Amazigh Spoken Digit Recognition using a Deep Learning Approach based on MFCC

Original Scientific Paper

Hossam Boulal
Sidi Mohamed Ben Abdellah University of Fez
Multidisciplinary faculty of Taza, LSI Laboratory
Taza, Morocco.
hossam.boulal@usmba.ac.ma

Mohamed Hamidi
Mohamed First University of Oujda,
Multidisciplinary faculty of Nador, Team of modeling and scientifc computing
Nador, Morocco.
m.hamidi@ump.ac.ma

Mustapha Abarkan
Sidi Mohamed Ben Abdellah University of Fez
Multidisciplinary faculty of Taza, LSI Laboratory
Taza, Morocco.
mustapha.abarkan@usmba.ac.ma

Jamal Barkani
Sidi Mohamed Ben Abdellah University of Fez
Multidisciplinary faculty of Taza, LSI Laboratory
Taza, Morocco.
jamal.barkani@usmba.ac.ma

Abstract *– The field of speech recognition has made human-machine voice interaction more convenient. Recognizing spoken digits is particularly useful for communication that involves numbers, such as providing a registration code, cellphone number, score, or account number. This article discusses our experience with Amazigh's Automatic Speech Recognition (ASR) using a deep learning-based approach. Our method involves using a convolutional neural network (CNN) with Mel-Frequency Cepstral Coefficients (MFCC) to analyze audio samples and generate spectrograms. We gathered a database of numerals from zero to nine spoken by 42 native Amazigh speakers, consisting of men and women between the ages of 20 and 40, to recognize Amazigh numerals. Our experimental results demonstrate that spoken digits in Amazigh can be recognized with an accuracy of 91.75%, 93% precision, and 92% recall. The preliminary outcomes we have achieved show great satisfaction when compared to the size of the training database. This motivates us to further enhance the system's performance in order to attain a higher rate of recognition. Our findings align with those reported in the existing literature.*

Keywords*: CNN, ASR, Deep learning, digit, Amazigh, speech*

1. INTRODUCTION

Amazigh is a North African language that is spoken by 50% of Moroccans. There are 4 vowels, 27 consonants, 2 semi-consonants, and 33 graphemes in the Amazigh language [1], This refers to the Tifinagh letters used in Morocco. Automatic Speech Recognition (ASR) is a challenging task in the Amazigh language due to its morphological complexity, language barrier, dialects, and resource limitations. ASR now has a wide range of applications that are both diverse and challenging, especially when combined with deep learning, a branch of machine learning [2]. It incorporates several techniques and features that mimic human thought and behavior. Deep learning algorithms modify the input values through a hierarchy of nonlinear transformations, and as an output, they create statistical models that are capable of independently anticipating future events [3]. Deep learning techniques require more processing power and a large amount of training data [2]. Artificial neural networks (ANN) [4], convolutional neural networks (CNN) [5], and recurrent neural networks (RNN) [6] are only a few examples of the numerous types of neural networks. The most popular deep neural network for understanding and interpreting visual data is a convolutional neural network [2].

In this work, we present the creation and evaluation of an Amazigh speech recognition system based on the MFCC and CNN approach. Our study focuses on harnessing the potential of these techniques to develop an accurate and efficient system for recognizing Amazigh speech. The focal point of this paper is our CNN model, which is designed to process input images of MFCC features that are extracted from audio signals. Our approach to addressing the complexity of audio classification involves reframing it as an image classification problem, using plots of the MFCC features as inputs to the CNN model. This innovative strategy is a key contribution to our work.

The rest of the paper is organized as follows: Sect. 2 presents the related works. Section 3 presents our methodology and system preparation. Section 4 shows the system's results and performance. Finally, Sect. 5 concludes the paper.

2. LITERATURE REVIEW

Several studies have recently examined Moroccan Amazigh speech recognition systems, with a particular emphasis on the Interactive Voice Response (IVR) system [7]. This system was evaluated by the authors for its ability to identify speaker-independent Amazigh digits in loud environments at different decibel levels. The researchers discovered that digits containing the consonant "S" were the most affected.

An automated voice recognition system for Amazigh has been developed by Telmem and Ghanou [8]. The authors trained and assessed the system for different Gaussian values ranging from 1 to 256 by using HMMs with 3 and 5 states. After training with 128 Gaussian mixture models and 5 HMM state numbers, the system performed best. They achieved a system rate of 90%.

Ghazi and Dawi [9] utilized Amazigh numerical construction principles in developing ASR. Their strategy involved generating associated numbers from an unrelated single number ranging from 1 to 10. This method enabled the expansion of the Amazigh accent in ASR. The researchers concluded that the results were satisfactory, taking into account the size of the learning database.

Satori and El Haoussi [10] have evaluated the independence of the speech recognition system through the use of "Alphadigits", a speech database containing Amazigh numbers and letters pronounced by native speakers. They built the system using CMU Sphinx, which is an HMM-based approach. The best results were achieved when they trained the system with 16 GMMs and 5 HMMs, with an accuracy rate of 92.89%.

Ghazi et al. [11] developed an Amazigh ASR based on the transcription of the Tifinaghe alphabet, which was authorized by the Royal Institute of Culture Amazigh (IRCAM). They compared the efficacy of dynamic programming and HMM approaches. Despite the limited number of speakers and the size of the dataset, the HMM method outperformed dynamic programming. This highlights the effectiveness of probabilistic techniques and stochastic models in speech recognition.

Zealouk O. et al. [12] conducted an assessment of the Sphinx toolset for open-source voice recognition. They specifically compared the use of Pocketsphinx and Sphinx-4 decoders, examining the tradeoff between identification accuracy and computational cost. Notably, the evaluation was carried out in the Amazigh language. Their system was trained to detect the first 10 Amazigh digits using 5 Hidden Markov model states (HMMs) with 16 Gaussian mixture models (GMMs) and the Mel frequency spectral coefficients (MFCCs). According to the gathered testing findings, the Pocketsphinx toolbox recorded the highest recognition rate.

Hamidi M. et al. [13] developed a safe VoIP network-based interactive Amazigh speech system. They examine interactive voice response (IVR) solutions for managing backup and firewall responsibilities in their research. Based on HMM automatic speech recognition, a voice platform was created. To increase security, the biological voiceprint is employed for identification and management tasks. The Amazigh language's 10 starting numerals, 33 alphabets, and five words have all been taught to the voice recognition system. They experimented with the network's remote voice control administration system by altering the IVR and ASR system parameters. According to their study, the system works best for admin tests when trained with three HMMs and eight GMMs. The non-admin recognition rate, however, is less than 5%, demonstrating the system's security.

Lounnas K. et al. [14] reported a series of research using a hybrid methodology to improve spoken digit recognition for under-resourced languages of the Maghreb region. The value of including a dialect identification module in an Automatic Speech Recognition (ASR) system has been shown in earlier research. To enable the ASR system to recognize numbers spoken in many languages, they trained their hybrid system on the Moroccan Arabic Dialect (MAD), Algerian Arabic Dialect (AAD), and Moroccan Amazigh Dialect (MAD) in addition to Modern Standard Arabic. They investigated two deep learning models and five machine learning classifiers, the second of which uses two pre-trained models: residual deep neural networks (RDN), and the first of which is based on convolutional neural networks (CNN) (Resnet50 and Resnet101). The CNN model outperforms the other recommended methods, increasing the spoken digit recognition system's performance by 20% for both Moroccan and Algerian languages, according to the data.

3. METHODOLOGY

There are six essential steps in the process for CNN-based digit identification in the Amazigh language, which is depicted in Fig. 1.

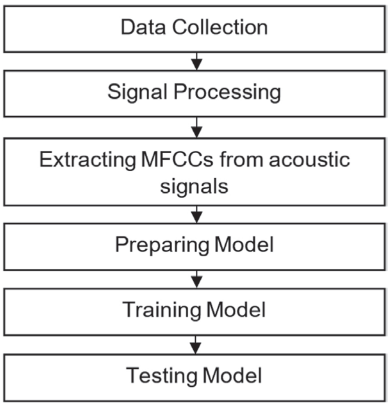


Fig. 1. Steps for proposed model

3.1. DATA COLLECTION

42 native Amazigh speakers were selected for this study to take part in the data collection. All of the volunteers that spoke were between the ages of 20 and 40. The audio database was recorded using the Audacity software. The numbers from 0 to 9 were spoken ten times each by each speaker. Table 1 displays the 0 to 9 digits of the dataset in Amazigh with English transcription. 4200 occurrences (42 speakers * 10 digits * 10 occurrences) were used for the proposed speech recognition model. We have chosen to divide the data into three categories: 60% for training, 20% for validation, and 20% for testing to assure the speaker-independent component.

Table 1. Dataset of digits (0 to 9) in Amazigh and English representation.

Amazigh digits	English transcription	Tifinagh transcription	Number of syllables	Syllables
ILEM	ZERO	ⵢⵍⴻⵎ	2	VCVC
YAN	ONE	ⵢⴰⵏ	1	CVC
SIN	TWO	ⵙⵉⵏ	1	CVC
KṚAḌ	THREE	ⴽⵕⴰⴹ	1	CCVC
KKUẒ	FOUR	ⴽⴽⵓⵥ	1	CCVC
SEMMUS	FIVE	ⵙⵎⵎⵓⵙ	2	CVCCVC
ṢḌIṢ	SIX	ⵙⴹⵉⵚ	1	CCVC
SA	SEVEN	ⵙⴰ	1	CV
TAM	EIGHT	ⵜⴰⵎ	1	CVC
TẒA	NINE	ⵜⵥⴰ	1	CCV

3.2. SIGNAL PROCESSING

Each speaker's recordings were saved as a ".wav" file. To ensure that all of the signal's digits were recorded, the right phrase was kept in the database, and the wrong ones were repeated until reliable and secure recordings were created, each utterance was repeated throughout the recording session.

Speakers were told to record all digits in one voice recording file (with 10 repetitions) to make the work of recording easier. The output files were cleaned up and reduced to a single-word signal (one repetition of each word in each audio file).

3.3. EXTRACTING MFCCS

We approach the challenge of categorizing audio as an image classification task in our method. To be employed with CNN models, audio recording data must be represented in the visual domain, that is the reason we opted to utilize MFCC (Mel Frequency Cepstral Coefficients [15-17].

Mel-Frequency Cepstral Coefficients (MFCC) is a widely used technique in digital signal processing and speech recognition. MFCC allows for the extraction of audio signal features that can be utilized in various signal processing applications and speech recognition tasks. The core concept behind MFCC is based on the observation that the human auditory system is attuned to logarithmically-spaced frequency bands, rather than linearly-spaced ones. To better simulate this experience, MFCC maps the audio spectrum onto the mel-scale, a logarithmic scale that more closely resembles how humans perceive sound. The first step in the MFCC extraction process is to preprocess the audio signal by removing noise and artifacts using methods like windowing and filtering. The signal is then split into brief frames that typically last 20 to 30 milliseconds. For each frame, a power spectrum is computed using a Fast Fourier Transform. A filter bank made up of triangle filters that overlap and are evenly placed on the Mel scale is then used to translate the power spectrum into the Mel scale. The discrete cosine transform (DCT) is then applied to the resulting mel-scaled spectrum to produce the MFCCs, a collection of coefficients that describe the signal's spectral envelope. Typically, just the first 10 to 20 MFCC coefficients are employed for speech recognition tasks since these are the ones that contain the majority of the data necessary to distinguish between various phonemes and words.

In our case, we took 13 Coefficients, plus Delta 1 and Delta 2, a total of 39 in each frame of 31.25 ms. Table 2 displays the parameters that were used to generate the MFCC.

Table 2. Librosa parameter values for MFCC generation

Parameters	Values
Sample rate	16000
Window length	512
hop length	128
Overlap	384
Window function	Hann
FFT Length	512

Below is an illustration of the grayscale MFCC spectrogram, generated using the librosa.feature.mfcc function from the librosa Python library for audio analysis [10].

Fig. 2. MFCC Spectrogram of a Digit Spoken in Amazigh

When we usually see an MFCC spectrogram (Fig. 2), we see it in color, the role of these colors is to facilitate the reading and understanding for the viewer, and it is not necessary, the spectral information can be represented using only black and white. The use of grayscale will be of great benefit at the level of computational cost, as using color, the input matrix dimension will be (256×256×3), but using Grayscale representation, the input dimension will be only (256×256×1).

3.4 PREPARING MODEL

The dataset consists of 4200 instances from both male and female speakers. It consists of 10 recordings of the numbers 0 through 9 being uttered. The dataset was recorded as 16000 Hz monophonic 16-bit audio files in .wav format using the Audacity program. Google Colab is used to construct and execute neural networks and acquire GPU and TPU as a runtime environment. The usage of it is free. The Librosa Python Library is used to extract spectrograms from audio files.

Training, validation, and testing data are split up into three groups with the ratios 60:20:20. The dataset is divided based on speakers, using 25 speakers for training, 8 speakers for validation, and 9 speakers for testing in order to have all the audio samples for the same speaker in only one split (training, or validation, or testing). It was carried out utilizing the train_test_split function of the Python "sklearn" module (see Fig. 3).

We preserved the same number of speakers across all datasets while reducing the number of repetitions for each number; for example, in the case of 100%, there are 10 iterations, 42 speakers, and 4200 total samples. While in the case of 70%, there is the same number of speakers (42), but there are only 7 repeats of each number, therefore there will be 2940 samples in total. Thus, if the percentage is 40%, there will be 1680 samples in total.

The Keras interface and TensorFlow library are used to implement our model in Python.

There are ten classes, numbered from 0 to 9, for each spectrogram picture. The following step is to separate the data into training, validation, and testing sets. Using sequential types, we built a layer-by-layer model in Keras.

The first convolutional layer (C1), which has 96 kernels with symmetrical shapes of sizes (11×11) and stride settings of (2×2) pixels, receives input pictures from the spectrogram during the training phase. A second convolution layer (C2) with 256 kernels of (5×5) is added after a maximum pooling layer (P1) of size (3×3) and a stride of (2×2). The second Maximum Pooling layer (P2) is identical to the first (P1). The last maximum pooling layer (P3) has a size of (2×2) and a final convolutional layer (C3) with 384 kernels of (3×3). Between the convolutional layers and the maximum pooling layers, batch normalization is implemented.

Fully connected layers (FC) of sizes 1024 (FC1), 1024 (FC2), and 10 (FC3) are positioned after the feature learning phase. Except for the last dense layer (FC3), which uses a Softmax activation function, all convolutional layers and dense layers employ the activation function ReLu (Rectified Linear Activation). A dropout layer with a value of 0.2 dropout rate is put after the Fully Connected layers (FC1) and (FC2). Fig. 4 depicts our CNN architecture.

Once all of the model's parameters have been specified, three parameters are utilized during compilation. Loss, metrics, and optimizers are all involved in the compilation process. SGD (Stochastic Gradient Descent) is incorporated from Keras with a momentum of 0.9 since it is an iterative method for maximizing the objective function. Sparse categorical cross-entropy is the name of the loss function that is used to estimate the performance of the model. The third parameter, the accuracy metric, is used to show the model's accuracy on the validation data.

3.5 TRAINING MODEL

A model is trained using training data, validation as test data, and the fit () function. The total number of data model cycles is determined by epochs. Up to a point, better model performance can be achieved by adding more epochs. The recommended model was used to analyze the neural network's performance across 250 epochs.

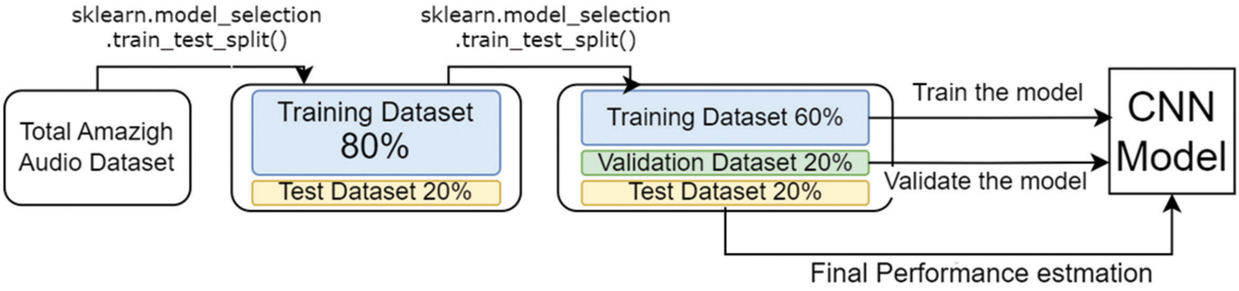


Fig. 3. Training, validation and test division of the dataset

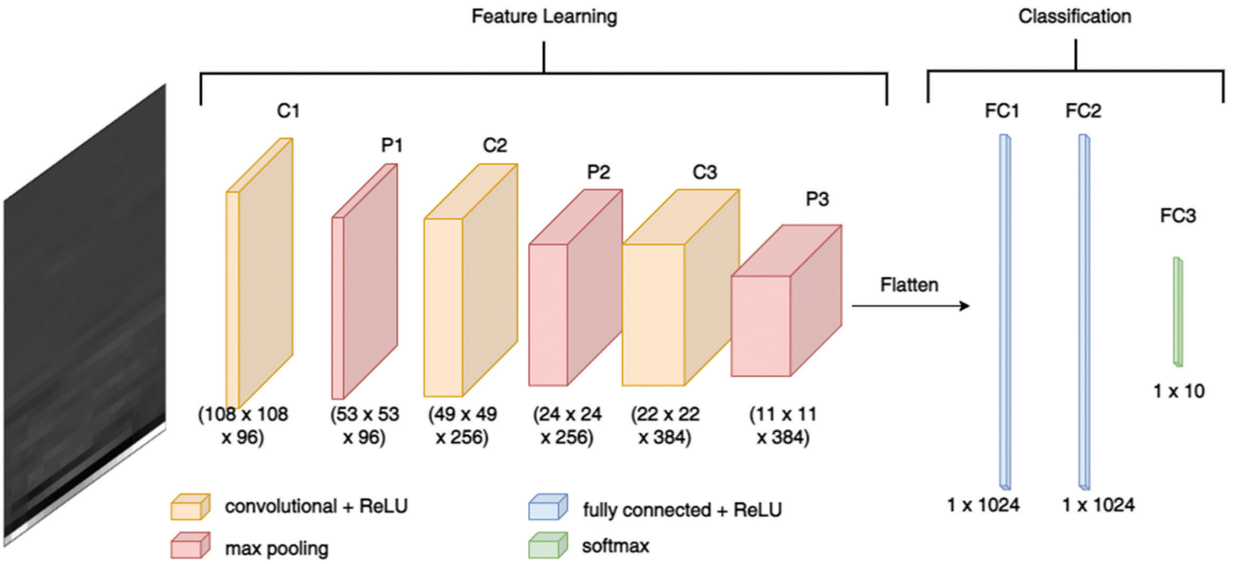


Fig. 4. Convolutional Neural Network

To determine test accuracy, one uses the model evaluation function. Sklearn's classification report and confusion matrix were used. To get information on the accuracy, recall, and f1-score of each test digit, the classification report was invoked.

4. RESULT AND DISCUSSION

The model is run three times with datasets of 1680, 2940, and 4200 digits, respectively. The outcomes of the outcome analysis utilizing several data sets are displayed in Table 3. The 4200 dataset values yielded the greatest results; the accuracy after the first iteration was 88.13%, with 89% precision and 89% recall. With 2940 samples in the second iteration, accuracy was 91.25%, precision was 91%, and recall was 91%. The third iteration produces 91.75% accuracy, 93% precision, and 92% recall with 4200 occurrences.

Table 3. Result analysis with various dataset sizes.

Experiment No.	Data set size	No. of Epochs	Batch size	Accuracy %	Precision %	Recall %
1	1680	250	32	88.13	89	88
2	2940	250	32	91.25	91	91
3	4200	250	32	91.75	93	92

The analysis demonstrates that when dataset sizes grow, recognition accuracy grows as well.

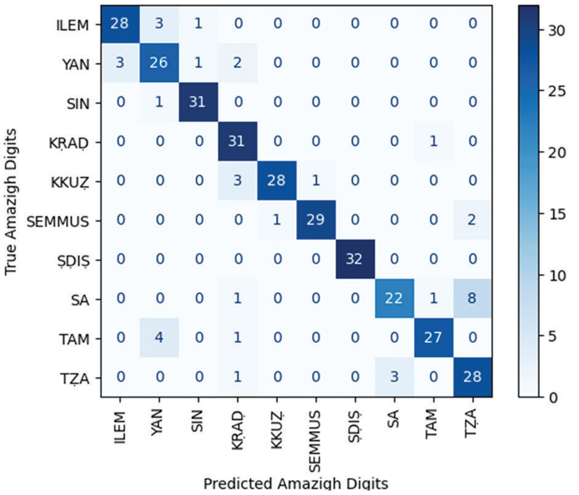
For a Dataset of 1680, 320 samples in testing (32 samples per digit):

Table 4. Classification report 40% dataset.

	precision	recall	f1-score
ILEM	90	88	89
YAN	76	81	79
SIN	94	97	95
KRAD	79	97	87
KKUZ	97	88	92
SEMMUS	97	91	94
ṢDIṢ	100	100	100
SA	88	69	77
TAM	93	84	89
TẒA	74	88	80

In the first experiment, the number "SDIS" had the greatest recall (100%), followed by "SIN", and "KRAD" (97% each of them). In addition, the number "SA" gave the worst recall (69%) (see Table 4).

As for Precision, the number "SDIS" had a perfect classification (100%), "KKUZ" and "SEMMUS" got second place with 97%, while numbers "TZA", "YAN" and "KRAD" got the worst (74%, 76%, and 79%). In this experiment, a significant difference was observed between precision and Recall, concerning the digit "SA" (19%) and digit "KRAD" (18%) (see Fig. 5).



	ILEM	YAN	SIN	KRAD	KKUẒ	SEMMUS	ṢDIṢ	SA	TAM	TẒA
ILEM	28	3	1	0	0	0	0	0	0	0
YAN	3	26	1	2	0	0	0	0	0	0
SIN	0	1	31	0	0	0	0	0	0	0
KRAD	0	0	0	31	0	0	0	0	1	0
KKUẒ	0	0	0	3	28	1	0	0	0	0
SEMMUS	0	0	0	0	1	29	0	0	0	2
ṢDIṢ	0	0	0	0	0	0	32	0	0	0
SA	0	0	0	1	0	0	0	22	1	8
TAM	0	4	0	1	0	0	0	0	27	0
TẒA	0	0	0	1	0	0	0	3	0	28

Fig. 5. The Confusion matrix of our CNN model with the 40% of the dataset.

For a Dataset of 2940, 560 samples in testing (56 samples per digit):

Table 5. Classification report 70% dataset.

	precision	recall	f1-score
ILEM	94	80	87
YAN	79	88	83
SIN	88	100	93
KRAD	97	100	98
KKUZ	100	100	100
SEMMUS	100	96	98
ṢḌIṢ	97	100	98
SA	84	77	80
TAM	96	88	92
TẒA	81	84	82

In the second experiment, and with the increase of data to 70%, a change occurred in the classification. "SDIS" has no longer the only perfect recall, "KKUZ", "KRAD", and "SIN" had also 100% of recall, the worst recall is still the commentator with numbers "ILEM" and "SA" (80% and 77%). Looking at the precision, the number "KKUZ" and "SEMMUS" have a perfect precision (100%), and the number "YAN" got the worst precision (79%) although his precision improved compared to the first experience. We noticed here that the biggest difference between Precision and Recall was with the number "ILEM" with a difference of (14%) (see Table 5 and Fig. 6).

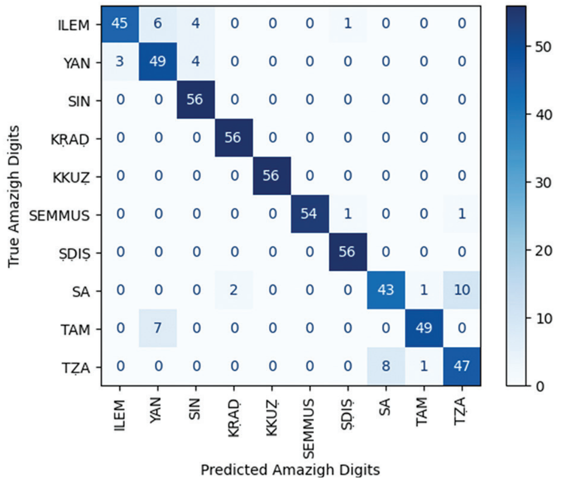


Fig. 6. The Confusion matrix of our CNN model with the 70% of the dataset

For a Dataset of 4200, 800 samples in testing (80 samples per digit):

Table 6. Classification report 100% dataset

	precision	recall	f1-score
ILEM	98	74	84
YAN	72	89	80
SIN	87	99	92
KRAD	99	100	99
KKUZ	98	100	99
SEMMUS	100	97	99
ṢḌIṢ	100	100	100
SA	87	84	85
TAM	100	88	93
TẒA	84	88	86

In the third experiment, what happened in the second experiment was confirmed, "SDIS", "KRAD", and "KKUZ" have perfect recall classification (100%), followed by "SIN" (99%), these top four, the same were the first in the previous experiment. The worst recalls are still like the previous experiment with an exchange in order, "SA" with 84%, and "ILEM" with 74%. Also in terms of Precision, the last five orders remained the same as in the previous experiment, while "SDIS", "SEMMUS" and "TAM" got perfect precision (100%) (see Table 6 and Fig. 7).

The numbers "SA" and "TZA" were frequently mentioned when discussing the worst classifications. The confusion matrix in Tables 4, 5, and 6 explains why "SA" and "TZA" received low ratings. We observe that our system often confuses the two digits "SA" and "TZA". We speculate that this may be due to their similar pronunciation.

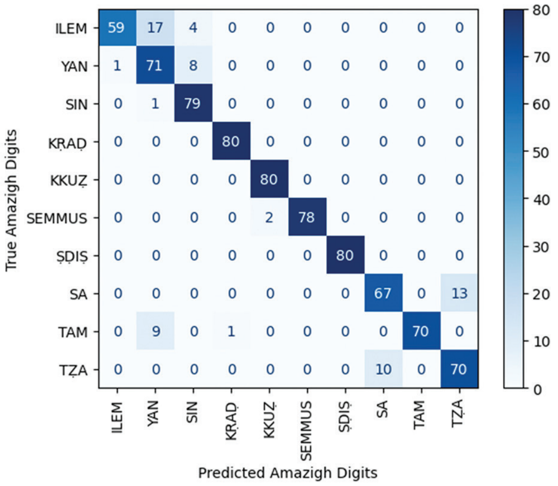


Fig. 7. The Confusion matrix of our CNN model with the 100% of the dataset

Our realized study has been contrasted with other studies that sought to identify speech in the Amazigh language. Table 7 presents a range of results obtained from prior studies. The Amazigh voice recognition system was created by Telmem, M., and Ghanou, Y [18]. It was built on MFCC and CNN, and it was able to recognize 33 letters of the Amazigh language with an accuracy of up to 93.90% from 4620 samples (80:10:10). Satori and El Haoussi (2014) [11] made use of a voice database that included native Amazigh speakers' utterances of Amazigh numerals and letters. This system was built using a system based on HMMs and CMU Sphinx, with feature extraction performed using MFCC. The greatest performance, with a score of 92.89%, came from the recognition results employing 16 Gaussian mixed models and five fixed HMM states. With the use of Pocketsphinx, Zealouk O. et al. [13] developed a system consisting of 5 HMMs and 16 GMMs. They also took advantage of the MFCC's feature extraction capabilities, which resulted in a top classification rate of 92.14% for Amazigh digits. Using a Raspberry Pi, Barkani, F. et al. [19] research Amazigh speech recognition and develop a system based on MFCCs, 3 HMMs, and 16 GMMs. Words in Amazigh were given to the algorithm, and it performed at its best (90.43%).

Table 7. Results of different approaches

Ref	year	Approach	Best parameters	Accuracy
Our Work	2023	MFCC + CNN	----------	91.75%
[13]	2022	MFCC + HMM + GMM	5 HMMs + 16 GMMs	92.14%
[18]	2021	MFCC + CNN	----------	93.90%
[19]	2020	MFCC + HMM + GMM	3 HMMs + 16 GMMs	90.43%
[11]	2014	MFCC + HMM + GMM	5 HMMs + 16 GMMs	92.89%

5. CONCLUSION

In this study, the first 10 Moroccan Amazigh digits were used to assess the ASR speaker-independent system. Our system, which uses MFCC Spectrograms as the basis for feature extraction, was built using the convolutional neural network. As part of this endeavor, the first 10 Amazigh language digits were developed as a speech database called "Amazigh digits," which was used in the system's training and testing stages. The findings we have provided show that our Amazigh ASR system is independent of the speaker and that it agrees with the findings of [13][18][19][11]. Our obtained results show that the best-achieved accuracy is 91.75%.

ACKNOWLEDGEMENT

We want to thank two specialists who have spared no effort in any assistance related to the Amazigh language. We would like to thank Dr. Younas LOUKILI and Mr. Youssef Kouraich, Both teachers of the Amazigh language, and employees of the Ministry of National Education in Morocco.

6. REFERENCES

[1] A. Boukous, "The planning of Standardizing Amazigh language The Moroccan Experience", Iles d imesli, Vol. 6, No. 1, 2014, pp. 7-23.

[2] A. A. Abdullah, M. M. Hassan, Y. T. Mustafa, "A review on bayesian deep learning in healthcare: Applications and challenges", IEEE Access, Vol. 10, 2022, pp. 36538-36562.

[3] J. H. Tailor, R. Rakholia, J. R. Saini, K. Kotecha, "Deep Learning Approach for Spoken Digit Recognition in Gujarati Language", International Journal of Advanced Computer Science and Applications, Vol. 13, No. 4, 2022.

[4] S. Bilgaiyan, S. Mishra, M. Das, "Effort estimation in agile software development using experimental validation of neural network models", International Journal of Information Technology, Vol. 11, No. 3, 2019, pp. 569-573.

[5] V. Jain, A. Jain, A. Chauhan, S. S. Kotla, A. Gautam, "American sign language recognition using support vector machine and convolutional neural network", International Journal of Information Technology, Vol. 13, 2021, pp. 1193-1200.

[6] D. Chhachhiya, A. Sharma, M. Gupta, "Designing optimal architecture of recurrent neural network (LSTM) with particle swarm optimization technique specifically for educational dataset", International Journal of Information Technology, Vol. 11, 2019, pp. 159-163.

[7] M. Hamidi, H. Satori, O. Zealouk, K. Satori, "Amazigh digits through interactive speech recognition system in noisy environment", International Journal of Speech Technology, Vol. 23, No. 1, 2020, pp. 101-109.

[8] M. Telmem, Y. Ghanou, "Estimation of the optimal HMM parameters for amazigh speech recognition system using CMU-Sphinx", Procedia Computer Science, Vol. 127, 2018, pp. 92-101.

[9] A. El Ghazi, C. Daoui, N. Idrissi, "Automatic Speech Recognition for tamazight enchained digits", World Journal Control Science and Engineering, Vol. 2, No. 1, 2014, pp. 1-5.

[10] H. Satori, F. ElHaoussi, "Investigation Amazigh speech recognition using CMU tools", International Journal of Speech Technology, Vol. 17, 2014, pp. 235-243.

[11] A. El Ghazi, C. Daoui, N. Idrissi, M. Fakir, B. Bouikhalene, "Système de reconnaissance automatique de la parole Amazigh à base de la transcription en alphabet Tifinagh", Revue Méditerranéenne des Télécommunications, Vol. 1, No. 2, 2011.

[12] Z. Ouissam, H. Mohamed, S. Hassan, "Investigation on speech recognition Accuracy via Sphinx toolkits", Proceedings of the 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, Meknes, Morocco, 3-4 March 2022, pp. 1-6.

[13] M. Hamidi, H. Satori, O. Zealouk, K. Satori, N. Laaidi, “Interactive voice response server voice network administration using hidden markov model speech recognition system”, Proceedings of the Second World Conference on Smart Trends in Systems, Security and Sustainability, London, UK, 30-31 October 2018, pp. 16-21.

[14] K. Lounnas, M. Abbas, M. Lichouri, M. Hamidi, H. Satori, H. Teffahi, “Enhancement of spoken digits recognition for under-resourced languages: case of Algerian and Moroccan dialects”, International Journal of Speech Technology, Vol. 25, No. 2, 2022, pp. 443-455.

[15] Q. Zhou et al. “Cough recognition based on mel-spectrogram and convolutional neural network”, Frontiers in Robotics and AI, Vol. 8, 2021, p. 580080.

[16] A. Ahmed, Y. Serrestou, K. Raoof, J.-F. Diouris, “Empirical Mode Decomposition-Based Feature Extraction for Environmental Sound Classification”, Sensors, Vol. 22, No. 20, 2022, p. 7717.

[17] K. W. Gunawan, A. A. Hidayat, T. W. Cenggoro, B. Pardamean, “A transfer learning strategy for owl sound classification by using image classification model with audio spectrogram”, International Journal on Electrical Engineering and Informatics, Vol. 13, No. 3, 2021, pp. 546-553.

[18] M. Telmem, Y. Ghanou, “The convolutional neural networks for Amazigh speech recognition system”, TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol. 19, No. 2, 2021, pp. 515-522.

[19] F. Barkani, H. Satori, M. Hamidi, O. Zealouk, N. Laaidi, “Amazigh speech recognition embedded system”, Proceedings of the 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, Meknes, Morocco, 16-19 April 2020, pp.