TIMIT and NTIMIT Phone Recognition Using Convolutional Neural Networks

Cornelius Glackin[1](✉), Julie Wall[2], Gérard Chollet[1], Nazim Dugan[1], and Nigel Cannings[1]

[1] Intelligent Voice Ltd., London, UK
neil.glackin@intelligentvoice.com
[2] University of East London, London, UK

Abstract. A novel application of convolutional neural networks to phone recognition is presented in this paper. Both the TIMIT and NTIMIT speech corpora have been employed. The phonetic transcriptions of these corpora have been used to label spectrogram segments for training the convolutional neural network. A sliding window extracted fixed sized images from the spectrograms produced for the TIMIT and NTIMIT utterances. These images were assigned to the appropriate phone class by parsing the TIMIT and NTIMIT phone transcriptions. The GoogLeNet convolutional neural network was implemented and trained using stochastic gradient descent with mini batches. Post training, phonetic rescoring was performed to map each phone set to the smaller standard set, i.e. the 61 phone set was mapped to the 39 phone set. Benchmark results of both datasets are presented for comparison to other state-of-the-art approaches. It will be shown that this convolutional neural network approach is particularly well suited to network noise and the distortion of speech data, as demonstrated by the state-of-the-art benchmark results for NTIMIT.

Keywords: Phone recognition · Convolutional Neural Network
TIMIT · NTIMIT

1 Introduction

Automatic Speech Recognition (ASR) typically involves multiple successive layers of hand-crafted feature extraction steps. This compresses the huge amounts of data produced from the raw audio ensuring that the training of the ASR does not take an unreasonably long time. With the adoption of GPGPUs and the so-called Deep Learning trend in recent years, data-driven approaches have overtaken the more traditional ASR pipelines. This means that audio data is automatically processed in its frequency form (e.g. spectrogram) with a Deep Neural Network (DNN), or more appropriately, since speech is temporal, a Recurrent Neural Network (RNN). These networks automate the feature extraction process and can be trained quickly with GPUs. The RNN then converts the spectrogram directly to phonetic symbols or text [1].

Of all the deep-learning technologies, Convolutional Neural Networks (CNNs) arguably demonstrate the most automated feature extraction pipeline. In this paper we have employed CNNs to process the spectrograms as they are well-known for their

M. De Marsico et al. (Eds.): ICPRAM 2018, LNCS 11351, pp. 89–100, 2019.
https://doi.org/10.1007/978-3-030-05499-1_5

state of the art performance for image processing tasks, and this has been adapted for learning the acoustic model component of an ASR system. The acoustic model is responsible for extracting acoustic features from speech and classifying them to symbol classes. The phonetic transcription of both the TIMIT and NTIMIT corpora will be used as the ‘ground truths’ for training, validation and testing the CNN acoustic model. This consists of spectrograms as input and the phones as class labels. The work presented in this paper builds on previously published work [2] but extends from TIMIT classification to additionally employ the NTIMIT speech corpus to achieve state of the art results. In addition, this paper includes more details regarding the analysis of errors in phone classification.

CNNs are inspired by receptive fields in the mammalian brain and have been typically employed for the classification of static images [3]. Mammalian receptive fields can be found in the V1 processing centres of the cortex responsible for vision and in the cochlear nucleus of the auditory processing areas [4]. They work by transforming the firing of sensory neurons depending on spatial input [5]. Typically, an inhibitory region surrounds the receptive field and suppresses any stimulus which is not captured by the bounds of the receptive field. In this way, receptive fields play a feature extraction role.

Fukushima developed the Neocognitron network inspired by the work of Hubel and Wiesel on receptive fields [6, 7]. The Neocognitron network provided an automated way for implementing feature extraction from images. This approach was advanced by LeCun by incorporating the convolution operations now commonplace in CNNs. It was LeCun that coined the term CNN, the most notable example of which was the LeNet5 architecture which was used to learn the MNIST handwritten character data set [8, 9]. LeNet5 was the first network to use convolutions and subsampling or pooling layers.

Since Ciresan’s innovative GPU implementation in 2011 [10], CNNs are now typically trained in parallel with a GPU. Selection of a suitable CNN architecture for classification of any data is dependent upon the amount of available resources and data required to train the networks. The depth of the architecture is positively correlated with the amount of training data required to train them. Additionally, for large network architectures, the number of parameters to be optimized becomes a factor. Perhaps the most efficient architecture to date is the GoogLeNet CNN. It has a relatively complex network structure as compared to AlexNets or VGG networks.

GoogLeNet’s main contribution is that it uses Inception modules, within which are convolution kernels extracting features of different sizes. There are 1×1, 3×3, and 5×5 pixel convolutions, typically an odd number so that the kernel can be centred on top of the image pixel in question. 1×1 convolutions are also used to reduce the dimensions of the feature vector, ensuring that the number of parameters to be optimised remains manageable. GoogLeNet’s reduced number of parameters was a significant innovation to the field. This is in comparison to its fore-runner AlexNet, which has 60 million parameters to GoogLeNet’s 4 million [3]. The pooling layer reduces the number of parameters, but its primary function is to make the network invariant to feature translation. The concatenation layer constructs a feature vector for processing by the next layer. This architecture was used in [2] and was retained here for comparison purposes. Intuitively it is arguably the ‘right’ depth as far as the volume of images available to train it. The volume of images used to train the networks presented here are slightly larger than the 2011 ImageNet dataset [19] that the GoogLeNet architecture was optimized to tackle.

2 Phone Recognition with TIMIT and NTIMIT

In this work, spectrograms derived from the TIMIT corpus have been used to train a CNN to perform acoustic modelling [11]. The TIMIT corpus, which has an accurate phone transcription, was designed in 1993 as a speech data resource for acoustic phonetic studies. It has been used extensively for the development and evaluation of ASR systems. TIMIT is the most accurately transcribed speech corpus in existence as it contains not only transcriptions of the text but also contains accurate timing of phones. This is impressive given that the average English speaker utters 14–15 phones a second. The corpus contains the broadband recordings of 630 people (438 male/192 female) reading ten phonetically rich sentences of eight major dialects of American English. It includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit 16 kHz speech waveform file for each utterance. TIMIT was commissioned by DARPA and worked on by many sites, including Texas Instruments (TI) and Massachusetts Institute of Technology (MIT), hence the corpus' name. Figure 1 features a spectrogram and illustrates the accuracy of the word and phone transcription for one of TIMIT's core training set utterances. A sliding window, shown in grayscale, moves over the 16 kHz (Short-Term Fourier Transform) STFT-based spectrogram. The resulting 256 * 256 pixel spectrogram patches are placed into phone classes according to the TIMIT transcription for training, validation and testing.

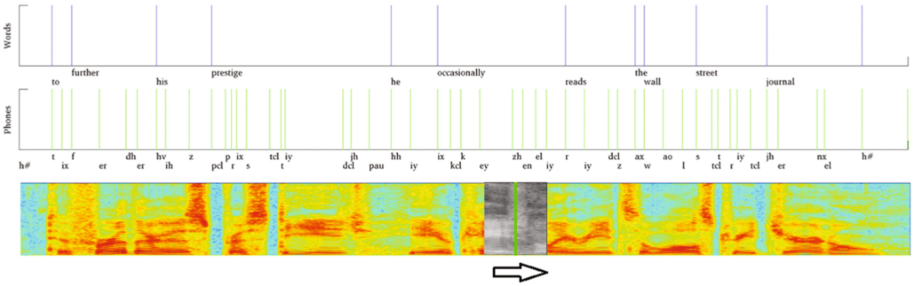

Fig. 1. Presentation of the images for GoogLeNet training [2].

A spectrogram was generated for every 160 samples. For 16 kHz of encoded audio, this corresponds to 10 ms as per the standard resolution required to find all the acoustic features the audio contains. The phone transcriptions are utilised to label each spectrogram according to the phone to which its centre most closely aligns to. Another option would have been to use the centre of the ground truth interval and calculate the Euclidean distance between the centre of the phone interval and the window length. However, this would have made assumptions about where the phone is centred within the interval, requiring an additional computationally expensive step in the labelling of the spectrogram windows.

The previous figure also illustrates the data preparation for the training, validation and testing sets using the sliding window approach. It shows how the phonetic transcription is used to label the 256 × 256 greyscale spectrogram patches as the sliding window passes over each of the TIMIT utterances. The labelled spectrograms were sorted according to the phone class to which they belong within each of the training, validation and testing sets. In the TIMIT corpus we use the standard core training setup. STFT-type spectrograms were used in particular as they could align the acoustic data and the phonetic symbols with timing that was as accurate as possible. NVIDIA's cuFFT library was used for the FFT component of spectrogram generation [18]. The distribution of the phones that were generated according to the TIMIT ground truth are shown in Fig. 2.

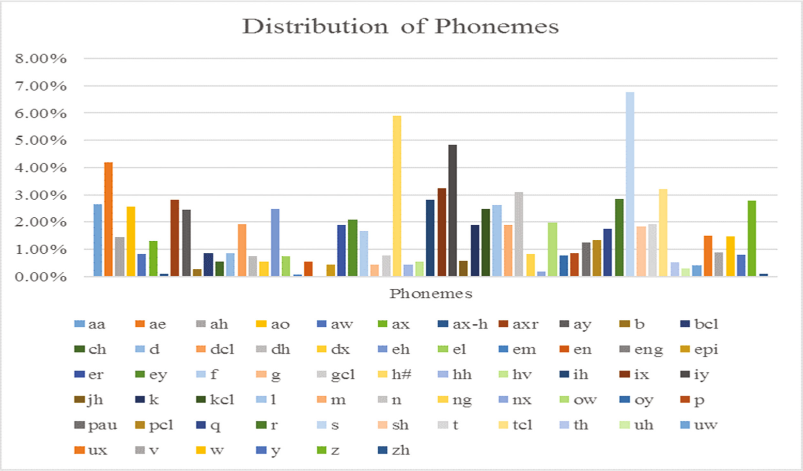

Fig. 2. Distribution of phones within the TIMIT transcription, note that the bars correspond to the alphabetically ordered phones in the key [2].

As evident from Fig. 2, the two largest classes are 's' and 'h#' (silence). Silence occurs at the beginning and end of each utterance. The distribution is highly non-uniform which makes training of the phone classes in the CNN challenging. The training data is the standard TIMIT core set, and the standard test set sub-directories DR1-4 and DR5-8 were used for validation and testing respectively. This partitioning resulted in 1,417,588 spectrogram patches in the training set, as well as 222,789 and 294,101 spectrograms in the validation and testing sets respectively.

2.1 TIMIT GoogLeNet Training and Inferencing

The GoogLeNet acoustic model in this work was trained with Stochastic Gradient Descent (SGD). Prior to the advent of Deep Learning, gradient descent was usually performed using the full batch of training samples in order to adapt the network weights in each training step. However, this approach is not easily parallelizable and thus cannot be implemented efficiently on a GPU. In contrast, SGD computes the gradient of the parameters on a single or few (mini batch) training samples. For larger datasets, such as the one utilised in this work, SGD performs qualitatively as well as batch methods but are faster to train.

We used a stepped learning rate with a 256 sample mini batch size. The resultant training graph is illustrated in Fig. 3. The GoogLeNet architecture produces a phone class prediction at three successive points in the network (loss1, loss2, and loss3). The NVIDIA DIGITS deep learning framework [20] which was employed for this implementation, reports the top-1 and top-5 predictions for each of these loss outputs. loss3 (the last network output) reports the highest accuracy which is 71.65% for classification of the 61 phones. For loss3/top-5 the accuracy is reported as 96.27%, which means that the correct phone was listed in the top five network output classifications over 96% of the time. As mentioned earlier, each spectrogram window contains 4–5 phones on average, and our results show that in the majority of cases these other phones were indeed correctly being reported in the top-5 network outputs.

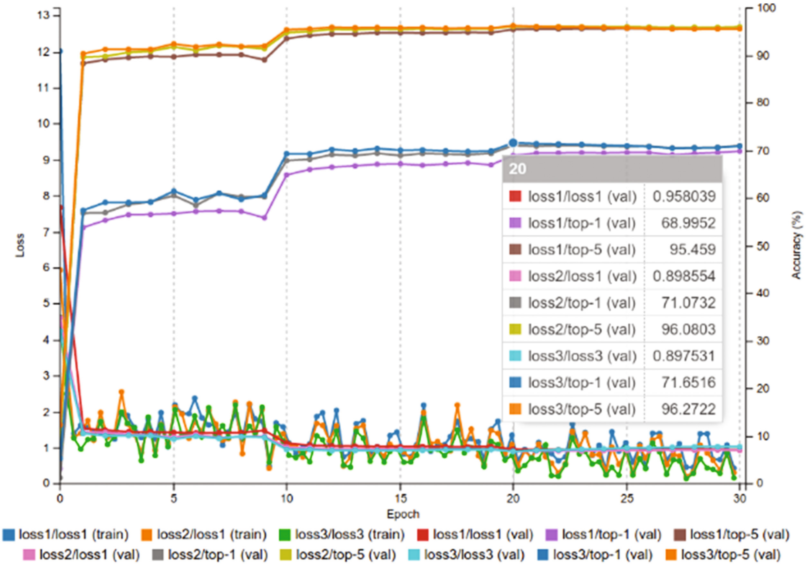

Fig. 3. TIMIT SGD training [2].

The network is trained using ~1.4 M images and uses the validation set (~223,000 images) to check training progress. A separate set (~294,000 images) was used to test the system, and the standard test set sub-directories DR1-4 and DR5-8 were used for validation and testing respectively. The validation set is kept separate from the training data and is only used to monitor the progress of the training, and to stop training if overfitting occurs. The highest value of the validation accuracy is used as the final system result and this was achieved at epoch 20, as can be seen in Fig. 3. With this final version of the system, we performed inferencing over the test set, Fig. 4 shows an example prediction from the system for a single sample of unseen test data. The output of the inferencing process contains many duplicates of phones due to the small increments of the sliding window position.

The 256 ms spectrogram windows typically can contain between 4 and 5 phones, with the average speaker uttering approximately 15 phones per second. The pooling layers in the CNN acoustic model provide flexibility in where the feature under question (phones in this case) can be within the 256 * 256 spectrogram image. This is useful for different orientations and scales of images in image classification and is also particularly useful for phone recognition where it is likely there will exist small errors in the training transcription.

During inferencing, the CNN acoustic model makes softmax predictions of all the phone classes for each of the test spectrograms, at three successive output stages of the network (Loss 1 to 3). We conducted some graphical analysis of the output confidences of the phones, colour coding the outputs for easier readability of the results, see Fig. 4. As can be seen from the loss-3 (accuracy), the network makes crisp classifications of usually only a single phone at a time. Given that this is unseen data, and that the comparison with the ground truth is good, we are confident that this network is an effective way to train an acoustic model.

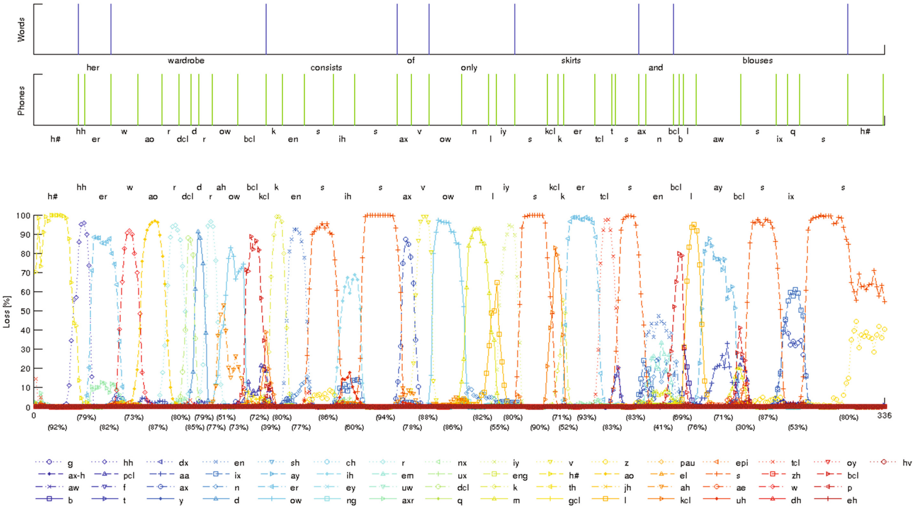

Fig. 4. Softmax network outputs for a test utterance [2].

2.2 Post-processing and Rescoring

Post-processing of the classification output was performed to remove duplicates produced by the fine granularity of the sliding window. It is the convention in the literature when reporting results for the TIMIT corpus to re-score the results for a smaller set of phones [12]. The phoneticians that scored TIMIT used 61 phone symbols. However, many of these phones in TIMIT are not conventionally used by other speech corpora and ASR systems. There are phone symbols called closures e.g. pcl, kcl, tcl, bcl, dcl, and gcl for example, which simply refer to the closing of the mouth before release of closure resulting in the p, k, t, b, d, or g phones being uttered respectively. Most acoustic models map these to the silence symbol ‘h#’. Remapping the output of the model inferencing for the unseen testing data to the smaller 39 phone set (See Table 1 for the rescroing mapping), resulted in a significant increase in accuracy from 71.655 (shown in Fig. 3) to 77.44% after rescoring.

This result, while not quite exceeding the 82.3% result reported by Graves [13] with bidirectional LSTMs, or the DNN with stochastic depth [14] which achieved a competitive accuracy of 80.9%, is nevertheless still comparable. The novel approach of Zhang et al. advocates an RNN-CNN hybrid based on MFCC features using conventional MFCC feature extraction with an RNN layer before a deep CNN structure [15]. This hybrid system achieved an impressive 82.67% accuracy. It is not surprising to us that the current state of the art is with a form of CNN [16] with an 83.5% test accuracy. Notably, a team from Microsoft recently presented a fusion system that achieved the state of the art accuracy for the Switchboard corpus. Each of the three ensemble members in the fusion system used some form of CNN architecture, particularly at the feature extraction part of the networks. It is becoming clear that CNNs are demonstrating superiority over RNNs for acoustic modelling.

Table 1. Rescoring mapping of phone symbols (61 to 39 symbols).

aa ->aa	epi ->sil	ow ->ow
ae ->ae	er ->er	oy ->oy
ah ->ah	ey ->ey	p ->p
ao ->aa	f ->sil	pau ->sil
aw ->aw	g ->g	pcl ->sil
ax ->ah	gcl ->sil	q ->sil
ax-h ->ah	h# ->sil	r ->r
axr ->er	hh ->hh	s ->s
ay ->ay	hv ->hh	sh ->sh
b ->b	ih ->ih	t ->t
bcl ->sil	ix ->ih	tcl ->sil
ch ->ch	iy ->iy	th ->th
d ->d	jh ->jh	uh ->uh
dcl ->sil	k ->k	uw ->uw
dh ->dh	kcl ->sil	ux ->uw
dx ->dx	l ->l	v ->v
eh ->eh	m ->m	w ->w
el ->l	n ->n	y ->y
em ->m	ng ->ng	z ->z
en ->n	nx ->n	zh ->sh
eng ->ng		

2.3 NTIMIT Experiments

For commercial speech recognition applications, it is vital to evaluate how ASR performs in the telephone setting. To ensure a fair comparison with the previously published TIMIT speech recognition paper we decided to train the system with NTIMIT [17]. NTIMIT (Network TIMIT) is the result of transmitting the TIMIT database over the telephone network. This results in loss of information in the signal due to the smaller passband of the telephone network, as well as distortions due to the network transmission. To quantify the effects of the network on the original TIMIT data, we calculated the normalized energy of TIMIT and NTIMIT in the [0, 8 kHz] frequency band. Figure 5 shows the absolute value of the normalized amplitudes of the TIMIT (top) and NTIMIT (bottom) corpora.

Both signals are shown here in the [0,8 kHz] range for the 16 kHz sample rate of the original audio as per Nyquist sampling theory. As can be seen from the figure, for TIMIT there is a smooth variation in signal amplitude from the low frequencies to the high frequencies in the entire frequency range. For the NTIMIT signal, it can be seen that after 3400 Hz the gradient of the amplitude stops varying (flatlines) and drops quickly before flatlining again at approximately 6800 Hz (which is likely due to some frequency folding). Consequently, it can be seen that there is very little useful information in the NTIMIT signal above 3400 Hz. Hence for the purposes of the NTIMIT experiments, we first downsampled all of the audio files to 6.5 kHz to ensure that the entire 256 × 256 pixel range resulting spectrogram input to the CNN represents the available speech signal.

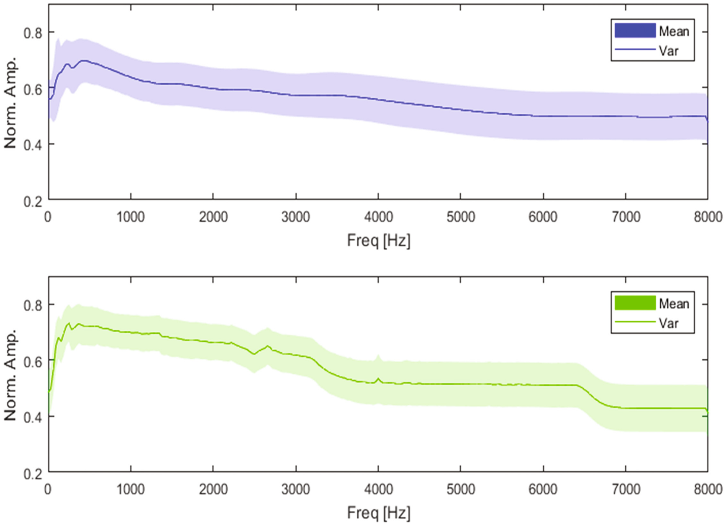

Fig. 5. Spectral profile of TIMIT (top) and NTIMIT (bottom). TIMIT shows a smooth transmission from low to high frequency, whereas NTIMIT has little useful information encoded above 3.3 kHz.

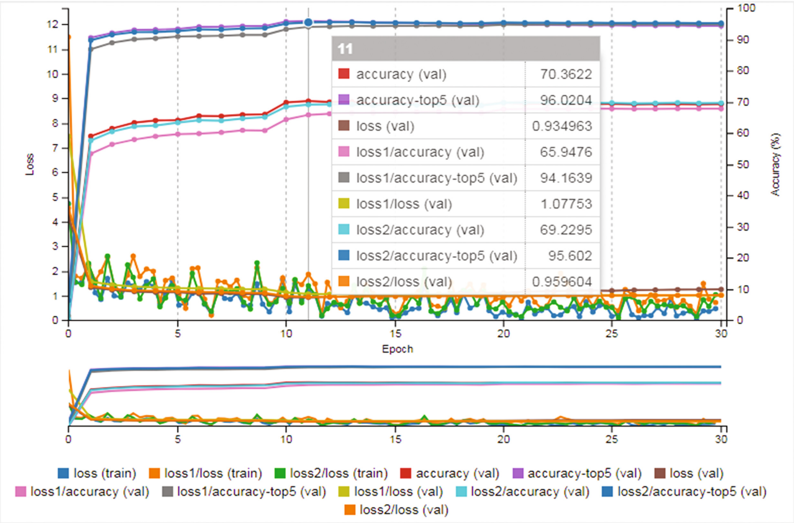

Fig. 6. NTIMIT stochastic gradient descent training.

A GoogLeNet CNN was trained from scratch with spectrograms generated from the downsampled NTIMIT training data. Figure 6 shows progress of the training results in terms of accuracy and loss. The best validation accuracy of 70.36% occurs at epoch 11, the top-5 accuracy achieved is 96.02%. Once again, rescoring was employed to convert the 61 phone set to 39 phones and the accuracy increased to 73.63% as a result.

Figure 7 shows the confusion matrix for the 61 phone set of the NTIMIT test results. The confusion matrix is calculated by accumulating the classification confidences for all 294,000 images in the NTIMIT test set and is informative with regards to the typical pattern of misclassification, in particular with regards to the top-5 performance. It can be quickly understood why the top-5 accuracy is so high, when looking across the rows of the confusion matrix, it can be seen that there are no more than 5 significant classifications for any actual phone. A typical misclassification occurs when comparing the classification of the ‘aa’ phone as the ‘r’ sound, although interestingly the results indicate that ‘r’ is rarely mistaken for ‘aa’. The original 61 phone set had many symbols for silence such as ‘pau’, ‘h#’, ‘sil’ and ‘q’, likely to describe the context of the silence in the original corpus and whether it was a silence at the beginning or end of an utterance (‘h#’), a pause (‘pau’), etc. However, ultimately an absence of speech regardless of context is easy to mistake when the spectrogram window is small and hence, as can be seen, misclassifications of these various silences are common. Given this, it is understandable why mapping all these silences as well as many of the closure phones, e.g. ‘bcl’, ‘dcl’, ‘kcl’ to ‘sil’ significantly improves the recognition accuracy. In order to assess how the rescoring improves matters we generated the confusion matrix for the rescored phones in Fig. 8.

Figure 8 shows the confusion matrix for the rescored phone matrices, for the purposes of isolating the remaining phone misclassifications we removed the now amalgamated ‘sil’ symbols. The main misclassifications can now be seen from the

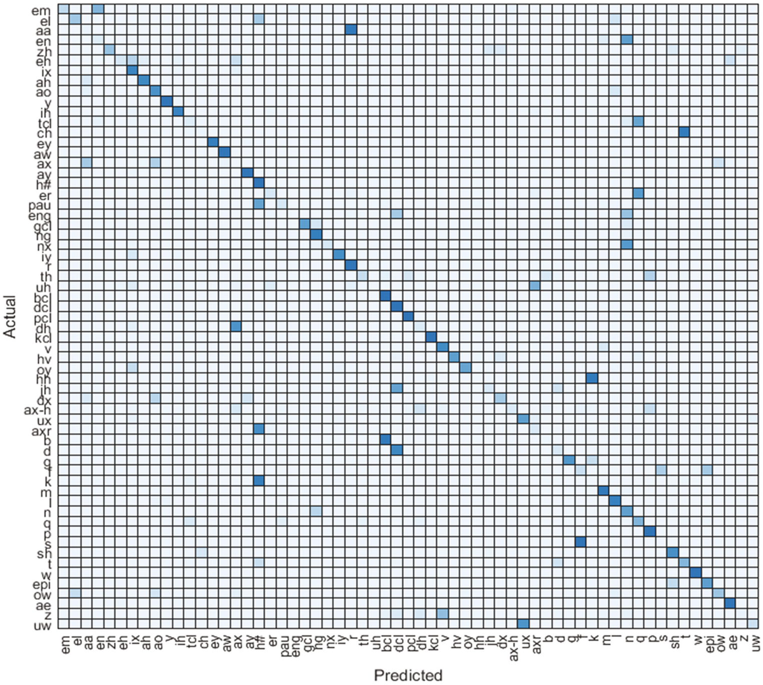

Fig. 7. Confusion matrix of actual against predicted phones.

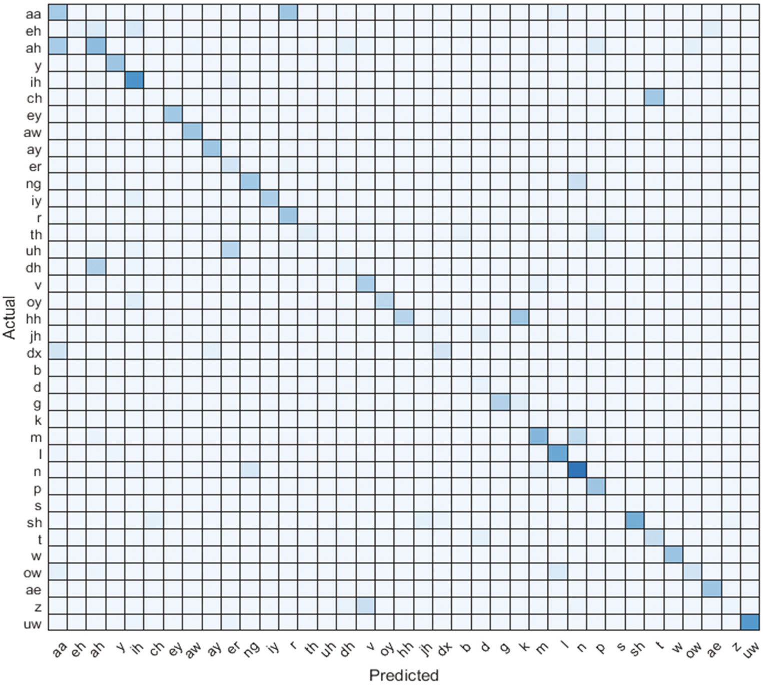

Fig. 8. Confusion matrix of actual against predicted phones for rescored phones.

figure. The ‘aa’ and ‘r’ symbols were not rescored and so this misclassification remains, similarly ‘ch’ and ‘sh’ are confused by the system. In general, these misclassifications are intuitively understandable and in an ASR system with a good quality language model, many of these typical misclassifications will be overcome by the probability distribution of phonetic sequences inherent in the language model.

3 Conclusions

This paper extended previously published work [2] concerning TIMIT phone classification using CNNs to the case of noisy telephone speech (NTIMIT corpus). Typically, we have found that NTIMIT results in the literature are around 10% less than for TIMIT. However, we have found that we are within 4% of the TIMIT CNN model performance in our tests which suggests that the CNN approach is much more noise robust. To our knowledge the NTIMIT benchmark results reported here are the state of the art.

CNNs are becoming a favoured method for feature extraction of speech data, now more commonly used as input layers to DNN-based ASR systems when used as input to an end to end speech recognition system. The reason for this is that CNNs automate the feature extraction process. We also found in this study, that CNNs seem to be more robust to challenging, noisy and distorted data of the type that is found in the NTIMIT corpus, as compared to other approaches.

References

1. Hannun, A., et al.: Deep speech: scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
2. Glackin, C., Wall, J., Chollet, G., Dugan, N., Cannings, N.: Convolutional neural networks for phoneme recognition. In: 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM) (2018)
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances Neural Information Processing System (NIPS), pp. 1097–1105 (2012)
4. Shamma, S.: On the role of space and time in auditory processing. Trends Cogn. Sci. **5**(8), 340–348 (2001)
5. Paulin, M.G.: A method for analysing neural computation using receptive fields in state space. Neural Netw. **11**(7), 1219–1228 (1998)
6. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980)
7. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in cat’s visual cortex. J. Physiol. (London) **160**, 106–154 (1962)
8. LeCun, Y., et al.: Handwritten digit recognition with a back-propagation network. In: Advances Neural Information Processing System (NIPS), pp. 396–404 (1990)
9. LeCun, Y.: The MNIST database of handwritten digits (1998). http://yann.lecun.com/exdb/mnist/

10. Ciresan, D.C., Meier, U., Masci, J., Gambardella, L., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: International Joint Conference on Artificial Intelligence (IJCAI), vol. 22, no. 1, pp. 1237–1242 (2011)
11. Garofolo, J., et al.: TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Linguistic Data Consortium, Web Download, Philadelphia (1993)
12. Lopes, C., Perdigao, F.: Phone recognition on the TIMIT database. In: Speech Technologies/Book 1, pp. 285–302 (2011)
13. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustic Speech Signal Process (ICASSP), pp. 6645–6649 (2013)
14. Chen, D., Zhang, W., Xu, X., Xing, X.: Deep networks with stochastic depth for acoustic modelling. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1–4 (2016)
15. Zhang, Z., Sun, Z., Liu, J., Chen, J., Huo, Z., Zhang, X.: Deep recurrent convolutional neural network: improving performance for speech recognition. arXiv 1611.07174 (2016)
16. Tóth, L.: Phone recognition with hierarchical convolutional deep maxout networks. EURASIP J. Audio, Speech, Music Process. **1**, 1–13 (2015)
17. Jankowski, C., Kalyanwamy, A., Basson, S., Spitz, J.: NTIMIT: a phonetically balanced, continuous speech, telephone bandwidth speech database. In: IEEE International Conference on Acoustic Speech Signal Processing (ICASSP) (1990)
18. CUDA CUFFT Library: NVIDIA (2007). https://docs.nvidia.com/cuda/cufft/index.html
19. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (2011). http://image-net.org/challenges/LSVRC/2011/index
20. NVIDIA DIGITS Interactive Deep Learning GPU Training System. https://developer.nvidia.com/digits