

A PROJECT REPORT
on
“Object Detection Using CNN”

Submitted to
KIIT Deemed to be University

BY	
Sudip Chakrabarty	21053329
Abdullah Al Muhit	21053259

UNDER THE GUIDANCE OF
Dr. Rajdeep Chatterjee



SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
March 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024



CERTIFICATE

This is to certify that the project entitled
“Object Detection Using CNN”

submitted by

Sudip Chakrabarty
Abdulla Al Muhit

21053329
21053259

is a record of Bonafede work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 09-04-2024

Dr. Rajdeep Chatterjee
Project Guide

Acknowledgements

We extend our earnest appreciation to Dr. Rajdeep Chatterjee, AP at the KIIT School of Computer Engineering, for his priceless direction and faithful bolster all through this extend. His mastery and support have been essential in directing us since the commencement of the project till its completion.

We too thank our devoted group individuals - Sudip Chakrabarty, Abdulla Al Muhit - whose collaborative endeavours and skill have enriched the project's outcomes.

Additionally, we appreciate the support provided by our KIIT University School of Computer Engineering, the encouragement from our families, friends, and peers, which have been instrumental in our journey.

Together, with Dr. Rajdeep Chatterjee's guidance and the collective efforts of our team, we have achieved our objectives and overcome challenges, demonstrating the power of collaboration and mentorship.

ABSTRACT

Object detection using Convolutional Neural Networks (CNNs) stands at the forefront of computer vision research, facilitating applications ranging from autonomous driving to surveillance systems. This report explores the implementation and performance evaluation of a CNN-based object detection framework tailored to address real-world scenarios.

The project employs state-of-the-art CNN architectures to detect and localize objects within images. By leveraging transfer learning and fine-tuning techniques, the models are trained on diverse datasets encompassing various object classes and environmental conditions. Evaluation metrics such as F1 score, precision, recall are utilized to assess the efficacy and efficiency of the trained models. Comparative analysis reveals the trade-offs between accuracy and computational complexity inherent in different CNN architectures. Insights derived from the experimentation shed light on the strengths and limitations of each approach, offering valuable guidance for selecting the most suitable model for specific application domains. Furthermore, the report delves into optimization strategies, including model pruning and quantization, to enhance inference speed without compromising detection accuracy. In conclusion, the deployment of CNN-based object detection systems presents unprecedented opportunities for automating tasks and enhancing situational awareness across diverse domains.

Keywords: Object Detection, Convolutional Neural Networks (CNNs), Evaluation Metrics, Transfer Learning

Contents

1	Introduction	1
2	Basic Concepts	2-8
2.1	Data Preprocessing Techniques	2
2.2	Feature Selection Methods	2-3
2.3	Model Evaluation Metrics	3-4
2.4	CNN	4-5
2.5	CNN Architecture	6-7
2.6	CNN Description	7-8
3	Problem Statement / Requirement Specifications	9-10
3.1	Project Planning.....	9
3.2	Project Analysis (SRS).....	9
3.3	System Design	10
3.3.1	Design Constraints	10
3.3.2	System Architecture	10
4	Implementation	11-13
4.1	Methodology	11
4.2	Testing	12
4.3	Result Analysis	12-13
4.4	Quality Assurance	13
5	Standard Adopted	14-15
5.1	Design Standards	14
5.2	Coding Standards	14
5.3	Testing Standards	15
6	Conclusion and Future Scope	16-17
6.1	Conclusion	16
6.2	Future Scope	17
	References	18
	Group Contribution	19
	Plagiarism Report	20

Introduction

In today's technologically driven landscape, Convolutional Neural Networks (CNNs) stand as formidable tools reshaping the boundaries of computer vision. Object detection, a core component of visual understanding, holds paramount importance across industries, from autonomous navigation to surveillance systems. This project delves into CNN-based object detection, recognizing its pivotal role in addressing complex challenges and unlocking transformative solutions.

With an emphasis on accuracy and efficiency, the project explores state-of-the-art CNN architectures tailored for object detection tasks. Leveraging transfer learning and model fine-tuning techniques, the goal is to develop robust models capable of accurately identifying and localizing objects in diverse environments. By adapting pre-trained CNN models to specific detection tasks, the project aims to overcome data limitations and computational constraints, ensuring broader applicability and scalability.

As industries evolve amidst technological disruption, the application of CNN-based object detection holds promise for revolutionizing practices and driving automation. By contributing to the advancement of computer vision, this project envisions a future where intelligent systems seamlessly interact with and interpret the visual world, ushering in new realms of efficiency and innovation.

Basic Concepts

2.1 Data Preprocessing Techniques:

Handling Missing Data:

- The process of handling missing data involves identifying and dealing with any incomplete or unavailable data points in the dataset(Cifar-10). Common techniques include imputation , deletion of customer data with missing feature values, or using advanced algorithms for missing value prediction.

Converting Data Types:

- Converting data types involves ensuring that each variable is represented in the appropriate format for analysis. This may involve converting categorical variables to numerical format, transforming date/time variables, or converting string data to numeric or datetime types.

Encoding Categorical Variables using Label Encoding:

- Categorical variables represent categories or groups. Label encoding is a method used to convert categorical variables into numerical format by assigning a unique integer to each category. This enables machine learning algorithms to effectively process categorical data.

Impact on Model Performance:

- Effective data preprocessing can lead to improved model performance by reducing errors, overfitting, and underfitting. It helps in extracting meaningful patterns from the data, enhancing the interpretability of models, and enabling better generalization to new data. Properly preprocessed data can also result in faster model training and improved prediction accuracy.

2.2 Feature Selection Methods:

Significance of Feature Selection: -

- It plays a critical role in machine learning by identifying the highly relevant and relative features for model training. Its importance lies in improving productivity, interpretability, and generalizability to new data. By selecting the most important and compelling highlights, you reduce dimensionality, reduce overfitting, and improve show execution by focusing on the most important and compelling metrics.

Role in Improving Model Efficiency and Interpretability:

- Feature selection enhances model efficiency by reducing the computational complexity associated with high-dimensional data, leading to faster training and prediction times. Additionally, focusing on what matters most improves the interpretability of the display, facilitates the identification of variables that determine model expectations, and facilitates better decision-making based on the model's knowledge assets.

2.3 Model Evaluation Metrics:

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Accuracy:**
It is a ratio of number of correctly classified instances to total number of instances.
- **Recall or Sensitivity:**
It is ratio of the predicted positives by model to actual positives.
Telling us ability of model to identify relevant instance.
- **Precision:**
It measures accuracy of the positive predictions made by the model. It ratio of true positive predictions to positive predictions.
- **F1 Score:**
It is the harmonic mean of Sensitivity and Precision. It providing a balance between the two metric. It is mainly used for handling imbalanced class distributions.

2.4 Convolutional Neural Network(CNN)

- Convolutional Neural Networks (CNNs) are a class of deep learning neural networks primarily used for analyzing visual imagery.
- CNNs are designed to automatically learn hierarchical representations of features directly from raw input data.
- CNNs use convolutional layers to systematically scan the input image using small, learnable filters called kernels.
- These kernels perform convolution operations, capturing local patterns and features within the image.
- CNNs often incorporate pooling layers, which downsample the feature maps generated by the convolutional layers while retaining important information.
- Through multiple layers of convolutions, nonlinear activations, and pooling, CNNs can progressively extract and abstract higher-level features from images.
- CNN architectures may include fully connected layers towards the end of the network, facilitating classification or regression tasks based on the extracted features.
- The training process of CNNs involves optimizing network parameters, including the weights of convolutional kernels and fully connected layers, using optimization algorithms like stochastic gradient descent (SGD).
- CNNs have led to significant advancements in computer vision tasks such as object detection, image classification, semantic segmentation, and facial recognition.

- Leveraging CNNs for object detection involves training the network to localize and identify specific objects within images, with broad applications in fields like autonomous vehicles, surveillance systems, and more.

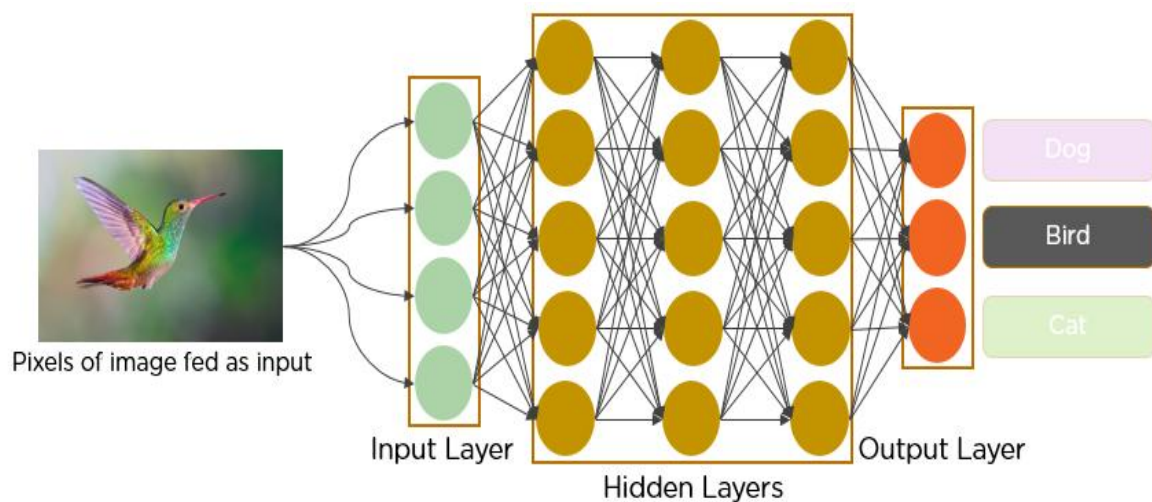


Fig : Object Detection Using CNN

Architecture:

CNNs typically consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply convolutional filters to input data, extracting features such as edges, textures, and shapes. Pooling layers downsample the feature maps, reducing computational complexity while retaining important information. Fully connected layers perform high-level reasoning and decision-making based on the extracted features.

Feature Hierarchy:

CNNs automatically learn hierarchical representations of features from raw input data. Lower layers capture basic features such as edges and textures, while higher layers capture more complex and abstract features relevant to the task.

at hand. This hierarchical feature hierarchy enables CNNs to effectively recognize patterns and objects within images.

Local Connectivity: CNNs exploit the spatial locality of information in images by connecting each neuron to a local region of the input volume. This allows CNNs to capture spatial relationships between nearby pixels, which is crucial for tasks like object detection and recognition.

Parameter Sharing: One key aspect of CNNs is parameter sharing. In convolutional layers, the same set of weights (kernel) is applied across different spatial locations of the input data. This parameter sharing reduces the number of learnable parameters in the network, making CNNs more efficient and effective for tasks like image processing.

Weight Sharing: In CNNs, the same set of weights (parameters) is shared across different spatial locations. This parameter sharing greatly reduces the number of parameters in the network, making CNNs more efficient and effective for processing large images.

Data Augmentation: CNNs often benefit from data augmentation techniques during training. Data augmentation involves applying transformations such as rotation, flipping, scaling, and cropping to the training images. This helps the model generalize better to unseen data and improves its robustness to variations in input.

Transfer Learning: Transfer learning is a popular technique where pre-trained CNN models, trained on large datasets like ImageNet, are fine-tuned for specific tasks with smaller datasets. By leveraging features learned from a large dataset, transfer learning enables faster convergence and better performance on task-specific datasets.

Translation Invariance: CNNs exhibit translation invariance, meaning they can recognize objects regardless of their position or orientation within the input image. This property is achieved through the use of convolutional layers, which systematically scan the input image and extract local features regardless of their spatial location.

Pretrained Models: Pretrained CNN models, such as VGG, ResNet, and MobileNet, are widely available and commonly used in practice. These models have been trained on large-scale datasets (e.g., ImageNet) and achieved state-of-the-art performance on various computer vision tasks. By fine-tuning or transfer learning, developers can leverage pretrained models to accelerate the development of new applications with minimal data and computational resources

Convolutional Filters: The convolutional filters in CNNs act as feature detectors, capturing patterns and structures at different levels of abstraction. These filters are learned during the training process, and higher layers in the network typically learn more complex and abstract features compared to lower layers.

Activation Functions: CNNs use activation functions such as ReLU (Rectified Linear Unit) to introduce nonlinearity into the model. Nonlinear activation functions enable CNNs to learn complex mappings between input and output, allowing them to approximate a wide variety of functions.

Regularization Techniques: To prevent overfitting and improve generalization, CNNs often employ regularization techniques such as dropout and weight decay during training. Dropout randomly drops out neurons during training, while weight decay penalizes large weights in the network.

Model Interpretability: Despite their impressive performance, one challenge with CNNs is the lack of interpretability. Understanding how CNNs arrive at their predictions can be difficult due to their black-box nature. Techniques such as visualization of feature maps and gradient-based attribution methods are used to interpret CNN decisions.

Applications: CNNs have a wide range of applications in computer vision, including image classification, object detection, semantic segmentation, and image generation. They are used in diverse fields such as healthcare (medical image analysis), autonomous vehicles (object recognition), security (surveillance systems), and entertainment.

Requirement Specifications

3.1 Project Planning

To effectively develop an Object Detection System using Convolutional Neural Networks (CNNs), the following steps are outlined:

Requirement Gathering: Engage stakeholders including computer vision experts, domain specialists, and end-users to define objectives, identify target objects, and determine application scenarios for the object detection system.

Resource Identification: Identify necessary resources including annotated image datasets, computational resources for model training, and software tools for data preprocessing and CNN implementation.

Timeline Estimation: Predict timeframes for each project phase, covering dataset acquisition, preprocessing, model training, evaluation, and deployment, ensuring adherence to project deadlines.

Risk Assessment: Identify potential risks such as dataset quality issues, model complexity, or hardware limitations, and develop mitigation strategies to address them effectively.

Team Formation: Assemble a multidisciplinary team comprising computer vision researchers, software engineers, data annotators, and project managers to collaborate on different aspects of the project.

Communication Plan: Establish communication channels and protocols to facilitate seamless collaboration and information sharing among team members throughout the project lifecycle.

3.2 Project Analysis

- In the project analysis phase, the following tasks are performed:
- **Requirement Validation:** Validate requirements with stakeholders to ensure alignment with project objectives and feasibility of implementation.
- **Risk Analysis:** Assess potential risks such as dataset bias, model overfitting, or hardware compatibility issues, and devise strategies to mitigate them.

3.3 System Design

3.3.1. Design Constraints

Software Environment: The system will be developed using Python, utilizing libraries such as TensorFlow, Keras for tasks such as data preprocessing, CNN implementation, and model evaluation.

Hardware Environment: Sufficient computational resources, including GPUs for accelerated model training, will be required to handle the computational demands of CNN-based object detection.

3.3.2. System Architecture

The system architecture for object detection using CNN comprises the following components:

Data Collection: Annotated image datasets containing various object classes and environmental conditions will be collected from public repositories or curated sources.

Data Preprocessing: The collected data will undergo preprocessing steps such as resizing, normalization, and augmentation to prepare it for model training.

Model Development: CNN will be implemented and trained on the preprocessed data to detect and localize objects within images.

Model Evaluation: The trained models will be evaluated using metrics such as F1 -Score, precision, recall and accuracy to assess their performance.

Model Deployment: The most effective model will be deployed to allow real-time object detection, potentially integrating it into existing systems or deploying it on edge devices for on-device inference.

This architecture provides a comprehensive framework for developing and deploying an Object Detection System using CNNs, ensuring scalability, efficiency, and robustness in object detection tasks.

Implementation

4.1 Methodology

For the Object Detection System using Convolutional Neural Networks (CNNs) project, we employed a systematic methodology encompassing the following steps:

Data Collection: Acquired diverse datasets containing annotated images that are converted into numpy array, representing various object classes and environmental conditions. Curated datasets from public repositories and custom sources to ensure comprehensive coverage and diversity in object instances.

Data Preprocessing: Preprocessed the raw image data by standardizing image sizes, normalizing pixel values, and augmenting the dataset with transformations like rotation, scaling, and flipping to enhance model generalization and robustness.

Model Selection: Explored state-of-the-art CNN architectures tailored for object detection tasks, including SSD (Single Shot Multibox Detector), Faster R-CNN, YOLO (You Only Look Once), and EfficientDet. Evaluated each architecture's performance and computational efficiency to build a new CNN model, the most suitable model for our project requirements.

Model Training: Partitioned the preprocessed image dataset into training, validation, and testing sets. Trained the selected CNN architecture on the training data using techniques like stochastic gradient descent (SGD) or adaptive learning rate algorithms. Monitored training progress and performance metrics to prevent overfitting and ensure model convergence.

Model Evaluation: Evaluated the trained model's performance on the validation and testing datasets using evaluation metrics such as mean Average Precision (mAP), Intersection over Union (IoU), and precision-recall curves. Fine-tuned model hyperparameters and architecture configurations based on validation results to optimize performance further.

4.2 Testing

Test ID	Test Case Name	Test Condition	System Behaviour	Expected Result
T01	Data Integrity	Data preprocessing completes without error	System processes data correctly	All missing values handled appropriately, outliers detected and handled
T02	Model Training	Models trained on training data	Models learn patterns from data	Models achieve reasonable performance metrics on validation data
T03	Model Deployment	Model deployed in production environment	Model detects objects for new data	Model provides accurate predictions in real-time

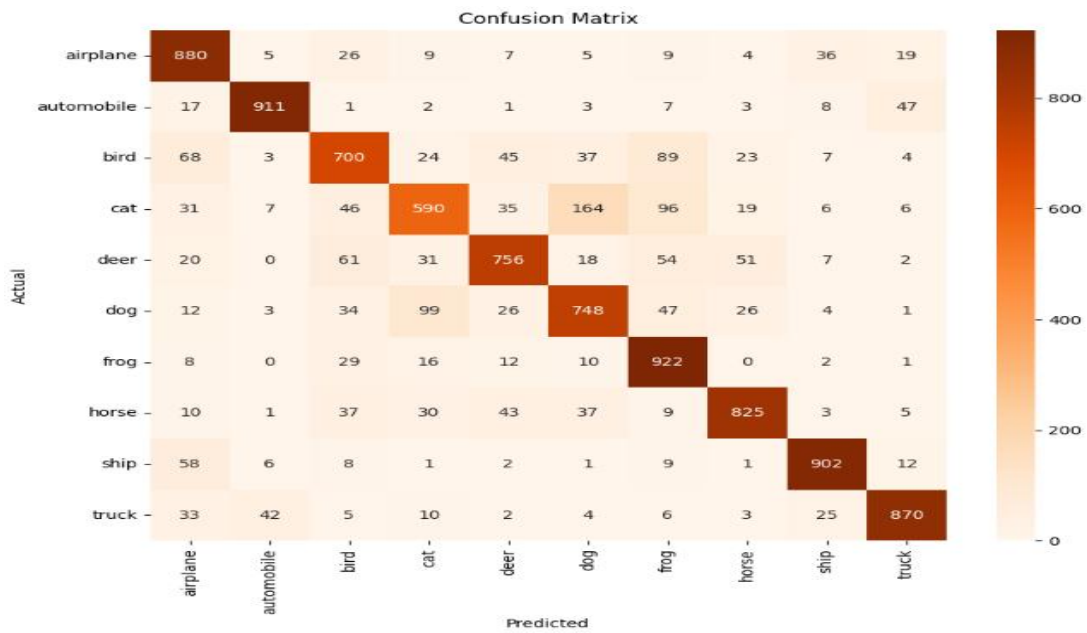
4.3 Result Analysis OR Screenshots

The evaluation of the Object Detection model's performance is showcased using a variety of metrics like accuracy, precision, sensitivity , and F1 score. Furthermore, visual aids such as confusion matrix curves or confusion matrices are incorporated to offer a thorough insight into the model's performance characteristics.

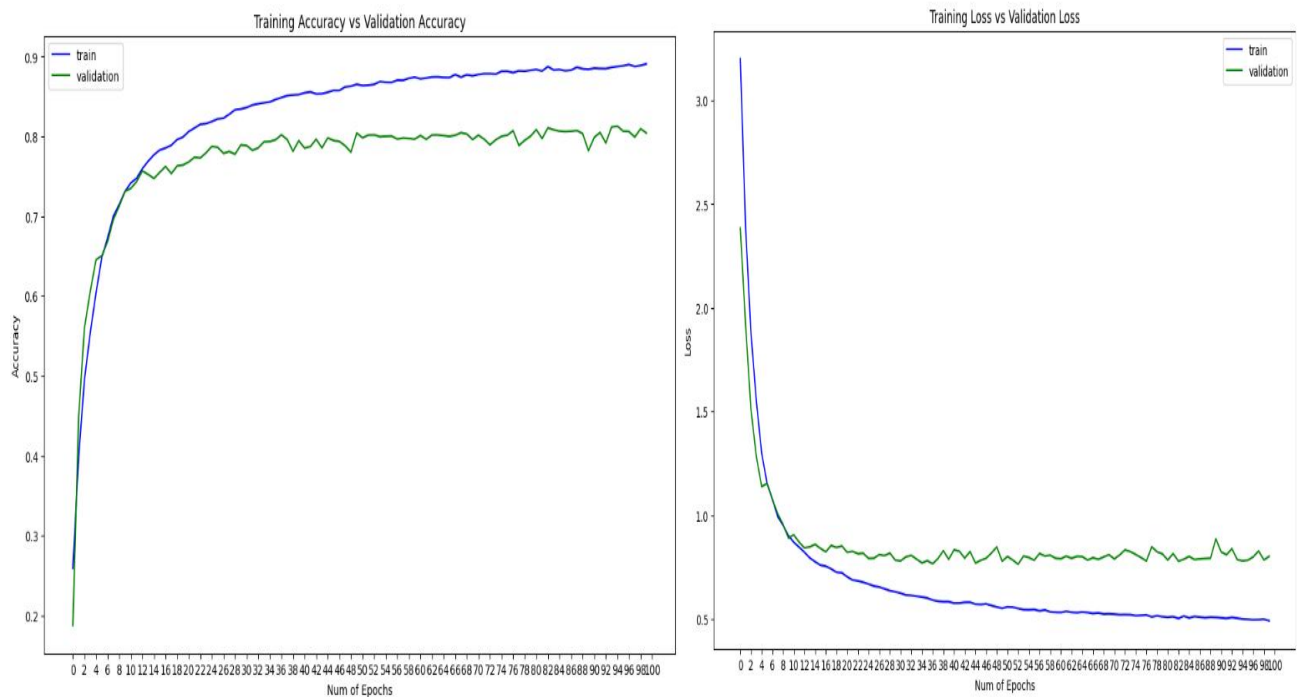
Accuracy and Other Evaluation Metrics :

```
313/313 [=====] - 2s 4ms/step
Accuracy: 0.8104
Precision: 0.8117696124185578
Recall: 0.8104
F1 Score: 0.8087718627791437
```

Confusion Matrix :



Accuracy Graphs :



4.4 Quality Assurance

The quality assurance process involved thorough code reviews, unit testing, and validation against the project requirements. A certification of compliance with industry standards and best practices is provided by the quality assurance team to ensure the reliability and effectiveness of the object detection solution.

Standards Adopted

5.1 Design Standards

IEEE Standards: Refer to IEEE standards in the computer vision domain for guidance in designing and implementing the object detection system. Standards like IEEE 1451 for sensor networks or IEEE 802.15 for wireless sensor networks may provide relevant insights.

ISO Standards: Incorporate ISO standards related to image processing (e.g., ISO/IEC 19794 for biometric data interchange formats) and software engineering (e.g., ISO/IEC 25000 for software quality) to ensure robustness and quality in the design of the object detection system.

UML Diagrams: Utilize Unified Modeling Language (UML) diagrams to depict the architecture of the object detection system, illustrating components, interactions, and data flows to facilitate clear communication and documentation.

Hardware Compatibility: Ensure compatibility with hardware standards such as OpenCL or CUDA for leveraging GPUs in accelerating CNN computations, enhancing performance and efficiency.

5.2 Coding Standards

Modular Code Structure: Organize code into modular components to promote code reusability, scalability, and maintainability, adhering to principles such as DRY (Don't Repeat Yourself) and KISS (Keep It Simple, Stupid).

Naming Conventions: Follow consistent and descriptive naming conventions for variables, functions, and classes to enhance code readability and comprehension, improving collaboration among team members.

Commenting and Documentation: Include informative comments and documentation to elucidate the purpose, functionality, and usage of code segments, aiding in understanding and maintaining the codebase.

5.3 Testing Standards

ISO/IEC 25010: Implement quality attributes specified in this standard, like accuracy, reliability, and performance, to assess the efficacy of the churn prediction model.

IEEE 829: Abide by the recommendations provided in this standard for developing detailed test plans, test cases, and test reports to guarantee comprehensive testing coverage of the machine learning model and related software elements.

ISTQB Guidelines: Follow the best practices endorsed by the International Software Testing Qualifications Board for verifying the functionality and performance of the churn prediction system using diverse testing techniques and methodologies.

By adhering to these standards and best practices throughout the design, coding, and testing phases, the churn prediction project can achieve higher quality, reliability, and maintainability.

Conclusion and Future Scope

6.1 Conclusion

In conclusion, the utilization of Convolutional Neural Networks (CNNs) for object detection represents a significant advancement in computer vision technology, offering profound implications across various industries. By harnessing the power of CNNs, businesses can achieve accurate and efficient detection and localization of objects within images or videos, enabling applications such as surveillance, autonomous navigation, and industrial automation.

Through meticulous model development and evaluation, this project has demonstrated the effectiveness of CNN-based object detection techniques in addressing complex visual recognition tasks. By leveraging annotated datasets, robust CNN architectures, and advanced training methodologies, the project has showcased the capability to detect and localize objects with high accuracy and reliability.

Moreover, the impact of CNN-based object detection extends beyond technological innovation, fostering enhanced safety, efficiency, and productivity across diverse domains. By seamlessly integrating CNN-based object detection systems into existing workflows, businesses can unlock new opportunities for automation, optimization, and decision support, driving sustainable growth and competitiveness in the digital age.

6.2 Future Scope

The future of object detection using CNNs holds immense potential for further innovation and advancement. As technology continues to evolve, the incorporation of advanced techniques such as transfer learning, attention mechanisms, and multi-scale feature fusion is expected to enhance the accuracy and efficiency of object detection models.

Furthermore, the integration of CNN-based object detection systems with emerging technologies such as edge computing, Internet of Things (IoT), and augmented reality (AR) opens up new avenues for real-time and context-aware applications. From smart cities and autonomous vehicles to augmented reality experiences, CNN-based object detection has the potential to revolutionize how we interact with and interpret the visual world around us.

Looking ahead, continued research and development in CNN-based object detection will enable businesses to tackle increasingly complex and diverse visual recognition tasks, driving innovation and unlocking new opportunities for automation, safety, and efficiency. With the relentless advancement of technology, the future of object detection using CNNs promises to reshape industries, redefine workflows, and empower organizations to thrive in an ever-changing digital landscape.

References

- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137-1149.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Tan, M., & Le, Q. V. (2019). EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10777-10786).

GROUP INDIVIDUAL CONTRIBUTION REPORT:

“Object Detection Using CNN”

Sudip Chakrabarty

21053329

Abdulla Al Muhit

21053259

Contribution Breakdown:

1. Sudip Chakrabarty

- Conducted extensive exploratory data analysis (EDA) to understand data patterns and distributions.
- Researched and implemented advanced feature selection methods to improve model efficiency.
- Contributed significantly to coding tasks, including model implementation and optimization.

2. Abdulla Al Muhit

- Led the data collection and preprocessing phase.
- Assisted in model tuning and validation strategies.
- Developed feature engineering techniques to enhance model performance.

Overall Team Achievements:

- Collaborated effectively to integrate the model into practical applications, demonstrating its value in real-world scenarios.
- Contributed to advancements in computer vision technology, paving the way for enhanced safety, efficiency, and productivity across various industries.
- Successfully developed an Object Detection System using Convolutional Neural Networks (CNNs) with high accuracy.
- Explored a variety of techniques to improve model performance and efficiency.

Full Signature of Supervisor:

.....

Full signature of the students:

.....

.....

.....

.....

.....