EXPLORING AUDIO EVENT DETECTION PIPELINES FOR URBAN SOUND CLASSIFICATION

Technical Report

Harris Nisar, Krishna Subramani, Ragini Gupta

University of Illinois Urbana-Champaign

ABSTRACT

”Machine Listening” aims to endow computers with the skills to understand and interpret sounds. This is simpler said than done. The main challenge with analyzing audio is the scarcity of properly annotated datasets. The introduction of SONYC-UST-V2 [1] has allowed exploration of learning based audio event detection pipelines. This dataset additionally provides Spatio-temporal context (STC) data to analyze if its inclusion improves classification performance. We study the baseline system used in classifying this dataset and analyze the effect of modifying 2 aspects of the baseline: (1) Transfer Learning and (2) Data Augmentation. In addition, we explore the use of simpler models like support vector machines (SVM). For completeness, we make available our repository to reproduce the results[1].

***Index Terms*—** Audio Event Detection, Transfer Learning, Data Augmentation, Spatio-temporal Context Data

1. INTRODUCTION

Audio event detection (AED) systems can be broadly classified as monophonic or polyphonic [2]. Monophonic systems take as input audio containing single classes and try to predict the class. Polyphonic systems take as input audio containing multiple classes (possibly overlapping) and predict the presence (and possibly the timestamp) of said classes. Figure 1 shows the generic structure for an AED system.

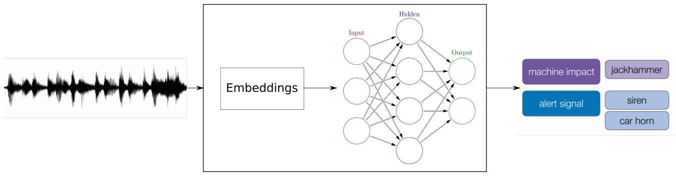


Figure 1: Generic AED System

The input representation can be the waveform, spectrum or mel-frequency spectrum. Embeddings are some aggregated features representation of the input, and a classifier classifies these embeddings into classes. Both steps can be combined to build an end-to-end system, such as a deep neural network.

With the advent of data-driven methods and computing power, researchers have shifted to machine learning approaches for AED. Audio suffers from one problem in contrast to the Computer Vision research community - lack of properly annotated large scale datasets. The DCASE[2] challenge and the introduction of the SONYC dataset give us the opportunity to experiment with machine learning based approaches to AED. These recordings have been obtained by placing sensors throughout New York City, and along with the sensor data, each sensor also records the timestamp and location of the recording (STC data). This dataset contains coarse and fine labels. Coarse labels tell us if any of the 8 coarse classes are present. Fine labels further define these coarse labels by splitting them into more specific groups. For example, a coarse label can be an engine and fine labels for this class are the different sizes of engine. The taxonomy of this dataset is described in Figure 2. In this report, we focus on detecting the presence (or absence) of 8 coarse noise sources in a 10 second audio recording. To accomplish this task, we look at the baseline system presented in [1].

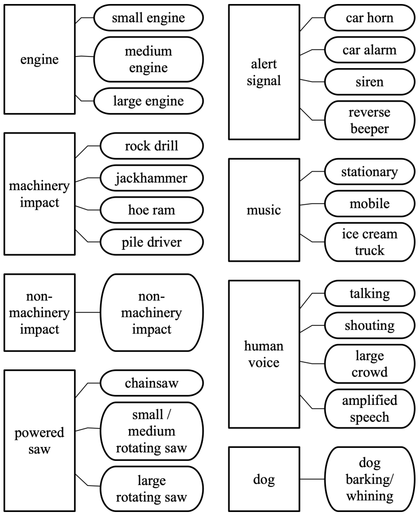


Figure 2: Taxonomy of SONYC dataset.

Here we present a literature review of other submissions to the DCASE challenge. Iqbal et al. [3] uses the mel-frequency spectrogram as input, and instead of concatenating the STC data as is, they pass the STC through a fully connected layer, and then concatenate the ‘processed’ STC with the the mel-spectrogram. An interesting observation they make is that spatial context is not of any use, and thus they only use the temporal context. Gaspón et al. [4] also proposes a similar architecture, but uses the log-mel spectrogram. They also observe that pre-trained feature extractors are sub-optimal. Bai et al. [5] applied a 9-layer convolutional neural network (CNN) to

[1] GitHub

[2] DCASE challenge

log-mel, log-linear and log-mel-h representaion of data. To prevent the imbalance between classes, they also applied data augmentation method to mix various samples to balance the dataset. Arnault et al. [6] uses a TALNet-like architecture, with some notable modifications. In particular, they use the encoder layer of a transformer. Their system takes as input log-mel spectograms and STC data and outputs a multilabel prediction vector. Kim et al. [7], implements an imagenet pre-trained model with the audio data. This is done by first utilizing the median harmonic percussive source separating methodology to separate the raw audio data (R) into its corresponding harmonic (H) and percussive components (P) to construct a multi-channel feature just like RGB. Finally, an image classification network, EfficientNet is used with the extracted features to make predictions.

2. METHODS

The baseline system used for the DCASE competition is a combination of Open-L[3] [8] to extract audio embeddings, and a multi-label, multi-layer perceptron model, using a single hidden layer of size 128 (with ReLU non-linearities) for classification. They also naively incorporate the STC data by appending it to the extracted embeddings. We first use a simpler SVM model as the classifier. We also propose the following changes to the baseline system:

1. Using a more "relevant" dataset to extract embeddings
2. Data Augmentation to create a polyphonic dataset by combining monophonic audio

For evaluating the classification performance, we will be reporting the Macro and Micro AUPRC (Area under Precision-Recall curve) across the 8 (coarse-grained) classes. Fixing a threshold τ for the classifier output, True Positives, True Negatives, False Positives and False Negatives are computed. Then, the precision and recall is computed. Varying τ from [0,1] gives the Precision-Recall curve, and AUPRC is approximated using the trapezoidal rule. The difference between the Macro and Micro AUPRC is that for the for the Macro AUPRC, each class is given equal importance. However, for the Micro AUPRC, each sample is given equal importance, which accounts for the class imbalance in our problem.

2.1. Support Vector Machine Classifier

Most linear classification predictive models do not support multi-label classification and may require adopting some meta-strategies for the same [9]. In order to address the given problem through a linear classification strategy, we implemented the Support Vector Machine (SVM) using a One-vs-All classification method and Principal Component Analysis (PCA) is applied apriori in order to find a reduced feature set from the audio files before training the model. In this task, we are working with time domain samples and applied PCA on the same. Figure 3 demonstrates a graph for variance against the number components in order to decide the near-optimal number of components for PCA. As observed from the graph, taking number of components as 100 gives a high variance of around 95% which is taken into consideration. Due to memory and computational complexity constraints, the model is trained on a subset of randomly sampled data of around 500 audio files.

The reduced audio dimensions are used for training the SVM subsequently. In this method, the multi-label learning problem is transformed into an ensemble of binary classification problems for each target class, such that, one classifier is trained for each class and fitted against all the remaining classes. Since 8 coarse grained labels are considered (y_i where i=1,2,3..8), SVM is applied on the reduced feature set X against each y_i. Thus, N=8 classifiers are trained against N-1 classes as shown in Figure 4. Each class makes a prediction for the membership of the (row) instance, and the combination of all classes prediction is returned as the multi-label output on the test audio sample. We quantify performance using raw accuracy values by counting how many of the predictions correctly identified the presence or absence of a sound class. In order to simplify the problem domain, for any sample containing multiple labels, we look at that sample as multiple, single labeled samples across all classes present.

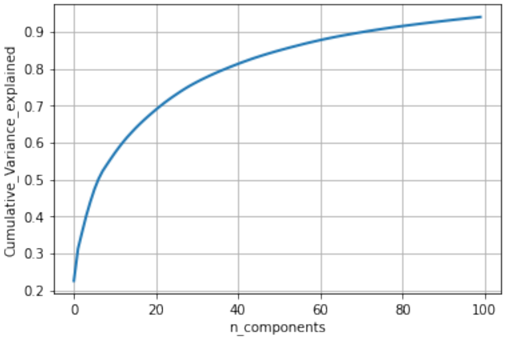


Figure 3: Number of Components Vs. Variance

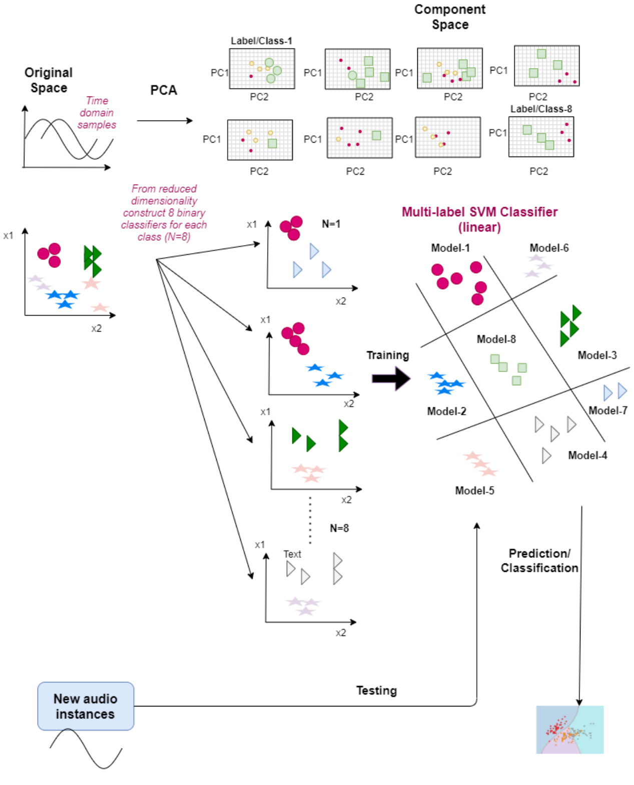


Figure 4: Multi-Label SVM Classification (Linear)

2.2. Transfer Learning

The embeddings for the baseline are extracted using a pre-trained CNN, Open-L^3. The model has been pretrained on the Audioset dataset, consisting of nearly a million hours of video and around ten thousand hours of audio. Cramer et al. [8] claim that the embeddings extracted by the model show promise in downstream audio classification tasks. They also claim that, contrary to what we would expect, matching the training data to the downstream classification task does not actually improve performance. What matters more is the amount of training data used, which would allow the network to learn sufficiently general feature extractors for multiple varieties of audio.

We try two methods of transfer learning:

1. Use a CNN as a pretrained feature extractor, similar to what is done in Open-L^3
2. Use the pretrained CNN to "fine-tune" our model

The difference here is we will train the CNN on a more relevant dataset. For this, we turn to the ESC-50 dataset [10]. Our motivation to use this dataset is 2-fold - (1). Similar to SONYC, this is also an environmental sound dataset, and the classes match well, and (2). The dataset is balanced among all classes, thus the classifier will not be biased to any specific class because of class imbalance.

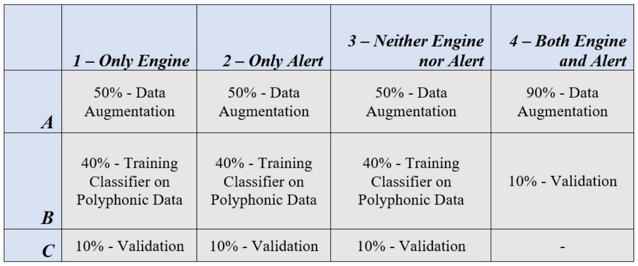
The network we use for extracting the embeddings is inspired from Open-L^3. It is a 2-layered CNN with 50 filters in each layer, followed by a fully connected 1-layer classifier (more details in our repository). For the pre-training, we train until the classifier achieves a reasonable accuracy on the ESC-50 dataset ($\approx 70\%$ train and 60 % test). Then, we use this the pre-trained layer to extract embeddings from the SONYC dataset, and train a classifier on these embeddings. Our observation was that the classifier did not train at all (i.e. the loss did not reduce). Hence we did not continue further with this approach.

One other domain with limited data that comes to mind is medical imaging. Raghu et al. [11] explore "fine-tuning", where the pre-trained network is further trained on the specific task, and show that it is a promising scheme for tasks with limited data. We also follow the same idea here; instead of extracting the features and training a classifier, we use the pre-trained model as an initialization, and update both the classifier and feature extraction weights with the SONYC dataset.

Unlike Open-L^3 which uses log-mel spectrograms, we use mel-frequency cepstral coefficients (MFCCs) for all of our transfer learning experiments. None of the other DCASE submissions have tried MFCCs as a representation. We observed that the MFCCs performed the best, for both transfer learning and without it.

2.3. Data Augmentation

Obtaining multilabel data can be challenging due to the difficulty of annotating samples. Here, we see if we can generate a multilabel dataset from single labeled data. In order to test the efficacy of mixing audio samples to generate new ones, we combined samples that contain a single label for two class types (engine and alert-signal). First, we collect all samples containing only an engine (group 1), samples containing only an alert-signal (group 2), samples containing neither an engine nor an alert signal (group 3), and samples containing both engines and alert signals (group 4). Groups 1, 2 3 are split further into 3 groups. 50% of these groups are kept to create new samples (1A,2A,3A), 40% are kept to train a classifier on non-augmented data (1B,2B,3B) and 10% are kept for validation (1C,2C,3C). 90% of group 4 is taken to train a classifier on non-augmented data (4A) and the remaining is used for validation (4B). These groups are summarized in Table 1



	1 – Only Engine	*2 – Only Alert*	*3 – Neither Engine nor Alert*	*4 – Both Engine and Alert*
A	50% - Data Augmentation	50% - Data Augmentation	50% - Data Augmentation	90% - Data Augmentation
B	40% - Training Classifier on Polyphonic Data	40% - Training Classifier on Polyphonic Data	40% - Training Classifier on Polyphonic Data	10% - Validation
C	10% - Validation	10% - Validation	10% - Validation	-

Table 1: Data Augmentation Groups

To generate a new, augmented sample, we generate a vector of 1's and 0's based on a binomial distribution. Using this vector, we combine a random sample from groups 1A, 2A and 3A. If the selection vector contains all 0's, we select a random sample from group 3A as the augmented sample. Figure 5 summarizes the augmentation scheme.

After we have generated augmented samples, have set aside samples for training on polyphonic data, and have set aside samples for validation, we extract embeddings using the Open-L^3 net and we extract Mel-frequency cepstral coefficients (MFCCs) on all samples. We train the same DCASE baseline network on both the augmented dataset and the already polyphonic data (1B,2B,3B,4A) for 200 epochs and compare performance. Learning rates were manually tuned to ensure smooth learning. We validate both models on groups 1C,2C,3C,4B. We quantify performance using raw accuracy values by counting how many of the predictions correctly identified the presence or absence of a sound class.

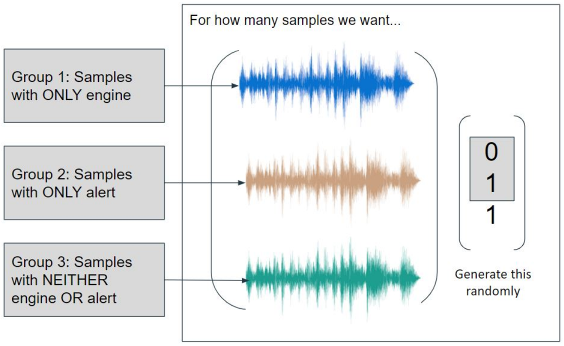

For how many samples we want...
Group 1: Samples with ONLY engine
Group 2: Samples with ONLY alert
Group 3: Samples with NEITHER engine OR alert
0
1
1
Generate this randomly

Figure 5: Data Augmentation Scheme

3. RESULTS

3.1. SVM

Figure 6 illustrates the accuracy score for predicting coarse labels using multi-label SVM classification model. For the subsampled dataset, the results indicate that the model performs significantly good for all the coarse labels with an average accuracy of around 86%.

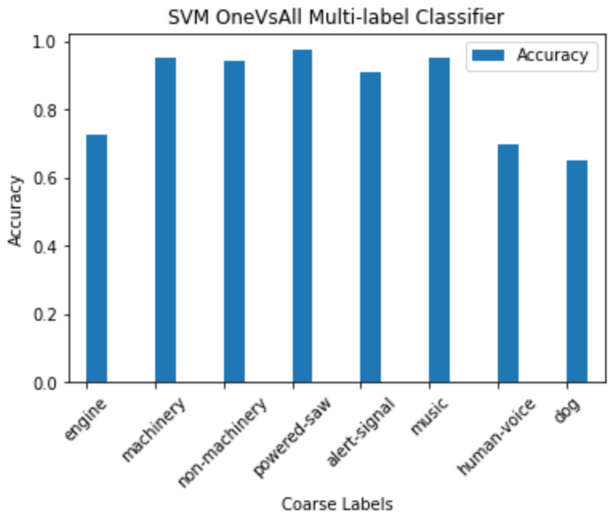


Figure 6: Accuracy score for SVM OneVsAll Multi-label classifier

Model	Macro	Micro
Baseline	0.51	0.75
ESC-50 fine-tuned	**0.56**	**0.80**
MFCC's random init	0.54	0.76

Table 2: AUPRC comparison across models

3.2. Transfer Learning

Table 2 compares our fine-tuned model to the baseline, showing improvement in both the macro and micro AUPRC. We also report the AUPRC results for a randomly initialized model using the MFCCs. Interestingly, fine-tuning via transfer learning only offers a slight improvement instead of using MFCCs and random initialization.

3.3. Data Augmentation

Figure 7 summarizes the results of the data augmentation scheme presented above. Polyphonic MFCCs and polyphonic Open-L^3 embeddings give the best performance based on the raw accuracy metric, converging to around 0.72. The raw accuracy of the augmented representations increase over epochs, suggesting that the model is learning. However, they do not do perform as well as the polyphonic data. When comparing performance of the augmented data representations, MFCCs (0.6) perform slightly better compared to Open-L^3 embeddings (0.58).

4. DISCUSSION

As for the linear classification methods, it is worth mentioning that Support Vector Machines are strictly constrained by separating classes based on the function that is learned i.e. one line distinguishing between the two categories. Therefore, one cannot adopt a conventional SVM based method for multi-label classification in which multiple classes need to be assigned to a sample instance. Thus, as a workaround a multi-label SVM based classifier is implemented by splitting into many classification problems. As observed from the results, the accuracy achieved with this method on the subsampled dataset is quite high of around 86% Also, the multi-label SVM classifier is easier to implement and more interpret-able in contrast to a neural network. However, the method is quite cumbersome with higher computational complexity as it depends on the number of classes/labels for prediction. Additionally, it would be worth exploring the performance of the trained SVM based linear model on the complete audio dataset for future work.

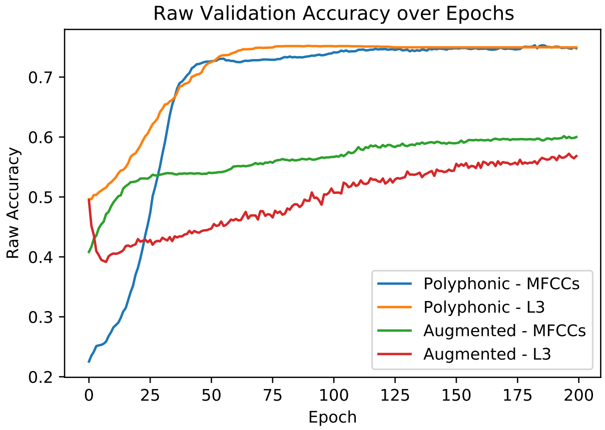


Figure 7: Raw accuracy values for the different data schemes and representations.

For the transfer learning, we observed that using pre-trained network embeddings failed. On further inspection of the pre-trained layer activations, we observed that they were all almost the same. We speculate that this is because the ESC-50 is quite different from the SONYC dataset (although the classes are the same), and hence the embeddings obtained did not generalize across datasets. This also confirms the findings from Cramer et al [8] i.e. a large dataset is needed to for the network to learn sufficiently general feature extractors across datasests. Another observation we make was the MFCCs perform almost as good as transfer learning. Since MFCCs only involve an additional DCT transform over the log-mel spectra, we wanted to see if using PCA instead of the DCT provided any additional improvements. However, because of the size of dataset in consideration, we could not load the entire dataset into memory (not even for an incremental-PCA). Replacing the DCT with a data-driven dimension reduction like PCA might be an interesting extension to explore.

Our data augmentation scheme gave some promising results. Both the MFCCs and Open-L^3 representations led to the baseline model learning as seen by the increasing accuracy across epochs. Importantly, we see that MFCCs do slightly better than the Open-L^3 embeddings when we train on augmented samples. This again points to the power of the MFCC representation when performing audio machine learning. Despite these positive results, our augmentation scheme did not perform as well as the polyphonic data. This could be because we combined audio samples by simply adding them. Performance may improve if audio samples were scaled before adding them together.

One of the major contributions of the SONYC dataset is the inclusion of STC data. However, a very important question to ask is, how do we incorporate this data to improve classification performance. We observe that the naive incorporation of STC in the baseline actually degrades performance as shown in Table 3, contrary

Model	Macro	Micro
With STC	0.51	0.75
Without STC	**0.58**	**0.78**

Table 3: Comparing performance with and without STC

to what you would expect. We also tried to boost two weak classifier trained individually on the features and the STC. However, even this approach preformed only as well as the baseline. A good line of future work would also be to use better representations of the STC data, maybe by representing the sensor network as a graph and using graph neural networks. Furthermore, STC data could be used to obtain other relevant information such as weather and traffic to see if this data boosts performance.

5. CONCLUSION

In this report, we present the DCASE Urban Sound Tagging challenge on the SONYC dataset. This dataset provides well annotated, multi-labeled audio samples, along with STC data for each sample. We explore the simple baseline system presented with the dataset which takes in as input audio embeddings extracted from the Open-L^3 network with the STC information simply appended to each embedding. This model serves as a basic machine listening pipeline. Then we perturb this pipeline to see if we can improve performance. We explore a simpler, linear model by training a SVM to solve the multilabel task. We also look to see if using transfer learning on more relevant datasets improve performance. In particular, we trained another network on a more targeted dataset (ESC-50) and fine-tune this network on the SONYC dataset. Furthermore, we also used the MFCC representation as input in all our transfer learning experiments which also gave similar performance to the baseline, highlighting the power of this representation. We also present a data augmentation scheme that serves to generate polyphonic data from monophonic samples. We show that this scheme has relatively good performance, and also discuss how we may improve it. Finally, we discuss the pitfalls of naively appending STC data to improve performance. Our work lays the foundation for how we can improve AED pipeline. However, the experiments we ran are only with subsets of the actual dataset so future works would be to generalize our approach to the entire dataset. Furthermore, we propose exploring different methods to incorporate STC data to improve performance.

6. REFERENCES

[1] M. Cartwright, J. Cramer, A. E. Mendez Mendez, Y. Wang, H.-H. Wu, V. Lostanlen, M. Fuentes, J. Salamon, and J. P. Bello, “An urban sound tagging dataset with spatiotemporal context,” DCASE2020 Challenge, Tech. Rep., October 2020.

[2] T. Chan and C. S. Chin, “A comprehensive review of polyphonic sound event detection,” *IEEE Access*, vol. 8, pp. 103 339–103 373, 2020.

[3] T. Iqbal, Y. Cao, M. D. Plumbley, and W. Wang, “Incorporating auxiliary data for urban sound tagging,” DCASE2020 Challenge, Tech. Rep., October 2020.

[4] I. Diez, P. Gonzalez, and I. Gonzalez, “Urban sound classification using convolutional neural networks for DCASE 2020 challenge,” DCASE2020 Challenge, Tech. Rep., October 2020.

[5] J. Bai, C. Chen, M. Wang, J. Chen, and X. Zhang, “Data augmentation based system for urban sound tagging,” DCASE2020 Challenge, Tech. Rep., 2020.

[6] A. Arnault and N. Riche, “Crnns for urban sound tagging with spatiotemporal context,” DCASE2020 Challenge, Tech. Rep., 2020.

[7] J. Kim, “Urban sound tagging using multi-channel audio feature with convolutional neural networks,” in *Detection and Classification of Acoustic Scenes and Events 2020*, 2020.

[8] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep audio embeddings,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[9] D. Fu, B. Zhou, and J. Hu, “Improving svm based multi-label classification by using label relationship,” in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–6.

[10] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[11] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, “Transfusion: Understanding transfer learning for medical imaging,” in *Advances in neural information processing systems*, 2019, pp. 3347–3357.