ASIMS: Acceleration Spectrograms Based Intelligent Mobility System for Vehicle Damage Detection

Sara Khan[1], Mehmed Yüksel[2] and Andre Ferreira[3]
[1]*Robert Bosch GmbH, Renningen, Germany*
[2]*DFKI GmbH Robotics Innovation Center, Bremen, Germany*
[3]*Robert Bosch Car Multimedia S.A., Braga, Portugal*

Keywords: Automobile, Machine Learning, Damage detection, Cosmetic Damages, Inertial Sensors, Autoencoders.

Abstract: Every vehicle is susceptible to several types of small physical damage such as dents and scratches. These damages can be seen as cosmetic damages as they impact the vehicle's visual and value but do not alter its main functions. Vehicle owners, insurance companies, and the car-rental/taxi-service companies are especially keen to detect the events that generate these kinds of damages. The ability to detect impact events is valuable to monitor the occurrence of possible damages to the vehicles. In this paper, we present a novel acceleration spectrogram-based Machine Learning (ML) approach for dynamic (real-time) small vehicle damage detection using inertial sensors. Inertial sensors are low-resource consumption sensors, which makes the proposed solution economical. Conventionally, inertial sensors are used in the airbag control system but they are not developed to detect impacts that generate minor damages. Most of the previous work on small impact detection either uses smartphone inertial data which is not accurate or focuses on static damage detection based on image sensory inputs. Our intelligent impact and damage detection ML-based system uses autoencoders as an automatic feature extractor using acceleration spectrograms and classifies the sensory encoded feature representation into damage or non-damage. It can achieve an accuracy of 0.8. This approach sets the stage for various potential research directions in damage detection.

1 INTRODUCTION

The common interest of car manufacturers, and car owners - from individual persons to car-sharing companies, passengers, or other players (insurance, inspection and service companies, etc.) is to monitor the vehicle status in various aspects.This vehicle status information is imperative to detect risks and anomalies or accurately predict malfunctions that may occur in non-autonomous or autonomous driving functions to avoid unexpected error cascades.

Traditionally, inertial sensors have been used in automotive systems for airbag control (Shi et al., 2008) and can help in detecting big damages. But what if we want to detect small damages like minor dents or scratches? Small damage detection information has a variety of applications in new mobility services like car-sharing and ride-sharing applications where vehicle damage information is very crucial to know. For example, a ride-sharing application involves a user picking up a vehicle from one point, using it for personal riding, and then dropping the vehicle at the vehicle drop point. If damage occurs during the user's ride, then the damage event detection system provides the damage information to the stakeholders even if they are imperceptible to take the corresponding actions for the occurred damage. Also, this damage prediction information can be used in identifying the damages easily and in turn reduces the tedious process of vehicle inspection by insurance companies while claiming insurance in case of vehicle damages (Li et al., 2021).

Thus, in this paper, we propose a small damage detection system where the goal is to detect events resulting in damages to the vehicle and classify the damages as cosmetic damages and non-damages. A damaging event can be defined as an event that has a physically negative impact on the vehicle structure but may not necessarily result in the vehicle malfunctioning. Previous research works in the field of damage detection are static large damage detection and use camera sensors data as the main input. On

Khan, S., Yüksel, M. and Ferreira, A.
ASIMS: Acceleration Spectrograms Based Intelligent Mobility System for Vehicle Damage Detection.
DOI: 10.5220/0011763200003479
In *Proceedings of the 9th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2023)*, pages 179-186
ISBN: 978-989-758-652-1; ISSN: 2184-495X

the contrary, our work focuses solely on small damage detection with inertial sensors which is a light weighted and efficient solution. Also, we build our model using real-life data, which has been not used before in general. In this paper, we propose a method for small damage detection using a machine learning approach termed as ASIMS. Traditional machine learning involves computing features to train machine learning models. But this process has high complexity and requires domain expertise (Vejdannik et al., 2018). Thus, we compute feature representation using the deep learning-based automatic feature extractor method approach. And, use this compressed representation for binary damage detection and classification.

2 RELATED WORK

Vehicle Damage Detection is an established domain in the literature. The damages to automobiles can either be significant or small. Figure 1 gives an overview of cosmetic (or small) damages and significant (or large) damages that are present on vehicles. Cosmetic damage image contains a slight scratch on the vehicle and negatively affects vehicle's appearance. Alternatively, significant damage image contains one or more damages to the vehicle's body which may impact its functionality too. In the automotive industry, there has been a substantial amount of prior research on damage detection. Most of the previous work concentrates on static damage detection using camera sensors. In the below sections, we summarise the current work on damage detection and also describe how our work is different from previous work and points in a different direction.

Cosmetic Damage Significant Damage

Figure 1: Kinds of damages.

2.1 Damage Detection Using Cameras

Researchers in (Li et al., 2021) aim to automatically detect and classify vehicle damages, to fasten the process of insurance claims. They have used a combination of deep learning and transfer learning for high performance and overcome training data limitations. Unlike other works, the authors concentrate on more relevant inputs by reducing the noise and irrelevant items. For the dataset, the authors have used web scraping to extract damages from Google images. Furthermore, they are manually labeled into different damage categories (e.g., bumper damage, no damage, glass damage, etc.). They first use Mask Regional-Convolutional Neural Network (Mask R-CNN) to crop the damaged car and remove other irrelevant items. And then (O'Shea and Nash, 2015) pre-trained on ImageNet (Deng et al., 2009) is used for classification. Using transfer learning, eight different architectures are evaluated. Among them, (Simonyan and Zisserman, 2015) performs the best with 87.5 accuracy. Along similar lines, the work in (Dwivedi et al., 2020) uses CNN and pre-trained models for classification and detection into 7 different classes. In (Kyu and Woraratpanya, 2020), the authors created their own dataset consisting of approximately 1200 images. They categorized damage severity into minor, moderate, and severe using a predefined VGG-16 model with L2 regularisation. In another work (Singh et al., 2019), the authors proposed an end-to-end system for claiming insurance for car damages. To do so, the authors propose a model that takes images as input and classifies them into non-damage or damage. It further localizes damage and classifies its severity into mild and severe. It also provides additional decisions on whether the damaged part needs to be replaced or not. For purpose of modeling, they have used instance segmentation models like PANet(Liu et al., 2018), Mask R-CNN, and ensemble version of both using transfer learning based on VGG16.

The drawback of camera sensor-based damage detection is it cannot be used for real-time detection. The cameras present in autonomous driving systems are used to capture surroundings rather than the car itself (Ess et al., 2010). The data from these camera sensors is huge and requires a lot of computation. Also, it is not economical to have such an expensive sensor setup for a fleet of shared vehicles.

2.2 Damage Detection Using Audio

On the contrary, not all of the previous work relies on camera sensors for damage detection. The authors in (Sammarco and Detyniecki, 2018) use audio signals from car impact to detect accidents using their created dataset (Sammarco and Detyniecki, 2019). They extract time and frequency domain features of input signals and use them for classification. Their proposed model is able to differentiate between crash sounds and other in-vehicle sounds. In (Choi et al., 2021), the authors use multi-modal data including both audio and video signals for crash detection. A Gated Recurrent Unit (GRU) and Convolutional Neural Network (CNN) based classifiers are used for the input. The results show that ensemble classifiers perform better than a single classifier. In (Hashimoto et al., 2019), the authors proposed a method to detect abnormal vibrations in cars. They used piezoelectric sensors waveform as an input and feature processing using Mel-frequency coefficients, and CNN for modeling provided optimal results. There are some other applications of damage detection in road damage and structural health monitoring using audio data. The work present in (Gontscharov et al., 2014) presents an approach for the automatic detection of minor vehicle body damages. The author uses a sensor network integrated vehicle body and uses acoustic vehicle noise level for structural vehicle damage detection.

2.3 Damage Detection Using Inertial Sensors

Conventional solutions use accelerometer data to detect more or fewer spikes caused by sudden changes in acceleration in one or more axes (Punetha et al., 2012). The abrupt changes in G-force trigger the air bag control for vehicle safety. But it is not able to detect impacts that are minor or cosmetic. Many previous works have used inertial sensors in smartphones for accident or incident detection in vehicles. The work in (Zaldivar et al., 2011) combines the acceleration sensors present in smartphones with vehicle's onboard diagnostics. With their onboard interface, they are able to detect accidents. Another work (White et al., 2011) presents a smartphone-based accident detection system named WreckWarch which records forces through accelerometers experienced in collisions. But none of the smartphone-based systems are accurate as smartphone data is hard to analyze due to calibration, noise, and rotation issues.Also, relying solely on acceleration data can lead to incorrect predictions in many situations. Bumps, potholes, and poor road conditions cause false alarms, while stationary rear-end collisions can be classified as normal acceleration (Sammarco and Detyniecki, 2018). Works in (Gontscharov et al., 2014), (Sammarco and Detyniecki, 2018) and (Punetha et al., 2012) motivates us to build a data-driven intelligent system for detecting small damages in real-time.

The main contributions of our work are:

1. Novel Machine Learning(ML) based approach using acceleration signal spectrogram for small vehicle damage detection.
2. Use of damage data acquired through field tests for empirical evaluation.

Table 1: Examples of different damage and non-damage events.

Damage Event	Non-Damage Event
Passive vehicle collision	Sun visor
Active vehicle collision	Trunk opening /closing

3. Overview of state-of-the art in automotive damage detection.

3 SMALL DAMAGE DETECTION

Small Damage Detection is a data-driven damage detection system that detects damages using accelerometer information. It is an event-based mechanism. Events are triggered while a person is riding the vehicle. Events are categorized into damage and non-damage events (Gontscharov et al., 2014). Table 1 shows some examples of damage and non-damage events. For example, a passive vehicle collision is a damage event that occurs when another vehicle or object hits our vehicle.

The damage detection system work as shown in Figure 2. When the sensory data hits a certain threshold, the event is triggered and the information within a certain window is collected. This raw information is passed through pre-processing step and fed into a trained machine-learning model. The prediction of the model is either damage or non-damage(or background). If damage occurs, the information is sent to the cloud service for further steps. The prediction system follows the CRISP-DM framework (Wirth and Hipp, 2000), which includes data understanding, and data preparation as part of data analysis, modeling, and evaluation.

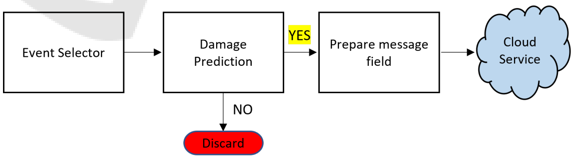

Figure 2: Small Damage Detection System.

3.1 Dataset and Pre-Processing

The primary source of data is real-field test data collected by creating the damage scenarios on vehicles. A small piece of hardware containing inertial sensors is mounted on the windshield of the car. Some of the cars used in data collection are BMW 5 series, BMW i3 and Mercedes GLA 180. The roads that were tested on to collect the data were categorised as asphalt, stone road, mud, dirt, snow and gravel. There were 10 different drivers who performed the maneuvers with the speed range from 10 km/hr to 100km/hr under different weather conditions like sunny, raining and cloudy. The database includes accelerometer and gyroscope sensory information along with timestamps. A continuous stream of data is collected keeping possible damage and non-damage events. Some examples of such events are mentioned in table 1. We gather those raw signals that cross a predetermined threshold as damaging occurrences have high values of inertial sensors. Then, a window size of one second is collected where it captures information of 100ms before the threshold and 900 ms after the threshold. The reason to use 100 ms before threshold is to give a small context before the event. This event information contains a damaging or non-damaging event. The inertial sensors are sampled at a frequency of 1600 Hz. Thus, for each event (of window size 1), it contains the x,y, and z-axis of acceleration values along with car-specific information and corresponding damage label. After this, the signals are passed through a low pass filter of 220Hz.

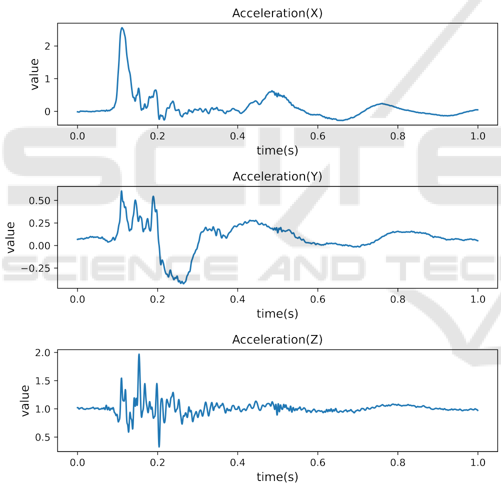

Figure 3: Acceleration x-axis, y-axis and z-axis for a damaged event.

Figure 3 and 4 depict the measurements from accelerometer and gyroscope in all three directions- X, Y, and Z axis. Acceleration and Gyroscope are both digital signals. Acceleration measures linear acceleration while the gyroscope provides angular velocity. In the above-discussed figures, we can see the rate of change of acceleration differs from the rate of change of gyroscope concerning time duration. Both of these figures are examples of a damage event in which a vehicle hits the left door and a dent is formed. The high spikes between intervals 0.1 and 0.4 clearly show the occurrence of a damage event that results in a dent. This work uses acceleration information only.

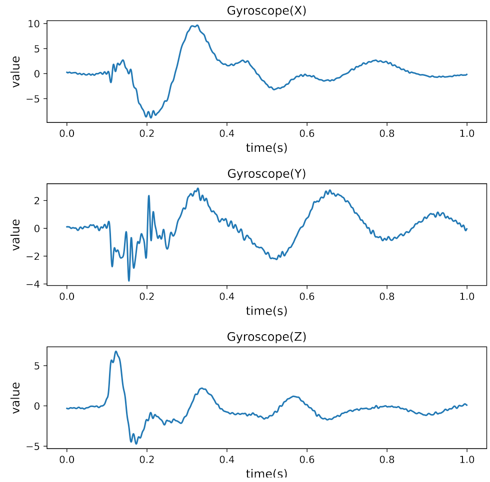

Figure 4: Gyroscope x-axis, y-axis and z-axis for a damaged event.

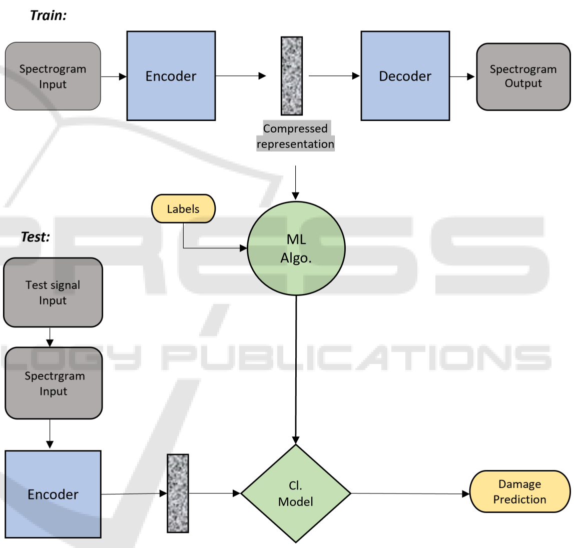

Figure 5: Modelling framework.

3.2 Methodology: ASIMS

The modelling approach of ASIMS system is divided into two sub-sections. The first sub-section explains the usage of deep learning as an automatic feature extractor to encode feature representations, which is trained using unsupervised learning. The second sub-section uses this encoded representation for a binary classification task, which is trained in a supervised technique using damage labels. Figure 5 depicts the modeling framework at the train and test level. It will be explained in the below sections.

3.2.1 Autoencoders as Feature Extractor

Autoencoder is an unsupervised learning technique that is used to reduce dimensionality and help reduce noise (Liu et al., 2016). It seeks an input image, converts into a latent or compressed representation, and reconstructs it as an output image. In recent years, many previous works have suggested autoencoder as an automatic feature extractor. Although it's a black box, it is fast and doesn't require domain expertise. As a result, we will employ it as a tool for feature extraction in this study.
To use autoencoders, we first convert the generated event signal information of window size of one second into a spectrogram image using Short-Time Fourier Transform (STFT) (Mateo and Talavera, 2017) at a sampling frequency of 1600 Hz. A spectrogram is a visual representation of a signal (French and Handy, 2007). This results in three spectrograms: x, y, and z for each window sample. Figure 6 depicts the

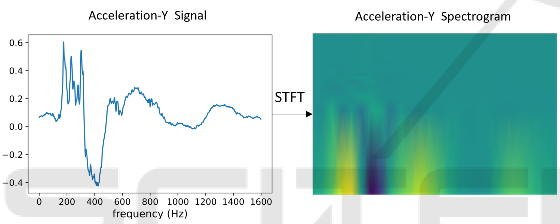

Figure 6: An acceleration y-axis signal into acceleration spectrogram through STFT.

transformation of the y-axis of the acceleration signal into a spectrogram. These generated spectrograms are trained using autoencoders as shown in the training section of Figure 5. The spectrograms are fed into an encoder which creates a compressed encoded representation. This encoded representation captures the characteristics of input data. Then, a decoder reconstructs the output based on latent representation. For our study, we discard the decoder and use latent representation to train our damage classification model.

3.2.2 Damage Prediction

The goal of damage detection is to build a Gradient Boosting (GB) based binary classification model, which takes the input as flattened encoded representation of a spectrogram obtained from autoencoder and outputs the damage predictions. This flattened encoded representation of an input image and the corresponding damage label are used as input-output pairs for supervised training. The trained model is then used for prediction as shown in the test section of Figure 5.

3.3 Implementation

This work was implemented in Python language. The modeling was done in two parts. The first sub-part involves converting signals into spectrograms. This was done using Matplotlib library (Hunter, 2007). The second subpart involves creating an encoded feature representation using auto-encoders. This was done using TensorFlow (Abadi et al., 2015) library. Grayscale spectrograms of X, Y, and Z were fed channel-wise into the autoencoder. Grayscale spectrograms contain the same information as color-mapped spectrograms (French and Handy, 2007). The colors only depict the aesthetic aspect of a spectrogram. The autoencoder architecture was a three-convolution layer architecture with batch normalization on each layer where the encoded representation contains 1024 features. The neurons in each layers are 200,100, 1 respectively and a 3*3 convolution filter on each layer. Figure 7 shows the training validation curve for a hundred epochs and a learning rate of 0.0001. The training was performed using Tesla V100-SXM2-32GB GPU containing 32 GB RAM. The data contained around unlabelled 50K image samples for feature representation. It was split into 80 percent training data and 20 percent validation data. The ratio between number of non-damaged samples to damaged samples is 99:1. This high imbalance data is a reflection of real-world scenario where damage cases are pretty low. The second part involves

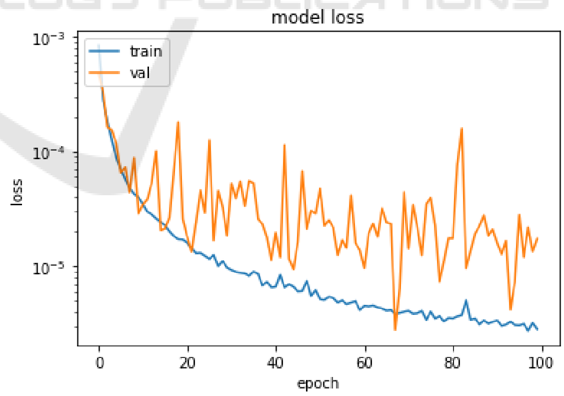

Figure 7: Training and validation curve for autoencoder architecture.

fitting the whole training data with encoded representation and corresponding binary labels in a supervised learning fashion. Our modeling framework uses Gradient Boosting (GB) as a classification model. Our proposed model was compared against Random forest (RF) and Gradient Boosting (GB) classical ML models without encoded representation. All ML algorithms are implemented using Scikit-learn (Pedregosa et al., 2011) library with their default settings. The results were evaluated using labeled training data containing approximately 2500 spectrograms for training and testing. The training and the testing split is 4:1.

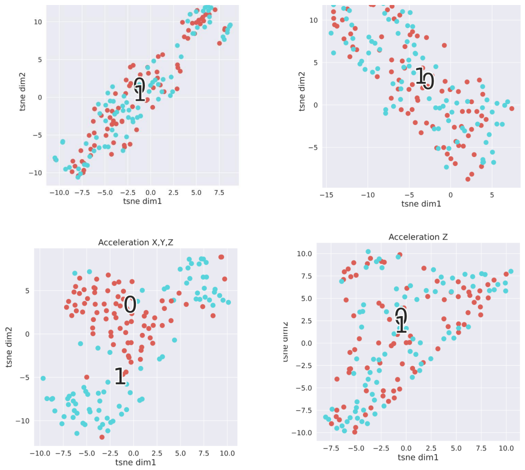

Figure 8: t-SNE representation using k-means. A value of 1 represents damage and 0 represents non-damage.

Figure 8 shows the data distribution of encoded training data using t-SNE (van der Maaten and Hinton, 2008) and clustered using the k-means algorithm into two clusters. The data contains approximately 150 equally balanced damaged and non-damaged samples. The orange color and turquoise color represent damage and non-damage samples respectively. As we can see that individual representations of each axis are non-separable. But when we consider the encoded representation altogether, two clusters are formed. Does this convey that clustering is enough for our problem? So, we increased the number of samples and performed clustering. Figure 9 shows the training data distribution with an increased number of samples. The data becomes more unbalanced as the number of samples rises. Thus, the clustering becomes intractable with increase in number of samples. This demonstrates that a simple unsupervised technique like clustering is not suitable for this problem. Therefore, this enabled us to design supervised learning based classification.

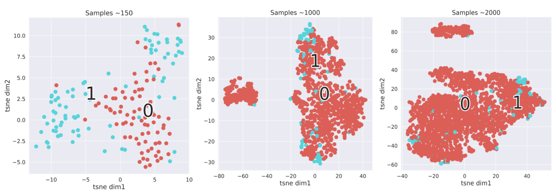

Figure 9: Data imbalance ratio increase with increase number of samples

4 RESULTS AND DISCUSSIONS

We use a test dataset containing 50K unlabeled samples for autoencoder results. The training part is evaluated using Linear Algebra (LA) norm. LA norm is a mean squared metric that measures the squared difference between the input and the reconstructed image. Table 2 displays LA norm results for the test dataset containing around 50K samples. The second column and third columns depict the minimum and maximum error between any image and its prediction of all the test samples. The low values explain that the feature extractor model has been able to generalize well. Figure 10 shows an example of an x-axis grayscale spectrogram with its test image and its reconstruction.

Table 2: Test results using autoenoder. L-R columns: number of test samples, minimum , maximum , mean and standard deviation of errors.

No.	Min.	Max.	Mean	SD.
50K	0.232	0.545	0.121	0.5342

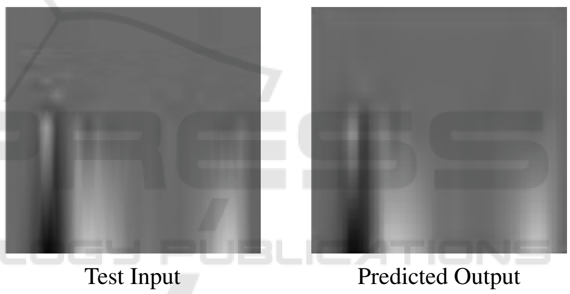

Figure 10: An input with dimensions 256*256 containing x-axis acceleration spectrogram image and its corresponding reconstruction with 0.079 error.

The latent representation of the trained model is then used for classification and compared to non-encoded representations. For the damage detection classification, we use metrics calculated using a confusion matrix for evaluation. Figure 11 summarises the results of the confusion matrix for our model that is trained and tested on approximately 500 spectrograms each. Every sample containing three spectrograms images per sample. A proportion of 0.14 and 0.05 are False Negatives and False Positives respectively. Door closing is a non-damage event that the model frequently interprets incorrectly and predicts as damage. By including better data in the model, this can be improved. Currently, STFT is used to construct spectrograms. Instead, Continuous Wavelet Transform (CWT) (Yunhui and Qiuqi, 2004) can be applied. As it provides finer details by creating bins of dynamic sizes. Table 3 summarises evaluation results. The training data contains around 6000 spectrogram images (three for each sample) and the test dataset contained approximately 1200 images. Accuracy, Precision, and F1 score are the metrics computed from the confusion matrix for evaluation. Our proposed model ASIMS is compared against Gradient Boosting (GB) present in the second row and Random Forest (RF) respectively. Both of these classification models are trained without encoded representation. The high values of accuracy and F1-score confirm that our approach outperforms the traditional ML models. Accuracy can be misleading when classes are imbalanced. Therefore, we measure F1-score as well. Thus, encoded representation has a positive effect on performance. Also, the number of features used in our model is one-third of the feature used by other models.

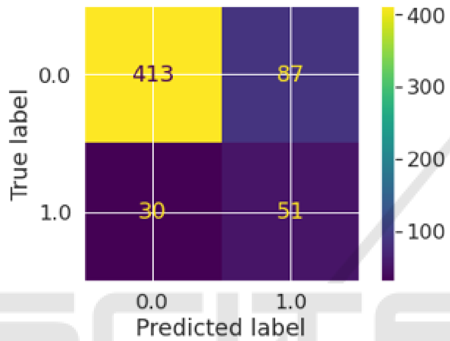

Figure 11: Confusion matrix for test data where 1 represents damage and 0 represents non-damage.

Table 3: Test results for damage classification.

Model	Features	Accuracy	Precision	F1
ASIMS	1024	**0.82**	0.84	**0.89**
GB	3072	0.70	1.00	0.09
RF	3072	0.20	0.82	0.82

5 CONCLUSIONS

This research introduces a semi-supervised machine learning approach for small damage detection based on acceleration spectrograms. Autoencoders are used to construct the encoded feature representation in an unsupervised manner, and this encoded representation is then utilised to train classifier with labels in a supervised manner. The majority of the earlier research focused on camera sensor-based static damage detection. Conventionally, inertial sensors are used in air bag control system to detect significant damages. However, they are unable to identify minor damages. We collected and used real field data with custom hardware solution as opposed to other research that work on synthetic or smartphone based data. As a result, our outcome is closer to a the real-world closer. We show how to anticipate damage occurrences using feature-encoded methods, and we assess our performance using well-known metrics like accuracy and F1 score. Gradient Boosting (GB) and Random Forest (RF) without encoded representations were used to compare our results, and we discovered that our model outperforms them both. As a result, various research directions can be started using this baseline results.

6 OUTLOOK

As a further extension of this work, first, we can include gyroscopic data. It may boost the performance of detection by reducing false positives and false negatives. Second, the dataset is imbalanced. And we can use techniques of class weights and over-sampling for improving results. Third, the current algorithm detects and classifies into damages and non-damages. It would be more helpful if we can further categorize into damage type and damage severity. Fourth, the dataset is highly asymmetric as undamaged samples outperform damaged samples which makes training the model quite difficult. Thus, we would like to explore advanced deep-learning models. Fifth, it will be helpful to compare the results of the autoencoder present in this study to traditional feature engineering methods. Lastly, the current framework includes inertial sensor information. But from different damage studies present in section 2, audio has helped in improving damage detection. Thus, we would consider using audio information as another modality, which may in turn help in improving the prediction performance.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Choi, J., Kong, C., Kim, G., and Lim, S. (2021). Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. *Expert Systems with Applications*, 183:115400.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Dwivedi, M., Malik, H. S., Omkar, S. N., Monis, E. B., Khanna, B., Samal, S. R., Tiwari, A., and Rathi, A. M. (2020). Deep learning-based car damage classification and detection.

Ess, A., Leibe, B., and Van Gool, L. (2010). Object detection and tracking for autonomous navigation in dynamic environments. *I. J. Robotic Res.*, 29:1707–1725.

French, M. and Handy, R. (2007). Spectrograms: Turning signals into pictures. *Journal of Engineering Technology*, 24:32–35.

Gontscharov, S., Baumgaertel, K., Kneifel, A., and Krieger, K.-L. (2014). Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing. *Procedia Technology*, 15.

Hashimoto, W., Hirota, M., Araki, T., Yamamoto, Y., Egi, M., Hirate, M., Maura, M., and Ishikawa, H. (2019). Detection of car abnormal vibration using machine learning. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 40–407.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Kyu, P. M. and Woraratpanya, K. (2020). Car damage detection and classification. In *Proceedings of the 11th International Conference on Advances in Information Technology*, IAIT2020, New York, NY, USA. Association for Computing Machinery.

Li, L., Ono, K., and Ngan, C.-K. (2021). A Deep Learning and Transfer Learning Approach for Vehicle Damage Detection. *The International FLAIRS Conference Proceedings*, 34(1).

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768.

Liu, S., Zhang, C., and Ma, J. (2016). Stacked autoencoders for feature extraction with neural networks. pages 377–384.

Mateo, C. and Talavera, J. (2017). Short-time fourier transform with the window size fixed in the frequency domain. *Digital Signal Processing*, 77.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *ArXiv e-prints*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Punetha, D., Kumar, D., and Mehta, V. (2012). Design and realization of the accelerometer based transportation system (ats). *International Journal of Computer Applications*, 49:17–20.

Sammarco, M. and Detyniecki, M. (2018). Crashzam: Sound-based car crash detection.

Sammarco, M. and Detyniecki, M. (2019). *Car Accident Detection and Reconstruction Through Sound Analysis with Crashzam*, pages 159–180.

Shi, G., Chan, C.-S., Zhang, G., Li, W., Leong, P., and Leung, K.-S. (2008). Towards a mobile airbag system using mems sensors and embedded intelligence. pages 634–639.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Singh, R., Ayyar, M. P., Sri Pavan, T. V., Gosain, S., and Shah, R. R. (2019). Automating car insurance claims using deep learning techniques. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 199–207.

van der Maaten, L. and Hinton, G. (2008). Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Vejdannik, M., Sadr, A., Albuquerque, V., and Tavares, J. (2018). *Signal Processing for NDE*, pages 1–19.

White, J., Thompson, C., Turner, H., Dougherty, B., and Schmidt, D. C. (2011). Wreckwatch: Automatic traffic accident detection and notification with smartphones. *Mob. Netw. Appl.*, 16(3):285–303.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*.

Yunhui, S. and Qiuqi, R. (2004). Continuous wavelet transforms. volume 1, pages 207– 210 vol.1.

Zaldivar, J., Calafate, C. M. T., Cano, J.-C., and Manzoni, P. (2011). Providing accident detection in vehicular networks through obd-ii devices and android-based smartphones. *2011 IEEE 36th Conference on Local Computer Networks*, pages 813–819.