

Analysis of Image Feature Extraction Techniques for Animal Face Detection

George Tzimas

November 2024

Abstract

This project investigates conventional feature extraction methods for classifying animal faces, comparing their effectiveness at distinguishing between the different classes. Using the LHI Animal Faces dataset, which includes a total of 20 classes, we applied Histogram of Oriented Gradients (HOG), Sobel filtering, and the Laplacian operator to extract shape, texture, and edge features. Each method was evaluated by training a Support Vector Classifier (SVC) on the extracted features, with performance metrics such as precision, recall, and F1-score used for comparison. Results demonstrated that HOG features were able to outperform the other methods across most metrics, being able to capture the structural and shape information critical for distinguishing between all the classes.

Keywords: animal classification, feature extraction, Histogram of Oriented Gradients, Sobel filter, Laplacian operator, Flask deployment

1 Introduction

Object detection and classification plays a crucial role in the domain of computer vision, and a vast amount of research has been conducted over the years with the goal of improving a computer's ability to achieve these tasks. Deep learning techniques have revolutionized that approach by utilizing complex convolutional neural networks (CNNs) architectures. These neural models have the ability to automatically learn hierarchical feature representations from the raw pixel data, enabling them to surpass traditional computer vision techniques of the past. CNNs such as AlexNet, VGGNet, GoogLeNet, and ResNet are a handful of examples of image classification models that have pushed the limits in terms of performance, depth, and architectural innovation in the field of computer vision. However, the one of their disadvantages is that they are often considered "black box" models, due to their internal structure/workings not being easily interpretable, making it difficult to understand how these models' decisions are made.

The challenge of interpretability associated with deep learning models is mitigated when using more traditional feature extraction and classification methods. A classic example of more traditional methods would be edge detection, where the edges within an image are extracted as features



for use in classification. Edge detection algorithms, such as Sobel, Laplace, and techniques like Histogram of Oriented Gradients (HOG), highlight the boundaries and contours of objects within an image, providing critical information about the shape and structure of the content.

In addition, when training a neural network from scratch, the requirement in terms of data volume cannot be easily met unless extensive labeled datasets are available. However, in cases where the dataset size is more limited, traditional feature extraction methods may prove to be more effective for the purposes of classification. These methods will allow us to extract interpretable and meaningful features that can be fed into simpler machine learning classification models.

2 Background

HOG is a well-established method for object detection and has been widely used for tasks like human and animal detection. [Dalal and Triggs, 2005] demonstrated HOG’s effectiveness in detecting humans by dividing images into cells, computing gradient histograms within these cells, and normalizing them across overlapping blocks to enhance robustness to local variations. Similarly, [Rangdal and Hanchate, 2014] applied HOG for animal detection, showing that HOG could capture texture and shape features essential for distinguishing animal species.

Although our image dataset contains face-only images, the principles from [Rybski et al., 2010] showed that HOG features could still be effective across varied poses and angles, thus offering even more adaptability for potential future work. In addition, if the dataset were to be expanded and hardware constraints were to become an issue, splitting the HOG computations between Field-Programmable Gate Array (FPGA) hardware and software could mitigate that issue, as was the case with [Ghaffari et al., 2020].

3 Methods

3.1 Dataset

The data used for this project is the [LHI Animal Faces](#) dataset, which consists of a total of 20 classes. Sample images for each class are provided in [Figure 1](#). These classes include 19 different animal species and one human class. Due to the multi-class prediction challenge of this dataset, it will allow us to evaluate how our feature extraction methods perform across a diverse range of categories. This evaluation will help assess the robustness and generalizability of the extracted features when distinguishing between different animal species and the additional human class.

3.2 Preprocessing

The dataset went through several preprocessing steps/stages in order to effectively extract their features:



Figure 1: Sample images for each of the 19 classes of animals, with the addition of one class of humans.

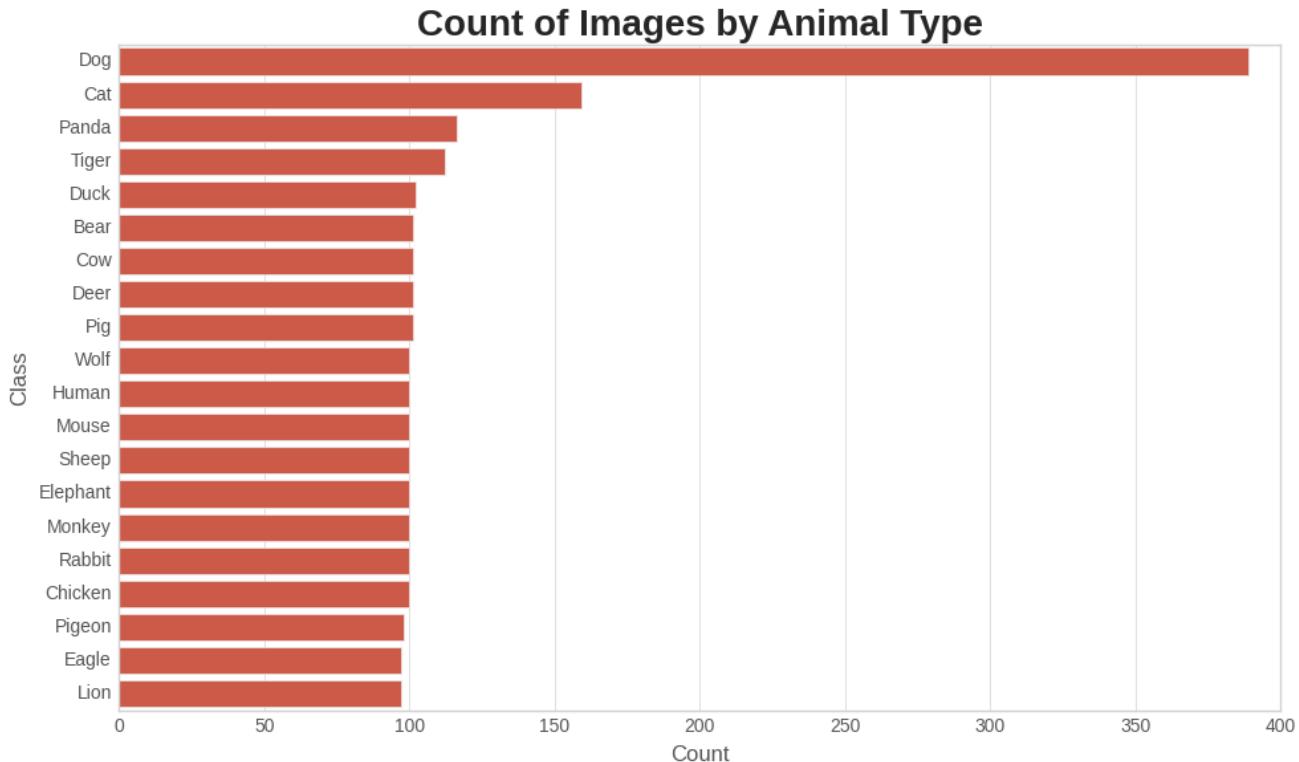


Figure 2: Counts of images per class.

1. **Grayscale transformation:** Grayscale images simplify the computations by reducing the data to a single intensity channel, making the feature extraction process more efficient without the additional complexity of color channels.
2. **Histogram equalization:** After converting the images to grayscale, histogram equalization was performed to enhance the contrast by spreading out the most frequent intensity values in an image. This results in the histogram of intensities approximating a uniform distribution, which ensures that the feature extraction methods are not biased by varying lighting conditions or low contrast.
3. **Adding Gaussian noise:** The last step involved adding Gaussian random noise to the image, with the goal of improving our classification model's ability to generalize to slightly



altered or noisier versions of the training data and prevent potential overfitting issues.

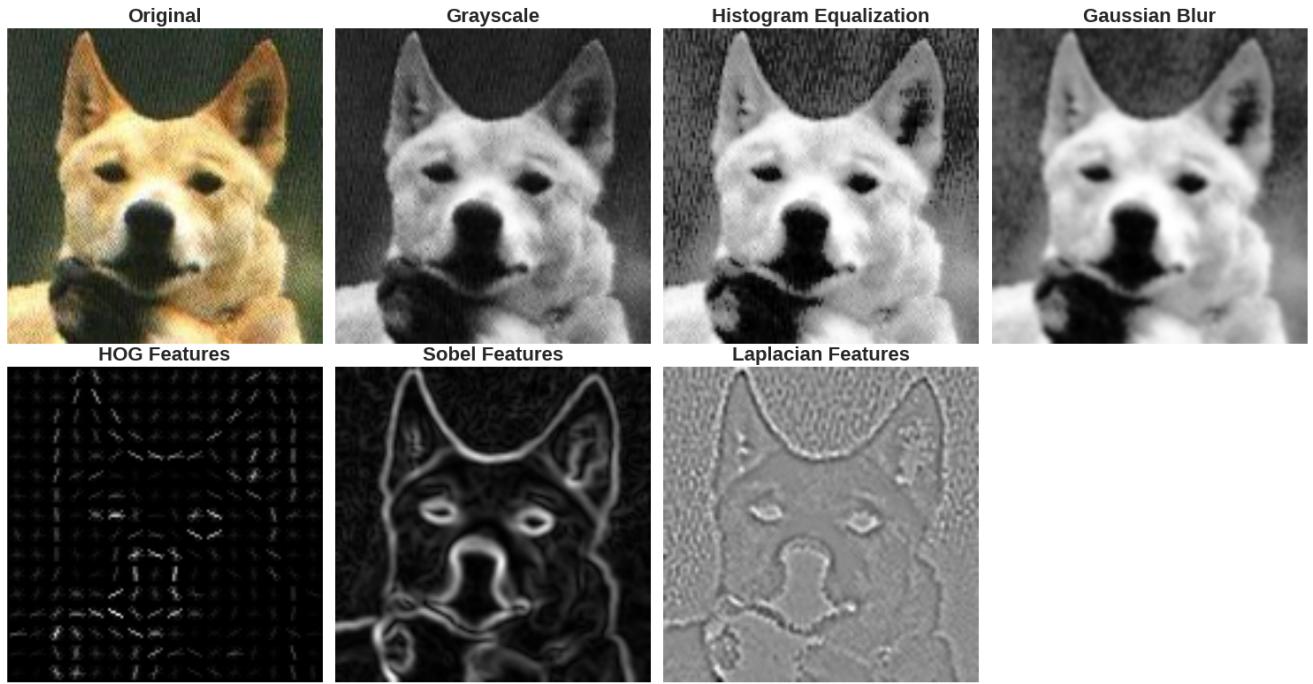


Figure 3: Preprocessing and final stages of a sample image.

3.3 Feature Extraction

After preprocessing all of the images, several edge detection methods were used to compare which method was better at extracting meaningful features to be used by a multinomial classification model.

1. **Sobel operator:** The Sobel operator works by computing the gradient of the image intensity at each pixel and highlighting regions of high spatial frequency that correspond to edges. It uses two convolution kernels, one for detecting horizontal changes and one for vertical changes. These kernels are applied to the image to calculate the intensity gradients in the x and y directions, respectively.

The gradient magnitude (G) and the direction (θ) are then computed for each pixel using the formulas above, and subsequently used as features for training the classification model.

2. **Laplacian Operator:** The Laplacian operator works by calculating the second-order derivative of the image. It captures regions where the intensity changes rapidly, highlighting edges more uniformly in all directions.

In order to ensure that edge responses are positive and represent edge strength uniformly, the absolute value of the Laplacian response is taken. To capture the overall edge information from the image, basic statistics like the mean, standard deviation and maximum are computed. Lastly, the image is divided into a 4x4 grid and for each cell, similar statistics are computed.



3. **Histogram of Oriented Gradients:** Similar to the Sobel operator, we begin by computing the gradient of the image in both horizontal and vertical directions, and the gradient magnitude and direction are calculated. The image then is divided into small, evenly sized, non-overlapping regions called cells (in most cases, 8x8 pixels). Within each cell, the gradient orientation and magnitude are used to create a histogram. To account for changes in lighting and contrast, the histograms are normalized over larger regions called blocks. A block typically contains multiple cells (in most cases, a 2x2 grid of cells). The histograms within each block are then concatenated and normalized.

3.4 Model Fitting and Evaluation

The full image dataset was split into training and testing sets, with 80% and 20% of the data allocated for training and testing, respectively. To evaluate and compare the three feature extraction methods –Sobel, Laplacian, and HOG–, a non-regularized Support Vector Classifier (SVC) model was trained separately on the features extracted by each method. Classification result tables for each of the methods are detailed in tables 4, 5, and 6.

After determining that HOG features had the best overall performance, various classification algorithms that allowed multi-class target variable were evaluated to identify the most effective model for distinguishing between animal species. Table 1 lists all of the models tested along with their performance across different metrics. Out of all the models, SVC had the best performance in terms of accuracy (0.79), sensitivity/recall (0.77) and the overall F1 score (0.77), making it the preferred choice for final model selection.

Table 1: Performance of different multinomial classification models on the HOG-extracted features.

Model	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbors	0.39	0.79	0.39	0.44
Support Vector Machine	0.77	0.78	0.77	0.77
Random Forest	0.45	0.69	0.45	0.43
Logistic Regression	0.76	0.77	0.76	0.76
SGD Classifier	0.62	0.76	0.62	0.65

4 Results

4.1 Model Performance

Out of the three methods, HOG features demonstrated the best performance across all classes in terms of precision, recall and F1 score. It was able to capture structure and shape information from the images more effectively than any of the other methods. As a result, it was selected as the final feature extraction method for training our final classification model.

To improve the performance of our SVC model, hyperparameter tuning was conducted to better fit the model to our data. This method allows us to systematically adjust the parameters



of our model, finding the optimal settings to enhance accuracy and generalizability ([Table 2](#)). The final model achieved a strong performance across all metrics in terms of weighted averages, with a precision of 0.774, recall of 0.746, and F1-score of 0.747 ([Table 3](#)).

Table 2: Optimal hyperparameters chosen by GridSearchCV.

Parameter	Value	Description
C	0.1	Controls the trade-off between a wider margin and training accuracy.
degree	2	Degree of the polynomial kernel function.
gamma	'scale'	Kernel coefficient controlling the influence of each training sample.
kernel	'linear'	Specifies the kernel type to be used in the algorithm.

Table 3: Final model performance metrics using hyperparameter-tuned model.

Class	precision	recall	f1-score	support
Bear	0.733	0.786	0.759	28.000
Cat	0.643	0.844	0.730	32.000
Chicken	0.867	0.650	0.743	20.000
Cow	0.813	0.650	0.722	20.000
Deer	1.000	0.850	0.919	20.000
Dog	0.607	0.910	0.728	78.000
Duck	0.703	0.765	0.732	34.000
Eagle	0.667	0.526	0.588	19.000
Elephant	0.824	0.700	0.757	20.000
Human	1.000	0.950	0.974	20.000
Lion	0.762	0.842	0.800	19.000
Monkey	0.917	0.550	0.688	20.000
Mouse	0.500	0.450	0.474	20.000
Panda	1.000	0.739	0.850	23.000
Pig	0.500	0.400	0.444	20.000
Pigeon	0.727	0.800	0.762	20.000
Rabbit	0.933	0.700	0.800	20.000
Sheep	0.842	0.800	0.821	20.000
Tiger	1.000	0.696	0.821	23.000
Wolf	1.000	0.750	0.857	20.000
accuracy	0.746	0.746	0.746	0.746
macro avg	0.802	0.718	0.748	496.000
weighted avg	0.774	0.746	0.747	496.000

4.2 Model Deployment

After completion of the model training process, the preprocessing steps and final model were deployed as a web application using the Flask framework, which provides an easy and intuitive way to interact with the classifier and display the model results. A simple HTML front-end was designed for the web application, allowing users to upload images and view results easily. When



Image Classification Flask App Using Machine Learning

Upload Image

No file selected.

Instructions:
Upload a file with extension ".png", ".jpg", or ".jpeg". For other image extensions, you will be **redirected** to Error 404 page.

Label	Confidence Score
Dog	0.16
Elephant	0.12
Sheep	0.12
Human	0.12
Duck	0.08



Developed by [Giorgos Tzimas](#)

Figure 4: Web interface of the Flask app with the ability to upload images and get prediction results.

an image is uploaded, it’s sent to the Flask backend, where the model processes it and returns the top five predicted class labels.

5 Conclusion

This project explored the effectiveness of various feature extraction methods —Sobel, Laplacian, and Histogram of Oriented Gradients (HOG)— for animal face detection and classification. The results showed that HOG was able to outperform Sobel and Laplacian in capturing the structural and shape information of the images. Combined with the use of SVC for class predictions, this image-to-prediction pipeline was able to achieve higher performance across all metrics when compared to the latter methods. While these methods may not match the complexity of advanced neural network models like CNNs, the results showed that they remain practical and effective options for image classification tasks.

5.1 Limitations

The biggest limitation when it comes to these feature extraction methods is that the model was trained specifically on images containing only facial structures of animals, which limits its ability to accurately classify images that feature the entire animal body or those with varying poses and backgrounds. It relies heavily on specific facial features to distinguish between classes, meaning it lacks the generalizability needed for full-body or more complex imagery.



5.2 Future Work

In order to address that limitation, the most prominent improvement to the image-to-prediction pipeline is being able to incorporate automated face detection methods, such as You Only Look Once (YOLO) or Haar cascades, to preprocess images by isolating the animal’s face before feature extraction. This would allow the model to be applied to full-body images or images with complex backgrounds by ensuring that only the relevant facial region is used for classification. Another potential improvement would involve the inclusion of color moments to improve the model’s ability to distinguish between different animal species, as was demonstrated by [Taner et al., 2023].

A Initial Classification Model Metrics

Table 4: SVC classification metrics for Sobel features.

Class	precision	recall	f1-score	support
Bear	0.429	0.450	0.439	20.000
Cat	0.581	0.563	0.571	32.000
Chicken	0.500	0.400	0.444	20.000
Cow	0.563	0.450	0.500	20.000
Deer	0.750	0.600	0.667	20.000
Dog	0.426	0.705	0.531	78.000
Duck	0.909	0.476	0.625	21.000
Eagle	0.588	0.526	0.556	19.000
Elephant	0.615	0.400	0.485	20.000
Human	0.591	0.650	0.619	20.000
Lion	0.269	0.368	0.311	19.000
Monkey	0.250	0.250	0.250	20.000
Mouse	0.333	0.250	0.286	20.000
Panda	0.842	0.696	0.762	23.000
Pig	0.500	0.350	0.412	20.000
Pigeon	0.500	0.350	0.412	20.000
Rabbit	0.412	0.350	0.378	20.000
Sheep	0.533	0.400	0.457	20.000
Tiger	0.815	0.957	0.880	23.000
Wolf	0.563	0.450	0.500	20.000
accuracy	0.516	0.516	0.516	0.516
macro avg	0.548	0.482	0.504	475.000
weighted avg	0.539	0.516	0.514	475.000



Table 5: SVC classification metrics for Laplacian features.

Class	precision	recall	f1-score	support
Bear	0.350	0.350	0.350	20.000
Cat	0.421	0.500	0.457	32.000
Chicken	0.222	0.300	0.255	20.000
Cow	0.222	0.200	0.211	20.000
Deer	0.500	0.450	0.474	20.000
Dog	0.301	0.526	0.383	78.000
Duck	0.476	0.476	0.476	21.000
Eagle	0.316	0.316	0.316	19.000
Elephant	0.364	0.400	0.381	20.000
Human	0.722	0.650	0.684	20.000
Lion	0.357	0.263	0.303	19.000
Monkey	0.167	0.100	0.125	20.000
Mouse	0.250	0.200	0.222	20.000
Panda	0.700	0.609	0.651	23.000
Pig	0.250	0.100	0.143	20.000
Pigeon	0.500	0.200	0.286	20.000
Rabbit	0.000	0.000	0.000	20.000
Sheep	0.357	0.250	0.294	20.000
Tiger	0.833	0.870	0.851	23.000
Wolf	0.500	0.400	0.444	20.000
accuracy	0.387	0.387	0.387	0.387
macro avg	0.390	0.358	0.365	475.000
weighted avg	0.386	0.387	0.375	475.000



Table 6: SVC classification metrics for HOG features.

Class	precision	recall	f1-score	support
Bear	0.810	0.850	0.829	20.000
Cat	0.684	0.813	0.743	32.000
Chicken	0.773	0.850	0.810	20.000
Cow	1.000	0.550	0.710	20.000
Deer	0.800	1.000	0.889	20.000
Dog	0.615	0.859	0.717	78.000
Duck	0.833	0.714	0.769	21.000
Eagle	0.667	0.632	0.649	19.000
Elephant	0.882	0.750	0.811	20.000
Human	1.000	1.000	1.000	20.000
Lion	0.769	0.526	0.625	19.000
Monkey	0.842	0.800	0.821	20.000
Mouse	0.737	0.700	0.718	20.000
Panda	0.947	0.783	0.857	23.000
Pig	0.800	0.600	0.686	20.000
Pigeon	0.875	0.700	0.778	20.000
Rabbit	0.857	0.600	0.706	20.000
Sheep	0.667	0.700	0.683	20.000
Tiger	0.875	0.913	0.894	23.000
Wolf	1.000	0.800	0.889	20.000
accuracy	0.773	0.773	0.773	0.773
macro avg	0.822	0.757	0.779	475.000
weighted avg	0.794	0.773	0.772	475.000

References

- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection.
- [Ghaffari et al., 2020] Ghaffari, S., Soleimani, P., Li, K. F., and Capson, D. W. (2020). Analysis and comparison of fpga-based histogram of oriented gradients implementations. *IEEE Access*, 8:79920–79934.
- [Rangdal and Hanchate, 2014] Rangdal, M. B. and Hanchate, D. B. (2014). Animal detection using histogram oriented gradient. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(2):178–183.
- [Rybski et al., 2010] Rybski, P. E., Huber, D., Morris, D. D., and Hoffman, R. (2010). Visual classification of coarse vehicle orientation using histogram of oriented gradients features. In *2010 IEEE Intelligent vehicles symposium*, pages 921–928. IEEE.
- [Taner et al., 2023] Taner, A., Mengstu, M. T., Selvi, K. Ç., Duran, H., Kabaş, Ö., Gür, İ., Karaköse, T., and Gheorghita, N.-E. (2023). Multiclass apple varieties classification using



machine learning with histogram of oriented gradient and color moments. *Applied Sciences*, 13(13):7682.