Emotive Classification of Audio Recordings Using Transfer Learning

Lukas Herron
Department of Physics
University of Florida

Timothy Martin
Department of Electrical Engineering
University of Florida

Jorge Perdomo
Department of Electrical Engineering
University of Florida

Gawtam Bhat
Department of Industrial and Systems Engineering
University of Florida

***Abstract*—Spectrograms are a convenient representation of audio signals which encode the frequency components of an audio signal over short temporal windows. Importantly, spectrograms may be represented as images, naturally lending themselves to classification by Convolutional Neural Networks (CNNs). We investigate the performance of 3 pretrained architectures, and finetune and optimize the best performing architecture. Finally, we utilize an ensemble of 4 finetuned CNNs following the Densenet architecture which are pretrained on the Imagenet dataset. We demonstrate that audio recording samples falling into eight emotive categories may be classified with 85.5% accuracy.**

I. Introduction

The classification of audio signals is of great interest, but has been given less attention than image classification, for which many accurate deep architectures have been developed. Fortunately, these domains are not mutually exclusive. Fourier analysis allows signals to be decomposed into their constituent frequencies. For the case of an audio signal, when the amplitudes (Fourier coefficients) of the frequencies are represented as an intensity color map on the frequency-time plane it is known as a spectrogram. And because this representation takes the form of an image, it lends itself to classification by Convolutional Neural Networks (CNNs). From a spectrogram other representations of the signal may be derived such as the MFCC coefficients of the signal, or the Δ-MFCC coefficients, both of which are tested for classification purposes.

Over the past decade many different deep architectures have been developed, and as a general rule, deeper architectures correlate with increased classification performance so long as there is a sufficient amount of data. However, when data is scarce, it is exceedingly difficult to train a deep network without overfitting. To circumvent this issue, transfer learning may be adopted where a network trained on one set of data is re-trained on another set with a small learning rate. For the example of a deep CNN trained on Imagenet, this allows all of the low level features which are extracted from a database of 1.2 million images to be applied to other classification tasks. This is one approach that we take to combat the lack of data present, as the assumption that low level features which are shared between many classes of images may also be present in spectrograms is not unreasonable.

In this paper, we will be presenting, implementing, and testing three different algorithms to train CNNs to accurately classify an audio dataset with eight classes of emotive speech recordings. The primary network we focus on is a DenseNet - a unique deep architecture which consists of densely connected blocks, each of which is made up of convolutional layers that are connected such that the feature maps generated by each layer are inputs to all subsequent layers in the block. This architecture is especially apt at classifying small datasets as the dense connections allow direct paths for the low-level features produced in the initial layers to be transmitted to the output layer of each dense block, allowing for a range of feature complexity; this is in contrast with non-densely connected networks which only use the more complex features produced just before the output layer for classification [1]. We also test an AlexNet which consists of a total of eight layers divided into five convolutional and 3 fully connected layers [2]. Finally we test a SqueezeNet, which is a specific type of AlexNet in which the number of parameters are reduced significantly by employing a number of design strategies. This network was chosen with the rationale that fewer parameters may lead to less overfitting [3]. In order, to analyze each of these architectures and test their accuracy, we will be training each of the CNNs on spectrograms produced from the audio time series signal with augmentation processes described below. After implementing, analyzing, and testing each of the three different architectures, the best performing one will be chosen for fine tuning of hyper-parameters. The final network will an ensemble of the optimal fine-tuned network.

II. Implementation

The model analysis was implemented in Python and requires the Torch, Torchvision, and Torchaudio libraries for feature extraction,preprocessing, and training. All of the models tested were trained on a Tesla K80 GPU. Additionally, four supplementary emotional speech datasets were used: RADVESS, SAVEE, TESS, and CREMA-D [4] [5] [6] [7]. Due to the size of some of the models tested, it may not be possible to fully reproduce the results discussed in this paper using a GPU with less than 8GB of RAM.

For clarity and reproducability, the parameters which were not varied during the experimentation process, or not changed after experimentation are the cross-entropy loss function, the batch size of 24, and the AdamW optimizer [8].

III. Experiments

All of the data, including that from the supplementary datasets, is preprocessed in the same way. Prior to feature extraction, each of the audio signals is resampled to be 100,000 units long in order to match the length of the samples collected in class. The samples are subsequently demeaned and normalized. Afterwards, preliminary testing was done to determine which features to extract. Two short training runs were performed on a Densenet with the three-channel inputs being either (S_1, S_2, S_2) or $(S_1, \mathrm{MFCC}, \Delta - \mathrm{MFCC})$, where S denotes a spectrogram. For the case when all three channels are occupied by spectrograms, each spectrogram is calculated with 128 frequency bins and 4096 Fourier components, but with different hop lengths and window lengths between each channel such that the overlap is constant (25%) between channels. More specifically, the parameters of each channel may be characterized in the form of $(\text{hop length}, \text{window length})$ as $(1024, 256), (2048, 512), (4096, 1024)$. Because different window sizes and hop lengths were used for each channel, each image is resized to be $(128, 256)$. Subsequently the log of the spectrogram is taken to make the entire image informative.

Preliminary results training the (S_1, S_2, S_3) and the $(S_1, \mathrm{MFCC}, \Delta - \mathrm{MFCC})$ representations on a DenseNet indicate the three channel spectrogram representation is superior. Therefore, only the three channel spectrogram representation was further considered for representation.

However, for spectrograms we run into a challenge: Neighboring temporal windows will be highly correlated with one another. Since convolutions extract features, by edge detection the fact that neighboring windows contain redundant information will hamper the networks ability to efficiently extract features. To remedy this, a ZCA-Mahalanobis transformation is performed on each spectrogram. If the frequency bins make up a basis for the spectrogram, a ZCA transform will find a rotation in that basis which decorrelates samples of the spectrogram. One sort of ZCA transform - the ZCA-Mahalanobis transformation - is special in that it maximally preserves the distances between samples in a least squares sense [9]. This transform is realized through a PCA rotation where all of the dimensions were preserved. Intuitively, applying this transformation introduces more edges into the image while preserving distances relations between ”samples” in the image.

We decided to employ transfer learning from models pretrained on ImageNet to classify the data. To decide which model to use, AlexNet, Squeezenet, and DenseNet are compared [2] [3] [1]. The last layer of each class was remapped to 8 outputs corresponding to the eight emotional classes being analyzed, and was trained on the class data with an 80% training 20% validation split for 60 epochs, with learning rates of 10^{-4}, a weight decay of 10^{-3}, and a cross-entropy loss metric. The optimizer used to perform the gradient descent was AdamW, as described in [8]. This optimizer is a variant of the adaptive Adam [10] optimizer with a weight decay constraint; we believe that promoting sparsity among the weights will help reduce overfitting in the deep architectures. The test accuracies of each are shown in Fig. 1, and it is clear that DenseNet is the best performing model. We attribute this to the fact that the layers in DenseNet are densely connected. This means that low level features are used in conjunction with more complex features to compute the network representation at the final convolutional layer. This has been demonstrated to reduce DenseNet’s tendency to overfit compared to other networks

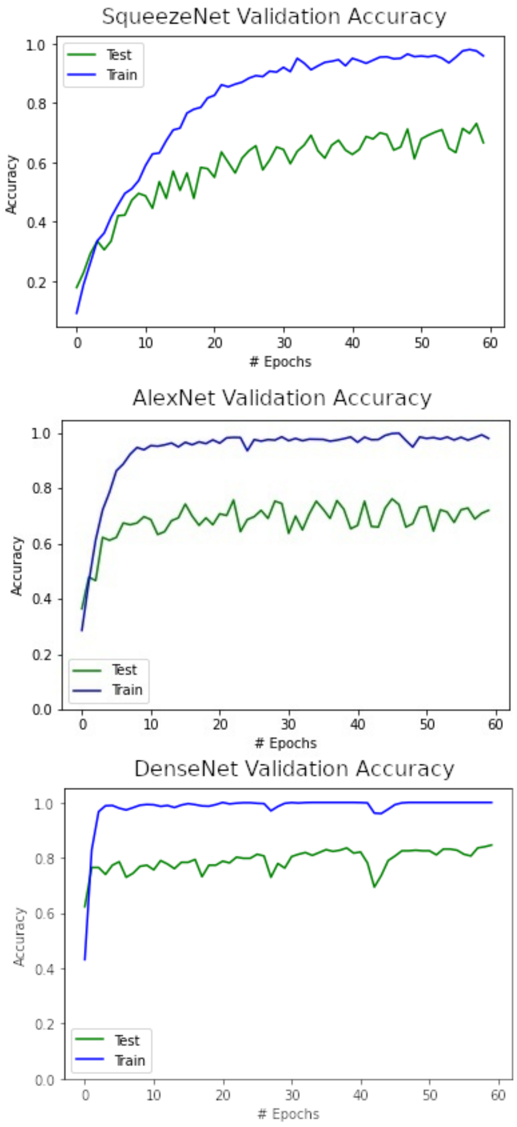

Fig. 1. **SqueezeNet, AlexNet, and Densenet training and validation accuracies.** Each of the networks was trained for 60 epochs with a learning rate of 10^{-4} and a weight decay of 10^{-3} using the AdamW optimizer. The training accuracy is denoted by ”Train” and the validation by ”Test”. Notably, Densenet has a more stable validation accuracy compared to the other two architectures. The different rates of approach to each architecture’s maximum validation accuracy is indicative of each requiring different hyper-parameters.

[1].From Figure 1 we chose to stop future training of the DenseNet once the accuracy had plateaued and before it began to degrade, which happens at approximately 50 epochs.

Once DenseNet was selected, we chose DenseNet-161 which is the largest pretrained model available. The size of the model was maximized to obtain the largest representational capacity possible. Henceforth, ”DenseNet” is taken to mean ”DenseNet-161”.

Network	**Data**	**Accuracy**
DenseNet	Class Dataset	81%
DenseNet	All data	75%
DenseNet	256 samples from each extra dataset	77%
DenseNet	256 samples from each excluding CREMA-D	82%
DenseNet	All data excluding CREMA-D	83%

TABLE I

Furthermore, we wish to incorporate supplementary data into the training phase to promote generalization. We analyzed which sets of data improve the accuracy of the model, the results of which are shown in Table 1. It is clear that samples from the CREMA-D dataset negatively impact the test accuracy of the network. This is likely because the samples from CREMA-D were only 2 seconds in length while the samples from all the other datasets were 3 seconds in length. Resampling the CREMA-D dataset to be the same length as the others likely distorted some features which are critical to spectrogram classification. The remaining datasets included various 3 second phrases said by dramatically trained and untrained individuals. Providing variations to the network in both these domains (uttered sentence and dramatic training) is observed to only increase the performance of the network.

With the network architecture and training data in order, we focus on improving the training accuracy by fine tuning the model’s hyper-parameters. We vary the schedule and magnitude of the learning rate which resulted in an observed 1% increase in accuracy. The schedules tested include a cyclical learning rate which increases then decreases according to a triangular function, and a multiplicative annealing which decreases the learning rate after a set number of epochs. The two performed similarly, but the multiplicative schedule was observed to result in a more stable validation accuracy. Ultimately we settled on a multiplicative decrease in which the learning rate begins at 10^{-3} and is decreased by a factor of 10 every 20 epochs. Thus, on epoch 50 (when the training stops), the learning rate is 10^{-5}.

We also experiment with changing the loss function. All of the above experiments were performed with a cross-entropy loss function, but recent results indicate that using a SVM rather than a fully connected final layer may mitigate overfitting [11]. Rather than implementing a SVM at the output of the convolutional layer we used a hinge loss function which implements a functionally equivalent margin maximization classification scheme. The multi-class hinge loss function is observed to be more stable than the cross entropy but resulted in an 1% decrease in validation accuracy, so the cross-entropy loss is utilized.

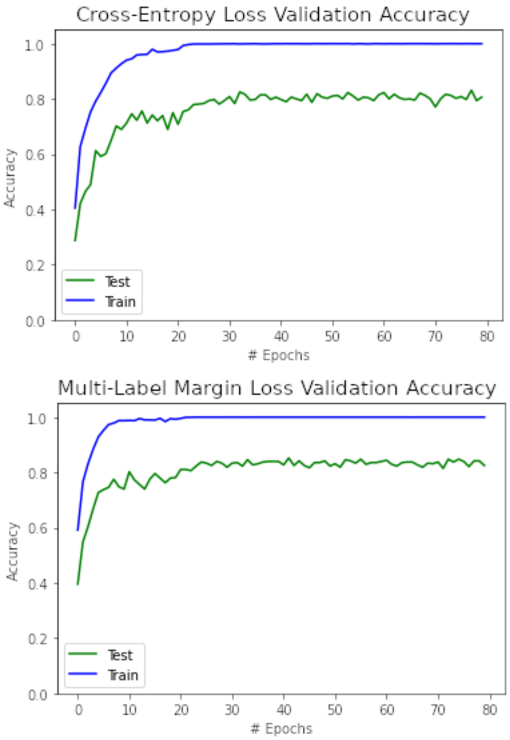

Fig. 2. **Cross-entropy loss and Multi-label margin loss performance.** Depicted above is a comparison of the cross-entropy and multi-label margin loss function performance. The cross entropy loss is a regressive one, while the multi-label margin loss operates on margin maximization.

We also investigate altering the batch size, but the model is found to be robust across the range of batch sizes that RAM constraints would allow. This result warrants little discussion but is included for posterity. All training discussed in this manuscript is performed with a batch size of 24 - the maximum that RAM constraints will allow for a K80 GPU.

The final optimization made is using an ensemble of DenseNets to make predictions. Each of the four DenseNets making up the ensemble were trained on the same data from the 80/20 training-validation split. Once all were trained, their outputs were averaged and the label with the maximum average output was considered the predicted class. Fig. 3 compares the performance of a single DenseNet to that of the ensemble on the validation set. From the figure, it is clear that using an ensemble to make predictions increases accuracy. Even though each DenseNet is trained on the same training set, the randomized batches ensure that each network recieves the training data in a different order. According to [12] this condition is sufficient for convergence to different minima and therefore different representations. For the confusion matrix corresponding to the ensemble prediction the accuracies are generally higher and exhibit less variation along the diagonal while the off-diagonal elements are smaller. Notably, in both cases the error is dominated by confusion between a predicted neutral label and a true calm label. This is likely due to calm

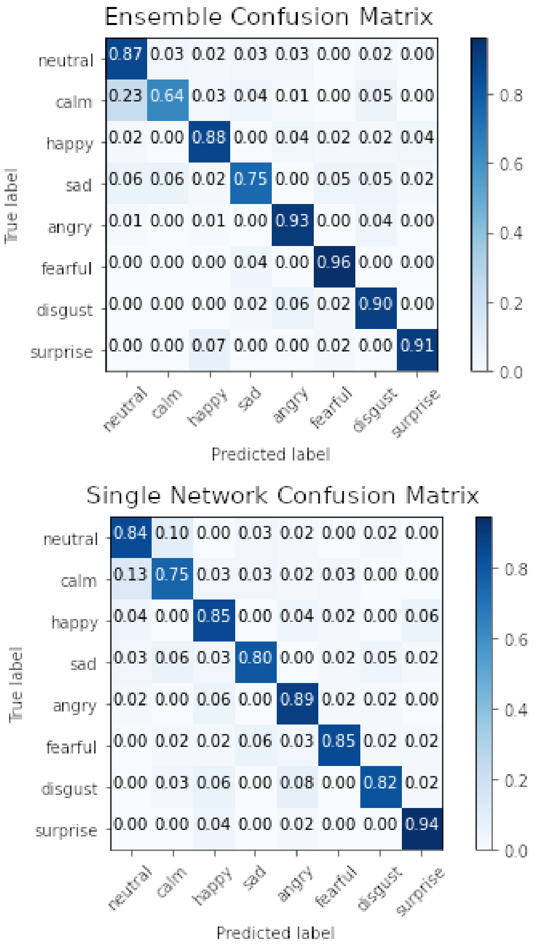

Fig. 3. **Comparing confusion in a single and ensemble DenseNet model.** The above plots depict the confusion of each of the 8 emotional classes. Note that the single DenseNet is not one of the DenseNets constituting the ensemble. Nonetheless, the improvement of the ensemble over the single network model is clearly illustrated. The primary difference arises in the confusion of the calm and neutral classes, which is increased in the ensemble. This is likely due to the limited number of calm samples which will result in increased variability in representation in the training and testing splits.

and neutral being similar emotions and the calm class being underrepresented in our extra data. This is supported by the confusion matrix being asymmetric; the true calm classes were often confused with the neutral class, while the true neutral class was rarely confused with the true calm class. Overall, the final ensemble model had an accuracy of 85.5% which was a 2.5% increase over the single network model.

Conclusions

Ultimately, through experimentation of feature spaces available to us, the usage of spectrograms that have undergone ZCA-Mahalanobis transforms seemed to perform the highest given the models tested. With this feature representation chosen, it was then determined that usage of transfer learning with models pretrained on ImageNet, more specifically the DenseNet-161 model, is the best performing model, resulting in a reasonably high accuracy in test. The model's learning rate, learning rate schedule, and loss function are then optimized. Finally, an ensemble of four DenseNets is used which results in a 2.5% increase in test accuracy over the single model, as well as a greater capacity for generalization.

This is not all to say that we have picked the globally best performing model and feature extraction methodologies, rather that this configuration yielded the highest experimental accuracy in both the training and test sets for the configurations which we tested. Further testing of Recurrent Neural Networks (RNN) and Residual Neural Networks (ResNets) may yield higher performance. In this analysis neither were considered due RNNs not being convolutional networks and ResNets have a tendency to overfit small datasets due to their large depth, though this may be counteracted by freezing the parameters in the early layers of the network. Another lead worth pursing is replacing the output layer of the network with a SVM classifier. The SVM would counteract the tendency of deep networks to overfit while still retaining their large representational capacity. There are likely otehr methods available that could either marginally or drastically improve performance.

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[3] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size," 2016.

[4] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS ONE*, vol. 13, no. 5, p. e0196391, May 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0196391

[5] P. J. S. Haq and J. Edge, "udio-visual feature selection and reduction for emotion classification," 2008.

[6] M. K. Pichora-Fuller and K. Dupuis, "Toronto emotional speech set (tess)," 2020. [Online]. Available: https://doi.org/10.5683/SP2/E8H2MF

[7] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-d: Crowd-sourced emotional multimodal actors dataset," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, Oct. 2014. [Online]. Available: https://doi.org/10.1109/taffc.2014.2336244

[8] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[9] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," *The American Statistician*, vol. 72, no. 4, pp. 309–314, Jan. 2018. [Online]. Available: https://doi.org/10.1080/00031305.2016.1277159

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[11] A. F. Agarap, "An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification," *CoRR*, vol. abs/1712.03541, 2017. [Online]. Available: http://arxiv.org/abs/1712.03541

[12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2017.