Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Amer Jusuf

INSIGHT INTO BLACK-BOX TESTING OF DEEP NEURAL NETWORKS

Consultant
Marussy Kristóf

BUDAPEST, 2024

1.Introduction

In the development of autonomous vehicles and other critical systems, the use of deep neural networks is becoming increasingly prevalent, particularly in the field of image recognition. Designing these systems requires not only precise engineering work and reliable architecture but also thorough testing of the system. Testing deep neural networks, which are neural networks with an increased number of hidden layers, poses a challenge for engineers due to the complexity of deep neural networks. During the process, if it is possible to organize misclassified images systematically, there is an opportunity to fine-tune the neural network. This fine-tuning can be done with images that are similar to the misclassified ones; for example, if a deep neural network incorrectly recognizes "stop" signs with a high percentage, it may be worthwhile to fine-tune it with images containing "stop" signs. As a result, training the neural network becomes more efficient, and the accuracy of the system can increase. However, fine-tuning the neural network with the misclassified images does not guarantee the increase of the accuracy.

Additionally, analyzing such errors can help identify which system components are sensitive to faulty operation. Additional security measures can be introduced around these components, contributing to the improvement of the performance and reliability of the critical system.

Furthermore, I discuss the theoretical and practical approaches of the described testing method, based on the publication titled “*Black-Box Testing of Deep Neural Networks through Test Case Diversity*” [1].

2.Measurement

2.1. Architecture

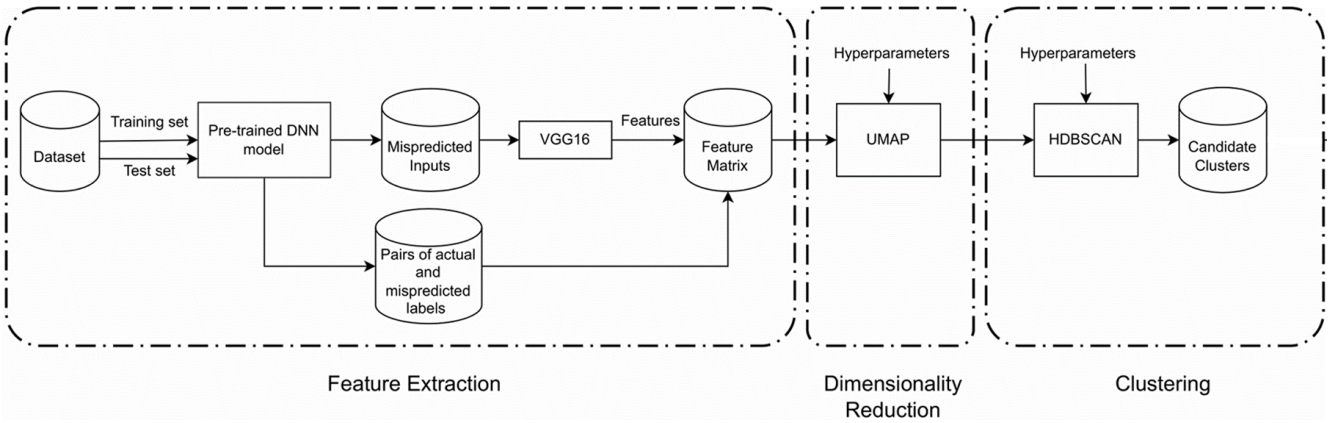


Figure 2.1 Clustering mispredictions
(Modified figure from 'Black-Box Testing of Deep Neural Networks through Test Case Diversity' [1])

Figure 2.1 presents a suitable architecture for clustering images misdetected by the deep neural network (DNN).

2.1.1 Dataset

The dataset represents the data that is evaluated during the testing process of the deep neural network. This article does not discuss requirements for the dataset, such as diversity of the images or the size of the dataset.

2.1.2 Pre-trained DNN model

The pre-trained DNN model represents the DNN that is being tested. The images from the dataset are processed by the model.

2.1.3 Mispredicted Inputs & Pairs of actual and mispredicted labels

Each misclassified image, and its ground truth label, and the misclassification label is stored for later evaluation.

2.1.4 VGG16

VGG16 is a pre-trained deep neural network specifically designed for image recognition tasks, utilizing convolutional neural network (CNN) architecture. It has been trained on a dataset of 14 million images.

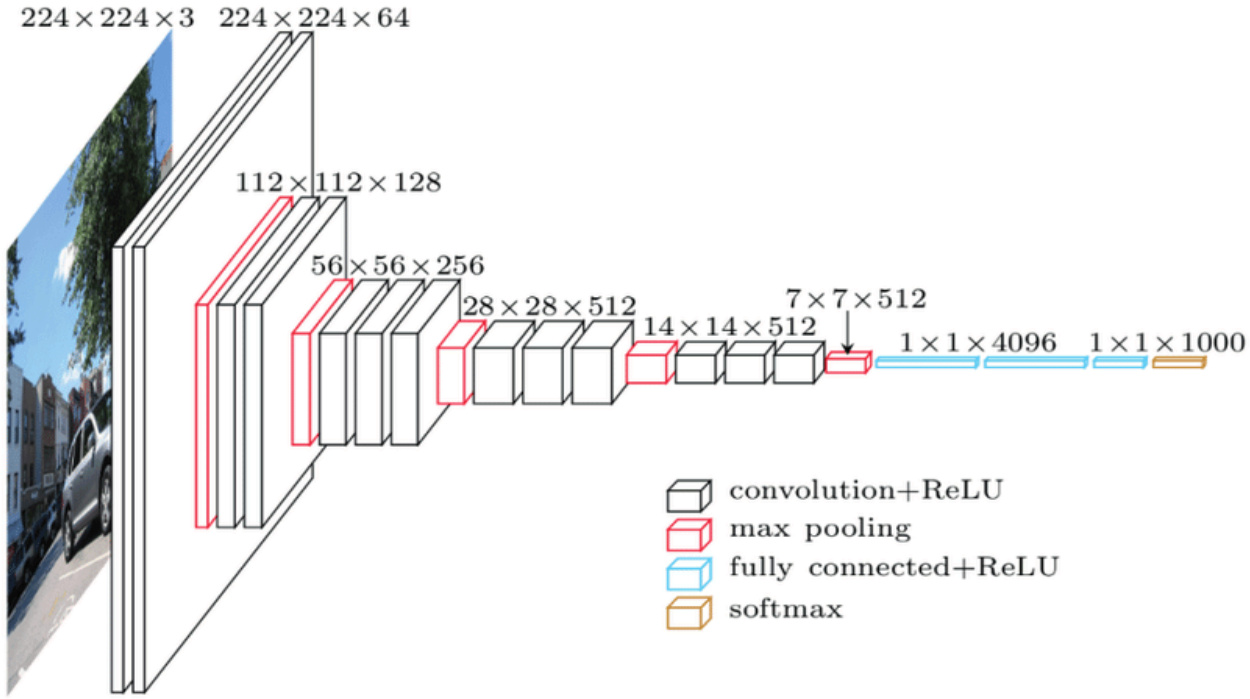


Figure 2.1.4 VGG16 architecture [2]

The VGG16 architecture is characterized by its simplicity and uniform structure. It comprises a total of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers use small 3x3 filters with a stride of 1, and max-pooling is performed using 2x2 filters. The use of small filters and stacking multiple convolutional layers on top of each other helps the network learn hierarchical features that increase in complexity gradually. The input layer accepts RGB images of size 224x224 pixels. In the feature grouping architecture, the "fully connected" top layer responsible for image classification is detached, so the output of the VGG16 with its top layer detached becomes a 1x1x512-dimensional feature vector.

In this measurement each misclassified image is evaluated by VGG16 to a 512 dimensional feature vector.

2.1.5 Feature Matrix

After extracting the feature vector from an image, their labels (ground truth label, misprediction label) are appended to the feature vector, thus in the Feature matrix the stored vectors are 514 dimensional vectors (512 dimensional feature vectors appended with 2 dimensional vectors of the labels).

2.1.6 UMAP

UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique primarily used for visualizing high-dimensional data in lower dimensions, typically two or three dimensions. It's similar in purpose to techniques like t-SNE (t-distributed stochastic neighbor embedding) and PCA (Principal Component Analysis), but it often preserves more of the global structure of the data.

In this measurement the dimension of vectors (514 dim.) in the Feature matrix are reduced to lower dimensions with the UMAP algorithm.

2.1.7 HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm designed to find clusters of varying densities in high-dimensional data. It builds upon the density-based clustering approach of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) but extends it to work with hierarchical clustering.

HDBSCAN can be parameterized, some of the key parameters are *Minimum cluster size* and *Min. samples.*

After using HDBSCAN algorithm on the reduced dimensional Feature vectors, the algorithm provides clusters, these clusters are the candidate clusters.

2.3. Measurements

2.3.1 Cifar10 dataset with self-made CNN

2.3.1.1 CIFAR10 Dataset

Cifar10 dataset[3] contains 60,000 images divided into 10 different classes. Each class consists of 6,000 images, and the dataset is split into training and test sets. The training set comprises 50,000 images, while the test set contains the remaining 10,000 images. The size of each image is 32x32 pixels.

The CIFAR-10 dataset is commonly used for evaluating and comparing machine learning algorithms.

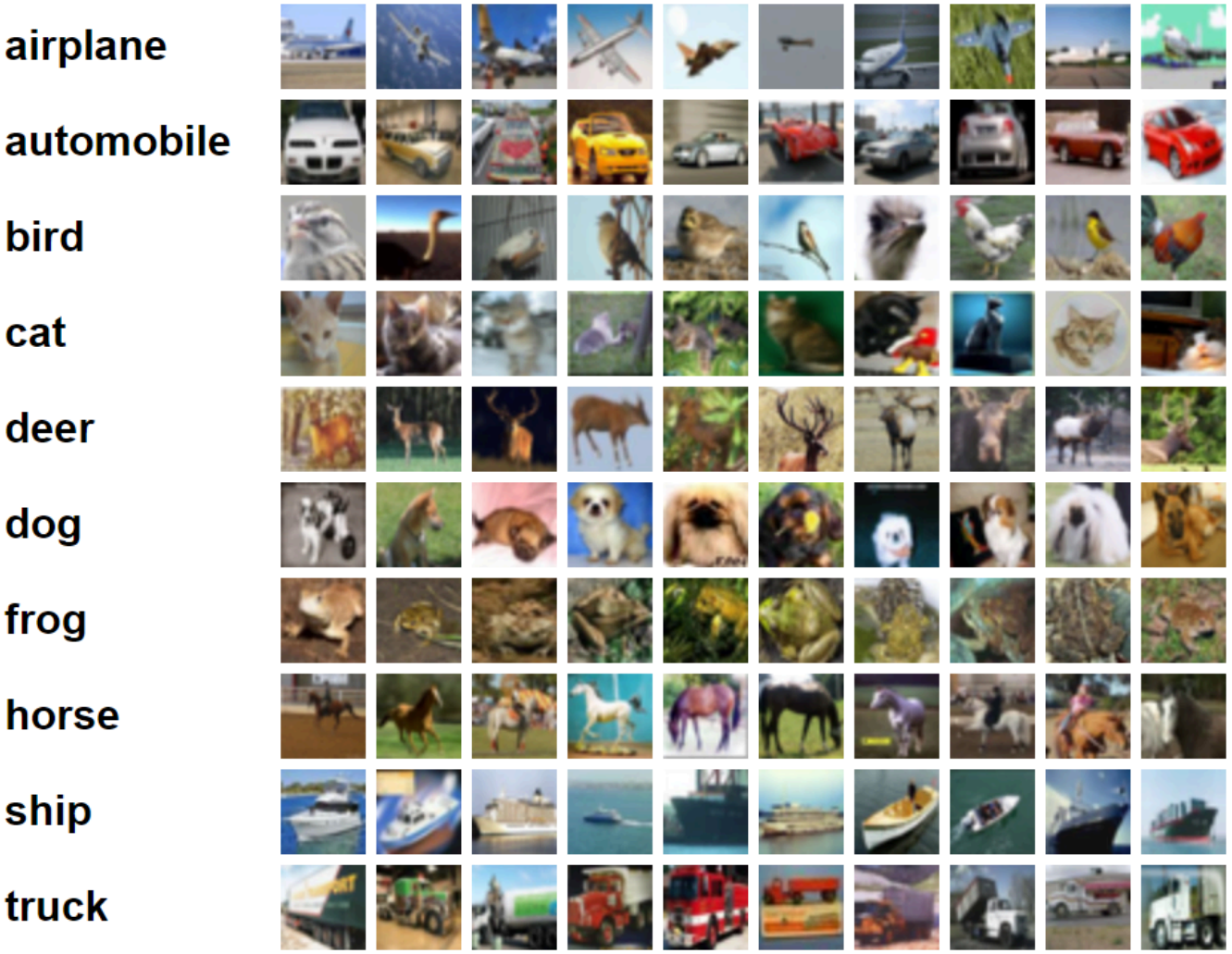


Figure 2.3.1 Classes of Cifar10 dataset [3]

2.3.1.2 Self-made convolutional neural network

Using Keras, python's built-in library, I made a convolutional neural network with 14 layers. The CNN was trained on the cifar10 dataset. The accuracy of the CNN on the test dataset is 83%.

2.3.1.3 Fault detection using clustering

After the measurement, HDBSCAN recognised 193 clusters:

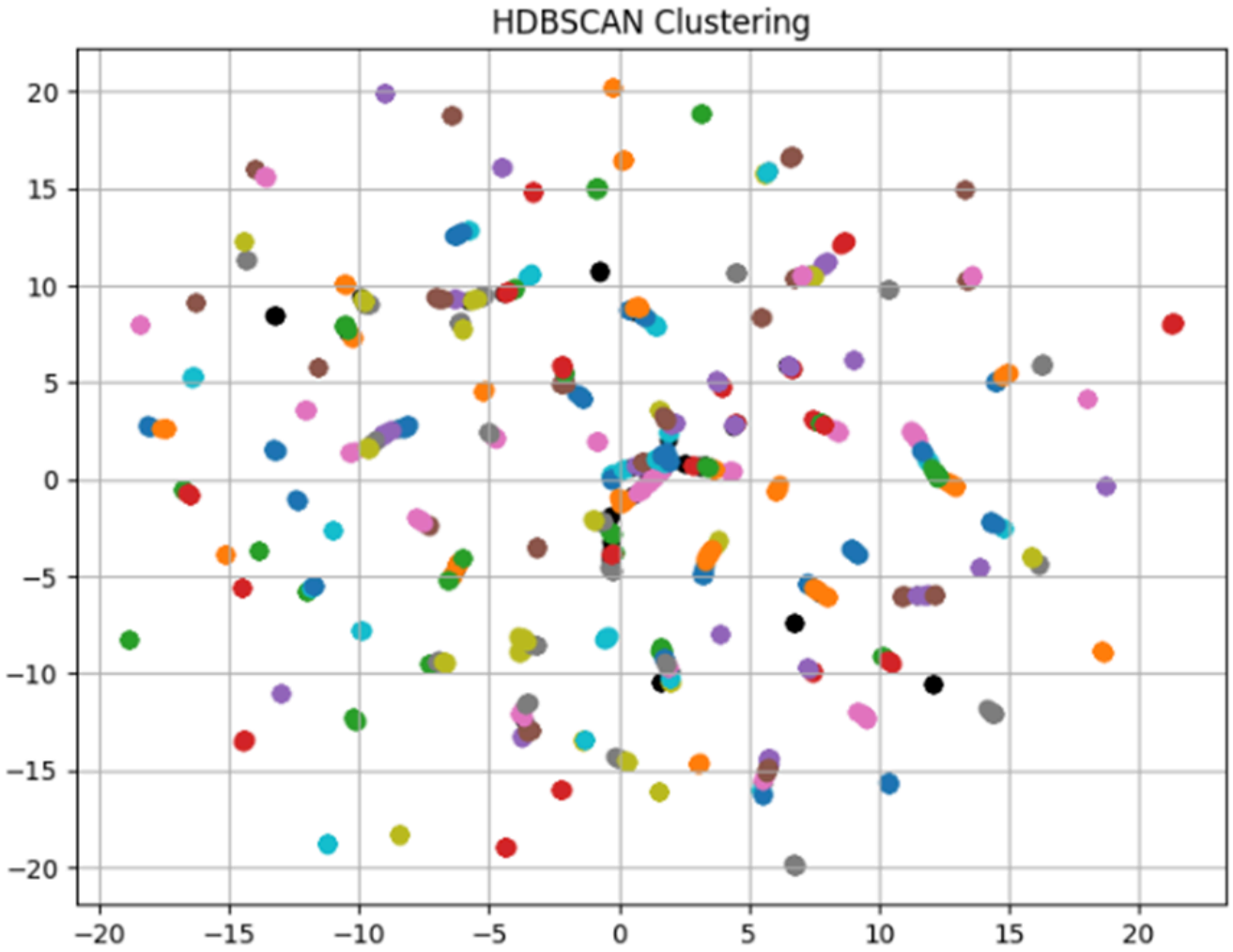


Figure 2.3.1.3 Clusters on Cifar10 dataset

Some of the clusters:

Cluster1

Cluster2

Cluster3

2.3.2 Dataset extracted from traffic simulator

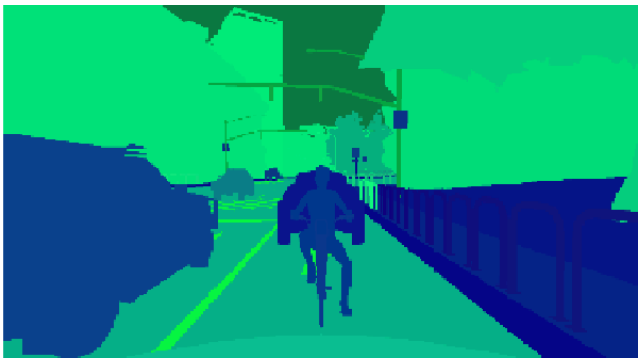
2.3.2.1 Dataset

The dataset contains images that were extracted from a traffic simulator:

Besides RGB images, the dataset contains the ground truth image, and a colormap that assigns an object to each RGB color:

Original Image

Ground Truth Image



2.3.2.2 BiSeNet model

BiSeNet, or Bilateral Segmentation Network, is a deep learning architecture designed for real-time semantic segmentation. It aims to balance the trade-off between high accuracy and fast processing speed, making it suitable for applications requiring real-time performance, such as autonomous driving and video surveillance.

In this measurement, I had access to the BiSeNet model output for the dataset images, as well as the corresponding colormap (the detected objects assigned to each color):

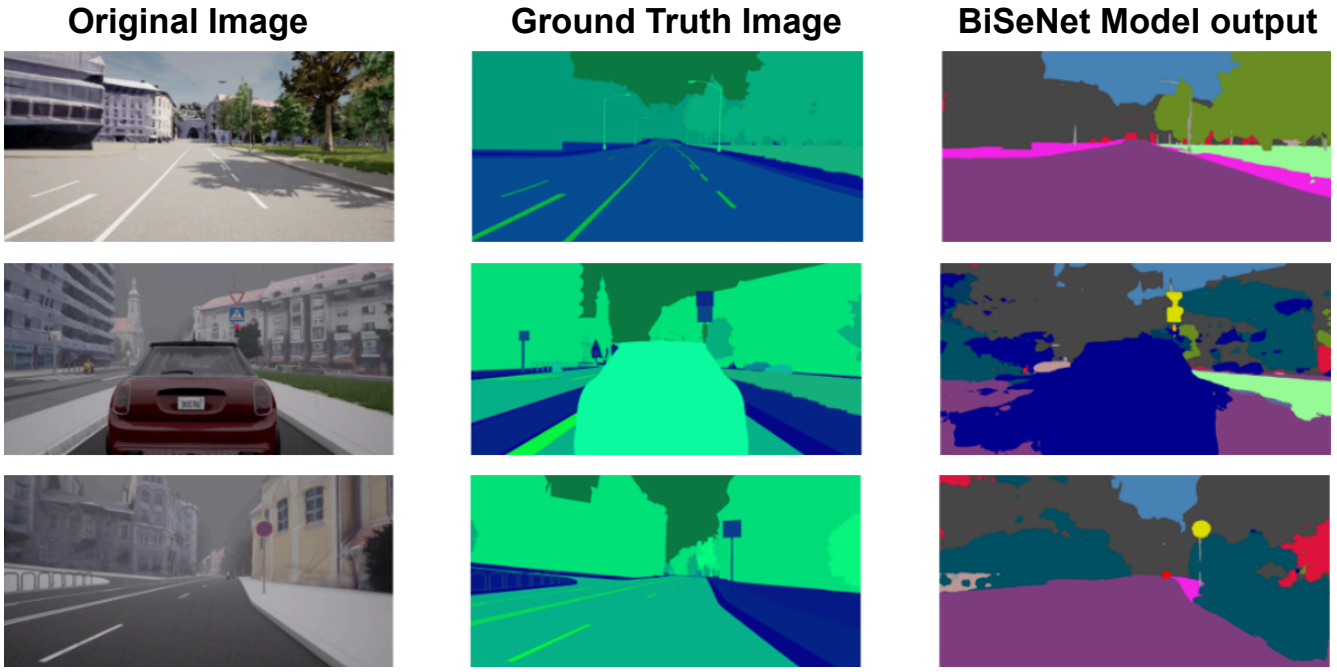
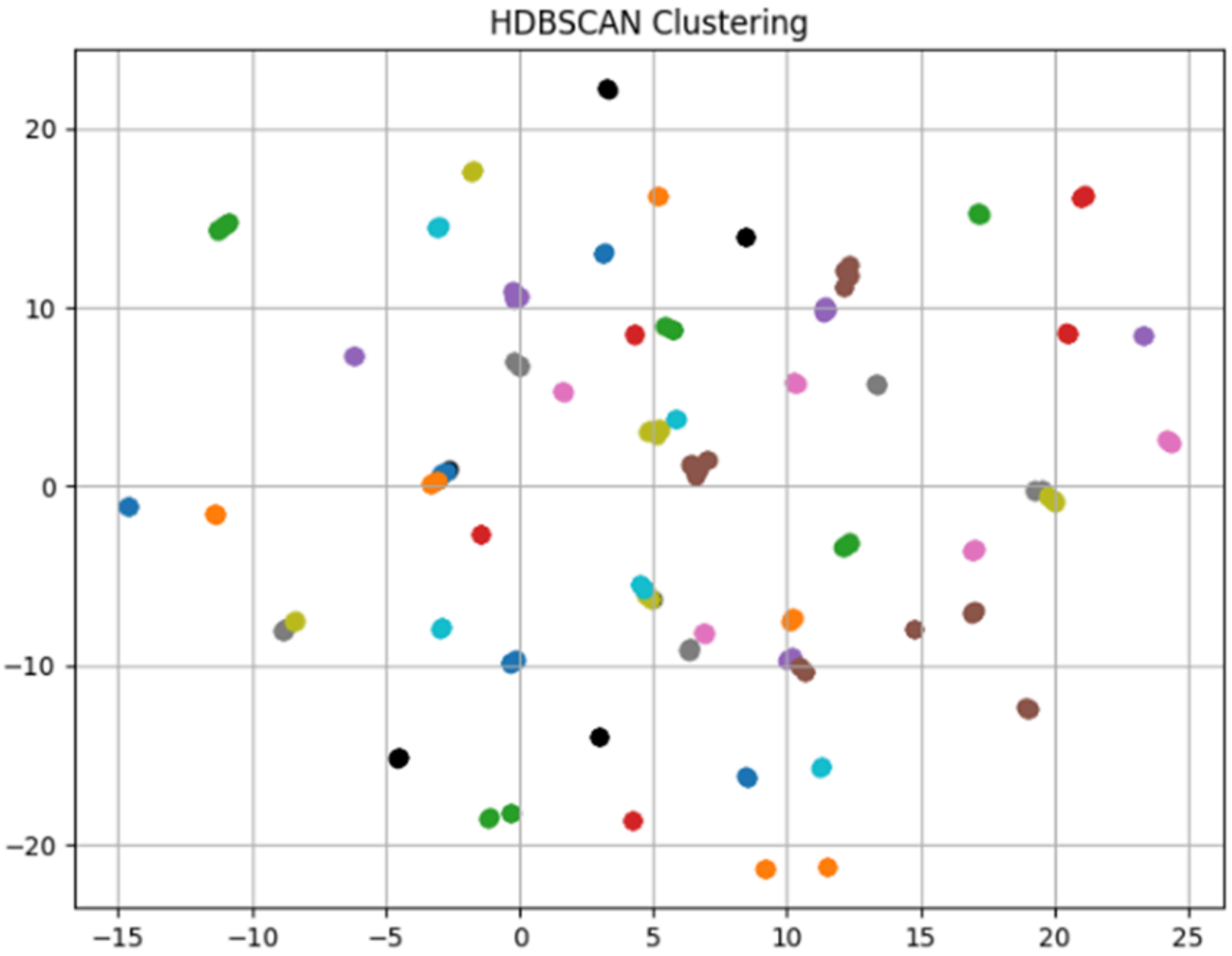


Figure 3.2.2.2 Measurement datasets

2.3.2.3 Fault detection using clustering

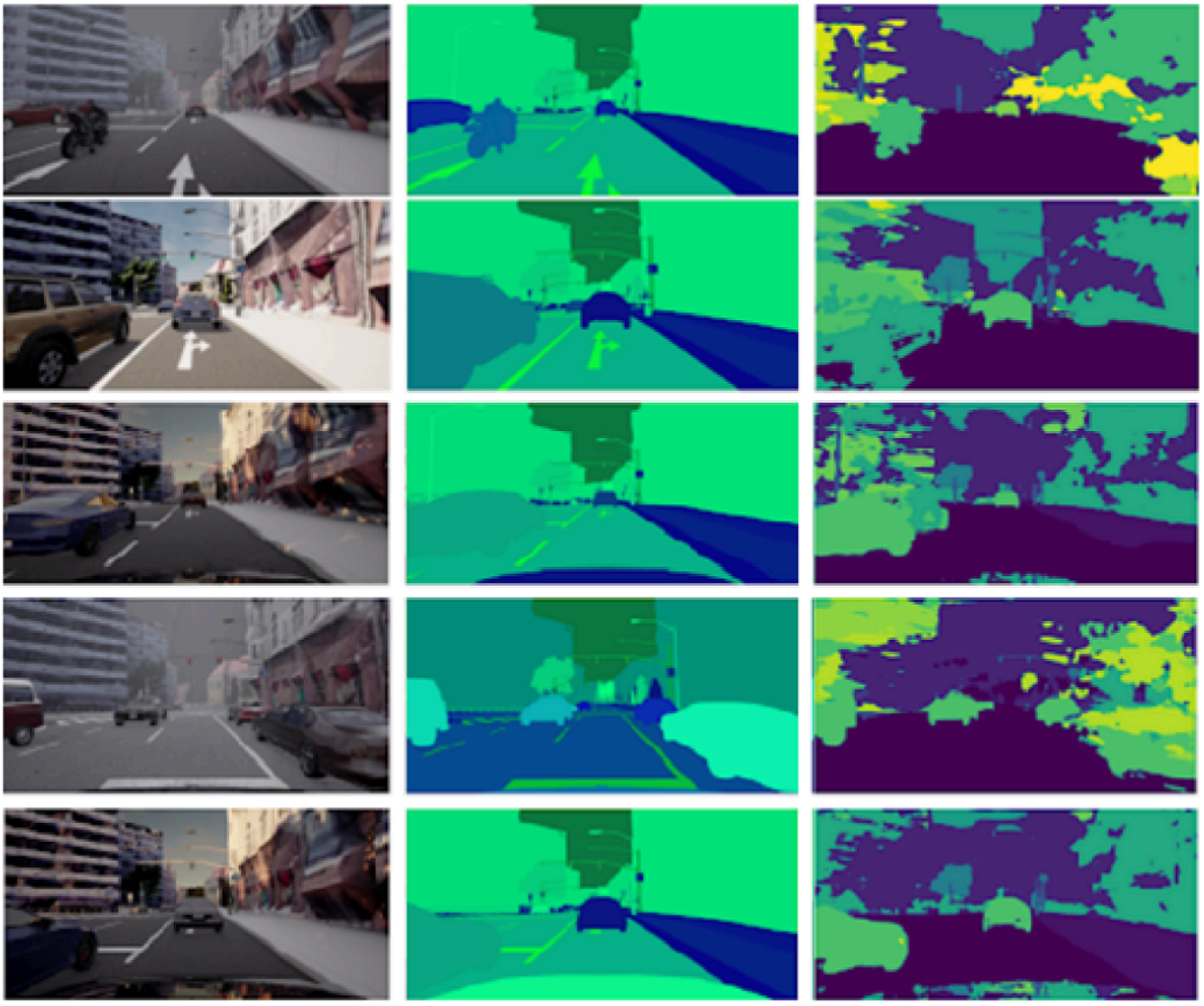
After the measurement, HDBSCAN recognised 50 clusters:



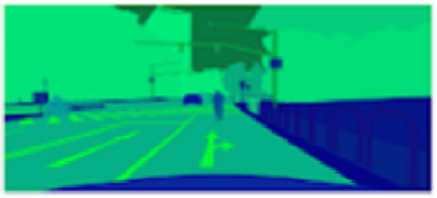
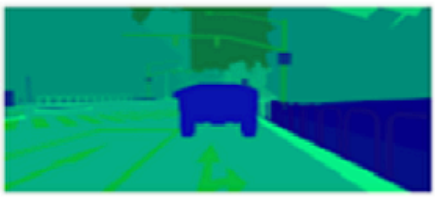
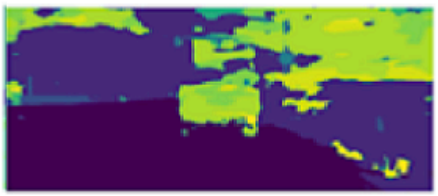
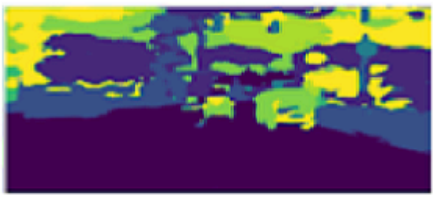
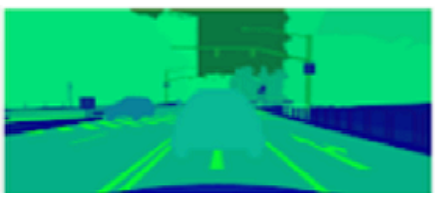
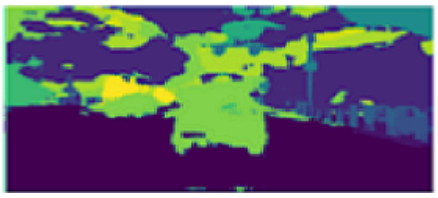
Some of the clusters:

Cluster1

Original Image	Ground Truth Image	BiSeNet Model output



Cluster2

Original Image	Ground Truth Image	BiSeNet Model output
		
		
		
		
		

3. Conclusion

The measurement can be considered successful, as the individual clusters are clearly distinguishable to the naked eye.

3.1 Cifar10 & Self made CNN

The individual clusters are well distinguishable, and based on the measurement, it can be inferred that the feature vectors of the images correlate with the specific error causes, as these vectors can be grouped, and seemingly similar images are found within the same cluster.

3.2 Traffic simulator dataset & BiSeNet model

In this measurement, it is also visually apparent that similar images are grouped into the same cluster. However, two factors degrade the final result. First, greater emphasis should be placed on the misdetection of cars and pedestrians among the images considered erroneous. Second, among many images, there are only a few framework differences, which should also be filtered out. Ignoring these factors, the most significant errors in the error groups are the misdetections of buildings in the background, which is not really relevant in an autonomous vehicle's image recognition.

References & Sources

[1] Zohreh Aghababaeyan , Manel Abdellatif , Lionel Briand , Fellow, IEEE, Ramesh S , and Mojtaba Bagherzadeh: Black-Box Testing of Deep Neural Networks through Test Case Diversity - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10041782

[2] VGG 16 dataset- https://www.researchgate.net/profile/Timea-Bezdan/publication/333242381/figure/fig2/AS:760979981860866@1558443174380/VGGNet-architecture-19.ppm

[3] Cifar10 dataset- https://www.cs.toronto.edu/~kriz/cifar.html