Kim. 2020. "Effectiveness of Transfer

Learning for Enhancing Tumor Classification with a Convolutional Neural Network on Frozen Sections." Scientific Reports 10 (1): 1–9. doi:10.1038/s41598-020-78129-0.

Li, Yihuan, Kang Li, Xuan Liu, Yanxia Wang, and Li Zhang. 2021. "Lithium-Ion Battery Capacity Estimation — A Pruned Convolutional Neural Network Approach Assisted with Transfer Learning." Applied Energy 285 (March): N.PAG. doi:10.1016/j.apenergy.2020.116410.

Roberto, Guilherme Freire, Alessandra Lumini, Leandro Alves Neves, and Marcelo Zanchetta do Nascimento. 2021. "Fractal Neural Network: A New Ensemble of Fractal Geometry and Convolutional Neural Networks for the Classification of Histology Images." Expert Systems with Applications 166 (March): N.PAG. doi:10.1016/j.eswa.2020.114103.

Sáez Trigueros, Daniel, Li Meng, and Margaret Hartnett. 2021. "Generating Photo-Realistic Training Data to Improve Face Recognition Accuracy." Neural Networks 134 (February): 86–94. doi:10.1016/j.neunet.2020.11.008.

Zhao, Shengyu, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. 2020. "Differentiable Augmentation for Data-Efficient GAN Training." *arXiv e-prints*: arXiv-2006.