Waste Classification by Fine-Tuning Pre-trained CNN and GAN

Amani Alsabei, Ashwaq Alsayed, Manar Alzahrani, Sarah Al-Shareef,
{s44280161,s44280168,s44280279}@st.uqu.edu.sa, saashareef@uqu.edu.sa
Computer Science Department, Umm Al-Qura University, Makkah, Saudi Arabia

Summary
Waste accumulation is becoming a significant challenge in most urban areas and if it continues unchecked, is poised to have severe repercussions on our environment and health. The massive industrialisation in our cities has been followed by a commensurate waste creation that has become a bottleneck for even waste management systems. While recycling is a viable solution for waste management, it can be daunting to classify waste material for recycling accurately. In this study, transfer learning models were proposed to automatically classify wastes based on six materials (cardboard, glass, metal, paper, plastic, and trash). The tested pre-trained models were ResNet50, VGG16, InceptionV3, and Xception. Data augmentation was done using a Generative Adversarial Network (GAN) with various image generation percentages. It was found that models based on Xception and VGG16 were more robust. In contrast, models based on ResNet50 and InceptionV3 were sensitive to the added machine-generated images as the accuracy degrades significantly compared to training with no artificial data.

Keywords:
deep learning, image classification, convolutional neural networks, transfer learning, waste classification, recycling.

1. Introduction

Rapid urbanisation, population growth, and industrialisation all contribute to global environmental pollution. The intense use of natural resources has become unavoidable. In contrast, the waste products produced due to ever-increasing consumption trends have reached levels that endanger the environment and human health in quantity and harmful content. Consumption, manufacturing, and chemical and physical properties are all considerations that can be used to classify waste products. In terms of human and environmental health, waste products are commodities that must be disposed of quickly. In summary, failing to prioritise recycling will result in financial losses and the waste of natural resources.

Currently, recycling facilities sort waste items manually for large objects and use a sequence of large filters to segregate smaller ones. With the increase of computing power, there have been rapid advancements in computer vision and image processing. Deep Learning with convolutional neural networks (CNN) as the core plays a pivotal role in this regard. Employing deep learning to identify and classify waste items will increase recycling facilities' efficiency and reduce expenses in both time and human resources, positively impacting the environment.

This paper is organised as follows. First, previous work conducted in waste classification is reviewed (Section 2), followed by a description of the data used in this study (Section 3). Then, the experimental design of the proposed experiments is explained (Section 4), and the results of these experiments are presented and discussed (Section 5). Finally, the study is concluded (Section 6).

2. Related Work

There has been an increase in the volume of research on automatic waste classification recently. This increase could be because heavily populated countries, such as China, implemented a compulsory municipal solid waste classification. In the reviewed work, municipal solid waste can be classified into either four or six classes. In four-class systems, such as [5,6,7,8,9], the classes are dry garbage, wet garbage (kitchen waste), recyclables and hazardous waste. In the six-class systems, such as [1,2,4,10], the focus is more on recyclable waste. The six classes are glass, metal, paper, cardboard and other trash. Fewer researchers, as [6,7], defined sub-classes within these major ones.

Most researchers employed deep learning approaches by utilising pre-trained CNN either as feature extractors, as in [4], or fine-tuned with in-domain data, also known as transfer learning, as in [1,2,3,4,5,6,7]. Table 1 summarises the performance of these models and the amount of data used in fine-tuning. Fewer researchers had trained their deep models, as in [1,3,5,8,9,10].

Deep CNNs have performed remarkably well on many of these works. However, they are heavily reliant on big data to avoid overfitting. For that, researchers tried to acquire their in-domain data from the internet using web crawlers. They either depended only on these web-harvested data, as [5,7], or added them to their own captured data, as in [8].

This study shares a similar purpose with the reviewed work above. Waste items will be classified into six categories using the transfer learning approach. However,

Manuscript received August 5, 2021
Manuscript revised August 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.9

Table 1
Performance summary for the reviewed work using fine-tuned pre-trained CNN models. #Class indicates the number of classes to be recognized by the system; Data shoes the amount of data used for fine-tuning the models and (+) and (-) in DA column indicates whether data augmentation was used or not, respectively.

Ref	#Class	Data	DA	Pre-trained Models	Accuracy
[1]	6	2.5k	+	Densenet121/169	95.0
				Inception	94.0
				ResNetV2	89.0
				MobileNet	84.0
				Xception	80.0
[2]	6	2.5k	-	MobileNet	87.2
[3]	3	8.3k	+	ResNeXt101	94.0
[4]	6	2.5k	-	Alexnet	87.1
				VGG16	90.0
				GoogLeNet	88.1
				ResNet	89.4
[5]	4	17.5k	-	VGG16	37.6
				ResNet50	47.5
[6]	4	10.5k	+	DenseNet121	95.4
				DenseNet169	96.4
				ResNet50	93.2
				ResNet101	93.3
				ResNeXt50	95.9
				ResNeXt101	95.5
				EfficientNet-B3	96.5
				EfficientNet-B4	96.8
[7]	4	2k	+	InceptionV3	93.2

as most of the studies above employed data augmentation to increase in-domain data, this study will explore using the Generative Adversarial Networks (GAN) to generate more in-domain data.

3. Methodology

The task in this study is to classify waste items into six categories. To achieve this, pre-trained CNN models will be fine-tuned using in-domain data. Four pre-trained models will be explored here: ResNet50, InceptionV3, Xception and VGG16. However, the disparity in the number of samples within classes, also known as class imbalance, push a classifier to favour classes with a high population, which reduces the classifier's overall performance. GAN will be utilised as an oversampling method to overcome the class imbalance in this study to generate additional in-domain samples. This section describes the techniques mentioned above.

3.1 Convolutional Neural Network (CNN)

Convolutional neural networks [11] have yielded tremendous results over the past few years in many different areas related to pattern recognition, from image processing to voice recognition. However, the most significant advantage of CNN is reducing the parameters in ANN, which prompted researchers and developers to turn to CNN more to solve complex problems that the ANN could not solve. Therefore, the most critical assumption about issues resolved by CNN should not contain spatially dependent features. The rationale behind using the CNN architecture as a classifier is that CNN provides the highest accuracy with the lowest computational power since it has a reduced image dimensionality.

3.2 Transfer Leaning and Pre-trained CNN Models

Transfer learning [12] transfers the knowledge from a related problem that has already been learned. Although most machine learning algorithms are designed to manipulate single problems, the development of algorithms that facilitate transfer learning is an essential topic in the community of machine learning. It has been customarily for most computer vision tasks to use a pre-trained CNN model and fine-tune it with in-domain data. A pre-trained CNN model is a specific model trained for image recognition tasks using a massive volume of data such as ImageNet [14]. Here, four pre-trained CNN models will be explored: Xception, InceptionV3, VGG16 and ResNet50.

InceptionV3 [14], also known as GoogLeNet, is a CNN architecture from the Inception family that includes label smoothing, factorised 7×7 convolutions. In addition, it uses an auxiliary classifier to propagate label information lower down the network and uses batch normalisation for layers in the sidehead, among other improvements.

Xception model [15] is an extreme variant of the Inception model. It employs a depth-wise separable convolutional layer that has been modified. It starts with a 1×1 pointwise convolution and then moves on to a 3×3 depth-wise convolution. For feature extraction, the Xception architecture has 36 convolutional layers. A logistic regression layer follows it after the convolutional layer.

The VGG16 model [16], which stands for Visual Geometry Group from Oxford, has 16 layers in the VGG model architecture. There are many features associated with this model. For example, It just composed of convolution and pooling layers in the model. The kernel size is 2×2 for max-pooling and 3×3 for the convolution. There are about 138 million parameters in the VGG16 model and trained on the ImageNet dataset to predict 1000 different classes.

ResNet50 [17] is a CNN made of 50 layers: 48 convolutional layers, one max pooling, and one average pooling layer. The model was trained on the ImageNet dataset, which had 1 million samples and 1000 classes. It produced outstanding results on the ImageNet dataset with a Top-1 error rate of 20.5% and a Top-1 error rate of 5.25%.

Table 2 summarises the main characteristic of these four architectures. VGG16 has the most significant number of trainable parameters but with the fewest layers among all four models.

Table 2
Details of the pre-trained models in this study.

Model	Parameters	Layers	Size (MB)
ResNet50	25.6M	50	98
Xception	22.9M	126	88
InceptionV3	23.9M	159	92
VGG16	138.4M	23	528

3.3 Generative Adversarial Network (GAN)

GAN [19] is the new type of neural network that lies under generative models. As an innovative way to train a generative model, GAN was recently introduced. It consists of two adversarial models: a generator and a discriminator. The generator is a generative model that captures the data distribution. At the same time, the discriminator is a discriminative model that measures the probability that a sample originated from the training data instead of the generator. A non-linear mapping function, such as a multi-layer perceptron, may be used for both models.

The generator and discriminator are both trained simultaneously using a two-player min-max game with a joint loss function. Parameters for the generator are updated to minimise the joint loss while the discriminator's parameters are adjusted to maximise it.

One of the GAN's extensions is Style Generative Adversarial Network [20], or StyleGAN for short. It introduced significant changes to the generator model, such as using a mapping network to connect points in latent space to an intermediate latent space and the intermediate latent space to monitor style at each point in the generator model.

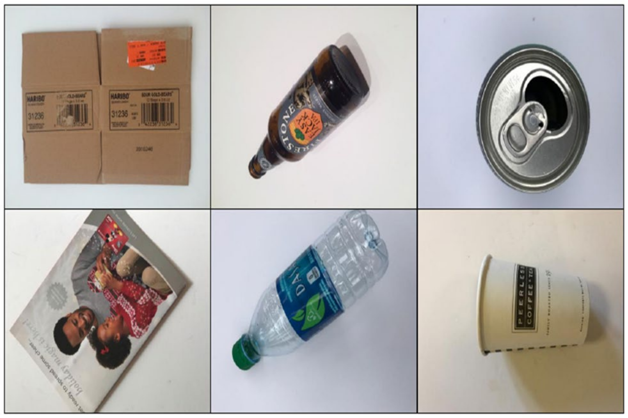

Figure 1: Samples from the TrashNet dataset [11].

4. TrashNet Dataset

In this study, the TrashNet dataset [11] is used. It consists of 2527 waste images classified into six classes: glass, metal, trash, paper and cardboard. Each image has a dimension of 512×384 and contains a single item captured on a solid background using sunlight and/or room lighting. Figure 1 illustrates a sample from each class in the dataset. For this study, TrashNet was split into three sets: training, validation and testing with ratios 70%, 13%, and 17%, respectively. Table 3 shows the sample's distribution among the classes for each subset. The number of samples per class is not balanced, as depicted in Figure 2.

Table 3
Sample distributions among the classes for training, validation, and testing subsets.

Class	Train	Valid	Test	All
Cardboard	287	46	70	403
Glass	354	65	82	501
Metal	286	56	68	410
Paper	403	83	108	594
Plastic	347	61	74	482
Trash	91	17	29	137
Total	**1768**	**328**	**413**	**2527**

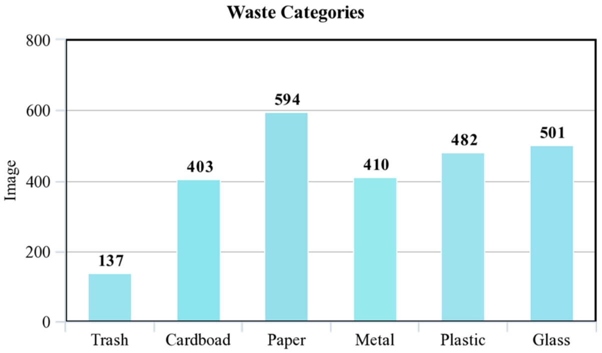

Figure 2: Number of samples per class within the TrashNet dataset.

5. Experiments

5.1 Experimental Design

All experiments were implemented and evaluated in Python and leverage TensorFlow and PyTorch [22] library using 16GB RAM and 1TB SSD machine. An overview of the implantation design is depicted in Figure 3.

The dataset is already split into 70:13:17 for training, validation, and testing in the first stage, respectively. Then, in stage two, fine-tuned models (transfer learning) are used based on ResNet50, Xception, InceptionV3 and VGG16 to get a robust trained classifier and fine-tune the parameters accurately. After that, GAN-based data augmentation is used to augment the dataset and introduce variability in the dataset without actually collecting new data [23]. Last, in the testing stage, the fine-tuned models are used to predict unknown labels using the testing subset, and the performance measurements will be calculated.

5.2 Data Preprocessing

Most of the pre-trained models were trained with images of size 224×224. Accordingly, images of the TrashNet dataset were rescaled using bicubic interpolation for optimum performance. In addition, all images were normalised on pixel-level using their mean and standard deviation to ensure that the model learns from the data faster.

5.3 Data Augmentation

Deep models are prone to overfitting during training. Data augmentation techniques generate additional in-domain data to reduce the possibility of overfitting the existing training set. New samples are generated by applying a single or a combination of geometric transformations such as rotation, translation, scaling, and flipping [24].

Geometric transformations are not the only way to generate new in-domain samples. Generative models can be trained on existing data to create additional samples from the same space. In this paper, the styleGAN [20] model was trained to generate 1000 images per class.

The amount of generated samples are controlled by an augmentation factor. In other words, if the augmentation factor was set to 0.2, then 20% additional images will be added to the training set. There were six settings for adding GAN-generated data augmentation (GDA) in this study: 0%, 20%, 50%, 100%, 150% and 200%.

5.4 Fine-tuning the Pre-trained CNN Models Preprocessing

For every pre-trained model in this study, the last layer, i.e. output layer, was modified to be equivalent to the number of classes, 6. Fine-tuning was performed in 2 steps. First, the weights were frozen for all layers except the newly added layers, which were trained with Adam optimiser using cross-entropy loss with a learning rate of 0.001. Then, the entire model was trained without freezing but with a learning rate of 0.00001.

6. Results and Discussion

Each pre-trained model was used as a core for six models with different ratios of added GDA. When used on validation and test sets, figures 4 and 5 summarise the

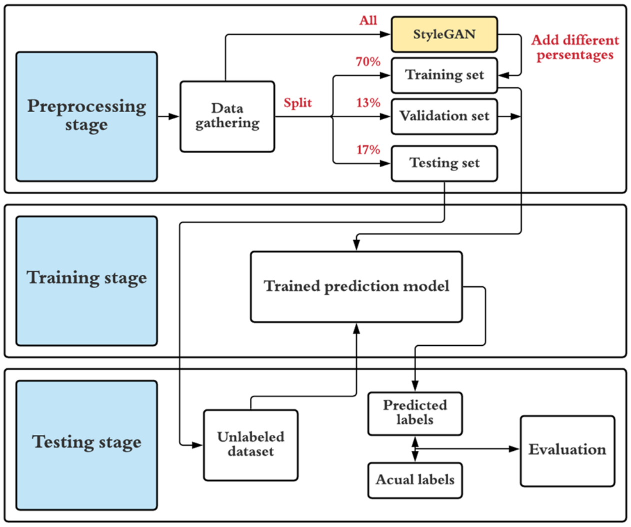

Figure 3: The experimental design followed in this study.

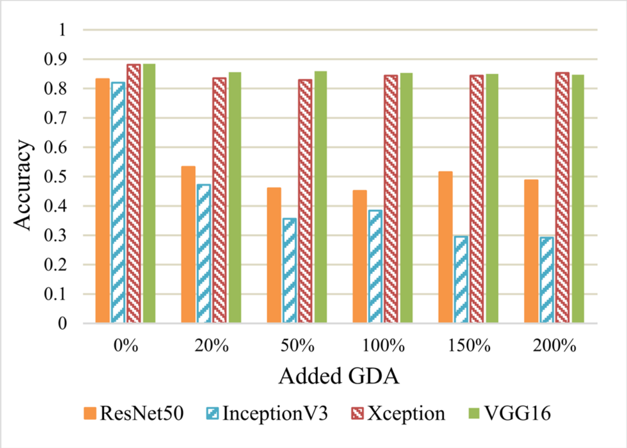

Figure 4: Performance of models on validation set when different rations of GDA were added to the training data.

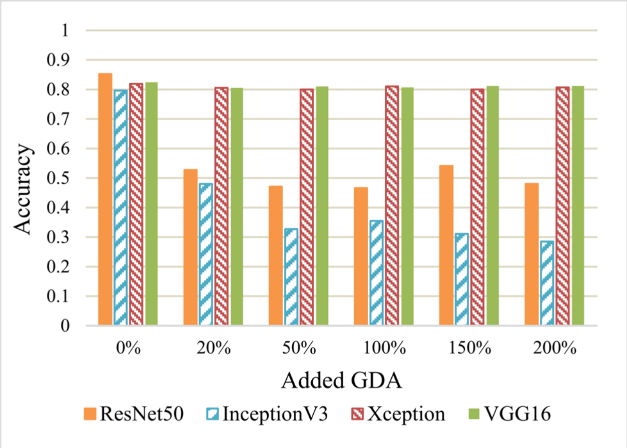

Figure 5: Performance of models on test set when different rations of GDA were added to the training data.

outcomes of using ResNet50, InceptionV3, Xception and VGG16, respectively.

The validation set was used to tune the hyperparameters of the models; hence, all models performed better when applied to the validation set compared to the test set.

Models based on ResNet50 and InceptionV3 fluctuated when GDA data was introduced, with a minimum drop of 37-39% accuracy when 20% GDA data were added. It even got worse when more GDA data was added to reach a reduction of 44-64% when 200% GDA data was used.

On the other hand, models based on Xception and VGG16 were not sensitive to the added GDA data, with a 1-2% performance drop.

Without adding any GDA, i.e. 0% augmentation factor, all models were performing better. It seems that GAN was not trained enough, and more data is required to achieve better results. GAN needs to be well-trained using many images to capture the distribution of the classes; otherwise, the model will generate a bad approximation of the distribution. Using under-trained GAN for data augmentation means adding more noise to the training data instead of in-domain samples, leading to confused models.

7. Conclusion

In this paper, a comparative study was performed to measure the impact of using GAN-based data augmentation with fine-tuned models. The performance of the final models was measured on a waste classification task using four pre-trained models as core models: ResNet50, IncepetionV3, Xception and VGG16. Additional data was generated using the GAN-based model, styleGAN, trained on the training set, with different ratios between 0% and 200%.

It was found that models based on ResNet50 and InceptionV3 were more sensitive to the added machine-generated images, whereas models based on Xception and VGG16 were more robust. The performance degradation was due to the undertrained styleGAN model, which generated noisy and abstract samples. So added data was considered as added noise. In future work, an enhanced GAN model would be trained to generate more realistic images.

References

[1] H. Wang, "Garbage recognition and classification system based on convolutional neural network vgg16," in 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEM-CSE). IEEE, 2020, pp. 252–255.

[2] Y. Liao, "A web-based dataset for garbage classification based on shanghai's rule," International Journal of Machine Learning and Computing, vol. 10, no. 4, 2020.

[3] M. Zeng, X. Lu, W. Xu, T. Zhou, and Y. Liu, "Public garbagenet: A deep learning framework for public garbage classification," in 2020 39th Chinese Control Conference (CCC). IEEE, 2020, pp. 7200–7205.

[4] L. Cao and W. Xiang, "Application of convolutional neural network based on transfer learning for garbage classification," in 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC).IEEE, 2020, pp. 1032–1036.

[5] R. Sidharth, P. Rohit, S. Vishagan, R. Karthika, and M. Ganesan, "Deep learning based smart garbage classifier for effective waste management," in2020 5th Inter-national Conference on Communication and ElectronicsSystems (ICCES). IEEE, 2020, pp. 1086–1089.

[6] M. Yang and G. Thung, "Classification of trash for recyclability status," CS229 Project Report, vol. 2016,2016.

[7] R.A.Aral,S¸.R.Keskin,M.Kaya, and M. Hacı ̈omero ̆glu, "Classification of trashnet dataset based on deep learning models," in2018 IEEEInternational Conference on Big Data (Big Data).IEEE, 2018, pp. 2058–2062.

[8] S. L. Rabano, M. K. Cabatuan, E. Sybingco, E. P. Dadios, and E. J. Calilung, "Common garbage classification using mobilenet," in2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). IEEE, 2018, pp. 1–4.

[9] U. Ozkaya and L. Seyfi, "Fine-tuning models comparisons on garbage classification for recyclability," arXivpreprint arXiv:1908.04393, 2019.

[10] A. H. Vo, M. T. Vo, T. Leet al., "A novel framework for trash classification using deep transfer learning," IEEEAccess, vol. 7, pp. 178 631–178 639, 2019.

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in2017International Conference on Engineering and Technology(ICET). Ieee, 2017, pp. 1–6.

[12] I. Kandel and M. Castelli, "Transfer learning with convolutional neural networks for diabetic retinopathy image classification. a review," Applied Sciences, vol. 10, no. 6,p. 2021, 2020.

[13] L. Torrey and J. Shavlik, "Transfer learning," in Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI Global,2010, pp. 242–264.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, andZ. Wojna, "Rethinking the inception architecture for computer

vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp.2818–2826.

[16] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition,2017, pp. 1251–1258.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXivpreprint arXiv:1409.1556, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu,D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," arXiv preprintarXiv:1406.2661, 2014.

[20] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF Conference on ComputerVision and Pattern Recognition, 2019, pp. 4401–4410.

[21] G. Thung and M. Yang, "Trashnet," GitHub repository,2016.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury,G. Chanan, T. Killeen, Z. Lin, N. Gimelshein,L.Antiga, A.Desmaison, A.Kopf, E.Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alch ́e-Buc, E. Fox, and R. Garnett, Eds.Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[23] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential data augmentation techniques for medical imaging classification tasks," in AMIA Annual Symposium Proceedings, vol. 2017.American Medical InformaticsAssociation, 2017, p. 979.

[24] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in2018 international interdisciplinary PhD workshop (IIPhDW). IEEE, 2018, pp. 117–12

Amani Alsabei received her B.S degree from Umm Al-Qura University, Makkah, Saudi Arabia, in 2017. She is currently a student in the M.S artificial intelligence at Computer Science Department at Umm Al-Qura University, Makkah, Saudi Arabia, and is expected to receive her degree in 2022

Ashwaq Alsayed received her B.S degree from King Abdulaziz University, Jeddah, Saudi Arabia, in 2017. She is currently a student in the M.S artificial intelligence at Computer Science Department at Umm Al-Qura University, Makkah, Saudi Arabia, and is expected to receive her degree in 2022.

Manar Alzahrani received her B.S degree from Taif University, Taif, Saudi Arabia, in 2018. She is currently a student in the M.S artificial intelligence at Computer Science Department at Umm Al-Qura University, Makkah, Saudi Arabia, and is expected to receive her degree in 2022.

Sarah Al-Shareef received the B.S. degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2005 and the M.S. degree in advanced computer science from Sheffield University, Sheffield, United Kingdom, in 2009 and a PhD degree in computer science from Sheffield University, Sheffield, United Kingdom, in 2015. Currently, she works as Assistant Professor in Computer Science Department at Umm Al-Qura University, Makkah, Saudi Arabia. Also, she is a member of IEEE Computer Society, IEEE Young Professional, IEEE Signal Processing Society. Her research interests include speech and Arabic technologies and especially automatic speech recognition ad acoustic modelling.