Urban Sound Classification: A CNN Exclusive

Name: Brijesh Patel, Victoria Chiu
SUNet ID: brijeshp, vchiu94

Abstract

1 Introduction

A growing research field has been the classification of environmental sound with applications to large-scale, content-based multimedia indexing and retrieval [1–3]. Classifying everyday urban sounds would expand the capabilities of audio sensor surveillance networks, noise control, and acoustic environmental planning [4]. Current challenges for this task include a lack of a common taxonomy and a scarceness of real world, annotated data [5]. Also, unlike the classification of music or speech, environmental (urban) sounds are often unstructured and lack a clear demarcation [3]. In this project, we will analyze urban acoustic environments with the goal of increasing classification accuracy.

2 Related Work

Based on a literature search on research in urban sound classification, we are proposing a convolutional neural network architecture with inputs of preprocessed mel-frequency cepstrum data. Two existing significant architectures used for urban sound classification are: SB-net (2016) and Zhous custom net (2017) [4, 6]. SB-net consists of three convolutional layers with pooling and two fully connected layers using time frequency patches as input (79% accuracy) [6]. Zhou's net builds upon SB-CNN with a smaller field size (2x2) and a deeper structure consisting of four, two dimensional convolutional layers, one fully connected layer and one output layer achieving a slightly better performance at 84% [4]. We will begin our analysis by re-implementing SB-CNN and making modifications to improve its performance, similar to Zhou's modifications.

3 Data Collection

We will be analyzing the UrbanSound8K dataset which begins to address the problems of lack of labeled data and a common taxonomy. This dataset consists of classes from real noise complaint data based on the Urban Sound Taxonomy proposed by Salamon, Jacoby, and Bello [5]. It was extracted from Freesound, an online sound repository, and consists of 8732 urban sound excerpts, all of which are real field-recordings. While still relatively small, this dataset has been used almost unanimously be researchers classifying urban sound. The majority of clips are of length 4 seconds (84%) with the remaining clips <4 seconds in length from one of the following 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music. The class sizes are roughly balanced with most around 11-12% of the dataset with the exception of car_horn (4.9%) and gun_shot (4.3%).

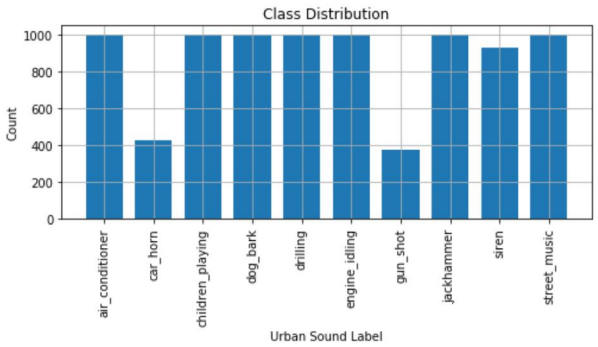

Figure 1: Distribution of classes in UrbanSound8k

3.1 Preprocessing and Feature Extraction

In environmental sound classification (ESC) and, in general, audio signal processing, Mel Frequency Cepstrum (MFC) analysis is commonly used in classification and analysis because it better approximates how humans perceive sound [7]. Due to its popularity in state of the art ESC literature, we decided to use the MFC spectrogram as an input feature for our baseline. We calculated the Mel-Frequency spectrogram and its corresponding deltas from the the 8732 audio samples. Since our dataset only has 8732 audio samples, we followed Salamon's work by splitting the clips into segments in order to have more examples to train and test our model. We split the segments into 50 % overlapping windows with length 633 ms. Then a Mel-Frequency spectrogram (Figure 2) and its delta were extracted with 128 mel bands and 128 frames [8]. The spectrograms and the deltas were then shaped into a two channel input for our Convolutional Neural Net model. Based on the size of the final dataset, we utilized 80% of the data for the training set, 10% for the validation set and 10% for the test set. The training dataset is of size (38550, 128, 128, 2) where the first dimension is the number of examples, the second and third dimensions are the bands and frames, and the fourth dimension represents the two layers, spectrograms and corresponding deltas.

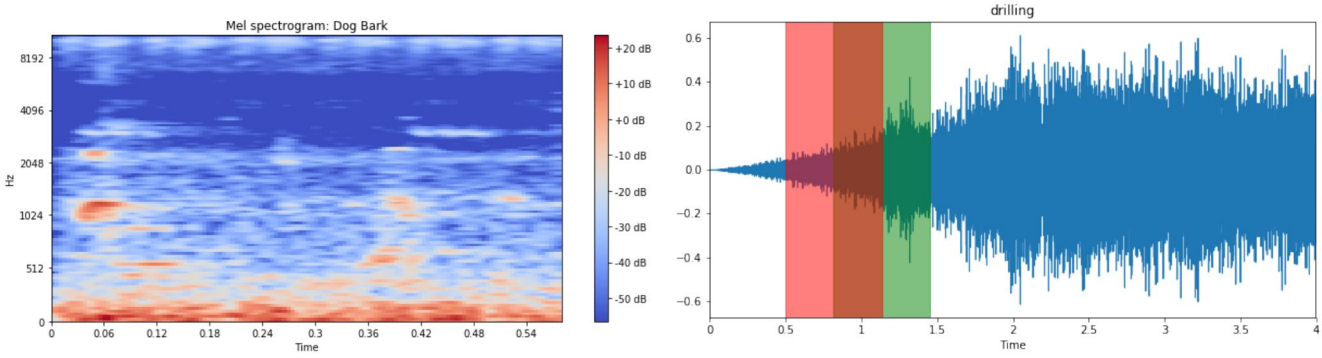

Figure 2: Mel spectrogram (left) and raw audio (right) waveforms from two classes in the UrbanSound8K Dataset. Overlapping boxes on the drilling image demonstrates clips with 50% overlap of 633ms duration.

4 Methods

4.1 Multinomial Logistic Regression

For our most basic baseline model, we sought to evaluate the performance of multinomial logistic regression. With multinomial logistic regression, the loss minimised is the multinomial loss fit across the entire probability distribution [9].

4.2 SB-CNN

For our primary baseline model, we implemented a 5 layer CNN developed by Salamon and Bello (SB-CNN) [6]. The 5 layers consist of 3 convolutional layers followed by 2 fully connected layers. The convolutional layers consist of a 2D convolution (5x5 filter size) followed by 2D max pooling (4x2). A ReLU activation function is used after every layer except the last one, which uses Softmax since this is a multiclass classification problem. During training, we use stochastic gradient descent, with a batch size of 32. We train using categorical cross entropy loss with a learning rate of 0.001.

4.3 BV-CNN

Based on our results from our re-implementation of SB-CNN, we made some modifications in an attempt to increase the performance of this model. SB-CNN had a large delta between the training and validation accuracies which we attributed to the combination of dropout and L2 regularization. We decreased dropout rate to try and alleviate this discrepancy between the train and validation accuracies. Next, we fine tuned the filter size, varying it from 3x3 to 7x7 and added an additional convolutional layer to each convolutional block in order to increase the depth of our model and to add more non-linearities. Our final architecture employs increasing dropout rate (0.1 in layer_4 and 0.2 in layer_5) and larger filter sizes (7x7) (Figure 3).

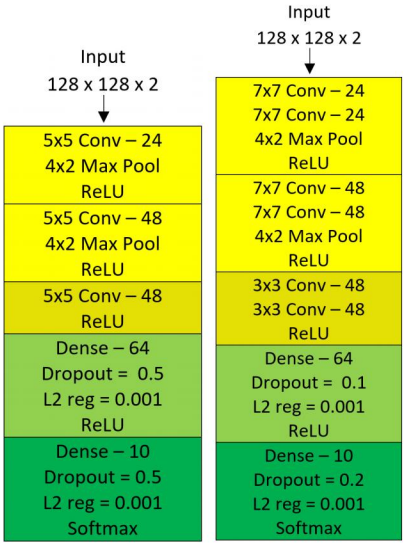

Figure 3: Block Diagram for SB-CNN(left) and BV-CNN (right).

4.4 Categorical Cross Entropy Loss

The categorical cross entropy loss is defined here as

$$-\sum_{c=1}^{31} y_{o,c} \log(p_{o,c})$$

where we have 10 classes, $y_{o,c}$ is the correct label and $p_{o,c}$ is the output probability of that label from the softmax layer. Since each urban sound can only belong to one category, the categorical cross entropy reduces to a single term $-\log(p_{o,c})$, the negative natural logarithm of the predicted probability. The closer the predicted probability of the correct label is to 1, the smaller the categorical cross entropy loss. As a limiting case, when the predicted probability of the correct label is 1, we achieve zero loss.

5 Results

The multinomial logistic regression classifier was a very light weight classifier with fast training. As expected, the linear classifier did not classify well (test accuracy = 55%).

The primary baseline was the re-implementation of SB-CNN. Our re-implementation of SB-CNN has a test accuracy of 87% which differs from the 79% test accuracy reported from the original SB-CNN [6] because we modified the input into the network.

SB-CNN had roughly a 30% delta between validation and training accuracy, implying an overuse of regularization (currently 50% dropout) (Figure 4). When we removed dropout entirely, validation accuracy did not improve after only 10 epochs. After decreasing the overall dropout rate while also employing relatively larger dropout rates at deeper layers, we obtained 92% test accuracy. Modifications to filter size and depth of the model seemingly resulted in some overall improvements, with a final test accuracy of BV-CNN at 94% (Figure 5).

Model	Parameters Changed	Training	Validation	Test
Logistic Regression				55%
SB-CNN [6]				79%
SB-CNN	Window Size, Frame Rate (Hop Size)	50.96%	87.83%	87%
BV-CNN	Dropout Rate	84.34%	92.01%	92%
BV-CNN	Filter Size, Dropout Rate	85.05%	91.78%	93%
BV-CNN	# of Filters, Filter Size, Dropout Rate	86.13%	93.33%	94%

Table 1: Summary of model accuracies

6 Discussion

From the analysis of SB-CNN to BV-CNN, it was clear that the majority of improvement from BV-CNN to SB-CNN was a result of two factors. The first factor was decreasing overall regularization, which caused severe underfitting of the training dataset in SB-CNN (Figure 4). The most significant improvement was due to the modification of our input to the CNN. Salamon and Bello segmented their data in 3s windows with a 23 ms frame rate when calculating the melspectrogram, while we segmented our data in 633 ms windows with a 4 ms frame rate. The smaller time window results in lowered time resolution, but higher frequency resolution, which increases the resolution of the features within the mel spectrogram (input into the neural network). Since this was the only difference between our implementation of SB-CNN and the original SB-CNN, the increased spectral resolution allowed for increased classification accuracy.

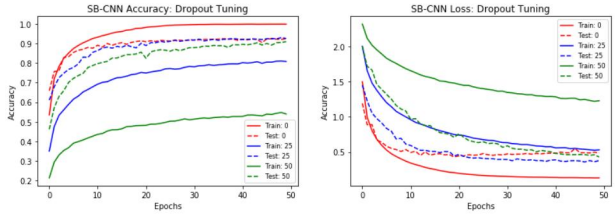

Figure 4: Dropout Tuning: Accuracy and Loss

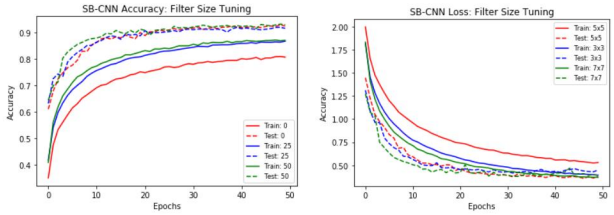

Figure 5: Filter Tuning: Accuracy and Loss

The categories that were most often misclassifed are street music, children playing, and gunshot. Street music and children playing are generally background sounds so they have a wider spectrum of sounds with a lower signal to background noise ratio which means that there is a less distinctive set of features in the mel spectrogram. Since gunshot has less than half the amount of data points of the other classes, this was an indicator that the classification accuracy would be less.

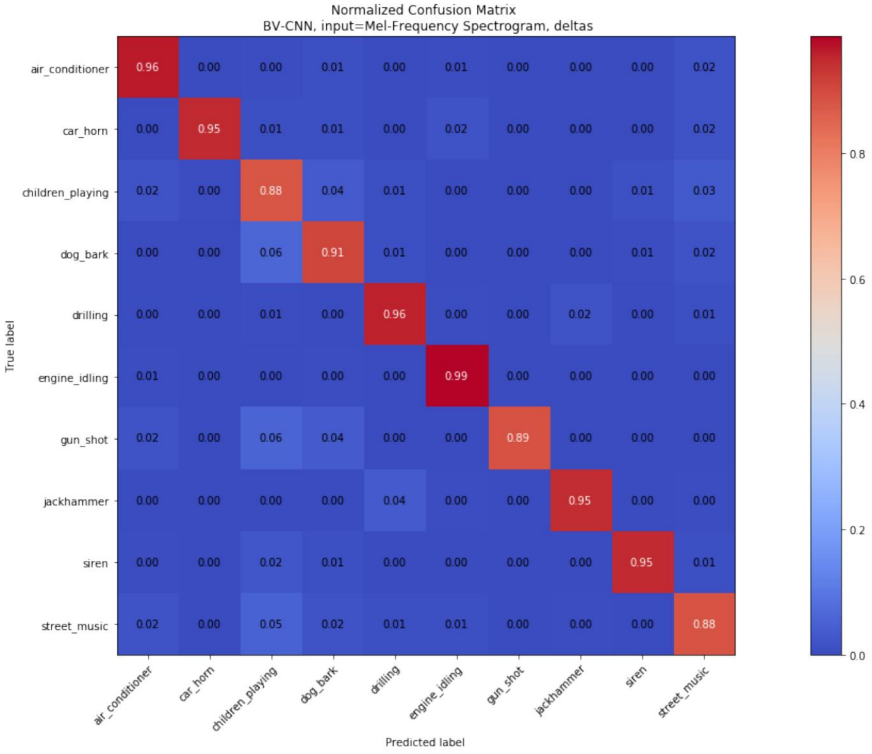

Figure 6: Normalized Confusion Matrix

7 Future Work

Our most significant improvements SB-CNN came from a feature extraction modification as opposed to an architecture modification or hyperparameter improvements. Therefore, in the future, experimentation with varying mel bands, frame rates, and window size as inputs to BV-CNN would likely yield better results. Due to the small size of the UrbanSound8K dataset (8732 clips), increasing the amount of examples via data augmentation would help BV-CNN train more effectively. Examples of possible data augmentation include pitch shifting, time shifting, and time stretching, which would be performed prior to mel spectrogram pre-processing [6].

Github Repo
https://github.com/Bergie1994/CS230

References

[1] R. Radhakrishnan, A. Divakaran, and A.Smaragdis, "Audio analysis for surveillance applications," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005*, 2005.

[2] M. Xu, C. Xu, L. Duan, J. Jin, and S. Luo, "Audio keywords generation for sports video analysis," *ACM TOMCCAP*, 2008.

[3] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia computer science*, 2017.

[4] H. Zhou, Y. Song, and H. Shu, "Using deep convolutional neural network to classify urban sounds," *TENCON 2017 - 2017 IEEE Region 10 Conference*, 2017.

[5] J. Salamon, C. Jacoby, and J. Bello, "A dataset and taxonomy for urban sound research," *22nd ACM International Conference on Multimedia*, 2014.

[6] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, 2017.

[7] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011.

[8] K. J. Piczak, "Environmental sound classification with convolutional neural networks," *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, 2015.

[9] Scikit-learn, "Logistic regression cv," https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html.