ISSN: 0974-8571 Vol.14 No. 1 June, 2022

International Journal of Computational Intelligence in Control

Single Deep Pipelined Intelligent Transport System for Pakistani Vehicles

M. Shoaib Hanif
Department of computer Science, University of the Punjab, Lahore, phdcsf14m508@pucit.edu.pk

Zubair Nawaz
Department of data science, University of the Punjab, Lahore, znawaz@pucit.edu.pk

Muhammad Kamran Malik
Department of information technology, University of the Punjab, Lahore kamran.malik@pucit.edu.pk

Date of Submission: 11th February 2022 Revised: 20th April 2022 Accepted: 15th June 2022

How to Cite: **M. Shoaib Hanif., (2022). Single Deep Pipelined Intelligent Transport System for Pakistani Vehicles.** ***International Journal of Computational Intelligence in Control, 14*(1), .**

Abstract – With the ever-increasing vehicular traffic day by day, we face several traffic management and law enforcement issues. Intelligent Transportation Systems (ITS) are required for monitoring and managing uprising road incidents and crimes to avoid human error and reduce manual effort. Vehicle analytics such as vehicle type, color, manufacture model, number plate recognition, and alert for blacklisted vehicles are integral to the ITS. Likewise, the rest of the world, Pakistan is also facing traffic management and law enforcement challenges. No significant work is done on Pakistan's data and environment. In this regard, we have gathered a large (150,000 images) local dataset of vehicle images labelling vehicle types, colors, and manufacturer models commonly used in Pakistan. Contrary to earlier studies, where individual features are extracted for each characteristic, our approach builds a single feature extractor that is used to detect these characteristics. This approach saves a lot of computations that make the vehicle analysis problem more responsive and real-time. In summary, a single pipeline takes the input video stream, detects the vehicle image, identify the vehicle type, manufacturer model, color, and number plate of the vehicle. The proposed study achieves 96% accuracy for vehicle type classification, 96% for vehicle color classification, 97% for vehicle manufacturer classification, and 98% accuracy with 0.125 CER for vehicle number plate recognition

Index Terms - Automatic number plate recognition (ANPR), Computer vision, Intelligent transportation system (ITS), Machine learning, Vehicle analytics

INTRODUCTION

Vehicular traffic is increasing drastically day by day in large and growing metropolitan areas worldwide, and it will continue to worsen with time. Traffic congestion represents an undoubted risk to the quality of urban life. It causes a reduction in traffic speed, increased journey time and cost, environmental pollution, and security issues as it is impossible to check every passing vehicle manually (Sanjana et al., 2021).

These circumstances in metropolitan cities invigorate an enormous interest in applying intelligent transportation systems for better traffic management and law (Bibri, 2018). Intelligent Transportation Systems (ITS) combines leading-edge information and communication technologies used in transportation and traffic management systems to improve transportation networks' safety, efficiency, and sustainability, reduce traffic congestion, enhance drivers' experiences, and meet transport policy goals. With the conception of smart city transmuting cities into digital societies, making the life of its citizens easy in every facet, Intelligent Transport System has become an indispensable component among all as our urban city yields to the growing pressure on our transportation system. Intelligent transportation systems (ITS) give us an intelligent and smart way to manage our transportation (Balasubramaniam et al., 2021).

This research identified the vehicle based on vehicle type, manufacturer, model, color, and number plate. Every passing vehicle number plate will be checked against the database of blacklist vehicles, e.g. (15 Call, Stolen Vehicle, Cloned & Fake Number Plate, E-Challan Defaulters, on-Custom Paid Vehicle, Suspected Illegal Activity, Involved in Commission of Crime, etc.). If the number plate matches these types of vehicles, an alarm for law enforcement agencies will be generated. Automatic license plate recognition systems typically have three stages: License Plate (LP) detection, character segmentation, and character recognition. The earlier stages require higher accuracy or almost perfection since failing to detect the licensed plate would probably lead to a failure in the following stages either. Many approaches search first for the vehicle and its licensed plate to reduce processing time and eliminate false positives. Although many existing algorithms work fine with knowledgeable systems for capturing images or under controlled conditions (Ritchie et al., 2021), accurately detecting and recognizing vehicles is still an inspirational task in an open environment (Deng et al., 2021). It holds extreme diversities in the patterns like size, distortion, occlusion, weather conditions, and complicated background, which has an immense effect on the system's accuracy (Yuval et al., 2021). Previously, vehicle Detection and Recognition only depended upon a few features of handcrafted images that can work in a specific environment and only focus from the frontal view. Even after many types of research have been done on the recognition and detection of vehicles, colors, and number plates, there were some shortcomings in them (Mufti and Shah, 2021).

Located in South Asia and surrounded by the Arabian Sea, India, and China, Pakistan possesses an ancient and distinctive culture. In Southeast Pakistan, Punjab is the most populous state with the fastest-growing economy. Lahore is the capital city of Punjab and the second-largest city in Pakistan. It has an 11 million population with a total area of 1772 km^2. Recently, urbanization accelerated by rapid economic growth has led to massive migration to the city, and Lahore has up to 17.7 million vehicles passing through its checkpoints every day. These types of vehicles are different, including two-wheeled up to six-wheeled.

For better surveillance, law enforcement, and traffic management, it is necessary to identify and record every passing vehicle based on Type (Motorcycle, Rickshaw, Car, Bus, Van, Ambulance, Tractor, etc.), Manufacturer Model (Suzuki Alto, Toyota Corolla, Honda city, etc.), Color (White, Black, Brown, Grey, Blue, Yellow, Green, Red, etc.) and Number plate. This research identified the vehicle based on vehicle type, manufacturer, model, color, and number plate.

LITERATURE REVIEW

Currently, vehicle analysis is done on its features separately, i.e., by detecting and classifying it into its categories, recognizing its colors, and reading letters and numbers from its number plate. Many researchers have studied and worked on these problems using machine learning and deep learning (Ahmad et al., 2021; Saxe et al., 2021).

Deep Learning Methods: Singh et al. (2021) experimented with analyzing noise produced by different vehicles on the road to tackle its adverse effects on human health. They performed vehicle classification and counting on four other vehicles and noise analysis. The model achieved accuracy between 1 to 0.95 with a minimum error between 3.3 to -15.5. Convolutional Neural Network (CNN) became ubiquitous in 2012 after beating image recognition with a high percentage. In a typical CNN, a sliding window, typically the size of 3x3, is moved on the image grid because the computer sees an image as a grid of numbers. They are estimated by multiplying various numbers in a smaller grid, now called a window. We have a much smaller representation of the image, which humans may not understand; however, computers can still recognize it. And this is flattened from a window to an array of numbers and fed to a simple feed-forward network, a simple artificial neural network. This FFN has one output layer and one layer to connect array output from previous CNN layers. The output layer of FFN identifies a class of images. There were a lot of full-fledged CNN networks anticipated for the classification and the feature extraction by the researcher. Some of the popular CNN networks are the VGG-19 (GhasemiDarehnaei et al., 2021) Network, Google Net (Yang et al., 2021) Network, and Mobile Net (Srinivasu et al., 2021). Fast R-CNN was proposed to give the image instead of 2000 regions to a dense or fully connected network, extracting the proposed regions for running through a classification algorithm (Sun et al., 2021). This reduced the time to identify the location and recognize images compared to RCNN. The feature map extracted from images is bounded in the classification model's region of interest (RoI). The last fully connected layer has a SoftMax function to classify images. The RoI was also offset, and loss is minimized for that as well while detecting an object. Even though Fast RCNN was faster than the original RCNN, this could have been improved and was so in the following algorithm by a different author than RCNN and Fast RCNN. This paper proposed extracting a feature map from an image and then proposing regions around them instead of regressing them by reducing loss, as seen in Fast RCNN. These new proposals were way fewer than regions submitted by RCNN, and this technique made Faster RCNN an even better approach for object detection. The Faster RCNN is not a selective region-based approach; that is why it is faster than the above two algorithms (Yi et al., 2021). Another difference between this algorithm and the above is that it uses a separate network for proposing regions instead of other algorithms.

Color Recognition: One of the significant parts of an Intelligent Traffic System (ITS) is vehicle color information. In this field, some researchers have proposed various solutions for color recognition for object detection of vehicles. The color recognition timeline is divided into two eras where; the first era is handcraft feature extraction coupled with classifiers, which focused on carefully designing various elements of vehicle colors. For example, color histogram (Vallese et al., 2021) and color moment (Sullivan, 2021). and color correlogram (Bhardwaj et al., 2021), and then these extracted features were used to train the classifiers such as SVMs, KNN, and ANNs. Some intelligent algorithms were proposed as follows. Chen et al. (2014) proposed a BoW model based on extracted features using color histogram and spatial information using SPM (Spatial Pyramid Matching) and FC (Feature Context). Then a vector established from Bag of Word (BoW) was run through linear SVM (Support Vector Machine). Wu et al. (2018) also proposed color features such as Hue and Saturation (H and S) in the color space of HSV. The color histogram constructed a two-dimensional vector and then used an SVM to classify it. Meenu et al. (2021) used a completely different approach for vehicle color recognition and classification, and his method was to use 2D histogram features and run an SVM classifier on them. A 2D histogram extracted hue and saturation (HS) from HSV color space. The average accuracy of 94.92% was achieved through this method. His dataset included 500 vehicles parked outdoors and five color classes containing white, black, red, blue, and yellow. Finally, Yu et al. (2021) proposed an approach that recognizes colors and classifies them using the similarity method. His system used a grid kernel that runs on HS of HSV color space, and he used the same dataset as Yu et al. (2021). And instead of observing the accuracy of their model, they used percussion and recall metrics to evaluate their model. Since their recall and precision were very close to 100%, it can be said that the model has achieved high accuracy. According to Nicolas et al. (2021), the State University of Korea proposed a vehicle retrieval system that uses natural language to make traffic management easy. The proposed system wrote sentences about a particular vehicle based on its attributes such as its color, type, position, and movement to distinguish it from other vehicles present on the road and achieved great accuracy in this regard

Number Plate Recognition: Another essential part of vehicle identification is number plate recognition. There have been many developments in this particular area as well. Pilli et al. (2021) and Saidani and Touati (2021) explained a deep learning method that recognized the vehicle by the registration number on its number plate using the Turkish dataset made by researchers. Several images were processed with filters like median blur smoothing, Adaptive Gaussian thresholding, and morphological transformations. The resulting images were fed to the CNN model for training. The outcome of the CNN model was sent to the LSTM network, which then was processed by the decryption algorithm. Ahmed and Ahmed (2021) created an end-to-end machine learning model for automatic vehicle identification through registration numbers on number plates. Feed from an Infrared camera was obtained and preprocessed by contrast enhancement and noise reduction as pre-processing steps. Then the number plate was localized, and a region of interest (RoI) was obtained. The next step was to use contour tracing to get the enhanced features of the image. Then Canny's edge detection was applied to find out the edges of the characters in the number plate. Segmentation was used to obtain different characters. In the end, patterns were marched through an ANN classifier for individual character recognition. Mufti and Shah (2021) conducted a survey that explained how Automatic Number Plate Recognition (ANPR) was done through various methodologies. Seventy-eight papers were evaluated based on the accuracy achieved by researchers in those papers. ANPR followed the following steps for complete recognition of number plates, and these are as follows: vehicle Image capture, number plate detection, Character segmentation, and Character recognition. Heuristics for a simple method for number plate detection are factors such as plate size, plate location, plate background, and screw. According to the survey, the best algorithm for number plate detection was Canny's edge detection. Character segmentation can be implemented using image binarization, CCA (Connected Component Analysis), and vertical and horizontal projection, producing better results. In the end, character recognition was done by Artificial Neural Networks, template matching, or Optical Character Recognition (OCR) techniques. Silvano et al. (2021) introduced a plate recognition system that used a deep learning approach. This was an OCR system with a customized dataset. The dataset was obtained from some online images, and data was augmented by adding noise and backgrounds to those images. For background, the SUN database and Stanford database were used. An object detection framework was used for number plate detection YOLO (You Only Look Once). Goel et al. (2021) used Convolutional Neural Network (CNN) for number plate detection and recognition. Its name had two parts; detection and recognition; the first part took the image frame, detected the number plate region, and then segmented it to recognize each character written in the Bengali Language with less resolution using a convolutional neural network

Vehicle Manufacturer-Model Classification: To classify the vehicle's maker and type, von Ziegler et al. (2021) proposed an analysis of previous models and research to decrease the time required for large datasets annotating large datasets. For this purpose, an online dataset from Turkey was used in a deep learning pipeline model. The model consisted of a combination of Single Shot Multiplexer Detector and Convolutional Neural Network; the combination increased the accuracy by 4% than using a single model. The model identified unlicensed vehicles by their model and manufacturer using their number plates. Khatri et al. (2021) classified the vehicle's model and manufacturer and proposed an intelligent traffic management system. The base network for this purpose consisted of Google Net, Caffe Net, and VGG-16. The pipeline of these models took the whole image as an input to detect if the vehicle was present or not and later classified it according to its manufacturer.

DATASET

Deep learning algorithms are data hungry, and the result depends on the quality of data collected. For the training of our deep learning-based model, it is required to get data of all vehicle types in different times, environments, weather, and lighting conditions. For data collection, we used cameras mounted at check posts and entry exits of Lahore city. The height of the cameras is between three to five meters mounted straight on the road. For data collection, we take the recoded videos of different cameras at different times (morning, noon, afternoon, night), locations, and angles with different weather conditions (fog, rain, smog, etc.). It was a very tedious task to get videos that contained data of all vehicle types, color, manufacturer models, and all licensed formats of number plates used in Pakistan.

After collecting the videos, we need to convert them into images to make them ready for annotation. We used “Free Video to JPG Converted v 5.0.101 build 201” to convert video to images. With the help of this software, we extracted images from the videos with every 5 seconds delay. After converting videos into images, the cleaning process starts in which we remove the duplicate images and only select images with pictures of vehicles. The cleaning process was done manually in which each image is reviewed by opening and deleting if it is a duplicate or does not have a vehicle in the image.

Dataset Annotation

After collecting videos, converting videos to images and cleaning the images, we started the process of data annotation. For our problem, it was necessary to assign multiple labels such as vehicle type, manufacturer model, color to the selected vehicle and then select the number plate of the same vehicle. For this purpose, we developed our custom multilabel data annotation software.

According to our best knowledge, no open source image annotation tools provide multilabel image annotation functionality. We explored some open source softwares like LabelImg for multiple label annotation but did not find any software with a user-friendly interface for the multilabel annotation. We need to select the whole bounding box for one label and then select this bounding box again on the same vehicle to give another label, and this creates ambiguity that one object belongs to more than one class. And while training, this information can confuse the model’s learning. Hence it was not easy to annotate the collected datasets using these softwares, and the use of these annotation softwares can increase the human resources for the annotation process. To overcome this issue, we developed our custom annotation software and annotation guideline to annotate the dataset. Using our multilabel annotation software, we saved upto 50% person-hours in the data annotation process. Five annotators worked 8 hours per day and take approximately 5 months to complete annotations of 150,000 images with 4 labels per vehicle.

Custom Data Annotation Software

For the annotation of vehicle images, the .Net Framework based desktop application is developed in Visual Studio 2019 using C# language. After installing the application, we can start the process of data annotation. Suppose we want to annotate a vehicle image; for this, an image is selected from our system by clicking the Open File button and then drawing a bounding box of the vehicle and number plate using the mouse cursor. After drawing these two bounding boxes, values of x, y, height, and width are added to the screen automatically. The next step is selecting the vehicle type, manufacturer model, color, and number plate value. Detailed annotation guidelines of this process are given next section. An example of this whole scenario is shown in Figure 1.

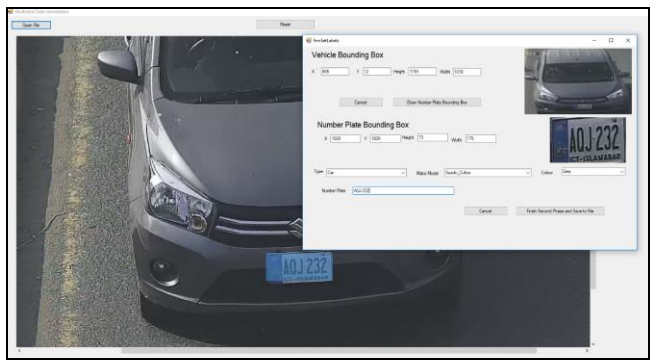

FIGURE 1: MULTILABEL DATA ANNOTATION SOFTWARE.

The output of this custom annotation tool is in the form of a JSON file which includes the information of all the objects annotated through the software. An example of the output is shown in Figure 2. During the training process, we provide the images with corresponding JSON files.

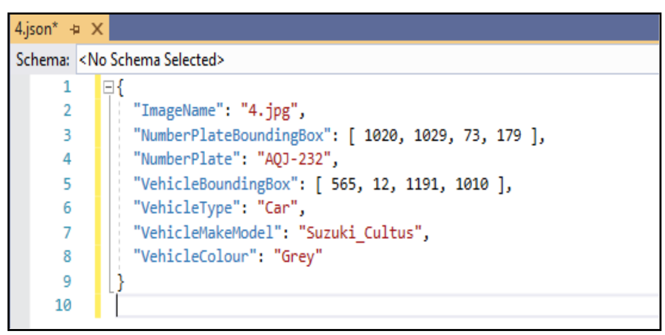

FIGURE 1: MULTILABEL DATA ANNOTATION JSON FILE FORMAT.

Annotation Guidelines

Data annotations are needed to prepare the dataset for training deep learning models. All the data must be annotated correctly as the results depend on the annotated data's correctness. To achieve this, we give the training in multilabel annotation software and also prepared the following annotation guidelines for our data annotators to ensure the correctness of our vehicle dataset

1. Install the custom-developed multilabel annotation using the .exe file
2. Open multilabel annotation software by double clicking the ICON of the Application.
3. Click the Open File button, select the images folder path, and select the image you want to annotate.
4. After selecting an image, it will open in the image view. Make sure there is a complete vehicle in the image. If there is no vehicle or broken image of the vehicle, then discard the image, but if the image has a good vehicle image, then continue the process of annotating
5. First, carefully draw the bounding box around the vehicle. The selected vehicle image will be viewed in another window. If the selected image is incorrect, click the cancel button and redraw the image.
6. After drawing the vehicle bounding box, we labelled the licensed number plate. Click the button to draw the number plate bounding box. Now you can draw the bounding box around the number plate by selecting the number plate using the mouse cursor.
7. After drawing both the bounding boxes, it is needed to give the labels to the selected vehicle and enter the number plate of the vehicle
8. Select the vehicle type (Car, bus, van, etc.) from the Type dropdown
9. Select the manufacturer model (Honda City, Toyota Corolla, etc.) from the dropdown.
10. Select the color of the vehicle from the color dropdown
11. Enter the number plate text of the selected number plate in the textbox.
12. After entering all details, make sure all the entered information is correct. If you doubt any vehicle type, color, or manufacturer model identification, discuss it with the team, especially the seniors.
13. Click on "Finish and Save to File" button to save this information in the JSON file.

Collecting the dataset was challenging as more than 30 licensed number plate format vehicles are used in Lahore, and they have different colors, plate sizes, font sizes, and territorial logos. All provinces and territories in Pakistan have different shapes and logos on the licensed number plates. The plate sizes of motorcycles, front and rear, Rickshaw, and other vehicles are also different in size and character spacing, as shown in table 7. That's why I needed to collect a huge amount of number plate data for a better prediction of the number plate. We collected more than 100,000 vehicles with licensed number plate data containing data from all Pakistan territories and provinces. Licensed format samples of different number plates in figures 3, 4.

FIGURE 3: CAR'S NEW NUMBER PLATE FORMAT IN PUNJAB.

FIGURE 4: NUMBER PLATE FORMATS USED IN DIFFERENT TERRITORIES OF PAKISTAN.

In our dataset, we gathered all types of vehicles most commonly used in Pakistan. We collected images of 10 different vehicle types: Motorcycle, Rickshaw, Car, Bus, Van, Truck, Ambulance, Tractor, Qingqi, and Loader. Collecting image datasets of all types of Vehicles was a very tedious task, particularly images of the ambulance, tractor, truck and loader. Few sample images of different vehicle types are shown in figures 5, 6, 7 and 8.

FIGURE 2: SAMPLE IMAGE OF CAR DATASET.

FIGURE 6: SAMPLE IMAGES FROM MOTORCYCLE DATASET.

FIGURE 7: SAMPLE IMAGES FROM RIKSHAW'S DATASET.

FIGURE 3: SAMPLE IMAGES FROM QINGQI DATASET.

In our study, we selected 10 classes of colors such as White, Black, Brown, Gray, Blue, Yellow, Green, Red, Silver, and Others.). For the color date set, we selected 6 vehicle types (Car, Bus, Van, Rikshaw, truck, ambulance) as color detection of the other four vehicle types (Motorcycle, Qingqi, Loader, Tractor) is not possible because of their body structure and multiple colors on the body. Data collection of the different color vehicles was very difficult, particularly green, red, yellow, and brown, as these colors are not commonly used in Pakistan. Figure 9 shows a sample of the color dataset.

FIGURE 9: SAMPLE IMAGE OF COLOUR DATASET.

Manufacturer model dataset, we divided cars into 41 classes (Suzuki WagonR, Honda City, Toyota Corolla, Kia Sportage etc.) That are most commonly used in Pakistan.

METHODOLOGY

In this study, the backbone architecture responsible for detecting vehicles in the early stage consisted of an optimized yolo with 50 layers and millions of parameters. Backbone architecture, as input, received the frames from the video stream. The video stream may be a live stream or any recorded video. The image is resized to 640X640 to feed into the vehicle detection model, which produces the bounding boxes of the vehicle. The result may contain many false positive bounding boxes. To avoid this, non-max suppression is applied, which works as a filter to the false bounding box and chooses the highest prediction probability bounding box.

After NMS, we applied an object tracking algorithm named CSRT (Channel and Spatial Reliability Tracking) tracker to track the vehicles. It helps to track the vehicle and gives a unique number to every new upcoming vehicle.

After detecting the vehicle from the backbone network, several parallel networks were also attached to it. These networks took the detected object as an input and then classified different vehicle attributes like vehicle type, color, manufacturer model, number plate detection, and recognition. The details of each network are described below, and the architecture diagram of the complete system is given below in Figure 10.

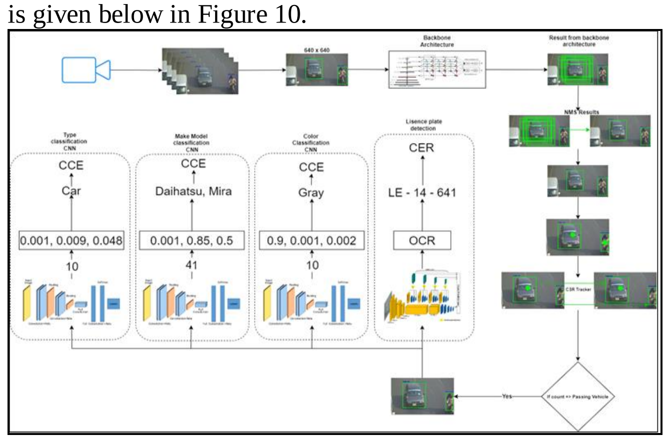

FIGURE 10: ARCHITECTURE DIAGRAM OF THE COMPLETE SYSTEM.

Vehicle color classification architecture is based on our suggested approach, extracting the feature from the fed input. The initial segment was the core architecture, which was a CNN architecture. The output from the previous network (backbone network) became the input of this network. The convolutional layers extract the feature maps, and then the fully connected layer in this classification network classifies the color accordingly from the 10 classes. The methodology diagram of color classification is shown in figure 11

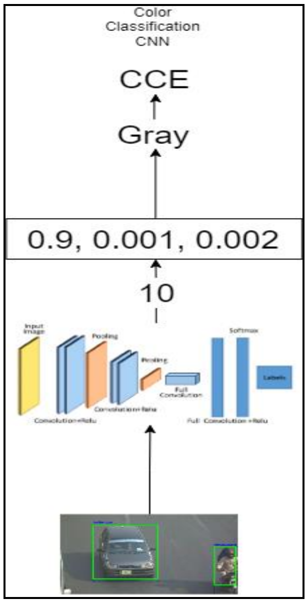

FIGURE 11: METHODOLOGY OF COLOR CLASSIFICATION.

The other parallel models running beside the main backbone architecture are vehicle manufacturer model and vehicle type classification. Same as in the case of color classification, the detected vehicle object was fed to this model. CNN architecture in these models is responsible for vehicle type and manufacturer model classification. Its flow diagram can be seen in figure 12.

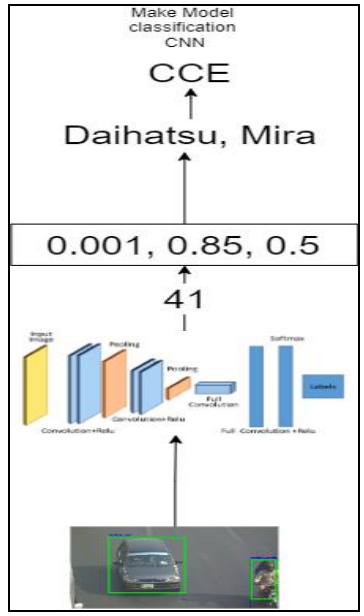

FIGURE 12: METHODOLOGY OF MANUFACTURER MODEL.

Table 1 explain the convolutional neural network used for the above detection and classification tasks. The first convolution layer is applied with stride 4, which has 96 filters of size 11X11. ReLU is the activation function utilized in this layer. The final feature map measures 55X55X96 pixels. The first Max pooling layer, 3X3 in size and stride 1, is next. The resulting feature map, measuring 53X53X96, is then obtained.

The second convolution process is then used. This time, the filter size is 2X2. There are 256 filters, and the stride is 2. Once more, ReLU (Rectified Linear Units) is the activation function used. The output size that we now receive is 26X26X256. Once more, we used stride 1 to apply a max-pooling layer 3X3 in size. The final feature map has the dimensions 24X24X256.

The third convolution operation is now applied, using 384 filters of size 3X3 stride 1. Once more, ReLU is employed as the activation function. The final feature map has the dimensions 22x22x384. Again, after the max pooling with the same parameters, we get a 20x20x384 feature map. As we delve further into the architecture, it is thus extracting more functionality. Additionally, the filter size is lessening, indicating that the original filter was larger, and as we move forward, the filter size is reducing. This causes a reduction in the form of the feature map.

The first and second layers are fully connected and have a ReLU activation function. The output layer with Softmax has 10 for color and vehicle type and 41 outputs for the manufacturer model, as shown in figure 13.

TABLE 1: CNN LAYERS ARCHITECTURE OF THE VEHICLE TYPE AND COLOR.

Layer Type	No. of filters	Feature Map Size	Kernel Size	No. of stride	Activation function
2D conv. Layer	96	227x227x3	(11,11)	(4,4)	ReLU
2D max pooling		55x55x96	(3,3)	(1,1)	
Batch normalization		53x53x96			

2D convolutional Layer	256	53x53x96	(3,3)	(2,2)	ReLU
2D max pooling		26x26x256	(3,3)	(1,1)	
Batch normalization		24x24x256			
2D convolutional Layer	384	24x24x256	(3,3)	(1,1)	ReLU
2D max pooling		22x22x384	(3,3)	(1,1)	
Batch normalization		20x20x384			
Flatten layer		20x20x384			
FC – 1	4096	153,600			ReLU
FC – 2	1024	4096			ReLU
Dropout					
FC – 3	10	1024			Softmax

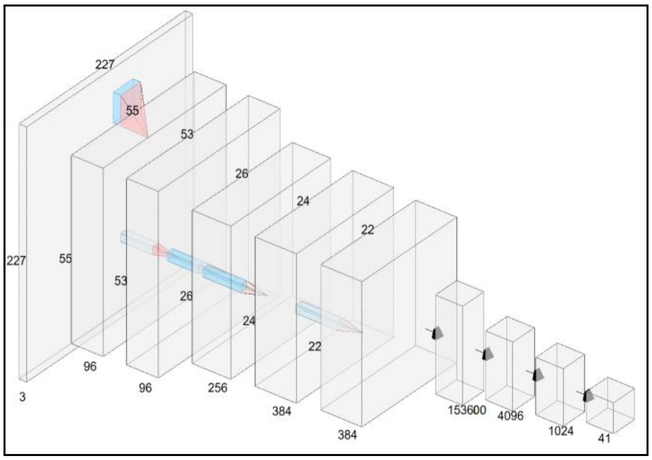

FIGURE13: NETWORK DIAGRAM OF THE MANUFACTURER MODELS.

There is a model for detecting a number plate in the final pipeline and recognizing its number. The details of this model and work are shown in figure 14. first of all, the vehicle image passed to this model detected the number plate region from the overall object.

Pakistani number plates are asymmetrical as each province and territory has distinctive color schemes, shape, plate size, font, and logos. So, it was challenging to detect such irregular number plates. We fine-tune a pre-trained deep learning model for better number plate detection from the vehicle image. The detected vehicle number plate image was in RGB format. We converted the RGB image into a grayscale image. The primary purpose of color conversion was to reduce the number of colors. Erosion and dilation are combined to remove small objects from the image and smooth the boundary of large objects, which is very helpful in removing the noise from the number plate image. The position of the number plate affects the performance of number plate recognition, so it is necessary to correct the skew and rotation of the number plate. Projection Profile Method in python is used to determine skew. The binary image is rotated at various angles to generate the histogram of pixels in each iteration. The skew angle is determined by comparing the maximum difference between peaks. Using this skew angle, rotated the image to correct the skew.

Optical character recognition extracted characters from number plate images. The major problem with character recognition in number plates was the large number of font styles used in the plate. We found that the best way to recognize characters from the image was to train a neural network to recognize the numbers and characters from the cropped image. In this way, the neural network was able to identify features associated with each character and number and used to recognize various fonts.

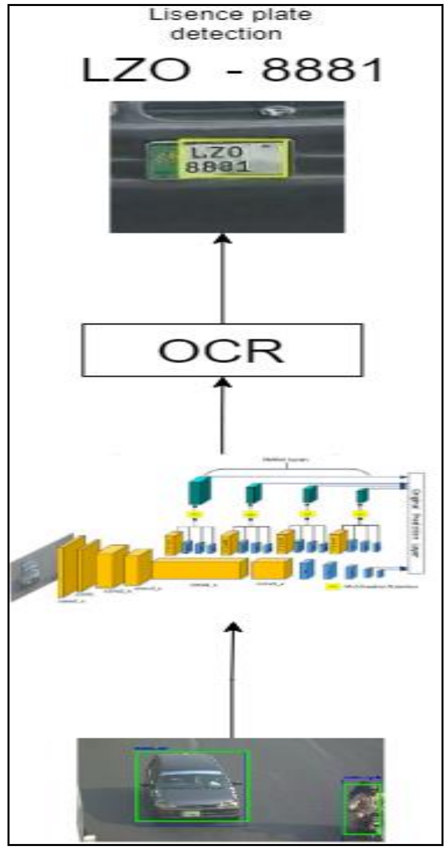

FIGURE 14: METHODOLOGY FOR ANPR

EXPERIMENTAL SETUP

For the training of our proposed model, 80% of the dataset was splitted for the training set and the rest for testing purposes. Dataset was shuffled before splitting, and 20% of data instances were randomly extracted for testing purposes. As the dataset was huge, the number of pictures for the training and testing set was sufficient for the model's training and comparable accuracy. A cluster computer consisting of '3' Tesla V100-PCIE GPUs of 32 GB with 128 GB of RAM capacity was used to train the proposed model. Moreover, a batch size of 64 was used with the momentum value of 0.8 for the 200,000 epochs. Initially, a 0.001 learning rate was used for the first 20,000 iterations, and then a decay rate of 0.1 was used for every 20,000 epochs. The first training phase of the vehicle detection took almost

26 hours. The approach made the SGD fall reasonably, resulting in lower loss values. Libraries and the other details of parameters are shown in the table 2.

TABLE 2: PARAMETERS OF EXPERIMENTAL SETUP.

Sr. No.	Name	Value
1.	GPU	Tesla V100-PCIE *3 (Each 32 GB)
2.	CPU	Dell PowerEdge R740
3.	OS	Linux (Ubuntu 18.04)
4.	Libraries	Pytorch, open cv, Pandas, NumPy, Matplotlib
5.	Batch Size	512
6.	No. of Epochs	200,000
7.	Learning Rate	0.001
8.	Decay Rate	0.1
9.	Dropouts	Dynamic
10.	Loss Function	Categorical cross-entropy
11.	Momentum value	0.9
12.	Training Data	80% of total data
13.	Testing Data	20% of total data

TABLE 31: ACCURACY W.R.T DROPOUT RATE.

Dropout Ratio	Train Accuracy	Test Accuracy
0.0	99.86	83.56
0.2	98.6	93.58
0.3	98.86	97.66
0.4	96.64	95.67

We used different dropout ratios to deal with overfitting. Initially, at a 0% dropout ratio, we achieved 99.86% accuracy on the train set and 83.56% on the test set. There is a large gap between train and test accuracy. So, we experimented with four different dropout ratios and got the best fit on the 0.3 dropout ratio, as shown in table 3.

FINDING AND DISCUSSION

In-depth details of experimental results of the proposed vehicle detection methodology, accuracy of the vehicle type, color, manufacturer model, number plate detection and the character error rate of number plate recognition.

Vehicle Detection: The detection of vehicles is done by the backbone architecture of our model. The graph below depicts the model's mean average precision on different IoU values. as we know, the mAP compares the ground-truth bounding box to the detected box and returns a score as explained in chapter 2. The higher the score of mAP, the more accurate it is in detection. In figure 15, Y-axis represents the IoU value, and the X-axis shows the mAP. The graph line shows with a higher value of IoU, mAP decreases and vice versa. Mean average precision increases as we decrease the IoU threshold, and the model achieved its maximum 0.926 value of mAP when the IoU threshold was 0.4. We selected the IoU threshold value of 0.5, on which we have 0.853 mAP

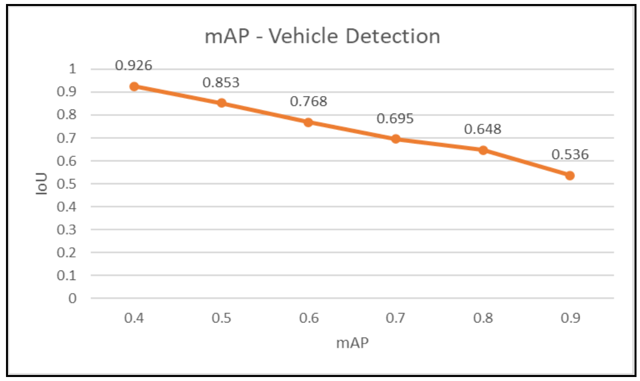

FIGURE15: MEAN AVERAGE PRECISION OF VEHICLE DETECTION.

Vehicle Type Detection: Figure 16 shows the categorical cross-entropy of vehicle type detection. The pattern in the graph depicts that as the number of epochs increases, the value of CCE drops near to zero. The lesser the value of CCE depicts better the performance of the model. Moreover figure 17 demonstrates that the accuracy of each class in the vehicle type detection is greater than 90%, and 96% is the overall accuracy of all vehicle type classes.

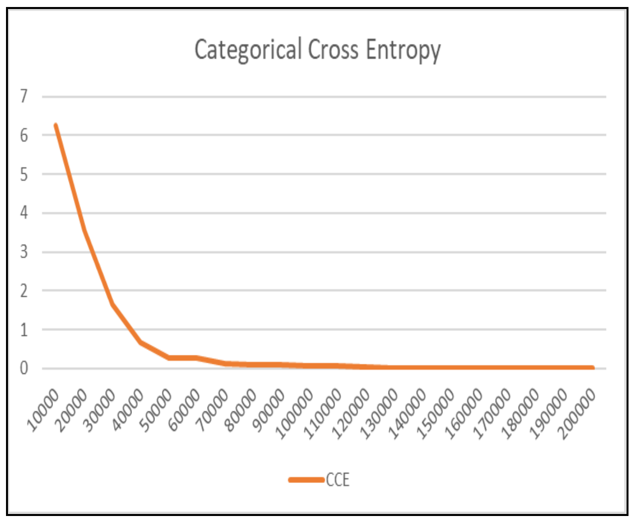

FIGURE 16: CATEGORICAL CROSS ENTROPY OF VEHICLE TYPE.

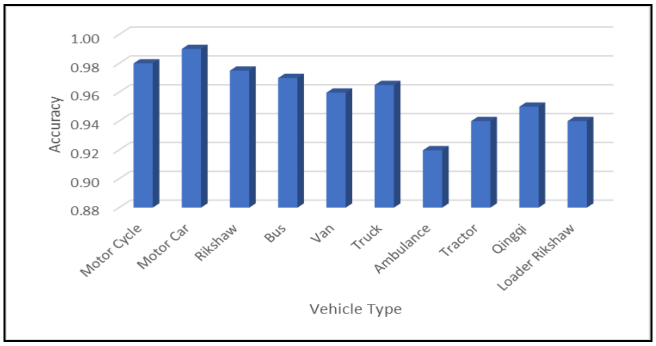

FIGURE 17: BAR CHART OF THE VEHICLE TYPE CLASSIFICATION ACCURACY.

Vehicle Manufacturer Model Detection: On the other hand, for the vehicle manufacturer classification, figure 18 shows that the CCE value decreases as we keep increasing the number of epochs. Vehicle manufacturer model classification accuracy values of each class is shown in figure 19 in which it is observed that the accuracy of each class in the manufacturer model is above 90%, and the average accuracy of all classes is 97%.

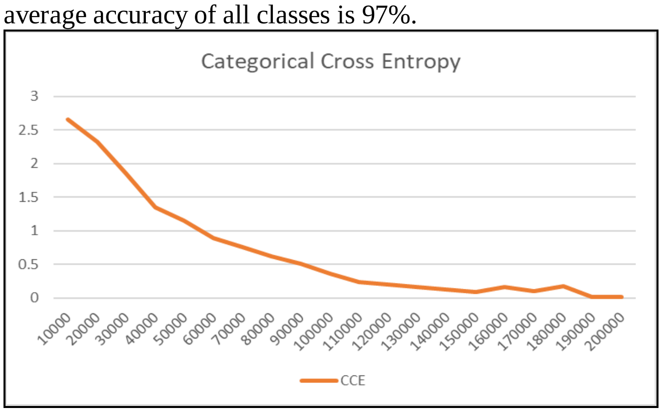

FIGURE 18: CATEGORICAL CROSS ENTROPY OF MANUFACTURER MODEL.

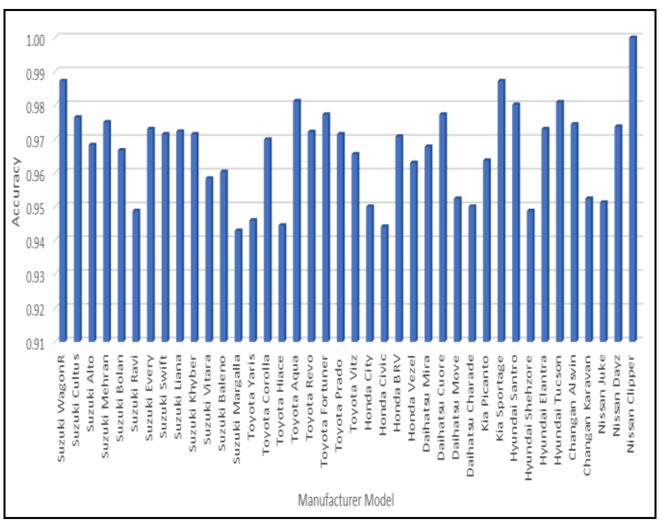

FIGURE19: BAR CHART OF THE VEHICLE MANUFACTURER MODEL DETECTION ACCURACY.

Table 4 shows several recent predominant approaches which are implemented in the area of vehicle manufacturer recognition. The recognition accuracy of our proposed method outperforms other handcrafted feature extraction methods, Implementing an efficient approach by looking at complete body shape rather than the logo because, in Pakistan, many people fabricated the logo in their cars. In addition, to the best of our knowledge, we used 41 classes, the maximum number of classes up until now.

TABLE 4: COMPARATIVE ANALYSIS OF VEHICLE MANUFACTURER CLASSIFICATION MODEL.

Ref	Recognition accuracy	Number of classes	Methodology
Psyllos et al. (2020)	93.7%	10	Merged Feature Matching
Burkhard et al. (2011)	90.33%	8	Fourier Shape Descriptor
Psyllos et al. (2010)	91.12	10	m-SIFT
Llorca et al. (2013)	90%	27	HOG
Huang et al. (2015)	94%	10	Traditional CNN
Huang et al. (2015)	94.21%	10	PCA CNN
This Study	97%	41	Proposed

Vehicle Color Detection: Figures 20, and 21 show the CCE and accuracies pattern for the color recognition of vehicles. The graph depicts that after 1,30,000 epochs, the value drops near to zero. A confusion matrix is also drawn for a better understanding of true positive and false positive in the case of vehicle color classification. Vehicle color classification of all classes is 96 %.

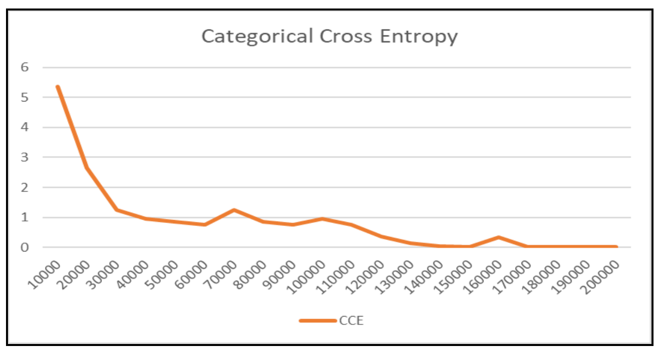

FIGURE 20: CATEGORICAL CROSS ENTROPY OF VEHICLE COLOR.

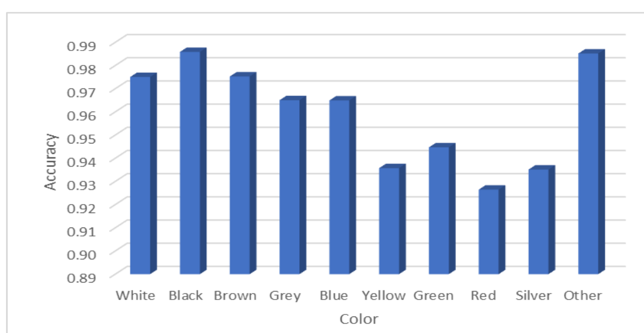

FIGURE 21: BAR CHART OF VEHICLE COLOR DETECTION ACCURACY.

Number Plate detection: figure 22 show the mean average precision value of the number plate detection model concerning the corresponding IoU. And Number plate recognition accuracy is 98.89 with 98.89 as shown in the table 5. Accuracy percentage of vehicle type, manufacturer model and color is also 96%,97% and 96% respectively.

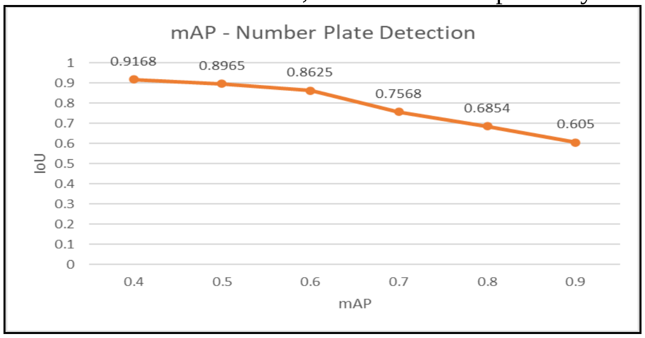

FIGURE 22: MEAN AVERAGE PRECISION OF NUMBER PLATE DETECTION.

TABLE 5: OVERALL ACCURACY OF THE PROPOSED STUDY

Model	**Accuracy**
Vehicle Type Classification	96%
Vehicle Manufacturer Classification	97%
Vehicle Color Classification	96%
Number Plate Recognition	98.89% with 0.125 CER

CONCLUSION

The proposed research work contributes to intelligent transportation systems (ITS) by making the vehicle analytics process autonomous. A robust vehicle analytic system is developed with fully optimized deep learning algorithms. We collected a huge dataset of all vehicle types most commonly used in Pakistan, which also contains data of more than 30 formats of licensed number plates used in different territories and provinces of Pakistan. Labelling this huge dataset is also a mammoth task. In this regard, multilabel annotation software was also developed for annotating the vehicles dataset for multilabel classification, reducing the 50% person-hours of the annotation process. Our proposed novel single pipelined deep learning model outperforms previous methodologies. We achieve 96% accuracy for vehicle type classification, 96% for vehicle color classification, 97% for vehicle manufacturer classification, and 98% accuracy with 0.125 CER for vehicle number plate recognition

In addition, we have developed two dashboards to provide insightful statistics about each passing vehicle for traffic management and law enforcement. Statistics like traffic peak hours, top traffic locations, and the type of vehicles passing from a specific location are very helpful for traffic management. The development of a blacklisted vehicle alarm system and vehicle search based on the number plate, vehicle type, manufacturer model, color, specific time, location, and the specific camera is very helpful for law enforcement agencies to track wanted blacklisted vehicles.

REFERENCES

[1] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), 4137-4150.

[2] Ahmed, A. A., & Ahmed, S. (2021). A Real-Time Car Towing Management System Using ML-Powered Automatic Number Plate Recognition. Algorithms, 14(11), 317.

[3] Balasubramaniam, A., Gul, M. J. J., Menon, V. G., & Paul, A. (2021). Blockchain for intelligent transport system. IETE Technical Review, 38(4), 438-449.

[4] Chen, P., Bai, X., & Liu, W. (2014). Vehicle color recognition on urban road by feature context. IEEE Transactions on Intelligent Transportation Systems, 15(5), 2340-2346.

[5] Deng, J., Hao, Y., Khokhar, M. S., Kumar, R., Cai, J., Kumar, J., & Aftab, M. U. (2021). Trends In-vehicle re-identification past, present, and future: A comprehensive review. Mathematics, 9(24), 3162-3174.

[6] Meenu, M., Kurade, C., Neelapu, B. C., Kalra, S., Ramaswamy, H. S., & Yu, Y. (2021). A concise review on food quality assessment using digital image processing. Trends in Food Science & Technology, 118, 106-124.

[7] Mufti, N., & Shah, S. A. A. (2021). Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors, 21(9), 3028.

[8] Sanjana, S., Shriya, V. R., Vaishnavi, G., & Ashwini, K. (2021). A review of various methodologies used for vehicle classification, helmet detection, and number plate recognition. Evolutionary Intelligence, 14(2), 979-987.

[9] Saxe, A., Nelli, S., & Summerfield, C. (2021). If deep learning is the answer, what is the question?. Nature Reviews Neuroscience, 22(1), 55-67.

[10] Singh, D., Upadhyay, R., Pannu, H. S., & Leray, D. (2021). Development of an adaptive neuro-fuzzy inference system based vehicular traffic noise prediction model. Journal of Ambient Intelligence and Humanized Computing, 12(2), 2685-2701.

[11] Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (2021). Classification of skin disease using deep learning neural networks with MobileNet V2

[12] Wu, H., & Zhao, J. (2018). An intelligent vision-based approach for helmet identification for work safety. Computers in Industry, 100, 267-277.

[13] Yang, N., Zhang, Z., Yang, J., Hong, Z., & Shi, J. (2021). A Convolutional Neural Network of GoogLeNet Applied in Mineral Prospectivity Prediction Based on Multi-source Geoinformation. Natural Resources Research, 30(6), 3905-3923.

[14] Yi, D., Su, J., & Chen, W. H. (2021). Probabilistic faster R-CNN with stochastic region proposing: Towards object detection and recognition in remote sensing imagery. Neurocomputing, 459, 290-301.

[15] Yuval, M., Alonso, I., Eyal, G., Tchernov, D., Loya, Y., Murillo, A. C., & Treibitz, T. (2021). Repeatable semantic reef mapping through photogrammetry and label-augmentation. Remote Sensing, 13(4), 659.

[16] Sullivan, M. (2021). A Kaleidoscope Moment of Color. Brushing, 5(1), 21.Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., ... & Luo, P. (2021). Sparse r-CNN:End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14454-14463).

[17] Saidani, T., & Touati, Y. E. (2021). A vehicle plate recognition system based on deep learning algorithms. Multimedia Tools and Applications, 80(30), 36237-36248.

[18] Vallese, F. D., Belén, F., Messina, P. V., de Araújo Gomes, A., & Pistonesi, M. F. (2021). Exploiting a gradient kinetics and color histogram in a single picture to second-order digital imaging data acquisition with MCR-ALS for the arsenic quantification in water. Sensors and Actuators B: Chemical, 342, 130079.

[19] Von Ziegler, L., Sturman, O., & Bohacek, J. (2021). Big behavior: challenges and opportunities in a new era of deep behavior profiling. Neuropsychopharmacology, 46(1), 33-44.

[20] Bibri, S. E. (2018). The IoT for smart, sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. Sustainable cities and society, 38, 230-253.

[21] Ritchie, K. L., Cartledge, C., Growns, B., Yan, A., Wang, Y., Guo, K., ... & White, D. (2021). Public attitudes towards the use of automated facial recognition technology in criminal justice systems worldwide. PloS one, 16(10), e0258241.

[22] Ghasemi Darehnaei, Z., Rastegar Fatemi, S. M. J., Mirhassani, S. M., & Fouladian, M. (2021). Ensemble Deep Learning Using Faster R-CNN and Genetic Algorithm for Vehicle Detection in UAV Images. IETE Journal of Research, 1-10.

[23] Bhardwaj, S., Pandove, G., & Dahiya, P. K. (2021). An Anatomy of a Hybrid Color Descriptor with a Neural Network Model to Enhance the Retrieval Accuracy of an Image Retrieval System. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(6), 1788-1807.

[24] Yu, L., Xiong, J., Fang, X., Yang, Z., Chen, Y., Lin, X., & Chen, S. (2021). A litchi fruit recognition method in a natural environment using RGB-D images. Biosystems Engineering, 204, 50-63.

[25] Nicolas, C., Kim, J., & Chi, S. (2021). Natural language processing-based characterization of top-down communication in smart cities for enhancing citizen alignment. Sustainable Cities and Society, 66, 102674.

[26] Pilli, B. R., Nagarajan, S., & Devabalan, P. (2021). Detecting the Vehicle's Number Plate in the Video Using Deep Learning Performance. Review of International Geographical Education Online, 11(5), 4315-4324.