Bangla Short Speech Commands Recognition Using Convolutional Neural Networks

Shakil Ahmed Sumon, Joydip Chowdhury, Sujit Debnath, Nabeel Mohammed
and Sifat Momen
Department of Electrical and Computer Engineering, North South University
Plot-15, Block-B, Bashundhara, Dhaka, Bangladesh
shakil.sumon@northsouth.edu, joydip.chowdhury@northsouth.edu, sujit.debnath@northsouth.edu,
nabeel.mohammed@northsouth.edu and sifat.momen@northsouth.edu

***Abstract*—Despite being one of the most widely spoken languages of the world, no significant efforts have been made in Bangla speech recognition. Speech recognition is a difficult task, particularly if the demand is to do so in noisy real-life conditions. In this study, Bangla short speech commands data set has been reported, where all the samples are taken in the real-life setting. Three different convolutional neural network (CNN) architectures have been designed to recognize those short speech commands. Mel-frequency cepstral coefficients (MFCC) features have been extracted from the audio files in one approach whereas only the raw audio files have been used in another CNN architecture. Lastly, a pre-trained model which is trained on a large English short speech commands data set has been fine-tuned by retraining on Bangla data set. Experimental results reveal that the MFCC model shows better accuracy in recognizing Bangla short speech commands where, surprisingly, the model predicting on raw audio data is very competitive. The models have shown proficiency in identifying single syllable words but encounter difficulties in recognizing multi-syllable commands.**

Keywords*—*Automatic Speech Recognition, Bangla Speech Recognition, Short Speech Commands, MFCC, Transfer learning, Convolutional neural network

I. Introduction

Speech is the most important aspect of communication among human beings. Human beings can naturally identify voices from different sets of sounds. However, the desire to automate tasks require constant interaction between humans and machines, and for that, automatic speech recognition (ASR) by machines gained a lot of attention over the last few decades [1]. Voice commands as input to machines have numerous advantages as it is quick in nature, hands-free and can be given remotely more feasible. People can easily create documents and control devices with the help of ASR. It makes technologies like home automation and remotely controlled unmanned vehicle a lot easy to operate. ASR can provide great help to businesses which have to provide live customer services through phone calls.

However, the above-mentioned applications are mostly available in English. Despite Bangla being the fifth most widely spoken language of the world, very few attempts have been taken for effective Bangla speech recognition. There are about 250 million people around the world who speak Bangla as their first language and 160 million of them reside inside Bangladesh which is a country of many contraints and challenges. A large portion of the population has access to mobile phones. However, most of them are not computer literate and typing and interacting in English is impractical for them. Bangla speech recognition applications can improve their interaction with technology and thus assist in improving the standard of living.

In this paper, we propose a convolutional neural network (CNN) based architecture for Bangla short speech recognition. We have collected utterances of 10 different words in real life noisy conditions and trained CNN-based models on them. Our paper is structured as follows: Section 2 discusses some relevant literature, Section 3 presents the collected dataset, Section 4 gives a brief overview about CNN, Section 5 the experimental setup is detailed, Section 6 provides the results and discussions on the findings and Section 7 concludes the paper.

II. Related Work

Convolutional neural networks have been used previously for speech recognition tasks. In this study [2], they found that CNNs reduce error rate by 6-10 percent on the TIMIT phone recognition dataset compared to deep neural networks (DNN). Moreover, they did some experiments using full weight sharing (FWS) and limited weight sharing (LWS) schemes and found out that LWS is more effective as it can learn feature patterns of different frequency bands. In this paper [3], a very deep convolutional neural network has been applied for noisy speech recognition. They experimented with the sizes of filters, input feature map, and pooling layers to find the optimum setup. They evaluated the proposed model in Aurora4task and AMI meeting transcription dataset and found out that very deep CNNs reduce word error rate (WER) significantly for noisy speech recognition. However, in [4], they have explored something significantly interesting. They have trained a deep belief network (DBN) with unlabeled data of call routing task and used the learned features form this network to initialize a feed-forward neural network

978-1-5386-8207-4/18/$31.00 ©2018 IEEE

which fine-tunes itself by back-propagation. They, then have compared three classic classifiers: support vector machine (SVM), maximum entropy (MaxEnt) and boosting with the DBN initialized network. They claim that the DBN initialized model has gained the accuracy which is equal to the best of the other baseline models. This study [5] outlines the possibility of using linear and log-linear stacking methods for ensemble learning for speech recognition using CNN, recurrent neural network (RNN) and fully connected DNN. [6] uses a DBN pre-trained neural network for large vocabulary speech recognition and claimed that it outperformed the baseline Gaussian mixture model-Hidden Markov model (GMM/HMM) which was built on a much larger dataset than the one they used.

Although, when it comes to Bangla speech recognition, not much work has been done. There are, however, a few praiseworthy attempts. [7] proposed a model which calculates the linear predictive coding (LPC) and cepstral coefficients to form vector quantization which is then fed to an artificial neural network (ANN). In this study [8], they introduced a Bangladeshi accented Bangla digit automatic speech recognition system. They used mel-frequency cepstral coefficients (MFCC) as a feature extraction method and feed the MFCC vectors to a HMM based classifier for recognition. However, in [10], they also built a digit recognizer and used MFCC analysis as a feature extractor but they feed those features to a back-propagation neural network instead. In this paper [9], they discussed four techniques of automatic speech recognition for Bangla words: MFCC, LPC, GMM and dynamic time wrapping (DTW). They compared these techniques in terms of recognition rate and elapsed time.

III. Theoretical overview of CNN

The convolutional neural networks (CNN) are a simple extension of the multi-layer perceptron model which can be considered as a diverse version of the standard neural networks [2] [10]. In this section, we briefly discuss the architecture and other dimensions of traditional CNNs which are used mainly for speech recognition purposes.

A. CNN Layer Architecture

A typical CNN for automatic speech recognition (ASR) introduces a different kind of network infrastructure compared to other artificial neural networks (ANN) and deep neural networks (DNN). However, traditional CNN consists of layers stacked together which are an input layer, a group of convolutional and pooling layers, several fully connected layers, and finally an output layer [10]. The convolutional and pooling layers, followed by fully connected layers are the main differences of CNN compared to other neural networks, and this kind of special layer architecture has significant practical consequences in terms of speech recognition.

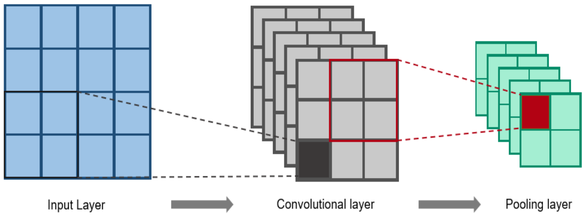

Fig. 1. Convolutional Neural Network

B. Convolution Layer

A convolutional layer organizes hidden layer such a way that can take the advantages of input layer which is in form of a two-dimensional input data. Each hidden unit of convolutional layer processes only a small part of the whole input space rather than connecting to all the inputs coming from the previous input layer and it applies some arbitrary filters of an arbitrary dimension which result in feature map. From the feature map, CNN can understand local features of the data.

C. Poolling Layer

Another layer is connected with one convolutional layer which is called pooling layer. Pooling layer reduces the dimensionality of the extracted feature maps by applying a window of an arbitrary size which is called stride. It can extract either max, average or the sum of the windows. In this case, max pooling is used which extracted the highest values for each window in the feature map.

D. Activation using rectified linear unit (ReLU)

Previously, logistic sigmoid and hyperbolic tangent have been used widely as non-linear activation functions in deep neural architectures like CNNs. But, recently some alternative solutions have emerged and the application of Rectified Linear Units (ReLUs) is one of the most commonly used alternatives. Additionally, ReLU is the common alternative solution, since it has several advantages over typical activation functions which are faster computation and more efficient gradient propagation, biological plausibility and sparse activation structure [11].

E. Mel Frequency Cepstral Coefficient (MFCC)

In automatic speech recognition, many feature selection techniques are widely used, and prominent among them is is Mel Frequency Cepstral Coefficient (MFCC). Generally, the MFCC feature extraction splits the audio signal into short timestamps, because audio signal varies too much in long timestamps. After the splitting audio signal into the short timestamps, the periodogram estimate of the power spectrum has been calculated for each frame. Then, the mel-filter bank is applied to the power spectrum and summed the filter energy. Furthermore, the logarithm of all filterbank energies has been calculated. Finally, the Discrete Cosine Transform (DCT) of the log filter bank energies are considered and keep the lower DCT coefficients of range 2-13 for ASR. Because the higher DCT coefficients downgrade the performance of ASR.

IV. Dataset

The resources for Bangla speech recognition are not widely available while there has been previous work on Bangla speech recognition, the dataset employed are not available publicly for research purpose. therefore we co9llected a small datset. We choose the words that are used on a daily basis. There is a lot of work has been done using the English data set provided by Google which is known as the Speech Commands dataset. The dataset contains 65,000 samples. The duration of each short words is not more than 1 second. The dataset contains 30 words with a variation of over 1000 utterances of the public, who contributed through the AIY website. This dataset was used to pre-train our proposed model. However, we choose only 10 classes from the English data set. The short speech words in the mentioned dataset are:

TABLE I
Short speech words in the mentioned data set

Bed	Go	On	No	One
Six	Three	Two	Yes	Zero

We decided to choose words for our own dataset considering the similarity of the phonemes of each word that are available in the English data set. The reasoning behind choosing those particular words is to achieve a higher accuracy considering the model is already pre-trained on Speech commands data set and is admittedly a subjective procedure. We were not able to get a large number of data samples, as we collected the audio samples manually by going person to person. Our dataset contains 10 classes of data sample; the duration of each utterance is less than 2 seconds. We managed to get about 100 samples per class. The words we choose to train our model are:

TABLE II
Short speech words in the mentioned data set

1	agerta	আগেরটা(previous)
2	aste	আস্তে(slowly)
3	at	আট(eight)
4	baba	বাবা(father)
5	bame jao	বামেযাও(go left)
6	bari	বাড়ি(house)
7	basa	বাসা(home)
8	bon	বোন(sister)
9	bondho koro	বন্ধকর(stop)
10	boro	বড়(big)

V. Experimental Setup

A. MFCC Model

We have experimented with two kinds of feature extraction. We have extracted mel-frequency cepstral coefficients (MFCC) features from the audio files and feed the features to a convolutional neural network architecture; which we are considering to call the MFCC model. Before training our model with the MFCC inputs we normalized those inputs. The MFCC model has one convolutional layer with 5 filters and each of the filters has a stride of 1. The convolutional layer is associated with a softmax layer which is the last layer of our model architecture. Regularization techniques like dropout of 0.25 and kernel regularizers l2 have been used to prevent the model from overfitting.

B. Raw Model

We have taken the raw audio files and feed them to a similar CNN architecture which we will call the raw model. The raw model has a similar architecture with one convolutional layer and a softmax layer. We applied a dropout rate of 0.8 in this model, which is the only difference between the MFCC model and the Raw model. The inputs of this model are the first 10000 values of each of the audio files since most audio files incorporate silences after 1 second. The inputs of this model, similar to the previous one, are being normalized as well.

C. Pre-trained Model

Our another approach was to use a pre-trained model before training with our data. Google has released a dataset of English short speech commands. We have extracted the MFCC features of the audio files and have trained a neural network with those features. The neural network has three convolutional layers which are associated with max-pooling and batch normalization layers. The first convolutional layer has 128 filters and second and third layers have 64 filters each. The filters of all three layers have a size of 3 × 3. We have saved the weights of the model while training and have used the weights as a mean for transfer learning.

In all the above models we have used categorical cross-entropy as loss function and adadelta as the optimizer.

VI. Results And Discussion

We have trained the models using the training portion of the data and have tested them against the audio files which were being allocated for the testing purpose. Table 1 shows the performance of the models in terms of percentage accuracy. We see that the MFCC model performs better than the other models but the model suffers from overfitting by a significant margin. The other two models achieve comparable test performance without any overfitting.

However, we have run 1000 epochs over the whole data set to train the MFCC model. Figure 2 shows the epochs

TABLE III
Performance of the models in percentage

Model	***Training Accuracy***	***Testing Accuracy***
MFCC	85.44	74.01
Raw	69.08	71.44
Transfer	68.06	73.00

vs accuracy graph of the model. Moreover, we have plotted the loss generated in every epoch in figure 3.

The raw model is being trained by running 1000 epochs over the entire data set. The accuracy and corresponding loss generated in every epoch are being plotted in Figure 4 and 5 respectively.

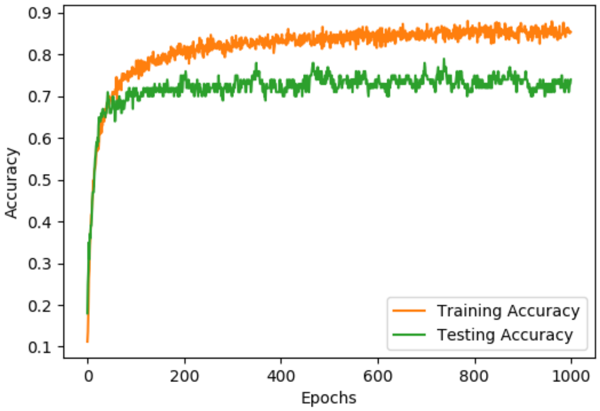

Fig. 2. Epochs vs accuracy of MFCC model

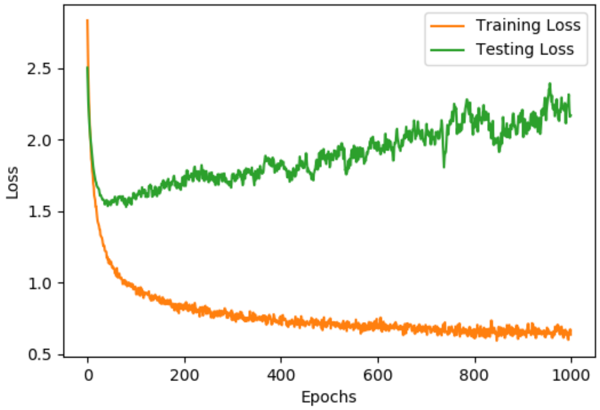

Fig. 3. Epochs vs loss of MFCC model

We loaded the weights generated while training the English dataset and retrained the model by running 1000 epochs over the Bangla short speech commands data set. Figure 6 and 7 represents the epochs vs accuracy and epochs vs loss graph of the transfer model respectively.

We have reserved 10 samples per class for testing purpose. We let the model predict the class label of these

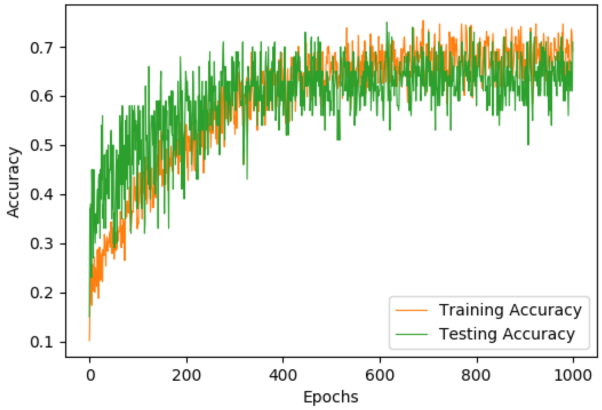

Fig. 4. Epochs vs accuracy of RAW model

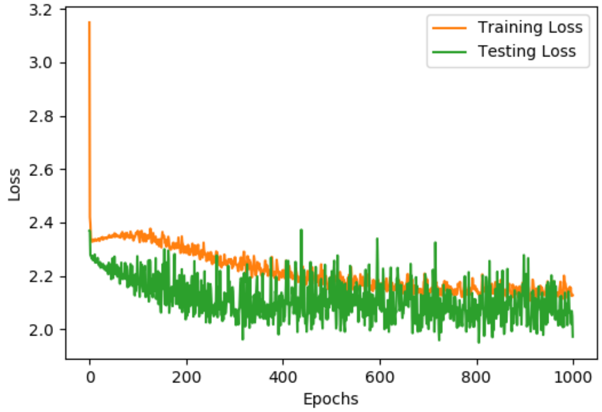

Fig. 5. Epochs vs loss of RAW model

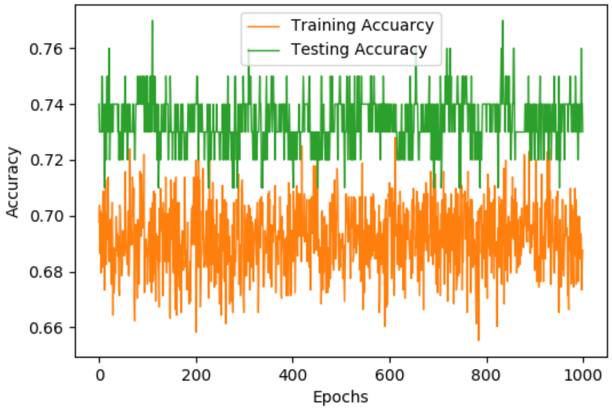

Fig. 6. Epochs vs accuarcy of Transfer model

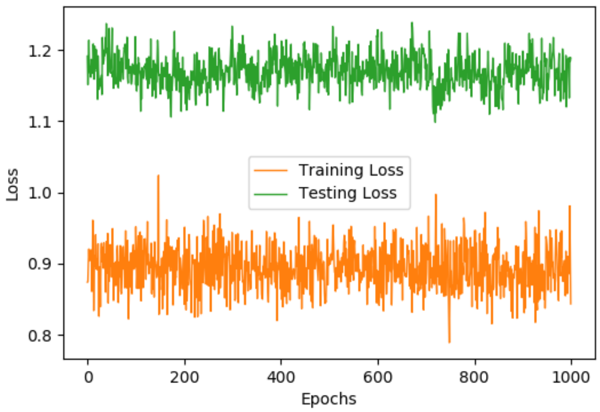

Fig. 7. Epochs vs loss of Transfer model

audio files and after that, we have generated confusion matrix from the predicted classes. Table 4 shows the confusion matrix of the MFCC model. We can see that the model does well in predicting ”bari”, ”basa”, ”bon”, and ”boro” but failed miserably in recognizing ”bame jao” and ”bondho koro”. The precision and recall per class of the model have been shown in table 5. Precision tells us about the classifier’s ability not to label a negative sample as a positive one. Recall tells us whether the model is good at finding all the positive samples in the dataset.

TABLE IV
Confusion Matrix of the MFCC model

	1	2	3	4	5	6	7	8	9	10
1	6	0	0	0	0	0	0	1	2	1
2	1	8	0	1	0	0	0	0	0	0
3	0	1	8	0	0	1	0	0	0	0
4	0	0	1	9	0	0	0	0	0	0
5	0	1	3	2	1	0	2	0	1	0
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	10	0	0
9	1	1	0	0	2	0	4	0	2	0
10	0	0	0	0	0	0	0	0	0	10

TABLE V
Precison and recall of MFCC model

Class Label	***Precision***	***Recall***
1	0.75	0.60
2	0.73	0.80
3	0.67	0.80
4	0.75	0.90
5	0.33	0.10
6	0.91	1.00
7	0.62	1.00
8	0.91	1.00
9	0.40	0.20
10	0.91	1.00

Table 6 and 7 represents the confusion matrix and precision and recall metrics of the raw model respectively. This model confuses between ”agerta” and at but does well in recognizing ”baba”, ”basa”, ”bon”, and ”boro”. The model recognizes ”bondho koro” as ”bon” as their first syllable is the same.

TABLE VI
Confusion Matrix of raw model

	1	2	3	4	5	6	7	8	9	10
1	5	1	3	0	0	0	0	1	0	0
2	0	8	1	0	0	0	0	1	0	0
3	2	0	8	0	0	0	0	0	0	0
4	0	0	0	9	0	1	0	0	0	0
5	2	1	2	0	3	0	0	0	0	2
6	1	0	0	0	0	9	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	10	0	0
9	2	0	0	0	1	0	0	6	1	0
10	0	1	0	0	0	0	0	0	0	9

TABLE VII
Precison and recall of raw model

Class Label	***Precision***	***Recall***
1	0.42	0.50
2	0.73	.80
3	0.57	.80
4	0.75	.30
5	1.00	.90
6	0.75	.30
7	0.90	.90
8	1.00	1.00
9	0.56	1.00
10	0.82	.90

Table 8 shows the confusion matrix of the transfer model and Table 9 represents the precision and recall metrics. It also does well in recognizing ”bari”, ”basa”, ”bon” but having difficulties identifying multi-syllable words ”bame jao” and ”bondho koro”. The model has mistaken ”agerta” as ”bame jao” and ”bari” as ”bondho koro”.

TABLE VIII
Confusion Matrix of Transfer model

	1	2	3	4	5	6	7	8	9	10
1	9	0	0	0	0	1	0	0	0	0
2	1	8	0	1	0	0	0	0	0	0
3	2	0	7	1	0	0	0	0	0	0
4	0	0	0	9	0	1	0	0	0	0
5	3	1	1	1	1	0	2	0	0	1
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	10	0	0
9	0	0	0	1	0	3	1	0	2	3
10	1	0	0	0	0	1	0	0	0	8

TABLE IX
Precison and recall of Transfer model

Class Label	***Precision***	***Recall***
1	0.56	0.90
2	0.89	.80
3	0.88	.70
4	0.69	.90
5	1.00	.10
6	0.62	1.00
7	0.77	1.00
8	1.00	1.00
9	1.00	.20
10	0.67	.80

VII. Conclusion And Future Work

The proposed research had been done with a relatively small dataset of Bangla short speech commands. The initial approach of the experiment was to observe how CNN performs with the small collected dataset. In addition we wanted to explore how to gain better performance using the small set of data. In the work reported in this paper, we have conducted multiple experiments, in particular, three different approaches were explored. The first approach was to extract the MFCC features from the audio signals and used them to train the CNN model, whereas in the second approach just raw audio signals were used as the input to the model. The third approach attempted to leverage features learned from English speech using transfer learning. Among all models, the MFCC model had given slightly better test performance since it showed better percentage accuracy but suffered from the over-fitting issue. All the models had difficulties in recognizing multi-syllable words. For future work, we would like to increase the number of samples per word in the dataset and also expand the number of words.

References

[1] B. H. Juangand and L. R. Rabiner. "Automatic Speech Recognition - A Brief History of the Technology Development". Jan 2005.

[2] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. "Convolutional Neural Networks for Speech Recognition". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014.

[3] Y. Qian, M. Bi, T. Tan, and K. Yu. "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2263–2276, Dec 2016.

[4] L. Deng and J. Platt. "Ensemble Deep Learning for Speech Recognition". Proc. Interspeech, Sep 2014.

[5] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke. "Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition". In *Proceedings of Interspeech 2012*, 2012.

[6] A. K. Paul, D. Das, and M. M. Kamal. "Bangla Speech Recognition System Using LPC and ANN". In *2009 Seventh International Conference on Advances in Pattern Recognition*, pages 171–174, Feb 2009.

[7] G. Muhammad, Y. A. Alotaibi, and M. N. Huda. "Automatic speech recognition for Bangla digits". In *2009 12th International Conference on Computers and Information Technology*, pages 379–383, Dec 2009.

[8] M. A. Hossain, M. M. Rahman, U. K. Prodhan, and M. F. Khan. "Implementation of Back-Propagation Neural Network for Isolated Bangla Speech Recognition". *CoRR*, 2013.

[9] M. A. Ali, M. Hossain, and M. N. Bhuiyan. "Automatic Speech Recognition Technique for Bangla Words". 2013.

[10] K. J. Piczak. "Environmental sound classification with Convolutional Neural Networks". In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep 2015.

[11] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks". In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Apr 2011.