sensors

MDPI

Article

A Novel FDLSR-Based Technique for View-Independent Vehicle Make and Model Recognition

Sobia Hayee [1], Fawad Hussain [1,*] and Muhammad Haroon Yousaf [1,2]

[1] Department of Computer Engineering, University of Engineering & Technology, Taxila 47050, Pakistan; sobia_h@yahoo.com (S.H.); haroon.yousaf@uettaxila.edu.pk (M.H.Y.)
[2] SWARM Robotics Lab, National Center of Robotics & Automation (NCRA), Taxila 47050, Pakistan
* Correspondence: fawad.hussain@uettaxila.edu.pk

Abstract: Vehicle make and model recognition (VMMR) is an important aspect of intelligent transportation systems (ITS). In VMMR systems, surveillance cameras capture vehicle images for real-time vehicle detection and recognition. These captured images pose challenges, including shadows, reflections, changes in weather and illumination, occlusions, and perspective distortion. Another significant challenge in VMMR is the multiclass classification. This scenario has two main categories: (a) multiplicity and (b) ambiguity. Multiplicity concerns the issue of different forms among car models manufactured by the same company, while the ambiguity problem arises when multiple models from the same manufacturer have visually similar appearances or when vehicle models of different makes have visually comparable rear/front views. This paper introduces a novel and robust VMMR model that can address the above-mentioned issues with accuracy comparable to state-of-the-art methods. Our proposed hybrid CNN model selects the best descriptive fine-grained features with the help of Fisher Discriminative Least Squares Regression (FDLSR). These features are extracted from a deep CNN model fine-tuned on the fine-grained vehicle datasets Stanford-196 and BoxCars21k. Using ResNet-152 features, our proposed model outperformed the SVM and FC layers in accuracy by 0.5% and 4% on Stanford-196 and 0.4 and 1% on BoxCars21k, respectively. Moreover, this model is well-suited for small-scale fine-grained vehicle datasets.

Keywords: VMMR; multiclass classification; ambiguity; multiplicity; hybrid CNN model; Fisher discriminative least squares regression; small-scale fine-grained vehicle datasets

Citation: Hayee, S.; Hussain, F.; Yousaf, M.H. A Novel FDLSR Based Technique for View-Independent Vehicle Make and Model Recognition. *Sensors* **2023**, *23*, 7920. https://doi.org/10.3390/s23187920

Academic Editors: Salvatore Carta, Silvio Barra and Alessandro Sebastian Podda

Received: 6 July 2023
Revised: 4 September 2023
Accepted: 4 September 2023
Published: 15 September 2023

1. Introduction

Intelligent transportation systems (ITS) are essential components of smart city initiatives in urban areas worldwide to achieve optimal, safe, and sustainable utilization of the available transportation infrastructure and maximum traffic efficiency. Automatic vehicle analysis is significant in any intelligent transportation system involving vehicle attribute recognition, such as vehicle re-identification, vehicle type recognition, and VMMR (vehicle make and model recognition). VMMR has many applications, such as in surveillance for policing and law enforcement, augmenting Automatic License Plate Recognition (ALPR) systems, advanced driver assistance systems (ADAS), electronic toll collection (ETC), self-driving cars, intelligent parking systems, measurement of traffic parameters like vehicle count, speed, and flow, as well as market analysis for car manufacturing companies. Traffic monitoring via VMMR is a critical tool for gathering statistics that aid in designing and planning sustainable and efficient transportation infrastructure.

VMMR is fraught with complications. The first is vehicle detection; the VMMR system should accurately locate vehicles in video images to perform feature extraction and classification. Numerous vehicle variations, such as color, size, and shape, make the problem challenging. Furthermore, under different lighting conditions and viewpoint variations, the visual properties of vehicles also change dramatically. The next task is to

classify the localized image regions into make and model categories. Unfortunately, good classification accuracy can be achieved only after addressing several issues. Firstly, the wide range of makes and models seen in practice can render the number of classes considered rather large, making it a challenging fine-grained classification problem. Next, different models from the same manufacturer (make) frequently share similar shape characteristics and are thus difficult to distinguish. Additionally, the same model can have various facelifts released by the manufacturer over the years, introducing intra-class variation.

For a long time, the performance of computer vision techniques was the primary bottleneck for camera-based traffic monitoring systems. However, the advent of deep learning has fundamentally altered the situation. Researchers must meet several challenges for a wholly integrated AI-based traffic surveillance infrastructure [1]. One of these is accident prevention and vehicle re-identification (reID), which allows a vehicle's route to be calculated for different areas thanks to its unique visual characteristics [2]. VMMR systems come into play in these scenarios, making it possible to detect a vehicle's brand, model, and color from the image. Our proposed approach and a real-time vehicle detection system can address this challenge. Image classification, in particular, has advanced to an entirely new level over the last decade, approaching human-level accuracy in several domains. An essential factor in this transformation is the availability of large-scale datasets. This paper treats the vehicle make and model classification as a fine-grained image classification problem. We use preexisting convolutional neural network (CNN) models for feature extraction and replace the fully connected (FC) layer with a customized classifier based on Fisher discriminative least squares regression (FDLSR) [3]. Our proposed method yields better results than standard transfer learning techniques. The main contributions of our paper are:

- Our technique combines deep features with FDLSR and SVM [4] to yield better classification accuracy.
- We have suggested a robust and efficient view-independent car make and model classification technique.
- Our proposed classifier can be trained on deep fine-grained features at low computational cost and has a short runtime.
- We have applied our proposed classifier to a number of publicly available datasets. The results obtained are comparable to state-of-the-art techniques.

The rest of the paper is arranged in the following manner. Section 2 describes the technical details of the proposed classifier in detail. Section 3 discusses the datasets used for training and testing our classifier, explaining the methodology of our proposed solution to vehicle make and model recognition, and Section 4 reports experimental results on Stanford Cars [5] and a Pakistani on-road car dataset. Finally, Section 5 contains concluding remarks and discusses future research directions.

2. Related Work

Fine-grained image classification aims to classify subcategories of a larger category through fine-grained images [5]. As our goal is fine-grained vehicle classification, we must build a model to identify the most discriminating image features. Therefore, it is vital to detect subtle differences in similar regions. Different subcategories generally have very similar appearances, but the various subcategories are occasionally inconsistent. Many visual disturbances, such as light intensity, occlusion, and blur, seriously reduce the classification accuracy of vehicles.

Vehicle analysis starts with vehicle detection. Once the vehicle is detected, we can classify it based on its class (car, bus, truck), make (Toyota, Honda, Ford), color (white, black, red, gray), or make and model (VMMR). VMMR methods belong to three main categories of fine-grained recognition: attention mechanism [6], high-dimensional feature coding [7,8], and specific characteristics [9]. To detect the primary class of a vehicle, several basic geometric parameters, such as length, width, and height, are approximated [10,11]. Kafai [12] and Grimson [13] processed spatial and edge-based vehicle features with a

Bayesian decision rule for classification. Kumar [14] detected vehicle logos using a Haar cascade classifier and trained an SVM classifier to classify vehicles into four categories. To classify vehicles, some researchers used adaptive background models [15], multiclass SVM-based models [16], and 3D vehicle features and models [17]. Zhang [18] proposed a modified form of the classified vector quantization (CVQ) approach for vehicle type recognition, rejecting low-confidence samples and achieving reliable classification results.

Vehicle type classification is also explored by using vehicle geographical features [19], edge-based features [20], histogram of gradient (HoG) features [21], contour point features [22], curvelet transform features [23], and contourlet transform features [24]. Some studies combined two features, such as wavelet and contourlet features, to improve results [25], as well as PHOG and Gabor features [26]. Dong et al. [27] achieved 83% to 98% accuracy. Liao et al. [9] proposed a strong-supervised DPM (SSDPM) for semantic segmentation of frontal vehicle images. Liao et al. used a novel symmetrical SURF descriptor to improve the discriminative powers of different parts, and the proposed method recognized the brand of each vehicle based on the weights of these parts. Hu and Psyllos [28] focused on brand recognition of a vehicle using discriminative pattern learning, car logo matching, and classification. Loua [29] implemented Lowe's [30] approach of keypoint localization and SIFT features for make and model vehicle recognition. It matched features tie-breakly, but the algorithm proved ineffective in overall vehicle make and model recognition. In addition to SIFT, other features based on edges, gradients, or corners [31], and MPEG-7 descriptors such as edge histograms [32] were also explored for VMMR purposes. In [31], He et al. used Sobel and Canny edge detectors to detect texture, boundaries, and line segment maps of headlamps and license plates. SURF descriptors gained the attention of many researchers due to their fast processing. Siddiqui et al. [33] extracted SURF features from vehicles' front or rear images and embedded them into a bag of sped-up robust features (BoSURF) histograms. Hsieh [34] used a grid division scheme and a combination of the histogram of gradient (HoG) and SURF descriptors to detect the region of interest and extract features from the vehicle. The low accuracy in [20] indicated that locally normalized Harris strengths (LNHS) were inefficient for the VMMR problem. However, the shape-based feature approaches, which extract features from vehicle backlights [35] and rear emblems [36], showed encouraging recognition rates in vehicle make and model recognition.

Model-based vehicle recognition uses the adaptive model [37], the approximate model [38], and the 3D model [39]. In [39], Prokaj and Medioni adopt the model-based approach and project the pose of a 3D CAD vehicle model to a 2D vehicle image to calculate the similarity score. Several classification approaches are proposed to improve VMMR classification. Psyllos et al. [40] classify SIFT features extracted from vehicle images using a probabilistic neural network. Pearce and Pears [20] investigate VMMR classification using the k-nearest neighbor classifier and the naive Bayes classifier. He et al. use neural networks and AdaBoost, SVM, and KNN for classification [31]. Random forest [41] and the nearest neighborhood classification approach [42] are also applied to identify the make and model of vehicles.

Fang et al. [43] proposed using CNNs to classify vehicles. SVM is also one of the popular classifiers in VMM classification [44]. A recent literature study shows that convolutional neural networks (CNNs) have set a new performance baseline in fine-grained visual classification [45–49]. Liu et al. [50] and Yang et al. [51] reinforced the viability of CNNs in fine-grained classification. Their work, GoogleNet, one of the first pre-trained deep learning models for fine-grained vehicle classification, outperformed the traditional approaches. Earlier research focused on auxiliary networks to learn local-level information for fine-grained classification. Krause et al. [52] proposed a fine-grained recognition method that worked without part annotations. They used the concept of alignment and segmentation to learn and detect useful parts. Xiao et al. [6] used three types of attention to extract relevant details of an image. They integrated these attentions to train deep nets. Zhang et al. [53] proposed an automatic fine-grained recognition approach, free of any object or part annotation. It extracted and pooled deep, distinctive filter responses and learned specific patterns signifi-

cantly and consistently. Wang et al. [54] emphasized mid-level representations of CNNs, which collected the class-level discriminative information end-to-end. Zhang et al. [55] addressed the constraints in pose-normalized representations for fine-grained classification. They introduced semantic part localization in convolutional neural networks and achieved state-of-the-art results. Fu [56] proposed a recurrent attention model that learns discriminative region attention and region-based feature representation at multiple scales without using bounding boxes. A novel part-stacked CNN proposed in [57] encodes the object-level and part-level cues simultaneously to model the subtle differences between the object parts. Hu [58] introduced spatially weighted pooling (SWP) layers in CNN, which pools extracted features by learning the discriminative spatial units. The proposed method surpassed previous fine-grained vehicle classification methods. Ma [59] improved the generalization ability of a CNN model by inserting a channel max pooling (CMP) layer between convolutional layers and the fully connected layers. In lightweight convolutional neural networks (LWCNNs) [60], network parameters are minimized and optimized by pre-training, fine-tuning training, and transfer training on a VMMR dataset [51].

Lam et al. [61] defined a heuristic function that scored the proposals of informative image parts and unified them via a long short-term memory (LSTM) network into a new deep recurrent architecture. Lin et al. [62] proposed a valve linkage function (VLF) for back-propagation chaining, improving the fine-grained classification performance of deep localization, alignment, and classification (LAC) systems. Zhang et al. [63] introduced the semantic part detection and abstraction (SPDA) approach in mid-level layers of an end-to-end CNN model. This approach shares the computation of convolutional filters and achieves state-of-the-art results in fine-grained classification. Different entropy loss functions were introduced to improve the performance of end-to-end neural networks. Deep CNNs with large-margin softmax (L-softmax) loss [7] created desired margins among features, made them more discriminative, and provided better classification results. The center loss was designed by Wen et al. [8] to improve inter-class dispensation and intra-class compactness. It learned the center of each class and restricted the distance of deep features from their respective classes. Focal loss [64] improved the dense object detection results by addressing the class imbalance problem and proposed training of hard-set examples only. Lin et al. [64] proposed a new loss function, introducing a regularization term to cross-entropy (CE) loss, which penalized the probability of a data point being assigned to a class other than its ground-truth class. The back-propagation algorithm used in CNN training typically optimizes the loss function. In contrast, in fine-grained classification, general and redundant features are undesirable. Ma et al. [59] addressed this problem by inserting a channel max pool layer between the convolutional layers and the fully connected layers of the CNN. This layer aimed to improve the generalization ability of the CNN by learning more discriminative features from a relatively lower number of feature maps. Experimental results demonstrated that CNNs with a CMP layer improved the classification accuracies on fine-grained vehicle classification with massively reduced parameters. Chang et al. [65] proposed a single loss, mutual-channel loss (MC-loss), applied directly to the feature channels to obtain class-aligned discriminative and diverse features. Naseer [66] also reduced the feature space by applying the genetic algorithm to deep features extracted from the VGG-16 CNN, fine-tuned on the frontal view of the vehicles, followed by an SVM classifier.

Our approach in this paper is similar to previous studies on fine-grained classification. Deep neural network (DNN) based deep learning (DL) techniques have demonstrated state-of-the-art results in VMM classification. Their ability to select features, transform, and classify data within a single framework, in particular, draws practitioners looking for ready-to-use solutions from raw data [67]. However, in severe data limitations or the absence of relevant transfer learning problems, DNN-based DL's advantages are drastically reduced [68]. We have proposed a hybrid CNN model fine-tuned on view-independent vehicle make and model datasets [5,69]. These datasets have a limited number of samples per class. The proposed model extracts deep features through the FC layer of a fine-tuned

CNN and produces the features that best describe a vehicle for fine-grained vehicle classification using the Fisher discriminative least squares regression (FDLSR) module [3]. It then trains a linear classifier on these discriminative features and makes predictions. Compared to a fine-tuned CNN, the proposed hybrid model improves recognition accuracy by 2.1%. The improved accuracy shows that the hybrid CNN model is more tolerant to view-independent, small-scale vehicle datasets than pure DNN-based DL models. CNNs undoubtedly demonstrate superior classification performance in VMMR systems. Previous approaches used auxiliary networks in CNNs, altered CNN architectures, and introduced different loss functions to CNNs for fine-grained vehicle classification. Specific methodologies that worked directly on CNN feature maps to improve their generalization ability also improved classification results. However, we observe that the advantages of DNN-based DL are drastically reduced in cases of severe data limitations or the absence of a relevant problem for transfer learning [68]. To address this problem and utilize CNN's ability to learn fine-grained features, we have proposed a hybrid CNN model fine-tuned on view-independent vehicle make and model datasets [5,69]. These datasets have a limited number of samples per class. The proposed model extracts deep features through an FC layer of a fine-tuned CNN and selects the most descriptive features using FDLSR. These transformed features exhibit improved inter-class disparity and intra-class similarity and are robust enough to be classified with a linear classifier. Table 1 lists some notable works in fine-grained image classification, especially VMMR.

Table 1. Summary of some Notable Works.

Year & Author	Objective	Dataset	Methodology	Result	Remarks
Biglari, M., 2018 [49]	To design a novel cascaded part-based system for VMMR	CompCars BVMMR	Novel greedy parts localization, and a practical multi-class data mining algorithm to detect discriminative vehicle region. Use of cascaded scheme to speed up the mechanism	up to 80% speed optimization. 97.01% accuracy on CompCars	Cascaded system performs with higher speed and accuracy than the baseline system.
Manzoor, M. A., 2019 [44]	To present a unique and robust real-time VMMR system which can handle unique set of challenges	NTOU-MMR	Used Histogram of Oriented Gradient (HOG) and GIST to represent the images and SVM and RF to classify the vehicles	97.20% with GIST features and SVM	System is well-suited for situations where vehicles are partially occluded, partially out of the image frame or poorly visible due to low lighting.
Benavides, N., 2019 [70]	Fine-tuning of a pre-trained CNN on a VMMR dataset	Stanford Cars	Transfer learning and fine-tuning of VGG16 Use of dropout, data augmentation and downsizing of dense layer	96.3% Top-5 Test Accuracy	Techniques used for dimensionality reduction are found crucial in fine-grained vehicle classification
Ma, Z., 2019 [59]	Improve generalization ability of CNNs for fine-grained classification	Stanford Cars, CompCars	Inserting Channel Max Pooling Layer (CMP) between the fully connected layers and the convolutional layer	97.89% by DenseNet161 on CompCar	CMP improves the performance of a network. It reduces the number of parameters in a neural network
Anwar, S., 2020 [71]	Comparison of general CNN classifiers and fine-grained classifiers	Stanford Cars, FGVC Aircrafts, Flowers, NA Birds	Transfer learning and fine-tuning of CNNs on FGVC datasets	94.5% by DenseNet161 on Stanford Cars	Traditional CNNs outperformed fine-grained classifiers in FGVC mainly because traditional CNNs are pre-trained CNNs.
Chang, D., 2020 [65]	Obtain fine-grained features using single-loss function	Birds, FGVC Aircraft, Flowers102, Stanford Car	Applied mutual-channel loss (MC-loss) directly to the feature channels	94.1% with ResNet-50 features on Stanford Cars	MC-Loss does not need fine-grained bounding-box. It can be applied to any network architecture. Does not need any extra parameter for tuning

Table 1. *Cont.*

Year & Author	Objective	Dataset	Methodology	Result	Remarks
Naseer, S., 2020 [66]	Proposal of a VMMR framework	NTOU MMR	Fine-tuning of VGG-16. Deep features extraction dimensionality reduction by GA. Classification using SVM.	98.20%	SVM performs better than any other classifier on fine-grained features and results are comparable to the state-of-the-art methods
Boukerche, A., 2021 [72]	LRAU to enhance the feature extraction ability of CNN architectures for VMMR.	Stanford Cars, CompCars, NTOU-MMR	Proposed LRAU extracts the discriminative part features by generating attention masks to locate the key points of a vehicle	93.94% on Stanford Cars	Model achieves excellent fine-grained recognition performance and can be used in a real-time environment

3. Proposed Methodology

In this section, we describe our proposed methodology in detail. Figure 1 provides an overview of our technique, and the subsequent sections describe each step in detail.

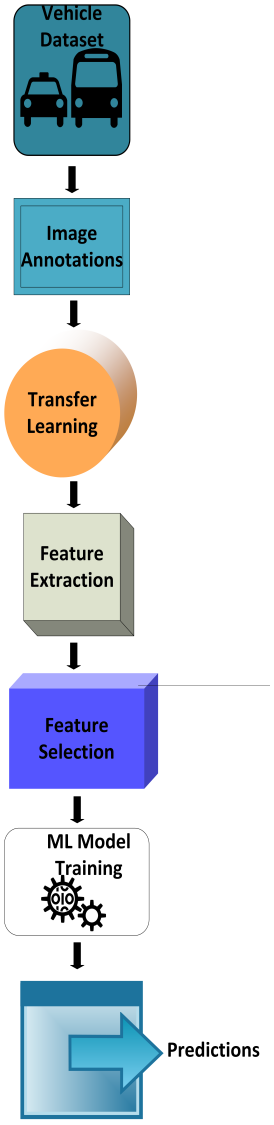

Figure 1. Proposed vehicle make and model recognition system.

3.1. Transfer Learning on Fine-Grained Vehicle Datasets

Deep neural networks trained on large-scale datasets like ImageNet [73] and COCO [74] have shown remarkable transfer learning capabilities. We fine-tune pre-trained CNNs (VGG-16, ResNet-50, and ResNet-152) to extract class-specific, fine-grained features. On our training data, we applied data augmentation. Data augmentation is essential and always recommended for small datasets. Random rotations, zooms, and horizontal flips are among the parameters of a data augmentation object. To perform transfer learning with VGG-16, we load its architecture (with pre-trained ImageNet weights) from the disc and remove the fully connected layers. Figure 2a shows the original CNN. Figure 2b depicts our network without the FC layer. We then define a new fully connected layer head and freeze all VGG-16 CONV layers. At this point, training our model will only tune our network head and not update the base weights (Figure 2c). We reset our training and validation generators before unfreezing the final set of CONV layers, then unfreeze the final set of CONV layers. Figure 2d shows the final stage, which is to train our model to fine-tune the FC layer head and the final CONV block.

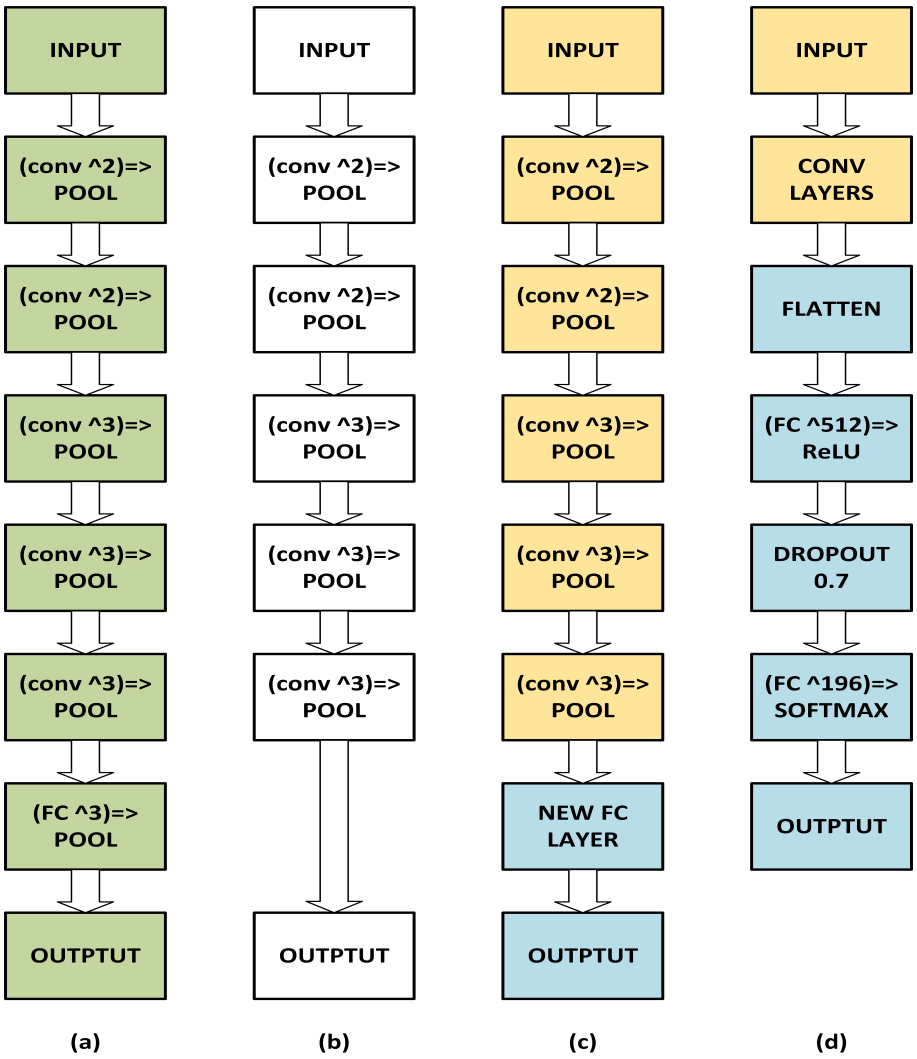

Figure 2. CNN architecture for transfer learning. (**a**) original CNN. (**b**) Our network without the FC layer. (**c**) Tuned network head without updating the base weights. (**d**) Fine-tuning of the FC layer head and the final CONV block.

3.2. Feature Extraction with Deep Learning

The architecture of a pre-trained neural network allows us to use it as an arbitrary feature extractor. The input image propagates forward and stops at the pre-defined layer, allowing us to retrieve features from that layer. We can use powerful CNN features this way. We take our fine-tuned VGG-16 network and, similarly, allow an image to propagate forward to the dense layer (the first hidden layer of our fully connected layer) and extract features from it. This dense layer produces a 2048-dimensional feature vector. We can repeat the feature extraction process for each image in the dataset, yielding a total of N $\times$ 2048-dimensional feature vectors.

3.3. Feature Engineering with Fisher Discriminative Least Squares Regression (FDLSR)

To understand FDLSR [3], suppose we have a system $QX = Y$ composed of a training dataset X with m features and n training examples. Let Q be the best-fit solution for the system such that $QX \approx Y$. We use the optimization function of a least squares regression (LSR) model to find Q. The least squares regression (LSR) model finds the best possible solution by minimizing the residual sum of the squared (RSS) error [75]. The optimization function is written as:

$$RSS = \sum (y_t - \hat{y}_t)^2. \tag{1}$$

However, solving a singular matrix for some RSS problems is difficult. Non-negative dragging values $\{\epsilon_{11}, \epsilon_{12}, \ldots, \epsilon_{34}\}$ are added in the regularized RSS function under a technique called ϵ-dragging. The ϵ-dragging technique improves the inter-class margins, but it is observed that the class margins do not change significantly with each iteration, and DLSR does not consider the intra-class compactness of the relaxed labels. The Fisher criterion is applied to the ϵ-draggings to address this issue, increase inter-class separability, and improve intra-class compactness during each iteration. Thus, the Fisher discriminative least squares regression (FDLSR) [3] model can be formulated as a discriminative least squares regression (DLSR) model inspired by the Fisher criterion and ϵ-dragging method:

$$min|QX - (Y + G.T)|^2 + \tau|Q|^2 + \lambda Fisher(Y + G.T), \tag{2}$$

where

$$T >= 0,$$

where Q is the projection matrix and S is the non-negative relaxation matrix. The matrix $Y + G \times T$ denotes the relaxed labels learned by the ϵ-dragging method. The first term is used to learn discriminative projection Q with relaxed regression labels, as shown in Equation (2). The third term aims to regularize the learned labels using the Fisher criterion. We introduce a transition variable H and rewrite our FDLSR model to understand better and optimize the Fisher function:

$$min|QX - H|^2 + \beta|H - (Y + G.T)|^2 + \tau|Q|^2 + \lambda Fisher(H) \tag{3}$$

$$Fisher(H) = \sum (|H - P|^2 + |P_m - P|^2 + |H|^2), \tag{4}$$

where

$$\beta, \lambda, \tau > 0$$

are scalars that weigh the corresponding terms in Equation (3), where P represents the relaxed labels of the mth class. P consists of N identical columns equal to the mean vector of all columns in H. P includes n identical columns equal to the mean vector of all columns in H.

To enhance intra-class compactness and inter-class separability of extracted features, we engineer the extracted features with the help of a Fisher discriminative least squares function in Equation (4). The extracted deep features X and their corresponding labels Y are loaded. These features are normalized, and their labels are converted into a one-

hot encoded matrix. The FDLSR function uses a feature matrix (X), a label matrix (Y), and parameters β, τ, and λ as input to formulate the projection matrix Q. The FDLSR function undergoes 30 iterations to find a convergent solution. The function updates the transition H, projection Q, and relaxation T matrix during each iteration. The FDLSR algorithm projects the training data into a lower-dimensional subspace of Q by taking its dot product with the projection matrix. The transformed training set is now of the size R ($c \times n$), where c represents the number of classes in a dataset. The pseudocode of FDLSR is showcased in Algorithm 1.

Algorithm 1 Fisher Discriminative Least Squares Regression (FDLSR).

Initialization:

$Q = YX^T(XX^T + \tau I)^{-1}; T = 0^{c \times n}; H = Y;$

$G = 2Y - 1^{c \times n}; \hat{P} = [P_1, P_2, P_3, ..., P_c]$

Let $i = 1; Q_{xy} = Q$

while $i < iterations_{max}$ **do**

$$H = \frac{QX + \beta(Y + G \cdot T) - \lambda P - 2\lambda\hat{P}}{1 + \beta + 2\lambda}$$

$Q = HX^T(XX^T + \tau I)^{-1}$

$T = max(G \cdot (H - Y), 0)$

if $\|Q - Q_{xy}\|_F^2 < 10^{-4}$ **then**

Stop

end if

$i = i + 1, Q_{xy} = Q$

end while

Output: Q

3.4. Feature Classification Using Linear Classifier

We assume that features extracted by a fine-tuned CNN model are already robust and discriminative, as CNN can learn non-linear features. Therefore, once we have these transformed features, we can train off-the-shelf machine learning models such as Linear SVM and KNN on these features to recognize a new set of images. Support vector machines (SVMs) [4] are the supervised machine learning algorithms for classification and regression problems. For linearly separable cases, the optimization function is:

$$y_i(mx_i + c) - 1 >= 0 \tag{5}$$

$$s.t. min\{1/2|w|^2\}.$$

For multiclass classification, $n(n-1)/2$ classifiers are trained in one-vs-one approach to classify samples from every pair of classes. The k-nearest neighbor algorithm considers the dimensions of the data points in a given space. It randomly selects data points from each class as class centers and calculates the distance between other samples and these center points. The commonly used metric to find the distance in a KNN algorithm is the Euclidean distance, which is given by:

$$d(x_1, x_2) = \sqrt{\sum(x_1 - x_2)}. \tag{6}$$

3.5. Overview of Proposed Algorithm

To conclude this section, we list the steps to implement our proposed algorithm.

Step 1: Load the dataset.

Step 2: Image preprocessing (annotations, augmentation).

Step 3: Fine-tune the most suitable CNN model pre-trained on the ImageNet dataset.

Step 4: Extract features from the fine-tuned CNN model's fully connected (FC) layer. (The FC layer first flattens the feature map and gives it a vector form. The fully connected layer receives input from the last pooling or convolutional layer. The number of channels in the output feature maps extracted from a pre-trained VGG-16 is fixed at 512 and that of ResNet50 or ResNet152 at 2048).

Step 5: Feature normalization.(Given the fixed size of the feature vector, this would produce 37,689 2048-dimension feature vectors and 8144 2048-dimension vectors for BoxCar21k and Stanford Cars, respectively.)

Step 6: Begin with an 80/20 training validation split for both datasets. (Both are small and increasing the validation set might overfit the CNN model.)

Step 7: Transform the features with FDLSR as described in detail in Section 3.3.

Step 8: Feature normalization. (After applying the Fisher discriminative least squares function, the feature vectors are dimensionally reduced, yielding 2048 $\times$ 87-dimensional vectors and 2048 $\times$ 196-dimensional vectors for BoxCar21k and Stanford Cars, respectively.)

Step 9: Train an off-the-shelf classifier. (e.g., SVM or KNN.)

Step 10: Predictions.

4. Experimental Results and Discussions

4.1. Datasets

We have chosen the Stanford Cars dataset [5] and BoxCars21k [47] for our research. We chose the Stanford Cars dataset for its many classes and a few instances in each class. It is one of the earliest benchmark datasets. The dataset contains 16,185 view-independent images belonging to 196 classes of cars. The data are split nearly 50/50, with 8144 training images and 8041 testing images. Classes are at the level of make, model, and year.

Figure 3 shows some images from the dataset. The sample images show the dataset's view-independent nature and different illumination conditions. The BoxCars21k dataset contains 63,750 vehicle images of 148 fine-grained classes (make, model, and model year). Based on the fine categorization of the make-model hierarchy, the dataset is divided into easy, hard, and medium subsets. There is a considerable variation in viewpoints in the dataset. The dataset provides a 3D bounding box for each image. We have worked on the hard split, containing 37,689 images for training and 18,939 for testing, belonging to 87 fine-grained classes. Figure 4 shows sample images from the dataset.

While carrying out experimentation for the choice of the best CNN model for feature extraction, another dataset was also used. Despite the ongoing research involving car make and model analysis, there is an absence of diverse datasets involving traffic dynamics in developing countries. Thus, we collected a comprehensive dataset that shall serve as a benchmark to further the research on traffic analytics to propose guidelines for ITS in developing countries like Pakistan. There are 129,000 images belonging to 94 different classes of vehicles on Pakistani roads to date. The dataset contains occluded images and partial and overhead camera views under low illumination. Images are labeled according to make, model, and generation; for example, HondaCity5 means Honda City 5th generation. Some examples are shown in Figure 5. Table 2 lists the main attributes of the datasets used for our experiments.

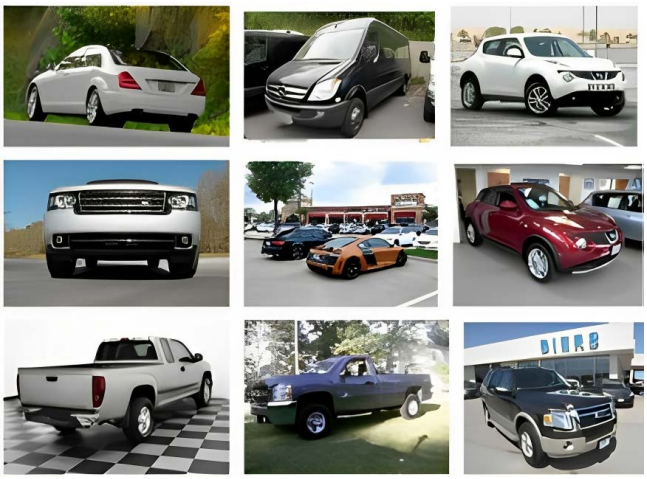

Figure 3. Images from the Stanford Cars dataset.

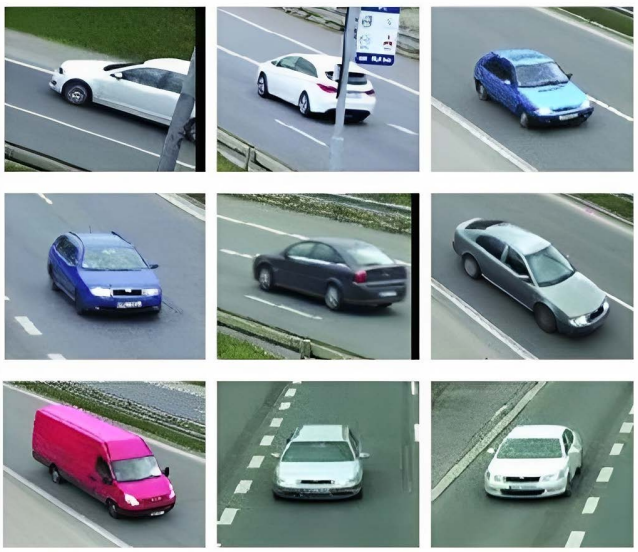

Figure 4. Images from the BoxCars21k dataset.

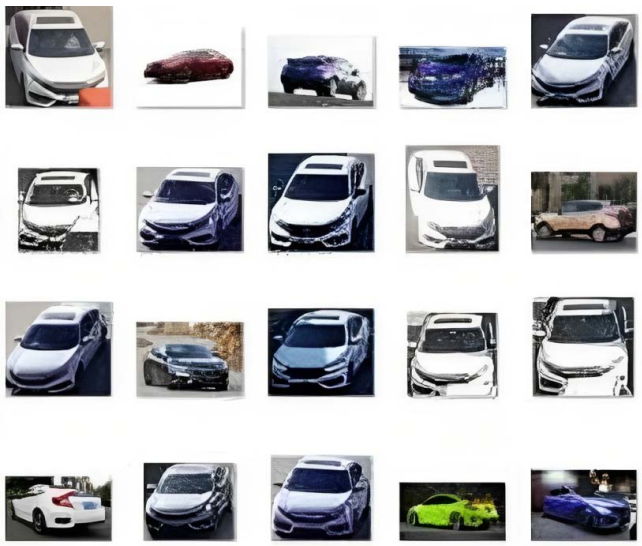

Figure 5. Images from the Pakistani cars dataset.

Table 2. Main attributes of datasets used in experiments.

Dataset	Year	Samples	Diversity	Annotations	Image Resolution	No. of Classes	Train/Test Split
BoxCars21k [47]	2016	63,750	View-Independent	Make, model year	Low	148	70/30
Stanford Cars [5]	2013	16,185	View-Independent	Make, model year	Mixed	196	50/50
Pakistani Cars	2022	129,000	View-Independent	Make, model generation	Low	94	60/40

4.2. Choice of CNN

Considering the relatively small size of our datasets, training a deep neural network (DNN) can easily lead to overfitting. In such a situation, transfer learning is the natural solution. Transfer learning can achieve better performance with a relatively small dataset. In our proposed system, we trained the following popular CNN models to choose the best-performing model for our proposed approach.

- ResNet50 [69]
- ResNet152 [69]
- VGG-16 [76]
- InceptionV3 [77]
- MobileNet [78]

The dataset contains images taken by different users, imaging devices, and multiple view angles, ensuring numerous variations. As a result, the cars are not well-aligned, and some images have irrelevant backgrounds. The data were gathered by collecting and cleaning images from the internet and then cropping and cleaning images from Pakistani overhead traffic videos taken at different locations. Pictures taken from the internet are automatically annotated using the title and description the sellers had provided for each post. Figure 4 shows some images of the Honda Civic 10th generation from the dataset.

Most of these models are trained on the ImageNet dataset [73], which makes these CNN models ideal candidates for transfer learning. Each chosen model has its advantages. ResNet models, being most famous for transfer learning, help tackle the vanishing gradient problem and increase the training speed. They provide higher accuracy, especially for classification problems. These models learn the difference among the already learned features. If the learned feature is not helpful, then the final decision weights are set to zero for that particular feature. The main strength of the VGG models is that they are easy to understand and explain. They are suitable for typical two-class problems like cats vs. dogs classification. InceptionV3 has many advantages, as it reduces computational cost. It trains faster than the VGG family. The size of the model is smaller than VGG. MobileNet offers several advantages over other state-of-the-art convolutional neural networks, including reduced network size, reduced number of parameters, and faster performance, and it is helpful for mobile applications. Even though MobileNet has the advantage of smaller size, fewer parameters, and fast performance, it is less accurate than other state-of-the-art networks. Table 3 lists the test accuracies achieved by our chosen models for Stanford Cars and the local Pakistani on-road cars dataset.

Table 3. Accuracies achieved.

No.	Model Name	Test Accuracies (Pakistani Cars)	Test Accuracies (Stanford Cars)
1	ResNet50	90%	92%
2	ResNet152	90%	82%
3	VGG-16	70.80%	71%
4	Inception V3	70%	58%
5	MobileNet		

4.3. Experimental Environment

All the experiments were performed on a GPU virtual machine with 16 GB RAM and a dual core CPU. Python 3.7 was used as the programming language.

4.4. Implementation Details

The most important thing to note is that the number of channels in the output feature maps extracted from a pre-trained VGG-16 (ResNet50, ResNet152) is fixed at 512 (2048). We fine-tune the pre-trained models with our proposed loss function to explore the pre-trained rich discriminative features of the VGG-16 (ResNet50, ResNet152) learned on a large ImageNet dataset. With the fixed size of the feature vector, this would produce 37,689 feature vectors of 2048 dimensions and 8144 vectors of 2048 dimensions for BoxCar21k (with 87 classes) and Stanford Cars (with 196 classes), respectively. After applying the Fisher discriminative least squares regression (FDLSR) function, the feature vectors are dimensionally reduced, yielding 2048 $\times$ 87-dimensional vectors and 2048 $\times$ 196-dimensional vectors for BoxCar21k and Stanford Cars, respectively.

To compare our approach with other state-of-the-art methods, we annotate and resize every image in the dataset to 224 $\times$ 224, then extract features using VGG-16 (ResNet50, ResNet152) pre-trained on ImageNet classification datasets. We began with an 80/20 training validation split for both datasets because both are small, and increasing the validation set might overfit the CNN model. We used stochastic gradient descent and batch normalization as regularizers. The learning rate of fully connected layers is kept at 0.0001, and we have trained no model for more than 100 epochs. Table 4 summarizes the hyperparameter values.

Table 4. Summary of hyperparameters.

Hyperparameters	Value	Rationale
Optimizer	SGD	Recommended for ResNet models [79]
Learning Rate	0.0001	Recommended for ResNet models [80] because we do not want to change too much what is previously learned
Batch Size	64	Best accuracy
Epoch	<100	Avoid overfitting

4.5. Evaluation Protocol and Measures

We conducted several experiments and analyzed our results to determine the best practices for vehicle make and model recognition using the chosen CNN architectures (VGG-16, ResNet50, and ResNet152). According to Table 3, these are the top three performing models. We have used top-1 and top-5 accuracy metrics to evaluate the performance of different fine-grained classification models. In fine-grained classification, differences between classes are pretty subtle, and the correct class is often in the top-k prediction, making top-k (k = 2, 3, 4, ...) accuracy significantly higher than top-1 accuracy. We have exploited this accuracy gap to understand the performance of different classification models. We compared different classification models in this section in terms of accuracies, computational complexity, and other factors such as runtime.

4.5.1. Test Accuracy Comparison

The different classification models trained on the same database with varying CNN model features have shown drastic variances in performance. Table 5 compares the accuracy of the FC layer, SVM, and our proposed classification model tested on the deep features of Stanford Cars. The highest top-1 test accuracy observed for the Stanford Cars database is 94.62% for our proposed model trained on fine-tuned ResNet152 features. The SVM model has performed better on ResNet50 features than other CNN features, with 94.44% accuracy, whereas the FC layer classification performance with ResNet152 features is comparably more convincing than others, with 90.37% top-1 accuracy. We observe that the accuracy gap between our proposed classifier's top-1 and top-5 accuracy is minimal and ranges between 4–11%. This range stretches to 5–15% with SVM and 8–18% in the case of the FC layer. Additionally, this accuracy gap is associated with the final loss of the classifier, and with a higher gap, the losses are also higher. Since our proposed classifier has decreased this

gap, minimal loss, i.e., 0.052, is observed by our classifier on ResNet152 features. The same trend is marked with the BoxCar21k database, as shown in Table 6.

Table 5. Stanford Cars test accuracies.

Classification Models	Feature Extracting Models	Top-1 Accuracy	Top-5 Accuracy	Final Loss
FC Layer	ResNet50	90.37	98.40	0.096
	ResNet152	90.52	98.15	0.094
	VGG-16	76.32	94.47	0.236
SVM	ResNet50	94.44	99.02	0.056
	ResNet152	94.17	99.03	0.053
	VGG-16	81.76	96.40	0.183
Ours	ResNet50	94.16	98.98	0.053
	ResNet152	94.62	99.09	0.052
	VGG-16	86.17	97.51	0.166

Table 6. BoxCars21k test accuracies.

Classification Models	Feature Extracting Models	Top-1 Accuracy	Top-5 Accuracy	Final Loss
FC Layer	ResNet50	96.15	99.56	0.038
	ResNet152	97.91	99.92	0.020
	VGG-16	93.91	99.25	0.060
SVM	ResNet50	98.40	99.99	0.016
	ResNet152	98.41	99.99	0.019
	VGG-16	93.21	99.44	0.082
Ours	ResNet50	98.74	100	0.013
	ResNet152	98.88	100	0.012
	VGG-16	96.84	100	0.031

4.5.2. Computational Complexity Comparison

The complexity analysis of the FDLSR algorithm in Figure 2 is as follows [3]. When we update T, computation complexity is $O(ndc)$. When updating Q, the complexity is

$$O(nd^2 + d^3).$$

Therefore, the final computational complexity of updating S is

$$O(ndc + nd^2 + d^3)$$

Since the number of training samples and classes is much smaller than the dimensionality of the feature vector, the main time-consuming step is computing

$$X^T \cdot (1/(XX^T + \beta I))$$

This term can be pre-computed because its value does not change during iteration. As a result, the final computational complexity of FDLSR [3] is

$$O((nd^2 + d^3) + 2tndc),$$

where t is the number of iterations, n is the number of samples, d is the dimensionality of the data, and c is the number of classes in the dataset. The computational complexity of SVM is

$$O(nd^2)$$

per iteration [4]. The proposed algorithm has the lowest computational complexity Since FDLSR converges in 30 iterations, while SVM takes 500 iterations to converge.

4.6. Runtime Comparison

The extracted feature vector has the dimension $2048 \times N$, where N is the number of sample images. Our proposed classifier has shown the lowest and almost equal runtime on both datasets (Figures 6 and 7). Even with different CNN models, the runtime is constant, which shows that the number of sample images and the nature of extracted features have no impact on the runtime of our classification model. We can observe that the nature and the order of hidden layers in the FC layer affect its runtime. Similarly, SVM depends on the nature of the training set, as it has a varying runtime with different CNN model features.

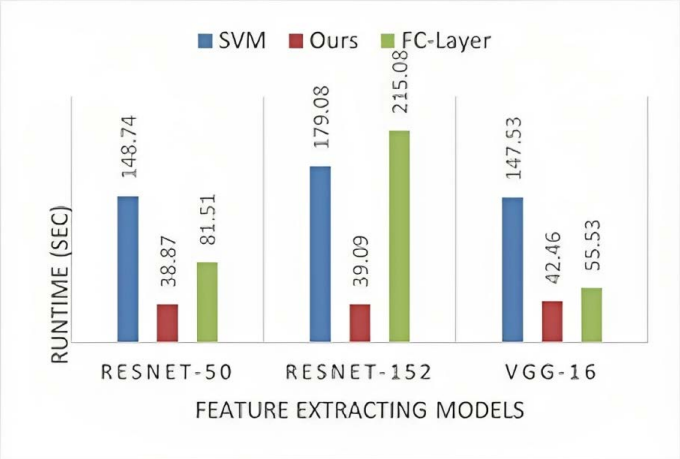

Figure 6. Runtime on the Stanford Cars dataset.

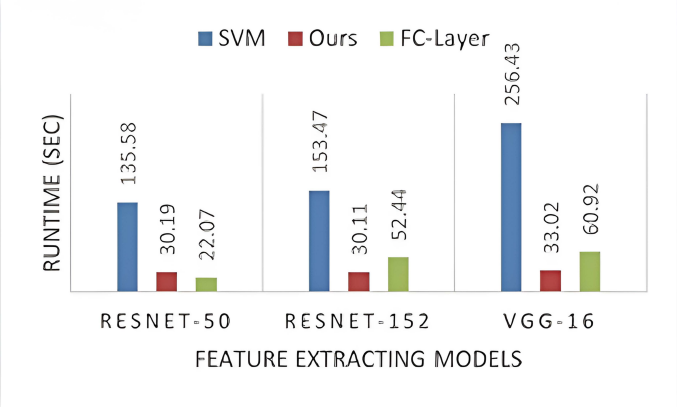

Figure 7. Runtime on the BoxCars21k dataset.

4.7. Comparisons with State-of-the-Art Methods

Our proposed approach for VMMR presented in this paper outperforms several related VMMR works regarding classification accuracy. A comparison of our work with the results of other associated works on the Stanford Cars dataset is presented in Table 7. We have used three main categories of fine-grained recognition methods to draw comparisons. The first category is based on the attention mechanism, which includes a fully convolutional attention network (FCAN) [81], recurrent attention CNN (RA-CNN) [56], multi-attention convolutional neural network (MA-CNN) [82], dynamic time recurrent attention model (DT-RAM) [83], and trilinear attention sampling network (TA-SN) [84]. The second category, which is high-dimensional feature coding, includes a bilinear convolutional neural network (BCNN) [85], kernel pooling (KP) [86], higher-order integration of hierarchical convolutional activations (HIHCA) [87], boosted convolutional neural network (Boost-CNN) [88], HBP, and HBP with aggregated slack mask (HBPASM) [89]. Moreover, the third category is based on vehicle-specific characteristics, which include dual cross-entropy loss

(DCEL) [90] and the global topology constraint network (GTCN) [91]. Using the ResNet152 model as the 379 feature extractor, the proposed fine-grained classification model achieves the best accuracy of 94.61% on the Stanford Cars dataset.

Table 7. Comparison of the Proposed Method with State-of-the-art Methods.

Method	Accuracy %
Channel Max Pooling (CMP) [59]	93.71
Spatially weighted pooling (SWP) [58]	93.1
Mutual-channel loss (MC) [65]	90.85
Recurrent-attention CNN (RA-CNN) [56]	92.5
Multi attention CNN (MA-CNN) [82]	92.8
Fully convolutional attention network (FCAN) [81]	89.1
Dynamic time recurrent attention model (DT-RAM) [83]	93.1
Trilinear attention sampling network (TA-SN) [84]	93.8
Kernel pooling (KP) [86]	92.4
Higher-order integration of hierarchical convolutional activations (HIHCA) [87]	91.7
Bilinear convolutional neural network (BCNN) [85]	92.1
Dual cross-entropy loss (DCEL) [90]	93.3
Global topology constraint network (GTCN) [91]	94.3
Our proposed method	94.61

5. Conclusions

This paper proposed a novel classifier based on FDLSR to solve the problem of view-independent car make and model classification. For our research, we have chosen the Stanford Cars dataset and BoxCars21k. The former was selected for its large number of classes and a small number of instances in each class, while the latter was selected for the considerable variation in viewpoints in the dataset. We also introduced a Pakistani cars dataset and conducted experiments for CNN selection on it. Preexisting CNN models were considered for feature extraction and after extensive experimentation, ResNet-50, ResNet-152, and VGG-16 were selected. Selected features were fed to our proposed classifier. Experimental results show that our proposed classifier achieves substantially better results than the existing state-of-the-art approaches. Our method deals with the main problem deep neural networks face, i.e., poor performance on a small training set. Due to FDLSR's ability to increase inter-class distance and decrease intra-class distances, class boundaries become more defined. We see superior performance on datasets with a large number of classes and with a small number of samples per class. Our proposed classifier has the shortest run time independent of the type of features fed to the classifier. For future work, we plan to conduct experiments on the Pakistani cars dataset and implement incremental learning for feature extraction.

Author Contributions: Conceptualization, S.H., F.H. and M.H.Y.; methodology, S.H.; implementation, S.H.; validation, S.H., F.H. and M.H.Y.; formal analysis, S.H.; investigation, S.H., F.H. and M.H.Y.; data curation, S.H.; writing—original draft preparation, S.H.; writing—review and editing, F.H. and M.H.Y.; supervision, F.H. and M.H.Y.; project administration, M.H.Y.; funding acquisition, M.H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the Higher Education Commission (HEC) of Pakistan for funding this project through a grant titled "Establishment of National Centre of Robotics and Automation (NCRA)" under Grant DF-1009-31.

Data Availability Statement: This research used publicly available datasets for experimentation and analysis purposes.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Atzori, A.; Barra, S.; Carta, S.; Fenu, G.; Podda, A.S. HEIMDALL: An AI-based infrastructure for traffic monitoring and anomalies detection. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerComWorkshops), Kassel, Germany, 22–26 March 2021 ; pp. 154–159.
2. Guerrero-Ibañez, J.; Contreras-Castillo, J.; Zeadally, S. Deep learning support for intelligent transportation systems. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4169. [CrossRef]
3. Chen, Z.; Wu, X.J.; Kittler, J. Fisher Discriminative Least Squares Regression for Image Classification. *arXiv* **2019**, arXiv:1903.07833.
4. Burges, C.J. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167. [CrossRef]
5. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3d object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 554–561.
6. Xiao, T.; Xu, Y.; Yang, K.; Zhang, J.; Peng, Y.; Zhang, Z. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 842–850.
7. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. *arXiv* **2016**, arXiv:1612.02295.
8. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 499–515.
9. Liao, L.; Hu, R.; Xiao, J.; Wang, Q.; Xiao, J.; Chen, J. Exploiting effects of parts in fine-grained categorization of vehicles. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 745–749.
10. Lai, A.H.; Yung, N.H.C. Vehicle-type identification through automated virtual loop assignment and block-based direction-biased motion estimation. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 86–97. [CrossRef]
11. Avery, R.P.;Wang, Y.; Rutherford, G.S. Length-based vehicle classification using images from uncalibrated video cameras. In Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749), Washington, DC, USA, 3–6 October 2004; pp. 737–742.
12. Kafai, M.; Bhanu, B. Dynamic Bayesian networks for vehicle classification in video. *IEEE Trans. Ind. Inform.* **2011**, *8*, 100–109. [CrossRef]
13. Ma, X.; Grimson, W.E.L. Edge-based rich representation for vehicle classification. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; Volume 2, pp. 1185–1192.
14. Kumar, T.S.; Sivanandam, S.N. A modified approach for detecting car in video using feature extraction techniques. *Eur. J. Sci. Res.* **2012**, *77*, 134–144.
15. Gupte, S.; Masoud, O.; Martin, R.F.; Papanikolopoulos, N.P. Detection and classification of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 37–47. [CrossRef]
16. Yoshida, T.; Mohottala, S.; Kagesawa, M.; Ikeuchi, K. Vehicle classification system with local-feature based algorithm using CG model images. *IEICE Trans. Inf. Syst.* **2002**, *85*, 1745–1752.
17. Chen, Z.; Ellis, T.; Velastin, S.A. Vehicle type categorization: A comparison of classification schemes. In Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 74–79.
18. Zhang, B.; Zhou, Y.; Pan, H. Vehicle classification with confidence by classified vector quantization. *IEEE Intell. Transp. Syst. Mag.* **2013**, *5*, 8–20. [CrossRef]
19. Daya, B.; Chauvet, P. Identification system of the type of vehicle. In Proceedings of the 2010 IEEE Fifth International conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Changsha, China, 23–26 September 2010; pp. 1607–1612.
20. Pearce, G.; Pears, N. Automatic make and model recognition from frontal images of cars. In Proceedings of the 2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Madrid, Spain, 30 August–2 September 2011; pp. 373–378.
21. Kamal, I. Car recognition for multiple data sets based on histogram of oriented gradients and support vector machines. In Proceedings of the 2012 International Conference on Multimedia Computing and Systems, Tangiers, Morocco, 10–12 May 2012; pp. 328–332.
22. Negri, P.; Clady, X.; Milgram, M.; Poulenard, R. An oriented-contour point based voting algorithm for vehicle type classification. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 1, pp. 574–577.
23. Kazemi, F.M.; Samadi, S.; Poorreza, H.R.; Akbarzadeh-T, M.R. Vehicle recognition using curvelet transform and SVM. In Proceedings of the Fourth International Conference on Information Technology (ITNG'07), Las Vegas, NA, USA, 2–4 April 2007; pp. 516–521.
24. Rahati, S.; Moravejian, R.; Kazemi, E.M.; Kazemi, F.M. Vehicle recognition using contourlet transform and SVM. In Proceedings of the Fifth International Conference on Information Technology: New Generations (Itng 2008), Las Vegas, NA, USA, 7–8 April 2008; pp. 894–898.
25. Arzani, M.M.; Jamzad, M. Car type recognition in highways based on wavelet and contourlet feature extraction. In Proceedings of the 2010 International Conference on Signal and Image Processing, Zhejiang, China, 9–11 April 2010; pp. 353–356.

26. Zhang, B. Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Trans. Intell. Transp.* **2012**, *14*, 322–332. [CrossRef]
27. Dong, Z.;Wu, Y.; Pei, M.; Jia, Y. Vehicle type classification using a semisupervised convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2247–2256. [CrossRef]
28. Psyllos, A.P.; Anagnostopoulos, C.N.E.; Kayafas, E. Vehicle logo recognition using a sift-based enhanced matching scheme. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 322–328. [CrossRef]
29. Dlagnekov, L.; Belongie, S. Recognizing Cars 2005. Available online: https://escholarship.org/uc/item/7b13d6cw (accessed on 3 September 2023).
30. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
31. He, H.; Shao, Z.; Tan, J. Recognition of car makes and models from a single traffic-camera image. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3182–3192. [CrossRef]
32. Manjunath, B.S.; Salembier, P.; Sikora, T. *Introduction to MPEG-7: Multimedia Content Description Interface*; John Wiley and Sons: Hoboken, NJ, USA, 2002.
33. Siddiqui, A.J.; Mammeri, A.; Boukerche, A. Real-time vehicle make and model recognition based on a bag of SURF features. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3205–3219. [CrossRef]
34. Hsieh, J.W.; Chen, L.C.; Chen, D.Y. Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 6–20. [CrossRef]
35. Santos, D.; Correia, P.L. Car recognition based on back lights and rear view features. In Proceedings of the 2009 10th Workshop on Image Analysis for Multimedia Interactive Services, London, UK, 6–8 May 2009; pp. 137–140.
36. Llorca, D.F.; Colás, D.; Daza, I.G.; Parra, I.; Sotelo, M. Vehicle model recognition using geometry and appearance of car emblems from rear view images. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 3094–3099.
37. Leotta, M.J.; Mundy, J.L. Vehicle surveillance with a generic, adaptive, 3d vehicle model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1457–1469. [CrossRef]
38. Guo, Y.; Rao, C.; Samarasekera, S.; Kim, J.; Kumar, R.; Sawhney, H. Matching vehicles under large pose transformations using approximate 3d models and piecewise mrf model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
39. Prokaj, J.; Medioni, G. 3-D model based vehicle recognition. In Proceedings of the 2009 Workshop on Applications of Computer Vision (WACV), Snowbird, UT, USA, 7–8 December 2009; pp. 1–7.
40. Psyllos, A.; Anagnostopoulos, C.N.; Kayafas, E. Vehicle model recognition from frontal view image measurements. *Comput. Stand. Interfaces* **2011**, *33*, 142–151. [CrossRef]
41. Manzoor, M.A.; Morgan, Y. Vehicle make and model recognition using random forest classification for intelligent transportation systems. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 148–154.
42. Tang, Y.; Zhang, C.; Gu, R.; Li, P.; Yang, B. Vehicle detection and recognition for intelligent traffic surveillance system. *Multimed. Tools Appl.* **2017**, *76*, 5817–5832. [CrossRef]
43. Fang, J.; Zhou, Y.; Yu, Y.; Du, S. Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1782–1792. [CrossRef]
44. Manzoor, M.A.; Morgan, Y.; Bais, A. Real-time vehicle make and model recognition system. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 611–629. [CrossRef]
45. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–28 June 2014; pp. 580–587.
46. Sun, Y.; Wang, X.; Tang, X. Deep learning face representation from predicting 10,000 classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–28 June 2014; pp. 1891–1898.
47. Sochor, J.; Herout, A.; Havel, J. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3006–3015.
48. Zhang, H.;Wang, K.; Tian, Y.; Gou, C.;Wang, F.Y. MFR-CNN: Incorporating multi-scale features and global information for traffic object detection. *IEEE Trans. Veh. Technol.* **2018**, *67*, 8019–8030. [CrossRef]
49. Biglari, M.; Soleimani, A.; Hassanpour, H. A cascaded part-based system for fine-grained vehicle classification. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 273–283. [CrossRef]
50. Liu, D.; Wang, Y. *Monza: Image Classification of Vehicle Make and Model Using Convolutional Neural Networks and Transfer Learning*; Stanford University: Stanford, CA, USA, 2017.
51. Yang, L.; Luo, P.; Change Loy, C.; Tang, X. A large-scale car dataset for fine-grained categorization and verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3973–3981.
52. Krause, J.; Jin, H.; Yang, J.; Fei-Fei, L. Fine-grained recognition without part annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5546–5555.
53. Zhang, X.; Xiong, H.; Zhou, W.; Lin, W.; Tian, Q. Picking deep filter responses for fine-grained image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1134–1142.

54. Wang, Y.; Morariu, V.I.; Davis, L.S. Learning a discriminative filter bank within a cnn for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4148–4157.
55. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for fine-grained category detection. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
56. Fu, J.; Zheng, H.; Mei, T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4438–4446.
57. Huang, S.; Xu, Z.; Tao, D.; Zhang, Y. Part-stacked CNN for fine-grained visual categorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1173–1182.
58. Hu, Q.; Wang, H.; Li, T.; Shen, C. Deep CNNs with spatially weighted pooling for fine-grained car recognition. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3147–3156. [CrossRef]
59. Ma, Z.; Chang, D.; Xie, J.; Ding, Y.; Wen, S.; Li, X.; Si, Z.; Guo, J. Fine-grained vehicle classification with channel max pooling modified CNNs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3224–3233. [CrossRef]
60. Zhang, Q.; Zhuo, L.; Zhang, S.; Li, J.; Zhang, H.; Li, X. Fine-grained vehicle recognition using lightweight convolutional neural network with combined learning strategy. In Proceedings of the 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), Xi'an, China, 13–16 September 2018; pp. 1–5.
61. Lam, M.; Mahasseni, B.; Todorovic, S. Fine-grained recognition as hsnet search for informative image parts. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2520–2529.
62. Lin, D.; Shen, X.; Lu, C.; Jia, J. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1666–1674.
63. Zhang, H.; Xu, T.; Elhoseiny, M.; Huang, X.; Zhang, S.; Elgammal, A.; Metaxas, D. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1143–1152.
64. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.
65. Chang, D.; Ding, Y.; Xie, J.; Bhunia, A.K.; Li, X.; Ma, Z.; Wu, M.; Guo, J.; Song, Y.Z. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Trans. Image Process.* **2020**, *29*, 4683–4695. [CrossRef]
66. Naseer, S.; Shah, S.M.A.; Aziz, S.; Khan, M.U.; Iqtidar, K. Vehicle make and model recognition using deep transfer learning and support vector machines. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
67. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
68. Gavrishchaka, V.; Yang, Z.; Miao, R.; Senyukova, O. Advantages of hybrid deep learning frameworks in applications with limited data. *Int. J. Mach. Learn. Comput.* **2018**, *8*, 549–558.
69. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
70. Benavides, C.N.; Tae, C. Fine Grained Image Classification for Vehicle Make and Model Using Convolutional Neural Network. *CS230 Standford*. 2019. Available online: https://cs230.stanford.edu/ (accessed on 3 September 2023).
71. Anwar, S.; Barnes, N.; Petersson, L. A systematic evaluation: Fine-grained CNN vs. traditional CNN classifiers. *arXiv* **2020**, arXiv:2003.11154.
72. Boukerche, A.; Ma, X. A novel smart lightweight visual attention model for fine-grained vehicle recognition. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 13846–13862. [CrossRef]
73. Deng, J.; Dong,W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
74. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
75. Ruppert, D. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Taylor & Francis: Abingdon, UK, 2004.
76. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
77. Szegedy, C.; Liu,W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp.1–9.
78. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
79. Naseer, I.; Akram, S.; Masood, T.; Jaffar, A.; Khan, M.A.; Mosavi, A. Performance analysis of state-of-the-art cnn architectures for luna16. *Sensors* **2022**, *22*, 4426. [CrossRef] [PubMed]
80. Transfer Learning: The Dos and Don'ts. Available online: https://medium.com/starschema-blog/transfer-learning-the-dos-and-donts-165729d66625 (accessed on 3 September 2023).

81. Liu, X.; Xia, T.; Wang, J.; Yang, Y.; Zhou, F.; Lin, Y. Fully convolutional attention networks for fine-grained recognition. *arXiv* **2016**, arXiv:1603.06765.
82. Zheng, H.; Fu, J.; Mei, T.; Luo, J. Learning multi-attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5209–5217.
83. Li, Z.; Yang, Y.; Liu, X.; Zhou, F.; Wen, S.; Xu, W. Dynamic computational time for visual attention. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 1199–1209.
84. Wang, C.; Cheng, J.; Wang, Y.; Qian, Y. Hierarchical scheme for vehicle make and model recognition. *Transp. Res. Rec.* **2021**, *2675*, 363–376. [CrossRef]
85. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear CNN models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1449–1457.
86. Cui, Y.; Zhou, F.; Wang, J.; Liu, X.; Lin, Y.; Belongie, S. Kernel pooling for convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2921–2930.
87. Cai, S.; Zuo,W.; Zhang, L. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 511–520.
88. Zhu, X.; Bain, M. B-CNN: Branch convolutional neural network for hierarchical classification. *arXiv* **2017**, arXiv:1709.09890.
89. Tan, M.;Wang, G.; Zhou, J.; Peng, Z.; Zheng, M. Fine-grained classification via hierarchical bilinear pooling with aggregated slack mask. *IEEE Access* **2019**, *7*, 117944–117953. [CrossRef]
90. Jaeger, S. A Dual Process Model for Optimizing Cross Entropy in Neural Networks. *arXiv* **2021**, arXiv:2104.13277.
91. Huang, Z.; Zhang, J.; Ma, L.; Mao, F. GTCN: Dynamic network embedding based on graph temporal convolution neural network. In Proceedings of the Intelligent Computing Theories and Application: 16th International Conference, ICIC 2020, Bari, Italy, 2–5 October 2020; pp. 583–593.