Total Recall: Understanding Traffic Signs using Deep Hierarchical Convolutional Neural Networks

Sourajit Saha*, Sharif Amit Kamran† and Ali Shihab Sabbir‡

Center for Cognitive Skill Enhancement
Independent University Bangladesh
Dhaka, Bangladesh

sourajit@iub.edu.bd*, sharifamit@iub.edu.bd†, asabbir@iub.edu.bd‡

Abstract—Recognizing Traffic Signs using intelligent systems can drastically reduce the number of accidents happening worldwide. With the arrival of Self-driving cars it has become a staple challenge to solve the automatic recognition of Traffic and Hand-held signs in the major streets. Various machine learning techniques like Random Forest, SVM as well as deep learning models has been proposed for classifying traffic signs. Though they reach state-of-the-art performance on a particular dataset, but fall short of tackling multiple Traffic Sign Recognition benchmarks. In this paper, we propose a novel one-for-all architecture that aces multiple benchmarks with better overall score than the state-of-the-art architectures. Our model is made of residual convolutional blocks with hierarchical dilated skip connections joined in steps. With this we score 99.33% Accuracy in German sign recognition benchmark and 99.17% Accuracy in Belgian traffic sign classification benchmark. Moreover, we propose a newly devised dilated residual learning representation technique which is very low in both memory and computational complexity.

Keywords—deep learning; traffic sign recognition; convolutional neural networks; residual neural network; computer vision

I. INTRODUCTION

As Self-driving cars are being introduced in major cities, intelligent traffic signs recognition has become an essential part of any autonomous driver-less vehicles [1]–[3]. Transitioning from a vehicle with driver to a driver-less vehicle should come in steps. The major issue seems to be the person driving the vehicle not being attentive enough to notice the traffic signs in due time. A safety harness should be placed which automates the recognition of traffic signs and alerts the driver in due time as they might suffer from fatigue or other causes [5], [6]. Solving this can greatly reduce the number of casualty in major roads and highways. Orthodox computer vision techniques [7] and machine learning based architectures were popular for traffic sign classification [8]–[11], yet failed miserably to state-of-the-art deep learning architectures in recent times. Currently, deep convolutional neural networks are capable of outperforming any traditional machine learning methods for traffic sign recognition. Though many deep learning models have been proposed for traffic sign classification [12], [13], they failed to provide sufficient evidence regarding overall performance for multiple traffic sign benchmarks. However, some of them scored nearly Top-1% accuracy for a particular benchmark like GTSRB data-set [14], [15], yet no further example of their models were shown for other popular benchmarks like BelgiumTS data-set [16]–[18] or LisaTSR benchmark [19]. Hence, it is evident that there is no single architecture which outperforms multiple benchmarks for Traffic Sign Recognition. It is important to mention that, a model should not be embedded on an autonomous vehicle before proving its accuracy and performance for multiple benchmark. In this paper, we propose a novel deep convolutional neural network architecture. It is accompanied by hierarchical structure with customized skip connections [20], [21] in steps. The skip connections are made with dilated convolutions [22] with different filter sizes to increase the receptive field of the feature extractors. With this architecture we reach 99.33% accuracy in German Traffic Sign Recognition data-set [14], [15]. On the other hand, we reach 99.17% accuracy in Belgian Traffic Sign data-set [16]–[18]. Concurrently, it can be embedded to any autonomous vehicle as its reaches competitive scores on multiple benchmark.

II. LITERATURE REVIEW

Various methodical approaches were adopted for recognizing traffic signs in the field of computer vision. However, most of them were clinging to only one set of benchmark. In this work, we discuss such methodologies starting from classical brute force approaches to modern learning representations. Most of them tackle the fundamental classification, detection and localization challenges surrounding traffic sign recognition.

A. Feature Extractor as Classification Technique

Classical approaches include techniques such as Histogram Oriented Gradients (HOG) [25], [26], Scale Invariant Feature Transformation (Sift) [7] and Sliding Window [27]. HOG based techniques were used for visual salience [25] and then for color exploitation for pedestrian detection [26]. Moreover,

TABLE I. Architecture Comparison with Large-Scale Classifiers on GTSRB.

| Architecture     | Top-1 Error, ϵ₁ | Top-5 Error, ϵ₅ | Parameters | Training Time |
|------------------|------------------|------------------|------------|---------------|
| Our              | 0.67%            | 0.34%            | 6.256 M    | 2.03 Hr       |
| Pre-Resnet-1001  | 0.71%            | 0.47%            | 10.2 M     | 6.07 Hr       |
| Resnet-50        | 0.73%            | 0.52%            | 23.8 M     | 4.55 Hr       |

Code is available at https://github.com/Sourajit2110/DilatedSkipTotalRecall
gradients of RGB images were computed along with different normalized, weighted histograms for finding the best detection algorithm to find pedestrians and signs. Furthermore, scale Invariant Feature Transform (SIFT) \cite{28} technique was used for classification of traffic signs, whereas sliding window \cite{27} approach was used to find candidate ROIs (Region of interest) within a small-sized window, and then further verified within a large-sized window for higher accuracy in object detection.

### B. Orthodox Machine Learning Techniques

Machine learning methods like support vector machines \cite{8}, \cite{29}, linear discriminant analysis (LDA) \cite{30}, kd-trees \cite{31}, maximally stable extremal regions \cite{32} and Random forest \cite{33} swept away the brute force approaches in traffic sign recognition. Concurrently, LDA is based on maximum likelihood estimate or maximum posteriori estimation between classes and class densities are represented by multivariate Gaussian and common co-variance matrix \cite{30}. However, discriminant function analysis is very similar to logistic regression, and both can be used to answer the same research questions \cite{34}. Logistic regression does not have as many presumptions and restrictions as linear discriminant analysis. However, when discriminant analysis supposition are met, it is more superior and stronger than logistic regression \cite{35}. In Random Forest a set of non pruned random decision trees are used to make an ensemble architecture through which the best classification scores are achieved \cite{33}. The decision trees are made with features selected randomly from the training set. For traffic sign recognition test data is validated by all the decision trees and the categorical output and probability scores are based on the majority voting. Support Vector Machines(SVM) is used for classification problem and it classifies the data by dividing the n-dimensional data plane with a hyper plane \cite{29}. SVM can transform the classification plane to higher dimensions using kernel trick \cite{42}. The method separates the non-linearly scattered data using a non-linear kernel function. The major problems for above techniques are, features needs to be hand-engineered and machine learning is heavily dependent on human-intervention \cite{9}, \cite{30}, \cite{33}. These approaches can not handle variable length images neither can converge better with data augmentation and low pre-processing.

### C. Deep Learning Approaches

To get rid of the drawback of the above mentioned techniques, new architectures based on deep learning algorithms were emerged \cite{37}–\cite{40}. The reasons being the increase in the amount of available computing resources and access to huge annotated data. Currently, almost all the state-of-art architectures for traffic signs are Convolutional neural networks. The first of its kind was the introduction of LeNet architecture \cite{41} for traffic sign recognition for German Traffic Sign data-set Benchmark Challenge \cite{14}, \cite{15}. What makes convolutional neural networks more accurate and easily implementable is it’s tower like structure that can process information and learn features in depth. There are variations in each block and layers, like convolution layer is the main feature extractor which uses filters with small receptive field to process input \cite{42}, pooling layer is used for reducing spatial dimension \cite{43} and then there is dense or fully-connected layer which takes input from all the neurons of the previous layers and shares the information to connected layers \cite{44}. A loss function is defined which is reduced by Back-propagation \cite{45}. Moreover, a new type of convolution called dilated convolution \cite{22} has been replacing vanilla convolution in latest architectures. The main intuition behind this is it increases receptive field exponentially if stacked on top of each other \cite{20}, whereas in vanilla convolutional layer, stacking increases receptive field linearly \cite{20}, \cite{44}.

### III. Proposed Methodology

#### A. Skip Connections in Convolutional Neural Networks

With the introduction of deep learning, the computer vision research community have been thriving to acquire precision level accuracy in image classification along with other computer vision tasks. Concurrently, we have seen deeper architectures like \cite{47} with 143.6 Million parameters and improvements like \cite{48} that goes even deeper with parameters as many as 23.8 Million. However with the depth increasing, the
accuracy does not idealistically get better. As the study [49], [50] suggests, accumulating more kernels causes performance degradation without provoking the high variance - high bias paradigm. Clearly, the recursive process of learning more features based on the previously learned features is not always optimal. Therefore, there has to be a trade-off between how much to learn from the already learned features with respect to the immediate previous layer and layers further beyond. To answer that intriguing question [24] first introduced deep residual learning, where each building block learns new features and simultaneously passes the previous layer as they are. This skip like connections gives the network the feasibility to attain features from layers further beyond when needed, affording the network to learn more interesting features with juxtaposition of features from any previous layers at any point in time, also known as the identity mapping. Further analysis of identity mapping [23] has confirmed that using batch normalization and activation before a convolutional layer afforded the network to learn more interesting features with respect to the already learned features with respect to the immediately previous layer.

As shown in Figure 1, a conventional residual block [23] has 2 pre-activation convolutions in it’s residual branch, the accumulation of which we denote by \( F^R(X_l) \) and identity mapping, \( H(X_l) = X_l \). The conventional Residual operations are demonstrated in Equation (1). In stead of using a bunch of these residual blocks we tried using a few with newly designed skip connections that are capable of learning more interesting features. Inside each residual block we embedded two more residual units, which goes inside the two composite functions as illustrated in Figure 1 therefore acting as one single residual block with two more residual blocks within. We call it Inner Residual Block and we denote it’s output function as \( F^{IR}(X_l) \). The computation of Inner Residual Block is illustrated by Equation (2), (3).

\[
X_{l+1} = F^R(X_l) + H(X_l) = F_2(F_1(X_l)) + H(X_l) \quad (1)
\]

\[
F^{IR}(X_l) = F_3(F_2(F_1(X_l)) + H(X_l)) + H(X_l) \quad (2)
\]

\[
X_{l+1} = F^{IR}(X_l) + H(X_l) \quad (3)
\]

\[
X_{l}^{IR} = X_l + \sum_{k=l}^{L-1} F_3(F_2(F_1(X_k) + H(X_k)) + H(X_k)) \quad (4)
\]

B. Dilated Convolutional Operation

Dilated convolutional operation has been a recent development that has lead to improved classification [22], [46] and segmentation [51] performances. A dilated convolution kernel uses the same number of operation as that of a regular kernel yet it skips the tensor by a fixed number of pixels, thus resulting in a extended receptive field. A regular convolution filter of size \((k \times k)\), where \(k = \alpha, \forall (\alpha > 1 \cap \alpha \in \mathbb{Z})\); has a receptive filed of \(E_1 = (k \times k)\) whereas, a dilated convolution kernel with size of \((k \times k)\) and dilation rate \(r = \beta, \forall (\beta > 1 \cap \beta \in \mathbb{Z})\); has a receptive field of \(E_2 = ((k + (k - 1)(r - 1)) \times ((k + (k - 1)(r - 1)))\). Equation (5) elaborates the extension in receptive field with no added computation where, \(\Delta E > 0\). We denote the number of parameters with a regular and dilated kernel for the \(l^{th}\) layer with depth, \(D[l]\) as \(P_1\) and \(P_2\) respectively. A comparative analysis of parameter requirement is formulated using Equation (6), (7), (8).

\[
\Delta E = (k - 1)(r - 1)(2k + (k - 1)(r - 1)) \quad (5)
\]

\[
P_1 = (k + (k - 1)(r - 1))^2 \times D[l] \times D[l-1] \quad (6)
\]

\[
P_2 = (k)^2 \times D[l] \times D[l-1] \quad (7)
\]

\[
P_1/P_2 = \left[ 1 + \frac{(k - 1)(r - 1)}{k} \right]^2 \quad (8)
\]

Since \((k - 1)(r - 1) \geq k\) therefore \(P_1 > P_2\), denoting that dilated convolution confirms a reduction in parameters. Given a input tensor of spatial dimension \((X \times X)\), a deep convolutional neural network takes 1 regular kernel with zero padding and 1 stride to reduce the spatial dimension to \(((X - (k - 1)) \times (X - (k - 1)))\), whereas, the same kernel with dilation rate \(r\) can reduce the tensor dimension to \(((X - (k - 1) \times r) \times (X - (k - 1) \times r))\). Intrinsically, running the same regular kernel \(r\) times would result in such spatial reduction.

C. Inner Residual Dilated Skip Convolution

Figure 2 depicts how we collocate both the dilation and residual unit concepts together. To posit the inner mechanism, we denote \(k_m = m\), where \(m\) represents both height and width of kernel \(k\). In each of our block we apply two \(k_3\) kernels to the input tensor in the first place. Then we skip a step and...
apply a \( k_3 \) kernel with dilation, \( r=2 \) and get to the same state that had used 2 kernels, which then gets added and passed into another \( k_3 \) kernel. Again we apply a \( k_3 \) with dilation rate, \( r=3 \) thus scraping the layout of using 3 kernels, which then we add with the \( 3^{rd} \) \( k_3 \) kernel results; ready to go as a output. However, we add a identity mapping before outputting the tensor and denote the dilated convolution function as \( R(X_1) \), residual output as \( F^{DIR}(X_1) \), number of parameters for non dilated inner residual unit and dilated inner residual unit as \( P^{IRR} \) and \( P^{DIR} \), respectively. Equation (9), (10), (11), (12) elaborates the detailed calculations of Dilated Inner Residual Blocks.

\[
F^{DIR}(X_1) = F_3(F_2(F_1(X_1) + R_1(X_1)) + R_2(X_1)) \quad (9) \\
X_{l+1} = F^{DIR}(X_l) + H(X_l) \quad (10) \\
X^{IRR}_l = X_l + \sum_{k=l}^{L-1} F^{DIR}(X_k) \quad (11) \\
P^{IRR}_l - P^{DIR}_l = D^{(l)} \times D^{(l-1)} \times ((K_2^2 - K_3^2) + (K_1^2 - K_3^2)) \quad (12)
\]

Since \( k_7 > k_5 > k_3 \) therefore \( P^{IRR}_l > P^{DIR}_l \), ergo our architecture assures a stringent reduction in parameters. Studies \cite{52, 53} have shown that divide and merge feature mapping helps learn coarse features, thus we have incorporated the idea in each of the dilated inner residual blocks. Figure 3 construes our network architecture while Table II interprets all the kernel operations and filter output dimensions. As each blocks are capable of simultaneously looking at the same tensor with different view points and retain all of the cor-relational tensors together therefore we address this approach as “Total Recall”. Furthermore, all the regular and dilated convolutions in all dilated inner residual blocks use the same number of kernels. The term \( D_\Omega \) used in Table II refers to the fact that all the kernels in a particular block uses \( \Omega \) of channels.

IV. EXPERIMENTATION

A. Data-set Preparation

We train and test our models on two sets of benchmark. The first one is German Traffic Sign Benchmark or GTSRB \cite{14, 15} and the second is Belgium Traffic Sign Benchmark, also known as BTSC \cite{16–18}. In this way, we get a viable model which performs very well on arbitrary data. There is essentially no difference between the data-sets except for the number of classes. All the RGB images from both the data-sets are cropped to \( 56 \times 56 \) spatial dimension for both test and training. Table III details all the information related to GTSRB and BTSC data-sets. Figure 4 illustrates a preview of the BTSC data-set.

B. The Neural Net Training and Performance Evaluation

\[
\mathcal{L}(\hat{y}, y) = -\frac{1}{N} \sum_{j} [y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j)] \quad (13)
\]

\[
\alpha^{[i-1]} = \begin{cases} 
\alpha^{[i-1]} & \text{if } \mathcal{L}^{[i-1]} \leq \mathcal{L}^{[i-2]} \leq \mathcal{L}^{[i-3]} \\
0.1\alpha^{[i-1]} & \text{if } 0.1\alpha^{[i-1]} \leq 10^{-12} \\
\alpha^{[i-1]} & \text{otherwise}
\end{cases} \quad (14)
\]

While training, we split the 39,209 GTSRB images into 1226 batches and the 4,575 BTSC images into 143 batches, each with 32 images per batch. We have used categorical cross-entropy loss function, \( \mathcal{L} \) to calculate approximation error using Equation (15), where \( \hat{y} \), \( y \) denotes prediction, true label correspondingly. Concurrently, to optimize the approximation we used Adam \cite{56} with a initial learning rate \( \alpha = 10^{-4} \) that updates itself using Equation (14) with reference to validation loss at \( i^{th} \) iteration. We trained both data-sets separately for 30 epochs on a 8 Gigabyte Nvidia GeForce GTX 1070 GPU using keras \cite{57}, a deep learning library. The GTSRB data-set was trained for 2.03 hours while the BTSC data-set took 0.23 hour to train. We ended up with the best performance (Table IV) on GTSRB benchmark as well as (Table V) on BTSC benchmark. Figure 6 shows the validation accuracy and loss plots after every iterations which depicts how quick the validation loss decreases while keeping a consistent accuracy, thus demonstrates the robustness of the proposed architecture (6.256 Million Parameters).

### TABLE III. Different Data-set

| Dataset  | Total Images | Training Image | Test Image | Image Size | Number of Classes |
|----------|--------------|----------------|------------|------------|------------------|
| GTSRB    | 51839        | 39209          | 12,630     | 15x15 to 250x250 | 43               |
| BTSC     | 7095         | 4575           | 2520       | 11x10 to 562x438 | 62               |

### TABLE IV. Architecture Comparison for GTSRB

| Models   | Top-1 Accuracy (%) |
|----------|---------------------|
| Ours     | 99.33               |
| PCNN     | 98.7                |
| μNet     | 98.9                |
| Human    | 98.84               |
| HOG      | 96.14               |
| CDNN     | 98.5                |
| Multi-scale-CNN | 98.31             |
| pLSA     | 98.14               |
| Capsule NN | 97.62              |
| Random Forest | 96.14             |

### TABLE V. Architecture Comparison for BTSC

| Models   | Top-1 Accuracy (%) |
|----------|---------------------|
| Ours     | 99.17               |
| HOG      | 98.34               |
| OneCNN   | 98.17               |

Fig. 4. Different traffic signs from Belgium Traffic Sign Classification Data-set
Fig. 5. View Point Variation and Feature Maps at Different Layers of our Dilated Inner Residual Deep Conv-Net. Each Layer’s findings are shown using both activation scale and hit-maps. Terms like Conv 2a $F_2(X)$, Conv 2a $R_1(X)$ used are inferred with juxtaposition of notations used in Figure 2 and Table II.

C. Observation and Justification

As Figure 5 demonstrates, the 2nd consecutive $[k_3]$ set of kernels in the 1st residual block (Conv 2a $F_2(X)$) succeed at finding round and straight edges whereas, the 1st dilated $[k_3, r = 2]$ set of kernels in the very same block (Conv 2a $R_1(X)$) detects noise around those edges thus leaving the findings blunt. However when added together (Conv 2a $F_2(X)$ + Conv 2a $R_1(X)$), they tend to trigger more robust features like precise edges and shades. Concurrently, deeper layers (Conv 2a $F_3(X)$ + Conv 2a $R_2(X)$) succeed at rendering the background details obsolete thus focusing further on the traffic signs rather than the background. Subsequently, the feature maps toward the end layers (Conv 4b $F_2(X)$ + Conv 4b $R_1(X)$) extract very abstract level details with each iteration of optimization. On the other hand, we also trained various large scale Image-net challenge winning image classifiers to compare our results, the details of which is shown in Table I.

V. Conclusion

As many traffic sign recognition CNN models as there exist, ours evidently requires the least amount of learnable parameters and training time. Furthermore, our CNN model is the first of its kind to be robust enough across different traffic sign recognition or classification benchmarks. With all the tech giants focusing more on producing self driving cars, such one-for-all as well as lightning fast traffic sign recognition platform would come in handy with numerous business values in perspective. Needless to say, when it comes to driving, understanding of the environment at precision level is the most important aspect. Consequently, faster recognition of the objects in the environment- of which traffic signs are an inevitable part of, determines how accurately a self driving car agent or a human driver for that matter would actuate in the environment. Furthermore, the devised methodologies and theories behind the making of this CNN architecture opens up possibilities of re-visualizing, rethinking and re-scaling in large-scale image classification which sure enough, is a domain for further research in the distant future.

ACKNOWLEDGMENT

We would like to thank “Center for Cognitive Skill Enhancement” Lab at Independent University Bangladesh for both the technical and inspirational supports provided.

REFERENCES

[1] M. Haloi and D. B. Jayagopi, “A robust lane detection and departure warning system,” in Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE, 2015, pp. 126–131.
[2] J. C. McCall and M. M. Trivedi, “Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation,” IEEE transactions on intelligent transportation systems, vol. 7, no. 1, pp. 20–37, 2006.
[3] M. Haloi and D. B. Jayagopi, “Vehicle local position estimation system,” arXiv preprint arXiv:1503.06648, 2015.
[4] C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel, “Driver-activity recognition in the context of conditionally autonomous driving,” in Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, 2015, pp. 1652–1657.
[5] J. R. Treat, “A study of precrash factors involved in traffic accidents,” HSRI Research Review, 1980.
[6] G. Zhang, K. K. Yan, X. Zhang, and Y. Li, “Traffic accidents involving fatigue driving and their extent of casualties,” Accident Analysis & Prevention, vol. 87, pp. 34–42, 2016.
[7] D. G. Lowe, “Object recognition from local scale-invariant features,” in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2. Ieee, 1999, pp. 1150–1157.
[8] T. Chen and S. Lu, “Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression,” IEEE Trans. Vehicular Technology, vol. 65, no. 6, pp. 4006–4015, 2016.
