

# EXDARKHIERARCHYNET: ENHANCING LOW-LIGHT IMAGE CLASSIFICATION WITH CNN ARCHITECTURES

**Yuesong Huang** \*

Department of Computer Science  
University of Rochester  
Rochester, NY 14627  
yhull16@u.rochester.edu

**Wentao Jiang** \*

Department of Computer Science  
Department of Mathematics  
University of Rochester  
Rochester, NY 14627  
wjiang20@u.rochester.edu

## ABSTRACT

Traditional Convolutional Neural Networks (CNNs) often falter in low-light image classification. This proposal suggests modifications to CNN architectures and specialized preprocessing techniques to enhance their performance in such conditions. Utilizing the “Exclusively Dark” image dataset<sup>1</sup>, this study aims to validate and showcase the advantages of our approach (Loh & Chan, 2019).

## 1 INTRODUCTION

The increasing application of visual data in areas such as security surveillance and autonomous driving demands robust image classification systems that can operate effectively under varied lighting conditions. Traditional Convolutional Neural Networks (CNNs) often struggle with low-light environments, encountering issues like noise amplification and detail loss that degrade classification performance. These challenges have significant real-world implications, including the potential for safety risks in autonomous vehicle navigation.

The “Exclusively Dark” dataset<sup>1</sup> (as in ‘see Loh & Chan (2019) for more information.), a collection rich in low-light images will serve as our benchmark., a collection of low-light images. We will train both traditional and modified models, comparing their performance based on metrics like accuracy, precision, recall, and many other metrics like F1-score. This evaluation will provide insights into the strengths and potential limitations of our approach.

Addressing these challenges, this paper proposes a dual-strategy enhancement to the CNN framework for low-light image classification. Firstly, we introduce a suite of preprocessing techniques tailored to mitigate the effects of darkness in images, including histogram equalization and noise reduction. Secondly, we establish a baseline CNN and then adapt and refine the architectures of established CNN models, such as VGG<sup>2</sup> (as in ‘see Simonyan & Zisserman (2015) for more information.) and ResNet<sup>3</sup> (as in ‘see He et al. (2016) for more information.). This adaptation involves incorporating specialized layers designed to extract and process low-light features more effectively.

Employing the “Exclusively Dark” image dataset<sup>1</sup>, our study benchmarks the performance of these enhanced models, assessing their capabilities through a comprehensive evaluation of accuracy, precision, recall, and F1-score metrics. Our goal extends beyond mere accuracy improvement; we aim to ensure that image-based systems remain reliable and secure, even in sub-optimal lighting conditions, thereby enhancing user safety and expanding the operational envelope of such technologies.

---

\*These authors contributed equally to this work.

<sup>1</sup><https://github.com/cs-chan/Exclusively-Dark-Image-Dataset>

<sup>2</sup><https://arxiv.org/abs/1409.1556>

<sup>3</sup><https://ieeexplore.ieee.org/document/7780459>

## 2 RELATED WORK

Our study employed the ExDark dataset<sup>1</sup>, a large collection of various-resolution images captured in extremely dark environments across 12 distinct classes. This dataset’s diversity made it suitable for advanced image processing and analysis.

**Data Volume:** The dataset contains approximately 7,363 individual records, featuring 23,711 objects across 12 classes. This section discusses how the images obscure nature in dark conditions affects classification outcomes.

**Average Dimension:** The average dimensions of images in our dataset were found to be  $819.266 \times 649.358$ . These dimensions were critical in determining the appropriate resizing and cropping parameters during preprocessing.

**RGB Statistical Description:** We provide a detailed statistical description of the RGB channels in our dataset (see Table 1). This information was foundational for our custom normalization process and was compared against standard ImageNet<sup>4</sup> (as in ‘see Deng et al. (2009) for more information.) normalization parameters.

Color Channel	Mean	Standard Deviation
Red	40.13	52.63
Green	32.12	44.80
Blue	25.95	41.23

Table 1: Statistical Description of ExDark Dataset<sup>1</sup>

For comparison, the default ImageNet<sup>4</sup> normalization parameters are mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225].

**Data Quality:** The data quality vary in the Dataset, as you can see in Table 1, the RGB is very low.

As shown in Figure 1 below, only a single class “Car” is involved, but images even can not be recognized by human eyes. This is a challenge for our model to classify the images.



Figure 1: Image with “Car” Class



Figure 2: “Bottle” and “People” Class

In Figure 2, the resolution of this image is very low, and the mouth of the bottle and the dark environment next to it are integrated and have no chromatic aberration.

<sup>4</sup><https://ieeexplore.ieee.org/document/5206848>

### 3 METHODS

#### 3.1 ANNOTATION PROCESSING AND ANALYSIS

We merged, analyzed, and streamlined the dataset annotations associated with each image. This included calculating the area of objects within the images and designating the largest object in each image as the main label for simplicity and efficiency. Additionally, we developed a custom tool to extract this information, including file names, spatial details, and class types, from each image's accompanying text file. This data was then systematically organized into a spreadsheet format, compatible with the `Pandas` library for efficient processing and analysis.

#### 3.2 IMAGE AUGMENTATION

**Resizing and Cropping** : The images were first resized to a  $512 \times 512$  resolution. A dynamic approach was then adopted for cropping, where we modified the scale range of 0.8 to 1.0 and an aspect ratio range of  $3/4$  to  $4/3$  for image preprocessing, ensuring the cropped images maintained a consistent aspect ratio. The images were finally resized to  $256 \times 256$  to standardize their dimensions.

**Random Vertical Flip** : To introduce variation and augment the dataset, we implemented a random vertical flip on the images.

#### 3.3 ADVANCED IMAGE ENHANCEMENT

Advanced image enhancement techniques were applied to improve image quality:

**Contrast Limited Adaptive Histogram Equalization (CLAHE)**: Enhances image contrast with a controlled clip limit. We will mainly use this approach.

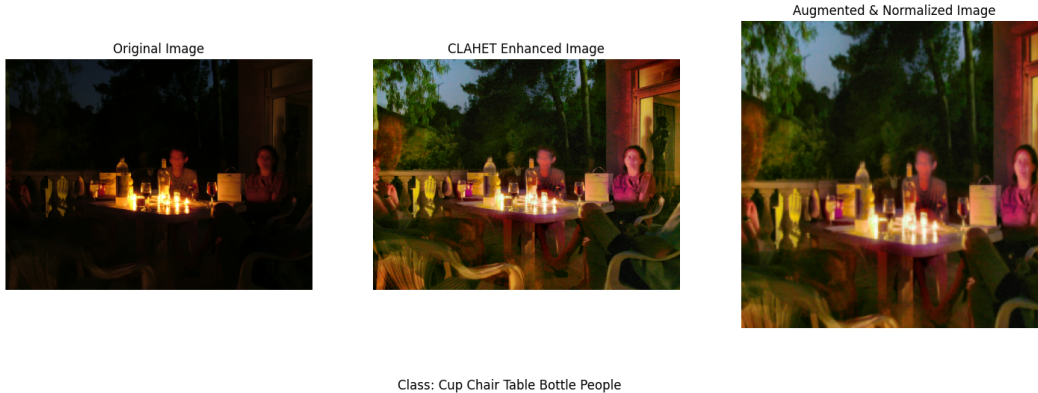


Figure 3: CLAHE

**Histogram Equalization (HE)**: Improves the contrast of the images.

**Dynamic Range Adjustment (DRA)**: Normalizes the range of pixel intensities.

**Gaussian Blur (GB)**: Reduces image noise, focusing on preserving quality.



Figure 4: Histogram Equalization

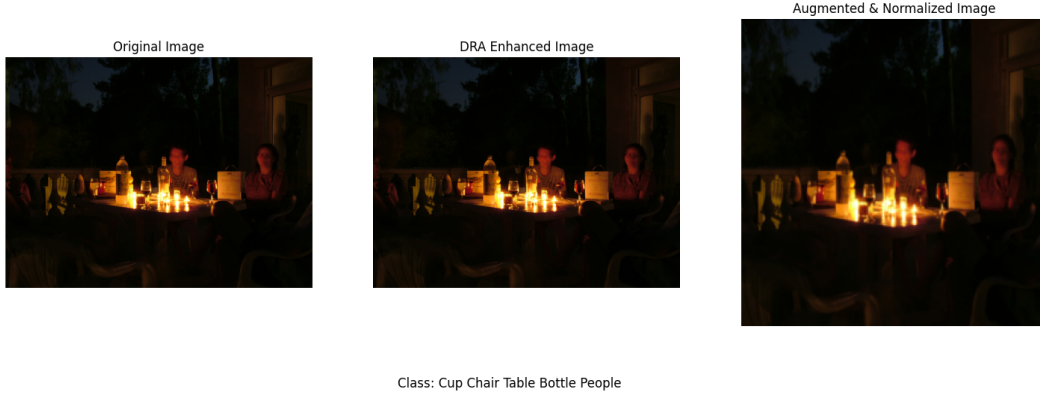


Figure 5: Histogram Equalization

### 3.4 TRANSFORMATION PIPELINE

A preprocessing pipeline integrated these methods, designed for flexibility and scalability to incorporate additional techniques as necessary. The advanced image enhancement and augmentation methods were cohesively integrated into a robust preprocessing pipeline, which was designed with flexibility and scalability in mind to accommodate additional techniques and modifications as needed.

### 3.5 NORMALIZATION

Two normalization methods were explored in our study: Custom normalization and ImageNet<sup>4</sup> normalization. The first approach utilized standard ImageNet<sup>4</sup> parameters, while the second involved computing the mean and standard deviation of our images, tailored specifically for the ExDark dataset<sup>1</sup>. Interestingly, even though our custom normalization was specifically designed for the ExDark dataset<sup>1</sup>, we observed that the standard ImageNet<sup>4</sup> parameters outperformed our method in VGG<sup>2</sup> and ResNet<sup>3</sup> models. Conversely, our custom normalization demonstrated superior performance in our specially designed baseline CNN. This finding suggests that the optimal normalization method is dependent on the model architecture and custom dataset.

### 3.6 CUSTOM DATASET PREPARATION

In processing the annotations, we prioritized the largest object in each image as the primary label, based on its spatial dimensions. This approach was adopted to enhance accuracy and reflect real-world scenarios where the most prominent object is often of the highest relevance.



Figure 6: Histogram Equalization

Our research necessitated creating custom datasets for the effective training and evaluation of various neural network models. We developed specialized datasets for binary (Creature vs. Auto), ternary (People vs. Cat vs. Dog), and multilabel classifications, each tailored to meet the specific requirements of these classification tasks.

**Stratified Splitting and Sampling:** We employed stratified sampling to divide the datasets into training and testing sets, ensuring balanced class representation. For the ternary dataset, a `WeightedRandomSampler` was used to mitigate class imbalance during training. A custom PyTorch dataset class was developed to streamline the loading and transformation of images and labels, integrating effectively with PyTorch’s `DataLoader`.

**Binary Classification Dataset:** This dataset was constructed by filtering annotations to include classes relevant for binary classification, such as **Creatures** (People, Cat, and Dog) and **Autos** (Car, Bus, Bicycle, and Motorbike). It played a crucial role in training models to distinguish between these broad categories.

**Ternary Classification Dataset:** We developed a ternary classification dataset by selecting appropriate classes from the annotations, specifically targeting “People”, “Cat”, and “Dog”. This dataset was designed to refine the classification process for images identified as “Creature” in the binary classification phase. It enables the models to further categorize a “Creature” into one of the three more specific categories, enhancing the accuracy and granularity of our classification system.

**Multilabel N-ary Classification Dataset:** Designed to handle images with 12 distinct labels, this dataset was vital for training models to simultaneously predict several classes, capturing the overlapping nature of objects in the ExDark dataset<sup>1</sup>.

### 3.7 MODELS

Our exploration of neural network models was comprehensive, focusing on three distinct architectures, each adapted for binary, ternary, and multilabel N-ary classification tasks. We chose to investigate a handcrafted CNN, and the well-established VGG19<sup>2</sup> and ResNet50<sup>3</sup> models, modifying each to meet the specific demands of the ExDark dataset<sup>1</sup>’s diverse and complex nature.

**Handcrafted CNN:** Our initial model, a custom CNN with three convolutional layers, was specifically developed to align with our dataset. However, we encountered challenges during training, as the model’s learning stagnated with a consistent loss around 0.6913, indicating underfitting. This led us to explore more sophisticated architectures.

**Modified ResNet50<sup>3</sup>:** We utilized the ResNet50<sup>3</sup> architecture, known for its effective training of deep networks through residual learning. The model was initialized with pre-trained weights, and its

final fully connected (FC) layer was adapted to accommodate the number of classes in our dataset, enhancing its suitability for our specific classification tasks.

**Modified VGG19<sup>2</sup>:** Our approach with VGG19<sup>2</sup> involved leveraging its deep and robust feature extraction capabilities. Like ResNet50<sup>3</sup>, we started with a pre-trained VGG19<sup>2</sup> model and modified its final FC layer to map to the dataset’s class structure, thus tailoring it for our classification requirements.

Each of these models - the custom CNN, modified ResNet50<sup>3</sup>, or modified VGG19<sup>2</sup> - was chosen for their unique architectures and learning capabilities. The custom CNN provided a baseline and a deeper understanding of the dataset’s complexity. In contrast, the modified versions of VGG19<sup>2</sup> and ResNet50<sup>3</sup> brought the advantage of advanced feature extraction and deep learning techniques, which were hypothesized to be more effective for the challenging conditions presented by the Ex-Dark dataset<sup>1</sup>.

## 4 EXPERIMENTS

### 4.1 HYPER-PARAMETERS

A substantial portion of our research, over 50%, was devoted to the meticulous process of identifying the most suitable hyper-parameters for our models. This critical aspect proved to be a challenging yet essential part of our study, greatly influencing model performance and accuracy.

- **Learning Rate:** A standard rate of  $1e - 5$  was used, with adjustments made for larger models.
- **Optimizer:** The Adam optimizer was employed across all models for its efficiency.
- **Criterion:** Various criteria were used depending on the classification task:
  - Binary: CrossEntropyLoss and BCEWithLogitsLoss.
  - Ternary: CrossEntropyLoss.
  - Multilabel N-ary: BCEWithLogitsLoss.
- **Sampler:** The WeightedRandomSampler was utilized to address class imbalance, particularly in the ternary classification.
- **Scale and Aspect Ratio:** We maintained a scale range of  $0.8 - 1.0$  and an aspect ratio range of  $3/4 - 4/3$  for image preprocessing.
- **Normalization:** Both ImageNet<sup>4</sup> parameters and custom RGB mean and standard deviation values were used.
- **Customized CNN Model Parameters:**
  - Convolutional layers: 3 layers.
  - Fully connected layers: 3 layers.
  - Pooling: MaxPool2d and AdaptiveAvgPool2d.
  - Activation functions: ReLU, Sigmoid, Softmax.
- **Label Selection Criteria:** The largest object’s spatial dimensions were used in each image as the primary label in enhancing classification accuracy.

### 4.2 BINARY CLASSIFICATION

**Model Introduction and Build:** Our binary classification study began with a Custom CNN, ResNet50<sup>3</sup>, and VGG19<sup>2</sup>. Each model was fine-tuned using BCEWithLogitsLoss, and CrossEntropyLoss was also evaluated for a thorough analysis. The primary objective was to accurately classify images into “Creature” and “Auto” categories.



**Classification Report and Analysis:** In our analysis, **BCEWithLogitsLoss** demonstrated superiority over **CrossEntropyLoss** in terms of precision, recall, and overall accuracy. The **Custom CNN** processed data over 20 epochs in 34 minutes, achieving an accuracy of 70.2899% and a loss of 0.4318. It showed a precision of 0.76 for ‘Creature’ and 0.66 for ‘Auto’, with a higher recall for ‘Creature’. **ResNet50<sup>3</sup>**, in 27 minutes, achieved an accuracy of 93.5818% with a loss of 0.0179, showing a superior precision of 0.95 for ‘Creature’ and 0.92 for ‘Auto’. **VGG19<sup>2</sup>**, within the same duration, reached an accuracy of 91.92% and a loss of 0.0074, slightly favoring ‘Auto’, with precision values of 0.93 and 0.91 for ‘Auto’ and ‘Creature’ respectively.

**Confusion Matrix Analysis:** The confusion matrices revealed the effectiveness of BCEWithLogitsLoss, particularly noteworthy in the **ResNet50<sup>3</sup>** model, which displayed a balanced classification ability. The **Custom CNN** accurately identified 333 instances of “Creature” while misclassifying 107, and correctly classified 346 instances of “Auto”. **ResNet50<sup>3</sup>** demonstrated a balanced performance with 478 true positives for “Creature” and 426 for “Auto”. **VGG19<sup>2</sup>** exhibited a slight bias towards “Creature”, with 481 true positives as opposed to 404 for “Auto”.

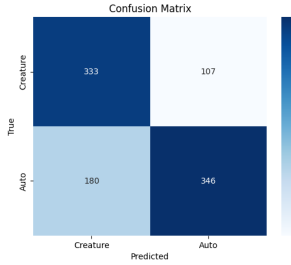


Figure 7: CNN Binary

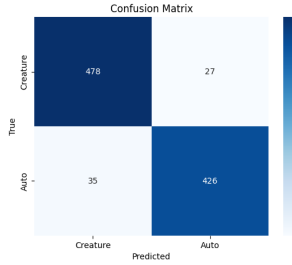


Figure 8: ResNet50 Binary

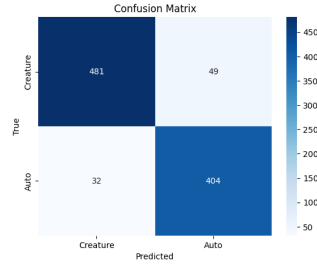


Figure 9: VGG19 Binary

**Overall Performance:** BCEWithLogitsLoss emerged as the superior criterion for binary classification, with ResNet50<sup>3</sup> slightly leading in terms of precision and accuracy.

**Experiences:** Our exploration into binary classification highlighted the importance of the right loss function choice. While both BCEWithLogitsLoss and CrossEntropyLoss were effective, BCEWithLogitsLoss was chosen for its superior performance, though CrossEntropyLoss’s slightly lesser results also provide valuable insights for future applications.

#### 4.3 TERNARY CLASSIFICATION

**Model Introduction and Build:** Expanding from binary to ternary classification, we focused on distinguishing among “People”, “Cat”, and “Dog”. We utilized a Custom CNN, ResNet50<sup>3</sup>, and VGG19<sup>2</sup>, each trained with BCEWithLogitsLoss, to tackle this more nuanced classification task.

**Classification Report and Analysis:** In the ternary classification task, each model exhibited distinct performances. The **Custom CNN** processed data over 20 epochs, ending with a loss of 0.8354 and an accuracy of 52.8265%, achieved in 34 minutes. It showed precision scores of 0.67 for “People”, 0.46 for “Cat”, and 0.31 for “Dog”, favoring “People” more. The **ResNet50<sup>3</sup>** model excelled with an accuracy of 92.3977% and a loss of 0.0560, showing high precision across all classes - 0.98 for “People”, 0.92 for “Cat”, and 0.82 for “Dog”. **VGG19<sup>2</sup>**, processing data in the same duration, achieved an accuracy of 90.64% with a loss of 0.0289. It displayed a high precision for “People”(0.95) but slightly lower for “Cat” and “Dog” at 0.78 and 0.86.

**Confusion Matrix Analysis:** The confusion matrices provided valuable insights into the classification behavior of each model in the ternary task. The **Custom CNN**’s matrix showed a tendency to correctly classify “People” with 178 true positives, but it also had notable misclassifications, as shown in the matrix below. In contrast, the **ResNet50<sup>3</sup>** model exhibited excellent balance and high accuracy, with impressive 240 true positives for “People”, and true positives for “Cat” (123) and “Dog” (111). Similarly, **VGG19<sup>2</sup>** demonstrated its proficiency, especially in classifying “People”

with 244 true positives, and performed well for “Cat” (118 true positives) and “Dog” (89 true positives), confirming its effectiveness in this challenging classification environment.

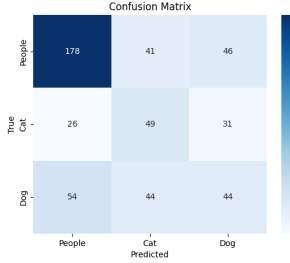


Figure 10: CNN Ternary

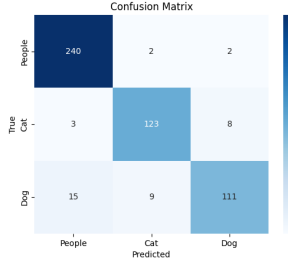


Figure 11: ResNet50 Ternary

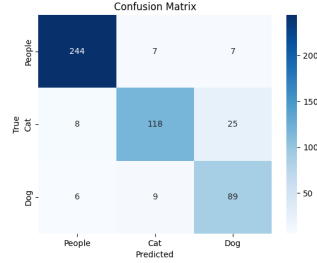


Figure 12: VGG19 Ternary

**Overall Performance:** ResNet50<sup>3</sup> and VGG19<sup>2</sup> exhibited robustness in ternary classification, efficiently managing the task’s complexity. The Custom CNN, while less accurate, provided valuable insights into the dynamics of classifying these closely related categories.

**Experiences:** The ternary classification phase underscored the challenge of dataset imbalance. The implementation of a WeightedRandomSampler was critical in achieving balanced exposure across classes, particularly benefiting the less represented “Cat” and “Dog” classes. This phase reinforced the importance of strategic sampling and advanced modeling techniques in complex classification tasks, and highlighted the necessity for refined tuning and richer datasets in multiclass machine learning scenarios.

#### 4.4 MULTILABEL N-ARY CLASSIFICATION

**Model Introduction and Build:** We ventured into the challenging realm of multilabel N-ary (12 classes) classification, employing our Custom CNN, ResNet50<sup>3</sup>, and VGG19<sup>2</sup> with BCEWithLogitsLoss. This phase marked a significant leap in complexity, requiring our models to discern multiple classes within a single image.

**Classification Report:** The **Custom CNN** struggled, managing only an 8.2154% accuracy over 20 epochs and 66 minutes of training. The model’s limitations were evident in its inconsistent precision and recall across various classes, with notable struggles in classifying “Bicycle” and “Bottle”. Conversely, **ResNet50<sup>3</sup>** demonstrated a robust performance, achieving a significant accuracy of 60.1494% in 51 minutes. It showed high precision in most classes, particularly excelling with “Boat” and “Dog”. **VGG19<sup>2</sup>** also performed commendably, reaching 54.8540% accuracy in 55 minutes. While slightly behind ResNet50<sup>3</sup>, VGG19<sup>2</sup> displayed substantial precision and recall, marking it as a capable model for multilabel classification tasks.

**Confusion Matrix Analysis:** **Custom CNN**’s revealed significant misclassifications, especially in complex categories. For instance, “Bicycle” saw only 18 true positives against 132 negatives, highlighting the model’s difficulty with finer-grained classification. Classes with more nuanced features, like “Bottle” and “Cup”, had true positives of only 0 and 47 respectively, suggesting that the CNN model struggled to distinguish between similar object categories. **ResNet50<sup>3</sup>** demonstrated a robust ability to correctly identify multiple labels. Notable performances include “Bicycle” and “Bus” with high true positive rates of 1307 and 1356, respectively. This model showed a high level of precision across most classes, although some, like “Motorbike” and “Table”, still presented room for improvement with lower true positive rates. **VGG19<sup>2</sup>**, then indicated a competent classification capability with a relatively even distribution of true positives across classes. However, there were instances of confusion, such as between “Dog” and “Motorbike”, where the true positive rates were not as high as expected. This suggests that while VGG19<sup>2</sup> was adept at multilabel classification, certain classes could benefit from further model refinement to reduce misclassification.



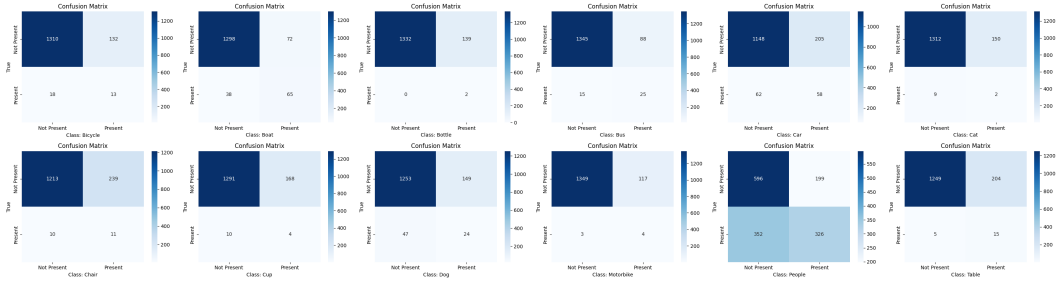


Figure 13: CNN Multilabel N-ary Confusion Matrix

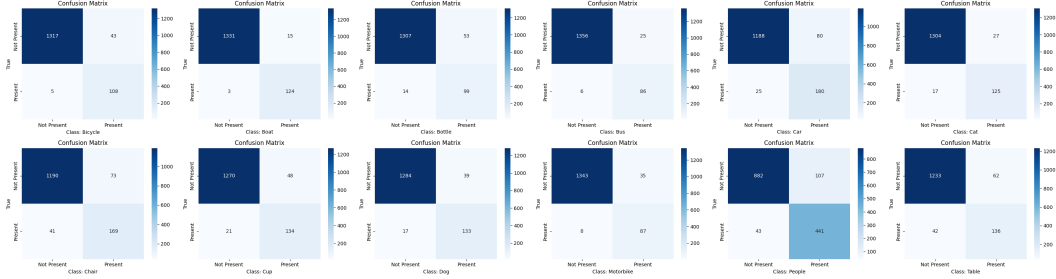


Figure 14: ResNet50 Multilabel N-ary Confusion Matrix

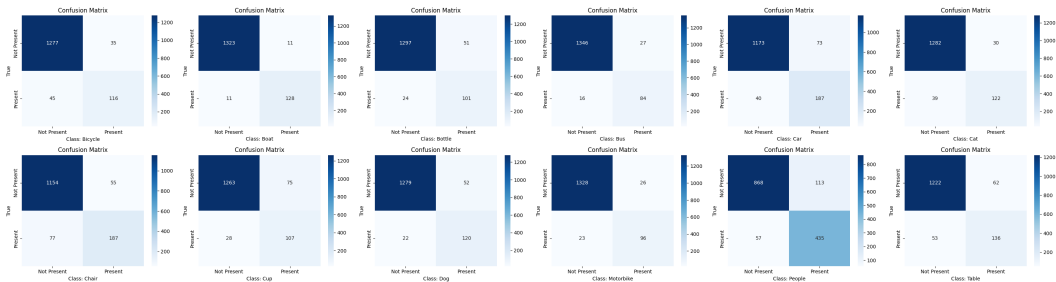


Figure 15: VGG19 Multilabel N-ary Confusion Matrix

**Overall Performance:** ResNet50<sup>3</sup> and VGG19<sup>2</sup> showcased their capabilities in the multilabel scenario, with ResNet50<sup>3</sup> slightly ahead in performance compared to VGG19<sup>2</sup>. The Custom CNN, in contrast, was less effective in this complex classification task.

**Experiences:** The multilabel classification phase of our research illuminated crucial challenges, notably the imbalance among different classes. We effectively addressed this using a WeightedRandomSampler, ensuring equitable representation in model training. However, a noticeable discrepancy emerged between high precision/recall for individual classes and lower overall accuracy. This underscored the complexity of the images and the necessity of sophisticated modeling techniques. The phase affirmed the need for advanced model architectures in multilabel tasks, emphasizing the significance of considering data distribution and characteristics for effective machine learning strategies.

#### 4.5 DISCUSSION

This study embarked on a comprehensive exploration of image classification across binary, ternary, and multilabel N-ary tasks, employing a custom CNN alongside established architectures such as ResNet50<sup>3</sup> and VGG19<sup>2</sup>. Notably, the models were benchmarked against a backdrop of rigorous hyperparameter tuning, which constituted over half of the research effort, underscoring its criticality in optimizing model performance.

##### 4.5.1 BINARY CLASSIFICATION

The binary classification task illuminated the distinct efficacies of different loss functions. Our custom CNN, though not as swift or accurate as its more sophisticated counterparts, nonetheless contributed valuable insights, particularly in the distinction between “Creature” and “Auto” classes. It was observed that the precision and recall metrics varied significantly between the classes, with a higher recall noted for “Creature”. This could be attributed to the innate model biases or the data distribution within the training set.

ResNet50<sup>3</sup> emerged as the front-runner, achieving a commendable balance between precision and recall across both classes. VGG19<sup>2</sup>’s performance was closely aligned with ResNet50<sup>3</sup>’s, albeit with a slight bias towards the “Creature” class. These results showcase the potential of advanced architectures in handling binary classifications with high accuracy, provided that the model is well-tuned.

##### 4.5.2 TERNARY CLASSIFICATION

Ternary classification expanded the challenge by introducing an additional class, thereby increasing the complexity of the task. The models were put to the test, with ResNet50<sup>3</sup> once again demonstrating its robustness, closely followed by VGG19<sup>2</sup>. The custom CNN struggled in this scenario, emphasizing the need for more complex model architectures when dealing with an increased number of classes. The WeightedRandomSampler played a pivotal role in mitigating the class imbalance, which was particularly pronounced in this classification scheme.

##### 4.5.3 MULTILABEL N-ARY CLASSIFICATION

Venturing into the realm of multilabel N-ary classification, the study highlighted the intricate nature of identifying multiple objects within a single image. The overall performance saw ResNet50<sup>3</sup> leading, with VGG19<sup>2</sup> close behind. The custom CNN faced challenges, reflecting the heightened complexity of the task at hand. Precision and recall metrics varied widely across classes, with some like “Bottle” achieving high precision but low recall, indicating a model reluctant to predict that class unless very certain.

The ResNet50<sup>3</sup> model demonstrated commendable performance in discerning “Boat” and “Dog” classes, while the custom CNN exhibited limitations in differentiating finer-grained categories.

#### 4.5.4 IMPLICATIONS AND FUTURE WORK

Our findings have significant implications for the development of image classification systems. It becomes evident that while custom models may offer a starting point, the sophistication offered by advanced pre-trained architectures like ResNet50<sup>3</sup> and VGG19<sup>2</sup> is indispensable for higher accuracy and balanced precision-recall metrics. Moreover, the use of weighted sampling to address class imbalance proved crucial, suggesting that future research should continue to focus on methods for equitable class representation.

In terms of future work, exploring additional data augmentation strategies, further hyperparameter optimization, and the integration of more complex models could yield improvements in classification tasks. Moreover, diving into the intricacies of dataset composition and investigating the potential of ensemble methods may provide pathways to enhance model robustness and accuracy.

## 5 CONCLUSION

Our study marks a significant advancement in the field of image classification under challenging conditions, demonstrating the effectiveness of our model pipeline. The research opens new avenues for innovative applications and integrations with other advanced machine-learning solutions.

Particularly noteworthy is our approach's impact on ternary classification, tailored for specific tasks. This success sets the stage for further exploration and application in fields requiring nuanced image interpretation, such as autonomous systems and advanced surveillance.

The transition from binary to multilabel N-ary classification not only deepened our understanding of the field's challenges but also established a foundation for future machine-learning explorations. Our model pipeline's adaptability and effectiveness highlight its potential for real-world applications, promising significant contributions to areas where precise classification is crucial.

In conclusion, this study paves the way for innovative applications and integrations in Deep Learning, reflecting our model pipeline's potential in areas demanding accurate and nuanced image interpretation.

## REFERENCES

- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding*, 178:1, 2019. doi: 10.1016/j.cviu.2018.10.010.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2015. doi: 10.48550/arXiv.1409.1556.