**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 22, No. 2, May 2021, pp. 1096∼1107
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22i2.pp1096-1107 

# ArSL-CNN: A convolutional neural network for arabic sign language gesture recognition

**Ali A. Alani**[1], **Georgina Cosma**[2]
[1]Department of Computer Science, University of Diyala, Diyala, Iraq
[2]Department of Computer Science, School of Science, Loughborough University, U.K

**Article Info**

***Article history:***

Received Jan 20, 2021
Revised March 13, 2021
Accepted March 20, 2021

***Keywords:***

Arabic sign language
CNNs
Convolutional neural networks
Deep learning
SMOTE

**ABSTRACT**

Sign language (SL) is a visual language means of communication for people with deafness or hearing impairments. In Arabic-speaking countries, there are many arabic sign languages (ArSL) and these use the same alphabets. This study proposes ArSL-CNN, a deep learning model that is based on a convolutional neural network (CNN) for translating Arabic SL (ArSL). Experiments were performed using a large ArSL dataset (ArSL2018) that contains 54,049 images of 32 sign language gestures, collected from forty participants. The results of the first experiments with the ArSL-CNN model returned a train and test accuracy of 98.80% and 96.59%, respectively. The results also revealed the impact of imbalanced data on model accuracy. For the second set of experiments, various re-sampling methods were applied to the dataset. Results revealed that applying the synthetic minority oversampling technique (SMOTE) improved the overall test accuracy from 96.59% to 97.29%, yielding a statistically significant improvement in test accuracy (p=0.016, $\alpha < 0.05$). The proposed ArSL-CNN model can be trained on a variety of Arabic sign languages and reduce the communication barriers encountered by deaf communities in Arabic-speaking countries.



***Corresponding Author:***

Ali A. Alani
Department of Computer Science
University of Diyala
Diyala, Iraq
Email: alialani@uodiyala.edu.iq

## 1. INTRODUCTION

Sign language (SL) is visual means of communication for people who have deafness or hearing-impairments, using gestures, facial expression, and body language [1], [2]. In 2019, the world health organization reported that approximately 466 million people, which is approximately 5% of the world's population, suffer from hearing impairment. Among these people, roughly 34 million are under the age of 18. A previous study predicted that this number would double by 2050 due to genetic factors, birth complications, infectious diseases, and chronic ear infections [3], [4]. Studies have been conducted to develop systems that can recognise the signs of various SLs [5]. Arabic SL (ArSL) recognition systems are currently in the development phase [6], and there exist limited SL recognition systems that can identify ArSL signs using deep learning methods. Two methods can be applied to SL recognition systems, namely, sensor and image-based methods [7]. Sensor-based methods require the user to wear instrumental gloves with sensors to recognise hand gestures. This approach requires interfacing multiple sensors with a glove to collect the gestures using sensor data, that is analysed for gesture recognition and translation tasks. Despite their accuracy and reliability, sensor-based methods have

several limitations, such as discomfort in using gloves overloaded with wires, sensors and other materials worn by the signer [8], [9]. By contrast, with image-based sign language gesture recognition, the signers are not required to use any kind of gloves or complicated devices. This technique provides users with more versatility than the sensor-based systems. However, intensive computations are necessary in the preprocessing phase to recognise the signs. Recent studies focus on the performance of image-based approaches in recognising ArSL [8], [9]. Image-based systems for recognising human signs are complex and multidisciplinary. These systems are developed using various machine learning (ML) methods, such as the artificial neural network [10], support vector machines (SVMs) [11], and elastic graph matching [12]. Deep learning (DL) algorithms have recently boosted many research fields, including image recognition and classification. DL algorithms are ML methods that have been utilised in various applications, such as medical image classification [13] and object recognition [14]. DL models comprise a neural network with more than one hidden layer that uses various levels of distribution to represent and learn the high-level abstractions of data. The objective of this study is to advance the research of ArSL and explore the capabilities of deep learning methods, specifically the convolutional neural network (CNN) method, for classifying ArSL gestures. This paper proposes a new deep learning model, ArSL-CNN, and explores the advantage of resampling techniques to address class imbalance in the dataset. The proposed ArSL-CNN is trained with images of hand signs in different lighting conditions and orientations to automatically recognise 32 ArSL signs [15]. The remainder of this paper is organised as follows. Related work is reviewed in Section 2. The design and architecture of the proposed ArSL-CNN model are presented in Section 3. Experiments and results are discussed in Section 4. A comparison of the proposed ArSL-CNN with state-of-the-art methods is discussed in Section 5. A conclusion and future research directions are provided in Section 6.

## 2. RELATED WORKS

Numerous methods are applied for SL recognition tasks. The two major approaches are the handcrafted feature engineering extraction and DL methods. The earliest known work on SL recognition is focused on the extraction of hand-engineered features, which are fed to learning algorithms for classification [16]. Consequently, the efficiency of these algorithms is highly dependent on handcrafted feature engineering [17]. Therefore, the accuracy results obtained using these approaches highly depend on extracted features. Ibrahim et al. [18] constructed a dataset containing 30 isolated words from children with hearing disabilities. The geometric features of the hands were formulated into feature vectors that were used for classification and automatic translation of the individual Arabic signs into text words. The accuracy of their proposed system reached 97%. Alzohairi et al. [9] applied the histogram of oriented gradients (HoG) feature descriptor for extracting features from ArSL image data, and then adopted the SVM algorithm for developing a ArSL image recognition system. The accuracy of their system reached 63.5%. Abdo et al. [19] applied the hidden Markov model and hand geometry with different hand shapes and forms to the task of Arabic alphabet and numbers sign language recognition and translation into speech or text. With Deep Learning, the features are extracted hierarchically in an automated manner by applying a series of transformations to the input images. The extracted features are the most robust ones, which means that complex problems are effectively modelled using DL architectures. Nagi et al. [20] proposed a hand gesture recognition system by using a CNN and used morphological image processing and colour segmentation to obtain hand contour edges and eliminate noise. Their proposed model achieved an accuracy of 96% on 6,000 sign images obtained from six gestures. By using the data collected by a Kinect sensor, Tang et al. [21] used a deep belief network (DBN) and a CNN for sign language recognition. Authors trained the DBN and CNN model using 36 different hand postures. The DBN model achieved 98.12% accuracy, which was higher than the accuracy obtained by the CNN model. Yang and Zhu [2] introduced a CNN system for the recognition of Chinese SL. The authors obtained video-based data by using 40 regular vocabularies. In the preprocessing stage, the authors enhanced the hand segmentation process and prevented the loss of important information during feature extraction. Moreover, they compared two different optimizers, namely, Adagrad and Adadelta, and their results revealed that the CNN model reached better accuracy when the Adadelta optimizer was used. Oyedotun and Khashman [5] adopted two DL methods, namely, CNN and stacked denoising autoencoder (SDAE) networks, to recognise 24 ASL alphabets. The samples were collected from the freely accessible Thomas Moeslund's gesture recognition database. Their test results showed that SDAE outperformed the CNN model in terms of overall average accuracy (92.83%). Eibadawy et al. [22] proposed a CNN-based framework for ArSL recognition to identify 25 signs. The accuracy values of this

model on the training and unseen data were 85% and 98%, respectively. Ghazanfar, et al. [1] proposed different CNN architectures using 54,049 sign images of more than 40 participants provided by [15]. Their results revealed the significant effect of the dataset size on the accuracy of the proposed model. By increasing the size of the dataset from 8,302 samples to 27,985 samples, the proposed model test accuracy increased from 80.3% to 93.9%. Also, increasing the size of the dataset from 33406 samples to 50000 samples resulted in a further increase in the proposed model test accuracy from 94.1% to 95.9%, respectively. Elsayed and Fathy [3] examined the capacity of ontology technologies (semantic web technologies) and DL to design a multiple sign language ontology for feature extraction using CNNs for the ArSL recognition task. Their findings revealed that the recognition rates of the ArSL training and testing sets were 98.06% and 88.87%, respectively. Although CNNs perform well with computer vision tasks, they require massive quantities of data to train the network. This disadvantage demands an enormous amount of time and computing capabilities. Several researchers use transfer learning techniques to minimise the processing time and the number of dataset samples needed to train the CNN model. Saleh and Issa [23] used transfer learning on a pre-trained network of VGG-16 and Resnet152 to boost performance in identifying 32 hand gestures from the ArSL dataset. To minimise the imbalance caused by the heterogeneity of the class sizes, random undersampling was applied to the dataset to reduce the number of images from 54,049 to 25,600. Their proposed method achieved testing accuracies of 99.4% and 99.6% for the VGG16 and Resnet152, respectively. Despite the latest developments in DL and the good precision of image classification and prediction achieved using CNN, imbalanced data can affect the performance of prediction models. Imbalanced data can impact a model's ability to learn and its usage in real-time situations. The translation of the sign language gestures into different formats, such as text and speech, should also be further investigated.

## 3. EXPERIMENT METHODOLOGY

This section describes the proposed ArSL-CNN architecture that was designed for classifying Arabic Sign Language gestures. This section also describes the ArSL dataset and the pre-processing techniques that were applied to the dataset.

### 3.1. Proposed ArSL-CNN architecture

CNNs have achieved several breakthroughs as a basic DL technique for image classification problems, such as object detection and hand gesture recognition [6], [21]. Table 1 shows the architecture of the proposed ArSLCNN. Three types of layers are used in the CNN algorithm, namely, convolutional, pooling and fully connected layers. The pooling layer decreases the spatial size of an input sequence. The complete CNN architecture is obtained through several stacks of the abovementioned layers. The ArSL-CNN model is composed of seven convolutional layers, four batch normalisation (BN) layers, four pooling layers, five dropout layers and one fully connected layer with rectified linear unit (ReLU). This model ends with an output layer that has a softmax activation function to yield the distribution of the probability over classes as shown in Figure 1.

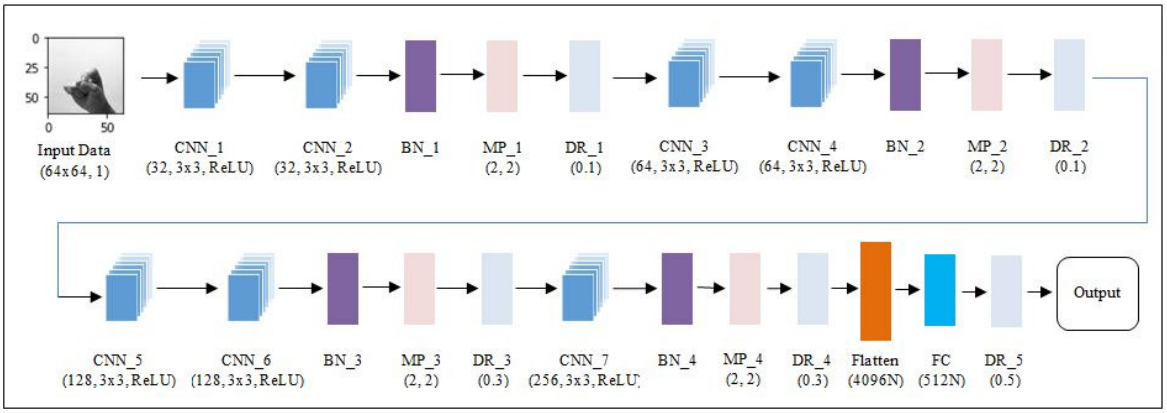


Figure 1. Architecture of the proposed ArSL-CNN model

Table 1 lists the detailed dimensions of each layer and operation. The first and second layers are convolutional layers that contain 32 feature maps and have a kernel size of 3×3. These layers are activated with ReLUs. The next layer is a BN layer, which aims to achieve the stable distribution of the activation values through training and normalise the inputs to a layer [24]. The fourth layer is a max pooling layer with a pool size of 2×2, and the objective of this layer is to decrease the number of parameters to minimise overfitting and decrease the computation time. The fifth layer is a dropout regularisation layer with the parameter set to 10%. The next layers are the third and fourth convolutional layers with 64 feature maps, a kernel size of 3×3 and ReLU activation function. These layers are followed by another max pooling layer with a pool size of 2×2, a BN layer and regularisation layers with parameter set to 10%. The next layers are the fifth and sixth convolutional layers with 128 feature maps, a kernel size of 3×3 and ReLU activation function, followed by another pooling layer with a pool size of 2×2, a BN layer and regularisation layers with parameter set to 30%. The last convolutional layer in the network is then laid after the regularisation layers. This layer with a ReLU activation function comprises 256 feature maps and has a kernel size of 3×3. The next layers are a max pooling layer with a size of 2×2, a BN layer and another dropout regularisation layer with the parameter set to 30%. The ArSL-CNN network architecture ends with fully connected units that contain a flatten layer, one fully connected layer, one dropout layer and the output layer. The flatten layer converts the 2D matrix data into a vector to allow the final output to be processed by standard fully connected layers. The second layer is a fully connected layer that contains 512 neurons with ReLU activation function. The dropout layer excludes 50% from neurons. The last layer is the output layer, which contains 32 neurons and is activated with a softmax activation function. The ArSL-CNN model is trained in a fully supervised way, and its parameters are optimised by minimising the cross-entropy loss function with the Adam version of stochastic gradient descent.

Table 1. Parameters of the ArSL-CNN architecture

| Layers | Layer Configuration | # Parameters |
|---|---|---|
| Convolution 1 | 32 filters, 3x3 kernel and ReLU | 320 |
| Convolution 2 | 32 filters, 3x3 kernel and ReLU | 9248 |
| batch Nor. 1 | - | 128 |
| Max-pooling 1 | 2x2 kernel | 0 |
| Dropout 1 | 0.1 | 0 |
| Convolution 3 | 64 filters, 3x3 kernel and ReLU | 18496 |
| Convolution 4 | 64 filters, 3x3 kernel and ReLU | 36928 |
| batch Nor. 2 | - | 256 |
| Max-pooling 2 | 2x2 kernel | 0 |
| Dropout 2 | 0.1 | 0 |
| Convolution 5 | 128 filters, 3x3 kernel and ReLU | 73856 |
| Convolution 6 | 128 filters, 3x3 kernel and ReLU | 174584 |
| Batch Nor. 3 | - | 512 |
| Max-pooling 3 | 2x2 kernel | 0 |
| Dropout 3 | 0.3 | 0 |
| Convolution 7 | 128 filters, 3x3 kernel and ReLU | 295168 |
| batch Nor. 4 | - | 1024 |
| Max-pooling 4 | 2x2 kernel | 0 |
| Dropout 4 | 0.3 | 0 |
| Flatten | 4096 Neurons | 0 |
| Fully connected | 512 Neurons | 209664 |
| Dropout | 0.5 | 0 |
| Output layer | Softmax 32 classes | 16416 |

### 3.2. Dataset description

The proposed ArSL-CNN architecture is trained and tested on the ArSL2018 [15], Arabic Sign Language (ArSL) dataset. The ArSL2018 dataset aims to provide an opportunity for researchers to develop automated ArSL recognition systems based on different machine learning methods. The original dataset consists of 54,049 RGB images distributed around 32 classes and the signs collected from more than 40 participants. The RGB images have different dimensions and many variations of images were presented through the use of different lighting and backgrounds. Samples of the dataset can be seen in Figure 2.


Figure 2. ArSL2018 dataset samples [15]

### 3.3. Image pre-processing

Data preprocessing is the implementation of various morphological activities to eliminate noise from the data. The ArSL2018 dataset includes sign language gesture images with different dimensions that were taken with varied illumination. Therefore, image preprocessing techniques are necessary to remove noise from the data before feeding them to the network. All sign images are firstly converted into greyscale images with a dimension of 64×64 to perform real-time classification. The greyscale colour space conversion allows operating in one channel only rather than processing in the three RGB channels. This conversion will minimise the number of parameters of the first convolutional layer two times and reduce the computational time. To increase the efficiency of the computation process and speed of the training stage, all images are normalised to set the range of the pixel values from 0 to 1. Then, the images are standardised by eliminating their means and scaling them to unit variance. To generate the training and testing sets, images are randomly selected from the dataset. The dataset is split into testing (20%) sets and training (80%) of which 20% is taken for validation. Figure 3 depicts that the number of samples for each class in the dataset is not balanced. Therefore, various resampling techniques have been applied to solve the imbalance problem amongst the classes. The details of this process are presented in Section 4.2., experimental results and discussion.

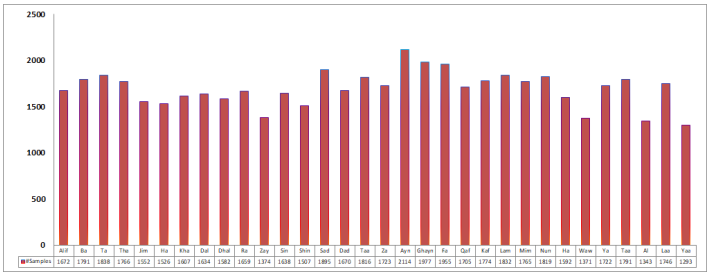


Figure 3. Number of samples in each class

## 4. RESULTS AND DISCUSSION

The experiment was conducted using Keras libraries and Python programming language that run on TensorFlow backend. The ArSL-CNN model was trained on a machine that has an NVIDIA K80 graphics processing unit (GPU), 64 GB random access memory, 12 GB memory and 100 GB solid state drive. To introduce randomness, the training dataset was shuffled before fed to the network to avoid bias towards certain parameters. The effectiveness of the proposed model was evaluated based on two independent experiments: (1) the proposed ArSL-CNN model was trained and tested using the original ArSL2018 dataset; and (2) the model was trained and tested using different resampling techniques to address the imbalance problem amongst the classes. The accuracy metric was adopted to determine the efficiency of the proposed CNN approach. In formula (1), A denotes the accuracy, TC and FC represent the number of correctly and incorrectly classified

instances, respectively. The calculated value is multiplied by 100 to turn it into a percentage.

$$A = \frac{TC}{TC + FC} \times 100 \tag{1}$$

For a class, the accuracy can be determined using (2).

$$Ac = \frac{TCc}{TCc + FCc} \times 100 \tag{2}$$

where, TCc represents the number of correctly classified instances which are from the class c, FCc represents the number of incorrectly classified instances which are from the class c. The final value is multiplied by 100 to get the percentage of the accuracy for each class.

### 4.1. Performance evaluation of the proposed ArSL-CNN model

The performance of the proposed ArSL-CNN model on the original ArSL2018 dataset is presented in Table 2. The training dataset consists of 54,049 images distributed over 32 ArSL gesture groups in a unified format. The training data were divided into batches with 128 samples each. The input and output layers have 4,096 and 32 neurons, respectively. The proposed ArSLCNN model was trained for multiple learning epochs. The training and testing accuracy values are summarised in Table 2. ArSL-CNN achieved the highest testing accuracy (96.59%) at 500 learning epochs. Figure 4 depicts the model accuracy when the proposed ArSL-CNN model is trained with 500 epochs. The training and testing performances are close to each other during different epochs which indicates that the model has not been overtrained.

Table 2. Classification accuracy and training time (minutes) obtained using the ArSL2018 original dataset

| No. Epochs | Training Acc. (%) | Testing Acc. (%) | Training Time (mins) |
|---|---|---|---|
| 500 | 98.80 | 96.59 | 66.1 |

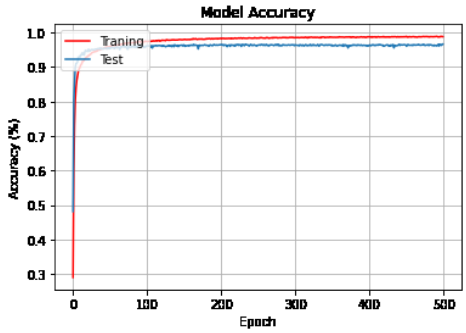


Figure 4. Accuracy of the proposed ArSL-CNN model obtained using the original ArSL2018 dataset

Table 3 indicates the accuracy of all 32 classes. From the table it can be observed that the number of testing samples across the classes varies considerably. It can also be observed that classes with the highest number of samples achieved a better accuracy than those the classes with fewer samples. For instance, the 'Waw' class contains 259 testing samples and its accuracy was 94.21%, whereas the 'Ayn' class contains 405 testing samples and its accuracy was 97.78%. These results revealed that the imbalanced distribution of the number of samples between classes may impact on the performance of the models, and in some cases, the model will be able to learn the classes that have more samples better than those with lower sample numbers. Therefore, it is important to apply techniques that can handle the imbalance problem between classes and to determine whether these techniques can improve classification performance, especially for the classes that contain smaller sample sizes. Therefore, resampling (over-sampling and under-sampling) methods are applied to the dataset and their impact on the performance of the ArSL-CNN model is explored (results are discussed in Section 4.2.).

Table 3. ArSL-CNN accuracy on the Ttest data using the original Arsl2018 dataset before applying sampling techniques (no sampling)

| Class No. | Class Name | #S[a] | #SCC[b] | Accuracy |
|---|---|---|---|---|
| 0 | Alif | 354 | 343 | 96.89 |
| 1 | Ba | 314 | 310 | 98.73 |
| 2 | Ta | 372 | 364 | 97.85 |
| 3 | Tha | 364 | 350 | 96.15 |
| 4 | Jim | 313 | 302 | 96.49 |
| 5 | Ha | 299 | 286 | 95.65 |
| 6 | Kha | 337 | 320 | 94.96 |
| 7 | Dal | 295 | 285 | 96.61 |
| 8 | Dhal | 328 | 318 | 96.95 |
| 9 | Ra | 310 | 303 | 97.74 |
| 10 | Zay | 265 | 259 | 97.74 |
| 11 | Sin | 336 | 317 | 94.35 |
| 12 | Shin | 316 | 294 | 93.04 |
| 13 | Sad | 388 | 380 | 97.94 |
| 14 | Dad | 361 | 350 | 96.95 |
| 15 | Taa | 355 | 349 | 98.31 |
| 16 | Za | 362 | 356 | 98.34 |
| 17 | Ayn | 405 | 396 | 97.78 |
| 18 | Ghayn | 376 | 361 | 96.01 |
| 19 | Fa | 391 | 377 | 96.42 |
| 20 | Qaf | 335 | 330 | 98.51 |
| 21 | Kaf | 371 | 362 | 97.57 |
| 22 | Lam | 354 | 340 | 96.05 |
| 23 | Mim | 346 | 338 | 97.69 |
| 24 | Nun | 353 | 349 | 98.87 |
| 25 | Ha | 325 | 318 | 97.85 |
| 26 | Waw | 259 | 244 | 94.21 |
| 27 | Ya | 365 | 352 | 96.44 |
| 28 | Taa | 374 | 345 | 92.25 |
| 29 | Al | 261 | 247 | 94.64 |
| 30 | Laa | 357 | 345 | 96.64 |
| 31 | Yaa | 269 | 256 | 95.17 |
| Total/ Average[c] | | 10810 | 10446 | 96.59 |

a. Number of samples in the test data
b. Number of samples correctly classified
c. The average is calculated by formula (2)

### 4.2. Results when using the ArSL-CNN model with oversampling and undersampling methods

The number of images per class in the ArSL2018 dataset is shown in Figure 3. As previously mentioned, the classes contain different sample sizes, and such discrepancies may result in an imbalance amongst the classes. The imbalance issue can have a negative effect on the classification results. To overcome this issue and reduce bias, resampling methods have been applied to balance the class distribution are classified into two groups: oversampling and undersampling methods [25]. The oversampling technique solves the imbalance amongst the classes by generating synthetic samples from minority samples. This approach can effectively improve the classification efficiency. However, increasing the number of samples in the minority classes will increase the training time. The oversampling process has two variations. The first is random minority oversampling (RMO), which randomly duplicates the minority class samples. The second is the synthetic minority oversampling technique (SMOTE), which is a sophisticated sampling technique that overcomes the issue of class imbalance by artificially generating samples through the interpolation of neighbouring data points [26]. The other method used for adjusting the balance of samples across ArSL2018 dataset classed was random minority under-sampling (RMU). The RMU strategy involves the random deletion of samples from majority classes until the dataset is balanced. A major drawback of this strategy is the possible loss of useful information. To correct the balance of samples amongst the classes in the ArSL2018 dataset, three resampling techniques, namely RMO, SMOTE, and RMU were applied to the dataset, and experiments were carried out to evaluate their impact on the task. Table 4 shows the results obtained using the three resampling methods. The findings reveal that the efficiency of the proposed ArSL-CNN model increases after applying the resampling techniques. The proposed model achieves training and testing accuracies of 99.14% and 97.21%, respectively,

by using the random oversampling method. The training and testing accuracy values after applying the undersampling method are 99.27% and 97.07%, respectively. By using SMOTE, the model obtains training and testing accuracies of 98.94% and 97.29%, respectively. This result implies that SMOTE outperforms the other two resampling methods in terms of the testing accuracy. The highest testing accuracy (97.29%) is achieved using SMOTE. This accuracy is higher than that obtained by implementing the ArSLCNN architecture on the original dataset (96.59%). These findings highlight the importance of having a balanced number of samples in each class in achieving high classification accuracy and minimising overfitting. Classes with small numbers of samples will reduce the accuracy of the proposed model.

Table 4. ArSL-CNN accuracy on the test data using the original Arsl2018 dataset before applying sampling techniques

| Resampling Technique | No. Epochs | Training Acc. (%) | Testing Acc. (%) | Training Time (mins) |
|---|---|---|---|---|
| RMU | 500 | 99.27 | 97.07 | 66.1 |
| RMO | 500 | 99.14 | 97.21 | 134.4 |
| SMOTE | 500 | 98.94 | 97.29 | 141.9 |

Figure 5 shows the confusion matrix generated by training the proposed ArSL-CNN model with SMOTE for 500 epochs. The diagonal elements in the confusion matrix reflect the number of correctly labelled images, whereas the off-diagonal elements denote the mislabeled images. The greater the sum of diagonal values of the confusion matrix, the higher the accuracy of the classification. The accuracy of the proposed ArSL-CNN model during various learning epochs after applying SMOTE is illustrated in Figure 6. The results reveal that the accuracy of the model on the training and testing sets increases for all learning epochs.

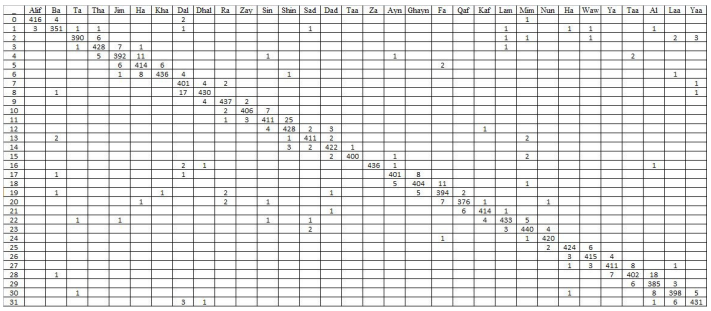

Figure 5. Confusion matrix of the proposed ArSL-CNN model with SMOTE

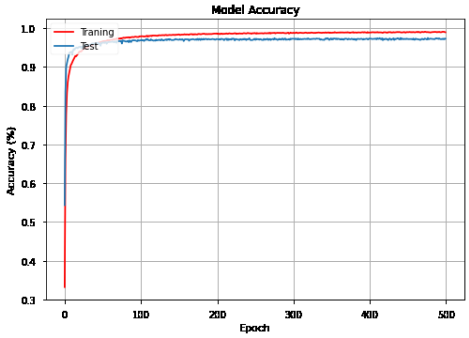


Figure 6. Accuracy of the proposed ArSL-CNN model with SMOTE

Furthermore, the accuracy per class is stated in Table 5. The experimental results show that ArSL-CNN obtained better classification efficiency when the RMU, RMO and SMOTE resampling method were applied. For instance, the number of samples in the 'Waw' class was 259 testing samples before using SMOTE

resampling method applied and the accuracy was 94.21%. However, after applying the SMOTE resampling method the number of samples increase from 259 to 422 testing samples and that led to increase the accuracy from 94.21% to 98.34%. These results approve the high impact of applying SMOTE resampling method to solve the imbalance problem and improve the overall accuracy of the proposed model.

### 4.3. Statistical analysis of the impact of the resampling methods applied to the ArSL2018 dataset on the performance of ArSL-CNN

Table 6 provides descriptive statistics of the test accuracy results when various sampling methods are applied to the ArSL2018 dataset. In Table 6, the first column is the sampling method applied to the dataset. The second column holds the mean test accuracy values across the 32 classes. The third column holds the standard deviation values which are a useful indicator of the stability of the model. The fourth and fifth columns show the minimum and maximum test accuracy values obtained and the last three columns hold information about the test accuracy value percentiles.

Table 5. ArSL-CNN accuracy on the test data after applying RMU, RMO and SMOTE

| Methods | RMU | | | RMO | | | SMOTE | | |
|---|---|---|---|---|---|---|---|---|---|
| Class Name | $\#S^a$ | $\#SCC^b$ | Accuracy | $\#S^a$ | $\#SCC^b$ | Accuracy | $\#S^a$ | $\#SCC^b$ | Accuracy |
| Alif | 351 | 348 | 99.15 | 423 | 419 | 99.05 | 423 | 416 | 98.35 |
| Ba | 366 | 361 | 98.63 | 362 | 348 | 96.13 | 362 | 351 | 96.96 |
| Ta | 365 | 359 | 98.36 | 404 | 390 | 96.53 | 404 | 390 | 96.53 |
| Tha | 342 | 333 | 97.37 | 438 | 426 | 97.26 | 438 | 428 | 97.72 |
| Jim | 298 | 288 | 96.64 | 412 | 401 | 97.33 | 412 | 392 | 95.15 |
| Ha | 266 | 254 | 95.49 | 428 | 411 | 96.03 | 428 | 414 | 96.73 |
| Kha | 337 | 324 | 96.14 | 451 | 438 | 97.12 | 451 | 436 | 96.67 |
| Dal | 324 | 308 | 95.06 | 408 | 394 | 96.57 | 408 | 401 | 98.28 |
| Dhal | 290 | 284 | 97.93 | 449 | 436 | 97.10 | 449 | 430 | 95.77 |
| Ra | 330 | 323 | 97.88 | 443 | 436 | 98.42 | 443 | 437 | 98.65 |
| Zay | 266 | 253 | 95.11 | 415 | 398 | 95.90 | 415 | 406 | 97.83 |
| Sin | 306 | 285 | 93.14 | 440 | 417 | 94.77 | 440 | 411 | 93.41 |
| Shin | 305 | 285 | 93.44 | 438 | 422 | 96.35 | 438 | 428 | 97.72 |
| Sad | 332 | 327 | 98.49 | 418 | 413 | 98.80 | 418 | 411 | 98.33 |
| Dad | 347 | 335 | 96.54 | 428 | 421 | 98.36 | 428 | 422 | 98.60 |
| Taa | 363 | 354 | 97.52 | 405 | 402 | 99.26 | 405 | 400 | 98.77 |
| Za | 333 | 327 | 98.20 | 441 | 439 | 99.55 | 441 | 436 | 98.87 |
| Ayn | 420 | 413 | 98.33 | 411 | 402 | 97.81 | 411 | 401 | 97.57 |
| Ghayn | 390 | 380 | 97.44 | 421 | 411 | 97.62 | 421 | 404 | 95.96 |
| Fa | 399 | 389 | 97.49 | 406 | 387 | 95.32 | 406 | 394 | 97.04 |
| Qaf | 339 | 328 | 96.76 | 389 | 378 | 97.17 | 389 | 376 | 96.66 |
| Kaf | 342 | 338 | 98.83 | 422 | 402 | 95.26 | 422 | 414 | 98.10 |
| Lam | 347 | 341 | 98.27 | 446 | 432 | 96.86 | 446 | 433 | 97.09 |
| Mim | 347 | 341 | 98.27 | 449 | 442 | 98.44 | 449 | 440 | 98.00 |
| Nun | 360 | 357 | 99.17 | 422 | 422 | 100.00 | 422 | 420 | 99.53 |
| Ha | 304 | 299 | 98.36 | 432 | 423 | 97.92 | 432 | 424 | 98.15 |
| Waw | 269 | 258 | 95.91 | 422 | 414 | 98.10 | 422 | 415 | 98.34 |
| Ya | 314 | 305 | 97.13 | 424 | 409 | 96.46 | 424 | 411 | 96.93 |
| Taa | 332 | 320 | 96.39 | 428 | 404 | 94.39 | 428 | 402 | 93.93 |
| Al | 262 | 255 | 97.33 | 394 | 387 | 98.22 | 394 | 385 | 97.72 |
| Laa | 363 | 343 | 94.49 | 413 | 395 | 95.64 | 413 | 398 | 96.37 |
| Yaa | 274 | 266 | 97.08 | 442 | 428 | 96.83 | 442 | 431 | 97.51 |
| Total/ Average$^c$ | 10583 | 10281 | 97.07 | 13524 | 13147 | 97.21 | 13524 | 13157 | 97.29 |

a. Number of samples in the test data
b. Number of samples correctly classified
c. The average is calculated by (2)

Table 6. Descriptive statistics of the test results when applying various sampling methods to the dataset

| Sampling method | Mean% | Std. Deviation | Minimum% | Maximum% | **Percentiles** | | |
|---|---|---|---|---|---|---|---|
| | | | | | 25th | 50th (Median) | 75th |
| No sampling | 96.59 | 1.64 | 92.25 | 98.87 | 95.74 | 96.77 | 97.83 |
| SMOTE | 97.29 | 1.37 | 93.41 | 99.53 | 96.66 | 97.65 | 98.32 |
| RMU | 97.07 | 1.57 | 93.14 | 99.17 | 96.20 | 97.41 | 98.32 |
| RMO | 97.21 | 1.40 | 94.39 | 100.00 | 96.19 | 97.15 | 98.33 |

Table 6 shows that the mean test accuracy value reached its highest, i.e. $\mu = 97.29\%$ = 97.29%, when the SMOTE resampling method was applied. With SMOTE, the proposed model achieved the lowest standard deviation value, i.e. $\sigma = 1.37$, and this suggests that applying SMOTE to the dataset results in a more stable prediction model. With RMO, the maximum test accuracy and the 75th percentile values were slightly higher than those of SMOTE, however, the higher standard deviation value of RMO suggests that using RMO results in a less stable model. The boxplots in Figure 7 illustrate the distribution of the test accuracy values with various sampling methods. Each boxplot in Figure 7 holds 32 values, where each value corresponds to a test accuracy value of the model for a particular class (note that there are 32 classes in the dataset as shown in Figure 3).

SMOTE has two outliers, and 'no sampling' and RMU have one outlier each, as shown in Figure 7. It is important to mention that the minimum value of SMOTE, i.e. 93.41%, is an outlier value as shown in Figure 7, and if the outliers are removed from SMOTE and RMU, then the minimum test accuracy values of SMOTE is the highest of all sampling methods, reaching 95.15% minimum test accuracy. Table 7 shows that applying sampling methods improves ArSL-CNN's performance, and the results suggest that best performance is achieved when SMOTE are applied to the dataset.

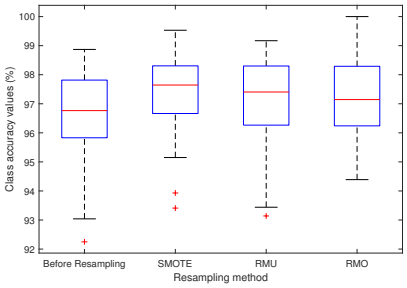


Figure 7. Boxplot of test accuracy values when various sampling methods are applied to the dataset

To determine whether the observed improvements in ArSL-CNN's performance when the SMOTE and other sampling methods are adopted are statistically significant at $\alpha = 0.05$, the non-parametric Wilcoxon Signed Ranks Test is applied to the test accuracy values obtained after applying the resampling methods to the dataset (see Table 7). The results revealed that when applying SMOTE, there is a statistically significant improvement in test accuracy (Z=-2.412, p=0.016). Indeed, there was also a weaker significant improvement in performance when applying the RMU and RMO sampling methods with p=0.042 and p=0.036 respectively. However, SMOTE achieved the most significant statistical improvement as indicated by the lowest p value. In conclusion, applying SMOTE resampling to adjust the class imbalance of the dataset significantly improves the test prediction accuracy of the model.

Table 7. Results of the wilcoxon signed ranks test applied to the test results

| Test Statistics[a] | No sampling vs. SMOTE | No sampling vs. RMU | No sampling vs. RMO |
|---|---|---|---|
| Z | -2.412b | -2.029b | -2.094b |
| Asymp. Sig. (2-tailed) | 0.016 | 0.042 | 0.036 |

a. Wilcoxon Signed Ranks Test
b. Based on positive ranks.
c. Based on negative ranks.

## 5. COMPARISON WITH STATE-OF-THE-ART METHODS

The performances of the proposed approach with existing state of-the-art techniques on the ArSL2018 dataset in terms of accuracy is shown in Table 8. The findings indicate that the proposed ArSL-CNN model when applying SMOTE resampling to the dataset is superior to two state-of-the-art methods in terms of overall accuracy [1], [3]. Ghazanfar et al. [1] used CNN and achieved an accuracy of 95.9%, whereas Elsayed and Fathy [3] applied semantic DL and obtained an accuracy of 88.8%. In comparison, our proposed method achieves an overall accuracy of 97.29%. This result implies the significance of providing a balanced number of samples to enhance the generalisation efficiency of CNN when training DL models.

Table 8. Comparison of the results obtained by the proposed approach and other previous methods

| Author | Methods used | #Sample | Recognition rate (%) |
|---|---|---|---|
| Elsayed and Fathy [3] | Semantic deep learning (SDL) | 54049 | 88.87 |
| Ghazanfar et al [1] | Convolutional neural network (CNN) | 54049 | 95.9 |
| Our approach 1 | ArSL-CNN | 54049 | 96.66 |
| Our approach 2 | ArSL-CNN +SMOTE | 67616 | 97.29 |

## 6. CONCLUSION

In this study, we established an Arabic sign recognition system by using a new ArSL-CNN architecture. Experiments were performed using the ArSL2018 dataset. The dataset originally consisted of 54,049 images collected from 40 users, which were then distributed amongst 32 classes. The proposed ArSL-CNN model initially achieved training and testing accuracies of 98.80% and 96.59%, respectively. The results demonstrated the challenges of working with imbalanced data and subsequently emphasized the importance of providing an adequate number of samples from each class to effectively train and test the DL models. The findings further revealed the effectiveness of the SMOTE oversampling method on the ArSL2018 dataset. The maximum classification accuracy of the proposed ArSL-CNN model was 97.29%. To our knowledge, this study was the first to investigate the class imbalance within the ArSL2018 dataset by using the SMOTE oversampling technique. Our future work will focus on evaluating the ArSL-CNN on more datasets, and investigating the performance recurrent neural networks for the task. In Arabic-speaking countries, there are many Arabic sign languages (ArSL) and these use the same alphabets. Such variation in ArSLs can create a communication barrier. Therefore, future work also involves using transfer learning to develop an advanced ArSL deep learning model that will work with ArSLs variations. Such a model can be utilised to overcome the communication barrier experienced by those using arabic sign language.

## REFERENCES

[1] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan, "An automatic arabic sign language recognition system based on deep cnn: An assistive system for the deaf and hard of hearing," *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 715-724, 2020, doi: 10.12785/ijcds/090418.

[2] S. Yang and Q. Zhu, "Video-based chinese sign language recognition using convolutional neural network," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. IEEE, pp. 929–934, 2017, doi: 10.1109/ICCSN.2017.8230247.

[3] E. K. Elsayed and D. R. Fathy, "Sign language semantic translation system using ontology and deep learning," *Sign*, vol. 11, no. 1, 2020, doi: 10.14569/IJACSA.2020.0110118.

[4] W. H. Organization, "Deafness and hearing loss 2019," [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[5] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3941–3951, 2017, doi: 10.1007/s00521-016-2294-8.

[6] A. A. Alani, G. Cosma, A. Taherkhani, and T. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," in *2018 4th International conference on information management (ICIM)*. IEEE, pp. 5–12, 2018, doi: 10.1109/INFOMAN.2018.8392660.

[7] K. Assaleh and M. Al-Rousan, "Recognition of arabic sign language alphabet using polynomial classifiers," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 13, p. 507614, 2005, doi: 10.1155/ASP.2005.2136.

[8] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. IEEE, pp. 960–965, 2014, doi: 10.1109/ISIE.2014.6864742.

[9] R. Alzohairi, R. Alghonaim, W. Alshehri, S. Aloqeely, M. Alzaidan, and O. Bchir, "Image based arabic sign language recognition system," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 3, 2018, doi: 0.14569/IJACSA.2018.090327.

[10] M. H. Rahman, J. Afrin *et al.*, "Hand gesture recognition using multiclass support vector machine," in *International Journal of Computer Applications*. published by Foundation of Computer Science, vol. 74, no. 1, pp. 39–43, 2013.

[11] S. K. Yewale and P. K. Bharne, "Hand gesture recognition using different algorithms based on artificial neural network," in *2011 International conference on emerging trends in networks and computer communications (ETNCC)*. IEEE, pp. 287–292, 2011, doi: 10.1109/ETNCC.2011.6255906.

[12] J. Triesch and C. Von Der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1449–1453, 2001, doi: 10.1109/34.977568.

[13] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016, doi: doi: 10.1109/TMI.2016.2535302.
[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[15] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "Arasl: Arabic alphabets sign language dataset," *Data in brief*, vol. 23, p. 103777, 2019.
[16] A. Tharwat, T. Gaber, A. E. Hassanien, M. K. Shahin, and B. Refaat, "Sift-based arabic sign language recognition system," in *Afro-european conference for industrial advancement*. Springer, 2015, pp. 359–370.
[17] Z. Wang, T. Jiang, B. Chang, and Z. Sui, "Chinese semantic role labeling with bidirectional recurrent neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1626–1631, doi: 10.18653/v1/D15-1186.
[18] N. B. Ibrahim, M. M. Selim, and H. H. Zayed, "An automatic arabic sign language recognition system (arslrs)," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 470–477, 2018, doi: 10.1016/j.jksuci.2017.09.007.
[19] M. Abdo, A. Hamdy, S. Salem, and E. M. Saad, "Arabic alphabet and numbers sign language recognition," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 11, pp. 209–214, 2015, doi: 10.14569/IJACSA.2015.061127.
[20] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE, pp. 342–347, 2011, doi: 10.1109/ICSIPA.2011.6144164.
[21] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A real-time hand posture recognition system using deep neural networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, pp. 1–23, 2015, doi: 10.1145/2735952.
[22] M. ElBadawy, A. Elons, H. A. Shedeed, and M. Tolba, "Arabic sign language recognition with 3d convolutional neural networks," in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE, pp. 66–71, 2017, doi: 10.1109/INTELCIS.2017.8260028.
[23] Y. Saleh and G. Issa, "Arabic sign language recognition through deep neural networks fine-tuning," 2020.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[25] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018, doi: 10.1016/j.neunet.2018.07.011.
[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

## BIOGRAPHIES OF AUTHORS

**Ali A. Alani** received his B.Sc. Degree in computer sciences from Diyala University, Diyala, Iraq in 2006 and M.Sc. Degree in Information Technology from Universiti Tenaga Nasional, Selangor, Malaysia in 2014. Recently, he is working as Assistant Lecturer in Department of computer sciences in university of Diyala, Diyala, Iraq. His research interests include Big data, Machine learning, Deep Learning and Computer vision.

**Dr. Georgina Cosma** received the Ph.D degree in Computer Science from the University of Warwick, UK, in 2008 and a First Class Honours BSc degree in Computer Science from Coventry University, UK, in 2003. She is currently a Senior Lecturer (Associate Professor) at the Department of Computer Science, Loughborough University, UK. Dr Cosma is a member of the IEEE Computer Society with Computational Intelligence, Big Data Community, and Brain Community memberships. Her research interests are in data science, artificial intelligence, natural language processing, and deep learning.