ISSN: 2278-4632
Vol-10 Issue-5 No. 7 May 2020

MULTI-TRAFFIC SCENE PERCEPTION BASED ON SUPERVISED LEARNING

Mrs.V.Swathi[1] B.Joseph Emmanuel[2] G.Nikhil[2] G.Shiva Sai Reddy[2] P.Sai Mahesh[2]
Assistant Professor[1] and B.Tech Students[2], Department of Computer Science and Engineering,
Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India

Abstract:

Traffic accidents are particularly serious at a rainy day, night without street lamp, overcast, rainy night, foggy day and many other low visibility conditions.Firstly, underlying visual features are extracted from multi-traffic scene images, and then the feature was expressed as an eight-dimensions feature matrix.Secondly, five supervised learning algorithms are used to train classifiers.

1.Introduction

1.1. Motivation

Traffic accidents are particularly serious at a rainy day, night without street lamp, overcast, rainy night, foggy day and many other low visibility conditions. Firstly, underlying visual features are extracted from multi-traffic scene images, and then the feature was expressed as an eightdimensions feature matrix. Secondly, five supervised learning algorithms are used to train classifiers.

1.2 Problem Definition

Little work has been done on weather related issues for in-vehicle camera systems so far. Payne and Singh propose classifying indoor and outdoor images by edge intensity. Lu *et al.* propose a sunny and cloudy weather classification method for single outdoor image. Lee and Kim propose intensity curves arranged to classfy four fog levels by a neural network. Zheng *et al.* present a novel framework for recognizing different weather conditions. Milford *et al.* present vision-based simultaneous localization and mapping in changing outdoor environments. Detecting critical changes of environments while driving is an important task in driver assistance systems. Liu *et al.* propose a vision- based skyline detection algorithm under image brightness variation. Fu *et al.* propose automatic traffic data collection under varying lighting conditions. Fritsch *et al.* use classifiers for detecting road area under multi-traffic scene. Wang *et al.* propose a multi-vehicle detection and tracking system and it is evaluated by roadway video captured in a variety of illumination and weather conditions. Satzoda and Trivedi propose a vehicle detection method on seven different datasets that captured varying road, traffic, and weather conditions.

1.3 Objective Of Project

Considering the fact that e-commercebanking and business related highly confidential and valuable information communicated within the network, it is needless to mention the importance of network traffic analysis to attain proper information security. Network traffic analysis and prediction resembles a proactive approach rather than reactive, where network is monitored to ensure that security breaches do not occur within network. The network traffic analysis is a significant stage for developing successful preventive congestion control schemes and to find out normal and malicious packets. These schemes target to avoid network congestion by distributing the network resources with respect to the forecasted traffic. The predictability of network traffic is of important benefits in many areas, such as dynamic bandwidth allocation, network security and network planning and predictive congestion control and so on. We can identify two categories of predictions: long and short period's predictions.

1.4 Limitations Of Project

Traffic prediction for long period gives a detailed forecasting of traffic models to evaluate future capacity requirements, and therefore permits for more minute planning and better decisions. Short period prediction (milli-seconds to minutes) is linked to dynamic resource allotment. It can be used to improve Quality of Service (QoS) mechanisms as well as for congestion control and for optimal resource management. It can also be used for routing packets. Several different techniques including time series models, modern data mining techniques, soft computing approaches, and neural networks are used for network traffic analysis and prediction. This paper presents a review of several techniques proposed, used and practiced for network traffic analysis and prediction. The distinctiveness and restrictions of previous researches are discussed and typical features of these network traffic analysis and prediction are also summarized. The remaining paper is organized as follows.

2. Existing System

Present vision driver assistance systems are designed to perform under goodnatured weather conditions. Highway traffic accidents bring huge losses to people's lives and property.The advanced driver assistance systems (ADAS) play a significant role in reducing traffic accidents .

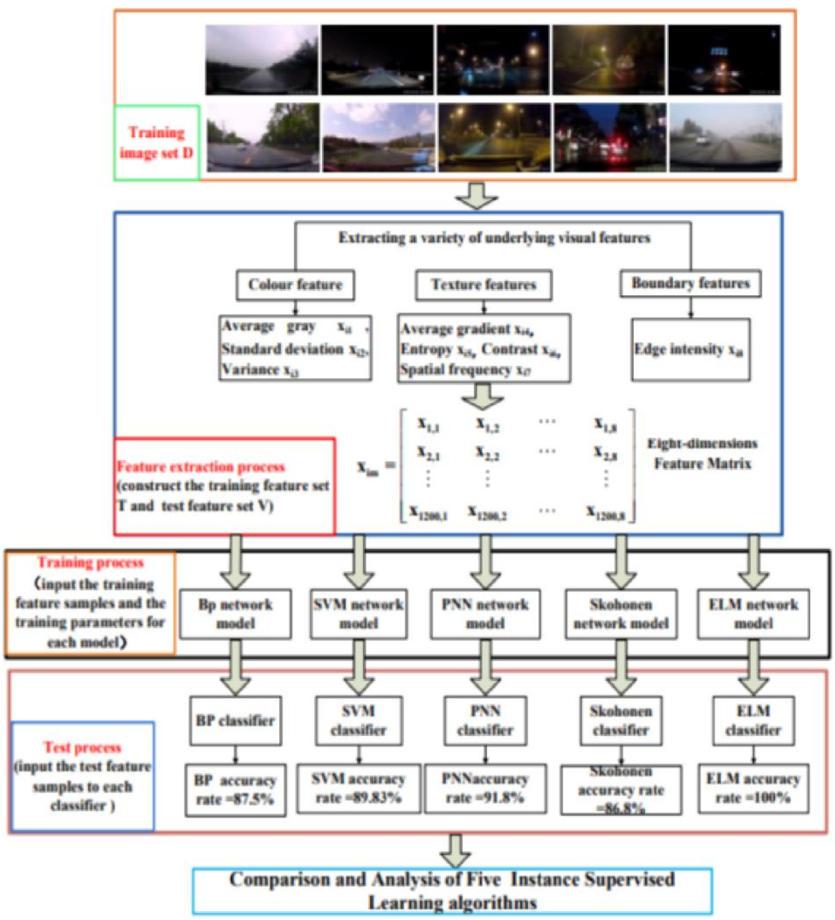
3. Proposed System

Multi-traffic scene perception of complex weather condition is a piece of valuable information for assistance systems. Based on different weather category, specialized approaches can be used to improve visibility. This will contribute to expand the application of ADAS.

Image feature extraction is the premise step of supervised learning. It is divided into global feature extraction and local feature extraction. In the work, we are interested in the entire image, the global feature descriptions are suitable and conducive to understand complex image. Therefore, multi-traffic scene perception more concerned about global features, such as color distribution, texture features.

Image feature extraction is the most important process in pattern recognition and it is the most efficient way to simplify high-dimensional image data. Because it is hard to obtain some information from the M _ N _ 3 dimensional image matrix. Therefore, owing to perceive multitraffic scene, the key information must be extracted from the image

4.Architecture



ISSN: 2278-4632
Vol-10 Issue-5 No. 7 May 2020

5.Implementation

Import libraries: In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. Various libraries that we will use for prediction

import numpy as np import random import pickle import matplotlib.pyplot as plt import cv2 from sklearn.utils import shuffle import tensorflow as tf from tensorflow.contrib.layers import flatten import time import csv

5.1. Load Pickle Data

The pickled data is a dictionary with 4 key/value pairs:

- 'features' is a 4-D array containing raw pixel data of the traffic sign images (num examples, width, height, channels).
- 'labels' is a 1-D array containing the label/class ID of the traffic sign. The file signnames.csv contains id -> name mappings for each ID.
- 'sizes' is a list containing tuples (width, height) representing the original width and height of the image.
- 'coords' is a list containing tuples (x1, y1, x2, y2) representing coordinates of a bounding box around the sign in the image. These coordinates assume the original images. The pickled data contains resized versions (32x32) of these images.

Here is a basic quantitative and visual summary of this data.

Number of training, validation, and test examples, image resolution and color channels, and number of classification classes

Here is a basic visualization of the content of the GTSRB dataset. Execute the cell multiple times to load new randomly selected images from the dataset. Change the grid_m and grid_n values and if needed the figsize to change the display.

we can see a histogram of the number of tracks per classification category in the training dataset, illustrating a huge variance.

Some categories such as "0 - Speed limit (20km/h)" and "37 - Go straight or left" make up only around 0.6% and 0.5% of the training data and test data, respectively, while others such as "2 - Speed limit (50km/h)" make up around 6.1% and 5.9% - a difference of factor 10. With as few as 7 tracks of training examples for some classes (i.e. 7 unique physical traffic sign instances), this is very little data to train on for some classes.

5 .2.Architecture And Configuring Testing

Now let's get to the actual model testing and training. Below I'll set up a test pipeline to test various network configurations. Thanks to the paper of Sermanet et al. I already have a starting point: I know that four layers (2 convolutional and 2 fully connected) are sufficient to achieve accuracies around 99%. Hence, my architecture experimentation will be limited to trying out different depths for the two convolutional layers.

The basic network that will be tested has two convolutional layers followed by two fully connected layers including the classifier, which uses softmax. Batch normalization and dropout (keep_rate=0.5) are performed on each hidden layer, and L2 regularization is added.

The parameters that will be varied are learning rate, L2 regularizaDon rate, and number of feature maps in the convoluDonal layers of the network. These tests will serve as a basis for further tweaking and will of course only provide a rough orientation: I only trained for 10 epochs due to limited computational resources, and I'm not testing nearly as many hyper parameter configurations and architectures as I'd like to. Nonetheless these tests provide a helpful basis.

36 configurations will be tested in total, namely all combinations of the following:

- 6 different learning rates, which are roughly by a factor of 3 apart from each other
- 3 different L2 regularization rates
- 2 alternatives for the number of feature maps in the two convolutional layers: 64-108 and 108-200

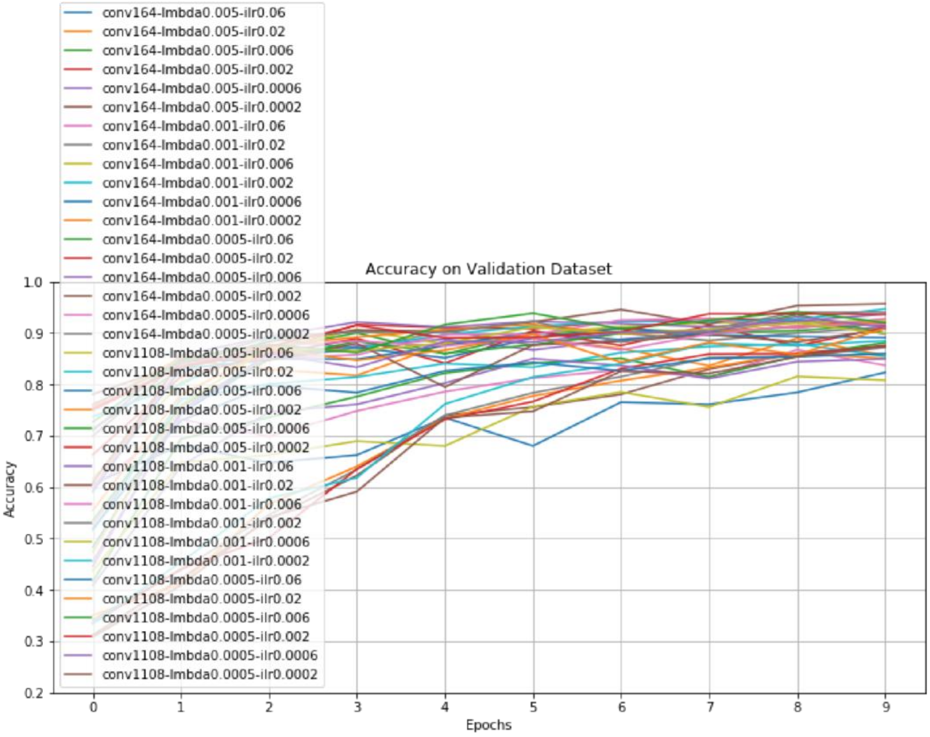
Below you find the top k configurations, ranked by their average accuracy over the last 4 epochs.

The result indicates the following:

- Learning rates in the region around 0.02-0.06 seem to perform best, regardless of the regularization rate or convolutional layer depth.
- A regularization factor of 0.0005 performs best - the smallest out of the 3 tested values. This indicates that an even smaller regularization rate might perform even better, or even no L2 regularization at all. This seems slightly counter-intuitive: The model seems to constantly overfit (the training accuracy is consistently extremely high during training), yet less L2 regularization leads to a better ability to generalize.
- It is inconclusive from these tests whether the larger or smaller number of convolutional feature maps performs better. The network with the fewer trainable parameters shows a

smoother learning curve on average, but I wouldn't read too much into that since we're only looking at 10 datapoints. Since both alternatives performed roughly equally well, in anticipation of augmenting the training dataset we'll go with the larger alternative.

- Below you find the results of the above tests. Granted, you can't see very much on a chart with 36 lines. Further below I've picked the top configurations among these 36 and displayed them in isolation.

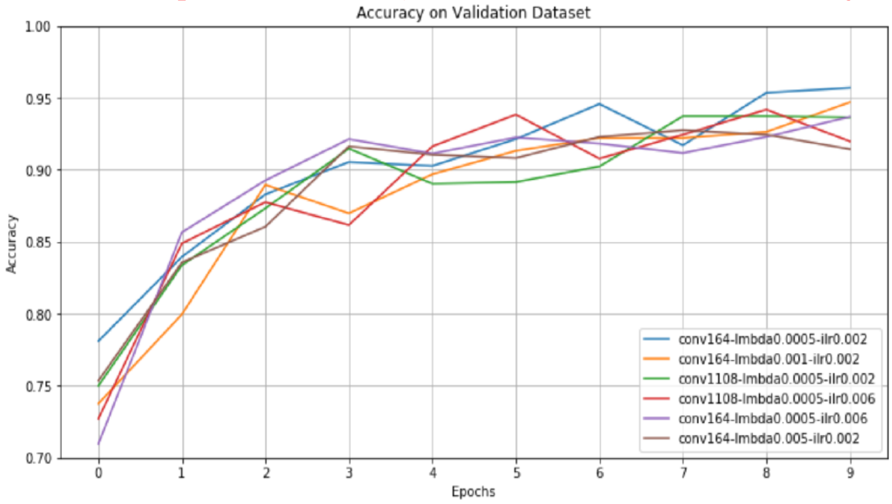


Below you find the top k configurations, ranked by their average accuracy over the last 4 epochs.

The result indicates the following:

- Learning rates in the region around 0.02-0.06 seem to perform best, regardless of the regularization rate or convolutional layer depth.
- A regularization factor of 0.0005 performs best - the smallest out of the 3 tested values. This indicates that an even smaller regularization rate might perform even better, or even no L2 regularization at all. This seems slightly counter-intuitive: The model seems to constantly overfit (the training accuracy is consistently extremely high during training), yet less L2 regularization leads to a better ability to generalize.
- It is inconclusive from these tests whether the larger or smaller number of convolutional feature maps performs better. The network with the fewer trainable parameters shows a smoother learning curve on average, but I wouldn't read too much into that since we're only

ISSN: 2278-4632
Vol-10 Issue-5 No. 7 May 2020



looking at 10 datapoints. Since both alternatives performed roughly equally well, in anticipation of augmenting the training dataset we'll go with the larger alternative.

5.3. Final Model Training

I'll pick a winner from the above experiments and see how much more I can get out of it by training on the expanded training dataset (note that the experiments above were conducted with the original training dataset), by training longer, playing with dropout and initialization methods, and by tweaking the learning and L2 regularization rates yet a bit more.

I'll use the following configuration from the first testing phase above as the candidate for the final model:

- 1 input channel, i.e. grayscale input
- 2 convolutional layers, 108-200
- Convolutional filter size of 5x5 with a stride of 1 in each direction for both convolutional layers
- Max-pooling layers after each of the convolutional layers with a ksize and stride of 2 in each direction
- 2 fully connected layer with 100-43, the second going into a softmax classifier
- Batch normalization before the non-linearities of each layer (not before softmax)
- ReLU non-linearities for the first three layers
- Initial learning rate of 0.002, decaying stepwise exponentially with a decay rate of 0.9
- L2 regularization rate of 0.0005
- Weights initialization according to He et al.

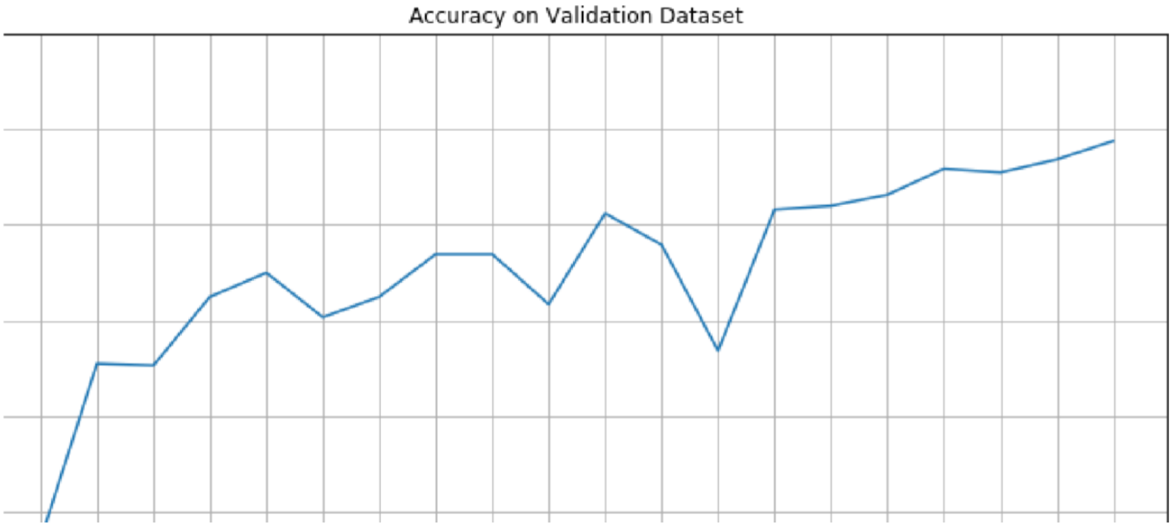
The weights are being initialized randomly according to He et al., i.e. using a normal distribution with mean zero and a standard deviation of 1/sqrt(n_inputs/2). This is identical to Xavier initialization except for the factor 1/2 in the square root that He et al. compute as needed to adjust for the use of ReLUs as non-linearities.

The winner of my final tests is the model_02 that you can see above. It does not use dropout and was trained on the expanded training dataset. Using dropout probably wouldn't have harmed, but it didn't seem necessary either.

Final small adjustments of the learning and L2 regularization rates yielded inconclusive results, indicating that an initial learning rate of 0.002 and an L2 regularization rate of 0.0005 are good values for this particular model. Training for longer than 20 epochs - I tried 30 and 40 - didn't yield clear improvements. Whether the weights were initialized with a constant global standard deviation of 0.01 or with the method proposed by He et al. didn't appear to make a structural difference - this model is probably too shallow for it to make much of a difference.

Below you can see how the accuracy on the validation dataset evolves, and the accuracy on the test dataset.

ACCURACY ON FINAL DATA SET=0.978



6 .Conclusion

Weather recognition based on road images is a brand-new and challenging subject, which is widely required in many fields. Hence, research of weather recognition based on images is in urgent demand, which can be used to recognize the weather conditions for many vision systems. Classification is a methodology to identify the type of optical characteristics for vision enhancement algorithms to make them more efficient.

In this paper, eight global underlying visual features are extracted and five supervised learning algorithms are used to perceive multi-traffic road scene. Firstly, our method extracts colour features, texture features and boundary feature which are used to evaluate the image quality. Thus, the extracted features are more comprehensive. Secondly, the ten categories traffic scene image are marked as labels 1-10. Owing to the category label represents the whole image, there is no need to mark the specific area or key point of image.

Thirdly, by using of five supervised learning that mentioned in Section IV, we can greatly simplify the manual annotation process of feature sample and improve the classifier efficiency. At last, experiments and comparisons are performed on large datasets to verify the effectiveness of the proposed method in Section V. It proved that the proposed eight features not only can accurately describe image characteristics, but also have strong robustness and stability at the complex weather environment and the ELM algorithm is superior to other algorithms. In the future, the proposed algorithms will need to be further verified by the larger image set. Integrated learning is a new paradigm in machine learning field. It is worth to be studied improve the generalization of a machine learning system. And visual image enhancement algorithms in fog and night time applied to general image are worth to be further studied.

References

[1] A. Payne and S. Singh, ``Indoor vs. outdoor scene classification in digital photographs,'' *Pattern Recognit.*, vol. 38, no. 10, pp. 1533_1545, Oct. 2005.

[2] C. Lu, D. Lin, J. Jia, and C.-K. Tang, ``Two-class weather classification,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2510_2524, Dec. 2017.

[3] Y. Lee and G. Kim, ``Fog level estimation using non-parametric intensity curves in road environments,'' *Electron. Lett.*, vol. 53, no. 21, pp. 1404_1406, Dec. 2017.

[4] C. Zheng, F. Zhang, H. Hou, C. Bi, M. Zhang, and B. Zhang, ``Active discriminative dictionary learning for weather recognition,'' *Math. ProblemsEng.*, vol. 2016, Mar. 2016, Art. no. 8272859. [5] M. Milford, E. Vig, W. Scheirer, and D. Cox, ``Vision-based simultaneous localization and mapping in changing outdoor environments,'' *J. FieldRobot.*, vol. 31, no. 5, pp. 814_836, Sep./Oct. 2014.

[6] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, ``Automatic change detection of driving environments in a vision-based driver assistance system,'' *IEEETrans. Neural Netw.*, vol. 14, no. 3, pp. 646_657, May 2003.

[7] Y. J. Liu, C. C. Chiu, and J. H. Yang, ``A robust vision-based skyline detection algorithm under different weather conditions,'' *IEEE Access*, vol. 5, pp. 22992_23009, 2017.

[8] T. Fu, J. Stipancic, S. Zangenehpour, L. Miranda-Moreno, and N. Saunier, ``Automatic traffic data collection under varying lighting and temperature conditions in multimodal

environments: Thermal versus visible spectrum video-based systems," *J. Adv. Transp.*, vol. 2017, Jan. 2017, Art. no. 5142732.

[9] J. Fritsch, T. Kuhnl, and F. Kummert, ``Monocular road terrain detection by combining visual and spatial information," *IEEE Trans. Intell. Transp.Syst.*, vol. 15, no. 4, pp. 1586_1596, Aug. 2014.

[10] K.Wang, Z. Huang, and Z. Zhong, ``Simultaneous multi-vehicle detection and tracking framework with pavement constraints based on machine learning and particle _lter algorithm," *Chin. J. Mech. Eng.*, vol. 27, no. 6, pp. 1169_1177, Nov. 2014.

[11] R. K. Satzoda and M. M. Trivedi, ``Multipart vehicle detection using symmetry-derived analysis and active learning," *IEEE Trans. Intell.Transp. Syst.*, vol. 17, no. 4, pp. 926_937, Apr. 2016.

[12] M. Chen, L. Jin, Y. Jiang, L. Gao, F.Wang, and X. Xie, ``Study on leading vehicle detection at night based on multisensor and image enhancement method," *Math. Problems Eng.*, vol. 2016, Art. no. 5810910, Aug. 2016.

[13] Y. Xu, J.Wen, L. Fei, and Z. Zhang, ``Reviewof video and image defogging algorithms and related studies on image restoration and enhancement," *IEEE Access*, vol. 4, pp. 165_188, 2016.

[14] R. Gallen, A. Cord, N. Hautière, É. Dumont, and D. Aubert, ``Nighttime visibility analysis and estimation method in the presence of dense fog," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 310_320, Feb. 2015.

[15] D. Gangodkar, P. Kumar, and A. Mittal, ``Robust segmentation of moving vehicles under complex outdoor conditions," *IEEE Trans. Intell. Transp.Syst.*, vol. 13, no. 4, pp. 1738_1752, Dec. 2012.

[16] H. Kuang, X. Zhang, Y. J. Li, L. L. H. Chan, and H. Yan, ``Nighttime vehicle detection based on bio-inspired image enhancement and weighted score-level feature fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 927_936, Apr. 2017. Now let's look at what really maIers, performance on the test dataset: