Vehicle Make & Model Recognition

by
Zhihao DAI

COP507 Computer Vision & Embedded Systems
Coursework Report

Loughborough University

Jan. 2020

Abstract

In this paper, a general architecture of Vehicle Make & Model Recognition (VMMR) system is designed and implemented. A variety of features extraction methods, dimensionality reduction methods, classification methods are evaluated and compared. The best performance is achieved when Square Mapped Gradients (SMG) is used for feature extraction, Principal Component Analysis (PCA) ($\sigma = 70$) for dimensionality reduction and Support Vector Machine (SVM) for classification. The best accuracy score is 99.62%, precision 99.77%, recall 99.62% and F1-score 99.70%.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

Automatic Number Plate Recognition (ANPR) systems are widely used for policing, traffic monitoring and access control. They have proven to be accurate and efficient under most scenarios. However, ANPR systems are vulnerable to plate cloning, forgery or erosion.

A Vehicle Make & Model Recognition (VMMR) system receives an image of a vehicle as input and outputs the make and model of that vehicle. Such system could strengthen the security of existing ANPR systems by providing a matching between vehicle types and number plates. For example, in access control, if the number plate is not registered under the detected vehicle type, a security warning is raised and manual intervention is required.

In this paper, we design and implement a VMMR system. The input to the system is a cropped frontal image of a vehicle and the output is the make and model of the vehicle.

1.1 Related Work

Due to the significance of VMMR systems, many approaches have been proposed for building VMMR systems in recent years. Petrovic and Cootes [13] extracted simple features such as Sobel Edge Response, Edge Orientation, Square Mapped Gradients from images in the database. Features are then represented and stored either in full dimension or in low dimension through Principal Component Analysis (PCA). Given a new image, the VMMR system predicts the vehicle type by finding the closest match in dot product distance. Their experiments on a dataset of 1132 frontal images of 77 vehicle classes showed that direct matching by Square Mapped Gradients features achieved the lowest vertification error of 3.5%.

AbdelMaseeh et al. [1] observed that unlike most object recognition tasks, VMMR poses a challenge of distingushing between similar classes under the same category (ie. vehicle).

Based on this observation, they proposed the combination of global and local descriptors for VMMR. While global shape descriptors capture differences across categories, local shape and appearance descriptors for segmented regions capture inter-class varieties. An image is matched to the class with the smallest weighted sum of global and local dissimilarity measures.

Pearce and Pears [11] suggested using Harris corner detectors [4] for features extraction and either K-Nearest Neighbour (KNN) or Naive Bayes Classifier for classification in VMMR systems. Local Harris strengths are computed through recursively dividing the image into quadrants and computing the sum of Harris corner response for each quadrant. Such features are then normalised through being devided by the sum of higher level strengths. For an input image of 150 by 150, a feature vector of Locally Normalised Harris Strengths (LNHS) of depth 5 is retrived and only one-twentieth the size of the original Harris corner response. Their experiments on a dataset of 262 frontal images of 74 vehicle classes showed that LNHS with Naive Bayes Classifier achieved the highest accuracy of 96%. Using LNHS as features speeds up the training of a classifier and does not reduce the accuracy.

Siddiqui et al. [14] proposed using Speeded Up Robust Features (SURF) [2] for features extraction and Support Vector Machines (SVM) for classification. Following Sivic and Zisserman's work on Bag-of-Features method [15], a dictionary (bag) of SURF features was constructed using K-Means clustering algorithm. An image can be then transformed into a fixed-length vector of visual words occurances and be fed into a SVM classifier for vehicle type recognition. High accuracy score of 94.84% was obtained on a large dataset of 6601 frontal images of 29 vehicle classes.

Zafar et al. [19] observed that dimensionality reduction methods used in many VMMR systems such as Principal Component Analysis (PCA) enhances the inner-class variance and can lead to miss-classification. In their setting, the raw pixel values of the image is directly projected to low-dimension space through Two Dimensional Linear Discriminant Analysis (2D-LDA) [8]. A match is found by minmizing the Euclidean distance to those in the training images set. The usage of 2D-LDA instead of PCA solves the variance problem by maximizing the ratio of intra-class variance to the inter-class variance. An accuracy score of 91% was obtained on a dataset of 271 frontal images of 25 vehicle classes (8 images per class for training and the rest for validation).

Zafar et al. [18] later proposed using localized Contourlet transform for features extraction, 2D-LDA for dimensionality reduction, and SVM for classification. They reported a boosted accuracy of 96% on the same frontal car images dataset in [19].

Fraz et al. [3] introduced an innovative framework of Mid-Level-Representation of densely sampled features into VMMR. The framework starts by extracting patches around key-points detected by Difference of Gaussians (DoG) detector. For each extracted patch, A set of

Scale-Invariant Feature Transform (SIFT) [9] feature descriptors are computed and reduced dimensionality by PCA. Fisher Vector [6], a Mid-Level-Representation (MLR), for the patch is then generated based on Gaussian Mixture Model (GMM), following Perronnin et al.'s work [12]. Fisher Vector for patches in images within the same class are visual words and collectively form a sub-lexicon. A lexicon of the training set images is essentially a collection sub-lexicons of all classes. Given a new image, the VMMR system extracts patches from the image, assigns each patch to a visual word by Euclidean distance within each sub-lexicon, classifies the image to the class (sub-lexicon) with the highest sum of similarity score of the word-patch matches. Fraz et al. reported an accuracy of 97.60% on the dataset used in [18] and 84.31% on a new dataset. The new dataset, coined 'Loughborough Cars (LC) Dataset', is composed of 1537 frontal images of 75 vehicle classes.

1.2 Dataset

There is a diverse set of datasets for VMMR task and Tafazzoli et al. [17] presented a thorough survey of them. Our proposed system is trained and evaluated on a superset of the dataset in [19, 18, 3] of 530 frontal images from 27 vehicle make and model classes. The dataset is pre-processed to extract Regions of Interest (ROI). See Section 3.1 for more details.

1.3 Comparison to Our Method

In this paper, we make use of Raw Image Pixels, Sobel Edge Response and Square Mapped Gradients following Petrovic and Cootes's [13] work, Locally Normalised Harris Strengths (LNHS) from Pearce and Pears's work [11], and Bag of Speeded Up Robust Features (SURF) from Siddiqui et al.'s work [14] interchangeably in our features extraction module. We use Principal Component Analysis (PCA) for optional dimensionality reduction module and either K-Nearest Neighbour (KNN) or Support Vector Machine (SVM) for classification module.

Despite the simplicity of features computation compared to Mid-Level-Representation in Fraz et al.'s work [3], our method achieves a higher accuracy score of 99% on the dataset.

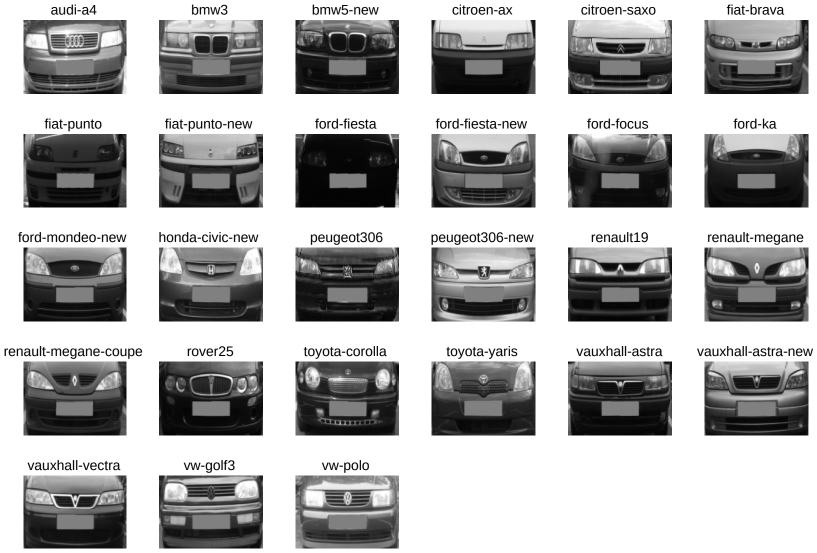

Figure 1.1: Samples of All 27 Vehicle Make and Model Classes in the Dataset.

Chapter 2

System Design

2.1 Assumptions

Several assumptions are made in our design and implementation of VMMR system.

1. The input to the system are frontal images of vehicles.
2. Region of Interest (ROI) can be extracted from the input image based on a pre-labeled bounding box of number plate.
3. The true make and model label for any input image is one of the 27 classes outlined in Section 1.2.

2.2 Block Diagram

A block diagram of our proposed VMMR system is presented in Figure 2.1.

At the first stage, **Features Extraction Module** extracts features from the input image I into a fix-length vector F. **Dimensionality Reduction Module** is optional and reduces the dimensionality of the high-dimensional vector F into low-dimensional vector F'. **Classification Module** is essentially a multi-class classifier that learns to assign incoming feature vectors to their corresponding true labels.

2.3 Features Extraction

A variety of features are interchangeably computed in Features Extraction Module. Among them, Raw Image, Sobel Edge Response (SER), Square Mapped Gradients (SMG) are parameter-free and first proposed to be used for VMMR by Petrovic and Cootes in [13].

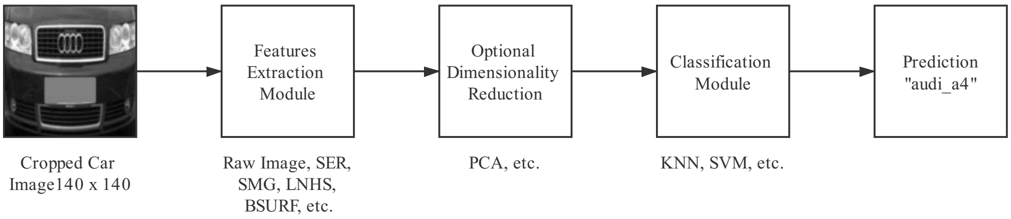

Figure 2.1: Block Diagram of Our Proposed VMMR System.

Locally Normalised Harris Strengths (LNHS) is first proposed by Pearce and Pears in [11]. Bag of Speeded Up Robust Features (BSURF) is first used by Siddiqui et al. for VMMR in [14].

In Section 3.3, the performance of the above features are compared and the effects of parameters of LNHS and BSURF on performance are examined.

2.3.1 Raw Image

Raw Image features are the image pixel values themselves. Hence, $F = I$.

2.3.2 Sobel Edge Response (SER)

SER (also named Sobel Gradient Map) is a map of weighted sum of pixels at 3-by-3 neighborhood.

$$S_{i,j} = \sum_{p=1}^{3} \sum_{q=1}^{3} W_{p,q} I_{i+p-2,j+q-2} \tag{2.1}$$

where in y-direction, $W = W^y$ is specified as follows.

$$W_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.2}$$

In x-direction, $W = W^x = W^y$.

The final feature vector F is the concatanation of S^x and S^y.

$$F = (S^x, S^y) \tag{2.3}$$

2.3.3 Square Mapped Gradients (SMG)

SMG describes the parallel and diagonal components of change in Sobel Edge Response. The parallel component M^p and diagonal componet M^d are computed as follows.

$$M^p_{i,j} = \begin{cases} 0, & \text{if } S^x_{i,j} = 0 \text{ and } S^y_{i,j} = 0 \\ \frac{{S^x_{i,j}}^2 - {S^y_{i,j}}^2}{{S^x_{i,j}}^2 + {S^y_{i,j}}^2}, & \text{otherwise} \end{cases} \tag{2.4}$$

$$M^d_{i,j} = \begin{cases} 0, & \text{if } S^x_{i,j} = 0 \text{ and } S^y_{i,j} = 0 \\ \frac{2 \cdot S^x_{i,j} S^y_{i,j}}{{S^x_{i,j}}^2 + {S^y_{i,j}}^2}, & \text{otherwise} \end{cases} \tag{2.5}$$

The final feature vector F is the concatanation of M^p and M^d.

$$F = (M^p, M^d) \tag{2.6}$$

2.3.4 Locally Normalised Harris Strengths (LNHS)

LNHS is a recurisve structure of Harris corner [4] features representaiton. Given an image, Harris corner strengths $M = \{M_c\}$ are first computed following Noble's suggsted formulation [10].

$$M_c = \frac{I_x^2 I_y^2 - (I_x I_y)^2}{I_x^2 + I_y^2} \tag{2.7}$$

where I_x and I_y are smoothed image derivatives in x-direction and y-direction respectively.

Local Harris corner strengths L are computed through recursively dividing the M into quadrants and computing the sum of Harris corner strengths M_c for each quadrant. Local strengths are then normalised into a vector of LNHS through being devided by the sum of higher level strengths.

For example, for depth of 1, M is first divided into 4 sub-matrices $M_1, M_2, \ldots, M_4$.

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \tag{2.8}$$

The overall LNHS feature vector F is equal to LNHS vector of depth 1, L_1.

$$L_1 = \{\frac{sum(M_i)}{\sum_i sum(M_i)} | i \in \{1, 2, 3, 4\}\} \tag{2.9}$$

For depth of 2, the above 4 sub-matrices $M_1, M_2, \ldots, M_4$ are further divided into quadrants respectively.

$$M_1 = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{1,3} & M_{1,4} \end{bmatrix} \tag{2.10}$$

LNHS vector of depth 2 are computed through dividing the sum of each quadrant by the sum of its higher level quadrant.

$$L_2 = \{\frac{sum(M_{i,j})}{\sum_j sum(M_{i,j})} | i, j \in \{1,2,3,4\}\} \tag{2.11}$$

The overall LNHS feature vector F is concantaion of LNHS vector of depth 1 and 2.

$$F = (L_1, L_2) \tag{2.12}$$

Depth d is the sole parameter in extracting LNHS features. Our implementation of LNHS is based on Harris corner detector by Peter Kovesi[1].

2.3.5 Bag of Speeded Up Robust Features (BSURF)

To compute a BSURF vector, SURF features [2] are first detected and extracted from the image. A dictionary (bag) of visual words can be constructed by grouping all SURF feature descriptors from the dataset into T clusters (viusal words) using K-Means algorithm.

Given any image I, each SURF descriptor extracted can then be assigned to the nearest among the above T visual words. The BSURF feature vector F for I is a vector for visual words occurrences of fixed length T.

The size of the dictionary T is the most important paramter in BSURF and is examined in our experiments.

2.4 Dimensionality Reduction

Features extracted from the images are usually correlated and can be reduced in dimensionality. Such reduction speeds up the training and reduces the complexity of a classifier, which help prevents over-fitting issues.

2.4.1 Principal Component Analysis (PCA)

PCA maps high-dimensional data into a new low-dimensional coordinate system through Singular Value Decomposition (SVD).

[1] https://www.peterkovesi.com/matlabfns/

Durining training, a high-dimensional matrix X is formed, where each row is a feature vector F extracted from a image in the dataset. SVM docomposes X into a product of 3 matrices.

$$X = U\Sigma W^T \tag{2.13}$$

where U is a m-by-m matrix, Σ is a m-by-n diagonal matrix and W^T is the transpose of W, a n-by-n matrix.

X is then reduced in dimensionality to produce X'.

$$X' = XW_L \tag{2.14}$$

where W_L only preserves the first L columns (principal components) of W.

Likewise, for any 1-by-n feature vector F, a new vector F' reduced in dimensionality can be derived.

$$F' = FW_L \tag{2.15}$$

The value of L can be determined by the total percentage σ of the total variance explained by principal components.

2.5 Classification

K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) are used interchangeably in Classification Module. Their perforamnce are examined and compared in Section 3.4.

2.5.1 K-Nearest Neighbour (KNN)

KNN is the simpest machine learning algorithm. A KNN classifier finds K-nearest neighbours for the input vector by Euclidean distance. The classifier then assigns each input vector to the most frequent class among its K-nearest neighbours.

The size of the clusters K is the sole parameter in KNN and its effects on VMMR perforamnce is examined in Section 3.4.

2.5.2 Support Vector Machine (SVM)

In binary classification, SVM classifies each feature vector by learning to map in-coming features vectors into a separate space. In multi-calss scnerario, C binary SVM classifiers are trained using the one-versus-all scheme, where C is the number of vehicle classes.

Chapter 3

Experiments and Results

Experiments for our VMMR system are conducted on a platform with Dual-Core Intel Core i5 (2.9 GHz), 16 GB 1867 MHz DDR3 and MATLAB R2019b.

3.1 Pre-processing

In the pre-processing stage, Regions Of Interest (ROI) are cropped out, converted to grayscale and scaled to a unified resolution of 140-by-140.

3.1.1 Duplicates Removal

The original dataset contains 2117 frontal car images in total. However, for each distinct car in the dataset, there exist multiple variations, which typically include a coloured non-cropping version, a grayscale downsampled version, as well as a grayscale downsampled and ROI-cropped version. Duplicates removal are achieved by preversing the coloured non-cropping version of size 640-by-480 only among those variations. A total of 500 ”original images” are retrieved.

For ”peugeot306” and ”citroen_saxo” classes, however, the coloured non-cropping versions are missing. In that case, the grayscale downsampled and ROI-cropped versions are chosen and both the cropping and converting steps are skipped.

3.1.2 Cropping

In the cropping stage, ROI are cropped out from the ”original images”. Locations of number plates in the ”original images” have already been pre-labeled. In this paper, ROI is defined in terms of the width w and the center (x_c, y_c) of the number plate in the image. Concretely, the

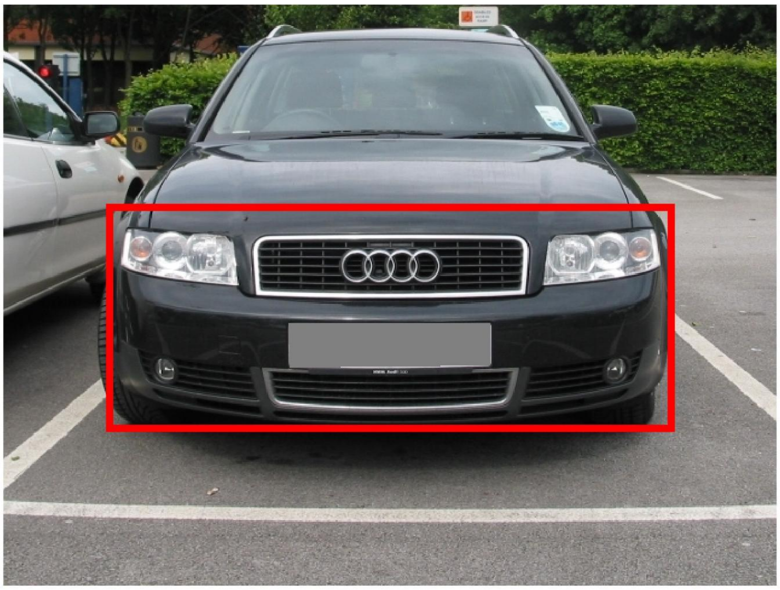

Figure 3.1: An Example of ROI Cropping on a ”Original Image” from ”audi_a4” Class.

rectangle bounding box of ROI is written as $[(x_c - 1.4w, y_c - 0.7w), ((x_c + 1.4w, y_c + 0.4w))]$. An example of ROI cropping is shown in Figure 3.1.

3.1.3 Converting to Grayscale

The cropped RGB images are converttted to grayscale image using Formula 3.1.

$$I = 0.2989R + 0.5870G + 0.1140B \tag{3.1}$$

The pre-processed dataset before scaling can be retrieved from GitHub [1].

3.1.4 Scaling

All images are scaled to a unified resolution of 140-by-140 at runtime. A set of scaled image samples for all classes are presented in Figure 1.1.

[1] https://github.com/daidahao/COP507-Vehicle-Make-Model-Recognition/tree/master/dataset

3.2 Cross-Validation

In our experiments, 5-fold cross-validation is applied to each model-paramter set. Accuracy, precision, recall and f1 scores averaged across class are reported for each set.

3.3 Effects of Features Extraction Methods

To evaluate the effects of features extraction methods, performance of VMMR systems using Raw Image, SER, SMG, LNHS, BSURF as features are evaluated and compared in Table 3.1. In addition, LNHS and BSURF are evaluated under a variety of parameter set. The default classification method is SVM and dimensionality reduction is skipped.

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Raw Image	90.00	89.61	89.87	89.74
SER	32.08	48.63	28.04	35.57
SMG	**99.62**	**99.70**	**99.63**	**99.66**
LNHS ($d = 1$)	3.77	3.47	4.80	4.03
LNHS ($d = 2$)	36.23	35.46	34.53	34.99
LNHS ($d = 3$)	88.68	89.45	89.14	89.29
LNHS ($d = 4$)	99.06	99.08	98.99	99.03
LNHS ($d = 5$)	**99.62**	99.62	99.58	99.60
LNHS ($d = 6$)	99.43	99.37	99.42	99.39
LNHS ($d = 7$)	96.23	97.44	96.62	97.03
BSURF ($T = 500$)	84.34	85.38	84.01	84.69
BSURF ($T = 1000$)	86.98	88.38	87.04	87.70
BSURF ($T = 2000$)	90.00	91.26	90.25	90.75
BSURF ($T = 4000$)	92.45	93.83	92.85	93.34
BSURF ($T = 8000$)	93.02	94.64	93.15	93.89
BSURF ($T = 16000$)	93.96	95.77	94.69	95.23
BSURF ($T = 32000$)	93.96	95.62	93.55	94.57

Table 3.1: Performance of VMMR System Using Different Features Extraction Methods.

SER achieves the poorest performance among three simplest features with accuracy score of 32.08%. The reason is that edge response are sensitive to noise and directional changes of edges. On the other hand, SMG performs the best among all features with accuracy score of 99.62%. The reason behind that is that SMG describes the parallel and diagonal components of change in SER and is robust to noise and directional changes of edges.

LNHS with depth of $d = 5$ and length of 1364 achieves comparable performance to SMG with length of 39200, having the same accuracy and a lower precision and recall scores. Deeper LNHS features, however, decrease the performance in all merits. One explanation for that would be shallower LNHS normalizes Harris corner strengths in a more robust manner. The deepest quardant of LNHS with $d = 5$ is normalized by the sum of Harris strengths on 76.56 pixels on average, compared to 4.79 pixels when $d = 7$.

BSURF features achieve poorer performance even when the length of feature vector is 32000 and close to SMG.

SMG, which outperforms all the other features in all merits, is the default features extraction method in the following sections.

3.4 Effects of Classification Methods

Performance of VMMR systems using KNN and SVM as classifier are evaluated and compared in Table 3.2. In addition, KNN are evaluated under a different values of K. The default classification method is SMG and dimensionality reduction is skipped.

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
KNN ($K = 1$)	99.43	99.69	99.59	99.64
KNN ($K = 3$)	99.43	99.67	99.38	99.52
KNN ($K = 5$)	**99.62**	99.76	99.52	99.64
KNN ($K = 7$)	99.43	99.56	99.29	99.42
KNN ($K = 9$)	98.87	99.20	98.78	98.99
SVM	**99.62**	**99.77**	**99.63**	**99.70**

Table 3.2: Performance of VMMR System Using Different Classification Methods.

KNN model performs best with accuracy of 99.62% when $K = 5$. There are 2 main reasons for that. First, a larger K value means more neighbours need to be found to support a prediction and thus the classifier is more robust. Second, some vehicle classes have fewer than 10 images durning training. Given a image from one of those classes, neighbours of the same class could be inadequate to form a majority.

SVM achieves the best performance in all merits with accuracy of 99.62% and F1-Score of 99.70%. Therefore, SVM is the default classification method for the following sections.

3.5 Effects of Dimensionality Reduction Methods

Performance of VMMR systems whether using dimensionality reduction are evaluated and compared in Table 3.3. In addition, PCA is evaluated under different σ values.

	Length	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Skipped	39200	**99.62**	**99.77**	99.63	**99.70**
PCA ($\sigma = 60$)	203	**99.62**	99.66	**99.70**	99.68
PCA ($\sigma = 70$)	270	**99.62**	**99.77**	99.63	**99.70**
PCA ($\sigma = 80$)	344	**99.62**	**99.77**	99.63	**99.70**
PCA ($\sigma = 90$)	429	**99.62**	**99.77**	99.63	**99.70**
PCA ($\sigma = 95$)	476	**99.62**	**99.77**	99.63	**99.70**
PCA ($\sigma = 99$)	581	**99.62**	**99.77**	99.63	**99.70**

Table 3.3: Performance of VMMR System Using Optional Dimensionality Reduction Method.

The introduction of dimensionality reduction into our system does not have any effect on the performance when σ for PCA is above 70. Using a feature vector of length 270 after PCA produces the same performance of using a full SMG feature vector of length 39200. The reason is that SMG on each pixel is highly correlated to neighbours and thus can be dramaitically reduced in dimensionality.

3.6 Best Performance

The best performance of our proposed VMMR system is achieved on the dataset when SMG is used for feature extraction, PCA ($\sigma = 70$) for dimensionality reduction and SVM for classification. The model produces an accuracy score of 99.62%, precision of 99.77%, recall of 99.62% and F1-score of 99.70%. A confusion matrix of the model is drawn in Figure 3.2, where predictions are made in 5-fold cross validation scheme.

Only 2 misclassifications are made on the dataset, as presented in Figure 3.3.

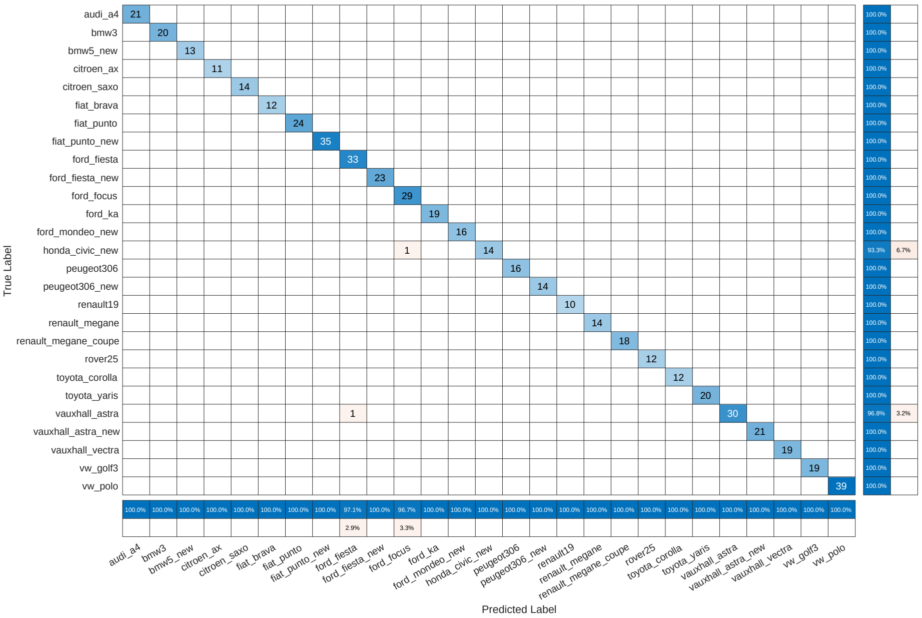

Figure 3.2: Confusion Matrix on Corss-Validated Dataset of Our Poposed VMMR System (SMG + PCA + SVM).

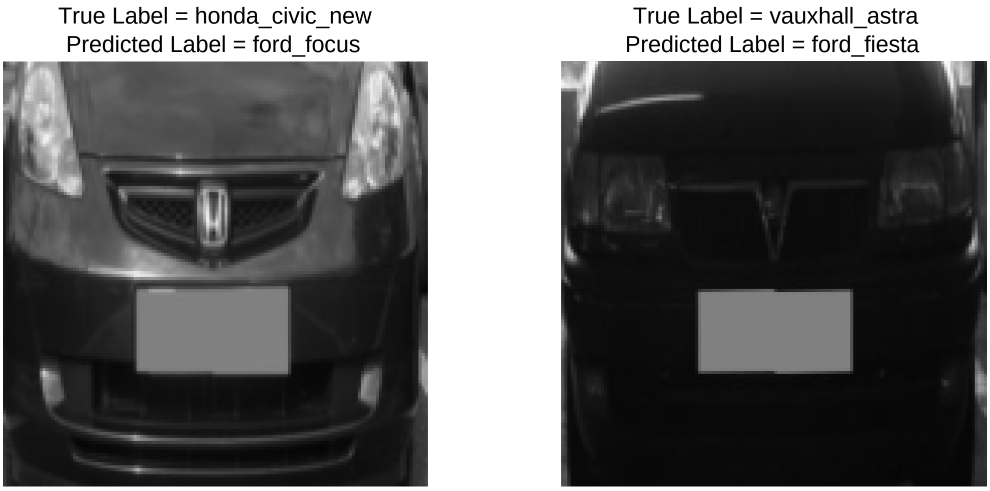
True Label = honda_civic_new
Predicted Label = ford_focus
True Label = vauxhall_astra
Predicted Label = ford_fiesta

Figure 3.3: Misclassifications on Corss-Validated Dataset of Our Poposed VMMR System (SMG + PCA + SVM).

Chapter 4

Convolution Neural Network Model

4.1 Architecture

For VMMR tasks, a simple Convolution Neural Network (CNN) model is also considered, as presented in 4.1.

The input image of size $(140, 140, 1)$ is fed into the first convolution layer and activated by a ReLU function. The convolution layer has 32 filters of $(5, 5)$, stride of 2 and produces an output of $(70, 70, 32)$. The layer picks up simple features such as edges, colours, curves from the image. It is followed by a max pooling layer with filter size of $(3, 3)$ and stride of 2 to reduce dimensionality. The output has a size of $(35, 35, 32)$ and is fed into the second convolution layer.

The second convolution layer has 32 filters of $(3, 3)$, stride of 2 and produces an output of $(18, 18, 32)$. The layer picks up more specific features such as squares, circles and triangles from the image. It is activated by a ReLU function and followed by a max pooling layer with filter size of $(3, 3)$ and stride of 2 to reduce dimensionality. The output has a size of $(9, 9, 32)$ and is fed into the third convolution layer.

The third convolution layer also has 32 filters of $(3, 3)$, stride of 1 and produces an output of $(9, 9, 32)$. This layer picks up high-level features such as headlights, front beams, logos from the image. It is activated by a ReLU function and followed by a max pooling layer with filter size of $(3, 3)$ and stride of 2 to reduce dimensionality. The output has a size of $(5, 5, 32)$ and is fed into the first fully connected layer.

The first fully connected layer has 256 hidden units and is activated by a ReLU function. This layer introduces more complexity into the network and allows high-level features located

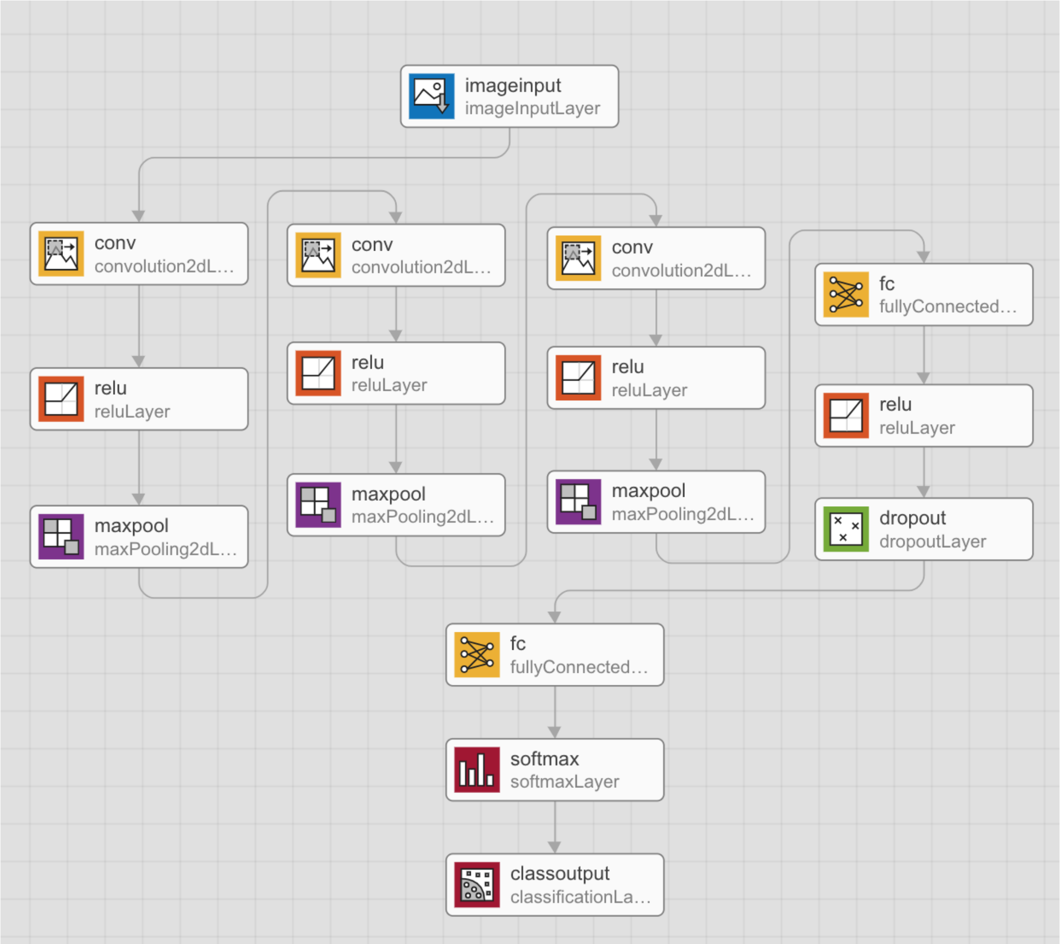

Figure 4.1: Architecture of Our Proposed CNN.

at different locations in the image to be combined. It is followed by a dropout layer with rate of 0.5, which introduces regularization into the model and prevents over-fitting. The output is fed into the second fully connected layer.

The first fully connected layer has 27 hidden units, the same size as the number of vehicle classes, and is activated by a Softmax function for the final classification output.

4.2 Overfitting Issues

Our proposed CNN model has $213,467$ parameters, much larger to other machine learning models used previously. Thus, the model is prune to overfitting. The following strategies can be adopted to prevent overfitting in the model.

1. Introduce a regularization term into the loss function. That is, the new loss function $L_{new}(W) = L(W) + \lambda R(W)$, where $R(W)$ is a regularization term for the weights W. Typically, $R(W) = W^2$ and it forces the network to make full use of the inputs at each layer.

2. Introduce a dropout layer in between different layers. The dropout layer randomly drops a percentage of its input, thus forcing the next layer to not rely on a small set of its inputs.

3. Early stopping. When the model is being trained on the dataset, the decrease in loss on the training set and the increase in loss on the validation set is a sign of overfitting. In such cases, the model should be stopped training to prevent further overfitting.

4.3 Insufficient Dataset

Given the small size of our dataset (530 images in total), the average number of parameters per image is 403. Such amount of paramters signal that the size of our dataset is insufficient. In the case that time and resouces are limited for collecting new data, several alternative solutions could be considered.

1. Apply data augmentation to the dataset to generate more images for training. The operations for data augmentation include rotation, scaling, shearing and translation.

2. Use pre-trained CNN models such as AlexNet[7], GoogLeNet[16] and ResNet[5] in a transfer learning scheme. Since our dataset is small, it is difficult to fine-tune the whole network. Therefore, only the last few layers of pre-trained CNN networks are replaced and re-trained.

Chapter 5

Conclusions

In this paper, a general architecture of VMMR system is proposed. Three main modules of the proposed system are Features Extraction Module, Dimensionality Reduction Module (optional) and Classification Module respectively.

A variety of features extraction methods are compared and it is concluded that SMG achives the best performance among all. Using SVM in Classification Module produces better performance than KNN under a variety set of K values. In addition, applying PCA as dimensionality reduction method to the feature vector does not affect the performance at all.

The best performance is achieved on the dataset when SMG is used for feature extraction, PCA ($\sigma = 70$) for dimensionality reduction and SVM for classification. The model produces an accuracy score of 99.62%, precision of 99.77%, recall of 99.62% and F1-score of 99.70%. Only 2 misclassifications are made.

References

[1] Meena AbdelMaseeh, Islam Badreldin, Mohamed F Abdelkader, and Motaz El Saban. Car Make and Model Recognition Combining Global and Local Cues. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 910–913. IEEE, 2012.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[3] Muhammad Fraz, Eran A Edirisinghe, and M Saquib Sarfraz. Mid-Level-Representation based Lexicon for Vehicle Make and Model Recognition. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 393–398. IEEE, 2014.

[4] Christopher G Harris, Mike Stephens, et al. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, volume 15, pages 10–5244. Citeseer, 1988.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Tommi Jaakkola and David Haussler. Exploiting Generative Models in Discriminative Classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 487–493, 1999.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[8] Ming Li and Baozong Yuan. 2D-LDA: A Statistical Linear Discriminant Analysis for Image Matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[9] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[10] Julia Alison Noble. *Descriptions of Image Surfaces.* PhD thesis, University of Oxford, 1989.

[11] Greg Pearce and Nick Pears. Automatic Make and Model Recognition from Frontal Images of Cars. In *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 373–378. IEEE, 2011.

[12] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156. Springer, 2010.

[13] Vladimir S Petrovic and Timothy F Cootes. Analysis of Features for Rigid Structure Vehicle Type Recognition. In *British Machine Vision Conference (BMVC)*, volume 2, pages 587–596, 2004.

[14] Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche. Real-Time Vehicle Make and Model Recognition Based on a Bag of SURF Features. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3205–3219, 2016.

[15] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, page 1470. IEEE, 2003.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper With Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[17] Faezeh Tafazzoli, Hichem Frigui, and Keishin Nishiyama. A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1–8, 2017.

[18] Iffat Zafar, Eran A Edirisinghe, and B Serpil Acar. Localized Contourlet Features in Vehicle Make and Model Recognition. In *Image Processing: Machine Vision Applications II*, volume 7251, page 725105. International Society for Optics and Photonics, 2009.

[19] Iffat Zafar, Eran A Edirisinghe, S Acar, and Helmut E Bez. Two Dimensional Statistical Linear Discriminant Analysis for Real-Time Robust Vehicle Type Recognition. In *Real-Time Image Processing 2007*, volume 6496, page 649602. International Society for Optics and Photonics, 2007.