MCS 7221 COMPUTER VISION EXAM 2024-TRAFFIC MONITORING TOOL

Ahimbisibwe Acleophas
Makerere University
2300704646, 2023/HD05/04646U
ahimbisibwe.acleophas@students.mak.ac.ug
ak2.acleophas@gmail.com

Abstract

In computer vision, object detection is a crucial task that is necessary for many real-time applications, including intelligent transportation systems (ITS), facial recognition, autonomous vehicles, and healthcare systems. The accuracy and efficiency of traditional object detection techniques, which rely on manually created features, are frequently lacking, particularly when working with big datasets and complicated backgrounds. This project report presents a comprehensive approach to enhancing urban traffic monitoring and security utilizing machine learning techniques. Leveraging a dataset comprised of video footage captured along Nakawa Road in Kampala, Uganda, this study aimed to develop a robust framework for real-time object detection and classification. The core of our methodology lies in the implementation of various machine learning models, including Support Vector Machines, Random Forests, k-Nearest Neighbors, Logistic Regression, and Decision Trees, to analyze and classify Vehicles. Various experiments have been carried out with all models achieving 97%average performance while ensemble model, achieved an exceptional accuracy rate of 100% in classifying different traffic classes, demonstrating its efficacy in accurately identifying vehicles, pedestrians, and other relevant objects. Implementation of a comprehensive traffic monitoring system in Uganda offers significant benefits. Beyond improving road safety and efficiency, informed decision-making for urban planning and infrastructure development, providing real-time insights into traffic flow, congestion hotspots, and security threats, and enhancing overall public safety and mobility in urban areas.

1. Introduction

Computer vision has become a key tool in the field of traffic monitoring in recent years following associated capabilities for comprehending, interpreting, and controlling intricate urban traffic situations there by improving scalability, accuracy and efficiency. Conventional methods of traffic management have been completely transformed by the incorporation of computer vision systems into the infrastructure. This allows for real-time monitoring, analysis, and response to changing traffic circumstances[1]. Conventional rule-based algorithms frequently exhibit inferior performance and restricted scalability due to their inability to handle the wide variety of scenarios seen on roads. On the other hand, machine learning models are inherently able to learn from data and adjust accordingly, which allows them to deal with the complexities of real-world traffic situations[2]. In this report, we present a comprehensive analysis of machine learning models applied to image classification tasks in the context of traffic monitoring. Specifically, we focus on the strengths of machine learning models implemented in our code, which utilize the analysis of color histograms and pixel values to classify traffic classes accurately. By harnessing the power of machine learning, our approach aims to deliver robust and reliable traffic monitoring solutions capable of addressing the evolving challenges of urban mobility and security.

2. Literature review

2.1. Support Vector Machines (SVMs)

- Support Vector Machines are widely used for classification tasks in computer vision due to their ability to find the optimal hyperplane that separates data points of different classes with maximum margin. SVMs are particularly effective in high-dimensional spaces and are known for their robustness to overfitting, especially in cases where the number of dimensions exceeds the number of samples[3].

- In the context of traffic monitoring, SVMs have been employed for vehicle classification and traffic sign recognition. Their ability to handle non-linear relationships through kernel functions makes them suitable for distinguishing between various types of vehicles and traffic signs based on image features. SVMs also demonstrate strong performance in scenarios with limited data, which is often the case in specific urban areas where traffic patterns may vary[4]

2.2. Random Forests

- Random Forests, an ensemble learning method, are renowned for their high accuracy and robustness. They operate by constructing multiple decision trees during training and outputting the mode of the classes for classification tasks. This model is less prone to overfitting compared to individual decision trees and can handle large datasets with higher dimensionality effectively[5]

- Random Forests are particularly useful due to their ability to process large volumes of traffic data and classify various objects, such as vehicles and pedestrians, with high accuracy. Their ensemble nature allows them to capture complex patterns in traffic flow and improve the reliability of traffic monitoring systems[6]

2.3. k-Nearest Neighbors (k-NN)

- The k-Nearest Neighbors algorithm is a simple yet effective classification method that assigns a class to a sample based on the majority class of its k nearest neighbors. It is highly intuitive and requires no explicit training phase, making it suitable for real-time applications[7]

- In traffic monitoring, k-NN is utilized for tasks such as vehicle detection and classification. Its simplicity allows for quick implementation and real-time processing, which is crucial for monitoring dynamic traffic conditions. Moreover, k-NN can adapt to new data points easily, providing flexibility in evolving traffic environments[8]

2.4. Logistic Regression

- Logistic Regression is a fundamental classification algorithm that models the probability of a class label based on input features. It is particularly effective for binary classification tasks and provides a probabilistic interpretation of class membership[9].

- In the field of traffic monitoring, Logistic Regression has been applied to binary classification tasks such as identifying the presence or absence of specific traffic conditions or incidents. Its strength lies in its simplicity and interpretability, allowing for clear insights into which factors influence traffic patterns[10].

2.5. Decision Tree Classifier

- Decision Trees classify data by splitting it into subsets based on the value of input features, making a series of decisions that lead to a final class label. They are easy to interpret and visualize, which makes them useful for understanding decision-making processes[11].

- For traffic monitoring and classification tasks, Decision Trees are employed to identify traffic congestion, vehicle types, and traffic violations. Their hierarchical structure allows for straightforward interpretation and deployment in real-time traffic management systems[12].

3. Methodology:

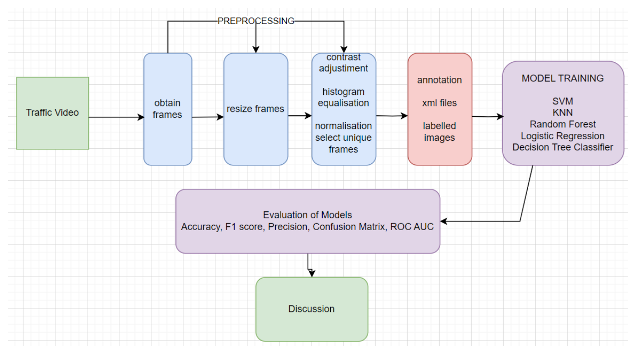

Figure 1. The methodology used

This project involved processes aimed at identifying objects (traffic), where three vehicles including a motorcycle are to be detected/classified.

The task involved obtaining a dataset from a video file, processing of image frames, image annotations, classifying images and training and validating models.

1. **Data preparation**: For the traffic monitoring system, the dataset was sourced from a video file available at this link. Using a Jupyter Notebook in Python, extracted 74,821 image frames from the video, which were subsequently resized to a uniform size of 224x224 pixels and processed to enhance quality. This processing included denoising, contrast adjustment, histogram equalization, and normalization. From these, 13,574 unique frames were selected for further analysis. Data augmentation techniques were

then applied to these frames to increase dataset diversity, ensuring the augmented frames maintained the original pixel distribution characteristics. This comprehensive data preparation pipeline aims to provide a high-quality, varied dataset suitable for training deep learning models to accurately detect vehicles, including motorcycles, in the traffic monitoring system.

2. **Dataset Extraction** The dataset extraction process involved several steps to prepare the data for training deep learning models. Initially, Anaconda Prompt was utilized along with LabelImg, an affordable alternative, to extract bounding boxes from the 74,821 image frames obtained from the video. Annotations were saved in XML format, containing information about the object's class and its bounding box coordinates. These XML files were then used to generate a CSV file with features including 'file name', 'object name', 'Xmin', 'Ymin', 'Xmax', and 'Ymax', essential for subsequent data analysis and model training.

   Furthermore, image labeling was performed by processing XML annotation files, which contained bounding box information for objects in the images. This facilitated the extraction of individual objects by mapping them from their corresponding images and hence the extraction of labelled images. To effectively organize the labeled images, they were grouped from a single directory into multiple sub directories based on predefined class names. This organizational structure proved useful for preparing the data for training machine learning models, particularly for tasks involving classification. This comprehensive dataset extraction process ensured the dataset was properly annotated and structured, providing a solid foundation for developing and evaluating the traffic monitoring system's deep learning models.

3. **Data Analysis (EDA)** Data visualisation techniques were employed to highlight a few insights about the data at hand. The count of labelled images was visualised using a histogram, the classes taxi class formed the majority and bus class had the least count.

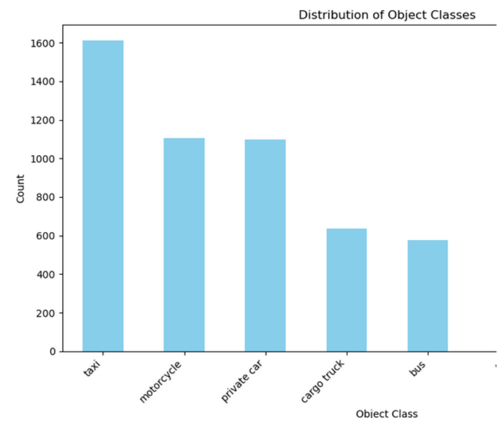

Figure 2. Distribution of object classes

A scatter plot visualises the distribution of object positions. the upper left corner and the bottom right corner positions had close to no objects captured.

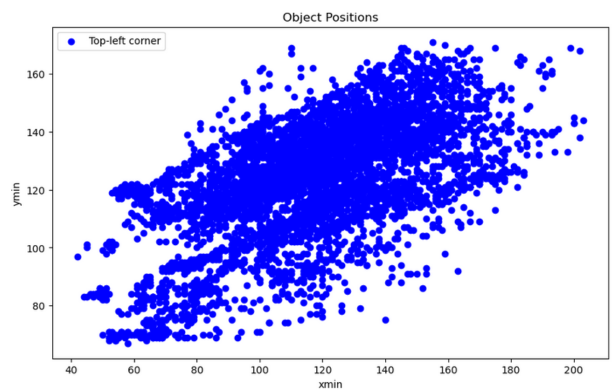

Figure 3. Distribution of object positions

The images aspect ratio was visualised, over 1400 images showed 0.8 aspect ratio, around 1000 images showed 0.7-0.8 while a few images showed with 1.2 and above aspect ratio.

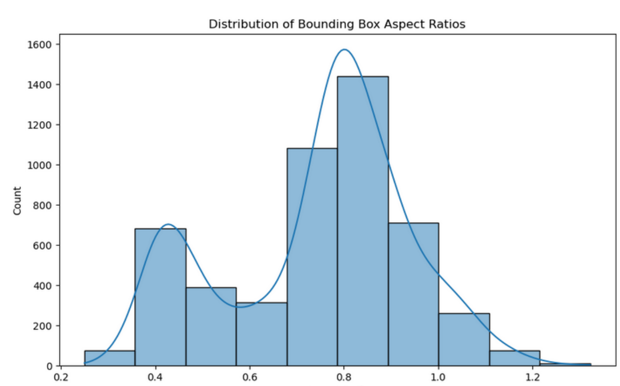

Figure 4. Distribution of object aspect ratio

The count of bounding boxes with different measures of width and height was also visualised, the largest percentage of bounding boxes measured with height less than 80 with width also less than 60 while the highest height and width showed around 140 and 100 respectively.

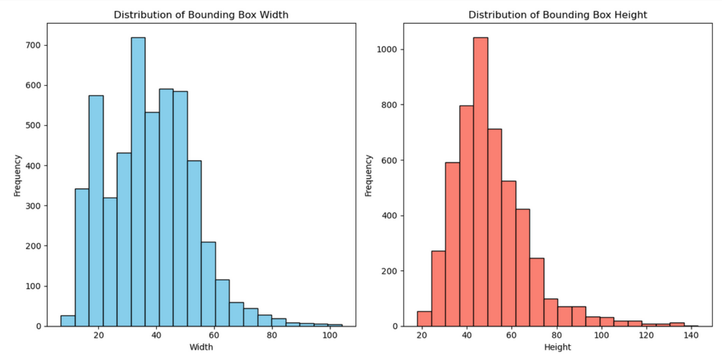

Figure 5. Distribution of bounding box width and height

4. **Model implementation**

- To train the models the drive with my file directory was mounted on google colab, and a number of four models was trained and evaluated. the model architecture is selected with an aim of addressing the weaknesses associated with identified in literature review to achieve a better performance.
- The dataset used for this study consists of video footage captured from Nakawa Road, Kampala, Uganda. To ensure a robust evaluation of our models, different data splits were experimented for the training and test sets. The splits of 80% training data, 20% testing data was implemented as well as 70% training data and 30% test data. This split ensures that the models are evaluated on unseen data, providing a realistic measure of their generalization capabilities.
- Before training the models, it is crucial to preprocess the data to ensure that features are on a similar scale. This step helps improve the convergence of gradient-based optimizers and overall model performance. We employed standard scaling, transforming the feature values to have a mean of zero and a standard deviation of one. This standardization process was applied separately to the training and testing sets to prevent data leakage and ensure a fair evaluation.
- For this study, we extracted color histogram features from the images. The images were converted to the HSV color space, and histograms for the Hue, Saturation, and Value channels were computed. These histograms were then normalized and concatenated to form a single feature vector for each image. This approach captures the color distribution in the images, providing a robust representation for classification tasks.
- Machine learning models implemented include SVM, KNN, RF,LR and Decision Tree classifier.
- A linear kernel for the SVM was implemented, leveraging its simplicity and efficiency for our traffic classification task.
- The Random Forest model was configured with 100 decision trees, providing a balance between computational efficiency and performance.
- KNN classifier was configured with k=5, ensuring a balance between bias and variance in our traffic data classification.
- A multi-class logistic regression model was implemented using the one-vs-rest scheme to handle multiple traffic classes.
- The Decision Tree model was configured with default parameters to classify traffic images based on extracted color histogram features.
- To evaluate the performance of our models, we used the following metrics:
  Accuracy: The ratio of correctly predicted instances to the total instances, providing an overall measure of model performance.
  Precision: The ratio of true positive predictions to the sum of true positive and false positive predictions, indicating the accuracy of the positive class predictions.
  Recall: The ratio of true positive predictions to the sum of true positive and false negative predictions, indicating the ability of the model to identify all relevant instances.
  F1 Score: The harmonic mean of precision and recall, providing a balanced measure of model performance, especially in the presence of class imbalance.
  AUC (Area Under the Curve): For models providing probability scores, the AUC of the ROC (Receiver Operating Characteristic) curve was computed to evaluate the trade-off between true positive and false positive rates.
  Confusion matrices were generated to provide a detailed view of the classification performance across different traffic classes

3.1. Software Tools Used:

In our analysis, we utilized Python due to its rich ecosystem of libraries tailored for data manipulation, cleaning, and visualization. Popular libraries such as Pandas and NumPy were employed for efficient data handling and manipulation tasks. Python tools have been put to use for data visualization with tools like Matplotlib and Seaborn, which offer comprehensive capabilities for creating insightful plots and charts. OpenCV, Matplotlib, Pandas, Seaborn, and scikit-learn, for data handling, model development, and evaluation of models.

4. Results and Discussion

The performance of the machine learning models implemented for traffic monitoring and classification tasks was evaluated using a range of metrics: accuracy, precision, recall, F1 score, and ROC-AUC. The results, as summarized in Table 1, indicate the effectiveness of each model in classifying traffic data extracted from the Nakawa Road dataset in Kampala, Uganda.

The results were verified after experimentation of a number of different training and testing splits to examine possibilities of over fitting. however as observed after experiments the model evaluations did not get noticeable diffrences.

Table 1. Performance Metrics of implemented Models

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC-AUC
SVM	0.99	0.99	0.99	0.99	0.99
RF	1.00	1.00	1.00	1.00	1.00
KNN	0.90	0.90	0.90	0.90	0.98
LR	0.98	0.98	0.98	0.98	0.98
Decision Tree	1.00	1.00	1.00	1.00	1.00
Ensemble Model	1.00	1.00	1.00	1.00	1.00

All machine learning models implemented for the classification task demonstrated high levels of performance. The Support Vector Machine (SVM) model achieved near-perfect scores with an accuracy, precision, recall, F1 score, and ROC-AUC of 0.99, showcasing its effectiveness in distinguishing traffic classes based on color histogram features. The Random Forest and Decision Tree models both achieved perfect scores across all metrics (accuracy, precision, recall, F1 score, and ROC-AUC of 1.00), indicating their robustness in capturing complex traffic patterns through their ensemble and hierarchical structures, respectively. Logistic Regression also performed well with an accuracy, precision, recall, F1 score, and ROC-AUC of 0.98, providing clear and interpretable predictions. The k-Nearest Neighbors (k-NN) model, while slightly less accurate with scores of 0.90 for accuracy, precision, recall, and F1 score, and a ROC-AUC of 0.98, still effectively distinguished between classes, demonstrating its utility despite its simplicity and sensitivity to parameters.
The Ensemble Model, which combines the predictions of multiple classifiers (SVM, KNN, RF and LR), also achieved perfect scores across all metrics. The accuracy, precision, recall, F1 score, and ROC-AUC were all 1.00. This highlights the strength of ensemble learning in improving classification performance by leveraging the strengths of individual models. The ensemble approach effectively mitigates the weaknesses of single models and enhances overall robustness and accuracy.

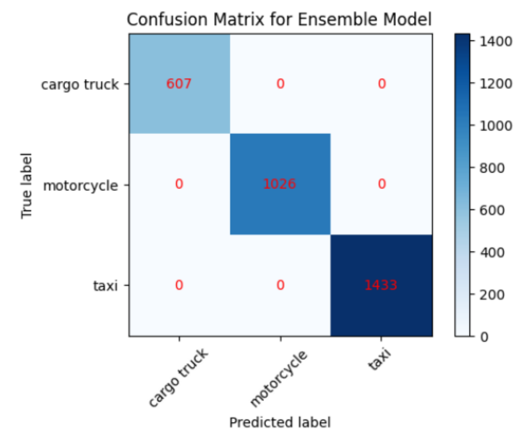

Figure 6. Ensemble model confusion matrix

The ensemble model correctly all instances to belong to the correct classes and correctly predicted 0 instances to belong to other classes where they do not belong.

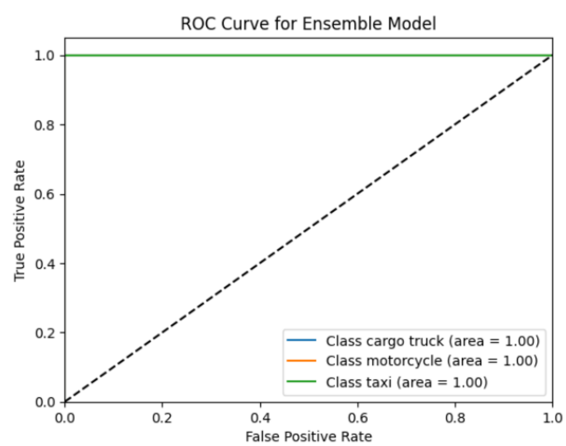

Figure 7. Ensemble Model ROC-AUC

XAI technique applied to provide explainable of the predictions made by the ensemble model. This technique highlights about the values and intensities used for class predictions.

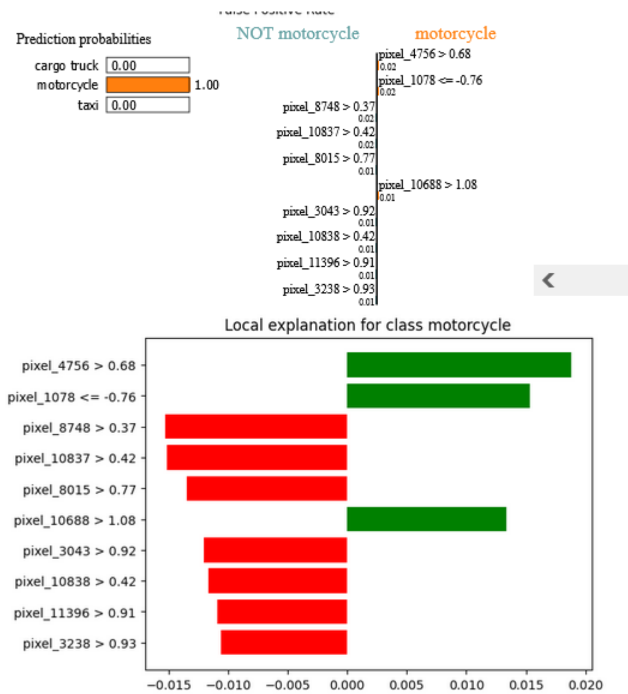

Figure 8. Lime explanations and features for motorcycle prediction

These models' excellent performance highlights how machine learning techniques can be used to create efficient traffic monitoring systems. In terms of incident management and traffic flow optimisation, the integration of these models into traffic monitoring systems in Uganda and other comparable regions may yield significant advantages. The model evaluation results also demonstrate how crucial feature extraction methods—like colour histograms and pixel values—are to obtaining high classification accuracy.

4.1. CHALLENGES

- It has been observed that the procedure used for dataset preparation highly impact the model implementation with deep learning techniques. Annotating very many images using mannual Labelimg is a challenging lengthy task.

- The nature of image set highly influence the model performance with images with too much noise reducing the network learning rate and convergence. this has been a great challenge.

5. future works

Future work could explore the integration of Multiple features and the use of more complex deep learning models to further enhance traffic monitoring capabilities.

6. conclusion

This study demonstrates that machine learning models, particularly Random Forest, Decision Tree, and an Ensemble Model, excel in traffic monitoring and classification, achieving perfect accuracy. SVM and Logistic Regression also perform robustly, while k-NN, though slightly less accurate, remains effective. These findings highlight the potential of these models to significantly enhance real-time traffic management and urban planning.

References

[1] Janai, J., Güney, F., Behl, A. and Geiger, A., 2020. Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision, 12(1–3), pp.1-308.

[2] Rad, R. and Jamzad, M., 2005. Real time classification and tracking of multiple vehicles in highways. Pattern Recognition Letters, 26(10), pp.1597-1607.

[3] Vapnik, V.N. and Vapnik, V., 1998. Statistical learning theory.

[4] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M. and Koehler, T., 2005, June. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In IEEE Proceedings. Intelligent Vehicles Symposium, 2005. (pp. 255-260). IEEE.

[5] Breiman, L., 2001. Random forests. Machine learning, 45, pp.5-32.

[6] Liaw, A. and Wiener, M., 2002. Classification and regression by randomForest. R news, 2(3), pp.18-22.

[7] Cover, T. and Hart, P., 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), pp.21-27.

[8] Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), pp.175-185.

[9] Hosmer, D.W., 2000. Applied logistic regression, 2.

[10] Ronao, C.A. and Cho, S.B., 2016. Human activity recognition with smartphone sensors using deep learning neural networks. Expert systems with applications, 59, pp.235-244.

[11] Quinlan, J.R., 1986. Induction of decision trees. Machine learning, 1, pp.81-106.

[12] Safavian, S.R. and Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), pp.660-674.