DECAR: Deep Clustering for learning general-purpose Audio Representations

Sreyan Ghosh⋆, Ashish Seth⋆, Sandesh Katta⋆, S. Umesh

Speech Lab, Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India

gsreyan@gmail.com, cs20s030@gmail.iitm.ac.in, sandesh.katta97@gmail.com, umeshs@ee.iitm.ac.in

Abstract

In this paper, we introduce DECAR (DEep Clustering for learning general-purpose Audio Representations), a self-supervised pre-training approach for learning general-purpose audio representations. Our system is based on clustering: it utilizes an offline clustering step to produce pseudo-labels and trains the network with a classification loss supervised by these pseudo-labels. We develop on top of recent advances in self-supervised learning for computer vision and design a lightweight, easy-to-use, self-supervised pre-training scheme for learning audio representations. We pre-train DECAR embeddings on a balanced subset of the large-scale AudioSet dataset and FSD50K and evaluate our representations on the LAPE Benchmark consisting of 11 downstream classification tasks, including speech, music, animal sounds, and acoustic scenes. Experimental results show that DECAR achieves results competitive to the state-of-the-art on both linear evaluation and transfer learning evaluation paradigms across all the downstream tasks in LAPE and performs better than other prior-art in literature with just 15% of the total amount of data available for pre-training. Furthermore, we conduct ablation studies identifying key design choices and also make all our code and pre-trained models publicly available.[1]

Index Terms: audio, speech, self-supervision, clustering

1. Introduction

Self-supervised representation learning (SSL) aims at learning high-level features from unlabeled data. It is rapidly closing the performance gap with supervised pre-training in a wide range of tasks across Computer Vision (CV) [1,2], Natural Language Processing (NLP) [3], and Speech Processing, especially Automatic Speech Recognition (ASR) [4,5]. Systems that leverage SSL algorithms to learn representations solve various tasks on large, unlabeled datasets. The weights learned are then either transferred for solving other downstream tasks under the transfer-learning paradigm or used as a feature extractor for input into another system for learning a different set of weights for another task under the linear-evaluation paradigm.

The goal of speech and audio representation learning is to learn a transformation from the acoustic signal that makes high-level information more accessible to downstream tasks. In the acoustics domain, the recent success of unsupervised speech representation learning has shown to outperform low-level features like Mel-spectrograms, waveform, or filter-banks in various Spoken Language Processing (SLP) downstream tasks [6]. The primary reason is that the former tends to learn more detailed information from speech and does not over-fit noise like the former. Despite recent progress, most research on SSL for speech and audio processing focuses mainly on the task of ASR [4,5] and ignores other speech and audio tasks. Though recent findings reveal that these representations can be utilized for full-stack speech processing [6,7], again, most of these techniques perform well only on other speech-based tasks like Speaker Verification (SV), Speech Emotion Recognition (SER), etc. But they fail to achieve good results on non-speech audio tasks such as acoustic scene detection, animal vocalizations, etc. [8]. We hypothesize that this might be due to an implicit language model that the model learns through solving a Masked Acoustic Modelling (MAM) task using individual speech frames. Moreover, SSL pre-trained models have been known to capture various kinds of phonetic, semantic, and syntactic information [9] due to its training paradigm, which makes them more suitable for Spoken Language Understanding (SLU) tasks.

The field of self-supervised audio representation learning has been relatively under-studied compared to SLU tasks, with very few works focusing on learning audio representations that could generalize across various tasks. We acknowledge that fully-supervised learning on any dataset can over-fit the task, and un-supervised training procedures could generalize better. In this space, the most recent works include [10,11,12,13,14,15,16]. Triplet-based objectives used in [10,13], heavily rely on the mining of negative samples, and the quality of learned features varies significantly with the sample generation scheme [11]. The methodology used in [12,16] learns general-purpose audio representations by solving a contrastive task. Contrastive learning systems typically work online and rely on many explicit pairwise feature comparisons, which is computationally challenging, given the fact they require large batches for mining negative samples. On the other hand, [14,15] propose the use of momentum encoders in a student-teacher training setup. However, the asymmetry of each network learning update introduced by the “stop-gradient” proves important in preventing trivial solutions in this kind of setup.

To alleviate the above problems, inspired by the DeepCluster framework in CV [1,7], we introduce DECAR: DEep Clustering for General-Purpose Audio Representations. It’s a simple self-supervised pre-training framework to learn general-purpose audio representations of sounds beyond and including speech. Our system requires no prior knowledge, minimal additional steps, alleviates dependency on large batches, and closely resembles a general supervised training procedure. Formally, DECAR employs an offline clustering step to generate noisy “pseudo-labels” from discrete audio samples and then learns by predicting cluster assignments using supervision from these “pseudo-labels”. We show that it is possible to obtain useful general-purpose audio features with a clustering framework.

We demonstrate the effectiveness of DECAR over a range of challenging, and diverse downstream tasks, including speech,
music, acoustic scenes, and animal sounds from the LAPE benchmark [10]. After pre-training on a relatively smaller unlabelled dataset, compared to prior work, we achieve results that outperform most other systems proposed in the literature and are competitive in performance to the state-of-the-art (SOTA) approach on the LAPE Benchmark with half the number of the trainable model parameters used for it.

2. Related Work

The past decade has seen impressive success in self-supervised learning in vision, speech, and text, pushing boundaries in low-resource downstream task settings [2 3 4]. Methodologies to learn representations from discrete or continuous input sequences in SLP use some form of Masked Acoustic Modelling (MAM), by solving different tasks like contrastive learning [4], reconstruction [27] and class prediction [3 4]. However, this learning from continuous speech frames makes models learn more about phonetic and semantic signals from speech, which is unsuitable for non-speech tasks. Beyond obvious algorithmic differences, this is also a major difference in learning methodology between audio and speech representation learning paradigms. For example, more specifically, HuBERT [5] and our approach include [10, 11, 12, 13, 14, 15, 16]. However, as discussed in section 1, most of these approaches are not robust and ours, where we learn not to use properties of individual frames but try to learn using a single representation for each audio file.

Common general-purpose audio representation learning approaches include [10 11 12 13 14 15 16]. However, as discussed in section 1, most of these approaches are not robust and ours, where we learn not to use properties of individual frames but try to learn using a single representation for each audio file. The DECAR pre-training framework for learning general-purpose audio representations

Figure 1: DECAR pre-training framework for learning general-purpose audio representations

3. Methodology

3.1. Problem Formulation

Let $X$ be an unlabelled dataset of size $N$ from a general domain where $X = \{x_1, \ldots, x_i, \ldots, x_N\}$ and $x_i$ is the $i^{th}$ audio sample from $X$. Also let $D^t$ be the task specific labeled dataset for task $t$ of size $I$, and $D^t = \{(x_1^{t}, y_1^{t}), \ldots, (x_i^{t}, y_i^{t}), \ldots, (x_I^{t}, y_I^{t})\}$ and $y$ is the corresponding label for audio sample $x$. Our primary aim here is to pre-train our ConvNet feature encoder using $X$ on the DECAR training framework, post which use $D^t$ to fine-tune our model with supervision on either of the transfer-learning or linear-evaluation setups.

3.2. Model Architecture

This section briefly describes the ConvNet feature encoder used in our experiments for learning general-purpose audio representations with the DECAR pre-training setup. For the ConvNet feature encoder, we resort to a more straightforward setup consisting of 3 convolutional encoder blocks. Each block consists of a single Conv2D layer, followed by Batch-Normalization, a ReLU activation function, and a MaxPool2D operation. These convolutional blocks are followed by an encoder head that consists of 2 fully-connected layers. Each of them is followed by a ReLU activation function and has a DropOut layer in between. Post obtaining the encoder output, we apply mean and max pool operations and finally sum the outputs to get a final embedding $e \in \mathbb{R}^d$ for each input audio sample where $d$ is a tunable hyper-parameter. This architecture has been previously used in [14 15] and achieved remarkable performance in addition to being resource-friendly. We attribute much of its success to the fact that the final fully-connected layers operate only on the frequency and channel dimensions.

During pre-training, we add a projection-head to our encoder with two fully-connected layers each with $j$ units, with the first one followed by Batch-Normalization and a ReLU activation function and finally an extra prototype head with $p$ units. Intuitively $p$ corresponds to the number of cluster centroids in our DECAR learning algorithm. Thus, during pre-training, our model outputs two vectors, $g$ and $z$ where $g = \text{projectionhead}(e)$ and $z = \text{prototypehead}(g)$. Post pre-training, we simply drop both the projection-head and prototype-head and pass $e$ to a fully-connected layer which we call the prediction-head with $t$ units where $t$ corresponds to the number of classes to predict in the downstream task.

3.3. DECAR Learning Algorithm

Our DECAR pre-training approach is inspired by DeepCluster-V2 proposed by [1][7]. The DECAR pre-training framework consists of two main phases 1) the Training Phase and 2) the Assignment Phase. In the next couple of sections, we will briefly discuss both.

3.3.1. Assignment Phase

The primary purpose of this phase is to obtain “pseudo-labels” $q$ for every unlabelled audio sample $x \in X$. To achieve this, we first store all the embeddings $\hat{e}_x$ obtained from our ConvNet \text{projectionhead}() in memory for the entire $X$ before applying Spherical K-means to cluster and get the “pseudo-labels” $q$ for
softmax \( \tau \) represent the temperature constant.

prototypehead

\[ x \]

every \( x \) as follows:

\[
\min_{C \in \mathbb{R}^{d \times K}} \frac{1}{N} \sum_{n=1}^{N} \min_{q} \mathbf{g}_n^T C q
\]

(1)

where \( C \) is the Centroid matrix and both \( \mathbf{g}_n \) and columns of \( C \) are \( I_2 \) normalized. \( K \) here represents number of clusters, \( \tilde{x} \approx x \) is an augmented and sampled version of the original audio sample, more about which can be found in Section 4.2

3.3.2. Training Phase

The primary purpose of this phase is to train the network using supervision from the “pseudo-labels” \( q \) obtained from the assignment phase. To do this, we first obtain the prototype \( p \) using:

\[
\mathbf{p}^{(k)} = \frac{\exp \left( \frac{1}{\tau} \mathbf{g}_n^T c_k \right)}{\sum_{k'} \exp \left( \frac{1}{\tau} \mathbf{g}_n^T c_{k'} \right)}
\]

(2)

where \( c_k \) corresponds to the \( k \)th column of \( C \) and \( \tau \) represent the temperature constant. \( p \) can also be viewed as \( \text{softmax}(z) \) where \( z \) is the output of the \( \text{prototypehead}(\cdot) \). Post this step, we minimize the multinomial logistic loss between \( p \) and \( q \) as follows:

\[
f(p, q) = - \sum_k q^{(k)} \log \mathbf{p}^{(k)}
\]

(3)

The prototypes are kept fixed during the training phase and updated for the entire \( X \) only once every epoch during the assignment phase. Similar to \([14]\), the assignment phase and training phase takes place in isolation only at the first epoch. After which, we use the embeddings \( g \) obtained from the previous epoch, which we store in memory at every iteration of the previous epoch right after the back-propagation step.

4. Experimental Setup

4.1. Datasets

For our experiments, we resort to the exact upstream pre-training and downstream fine-tuning dataset settings proposed by the LAPE Benchmark \([16]\). The choice of upstream pre-training data fits well with the low-resource SSL pre-training paradigm, which alleviates the problem of scale and accessibility in SSL \([28]\). We use the balanced subset of the large-scale AudioSet dataset (≈ 0.2 million audio files) and FSD50K (≈ 0.05 million audio files). This accounts for about 15% of the total AudioSet dataset (2 million audio files) commonly used in general-purpose audio representation learning literature \([12,14]\). For downstream classification, LAPE proposes to evaluate representations on 11 challenging classification tasks, including speech, music, animal sounds, and acoustic scenes. Further details regarding all the datasets used for the downstream tasks are shown in Table 1

| Downstream Task (DT) | Target | No. of Classes | No. of Samples | Avg. Duration |
|----------------------|--------|----------------|----------------|--------------|
| Speech               |        |                |                |              |
| LibriSpeech (LBS) \([18]\) | Speaker Identification | 585 | 28,538 | 12.69 |
| VoxCeleb 1 (VC) \([19]\) | Speaker Identification | 1,211 | 153,397 | 8.20 |
| IEMOCAP (IC) \([20]\) | Emotion Recognition | 4 | 4,490 | 4.49 |
| Speech Commands V1 (SC-V1) \([21]\) | Keyword Recognition | 12 | 64,721 | 0.98 |
| Speech Commands V2 (SC-V2) \([21]\) | Keyword Recognition | 12/35 | 105,829 | 0.98 |
| VoxForge (VF) \([22]\) | Language Identification | 6 | 176,438 | 6.68 |
| Non-Speech            |        |                |                |              |
| Bird Song Detection (BSD) \([23]\) | Song detection | 2 | 15,690 | 10.08 |
| NSynth (NS) \([24]\) | Musical Instruments Identification | 11 | 301,883 | 4.00 |
| TUT-Urban 2018 (TUT) \([25]\) | Acoustic Scene Classification | 10 | 8640 | 10.00 |
| UrbanSound8K (US8K) \([26]\) | Acoustic Scene Classification | 10 | 8732 | 4.00 |

Table 1: Dataset statistics for downstream benchmark tasks. The settings are consistent with the LAPE Benchmark proposed in \([16]\)
Table 2: Result comparison of DECAR with prior-art. Results for approaches other than DECAR have been taken from literature. ‘−’ signifies that results were not reported for these tasks by these methods. All results are reported in “linear evaluation / transfer learning” format.

| DT | TRILL | COLA | BYOL-A | Wav2Vec | SSAST | DeLoRes-S | DeLoRes-M | DECAR |
|----|-------|------|--------|---------|-------|-----------|-----------|-------|
| **Speech** | | | | | | | | |
| SC-V1 | 74.0/- | 71.7/98.1 | -/- | -/- | -/- | 96.2 | -/- | 94.0/97.7 |
| SC-V2 (12) | 74.0/91.2 | -/- | 91.0/- | -/- | -/- | 85.4/97.8 | -/- | 93.3/97.9 |
| SC-V2 (35) | -/- | 62.4/95.5 | 92.2/- | -/- | -/- | -/- | -/- | 98.2 |
| LS | -/- | 100.0/100.0 | -/- | -/- | -/- | 90.0/95.3 | -/- | 95.7/95.7 |
| VC | 17.7/17.6 | 29.9/37.7 | 40.1/- | -/- | -/- | -/- | -/- | 66.6 |
| IC | -/- | -/- | -/- | -/- | -/- | -/- | -/- | 59.8 |
| VF | 88.1/94.1 | 71.3/82.9 | **90.2/-** | -/- | -/- | -/- | -/- | 76.5/95.6 |
| **Non-Speech** | | | | | | | | |
| NS | -/- | 63.4/73.0 | 74.1/- | -/- | -/- | -/- | -/- | 66.3/78.6 |
| BSD | -/- | 77.0/73.0 | -/- | -/- | -/- | -/- | -/- | 86.7/90.3 |
| TUT | -/- | **94.0/99.2** | -/- | -/- | -/- | -/- | -/- | 58.6/64.1 |
| US8K | -/- | -/- | 79.1/- | -/- | -/- | -/- | -/- | 71.2/74.2 |

4.3. Experimental Results

Table 2 reports the accuracy of DECAR embeddings over 11 downstream classification tasks from the LAPE Benchmark. Consistent with prior-art in literature, we evaluate the effectiveness of DECAR embeddings on both linear evaluation and transfer learning setups. For linear evaluation we keep all parameters but the task-specific prediction-head frozen while downstream fine-tuning. For transfer learning all parameters of the model are kept trainable. We only compare the latest and the best (in terms of performance) work in literature to space constraints. Not all methods in the literature report results on all 11 tasks, and in Table 2 we report only what they do in linear evaluation / transfer learning format. Except for DeLoRes-S and DeLoRes-M, which are trained and evaluated on LAPE, all other approaches in Table 2 were pre-trained on all of AudioSet, which is about 10x the data proposed by LAPE.

As we see, DECAR is competitive with prior-art in literature when pre-trained on only 15% of the total available data. A lot of these results were also reported using larger models with more parameters, a detailed comparison of which with our model can be found in [10]. In the low-resource pre-training paradigm, DECAR outperforms DeLoRes-S on all 11 tasks by a significant margin and is competitive with DeLoRes-M, which, as mentioned earlier, is more compute hungry than DECAR.

Table 3: Comparison between DeepCluster-v1 and DeepCluster-v2 in DECAR pre-training framework

| DT | DeepCluster-v1 | DeepCluster-v2 (proposed) |
|----|----------------|--------------------------|
| SP-V1 | 82.3 | 91.6 |
| SP-V2 (12) | 83.0 | 90.6 |
| SP-V2 (35) | 73.6 | 87.2 |
| LS | 91.0 | 92.5 |
| VC | 25.6 | 33.0 |
| IC | 63.2 | 65.2 |
| VF | 74.1 | 78.2 |
| NS | 70.7 | 69.8 |
| BSD | 87.7 | 88.5 |
| TUT | 62.5 | 64.6 |
| US8K | 70.1 | 73.2 |
| **Avg** | 71.2 | 75.6 |

5. DeepCluster-v1 vs v2

Our work is based on DeepCluster-v2 [17] which was proposed to an extension to the original work proposed in [11]. DeepCluster-v2 proposes several improvements over DeepCluster-v1, including avoiding an extra forward pass while SSL pre-training by implementing a memory bank, which was a time-consuming affair. Adding to this, DeepCluster-v2 avoids instability while pre-training by not re-initializing the projection-head, at every iteration like DeepCluster-v1, and also discards the re-assignment trick which helped DeepCluster-v1 not collapse. In this section, we compare the performance of DECAR, when pre-trained using DeepCluster-v1 approach, on the same experimental setup settings as the former. Table 3 reports results comparing both and as we clearly see, DeepCluster-v2 clearly outperforms its v1 counterpart by a significant margin. Here we would like to add that the DeepCluster-v1 training paradigm has been recently explored by [3], but our work is the first to adopt v2 to the acoustic domain. Our work might act as a stimulus to steer future research in this direction.

6. Conclusion and Future Work

In this paper, we present DECAR, a simple and easy-to-implement self-supervised audio representation learning algorithm that alleviates the need for large batches, too many additional steps, or prior domain knowledge and avoids collapse through its nature of implementation. Our system achieves results competitive to SOTA on the LAPE Benchmark. It performs better than other approaches in literature with just 15% of the total amount of data available for pre-training. Future work would include exploring better ways under the clustering paradigm to push performance in low-resource SSL for audio.
7. References

[1] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in ECCV 2018, pp. 132–149.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in ICML 2020, pp. 1597–1607.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” NeurIPS 2020, vol. 33, pp. 12449–12460.

[5] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 3451–3460, 2021.

[6] S. Wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. Tiek Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. Y. Lee, “Superb: Speech processing universal performance benchmark,” in Interspeech 2021, pp. 1194–1198.

[7] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshokha, X. Xiao et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” arXiv preprint arXiv:2110.13900, 2021.

[8] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, “Ssaat: Self-supervised audio spectrogram transformer,” arXiv preprint arXiv:2110.09784, 2021.

[9] A. Pasad, J.-C. Chou, and K. Livescu, “Layer-wise analysis of a self-supervised speech representation model,” arXiv preprint arXiv:2107.04734, 2021.

[10] A. Jansen, M. Plakal, R. Pandya, D. P. Ellis, S. Hershhey, J. Liu, R. C. Moore, and R. A. Saurous, “Unsupervised learning of semantic audio representations,” in IEEE ICASSP 2018, pp. 126–130.

[11] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. D. C. Quirty, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, “Towards learning a universal non-semantic representation of speech,” Interspeech 2020.

[12] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” in IEEE ICASSP 2021, pp. 3875–3879.

[13] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. D. C. Quirty, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, “Towards learning a universal non-semantic representation of speech,” arXiv preprint arXiv:2002.12764, 2020.

[14] D. Nizam, D. Takeuchi, Y. Oshishi, N. Harada, and K. Kashino, “Byol for audio: Self-supervised learning for general-purpose audio representation,” in IEEE IJCNN 2021, pp. 1–8.

[15] Y. Gong, Y.-A. Chung, and J. Glass, “Ast: Audio spectrogram transformer,” arXiv preprint arXiv:2104.01778, 2021.

[16] S. Ghosh, A. Seth, and S. Umesh, “Delores: Decorrelating latent spaces for low-resource audio representation learning,” arXiv preprint arXiv:2203.13628, 2022.

[17] M. Caron, I. Misra, J. Marral, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrastive cluster assignments,” NeurIPS 2020, vol. 33, pp. 9912–9924.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015, pp. 5206–5210.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” arXiv preprint arXiv:1706.08612, 2017.

[20] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” LREC, vol. 42, no. 4, pp. 335–359, 2008.

[21] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” arXiv preprint arXiv:1804.03209, 2018.

[22] Voxforge.org, “Free speech... recognition (linux, windows and mac) - voxforge.org,” accessed 06/25/2014.

[23] D. Stowell, M. D. Wood, H. Pamula, Y. Stilianou, and H. Glotin, “Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge,” Methods in Ecology and Evolution, vol. 10, no. 3, pp. 368–380, 2019.

[24] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” arXiv preprint arXiv:1510.08484, 2015.

[25] A. Mesaros, T. Heititolaa, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” arXiv preprint arXiv:1807.09840, 2018.

[26] J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research,” in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 1041–1044.

[27] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in IEEE ICASSP 2020, pp. 6419–6423.

[28] A. Hannun, “The history of speech recognition to the year 2030,” arXiv preprint arXiv:2108.00084, 2021.