INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

KITE: Automatic speech recognition for unmanned aerial vehicles

Dan Oneață[1,2], *Horia Cucu*[1,2]

[1]University POLITEHNICA of Bucharest, Romania
[2]Autonomous Systems, Bucharest, Romania
dan.oneata@speed.pub.ro, horia.cucu@speed.pub.ro

Abstract

This paper addresses the problem of building a speech recognition system attuned to the control of unmanned aerial vehicles (UAVs). Even though UAVs are becoming widespread, the task of creating voice interfaces for them is largely unaddressed. To this end, we introduce a multi-modal evaluation dataset for UAV control, consisting of spoken commands and associated images, which represent the visual context of what the UAV "sees" when the pilot utters the command. We provide baseline results and address two research directions: *(i)* how robust the language models are, given an incomplete list of commands at train time; *(ii)* how to incorporate visual information in the language model. We find that recurrent neural networks (RNNs) are a solution to both tasks: they can be successfully adapted using a small number of commands and they can be extended to use visual cues. Our results show that the image-based RNN outperforms its text-only counterpart even if the command–image training associations are automatically generated and inherently imperfect.
Index terms: automatic speech recognition, multi-modal learning, domain adaptation

1. Introduction

As unmanned aerial vehicles (UAVs) are reaching consumer-level production, we expect an increasing effort into making them more accessible. One way to achieve accessibility is by designing interfaces that are easier to operate. The typical interface for UAVs relies on windows, icons, menus, pointers (WIMP), but recent research proposes a variety of interfaces, such as gestures [1, 2], gaze [3] or speech [4, 5]. Our work addresses the last category—controlling an UAV by spoken commands (we assume the utterances are recorded from the pilot's headset and transcribed by a system located on the ground).

We are interested in transcribing a diverse set of commands, from simpler, movement-related instructions (such as, *turn right* or *move up*) to more elaborate ones, specific to certain operational scenarios (such as, *zoom on the poacher shooting the rhinoceros* or *what type of trees does the truck carry?*). Being able to accurately recognize a wide range of instructions is a prerequisite in creating systems that can connect language to perception and action. The high-level queries we consider are similar to those encountered in situated language understanding for human-robot interaction [6] or visual question answering [7].

The first step towards building systems for UAV control is having a way of evaluating and comparing them. In this paper, we propose an evaluation dataset for this task, named KITE eval. While in other communities (*e.g.*, computer vision) UAV-related datasets are emerging [8, 9], we are, to the best of our knowledge, the first to introduce such a database for speech. In choosing the commands, we took inspiration from UAV pilots and tried to address relevant scenarios in which UAVs could be used; figure 1 shows a sample of commands for two such scenarios.

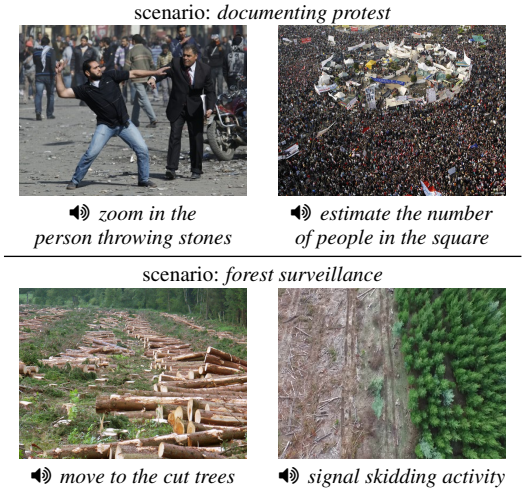

Figure 1: *Examples of commands and images from* KITE *eval.*

A baseline method for our task is a generic speech recognition system. However, since there is a domain mismatch between existing datasets and KITE eval, we do not expect such a system to perform particularly well. As an improvement, we consider adapting the generic system to the task at hand; in particular, we experiment with adapting the language models to UAV commands. While we can have a good idea of the type of commands given by a pilot in a particular scenario, it is unavoidable to encounter unforeseen commands at test time. To quantify such errors, we carry the adaptation procedure by varying the overlap between the training and testing commands.

A given command depends on a number of factors, *e.g.*, visual context, type of scenario, previous commands. In this paper, we try to leverage the visual information—what the UAV and its pilot are "seeing" when a command is uttered. Consider, for example, an UAV performing a forest surveillance operation, floating close to the border of a forest and a muddy path, as in the bottom right image from figure 1; given this picture, we expect a command like *signal skidding activity* rather than *signal skiing activity*. To handle such cases, we modify the language model to incorporate visual information and extend the evaluation set by manually assigning a relevant image to each command in KITE eval. Collecting manual associations for the training set is an expensive, if not prohibitive, task. Instead, we propose a semi-automatic procedure to generate correspondences between images and commands by relying on existing image databases.

To summarize, our main contributions are: 1. We introduce a multi-modal evaluation dataset for UAV control, where each command has associated its utterance and a relevant image (§3). 2. We build a baseline speech recognition system by using external data and compare it to improved models that are adapted

http://dx.doi.org/10.21437/Interspeech.2019-1390

on various amounts of data (§4 and §5). 3. We augment the language model to include visual information and use semi-automatic procedures to generate command–image associations as training data (§4 and §5).

2. Related work

We discuss two research directions related to our work.

Speech recognition for UAV control. The task of speech recognition for UAV control is relatively unexplored and the few published works on this topic [4, 5, 10] focus on recognition of simple commands: the authors of [4] predict a fixed set of nine commands using a classification pipeline based on audio features, such as energy and MFCC; the method in [10] recognizes commands to navigate through menus, operations which were previously achieved through keyboard presses.

Multi-modal learning. Systems that use multiple types of sensory data (*e.g.*, audio, visual, language) are known as *multi-modal* systems. Many works focus on combinations of two out of the three aforementioned modalities. Arguably, the most common combination is audio–language, as it includes the task of speech recognition, but the other two combinations, vision–language and audio–visual, are seeing increased attention.

Vision-language systems are used in tasks such as image captioning [11, 12] or visual question answering [7, 13]. Many such systems model the language in the context of an image by using an RNN to estimate the distribution over the words and a convolutional network to extract visual features [12, 14, 15, 16]; we use a similar architecture. Audio-visual systems target tasks such as image retrieval by speech [17, 18, 19], embedding learning [18, 20], speech-prompted object localization [19] or semantic keyword spotting [21]. The typical approach exploits statistical correspondences and learns embeddings for the two modalities, utterances and images, to a common sub-space.

The work of Sun *et al.* [22] combines all three modalities and is most similar to ours: they attempt to improve an ASR system based on a language model that takes the context image as input. We differ from them by taking other architectural decisions and, more importantly, by assuming a scenario with small amounts of data. For this reason we have to rely on out-of-domain datasets for initialization and semi-automatic methods to generate training data.

3. Dataset

In this section we introduce the KITE dataset,[1] a multi-modal dataset for UAV control. The dataset consists of three types of modalities: language (commands), audio (utterances), vision (images). We have build the dataset by first deciding on a set of commands, then recording the spoken utterances, and, finally, associating a image to each command.

We identified three types of UAV-specific commands: *(i)* movement-related, *(ii)* camera-related, and *(iii)* scenario-specific. The scenario-specific category was further split by considering seven types of scenarios, which we thought of interest for UAV applications: 1. *documenting a protest*; 2. *forest surveillance*; 3. *train surveillance*; 4. *anti-poaching operation*; 5. *natural disaster rescue operations*; 6. *ski monitoring*; 7. *sea monitoring*. We collaborated with UAV pilots to prepare a list of possible English commands based on these scenarios, which we then distilled into a finite state grammar (FSG), named *gold FSG*. The FSG representation has several advantages over a raw list of commands: *(i)* it is more compact, given that many commands overlap; *(ii)* it allows us to sample new commands, which are similar, but not necessarily identical, to the ones proposed by the pilots; *(iii)* it enables us to create new datasets, which are used in our experimental procedure for training.

[1] The dataset is available at `http://kite.speed.pub.ro`.

Table 1: *Statistics for* KITE *train. For each dataset of size n, we report the number of unique commands and the number of commands in the evaluation set. We report the mean and the standard deviation over the five folds. The last row indicates that the FSG can generate over 35K commands.*

		overlap with evaluation set	
n	unique	number	proportion (%)
2,048	1,313.2 ± 12.9	1,705.6 ± 10.1	59.2 ± 0.4
4,096	2,192.4 ± 37.0	1,900.8 ± 12.3	66.0 ± 0.4
8,192	3,532.2 ± 22.3	2,147.8 ± 14.1	74.6 ± 0.5
16,384	5,420.0 ± 52.4	2,360.8 ± 8.9	82.0 ± 0.3
32,768	7,795.8 ± 45.2	2,562.0 ± 9.3	89.0 ± 0.3
65,536	10,587.4 ± 31.3	2,702.4 ± 7.2	93.8 ± 0.2
∞	35,753	2,880	100

Evaluation dataset. We selected a set of 2,880 commands and recorded their utterances with the help of 16 L2 English speakers. Each speaker was assigned 180 commands: 20 movement-related, 20 camera-related, 20 for each of the seven scenarios. The utterances were recorded using a web application and allowed speakers to use their own recording environment. On average a command has five words and lasts about 3.5 seconds. Some examples of commands can be found in figure 1. This dataset is used for evaluation and we name it KITE eval. The recordings were done in noiseless conditions, but in order to simulate a real-world scenario we have corrupted the audio files with noise. We have selected noise samples corresponding to outdoor noises from the MUSAN dataset [23] and added them to the spoken utterances using a signal-noise ratio (SNR) of 10.

Training dataset. Based on the gold FSG we sampled datasets of different sizes, from 2,048 to 65,536 ($2^{11}, \ldots, 2^{16}$) commands; the varying number of commands simulates scenarios where we have access to different amounts of data. In order to account for the variability in the sampling process, we generate five folds for each dataset size. This dataset is used for training and we name it KITE train. Note that the training set is text-only and is employed in domain adaptation of the language models. Table 1 reports statistics of the dataset.

3.1. Visual extension

We extended both the training and evaluation parts with a visual component, by assigning each command with a relevant image (an image that might have been observed when that particular command has been uttered).

Evaluation dataset. The associations were done by searching images on search engines (such as Google Images) using queries that were related to a given command. We asked the participants to select images which are taken from a higher perspective, similar to what a UAV would record, but for some commands it was difficult to meet this requirement. See figure 1 for some examples of commands and their associated images.

Training dataset. Obtaining manual image–command associations for the training data would have been prohibitively expensive, so we relied on a semi-automatic approach. The idea was to link keywords from commands to the image classes from standard computer vision databases. Figure 2 shows an example

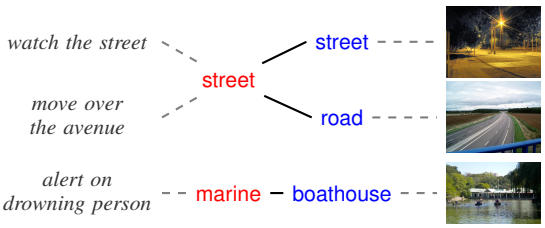

Figure 2: *Generated command–image associations.*

of creating such associations.

Instead of directly linking each command on the left to a corresponding image on the right, we just associate keywords (in red) to image classes (in blue). The effort is much reduced, because there are far fewer keywords and classes than commands and images. The command–keyword associations are done automatically by searching expressions in the commands, while the class–image associations are obtained from computer vision datasets, in our case ImageNet [24] and MIT Places [25].

4. Methodology

Our speech recognition system is based on an acoustic model and two language models. The acoustic model consists of a time delay neural network [26] and is implemented in Kaldi [27]. The first language model is used for decoding and it is either a finite state grammar (FSG) or an n-gram. The second language model is used for re-scoring and, hence, it is richer and more flexible than the first one; we use either a larger n-gram or an RNN.

We make use of existing databases to learn powerful representations. For acoustics we learn the model entirely on external data (the TED-LIUM dataset), while for language models we use external datasets to learn an initial set of parameters, which are then adapted to our task by fine-tuning. Figure 3 shows our systems' components and the use of datasets in our methodology; these are further presented in subsections 4.1 and 4.2.

4.1. Language models

In the following, we describe the language models and their adaptation procedure to the task of drone control.

FSG. Given the training corpus of commands, KITE train, we use the FSG to construct a language model that allows only the commands that appear in the training set. Practically, the training commands are joined by the `or` operator. The FSG is used only for decoding. Apart from the FSG built from the training data, we have also experimented with the gold FSG, which was manually created as described in section 3.

N-gram. We use two types of n-gram models, both of order four, one using 2M n-grams, while the other, 10M. The smaller n-gram is used for decoding and the larger one for rescoring. We perform domain adaptation by training a generic model (trained on the CANTAB dataset) and a specific one (trained on KITE train) and then interpolating the two [28]; we use a coefficient of 0.9 for the domain-specific model.

RNN. We use a recurrent network with long short-term memory cells (LSTM; [29]) to model the probability over the next word given a sequence. We "tie" the input and output embedding matrices [30, 31], as this has been shown to improve performance [32, 33]. The vocabulary is fixed at 10,000 words and the embedding size and the hidden size of the LSTM cells is set to 512; the resulting network has around 9.4M parameters.

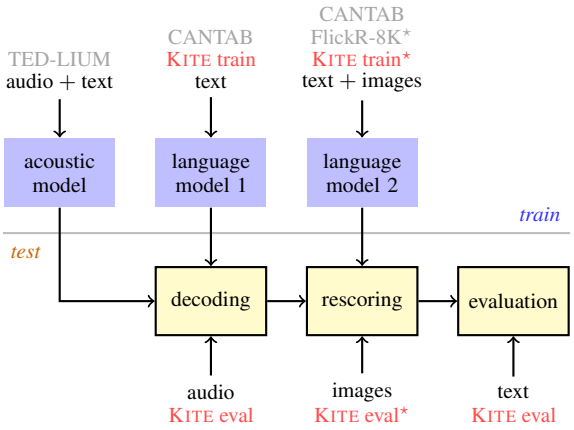

Figure 3: *Methodological overview. Our ASR system consists of an acoustic model and two language models.We initialize these components on generic datasets (in gray), and then adapt them to domain-specific data (in red). The datasets that have a visual component are marked with a star.*

Inspired by the results of Melis *et al.* [33], we optimize the hyper-parameters (*e.g.*, drop-out, learning rate) using an automatic procedure [34]. We perform domain adaptation by fine-tuning the network [35]: we train a generic model on a source dataset (CANTAB), and then continue updating the weights for a fixed number of epochs (25) on the target dataset (KITE train). The fine-tuning procedure allows us to train on smaller datasets, which otherwise would be difficult to fit. The RNN language model is used only for rescoring by reordering the n-best list.

Multi-modal RNN. We extend the RNN with an additional component, an encoder, which takes in an image, extracts features and passes them to the first LSTM cell. This architecture allows us to estimate the distribution over the next word w_t given the preceding words $\mathbf{w}_{1:t-1}$ and the visual context $\mathbf{v}$, that is, $p(w_t|\mathbf{w}_{1:t-1}, \mathbf{v})$. The encoding network is a residual network with 152 layers, ResNet152 [36], and is pre-trained on the ImageNet dataset [24]; the rest of the network (the text-only part) is pre-trained on CANTAB. We follow with a second pre-training step, by using the multi-modal FlickR-8K dataset. Finally, as in the case of the text-only RNN, we use fine-tuning to perform domain adaptation.

4.2. Auxiliary datasets

In addition to the KITE dataset, we have also used the following public datasets:

TED-LIUM [37] is a speech recognition corpus containing recordings of almost 1,500 TED talks (around 200 h). We use it to train the acoustic model.

CANTAB [38] is a large text corpus (14M sentences, 252M words), collected from multiple sources, such as news or law. We use it to initialize generic language models.

FlickR-8K [39] is an image–text corpus: there are 8,000 images, each described by five captions (in total around 500K words). We use this corpus to initialize the multi-modal language model.

5. Experimental results

In this section we present the results on KITE eval dataset.

Table 2: *Word error rate (WER) on the proposed* KITE *dataset using adapted language models. Rows 1–4 correspond to text-only models, while rows 5–6 correspond to multi-modal models. We report the mean and two times the standard error across the five folds for each training size, n. A system using an unadapted language model obtains 56.2% WER. Figures in italics indicate that the improvements from the best text-only model (row 4) to the multi-modal one (row 5) are statistically significant (based on McNemar's test at $p = 0.05$).*

				number of training sentences (overlap with evaluation set)					
	decoding	rescoring	img. assoc.	2,048 (59%)	4,096 (66%)	8,192 (75%)	16,384 (82%)	32,768 (89%)	65,536 (94%)
1	FSG	—		26.22 ± 0.1	22.35 ± 0.3	19.02 ± 0.2	16.32 ± 0.1	14.52 ± 0.1	13.19 ± 0.1
2	n-gram small	—		15.91 ± 0.2	15.11 ± 0.1	14.65 ± 0.1	14.43 ± 0.1	14.18 ± 0.0	14.30 ± 0.1
3	n-gram small	n-gram large		15.27 ± 0.2	14.53 ± 0.3	13.67 ± 0.1	13.40 ± 0.1	12.98 ± 0.0	12.88 ± 0.2
4	n-gram small	RNN		13.57 ± 0.1	12.43 ± 0.2	12.09 ± 0.1	11.89 ± 0.1	11.64 ± 0.1	11.48 ± 0.1
5	n-gram small	RNN	annotated	13.43 ± 0.1	12.07 ± 0.2	*11.31 ± 0.1*	*11.00 ± 0.1*	*10.78 ± 0.1*	*10.73 ± 0.1*
6	n-gram small	RNN	generated	13.49 ± 0.1	12.01 ± 0.2	11.32 ± 0.1	10.93 ± 0.2	10.60 ± 0.1	10.45 ± 0.0

Figure 4: *Transcriptions of commands using the text-only RNN (txt) or the multi-modal RNN (img) language model. The groundtruth is denoted by gt. The first row shows success cases, while the last one shows failure cases.*

Baseline systems. We compare the domain-adapted models against two baseline methods. Both systems use the same acoustic model, which is trained on the TED-LIUM dataset, but they differ in terms of the language model and the data used to train it: the first system uses an n-gram trained on the CANTAB dataset and corresponds to a generic, unadapted speech recognition system; the second system uses the gold finite state grammar (FSG), from which we sampled commands. The unadapted system obtains a word error rate of 56.2%, while the system relying on the gold FSG obtains an WER of 11.7%. These results highlight the importance of adapting the language model to the target domain.

Domain adaptation. The next experiment considers the case when we have access only to a partial list of commands at train time and we use those for domain adaptation. The results for multiple language models and varying amounts of data are shown in table 2; in particular, the first part of the table, rows 1–4, presents the results for text-only adaptation. As expected, the performance improves with more data and more flexible models. The FSG is more reliant on data and it converges towards a good performance at a slower rate than the other models. This behavior is expected, because if the exact command is missing from the training set, the FSG is unable to predict, whereas the other models are more flexible and can interpolate missing words. Rescoring improves the results further, with the RNN out-performing the n-gram model (rows 3 and 4).

Multi-modal experiments. In the final experiment, we focus on language models that use visual information; the corresponding results are in the second part of table 2, rows 5–6. The two experiments differ in the images used for evaluation: row 5 uses the standard KITE eval set of images collected manually from the internet; row 6 uses an automatically selected set of images from ImageNet and FlickR-8K, similar to what we have done for the train set. There are three main observations. First, we notice that the visual information helps improve over the text-only model. Second, the improvements are noticeable when we increase the data size, because the network is larger and needs more data to learn. Third, having a different distribution at test time and possibly imperfect correspondences at train time, does not impact the results: the differences between rows 5 and 6 are not statistical significant, while we still obtain statistically significant improvements over the text-only model (row 4).

In figure 4 we present qualitative results for the text-only (n-gram with RNN rescoring) and multi-modal models (n-gram with image RNN). We show both success (first row) and failure cases (second row). The multi-modal model is able to correct phonetically similar pairs which have visual grounding (*e.g.*, buoy–boy, track–truck, trail–train), but there are still cases where it biases too strongly towards the visual context (second row).

6. Conclusions

We have introduced a multi-modal dataset, KITE, for recognition of UAV commands. Its evaluation part was manually annotated and curated, while the training part relied on more automatic approaches. While the command–image associations used for training are likely to be imperfect, we have consistently found improvements over a text-only model. This result confirms the benefits of the visual context for transcribing. We have also shown the importance of adapting the language model and the benefits of using a more flexible model, as its performance is less reliant on the quantity of data. Finally, we conclude with a couple of research directions that can be carried around our dataset: *(i)* grounding the uttered commands in the images as a way of obtaining feedback from the system; *(ii)* improving the acoustic model by making it more robust to outdoor noises.

7. Acknowledgments

This research was partially supported by the POC-2015 P39-287 IAVPLN project.

8. References

[1] T. Naseer, J. Sturm, and D. Cremers, "FollowMe: Person following and gesture recognition with a quadrocopter," in *International Conference on Intelligent Robots and Systems*, 2013, pp. 624–630.

[2] E. Peshkova, M. Hitz, and B. Kaufmann, "Natural interaction techniques for an unmanned aerial vehicle system," *IEEE Pervasive Computing*, no. 1, pp. 34–42, 2017.

[3] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 617–623.

[4] S. Supimros and S. Wongthanavasu, "Speech recognition-based control system for drone," in *ICT International Student Project Conference*, 3 2014, pp. 107–110.

[5] M. Landau and S. van Delden, "A system architecture for hands-free UAV drone control using intuitive voice commands," in *IEEE International Conference on Human-Robot Interaction*, ser. HRI '17. New York, NY, USA: ACM, 2017, pp. 181–182.

[6] P. Gorniak and D. Roy, "Situated language understanding as filtering perceived affordances," *Cognitive science*, vol. 31, no. 2, pp. 197–231, 2007.

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "VQA: Visual question answering," in *International Conference on Computer Vision*, 2015, pp. 2425–2433.

[8] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 445–461.

[9] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu, "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[10] M. Draper, G. Calhoun, H. Ruff, D. Williamson, and B. , "Manual versus speech input for unmanned aerial vehicle control station operations," in *Human Factors and Ergonomics Society Annual Meeting*, vol. 47, 10 2003, pp. 109–113.

[11] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *International Conference on Machine Learning*, 2014, pp. 595–603.

[12] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[13] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[14] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain images with multimodal recurrent neural networks," *arXiv preprint arXiv:1410.1090*, 2014.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[17] G. Synnaeve, M. Versteegh, and E. Dupoux, "Learning words from images and speech," in *NIPS Workshop on Learning Semantics*, 2014.

[18] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *Advances in Neural Information Processing Systems*, 2016, pp. 1858–1866.

[19] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass, "Jointly discovering visual objects and spoken words from raw sensory input," in *European Conference on Computer Vision*, 2018, pp. 649–665.

[20] D. Harwath and J. Glass, "Deep multimodal semantic embeddings for speech and images," in *Workshop on Automatic Speech Recognition and Understanding*, 2015, pp. 237–244.

[21] H. Kamper, G. Shakhnarovich, and K. Livescu, "Semantic speech retrieval with a visually grounded model of untranscribed speech," *Transactions on Audio, Speech and Language Processing*, vol. 27, no. 1, pp. 89–98, 2019.

[22] F. Sun, D. Harwath, and J. Glass, "Look, listen, and decode: Multimodal speech recognition with images," in *Spoken Language Technology Workshop*, 2016, pp. 573–578.

[23] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 12 2015.

[25] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

[26] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." in *Interspeech*, 2015, pp. 3214–3218.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 12 2011.

[28] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

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

[30] H. Inan, K. Khosravi, and R. Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *arXiv preprint arXiv:1611.01462*, 2016.

[31] O. Press and L. Wolf, "Using the output embedding to improve language models," *European Chapter of the Association for Computational Linguistics*, p. 157, 2017.

[32] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," *arXiv preprint arXiv:1708.02182*, 2017.

[33] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," in *International Conference on Learning Representations*, 2018.

[34] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *Journal of Machine Learning Research*, 2013.

[35] S. Gangireddy, P. Swietojanski, P. Bell, and S. Renals, "Unsupervised adaptation of recurrent neural network language models," in *Interspeech*, 9 2016, pp. 2333–2337.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[37] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *LREC*, 2014, pp. 3935–3939.

[38] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," *arXiv preprint arXiv:1502.00512*, 2015.

[39] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.