Speech Emotion Recognition Using Convolutional Neural Networks For Use in Personal Robots

Michael Gislason

4 May 2020

Contents

1 Introduction

In the past few decades, increased access to advanced technology has radically transformed human life. Perhaps nothing better symbolizes this change than the widespread adoption of robots throughout the world. Robots are now in stores, in our homes and on our streets. Although robots have been in use in manufacturing for some time, the development of humanoid robots designed to interact with people in a human-like fashion presents an entirely new set of challenges for developers and programmers. To optimize human-robot interaction, beyond being capable of Automatic Speech Recognition (ASR), which transforms a speech sample into text, robots also must be able to take into account the underlying emotional state of the speaker to fully grasp intended meaning.

Thus, speech emotion recognition (SER) is a major area of focus for researchers. Without consciously realizing it, beyond processing the literal string of words spoken in an utterance, humans take into account both prosody (the acoustic aspects of an utterance that reflect emotional attitude and emphasis, like pitch, speed of sounds, and loudness), as well as kinesics (i.e. body language, including the movement of eyes, physical gestures, posture, etc) when interpreting what other people wish to communicate [1]. Traditional machine learning has proven to be moderately effective for the purpose of extracting prosodic features from utterances, but it has proven to difficult to generalize models beyond the context of artificial databases of utterances and associated emotional labels. Therefore, deep learning seems best suited for speech emotion recognition. By means of transforming speech samples captured as .wav files into Mel spectrograms, we can model SER as an image classification problem. In this paper, I propose two ”lightweight”, computationally inexpensive Convolutional Neural Network (CNN) architectures designed to extract prosodic features, and evaluate their performance. Furthermore, the effects on performance of techniques like dropout and batch normalization which have proven useful in increasing the generalization ability of neural networks and augmenting the training power of small datasets, will be evaluated. Finally, possible applications for these models are discussed, particularly in the context of personal robots like Aldebaran Robotics’ NAO, as well as possible areas of future research.

2 Related Work

As stated above, speech emotion recognition using deep learning techniques is an active of research in the field of natural language processing.

Tripathi *et al* [2] used a multi-modal approach to achieve results superior to previously state-of-the-art models on the IEMOCAP database, which is often used to train models for English language speech emotion recognition.Their team used four parallel convolutional layers with different kernel sizes to extract high-level prosodic features of varying size from audio speech samples, as well as separately deploying a text-based model which takes speech transcription (i.e. text) as input and creates word-embeddings (using Google's Word2Vec), which are then fed into a sequential CNN to extract emotive content. Finally the output of the audio and text-based neural is concatenated to output an emotional label. This multi-modal approach achieved high accuracy and showed good generalization ability. However, it is computationally expensive (involving two separate, dense neural networks and covolutional layers which learn 200 filters each) and is therefore as of yet unsuitable for use in robots with limited computing capacity.

Zheng *et al* [3] proposed an SER system which uses a lightweight CNN to extract prosodic features, and is specifically designed for use in personal robots. There is one sequential path of 5 separate convolutional layers, each of which learns only 16 filters; a random forest (RF) classifier is used to output a categorical emotional label using a database of Chinese language, emotionally-labeled utterances. Therefore, the model is computationally inexpensive in comparison with the model proposed by Tripathi *et al*, and is therefore suitable for use in personal robots like the Nao.

3 Speech Emotion Recognition Using Convolutional Neural Networks

3.1 Data Preprocessing

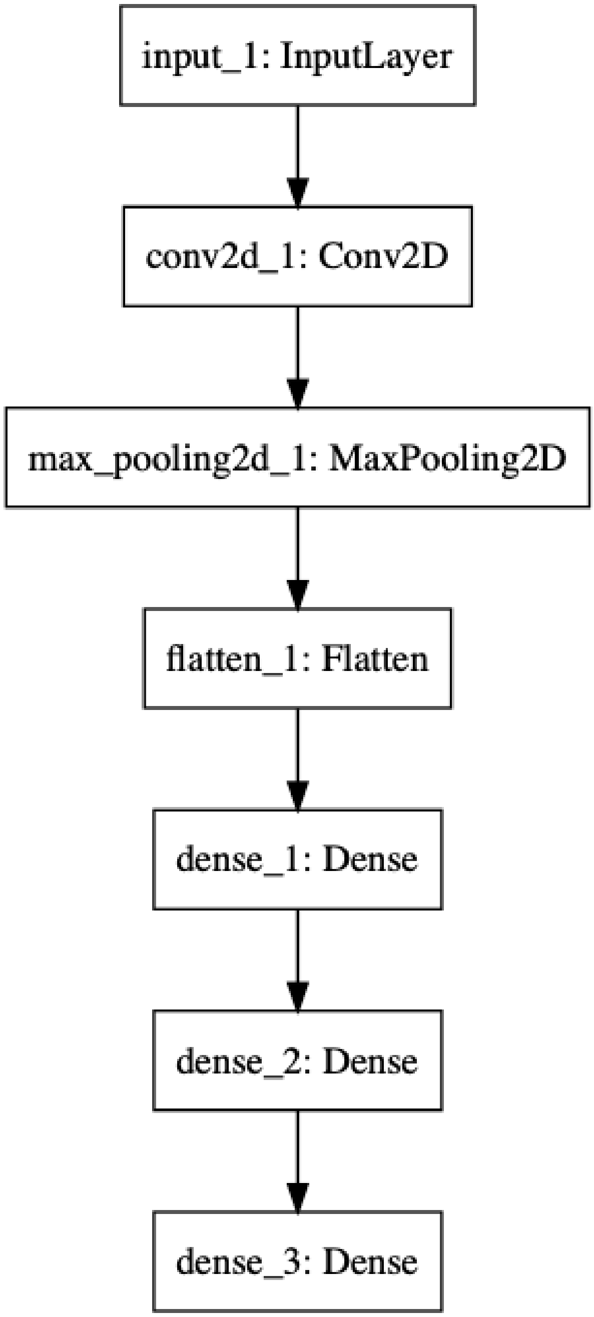
A spectrogram is an image-like representation of speech over time and frequency. It captures and visually represents prosodic features of natural language like pitch, speed of sounds, and loudness, which are lost when speech is converted directly to text[2]. Both of the architectures we propose use the Mel-frequency spectrogram as the input into a two-dimensional CNN. Luckily, the *librosa* Python library makes it quite easy to create Mel Spectrograms from common audio file formats like .wav.

First, the databases with which we will train, validate, and test our model are read in from directories. Three of the most popular English language databases used for speech emotion recognition are IEMOCAP (Interactive Emotional Dyadic Motion Capture), TESS (Toronto Emotional Speech Set) and RAVDESS (the Ryerson Audio-Visual Database of Emotional Speech and Song.) Due to the current university shutdown, we were unable to make contact with USC's SAIL laboratory to request access to IEMOCAP, and thus were only able to use RAVDESS and TESS for the purposes of this experiment. We chose to limit our experiment to 4 emotional classes (sad, happy, neutral, and angry), which compose 27%, 27%, 19% and 27%, respectively, of our total dataset of 3,551 audio files with an associated emotional label. 80% of the data was reserved for training, and of the remaining 20% two-thirds was reserved for the validation set and one-third (approximately 180 audio files) was reserved for the final test/evaluation. It is necessary to reserve a testing set not used for training or validation to convincingly demonstrate that the system generalizes well on unseen data.

Then, since convolutional neural networks require a fixed input size, the .wav file is trimmed or padded to 6 seconds. Then, we create a Mel Spectrogram using a sample rate of 22050 HZ, Fast Fourier Transform windows of length 2048, a hop length equal to 512, and 128 Mel coefficients per window, all of which are default parameters for *librosa*'s feature.melspectrogram method. Finally, the spectrogram is log-scaled by converting its power to decibels.

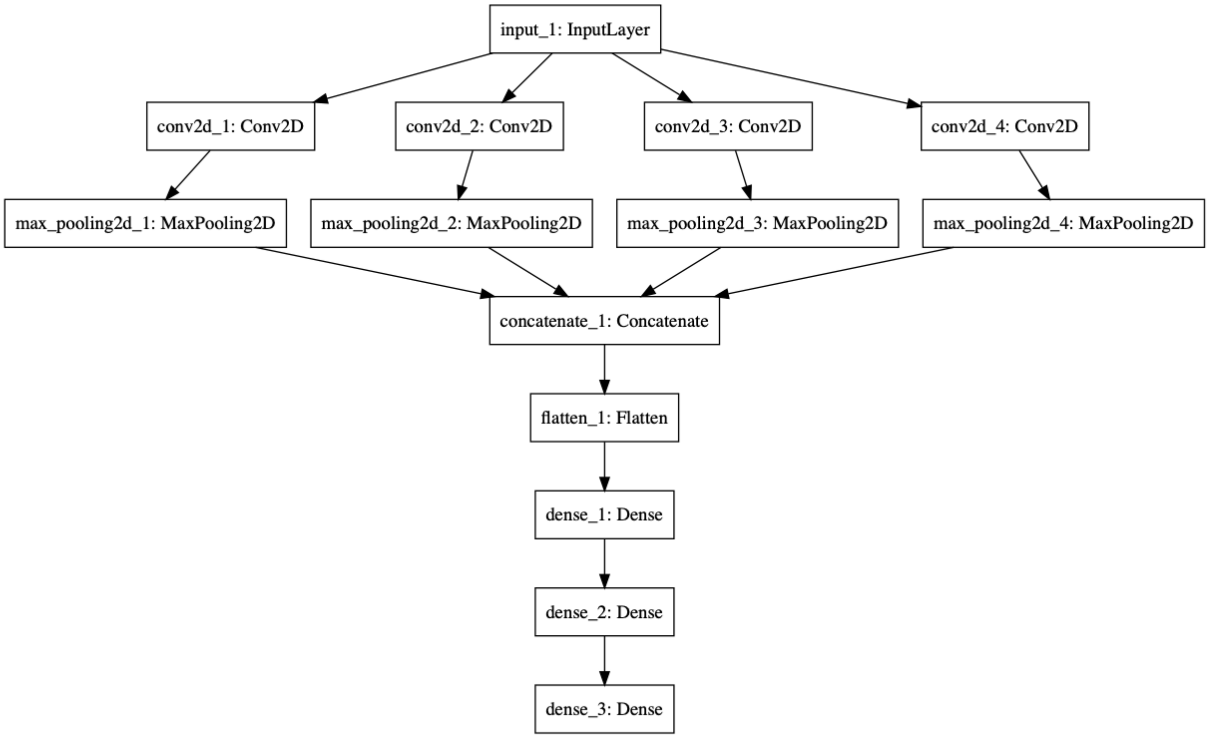
3.2 Model Architectures

Our goal is to accurately classify speech samples into one of four emotional categories; for this purpose, two main model architectures were created, trained and evaluated. First, a relatively simple sequential CNN employing a single convolutional layer (Architecture A):



The input layer takes Mel spectrograms as input. The convolutional layer uses a kernel size of 12 x 16, and learns 16 filters. Then, the max-pooling layer uses a pool size of 6 x 8, so that each filter generates 4 features. Then, successive dense layers (employing the reLu activation function) reduce our output to 32 and 16 neurons, before the final dense layers uses the Softmax function to produce output of the form *(happy, sad, neutral, angry)*, such that the sum of $happy + sad + neutral + angry = 1$, and each is in the range (0, 1.) Then, the argmax of this tuple provides us with an emotional label.

Next, Architecture B:

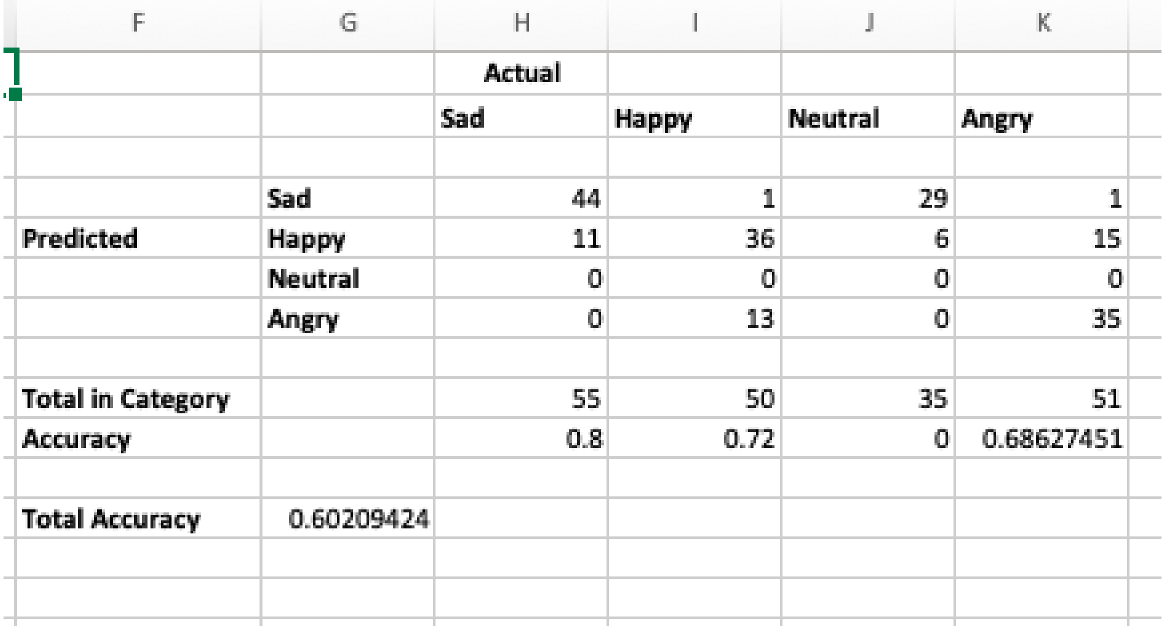


This model (Architecture B) is similar to the above, but as in Tripathi *et al*'s model, four parallel convolutional layers are employed, with kernel sizes 12 x 16, 18 x 24, 24 x 32 and 32 x 40, and each layer separately learns 16 filters. Again, we employ max pooling layers with a pool size with dimensions half that of our convolutional layers so that 4 features are extracted from each filter. After max pooling, the output of our 4 convolutional layers is concatenated, after which the model the functions exactly as in Architecture A.

3.3 Experimental Results

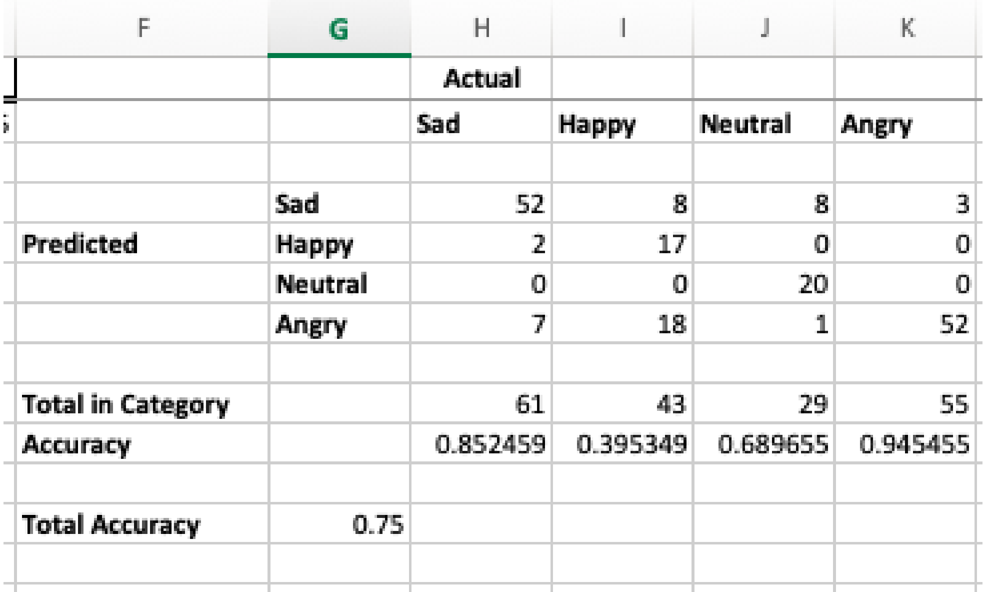
Models corresponding to Architecture A and Architecture B were created and trained for 20 epochs using the *Keras* Python library. The emotional labels predicted by each model and the actual emotional labels for the final test set of approximately 180 elements were compared to generate the following confusion matrices:

Architecture A:



F	G	H	I	J	K
		Actual			
		Sad	Happy	Neutral	Angry
	Sad	44	1	29	1
Predicted	Happy	11	36	6	15
	Neutral	0	0	0	0
	Angry	0	13	0	35
Total in Category		55	50	35	51
Accuracy		0.8	0.72	0	0.68627451
Total Accuracy	0.60209424				

Architecture B:



F	G	H	I	J	K
		Actual			
		Sad	Happy	Neutral	Angry
	Sad	52	8	8	3
Predicted	Happy	2	17	0	0
	Neutral	0	0	20	0
	Angry	7	18	1	52
Total in Category		61	43	29	55
Accuracy		0.852459	0.395349	0.689655	0.945455
Total Accuracy	0.75				

Several results are immediately interesting. Architecture B, employing convolutional layers in parallel, achieved 75% accuracy on the final test set whereas Architecture A only achieved 60.2% accuracy on the final test set. This seems to suggest that even when parallel convolutional layers learn relatively few filters (16 as compared to the 200 in Tripathi *et al*'s model), employing differently-sized kernels for each of the layers increases the number of prosodic features identified.

Furthermore, we see that Architecture A achieved 0% accuracy on the neutral category in the final evaluation. This suggests that its architecture is not learning enough features to effectively conduct SER, and is instead overfitting the training dataset (in which neutral speech samples are the least well-representated among our four emotional categories), and in effect has just learned to not guess the least common emotional label. Thus, it seems likely that Architecture A would not generalize well at all beyond this particular training dataset (TESS and RAVDESS.)

In contrast, we see that Architecture B performs extremely well when identifying sad and angry speech samples (85.2% accuracy and 94.5% accuracy, respectively), achieves 69% accuracy on neutral samples but only achieves 39.5% accuracy for speech samples labelled as happy. Nonetheless, even 39.5% accuracy is superior to guessing at random, which suggests that Architecture B is in fact identifying characteristic prosodic features for each of the emotional classes.

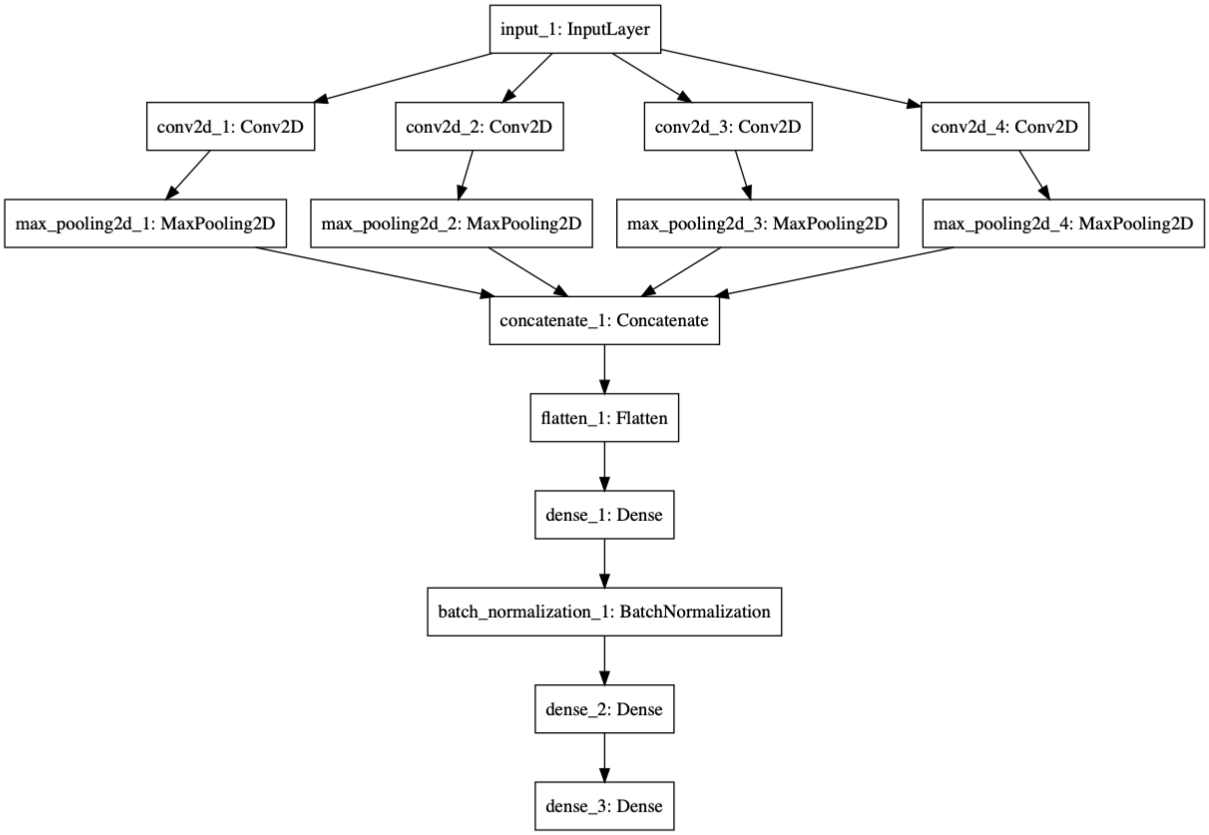
3.4 Experimental Results with Data Augmentation

Given the relatively small dataset with which we are working, which contains only 3,551 labelled speech samples, overfitting our dataset is a real concern: overfitting "has the effect of the model learning the statistical noise in the training data, which results in poor performance when the model is evaluated on new data, e.g. a test dataset. Generalization error increases due to overfitting" [4]. Several techniques exist to reduce overfitting and decrease generalization error: two commonly used methods are dropout and batch normalization.

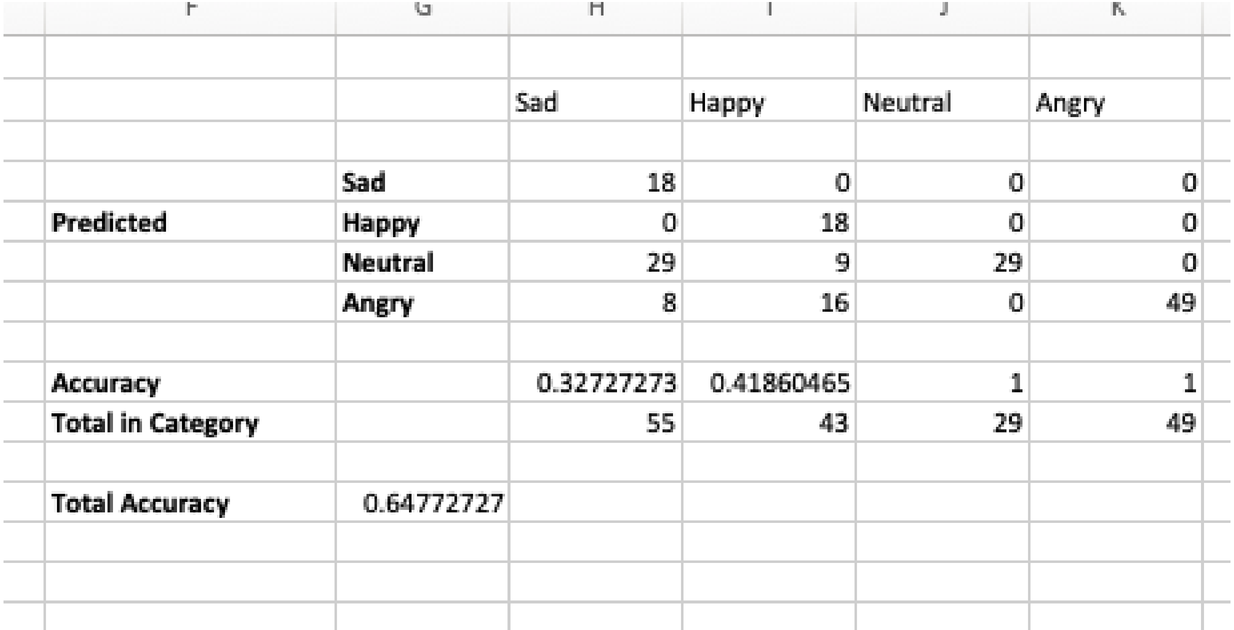
Dropout regularization is when some proportion of outputs of a particular layer are randomly ignored; this makes the training process more noisy, which "suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model more robust" [4]. Unfortunately, evaluation of the effects of dropout was made unfeasible by the technological limitations of the machine training the models (a personal laptop with a single 2.0 GHz processor.) Since the duration of a full training cycle of 20 epochs for any model already exceeded 24 hours, it was impractical to train the model incorporating dropout for hundreds of epochs.

Batch normalization is "a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks", and batch normalization may also have the effect of reducing generalization error by means of its regularization of inputs or the activations of a prior layer [5]. Since Architecture B performed best, it was tested with batch normalization after the first dense, fully-connected layer.

Architecture C:


input_1: InputLayer
conv2d_1: Conv2D
conv2d_2: Conv2D
conv2d_3: Conv2D
conv2d_4: Conv2D
max_pooling2d_1: MaxPooling2D
max_pooling2d_2: MaxPooling2D
max_pooling2d_3: MaxPooling2D
max_pooling2d_4: MaxPooling2D
concatenate_1: Concatenate
flatten_1: Flatten
dense_1: Dense
batch_normalization_1: BatchNormalization
dense_2: Dense
dense_3: Dense

The experimental results for Architecture C are presented in the same format as for A and B:



F	G	H	I	J	K
		Sad	Happy	Neutral	Angry
	Sad	18	0	0	0
Predicted	Happy	0	18	0	0
	Neutral	29	9	29	0
	Angry	8	16	0	49
Accuracy		0.32727273	0.41860465	1	1
Total in Category		55	43	29	49
Total Accuracy	0.64772727				

As compared with Architecture B, total accuracy declined from 75.0% to 64.8%. This isn't immediately promising, but there are some interesting results. Architecture C incorrectly identified 62 of the 176 speech samples in the final test set. Of these 62, 38 (61.2%) were incorrectly identified as neutral. For Architecture B, 47 speech samples of 188 were misidentified, of which 0 of 47 were incorrectly classified as neutral. Furthermore, Architecture C correctly identified 29 of 29 neutral samples in its final test set, whereas Architecture B correctly identified 20 of 29 neutral samples.

This finding is significant, since if the model fails to identify an emotional label, defaulting to neutral is desirable. If, for example, a person is angry, and the model fails to correctly classify said person's speech sample as angry, we would much prefer that our model/robot offer a neutral response rather than a happy response. Thus, despite achieving a lower percentage total accuracy, Architecture C seems most suitable for further testing and development.

4 Future Research

There are several, relatively simple ways in which the precision and accuracy of the SER model proposed in this paper for use in personal robots can be improved.

Firstly, there is an obvious need for more and more varied training data. The IEMOCAP database alone contains thousands more natural speech samples in the four categories used (happy, sad, neutral, angry). Beyond simply improving the accuracy and generalization ability of the mode, more data will also allow a greater number of speech samples to be reserved for the final testing set, which will facilitate more detailed analysis of experimental results and trends in the model's predictions. A less homogenous dataset with a greater number of speakers (RAVDESS uses, for example, 24 speakers and TESS 2) will most likely improve the ability of the model to accurately perform SER in a variety of contexts.

Secondly, I would suggest that larger and deeper models (involving both more dense, fully-connected layers as well as more convolutional layers) be tested as well especially for use in robots. These models will likely achieve greater accuracy, at the cost of increasing the computational expense processing an input through the neural network. It is true that personal robots like the NAO have a limited computing capacity, but as of yet lack of access has prevented the investigation of the maximum neural network parameters feasible on such a robot. Suppose that we wish that all speech samples be processed and an emotional label extracted in $t = 0.5$ seconds, so that the robot can respond quickly to natural speech. Since neural networks can be pre-trained, the factor limiting the size and depth of the neural networks should be this time t required for the robot's computer to preprocess an audio file, input said file into the neural network, and then output an emotional label. In practice, the processing power of the testing computer and the length of experiments limited the number of parameters in each of my proposed model architectures and experiments.

Thus, it is possible that CNNs significantly larger and deeper than those proposed above are suitable for use in personal robots like the NAO. In any case, it should be relatively simple to ascertain an upper limit on neural network size once access to the robotics laboratory computer is restored. Both dropout regularization and batch normalization seem promising in their ability to possibly increase the generalization ability of the SER models. Without the constraints of a laptop computer's hardware, a variety of configurations of dropout layers and batch normalization can be easily tested for the necessary number of epochs.

When looking at the problem of SER generally, it seems obvious that a multi-modal approach holds the greatest hope for the kind of high-accuracy, generalizable SER system needed for commercial use in personal robots. Given the wide availability of Automatic Speech Recognition technology, and existing use in personal robots, it seems feasible to simultaneously deploy 1) a text-based deep neural network for SER, 2) a separate deep neural network which uses spectrograms of speech samples to categorically label samples by prosodic features and 3) a dynamic neural network which can use a robot's many cameras to identify kinesic features (i.e. body language.) Then, an error in emotion recognition by one of three neural networks can be corrected by the other two.

Once accurate SER is achieved, for a given speech sample recorded by the robot, this information about the speaker's emotional state can be used to dynamically modulate the content of the robot's response, as well as the prosodic and kinesic features of the robot's vocalizations, producing a robot able to communicate in a manner more human-like and less "robotic". [6]

References

[1] Esposito, Anna, et al. “On the Amount of Semantic Information Conveyed by Gestures.” 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), 2015, doi:10.1109/ictai.2015.100.

[2] Tripathi, Suraj, et al. “Deep Learning Based Emotion Recognition System Using Speech Features and Transcriptions.” Samsung R&amp;D Institute India – Bangalore, 2019.

[3] Zheng, Li, et al. “Speech Emotion Recognition Based on Convolution Neural Network Combined with Random Forest.” 2018 Chinese Control And Decision Conference (CCDC), 2018, doi:10.1109/ccdc.2018.8407844.

[4] Brownlee, Jason. “A Gentle Introduction to Dropout for Regularizing Deep Neural Networks.” Machine Learning Mastery, 6 Aug. 2019, machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[5] Brownlee, Jason. “A Gentle Introduction to Batch Normalization for Regularizing Deep Neural Networks.” Machine Learning Mastery, 6 Aug. 2019, machinelearningmastery.com/batch-normalization-for-regularizing-deep-neural-networks/.

[6] Sylvester, Peter, and David Claveau. “A System to Convey the Emotional Content of Text Using a Humanoid Robot.” 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), 2016, doi:10.1109/ictai.2016.0162.