Mind Your Accent: An Empirical Study on Differences in Performance of Various Machine Learning Models for Accent Classification

Yeabsira Gebreegziabher* Daniel Lumbu† Ethan Masadde‡
Carleton College Carleton College Carleton College

Abstract

As speech-based AI becomes more prevalent, concerns have grown about its performance when being used by people of non-native English accents. A possible solution that has been proposed ofr this is accent classification to inform the type of speech recognition system that is used This study evaluates a range of machine learning models for accent classification using data obtained from the VoxForge corpus. We compare traditional models such as logistic regression and SVMs with deep learning architectures including a feedforward network, a regular CNNs, a novel VFNet architecture, and a CNN-RNN hybrid. Results show that SVMs perform best on mean MFCC inputs, while the base CNN model outperforms others when using full MFCC matrices. Performance declined when using Mel Spectrograms, suggesting that MFCCs are a more effective feature representation for this task. Our findings highlight the importance of input representation and model architecture in accent classification problems.

1 Introduction

In recent years, there has been a significant uptick in the number of AI agents being deployed in the wild. These agents are used for various tasks, from general-purpose chatbots that can engage in casual conversations to customer service assistants used by big corporations (among others) to optimize the process of responding to and addressing customer needs. Advances in machine learning technology have allowed for improvements in the performance of the automatic speech recognition systems deployed to handle these various tasks.

While this has increased different companies' capacity to address customer needs, it has been found that people with specific accents often have a difficult experience with automated customer service calls. [6] Accents are generally varied patterns of speech that different speakers of a language exhibit, and they tend to cluster among speakers of similar languages that are of similar ethnic and geographic backgrounds. Accents can introduce systematic variations in pronunciation, intonation, and rhythm, which pose challenges to traditional automatic speech processing systems.

For example, a survey of 3000 Americans was able to identify the top five regional accents that AI struggles to understand. [7] This survey did not even specifically examine the experiences of persons of marginalized communities, especially those of foreign origin, who are historically underrepresented in training datasets for these models. In an increasingly globalized world, these speech recognition systems must be robust to the different accents exhibited by different speakers of the same language.

One approach to addressing this issue is using accent classifiers to improve the customer experience. Identifying the accent/manner of speech of the user could facilitate the use of a cascading method, in which the a user's accent determines which subsequent model should be used for better tailored speech recognition. This could in turn reduce the computational complexity that may come with training and deploying an adequately robust speech recognition model.

Accent classifiers in machine learning take an audio utterance as input and assign it an accent class. Past researchers have proposed a variety of approaches that employ different methods and data representations to tackle this task. In our project we consider a handful of these approaches and evaluate their effectiveness by comparing the model performance and investigating other model characteristics such as interpretability.

2 Related Work

In past research, many researchers have often utilized feature extraction of the audio utterances to be classified as opposed to the raw audio data itself. These include spectrograms and MFCCs (Mel Frequency Cepstral Coefficients) among others. Researchers have also directly used phonetic and lexical properties to train accent classifiers. [8] [12] [13] These properties were commonly used in some aggregated formt to train traditional models such as Gaussian Mixture Models (GMMs), Support Vector Machines (SVMs),Gradient Boosting and other classical methods not explicitly used for audio data.

While these methods can yield significantly accurate models, some have leveraged the rise of deep learning approaches and proposed more complex models for accomplishing the accent classification task. By taking advantage of the image-like 2-dimensional nature afforded by MFCCs and spectograms, the proposed models have been mainly based on Convolutoinal Neural Network architectures. These architectures seek to outperform the previous state of the art.

For example, Ahmed et al. [1] introduce VFNet which uses multiple filters of varying size after the convolution. Additionally, in their 2023 publication Song et al. [10] proposed three slightly more complicated networks, including the MPSA-Densenet which combines multitask learning and the PSA module attention mechanism with Densenet to achieve state of the art results. Others, such as Zhang et al. [13] have also applied transformer based architectures in their attempts to solve the solution. Both these models did better than traditional approaches. The aforementioned VFNet architecture is among those explained and explored at a more granular level later in this report.

3 Data

The data used in this project was obtained from the VoxForge Speech-Corpus [11] which is an open speech dataset for use with speech recognition models. The dataset is well suited for the task because it includes a variety of audio recordings from volunteers with a range of English accents. To obtain the data, we scraped a number of *.tgz* files from the *16kHz_16bit* sub-folder of the available Audios in the speech corpus. We then extracted the relevant metadata including the audio, accent class, and gender of the speakers. Using this data, we filtered our 6320 observations for specific target accents, i.e. American English, Canadian English, and European English and grouped these observations into their respective classes, allowing for a maximum of 200 observations per class.

*e-mail: gebreegziabhery@carleton.edu
†e-mail: lumbud@carleton.edu
‡e-mail: masaddee@carleton.edu

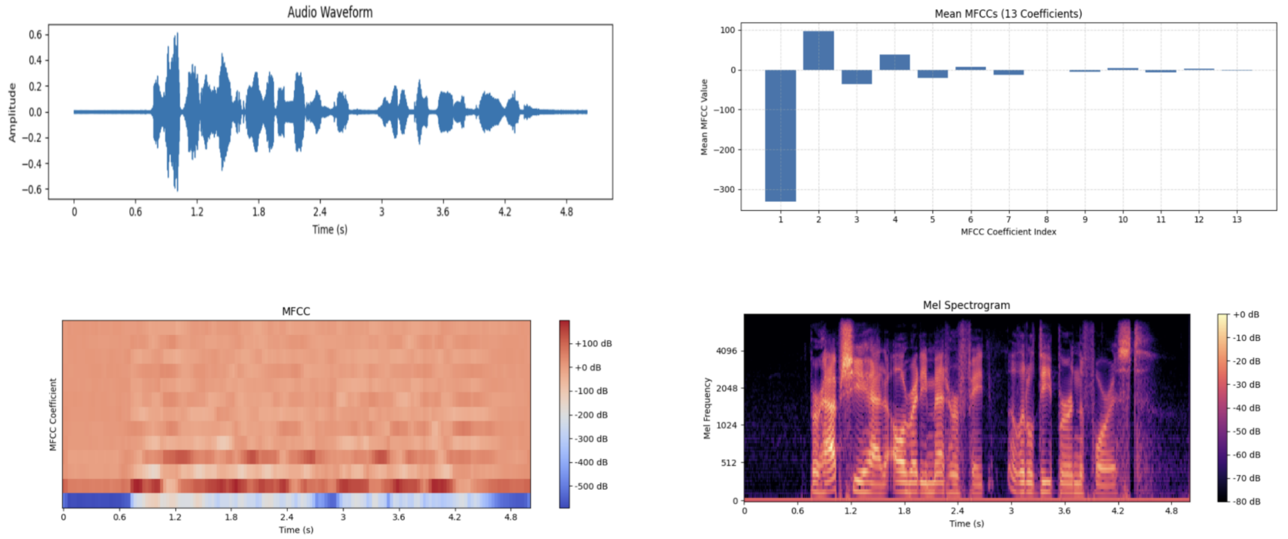


Figure 1: Visualizations of the different ways to represent an example speech signal in the American Accent class. Shown here is the amplitude waveform, (top left), the mean MFCC value for each of 13 coeffecients defined (top right), the gross MFCC for the raw audio (bottom left), and the resulting Mel spectrogram (bottom left). In this project, we utilize the mean MFCC values to train logistic regression, SVM, and a feed forward network, while we use the gross MFCC and Mel Spectrogram to train our CNN-based architectures.

During preprocessing, we extracted different features for the different types of models. For both types of models we used the Mel-Frequency Cepstral Coefficients (MFCCs) with 13 features from the different audio samples. We chose to use MFCCs because they extract non-linear representation of the speech signal very similar to what is done by the human ear. Because all the audios were of many different durations, for the logistic regression and SVM models, we calculated the mean MFCC features for each audio to summarize the spectral distribution across time. For the neural networks, we maintained the temporal dimension of the MFCCs but restricted each audio to produce 200 frames through padding (for shorter audios) and truncating (for longer audios). Additionally, we generated spectrogram distributions for each audio and saved these as images to compare the effect of changing the input type on model performance. Visualizations of these different features can be seen in Fig.1. All resulting datasets were split, withholding 20% of the data for validation and testing model performance.

4 Methods

In our project, we explored four different types of models, all well suited for the classification task: logistic regression, Support Vector Machines, feed forward networks, and Convolutional Neural Networks. The logistic regression, SVM, and feed forward models were trained on the mean MFCCs accross the entire duration of the audio from the observations, while the CNNs were trained directly on the MFCCs and later the Mel Spectrograms we obtained from the audios restricted to 200 frames.

We employed the logistic regression model from Scikit learn as our baseline. We used multinomial regression since we framed our task as a multiclassification problem. Similar to most of the models we discuss in this report, the model works by taking in an input vector, $\mathbf{x}$, with a goal of predicting the accent class, $\mathbf{y} \in \{0,1,2\}$. In multinomial regression, the model computes the probability of an observation belonging to any of our three target classes. It does this by fitting a separate binary logistic regression model for each of our outcome possibilities and predicting the class with the highest probability given your input. The general equation for this method is as follows [5]:

$$\begin{aligned} p_j(\mathbf{x}) &:= P[Y = j \mid X_1 = x_1, \ldots, X_p = x_p] \\ &= \frac{e^{\beta_{0j}+\beta_{1j}X_1+\cdots+\beta_{pj}X_p}}{1+\sum_{\ell=1}^{J-1} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\cdots+\beta_{p\ell}X_p}} \end{aligned} \tag{1}$$

where Y is the predicted class, X is the input vector with p features and J is the number of classes, such that $j \in \{0,\ldots,J-1\}$.

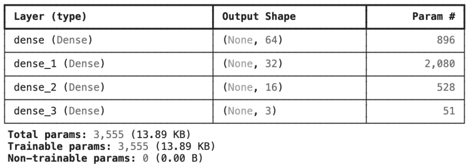
As previously mentioned, we also employed an SVM model to compare to our baseline. To achieve multiclassificaiton functionality, we employed Scikit Learn's default one-vs-one scheme. For a binary classification task, an SVM works by trying to find the best decision boundary (a hyperplane) to split the classes by maximizing the distance between said margin and the closest points of either class, which effectively act as the support vectors. To extend this approach to a multiclassification problem space, in the one-versus-one scheme, a separate model is trained for each pair of classes and each classifier votes for a class. The observation is then assigned the class with the highest votes. To allow our model to effectively determine the optimal decision boundary, we specified a Gaussian kernel function which enables the model to separate the classes according to spacing relationships based on higher dimensional positions that those that are directly encoded into the data. The SVC module provided by Scikit Learn frames the SVM problem as follows [4]: given training vectors $x_i \in \mathbf{R}^p$, $i = 1,\ldots,n$, in two classes, and a vector $y \in \{1,-1\}^n$, our goal is to find $w \in \mathbf{R}^p$ and $b \in \mathbf{R}$ such that the prediction given by $\text{sign}(w^T\phi(x)+b)$ is correct for most samples. SVC thus solves the following binary classification problem:

$$\begin{gathered} \min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \\ \text{subject to } y_i(w^T\phi(x_i)+b) \geq 1-\zeta_i, \\ \zeta_i \geq 0, i = 1, \ldots, n \end{gathered} \tag{2}$$

where ζ_i is a distance term dictating how much a sample is allowed to deviate from the correct decision boundary.

To explore how deep learning methods compare to our more traditional models, we implemented and trained one feed forward network along with several convolutional neural network (CNN) architectures. Unlike logistic regression and SVMs, which were trained on mean MFCC vectors, all deep learning models (except the feed forward network) were trained on fixed-length MFCC feature as well as fixed-length mel spectrogram image-like matrices corresponding to the first 200 frames of each audio sample. The goal in doing this was to retain temporal structure in the data and allow the CNN to leverage local time-frequency correlations that are critical in spoken language, as has been done in previous work. [10] [1]

As a first exploration into the potential of applying neural networks to this task, we developed a deep feed forward network with three hidden layers that also took in an audio samples mean MFCC averaged across time and fit itself to predict the audio sample's true accent class. For each hidden layer, we used the ReLU activation function (which applies the activation function $x := max(0,x)$) and used Softmax for the final output layer.



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	896
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 3)	51

Total params: 3,555 (13.89 KB)
Trainable params: 3,555 (13.89 KB)
Non-trainable params: 0 (0.00 B)

Figure 2: Model summary for our feed forward neural network.

The second family of neural networks that we explored was CNNs, which have the advantage of learning from the temporal and spatial relationship inherent to the audio data. This provides sufficient contrast to the aggregation approach performed for the aforementioned approaches, in which a lot of this information is lost. The baseline CNN (fig reference) was a simple network consisting of two consecutive convolutional layers with filters of size 3×3 each with batch normalization to speed up trinaing and max pooling. As is general practice, these were followed by a dense layer and an output layer which provided the "probability" of a specific observation belonging to each class. We also added a dropout layer to make our model more robust.

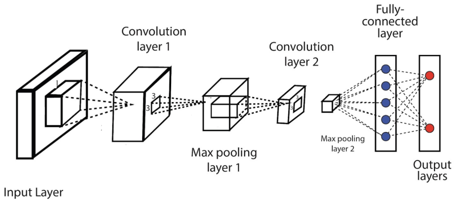


Figure 3: A visual representation of a CNN from [3] that is very similar to our base CNN model architecture.

Another variation of CNNs that we experimented with was an implementation of the novel architecture, VFNet introduced by (/cite) which uses multiple filters of variable size at a single convolution step followed by maxpooling and concatenation of the various filter outputs. The output of this layer is then passed through a fully connected layer to get the accent label of the speech vector used as input. In our implementation, we achieved the variable filter functionality by defining separate convolutional layers for each filter size and merging them using the concatenate feature provided by the Keras API. The model architecture of VFNet provided by the authors can be seen below in Fig.4

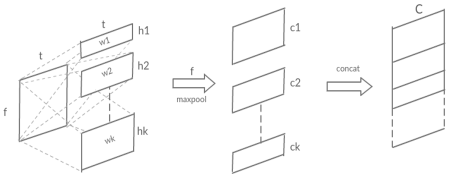
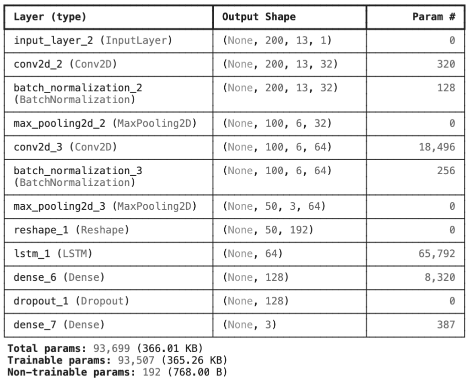


Figure 4: A visual representation of the VFNet architecture from the authors of [1].

Finally, we also modified our base CNN to construct a hybrid CNN-RNN model to account for the temporal dependencies inherent to a speech signal. This is an approach similar to what is done by Ashraf et al. in [2] for music classificaiton. We did this by adding a Long Short-Term Memory (LSTM) layer from the Keras API in between the convolutional and dense layers. After the convolutional layers extracted spatial features from the MFCC input, we reshaped the output into a sequence suitable for the LSTM. This layer effectively processes the sequence, one time step at a time, using a sort of gating mechanism that performs "forgetting" of some information at each time step and then combines this partially forgotten information with the new information in the following time step. [9] This process should allow this model to learn key differences in accented speech such as patterns of intonation. The summary of the resulting network is provided below in Fig.5.



Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 200, 13, 1)	0
conv2d_2 (Conv2D)	(None, 200, 13, 32)	320
batch_normalization_2 (BatchNormalization)	(None, 200, 13, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 100, 6, 32)	0
conv2d_3 (Conv2D)	(None, 100, 6, 64)	18,496
batch_normalization_3 (BatchNormalization)	(None, 100, 6, 64)	256
max_pooling2d_3 (MaxPooling2D)	(None, 50, 3, 64)	0
reshape_1 (Reshape)	(None, 50, 192)	0
lstm_1 (LSTM)	(None, 64)	65,792
dense_6 (Dense)	(None, 128)	8,320
dropout_1 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 3)	387

Total params: 93,699 (366.01 KB)
Trainable params: 93,507 (365.26 KB)
Non-trainable params: 192 (768.00 B)

Figure 5: Model summary for our hybrid CNN and RNN model prior to adding regularization.

All neural networks were trained using the Adam optimizer and sparse categorical entropy loss.

5 Experiments and Results

Our experiments mainly explored the effect on performance caused by differences in model architecture/complexity and model input. As mentioned earlier, we considered traditional models such as Logistic Regression and Support Vector Machines as well as various CNN-based architectures.

5.1 Model Complexity

To compare the effect of varying model architecture, the accuracy results of our Logistic Regression, SVM, and Feed forward network are detailed in table 1 below. From this table, it is clear that Support Vector Machine performs the best on mean MFCC data. While the feed forward network performs almost just as good, the performance of the multinomial logistic model is dramatically worse than the others. This is likely due to the SVM's ability to model non-linear boundaries in high-dimensional feature spaces via the

RBF kernel. Considering the confusion matrices in Fig.6, all models consistently performed best on European English but performed relatively similarly on Canadian and American English.

Model	**Test Accuracy**
Logistic Regression	0.47
Support Vector Machine	0.89
Feed Forward Network	0.84

Table 1: Test Accuracies for our simpler models.

5.2 Model Input on Models of varying Complexity

In investigating the effect of changing the type of model input on the performance of the model, we considered 3 different models, base CNN, a hybrid CNN + RNN architecture, as well as VFNet. In each of our experimental trials, all models performed dramatically worse on Mel Spectrogram encoded data than on MFCC. The resulting train and validation accuracies can be seen in Fig.7 While MFCCs are most commonly used in literature, there has been evidence that using Mel Spectrogram feature extraction can yield effective classifiers. However, our results do not support this idea.

6 Discussion and future work

From the results observed during our experiments, it appears that the mean MFCC coeffecients for an entire audio sample prove to be the most reliable predictor of the accent class of that speech signal. When compared with the cases in which we used the MFCC features for the entire audio, performance for the former was significantly better. Additionally, the fact that the SVM model performed best on the data may prove that a model of moderate complexity is better suited for such data. For the cases in which we varied model input (i.e. MFCC vs Mel Spectrogram), the MFCC cases performed much better than those in whihc we used Mel Spectogram informaition despite the latter observations containing more features for each datapoint. Additionally, using Mel Frequency Spectograms dramatically increased our training time from 4̃0 seconds per epoch to 4̃00 seconds per epoch in each case. The dramatic difference in performance between the two suggests that the added computational overhead is effectively unnecessary.

However, considering that we limited both MFCC and Mel Spectrogram to the first 200 frames, it is possible that this is the reason for such a dramatically low accuracy in the latter case and subpar performance in the former. Future work should perform more detailed preprocessing of data including removing silences and amplifying important aspects of the speech signals used to train the model. Additionally, exploring the effects of normalizing data on model performance is also worthwhile.

Lastly, we did not perform any hyper parameter finetuning on our models as it was outside the specific goal of our project, especially since a number of the models we considered were of great complexity. It is very likely that taking the time to determine the optimal combination of hyperparameters through, for example, cross validation could lead to significantly better model performance. The accuracy graphs we obtain for the Mel Spectrogram case provide strong support for this idea as they show that it would Employing regularization to the model

References

[1] A. Ahmed, P. Tangri, A. Panda, D. Ramani, and S. Karmakar. Vfnet: A convolutional architecture for accent classification. *2019 IEEE 16th India Council International Conference (INDICON)*, Dec 2019. doi: 10.1109/indicon47234.2019.9030363

[2] M. Ashraf, F. Abid, I. U. Din, J. Rasheed, M. Yesiltepe, S. F. Yeo, and M. T. Ersoy. A hybrid cnn and rnn variant model for music classification. *Applied Sciences*, 13(3):1476, Jan 2023. doi: 10.3390/app13031476

[3] T. Balodi. Convolutional neural network with python code explanation: Convolutional layer: Max pooling in cnn.

[4] D. Cournapeau.

[5] E. García-Portugués. Notes for predictive modeling.

[6] A. S. Gillis. *How AI speech recognition shows bias toward different accents* [Online]. Available: `https://www.techtarget.com/WhatIs/feature/How-AI-speech-recognition-shows-bias-toward-different-accents`, 2024. [Accessed: 05 June 2025].

[7] L. Jiménez. *Survey: Which Accents Does AI Find Hardest to Understand?* [Online]. Available: `https://guide2fluency.com/language-resources/which-accents-ai-hardest-to-understand/`, 2024. [Accessed: 06 June 2025].

[8] Q. Jin, Schultz, and Waibel. Speaker identification using multilingual phone strings. *IEEE International Conference on Acoustics Speech and Signal Processing*, 2002. doi: 10.1109/icassp.2002.1005697

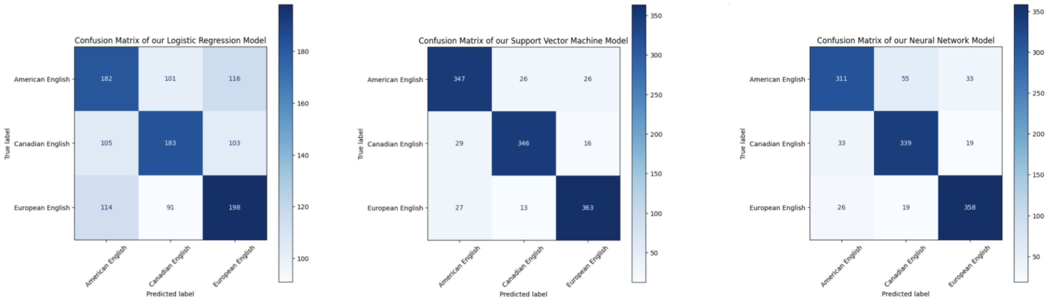
[9] C. Olah. Understanding lstm networks, Aug 2015.

[10] T. Song, L. T. Nguyen, and T. V. Ta. Mpsa-densenet: A novel deep learning model for english accent classification. *Computer Speech amp; Language*, 89:101676, Jan 2025. doi: 10.1016/j.csl.2024.101676

[11] Voxforge.org. Free speech... recognition (linux, windows and mac) - voxforge.org.
urlhttp://www.voxforge.org/. accessed 06/5/2025.

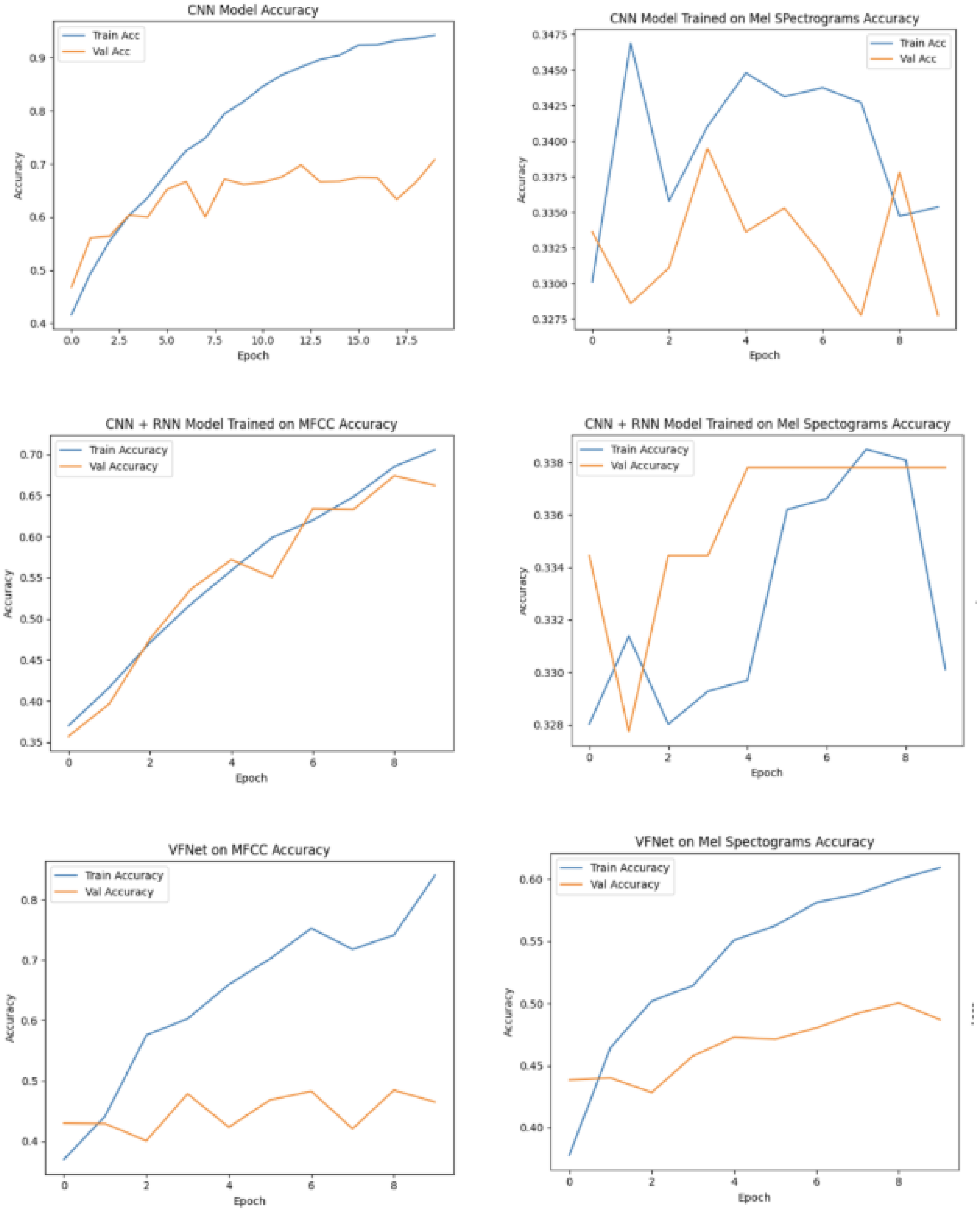
[12] Z. Zhang, Y. Wang, and J. Yang. Accent recognition with hybrid phonetic features. *Sensors*, 21(18):6258, Sep 2021. doi: 10.3390/s21186258

[13] Z. Zhang, Y. Wang, J. Yang, and X. Chen. Accent recognition with hybrid phonetic features. *Sensors*, 21(18):6258, Sep 2021. doi: 10.3390/s21186258



heightheight

Figure 6: Confusion Matrices for Logistic Regression (left), SVM (middle), and feed forward network (right).



heightheight

Figure 7: Training and validation accuracies against epoch for base CNN (top), CNN+RNN (middle), and VFNet (bottom).