Contents

List of Figures

List of Tables

1

Timing Analysis of Synchronous Neural Networks

2

Run-time Enforcement of Synchronous Neural Networks

3 Run-time Verification of Synchronous Neural Networks

3.1 Methodology

The system designed for this case study was made to reflect a Autonomous Vehicle (AV) and its object detection mechanisms. The system used multiple techniques to tackle the inherent issues of the AV system, i.e. weakness to perturbed inputs and misclassification detection. The system's sensors include an overhead, 360° Light Detection and Ranging (LiDAR) apparatus, and a single, frontal facing camera. A solitary camera was sufficient to prove the efficacy of this solution, however it is to be noted that AV systems generally use multiple cameras, facing different directions, so that the controller can make properly informed decisions. The system used can be seen in Figure 3.1.

The LiDAR for this system was accurate 93% of the time [5], to closely simulate a real LiDAR system. The simulated camera outputs consisted of test images from both the Visual Object Classes (VOC) [1] and German Traffic Sign Recognition Benchmark (GTSRB) [4] datasets, in a combination of people, vehicles and various traffic signs. The LiDAR and camera outputs were handled by different parts of the controller. The camera outputs were fed into a Meta Neural Network (MNN) (see Figure 3.2) where they were classified by shape, colour and object type.

Utilising synchronous semantics, a Meta Neural Network (MNN), containing three other MNN ensembles, was created. Each ensemble synchronously combined the outputs of three different convolutional Synchronous Neural Networks (SNNs) [3], providing increased prediction accuracy for shape, colour and object type. These ensembles ran in synchronous concurrency, each taking four logical ticks to run. The outputs of each ensemble were then combined into batch of outputs forming the *predicted output.*

The system controller was encapsulated by a run-time enforcer [2] that used sensor fusion to check for misclassifications made by the MNN. A safety automata (timed automata) **??** was designed to verify predictions made by the MNN and ensure utmost safety at all times. The automata started in a safe state, where control of the vehicle was autonomously handled by the system controller. If a misclassification was detected, the enforced policy entered an unstable state, still under autonomous control. Once enough time passed without further misclassifications, the vehicle entered the safe state again. However, if another misclassification was detected while unstable, the enforced policy entered a violation state and forced control of the AV to the driver. The vehicle would not enter autonomous mode again until the system was restarted. A diagram of the enforced policy's safety (timed) automaton is shown in Figure 3.3. This type of run-time enforcement, where neither the inputs nor outputs of the sensors or controller are enforced, has been termed as *run-time verification. Run-time verification* refers to the verification of system parameters during run-time, while ensuring that the system is aware of any failed guards in the enforced policy.

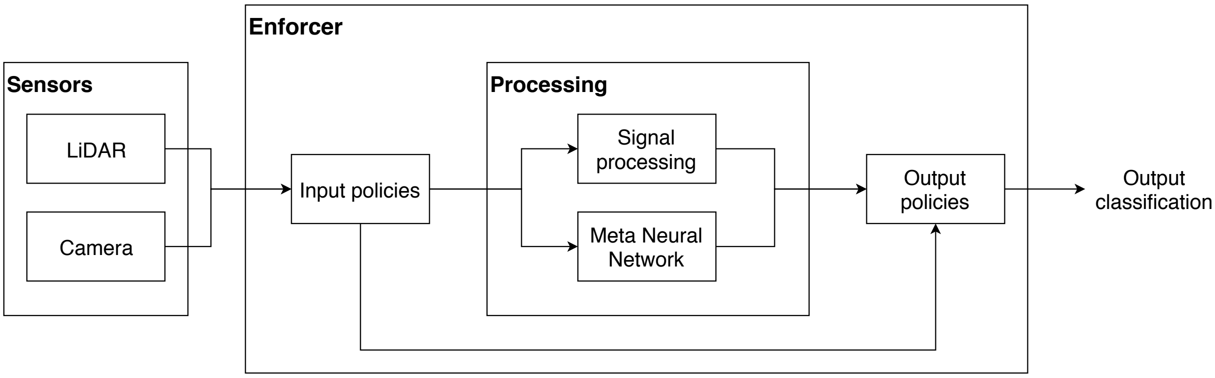


Figure 3.1: Block diagram showing the AV system with enforcer

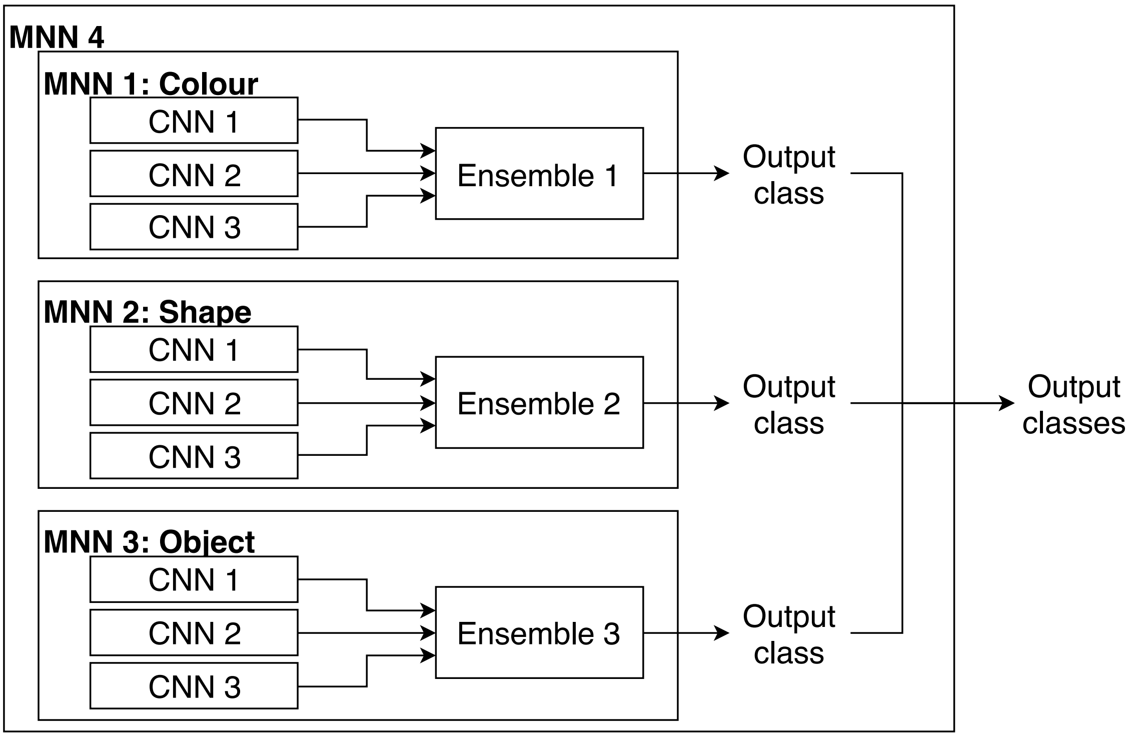
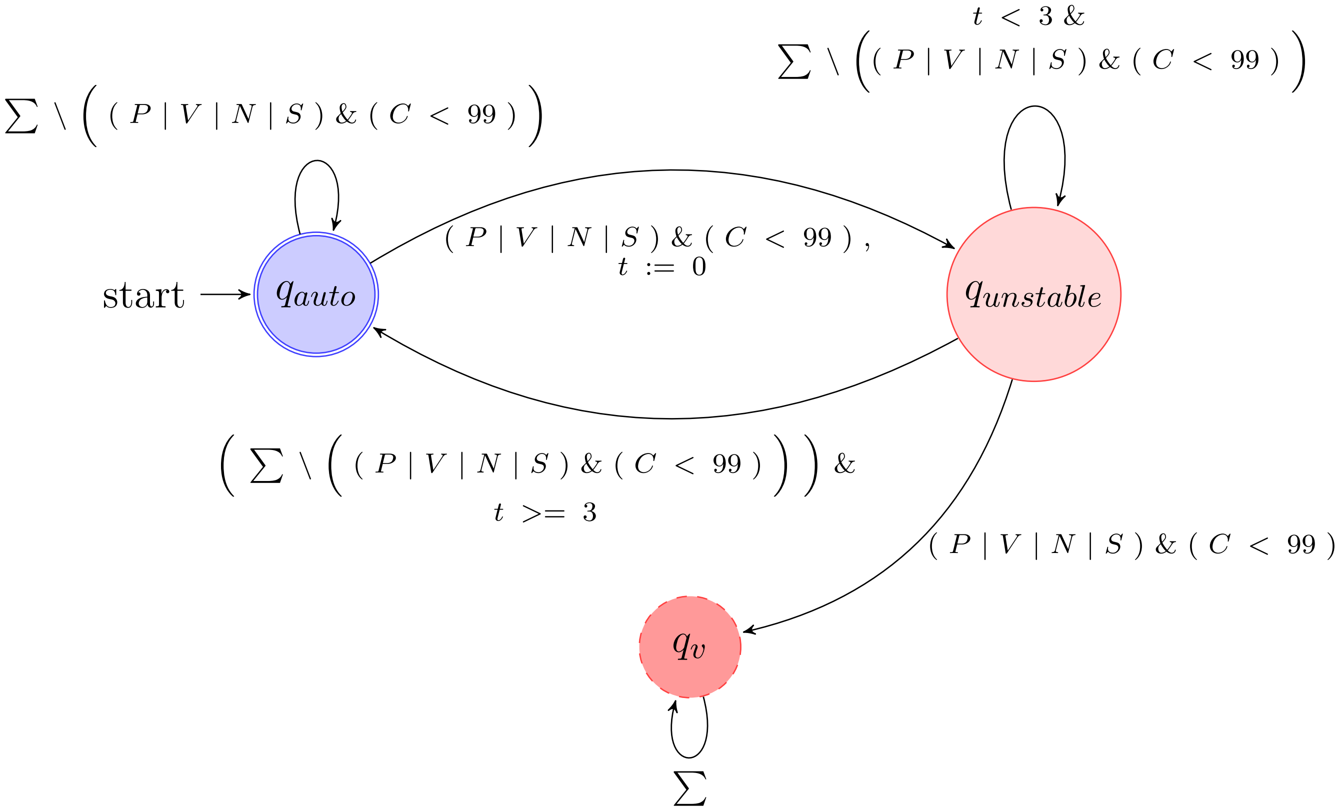


Figure 3.2: Block diagram showing the Meta Neural Network ensemble


$\Sigma \setminus \Big((P \mid V \mid N \mid S) \;\&\; (C < 99) \Big)$
$t < 3 \;\&$
$\Sigma \setminus \Big((P \mid V \mid N \mid S) \;\&\; (C < 99) \Big)$
start
q_{auto}
$(P \mid V \mid N \mid S) \;\&\; (C < 99),$
$t := 0$
$q_{unstable}$
$\Big(\Sigma \setminus \Big((P \mid V \mid N \mid S) \;\&\; (C < 99) \Big) \Big) \;\&$
$t >= 3$
$(P \mid V \mid N \mid S) \;\&\; (C < 99)$
q_v
Σ

- P: Misclassification of a person.
- V: Misclassification of a vehicle.
- N: Classification of an object when there is nothing.
- S: Misclassification of a traffic sign.
- C: Confidence rating of the SNN classification.
- t: Timer for the unstable state.

Figure 3.3: Enforcer policy for the AV prediction system

Test case	Person	Vehicle	Nothing of relevance	Street sign	Total
Number of misclassifications	3.45	9.29	6.65	38.56	57.96
Caught misclassifications	3.06	8.83	6.51	37.45	55.85
False negatives	3.87	2.16	0.28	7.53	13.84
Missed misclassifications	4.26	2.62	0.42	8.64	15.94

Table 3.1: Table showing results of the AV prediction SNN using perturbed images

3.2 Results

A
Appendix

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] S. Pinisetty, P. Roop, S. Smyth, N. Allen, S. Tripakis, and R. von Hanxleden, "Runtime enforcement of cyber-physical systems," vol. 16, pp. 1–25, 09 2017.

[3] P. S. Roop, H. Pearce, and K. Monadjem, "Synchronous neural networks for cyber-physical systems," in *MEMOCODE '18 Proceedings of the 16th ACM-IEEE International Conference on Formal Methods and Models for System Design*, 2018.

[4] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457

[5] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, "LiDAR and Camera Detection Fusion in a Real Time Industrial Multi-Sensor Collision Avoidance System," *ArXiv e-prints*, Jul. 2018.