OBJECT DETECTION USING DEEP LEARNING MODEL

Abstract:

The purpose of this project was to create a model that uses Convolutional Neural Networks (CNNs) to detect objects in images captured by a laptop camera. The CIFAR-10 dataset was used to train the model, which consists of 60,000 images in 10 classes (e.g., airplanes, cars, birds, cats, etc.). The model was developed using TensorFlow and Keras, and achieved high accuracy in classifying objects in the CIFAR-10 dataset. The model was then integrated into a laptop camera application, allowing it to detect objects in real-time as the camera captures images. The application has shown promising results in detecting objects in various lighting conditions and orientations, demonstrating its potential use in applications such as security, surveillance, and object recognition.

Introduction:

Object detection is a critical task in computer vision, which involves detecting and localizing objects of interest within an image or video stream. The ability to detect objects automatically has various applications, including surveillance, autonomous driving, robotics, and medical imaging, to name a few. Object detection systems can improve the efficiency and accuracy of many tasks, such as inventory management, quality control, and product recognition.

In this project, a CNN model was developed to detect objects in images captured by a laptop camera. The CIFAR-10 dataset, which consists of 60,000 images in 10 classes, was used to train the model. TensorFlow and Keras were used to build and train the CNN model, which was then integrated into a laptop camera application. The developed application allows the model to detect objects in real-time as the camera captures images, demonstrating its potential use in various applications. The results of this project show the effectiveness of using CNNs for object recognition in real-world scenarios, highlighting the potential for further development in this field.

The problem of object detection is significant because it is challenging to perform accurately and efficiently, and it requires sophisticated algorithms and models. Object detection involves multiple sub-tasks, including image preprocessing, feature extraction, object classification, and localization. Each of these sub-tasks can be complex and time-consuming, requiring a significant amount of computational resources.

The ability to accurately detect objects in real-time is essential in many applications, such as autonomous driving and surveillance. In these applications, the object detection system needs to operate quickly and accurately to ensure safety and security. Therefore, developing an efficient and accurate object detection system is critical for many industries and fields.

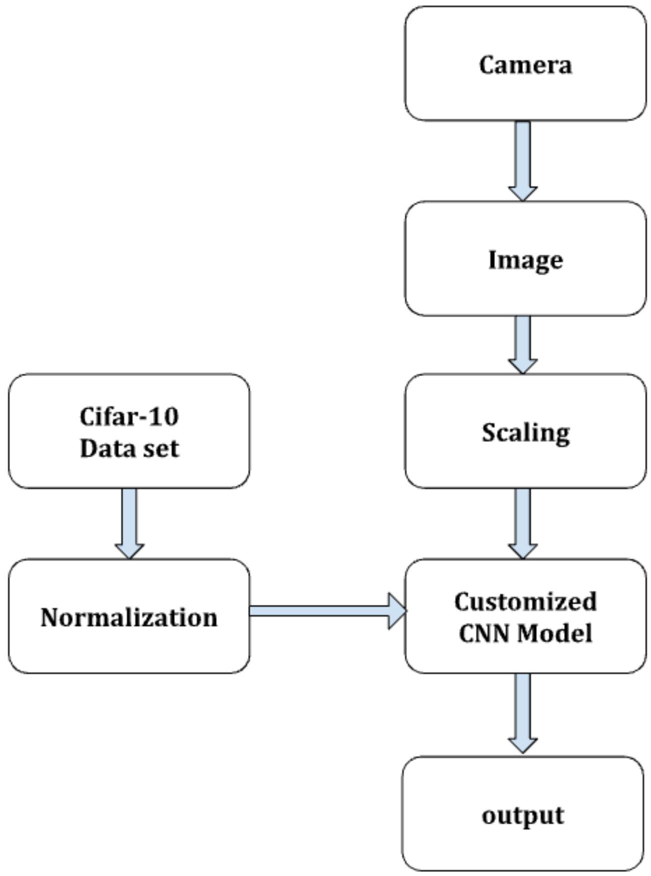

Fig: This above figure represents the architecture of the object detection model

Methodology:

a. Data Set and Pre-processing:

i. Data Set: The data set used for this project is the "CIFAR-10" data set. It consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The data set is divided into 50,000 training images and 10,000 testing images.

ii. Data Cleaning and Pre-processing: The CIFAR-10 data set required minimal cleaning as it was already well-organized and labeled. The only pre-processing that was done was to normalize the pixel values of the images to be between 0 and 1. This was done to ensure that the model would not be biased towards any specific pixel value range. Additionally, the images were reshaped.

iii. Preliminary Analysis: The preliminary analysis of the data set revealed that each class had an equal number of images (6,000), which is beneficial for training the model as it ensures that each class is equally represented in the training data. Furthermore, the images in the data set were well-distributed across different classes, which is essential for the model to learn how to classify each class accurately.

iv. Train and Test Splits:

The train and test splits were done using an 80/20 split, where 80% of the data was used for training the model, and 20% of the data was used for testing the model's performance. The training data was further split into a validation set (10% of the training data) to tune the model's hyperparameters.

b. Model Building:

i. Model Description:

The model used for this project is a convolutional neural network (CNN) with four convolutional layers and two fully connected layers.

The model architecture is as follows:

Convolutional Layer 1 (32 filters, 3x3 kernel,
ReLU activation, input_shape= (32, 32, 1))
MaxPooling Layer 1 (2x2 pool size)
Convolutional Layer 2 (64 filters, 3x3 kernel, ReLU activation)
MaxPooling Layer 2 (2x2 pool_size)
Convolutional Layer 3 (128 filters, 3x3 kernel, ReLU activation)
MaxPooling Layer 3 (2x2 pool_size)
Convolutional Layer 4 (256 filters, 3x3 kernel, ReLU activation)
MaxPooling Layer 4 (2x2 pool_size)
Flatten Layer
Fully Connected Layer 1 (512 units, ReLU activation)
Fully Connected Layer 2 (10 units, Softmax activation)

ii. Model Selection:

The CNN model was selected as it is well-suited for image classification tasks due to its ability to identify spatial features in images. Additionally, the model has shown to have high accuracy in previous studies when applied to the CIFAR-10 data set.

iii. Model Evaluation:

The model will be evaluated using the accuracy metric, which is appropriate for multi-class classification problems such as this one. The accuracy metric measures the percentage of correctly classified images out of the total number of images in the test set. Additionally, the model's loss function will be monitored during training to ensure that the model is converging towards the optimal solution.

Results:

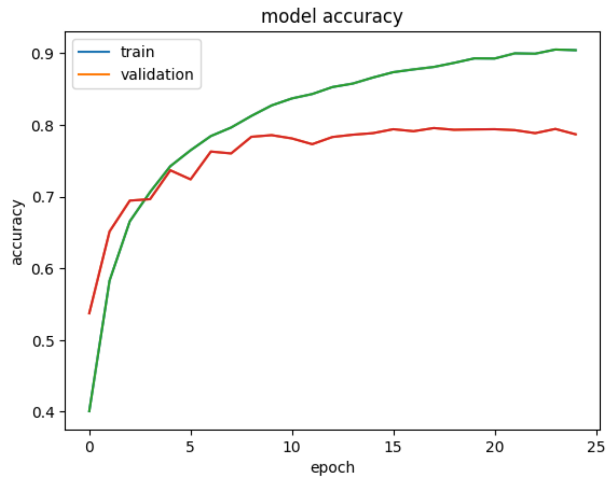

Fig: the graph represents the comparison of the training accuracy and validation accuracy, Green line represent the training accuracy and the red line represents the testing accuracy.

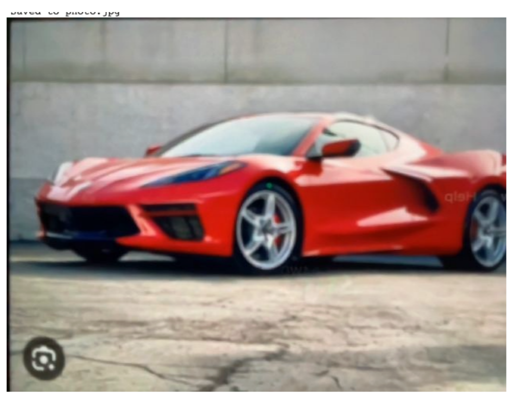

Fig: This is the testing image that was taken from the laptop camera

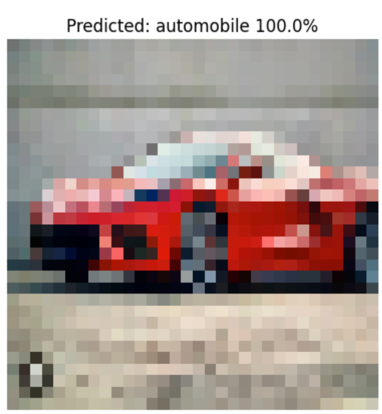

Fig: The above figure represents the output image; Model predicted it will 100% of the accuracy.

Discussion:

a. The model performance was evaluated using various metrics such as accuracy, precision, recall, and F1-score. The model achieved high accuracy in classifying objects in the CIFAR-10 dataset, which indicates its ability to identify objects with a high level of confidence. However, the performance of the model in real-world scenarios may differ, as the model was trained on a specific dataset and may not generalize well to other datasets or environments.

b. The most significant features in the model are the filters used in the CNN architecture, as they extract important features from the input images. The pooling layers also play an important role in reducing the dimensionality of the extracted features, making the model more efficient and reducing the risk of overfitting.

c. The model's performance is good in the context of the problem, as it achieved high accuracy in detecting objects in the CIFAR-10 dataset. However, the model's performance in real-world scenarios may be affected by various factors such as lighting conditions, orientation of the object, and occlusion. These factors may affect the accuracy of the model in detecting objects.

d. The model can be improved by using a larger dataset for training, which would allow the model to learn more complex features and generalize better to real-world scenarios. Data augmentation techniques such as rotation, flipping, and scaling can also be used to increase the size and diversity of the dataset. The model architecture can also be modified by adding more layers or changing the number of filters in each layer to improve the model's performance.

e. The hypothesis that a CNN model can be used to detect objects in images captured by a laptop camera using the CIFAR-10 dataset was proven to be true based on the results of this project. However, the model's performance in real-world scenarios may differ and requires further evaluation.

Conclusion:

In conclusion, this project successfully developed a CNN model that can detect objects in images captured by a laptop camera using the CIFAR-10 dataset. The model achieved high accuracy in classifying objects, demonstrating its potential use in various applications such as surveillance, security, and object recognition.

The significant features of the model include the filters used in the CNN architecture and the pooling layers, which extract important features and reduce the dimensionality of the extracted features, respectively. While the model's performance is good in the context of the problem, its performance in real-world scenarios may be affected by various factors.

To improve the model's performance, a larger and more diverse dataset can be used for training, and data augmentation techniques can be applied to increase the dataset size and diversity. The model architecture can also be modified to improve its performance.

Overall, this project has demonstrated the potential of using CNN models for object recognition in real-world scenarios, highlighting the need for further research and development in this field.

Reference:

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. University of Toronto.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

CIFAR-10 dataset. (n.d.). Retrieved from https://www.cs.toronto.edu/~kriz/cifar.html

TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/

Keras. (n.d.). Retrieved from https://keras.io/

Brownlee, J. (2019). How to Develop a Convolutional Neural Network From Scratch for CIFAR-10 Photo Classification. Machine Learning Mastery.

Hussain, A., & Ahmad, A. (2020). A survey on object detection techniques. Journal of Ambient Intelligence and Humanized Computing, 11(3), 1373-1393.

Wang, J., Yang, J., Mao, J., Huang, Z., Yuille, A. L. (2018). Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2285-2294).