INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Speaker Code Based Speaker Adaptive Training Using Model Agnostic Meta-learning

Huaxin Wu[1]*, Genshun Wan*[2]*, Jia Pan*[2]

[1]iFlytek Research, iFlytek Co., Ltd
[2]University of Science and Technology of China
hxwu2@iflytek.com, gswan@mail.ustc.edu.cn, panjia@mail.ustc.edu.cn

Abstract

The performance of automatic speech recognition systems can be improved by speaker adaptive training (SAT), which adapts an acoustic model to compensate for the mismatch between training and testing conditions. Speaker code learning is one of the useful ways for speaker adaptive training. It learns a set of speaker dependent codes together with speaker independent acoustic model in order to remove speaker variation. Conventionally, speaker dependent codes and speaker independent acoustic model are jointly optimized. However, this could make it difficult to decouple the speaker code from the acoustic model. In this paper, we take the speaker code based SAT as a meta-learning task. The acoustic model is considered as meta-knowledge, while speaker code is considered as task specific knowledge. Experiments on the Switchboard task show that our method can not only learn a good speaker code, but also improve the performance of the acoustic model even without speaker code.

Index Terms: automatic speech recognition, speaker adaptive training, model-agnostic meta-learning

1. Introduction

Recently, the accuracy of automatic speech recognition (ASR) has been greatly improved by the use of deep neural network (DNN) acoustic models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [1–3]. However, the performance is still unsatisfactory if the acoustic condition of the test data is mismatched to that of the training data, such as for speakers who have not been seen by the acoustic model. In response, speaker adaptive training (SAT) is one of the effective approaches to improve the performance of ASR on these conditions. SAT reduces the mismatch by removing speaker variance during training of the acoustic model, and it allows the acoustic model to focus solely on modelling phonetic variations.

In recent years, a lot of SAT approaches have been proposed. They can be divided into two groups: auxiliary feature methods and adversarial learning methods.

Auxiliary feature methods use auxiliary features to inform the acoustic model about speaker identity. In [4–7], speaker i-vectors or bottleneck vectors are obtained using a pretrained speaker recognition model. Then, acoustic features concatenated with the corresponding speaker vectors are fed to a DNN-based acoustic model. In [8–10], the authors use speaker codes which are learned together with the acoustic model to represent speaker identity information. In order to highlight the importance of the speaker embeddings in adaptation and provide speaker identity information more effectively, [11, 12] try to generate the speaker dependent parameters via a controller network that takes speaker embeddings as input.

Adversarial learning is also used to perform speaker adaptive training. Inspired by the methods used in domain adaptation [13–15], [16] optimizes the acoustic model and the speaker classification model jointly via adversarial learning. In [17], a reconstruction network is trained to predict the input speaker i-vector. The mean-squared error loss of the i-vector reconstruction and the cross-entropy loss of the acoustic model are jointly optimized through adversarial multitask learning.

Despite the progress, it is still a challenge of speaker adaptation to improve performance on test data as much as possible without overfitting, which is especially important in a rapid adaptation setting when we use only a small amount of adaptation data. Auxiliary feature methods need speaker identity information, but how to get the information is a question that needs to be considered. Features similar to i-vector [4–7] are extracted from other pre-trained models. They may not fit perfectly with current acoustic models. Speaker code method [8–10] is a useful way to provide information about speaker identity. But speaker codes and the acoustic model are optimized together in the same time. This training strategy makes it difficult to decouple them. What we really want is a speaker dependent code who only related to speaker identity, and a speaker independent acoustic model who never cares about speaker identity.

Adversarial learning methods aim to map the input speech frames from different speakers into speaker-invariant hidden features, so that further classification will be based on representations with the speaker factor already normalized out. They do not perform adaptive training on test speakers. For example, when we already have a small number of labeled speaker utterances, how to use those labeled utterances to improve the model's performance on unlabeled utterances of the same speaker is something that the adversarial learning method cannot do.

For this purpose, we introduce a meta-learning method called Model-Agnostic Meta-Learning (MAML) [18] to the speaker code based SAT framework. We consider automatic speech recognition on a specific speaker as a specific task. The acoustic model learns meta-knowledge across all speakers, and speaker code learns task-specific knowledge which indicates the speaker identity. We evaluated the effectiveness of the proposed method on the Switchboard dataset. The experiments reveal that our method can not only learn a good speaker code to improve the performance on target speaker, but also improve the performance of the acoustic model even without speaker code.

The remainder of this paper is organized as follows. In Section 2, we briefly review the speaker code method and the standard MAML algorithm as related works. In Section 3, we present two different methods to apply the MAML algorithm to speaker code based speaker adaptive training. In Section 4, we report and discuss our experimental results on the Switchboard

http://dx.doi.org/10.21437/Interspeech.2020-2296

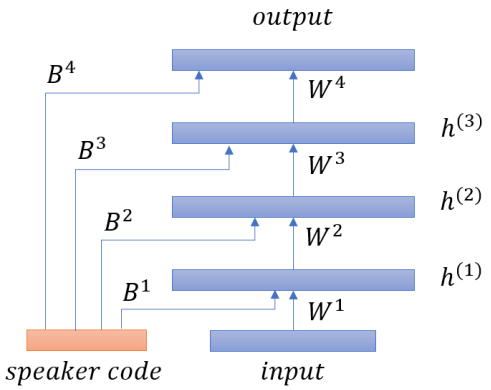

Figure 1: *Illustration of the model structure for speaker code based SAT.*

task. Finally, the paper is concluded in Section 5.

2. Related Work

2.1. Speaker code based SAT

Assuming that we have an (L+1)-layer DNN acoustic model consisting of weight matrices, denoted as $\mathbf{W}^l(1 \leq l \leq L+1)$, and the data come from C different speakers in total, we should have C different speaker codes, denoted as $\mathbf{s}^{(c)}(1 \leq c \leq C)$. Each speaker code is simply a vector, whose dimension can be freely adjusted. As shown in Fig. 1, these speaker codes are fed into some particular layers of DNN through another set of connection weights, denoted as $\mathbf{B}^l(l \in \mathcal{L})$, $\mathcal{L}$ stands for the number of layers connected with the speaker codes. For any layer $l(l \in \mathcal{L})$, it receives input features from both the lower layer $l-1$ and the speaker code, the output features in these layers are computed as follows:

$$\mathbf{h}^l = \mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{B}^l \mathbf{s}^{(c)} \quad (\forall l \in \mathcal{L}) \tag{1}$$

In the learning process, both speaker codes and their connection weight matrices are all randomly initialized. The weights of the DNN can be initialized from a pretrained ASR model. After that, all of these parameters are jointly learned using the standard BP algorithm. In the testing stage, a new speaker code is learned based on a small amount of adaptation data for each speaker while the other parameters of the acoustic model are frozen. The learned speaker code is used for all the utterances of the corresponding speaker. Experiments on the TIMIT and Switchboard task have shown that the speaker code method is quite effective to adapt large DNN models using only a small amount of adaptation data.

2.2. Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (MAML) [18] is a popular meta-learning framework. MAML learns initialization parameters θ_0 by meta training $\mathcal{M}_{train}$ such that the model can perform well on query set after a few steps of gradient descent. Support set $\mathcal{S}$ are used to calculate loss used for gradient computation. Suppose model f is initialized as f_{θ_0}, let $\theta_N = Adapt(\theta_0; L, \mathcal{S}, N)$ be the model parameters updated through N steps of gradient descent where the loss function is L computed on support set $\mathcal{S}$. The optimization problem is defined as Eq.(2), which minimizes the loss of f_{θ_N} on query set $\mathcal{Q}$:

$$\min_{\theta_0} L(\theta_N; \mathcal{Q}) = \min_{\theta_0} L(Adapt(\theta_0; L, \mathcal{S}, N); \mathcal{Q}) \tag{2}$$

In speech applications, MAML has been applied to ASR. For example, in [19], MAML is proved helpful for cross-language speech recognition. The results showed that MAML based approach significantly outperforms the state-of-the-art multitask pretraining approach on all target languages.

3. Proposed Method

For some parametric model f_θ, MAML aims to find a set of initial parameters θ_0 which can be used to fast adapt to any new task sampled from the same distribution. In other words, the parameters θ_0 could be regarded as meta-knowledge. For speaker code based SAT method, $\mathbf{W}^l$ as well as $\mathbf{B}^l$ are speaker independent, so they can alse be regarded as meta-knowledge across SAT procedure. This is our motivation of using MAML in speaker code based SAT. We propose two different methods to apply MAML.

3.1. SAT with zero-initialized speaker code (SAT-ZISC)

In the speaker code based SAT method, the speaker independent parameters are denoted as $\theta = (W, B)$. The adaptation procedure, or "inner loop", is formulated as following:

$$\mathbf{s}_N^{(c)} = Adapt(\mathbf{s}_0^{(c)}, \theta; L, \mathcal{S}, N) \tag{3}$$

where $\mathbf{s}_0^{(c)}$ is the initial speaker code for speaker c, L is the loss function, $\mathcal{S}$ is the support set which contains utterances of speaker c. During the adaptation procedure, or "inner loop", we freeze speaker independent parameters θ, and update initial speaker code by gradient descent. For example, when using one step of gradient update, the process can be described as following:

$$\mathbf{s}^{(c)} \leftarrow \mathbf{s}^{(c)} - \alpha \nabla_{\mathbf{s}^{(c)}} L(\mathbf{s}^{(c)}, \theta) \tag{4}$$

Generally, N adaptation steps could be applied to get adapted speaker code $\mathbf{s}_N^{(c)}$.The step size α and the number of steps N are fixed as hyperparameter.

When we get adapted speaker code $\mathbf{s}_N^{(c)}$, we could compute the loss on query set $\mathcal{Q}$, which contains some different utterances of the same speaker. And then we update the speaker independent parameters θ by gradient descent. The final optimization problem, what we referred as the "outer loop" of meta-learning, is defined as Eq.(5):

$$\min_{\theta} L(\mathbf{s}_N^{(c)}; \mathcal{Q}) = \min_{\theta} L(Adapt(\mathbf{s}_0^{(c)}, \theta; L, \mathcal{S}, N); \mathcal{Q}) \tag{5}$$

Fig. 2 shows the architecture of the proposed SAT-ZISC method. Since the acoustic model weights $\mathbf{W}$ are initialized from a pretrained ASR model and the connection weights $\mathbf{B}$ are initialized from scratch, we initialize all speaker codes $\mathbf{s}_0^{(c)}(1 \leq c \leq C)$ to zero vector, so they have no effect on the original acoustic model at the beginning, and make the training process more stable. At the end of the "inner loop " learning, the speaker code $\mathbf{s}_0^{(c)}$ will be saved and used as a initial speaker code for next inner-loop learning.

It is worth mentioning that only the speaker independent parameters θ are updated in the "outer loop" while the adapted speaker code keeps unchanged, which means we do not need to compute second derivatives as the original MAML does.

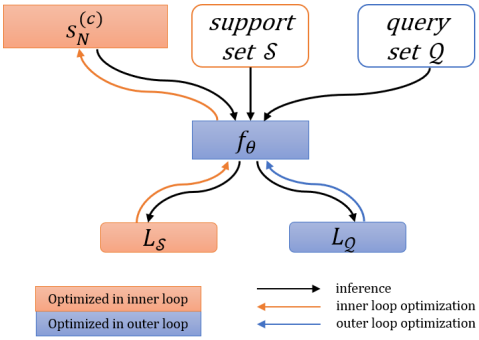

Figure 2: *Overview of the architecture of SAT-ZISC.*

In the testing stage, a new speaker code is first initialized to be zero, and then it is updated based on the derivatives of the adaptation data as Eq.(3). The learned speaker code will be fed to the model as in Eq.(1) for testing purpose.

3.2. SAT with meta-initialized speaker code (SAT-MISC)

Learning a good speaker code from a zero-initialized vector might be a hard work. In this section, we employ a new mothed called meta-initialized speaker code. Here not only the model parameters $\theta = (\mathbf{W}, \mathbf{B})$, but also the initial speaker code $\mathbf{s}_0$ are regarded as speaker independent parameters. As Fig. 3 shows, all speakers share a same initial speaker code $\mathbf{s}_0$. In the process of speaker adaptation, the initial speaker code is updated based on adaptation data of the speaker, to become a speaker dependent code $\mathbf{s}_N^{(c)}$:

$$\mathbf{s}_N^{(c)} = Adapt(\mathbf{s}_0, \theta; L, \mathcal{S}, N) \tag{6}$$

The final optimization problem, or "outer loop" is different from eq. It is formulated as following:

$$\min_{\theta, \mathbf{s}_0} L(\mathbf{s}_N^{(c)}; \mathcal{Q}) = \min_{\theta, \mathbf{s}_0} L(Adapt(\mathbf{s}_0, \theta; L, \mathcal{S}, N); \mathcal{Q}) \tag{7}$$

After training, we aim to get a speaker independent model and a initial speaker code $\mathbf{s}_0$ that is more suitable to speaker adaptation.

It is worth mentioning that during the training stage, we use second order derivatives to train the initial speaker code $\mathbf{s}_0$ according to Eq.(2). This could bring a significant computational expense. For computation efficiency, some previous works [18, 20] ignored the second-order term, which were alse known as First-order MAML(FOMAML). But we found that the training process was very difficult to converge when using FOMAML in our experiments. Finally we used second-order MAML in the SAT-MISC method, and its training speed was about 30% slower than the SAT-ZISC method.

4. Experiments

4.1. Dataset

All experiments were performed on the Switchboard (SWB) dataset. The training data of the SWB task [21] consists of 20-hour English CALLHOME and 309-hour Switchboard-I dataset, including a total of 5110 speakers. The SWB part of NIST 2000 Hub5 evaluation set is taken as test set, which contains 1831 utterances from 40 speakers in total. We use 20 utterances for each speaker to do speaker adaptive training. So the final test set contains 1031 utterances.

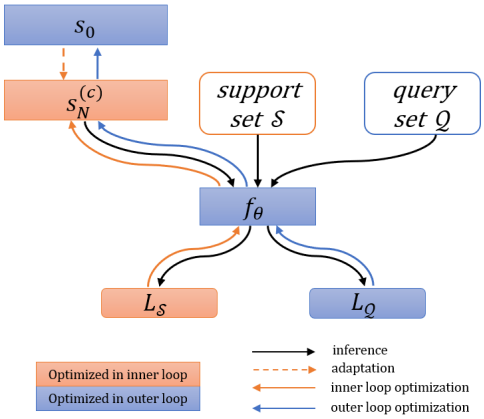

Figure 3: *Overview of the architecture of SAT-MISC.*

4.2. Baseline setup

The speaker independent baseline is trained with a VGG-like [22] model architecture based on frame-level cross-entropy criterion. The inputs of the model were the 40-dimensional log Mel-scale filter-bank features. The architecture of the model mainly consisted of convolutional and pooling layers, and each convolutional layer was equipped with a standard ReLU activation function. We shuffled the utterances in training data and grouped them into minibatches with a limit of 2048 frames per minibatch to speed up training. Stochastic gradient descent was used as the optimizer, and the initial learning rate was set to 0.02.

The speaker code baseline is trained based on the speaker independent baseline model. We connected the speaker code to the first layers of each convolutional block (layer conv0, conv1, conv5, conv9, conv13) in the VGG-like model. The connection method is as described in the Eq.(1). The DNN weights W were initialized from the speaker independent baseline and the connection weights $\mathbf{B}$ were randomly initialized. At the beginning, we initialized the speaker code $\mathbf{s}^{(c)}$ randomly, but found that the training process was very difficult to converge. So we initialized all speaker codes to zero vector. After that, all of the parameters were jointly optimized. The learning rate of $\mathbf{W}$ and $\mathbf{s}^{(c)}$ was set to 0.02, and the learning rate of $\mathbf{B}$ was set to 0.4. Table 1 reports the word error rate (WER) of the baseline models.

Table 1: *Performance of the baseline models on SWB.*

Method	**WER**	**WERR**
baseline	13.8	–
SC-baseline with adapted speaker code	13.5	2.2%

4.3. Results of the proposed method

Table 2 reports the performance of our proposed methods. For SC-baseline model, firstly we fed the model with a zero initialized speaker code that has no effect on acoustic parameters. It achieves only a relative 0.7% WER reduction (WERR) compared with the baseline model. After training on 20 utterances of the target speaker to get an adapted speaker code, it achieves a relative 2.2% WER reduction compared with the baseline model.

Both of our proposed SAT methods use 2 adaptation steps in the "inner loop" in the MAML training stage. In the test stage, the speaker code is optimized until it converges. As for the SAT-ZISC method, we also fed the model with a zero initialized speaker code to test the performance of the speaker independent acoustic model. It achieves a relative 3.6% WER reduction compared with the baseline model, which is much better than SC-baseline. When using an adapted speaker code, it achieves a relative 4.3% WER reduction compared with the baseline model.

When receiving a zero initialized speaker code, the SAT-MISC method achieves a relative 3.6% WER reduction compared with the baseline model. This result shows that the performance of our acoustic model has been improved. In this method, the model has an initial speaker code, and it is a part of speaker independent parameters. When we fed the model with the initial speaker code, it achieves a relative 4.3% WER reduction compared with the baseline model. If we get an adapted speaker code based on this initial speaker code, we will get a result of a relative 5.8% WER reduction compared with the baseline model.

Table 2: *Performance of the proposed method on SWB.*

Method	**WER**	**WERR**
baseline	13.8	–
SC-baseline with adapted speaker code	13.5	2.2%
SC-baseline with zero speaker code	13.7	0.7%
SAT-ZISC with zero speaker code	13.3	3.6%
SAT-ZISC with adapted speaker code	13.2	4.3%
SAT-MISC with zero speaker code	13.3	3.6%
SAT-MISC with initial speaker code	13.2	4.3%
SAT-MISC with adapted speaker code	13.0	5.8%

The results show that both SAT-ZISC and SAT-MISC are able to get a better speaker independent acoustic model. This can be considered as the benefit of meta-learning. In the process of ASR, speaker independent acoustic model is a meta-knowledge across all speakers. Applying MAML to speaker adaptive training makes the model easier to extract this meta-knowledge. When using adapted speaker code, SAT-MISC is superior than SAT-ZISC. This result shows that learning from a good initialized speaker code is better than learning from zero.

We also investigated the impact of the number of adaptation steps N in proposed methods. We find that SAT-ZISC need more steps to learn a useful speaker code than SAT-MISC. As Table 3 shows, SAT-ZISC need 12 adaptation steps to achieve best result while SAT-MISC need only 2 adaptation steps. This is a proof that learning from a good initialized speaker code is much easier than learning from zero. Table 3 also shows that more adaptation steps will not bring further improvement, and too many adaptation steps may degrade the performance of the model due to overfitting.

5. Conclusions

In this study, we have proposed two speaker code based speaker adaptive training methods with meta-learning approach. Both of the methods use the MAML algorithm. The speaker code is updated in the MAML's inner loop, and the speaker independent parameters are optimized in the MAML's outer loop. The results on the Switchboard task show that our methods not only learn a suitable speaker code, but also significantly improve the performance of the acoustic model. Both of the acoustic models of our method achieve a relative 3.6% WER reduction (WERR) compared with the baseline model. After using adapted speaker code, the SAT-ZISC method achieves a relative 4.3% WER reduction compared with the baseline, and the SAT-MISC method achieves a relative 5.8% WER reduction compared with the baseline. In future work, we plan to use more corpora to evaluate the effectiveness of SAT-ZISC and SAT-MISC extensively. Besides, based on MAML's model agnostic property, our approaches can be applied to a wide range of network architectures.

Table 3: *Impact of different adaptation steps on SWB.*

Method	**adaptation steps**	**WER**	**WERR**
baseline	–	13.8	–
SAT-ZISC	0	13.3	3.6%
	2	13.3	3.6%
	6	13.3	3.6%
	12	13.2	4.3%
	20	13.3	3.6%
	35	13.4	2.9%
SAT-MISC	0	13.2	4.3%
	2	13.0	5.8%
	6	13.2	4.3%

6. References

[1] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of Interspeech*, 2014, pp. 338–342.

[2] O. Abdel-Hamid, A. rahman Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Proceedings of ICASSP*, 2012, pp. 4277–4280.

[3] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. Lecun, "Very deep multilingual convolutional neural networks for lvcsr," in *Proceedings of ICASSP*, 2016, pp. 4955–4959.

[4] P. Karanasou, Y. Wang, M. J. Gales, and P. C. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proceedings of Interspeech*, 2014, pp. 2180–2184.

[5] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013, pp. 55–59.

[6] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 11, pp. 1938–1949, 2015.

[7] P. Cardinal, N. Dehak, Y. Zhang, and J. Glass, "Speaker adaptation using the i-vector technique for bottleneck features," in *Proceedings of Interspeech*, 2015, pp. 2867–2871.

[8] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *Proceedings of ICASSP*, 2013, pp. 7942–7946.

[9] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid dnn/hmm model for fast speaker adaptation in lvcsr based on speaker code," in *Proceedings of ICASSP*, 2014, pp. 6389–6393.

[10] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[11] X. Cui, V. Goel, and G. Saon, "Embedding-based speaker adaptive training of deep neural networks," in *Proceedings of Interspeech*, 2017, pp. 122–126.

[12] Y. Zhao, J. Li, S. Zhang, L. Chen, and Y. Gong, "Domain and speaker adaptation for cortana speech recognition," in *Proceedings of ICASSP*, 2018, pp. 5984–5988.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *JMLR Workshop and Conference Proceedings*, vol. 37, pp. 1180–1189, 2015.

[14] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proceedings of NIPS*, 2016, pp. 343–351.

[15] Z. Meng, J. Li, Y. Gong, and B. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *Proceedings of ICASSP*, 2018, pp. 5949–5953.

[16] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B. Juang, "Speaker-invariant training via adversarial learning," in *Proceedings of ICASSP*, 2018, pp. 5969–5973.

[17] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, and et al., "English conversational telephone speech recognition by humans and machines," in *Proceedings of Interspeech*, 2017.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[19] J.-Y. Hsu, Y.-J. Chen, and H.-y. Lee, "Meta learning for end-to-end low-resource speech recognition," *arXiv preprint arXiv:1910.12094*, 2019.

[20] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, vol. 2, p. 2, 2018.

[21] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of ICASSP*, 1992, pp. 517–520.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *CoRR*, 2014.