Abstract— We propose a new traffic-sign detection method based on a weakly-supervised multi-purpose single convolutional neural architecture. The base classification network used is the very light convolutional architecture, MobileNetv2, which is used as a region proposal network, in addition to being a classification network. The method is divided into two stages; in the first stage MobileNetv2 is trained to suggest certain regions within the image to classify, while in the second stage it is trained to be a traffic-sign classifier. The method attained few milliseconds of processing time for single image or frame testing and averaged about 55 milliseconds on 800x1300 resolution, while maintaining an acceptable accuracy. This method takes advantage of weak supervision which completely eliminates the time required for dataset annotation. We trained and tested our proposed technique on two datasets that have been broadly used for traffic-sign recognition and detection: the German Traffic Signs Recognition Benchmark (GTSRB) and the German Traffic Signs Detection Benchmark (GTSDB).

I. INTRODUCTION

Traffic sign detection is a common problem in computer vision, due to the importance of such systems for self-driving vehicles and advanced driver assistance systems (ADAS). Many methods have been developed in recent years to accurately localize and detect traffic signs on the road and at the same time maintain real time processing to keep up with vehicle speed. While most of the efficient methods for object detection such as YOLOv3[1], SSD[2] and RetinaNet[3] achieve excellent performance in terms of both accuracy and processing time, all of them are fully supervised and thus still require performing a cumbersome and time-consuming annotation operation for every single image in the training set. This full supervision problem means it is difficult to train any of these networks mentioned above, which makes it appealing to employ weak supervision to overcome this difficulty and save significant time that would have otherwise been lost in annotation. However, weak supervision enables us to address the issue at the expense of detection accuracy. Weak supervision has been used effectively as a general object detection method, but not specifically for traffic sign detection. We see that traffic sign detection is much harder than general object detection, since in most of the cases the traffic sign is so small that it only occupies a few pixels of the image; even when it becomes extremely close to the vehicle, it still occupies a very small area when compared to the total size of the image, at most perhaps 1/15 of the image. This makes traffic sign detection more difficult than general object detection when the objects occupy a large portion of the

II. RELATED WORK

A. Feature descriptor-based object detection

Many methods have been introduced to offer a solution for traffic sign detection and localization. The main method which has been commonly used in recent years relies mainly on features descriptors and Histogram of Oriented
Gradients (HOG) to extract the main features of the traffic signs, then using Support Vector Machine (SVM) classifier to classify the images based on the features extracted. This method has been used successfully by Wang et al. [6], Samuele et al. [7] and Yang et al. [8], amongst many other researchers, to provide effective solutions with high precision/recall, which in some cases reached around 100%. However, the speed of processing was an issue; also, extracting the features using HOG was computationally expensive and time-consuming, which made it inconvenient for real time application.

A. Fully supervised object detection

Other methods were introduced as fully supervised object detectors by training a Convolutional Neural Network (CNN) to predict the bounding boxes of the objects in the scene, in addition to predicting the class probability of every object. As mentioned, a lot of time is required to annotate the training data in this method (which must be a big dataset to well-train the network), meaning that a significant amount of time is needed to complete the annotations. The best examples of fully supervised methods are Single Shot Detector (SSD) [2], YOLOv2 [9], YOLOv3 [1] and RetinaNet [3]. All of these networks are perfect object detectors with an excellent mean average precision of around 30–37. They are all real time object detectors, with the best of them having a frame rate of 45 fps (YOLOv2) and 25 fps (YOLOv3), although the image size in most cases is still small (320x320 and 448x448 in both methods). However, as the image size increases, the frame rate drops significantly, so the accuracy increases.

B. Weakly supervised object detection

Weakly supervised methods usually suffer from low accuracy and mean average precision, but their great advantage is saving time, with there being no need for image annotation or computationally expensive feature extraction (as discussed in the two previous methods). Most of the weakly supervised object detection (WSOD) methods utilize the Multiple Instance Learning Pipeline (MIL), where mainly a classification network is used to obtain information about the locations of the objects in a scene. Many methods have been introduced as WSOD methods, such as Bilen and Vedaldi [10] presenting a deep network approach for WSOD in which a classification network is giving a weighted sum of proposed regions. Li. et al. [11] offer another approach, training an image classifier to mask out the image to find the objects in a scene, and Ge et al. [12] proposed a multi-tasking technique of joint optimization to perform recognition, detection and segmentation simultaneously to improve the detection accuracy. All of the previous methods are different from our approach, the main difference being that most of these weak supervised methods are general object detectors, which do not focus on detecting the small traffic signs which are hard to detect in a weak supervision fashion.

III. PROPOSED METHOD

A. The big picture

The proposed method mainly uses MobileNetv2 [4] as the main CNN network. The architecture of MobileNetv2 consists of a fully convolutional layer with 32 filters, followed by 19 residual layers. It uses 3x3 filter size for the convolutional layers and employs batch normalization and drop-out in training. MobileNetv2 has been used for two different tasks:
classification and region proposal. The network used was pre-trained on the ILSVRC ImageNet dataset for classification and fine-tuned on the GTSRB [13] dataset to work as an image classifier. It was also fine-tuned to perform ‘Objectness’ or ROI proposal; in other words, the second network aims to find the locations in the image where there are traffic signs. Fig. 1. shows a detailed graph of the proposed method.

B. Training procedure

First, training a traffic sign classifier on the GTSRB [13] dataset was a straightforward process, due to the pre-trained network making it easy to fine-tune the network on the new dataset while obtaining a high level of accuracy. The GTSRB dataset consists of 43 different traffic sign classes which fall under three main categories: Prohibitory, Mandatory and Danger. One extra class has been added to the dataset, the background class, to indicate the absence of a traffic sign. The background training images were generated by cropping large images to many small images from the GTSDB[14] training dataset where there are no traffic signs in the scene. The classification network has been trained with an input shape of 128x128x3 with a loss function of categorical cross entropy loss.

Second, training MobileNetv2 for proposing regions of interest (ROI) has been done by fine-tuning the pre-trained network on just 2 classes: traffic sign class and background class, by giving all the images in the GTSRB training set as traffic sign class training data and those cropped in the first stage as background class training data. This network has also been trained using an input shape of 128x128x3 with a loss function of binary cross entropy loss. The last fully connected layer has been changed to a convolutional layer followed by Softmax4D to give class prediction for every pixel in the input image, indicating whether there is a traffic sign or not in the output heatmap. It is important to note that the proposed regions are few and accurate when compared to other region proposal network methods, and during testing no sliding window is applied like in the faster RCNN [15] method.

C. Testing procedure

While testing, the two networks are used back-to-back. Initially, the region of interest proposal network is fed with the testing image, which then produces a heatmap proposing regions of interest. The coordinates of the proposed regions are obtained and the corresponding regions are cropped from the input images and fed to the classification network to classify them. Depending on the class probability of each image crop, a bounding box is drawn with the same coordinates of the cropping coordinates around the traffic sign with its class probability as shown in Fig.1. If the cropped image is classified as ‘background’, its bounding box is discarded. Some calibration for the cropping coordinates and the bounding box is applied to remove any error caused by the mismatch between the image size and the heatmap size.

IV. EXPERIMENTS AND RESULTS

A. German Traffic Sign Detection Benchmark (GTSDB)

The GTSDB[14] dataset is a detection dataset consisting of 900 images divided into 600 images for training and 300 for testing of video frames, which were captured in Germany in urban and rural areas. The images are RGB images in ppm format with a resolution of 800x1300. The testing images have been used to evaluate our method. The proposed method was trained and tested on Nvidia TITAN Xp Gpu, Intel(R) Core(TM) i7-8700 CPU @3.20GHz processor.

B. Results Evaluation by GTSDB metrics

The testing images have been fed to the combined network, as mentioned before in detail, and the results have been evaluated by GTSDB metrics which calculate recall/precision for every one of the three categories: Prohibitory, Mandatory and Danger. Precision is calculated by dividing the true positive detections over all detections, while recall is calculated by dividing the true positive detections over all the ground truth images. Table I shows the GTSDB test set experiment result of precision/recall over the three categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prohibitory</td>
<td>0.35</td>
<td>0.3</td>
</tr>
<tr>
<td>Mandatory</td>
<td>0.45</td>
<td>0.3</td>
</tr>
<tr>
<td>Danger</td>
<td>0.5</td>
<td>0.4</td>
</tr>
</tbody>
</table>

C. Results Evaluation by the Pascal VOC metric

The Pascal VOC evaluation metric is the mean average precision (mAP) over all classes using the recall/precision values employing the intersection over union of 0.5 in the method. The total mean average precision obtained is 15.03 which is acceptable for detection. Table II shows processing time and the mAP of our technique and three other techniques which use fully supervised methods, as we did not find any traffic sign detection method based on weak supervision to compare with.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Processing time (ms)</th>
<th>mAP @0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (800x1300)</td>
<td>55.04</td>
<td>15.03</td>
</tr>
<tr>
<td>YOLOv3 (488x488)</td>
<td>21.48</td>
<td>21.83</td>
</tr>
<tr>
<td>Faster RCNN (224x224)</td>
<td>123.27</td>
<td>24.9</td>
</tr>
<tr>
<td>SSD MobileNet (224x224)</td>
<td>15.14</td>
<td>20.64</td>
</tr>
</tbody>
</table>

D. Detection speed and frame rate

The obtained processing time per image was excellent, using the original GTSDB image size which is 800x1300. The average processing time per image was 55 milliseconds which is about 18 fps. Many image sizes have been tested as shown in Table III.
Table III

<table>
<thead>
<tr>
<th>Image size</th>
<th>Frames per second</th>
<th>mAP @ 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1080x1920</td>
<td>10.1</td>
<td>19.10</td>
</tr>
<tr>
<td>800x1300(original)</td>
<td>18.18</td>
<td>15.03</td>
</tr>
<tr>
<td>720x1280</td>
<td>20.8</td>
<td>10.04</td>
</tr>
<tr>
<td>480x640</td>
<td>40</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Notes on the results

The results can be seen as acceptable and not perfect since there were many factors that negatively affected our approach:

1) The tiny traffic signs in the scene

Many of the testing images contain tiny traffic signs which were extremely far away from the vehicle during the video. These tiny traffic signs are around (or even smaller than) 30x30 pixels in images of 800x1300 pixels, which represents a mere 2.5% of the image size. This occurs in many images and the signs are impossible to be detected and recognized using our approach because the heatmap is 1/32 of the original image size. Therefore, the smallest object that can be proposed by the ROI proposal network is 32x32 pixels.

This problem of missing the tiny traffic signs was reflected in the recall/precision and also the mean average precision, lowering their values.

2) The test set does not cover all traffic sign classes

The test set does not distribute the samples over the whole range of different traffic sign classes, although we think this was unintentional because the original evaluation method for the GTSTDB is over the main three categories (Prohibitory, Mandatory and Danger) and not over the classes. This issue also affected the mean average precision since mAP is calculated over the classes and not over the categories.

3) There are minor coordinate errors in bounding boxes

The final bounding box does not perfectly fit the traffic sign because, as mentioned before, the heatmap is 1/32 of the input image size. The heatmap’s width and height should be an integer number of pixels, so any fraction is neglected when obtaining the heatmap, so the heatmap gives less localization details.

V. Conclusion and future work

The weak supervision in object detection is very useful since there is no need for computationally expensive features, like methods based on feature descriptors, or very expensive image annotation in the fully supervised methods which are extremely time consuming. The weak supervision provides fast training while giving a very good frame speed in testing, which can work in real time. Also the small size of the software implemented makes it easy to be deployed in embedded computers with limited memory and limited GPU power.

We should focus the future work in two areas:

1) Improving the recall/precision and so the mAP

This can be done by modifying MobileNetv2 architecture to give larger heatmaps without losing the deep representation, in order to give the network the ability to detect smaller objects.

2) Apply the anchor boxes concept

Using the concept of anchor boxes will improve the bounding box prediction by generating many different shapes of bounding box, and trying to find the best bounding box that perfectly fits the detected object.

Acknowledgment

This research was supported by the KIAT (Korea Institute for Advancement of Technology) grant funded by the Korean Government (MSS: Ministry of SMEs and Startups) (No. S2755555, HRD program for 2019) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2017R1D1A3B03034604).

Reference