Integrating Object Detection and Deep Convolutional Neural Networks for Cat Breed Classification

Yiming Feng
College of Engineering, University of California, Davis, California, 95616, U.S.A.

Keywords: Cat Breed Classification, Deep Learning, Convolutional Neural Network.

Abstract: This study presents a novel approach to cat breed identification using a combination of object detection and deep learning classification models. The project's primary objective is to assist animal shelters and adoption centers by accurately identifying the breeds of homeless cats from images, thereby enhancing the cats' chances of finding suitable homes and facilitating targeted healthcare. This work utilized the convolutional neural network model, pre-trained on ImageNet, and integrated it with OpenCV for initial cat detection. The dataset, comprising five major cat breeds—Calico, Persian, Siamese, Tortoiseshell, and Tuxedo—was subjected to rigorous preprocessing, including image and metadata matching, cat detection, data augmentation, and rescaling. The model was trained and tested for breed classification, achieving an impressive accuracy of 87%. The integration of detection and classification not only improved the model's focus on relevant image features but also bolstered its robustness against background noise and variations in image quality. The findings underscore the potential of deep learning in animal breed identification, offering a scalable solution for broader applications in animal care and research.

1 INTRODUCTION

The identification of cat breeds is crucial for animal hospitals and adoption shelters. For hospitals, cat breeds are important because different cat breeds may have distinct common health problems (Zhang, 2020). Each breed may come with unique sets of genetic predispositions, making certain breeds more vulnerable to specific alignment. By identifying cat breeds, veterinarians can decide which conditions to monitor and offer targeted preventative care. The breed identification can enhance diagnostic accuracy, expedite treatment, and ultimately lead to better health outcomes for cats (Hamdi, 2023). Furthermore, understanding the behavioral tendencies of different breeds allows the hospital to ensure a more comfortable experience for both the cat and the owners.

From the shelter's perspective, breed identification is crucial in facilitating successful adoption. Each breed had distinct characteristics and personality traits, activity levels, and care requirements. By accurately identifying a cat's breed, shelters can provide potential adopters with more detailed information to help them select the cat that matches their lifestyle and expectations. This not only increases the probability of successful, long-term adoption but also reduces the risk of the cat being returned due to mismatched expectations. Additionally, breed identification can help manage the population. For instance, if a particular breed is considered to be overpopulation, the shelter can implement specific breeding control measures. Conversely, for any rare or endangered breeds, shelters can focus on preservation efforts.

Deep learning is particularly well-suited for tasks like cat breed identification due to its ability to process large datasets and detect complicated patterns that differentiate breeds (LeCun, 2015). Models like Visual Geometry Group (VGG)16 excel at automatically learning features from images, allowing the system to identify subtle distinctions in characteristics like fur texture, eye shape, and markings (Simonyan, 2014). By using tools such as PyTorch and OpenCV, deep learning can effectively focus on and analyze specific features within images, which is crucial for accurate detection and classification (Paszke, 2019). Image augmentation is applied for helping the model avoid overfitting and improving its ability to generalize to new, unseen images. Moreover, deep learning models are highly scalable, meaning they can adapt to larger datasets as

Feng, Y.
Integrating Object Detection and Deep Convolutional Neural Networks for Cat Breed Classification.
DOI: 10.5220/0013337500004558

In *Proceedings of the 1st International Conference on Modern Logistics and Supply Chain Management (MLSCM 2024)*, pages 433-438
ISBN: 978-989-758-738-2

more breeds are added, ensuring that the model continues to perform well as the scope of the task expands (Shrestha, 2019). This combination of large-scale data handling, feature learning, and scalability makes deep learning the ideal choice for the project

Among existing works, improving classification model structures often involves architectural innovations, transfer learning, and data augmentation to enhance accuracy and efficiency (Shrestha, 2019). Background removal is a critical technique that further boosts classification performance by isolating the cat from irrelevant visual noise, allowing the model to focus on essential features like fur patterns and facial structure. This not only leads to more accurate breed identification but also improves the detection of the cat's position within an image, reducing errors caused by complex or cluttered backgrounds. By refining the focus on the subject, background removal ensures that the model can more effectively analyze and classify the relevant details (Alzubaidi, 2021).

This paper contributes to the field by integrating detection and classification techniques to enhance the accuracy of cat breed identification. By combining these two approaches, the model not only accurately identifies the breed of a cat but also precisely detects its position within the image. This dual approach ensures that the model focuses on the most relevant features for classification, reducing errors and improving overall performance. The paper demonstrates that this combination leads to a more robust and reliable system for breed identification, particularly in complex visual environments where background noise and positional inaccuracies could otherwise hinder classification accuracy.

2 METHOD

2.1 Dataset and Preprocessing

This paper worked with a dataset containing images of five cat breeds: Calico, Persian, Siamese, Tortoiseshell, and Tuxedo (MA, 2019). This dataset was put together to train a model that can identify these breeds. The dataset is moderate in size and provides enough examples of each breed for the model to learn meaningful patterns.

This work took several steps to get the dataset ready for training:

(1) Image and Metadata Matching: This work started by loading the data into a Pandas DataFrame. The next task was to find the actual image files. To do this, this work scanned through the new_images folder and matched each file to the respective ID by checking the first part of the filename. If an image file didn't have a match, that row is excluded from the dataset to make sure everything lined up.

(2) Cat Detection: This work wanted the model to focus on the cats, so YOLOv5 is leveraged to detect cats in the images (Wu, 2021). This model is trained to recognize objects like cats, so it was able to highlight the areas of interest (Liu, 2020). By running the detection, images are filtered out that didn't contain cats and processed only the ones that did.

(3) Splitting the Data: After cleaning the data, this work split it into training and validation sets. This work used an 80/20 split, meaning that 80% of the images were used for training the model, and the remaining 20% were kept aside to test how well the model works on unseen data.

(4) Data Augmentation: To make the model more robust, this work applied data augmentation. For example, images are randomly rotated, zoomed in or out, shifted, and flipped. This helps the model become more adaptable and prevents it from overfitting to the training data, including 1. Rescaling: this work rescaled the pixel values to fall between 0 and 1 for better training. 2. Rotation, Shifting, and Flipping: this work added random changes to the image position and orientation to mimic different viewing angles. 3. Zooming: this work zoomed in and out to simulate different distances from the camera.

(5) For validation, augmentations are not conducted; images are only rescaled so the model could be tested on the original, unaltered images.

(6) Data Generators: Lastly, this work created data generators that continuously feed batches of images and their labels into the model during training. This is a more efficient way to handle large datasets since it doesn't load all the data into memory simultaneously.

By following these steps, this work ensured the dataset was properly prepared for training, giving the model a good balance of variety and consistency for learning.

2.2 Model Architecture

2.2.1 VGG16 for Classification

For the classification component of this project, this work employs the VGG16 architecture, which has proven to be a robust and effective convolutional neural network (CNN) for image classification tasks. Originally introduced by Simonyan and Zisserman in 2014, VGG16 is characterized by its uniform architecture, utilizing small 3x3 convolutional filters across 16 layers, followed by max-pooling layers.

This design enables the network to progressively capture complex hierarchical features, a critical advantage when distinguishing between visually similar categories such as different cat breeds.

One of the primary benefits of VGG16 lies in its compatibility with transfer learning. By leveraging pre-trained weights from large-scale datasets like ImageNet, the model inherits a sophisticated understanding of generic image features, allowing it to adapt efficiently to more specific tasks such as breed identification. The pre-trained VGG16 architecture thus serves as a feature extractor, reducing the need for extensive training while improving accuracy and convergence time. Despite its large size and memory demands, VGG16 offers a high level of performance in extracting the intricate visual patterns required for fine-grained classification.

2.2.2 OpenCV for Detection

In parallel to classification, a detection model has been integrated to pre-process the input images by isolating regions of interest (ROI). This step is essential, as it mitigates the impact of irrelevant image regions that could mislead the classifier, such as background noise or extraneous objects. By focusing the classification model on the relevant portions of the image—typically the cat's face or body—the detection model enhances the overall performance of the classification system.

The detection model operates using OpenCV, which facilitates object detection through techniques like Haar cascades or sliding window methods. Once the detection phase identifies the relevant ROI, these regions are cropped, resized, and normalized before being passed into the VGG16 model for classification. This pre-processing step ensures that the classifier operates on refined inputs, leading to improved precision and recall metrics.

By integrating detection and classification, the model architecture optimizes the flow of data. The detection module reduces noise and irrelevant features, effectively acting as a filter, and allowing the VGG16 classifier to focus solely on breed-specific features. This layered structure has led to significant improvements in accuracy, particularly when evaluating metrics such as precision, recall, and F1-score. The combined system thus demonstrates enhanced robustness and generalization ability compared to a standalone classifier.

This integration reflects a synergistic approach where detection and classification work together to improve the model's overall efficacy in recognizing and categorizing different cat breeds.

2.3 Evaluation Metrics

This work leverages representative classification indexes for performance evaluation. Including accuracy, precision, recall, F1-score, confusion matrix, and Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC)

3 RESULTS

3.1 Experimental Details

In the realm of hyperparameters and model architecture, this work employed the VGG16 base model, pre-trained on ImageNet dataset. To adapt this model to the specific task, custom layers are introduced: a global average pooling layer, followed by a Dense layer with 1024 units with a Rectified Linear Unit (ReLU) activation function to introduce non-linearity. The final Dense layer was designed with units corresponding to the number of cat breeds, utilizing a Softmax activation function to facilitate classification.

For the optimization, the Adam optimizer is used for fine-tuning. The initial training utilized a default learning rate of 0.001, while for fine-tuning, this paper adopted the learning rate of 1e-4.

The training process was with a batch size of 32. This work allocated 40 epochs for the initial training phase and 30 epochs for fine-tuning. The best-performing model based on validation accuracy is saved and the early stopping strategy is conducted.

The hardware setup included an NVIDIA GeForce RTX 3070 GPU. The software environment consisted of Ubuntu 22.04, CUDA 11, with key Python packages such as TensorFlow 2.17, Keras 3.4, and PyTorch 2.1.

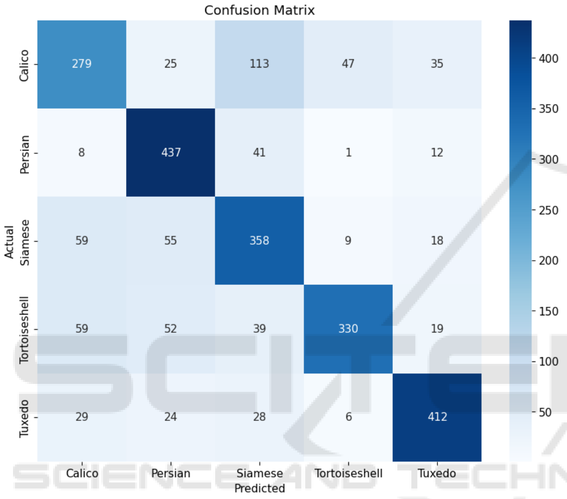
3.2 Performance Comparison

In the results section, the study reported a significant enhancement in cat breed identification accuracy by the integration of object detection and breed classification models.

The initial classification model, utilizing only the VGG16 architecture, achieved an accuracy of 73%, as demonstrated in Table 1, Figure 1 and Figure 2. However, by incorporating YOLOv5 for object detection prior to breed classification, the model's accuracy improved to 87%, as illustrated in Table 2, Figure 3 and Figure 4. This increase underscores the effectiveness of focusing the classification model on the detected regions of interest within the images.

Table 1: Performance using classification model only.

	Precision	Recall	F1-score
Calico	0.6429	0.5591	0.5981
Persian	0.7369	0.8758	0.8004
Siamese	0.6183	0.7174	0.6642
Tortoiseshell	0.8397	0.6613	0.7399
Tuxedo	0.8306	0.8257	0.8281
Average	0.7337	0.7279	0.7261
Average Acc	0.7728		



	Calico	Persian	Siamese	Tortoiseshell	Tuxedo
Calico	279	25	113	47	35
Persian	8	437	41	1	12
Siamese	59	55	358	9	18
Tortoiseshell	59	52	39	330	19
Tuxedo	29	24	28	6	412

Figure 1: Confusion matrix using classification model only (Figure Credits: Original).

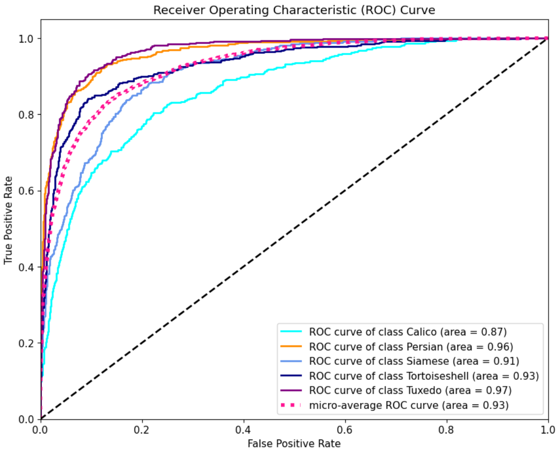
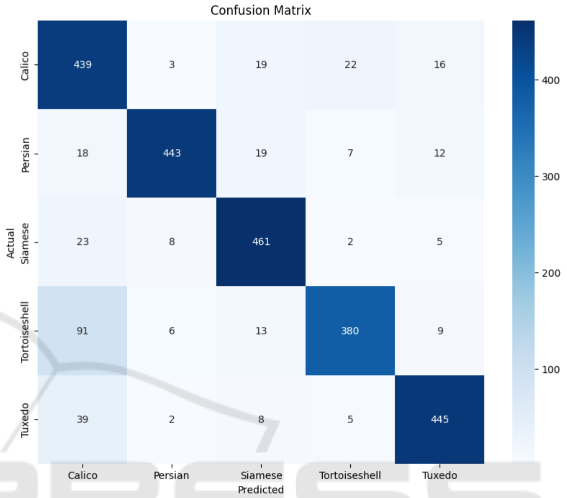


Figure 2: ROC curve using classification model only (Figure Credits: Original).

Table 2: Performance using both detection and classification models.

	Precision	Recall	F1-score
Calico	0.7197	0.8798	0.7917
Persian	0.9589	0.8878	0.9220
Siamese	0.8865	0.9238	0.9048
Tortoiseshell	0.9135	0.7615	0.8306
Tuxedo	0.9138	0.8918	0.9026
Average	0.8689	0.8689	0.8689
Average Acc	0.8689		



	Calico	Persian	Siamese	Tortoiseshell	Tuxedo
Calico	439	3	19	22	16
Persian	18	443	19	7	12
Siamese	23	8	461	2	5
Tortoiseshell	91	6	13	380	9
Tuxedo	39	2	8	5	445

Figure 3: Confusion matrix using both detection and classification models (Figure Credits: Original).

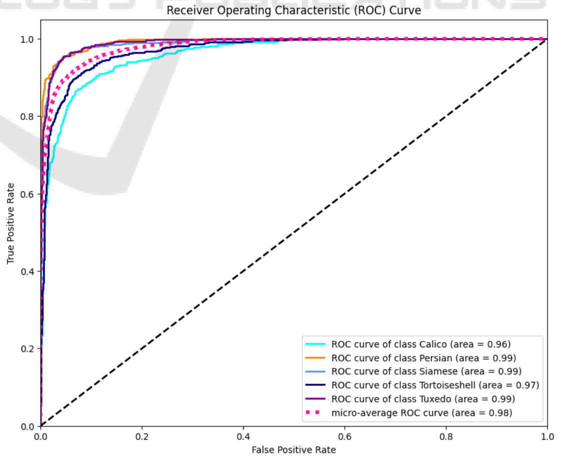


Figure 4: ROC curve using both detection and classification models (Figure Credits: Original).

4 DISCUSSIONS

4.1 Detection and Classification Performance

The combined approach of using YOLOv5 for object detection and VGG16 for breed classification provided insightful results. The detection model effectively identified cats in the images, isolating them from backgrounds or other objects. Once detected, the classification model accurately identified the cat breeds. In many cases, the pipeline worked as expected, especially when the images clearly depict the cats in non-occluded environments.

However, some limitations emerged. For images where the cat's pose or lighting conditions were less ideal, the classification model struggled to correctly distinguish between certain breeds, particularly those with visually similar traits (e.g., Calico and Tortoiseshell). This could be attributed to the subtle visual differences that define these breeds, which may not have been adequately represented in the training data. Additionally, the quality of the bounding boxes generated by YOLOv5 impacted the classification outcome. Incorrect or incomplete detection (e.g., when only part of the cat is detected) negatively affected the subsequent breed classification.

Overall, combining detection with classification yielded promising results, demonstrating that integrating object detection with breed classification can be a powerful method for handling more complex datasets where target objects need to be isolated before classification.

4.2 Limitations

Despite the promising results, this study faces several limitations: (1) Data Imbalance: Some breeds, particularly less common ones like Tortoiseshell, were underrepresented in the dataset. This likely led to a bias in classification, with the model performing better on breeds that were more frequent in the training data. (2) Detection Quality: YOLOv5 performed well in most cases, but when the detected bounding box was inaccurate or incomplete, the subsequent classification suffered. The pipeline is heavily reliant on high-quality detection to ensure that only relevant parts of the image are passed to the classifier. (3) Visual Similarity Between Breeds: Some breeds, such as Calico and Tortoiseshell, have overlapping visual characteristics, making it challenging for the classifier to distinguish between them. The model struggled more with images where breed-specific traits were not prominently visible. (4) Image Quality and Pose Variability: Images with poor lighting, unusual angles, or occluded cats affected both detection and classification accuracy. This highlights a limitation in dealing with real-world conditions, where images are often far from perfect. (5) Overfitting: While data augmentation helped reduce overfitting, there was still some indication that the model performed better on the training data than on the validation data, suggesting that more diverse augmentation techniques or larger datasets are needed.

4.3 Future Work

To build on the foundation of this study, several avenues for future research and improvement are proposed: (1) Increasing Dataset Diversity: Expanding the dataset to include more samples for underrepresented breeds and more varied real-world conditions (e.g., different lighting, poses, and backgrounds) will help create a more robust model capable of generalizing across various scenarios. (2) Advanced Data Augmentation: Implementing more sophisticated augmentation techniques (e.g., adding synthetic occlusions, more radical lighting changes) would help the model become more resilient to common visual challenges in real-world images. Additionally, techniques such as MixUp or CutMix can help address class imbalances. (3) Improved Detection Models: Exploring more advanced object detection models or fine-tuning YOLOv5 to better capture the full extent of each cat's body, regardless of pose, could improve the quality of the bounding boxes and enhance breed classification performance. (4) Multi-task Learning: Rather than treating detection and classification as separate stages, a unified multi-task model could be developed to detect cats and predict their breeds simultaneously. This may reduce the errors that arise from passing detections between models. (5) Fine-tuning Pretrained Models: The classification model could benefit from further fine-tuning on domain-specific data. Given that VGG16 was originally trained on a general dataset (ImageNet), additional fine-tuning on a larger, more diverse set of cat images could improve its ability to differentiate between similar breeds. (6) Handling Occlusions and Partial Detections: Introducing models that can handle partial detections, or training the system to classify cats based on partial features, could be an interesting research direction, particularly for real-world scenarios like shelter environments, where cats are often not fully visible. (7) Practical Applications: The findings from this study can be applied to real-world use cases such as assisting shelters in identifying cat breeds for potential adopters or helping veterinarians identify breeds for better healthcare recommendations. Expanding this approach to cover more breeds or integrating it into mobile apps could be practical next steps.

By addressing these limitations and exploring future directions, the study could be further enhanced to provide a robust and widely applicable tool for cat breed identification in various contexts.

5 CONCLUSIONS

This study combined two techniques to identify cat breeds from images: object detection and breed classification. This work first used YOLOv5 to locate cats within images, separating them from the background and other objects. After detecting the cats, the author employed the VGG16 model to classify each detected cat into one of five breeds: Calico, Persian, Siamese, Tortoiseshell, and Tuxedo. This method ensured that the breed classification was based solely on the detected cat regions, which improved the accuracy of the predictions. The combined approach effectively identified cats and classified their breeds in most cases. While the overall results were positive, this work encountered some difficulties with images where cats were partially visible or where breeds were visually similar. These issues occasionally led to less accurate breed classifications. Despite these challenges, the integration of detection and classification proved to be a useful method for handling complex image data. This study highlights the effectiveness of combining object detection with breed classification to improve image analysis. Although the results are promising, there are areas for improvement. Future research should focus on enhancing detection accuracy, expanding the dataset, and exploring more advanced models. These steps could lead to even better performance and practical applications, such as aiding animal shelters in identifying and managing cat breeds more efficiently.

REFERENCES

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*, 1-74.

Hamdi, E. B., Sunaryo, J. A., & Prasetyo, S. Y. 2023. Fusion of pretrained CNN models for cat breed classification: A comparative study. In *E3S Web of Conferences*. 426, 01014.

LeCun, Y., Bengio, Y., & Hinton, G. 2015. Deep learning. *nature*, *521*(7553), 436-444.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*, 261-318.

MA, Cat Breeds Dataset, 2019 URL: https://www.kaggle.com/datasets/ma7555/cat-breeds-dataset. Last Accessed:2024/09/09.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Simonyan, K., & Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Shrestha, A., & Mahmood, A. 2019. Review of deep learning algorithms and architectures. *IEEE access*, *7*, 53040-53065.

Wu, W., Liu, H., Li, L., Long, Y., Wang, X., Wang, Z., ... & Chang, Y. 2021. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PloS one*, *16*(10), e0259283.

Zhang, Y., Gao, J., & Zhou, H. 2020. Breeds classification with deep convolutional neural network. In *Proceedings of the 2020 12th international conference on machine learning and computing*, 145-151.