Automated Failure Detection in Computer Vision Systems

H. Yan — University of Waterloo, ON, Canada
A. Achkar — Miovision Technologies Inc., ON, Canada
A. Mishra — Miovision Technologies Inc., ON, Canada
K. Naik — University of Waterloo, ON, Canada

Abstract

Human validation of computer vision systems increase their operating costs and limits their scale. Automated failure detection can mitigate these constraints and is thus of great importance to the computer vision industry. Here, we apply a deep neural network to detect computer vision failures on vehicle detection tasks. The proposed model is a convolution neural network that estimates the output quality of a vehicle detector. We train the network to learn to estimate a pixel-level F_1 score between the vehicle detector and human annotated data. The model generalizes well to testing data, providing a mechanism for identifying detection failures.

1 Introduction

Computer vision is used widely to automate object localization and classification in images and video [1, 4] and often fails to perform without providing any warning [2]. Identifying the cases where computer vision system fails to perform is a fundamental computer vision problem [2, 3]. At Miovision, we have developed a vehicle detector that assigns each pixel of a video frame probability of being a vehicle. Fig. 1 shows an example of this detector's output on a typical road scene. Although in this case the detector produces a faithful output, an important challenge we face is to identify cases where the detector fails to produce a reliable output. Manual verification of the detector's output on thousands of hours of such video is impractical in terms of time and cost. Here, we present a failure detection module (FDM) that automates failure detection using a convolutional neural network, greatly reducing the need for human validation.

Fig. 1: Left: a typical road scene with 6 vehicles. Right: the output of our vehicle detector.

2 Methodology

To automate failure detection on our experimental vehicle detection model (VDM), we have developed a neural network model that estimates the quality of the VDM's output. Note that this VDM is an early prototype of a pixel-level classifier and is not used in production at Miovision. The workflow for this system is illustrated in Fig. 2. The failure detection module (FDM) is trained in a supervised setting on a dataset constructed from human annotated images and the output of the VDM. The VDM output and ground truth are used to compute the ground truth F_1 score, Eq. (1), and the FDM estimates this as $\tilde{F}_1$. The input to the FDM is a four channel image obtained by concatenating the red (R), green (G) and blue (B) channels of the input image with the output probabilities of the VDM. F_1 score is a suitable confidence measure as it identifies both false positives and false negatives, cases that should trigger further human validation. The optimization minimizes the mean absolute error between F_1 and $\tilde{F}_1$.

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (1)$$

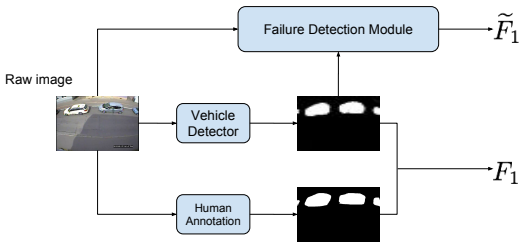

Fig. 2: Workflow of failure detection module. A raw image is processed by the vehicle detector and compared to annotated data to calculate the ground truth F_1 score. Given the raw image and the output of the vehicle detector, the failure detection module calculates estimate $\tilde{F}_1$.

2.1 Model Selection

Since the FDM should work as a confidence estimator for our VDM, one of our goals is to minimize computational resources and thus choose a relatively small network. At the same time, it has to be capable of performing fairly robust pattern recognition tasks. Namely, it must identify whether the overlap between the raw image pixels and the output of the VDM are consistent with vehicles being in those locations and not missing from the VDM output. Additionally, the network may be faced with scenarios where objects are present in both the raw pixels and VDM output, but these objects are not vehicles. Challenging examples are distortions, bright lights, and shadows. The FDM should identify these failures, so it cannot be overly simplistic. With these constraints in mind, we chose to build a model based on a simple network with 2 convolutional layers and 2 fully connected layers.[1] Despite its small size, this network can strongly discriminates classes on the CIFAR-10 dataset.

2.2 Model Architecture

The failure detection network consists of 3 convolutional layers and 3 fully connected layers (including output layer) as shown in Fig. 3. Pooling (3×3, 2×2 stride), ReLU activations and normalization layers between each hidden layer are omitted in Fig. 3. This network is larger than the simple 2 layer model but can still be trained efficiently (15.8 examples/sec on CPU and 60 examples/sec on GPU).

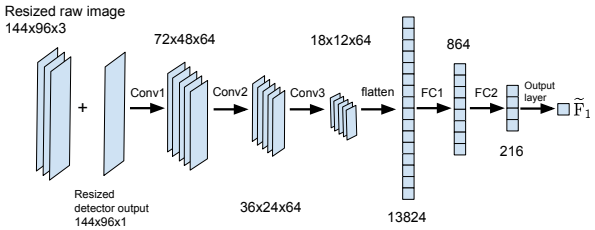

Fig. 3: Failure detection neural network. Using features from the 4 channel inputs (RGB + vehicle probability), this network is trained to predict an estimated $\tilde{F}_1$ score.

As 32×32 is too small to represent the outline of vehicles, we enlarged the size of our layers in comparison to the 2 layer CIFAR-10 model and we added an additional convolutional layer. The FDM network takes $144 \times 96 \times 4$ input consisting of the raw image and detector output. The output layer has only a single feature that represents the $\tilde{F}_1$ score. Training of this network involves minimizing absolute errors, and so can be thought of as a regression problem. The final layer is activated with a logistic function, mapping the output to the 0 to 1 range. The optimizaiton loss function is defined as the absolute error between $\tilde{F}_1$ and F_1.

3 Experiemntal setup and evaluation

3.1 Dataset

In an average month, Miovision Technologies Inc. processes $50,000$ hours of video. A small subset of these videos, consisting of complex environmental, geometric or traffic density factors are annotated by human agents for machine learning and verification tasks. The proposed failure mode detection system is trained (tested) using 8196 (1696) such annotated video frames. Our training set excludes video frames containing very small objects (< 4 pixels). Our training data set is also enhanced synthetically by applying random transformations such as flipping and adding noise.

Our vehicle detection model converts RGB stills from videos to vehicle probability maps that have a resolution of 41×41. To retain the details of raw images and to ensure computational efficiency, the raw input images and gray scale probability maps are resized to 144×96 as the input to the FDM. The RGB image is then stacked with the probability map to create a 4 channel input for the FDM.

The ground truth F_1 score (see Eq. 1) for training the FDM model are created by calculating F_1 between human annotated vehicle masks and a binarized probability map (variations of the binarization threshold are discussed in Sec. 4). Examples of raw RGB images, ground truth human annotated masks and probability maps are shown in Fig. 4. In Fig. 4, the upper probability map has an F_1 score of 0.89 while the lower set has an F_1 score of 0.34.

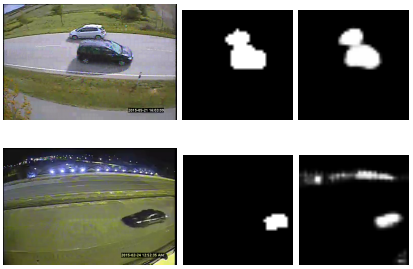

Fig. 4: Left: Input RGB images. Middle: human annotated vehicle mask. Right: vehicle detector probability map. In the top (bottom) image, the vehicle detector has an F_1 score of 0.89 (0.34). The blurry nighttime scene with bright lights is a case where this vehicle detector produces a number of false positives. The FDM predicts $\tilde{F}_1$ of 0.82 (0.36) for these, capturing the ground truth quite closely.

4 Experimental Observations

The performance of FDM is evaluated by applying two separate binarization thresholds on VDM's output probability map. In the first case, we consider any non-zero probability as representative of a vehicle. In Fig. 5, we show the relation between the ground truth F_1 and predicted $\tilde{F})_1$ on the testing set with this zero threshold. We find a wide distribution of F_1 score. This is likely due to an overemphasis on small signals that produce a number of false positives with a zero binarization threshold. Nevertheless, the predicted $\tilde{F})_1$ from our FDM is strongly correlates to the ground truth. The pearson correlation coefficient (PCC) is 0.88 and the mean error is 0.06. This indicates that the FDM converged to a generalized solution.

Using a more conservative binarization threshold of 0.15 helps to eliminate noisy signal from the probability map. As shown in Fig. 6 the ground truth F_1 clusters much higher (near 0.8) and has less variance than before. In this case, we find a PCC of 0.89 and mean error of 0.06, similar to before. However, the estimator seems to be slightly overconfident for low true F_1 score. We suspect this is due to a class imbalance in the training data.

4.1 Discussion and Future Work

This preliminary work shows promising applications of neural networks to detecting failures from our prototype pixel-level vehicle classifier. There is still some work needed to tune hyperparameters and modify the FDM model's network for practical uses.

An area we will be paying close attention to is the slight overconfidence of the FDM for low scoring samples. This may caused by bias in our training data. An attempt to account for this bias

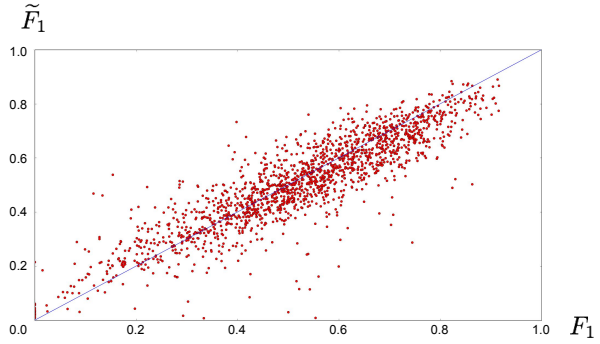

Fig. 5: Correlation between F_1 and $\tilde{F}_1$ for binarization threshold of zero. This case corresponds to treating all non-zero probability as belonging to a vehicle, and thus includes a lot of noise (false positives). The PCC is 0.88 and mean error is 0.06.

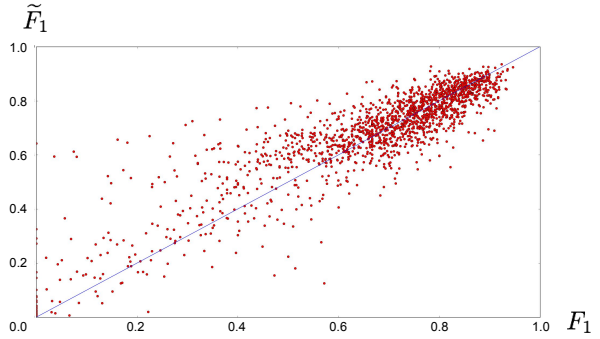

Fig. 6: Correlation between F_1 and $\tilde{F}_1$ for binarization threshold of 0.15. This case largely eliminates noise from the probability maps. The PCC is 0.89 and mean error is 0.06.

was made with synethetic data augmentation but we have yet to try balancing the training data more deliberately. In addition, we would like to greatly increase the number of training samples, but it takes time to generate enough human annotated samples. Lastly, we would like to set the optimal binarization threshold based on a global optimization.

5 Conclusion

An automated failure detection system was developed using a deep neural network architecture. This network was trained using the output of our experimental pixel-level vehicle detector. Although there are a number of further experiments to conduct, this preliminary effort validates the feasibility of applying convolutional neural networks to failure detection in computer vision tasks.

References

[1] A. Krizhevsky, and G. Hinton. "Learning multiple layers of features from tiny images," 2009.

[2] Z., Peng, et al., "Predicting failures of vision systems." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[3] S. C. Huang, "An Advanced Motion Detection Algorithm With Video Quality Analysis for Video Surveillance Systems," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 1, pp. 1-14, Jan. 2011.

[4] O. Javed and M. Shah. "Tracking and Object Classification for Automated Surveillance", In Proceedings of the 7th European Conference on Computer Vision-Part IV (ECCV '02), Springer-Verlag, London, UK, UK, 343-357. 2002.