Novelties in Intelligent Digital Systems
C. Frasson et al. (Eds.)

doi:10.3233/FAIA210090

ActiveCrowds: A Human-in-the-Loop Machine Learning Framework

Lazaros TOUMANIDIS [a,1], Panagiotis KASNESIS [a],
Christos CHATZIGEORGIOU [a], Michail FEIDAKIS [a] and
Charalampos PATRIKAKIS [a]

[a] *University of West Attica, dept. of Electrical & Electronics Engineering, Egaleo, Greece*

Abstract. A widespread practice in machine learning solutions is the continuous use of human intelligence to increase their quality and efficiency. A common problem in such solutions is the requirement of a large amount of labeled data. In this paper, we present a practical implementation of the human-in-the-loop computing practice, which includes the combination of active and transfer learning for sophisticated data sampling and weight initialization respectively, and a cross-platform mobile application for crowdsourcing data annotation tasks. We study the use of the proposed framework to a post-event building reconnaissance scenario, where we utilized the implementation of an existing pre-trained computer vision model, an image binary classification solution built on top of it, and max entropy and random sampling as uncertainty sampling methods for the active learning step. Multiple annotations with majority voting as quality assurance are required for new human-annotated images to be added on the train set and retrain the model. We provide the results and discuss our next steps.

Keywords. Active Learning, Transfer Learning, Crowdsourcing, Mobile Computing

1. Introduction

There is no doubt that Machine Learning (ML), as a field of Artificial Intelligence (AI) and consequently as part of the 4th Industrial Revolution [1], has changed our work, our social interactions, our life in general. One of the major bottlenecks and open research topic in ML is the massive data collection and labeling requirement, which is essential for training supervised learning models. Towards, Transfer Learning (TL) constitutes a technique that reduces the amount of data that need to be used during the training and testing process. The objective of TL is to take advantage of data from an existing setting to extract information that may be useful when learning or even when making predictions in a different setting [2].

[1] Corresponding Author: Toumanidis Lazaros, University of West Attica, dept. of Electrical & Electronics Engineering, Egaleo, Greece; E-mail: laztoum@uniwa.gr.

In the case of supervised learning, further reduction in the data labeling process is achieved using Active Learning (AL) that reduces data labeling effort by actively selecting the most informative instances to be queried for labeling. In contrast with Passive Learning, where all labeled samples are obtained once and without a reference to the learning algorithm, new samples are interactively chosen from a pool of unlabeled data that may improve the learning process resulting on the reduction of the total amount of samples needed.

Crowdsourcing is a common technique for both data acquisition and data labeling, where a large group of people, not necessarily domain experts, are asked to either provide or label existing samples. The term crowdsourcing appeared in [3] as a paradigm, in which a specific service, information or task is offloaded to a crowd of individuals, often connected through a common goal or interest, as in an online community.

Using the above three techniques, i.e. TL, AL, and Crowdsourcing, we have implemented an end-to-end solution that can be applied in several domains in supervised ML, including computer vision, text and audio processing. Also, it can be easily used on-field to feed any ML pipeline by exploiting a user-friendly mobile application. Tasks are initialized with existing models using TL, and new samples are appended to the training data set using AL. Crowdsourcing is used for the labeling step of each task, by deploying a mobile application as an interface. In this paper, we apply our solution to post-event building reconnaissance use case, demonstrating its capabilities on reducing the amount of annotated data needed. In particular, we selected a binary image classification task using ResNet [4] as the initial model, a smaller image dataset with images of buildings that are either collapsed or not [5], and uncertainty sampling for querying new samples.

2. Related Work

TL techniques have been successfully applied in real-world applications [6] in several domains, including Natural Language Processing [7,8], Computer Vision [9,10], and Audio Processing [11,12]. In [13], Yang et al. present a study on the combination of TL and AL, focusing on the improvement of the learning accuracy, concerning the size of the required labels. Zhou et al. [14] present active fine-tuning (AFT), a new algorithm which naturally integrates AL and TL. Their results include a significant cut in the annotation cost compared to the state-of-the-art method [15]. High detection accuracy is achieved in [16], where Feng et al. use a deep residual network for defect detection and classification in infrastructure surface images, and apply AL strategy, asking experts to label the most informative subset of new images to retrain the network. Tong and Koller in [17] propose the use of Support Vector Machines (SVM) for samples selection in AL. By applying the latter to text classification, they achieve a significant reduction in the need for labeled instances.

There have been several studies on measuring and ensuring the quality of crowdsourced data. Wang et al. in [18] propose ARTSense, a framework to solve the problem of “trust without identity” in mobile sensing. The solution consists of a privacy-preserving provenance model, a data trust assessment scheme, and an anonymous reputation management protocol. Tian et al. have presented Max-Margin Majority voting as a new approach for improving the discriminative ability of majority voting in crowdsourcing [19].

Regarding the combination of AL and Crowdsourcing for data labeling, Fang et al. focus on the optimization of the sampling techniques in [20]. Zhang et al. propose a deep computational model with crowdsourcing for industrial IoT Big Data, trying to prevent overfitting and aggregate adequately labeled samples to train a model's parameters [21]. Song et al. propose a confidence-based Crowdsourcing approach for data labeling, in which the confidence of the crowd workers has been considered for aggregating the results [22]. A combination of the learning algorithm uncertainty, and the uncertainty derived from the crowdsourced answers, is used to define a score function, for new instances selection. Zhao et al. study the use of AL and Crowdsourcing for activity recognition [23]. In their work, they present three methods to choose the most informative data points based on low confidence for the most probable activity class, the minimum difference between the confidence of the most and second most probable class, and high entropy among the probability of classes. A combination of ontological knowledge and AL is presented in Civitarese et al. [24], in which users' feedback is deployed to refine the correlations among sensor events and activity types initially extracted from a high-level ontology, in order to mine temporal patterns of sensor events that are frequently generated by the execution of specific activities.

In the domain of sound processing, MoodSwings [25] and TagAtune [26] are used for data labeling with mobile applications. The former is focusing on audio mood labeling, and the latter, on tagging music albums. A more generic approach that includes several tasks is provided in 'Crowdsource' [2] for Android and in 'Unbiased WorkForce' for Android[3] and iOS[4].

In this paper, we present an implementation that includes the three techniques mentioned (TL, AL, Crowdsourcing) above into one extendable platform, providing both the means for training new models, and a user interface for crowdsourcing the data labeling requirement.

3. ActiveCrowds

In this section, we present 'ActiveCrowds', a platform that can be used for managing ML related tasks, utilizing the aforementioned techniques (TL, AL, Crowdsourcing). It allows ML tasks to be carried out, utilizing the advantages of each technique. It aims at providing a flexible tool that could be used in several domains and extended with new implementations of either ML or crowdsourcing techniques.

3.1. Overview

A high-level overview of a task is displayed in Figure 1. First, a model initialization is required, using an existing pre-trained model. After collecting enough

[2] https://play.google.com/store/apps/details?id=com.google.android.apps.village.boond

[3] https://play.google.com/store/apps/details?id=io.datax.workstation

[4] https://apps.apple.com/tt/app/unbiased-workforce/id1502361378

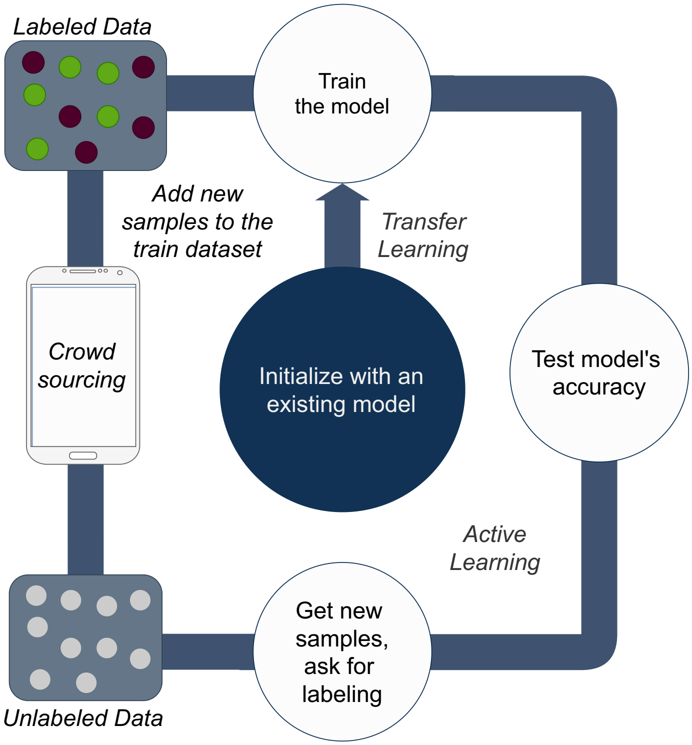

Figure 1. Overview

samples, we train and validate the model; we compute its accuracy on a test data set, we get new samples using one of the available sampling strategies and ask humans for labeling. The new samples that are labeled using the mobile application are added to the training set, and another loop of training–validation–testing sampling begins.

The platform is split into three main modules: (a) The core module related with the training, evaluation and testing of the available models, as well as the process of sampling entries to be labeled. (b) A web service acting as an interface to the core module. (c) A mobile application that is used from the users to complete the labeling tasks. For the core module, we have implemented a Python package that includes an abstract base class for the train, evaluation, testing of a model, and the sampling process. An image binary classification class implements the above methods on top of Torchvision and PyTorch [27]. We have also developed a storage class, responsible for handling the relevant datasets, and a scenario class that is used to hold information about the train–evaluation–test sample loop parameters.

Read and write access to the machine learning tasks is available through a REST API implemented using the Flask framework. The task parameters and their current state are stored in a PostgreSQL database. Background tasks that are triggered using the API, are performed using the Celery library with Redis being used as the message broker and the results' back-end storage. The mobile

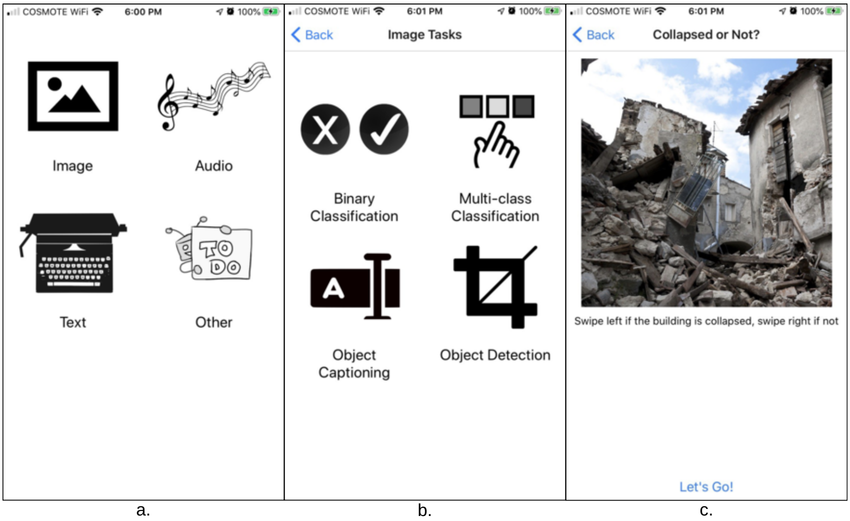
COSMOTE WiFi 6:00 PM 100%
Image
Audio
Text
Other
COSMOTE WiFi 6:01 PM 100%
Back
Image Tasks
Binary Classification
Multi-class Classification
Object Captioning
Object Detection
COSMOTE WiFi 6:01 PM 100%
Back
Collapsed or Not?
Swipe left if the building is collapsed, swipe right if not
Let's Go!
a.
b.
c.

Figure 2. Mobile application screens: a) Task categories, b) Computer Vision tasks and c) Binary Classification task

application is implemented using the React Native framework, making it available to run on both Android and iOS based devices. A task initialization requires:

- The pre-trained model to be used,
- The required dataset sizes for the train, evaluation, and testing steps,
- The query strategy to be used for new samples,
- The minimum number of peer answers and the minimum required accuracy for the labeling decision

3.2. Implementation

An instance of the Scenario class consists of:

- The required length of samples to be used for the active learning phases, i.e., training, evaluation, testing, sampling, sampling strategy (currently one of: *Minimum Margin*, *Least Confidence* and *Max Entropy*),
- The category of the task that is image, audio, text or other,
- The task's subcategory, which the case of an image category may be a binary or multi-class classification, object detection or object captioning.

The REST API currently includes endpoints for the available tasks listing, new task creation, getting task details by its id or updating or deleting one, and a task's labeling answers submission endpoint. On each submission, the task's status is updated, the criterion about deciding for a new label is checked, and if applicable a new active learning loop is started. The mobile application, as depicted in Figure 2 consists of two main screens, (a) the lists of the task categories and subcategories respectively are presented after being fetched from the server, and (b) the tasks' specific screens. In the case of image binary classification, a cover image and a small description are presented, and a batch of the selected samples that need to be labeled are sequentially presented. After completing the batch labeling, the user answers are sent to the server, and the task is marked as locked. When new samples are available, the task is again available for a new submission.

4. Use Case Scenario

The proposed framework has been applied in a post-event building reconnaissance use case scenario. In such an event, a tremendous amount of perishable visual data can be generated in just a few days. As a result, data annotation from trained professional engineers about whether a building has collapsed or not, is time-consuming; it might take days to be completed, so there is a need of an autonomous classification tool [5]. Additionally, to train a deep Convolutional Neural Network (CNN) efficiently, a huge amount of annotated data is required. Therefore, we study the use of our tool in a simulated scenario, in which an earthquake has taken place damaging a lot of buildings.

In more technical terms, we study a binary classification problem, which deals with the prediction on whether a depicted building is collapsed or not. The dataset we used consists of 1850 images of collapsed buildings and 3420 of non-collapsed, for a total of 5270 images. Non-collapsed images mainly consist of undamaged buildings, damaged buildings, and irrelevant pictures, which represents a typical data set collected during an earthquake reconnaissance mission. It has been successfully used on a post-event building reconnaissance study in [5] with a collapse classification accuracy of about 91.5%, trained using the AlexNet CNN [28]. 1000 of them were used for validation and testing of the trained model, and the rest of them, as a pool of new samples to select from, for the AL sampling step (Figure 3, samples of the training dataset).

The task is created using the above dataset, while the initial parameters are set as follows:

- A team of 20 trained professional engineers investigates the area where the earthquake has taken place
- We have an initial training set of 30 images
- A fixed number of 30 images to be appended on this set for a train-validation-test loop of the model, and
- A peer accuracy of 80% with a minimum number of required peers (end-users) of 10, meaning that to decide for a class of an unlabeled image, input from at least 10 peers is required, and the 80% of the answers have to be the same for this image.

The server and the core modules are hosted on a computer workstation equipped with an NVIDIA GTX Titan X GPU, featuring 12 gigabytes RAM, 3072 CUDA cores, and bandwidth of 336.5 GB/s. We used Python as the programming language, and specifically the PyTorch library. To accelerate the tensor multiplications, we used CUDA Toolkit also supported by the cuDNN, which is the NVIDIA GPU-accelerated library for deep neural networks. The software is installed on a Linux operating system.

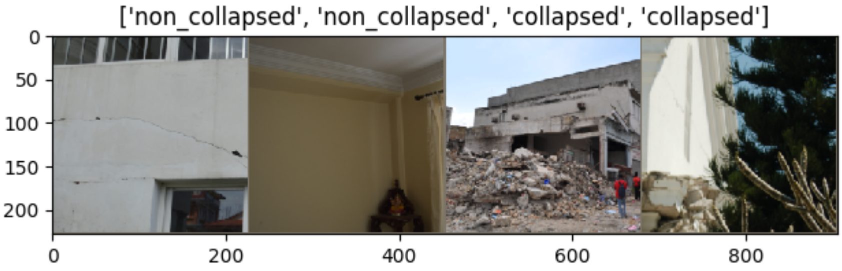
['non_collapsed', 'non_collapsed', 'collapsed', 'collapsed']
0
50
100
150
200
0
200
400
600
800

Figure 3. Train Preview

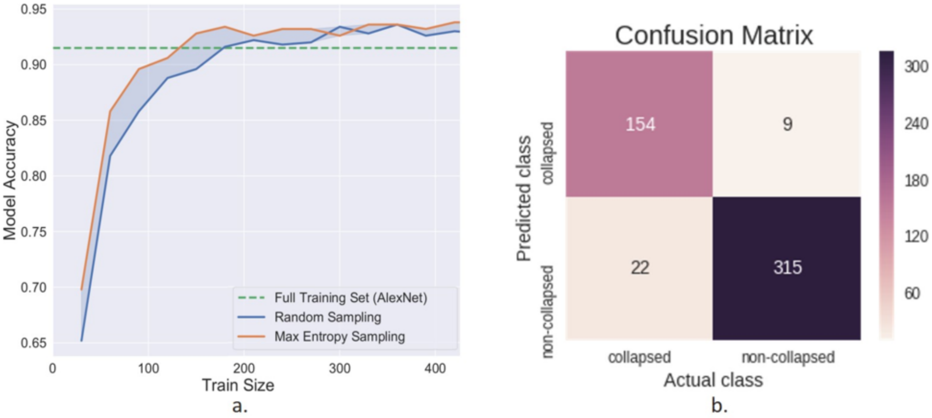

Figure 4. Obtained results: a) model accuracy and b) confusion matrix

Margin sampling and random sampling were deployed for the labelling of new images. As expected, testing the least confidence margin and entropy sampling methods gave precisely the same results with margin sampling, since this is a binary classification problem, and these three methods provide the same results in this case. We achieved an accuracy of about 93.5% with a training size of fewer than 420 images (Figure 4a). Moreover, it is worth mentioning that by exploiting the AL feature of our framework, we achieved almost 1% higher accuracy (92.47%) than the one reported in [5] (91.5%) using only 150 labeled images, which is around 0.09% of the labeled used in [5]. Testing the model on the 500 samples of the test set, gave us a result of 315 true-negative and 154 true-positive predictions presented in the form of a confusion matrix (Figure 4b).

5. Conclusions

This paper presents a novel framework that can be used in ML related tasks, utilizing the methods of Transfer Learning, Active Learning and Crowdsourcing. We successfully deployed and evaluated the platform using a post-earthquake building reconnaissance use case and by setting an image binary classification problem of collapsed building identification, achieving State-of-the-Art results, while using only few labeled instances. We expect that the proposed framework will be a breakthrough in domains where data labelling is both vital and time-consuming. Our next steps include the evaluation of more tasks both in the computer vision domain and in other domains like audio and text processing. Moreover, we intend to explore more active learning query strategies, as well as more methods for the quality assurance of the crowdsourced labels.

Acknowledgment

We would like to gratefully acknowledge Chul Min Yeum Shirley J. Dyke and Julio Ramirez from the Lyles School of Civil Engineering Purdue University for sharing the dataset which was used in our experiments, and the support of NVIDIA for a Titan X GPU card donation that has been used in this research.
This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EDK-03487).

References

[1] K. Schwab, *The fourth industrial revolution.* Currency, 2017.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series).* MIT Press, 2016.

[3] J. Howe, “The rise of crowdsourcing,” *Wired magazine*, vol. 14, pp. 1–4, June 2006.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 6 2016.

[5] C. M. Yeum, S. J. Dyke, and J. Ramirez, “Visual data classification in post-event building reconnaissance,” *Engineering Structures*, vol. 155, pp. 16–24, 2018.

[6] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, “Transferring naive bayes classifiers for text classification,” in *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI’07, pp. 540–545, AAAI Press, 2007.

[8] X. Ling, G.-R. Xue, W. Dai, Y. Jiang, Q. Yang, and Y. Yu, “Can chinese web pages be classified with english data source?,” in *Proceedings of the 17th International Conference on World Wide Web*, WWW ’08, (New York, NY, USA), pp. 969–978, Association for Computing Machinery, 2008.

[9] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, “Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection,” *Construction and Building Materials*, vol. 157, pp. 322–330, Dec. 2017.

[10] A. Kensert, P. J. Harrison, and O. Spjuth, “Transfer learning with deep convolutional neural networks for classifying cellular morphological changes,” *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, vol. 24, no. 4, pp. 466–475, 2019.

[11] W.-N. Hsu, D. Harwath, and J. Glass, “Transfer learning from audio-visual grounding to speech recognition,” *arXiv preprint arXiv:1907.04355*, 2019.

[12] M. E. A. Elshaer, S. Wisdom, and T. Mishra, “Transfer learning from sound representations for anger detection in speech,” *arXiv preprint arXiv:1902.02120*, 2019.

[13] L. Yang, S. Hanneke, and J. Carbonell, “A theory of transfer learning with applications to active learning,” *Machine Learning*, vol. 90, pp. 161–189, Feb. 2013.

[14] Z. Zhou, J. Shin, R. Feng, R. T. Hurst, C. B. Kendall, and J. Liang, "Integrating active learning and transfer learning for carotid intima-media thickness video interpretation," *Journal of Digital Imaging*, vol. 32, pp. 290–299, Apr. 2019.

[15] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[16] C. Feng, M. Y. Liu, C. C. Kao, and T. Y. Lee, "Deep active learning for civil infrastructure defect detection and classification," in *Computing in Civil Engineering 2017* (N. El-Gohary, P. Tang, N. El-Gohary, K.-Y. Lin, K.-Y. Lin, and P. Tang, eds.), pp. 298–306, American Society of Civil Engineers (ASCE), Jan. 2017.

[17] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*, pp. 107–118, ACM, 2001.

[18] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 13, pp. 2777–2790, Dec. 2014.

[19] T. Tian, J. Zhu, and Y. Qiaoben, "Max-margin majority voting for learning from crowds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2480–2494, Oct. 2019.

[20] M. Fang, J. Yin, and D. Tao, "Active learning for crowdsourcing using knowledge transfer," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pp. 1809–1815, AAAI Press, 2014.

[21] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, "An adaptive dropout deep computation model for industrial iot big data learning with crowdsourcing to cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2330–2337, 2019.

[22] J. Song, H. Wang, Y. Gao, and B. An, "Active learning with confidence-based answers for crowdsourcing labeling tasks," *Knowledge-Based Systems*, vol. 159, pp. 244–258, 2018.

[23] L. Zhao, G. Sukthankar, and R. Sukthankar, "Robust active learning using crowdsourced annotations for activity recognition," in *Proceedings of the 11th AAAI Conference on Human Computation*, AAAIWS'11-11, pp. 74–79, AAAI Press, 2011.

[24] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, and H. Stuckenschmidt, "newnectar: Collaborative active learning for knowledge-based probabilistic activity recognition," *Pervasive and Mobile Computing*, vol. 56, pp. 88–105, 2019.

[25] Y. Kim, E. Schmidt, and L. Emelle, "Moodswings: A collaborative game for music mood label collection.," in *ISMIR*, pp. 231–236, 01 2008.

[26] E. L. M. Law, L. V. Ahn, R. B. Dannenberg, and M. Crawford, "Tagatune: A game for music and sound annotation," 2007.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (Red Hook, NY, USA), pp. 1097–1105, Curran Associates Inc., 2012.