Neural Networks 92 (2017) 60–68

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2017 Special Issue

Evaluating deep learning architectures for Speech Emotion Recognition

Haytham M. Fayek [a,*], Margaret Lech [a], Lawrence Cavedon [b]

[a] School of Engineering, RMIT University, Melbourne VIC 3001, Australia
[b] School of Science, RMIT University, Melbourne VIC 3001, Australia

ARTICLE INFO

Article history:
Available online 21 March 2017

Keywords:
Affective computing
Deep learning
Emotion recognition
Neural networks
Speech recognition

ABSTRACT

Speech Emotion Recognition (SER) can be regarded as a static or dynamic classification problem, which makes SER an excellent test bed for investigating and comparing various deep learning architectures. We describe a frame-based formulation to SER that relies on minimal speech processing and end-to-end deep learning to model intra-utterance dynamics. We use the proposed SER system to empirically explore feed-forward and recurrent neural network architectures and their variants. Experiments conducted illuminate the advantages and limitations of these architectures in paralinguistic speech recognition and emotion recognition in particular. As a result of our exploration, we report state-of-the-art results on the IEMOCAP database for speaker-independent SER and present quantitative and qualitative assessments of the models' performances.

1. Introduction

In recent years, deep learning in neural networks has achieved tremendous success in various domains that led to multiple deep learning architectures emerging as effective models across numerous tasks. Feed-forward architectures such as Deep Neural Networks (DNNs) and Convolutional Neural Networks (ConvNets) have been particularly successful in image and video processing as well as speech recognition, while recurrent architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) RNNs have been effective in speech recognition and natural language processing (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). These architectures process and model information in different ways and have their own advantages and limitations. For instance, ConvNets are able to deal with high-dimensional inputs and learn features that are invariant to small variations and distortions (Krizhevsky, Sutskever, & Hinton, 2012), whereas LSTM-RNNs are able to deal with variable length inputs and model sequential data with long range context (Graves, 2008).

In this paper, we investigate the application of end-to-end deep learning to Speech Emotion Recognition (SER) and critically explore how each of these architectures can be employed in this task. SER can be regarded as a static or dynamic classification problem, which has motivated two popular formulations in the literature to the task (Ververidis & Kotropoulos, 2006): *turn-based processing* (also known as static modeling), which aims to recognize emotions from a complete utterance; or *frame-based processing* (also known as dynamic modeling), which aims to recognize emotions at the frame level. In either formulation, SER can be employed in stand-alone applications; e.g. emotion monitoring, or integrated into other systems for emotional awareness; e.g. integrating SER into Automatic Speech Recognition (ASR) to improve its capability in dealing with emotional speech (Cowie et al., 2001; Fayek, Lech, & Cavedon, 2016b; Fernandez, 2004). Frame-based processing is more robust since it does not rely on segmenting the input speech into utterances and can model intra-utterance emotion dynamics (Arias, Busso, & Yoma, 2013; Fayek, Lech, & Cavedon, 2015). However, empirical comparisons between frame-based processing and turn-based processing in prior work have demonstrated the superiority of the latter (Schuller, Vlasenko, Eyben, Rigoll, & Wendemuth, 2009; Vlasenko, Schuller, Wendemuth, & Rigoll, 2007).

Whether performing turn-based processing or frame-based processing, most of the research effort in the last decade has been devoted to selecting an optimal set of features (Schuller et al., 2010). Despite the effort, little success has been achieved in realizing such a set of features that performs consistently over different conditions and multiple data sets (Eyben, Scherer et al., 2015). Thus, researchers have resorted to brute-force high-dimensional

* Corresponding author.
E-mail addresses: haytham.fayek@ieee.org (H.M. Fayek), margaret.lech@rmit.edu.au (M. Lech), lawrence.cavedon@rmit.edu.au (L. Cavedon).

http://dx.doi.org/10.1016/j.neunet.2017.02.013
0893-6080/

features sets that comprise many acoustic parameters in an attempt to capture all variances (Tahon & Devillers, 2016). Such high-dimensional feature sets complicate the learning process in most machine learning algorithms, increase the likelihood of overfitting and hinder generalization. Moreover, the computation of many acoustic parameters is computationally expensive and may be difficult to apply on a large scale or with limited resources (Eyben, Huber, Marchi, Schuller, & Schuller, 2015). Therefore, it is highly pertinent to investigate the application of deep learning to SER to alleviate the problem of feature engineering and selection and achieve an SER with a simple pipeline and low latency. Moreover, SER is an excellent test bed for exploring various deep learning architectures since the task itself can be formulated in multiple ways.

Deep learning has been applied to SER in prior work, as discussed in Section 2. However, with different data subsets and under various experiment conditions involved in prior studies, it is difficult to directly compare various deep learning models. To the best of our knowledge, our work provides the first empirical exploration of various deep learning formulations and architectures applied to SER. As a result, we report state-of-the-art results on the popular Interactive Emotional Dyadic Motion Capture (IEMOCAP) database (Busso et al., 2008) for speaker-independent SER.

The remainder of this paper is divided into seven sections. In the following section, related work is reviewed, highlighting recent advances. In Section 3, a review of deep learning is presented focusing on the architectures and methods used in this paper. In Section 4, the proposed SER system is explained. In Section 5, the experimental setup is described, depicting the data, its preprocessing, the computational setup and the training recipe. Experiments performed and their results are presented in Section 6 and discussed in Section 7. Finally, the paper is concluded in Section 8.

2. Related work

Work on SER prior to 2011 is well reviewed in the literature (Ayadi, Kamel, & Karray, 2011; Petta, Pelachaud, & Cowie, 2011; Ververidis & Kotropoulos, 2006). Since DNNs displaced Gaussian Mixture Models (GMMs) for acoustic modeling in ASR (Hinton et al., 2012; Mohamed, Dahl, & Hinton, 2012), researchers have attempted to employ DNNs for other speech applications as well, and specifically for SER. Stuhlsatz et al. (2011) proposed a DNN Generalized Discriminant Analysis to deal with high-dimensional feature sets in SER, demonstrating better performance than Support Vector Machines (SVM) on the same set of features. In Li et al. (2013) a hybrid DNN–Hidden Markov Model (HMM) trained on Mel-Frequency Cepstral Coefficients (MFCCs) was proposed for SER and compared to a GMM–HMM indicating improved results. Han, Yu, and Tashev (2014) used a DNN to extract features from speech segments, which were then used to construct utterance-level SER features that were fed into an Extreme Learning Machine (ELM) for utterance-level classification outperforming other techniques. In Fayek, Lech, and Cavedon (2016a), a DNN was used to learn a mapping from Fourier-transform based filter banks to emotion classes using soft labels generated from multiple annotators to model the subjectiveness in emotion recognition which yielded improved performance compared to ground truth labels obtained by majority voting between the same annotators.

More recently, alternative neural network architectures for SER were also investigated. Mao, Dong, Huang, and Zhan (2014) used a ConvNet in a two-stage SER scheme that involves learning local invariant features using a sparse auto-encoder from speech spectrograms, processed using Principal Component Analysis (PCA) followed by salient discriminative feature analysis to extract discriminative features demonstrating competitive results. Tian, Moore, and Lai (2015) compared knowledge-inspired disfluency and non-verbal vocalization features in emotional speech against a feature set comprising acoustic parameters aggregated using statistical functionals, by using LSTM-RNNs as well as SVM, where the former was shown to yield better results given enough data.

This study differs from prior studies in several ways. We focus on a frame-based formulation for SER, aiming to achieve a system with a simple pipeline and low latency by modeling intra-utterance emotion dynamics. Moreover, most previous studies relied on some form of high-level features, while in this paper we strive for minimal speech processing and rely on deep learning to automate the process of feature extraction. Furthermore, we use uniform data subsets and experiment conditions promoting comparisons across various deep learning models, which has not been investigated in previous studies.

3. Deep learning: An overview

Deep learning in neural networks is the approach of composing networks into multiple layers of processing with the aim of learning multiple levels of abstraction (Goodfellow, Bengio, & Courville, 2016; LeCun et al., 2015). In doing so, the network can adaptively learn low-level features from raw data and higher-level features from low-level ones in a hierarchical manner, nullifying the over-dependence of shallow networks on feature engineering. The remainder of this section reviews the architectures, learning procedures and regularization methods used in this paper.

3.1. Architectures

The two most popular neural network architectures are the feed-forward (acyclic) architecture and the recurrent (cyclic) architecture (Schmidhuber, 2015). Feed-forward neural network architectures comprise multiple layers of transformations and nonlinearity with the output of each layer feeding the subsequent layer. A feed-forward fully-connected multi-layer neural network — also known as Deep Neural Network (DNN) — can be modeled by iterating over Eqs. (1) and (2):

$$\mathbf{h}^{(l)} = \mathbf{y}^{(l-1)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)} \tag{1}$$

$$\mathbf{y}^{(l)} = \phi(\mathbf{h}^{(l)}) \tag{2}$$

where $l \in \{1, \ldots, L\}$ denotes the lth layer, $\mathbf{h}^{(l)} \in \mathbb{R}^{n_o}$ is a vector of preactivations of layer l, $\mathbf{y}^{(l-1)} \in \mathbb{R}^{n_i}$ is the output of the previous layer $(l-1)$ and input to layer l, $\mathbf{W}^{(l)} \in \mathbb{R}^{n_i \times n_o}$ is a matrix of learnable weights of layer l, $\mathbf{b}^{(l)} \in \mathbb{R}^{n_o}$ is a vector of learnable biases of layer l, $\mathbf{y}^{(l)} \in \mathbb{R}^{n_o}$ is the output of layer l, $\mathbf{y}^{(0)}$ is the input to the model, $\mathbf{y}^{(L)}$ is the output of the final layer L and the model, and ϕ is a nonlinear activation function applied element-wise. The activation function used in this paper for feed-forward architectures is the Rectified Linear Unit (ReLU) as in Eq. (3) due to its advantages over other activation functions, such as computational simplicity and faster learning convergence (Glorot, Bordes, & Bengio, 2011).

$$\phi(z) = \max(0, z). \tag{3}$$

To provide a probabilistic interpretation of the model's output, the output layer L utilizes a softmax nonlinearity instead of the nonlinear function used in previous layers as in Eq. (4):

$$softmax(z_k) = \frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}} \tag{4}$$

where K is the number of output classes.

A popular variant of the feed-forward neural network architecture is the Convolutional Neural Network (ConvNet) (LeCun et al., 1990), which leverages three ideas: sparse interactions; parameter sharing; and equivariant representations. This can be achieved

by replacing the affine transformation in Eq. (1) with a convolution operation as in Eq. (5) and adding another layer called pooling, which aims to merge semantically similar features using a subsampling operation such as maximization.

$$\mathbf{h}^{(l)} = \mathbf{y}^{(l-1)} * \mathbf{W}^{(l)} + \mathbf{b}^{(l)} \tag{5}$$

where in this case, $\mathbf{W}^{(l)} \in \mathbb{R}^{m \times j \times k}$ is a tensor of m learnable filters, each of which is of height j and width k. Following recent work (He, Zhang, Ren, & Sun, 2016; Simonyan & Zisserman, 2015), subsampling is performed in this work by adjusting the stride in convolution layers rather than an explicit pooling layer.

The recurrent architecture extends the notion of a typical feed-forward architecture by adding inter-layer and self connections to units in the recurrent layer (Graves, 2008), which can be modeled using Eq. (6) in place of Eq. (1). This makes such type of architectures particularly suitable for tasks that involve sequential inputs such as speech.

$$\mathbf{h}_t^{(l)} = \mathbf{y}_t^{(l-1)}\mathbf{W}_y^{(l)} + \mathbf{s}_{t-1}^{(l)}\mathbf{W}_s^{(l)} + \mathbf{b}^{(l)} \tag{6}$$

where t denotes the time step, $\mathbf{h}_t^{(l)} \in \mathbb{R}^{n_o}$ is a vector of preactivations of layer l at time step t, $\mathbf{y}_t^{(l-1)} \in \mathbb{R}^{n_i}$ is the output of the previous layer $(l-1)$ at time step t and input to layer l at time step t, $\mathbf{W}_y^{(l)} \in \mathbb{R}^{n_i \times n_o}$ is a matrix of learnable weights of layer l, $\mathbf{s}_{t-1}^{(l)} \in \mathbb{R}^{n_o}$ is the state of layer l at the previous time step $(t-1)$, $\mathbf{W}_s^{(l)} \in \mathbb{R}^{n_o \times n_o}$ is a matrix of learnable weights of layer l, and $\mathbf{b}^{(l)} \in \mathbb{R}^{n_o}$ is a vector of learnable biases of layer l. For recurrent architectures, sigmoid functions such as the logistic function as in Eq. (7) and the hyperbolic tangent (tanh) function were used as the activation function instead of ReLUs, as ReLUs amplify the exploding gradient problem in recurrent architectures due to their unbounded nature.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

A popular variant of the recurrent architectures is the Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), which uses an explicit memory cell to better learn long-term dependencies. An LSTM cell can be modeled using Eqs. (8)–(12):

$$\mathbf{i}_t^{(l)} = \sigma\left(\mathbf{W}_{yi}\mathbf{y}_t^{(l-1)} + \mathbf{W}_{hi}\mathbf{h}_{t-1}^{(l)} + \mathbf{W}_{ci}\mathbf{c}_{t-1}^{(l)} + \mathbf{b}_i^{(l)}\right) \tag{8}$$

$$\mathbf{f}_t^{(l)} = \sigma\left(\mathbf{W}_{yf}\mathbf{y}_t^{(l-1)} + \mathbf{W}_{hf}\mathbf{h}_{t-1}^{(l)} + \mathbf{W}_{cf}\mathbf{c}_{t-1}^{(l)} + \mathbf{b}_f^{(l)}\right) \tag{9}$$

$$\mathbf{c}_t^{(l)} = \mathbf{f}_t\mathbf{c}_{t-1}^{(l)} + i_t \tanh\left(\mathbf{W}_{yc}\mathbf{y}_t^{(l-1)} + \mathbf{W}_{hc}\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_c^{(l)}\right) \tag{10}$$

$$\mathbf{o}_t^{(l)} = \sigma\left(\mathbf{W}_{yo}\mathbf{y}_t^{(l-1)} + \mathbf{W}_{ho}\mathbf{h}_{t-1}^{(l)} + \mathbf{W}_{co}c_t^{(l)} + \mathbf{b}_o^{(l)}\right) \tag{11}$$

$$\mathbf{h}_t^{(l)} = \mathbf{o}_t^{(l)} \tanh(\mathbf{c}_t^{(l)}) \tag{12}$$

where σ is the logistic sigmoid function in Eq. (7), and $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$ and $\mathbf{c}$ are the input gate, forget gate, output gate and cell activation vectors respectively, all of which are the same size as the vector $\mathbf{h}$. The weight matrices from the cell to gate vectors, e.g. $\mathbf{W}_{ci}$, are diagonals such that each element in each gate vector only receives input from the same element of the cell vector (Graves, Mohamed, & Hinton, 2013).

3.2. Learning

Learning is formulated as an optimization problem to minimize a cost function. The cost function used in this paper is the cross-entropy cost function in Eq. (13):

$$C = -\sum_{k=1}^{K} \hat{\mathbf{y}}_k \log(\mathbf{y}_k^{(L)}) \tag{13}$$

where $\hat{\mathbf{y}} \in \{0, 1\}^K$ is a one-of-K encoded label and $\mathbf{y}^{(L)}$ is the output of the model.

The gradients are computed by differentiating the cost function with respect to the model parameters using a mini-batch of data examples sampled from the training data and backpropagated to prior layers using the backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986). Training recurrent architectures requires modification to the backpropagation algorithm to compute the gradients with respect to the parameters and states of the model, which is known as the backpropagation through time algorithm (Werbos, 1988).

Gradient descent or one of its variants is used to update the parameters of the model using the gradients computed. A per-parameter adaptive variant of gradient descent called RMSProp (Dauphin, de Vries, & Bengio, 2015; Tieleman & Hinton, 2012) was used in this paper, which uses gradient information to adjust the learning rate as in Eqs. (14) and (15):

$$r := \eta r + (1 - \eta)(\partial C/\partial w)^2 \tag{14}$$

$$w := w - \alpha \frac{\partial C/\partial w}{\sqrt{r} + \epsilon} \tag{15}$$

where r is a leaky moving average of the squared gradient and η and α are hyperparameters denoting the decay rate and the learning rate respectively.

3.3. Regularization

Deep architectures are prone to overfitting, which makes regularization an essential ingredient in their success. In this paper, three regularization techniques were used: l_2 weight decay, which penalizes the l_2 norm of the weights of the model; dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), which stochastically omits units in the model during training preventing co-adaptation of units; and Batch Normalization (BatchNorm) (Ioffe & Szegedy, 2015), which aims to reduce the internal covariate shift in deep architectures by normalizing the means and standard deviations of layer preactivations as in Eq. (16):

$$\text{BatchNorm}(\mathbf{z}^{(l)}; \gamma, \beta) = \beta + \gamma \frac{\mathbf{z}^{(l)} - \widehat{\mathbb{E}}(\mathbf{z}^{(l)})}{\sqrt{\widehat{\text{Var}}(\mathbf{z}^{(l)}) + \epsilon}} \tag{16}$$

where $\gamma \in \mathbb{R}^{n_o}$, $\beta \in \mathbb{R}^{n_o}$ are model parameters that determine the mean and standard deviation of the layer preactivations respectively and $\widehat{\mathbb{E}}$ and $\widehat{\text{Var}}$ are estimates of the sample mean and sample variance of the normalized preactivations respectively.

Unlike l_2 weight decay, which is employed by simply modifying the cost function, dropout and BatchNorm require modifying the architecture of the model in that BatchNorm can be treated as an additional layer added before the nonlinearity layer, while dropout is applied after the nonlinearity layer.

4. Proposed speech emotion recognition system

Fig. 1 is a sketch of the proposed SER system which follows a frame-based processing formulation that utilizes Fourier-transform based filter bank speech spectrograms and a deep multi-layered neural network to predict emotion class probabilities for each frame in the input utterance.

Let $X \in \mathbb{R}^{N \times T}$ be a speech utterance or speech stream sliced into a time-series sequence of T frames, each of which is a $\mathbb{R}^N$ vector of audio features. The aim is to rely on minimal speech processing and thus each frame is represented by Fourier-transform based log Mel-scaled N filter banks. The goal of the model is to predict $p(y_t|x)$, where $x \in X$ is a number of concatenated frames, $x_{t-l} \| \cdots \| x_t \| \cdots \| x_{t+r}$, where x_t is the target frame, l is the number

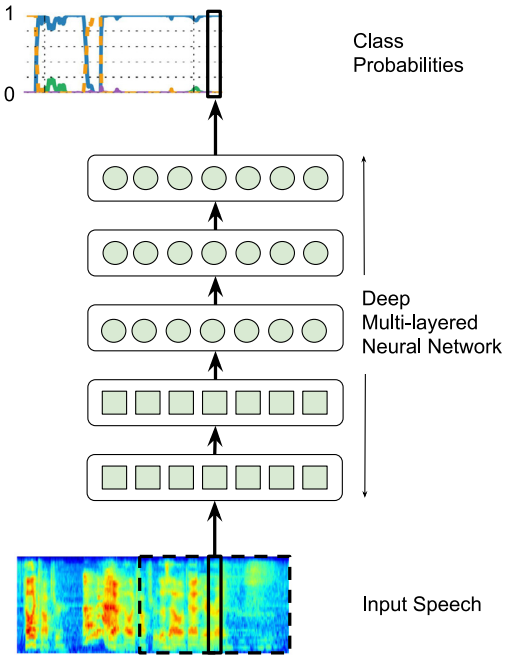

Fig. 1. Overview of the proposed SER system. A deep multi-layered neural network composed of several fully-connected, convolutional or recurrent layers ingests a target frame (solid line) concatenated with a number of context frames (dotted line) to predict the posterior class probabilities corresponding to the target frame.

of past context frames and r is the number of future frames, and y_t is the predicted output for frame x_t, which is either one emotion class or *silence*. *Silence* was added to the output classes since silence as well as unvoiced speech were not removed from the input speech utterance as it has been shown that silence and other disfluencies are effective cues in emotion recognition (Tian, Lai, & Moore, 2015).

The proposed model is a deep multi-layered neural network. We experiment with several neural network architectures as presented in Section 6. It is important to note that the model is able to deal with utterances of variable length, independent of the choice of architecture since the model predicts $p(y_t|x), \forall t \in \{1, \ldots, T\}$: this only requires the target frame x_t and past l and future r context frames, which are fixed prior. Since emotions manifest in speech in a slow manner, one may not necessarily predict the class of every single frame in an utterance or speech stream but may rely on predicting the class of a frame sampled every few frames, depending on application requirements. The output of the model may be aggregated over the entire utterance to perform utterance-level classification if desired.

5. Experimental setup

In this section, we introduce the data and its preprocessing, the computational setup and training recipe used in this paper.

5.1. Data and preprocessing

The popular Interactive Emotional Dyadic Motion Capture (IEMOCAP) database (Busso et al., 2008) was used. It comprises 12 h of audio-visual recordings divided into five sessions. Each session is composed of two actors, a male and a female, performing emotional scripts as well as improvised scenarios. In total, the database comprises 10 039 utterances with an average duration of 4.5 s. Utterances were labeled by three annotators using categorical labels. The database predominantly focused on five emotions: *anger*, *happiness*, *sadness*, *neutral* and *frustration*; however, annotators were not limited to these emotions during annotation. Ground truth labels were obtained by majority voting, where 74.6% of the utterances were agreed upon by at least two annotators. Utterances that were labeled differently by all three annotators were discarded in this study. To be consistent with other studies on the same database (Kim, Lee, & Provost, 2013; Mariooryad & Busso, 2013; Shah, Chakrabarti, & Spanias, 2014), utterances that bore only the following four emotions were included: *anger*, *happiness*, *sadness* and *neutral*, with *excitement* considered as *happiness*.

An eight-fold Leave-One-Speaker-Out (LOSO) cross-validation scheme (Schuller, Vlasenko et al., 2009) was employed in all experiments using the eight speakers in the first four sessions. Both speakers in the fifth session were used to cross-validate the hyperparameters of the models and to apply early stopping during training and therefore were not included in the cross-validation folds so as to not bias the results (Refaeilzadeh, Tang, & Liu, 2009).

Audio was analyzed using a 25 ms Hamming window with a stride of 10 ms. Log Fourier-transform based filter banks with 40 coefficients distributed on a Mel scale were extracted from each frame. The mean and variance were normalized per coefficient for each fold using the mean and variance computed using the training subset only. No speaker dependent operations were performed.

Since the data was labeled at an utterance-level, all frames in an utterance inherited the utterance label. A voice activity detector was then used to label silent frames and *silence* was added as an additional class to the four previously mentioned emotion classes; i.e. a frame has either the same label as its parent utterance or the *silence* label. The underlying assumption here is that frames in an utterance convey the same emotion as the parent utterance, which concurs with the same assumption made when a categorical label was assigned to the entire utterance; nevertheless, this assumption is eased by labeling silent and unvoiced frames as *silence*.

5.2. Computational setup

Due to the large number of experiments carried out in this paper, several computational resources were exploited at different stages. Some experiments were carried out on a cluster of CPUs, while others were carried out using Graphics Processing Units (GPUs) to accelerate the training process. The Kaldi toolkit (Povey et al., 2011) was used for speech processing and analysis. The neural networks and training algorithms were implemented in MATLAB and C. Training time varied significantly between different models with an average duration of 2 days; however the largest model took 14 days to train on a GPU. The code is available at http://github.com/haythamfayek/SER.

5.3. Training recipe

The parameters of the neural networks were initialized from a Gaussian distribution with zero mean and $\sqrt{2/n_i}$ standard deviation, where n_i is the number of inputs to the layer, as recommended by He et al. (2016). Mini-batch stochastic gradient descent with a batch size of 256 and RMSProp per-parameter adaptive learning rate were used to optimize the parameters with respect to a cross-entropy cost function. The base learning rate was set to $\alpha = 1 \times 10^{-2}$ and annealed by a factor of 10 when the error plateaus. The decay rate was set to $\eta = 0.99$. Fully-connected layers were regularized using dropout with a retention probability $P = 0.5$. Convolutional layers were regularized using l_2 weight decay with penalty $\lambda = 1 \times 10^{-3}$. LSTM layers were regularized using dropout with a retention probability $P = 0.5$ and the gradients were clipped to lie in range $[-5, 5]$. BatchNorm was used after every fully-connected or convolutional layer. The validation set was used to perform early-stopping during training, such that training halts when the learning rate reaches $\alpha = 1 \times 10^{-8}$; and the model with the best accuracy on the validation set during training was selected. These hyperparameters were chosen based on experimental trials using the validation set.

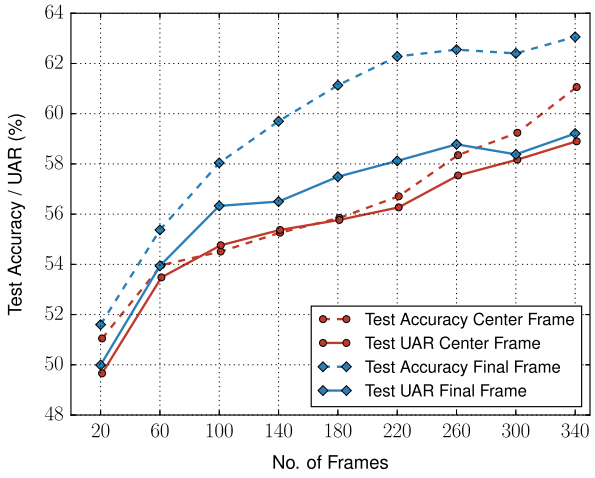

Fig. 2. Test accuracy and UAR of a DNN with various number of context frames.

6. Experiments and results

As is standard practice in the field of automatic emotion recognition, results are reported using Accuracy and Unweighted Average Recall (UAR) to reflect imbalanced classes (Schuller, Steidl, & Batliner, 2009). Both metrics are reported for the average of the eight-fold LOSO cross-validation scheme. The output of the model during evaluation was the class with the highest posterior probability as in Eq. (17):

$$\ell_t = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, p(y_t|x) \tag{17}$$

where $K = 5$ is the number of output classes and ℓ_t is the predicted label of frame t.

6.1. Feed-forward architectures

The first experiment was conducted to investigate the number of context frames required for SER. We hypothesized that unlike ASR, SER does not rely on future context but does require a large number of past context frames. Therefore, the models were trained in two configurations: (1) the first configuration was to predict $p(y_t|x)$, where x is $x_{t-c}\| \cdots \|x_t\| \cdots \|x_{t+c}$ for various values of c, i.e. predict the class label of the center frame; (2) the second configuration was to predict $p(y_t|x)$, where x is $x_{t-c}\| \cdots \|x_t$ for various values of c, i.e. predict the class label of the final frame. The model used in this experiment was a DNN with 5 hidden layers; each was composed of 1024 fully-connected units with BatchNorm, ReLU and dropout layers interspersed in-between and a softmax output layer. This architecture was selected based on the best UAR on the validation set, which was excluded from the cross-validation scheme. Fig. 2 is a plot of the test accuracy and UAR of the model in both configurations for various numbers of context frames.

Two observations are immediately evident from the results in Fig. 2: (1) the performance of the system is directly proportional to the number of context frames until it starts to plateau after 220 frames and (2) the future context has a minor contribution to the performance of the system as hypothesized. Since a large number of context frames lead to an increase in the dimensionality of the input and may increase overfitting (as shown in Fig. 2), a good trade-off between the number of context frames and the performance of the system would lie between 2–3 s of speech.

ConvNets are able to learn features that are insensitive to small variations in the input speech which can help in disentangling speaker-dependent variations as well as other sources of distortion such as noise. Moreover, ConvNets are able to deal with high-dimensional input, which in this case is due to the large number of context frames required. In this experiment, we present an in-depth exploration of various ConvNet architectures and demonstrate the effect of the number of convolutional and fully-connected layers, number of filters, size of filters and type of convolution (spatial vs temporal) on the performance of the system. Table 1 lists various ConvNet architectures and their respective test accuracy and UAR. All experiments were conducted using 259 past context frames and no future context frames, which correspond to approximately 2.6 s of speech; i.e. the input dimensionality is 40 filter banks $\times$ 260 frames.

From the results listed in the first segment of Table 1, the benefit of network depth can be observed. The best results were obtained using 2 convolutional layers followed by 2–3 fully-connected layers. Adding more layers to the network did not yield any performance gain and resulted in overfitting. The results in the second segment of Table 1 demonstrate the effect of the filter size on the performance of the model. It can be seen that similar to other speech applications, SER requires a relatively large filter with an optimal size of 10×10. Temporal convolution did perform slightly worse than spatial convolution as demonstrated in the final segment of Table 1.

6.2. Recurrent architectures

In the next set of experiments, we investigate how LSTM-RNNs can be employed for the proposed SER system. LSTM-RNNs can be trained in several ways, such as Sequence-to-Sequence, where a model is trained to ingest a sequence of frames and output a sequence of class labels; or Sequence-to-One, where a model is trained to ingest a sequence of frames and output a class label. Sequence-to-Sequence training may seem to be a better fit to the proposed system; however, preliminary experiments demonstrated the superiority of Sequence-to-One training, as Sequence-to-Sequence training failed to converge in most cases or had poor performance otherwise. Therefore, Sequence-to-One training was used in our experiments: the model was trained to ingest a sequence of frames, frame-by-frame, and predict a class label for the final frame, $p(y_t|x)$, where x is $x_{t-c}\| \cdots \|x_t$ and c is the number of context frames (sequence length).

LSTM-RNNs can handle sequences of arbitrary lengths. However, the effect of the sequence length, on which the model was trained, on the ability of the model to handle arbitrary sequence lengths is not well-studied. Hence, several models were trained using various training sequence lengths {20, 60, 100, 200}, where LSTM-RNN-c denotes the training sequence length c on which the model was trained; it was then evaluated on a number of test sequence lengths {20, 60, 100, 200, 260, 300}. An extra model was trained on sequence length c chosen randomly at each iteration such that $c \in \{20, 60, 100, 200\}$, denoted LSTM-RNN-R. The model used in this experiment was a 2-layered LSTM-RNN with 256 units in each hidden layer and dropout interspersed in-between and a softmax output layer. This architecture was selected based on the best UAR on the validation set, which was excluded from the cross-validation scheme. Figs. 3 and 4 depict the accuracy and UAR respectively of the LSTM-RNNs trained and evaluated on various sequence lengths.

Results in Figs. 3 and 4 demonstrate a similar trend in that models trained on short sequences did not perform as well on long sequences and vice versa. In addition, noticeable gains in performance could be achieved by increasing the number of context frames (test sequence length). The best performance at each test sequence length was obtained by the model trained on the same sequence length and the performance degraded gradually as the test sequence length deviated from the training sequence

Table 1
Test accuracy and UAR of various ConvNet architectures. FC(n_o) denotes a fully-connected layer of n_o units followed by BatchNorm, ReLUs and dropout. Conv($m \times j \times k$) and Conv1D($m \times j \times k$) denote a spatial convolutional layer and a temporal convolutional layer respectively of m filters each of size $j \times k$ with a stride of 2 followed by BatchNorm and ReLUs. Softmax(n_o) denotes a softmax output layer of n_o units followed by a softmax operation.

Architecture	Test Accuracy (%)	Test UAR (%)
Conv(32 × 4 × 4) — FC(1024) — Softmax(1024)	62.27	58.30
Conv(32 × 4 × 4) — FC(1024)× 2 — Softmax(1024)	62.78	58.87
Conv(32 × 4 × 4) — Conv(64 × 3 × 3) — FC(1024) — Softmax(1024)	62.58	58.71
Conv(32 × 4 × 4) — Conv(64 × 3 × 3) — FC(1024)× 2 — Softmax(1024)	63.16	58.56
Conv(16 × 4 × 4) — Conv(32 × 3 × 3) — FC(716)× 2 — Softmax(716)	63.34	59.30
Conv(32 × 4 × 4) — Conv(64 × 3 × 3) — FC(1024)× 3 — Softmax(1024)	63.82	58.92
Conv(16 × 4 × 4) — Conv(32 × 3 × 3) — FC(716)× 3 — Softmax(716)	62.90	58.17
Conv(16 × 6 × 6) — Conv(32 × 6 × 6) — FC(716)× 2 — Softmax(716)	63.51	59.50
Conv(16 × 10 × 10) — Conv(32 × 10 × 10) — FC(716)× 2 — Softmax(716)	**64.78**	**60.89**
Conv(16 × 14 × 14) — Conv(32 × 14 × 14) — FC(716)× 2 — Softmax(716)	62.84	58.30
Conv(16 × 10 × 18) — Conv(32 × 10 × 18) — FC(716)× 2 — Softmax(716)	63.07	58.79
Conv1D(64 × 40 × 4) — FC(1024)× 2 — Softmax(1024)	62.41	58.38
Conv1D(64 × 40 × 8) — FC(1024)× 2 — Softmax(1024)	62.98	59.07
Conv1D(64 × 40 × 16) — FC(1024)× 2 — Softmax(1024)	62.91	58.49

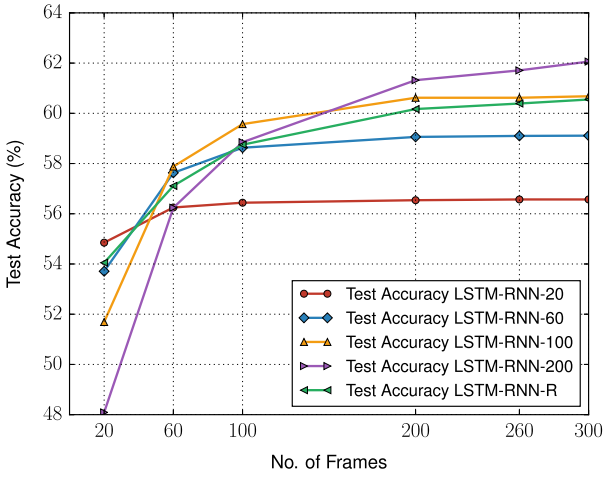

Fig. 3. Test accuracy of an LSTM-RNN with various number of context frames. LSTM-RNN-c denotes the sequence length which the model was trained on. The no. of frames denotes the sequence length which the model was evaluated on.

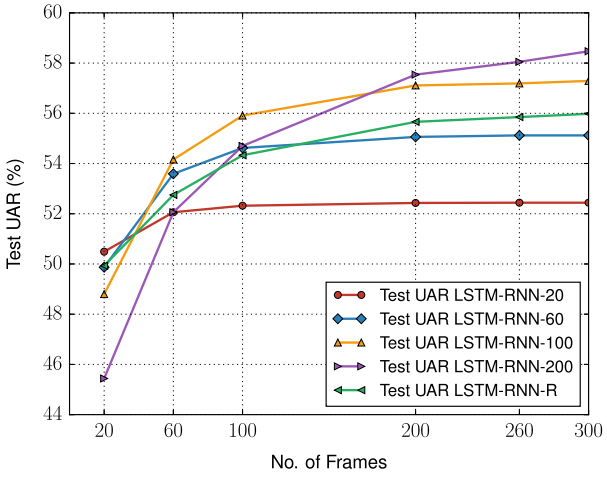

Fig. 4. Test UAR of an LSTM-RNN with various number of context frames. LSTM-RNN-c denotes the sequence length which the model was trained on. The no. of frames denotes the sequence length which the model was evaluated on.

length. Moreover, by varying the sequence length when training LSTM-RNN-R, the model did learn to perform well on various test sequence lengths. On average, LSTM-RNN-100 yielded the best UAR averaged over all test sequence lengths, followed by LSTM-RNN-R.

7. Discussion

The number of context frames was a major contributing factor in the performance of the system. All architectures benefited from using a large number of context frames. Key to harnessing the information in these frames without overfitting was using recent advances in regularization methods such as dropout and BatchNorm, otherwise the accompanied increase in dimensionality of the input data would have been problematic.

Table 2 lists the best model from each architecture and their respective accuracy and UAR trained and evaluated under the same data subsets and experiment conditions. As stated earlier, SER can be regarded as a static or dynamic classification problem, which makes it an excellent test bed for conducting a comparison between these architectures. In this case, the ConvNet and the DNN can be regarded in this formulation as static classifiers that process a number of concatenated frames jointly to predict a class label, whereas the LSTM-RNN can be regarded in this formulation as a dynamic classifier that processes a sequence of frames, frame-by-frame, to predict a class label. The results in Table 2 suggest that the static component in speech is more discriminative for SER than the dynamic component. This is likely due to the dominant presence of the linguistic aspect in the dynamic component of speech, which hinders the recognition of paralinguistic components such as emotions. We speculate that for this reason and due to the ConvNet's ability to learn discriminative features invariant to small variations, the ConvNet yielded the best accuracy and UAR followed by the DNN then the LSTM-RNN.

Trends in Table 1 and the best ConvNet architecture reported in this work are similar to those reported in other speech applications and particularly in ASR (Abdel-Hamid et al., 2014). This may pave the way for a single architecture to be used in a multi-task setting for speech processing (Fayek et al., 2016b).

Fig. 5 illustrates the output of the proposed SER system using a ConvNet for a number of selected utterances, all of which were from the test subset of the data. Qualitative assessment of the system's output indicates that the network has learned to model the intra-utterance emotion dynamics with high confidence and is able to transition smoothly from one class to another, capturing brief pauses and mixed emotions as shown in Fig. 5. It is particularly interesting to note the system's output in Fig. 5(e), which has classified the first half of the utterance as *neutral*

Table 2
Test accuracy and UAR for various network architectures.

Model	Test Accuracy (%)	Test UAR (%)
FC(1024)× 5 — Softmax(1024)	62.55	58.78
Conv(16 × 10 × 10) — Conv(32 × 10 × 10) — FC(716)× 2 — Softmax(716)	**64.78**	**60.89**
LSTM-RNN(256)× 2 - Softmax(256)	61.71	58.05

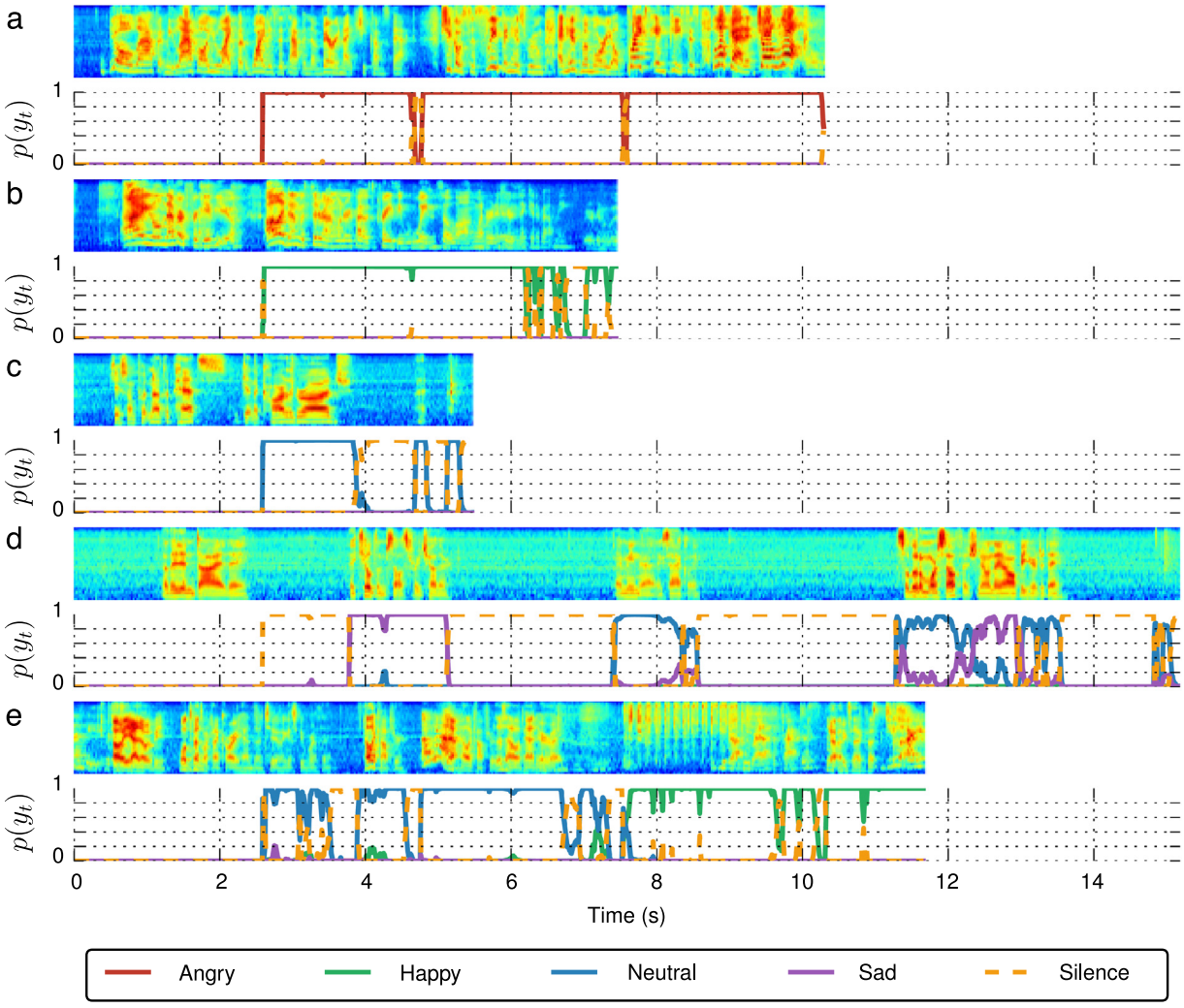

Fig. 5. Input speech utterances (top) and corresponding aligned output (below) of the proposed SER system for a number of utterances from the test subset. The output is the posterior class probabilities $p(y_t)$ denoting the confidence of the model. Transcripts: (a): Oh, laugh at me all you like but why does this happen every night she comes back? She goes to sleep in his room and his memorial breaks in pieces. Look at it, Joe look.: Angry. (b): I will never forgive you. All I'd done was sit around wondering if I was crazy waiting so long, wondering if you were thinking about me.: Happy. (c): OKay. So I am putting out the pets, getting the car our the garage.: Neutral. (d): They didn't die. They killed themselves for each other. I mean that, exactly. Just a little more selfish and they would all be here today.: Sad. (e): Oh yeah, that would be. Well, depends on what type of car you had, though too. I guess it would be worth it. helicopter. Yeah, helicopter. There is a helipad there, right? Yeah, exactly.: Happy.

and the second half of the utterance as *happy* conforming to our manual inspection, whereas the database annotators assigned the *happy* label to the entire utterance. In some cases, as in Fig. 5(d,e), the system did not predict the correct output class with high confidence across the entire utterance, which may suggest that a smoothing function over the network may offer additional improvements in some cases; however, this is beyond the scope of this work.

To compare the results reported in this paper with prior work in the literature, which mostly relies on utterance-based classification, the posterior class probabilities computed for each frame in an utterance were averaged across all frames in that utterance and an utterance-based label was selected based on the maximum average class probabilities, ignoring the *silence* label, as per Eq. (18):

$$\ell_u = \underset{k=1,\ldots,K}{\operatorname{argmax}} \frac{\sum_{t=1}^{T} p(y_t|x)}{T} \tag{18}$$

where $K = 4$ is the number of output classes, i.e. ignoring the *silence* class, and ℓ_u is the utterance-level predicted label.

Table 3 shows the SER results reported in prior work on the IEMOCAP database. Note that differences in data subsets used and other experiment conditions should be taken into consideration when comparing these results against each other, c.f. Han et al. (2014), Lee, Mower, Busso, Lee, and Narayanan (2011), Mariooryad and Busso (2013) and Shah et al. (2014) for more details. As can be seen from Table 3, the proposed SER system outperforms all other speaker-independent methods. In addition, the proposed SER system offers other advantages such as real-time output since

Table 3
SER results reported in prior work on the IEMOCAP database. Note that differences in data subsets used and other experiment conditions should be taken into consideration when comparing the following results against each other, c.f. references for more details.

Method	Test Accuracy (%)	Test UAR (%)	Notes
DNN + ELM (Han et al., 2014)	54.3	48.2	–
SVM (Mariooryad & Busso, 2013)	53.99	50.64	Better performance was reported by incorporating other modalities.
Replicated Softmax Models + SVM (Shah et al., 2014)	–	57.39	Better performance was reported by incorporating other modalities.
Hierarchical Binary Decision Tree (Lee et al., 2011)	–	58.46	Speaker-Dependent Normalization
Proposed SER system (Utterance-Based)	**57.74**	**58.28**	
Proposed SER system (Frame-Based)	**64.78**	**60.89**	

it does not depend on future context. Moreover, the system is able to deal with utterances of arbitrary length with no degradation in performance. Furthermore, the system can handle utterances that contain more than an emotion class, as demonstrated in Fig. 5(e), which would not be possible in an utterance-based formulation.

8. Conclusion

Various deep learning architectures were explored on a Speech Emotion Recognition (SER) task. Experiments conducted illuminate how feed-forward and recurrent neural network architectures and their variants could be employed for paralinguistic speech recognition, particularly emotion recognition. Convolutional Neural Networks (ConvNets) demonstrated better discriminative performance compared to other architectures. As a result of our exploration, the proposed SER system which relies on minimal speech processing and end-to-end deep learning, in a frame-based formulation, yields state-of-the-art results on the IEMOCAP database for speaker-independent SER.

Future work can be pursued in several directions. The proposed SER system can be integrated with automatic speech recognition, employing joint knowledge of the linguistic and paralinguistic components of speech to achieve a unified model for speech processing. More generally, observations made in this work as a result of exploring various architectures could be beneficial for devising further architectural innovations in deep learning that can exploit advantages of current models and address their limitations.

Acknowledgments

This research was funded by the Vice-Chancellor's Ph.D. Scholarship (VCPS) from RMIT University. We gratefully acknowledge the support of NVIDIA Corporation with the donation of one of the Tesla K40 GPUs used in this research.

References

Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*, 1533–1545.

Arias, J. P., Busso, C., & Yoma, N. B. (2013). Energy and F0 contour modeling with functional data analysis for emotional speech detection. In *Interspeech* (pp. 2871–2875).

Ayadi, M. E., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, *44*, 572–587.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., et al. (2008). IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, *42*, 335–359. http://dx.doi.org/10.1007/s10579-008-9076-6.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., et al. (2001). Emotion recognition in human–computer interaction. *IEEE Signal Processing Magazine*, *18*, 32–80. http://dx.doi.org/10.1109/79.911197.

Dauphin, Y., de Vries, H., & Bengio, Y. (2015). Equilibrated adaptive learning rates for non-convex optimization. In *Advances in neural information processing systems* (pp. 1504–1512).

Eyben, F., Huber, B., Marchi, E., Schuller, D., & Schuller, B. (2015). Real-time robust recognition of speakers' emotions and characteristics on mobile platforms. In *2015 international conference on affective computing and intelligent interaction (ACII)* (pp. 778–780). http://dx.doi.org/10.1109/ACII.2015.7344658.

Eyben, F., Scherer, K., Schuller, B., Sundberg, J., Andre, E., Busso, C., et al. (2015). The Geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 190–202. http://dx.doi.org/10.1109/TAFFC.2015.2457417.

Fayek, H. M., Lech, M., & Cavedon, L. (2015). Towards real-time speech emotion recognition using deep neural networks. In *2015 9th international conference on signal processing and communication systems (ICSPCS)* (pp. 1–5). http://dx.doi.org/10.1109/ICSPCS.2015.7391796.

Fayek, H. M., Lech, M., & Cavedon, L. (2016a). Modeling subjectiveness in emotion recognition with deep neural networks: Ensembles vs soft labels. In *2016 international joint conference on neural networks (IJCNN)* (pp. 566–570). http://dx.doi.org/10.1109/IJCNN.2016.7727250.

Fayek, H. M., Lech, M., & Cavedon, L. (2016b). On the correlation and transferability of features between automatic speech recognition and speech emotion recognition. In *Interspeech 2016* (pp. 3618–3622). http://dx.doi.org/10.21437/Interspeech.2016-868.

Fernandez, R. (2004). *A computational model for the automatic recognition of affect in speech*. (Ph.D. thesis), School of Architecture and Planning, Massachusetts Institute of Technology.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *JMLR W&CP: Proceedings of the fourteenth international conference on artificial intelligence and statistics*, AISTATS 2011.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Graves, A. (2008). *Supervised sequence labelling with recurrent neural networks*. (Ph.D. thesis), Technische Universitat Munchen.

Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645–6649). http://dx.doi.org/10.1109/ICASSP.2013.6638947.

Han, K., Yu, D., & Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *Interspeech 2014*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*, 82–97. http://dx.doi.org/10.1109/MSP.2012.2205597.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning* (pp. 448–456).

Kim, Y., Lee, H., & Provost, E. M. (2013). Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 3687–3691). http://dx.doi.org/10.1109/ICASSP.2013.6638346.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Neural information processing systems* (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. Insight.

LeCun, Y., Boser, B., Denker, J.S., Howard, R.E., Habbard, W., & Jackel, L.D. et al. (1990). Advances in neural information processing systems 2. chapter Handwritten digit recognition with a back-propagation network. (pp. 396–404).

Lee, C.-C., Mower, E., Busso, C., Lee, S., & Narayanan, S. (2011). Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, *53*, 1162–1171.

Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., & Gonzalez, I. et al. (2013). Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition. In *2013 humaine association conference on affective computing and intelligent interaction (ACII)* (pp. 312–317).

Mao, Q., Dong, M., Huang, Z., & Zhan, Y. (2014). Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE Transactions on Multimedia*, *16*, 2203–2213.

Mariooryad, S., & Busso, C. (2013). Exploring cross-modality affective reactions for audiovisual emotion recognition. *IEEE Transactions on Affective Computing*, *4*, 183–196. http://dx.doi.org/10.1109/T-AFFC.2013.11.

Mohamed, A., Dahl, G., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*, 14–22. http://dx.doi.org/10.1109/TASL.2011.2109382.

Petta, P., Pelachaud, C., & Cowie, R. (2011). Emotion-oriented systems. In *The humaine handbook*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems* (pp. 532–538). Springer.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. http://dx.doi.org/10.1016/j.neunet.2014.09.003.

Schuller, B., Steidl, S., & Batliner, A. (2009). The interspeech 2009 emotion challenge. In *Interspeech, vol. 2009* (pp. 312–315).

Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C. A., et al. (2010). The interspeech 2010 paralinguistic challenge. In *Interspeech* (pp. 2794–2797).

Schuller, B., Vlasenko, B., Eyben, F., Rigoll, G., & Wendemuth, A. (2009). Acoustic emotion recognition: A benchmark comparison of performances. In *IEEE workshop on automatic speech recognition understanding, 2009* (pp. 552–557).

Shah, M., Chakrabarti, C., & Spanias, A. (2014). A multi-modal approach to emotion recognition using undirected topic models. In *IEEE international symposium on circuits and systems (ISCAS)* (pp. 754–757).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations (ICLR)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stuhlsatz, A., Meyer, C., Eyben, F., Zielke, T., Meier, G., & Schuller, B. (2011). Deep neural networks for acoustic emotion recognition: Raising the benchmarks. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5688–5691).

Tahon, M., & Devillers, L. (2016). Towards a small set of robust acoustic features for emotion recognition: Challenges. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*, 16–28.

Tian, L., Lai, C., & Moore, J. (2015). Recognising emotions in dialogues with disfluencies and non-verbal vocalisations. In R. Lickley (Ed.), *The 7th workshop on disfluency in spontaneous speech*.

Tian, L., Moore, J., & Lai, C. (2015). Emotion recognition in spontaneous and acted dialogues. In *2015 International conference on affective computing and intelligent interaction (ACII)* (pp. 698–704).

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*.

Ververidis, D., & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech Communication*, *48*, 1162–1181.

Vlasenko, B., Schuller, B., Wendemuth, A., & Rigoll, G. (2007). Frame vs. turn-level: emotion recognition from speech considering static and dynamic processing. In *Affective computing and intelligent interaction* (pp. 139–147). Springer.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, *1*, 339–356. http://dx.doi.org/10.1016/0893-6080(88)90007-X.