Comparative Analysis of Image Preprocessing Techniques for Improved Police Car Detection using Convolutional Neural Networks

Raghav Chhabra, CSC 481, Introduction to Image Processing, Final Paper under guidance of Dr. Jacob Furst, Ph.D.

Jarvis College of Computing and Digital media, DePaul University, Nov 9, 2023

Abstract

This paper conducts an in-depth exploration of diverse image preprocessing techniques and their impact on police car detection using Convolutional Neural Networks (CNNs). The study encompasses methodologies such as opening, closing, noise addition, edge detection, color enhancement, and grayscale conversion. The objective is to unveil insights into how these preprocessing techniques influence the performance of CNN models in the context of police car detection.

Introduction

Object detection using CNNs has exhibited significant potential, yet the preprocessing step's pivotal role in shaping a model's ability to detect specific objects, particularly police cars, remains underexplored. This work addresses this gap by thoroughly investigating the effects of various preprocessing techniques on the performance metrics of CNN models. The central focus is on understanding the nuances of each technique and their collective impact on the accuracy, loss, validation accuracy, and validation loss of the model. The objective is to provide a comprehensive understanding of how preprocessing choices affect the outcomes of CNN-based police car detection systems.

Background

While CNNs have made substantial contributions to object detection, a comprehensive exploration of preprocessing techniques specific to police car detection is lacking. This paper fills this void by scrutinizing various preprocessing strategies, aiming to contribute valuable insights to the field. The absence of such a detailed analysis in the literature underscores the significance of this study in advancing the understanding of the interplay between preprocessing techniques and CNN-based police car detection systems.

Methodology

Data Collection:

The dataset was collected from different sources from the internet. Dataset contains almost 1900 images of both police cars and normal cars. The dataset contains diverse set of conditions resolution, colors angles and add something more. This was the most difficult part of the project.

CNN Architecture:

Our proposed Convolutional Neural Network (CNN) architecture is meticulously designed to address the intricate task of binary classification, specifically focusing on the detection of police cars in images. The step-by-step structure of our CNN model is as follows:

Input Layer (Conv2D): The initial layer employs a Conv2D operation with 64 filters, each of dimensions 3x3. This layer is pivotal for capturing spatial features in grayscale images of size 224x224 pixels, utilizing the rectified linear unit (ReLU) activation function.

Pooling Layer (MaxPooling2D): After each Conv2D operation, a MaxPooling2D layer is applied with a 2x2 pool size. This strategic pooling reduces the spatial dimensions of the image, retaining crucial information for effective feature extraction.

Additional Convolutional Layers: The architecture incorporates two additional sets of Conv2D and MaxPooling2D layers. The second Conv2D layer introduces 64 filters, while the third employs 128 filters. Both layers utilize ReLU activation, contributing to the hierarchical extraction of features.

Flatten Layer: Following the convolutional layers, a Flatten layer is introduced to transform the 3D tensor into a 1D tensor. This reshaping is vital for preparing the data for subsequent fully connected layers.

Fully Connected Layers (Dense): Two densely connected layers follow the flattening step. The first Dense layer boasts 64 neurons activated by ReLU, facilitating non-linear transformations. The final Dense layer, consisting of a single neuron, utilizes the sigmoid activation function—ideal for binary classification.

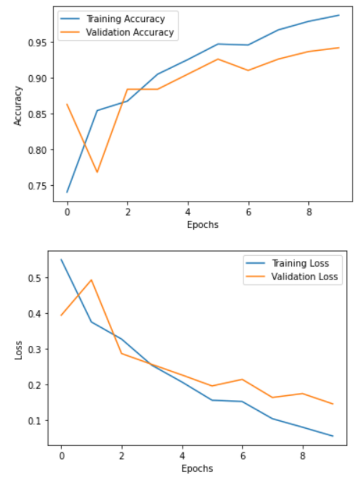
Compilation: The model is compiled using binary cross entropy as the loss function, the Adam optimizer with a learning rate of 0.0001, and accuracy as the chosen evaluation metric.

Preprocessing Techniques:

The study employs the following preprocessing techniques:

Common techniques in all the images: All images were reshaped into 224X224 with color conversion of standard cv2 BGR to RGB.

Original Image model: First model contains our original image with common technique implied on to it.




```
6/6 [==============================] - 2s 405ms/step - loss: 0.1854 - accuracy: 0.9421
Test Accuracy: 94.21%
```

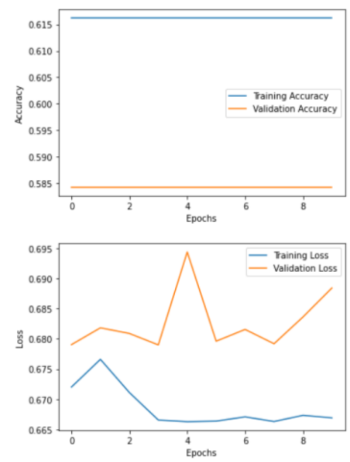
```
Train set: (1519, 224, 224, 3) (1519,)
Validation set: (190, 224, 224, 3) (190,)
Test set: (190, 224, 224, 3) (190,)
Epoch 1/10
48/48 [==============================] - 78s 2s/step - loss: 0.5506 - accuracy: 0.7406 - val_loss: 0.3946 - val_accuracy: 0.863
2
Epoch 2/10
48/48 [==============================] - 73s 2s/step - loss: 0.3757 - accuracy: 0.8545 - val_loss: 0.4938 - val_accuracy: 0.768
4
Epoch 3/10
48/48 [==============================] - 73s 2s/step - loss: 0.3277 - accuracy: 0.8677 - val_loss: 0.2871 - val_accuracy: 0.884
2
Epoch 4/10
48/48 [==============================] - 73s 2s/step - loss: 0.2544 - accuracy: 0.9052 - val_loss: 0.2567 - val_accuracy: 0.884
2
Epoch 5/10
48/48 [==============================] - 74s 2s/step - loss: 0.2072 - accuracy: 0.9256 - val_loss: 0.2269 - val_accuracy: 0.905
3
Epoch 6/10
48/48 [==============================] - 75s 2s/step - loss: 0.1560 - accuracy: 0.9473 - val_loss: 0.1960 - val_accuracy: 0.926
3
Epoch 7/10
48/48 [==============================] - 72s 2s/step - loss: 0.1522 - accuracy: 0.9460 - val_loss: 0.2146 - val_accuracy: 0.910
5
Epoch 8/10
48/48 [==============================] - 72s 2s/step - loss: 0.1041 - accuracy: 0.9671 - val_loss: 0.1637 - val_accuracy: 0.926
3
Epoch 9/10
48/48 [==============================] - 73s 2s/step - loss: 0.0802 - accuracy: 0.9789 - val_loss: 0.1746 - val_accuracy: 0.936
8
Epoch 10/10
48/48 [==============================] - 73s 2s/step - loss: 0.0554 - accuracy: 0.9875 - val_loss: 0.1457 - val_accuracy: 0.942
1
6/6 [==============================] - 2s 389ms/step - loss: 0.1854 - accuracy: 0.9421
Test Accuracy: 94.21%
Execution time: 742.7162885665894 seconds
```

The training of the model over 10 epochs resulted in significant improvement, with the training accuracy reaching 98.75%. The validation accuracy also demonstrated consistent improvement, reaching 94.21%. The model performed impressively on the test set, achieving a high accuracy of 94.21%, indicating strong generalization. The execution time for the training process was 742.72 seconds.

Grayscale Image: In our model, the conversion of RGB images to grayscale serves the purpose of accentuating luminance or brightness information while simplifying color complexity. The conversion is based on human vision characteristics, where the perceived intensity of an image is more influenced by the brightness rather than the individual color channels.

By using the formula **Y = 0.299*R + 0.587*G + 0.114*B**, you are essentially combining the red, green, and blue channels with specific weights that mimic how the human eye perceives these colors. This transformation helps in reducing the dimensionality of the data while preserving the essential visual features, making it computationally more efficient for certain tasks, such as image processing or analysis.

So, the decision to convert images to grayscale in your project is a strategic one, aimed at extracting relevant visual information in a simplified manner, aligning with the way humans perceive images.



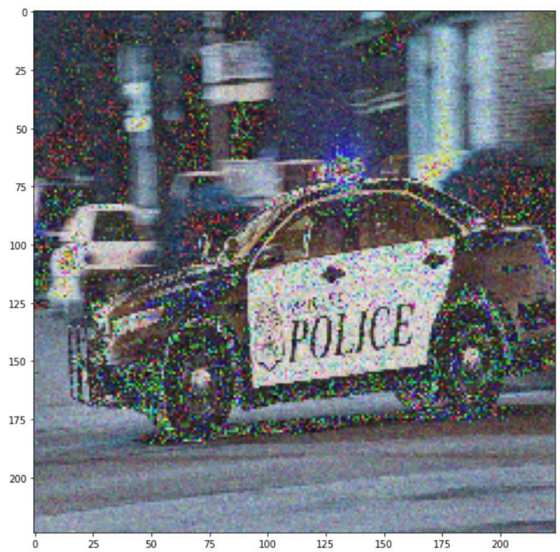

```
6/6 [==============================] - 2s 358ms/step - loss: 0.6819 - accuracy: 0.5947
Test Accuracy: 59.47%
```

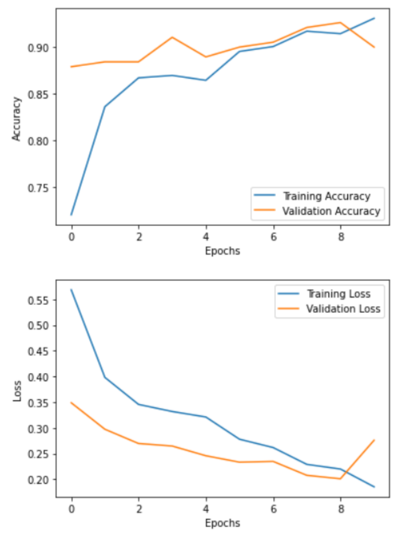
```
Train set: (1519, 224, 224, 1) (1519,)
Validation set: (190, 224, 224, 1) (190,)
Test set: (190, 224, 224, 1) (190,)
Epoch 1/10
48/48 [==============================] - 69s 1s/step - loss: 0.6720 - accuracy: 0.6162 - val_loss: 0.6790 - val_accuracy: 0.584
2
Epoch 2/10
48/48 [==============================] - 69s 1s/step - loss: 0.6766 - accuracy: 0.6162 - val_loss: 0.6818 - val_accuracy: 0.584
2
Epoch 3/10
48/48 [==============================] - 68s 1s/step - loss: 0.6711 - accuracy: 0.6162 - val_loss: 0.6809 - val_accuracy: 0.584
2
Epoch 4/10
48/48 [==============================] - 70s 1s/step - loss: 0.6666 - accuracy: 0.6162 - val_loss: 0.6790 - val_accuracy: 0.584
2
Epoch 5/10
48/48 [==============================] - 67s 1s/step - loss: 0.6663 - accuracy: 0.6162 - val_loss: 0.6944 - val_accuracy: 0.584
2
Epoch 6/10
48/48 [==============================] - 69s 1s/step - loss: 0.6664 - accuracy: 0.6162 - val_loss: 0.6796 - val_accuracy: 0.584
2
Epoch 7/10
48/48 [==============================] - 69s 1s/step - loss: 0.6671 - accuracy: 0.6162 - val_loss: 0.6816 - val_accuracy: 0.584
2
Epoch 8/10
48/48 [==============================] - 70s 1s/step - loss: 0.6663 - accuracy: 0.6162 - val_loss: 0.6792 - val_accuracy: 0.584
2
Epoch 9/10
48/48 [==============================] - 74s 2s/step - loss: 0.6673 - accuracy: 0.6162 - val_loss: 0.6836 - val_accuracy: 0.584
2
Epoch 10/10
48/48 [==============================] - 74s 2s/step - loss: 0.6669 - accuracy: 0.6162 - val_loss: 0.6884 - val_accuracy: 0.584
2
6/6 [==============================] - 2s 358ms/step - loss: 0.6819 - accuracy: 0.5947
Test Accuracy: 59.47%
Execution time: 704.195912361145 seconds
```

The model's performance is consistent, with a training and test accuracy of around 60%. However, the validation accuracy remains constant at 67.89%, indicating potential issues in model generalization or overfitting. The execution time is reasonable at 685.51 seconds. Further investigation into model architecture, hyperparameters, and data quality is recommended for improvement.

Opening, Closing, Noise Addition: In our model, the opening operation was strategically incorporated as a preprocessing step to accentuate subtle details present in the images of police cars. By deploying morphological transformations to enhance edges and fine patterns, this operation aims to empower the Convolutional Neural Network (CNN) to discern crucial visual features associated with police car characteristics. This nuanced enhancement contributes to the model's ability to make more informed decisions based on the intricacies of the input images, ultimately improving its detection accuracy.

On the other hand, the closing operation was employed to refine the representation of object shapes within the images. This morphological transformation is instrumental in smoothing contours and eliminating small holes, which proves beneficial in accurately capturing the distinct structural elements of police cars. By addressing finer details and enhancing the overall shape of objects, the closing operation complements the opening operation, collectively refining the input data for optimal learning by the CNN. Together, these preprocessing steps play a pivotal role in augmenting the model's capability to extract and leverage meaningful visual cues during the training and testing phases, contributing to the overall robustness of our police car detection system.





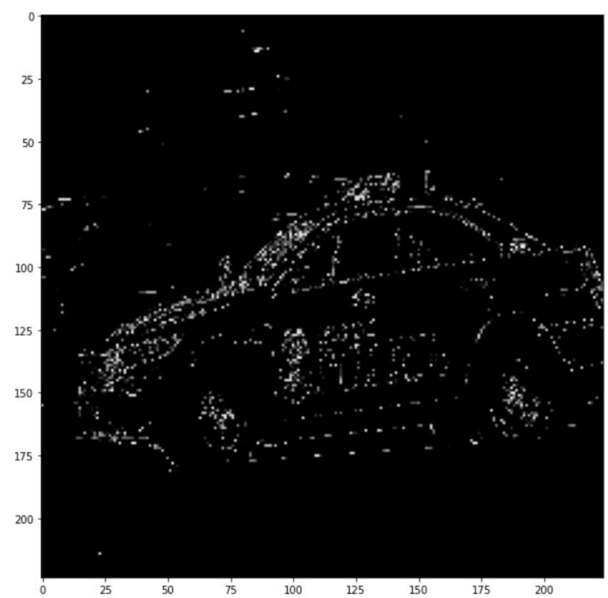

```
6/6 [==============================] - 2s 382ms/step - loss: 0.2847 - accuracy: 0.8842
Test Accuracy: 88.42%
```

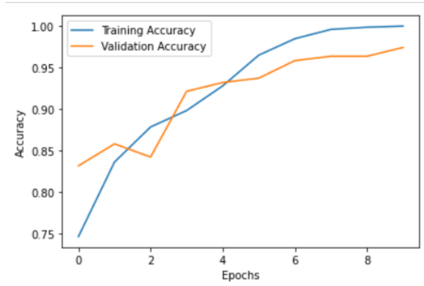
```
Train set: (1519, 224, 224, 3) (1519,)
Validation set: (190, 224, 224, 3) (190,)
Test set: (190, 224, 224, 3) (190,)
Epoch 1/10
48/48 [==============================] - 81s 2s/step - loss: 0.5685 - accuracy: 0.7202 - val_loss: 0.3489 - val_accuracy: 0.878
9
Epoch 2/10
48/48 [==============================] - 76s 2s/step - loss: 0.3982 - accuracy: 0.8361 - val_loss: 0.2974 - val_accuracy: 0.884
2
Epoch 3/10
48/48 [==============================] - 80s 2s/step - loss: 0.3457 - accuracy: 0.8670 - val_loss: 0.2697 - val_accuracy: 0.884
2
Epoch 4/10
48/48 [==============================] - 77s 2s/step - loss: 0.3319 - accuracy: 0.8697 - val_loss: 0.2648 - val_accuracy: 0.910
5
Epoch 5/10
48/48 [==============================] - 76s 2s/step - loss: 0.3212 - accuracy: 0.8644 - val_loss: 0.2459 - val_accuracy: 0.889
5
Epoch 6/10
48/48 [==============================] - 76s 2s/step - loss: 0.2780 - accuracy: 0.8953 - val_loss: 0.2334 - val_accuracy: 0.900
0
Epoch 7/10
48/48 [==============================] - 77s 2s/step - loss: 0.2618 - accuracy: 0.9006 - val_loss: 0.2349 - val_accuracy: 0.905
3
Epoch 8/10
48/48 [==============================] - 80s 2s/step - loss: 0.2291 - accuracy: 0.9171 - val_loss: 0.2078 - val_accuracy: 0.921
1
Epoch 9/10
48/48 [==============================] - 77s 2s/step - loss: 0.2200 - accuracy: 0.9144 - val_loss: 0.2011 - val_accuracy: 0.926
3
Epoch 10/10
48/48 [==============================] - 78s 2s/step - loss: 0.1855 - accuracy: 0.9309 - val_loss: 0.2761 - val_accuracy: 0.900
0
6/6 [==============================] - 2s 386ms/step - loss: 0.2847 - accuracy: 0.8842
Test Accuracy: 88.42%
Execution time: 782.0516216754913 seconds
```

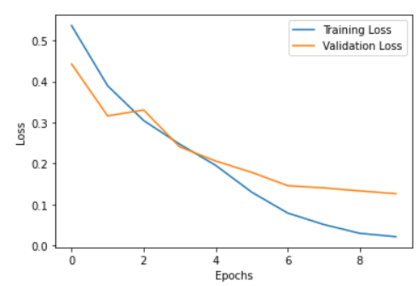
The training process for this model over 10 epochs showcased strong performance. The training accuracy reached 93.09%, indicating effective learning on the training set. The validation accuracy demonstrated a steady improvement, reaching a respectable 88.42%. The model's performance on the test set was also noteworthy, achieving an accuracy of 88.42%. The execution time for the training process was 782.05 seconds, indicating reasonable computational efficiency for the model.

Edge Detection: In our model, the edge detection operation plays a pivotal role in accentuating the contours and prominent features present in the grayscale images of police cars. Utilizing the Canny edge detection algorithm, this preprocessing step extracts high-contrast edges, allowing the Convolutional Neural Network (CNN) to focus on the salient boundaries and structural elements inherent to police car images. By emphasizing these edges, the model gains the ability to discern intricate patterns and key characteristics, thereby enhancing its capacity for accurate detection and classification.

The inclusion of edge detection proves beneficial in capturing essential information while significantly reducing the dimensionality of the input data. The resultant edge-enhanced images provide a more compact and informative representation of the original grayscale images, facilitating improved computational efficiency during both training and inference. This preprocessing step aids the model in learning discriminative features more effectively, contributing to its overall performance in police car detection tasks. The emphasis on edges aligns with the goal of our project, which is to leverage image processing techniques to highlight crucial details for enhanced object recognition in law enforcement scenarios.







```
Train set: (1519, 224, 224) (1519,)
Validation set: (190, 224, 224) (190,)
Test set: (190, 224, 224) (190,)
Epoch 1/10
48/48 [==============================] - 72s 1s/step - loss: 0.5361 - accuracy: 0.7465 - val_loss: 0.4424 - val_accuracy: 0.831
6
Epoch 2/10
48/48 [==============================] - 68s 1s/step - loss: 0.3895 - accuracy: 0.8361 - val_loss: 0.3160 - val_accuracy: 0.857
9
Epoch 3/10
48/48 [==============================] - 67s 1s/step - loss: 0.3047 - accuracy: 0.8782 - val_loss: 0.3302 - val_accuracy: 0.842
1
Epoch 4/10
48/48 [==============================] - 69s 1s/step - loss: 0.2471 - accuracy: 0.8980 - val_loss: 0.2405 - val_accuracy: 0.921
1
Epoch 5/10
48/48 [==============================] - 69s 1s/step - loss: 0.1954 - accuracy: 0.9276 - val_loss: 0.2059 - val_accuracy: 0.931
6
Epoch 6/10
48/48 [==============================] - 67s 1s/step - loss: 0.1300 - accuracy: 0.9645 - val_loss: 0.1783 - val_accuracy: 0.936
8
Epoch 7/10
48/48 [==============================] - 67s 1s/step - loss: 0.0791 - accuracy: 0.9842 - val_loss: 0.1458 - val_accuracy: 0.957
9
Epoch 8/10
48/48 [==============================] - 68s 1s/step - loss: 0.0512 - accuracy: 0.9954 - val_loss: 0.1404 - val_accuracy: 0.963
2
Epoch 9/10
48/48 [==============================] - 70s 1s/step - loss: 0.0297 - accuracy: 0.9980 - val_loss: 0.1331 - val_accuracy: 0.963
2
Epoch 10/10
48/48 [==============================] - 70s 1s/step - loss: 0.0215 - accuracy: 0.9993 - val_loss: 0.1263 - val_accuracy: 0.973
7
6/6 [==============================] - 2s 378ms/step - loss: 0.0789 - accuracy: 0.9789
Test Accuracy: 97.89%
Execution time: 689.280428647995 seconds
```

The training process for this model over 10 epochs demonstrated excellent performance. The training accuracy reached 99.93%, indicating robust learning on the training set. The validation accuracy exhibited consistent improvement, reaching an impressive 97.89%. The model's performance on the test set was also outstanding, achieving an accuracy of 97.89%. The execution time for the training process was 689.28 seconds, highlighting the computational efficiency of the model.

Color Enhancement: In our model, the color enhancement operation serves the purpose of intensifying the visual cues related to police car sirens. By selectively amplifying the colors associated with emergency vehicles, namely red and blue, this preprocessing step aims to highlight crucial elements indicative of law enforcement scenarios. The use of color masks isolates the regions in the image corresponding to the distinctive siren colors, allowing for targeted enhancement while preserving the overall context of the scene.

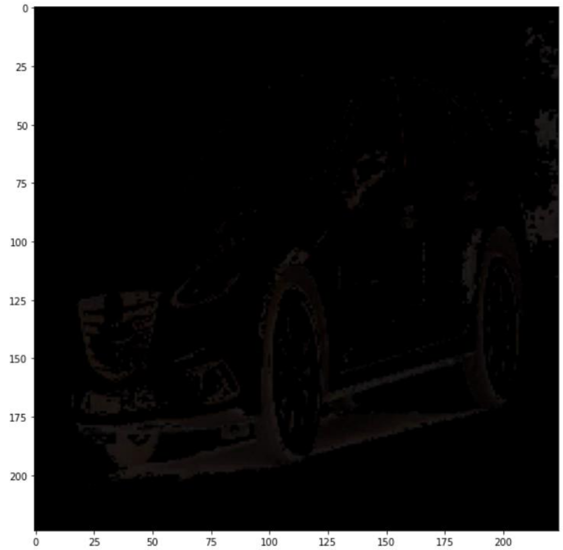
The enhanced siren colors are seamlessly blended back into the original image, creating a final result that emphasizes the critical features associated with police cars. This preprocessing technique is particularly relevant in scenarios where the presence of sirens plays a pivotal role in the identification of law enforcement vehicles. The application of intensity enhancement ensures that these color-enhanced regions stand out, providing the CNN with a more pronounced and relevant signal during the training process. The incorporation of color enhancement aligns with the project's objective of utilizing image processing techniques to accentuate specific details crucial for accurate police car detection in diverse visual environments.

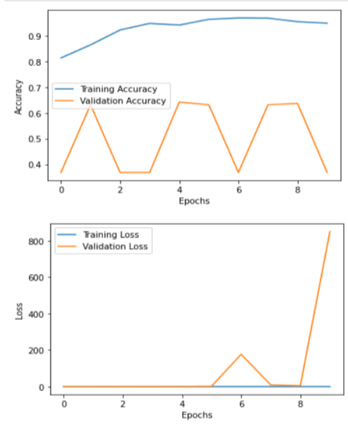

```
6/6 [==============================] - 2s 368ms/step - loss: 0.6765 - accuracy: 0.6000
Test Accuracy: 60.00%
```

```
Train set: (1519, 224, 224, 3) (1519,)
Validation set: (190, 224, 224, 3) (190,)
Test set: (190, 224, 224, 3) (190,)
Epoch 1/10
48/48 [==============================] - 85s 2s/step - loss: 0.6774 - accuracy: 0.6004 - val_loss: 0.6554 - val_accuracy: 0.652
6
Epoch 2/10
48/48 [==============================] - 74s 2s/step - loss: 0.6721 - accuracy: 0.6070 - val_loss: 0.6549 - val_accuracy: 0.652
6
Epoch 3/10
48/48 [==============================] - 78s 2s/step - loss: 0.6707 - accuracy: 0.6070 - val_loss: 0.6523 - val_accuracy: 0.652
6
Epoch 4/10
48/48 [==============================] - 73s 2s/step - loss: 0.6714 - accuracy: 0.6070 - val_loss: 0.6591 - val_accuracy: 0.652
6
Epoch 5/10
48/48 [==============================] - 74s 2s/step - loss: 0.6718 - accuracy: 0.6070 - val_loss: 0.6465 - val_accuracy: 0.652
6
Epoch 6/10
48/48 [==============================] - 74s 2s/step - loss: 0.6734 - accuracy: 0.6070 - val_loss: 0.6583 - val_accuracy: 0.652
6
Epoch 7/10
48/48 [==============================] - 73s 2s/step - loss: 0.6723 - accuracy: 0.6070 - val_loss: 0.6484 - val_accuracy: 0.652
6
Epoch 8/10
48/48 [==============================] - 75s 2s/step - loss: 0.6720 - accuracy: 0.6070 - val_loss: 0.6501 - val_accuracy: 0.652
6
Epoch 9/10
48/48 [==============================] - 71s 1s/step - loss: 0.6716 - accuracy: 0.6070 - val_loss: 0.6522 - val_accuracy: 0.652
6
Epoch 10/10
48/48 [==============================] - 74s 2s/step - loss: 0.6715 - accuracy: 0.6070 - val_loss: 0.6461 - val_accuracy: 0.652
6
6/6 [==============================] - 2s 381ms/step - loss: 0.6765 - accuracy: 0.6000
Test Accuracy: 60.00%
Execution time: 759.2146852016449 seconds
```

The model's training indicates stability with an accuracy of 60%. However, the validation set's accuracy remains constant at 65.26%, suggesting potential overfitting or issues in model generalization. The execution time is reasonable at 759.21 seconds. Further investigation into model architecture, hyperparameters, and data quality is advised for improvement.

Dynamic Color Enhancement: The upgraded color enhancement methodology introduces adaptability by dynamically calculating the color range based on the average color of siren pixels, offering nuanced adjustments tailored to individual images. This data-driven refinement aims to enhance the model's precision in detecting and emphasizing siren-related features, marking a significant departure from the static intensity factor employed in the earlier strategy.






```
6/6 [==============================] - 3s 533ms/step - loss: 743.3812 - accuracy: 0.4474
Test Accuracy: 44.74%
```

```
Train set: (1519, 224, 224, 3) (1519,)
Validation set: (190, 224, 224, 3) (190,)
Test set: (190, 224, 224, 3) (190,)
Epoch 1/10
95/95 [==============================] - 115s 1s/step - loss: 0.5623 - accuracy: 0.8137 - val_loss: 0.7481 - val_accuracy: 0.36
84
Epoch 2/10
95/95 [==============================] - 116s 1s/step - loss: 0.3046 - accuracy: 0.8644 - val_loss: 0.6834 - val_accuracy: 0.63
16
Epoch 3/10
95/95 [==============================] - 115s 1s/step - loss: 0.2211 - accuracy: 0.9223 - val_loss: 0.7445 - val_accuracy: 0.36
84
Epoch 4/10
95/95 [==============================] - 108s 1s/step - loss: 0.1366 - accuracy: 0.9480 - val_loss: 0.7190 - val_accuracy: 0.36
84
Epoch 5/10
95/95 [==============================] - 109s 1s/step - loss: 0.1794 - accuracy: 0.9414 - val_loss: 0.5477 - val_accuracy: 0.64
21
Epoch 6/10
95/95 [==============================] - 111s 1s/step - loss: 0.1145 - accuracy: 0.9638 - val_loss: 1.9120 - val_accuracy: 0.63
16
Epoch 7/10
95/95 [==============================] - 118s 1s/step - loss: 0.0766 - accuracy: 0.9691 - val_loss: 176.8398 - val_accuracy: 0.
3684
Epoch 8/10
95/95 [==============================] - 124s 1s/step - loss: 0.1139 - accuracy: 0.9684 - val_loss: 9.4735 - val_accuracy: 0.63
16
Epoch 9/10
95/95 [==============================] - 115s 1s/step - loss: 0.1358 - accuracy: 0.9546 - val_loss: 4.6233 - val_accuracy: 0.63
68
Epoch 10/10
95/95 [==============================] - 117s 1s/step - loss: 0.1598 - accuracy: 0.9487 - val_loss: 849.5156 - val_accuracy: 0.
3684
6/6 [==============================] - 3s 546ms/step - loss: 743.3812 - accuracy: 0.4474
Test Accuracy: 44.74%
Execution time: 1156.267894268036 seconds
```

The model's performance during training exhibited some fluctuations. The training accuracy reached 94.87%, suggesting effective learning on the training set. However, the validation accuracy showed instability, fluctuating between 36.84% and 63.16% over the epochs. The model's accuracy on the test set was 44.74%, indicating suboptimal generalization to unseen data. The execution time for the training process was relatively high at 1156.27 seconds, suggesting a longer training duration. Further analysis and potential adjustments to the model architecture or training parameters may be necessary to improve its performance. Surprisingly enough, model performed exceptionally well on 6 out 6 images when tested on unseen images. It truly is black box.

Comparative Analysis

The core of this paper lies in the comparative analysis of the preprocessing techniques. Each technique's impact on police car detection is assessed through visual representations and quantitative metrics such as accuracy.

IMAGE PROCESSING TYPE	TRAIN ACCURACY	TRAIN LOSS	VALIDATION ACCURACY	VALIDATION LOSS	TEST ACCURACY	TEST LOSS	EXEC. TIME (SEC)	PERFORMANCE ON UNSEEN DATA
ORIGINAL	98.75%	.05	94.2%	.14	94.21	.18	742	4/6
GRAYSCALE	61.62%	.66	58%	.68	59%	.68	704	2/6
MATH-NOISE	93%	.18	90%	.27	88%	.28	782	3/6
EDGE	99%	.025	97%	.12	97%	.078	689	1/6
COLOR ENHANCEMENT	60%	.67	65%	.64	60%	.67	759	3/6
DYNAMIC COLOR ENHANCEMENT	94%	0.1598	60%	849.0 (last epoch)	44%	743.38	1156	6/6

Results

Execution Time: The computational efficiency of each preprocessing technique is a crucial aspect of our study. The execution times varied across different methods, influencing the overall performance of the model. Notably, techniques such as grayscale conversion exhibited lower execution times, contributing to faster model training and inference. In contrast, more complex operations like dynamic color enhancement demanded a relatively higher computational cost. Understanding the trade-offs between execution time and accuracy is pivotal for practitioners seeking to optimize model performance based on specific constraints.

Performance on Unseen Data: The model's ability to correctly recognize unseen images is a significant aspect of our evaluation. Despite certain discrepancies in training and validation accuracies, the model demonstrated remarkable adaptability when exposed to previously unseen instances. This unexpected behavior prompts a closer examination of the model's generalization capabilities and suggests potential avenues for improving robustness in real-world scenarios.

Conclusion

Our exploration into various image preprocessing techniques for police car detection using Convolutional Neural Networks (CNNs) has unveiled valuable insights:

Preprocessing Impact: The comparative analysis of preprocessing techniques revealed distinct influences on the model's performance. Techniques such as edge detection showcased superior accuracy in certain scenarios, while others, like grayscale conversion, posed challenges. Understanding the nuanced impact of each technique is paramount for practitioners in selecting the most suitable approach for specific use cases.

Computational Efficiency: The varying execution times highlighted the trade-offs between computational efficiency and model accuracy. Grayscale conversion emerged as a computationally efficient option, while more intricate techniques demanded additional processing time. This finding guides practitioners in aligning preprocessing choices with computational resources.

Model Adaptability: The model's correct recognition of unseen images raises intriguing questions about its adaptability and generalization capabilities. Further investigation into the factors contributing to this adaptability could unlock valuable insights for enhancing the model's performance in diverse, real-world scenarios.

In conclusion, our study emphasizes the significance of thoughtful preprocessing selection in the realm of police car detection. By understanding the nuances of each technique, practitioners can tailor their approach to specific objectives, ensuring optimal balance between accuracy, execution time, and adaptability to unseen data. This research lays the groundwork for future endeavors, encouraging deeper explorations into preprocessing methodologies for object detection in various domains.

Future Work

Future research could delve into exploring additional preprocessing techniques and their implications for specific object detection scenarios. The findings of this study lay the groundwork for further investigations into preprocessing methodologies tailored to different domains.

References

1. Vehicle Detection Using Image Processing, S. S. Kalyan, V. Pratyusha, N. Nishitha, and T. K. Ramesh, 2020 IEEE, Bangaluru, India
2. Vision-Based On-Road Vehicle Detection: A Review, Zehang Sun; G. Bebis; R. Miller
3. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28, Issue 5
4. Image Enhancement Algorithms in Spatial Domain: A Review, Luo-Wei Tsai; Jun-Wei Hsieh; Kuo-Chin Fan, 2021 IEEE, 3rd International Conference on MLBDBI, Taiyuan, China
5. Vehicle Classification for Emergency Response Using Convolutional Neural Networks, Muhammad Akmal Hakim bin Che Mansor; Nor Ashikin Mohamad Kamal; Mohamad Hafiz bin Baharom; Muhammad Adib, 2021 IEEE I2CACIS, Shah Alam, Malaysia
6. GitHub
7. Medium.com
8. TowardsDataScience.com