Traffic Sign Detection and Recognition under Challenging Conditions

Shivam Raval (201501088), Jay Mohta (201501036),
Om Thakkar (201501109), Anushree Rankawat (201501007),
Himol Shah (201501098), Faraaz Kakiwala (201501043)

Computer Vision (CSP502)
School of Engineering and Applied Science,
Ahmedabad University

November 13, 2018

Outline

Outline

Background and Motivation

- Robust and reliable traffic sign detection and recognition becomes necessary to bring autonomous vehicles onto our roads.
- A few challenges in automating the process are signs with the same general meaning, such as the various speed limits, have a common general appearance, leading to subsets of traffic signs that are very similar to each other.
- Illumination changes, partial occlusions, rotations, and weather conditions further increase the range of variations in visual appearance a classifier has to cope with.

Outline

Problem Statement

- Given a very novel data-set consisting of diverse video sequences, achieve robust as well as rapid detection and recognition of traffic signs, resulting into improved accuracy.

Outline

Literature Review : Datasets

- German Traffic Sign Recognition Benchmark (GTSRB) [1]
- Challenging Real and Unreal Datasets (CURE)-TSD [2]
- Tencent 100k [3]

---

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural Networks, vol. 32, pp. 323-332, 2012.

[2] D. Temel and G. AlRegib, Traffic Signs in the Wild: Highlights from the IEEE Video and Image Processing Cup 2017 Student Competition [SP Competitions], in IEEE Signal Processing Magazine, vol. 35, no. 2, pp. 154-161, March 2018.

[3] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," 2016 IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2110-2118, 2016.

Outline

Literature Review : Current State of Art

- Traffic Sign Detection with Multi-Scale Convolutional Networks [4]
- Traffic Sign Classification Using Deep Inception Based Convolutional Networks [5]
- Traffic Sign Detection and Classification in Wild [6]
- Novel Deep Learning Model for Traffic Sign Detection using Capsule Networks [7]

---

[4] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," Neural Networks (IJCNN), 2011 Int. Jt. Conf., pp. 2809-2813, 2011.

[5] M. Haloi, "Traffic Sign Classification Using Deep Inception Based Convolutional Networks," pp. 11-15, 2015.

[6] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2110-2118, 2016.

[7] A. D. Kumar, "Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks. (arXiv:1805.04424v1 [cs.CV])," 2018.

Outline

Outline

Block Diagram of Overall Approach

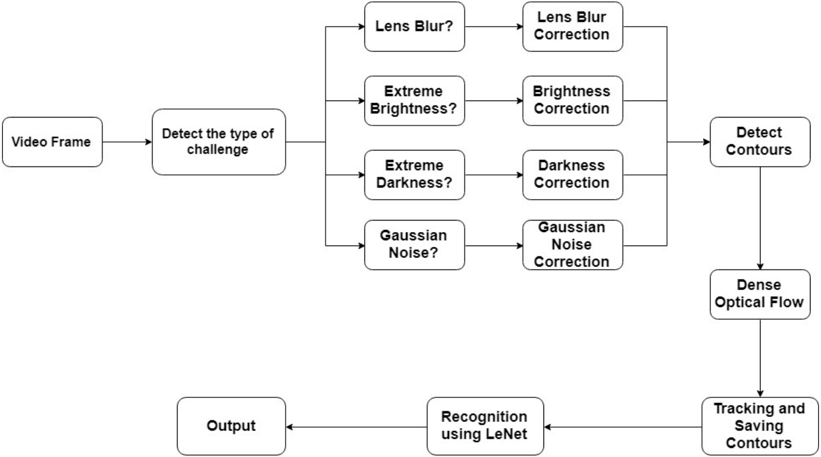

Figure: Block Diagram

Outline

Outline

General Approach for Traffic Sign Detection

- Extract video frames from video sequences, and detect the type of challenge based on the input frame (e.g., blur, noise, brightness, darkness, etc.) (How to detect the type of challenge has been explained in subsequent slides)
- Apply respective correction techniques in order to remove noise and other unnecessary elements from the frame before classifying it.
- Detect Contours (method explained in subsequent slides)
- Classification of traffic sign using LeNet architecture.

Outline

Detecting the type of challenge

- Detect whether image is blur?
  - Using Laplacian Kernel - We take a single channel of an image and convolve it with a 3 x 3 Laplacian Kernel. Then, we obtain variance of the response. If the variance is very low, it implies that there is a tiny spread of responses, indicating there are very little edges in the image. As we know, the more an image is blurred, the lesser are the edges.
- Detect whether image is extra bright/dark?
  - To detect whether an image is extra bright or dark we extract the RGB information from the image and averaging the three values gives the measure of brightness or darkness.
- Detect whether image has Gaussian noise?
  - To detect whether Gaussian noise is present in an image we look at the histogram of the image, if the noise is present the histogram follows a curve like Gaussian distribution

Outline

Resolving Extreme Brightness to perform detection (1/2)

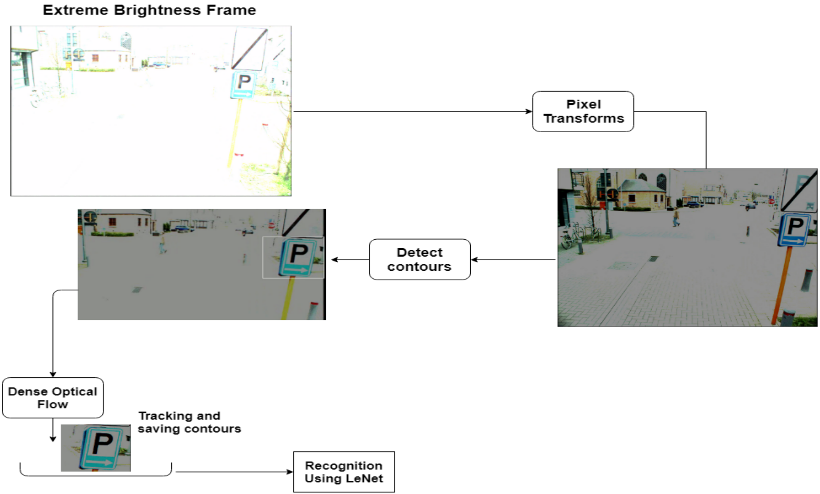

Figure: Resolve Extreme Brightness

Resolving Extreme Brightness to perform detection (2/2): Approach using Pixel Transforms

- In this kind of image processing transform, each output pixel's value depends on only the corresponding input pixel value (plus, potentially, some globally collected information or parameters).

$$g(i,j) = \alpha f(i,j) + \beta$$

  where, $f(x)$ is source image pixels, $g(x)$ is output image pixels, α controls contrast and β controls brightness/darkness.
- Usually, this approach is used to increase brightness of an image ($\alpha \in [1.0, 3.0]$ and $\beta \in [0, 100]$). However, we used this approach to reduce brightness by tuning parameters as: $\alpha \in [0.0, 1.0]$ and $\beta \in [-100, 0]$.

Outline

Resolving Extreme Darkness to perform detection (1/3)

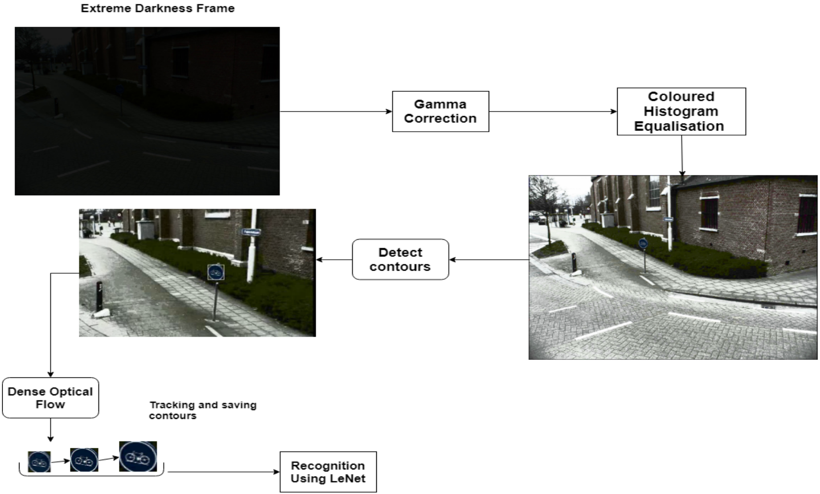

Figure: Resolve Extreme Darkness

Resolving Extreme Darkness to perform detection (2/3): Approach using Gamma Correction and Coloured Histogram Equalization

- Gamma Correction is used to correct the brightness of an image by using a non linear transformation between the input values and the mapped output values.

$$O = \left(\frac{I}{255}\right)^{\gamma} * 255$$

- As this relation is non linear, the effect will not be the same for all the pixels and will depend to their original value.
- When $\gamma < 1$, the original dark regions will be brighter and when $\gamma > 1$, the original brighter regions will be darker.

Resolving Extreme Darkness to perform detection (3/3): Approach using Gamma Correction and Coloured Histogram Equalization

- After using Gamma Correction techniques, the contrast of the image is reduced sometimes to such an extent where it results to a lot of loss in information. Hence, we use Histogram Equalization technique in order to improve contrast of the image.

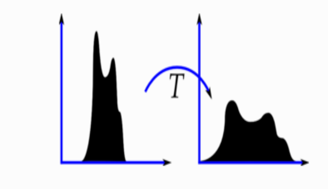

Figure: Histogram Equalization Transform

Outline

Removing Gaussian Noise to perform detection (1/2)

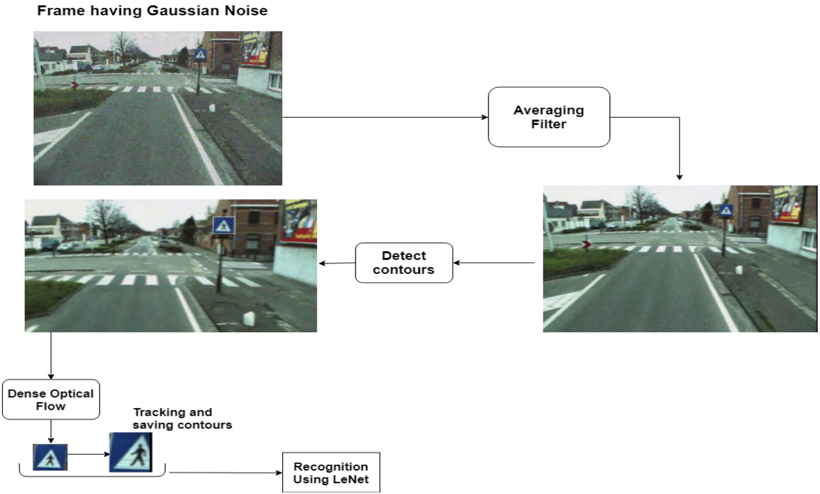

Figure: Removing Gaussian Noise

Removing Gaussian Noise to perform detection (2/2): Approach using Mean Filtering

- Image filtering is used to remove noise and any undesired features from an image, creating a better and enhanced version of that image.
- Mean/Average filtering is an example of a linear filter. It has the effect of smoothing the image, removing noise from the image and increasing its brightness.
- We tried working with other non-linear filters like Median Filtering, which usually performs better than Mean filtering. However, median filtering has a disadvantage that it looses out fine details along with noise. Anything relatively small in size compared to the size of the neighborhood will have minimal affect on the value of the median, and will be filtered out. Hence, we opted to employ mean filtering.

Outline

Removing Lens Blurring to perform detection (1/2)

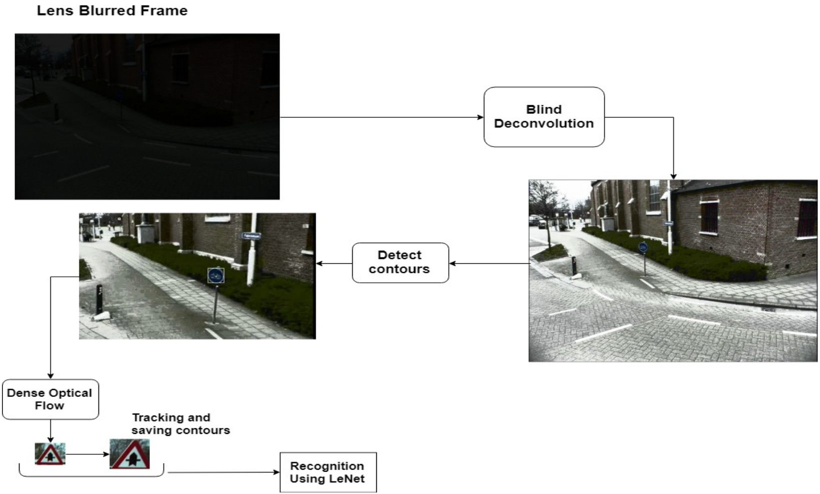

Figure: Resolve Extreme Brightness

Removing Lens Blurring to perform detection (2/2): Approach using Blind Deconvolution

- Deconvolution technique permits recovery of target scene from a single or set of "blurred" images in presence of poorly determined or unknown point spread function (PSF)
- For Blind Deconvolution technique, the PSF is estimated from the image or image set, allowing deconvolution to be performed.
- Blind Deconvolution can be performed iteratively, whereby each iteration improves the estimation of PSF. A good estimate of PSF is helpful for quicker convergence but not necessary.

Outline

Detecting Contours using Hough Transform (1/2)

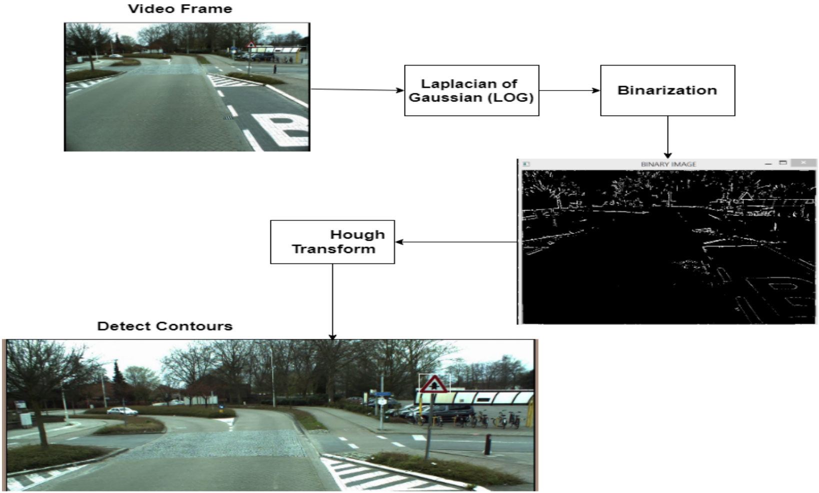

Figure: Block Diagram

Detecting Contours using Hough Transform (2/2)

- Apply Laplacian of Gaussian (LoG) to get the borders of different objects present in the new frame of a video.
- Make contours by Image Binarization.
- Pass the binary image through Generalized Hough Transform.
- The result of above step is passed through SVM classifier to predict whether the given bounding box contains a traffic sign or not.

Outline

Tracking detected Traffic Sign using Dense Optical Flow

- Optical Flow: movement of objects in the image across consecutive image frames.
- Dense Optical Flow Algorithm is based on Gunner Farneback approach.
- Since detection of traffic sign is not feasible in every frame, we detect the traffic sign in one frame (output of SVM classifier), and track the sign using Dense Optical Flow approach.

Outline

Outline

Classification of Traffic Signs using Le-Net

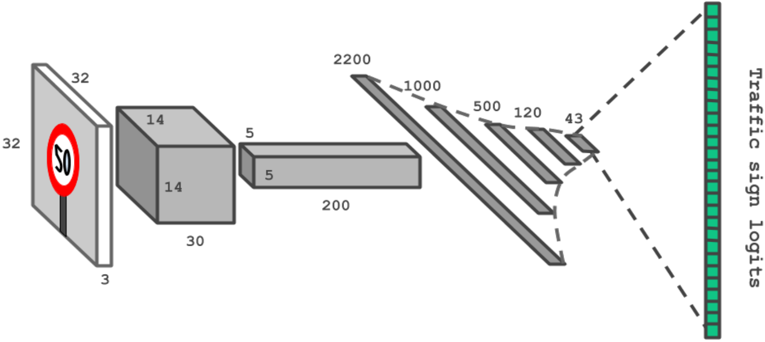

Figure: Le-Net Architecture

a

[a]https://github.com/muddassir235/German-Traffic-Sign-Classifier/tree/master

Outline

Specifications of Model

- Number of Layers: 7
  - Layer 1: Convolutional (30 5x5 filters)
  - Layer 2: Convolutional (200 5x5 filters)
  - Layer 3: Fully connected (2200 depth)
  - Layer 4: Fully connected (1000 depth)
  - Layer 5: Fully connected (500 depth)
  - Layer 6: Fully connected (120 depth)
  - Layer 7: Output Layer (43 Traffic Sign Classes)
- Activation function: ReLU
- Dropout: 0.5 throughout fully connected layers
- L2 regularization of 1e-6 is also applied.

Outline

Results

Figure: Traffic Sign Classification

- Training Accuracy reported: 99.8%
- Testing Accuracy reported: 95.8%

Acknowledgement

We would like to express our gratitude to Dr. Mehul Raval for supporting this work. Also, we are immensely grateful to Mr. Vandit Gajjar for providing extensive help during the course of this work.

Thank You!

Outline

References (1/2)

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural Networks, vol. 32, pp. 323-332, 2012.

[2] D. Temel and G. AlRegib, Traffic Signs in the Wild: Highlights from the IEEE Video and Image Processing Cup 2017 Student Competition [SP Competitions], in IEEE Signal Processing Magazine, vol. 35, no. 2, pp. 154-161, March 2018.

[3] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," 2016 IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2110-2118, 2016.

[4] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," Neural Networks (IJCNN), 2011 Int. Jt. Conf., pp. 2809-2813, 2011.

References (2/2)

[5] M. Haloi, "Traffic Sign Classification Using Deep Inception Based Convolutional Networks," pp. 11-15, 2015.

[6] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2110-2118, 2016.

[7] A. D. Kumar, "Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks. (arXiv:1805.04424v1 [cs.CV])," 2018.

[8] https://github.com/muddassir235/German-Traffic-Sign-Classifier/tree/master