Cross-dataset performance evaluation of deep learning distracted driver detection algorithms

Frank Zandamela[1*], *Terence* Ratshidaho[1], *Fred* Nicolls[2], and *Gene* Stoltz[1]

[1] Council for Scientific and Industrial Research, Optronic Sensor Systems, South Africa
[2]University of Cape Town, Department of Electrical Engineering, South Africa

Abstract. Deep learning has gained traction due its supremacy in terms of accuracy and ability to automatically learn features from input data. However, deep learning algorithms can sometimes be flawed due to many factors such as training dataset, parameters, and choice of algorithms. Few studies have evaluated the robustness of deep learning distracted driver detection algorithms. The studies evaluate the algorithms on a single dataset and do not consider cross-dataset performance. A problem arises because cross-dataset performance often implies model generalisation ability. Deploying a model in the real world without knowing its cross-dataset performance could lead to catastrophic events. The paper investigates the cross-dataset performance of deep learning distracted driver detection algorithms. Experimental results found reveal that deep learning distracted driver detection algorithms do not generalise well on unknown datasets for CNN models that use the whole image for prediction. The cross-dataset performance evaluations shed light on future research in developing robust deep learning distracted driver detection algorithms.

1 Introduction

The success of deep learning in other real-world applications such as number plate recognition for vehicle access control, has inspired the development of deep learning-based approaches to remedy the problem of distracted driver detection [1]. This move is done to reduce the number of distracted driver-related road accidents. The approaches proposed in the literature use many different techniques such as ensemble of convolutional neural networks (CNNs), combining CNN features and HOG features, and a hybrid of CNNs and recurrent neural networks (RNNs) [1]. In addition, different datasets are used for training and testing these approaches.

With continuing advancements in deep learning, it is imperative to evaluate the performance of proposed distracted driver detection algorithms. Such evaluations not only do they help in generating reference work that can be used for the selection of distracted driver detection algorithms but may provide important insights on the usage of distracted driver detection techniques, evaluation metrics and datasets used for distracted driver detection. Such lower-level type of insights might not be obtained from the original publications of the algorithms.

* Corresponding author: fzandamela@csir.co.za

Currently, in the literature, there is a lack of a comprehensive study that evaluates the cross-dataset performance of distracted driver detection algorithms [2, 3]. Most approaches are published with comparative performance results. Such evaluations are not only incomprehensive but also do not consider cross-dataset performance. Cross-dataset performance is important since it generally indicates the robustness and generalising ability of a learning model. The generalising ability of a model gives a good indication of the model's likelihood to fail when deployed in a real-world system. This study seeks to answer one critical question: to what extent can deep learning distracted driver detection algorithms generalise on image datasets they were not trained on? This is addressed by evaluating the performance of state-of-the-art deep learning-based distracted driver detection algorithms on widely used benchmark datasets. Most importantly, an in-depth evaluation and analysis of the cross-dataset performance of the algorithms is carried out.

The primary contributions of this work can be summarised as follows:

i. First comprehensive study that evaluates the cross-dataset performance of deep learning-based distracted driver detection algorithms.

ii. Experimental results on widely used distracted driver detection image datasets are provided. By so doing, the issue of dataset bias is addressed, and the cross-dataset performance of the algorithms is analysed. Class activation maps are used to further analyse any performance differences.

iii. The work may serve as reference work that can be used to guide the selection of distracted driver detection algorithms for different applications. Additionally, the article can generate research leads that can be pursued by other researchers.

2 Related work

Datasets. Datasets play a vital role in the successful application of deep learning on real world problems. This is because deep learning algorithms establish patterns based on features learned from the training dataset. Such is also the case in the task of distracted driver detection. The first dataset in the area of driving behaviour analysis and distracted driving was introduced by Zhao *et al.* [4, 5]. The dataset has side view images of the driver performing four driver activities: (i) grasping the steering wheel; (ii) operating the shift lever; (iii) eating a cake; and (iv) talking on a cellular phone. However, the dataset is not publicly available and all the papers ([6–8]) that benchmarked using the dataset are affiliated with either Southeast University, Xi'an Jiaotong-Liverpool University, or Liverpool University, and they have at least one shared author [9]. A total of 20 participants, 10 male and ten female, were involved in the development of the dataset. Later, the State Farm Insurance Company released a dataset in a quest to find out if computer vision can spot distracted drivers. The insurance company held a competition named State Farm Distracted Driver Detection [10] on Kaggle. The State Farm dataset consists of 2D dashboard camera images, showing 10 different driving postures.

Despite the State Farm dataset being public, the State Farm dataset was only limited to the purpose of the State Farm Distracted Driver Detection competition. Due to the lack of a quality dataset, Billah *et al.* [11] created a four-class distracted driver dataset called EEE BUET Distracted Driving [12]. The dataset was created using a Sony Cyber Shot 14.1 mega pixels camera that was affixed on the front windscreen facing the driver inside the vehicles. The four distracted driving activities in the dataset include talking on cell phone, texting on cell phone, eating, and operating cabinet equipment. A total of 13 participants took part in the development of the dataset.

Inspired by the State Farm dataset, Eraqi *et al.* created a similar dataset called AUC Distracted Driver Dataset [9, 13]. The dataset was made public subject to signing an agreement form. A two-phase data collection method was followed – in the first phase, the ASUS ZenFone smartphone (Model ZD551KL) rear camera was used, and the DS325 Sony DepthSense camera was used in the second phase. In the project, 44 drivers from 7 different counties were involved, of which 29 were males and 15 were females. However, it has been reported that the AUC dataset is not balanced, for example, the reaching behind class is only represents 7% of the complete data points [14]. In contrast, the normal driving class represents 21% of the complete dataset. In addition, not all drivers participated in all distraction activities. To remedy the shortcomings of the AUC dataset, Ezzouhri et al. [14] introduced a more balanced distracted driver detection dataset with 9 participants.

Evaluation methods. Most distracted driver detection algorithms are published with comparative evaluations. For example, Yan *et al.* [8] proposed a CNN-based approach that recognises driving posture based on the position of the hand and evaluated the proposed approach on three datasets. While other authors ([1], [15–17]) compare the performance of the proposed method to other approaches. However, the focus of these papers is on the proposed algorithms and the evaluations are not comprehensive. Recently, Ezzouhri *et al.* [14] evaluated their proposed driver body parts segmentation-based distracted driver detection algorithm on their custom dataset and a widely used benchmark dataset (AUC Distracted Driver Dataset [9, 13]). The main contribution of the authors was on the proposed algorithm and the created dataset. The cross-dataset performance evaluations were based on the AUC dataset only and a few CNN-based algorithms.

Recently, Kashevnik *et al.* [18] presented an extensive literature survey on distracted driver detection and outlined the entire chain of distracted driver detection from sensor data acquisition to data pre-processing, behaviour inference, and distraction type inference. Similarly, Huang *et al.* [19] provided an extensive literature survey on vision-based distracted driver detection algorithms. Despite these studies being compressive and providing current state of the knowledge on distracted driver detection, none of them evaluate and analyse the performance of distracted driver detection algorithms. In another study, the authors [2] presented a literature review on distracted driver detection algorithms and then proceeded to evaluate the performance of ten deep learning-based algorithms using a dataset called AUC Distracted Driver Dataset [9, 13].

3 Experimental setup

3.1 Algorithms

In this study, a total of six state-of-the-art algorithms with publicly available code or where authors provided code upon request were evaluated. In the event where code is not available, the authors implemented similar algorithms based on the original publications. State-of-the-art representative algorithms were selected based on performance results reported by other researchers [2, 3]. In addition, representative commonly used and recent algorithms were selected. The selected deep learning distracted driver detection algorithms can be broadly grouped into the following approaches: transfer learning, CNNs combined with other features or pre-processing stage, hybrid of CNNs with sequence models, and human pose estimation-based algorithms. Table 1 shows the complete list of algorithms that were evaluated with the corresponding approach used.

Table 1. List of algorithms evaluated.

Algorithm	Approach
ResNet50 [20]	Transfer learning
EfficientNetB0 [21]	Transfer learning
Leekha_GrabCut [22]	Background removal + CNN
ConvLSTM [23]	Convolutional LSTM layers
CNN LSTM [1]	Combination of Convolutional and LSTM layers in
CNN-Pose estimation [24]	Combines a CNN predictions and predictions of a random forest algorithm trained on detected human key points

3.2 Datasets

The primary objective of this study is to evaluate the cross-dataset performance of deep learning distracted driver detection algorithms. To achieve this objective, three distracted driver detection image datasets will be used. The datasets include AUC2 dataset, driver distraction dataset introduced by Ezzouhri *et al.* [14] (EZZ2021), and the State Farm dataset. The AUC2 and State Farm datasets were selected based on their wide usage in benchmarking distracted driver detection algorithms. The datasets are relatively large and considers 9 distracted activities. The EZZ2021 dataset was recently introduced and is similar to the AUC2 and State Farm datasets with 9 distracted driver classes and a save driving class. Fig. 1 shows sample images from the EZZ2021 dataset. The different classes and driver postures in the three datasets are shown in Table 2.

Table 2. Classes in the EZZ2021, AUC2, STF datasets.

Class	Driver action
C0	Safe driving
C1	Text right
C2	Talk right
C3	Text left
C4	Talk left
C5	Adjust radio
C6	Drinking
C7	Reach behind
C8	Make-up
C9	Talking to passenger

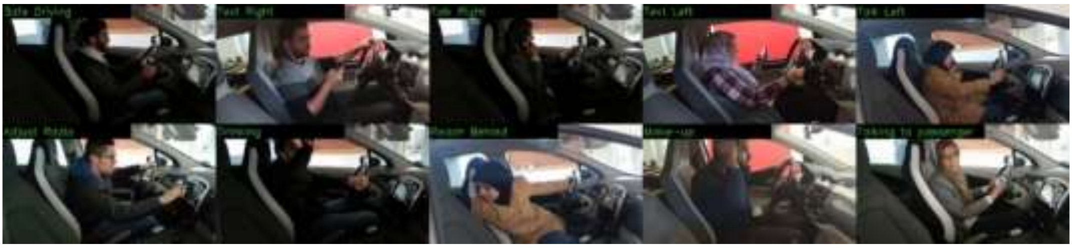


Fig. 1: Images randomly sampled from the EZZ2021 dataset.

Table 3 shows the distracted driver detection image datasets that will be used in the study with corresponding environment were the datasets were created (real or synthetic), type of distractions, number of drivers, and size of the datasets.

Table 3. Image datasets used. *=not mentioned.

Image Dataset	Environment	Type of distractions	Participants	Image samples
AUC2	Real	1 save driving, 9 distracted activities	44	32.7k
EZZ2021	Real	1 save driving, 9 distracted activities	9	29.2k
State Farm (STF)	Real	1 save driving, 9 distracted activities	*	22.4k

3.3 Evaluation metrics

To evaluate the performance of an algorithm in detecting a distracted driver, the performance of an algorithm will be determined by comparing the classification accuracy. Accuracy is the simplest and commonly used indication of the performance of a machine algorithm. Accuracy gives the number of correct predications a model has made over the total number of observations in the test set. In addition, to compare the performance of the algorithms per class, the weighted harmonic mean of the precision and recall performance metrics, i.e., the F-measure (F1-score), will be used. For further analysis, class activation maps (CAMs) will be used. CAMs help in understanding what a CNN "see" and how it arrived at the final prediction. Specifically, an approach called Grad-CAM [25] will be used. Grad-CAM works by finding the final convolutional layer in the network and then examining the gradient information flowing into that layer. The output of Grad-CAM is a heatmap visualization for a given class label (either the top, predicted label or an arbitrary label we select for debugging). We can use this heatmap to visually verify where in the image the CNN is looking.

4 Evaluation and analysis

4.1 Training and evaluation

4.1.1 Evaluation procedure

Each distracted driver detection image dataset was split into three sets, i.e., training, validation, and testing. Training sets were used for training and validation sets were used for hyperparameter tuning. While the test set was used for cross-dataset performance evaluation. Each algorithm was trained separately on each dataset and tested against all three datasets.

4.1.2 Training procedure

Transfer learning approaches. ResNet50 and EfficientNetB0 architectures pre-trained on ImageNet were fine-tuned to each of the three datasets using transfer learning framework. The top layers (head) were replaced by a GlobalAveragePooling2D layer, followed a by a

Dropout layer and a fully connected layer with 10 neurons. Table 4 shows that hyperparameters used for training.

Leekha_GrabCut. For the Leekha_GrabCut algorithm, an EfficientNetB0 model pre-trained on ImageNet was fine-tuned to the three image datasets. However, the GrabCut background removal algorithm was incorporated as a pre-processing stage to the data pipeline used for training the Leekha_GrabCut algorithm.

convLSTM. A convLSTM model with 4 ConvLSTM2D recurrent layers was used. Each ConvLSTM2D recurrent layer was followed by a Maxpooling3D layer and a Dropout layer. The Maxpooling3D layer reduces dimensions of the frames and avoid unnecessary computations. Dropout layers help prevent overfitting the model on the data.

CNN LSTM. The CNN LSTM model was built using the AlexNet architecture and an LSTM layer with 50 units. A fully connected layer with 10 neurons and a softmax activation function was used for class prediction. For both convLSTM and CNN LSTM models, the datasets were prepared to be sequence data with five images.

CNN-Pose. The CNN-Pose algorithm consists of a fine-tuned EfficientNetB0 architecture using transfer learning and a Random Forest machine learning model trained on detected human key points obtained through pose estimation. The final prediction was a combination of predictions from the CNN and Random Forest models multiplied by two different coefficients that add up to one. For this study, the coefficients were obtained using a grid search for each dataset. Table 5 shows the coefficients that were obtained for the CNN-Pose algorithm in the three datasets.

The models were implemented using Python 3.6, scikit-learn, NumPy, OpenCV-Python, PyTorch, and TensorFlow. The training and validation information of all algorithms is shown in Table 6.

Table 4. Hyperparameters used for training the algorithms.

Algorithm	Learning rate	Optimizer	Dropout	Filters
ResNet50	Head: 0.001, 15 epochs Fine-tuning: 1e-5	Adam Adam	0.2 -	- -
EfficientNetB0	Head: 0.001, 15 epochs Fine-tuning: 1e-5	Adam Adam	0.2 -	- -
Leekha_GrabCut	Head: 0.001, 15 epochs Fine-tuning: 1e-5	Adam Adam	0.2 -	- -
ConvLSTM	0.001	Adam	0.2	4-8-14-16
CNN LSTM	0.001	Adam	0.25	CNN: 96-256-384-384-256 LSTM: 50
CNN-Pose estimation	CNN: 0.001	Adam	0.2	-

Table 5. Coefficients obtained for the CNN-Pose estimation algorithm

Dataset	Coefficients (cnn, pose)
EZZ2021	(0.3, 0.7)
AUC2	(0.4, 0.6)
STF	(0.3, 0.7)

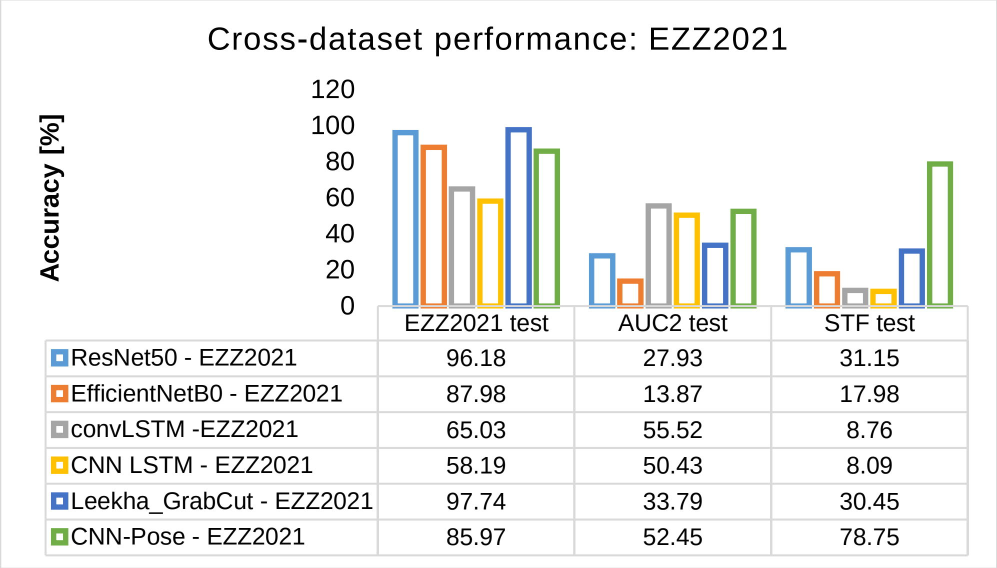
Table 6. Training and development (validation) information.

Algorithm	**EZZ2021**		**AUC2**		**STF**	
	Train acc	Dev acc	Train acc	Dev acc	Train acc	Dev acc
ResNet50	100	90	99.42	97.30	99.75	99.70
EfficientNetB0	99.99	99.80	99.64	97.8	99.91	99.80
convLSTM	100	100	97.52	92	99.08	99
CNN LSTM	91.90	92.50	80.06	70	92.65	93
Leekha_GrabCut	99.90	99.27	97.78	44.92	96.47	90.10
CNN-Pose estimation	-	95.83	-	94	-	93.64

4.2 Cross-dataset performance evaluation and analysis

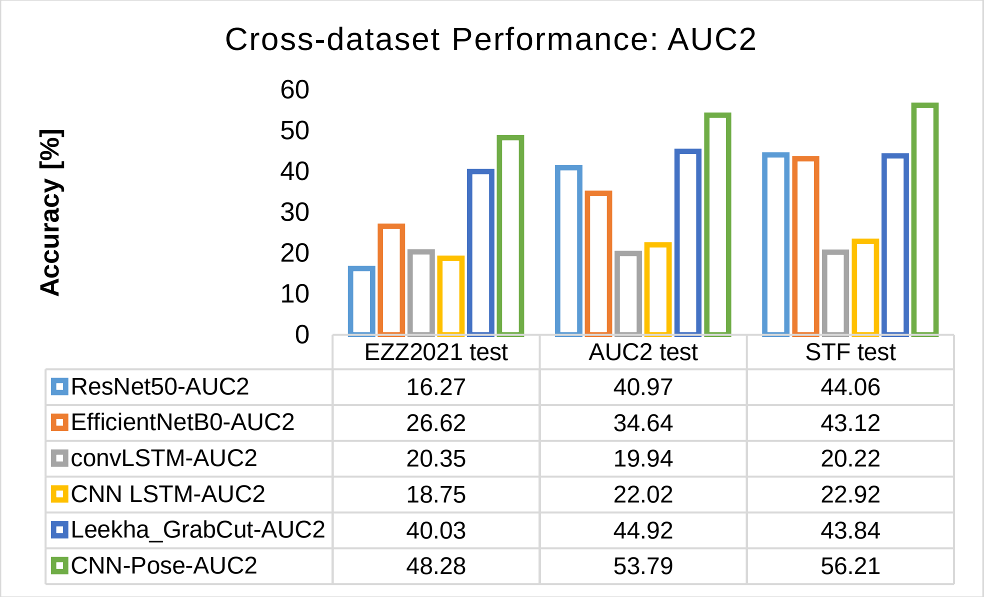
Fig. 2 through Fig. 4 show bar graphs comparing the classification accuracy of the 6 evaluated algorithms on the 3 distracted driver test datasets. Based on the bar graphs, the following observations can be made:

- All algorithms trained on the EZZ2021 training dataset perform very well on the EZZ2021 test set except for convLSTM and CNN LSTM algorithms which have average accuracies.
- Algorithms trained on the EZZ2021 training set did not perform well on both the AUC2 and STF test sets.
- The CNN-Pose algorithm trained on the EZZ2021 training dataset has a better performance (AUC2: 52.45%, STF: 78.75%) compared to the other five algorithms evaluated.
- Algorithms trained on the AUC2 training dataset did not perform well on the AUC2 test dataset. The algorithms also have low accuracies on both EZZ2021 and STF test sets.
- Algorithms trained on the STF training dataset perform very well on the STF test set, except for the CNN-Pose algorithm with a 73.91% classification accuracy on the STF test set.
- Algorithms trained on the STF training set did not perform well on both the AUC2 and EZZ2021 test sets.
- Compared to all other 5 algorithms evaluated, the CNN-Pose algorithm has a better performance across all thee test image datasets irrespective of the training dataset used. The Leekha_GrabCut algorithm has the second-best performance across the three datasets.



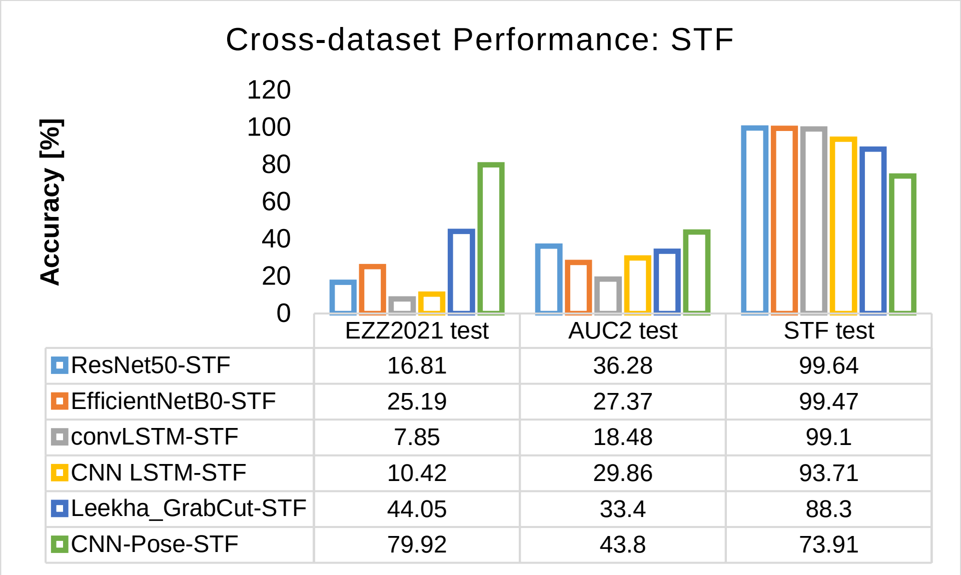
	EZZ2021 test	AUC2 test	STF test
🟦 ResNet50 - EZZ2021	96.18	27.93	31.15
🟧 EfficientNetB0 - EZZ2021	87.98	13.87	17.98
⬜ convLSTM -EZZ2021	65.03	55.52	8.76
🟨 CNN LSTM - EZZ2021	58.19	50.43	8.09
🟦 Leekha_GrabCut - EZZ2021	97.74	33.79	30.45
🟩 CNN-Pose - EZZ2021	85.97	52.45	78.75

Fig. 2: Cross-dataset performance for algorithms trained on the EZZ2021 dataset.



	EZZ2021 test	AUC2 test	STF test
🟦 ResNet50-AUC2	16.27	40.97	44.06
🟧 EfficientNetB0-AUC2	26.62	34.64	43.12
⬜ convLSTM-AUC2	20.35	19.94	20.22
🟨 CNN LSTM-AUC2	18.75	22.02	22.92
🟦 Leekha_GrabCut-AUC2	40.03	44.92	43.84
🟩 CNN-Pose-AUC2	48.28	53.79	56.21

Fig. 3: Cross-dataset performance of the algorithms trained on the AUC2 dataset.



	EZZ2021 test	AUC2 test	STF test
ResNet50-STF	16.81	36.28	99.64
EfficientNetB0-STF	25.19	27.37	99.47
convLSTM-STF	7.85	18.48	99.1
CNN LSTM-STF	10.42	29.86	93.71
Leekha_GrabCut-STF	44.05	33.4	88.3
CNN-Pose-STF	79.92	43.8	73.91

Fig. 4: Cross-dataset performance of algorithms trained on the STF dataset.

For further analysis, the F1-score was used to compare the performance of the algorithms on the safe driving class. Table 7 through Table 12 show the results of the algorithms when trained and tested on each of the three datasets. It can be observed that all algorithms perform well in detecting a driver in a safe driving posture when tested on a testing dataset that comes from the same dataset as the training dataset used. In contrast, the algorithms do not do well in detecting a driver in a safe driving posture when tested on a testing dataset that does not come from the same dataset as the training dataset. These observations correspond to the observations made above based on the classification accuracy of the models. This was expected since training and testing datasets from same dataset generally have the same characteristics such as same camera viewpoint, drivers used, and cars used. The results also reveal that algorithms trained on the AUC2 dataset do perform well on all perform across the three testing datasets. In addition, all algorithms perform well when tested on the EZZ2021 and STF test datasets compared to when tested on the AUC2 test dataset.

Based on Table 7 through Table 12, it can be observed that the CNN-Pose algorithm has the best overall performance across all three test datasets. The Leekha_GrabCut algorithm has the second-best performance. While the convLSTM and CNN LSTM algorithms have the worst performance on the three testing datasets.

The detailed per-class and overall performance of all algorithms can be found in Appendix A of this paper.

Table 7. Performance of the ResNet50 model on the safe driving class.

Safe driving class					
ResNet50		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.85	0.52	0.31	***0.56***
	AUC2	0.19	0.61	0.39	***0.40***
	STF	0.00	0.50	0.95	***0.48***
	Average F1-score	***0.34***	***0.54***	***0.48***	

Table 8. Performance of the EfficientNetB0 model on the safe driving class.

Safe driving class					
EfficientNetB0		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.85	0.27	0.19	***0.44***
	AUC2	0.54	0.35	0.39	***0.43***
	STF	0.44	0.24	0.88	***0.52***
	Average F1-score	***0.61***	***0.29***	***0.49***	

Table 9. Performance of the convLSTM model on the safe driving class.

Safe driving class					
convLSTM		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.97	0.00	0.00	***0.32***
	AUC2	0.45	0.04	0.11	***0.2***
	STF	0.00	0.23	0.97	***0.4***
	Average F1-score	***0.47***	***0.09***	***0.36***	

Table 10. Performance of the CNN LSTM model on the safe driving class.

Safe driving class					
CNN LSTM		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.62	0.04	0.00	***0.22***
	AUC2	0.13	0.39	0.22	***0.25***
	STF	0.15	0.09	0.87	***0.37***
	Average F1-score	***0.30***	***0.17***	***0.36***	

Table 11. Performance of the Leekha_GrabCut model on the safe driving class.

Safe driving class					
Leekha_GrabCut		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.94	0.46	0.27	***0.56***
	AUC2	0.39	0.58	0.43	***0.47***
	STF	0.29	0.27	0.88	***0.48***
	Average F1-score	***0.54***	***0.44***	***0.53***	

Table 12. Performance of the CNN-Pose model on the safe driving class.

Safe driving class					
CNN-Pose		**Test dataset**			
		EZZ2021	**ACU2**	**STF**	***Average F1-score***
Trained	**EZZ2021**	0.99	0.69	0.53	***0.74***
	AUC2	0.78	0.60	0.56	***0.65***
	STF	0.98	0.60	0.96	***0.85***
	Average F1-score	***0.92***	***0.63***	***0.68***	

To understand what features are used by the CNN models when making predictions, Grad-CAM was used. Fig. 5 shows a sample output of Grad-CAM when applied on the ResNet50 models. Due to space limitations, Grad-CAM outputs of all the algorithms were not included in the paper. However, based on the Grad-CAM analysis, the following observations were made:

- The models seem to be looking for the right features or regions of the image when making a prediction. This is especially true for test datasets which are from the same dataset as the training dataset.
- Although the models learn important features, they also learn features that are not important. This especially the case when the whole image is used for training. For example, for the make-up class, the models look for hands on the head, face or an opened front mirror. This causes model confusion on images where the car has the front mirror opened since the models take shortcuts.
- The models look for the position of two hands (specifically, the forearms) in relation to the steering wheel when predicting the safe driving driver posture. The models get confused when they only see one forearm. Some images were taken too close to the driver and as a result the two arms are not clearly visible. As a result, models seem to be struggling when the driver is closer to the camera.
- The presence of a cell phone around the driver confuses the model to predict classes that involve the presence of a cell phone.

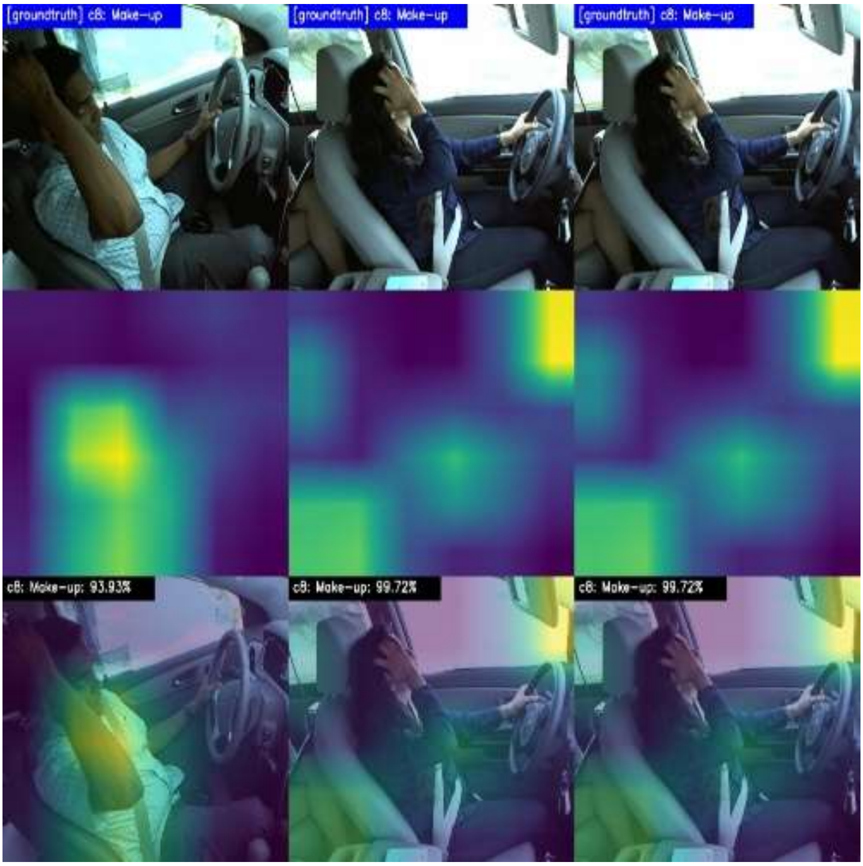

[groundtruth] c8: Make-up
[groundtruth] c8: Make-up
[groundtruth] c8: Make-up
c8: Make-up: 93.93%
c8: Make-up: 99.72%
c8: Make-up: 99.72%

Fig. 5: Sample output of Grad-CAM when applied on the ResNet50 model.

The results and analysis above suggest that:

- The CNN-Pose model has better generalising ability compared to the other algorithms due to a better cross-dataset performance. This can be attributed to the fact the model takes advantage of both rich features learnt by the CNN and human key points which are less variable.
- The second-best model is the Leekha_GrabCut. The GrabCut algorithm removes background noise, forcing a model to focus on the body posture of the driver during training. This could explain why the Leekha_GrabCut algorithm can obtain reasonable cross-dataset performance, compared to the other algorithms, since it reduces dataset-to-dataset variability by removing objects that are not important in detecting a distracted driver.
- The characteristics of the AUC2 dataset negatively affect the performance of algorithms trained on it. Based on the initial splits provided with the datasets, the major difference between the three datasets is that in the AUC2 dataset, drivers in the training set are not in the testing set. In contrast, in the EZZ2021 and STF datasets, drivers in the training datasets are also in the testing datasets. In the AUC2 dataset, drivers do not participate in all driver posture activities. These differences could explain why models trained on the AUC2 training dataset do not perform well on the AUC2 test dataset. This could also explain why models trained EZZ2021 and

STF training datasets perform well on their respective testing datasets. The authors also attribute the poor performance of algorithms trained on the AUC2 dataset to the fact that the dataset is relatively large but not diverse. Each driver in the dataset performs the same driver activity for more than 20 times with very little differences between the image frames. This creates an opportunity for shortcut learning which can easily arise due to a systematic relationship between the driver and background or context [26].

- CNN models that use the whole image without background noise removal or without considering other features that are less variable, do not generalise well on new data. This can be attributed to fact that the three datasets are large but not diverse.
- In general, all distracted driver detection algorithms do not perform exceptionally well when tested on image datasets that they were not trained on. This is especially the case for CNN models that use the whole image without background noise removal or using features that are less variable. The authors attribute this overall poor cross-dataset performance to the datasets used for training. The datasets are relatively large but not diverse, i.e., lack of high data variance. As a result, deep learning distracted driver detection algorithms resort to shortcut learning which significantly reduces their ability to generalise to new data.

5 Conclusions and future work

This work sought to find the extent to which deep learning distracted driver detection algorithms can generalise to new data that was not used for training. A cross-dataset performance evaluation study was carried out. Based on the analysis in section 4, it was found that, in general, deep learning distracted driver detection algorithms do not perform very well on testing datasets that do not come from the same dataset as the training dataset. Based on the findings of the study, the authors suggest that future work should:

- Create large and diverse distracted driver detection image datasets. To reduce, the effort required, synthetic image data generation using AI (for example, generative adversarial networks (GANs)) and CGI can be explored.
- Work towards creating features that are less variable from dataset-to-dataset. In the CNN-Pose model, the pose estimation model was given more weight than the CNN. This may suggest that using features derived from detected human key points (pose estimation) can result to a model with better cross-dataset performance.

References

1. J. M. Mase, P. Chapman, G. P. Figueredo, and M. Torres Torres, *A Hybrid Deep Learning Approach for Driver Distraction Detection*, in International Conference on Information and Communication Technology Convergence, ICTC, pp. 1-6 (2020)
2. J. M. Mase, P. Chapman, G. P. Figueredo, and M. Torres Torres, *Benchmarking Deep Learning Models for Driver Distraction Detection*, In International Conference on Machine Learning, Optimization, and Data Science, pp. 103-117 (2020)
3. M. H. Saad, M. I. Khalil, and H. M. Abbas, *End-To-End Driver Distraction*

Recognition Using Novel Low Lighting Support Dataset, in 2020 15th International Conference on Computer Engineering and Systems, ICCES, pp. 1-6 (2020)

4. C. H. Zhao, B. L. Zhang, J. He, and J. Lian, *Recognition of driving postures by contourlet transform and random forests*, IET Intell. Transp. Syst., **6**, pp. 161–168 (2012)

5. C. Zhao, B. Zhang, J. Lian, J. He, T. Lin, and X. Zhang, *Classification of driving postures by support vector machines*, in 2011 sixth International Conference on Image and Graphics, pp. 926-930 (2011)

6. C. Zhao, Y. Gao, J. He, and J. Lian, *Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier*, Engineering Applications of Artificial Intelligence, **25**, pp. 1677–1686 (2012)

7. C. H. Zhao, B. L. Zhang, X. Z. Zhang, S. Q. Zhao, and H. X. Li, Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers, Neural Computing and Applications, **22**, pp. 175–184 (2013)

8. C. Yan, F. Coenen, and B. Zhang, *Driving posture recognition by convolutional neural networks*, IET Computer Vision, **10**, pp. 103–114 (2016)

9. H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, *Driver distraction identification with an ensemble of convolutional neural networks*, Journal of Advanced Transportation (2019)

10. "State Farm Distracted Driver Detection | Kaggle." https://www.kaggle.com/c/state-farm-distracted-driver-detection (accessed Mar. 28, 2022).

11. T. Billah, S. M. M. Rahman, M. O. Ahmad, and M. N. S. Swamy, *Recognizing distractions for assistive driving by tracking body parts*, IEEE Transactions on Circuits and Systems for Video Technology, **29**, pp. 1048–1062 (2019)

12. S. M. M. Rahman, *EEE BUET Distracted Driving (EBDD) Video Database*, https://mahbubur.buet.ac.bd/resources/ebdd_database.htm (accessed Mar. 28, 2022).

13. H. M. Eraqi, *Distracted Driver Dataset*, https://heshameraqi.github.io/distraction_detection (accessed Mar. 28, 2022).

14. A. Ezzouhri, Z. Charouh, M. Ghogho, and Z. Guennoun, *Robust Deep Learning-Based Driver Distraction Detection and Classification*, IEEE Access, **9**, pp. 168080–168092 (2021)

15. J. Wang, Z. C. Wu, F. Li, and J. Zhang, *A data augmentation approach to distracted driving detection*, Future Internet, **13**, pp. 1–11 (2021)

16. M. R. Arefin, F. Makhmudkhujaev, O. Chae, and J. Kim, *Aggregating CNN and HOG features for Real-Time Distracted Driver Detection*, In 2019 IEEE International Conference on Consumer Electronics, ICCE 2019, pp. 12–14, (2019)

17. B. Baheti, S. Gajre, and S. Talbar, *Detection of distracted driver using convolutional neural network*, In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 1145–1151 (2018)

18. A. Kashevnik, R. Shchedrin, C. Kaiser, and A. Stocker, *Driver Distraction Detection Methods: A Literature Review and Framework*, IEEE Access, **9**, pp. 60063–60076 (2021)

19. W. Li, J. Huang, G. Xie, F. Karray, and R. Li, *A survey on vision-based driver*

distraction analysis, Journal of Systems Architecture, **121**, p. 102319 (2021)

20. K. He and J. Sun, *Deep Residual Learning for Image Recognition*, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
21. M. Tan and Q. V Le, *EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks*, In International conference on machine learning, PMLR, pp. 6105-6114 (2019)
22. M. Leekha, M. Goswami, R. R. Shah, Y. Yin, and R. Zimmermann, *Are you paying attention? Detecting distracted driving in real-time*, In 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), pp. 171–180 (2019)
23. X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, *Convolutional LSTM network: A machine learning approach for precipitation nowcasting,* Advances in neural information processing systems, **28**, pp. 802–810 (2015)
24. M. Cetinkaya and T. Acarman, *Driver activity recognition using deep learning and human pose estimation*, In 2021 International Conference on INnovations in Intelligent SysTems and Applications, INSTA, pp. 1–5 (2021)
25. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization*, In Proceedings of the IEEE international conference on computer vision, **128**, pp. 336–359 (2020)
26. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M. and Wichmann, F.A, *Shortcut learning in deep neural networks*, Nature Machine Intelligence, **2**, pp. 665–673 (2020)

Appendix A
Per-class performance results of the algorithms

Table 13. The results of the ResNet50 model on the three datasets.

Class	**ResNet50-EZZ2021**			**ResNet50-AUC2**			**ResNet50-STF**		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.85**	**0.52**	**0.31**	**0.19**	**0.61**	**0.39**	**0.00**	**0.50**	**0.95**
Text right	0.99	0.07	0.43	0.00	0.39	0.35	0.04	0.34	0.96
Talk right	0.98	0.51	0.48	0.01	0.23	0.42	0.00	0.00	0.92
Text left	1.00	0.00	0.26	0.00	0.32	0.54	0.28	0.00	0.99
Talk left	0.92	0.00	0.29	0.05	0.53	0.56	0.01	0.58	0.97
Adjust radio	0.93	0.00	0.19	0.05	0.68	0.64	0.00	0.86	0.94
Drinking	0.99	0.00	0.00	0.01	0.23	0.06	0.22	0.00	0.93
Reach behind	1.00	0.19	0.32	0.33	0.31	0.57	0.25	0.42	0.98
Make-up	0.99	0.10	0.11	0.16	0.40	0.33	0.15	0.22	0.88
Talking to passenger	0.97	0.44	0.39	0.00	0.07	0.45	0.01	0.32	0.79
Overall accuracy	***96.18***	***27.93***	***31.15***	***16.27***	***40.97***	***44.06***	***16.81***	***36.28***	***99.64***

Table 14. The results of the EfficientNetB0 model on the three datasets.

Class	**EfficientNetB0-EZZ2021**			**EfficientNetB0-AUC2**			**EfficientNetB0-STF**		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.85**	**0.27**	**0.19**	**0.54**	**0.35**	**0.39**	**0.44**	**0.24**	**0.88**
Text right	0.87	0.00	0.16	0.15	0.06	0.35	0.03	0.09	0.94
Talk right	0.86	0.00	0.31	0.04	0.55	0.42	0.00	0.00	0.87
Text left	0.93	0.00	0.08	0.26	0.15	0.54	0.30	0.29	0.95
Talk left	0.84	0.00	0.15	0.02	0.67	0.56	0.20	0.54	0.91
Adjust radio	0.76	0.04	0.17	0.04	0.52	0.64	0.00	0.06	0.96
Drinking	0.95	0.24	0.06	0.43	0.38	0.06	0.22	0.10	0.91
Reach behind	0.96	0.21	0.16	0.35	0.62	0.57	0.12	0.40	0.91
Make-up	0.81	0.09	0.03	0.24	0.17	0.33	0.11	0.09	0.83
Talking to passenger	0.95	0.10	0.33	0.38	0.00	0.45	0.32	0.08	0.87
Overall accuracy	***87.98***	***13.87***	***17.98***	***26.62***	***34.64***	***43.12***	***18.12***	***15.27***	***90.39***

Table 15. The results of the convLSTM model on the three datasets.

Class	convLSTM-EZZ2021			convLSTM-AUC2			convLSTM-STF		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.97**	**0.00**	**0.00**	**0.45**	**0.04**	**0.11**	**0.00**	**0.23**	**0.97**
Text right	0.99	0.00	0.09	0.18	0.10	0.04	0.16	0.21	1.00
Talk right	0.97	0.05	0.13	0.00	0.08	0.00	0.00	0.25	1.00
Text left	0.98	0.00	0.17	0.00	0.09	0.00	0.00	0.00	0.99
Talk left	0.99	0.00	0.03	0.00	0.44	0.20	0.00	0.00	1.00
Adjust radio	1.00	0.00	0.03	0.42	0.11	0.17	0.02	0.11	1.00
Drinking	0.99	0.00	0.09	0.18	0.14	0.10	0.05	0.00	0.99
Reach behind	1.00	0.02	0.08	0.19	0.39	0.32	0.08	0.28	1.00
Make-up	0.99	0.00	0.09	0.11	0.14	0.17	0.00	0.00	0.99
Talking to passenger	1.00	0.03	0.05	0.21	0.10	0.35	0.00	0.20	0.98
Overall accuracy	***98.91***	***1.90***	***8.76***	***20.43***	***17.61***	***20.22***	***7.85***	***18.57***	***99.10***

Table 16. The results of the CNN-LSTM model on the three datasets.

Class	CNN-LSTM-EZZ2021			CNN-LSTM-AUC2			CNN-LSTM-STF		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.62**	**0.04**	**0.00**	**0.13**	**0.39**	**0.22**	**0.15**	**0.09**	**0.87**
Text right	0.89	0.00	0.00	0.02	0.00	0.00	0.03	0.10	0.99
Talk right	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.95
Text left	0.88	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.85
Talk left	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96
Adjust radio	0.97	0.00	0.00	0.00	0.00	0.00	0.24	0.82	0.99
Drinking	0.94	0.00	0.00	0.00	0.00	0.32	0.00	0.07	0.98
Reach behind	0.94	0.13	0.15	0.00	0.00	0.28	0.11	0.00	0.94
Make-up	0.90	0.13	0.04	0.03	0.00	0.00	0.03	0.25	0.93
Talking to passenger	0.95	0.00	0.00	0.16	0.24	0.18	0.04	0.25	0.93
Overall accuracy	***88.23***	***7.14***	***8.76***	***8.34***	***23.81***	***13.26***	***10.42***	***30.00***	***94.00***

Table 17. The results of the Leekha_GrabCut model on the three datasets.

Class	**Leekha_GrabCut-EZZ2021**			**Leekha_GrabCut-AUC2**			**Leekha_GrabCut-STF**		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.94**	**0.46**	**0.27**	**0.39**	**0.58**	**0.43**	**0.29**	**0.27**	**0.88**
Text right	1.00	0.38	0.34	0.26	0.08	0.37	0.55	0.11	0.94
Talk right	0.95	0.38	0.35	0.35	0.12	0.39	0.30	0.07	0.87
Text left	0.98	0.50	0.31	0.54	0.70	0.62	0.52	0.48	0.95
Talk left	0.96	0.30	0.40	0.42	0.64	0.57	0.48	0.76	0.91
Adjust radio	0.99	0.28	0.24	0.56	0.50	0.39	0.38	0.35	0.96
Drinking	1.00	0.42	0.21	0.21	0.58	0.46	0.46	0.30	0.91
Reach behind	0.99	0.20	0.35	0.49	0.40	0.53	0.30	0.34	0.91
Make-up	0.98	0.09	0.15	0.19	0.25	0.30	0.18	0.20	0.83
Talking to passenger	0.98	0.25	0.32	0.61	0.28	0.42	0.61	0.38	0.87
Overall accuracy	***97.71***	***32.31***	***30.53***	***40.06***	***44.23***	***43.88***	***42.03***	***33.43***	***88.30***

Table 18. The results of the CNN-Pose model on the three datasets.

Class	**CNN-Pose-EZZ2021**			**CNN-Pose-AUC2**			**CNN-Pose–STF**		
	EZZ2021 test – F1	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***	***EZZ2021 test-F1***	***AUC2 test-F1***	***STF test-F1***
Safe driving	**0.99**	**0.69**	**0.53**	**0.78**	**0.60**	**0.56**	**0.98**	**0.60**	**0.96**
Text right	1.00	0.48	0.62	0.68	0.34	0.56	1.00	0.59	0.98
Talk right	0.99	0.15	0.58	0.55	0.69	0.58	0.98	0.05	0.98
Text left	0.99	0.62	0.48	0.64	0.61	0.45	0.97	0.82	1.00
Talk left	1.00	0.99	0.89	0.68	0.69	0.79	0.99	0.89	0.99
Adjust radio	1.00	0.43	0.39	0.56	0.66	0.63	0.99	0.52	0.99
Drinking	1.00	0.63	0.19	0.52	0.50	0.27	0.97	0.18	0.98
Reach behind	0.99	0.39	0.50	0.61	0.56	0.77	0.97	0.63	1.00
Make-up	0.99	0.30	0.38	0.36	0.23	0.36	0.96	0.22	0.95
Talking to passenger	0.99	0.53	0.56	0.77	0.48	0.68	0.96	0.64	0.94
Overall accuracy	***99.35***	***52.45***	***53.52***	***58.59***	***51.49***	***56.48***	***97.53***	***50.82***	***97.76***

Rebuttal

Comment/Correction	**Page**	**Reviewer**	**Reaction**
Indicate the importance of deep learning	Pg.1 – Abstract	2	Added two sentences: *Deep learning has gained traction due its supremacy in terms of accuracy and ability to automatically learn features from input data. However, deep learning algorithms can sometimes be flawed due to many factors such as training dataset, parameters, and choice of algorithms*
Remove “This is especially the case” and join the sentences	Pg.1 – Abstract	2	Removed “This is especially the case” and joined the two sentences
Rewrite opening paragraph in the introduction and add reference	Pg.1 – introduction	2	Rephrased opening paragraph and included reference [1]
Fix referencing	pg. 1 – 5	2	Changed reference citation from [ref1], [ref2] to **[ref1, ref2]**
Change figure labels	All	2	Changed figure labels from Figure x to Fig. x
Add full stop at the end of figure and table captions	All	2	Added a full stop (.) at the end of figure and table captions
Grammar errors	All	2	Accepted all changes recommended by the reviewer.