Anomaly and Corner Case Detection in Autonomous Driving Scenarios

Siddhant Chandgadkar
Tamim Faruk
Tianyu Guo
Mustafa Shahbaz

Abstract (5%)

With autonomous driving systems taking the forefront of technological innovation in the automotive industry today, there have been many calls to increasing their robustness. One of the areas of development is corner case and anomaly detection in self-driving situations, underlying any events during a driving scenario which are unusual in nature. For example, people walking on highways or cars driving on sidewalks. Detection of these cases can help autonomous driving systems in two ways: firstly, it can contribute to their deep learning models to recognize these events in the future, and secondly, it can help software engineers to develop solutions to navigate around these corner cases. This investigation works to develop a pre-trained model to detect these corner cases, from the Cityscapes dataset in the urban environment of Stuttgart, Germany. Through three methods, this research paper was able to evaluate different ways of identifying any misalignments between predicted scenarios for a corner case, and actual scenarios for a corner case. These included using the PredNet architecture, to identify misalignments between frames, as well as using the DeepLabv3 framework to identify object misalignments within images. As well, segmentation maps from the Cityscapes dataset were used to evaluate for misalignments, and these results were output in the form of mean-squared errors, between the actual frame / segmentation map, and the predicted frame / segmentation map. The results showed that the frame misalignment through the PredNet framework worked to identify 77.8% of corner cases, while the object misalignment through pixel alignment worked to identify corner cases using a threshold of 20%-pixel misalignment between predicted frames and actual frames. Finally, the third method of segmentation maps output similar results to the pixel misalignment method. Linear regression was used as a method of learning some of the trends occurring within the dataset, when identifying anomalies within driving scenarios. Some issues to point out were the misclassification of certain objects and scenarios within the benchmark, Cityscapes dataset. However, this research proved to be important in identifying several options in improving the robustness of autonomous driving technology, with reference to corner cases.

Introduction (10%)

Machine intelligence and learning is ever more important within the autonomous driving world, with predictions of various driving scenarios an important focus for developing a robust self-driving system. It is imperative that these models offer a high degree of accuracy, in uncertain road environments. A lot of the frameworks used today are reliant on deep learning, which is often known to be a “black box” type environment, where a lot of the machine intelligence in outputting predictions is unknown. With this uncertainty, any errors in prediction models can exacerbate the impact a faulty self-driving system can have on its surrounding environment. Hence, a type of ‘accountability’ is necessary within these systems, pointing out probable errors within prediction models, and where they may have occurred. This can be used to tweak parts of a prediction model, in order to improve the robustness of a self-driving system.

The method which is presently followed, in terms of assessing this accountability, is corner case and anomaly detections between predicted models and actual environments. According to [1], a corner case can be identified “if there is a non-predictable relevant object/class in a relevant location”. Thus, if there is an unusual occurrence within a driving situation, corner case detection can be used to relay this information to the self-driving system. Corner cases could be situations where it is predicted that two separate entities (i.e. a person and a sidewalk) are merged together in a predicted frame, which could point out a flaw in the self-driving system. Moreover, improbable occurrences, such as a person riding a skateboard down a highway, can also be considered as an anomaly. Anomalies are defined as a situation which does not conform to normal behaviour, and these fall under the subset of corner case characteristics [1]. These anomalies are of high importance, as self-driving algorithms get trained to work with increasingly ‘edgy’ data, in order to develop high robustness.

This investigation aims to propose three different methods to enable corner case detection in various road scenarios. These include identifying the difference between predicted frames of a video capture, and the actual frame, by assessing the alignment of pixels between the two frames, as well as object segmentation and recognition. The second method checks for misalignments between edges between two frames, as a means of determining if a corner case has been identified. Finally, the third method aims to compare the segmentation maps of the predicted frames and the actual frames. If there appears to be major discrepancies between various classes / labels within a predicted map and an actual segmentation map, this could heed warning to a potential corner case. These three models are then analyzed, with a confidence interval applied to their error (in most cases, the mean squared error) in order to output a conclusion of whether various scenarios do indeed produce possible edge / corner cases.

Background Review (15%)

The main paper of reference in this investigation is [1]. The paper offers a method of determining corner cases through its own segmentation and image prediction networks. These networks are then trained on the Cityscapes dataset (see [2]), which is also used as part of this investigation (and will explained further in the Dataset section). The Cityscape dataset also provides a benchmark for the image segmentation, by providing the accurate images, as to be compared with the predicted models. [1] follows through by applying semantic segmentation on images obtained from the dataset, through its an enhanced version of the DeepLabv3 framework, to output predicted frames and segmentation maps. In this investigation, simply the current DeepLabv3 framework will be used as a comparison to the framework used in [1]'s study. Results from the study in [1] demonstrate a 78.5% mean intersection over union (mIoU), when compared to the ground truth of the Cityscapes dataset, through the semantic segmentation method. Another method also followed in [1] is the image prediction model using PredNet, which is a predictive neural network that can learn how to follow synthetic objects in images, and, separately as comparison between two image prediction models, a predictive autoencoder [3]. In both cases, the mean squared errors are evaluated between the actual frame and the predicted image, in order to output a confidence of whether a corner case or anomaly was detected or not.

Preceding this valuable paper for this investigation, there has been significant research into corner cases within the autonomous driving field. In terms of anomaly detection, [4] investigates how the Mahalanobis distance (distance between a point P and a distribution of points D) can be used to evaluate distances from sensor data vectors. These distances could then be used to derive whether any anomalous activity was recorded on the self-driving system. Additionally, particle filtering and Kalman filtering have been proposed ideas, when assessing whether any novel situations, that have not been recorded before, appear within a certain dataset [1].

With regards to semantic segmentation, which rely on fully convolutional networks, all current architectures label all the pixels of a certain image [5]. For example, any pixels attributed to a person will be given the semantic label of a "human". Thus, a lot of current networks for anomaly and corner case detection attempt to identify if there are any discrepancies with predicted pixel positions and actual pixel positions, based on their labels.

Finally, with regards to image predictions, there has been much research invested into devising future image frames, through possible unsupervised learning [1]. For example, [6] was able to apply predictions on certain areas of images, which are extracted using a CNN (convolutional neural network). This method, using a long short-term memory autoencoder, was able to further ameliorate the outputs for classification tasks in predicted image frames [6].

Application/Dataset (15%)

As mentioned previously, the dataset being used for this investigation is the Cityscapes dataset. This dataset offers a more logical and semantic understanding for city-based scenarios, with labels and classes associated with various components of its images [2]. The specific city of focus within this dataset is Stuttgart, from which 5000 images were polled for various anomaly detection situations. Some features of this dataset included the ability to annotate various classes and labels of a frame based on pixel density, situational instances, and plain semantics. Approximately 30 classes for classification were available from this dataset; however, to reduce complexity of this investigation, the main classes of interest were the road, person, sidewalk and overall terrain. Defining these classes, the road was identified as any area in which a car normally drives. The person was identified as any human not within a car, within the frame, as well as any attributes in the possession of said human. The sidewalk was defined as any designated areas for pedestrians and cyclists, while the terrain was recognized as grass or any other parts not meant to be driven on by vehicles.

Based on the method (see proposed scheme and algorithms), through the DeepLabv3 framework, a pre-trained model was produced using the Cityscapes dataset. Segmentation maps of the various classes could be produced using the DeepLabv3 framework, as well as the image prediction network, as followed through the experimentation completed in [1]. According to [1], the Cityscapes dataset is “a widely used benchmark for semantic segmentation” in autonomous driving research. An example of a segmentation map for the Stuttgart dataset can be seen in the figure below, with various labels / colours given to the road, sidewalk, people, etc.

Figure 1: Example of Label Classification and Annotations in Stuttgart Frame

Through the DeepLabv3 framework, prediction models based on this output approximately 81.3% accuracy, for its pixel-level semantic labeling (evaluated through its Intersection over Union (IoU) on class-level). This accuracy was calculated based on the ratio of true positive pixels to the sum of true positive, false positive and false negative pixels in the prediction models. Thus, with the 81.3% accuracy, it was important to be wary of any discrepancies in data, where faulty predictions in pixels could potentially be due to misclassifications.

Proposed Scheme/Algorithms (10%)

Three methods were proposed in this investigation, each with different advantages and results.

Method 1: Misalignment with Next Frame Prediction

One major part of today's autonomous driving system is predicting trajectories of surrounding traffic participants. Corner case road scenarios in trajectory prediction are generally scenes where traffic participants create unusual movement. The first method of this research aims to capture such type of scenes by comparing the difference between a video frame with a predicted video frame at the same timestamp.

First, the stuttgart_00 dataset (from Cityscapes) is fed into the original pre-trained PredNet from [3], which generates a list of predicted video frames. Here, PredNet uses a convolutional Long Short-Term Memory network to predict the next frame based on previous movement trajectories of objects in the video. After this, the difference between each original frame and its corresponding predicted frames are calculated, represented by a list of Mean Square Errors (MSE). Lastly, a linear regression algorithm is used detect the outliers in the data, which represents the timestamps where there is unusual movement of traffic participants in the video. At the same time, linear regression also filters out the noise generated due to the blurriness in the predicted frames.

Method 2: Object and Road Type Misalignment

The second method sees the team use the DeepLabv3 semantic segmentation model, to find any misalignments and anomalies frame-by-frame. In order to understand the basis of misalignments, the team must first re-evaluate the definition of a corner case, as defined in the Introduction section. A specific rule set of situations that should not occur can be made, to satisfy this definition. For the purposes of this paper, in order to avoid complexity of specific cases, and because the team already knows the dataset and what types of anomalies exist, the ruleset is as follows:

> *Rule 1:* A person should not be walking (overlapping) with the road (which is separately classified from the sidewalk or crosswalk).
> *Rule 2:* A car should not be driving on the sidewalk.
> *Rule 3:* A car should not be driving on terrain.

Any misalignments are frames where any one of these rules are breached with a high percentage of overlay.

The stuttgart_00 dataset (from Cityscapes) contains 600 images, so this general set of steps is looped over for all 600 images. First, the image is run through DeepLabv3 to segment it into its different classes. The result of running this model produces a segmentation map. Then, for each ruleset, the two classes appearing within it are isolated (i.e. 'person' class from 'road' class). This

is done by searching the whole segmentation map and finding any label that does not belong to the labels of interest, and then assigning them a value of 255 (i.e. a value that means nothing in terms of the current labeling). The Canny edge detection algorithm, developed by John F. Canny in 1986, is then used for edge detection in the map [7]. It uses a multi-stage algorithm to detect a wide range of edges in images. This turns the segmentation map into a binary image where a value of 1 represents a pixel where the edge of a labeled object lies, and a value of 0 represents everything else.

At this point, two edge detection maps for the same image are produced, one for each label. The two maps are then overlapped and any overlapping pixels (edges) are counted by scrolling with a bounding box and checking all the pixels around it to see if two edges are overlapping. At this point we have all the pixels that are overlapped in the image, which are called the boundary pixels. To find the percentage overlap between these pixels with respect to the object and class, the minimum amount of pixels between the two labels are found, and this value is divided by the boundary pixels present within the image (theoretically speaking, a high percentage overlap would mean that most of the object is overlapping with another class that it shouldn't be, as defined by the anomaly algorithm). This method is followed for all the other rules in the ruleset.

Method 3: ROI Misalignment with Next Segmentation Map Prediction

The third method is similar to that of the second method. The DeepLabv3 framework is used with the same pretrained model on the Cityscapes dataset just as before. However, the result is derived differently: in this case, the segmentation map of the current frame is compared with that of the previous frame in terms of their mean squared error (MSE). The rationale behind this decision is similar to that found in the first method – if there is a sufficient difference in how the frames are segmented, this means there must have been a sudden change in the frame. Performing this operation on the segmentation is useful because the segmentation of each object is uniform in colour. Thus, the calculation for MSE will not be prone to error imposed by colour differences within objects. Furthermore, each object will be a distinctly different colour so the presence of an unexpected object in one frame in relation to its previous frame will introduce a clear, and discernible disparity in the MSE.

Based on these three methods, it would be possible to obtain mean-squared error and misalignment graphs, in a similar structure to the benchmark achieved in [1], as shown in the figure below. Mainly looking at the red line, through the analysis in [1], peaks for corner case detection were found in instances of humans walking on the road, as an example.

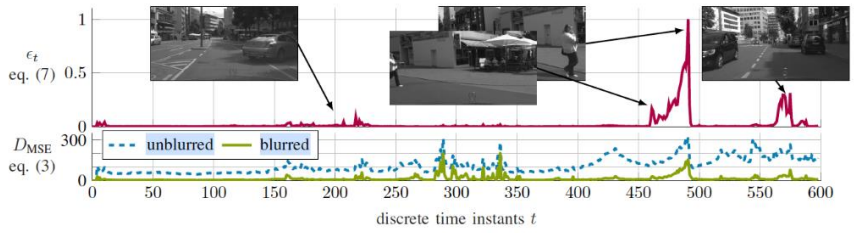

Figure 2: Benchmark MSE Output from [1]

Experiments and Results (40%)

Method 1: Misalignment with Next Frame Prediction

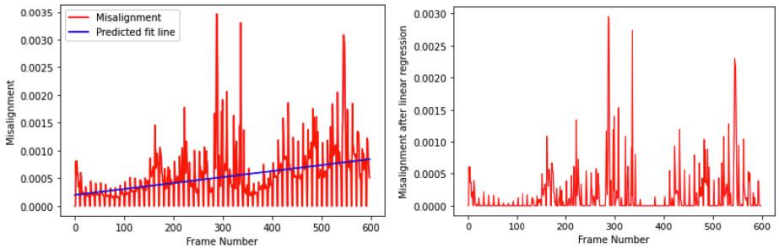

Figure 3: Misalignment (in MSE, w/ linear regression filter – blue line) between a frame and its corresponding predicted frame

After taking a manual threshold of MSE = 0.0008, 77.8% of nine corner cases detected align with the detection result from the benchmark. The few exceptions are caused by the model's inability to quickly adjust to the rapid change of lighting in the frames. At the same time, there is more noise in this method's result compared to the benchmark. Both problems are present, since the final benchmark data uses the Predictive Autoencoder instead of PredNet. This autoencoder is trained on the Cityscapes dataset, adapting to its frame rate, vehicle speed, image resolution, image aspect ratio, and common object movement. Lastly, there are discontinuities in the output, occurring once every ten frames, since PredNet predicts ten frames at every instance.

Method 2: (Object and Road Type Misalignment)

Table I: Key Regions for Anomaly Detection (Method 2)

Frame of Anomaly	Cause of Anomaly Flag
~160	Person/Road
~170	Person/Road
~180	Car/Terrain
~205	Car/Sidewalk
~218	Car/Sidewalk
~460	Person/Road
~470	Person/Road
~490	Person/Road
~568	Person/Road & Car/Sidewalk
~573	Person/Road & Car/Sidewalk
~585	Person/Road & Car/Sidewalk

The following figure displays the results of the overlapped individual rulesets. For reference, the ground truths, as seen in the results of [1], are in 11 key regions. Table I summarizes these regions and their flags. It should be stated again that the images pass through a pre-trained Cityscapes segmentation dataset with 81.3% accuracy. Thus, one immediate source of error in the results is misrepresentation of objects due to faulty initial labelling.

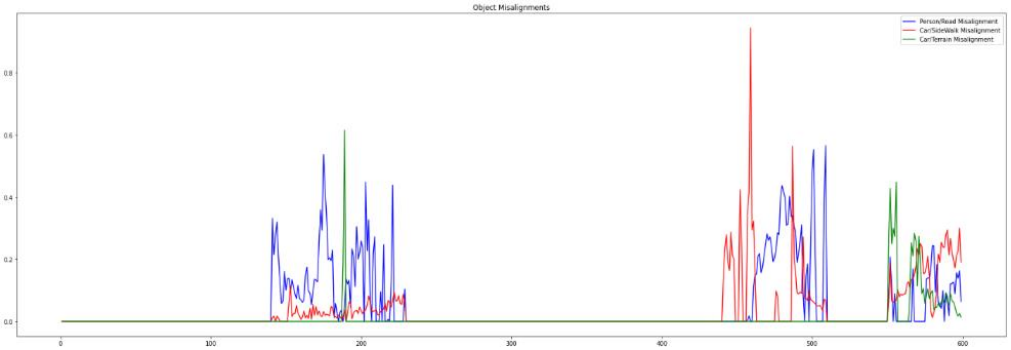

Figure 4: Object Misalignments for Person/Road (Blue), Car/Sidewalk (Red), Car/Terrain (Green) – Frame vs. % Misalignment

First, focusing on person/road misalignments, scores above 0.2 are registered, meaning that 20%-pixel overlap was seen between the two classes. Any region above 0.2 overlap is therefore seen as viable data. Although this may be a small confidence interval, the goal of the experiment

is to mainly recognize that some sort of overlap exists between predicted and actual images. Next, analyzing car/sidewalk misalignments, the amount of overlap is very small in frames 205-218. There was much more overlap for frames 568-585, exceeding the 0.2 threshold. However, significant error was observed in frames 460-510, a region that should see only person/road misalignments. The main source of this error is misclassification in the pre-trained dataset. Finally, car/terrain misalignments demonstrated significant overlap in frame 180 (above 0.6). There was a lot of error at the end of the model, due to an 'irregular' overlap. The top and sides of the cars overlapped with the terrain, which is not a specific region of interest.

In future iterations, a slightly different approach on checking overlaps could be followed, to lessen the impact of errors propagated from the dataset. In this approach, rather than checking all possible overlaps, it might be more efficient to check the bottom 20% - 30% of an object overlapping on a class. The pseudocode for this method can be seen below.

```
# Fixating on only bottom 30% of the image:
# segmap = <>
# person_map = segmap[segmap == person]
# Do connected components on all person labels to get instances of people
# Take bounding box on each person
# Take the top 70% of the height of the bounding box
# and set to background on the original seg_map
# input modified seg_map into get_overlap_perc for more confident overlap
```

Figure 5: Pseudocode Example for Future Iterations of Method 2

Method 3: ROI Misalignment with Next Segmentation Map Prediction

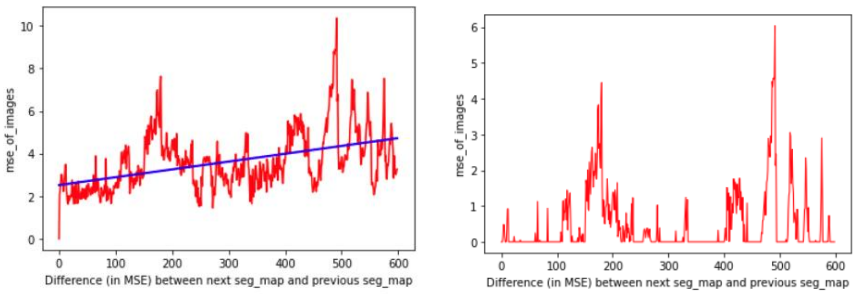

Figure 6: Left - MSE between Each Frame with Linear Regression-based Predicted Fit (Blue Line); Right - MSE with Linear Regression as Confidence Threshold

Table II: Key Regions for Anomaly Detection (Method 3)

Frame of Anomaly	Correlated anomaly
160	Person/Road
208	Car/Sidewalk
236	Car/Sidewalk
441	Person/Road
490	Person/Road
573	Person/Road & Car/Sidewalk

Using peak detection with a threshold of 1 (removing noise in the signal), the following frames in Table II are introduced. Note, that the heading in the table is changed to 'correlated anomaly' based on the correspondence of these findings with visual validation of the anomaly. This distinction is important because this method does not keep track of the overlap of objects as with the previous method. The results, from Figure 6, do substantiate the hypothesis that the sudden appearance of objects will alter the MSE significantly. This is clearly discernible through the segmentation maps of the frames. This method is prone to the flaws of the method #2 since they are both based on the same underlying model. In order to improve this method, one potential feature would be to include a way to determine whether or not the frames are a part of the same scenario as is the case with frames 208-236 and 441-490 and then return them accordingly.

References (5%)

[1] J. Bolte, A. Bar, D. Lipinski and T. Fingscheidt, "Towards Corner Case Detection for Autonomous Driving", *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019. Available: 10.1109/ivs.2019.8813817 [Accessed 01 April 2020].

[2] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Available: 10.1109/cvpr.2016.350 [Accessed 04 April 2020].

[3] W. Lotter, G. Kreiman and D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning", *ICLR*, 2017. [Accessed 10 April 2020].

[4] R. Lin, E. Khalastchi and G. Kaminka, "Detecting anomalies in unmanned vehicles using the Mahalanobis distance", *2010 IEEE International Conference on Robotics and Automation*, 2010. Available: 10.1109/robot.2010.5509781 [Accessed 14 April 2020].

[5] A. Liu, Y. Yang, Q. Sun and Q. Xu, "A Deep Fully Convolution Neural Network for Semantic Segmentation Based on Adaptive Feature Fusion," *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, Zhengzhou, 2018, pp. 16-20.

[6] N. Srivastava, E. Mansimov and R. Salakhudinov, "Unsupervised Learning of Video Representations Using LSTMs", *ICML*, 2015. [Accessed 11 April 2020].

[7] L. Yuan and X. Xu, "Adaptive Image Edge Detection Algorithm Based on Canny Operator," *2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS)*, Harbin, 2015, pp. 28-31.