CONCATS: MULTI-BREED CAT DETECTION AND CLASSIFICATION IN COMPOSITE IMAGES USING YOLO

Afsana Salahudeen, Batool Taraif, Raghad Alsalman

Columbia University

ABSTRACT

Cats are beloved companions among the most popular pets worldwide, and are cherished for their diverse breeds and unique characteristics. Recent advancements in image classification have made breed identification possible. However, most prior research focuses on single-object images, limiting their applicability to real-world scenarios with pictures that have multiple objects. Our paper explores the use of the YOLO (You Only Look Once) object detection framework to detect and classify cat breeds in composite images containing multiple cats. We used YOLO for object detection on a curated dataset that represents 10 different breeds and used these results to feed in the detected cats into a cat breed classifier.

1. INTRODUCTION

Image classification is a complex task in computer vision because it requires the model to distinguish between classes with based on subtle differences in the image. One practical application of this task, which we explore in this report, is the classification of multiple cat breeds from images. The objective is to detect all the cats in an image and classify each individual cat's breed. Cats share many visual similarities, making it challenging to accurately classify different breeds without a model capable of recognizing intricate patterns, textures, and shapes, all of which are critical factors in cat breed classification. Moreover, variations in lighting, poses, and complex backgrounds further complicate the task. Since the images used in this project often contain multiple cats of different breeds, the problem becomes even more intricate, highlighting the need for a precise and targeted solution.

The goal of this project is to design and implement a deep learning pipeline for the classification of multiple cat breeds. To achieve this, we designed a two-step approach. First, we utilized the You Only Look Once (YOLO) object detection model [1] to identify the cats in an image and create bounding boxes around the detected regions. These regions were then annotated and passed to a breed classification model, which uses residual networks, namely ResNet50 and ResNet18 [2], to classify each cat's breed. This modular design ensures that the model focuses on the most relevant parts of the image, thereby improving accuracy and efficiency. We experimented with variations in model design, training strategies, and hyperparameter tuning. Through these experiments, we aimed to identify the best-performing configuration for this pipeline and provide insights into effective techniques for this task.

Cat breed classification has numerous practical applications, such as aiding veterinarians, pet owners, and animal welfare organizations in identifying and understanding cats of different breeds. Beyond this, the methodologies discussed in this project have broader relevance and can be applied to other classification problems, such as animal species identification. By addressing the challenges of multi-class cat breed classification, this project contributes to the growing field of deep learning for computer vision.

2. RELATED WORKS

Image classification is a very prevalent part of computer vision and is one of high interest. In early image classification, approaches relied on features combined with classical machine learning methods, such as Support Vector Machines (SVM). However, the introduction of deep learning has revolutionized this field. Deep learning allows for convolutional neural networks (CNN) to extract hierarchical features. In recent years, several studies have explored breed classification using deep learning techniques. A domestic cat breed classification study used VGGNet and Inception-v3 for cat breed recognition, and achieved an accuracy of 84%. [3] There have been many studies for dog breed classification as well, including studies that use YOLO for dog face detection to categorize different dog breeds. [4] [5]. However, although there have been many works on breed classification, there's far less works that explore identifying multiple animal breeds in the same image.

In order to detect multiple cats of different breeds within a single image, we used the YOLO model [1] which brought forth a new-age object detection system that allows for real time object detection by combining object localization and classification in a single forward pass. The latest development of YOLO, YOLOv8 which is used in our model, can detect multiple objects within an image with high speed and accuracy. This model was suitable for our task of identifying cats in images because of YOLO's ability to pinpoint many objects in cluttered environments. This aligned well with our task of analyzing images containing multiple cats of different breeds.

For the classification portion, residual networks (ResNets) [2] offer robustness and scalability depending on the dataset and model requirements. ResNet50 is a commonly used ResNet architecture that utilizes residual connections to address the vanishing gradient problem, which is critical for training deep networks effectively. This model is particularly effective for classification tasks that require precise image analysis, where subtle differences between images can significantly impact the classification result.

This project builds upon these advancements by integrating YOLO for object detection and ResNet architectures for classification. While the individual components of our pipeline already exist, the combination of these methods specifically tailored for the multi-class cat breed classification problem offers a unique contribution to the field. Furthermore, we experiment with multiple variants of ResNet, different optimizers, and training strategies to evaluate their effectiveness. This approach not only addresses the challenges of the task but also provides insights into how state of the art methods can be optimized for image classification in new ways.

3. METHODS

3.1. Data Collection and Augmentation

The dataset was sourced from Kaggle[6], containing high-resolution images of individual cats categorized by breed. The Cat Breeds Dataset contains a collection of cat images captured using the PetFinder API. The dataset originally contains over 67 cat breeds, but we picked out 10 different breeds that have unique visual characteristics like differing colors and patterns: Domestic Short Hair, Persian, Bengal, Siamese, Calico, Maine Coon, Bombay, Tortoiseshell, Russian Blue, and Turkish Angora. To simulate scenarios with multiple cats of different breeds in a single image, we created composite images by concatenating the sourced cat images, as presented in Figure 1. Each image was resized to 224x224 pixels to ensure consistency and compatibility with the classification network. Composite images were generated in three configurations:

- 2-breed images: horizontal concatenation of two cats
- 3-breed images: horizontal concatenation of three cats
- 4-breed images: a 2x2 grid of four cats, ordered top-left, top-right, bottom-left, and bottom-right

For each composite image, a JSON file documented the breed labels for each section. Images were processed in batches and saved in structured directories based on their configuration type.

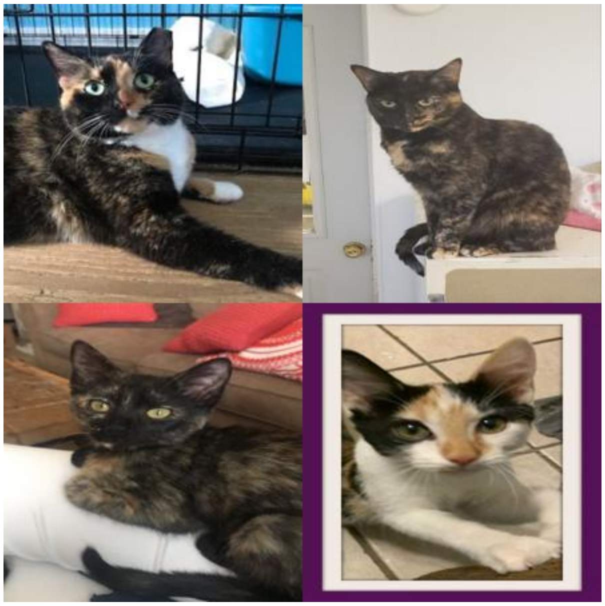

Fig. 1. Example of a four-breed composite image. From left to right, top to bottom, the breeds are Calico, Tortoiseshell, Tortoiseshell, and Calico

3.2. Framework

The pipeline for this project integrates object detection and classification in a two step approach. First, the YOLOv8 model is used to detect cats in an image, generating bounding boxes for all detected regions containing cats. These bounding boxes are then cropped accordingly and passed to a classification network for breed identification. The classification network leverages ResNet architectures, to classify each detected cat into one of the predefined breeds. This design ensures that the model focuses on relevant regions of the image.

The next step is then object detection using YOLO. In particular, we used YOLOv8 which has high speed and accuracy in detecting multiple objects in cluttered environments. YOLOv8 was pretrained on the COCO dataset, which includes a ’cat’ class, and this ID was used to find all cats in the image. All images from the two-breed, three-breed, and four-breed configuration types were used for YOLOv8’s inference, it outputs bounding boxes for each detected cat in an image, which are then used to isolate individual regions for classification, and a JSON file was written to document all annotations. Because we concatenated images together with different breeds, we needed a way to update the labels for the bounding boxes so that the correct breed is saved for each region. We hard-coded the original image dimensions for the concatenated image so that we had general bounds for each of the two-breed, three-breed, and four-breed sets so that the labels were all associated with their proper original images. Then, once YOLO detected a cat in the image, we checked where the bounding box falls in the concatenated image and it gets the breed label of the section the bounding box is annotated in. This approach allows the system to effectively handle images that contain multiple cats of different breeds. Finally, we update the JSON to write the labels for its corresponding bounding box. To prepare the data for training in the next portion, all detected cat data including the cropped cat bounding boxes and labels were shuffled and split into 80% training and 20% testing datasets.

The second step involves classification using ResNet architectures. ResNet50 is structured as a stack of residual blocks, where each block contains convolutional layers and a skip connection that bypasses one or more layers. This skip connection adds the input of the block to its output, ensuring that gradients flow effectively during backpropagation. This also helps address the vanishing gradient problem, making ResNet50 highly effective on complex tasks. We used a custom classification head with a dense layer with 512 units and ReLU activation.

To preprocess the data, we resized and edited the coloring of the image to standardize it, and loaded the label for each. The cropped bounding box regions are resized to 224x224 pixels which is necessary for the input in a ResNet model. Both ResNet50 and ResNet18 were chosen for implementation and comparison.

This pipeline, integrating YOLOv8 for object detection and ResNet architectures for classification, represents a novel combination specifically tailored for the multi-class cat breed classification problem. By experimenting with multiple architectures, optimizers, and training strategies, we aim to provide insights into optimizing this classification task.

4. EXPERIMENTS

4.1. Single Cat Classification

Before testing the full pipeline with multi-cat composite images, we first evaluated the performance of the cat breed classification model on a dataset of single-cat images. This experiment aimed to establish a baseline for the classification performance of individual cat breeds, and ensure that the model could distinguish between the ten selected breeds effectively. For this, the ResNet50 architecture was fine-tuned on the single-cat dataset with similar configurations to the multi-cat experiment.

The ResNet50 model was initialized with pre-trained ImageNet weights, and the training setup included freezing all but the last six layers and replacing the final classification layer with a dropout layer of 0.5, and a dense layer. Freezing most of the network allowed the model to leverage general image features learned during pretraining, while fine-tuning the final layer adapted it to the cat breed classification task. The single-cat dataset contained 23,396 images with 80% allocated for training and 20% for testing. A detailed classification report provides precision, recall, and F-1 scores for each breed, which shows performance differences among breeds. Breeds like Persian and Siamese showed the highest classification accuracy, with F-1 scores exceeding 0.80, while Turkish Angora and Domestic Short Hair were more challenging to classify, thus having lower recall rates. The model achieved a 70% accuracy, meaning that it has a good ability to identify subtle breed-specific features.

Table 1. Classification Report for Single Cat Classification

Breed	**Precision**	**Recall**	**F1-Score**
Bengal	0.77	0.71	0.74
Bombay	0.71	0.71	0.71
Calico	0.67	0.78	0.72
Domestic Short Hair	0.54	0.50	0.52
Maine Coon	0.55	0.45	0.50
Persian	0.90	0.79	0.84
Russian Blue	0.76	0.84	0.80
Siamese	0.70	0.92	0.79
Tortoiseshell	0.72	0.71	0.71
Turkish Angora	0.77	0.30	0.43
Accuracy		0.70	
Macro Avg	0.71	0.67	0.68
Weighted Avg	0.71	0.70	0.70

4.2. Multi-Cat Classification

Experiments were done on the classification portion of the pipeline. Both ResNet50 and ResNet18 models were initialized with ImageNet pre-trained weights and fine tuned on the cropped cat breed dataset using cross entropy loss.

The first iteration was done on the ResNet50 model with all layers except the last 6 layers frozen and the last layer replaced with a dropout layer of 0.5 and linear layers so that the model's output layer is aligned with the number of breeds in the label set. The optimizer used was Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and momentum of 0.8. Momentum allows the model to converge faster by using a combination of the current gradient and a fraction of the previous update, smoothing out oscillations. Along with this, 20 epochs were used.

The second iteration was also done on the ResNet50 model using the same specifications as the previous model except for the optimizer and epochs. In this instance, the optimizer used was Adaptive Moment Estimation (Adam) with a learning rate of 0.001 and 15 epochs.

The final model used the ResNet18 architecture. All layers except the last 3 layers were frozen and the last layer was replaced with a dropout layer of 0.3 and linear layers so that the model's output layer is aligned with the number of breeds in the label set. This model also used 15 epochs and was trained using an Adam optimizer with a learning rate of 0.001. These model trainings allowed us to leverage the pre-trained

ResNet model and analyze which specifications worked best for our project.

5. RESULTS

5.1. Single-Cat Classification

The ResNet50 model was first evaluated on single-cat images to establish a baseline performance for cat breed classification. The model achieved an overall accuracy of 70%, demonstrating its ability to differentiate between the 10 selected breeds. Detailed performance metrics are presented in Table 1. Breeds like Persian and Siamese performed best, with F1-scores exceeding 0.80, likely due to their distinct visual features. However, breeds such as Turkish Angora and Domestic Short Hair posed challenges, showing lower recall rates due to subtle or inconsistent distinguishing features. Another note is that considering the dataset was taken from a pet adoption website, a lot of cats with unknown breeds will get labeled as Domestic Short Hair. In hindsight, it would have been better to not include Domestic Short Hair, as there was a lot of mislabeling in the dataset, and DSH in general is not a breed with distinctive visuals. This became even more apparent after creating a confusion matrix with the results from the classification model.

5.2. Multi-Cat Classification

For the multi-cat classification task, the YOLOv8 model successfully detected multiple cats in the composite images with a detection rate of 79%. The bounding boxes generated by YOLO were used to crop individual cat regions, which were then classified using ResNet50 and ResNet18 models. In experiments with the ResNet50 model, two variations were tested. One with the SDG optimizer, which achieved an accuracy of 68% on the test set, and one with the Adam optimizer, which achieved an accuracy of 72%.

The ResNet model, which was trained with the Adam optimizer, achieved an accuracy of 69%. Although it is slightly less accurate than the Adam-optimized ResNet50 model, it is still performing well considering it's reduced computational requirements. Table 2 summarizes the performance metrics across different configurations, highlighting the strengths of each model and training strategy.

Table 2. Performance Metrics for Multi-Cat Classification

Model	**Optimizer**	**Epochs**	**Accuracy (%)**
ResNet50	SGD	20	68
ResNet50	Adam	15	72
ResNet18	Adam	15	69

6. DISCUSSION

The results highlight the effectiveness of a two-step pipeline integrating YOLOv8 for object detection and ResNet architectures for classification in multi-cat breed identification tasks. Using YOLOv8's high accuracy in cluttered environments, the pipeline successfully isolated individual cats in composite images, giving us more accurate breed classification. Some of the insights we got from the single-cat classification was how the baseline experiment showed us the importance of breed-specific features for classification accuracy. Breeds with distinctive visual traits, like Persian and Siamese cats, achieved higher metrics, while less visually distinctive breeds like Domestic Short Hairs presented classification difficulties. For Multi-Cat Classification, the composite images introduced challenges with the bounding box annotations and label accuracies. Manually adjusting bounding box labels ensured consistency and contributed to the effectiveness of the pipeline. Although YOLOv8 mostly performed well in detecting cats in the composite images, some failure cases were observed. Figure 2 shows the ground-truth cat detections in green, and the YOLO detections in red boxes. In most failure cases, the cat detection failed due to the presence of a human in the image, or if the cat was really far away from the camera. We saw that ResNet50, particularly when fine-tuned with Adam, outperformed other configurations. Overall, this study demonstrated that combining object detection and classification in a deep learning pipeline can effectively address challenges in multi-object image classification. This methodology can be applied more broadly in other applications, like

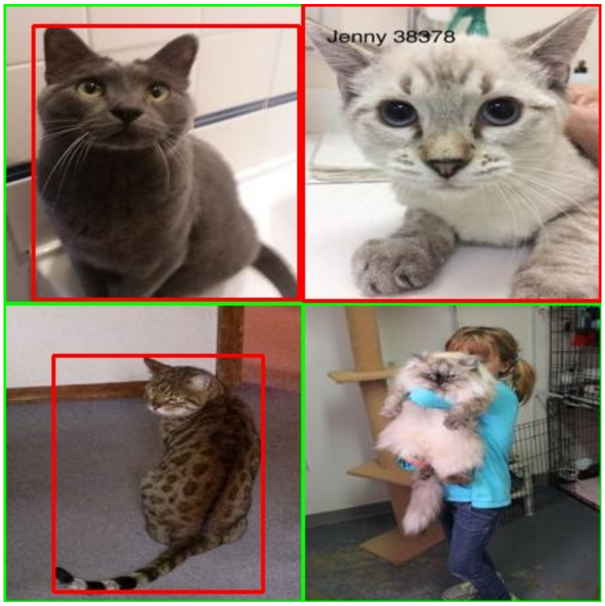

Fig. 2. Example of a four-breed composite image after being run through YOLO. The cat on the bottom right was not detected.

animal species identification, or any other multi-class classification task. Further work could explore the inclusion of additional breeds, further optimization of hyperparameters, or integrating more advanced architectures.

7. CONCLUSION

This paper presented a novel approach for multi-breed cat detection and classification in composite images, using YOLOv8 for object detection and ResNet architectures for classification. The two-step pipeline had great performance in challenging real-world scenarios with cluttered environments and multiple objects. This shows the potential of integrating state-of-the-art object detection and classification models to address complex computer vision tasks like breed classification.

While the pipeline achieved promising results, there were failure cases, such as missed detections in crowded settings where more than one cat was detected, or where a human was detected as well. Future work should focus on enhancing detection accuracy, incorporating more diverse datasets, and exploring advanced architectures to further refine the pipeline.

By contributing insights into multi-class object detection and classification, this study lays a foundation for future research and applications in animal identification and beyond.

8. AUTHOR CONTRIBUTION STATEMENT

Afsana Salahudeen: Implemented the YOLOv8 object detection module and fine-tuned the ResNet models for classification. Prepared data with the object detection results. Debugged and optimized the initial multi-cat ResNet50 implementation. Conducted the multi-cat classification experiments with ResNet50 and ResNet18.

Batool Taraif: Conceptualized the research problem, designed the overall pipeline architecture, implemented Single Cat Classification with ResNet50. Conducted initial experiment with ResNet50 for multi-cat classification, identifying challenging areas before debugging and refinement by Afsana. Added the generation of images with bounding boxes to identify missed detections and explored which breeds were frequently missed by YOLO.

Raghad Al-Salman: Curated the dataset, developed the composite image generation methodology and ensured that it can be processed through the classification architectures, helped with debugging code, created the presentation slides for the project.

9. REFERENCES

[1] J Redmon, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Ruihao Zhang, “Classification and identification of domestic catsbased on deep learning,” in *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, 2021, pp. 106–110.

[4] Akash Varshney, Abhay Katiyar, Aman Kumar Singh, and Surendra Singh Chauhan, “Dog breed classification using deep learning,” in *2021 International Conference on Intelligent Technologies (CONIT)*, 2021, pp. 1–5.

[5] Changqing Wang, Jiaxiang Wang, Quancheng Du, and Xiangyu Yang, “Dog breed classification based on deep learning,” in *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, 2020, pp. 209–212.

[6] ma7555, “Cat breeds dataset,” 2024, Accessed: 2024-12-06.