CLASSIFICATION VS. REGRESSION IN SUPERVISED LEARNING FOR SINGLE CHANNEL SPEAKER COUNT ESTIMATION

Fabian-Robert Stöter, Soumitro Chakrabarty, Bernd Edler, Emanuël A. P. Habets

International Audio Laboratories Erlangen, Germany*

ABSTRACT

The task of estimating the maximum number of concurrent speakers from single channel mixtures is important for various audio-based applications, such as blind source separation, speaker diarisation, audio surveillance or auditory scene classification. Building upon powerful machine learning methodology, we develop a Deep Neural Network (DNN) that estimates a speaker count. While DNNs efficiently map input representations to output targets, it remains unclear how to best handle the network output to infer integer source count estimates, as a discrete count estimate can either be tackled as a regression or a classification problem. In this paper, we investigate this important design decision and also address complementary parameter choices such as the input representation. We evaluate a state-of-the-art DNN audio model based on a Bi-directional Long Short-Term Memory network architecture for speaker count estimations. Through experimental evaluations aimed at identifying the best overall strategy for the task and show results for five seconds speech segments in mixtures of up to ten speakers.

Index Terms— speaker count estimation, number of concurrent speakers, overlapped speech, cocktail-party

1. INTRODUCTION

In a "cocktail-party" scenario with many concurrent speakers, different applications may be envisioned such as localization, crowd monitoring, surveillance, speech recognition or speaker separation. In this scenario, a typical assumption is that the number of concurrent speakers is known, which turns out to be of paramount importance for the effectiveness of subsequent processings. Unfortunately, in real world applications, information about the actual number of concurrent speakers is often not available. Surprisingly, very few methods have been proposed to address the task of counting the number of speakers.

Estimating the maximum number of concurrent speakers is closely related to the more difficult problem of *identifying* them, which is the topic of speaker diarisation (who speaks when) [1]. We call a system that identifies speakers first, "counting by detection". These systems often use segments where only one speaker is active to discriminate the speakers. Then comparisons of found segments are made to discriminate and temporally locate the speakers within a given recording. When sources are fully overlapped as in real "cocktail party" environments, such a segmentation is hardly feasible. And when a speaker overlap is as prevalent as in a "cocktail-party" scenario, developing an algorithm to detect speakers is challenging. In this study we therefore attempt to directly estimate a speaker count instead of counting them after identification. We refer to this strategy as "direct count estimation".

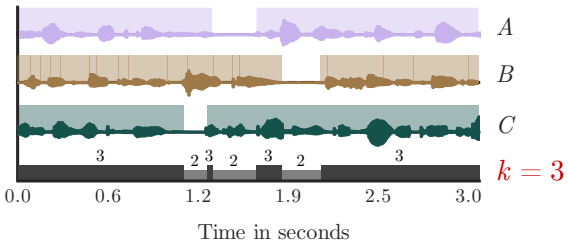


Fig. 1: Illustration of our application scenario of three concurrent speakers and their respective speech activity. Bottom plot shows the number of concurrently active speakers and its maximum k which is our targeted output.

In multichannel signal processing, count estimation usually is achieved by estimating directions of arrival (DoA) and clustering them [2, 3]. The first single channel method, based on thresholding amplitude modulation patterns, was proposed in [4]. In [5], the authors propose an energy feature based on temporally averaged mel-scale filter outputs. In a more recent work [6], the number of speakers is estimated by applying hierarchical clustering on fixed-length audio segments. The main weakness of this method is to rely on the assumption that there are segments where only one speaker is active. In another vein, Andrei et.al. [7] proposed an algorithm which correlates single frames of multi-speaker mixtures with a set of single-speaker utterances. Motivated by the recent and impressive successes of deep learning approaches in various audio-related tasks [8, 9], we focus on developing such a method for direct count estimation. In computer vision, (object) count estimation using DNNs has recently achieved state-of-the-art performance [10–18]. Two main paradigms may be found in the literature for this purpose, namely, *regression* and *classification*.

In this work we want to build upon these findings to achieve direct count estimation of speakers. Our main contributions are: i) to formulate the speaker count estimation problem as either a classification or a regression task, and ii) to propose a neural network architecture based on a state-of-the-art BLSTM network, to infer the number of speakers from short audio segments of 5s. Finally, we present experimental results for the different problem formulations as well as input feature representations to identify the best strategy for this task. For the sake of reproducibility, pre-trained network and the test dataset are made available for download on the accompanying website.[1]

*A joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits (IIS).

[1] https://www.audiolabs-erlangen.de/resources/2017-CountNet

2. PROBLEM FORMULATION

We consider the task of estimating the maximum number of concurrent speakers, $k \in \mathbb{Z}_0^+$, in a single channel audio mixture $\mathbf{x}$. This is achieved by applying a mapping from $\mathbf{x}$ to k. Let $\mathbf{x}$ be a time domain signal of N samples, representing a linear mixture of L unique single speaker speech signals $\mathbf{s}_l$. Naturally, not all speakers $l = 1, \ldots, L$ are active at every time instance. We therefore, for each sample n, introduce a latent binary *speech activity* variable $v_{nl} \in \{0, 1\}$. Then, our task is to estimate

$$k = \max_n \left(\sum_{l=1}^{L} v_{nl} \right). \tag{1}$$

It can be seen that our proposed task of estimating $k \leq L$ is more closely related to source separation whereas the estimation of L itself is more useful for tasks where speakers do not overlap. We assume that no additional prior information except the maximum number of concurrent speakers, $k_{\max}$, is available, representing an upper limit for estimation. In Figure 1, we illustrate our setup in a "cocktail-party" scenario featuring $L = k = 3$ speakers. For the DNN system proposed in this paper, we use a non-negative time-frequency (TF) input representation $\mathbf{X} \in \mathbb{R}_+^{D \times F}$ instead of $\mathbf{x}$, where D and F denote the total number of time frames and frequency sub-bands, respectively.

2.1. Estimation in a Deep Learning Framework

In this study, we choose a deep neural network (DNN) as the mapping function f_θ from the input $\mathbf{X}$ to the output y, given by $y = f_\theta(\mathbf{X})$, where the optimal parameters (weights) θ are learned via supervised training. The output of the DNN is not necessarily the direct source count k, therefore we introduce $q(\cdot)$ as a *decision function*, such that

$$\hat{k} = q\left(f_\theta(\boldsymbol{X})\right). \tag{2}$$

The DNN is trained in a supervised manner using a training database of $\{\mathbf{X}, k\}$ examples. In this work, we want to investigate three different choices for the output distributions of the DNN, as well as the corresponding decision functions $q(\cdot)$.

Classification: Here the output distribution is directly taken as *discrete*, discarding any meaning concerning the ordering of the different possible values. Given some particular input $\mathbf{X}$, the network generates the posterior output probability for $(k_{\max} + 1)$ classes (including $k = 0$) using the softmax activation function, and a maximum a posteriori (MAP) decision function is chosen that simply picks the most likely class $q = \arg\max(\cdot)$. Notwithstanding its conceptual simplicity, classification has two drawbacks. First, the intuitive ranking between different estimates is lost: e.g. $p(k = 6)$ may not depend on $p(k = 5)$. Second, the largest possible count $k_{\max}$ is given *a priori*. Despite these limitations, classification-based approaches have successfully been applied in deep neural networks for counting objects [12, 13, 18] in images.

Gaussian Regression: In regression, k is derived from an output distribution defined on the real line. The output distribution in this setting is assumed to be Gaussian and the associated cost function is the classical squared error. During inference and given the output $f_\theta(\mathbf{X})$ of the network, the best discrete value that is consistent with the model is simply the rounding operator $q = [\cdot]$. Gaussian regression has achieved state-of-the-art counting performance in computer vision using deep learning frameworks [14–17].

Discrete Poisson modelling: When it comes to modelling count data, it is often shown effective to adopt the Poisson distribution [19]. First, this strategy retains the advantage of the classification approach to directly pick a probabilistic model over the actual discrete observations, avoiding the somewhat artificial trick of introducing a latent variable that would be rounded to yield the observation. Second, the model avoids the inconvenience of the classification approach to completely drop dependencies between classes.

Due to these advantages, the Poisson distribution has been used in studies devising deep architectures for counting systems [20]. For instance in [19–21], it is shown that the number of objects in images can be well modelled by the Poisson distribution. Inspired by these previous works, we also consider the Poisson output distribution $\mathcal{P}\left(k \mid f_\theta(\mathbf{X})\right)$ where $\mathcal{P}\left(\cdot \mid \lambda\right)$ denotes the Poisson distribution with scale parameter λ.

In this setup, the cost function at learning time is the Poisson negative log-likelihood and the deep architecture at test time provides the predicted scale parameter $f_\theta(\mathbf{X}) \in \mathbb{R}_+$, which summarizes the whole output distribution.

As a decision function q in this setting, we considered several alternatives. A first option is to again resort to MAP estimation and pick the mode $[f_\theta(\mathbf{X})]$ of the distribution as a point estimate. However, experiments showed that the posterior median yields better estimates, and is given by

$$q\left(f_\theta(\mathbf{X})\right) = \operatorname*{argmin}_{\hat{k}} \sum_{k=0}^{\infty} \left|\hat{k} - k\right| \mathcal{P}\left(k \mid f_\theta(\mathbf{X})\right) \tag{3a}$$

$$= \text{median}\left(k \sim \mathcal{P}\left(f_\theta(\mathbf{X})\right)\right) \tag{3b}$$

where the median of a Poisson distributed random variable was approximated given the expression in [22].

3. PROPOSED MODEL

Various audio-related applications share common DNN architecture designs, often found by incorporating domain knowledge and through extensive hyperparameter searches. For our proposed task of source count estimation, however, domain knowledge is difficult to incorporate, as this study aims at revealing the best strategy to address the problem. Therefore, we use a network built upon an existing BLSTM-RNN architecture, that has already shown a considerable amount of generalization for various audio applications [23, 24].

A recurrent neural network (RNN) is very similar to a fully connected network, except that RNN applies the same set of weights recursively over an input sequence. RNNs can detect structure in sequential data of arbitrary length. This makes it ideal to model time series, however, in practice, the temporal context learnt is limited to only a few time instances, because of the vanishing gradient problem [25].

To alleviate this problem, forgetting factors (also called gating) were proposed. One of the most popular gated recurrent cells is the Long Short-Term Memory (LSTM) [26] cell. Its effectiveness has been proven in various applications and LSTMs are the state-of-the-art approach for speech recognition [27] or singing voice detection [23]. In this work we employed a bi-directional LSTM (BLSTM) with three hidden layers whose sizes are 30, 20 and 40 similar to the architecture introduced in [23]. A BLSTM is more robust compared to a simple LSTM, since input information from both past as well as the future in used to learn the weights. For further information on BLSTMs, the reader is referred to [28].

Output Type	Activation	Dim.	Loss
Classification	Softmax	$\mathbb{B}^{k_{\max}+1}$	Cat. cross entropy
Gaussian Regr.	Linear	$\mathbb{R}^1$	MSE
Poisson Regr.	Exponential	$\mathbb{R}^1$	Neg. log likelihood

Table 1: Output Activation Functions and Loss Functions

For a given input sequence, the output of a recurrent layer is either only the last step output or a full sequence. We found that employing the full sequence output of the last recurrent layer before feeding it into the fully connected output layer is important in the context of RNNs for count estimation. Furthermore we added a temporal max pooling layer with pooling size 2 to reduce the number of parameters for the fully connected layer. Temporal max pooling intuitively fits to our problem formulation which in itself is a maximum of the number of sources in a specific number of frames.

As we introduced in Section 2.1, the count estimation problem can be addressed using three different strategies. For each of the decision functions a suitable output activation and loss is used as shown in Table 1. Except for these (output) parameters, all models have the same parameters.

4. TRAINING

Since a realistic dataset of fully overlapped speakers is not available, we chose to generate synthetic mixtures. We recognize that in a simulated "cocktail-party" environment, mixtures lack the conversational aspect of human communication but provide a controlled environment which helps understand how a DNN solves the count estimation problem. As we aim for a speaker independent solution, we selected a speech corpus with a large number of different speakers instead of large number of utterances, yielding a larger number of unique mixtures. For training we selected the *LibriSpeech clean-360* [29] dataset which includes 363 hours of clean speech of English utterances from 921 speakers (439 female and 482 male speakers). As revealed in Section 2, the maximum number of concurrent speakers k requires annotation of the activity of each individual speaker. Even though *LibriSpeech* comes with annotations, they often are not consistent across different corpora. We therefore generated annotations based on a voice activity detection algorithm (VAD). In this work, we used the implementation from the *Chromium Web Browser* that is part of the WebRTC Standard [30].

To generate a single training sample $\{\mathbf{X}, k\}$, we draw a unique set of L speakers from the corpus. For each of the speakers we then select a random utterance, resampled to 16 kHz sampling rate and apply VAD. The VAD method was configured using a hop size of 10 ms. Further, the VAD estimate was used to remove silence in the beginning and the end of an utterance recording. In the next step, more utterances from the same speaker are drawn from the corpus until the desired duration is reached. Both, the audio recording and the VAD annotation of each utterance is concatenated. The procedure is repeated for all speakers so that L time domain signals are created. The signals are mixed and peak normalized to avoid clipping. Mixtures are then transformed to a time-frequency matrix $X \in D \times F$ as defined in Section 2. The ground truth output k are then computed using the VAD matrix based on Equation 1.

We follow the proposal of [10] and include non-speech examples in our training data to avoid using zero input samples for $k = 0$. For this, we used the TUT Acoustic Scenes dataset [31] to create negative training samples using the same procedure as described above. Because these environmental sounds could include speech, scenes with `cafe/restaurant`, `grocery store` and `metro station` were omitted.

As our application closely relates to source separation it is desirable for our trained DNN system to be robust against gain variations. We therefore find it important to make sure that the DNN cannot leverage the gain factors of the mixture. We found that the averaged energy of one bin across all frames of the input sample already is a solid indicator for the number of speakers. To accomodate these findings, we normalize $\mathbf{X}$ to the average Euclidean norm of all frames as used in [32]. Additionally, as common in machine learning, we scale the normalized input representation so that the feature dimensions have zero mean and unit variance/standard deviation across the whole training dataset.

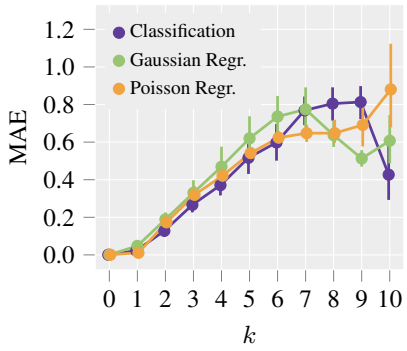
To train the network we use Poisson sampling to balance the number of samples T_k for each k. For our experiments we chose a medium-sized training dataset of $T_k = 1820$ samples $\forall k \in [0, \ldots, 10]$ resulting in a total of 20,020 training items, each containing 10 seconds of audio, resulting in 55.55 hours of training material. The actual duration of each input is reduced to five seconds by selecting a random excerpt from each mixture. For each excerpt Equation 1 is evaluated to generate a single sample, then combined into mini-batches of 32 samples. This way the network is seeing slightly different samples (in different order) in each training epoch. We found this procedure (also used in [33]) to help speeding up the stochastic gradient based training process. The DNN is trained using the ADAM optimizer [34]. In addition to the training dataset we created a separate validation dataset of $T_k = 5720$ samples using a different set of speakers from *LibriSpeech dev-clean*. Early stopping is applied by monitoring the validation loss to reduce the effect of overfitting. Training never exceeded 50 epochs.

5. EVALUATION

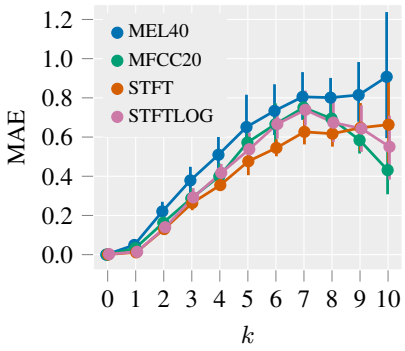
We evaluated our proposed network architecture with two main parameters: the three proposed output distributions (see Section 2) and four different input representations. To allow for a controlled test environment and at the same time limit the number of training iterations, we fix certain parameters: In our experiment all speakers were mixed to 0 dB SNR. For all experimental parameters we ran the training three times with different random seeds for each run and report averaged the results to minimize random effects caused by early stopping. We used the *LibriSpeech test-clean* subsets to generate 5720 unique and unseen speaker mixtures of five seconds duration for the test set with $k_{\max} = L = 10$.
Since we are dealing with a novel task description, related speaker count estimation techniques like those introduced in Section 1, could hardly be used as baselines. Specifically, [6] does not work on fully overlapped speech, [7] does not scale to the size of our dataset, since it requires to cross-correlate the full database against another. Finally, [5] proposes a feature but does not employ a fully automated system. We translated this method into a data-driven approach and employed a vector quantizer to get an optimal mapping with respect to the sum of squares criterion (we refer to this as *VQ*).

5.1. Input Representations

For our task, we chose several different input representations, well-established in speech application. We expect that a high frequency resolution is needed to discriminate time frequency bins with overlapped speech segments from those that only belong to a single speaker. We compared the following input representations that were



(a) Output Distribution



(b) Feature Representation

Fig. 2: Mean absolute error (MAE) over ground truth count $k = [0 \dots 10]$. Error bars show the 95% confidence intervals. Results are averaged over the different feature representations (a) and output distributions (b).

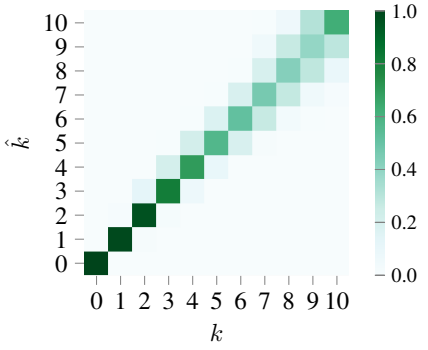


Fig. 3: Normalized confusion matrix showing $\hat{k}$ over k for the test data of the best performing network (Output Distribution: Classification, Feature Representation: STFT).

all based on a frame length of 25 ms:

STFT: magnitude of the Short-time Fourier transform computed using Hann windows. The resulting input is $X \in \mathbb{R}^{500\times201}$.

LOGSTFT: logarithmically scaled magnitudes from STFT representation using $\log(1 + STFT)$. The resulting input is $X \in \mathbb{R}^{500\times201}$.

MEL40: compute mapping from the STFT output directly onto Mel basis using 40 triangular filters. The resulting input is $X \in \mathbb{R}^{500\times40}$.

MFCC20: First 20 Mel-frequency cepstral coefficients. The resulting input is $X \in \mathbb{R}^{500\times20}$.

Before feature transformation, all input files were resampled to 16 kHz sampling rate. All features are computed using a hop size of 10 ms.

5.2. Metric

While the intermediate output y is treated as either a classification or a regression problem (See Section 2), we evaluate the final output k as a discrete regression problem. We therefore evaluate the performance using the mean absolute error (MAE), which is also used to evaluate other count related tasks (c.f. [14, 20]).

5.3. Results

To find the best parameters, we performed training and evaluation for the parameters, resulting in 36 trained networks. On average each network was trained 33 epochs before early stopping was engaged. Training duration was about 800 seconds per epoch on a NVIDIA GTX 1080 GPU. We present the results in terms of input representation and output distribution in Figure 2. One can see that the overall trend of the count error in MAE is similar regardless of the parametrisation: all models are able to reliably distinguish between $k = 0$ and $k = 1$, followed by a nearly linear increase in MAE for $k = [1 \dots 7]$. For $k > 7$ it can be seen that the classification networks have learned the maximum of k across the dataset, hence the prediction error decreases when k reaches its maximum. This is because classification based models intrinsically have access to the maximum number of sources determined by the output vector dimensionality.

Figure 2a indicates that *Classification* outperforms the other two distributions while Poisson regression performs better than Gaussian regression which confirms the findings made in [20] on object counts. With respect to the input representation, as shown in Figure 2b, despite its larger input dimension, choosing linear STFT as generally results in a better performance compared to *MEL40*, *LOGSTFT* or *MFCC20*.

A detailed analysis of all distribution and feature combinations, not shown here due to space constraints, reveals that STFT + *Classification* performs best. This model achieves results of (MAE 0.38 ± 0.28) for $k = [0 \dots 10]$ while the *VQ* baseline (MAE 2.41 ± 1.08) only performs slightly better than a mean estimator predicting $\hat{k} = 5$ (MAE 2.73 ± 1.63). To show the level of overestimation or underestimation, we depict all responses in a confusion matrix (see Figure 3). Unlike humans that generally tend to underestimate for the task of speaker count estimation [35], one can see that our proposed model slightly overestimates for smaller k.

6. CONCLUSION AND OUTLOOK

We introduced the task of estimating the maximum number of concurrent speakers in a simulated "cocktail-party" environment using a data-driven approach. We evaluated three different methods to output integer source count estimates in conjunction with defining cost function over which to optimize. Our experiments revealed a trade-off between better overall performance but requiring the maximum number of speakers to be estimated as prior knowledge (classification) and slightly worse performance when treated as a regression problem using Poisson distribution. Furthermore, we investigated and evaluated suitable input representations. Our final proposed BLSTM based classification model achieves mean absolute error of less than 0.4 speakers for zero to ten speakers. We think this first study on data-driven speaker count estimation opens the field to interesting and new research. Future work would be to evaluate and optimize other network structures such as convolutional neural networks and investigate the strategy a machine learning source count model pursues to solve the problem.

Acknowledgements

The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

7. REFERENCES

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1557–1565, 2006.

[2] S. Mirzaei, Y. Norouzi, et al., "Blind audio source counting and separation of anechoic mixtures using the multichannel complex nmf framework," *Signal Processing*, vol. 115, pp. 27–37, 2015.

[3] O. Walter, L. Drude, and R. Haeb-Umbach, "Source counting in speech mixtures by nonparametric bayesian estimation of an infinite Gaussian mixture model," in *Proc. IEEE (ICASSP)*, 2015, pp. 459–463.

[4] T. Arai, "Estimating number of speakers by the modulation characteristics of speech," in *Proc. IEEE (ICASSP)*, 2003, vol. 2, pp. II–197.

[5] H. Sayoud and S. Ouamour, "Proposal of a new confidence parameter estimating the number of speakers-an experimental investigation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 2, pp. 101–109, 2010.

[6] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in *Proceedings of the 2013 ACM UbiComb 13*. ACM, 2013, pp. 43–52.

[7] V. Andrei, H. Cucuand, A. Buzo, and C. Burileanu, "Counting competing speakers in a time frame - human versus computer," in *Proc. Interspeech Conf.*, 2015.

[8] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE (ICASSP)*, 2017.

[9] J.R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE (ICASSP)*, Mar. 2016, pp. 31–35.

[10] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proc. ACM Intl. Conference on Multimedia (ACMMM)*. ACM, 2015, pp. 1299–1302.

[11] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] A. Khan, S. Gould, and M. Salzmann, "Deep convolutional neural networks for human embryonic cell counting," in *European Conference on Computer Vision*. Springer, 2016, pp. 339–348.

[13] S. Seguí, O. Pujol, and J. Vitria, "Learning to count with deep object features," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 90–96.

[14] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 833–841.

[15] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–498.

[16] M. Marsden, K. McGuiness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," *arXiv preprint arXiv:1612.00220*, 2016.

[17] L. Boominathan, S. SS. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proc. ACM Intl. Conference on Multimedia (ACMMM)*. ACM, 2016, pp. 640–644.

[18] J. Zhang, S. Ma, M. Sameki, S. Sclaroff, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech, "Salient object subitizing," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CCVPR)*, 2015, pp. 4045–4054.

[19] N. Fallah, H. Gu, K. Mohammad, S. A. Seyyedsalehi, K. Nourijelyani, and M. R. Eshraghian, "Nonlinear poisson regression using neural networks: a simulation study," *Neural Computing and Applications*, vol. 18, no. 8, pp. 939, 2009.

[20] S. H. Rezatofighi, V. Kumar BG, A. Milan, E. Abbasnejad, A. Dick, and I. Reid, "DeepSetNet: Predicting sets with deep neural networks," in *Proc. IEEE Intl. Conference on Computer Vision (ICCV)*, 2017.

[21] A. B. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *Proc. IEEE Intl. Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 545–551.

[22] K. P. Choi, "On the medians of gamma distributions and an equation of ramanujan," *Proceedings of the American Mathematical Society*, vol. 121, no. 1, pp. 245–251, 1994.

[23] S. Leglaive, R. Hennequin, and R. Badeau, "Singing voice detection with deep recurrent neural networks," in *Proc. IEEE (ICASSP)*, April 2015, pp. 121–125.

[24] G. Hagerer, V. Pandit, F. Eyben, and B. Schuller, "Enhancing LSTM RNN-Based speech overlap detection by artificially mixed data," in *Proc. Audio Eng. Soc. Conference on Semantic Audio*, Jun 2017.

[25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

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

[27] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE (ICASSP)*, May 2013, pp. 6645–6649.

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

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. IEEE (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[30] "WebRTC VAD v2.0.10," `https://github.com/wiseman/py-webrtcvad/tree/2.0.10`.

[31] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, 2016.

[32] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. IEEE (ICASSP)*, April 2015, pp. 2135–2139.

[33] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples," in *Proc. of ISMIR)*, 2016, pp. 44–50.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[35] T. Kawashima and T. Sato, "Perceptual limits in a simulated cocktail party," *Attention, Perception and Psychophysics*, vol. 77, no. 6, pp. 2108–2120, 2015.