Jamie McQuire
CSC8635
J.Mcquire2@newcastle.ac.uk

Deep Learning for Automatic Speech Recognition

2020/01/24

1 Abstract

Automatic speech recognition is an important area of research in natural language processing that focuses on the classification of human voice signals with computational algorithms. The recognition of human voice commands offers potential benefits in developing future devices for smart homes and the assistance of people with visual impairments who might struggle to operate digital devices not tailored to them. This project explores the use of deep learning techniques for the classification of a vocabulary of human voice commands, providing a comparison between the performances of different types of deep learning models for audio recognition. The audio files used in this report are taken from Google's speech commands dataset, and the deep learning models are trained to classify a subset of 6 words and the full vocabulary specified by the Kaggle competition. The voice data was transformed into log-spectrograms allowing for the classification of the audio signals to be done using computer vision algorithms. The performances of the models were evaluated based on the correct classification of each command, and if the models could correctly identify when an audio signal contained silence. We found that the ResNet model achieved the highest accuracy for both sets of words, with the model achieving an accuracy of 84.6% on the competition's testing set. This work expanded on existing papers that treated the audio recognition problem as an image recognition problem by using both deep residual learning and recurrent models for image classification, comparing the results to the basic convolutional neural network model. The use of other well established deep learning models could improve the performance of the audio recognition model and should be investigated in future research.

2 Introduction

Automatic Speech Recognition (ASR) is a subfield of Natural Language Processing (NLP) which focuses on the use of algorithmic techniques for the translation of human voice signals into text. The transfer of information between different systems is achieved using a multitude of different methods. Human-to-Human communication is characterised by vocal and text-based communication methods, whereas human-to-machine communication is almost exclusively dominated by text-based communication via input and output devices. ASR allows for human-to-machine communication to adopt a more human-to-human approach to information transfer. There has been major growth in the commercial popularity of ASR systems, with many different companies such as Apple and Amazon utilising ASR to develop virtual artificial intelligence (AI) assistants, such as Siri and Alexa. The AI assistants can leverage ASR to carry out tasks following voice commands from the users. As well as successes in digital smart homes, ASR systems can potentially offer life-saving benefits when integrated into motor vehicle systems. Voice-enabled technology can potentially provide a viable alternative for drivers who need to urgently access their mobile devices, hopefully reducing the number of potential distracted drivers on the roads. ASR systems can also offer potential healthcare benefits by providing visually impaired or elderly people with a method of communicating with digital devices. Deep learning methods have found successes in ASR and other fields such as computer vision, medicine, and finance [1]. The models are considered to be state-of-the-art and have been utilised in many different prize winning competitions due to their ability at detecting patterns and trends in data. Models such as Deep Speech [2] have found incredible successes in speech recognition by simplifying the traditional architectures of end-to-end speech recognition systems while maintaining a high level of performance.

This report explores the use of deep learning techniques for ASR, utilising Google's Speech Commands dataset [3] to train a machine learning classification algorithm to learn audio commands. The report will begin by providing an overview of the different methods employed in designing and implementing the ASR system. The typical data pipeline for an ASR system can be seen in Figure 1. The performance of the machine learning classifiers are then evaluated when trained to identify only 6 words from the dataset. This is only a small subset of the dataset and allows for us to explore how the different machine learning models will perform on the larger dataset. The models used in the report are evaluated based on the predictions made by the model. The report then investigates how the deep learning algorithms perform when the vocabulary is expanded to the specified competition size, and the performance of the algorithms are evaluated using the Kaggle competition's testing set.

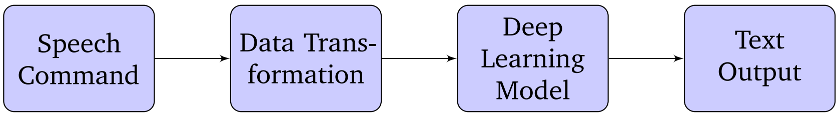


Figure 1: The data pipeline for an automatic speech recognition system.

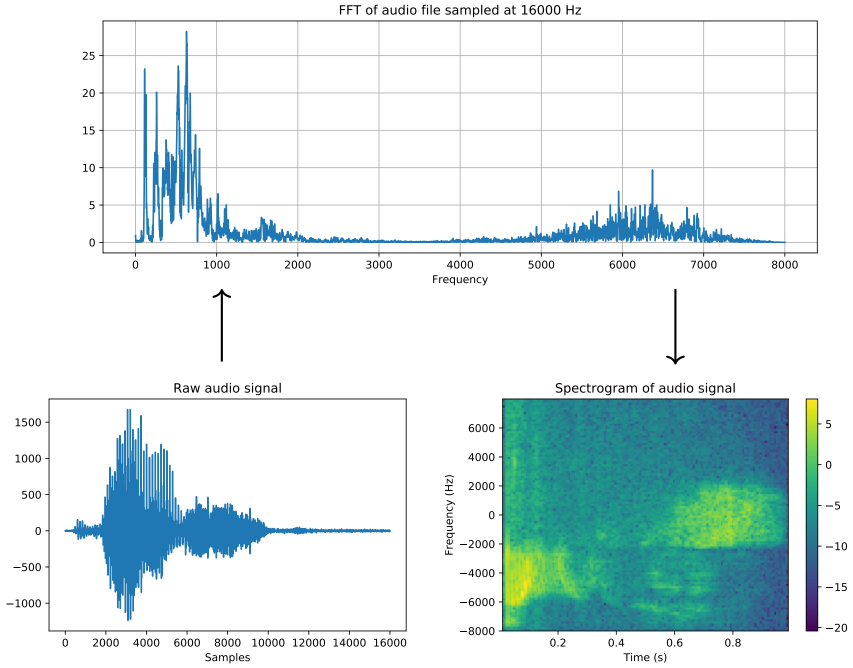


Figure 2: The data transformation process mapping the raw audio signal to a spectrogram.

3 Methods

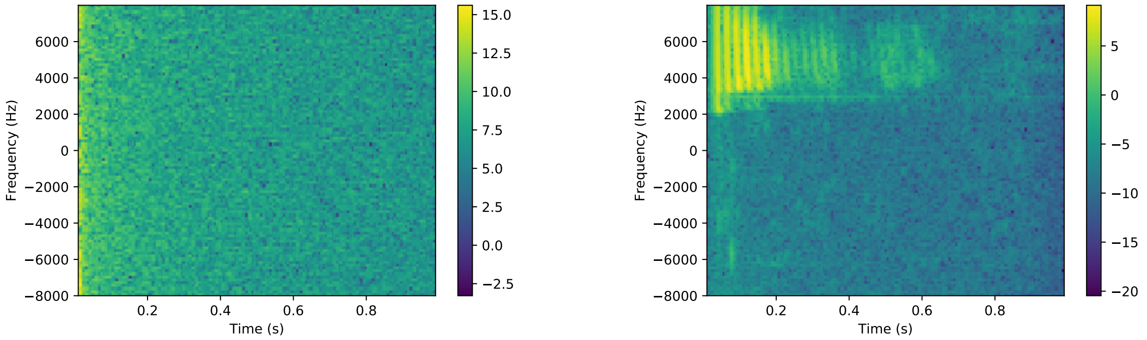
3.1 Data Pre-Processing

3.1.1 Data Representation

Google's Speech Commands dataset [3] contains audio files for a vocabulary of different spoken words and background noises. In their raw form, audio files are not compatible with machine learning models, thereby meaning that a suitable data transformation has to be employed to convert the data into a form that the machine learning models can understand. The audio files can be converted into a 1D time series vector in the Python environment using the SciPy package [4]. All of the audio files are sampled at a constant rate of $16,000$ kHz, meaning that no up-sampling or down-sampling is required when working between the different types of audio files. Although the data is now in a form that could be understood by the machine learning algorithms, we are going to transform the time series vector into a log-spectrogram. A spectrogram is an image representation of the frequency spectrum of a time-varying signal. The x-axis of a spectrogram represents time, and the y-axis represents frequency. The colour of the spectrogram pixel represents the power of the frequency component within the audio signal. The spectrogram is computed by applying a Discrete Fourier Transform (DFT) to the discrete-in-time signal to compute the estimated frequency spectrum [5]. The DFT is computed using the Fast Fourier Transform (FFT) algorithm, with a Hanning windowing function applied. The resulting image allows for a visualisation of how the different frequency components in the audio file's spectrum changes with time. The decision to convert the time series vector into a 2D image was motivated by the breakthrough successes of deep learning in the image recognition field, where convolutional neural networks (CNN) are prevalent [1]. The authors in [6] achieved positive results when treating the audio recognition problem as an image recognition problem, indicating that this approach to the classification of human voice commands is likely to be successful. From Figure 2, we can observe how the raw audio signal in the time domain is translated into the frequency domain via the FFT, with the output of the FFT being represented as a log-spectrogram.

3.1.2 Generation of Silence

An important task in the data pre-processing stage of the project was the generation of audio data that could be used to represent silence. The audio recognition challenge specified that one of the categories that the algorithm should be able to correctly classify is audio files which do not contain any human voice commands. The dataset used in this report contained 6 different audio files for the representation of background noise; this included ideal representations of noise, such as pink and white noise, and general household background noises, such as running taps and washing the dishes. The comparatively small size of the training data for the background noises indicated that we would have to apply some additional techniques to generate more audio samples to represent silence. The background noises were converted into 1D time series vectors, and then separated into one-second segments to create additional files. The sample size was further increased by adding random Gaussian noise to the time series data to generate more representations of silence. The resulting time series vectors were then converted back into audio files, which were then transformed into spectrograms. From Figure 3, we can observe that the spectrogram of the voice command is significantly different to the spectrogram of the background noise, suggesting that the image recognition network should be able to correctly classify when there is no voice command at the input.



(a) Spectrogram of pink noise.

(b) Spectrogram of a person saying “wow”

Figure 3: The two different spectrograms represent audio files where a voice command is present/absent.

3.2 Deep Learning Models

This section of the report will present a brief introduction about deep learning for image recognition, and detail the architecture of the different deep learning models used in the report. This report will consider the use of 3 different neural network models for the classification of the audio files.

3.2.1 Convolutional Neural Network

CNN are ubiquitous in the field of image recognition, with many successful prize winning competitors implementing them in their models. CNN are able to utilise three main ideas which make them successful in the field of machine learning: sparse interactions, parameter sharing, and equivariant representations [1]. The architecture of the simple (CNN) model used in this report was inspired by the CIFAR-10 CNN model from Keras [7]. The CIFAR-10 is one of the most popular computer vision databases, with many different researchers focusing on developing computer vision models to achieve state-of-the-art results on this data. Figure 4 highlights the components used in the machine learning model. The base layer employed in this model is the convolutional layer; the layer utilises the convolution operation between the input 2D image and a kernel to produce a 2D activation map [1]. The convolutional layers can provide a good method for extracting features such as edges, end-points, and corners from image data. The pooling layer is another key component of the CNN architecture; the layer provides a method of down-sampling the activation maps that are outputted from the convolutional layer. The benefits of using pooling layers is that it makes the output of the CNN locally translation invariant, meaning that the output is more robust to spatial changes of the feature in the image [1].

3.2.2 Recurrent Neural Network

Recurrent neural networks (RNN) are a type of artificial neural network that has been widely utilised with time varying sequential data, such as audio, video, and text [8]. The defining feature of a RNN is the feedback connection that allows for the network to update the current state based on the past states and the current

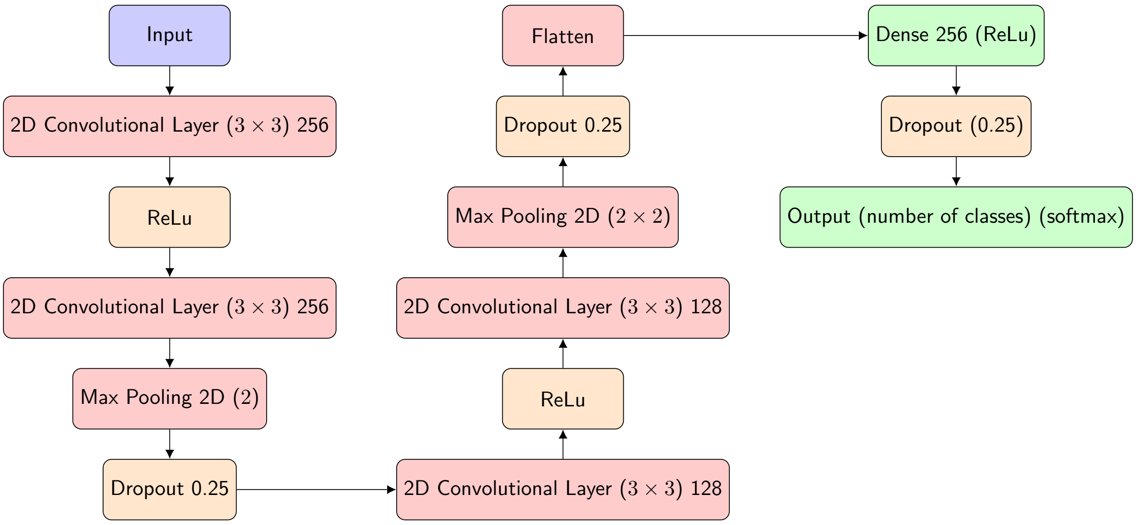

Input
2D Convolutional Layer (3×3) 256
ReLu
2D Convolutional Layer (3×3) 256
Max Pooling 2D (2)
Dropout 0.25
2D Convolutional Layer (3×3) 128
ReLu
2D Convolutional Layer (3×3) 128
Max Pooling 2D (2×2)
Dropout 0.25
Flatten
Dense 256 (ReLu)
Dropout (0.25)
Output (number of classes) (softmax)

Figure 4: The simple convolutional neural network architecture used in this report.

input data. The authors in [9] proposed an efficient gradient based method known as Long Short-Term Memory (LSTM). The main feature of the LSTM model is the introduction of self-loops which can provide a path for the gradient to flow during long periods of time [1]. LSTM addresses efficeincy issues that the RNN faced when learning to store information over extended intervals of time. LSTM layers have a very powerful learning capability and have been included in many different audio related tasks [8]. The use of LSTM-based models for the audio classification challenge is highly appropriate because of the time-dependent structure of the spectrogram. The spectrograms x-axis is discrete-in-time, meaning that a LSTM-model should benefit from learning with the previous inputs to the system. Before being inputted to the model, the spectrograms channel dimension is removed. Figure 5 shows the structure of the RNN model used in this report.

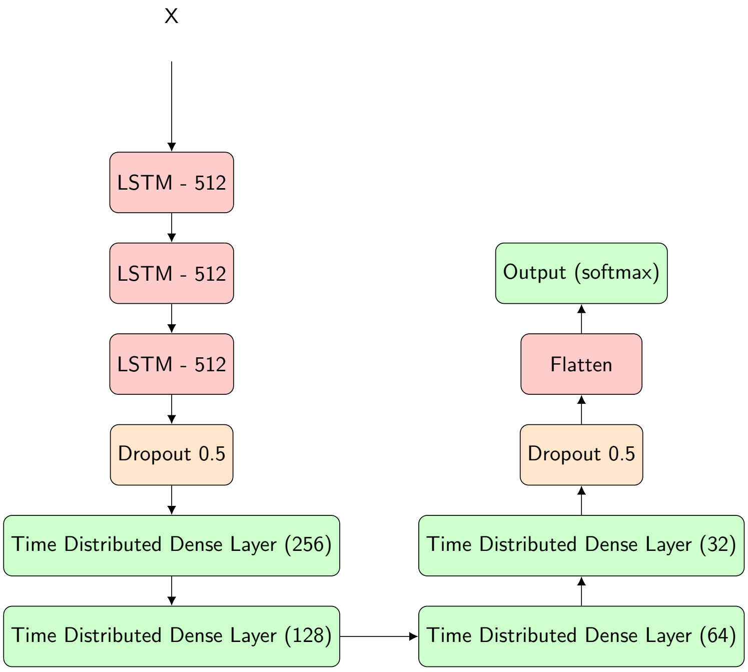
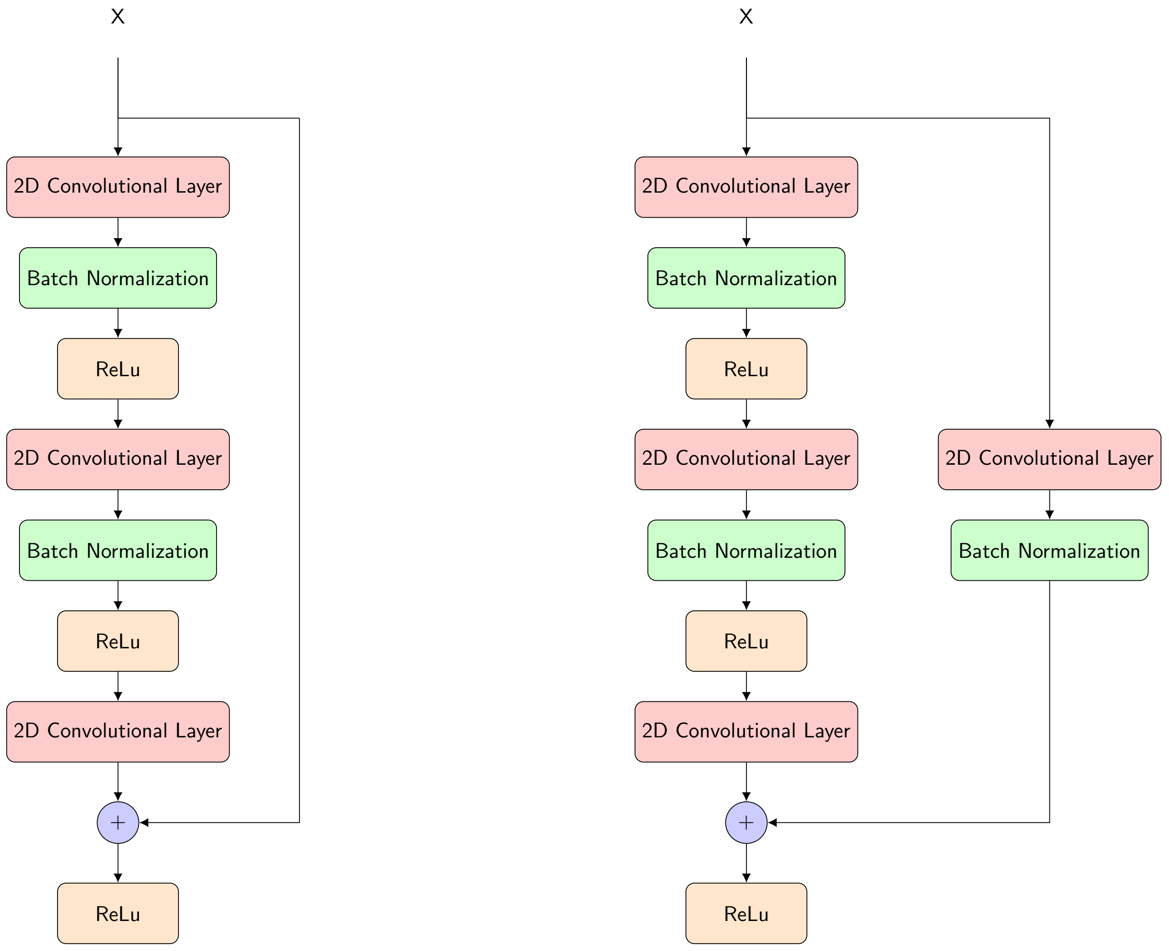

X
LSTM - 512
LSTM - 512
LSTM - 512
Dropout 0.5
Time Distributed Dense Layer (256)
Time Distributed Dense Layer (128)
Time Distributed Dense Layer (64)
Time Distributed Dense Layer (32)
Dropout 0.5
Flatten
Output (softmax)

Figure 5: The recurrent neural network model used in this report.

3.2.3 Deep Residual Neural Network

The deep residual neural network (ResNet) is a deep neural network architecture that was first introduced in [10]. The ResNet architecture provided an easier approach to training very deep neural networks by presenting a residual learning framework. The model was highly successful in the field of computer vision as it won 1^{st} place on the ILSVRC 2015 classification task [10]. It has been shown previously that deep networks naturally integrate hierarchical levels of features, and that the levels of features will benefit from increasing the depth of



(a) The identity block

(b) The convolutional block.

Figure 6: The identity and convolutional blocks architecture in the ResNet neural network.

the neural network [1]. We find when we stack more layers in the neural network that the gradients of the loss function start to approach zero, making the neural network harder to train; this problem is commonly known as the vanishing/exploding gradient problem [11]. There also exists a degradation problem where larger models will experience a diminished accuracy when being trained, which is not a consequence of overfitting [10]. Residual learning provides a solution to both the vanishing/exploding gradient problem, and the degradation problem. The deep residual neural network being implemented in this report is ResNet-50. The main components of the ResNet architecture are the identity block and the convolutional block. Figure 6a shows the structure of the identity block and Figure 6b shows the structure of the convolutional block. Both blocks have a ’skip’ connection present; the skip connection allows for the gradient to flow through an alternative path during back propagation, this will prevent the gradient from vanishing when we construct deeper networks. Figure 7 shows the full architecture of the ResNet-50 model used in this report. The ResNet combines the identity and convolutional blocks to form a very deep neural network that has a high performance with image recognition, and does not suffer from any of the previously discussed issues.

3.3 Training and Evaluating the Models

The models were trained and evaluated in a Google Colaboratory environment; this was due to the size of the training data and the depth of the models imposing a large computational workload. Google Colaboratory provides a free graphics processing unit (GPU) that can be used as a hardware accelerator, significantly reducing the time taken to train the neural networks. For the 6-word vocabulary experiment, each neural network was trained and tested using the same train, validation, and testing files. The vocabulary size was selected as 6 because it would allow for us to observe how well the classifiers work at differentiating between individual words with no silence included. Having the smaller training size also offered benefits in reducing the training time of the models. When the full vocabulary was used, the model was trained to predict 12 classes including silence, with the final class being unknown which represented a human voice command that was not any of the specified labels. The full vocabulary experiments had the training and validation sets specified by the competition, with an external testing set that would be evaluated by the competition’s results

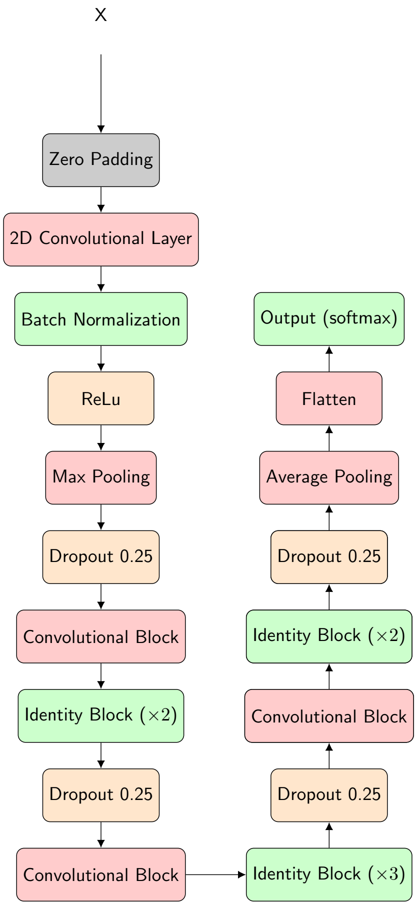


Figure 7: The ResNet 50 architecture used in this report.

on Kaggle. The final predictions made by the models on the testing sets were uploaded to Kaggle and a public and private score was returned. The unknown class had a disproportionate amount of data relative to the other classes; this meant that we had to include class weights when fitting the trained model to compensate for the uneven training groups. Before being used to train the neural networks, the image data's pixels were scaled by a factor of $\frac{1}{255}$ as neural networks prefer to work with standardized data [6]. All of the DL models used in this report have a categorical cross-entropy loss function, which is optimized using the Adam optimizer. To maintain consistency, all of the models are trained over 20 epochs with a batch size of 128. The models are evaluated in terms of the accuracy of the voice command predictions.

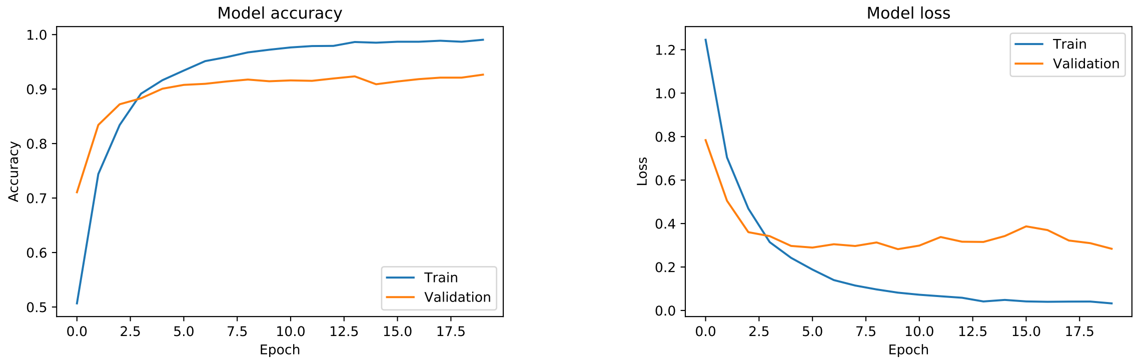
4 Results

4.1 6-Word Vocabulary

In this section of the report the deep learning models are trained and evaluated for correctly identifying 6 words from the speech recognition dataset. The words are: yes, no, up, down, left, and right. The models are going to be trained and evaluated using the training and validation datasets. The independent testing set is going to be used at the end of the training process to determine how well the deep learning models perform when being evaluated on unseen data.

4.1.1 Convolutional Neural Network

We can observe in Figure 8 how the accuracy and loss changes when the CNN is trained to correctly identify the 6 words. We can see in Figure 8a that the accuracy of validation set for the model has no significant increase after 5 epochs; we can also observe that the validation loss appears to no longer decrease after 5 epochs in Figure 8b. This is indicative of the model starting to overfit the training data, as the model's training accuracy is no longer representative of the validation accuracy. This allows for us to now train the model for only 5 epochs and then evaluate on the testing data to reduce the effects of overfitting. Figure 9 shows the confusion matrix for the classifications made by the final model. The model appears to correctly classify the voice commands with only a few incorrect classifications.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

Figure 8: Plot of the accuracy and the loss for the convolutional neural network trained on the 6-word vocabulary dataset.

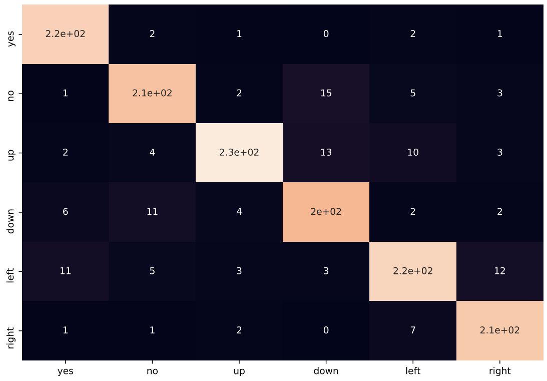
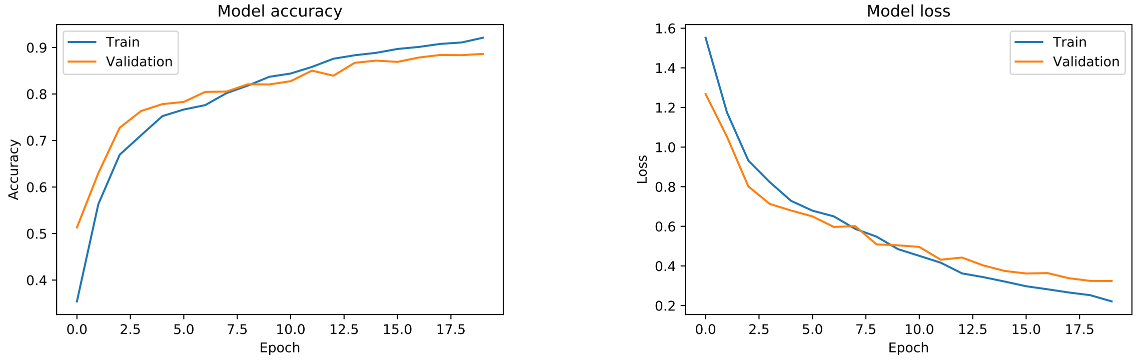


Figure 9: Plot of the confusion matrix of the convolutional neural network's testing set classifications.

4.1.2 Recurrent Neural Network

From Figure 10 we can observe how the values of the accuracy and loss for the training and validation data changes when the machine learning algorithm is trained. We can see in Figures 10a and 10b that the accuracy and loss increases and decreases, respectively. This starts to become relatively slow as the number of epochs increases. Figure 11 shows that the predictions made the model appear to be mostly correct, with only a small proportion of audio files being incorrectly classified.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

Figure 10: Plot of the accuracy and the loss for the recurrent neural network trained on the 6-word vocabulary dataset.

4.1.3 Deep Residual Neural Network

Figure 12 shows the plots of the accuracy and loss values when the ResNet model is trained over 20 epochs. From Figure 12a, we can observe that the accuracy of the predictions of the training set made by the ResNet

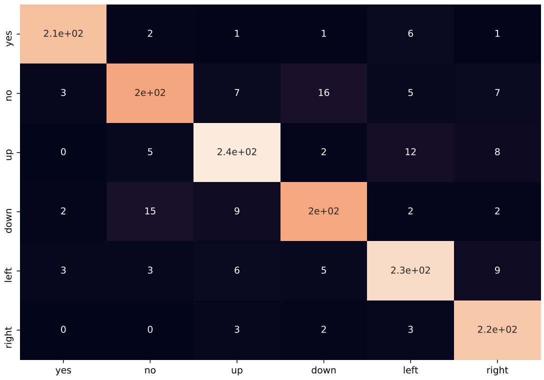
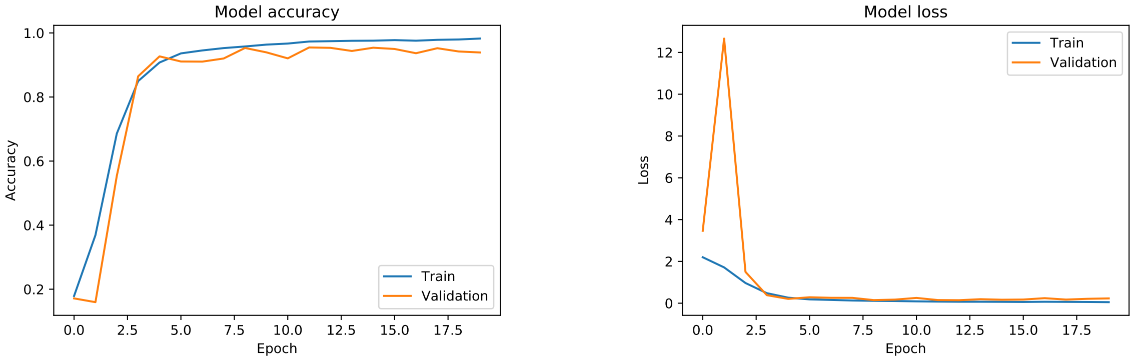


Figure 11: Plot of the confusion matrix of the recurrent neural network's testing set classifications.

appears to increase with the accuracy of the predictions from the validation set. We can also observe that in Figure 12b that the loss values for both the training and validation sets appears to remain relatively consistent over the training epochs. There appears to be a spike in the value of the loss at the start of the training process, but this appears to die out quickly with further training. The plot of the confusion matrix in Figure 13 shows that the ResNet appears to correctly classify the majority of audio files, with only a minor number of incorrect classifications.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

Figure 12: Plot of the accuracy and the loss for the ResNet model trained on the 6-word vocabulary dataset.

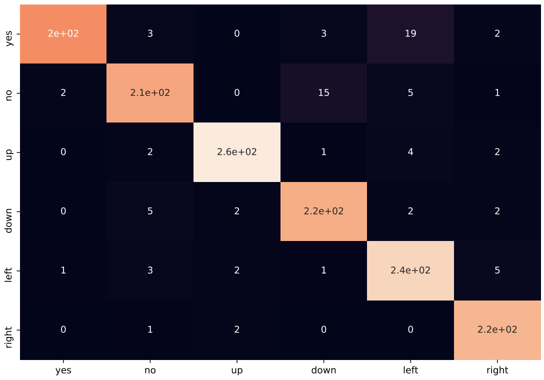


Figure 13: Plot of the confusion matrix of the ResNet model's testing set classifications.

4.1.4 Model Evaluation

Table 1 provides a summary of the accuracy values following the training on the final model. We can see that the ResNet model appears to achieve the best validation and testing accuracy for all of the models. The RNN model achieves the lowest training, validation, and testing accuracy, suggesting that the recurrent model is not as well suited to the image recognition problem. However, the RNN model's values are still relatively high, indicating that a more established RNN architecture could be used to achieve better results.

Model	Training (%)	Validation (%)	Testing (%)
CNN	94.2	91.2	90.6
RNN	92.1	88.6	90.1
ResNet	98.3	93.9	94.0

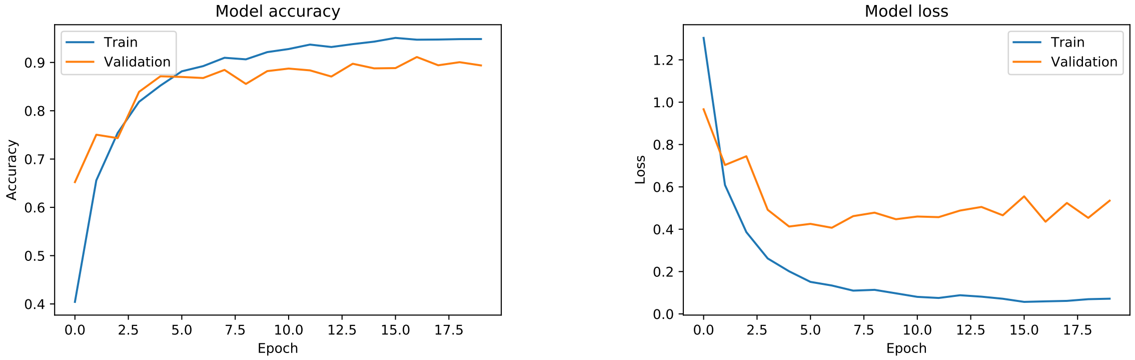
Table 1: Comparison of the training, validation, and testing accuracy of the models.

4.2 Full Vocabulary

This section of the report will train and evaluate the deep learning models for correctly identifying the audio commands specified by the Kaggle challenge. The audio categories are: yes, no, up, down, left, right, on, off, stop, go, silence, and unknown. The unknown category contains every other voice command that is not already specified. The models are going to be trained and evaluated using the training and validation sets. The challenge provided us with an independent testing set which the models will use to make predictions with. The predictions made by the model will be uploaded to Kaggle, and the public and private accuracy scores will be used to compare the model's performances.

4.2.1 Convolutional Neural Network

From Figure 14 we can see how the model's accuracy and loss values change when the model is trained. Figure 14a shows that the accuracy of the training and validation sets increases during training, with the accuracy of both sets seeming to increase slowly past 5 epochs. We can also see in Figure 14b that the loss appears to decrease and then become steady after 5 epochs. Because of the increase in the accuracy of both the training and validation sets, we trained the final model for the competition submission on 20 epochs.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

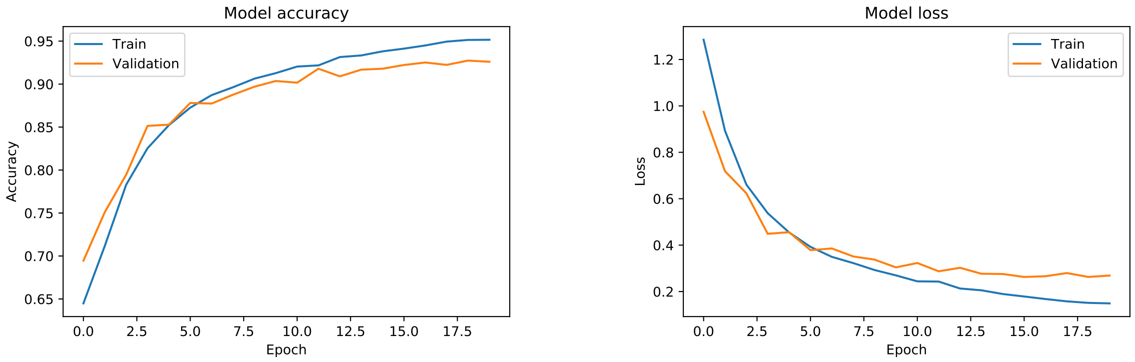
Figure 14: Plot of the accuracy and the loss for the convolutional neural network trained on the full competition dataset.

4.2.2 Recurrent Neural Network

Figure 14 shows how the model's accuracy and loss changes over the training epochs. We can see in Figure 15a that the accuracy of both the training and validation sets appears to steadily increase over the training epochs, with a slightly higher accuracy for the training data. The loss of the training and validation sets appears to decrease over the training epochs in Figure 14b, indicating that the model fits very well to the data. The final model was trained for 20 epochs because of these results.

4.2.3 Deep Residual Neural Network

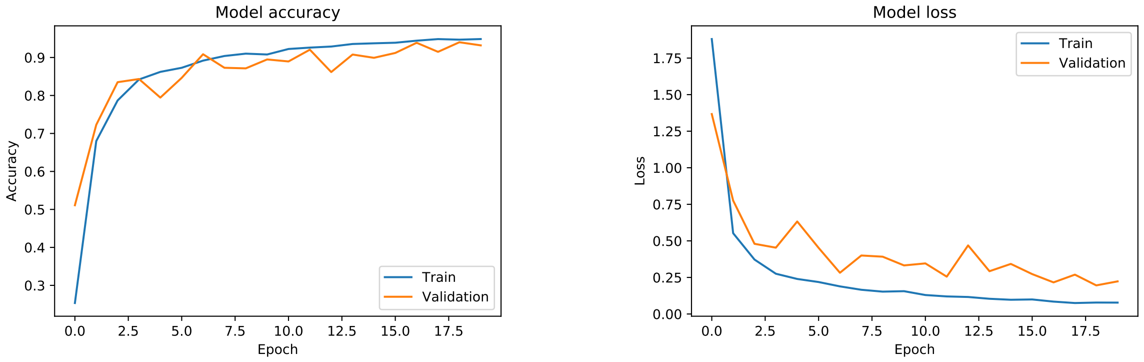
We can see in Figure 16 how the ResNet model's accuracy and loss changes over the training epochs. Figure 16a shows that the accuracy of the training and validation sets sharply increases over the first few epochs, and then slowly increases over the remaining epochs. The loss of the training and validation sets appears to decrease sharply before starting to slowly decrease in Figure 16b. It is apparent that the validation set's accuracy has fluctuations in its plot that slowly die out, suggesting that the set number of training epochs is necessary to help the model generalise well to the training data.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

Figure 15: Plot of the accuracy and the loss for the recurrent neural network trained on the full competition dataset.



(a) Accuracy of the training and validation sets.

(b) Loss of the training and validation sets.

Figure 16: Plot of the accuracy and the loss for the ResNet model trained on the full competition dataset.

Model	Training (%)	Validation (%)
CNN	94.9	89.4
RNN	92.1	88.6
ResNet	98.3	93.9

Table 2: Comparison of the training and validation set's accuracy for the models.

4.3 Competition Results

From Table 3, we can evaluate how the models performed on the testing set for the competition. The ResNet model achieved the highest score out of all the models and positioned at around 327^{th} in the competition. This is a respectable score with the model being 6% in accuracy off the 1^{st} placed model. It is apparent that the large size of the testing data means that the algorithm did not generalise as well to the data. This means to potentially achieve a better score, we would either have to adopt a different deep learning architecture or expose the model to a wider variety of different audio files during training.

Model	Private Score (%)	Public Score (%)
CNN	71.5	70.5
RNN	75.1	74.5
ResNet	84.9	84.6

Table 3: Comparison of the public and private scores of the model's predictions for the competition.

5 Discussion

This section of the report will provide a discussion and analysis of the results in Section 4. When evaluating the performance of the ASR models for correctly classifying the 6-word voabulary dataset we have to compare the accuracy of the different deep learning models. The ResNet model has the highest accuracy for the training, testing, and validation datasets; this indicates that the model is the most suitable for the audio recognition challenge. The ResNet model potentially outperforms the CNN model because there is likely to be more hidden features present in the spectrogram that the shallow CNN model cannot extract. It is also important to note that the ResNet model has previously been considered to be state-of-the-art, thereby meaning that when it comes to image recognition, the ResNet model is likely to outperform the simpler model. The RNN model appeared to not have performed as well as the models that used convolutional layers. Although the spectrogram is a time series and definitely benefits from the LSTM layers, it is clear that the inclusion of convolutional layers is essential for a spectrogram-based speech recognition system. From Figures 9, 11, and 13 we can compare how the models performed at correctly classifying individual words in the 6-word vocabulary. We can see that both the CNN and RNN models incorrectly classified "down" as "no"" the most frequently; the ResNet model incorrectly classified this as frequently, but had the most incorrect predictions with "left" as "yes". The ResNet model had comparatively less incorrect classifications with the other words than the other models. When the vocabulary of the dataset was expanded to the competition's size, it became evidently clear that the ResNet model was the winner in terms of correctly classifying the voice commands; this is supported by the high public and private scores in Table 3.

Because of the successes with the deep residual learning model, future work in this area should either investigate other well established image recognition networks, or seek to increase the depth of the neural network. The RNN outperformed the CNN in the competition, indicating that a combination of LSTM and convolutional layers could potentially produce strong results, however, further research would need to be carried out to see if any existing networks have achieved meaningful results with this combination. An important aspect of the performance of the model is the quality of the data and its representation that is used to train the machine learning algorithms. The lower scores from the competition when compared to the scores from the validation set is an indication that the audio used to evaluate the model is dissimilar to the audio used to train the model. This could be due to a variety of different reasons, such as more noise being present in the recording, or the use of different people's voices. Therefore, it is evidently clear that in order to build an ASR system that could be deployed to hardware, the deep learning model will have to be significantly more resilient to perturbations and irregularities in the voice commands. The integration of Hidden Markov Models (HMM) into the ASR system could be a potential avenue for future work as HMMs have been shown to be robust to noisy audio [12]. Further investigation into the use of digital signal processing (DSP) techniques could be considered to be beneficial to the work, with the mixing of the training signals with noise signals potentially helping the deep learning models become accustomed to audio files that are of lower quality. Another future area of research with this project is the development of low-latency ASR networks for deployment onto mobile devices. This was considered by the authors in [6] who investigated low-latency CNN for audio recognition using spectrograms. The researchers in [13] presented a case study about deep learning models being integrated into mobile devices, and how larger models could be optimized to run locally on mobile devices with technologies such as Tensorflow Lite and Tensorflow Mobile.

6 Conclusion

This report has explored the application of machine learning for ASR through the employment of deep learning models for the classification of human voice commands. ASR is a very large field within NLP with many different researchers utilising different computational techniques to provide a method of transferring information to digital devices through human voice commands. This work used DSP techniques to transform the voice data into log-spectrograms to interpret the audio recognition problem as a computer vision problem, allowing for models that utilise both convolutional and recurrent layers to address the image recognition problem. The report explored the use of both CNN and RNN models that were designed by us, and a deep residual neural network that was taken directly from existing literature [10]. The investigation began by evaluating the different models ability at classifying a small subset of voice commands from the speech recognition dataset. This allowed us to form an understanding of how the models would perform when the vocabulary was expanded, and how well the learning algorithms classify different voice commands, which was visualized with confusion matrices. The report then expanded the vocabulary of the voice commands, and trained and evaluated the models for submission to the Kaggle competition. The ResNet model found the most success across all of the evaluations, with the model achieving a very respectable 84.6% public accuracy score, ranking us around 327^{th} in the public competition. The report then concluded with a short discussion of the experimental results, followed by a reflection into the potential future work that could be done in this area of research. Deep learning has massively contributed to the growth of ASR, driving the potential future of the voice-activated digital devices that will be ubiquitous in every smart-home, and paving the way for the next-generation of AI personal assistants.

7 Personal Remarks

This report has allowed for me to explore an area of research that I consider to be very interesting as an Electronic Engineering graduate. I have had previous exposure to digital signal processing and machine learning through undergraduate studies, and found the project to be very engaging when implementing techniques from these two fields together. I have really been engaged with the reading that I have done in this area, and would hopefully like to continue with it. I found the process of converting the audio files into spectrograms to be very challenging as I had to research the different methods and libraries that I would require. I found Google Colab essential for the project as the free GPU significantly cut down the time taken for my algorithms to train, and would definitely use again in a future project, or invest in a paid cloud computing service for projects with even more training data. In future machine learning work, I would dedicate more time to learning more established neural network models and seeing if I could implement them in a coding environment. I would also take an object-oriented approach to building the models so that I would not have to write out all of the functions for each Python notebook. I found submitting to the Kaggle competition extremely rewarding, and I had to stop myself from trying to achieve an even better score! This project has allowed for me to really engage with a modern deep learning area of research and build on my skills as a data scientist.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[3] P. Warden, "Speech commands: A public dataset for single-word speech recognition.," *Dataset available from http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz*, 2017.

[4] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, "SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python," *arXiv e-prints*, p. arXiv:1907.10121, Jul 2019.

[5] J. G. Proakis, *Digital signal processing: principles algorithms and applications*. Pearson Education India, 2001.

[6] S. K. Gouda, S. Kanetkar, D. Harrison, and M. K. Warmuth, "Speech recognition: Keyword spotting through image recognition," *arXiv preprint arXiv:1803.03759*, 2018.

[7] F. Chollet, "keras." `https://github.com/fchollet/keras`, 2015.

[8] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] 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*, pp. 770–778, 2016.

[11] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[12] S. Bengio, "An asynchronous hidden markov model for audio-visual speech recognition," in *Advances in Neural Information Processing Systems*, pp. 1237–1244, 2003.

[13] P. Sanabria, J. I. Benedetto, A. Neyem, J. Navon, and C. Poellabauer, "Code offloading solutions for audio processing in mobile healthcare applications: a case study," in *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pp. 117–121, IEEE, 2018.