IMPROVING ACCURACY OF NONPARAMETRIC TRANSFER LEARNING VIA VECTOR SEGMENTATION

Vincent Gripon, Ghouthi B. Hacene
IMT Atlantique
Electronics Dept.

Matthias Löwe
University of Münster
Fachbereich Mathematik
und Informatik

Franck Vermet
Université de Brest
Laboratoire de Mathématiques

ABSTRACT

Transfer learning using deep neural networks as feature extractors has become increasingly popular over the past few years. It allows to obtain state-of-the-art accuracy on datasets too small to train a deep neural network on its own, and it provides cutting edge descriptors that, combined with nonparametric learning methods, allow rapid and flexible deployment of performing solutions in computationally restricted settings. In this paper, we are interested in showing that the features extracted using deep neural networks have specific properties which can be used to improve accuracy of downstream nonparametric learning methods. Namely, we demonstrate that for some distributions where information is embedded in a few coordinates, segmenting feature vectors can lead to better accuracy. We show how this model can be applied to real datasets by performing experiments using three mainstream deep neural network feature extractors and four databases, in vision and audio.

***Index Terms*—** transfer learning, deep neural networks, nearest neighbor search

1. INTRODUCTION

Transfer learning consists in training a learning method on a first dataset to be used on a second, distinct one. In this context, using Deep Neural Networks (DNNs) [1, 2, 3] (in particular convolutional neural networks on vision datasets) has become increasingly popular over the past few years. Indeed, the features extracted by state-of-the-art deep neural networks are so good that they allow, in some settings, to reach the best known accuracy when applied to other datasets and combined with simple classification routines. One of the key interest in using transfer learning methods is to avoid the heavy computational cost of training DNNs. Therefore, it is possible to exploit their accuracy on embedded devices such as small robots or smartphones [4, 5]. In this context, nonparametric methods such as k-Nearest Neighbors (k-NN) are particularly attractive for their ability to handle both class-incremental and example-incremental properties [6, 7].

It is interesting to point out that DNNs are trained to extract features well suited to perform a given classification task. In order for these features to become usable in other contexts (e.g. a new classification task), broad databases containing a large variety of classes should be used. As a direct consequence, it is expected that a significant part of the extracted feature vectors is useless for solving the new task at hand. Consequently, e..g., in the field of approximate nearest neighbor search for classification it is often observed that methods based on Product Quantization [8] and its derivatives can lead for certain choices of parameters to better performance than an exhaustive search on raw data. In these methods, the search space is split into quantized subspaces in which the search is performed independently.

In this paper we are interested in showing that, more generally, segmentation of feature vectors (obtained with pretrained DNNs) in multiple subvectors for which the search is performed independently can result in higher overall accuracy. We are interested in answering the following questions:

- Are there simple convincing mathematical models in which such improvement exists?
- How do such improvements depend on parameters?
- Does this apply to real world data?

To answer these questions, we describe the mathematical core of the classification procedure in Section 2. Section 3 contains an example where this procedure is not successful as well as a couple of situations where segmentation not only helps, but does provide the right class of the test pattern with probability converging to 1 as dimension becomes large, while a comparison of the Euclidean distance as well as a comparison coordinate by coordinate fails with probability at least one half. Section 4 contains experiments on real datasets. Section 5 is a conclusion.

2. CLASSIFICATION BY SEGMENTATION

In this section we will give a mathematical framework of the procedure we have in mind. We start with classes of data

$\mathcal{C}_1, \ldots, \mathcal{C}_K \subseteq \mathbb{R}^d$, where we always assume that the dimension d is a large parameter. To simplify matters assume that all of these classes have M elements. Now choose c, such that c divides d and write $\mathbb{R}^d = \bigotimes_{j=1}^{c} \mathbb{R}^{d/c}$. The $j'th$ of these subspaces will also be denoted by $(\mathbb{R}^{d/c})_j$. For each j take a dictionary D_j of nK segments such that D_j contains n segments of each class $\mathcal{C}_k$ drawn uniformly at random. Thus, we write each word $w \in \mathcal{C}_k$, $k = 1, \ldots, K$ as $w = w_1 \circ \ldots \circ w_c$, where each of the $w_j \in (\mathbb{R}^{d/c})_j$ and "$\circ$" denotes concatenation. For each class $\mathcal{C}_k$, $k = 1, \ldots, K$ we pick n segments of words w_j in $\mathcal{C}_k$ uniformly at random without replacement and put them into the dictionary D_j. Given a fresh word z we parse it in the same way into $z = z_1 \circ \ldots \circ z_c$. Then for each $j = 1, \ldots, c$ we find

$$\tilde{w}_j := \text{argmin}\{||w_j - z_j||_{d/c} : w_j \in D_j\}. \qquad (1)$$

Here $||\cdot||_{d/c}$ denotes Euclidean distance in $\mathbb{R}^{d/c}$. Let the random variable U_j take the value k, if $\tilde{w}_j$ is the j'th segment of a word $w \in \mathcal{C}_k$. If several words minimize the distance in (1) we let U_j take any of the values of the classes corresponding to these words with equal probability, as a tie-breaking rule. Finally one takes

$$\chi(z) := \text{argmax}_k \sum_{j=1}^{c} \{\mathbb{I}_{U_j = k}, k = 1, \ldots K\}$$

hence the class that is most often found by the above procedure. Again we add a tie-breaking rule, if this class is not unique. The c-segmentation procedure assigns z the class $\chi(z)$.

3. SITUATIONS WHERE SEGMENTATION IS OR IS NOT FAVORABLE

3.1. When segmentation does not help

We start with an example that shows that the data need to have a special structure for the segmentation technique to be supportive. In this subsection we will assume that we only have two classes $\mathcal{C}_1$ and $\mathcal{C}_2$. These are built in the following way. Take $m_1, m_2 \in \{-1, +1\}^d$ uniformly at random as "base vectors" of the two classes. Assume $0 < \varrho < \frac{1}{2}$. Then for $\mu = 1, \ldots, M$ let $Y^\mu, \tilde{Y}^\mu \in \{-1, +1\}^d$ be i.i.d. vectors with i.i.d. coordinates such that $\mathbb{P}(Y_1^1 = 1) = 1 - \varrho$ and $\mathbb{P}(Y_1^1 = -1) = \varrho$. We define $\mathcal{C}_1 := \{Y^\mu \times m_1, \mu = 1, \ldots, M\}$ and $\mathcal{C}_2 := \{\tilde{Y}^\mu \times m_2, \mu = 1, \ldots, M\}$. Here the multiplication is pointwise, i.e. $(Y^\mu \times m_1)_i := Y_i^\mu m_{1,i}$. Also let us assume that for $c \in \mathbb{N}$ such that c divides d and each $1 \leq j \leq c$ we take a dictionary D_j consisting of two segments, one, w_j^1, belonging to class $\mathcal{C}_1$, one, w_j^2, belonging to class $\mathcal{C}_2$, only. This will help to facilitate computations. In this setting we claim that c-segmentation does not improve the accuracy of the naive Euclidean approach.

Proposition 3.1 *In the above situation assume that c and d/c are odd (to avoid the discussion of tie-breaks) and that w is distributed like a word from $\mathcal{C}_1$ (but independent of all words in all classes). Then there is a number $I > 0$ such*

$$\lim_{d \to \infty} \frac{1}{d} \log \mathbb{P}(w \textit{ is classified } \mathcal{C}_2 \textit{ with } c = 1) = -I.$$

Whilst

$$\lim_{d \to \infty} \frac{1}{d} \log \mathbb{P}(w \textit{ is classified } \mathcal{C}_2 \textit{ with } 1 \ll c \ll d) = -I/2.$$

Proof: Refer to the extended version [9]. □

3.2. When segmentation does help

In the previous paragraph we saw that there are natural situations where the simplest case, when one does not partition vectors at all, is the best. However, the situation described there is close to a situation where using pieces of size $1 \ll c \ll d$ not only gives a better result than $c = 1$ and $c = d$, but also the results for the latter choice are useless.

We will start by describing a basic situation and then discuss possible extensions. All these models are influenced by the observation that the data classified in [10] seem to suffer from occasional large outliers.

Our first basic situation will be given by $K = 2$ classes $\mathcal{C}_1$ and $\mathcal{C}_2$ where

$$\mathcal{C}_1 := \{Y^\mu + m_1, \mu = 1, \ldots, M\}$$
$$\text{and} \quad \mathcal{C}_2 := \{\tilde{Y}^\mu + m_2, \mu = 1, \ldots, M\}.$$

This time (for the sake of keeping things easy), $m_1 = (0, \ldots, 0)$ and $m_2 = (1, \underbrace{0, \ldots, 0}_{l-1 \text{ times}}, 1, \underbrace{0, \ldots, 0}_{l-1 \text{ times}}, 1 \ldots 0)$ i.e. m_2 has a 1 at regular positions. Moreover the Y^μ, and $\tilde{Y}^\mu$ are i.i.d vectors in $\mathbb{R}^d$ such that $\mathbb{P}(Y_1^1 = N) = \mathbb{P}(\tilde{Y}_1^1 = N) = p = 1 - \mathbb{P}(Y_1^1 = 0) = 1 - \mathbb{P}(\tilde{Y}_1^1 = 0)$ and p and N will be chosen in the sequel.

Given c for each segment $1 \leq j \leq c$ we will again take a short dictionary D_j consisting of one segment w_j^1 of a word from $\mathcal{C}_1$ and one segment w_j^2 from a word of $\mathcal{C}_2$. Assume again we want to classify a word w that is distributed like a word from $\mathcal{C}_1$ (but independent of all words in all classes). We start with the observation, that for small l and small p the coordinate by coordinate comparison will fail.

Proposition 3.2 *Assume that $l \leq d^{1/4}$, $N > 0$, and that $p \leq \frac{1}{l}$. Then for $c = d$, i.e. the coordinate by coordinate comparison,*

$$\mathbb{P}(w \textit{ is classified correctly}) \to \frac{1}{2} \quad \textit{as } d \to \infty. \qquad (2)$$

Remark 3.3 *Observe that the situation described in* (2) *is a worst case scenario when one has two classes only. Indeed if the probability on the right were even smaller than $\frac{1}{2}$ one could use the reverse method and decide just the opposite of the proposed classification to get a better result.*

Proof: Due to the independence of the random parts of the coordinates of the words, we may assume without loss of generality that all the segments from class one in D_j stem from the same word $w^1 = m_1 + Y$, all the segments from class two stem from the same word $w^2 = m_2 + \tilde{Y}$. Moreover, we write $z = m_1 + \overline{Y}$. Consider the set

$$S := \{i : \exists n, i = 1 + nl \text{ or } Y_i = N \text{ or } \tilde{Y}_i = N \text{ or } \overline{Y_i} = N\}$$

According to our assumptions, $|S| \leq d^{1/3}$ with high probability, i.e. with probability converging to 1 as $d \to \infty$ and thus $|\{1, \ldots, d\} \setminus S| \geq d - d^{1/3}$ with high probability. But for all coordinates i in $\{1, \ldots, d\} \setminus S$ one has that $|w_i - w_i^1| = |w_i - w_i^2|$, therefore the tie-breaking rule decides with probability one half for class $\mathcal{C}_1$. As the fluctuations of this random decision by the Central Limit Theorem are of order $\sqrt{d}$ and therefore larger than any "signal" one might obtain from S, the statement follows. □

But also the Euclidean distance $c = 1$ fails as a classification rule for a wide range of parameters.

Proposition 3.4 *If $l \to \infty$ and $p \gg \max(\frac{1}{d}, \frac{1}{N^2 l})$, we have for the Euclidean distance rule $c = 1$*

$$\mathbb{P}(w \textit{ is classified correctly}) \to \frac{1}{2} \quad \textit{as } d \to \infty. \tag{3}$$

Proof: Refer to the extended version [9]. □

The question remains, of course, whether there is any segmentation method that works in this case. Fortunately, the answer is yes.

Proposition 3.5 *If $l \ll \frac{1}{p}$ the c-segmentation rule with $c = d/l$ works, more precisely*

$$\mathbb{P}(w \textit{ is misclassified}) \to 0, \quad \textit{as } l \to \infty \textit{ and } d \to \infty.$$

Proof: Refer to the extended version [9]. □

Altogether we have shown

Theorem 3.6 *In the model described above assume that $d, l \to \infty$, $l \leq d^{\frac{1}{4}}$, $p \ll \frac{1}{l}$, but $p \gg \max(\frac{1}{d}, \frac{1}{N^2 l})$, then for $c = d$ and $c = 1$ we have $\mathbb{P}(w \textit{ is misclassified}) \to 1/2$, while for $c = d/l$, we have $\mathbb{P}(w \textit{ is misclassified}) \to 0$.*

Up to now we have just discussed the basic example to illustrate which statistical properties of the classes favor segmentation in the classification process. Let us comment on some variants of the above model. A first natural extension of the model is to consider more than two classes. In the above setting it is obvious that we can build up to $l+1$ classes, where again $\mathcal{C}_1 := \{Y^{1,\mu} + m_1, \mu = 1, \ldots, M\}$ with $m_1 = 0$, and for $k = 2, \ldots, K$, $\mathcal{C}_k := \{Y^{k,\mu} + m_k, \mu = 1, \ldots, M\}$, where all the $(Y^{k,\mu})$ are i.i.d. random vectors in $\mathbb{R}^d$ with $\mathbb{P}(Y_i^{k,\mu} = N) = p = 1 - \mathbb{P}(Y_i^{k,\mu} = 0)$ and the vectors $(m_k)_{k=2,\ldots,K}$ are concatenations of strings of length l, such that for each of these strings contains all coordinates but one are 0, the remaining coordinate is 1, and the 1s are placed at different positions for different m_k. Then Theorem 3.6 translates to

Theorem 3.7 *In the model described above assume that $d, l \to \infty$, $l \leq d^{\frac{1}{4}}$, $p \ll \frac{1}{l}$, but $p \gg \max(\frac{1}{d}, \frac{1}{N^2 l})$, then for $c = d$ and $c = 1$ we have $\mathbb{P}(w \textit{ is misclassified}) \to 1/K$ while for $c = d/l$, we have $\mathbb{P}(w \textit{ is misclassified}) \to 0$.*

Altogether the case of two classes is generic ([9]). Therefore we will discuss other variants of the model for this case only.

Another obvious modification of the model at the beginning of this subsection one might discuss is the influence of a larger dictionary. So let us assume now that the dictionaries D_j contain ν segments of each class, i.e. $D_j := \{w_j^{1,1}, \ldots w_j^{1,\nu}, w_j^{2,1}, \ldots w_j^{2,\nu}\}$ and again the words are of the form $w^{1,\mu} := m_1 + Y^{1,\mu}$, $w^{2,\mu} := m_2 + Y^{2,\mu}$. Again we will check whether Propositions 3.2 to 3.5 remain true. For the proof of Proposition 3.2 we let $w = m_1 + Y$ be distributed as a word from class one. Define the set S in the proof of Proposition 3.2 now as the set of coordinates that are not of the form $1 + jl$ and such that none of the Bernoulli's in the dictionaries is 1. Then again $|S| \leq d^{1/3}$ with high probability, which implies that Proposition 3.2 remains true. However, the behaviour of the Euclidean distance rule Proposition 3.4 improves, if the dictionaries become larger. Indeed w is classified correctly, if there exists a word $w^{1,n} \in \{w^{1,1}, \ldots, w^{1,\nu}\}$ such that

$$\sum_i \mathbb{I}_{Y_i^{1,n} \neq Y_i} < \sum_i \mathbb{I}_{Y_i^{2,s} \neq Y_i} \quad \forall s = 1, \ldots, \nu.$$

The probability that this holds true for a fixed n is asymptotically $(\frac{1}{2})^\nu$. So the probability that such an n does not exist is given by $1 - (1 - (\frac{1}{2})^\nu)^\nu$, which is smaller than $1/2$ but for ν not depending on d still not 0. However, also the accuracy of the c-segmentation method with $c = d/l$ as in Proposition 3.5 improves and this basically for the same reasons: If there is one segment of a word of class $\mathcal{C}_1$ in the j'th dictionary such that all its Y-variables in this segment are 0, one classifies w correctly. And this probability, of course, increases, as ν becomes larger.

One might, of course, ask which features of the model discussed in this subsection are decisive for the c-segmentation method to be favorable. These features are:

a) The vectors in each class are *rare* but *large* perturbations of a base vector. Most of the coordinates of the base vectors of two distinct classes agree.

b) The perturbations are much rarer than the frequencies of the coordinates in which the base vectors disagree.

However, analyzing the data used in [10] one sees that our models above describe well the behavior of one class, but not that of two classes simultaneously. Indeed, there is some evidence, that the coordinates that take large values in a class are, also likely to take large values in another, but the variance is larger for those that take large values. To take this into

account, we change our original model in the following way:

$$\mathcal{C}_1 := \{N^\mu Y + m_1, \mu = 1, \ldots, M\}$$
$$\mathcal{C}_2 := \{\tilde{N}^\mu Y + m_2, \mu = 1, \ldots, M\}.$$

Moreover Y is a vector of i.i.d. Bernoulli random variables with parameter p, i.e. $\mathbb{P}(Y_1 = 1) = p = 1 - \mathbb{P}(Y_1 = 0)$. Finally the N^μ and $\tilde{N}^\mu$ are i.i.d. random variables in $\mathbb{R}^d$ with positive, i.i.d. components, such that $\mathbb{P}[N_1^\mu \geq a] = 1$, for some $a > 0$. Again we will take dictionaries D_j that only contain one segment w_j^1 and w_j^2 of each class and we want to classify a word w that is distributed as a word from $\mathcal{C}_1$ correctly. Assume that $w^1 = YN, w_2 = m_2 + Y\tilde{N}$ and $w = Y\overline{N}$. Again we obtain

Theorem 3.8 *In the model described above assume that* $d, l \to \infty$, $l \leq d^{\frac{1}{4}}$, $p \ll \frac{1}{l}$, *but* $p \gg \max(\frac{1}{d}, \frac{1}{a^2 l})$, *then for* $c = d$ *and* $c = 1$ *we have* $\mathbb{P}(w$ *is misclassified*$) \to 1/2$ *while for* $c = d/l$, *we have* $\mathbb{P}(w$ *is misclassified*$) \to 0$.

The proof is only a slight modification of the proof of Theorem 3.6.

4. EXPERIMENTS

In this section we derive experiments on real-world data. In our experiments, we use three distinct DNNs. Two of them are related to vision tasks and perform feature extraction from raw input images, namely Inception V3 [11] and SqueezeNet [5]. Both these networks have been trained using 1'000 classes from the ImageNet dataset. As far as Inception V3 is concerned, we use the features obtained before the first fully connected layer. It consists of a vector with 2'048 dimensions. The inputs are images scaled to 299x299 pixels. For the SqueezeNet network, we use the penultimate layer (containing 1'000 dimensions) as our feature extractor. Input images contain 227x227 pixels. The last DNN we use has been trained on AudioSet [12], a dataset that consists of more than 2'000'000 distinct audio tracks extracted from videos on YouTube. The extracted features contain 1'280 dimensions which are the concatenation of ten 128 dimensions feature vectors, one per second of the corresponding audio track.

We perform tests on four datasets: CIFAR10, two subsets of ImageNet made of 10 classes sampled randomly from those that where not used to train the DNNs, and a subset of 10 classes used to train AudioSet. More details are available in the extended version of this paper [9].

We use k-NN as our nonparametric method to obtain a classification accuracy. Note that in the case of c segments, the decision is taken using kc votes instead of k, since each subspace performs a k-NN. We observe that in all scenarios, the optimal solution corresponds to an intermediate number of segments. The case of AudioSet is interesting as there is a local maximum in accuracy which corresponds to 10 segments, which occurs when considering the ten 128 dimension feature vectors independently, but the global maximum is for 40 segments. Note that the complexity of the method does not depend on c, as both memory and number of operations boils down to the product of the number of training vectors and their dimension. Table 1 summarizes our results.

Inception V3, 1-NN

c	1	4	16	64	256
CIFAR10	0.8519	0.8652	**0.8781**	0.8651	0.8347
ImageNet1	0.9328	0.9354	0.9424	**0.9439**	0.9081
ImageNet2	0.9438	0.9451	**0.9524**	0.9464	0.9171

Inception V3, 5-NN

c	1	4	16	64	256
CIFAR10	0.8689	**0.8761**	0.8759	0.8668	0.8461
ImageNet1	0.9389	**0.9450**	0.9429	0.9394	0.9202
ImageNet2	0.9467	0.9498	**0.9511**	0.9488	0.9303

SqueezeNet, 1-NN

c	1	5	20	100	200
CIFAR10	0.6839	0.7069	**0.7472**	0.6890	0.6225
ImageNet1	0.8854	0.8900	**0.9001**	0.8784	0.8466
ImageNet2	0.8737	0.8802	**0.8926**	0.8669	0.8267

SqueezeNet, 5-NN

c	1	5	20	100	200
CIFAR10	0.7284	0.7483	**0.7566**	0.6954	0.6371
ImageNet1	0.8985	0.8965	**0.8980**	0.8698	0.8501
ImageNet2	0.8862	**0.8901**	0.8893	0.8591	0.8280

AudioSet

c	1	2	10	20	40	160
1-NN	0.605	0.621	0.704	0.698	**0.724**	0.660
5-NN	0.564	0.649	0.704	0.718	**0.727**	0.668

Table 1. Accuracy of classification, depending on the feature extractor used, the dataset, the number of segments c and the number of nearest neighbors k the decision is based upon. Best scores are marked in bold.

5. CONCLUSION

Transfer learning is a popular method to obtain cutting edge descriptors that can be exploited to classify new data. When combined with nonparametric methods such as k-nearest neighbor search, it provides a lightweight incremental solution that is suitable for devices with limited energy or computational capabilities. We have shown that segmenting vectors to perform k-nearest neighbor search in obtained subspaces can result in significant improvements in accuracy. Moreover, this change has no cost on memory usage neither on the number of operations required to fulfill the task. Interestingly, this method can be thought about as an alternative to increasing the number of neighbors k to consider when taking a decision.

Future work include considering other downstream classification techniques such as support vector machines and logistic regression.

6. REFERENCES

[1] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[4] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, et al., "Going deeper with embedded fpga platform for convolutional neural network," in *Proceedings of the ACM International Symposium on Field-Programmable Gate Arrays*, 2016, pp. 26–35.

[5] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[6] Robi Polikar, Lalita Udpa, Satish S Udpa, and Vasant Honavar, "Learn++: an incremental learning algorithm for multilayer perceptron networks," in *Proceedings of the IEEE conference on Acoustics, Speech, and Signal Processing*, 2000, vol. 6, pp. 3414–3417.

[7] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 31, no. 4, pp. 497–508, 2001.

[8] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[9] V. Gripon, G. B. Hacene, M. Löwe, and F. Vermet, "Improving Accuracy of Nonparametric Transfer Learning via Vector Segmentation," *ArXiv e-prints, 1710.08637*, Oct. 2017.

[10] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael Rabbat, and Hervé Jégou, "Memory vectors for similarity search in high-dimensional spaces," *IEEE Transactions on Big Data*, 2017, In press.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[12] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE ICASSP*, 2017.