ISE 6740 – Summer 2021
Project Final Report

Team Member Names: Teresa Dong, Henry Dong

Project Title: *Comparison of Classic ML Algorithms to Identify Dog Breeds in Images*

Table of Contents

Background

Problem Statement

In 2020, as the coronavirus pandemic began sweeping across the nation, something peculiar happened: dog shelters across America began running out of dogs for adoption. Isolated and alone, mankind was once again saved from its own misery by mankind's best friend. Yet even more so than finding a dog, one of the biggest challenges first-time dog owners faced was the question: what's the right dog breed for me?

While every dog is unique, each dog breed generally shares certain characteristics that make its members more or less suitable for different owners. Some breeds, such as Newfoundland, grow to enormous sizes that make them unsuitable for urban pet-lovers. Other breeds, such as Afghan Hounds, are notoriously disobedient and hard to train, and therefore a bad choice for first-time dog owners. Therefore, it's imperative for first-time dog owners to ensure the dog they're thinking about adopting is a breed suitable for their needs.

The most obvious solution for this problem would be for prospective owners to order a DNA test for each dog of interest. Yet at well over $100 a test with two to four weeks lead time before receiving results, DNA tests are all but the most dedicated prospective buyers, especially in a market where dogs are often snatched up as soon as they become available. This raises the question: is there a way prospective dog owners can *visually* identify a given dog's dominant breeds?

With over 190 AKC-recognized (American Kennel Association) distinct dog breeds, and thousands more mixed-breeds, the simple answer is no: visual identification of a dog's dominant breed(s) is often a difficult task for even trained dog experts, much-less untrained first-time owners. This is especially true for both shelter dogs, who typically don't have well-documented genealogies, as well as designer dogs, whom unscrupulous breeders often try to pass off as those of more valuable, trending breeds to unsuspecting buyers.

Through the power of image classification, we address this problem of identifying dog breeds by appearance and help more people find their new best friend.

Data Source

The dataset we use for this analysis is the Stanford Dogs Dataset (Khosla et al., 2011). The dataset itself is an example of fine-grained classification, where there are likely smaller differences among dogs between breeds than there are within a given breed. This provides a much more difficult scenario for image classification. Students at Tsinghua university even created a new dataset Tsinghua Dogs Dataset that they claim has even more diversity within breeds (Zou et al., 2020). However, due to the availability of data, we will still be using the Stanford Dataset which is easily available on Kaggle (Li, 2019) (Li, 2019).

The Stanford Dogs dataset is 20,580 images and 20,580 annotations over 120 breeds. For each image, there is an annotations file that lists the size of the image, breed of the dog, and bounding boxes of the dog in the image and the folder which we will use as the class labels. We will be using the images as an input and annotations file to crop the image as seems to be the preferred way in many Kaggle notebooks for our image classifier (Devang, 2021).

Methodology

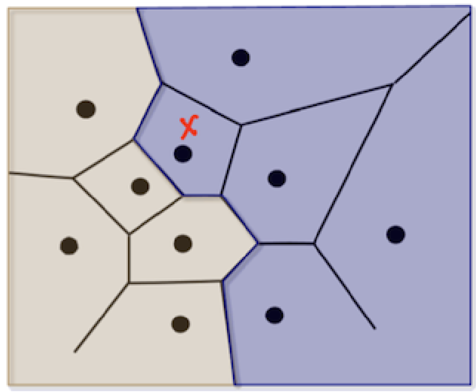
Convolutional Neural Networks (CNN) is widely cited as the preferred algorithm for Image Classification. Yet, with most implementations of CNN in industry typically being built atop pre-trained, ImageNet-based models, a large visual database of over 14 million hand-annotated images, it’s fair to wonder how much of this performance superiority is due to the superiority of the ImageNet training set, and how much of it is due to the superiority of the CNN technique itself. To explore this question we explore the performance of cold-start CNN, which we did not cover in class, against the performance of the various ML techniques we did cover in class (KNN, Logistic Regression, SVM, Random Forest and Neural Networks). In addition, we also compare the performance of these techniques against the conventional best-practice pre-trained CNN technique (transfer learning). Evaluation of these techniques and their resultant will be in terms of accuracy and time to train the model.

Classic ML Algorithms

We start our comparison by building models with the classic ML algorithms covered in our course. Starting with these more familiar approaches helped us establish a strong performance baseline with which to compare our CNN techniques against. Multi-class classification algorithms evaluated in our comparison are: KNN, Logistic Regression, SVM, Neural Networks and Random Forest. We begin by applying these algorithms without any feature engineering, e.g. on normalized raw image pixels themselves.

KNN

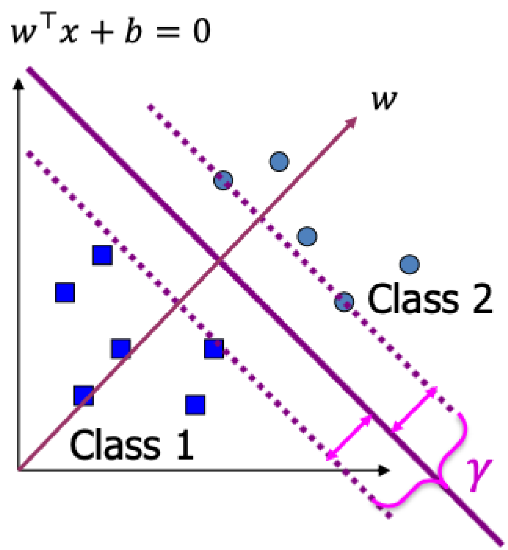
K-Nearest Neighbors (KNN) is very simple to understand and easy to implement. To classify a new data point, it takes the majority vote over the K training points closest to x. It is non-parametric, so there are no assumptions and no models to be built. In terms of disadvantages, it’s speed tends to decrease as the dataset grows and is sensitive to imbalanced classes and outliers (Genesis, 2018).

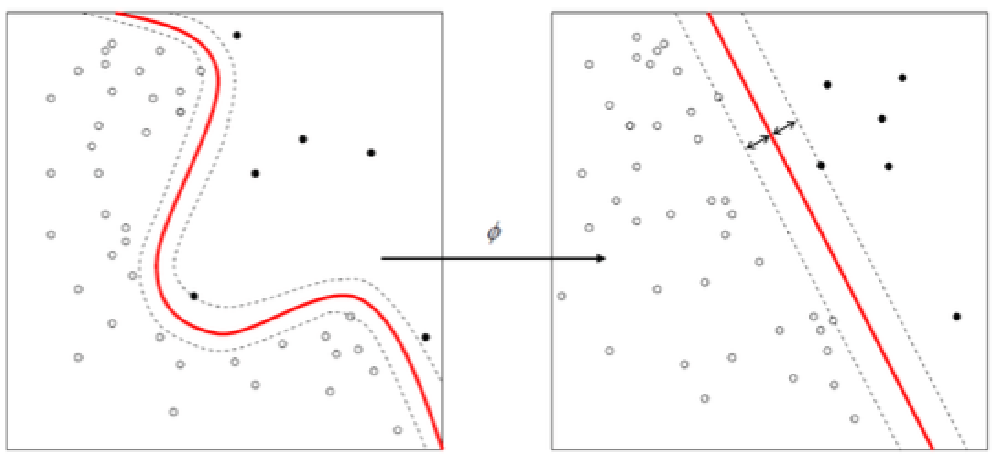


$$\underbrace{f(x)}_{\text{predicted label for test point}} := sign\left(\sum_{i \in I_k(x)} y^i\right)$$

SVM

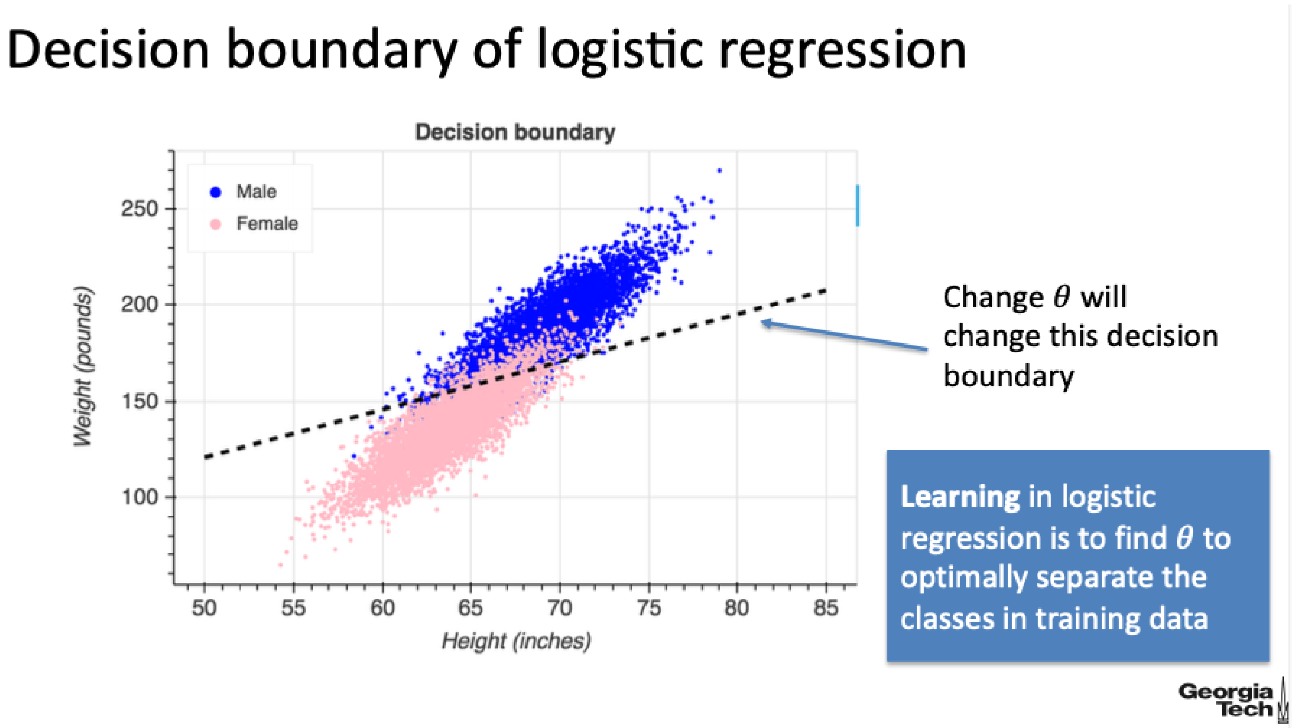
Similar to KNN, the Support Vector Machine (SVM) is another geometric intuition classifier. However, instead of examining a given point’s K neighbors, SVM tries to maximize the distance between the closest points to the margin. Using kernels, we can make the decision boundary non-linear. However, SVM really works better when classes are more linearly separable, so we don’t expect SVM to perform well with our problem.





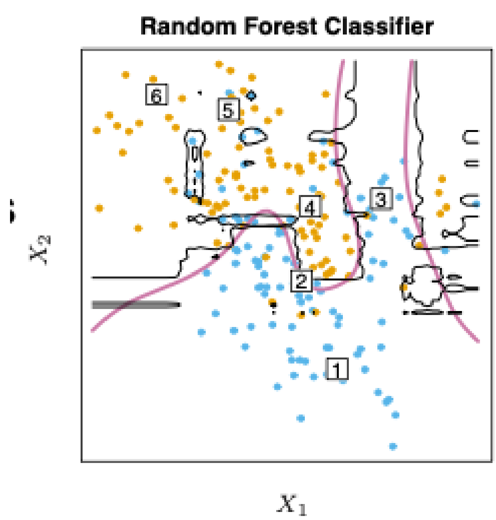
Logistic Regression

Logistic regression focuses on maximizing the probability of the data. It can work better for classes that are not clearly separable (e.g. SVM) and are easy to interpret due to the presence of coefficients. However, it has some stronger assumptions. The decision boundary is assumed to be linear, and there shouldn't be multicollinearity between variables. Since the images are likely non-linear, we don't expect our logistic regression model to perform well for our data.



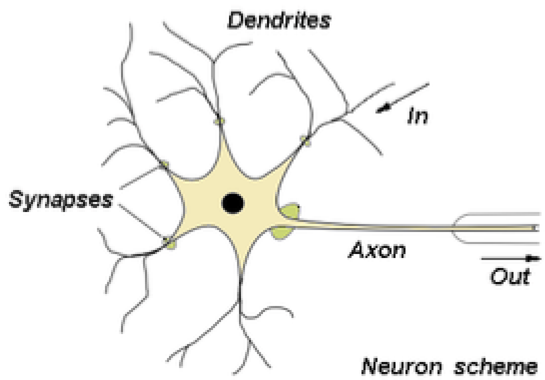
Random Forest

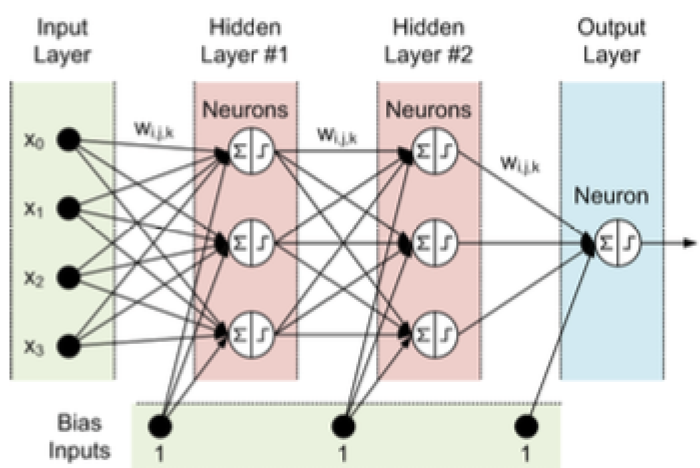
Random Forest (RF) is a supervised learning method that uses Decision Trees as building blocks. It bootstraps samples from the training data and then grows a tree for each batch of bootstrapped samples. Finally, it combines the samples by taking a majority vote across trees (for classification). RF is robust to outliers, doesn't need the assumption of linearity, and runs efficiently on large datasets. However, RF are slow to train and hard to interpret, as the method doesn't output coefficients like regression-based methods would.



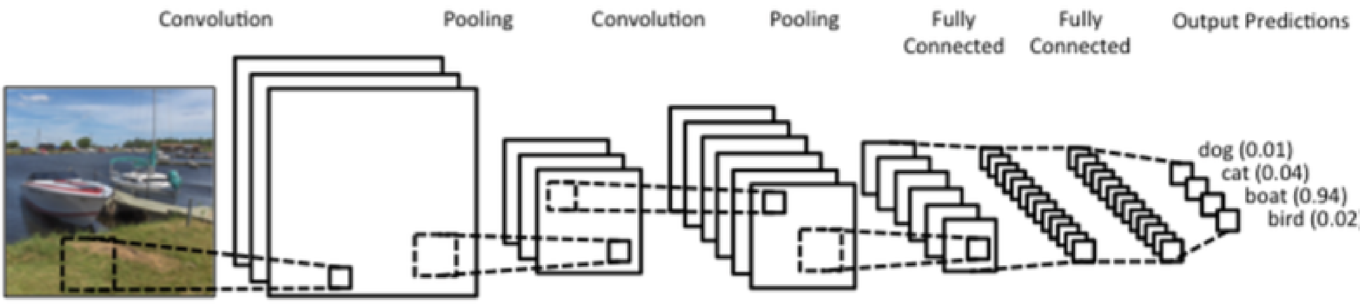
Neural Networks

Neural networks (NN) are an artificial way to represent actual biological neurons (see comparisons below). A neural network is made up of many layers, whereby weights are applied to various aspects of the input. Then, the layers are combined using various nonlinear transformations to give a final classification using functions such as softmax. It works well with uncertainty and incompleteness. However, it is very computationally intensive and requires parallel processing.





This is why we've decided to use TensorFlow and Keras libraries to build the Convolutional Neural Networks used in this analysis. These libraries allow our NN to be run using GPUs instead of CPUs. In addition, Convolutional Neural Networks work especially well with images, as it only looks at portions of images at a time, with pooling steps in between to create new feature maps that can be analyzed further.



Feature Engineering

After building our initial models using raw pixels, we attempted to build improved models through basic feature engineering via extraction of global feature descriptors. According (IIango, 2017) feature engineering for classic algorithms include global descriptors that take the entire image for processing vs. local descriptors that focus on parts of an image. In particular, dog faces, an example of a local descriptor, are very popular for image classification, but more computationally costly and time-consuming to implement (Liu et al.). Therefore, to provide more realistic and practical comparisons for our problem, the commonly-used Global Feature Descriptors extracted include color, shape and texture:

- *Color*
  - **Color Histogram:** number of pixels that have each color in each of a fixed list of color ranges that span the image's color space.
- *Texture*
  - **Haralick Texture**: calculated from a Gray Level Co-occurence Matrix (GLCM), which counts the co-occurence of neighboring grey levels. This reflects the spatial distribution of tonal variations within our picture.
- *Shape*
  - **Hu Moments:** a set of 7 numbers calculated using central moments thar are invariant to image transformations: translation, scale, rotation, and reflection

Afterwards, we re-ran the classic ML algorithms used to build our initial models on our new engineered features to see if there is an improvement against all performance metrics and time.

Convolutional Neural Network

Cold Start

Following the completion of our classic ML models, we created a Convolutional Neural Network (CNN) algorithm from scratch (cold start). The benchmark model from (Le, 2019) showed that expected accuracy for the model was about 44% on the entire dataset. On our subset dataset, we'd expect the model performance to be similar.

Transfer Learning

Afterwards, we applied a pre-trained CNN algorithm, as per industry standard, for benchmarking purposes. Zou et al. 2020 obtained performances as high as 83% on their very similar Tsinghua Dogs Dataset with ResNet50 and Inception V3.

Evaluation and Final Model Results

In our final results, we first evaluate the technical aspects and resultant learnings of our models. Then, we discuss the practical implications of our results and provide key recommendations for prospective dog owners.

Comparing Model Performance (Top 20 Breeds)

First, we compare the models above using the following metrics: accuracy and time to train the model. Then, we examine the resulting classifications to better understand our results and identify where, if any, each model performs well and where each model struggles. Next, we discuss the pros and cons of each classifier, and why there may still be value in using classic classifiers.

We ran the model for the top 20 breeds (as defined by number of available images). Below is a summary of the best model from each of our 4 method types.

Method Type	Best Model	Subset	Time to Train (seconds)	Time Improvement (from Worst)	Accuracy	Accuracy Improvement (from Worst)
Classic ML (Raw Pixel)	SVM	Top 20 Breeds	11,294	0.0x	29.6%	0.0%

Classic ML (Feature Engineering)	Random Forest	Top 20 Breeds	**19**	**588.8x**	32.3%	2.7%
CNN	Manual Cold Start	Top 20 Breeds	5,181	1.2x	47.2%	17.5%
CNN	Transfer Learning	Top 20 Breeds	25	447.7x	**65.7%**	**36.1%**

Classic ML with Raw Pixel

Unsurprisingly, building models on raw pixels (non-data engineered) took the longest time: up to 11,294 seconds (3+ hours) for SVM. Accuracy ranged from 5.18% (Neural Networks) to 29.72% (SVM). All raw pixel models were built using default parameters.

Despite our initial expectations, SVM actually performed best among our classic models, albeit still relatively poorly, while Neural Networks performed absolutely abysmally.

Model	Subset	Time to Train (seconds)	Accuracy
KNN	Top 20 Breeds	**44**	17.18%
SVM	Top 20 Breeds	11,294	**29.62%**
Logistic Regression	Top 20 Breeds	910	20.63%
Random Forest	Top 20 Breeds	850	26.57%
Neural Networks	Top 20 Breeds	3,759	5.18%

Classic ML with Feature Engineering (Global Feature Descriptors)

For more practical comparisons, we also ran our Classic ML algorithms using data engineered, commonly-used global features (texture - haralick, color - color histogram, and shape - Hu moments).

Because these data-engineered models are only built on a handful global features, rather than the plethora of raw pixels, the time to run these models was far shorter for every implemented algorithm. For example, it took just 19 seconds to run SVM on global feature descriptors compared to 3+ hours for SVM on Raw Pixels, with very little loss in accuracy (29.62% for raw pixel vs. 27.80% for global descriptors).

More important than time to model, we also observed dramatic improvements in accuracy for nearly all implemented algorithms. The lone exception was the aforementioned SVM model, which as previously mentioned, actually exhibited a small -1.8% decline in accuracy relative to its raw-pixel variant. The best performing model built on global feature descriptors was our RF model, with an accuracy of 32.34%. This represents a 5.8% improvement over its raw-pixel variant while also outperforming the best raw-pixel model (SVM) by +2.7%.

Furthemore, Neural Networks went from the worst performing model using raw pixels to the second-best performing model using global descriptors (5.18% for raw pixel vs. 28.39% for global descriptors). This dramatic +23.2% accuracy improvement underscored the importance of feature engineering/selection for Neural Networks, and lent credence to our assumption that CNN would likely be the best performing algorithm.

Models	Subset	Time to Train (seconds)	Time Improvement (over Raw Pixel variant)	Accuracy	Accuracy Improvement (over Raw Pixel variant)
KNN	Top 20 Breeds	**0**	134.1x	22.12%	4.9%
SVM	Top 20 Breeds	19	**583.0x**	27.80%	-1.8%
Logistic Regression	Top 20 Breeds	4	222.9x	26.87%	6.2%
Random Forest	Top 20 Breeds	19	43.4x	**32.34%**	5.8%
Neural Networks	Top 20 Breeds	43	86.6x	28.39%	**23.2%**

Convolutional Neural Networks (Cold Start)

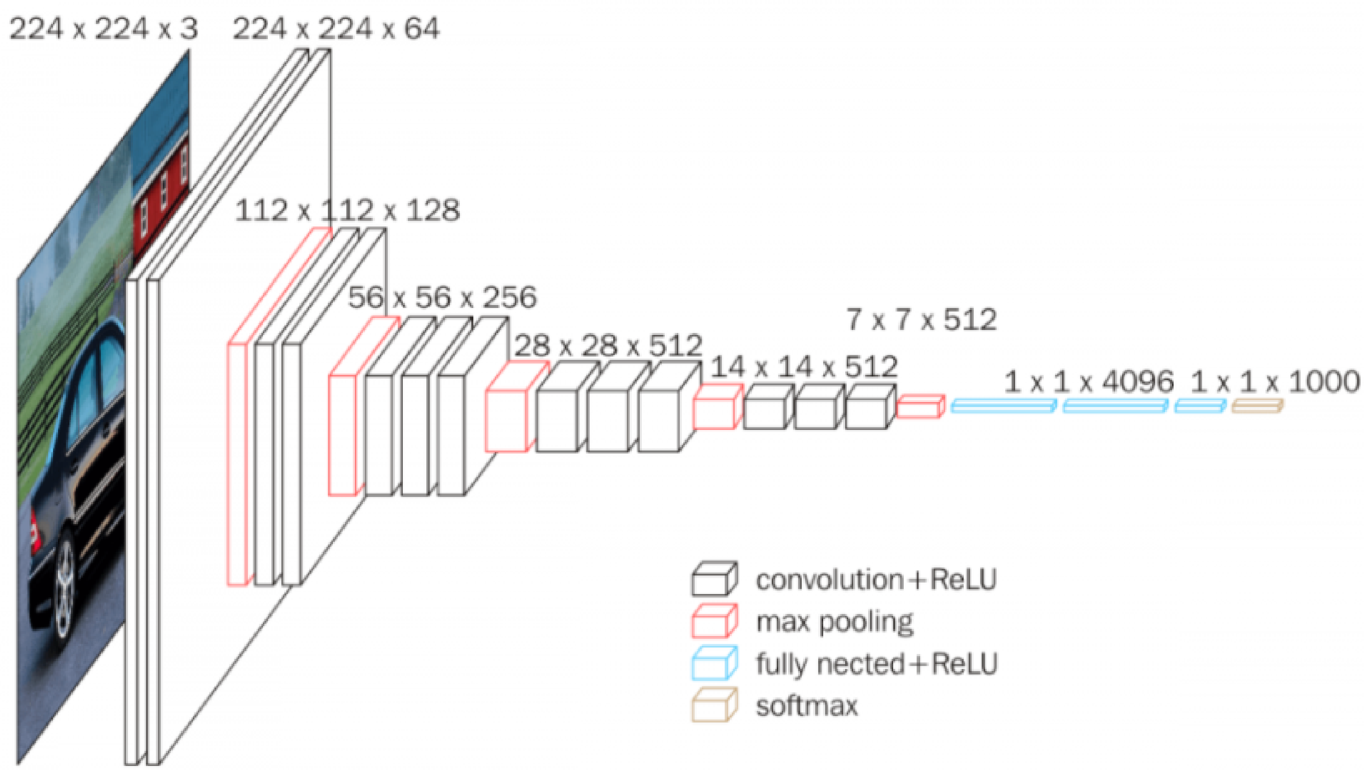
With our Classic ML baselines established, we trained a Convolutional Neural Network (CNN) from scratch. A CNN is a special class of neural networks where filtered layers (convolution) are combined to create a feature map. Therefore, CNN is itself a form of feature engineering, so feature selection is not needed.

For our basic, cold-start CNN, we created a 5-layer CNN model. After fitting the model on the raw pixels, we observed an accuracy of 47.15%, which is +14.8% higher than our best classic ML model (RF using Global Descriptors). Unfortunately, our cold start model was also painfully slow to train, requiring 5,181 seconds or almost two hours. This is slower than all but our SVM using Raw Pixels model.

Models	Subset	Time to Train (seconds)	Accuracy
CNN Manual Cold Start	Top 20 Breeds	**5,181**	**47.15%**

Convolutional Neural Networks (Transfer Learning)

The beauty of CNN is that we can leverage Transfer Learning, which as its name implies, takes a model trained for a task as the starting point for a model on a second task. This use of pre-trained models as a starting point is, as previously mentioned, is industry best practice. In our case, this meant leveraging an image classification model trained on a much larger dataset, such as ImageNet, and using it as the starting point for our model on our much smaller dog dataset. For our implementation, we chose the famous VGG16 model first proposed in a paper by researchers at the University of Oxford (Simonyan & Zisserman, 2015). While there are other newer, higher-performing models available, we chose VGG16 primarily due to the relatively simplicity of its implementation (Neurohive, 2018).



As expected, the performance improvements were dramatic. In just 25 seconds, we were able to train a classifier that had 65.70% accuracy, or +18.55% better than our Cold Start CNN model.

Models	Subset	Time to Train (seconds)	Time Improvement (over Cold Start variant)	Accuracy	Accuracy Improvement (over Cold Start variant)
CNN Transfer Learning	Top 20 Breeds	**25**	**204.8x**	**65.70%**	**18.55%**

Comparing Model Performance (All 120 Breeds)

Using the learnings from above, we will be comparing the Classic ML with Feature Engineering and the CNN Transfer Learning Method as those two methods provided the fastest training times and accuracy.

Method Type	Best Model	Subset	Time to Train (seconds)	Time Improvement (from Worst)	Accuracy	Accuracy Improvement (from Worst)
Classic ML (Feature Engineering)	Random Forest	Top 20 Breeds	**399**	**51.6x**	11.9%	0.0%
CNN	Transfer Learning	Top 20 Breeds	20,982	0.0x	**78.1%**	66.3%

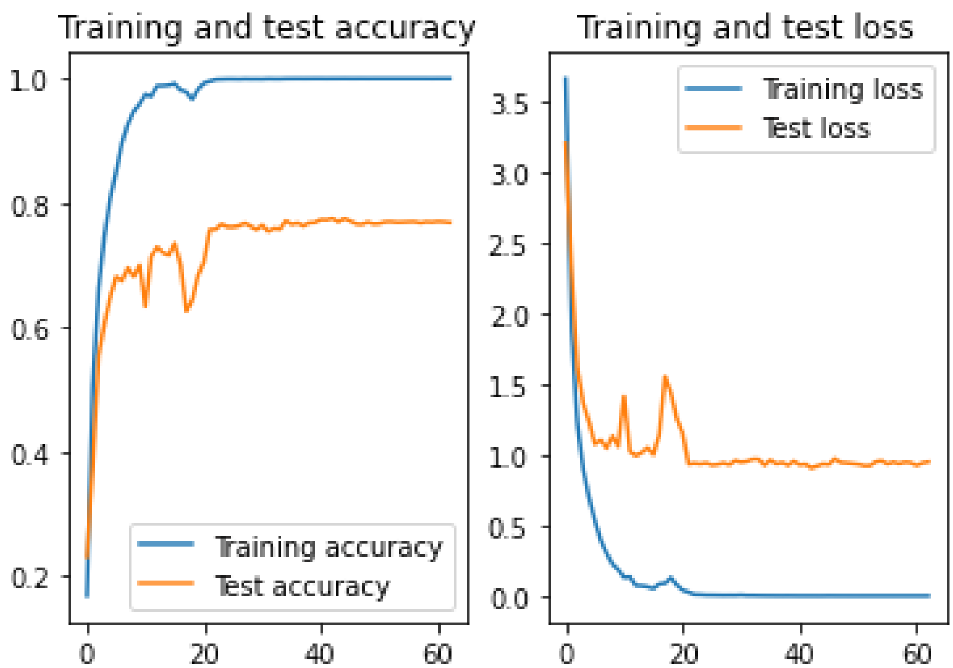
Classic ML with Feature Engineering (Global Feature Descriptors)

Among all the Classic ML with Feature Engineering models, Random Forest performed the best with an accuracy of 11.85%. This is much lower than the respective score when only modeling on the Top 20 Breeds, which indicates that there is likely much more cross-group similarity than within-group similarity among all 120 breeds as compared to the top 20 breeds.

Model	Subset	Time to Train (seconds)	Accuracy
KNN	All 120 Breeds	**5**	6.46%
SVM	All 120 Breeds	529	8.27%
Logistic Regression	All 120 Breeds	112	8.21%
Random Forest	All 120 Breeds	399	**11.85%**
Neural Networks	All 120 Breeds	395	9.33%

Convolutional Neural Networks (Transfer Learning)

Despite taking a very long time to train (almost 6 hours), we were able to get up to 78.1% accuracy with CNN Transfer Learning using VGG16. Considering this is a higher accuracy, this is a trade-off we are willing to make. In addition this high accuracy was also in line with what the makers of the Tsinghua dataset had on their dogs dataset (eg ~83%).



Examining the training accuracy vs. loss charts, we saw that the accuracy of ~80% was reached after 20 epochs though the changes in accuracy didn’t really stabilize until the 63 epoch.

Practical Implications and Recommendations

Now that we’ve covered the technical aspects of our results, we can discuss their practical implications. One of the most important potential implications is understanding why certain dog breeds are often misclassified. As stated in our problem statement, not every dog breed is right for every dog owner. Especially when dogs are still young and growing, it’s not always visually apparent what breed a dog is. Analysis of misclassifications can be useful to help determine similar breeds. With the ongoing dog shortage, it may be helpful for prospective dog owners to find a dog of breeds similar to those of interest.

Comparing ML Methods

- **Classic ML models do not work well with RAW data.** For our analysis, when training on raw data, classic models took anywhere from 44 seconds to 11,294 seconds (3+ hours) and, worse, had lower accuracies than our alternative methods.
- **Feature Engineering substantially improves Training Time without sacrificing Accuracy (compared to Classic ML models on RAW data).** Re-running our Classic ML models on our extracted global feature descriptors, we observed similar, slightly better, accuracies than those from their RAW data variants, but with results achieved in substantially shorter periods of time, ranging from 0 to 43 seconds (when looking at the Top 20 breeds). The training time increased to about 5-529 seconds when looking at the Top 120 Breeds)
- **CNN (Convolutional Neural Networks) performs even better than our best Classic ML models without the need for feature engineering.** Despite the improvements from feature engineering, our classic ML models could not match the performance of even our cold-start CNN. This is because the Convolutional and Pooling layers automatically perform the feature engineering for us. However, for Convolutional Neural Networks to work well, it does need time to train as training the CNN with cold start took almost two hours though we were able to increase our accuracy by 15 points compared to the next best Classic ML model.
- **CNN model built on transfer learning performed best**. Our VGG-16 based model acheived an accuracy of 65.70% in just under 25 seconds. While we only tried one transfer learning-based CNN model, there are many other, higher performing transfer learning models that can further improve our modeling performance. When examining all 120 breeds, our CNN model with Transfer Learning took almost 6 hours to train, however we were able to get accuracy up to 78%.

Therefore, we can conclude that the conventional belief that has made CNN the most popular model for image classifications is indeed justified. CNN offers the following advantages over Classic ML based models:

1. CNN offers transfer learning capability (e.g., use previously trained weights)
2. Using these predefined models it is able to achieve higher accuracy in a much shorter amount of time (when looking at a small number of categories)
3. When looking at a large number of categories, CNN is still able to get a high accuracy in a reasonable amount of time (eg under 6 hours)

Inferences from Model Output

Regardless of the methodology used, the output of the classifier can be used in many different and practical ways. Interpreting the output from our model, we address the following questions a potential dog-owners may ask:

1. Which do breeds are most visually distinctive
2. Which other dog breeds are most visually similar to my favorite breed?

Which dog breeds are most visually distinctive?

Alternate ways to ask this question:

- “Which dog breeds can I most easily tell apart?”
- “Which dog breeds can I adopt and be most confident that I’m not getting a different/wrong breed?”

How can our model help answer this question?

With our model output, we can *quantify* how “distinctive” each breed is by examining the classification report of the classifier.

- **Precision**:
  - $Precision\ =\ \frac{TP}{TP+FP}$
  - With Precision, we can understand which dog breeds are easiest to *classify.*
    - In layman’s terms, Precision tells us, among all dog pictures classified as being a member of a given breed (TP + FP), how many dogs were correctly tagged by our model as members of that breed (TP).
  - Easiest to classify breeds:
    - Of all images *classified* as Bedlington Terrier, 97% were *truly* (TP) Bedlington terrier.
    - Likewise, of all images *classified* as German Short-haired {ointer, 97% were *truly (TP)* German_short-haired_pointer.
- **Recall**:
  - $Recall\ =\ \frac{TP}{TP+FN}$
  - With Recall, we can understand which dog breeds are easiest to *identify*.
    - In layman’s terms, Recall tells us that, among all the dog pictures that were actually from a given breed (TP+FN), how many were correctly tagged by our model as members of that breed (TP)
  - Easiest to identify breed:
    - Of all the images that were actually Bedlington Terrier, 97% were correctly identified as Bedlington Terrier.

- **F1 score:**
    - $F1 \ = 2 * \frac{precision * recall}{precision + recall}$
    - With F1, by combining Precision and Recall, we can understand which dogs are most *unique.*
        - F1 is the harmonic mean of Precision and Recall.
    - Most unique breeds:
        - Similarly, the Bedlington Terrier, German Short-Haired Pointer, Gordon Setter, and Saint Bernard are among the most unique breeds, as all have the highest F1-score of 97%. Photo samples from each of these breeds can be found below.

Model Performance for All 120 Breeds

Breed Name	Breed Code	Precision	Recall	F1-score	Support
Bedlington Terrier	n02093647	97%	97%	97%	36
German Short-Haired Pointer	n02100236	97%	97%	97%	30
Gordon Setter	n02101006	97%	97%	97%	30
Saint Bernard	n02109525	97%	97%	97%	34
Papillon	n02086910	93%	95%	94%	39
Ibizan Hound	n02091244	92%	95%	93%	37
Irish Setter	n02100877	97%	90%	93%	31
Sussex Spaniel	n02102480	100%	87%	93%	30
Brabancon Griffon	n02112706	93%	93%	93%	30
Schipperke	n02104365	90%	93%	92%	30
Groenendael	n02105056	88%	97%	92%	30
Chow	n02112137	97%	87%	92%	39
Bernese Mountain Dog	n02107683	86%	98%	91%	43
Pomeranian	n02112018	89%	93%	91%	43
Afghan Hound	n02088094	88%	91%	90%	47
Blenheim Spaniel	n02086646	94%	84%	89%	37
Norwegian Elkhound	n02091467	88%	90%	89%	39
Border Terrier	n02093754	84%	94%	89%	34

Sealyham Terrier	n02095889	94%	85%	89%	40
Scotch Terrier	n02097298	93%	84%	88%	31
German Shepherd	n02106662	93%	83%	88%	30
Keeshond	n02112350	85%	90%	88%	31
Weimaraner	n02092339	90%	84%	87%	32
Welsh Springer Spaniel	n02102177	87%	87%	87%	30
Komondor	n02105505	84%	90%	87%	30
Leonberg	n02111129	84%	90%	87%	42
Samoyed	n02111889	86%	88%	87%	43
Chihuahua	n02085620	89%	83%	86%	30
Japanese Spaniel	n02085782	81%	92%	86%	37
Black-And-Tan Coonhound	n02089078	84%	87%	86%	31
Clumber	n02101556	80%	93%	86%	30
Old English Sheepdog	n02105641	79%	94%	86%	33
Vizsla	n02100583	86%	83%	85%	30
Bull Mastiff	n02108422	78%	94%	85%	31
African Hunting Dog	n02116738	79%	91%	85%	33
Bluetick	n02088632	85%	82%	84%	34
Saluki	n02091831	83%	85%	84%	40
Airedale	n02096051	83%	85%	84%	40
Dandie Dinmont	n02096437	79%	86%	83%	36
Basenji	n02110806	81%	83%	82%	41
Borzoi	n02090622	85%	77%	81%	30
Rottweiler	n02106550	76%	87%	81%	30
Dhole	n02115913	76%	87%	81%	30
Kerry Blue Terrier	n02093859	71%	91%	80%	35
Doberman	n02107142	88%	73%	80%	30
Affenpinscher	n02110627	80%	80%	80%	30
Pembroke	n02113023	73%	89%	80%	36
Otterhound	n02091635	77%	80%	79%	30
Malinois	n02105162	77%	80%	79%	30
Briard	n02105251	82%	77%	79%	30
Bouvier Des Flandres	n02106382	76%	83%	79%	30

Greater Swiss Mountain Dog	n02107574	79%	79%	79%	33
Entlebucher	n02108000	75%	82%	79%	40
Mexican Hairless	n02113978	80%	77%	79%	31
Pekinese	n02086079	77%	79%	78%	29
West Highland White Terrier	n02098286	71%	88%	78%	33
Golden Retriever	n02099601	70%	87%	78%	30
Brittany Spaniel	n02101388	74%	83%	78%	30
Miniature Pinscher	n02107312	73%	83%	78%	36
Great Pyrenees	n02111500	77%	79%	78%	42
Maltese Dog	n02085936	73%	82%	77%	50
Bloodhound	n02088466	70%	84%	77%	37
Scottish Deerhound	n02092002	78%	76%	77%	46
Curly-Coated Retriever	n02099429	71%	83%	77%	30
English Setter	n02100735	82%	72%	77%	32
English Springer	n02102040	77%	77%	77%	31
Cairn	n02096177	75%	77%	76%	39
Boston Bull	n02096585	72%	81%	76%	36
Cocker Spaniel	n02102318	88%	68%	76%	31
Boxer	n02108089	79%	73%	76%	30
French Bulldog	n02108915	81%	71%	76%	31
Toy Terrier	n02087046	74%	76%	75%	34
Flat-Coated Retriever	n02099267	74%	77%	75%	30
Chesapeake Bay Retriever	n02099849	69%	82%	75%	33
Irish Water Spaniel	n02102973	74%	77%	75%	30
Border Collie	n02106166	68%	83%	75%	30
Tibetan Mastiff	n02108551	81%	70%	75%	30
Standard Poodle	n02113799	77%	74%	75%	31
Wire-Haired Fox Terrier	n02095314	74%	74%	74%	31
Malamute	n02110063	71%	77%	74%	35
Pug	n02110958	76%	72%	74%	40
Silky Terrier	n02097658	71%	75%	73%	36
Irish Terrier	n02093991	74%	70%	72%	33

Norfolk Terrier	n02094114	81%	65%	72%	34
Norwich Terrier	n02094258	71%	73%	72%	37
English Foxhound	n02089973	80%	65%	71%	31
Irish Wolfhound	n02090721	73%	70%	71%	43
Italian Greyhound	n02091032	68%	72%	70%	36
Australian Terrier	n02096294	74%	67%	70%	39
Giant Schnauzer	n02097130	72%	68%	70%	31
Kuvasz	n02104029	70%	70%	70%	30
Toy Poodle	n02113624	70%	70%	70%	30
Basset	n02088238	72%	66%	69%	35
Shetland Sheepdog	n02105855	67%	71%	69%	31
Whippet	n02091134	69%	68%	68%	37
Lakeland Terrier	n02095570	71%	64%	68%	39
Miniature Schnauzer	n02097047	61%	77%	68%	30
Tibetan Terrier	n02097474	76%	61%	68%	41
Labrador Retriever	n02099712	74%	59%	66%	34
Newfoundland	n02111277	58%	77%	66%	39
Shih-Tzu	n02086240	66%	64%	65%	42
Rhodesian Ridgeback	n02087394	63%	65%	64%	34
Yorkshire Terrier	n02094433	62%	66%	64%	32
Kelpie	n02105412	74%	57%	64%	30
Beagle	n02088364	62%	64%	63%	39
Dingo	n02115641	62%	65%	63%	31
Standard Schnauzer	n02097209	76%	52%	62%	31
Redbone	n02090379	67%	55%	60%	29
Walker Hound	n02089867	61%	57%	59%	30
Great Dane	n02109047	75%	48%	59%	31
Soft-Coated Wheaten Terrier	n02098105	59%	55%	57%	31
Lhasa	n02098413	61%	54%	57%	37
Siberian Husky	n02110185	56%	58%	57%	38
Staffordshire Bullterrier	n02093256	55%	55%	55%	31
Cardigan	n02113186	72%	42%	53%	31

Collie	n02106030	65%	43%	52%	30
Appenzeller	n02107908	58%	47%	52%	30
Miniature Poodle	n02113712	60%	39%	47%	31
American Staffordshire Terrier	n02093428	41%	44%	42%	32
Eskimo Dog	n02109961	41%	30%	35%	30

Examples of most unique looking breeds as quantified by F1-score

Bedlington terrier Images

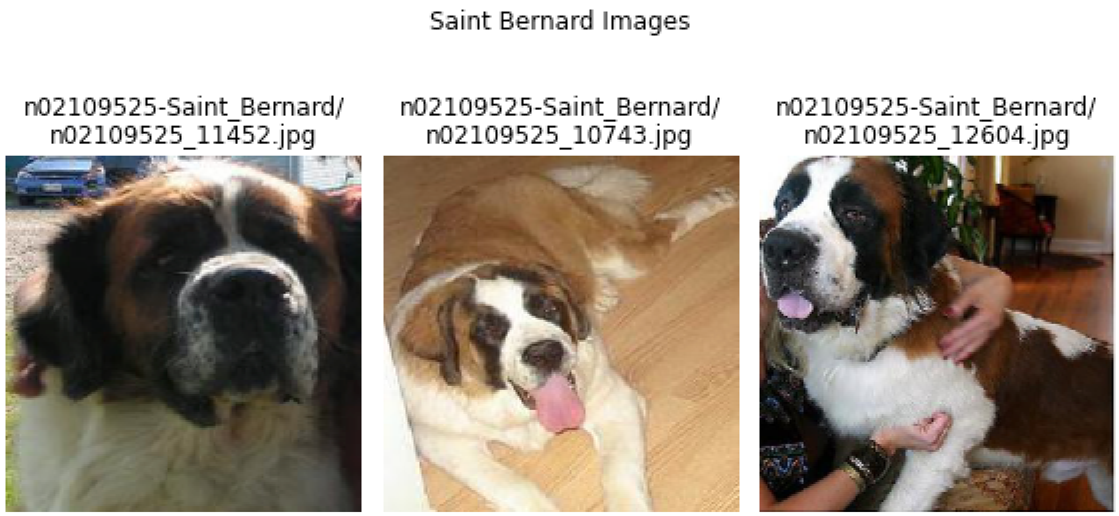
n02093647-Bedlington_terrier/ n02093647_1495.jpg | n02093647-Bedlington_terrier/ n02093647_1238.jpg | n02093647-Bedlington_terrier/ n02093647_1081.jpg

German Short-Haired Pointer Images

n02100236-German_short-haired_pointer/ n02100236_1529.jpg | n02100236-German_short-haired_pointer/ n02100236_1673.jpg | n02100236-German_short-haired_pointer/ n02100236_156.jpg

Gordon setter Images

n02101006-Gordon_setter/ n02101006_1354.jpg | n02101006-Gordon_setter/ n02101006_114.jpg | n02101006-Gordon_setter/ n02101006_18.jpg



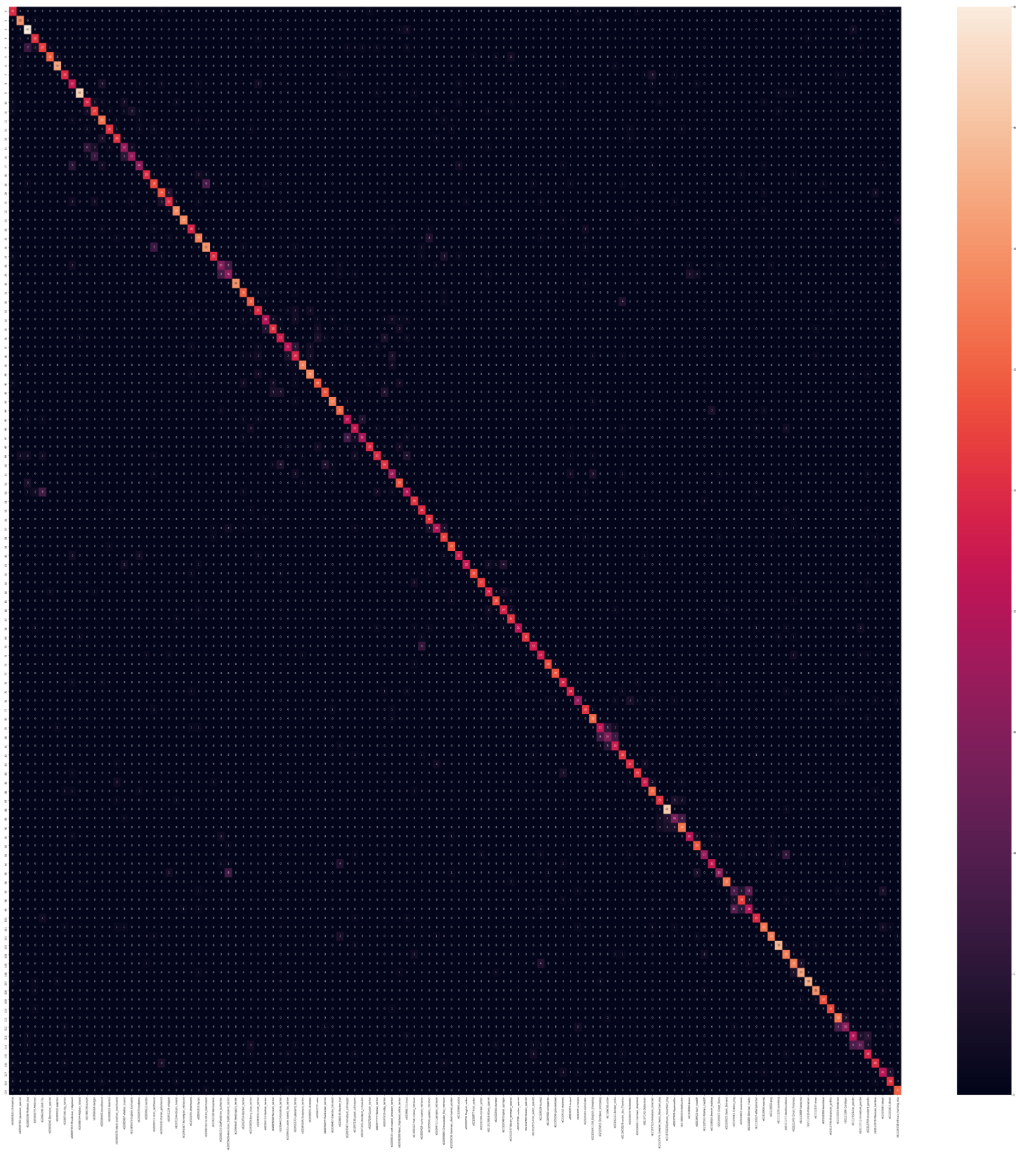
Which other dog breeds are similar to my favorite breed?

Alternate ways to ask this question:

- "I can't find the dog I want. What other dog breeds look like the one I want?"
- "I can't tell what breed this dog is. What breeds could it be?"

On the other hand, misclassifications can be used to identify which breeds look the most similar. For our best classifier (CNN transfer learning), we found that Eskimo dogs are often confused with Siberian Huskies, while the Scottish Deerhound and often confused with Irish Wolfhounds and Staffordshire Bullterriers and often confused with American Staffordshire terriers. The visual similarity between different dog breeds is summarized in Confusion Matrix below.

Confusion Matrix of Dog Breed Similarity



Since the confusion matrix is a bit difficult to read, I have also revisualized the misclassifications as a table below:

Top 10 Confused Breeds

Breed Name [A]	Breed Code [A]	Breed Name [B]	Breed Code [B]	Misclassification Count
Eskimo Dog	n02109961	Siberian Husky	n02110185	21
Irish Wolfhound	n02090721	Scottish Deerhound	n02092002	16
Staffordshire Bullterrier	n02093256	American Staffordshire Terrier	n02093428	16
Shetland Sheepdog	n02105855	Collie	n02106030	13
Miniature Schnauzer	n02097047	Standard Schnauzer	n02097209	12
Collie	n02106030	Border Collie	n02106166	12
Pembroke	n02113023	Cardigan	n02113186	12
Toy Poodle	n02113624	Miniature Poodle	n02113712	12
Appenzeller	n02107908	Entlebucher	n02108000	11
Beagle	n02088364	English Foxhound	n02089973	10

Case Study: Which other dog breeds are the most similar to the Eskimo Dog?

Let’s examine the top most confused breeds, the Eskimo Dog and the Siberian Husky:

Reference Images - Siberian Husky

Reference Images - Eskimo Dogs misclassified as Siberian Huskies

To the naked eye, these two breeds look almost identical [and equally adorable], so it's no surprise the model had trouble differentiating them. Examining the detailed output from one specific misclassified picture, we see that the Eskimo Dog is the second most likely breed, though its respective probability is almost 40% lower than that of the Siberian Husky.

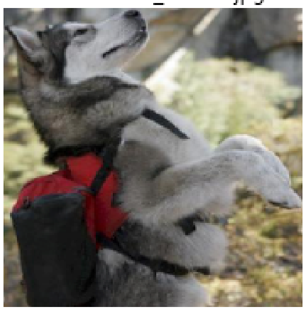
Examination of one of misclassification of Eskimo Dog as Siberian Husky

```
n02109961-Eskimo_dog/n02109961_11511.jpg
0.576   :   (99, 'n02110185-Siberian_husky')
0.1652   :   (97, 'n02109961-Eskimo_dog')
0.16   :   (98, 'n02110063-malamute')
0.07385   :   (79, 'n02105855-Shetland_sheepdog')
0.006737   :   (25, 'n02091831-Saluki')
0.006477   :   (81, 'n02106166-Border_collie')
0.00463   :   (18, 'n02090622-borzoi')
0.002449   :   (80, 'n02106030-collie')
0.001683   :   (105, 'n02111500-Great_Pyrenees')
0.000453   :   (106, 'n02111889-Samoyed')
```

These classification probabilities are also useful for examining what visual features may have led to misclassification, and thus, also useful in suggesting similar looking dogs. Let's examine the next 3 most likely breeds for this example dog.

Malamute As Eskimo Dog (16% probability, 3 total misclassified cases)

n02110063-malamute/
n02110063_10768.jpg



n02110063-malamute/
n02110063_10787.jpg

n02110063-malamute/
n02110063_11814.jpg

Shetland Sheepdog As Eskimo Dog (7.4% probability, 0 misclassified cases)

n02105855-Shetland_sheepdog/
n02105855_11668.jpg

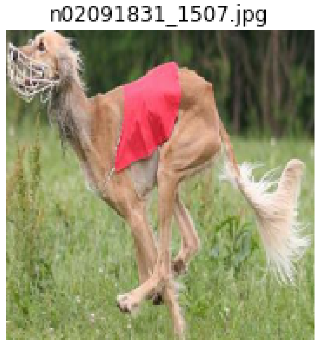
n02105855-Shetland_sheepdog/
n02105855_10608.jpg

n02105855-Shetland_sheepdog/
n02105855_11955.jpg

Saluki as Eskimo Dog (.7% probability, 0 misclassfied cases)

n02091831-Saluki/
n02091831_1495.jpg

n02091831-Saluki/
n02091831_1507.jpg



n02091831-Saluki/
n02091831_13354.jpg

Examining the images above, we can see both the power of our model at identifying the Eskimo Dog breed, as well as why the model assigned probability to each of these 3 breeds. Though the Eskimo dog is much more similar to the most frequently confused-for Siberian Husky, we can see that all of these breeds exhibit substantial multicoloring, specifically large sections of white fur. Furthermore, the Malamute in particular also shares the bicolor pattern of the Eskimo Dog and Siberian Husky breeds—white faces, white underbellies, and colored backs—albeit with a chubbier body, which is hard to differentiate in the resized images.

Therefore, we can conclude that if a person likes the look of the Eskimo Dog, the Siberian Husky and the Malamute would be very good alternative breeds to recommend if an Eskimo Dog is not available. In fact, the debate over Husky vs. Malamute is so common that the American Kennel Club has written an article comparing the two breeds (Gibeault, 2019). The reason that Eskimo Dogs are not mentioned in this aforementioned article is likely due to its rarity. Wikipedia lists the Canadian Eskimo Dog as endangered with only 300 purebreds left in 2008 (*Canadian Eskimo Dog*, 2021). Thus, by adopting an alternative breed suggested by the model, a prospective dog-owner can also help to reduce demand pressure on the few remaining Eskimo Dogs.

Conclusion

In this paper, we examined the performance of *Classic ML Methods for Image Classification* against the industry preferred method of *Convolutional Neural Networks (CNN)* on the Stanford Dogs dataset. We demonstrated that *Feature Engineering* is needed for *Classic ML* Methods as training time on engineered features is greatly reduced when compared to training time on raw pixels, with minimal impact on accuracy. On our subsampled population of Top 20 Breeds, we also demonstrated that CNN performed better than *Classic ML* methods with much higher accuracy (+15% gain in accuracy for *Cold Start CNN* vs. *Best Classic ML Model*). We also demonstrated the power of *Transfer Learning*, as on our Top 20 breeds subsample, we achieved an increased accuracy of 65% under just 25 seconds. When applied to the complete population of Stanford Dogs, we demonstrated that *CNN* exhibited MUCH higher accuracy than *Classic ML Methods with Feature Engineering* (+66% gain in accuracy), although the CNN model took about 51.6x longer to train than the *Classic ML Methods with Feature Engineering* model.

In addition, we also demonstrated some applications of fine-tuned image classification. Using our classifier, we were able to correctly identify the breed of a dog from an image with almost 80% accuracy. This is on par with the performance Tsinghua researchers achieved on their own Tsinghua dogs dataset. Furthermore, we demonstrated that the misclassifications themselves were also useful and informative. Through interpretation of misclassifications, we were able to find similar breeds by examining the probabilities a given dog photo portrayed a dog of another breed returned by our image classifier. Specifically, we used the example of the misclassification of an Eskimo Dog as a Siberian Husky to show that the Siberian Husky and Alaska Malamute would be great alternatives for an Eskimo Dog. This is important considering the relative scarcity of the Eskimo Dogs. Using our classifier, users would be able to find other breeds of interest if their desired breed is rare (in the case of the Eskimo dog) or unavailable.

Bibliography

Brendel, C. (2020, May 14). *Quickly Compare Multiple Models*. Towards Data Science. Retrieved July 12, 2021, from https://towardsdatascience.com/quickly-test-multiple-models-a98477476f0

Canadian Eskimo Dog. (2021, August 2). Wikipedia. https://en.wikipedia.org/wiki/Canadian_Eskimo_Dog

Devang. (2021, February 6). *Transfer Learning with Keras and EfficientNets*. Kaggle. Retrieved August 2, 2021, from https://www.kaggle.com/devang/transfer-learning-with-keras-and-efficientnets

Genesis. (2018, September 25). *Pros and Cons of K-Nearest Neighbors*. Genesis. Retrieved July 28, 2021, from https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/

Gibeault, S. (2019). *Meet Two Similar Yet Different Breeds: Siberian Husky and Alaskan Malamute*. American Kennel Club. https://www.akc.org/expert-advice/dog-breeds/siberian-husky-vs-alaskan-malamute/

IIango, G. (2017, January 28). *Image Classification using Python and Scikit-learn*. Image Classification using Python and Scikit-learn. Retrieved 7 16, 2021, from https://gogul.dev/software/image-classification-python

Khosla, A., Jayadevaprakash, N., Yao, B., & Li, F.-F. (2011). Novel dataset for Fine-Grained Image Categorization. *First Workshop on Fine-Grained Visual Categorization (FGVC)*, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://people.csail.mit.edu/khosla/papers/fgvc2011.pdf

Le, J. (2019, March 15). *How to easily build a Dog breed Image classification model*. Deep Learning. Retrieved 7 15, 2021, from https://medium.com/nanonets/how-to-easily-build-a-dog-breed-image-classification-model-2fd214419cde

Li, J. (2019, November 12). *Stanford Dogs Dataset*. Kaggle. Retrieved July 15, 2021, from https://www.kaggle.com/jessicali9530/stanford-dogs-dataset

Liu, J., Kanazawa, A., Jacobs, D., & Belhumeur, P. (n.d.). Dog Breed Classification Using Part Localization. *Europe Conference on Computer Vision*, *2012*(2012), 172-185.

Neurohive. (2018, November 20). *VGG16 – Convolutional Network for Classification and Detection*. Neurohive. Retrieved July 30, 2021, from https://neurohive.io/en/popular-networks/vgg16/

Simonyan, K., & Zisserman, A. (2015, April 10). Very deep convolutional networks for large-scale image recognition. *ICLR*. https://arxiv.org/abs/1409.1556

VBoW Pt 1 - Image Classification in Python with SIFT Features. (2016, May 6). VBoW Pt 1 - Image Classification in Python with SIFT Features. Retrieved July 14, 2021, from https://ianlondon.github.io/blog/how-to-sift-opencv/

Zou, D.-N., Zhang, S.-H., Mu, T.-J., & Zhang, M. (2020, October 01). A new dataset of dog breed images and a benchmark for finegrained classification. *Computational Visual Media*, *6*, 477-487. https://doi.org/10.1007/s41095-020-0184-6