Iraqi Journal of Science, 2022, Vol. 63, No. 9, pp: 3977-3993
DOI: 10.24996/ijs.2022.63.9.29

ISSN: 0067-2904

Intelligent System for Multi-Layer Lip Reading Based Microlearning

Nada Hussain Ali [1*], Matheel E. Abdulmunim [2], Akbas Ezaldeen Ali [2]
[1]*Department of Computer Technology Engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq.*
[2]*Department of Computer Sciences, University of Technology, Baghdad, Iraq*

Received: 22/9/2021 Accepted: 19/11/2021 Published: 30/9/2022

Abstract

Intelligent systems can be used to build systems that simulate human behavior. One such system is lip reading. Hence, lip reading is considered one of the hardest problems in image analysis, and thus machine learning is used to solve this problem, which achieves remarkable results, especially when using a deep neural network, in which it dives deeply into the texture of any input. Microlearning is the new trend in E-learning. It is based on small pieces of information to make the learning process easier and more productive. In this paper, a proposed system for multi-layer lip reading is presented. The proposed system is based on micro content (letters) to achieve the lip reading process using deep learning and auto-correction models for error detection and correction of the accumulated words spelled by the lip reading system.The proposed systemwas implemented on a proposed dataset gathered using 20 persons of different ages and genders. The model achieved 98% accuracy in micro content recognition and 73.6% accuracy in accumulated word lip reading.

Keywords: Intelligent systems, deep learning, lip reading, microlearning, Levenshtein distance.

نظام ذكي متعدد المستويات لقراءة الشفاه بالاعتماد على التعلم المجهري

ندى حسين علي[1*], مثيل عمادالدين عبدالمنعم[2], اقباس عزالدين علي[2]
[1] جامعة الامام جعفر الصادق (ع) ، بغداد، العراق
[2] قسم علوم الحاسوب, الجامعة التكنولوجية، بغداد، العراق

الخلاصة

الانظمة الذكية هي الانظمة التي تستعمل لتحاكي سلوك الانسان. واحد من هذه الانظمة هو نظام قراءه الشفاه. قراءة الشفاه تعتبر من اصعب المشاكل في مجال تحليل الصور ولذلك تم استعمال تعلم الماكنة لحل هذه المشكلة والذي حقق نتائج جيدة خصوصا عند استعمال الشبكات العصبية العميقة والتي تغوص في عمق النسيج لاي نوع من البيانات المدخلة. التعلم المجهري هو التوجه الجديد في عالم التعلم الالكتروني والذي يعتمد على قطع صغيرة من المعلومات لتسهيل عملية التعلم وجعلها اكثر انتاجية. في هذا البحث تم تقديم نظام مقترح لقراءة الشفاه متعدد المستويات, النظام المقترح يعتمد على المحتوى المجهري المتمثل بالحروف لتحقيق عملية قراءه الشفاه باستعمال التعلم العميق والتصحيح التلقائي لاكتشاف الاخطاء وتصحيحها للكلمات المجمعة من قبل النظام المقترح. النظام المقترح تم تطبيقه على مجموعة بيانات مقترحة جمعت باستعمال 20

*Email: nada.hussien@sadiq.edu.iq

شخصا من مختلف الاعمار والاجناس. حقق النظام المقترح دقة بنسبة 98% لعملية تمييز المحتوى المجهري و 73.6% لعملية قراءه كلمات كاملة.

1. Introduction

In machine learning, visual speech recognition (VSR), also known as automatic lip-reading, is the process of recognizing the words through processing and observing the visual lip movement of a speaker's talking without any audio input. Although visual information alone cannot be considered sufficient to provide normal speech with intelligibility, it may succeed in a few cases, particularly when the number of words to be recognized is limited [1]. Visual lip-reading plays an important role in the interaction between human and computer in noisy environments where audio speech may be difficult to recognize. It can also be very useful for the hearing impaired as a hearing aid tool [2]. Despite the fact that audio signals are much more informative than video signals, it has been noticed that most people use lip-reading gestures to understand speech [3]. Lip reading is a difficult task for both machines and humans due to the considerably high similarity of lip shape and movements corresponding to uttering letters (e.g., letters b and p, or d and t). In addition to the lip movement, lip size, wrinkles around the mouth, orientation, brightness and the environment around the speaker also affect the quality of the detected words [4]. Microlearning presents the opportunity to absorb and retain the information provided and the activities that are more digestible and manageable. The method by which micro-learning identifies small portions of learning content that are made up of fine-grained and loosely-coupled learning activities that are interconnected and shortened learning activities that focus on the individual learning needs [5]. Deep networks, which are considered robust and precise learning techniques, are able to learn from data in the same way that babies are able to learn from the world around them, starting with fresh eyesight and gradually acquiring more skills needed to navigate the environments around them. Many difficult problems can be solved using the same learning networks; their solutions can be generalized and require much less work than writing a different program for each problem. The deep learning revolution has two convoluted themes: how Artificial Intelligence (AI) evolved and how human intelligence is evolving. The difference between the two types of intelligence is the time needed for evolution. Human intelligence took many years to evolve, but AI is evolving faster on a trajectory measured in decades. The conversion from an AI based on logic, symbols, and rules to a deep learning approach based on learning algorithms and big data is not easy [6]. Deep learning techniques will be the efficient solution that empowers classification techniques spatially on images [7].

Auto-correction is a trail to diminish the number of errors that can be generated by lip reading systems and to improve their accuracy, many error-correction techniques were visualized, several of them are manual works by post-editing the recognized output reflex to correct misspellings, while other techniques works on different ways[8]. Spelling correction is a well-versed task in Natural Language Processing (NLP). Automatic spelling correction is an important field for many applications in NLP, like text summarization, web search engines, sentiment analysis, and speech recognition [9].

Spell checking is the task of knowing which words are misspelled and correcting them [10]. There are three sub-problems in the spell checking task: non-word error detection, isolated-word error correction, and context-dependent error correction. Non-word error detection and correction techniques are divided into two types: n-gram and dictionary lookup. The most commonly used technique in optical character recognition is n-gram analysis, which works by finding unusual sequences of characters and considering them as an indicator for error. A more commonly used technique in spelling correction systems is dictionary lookup. This technique considers any word that does not appear in the dictionary as a misspelled word. Isolated-word spelling correction systems mostly use a form of minimum edit distance to rank or generate suggestions. Damerau discovered that over 80% of spelling errors involve one or

more of the following operations: a deleted letter, a letter substituted for another, an inserted letter, or two letters transposed or switched. Context-dependent error correction is used in cases where a word that is correctly spelled is replaced with another word. Statistical language models are the techniques that are used to detect ill-formed sequences of words. Word spelling correction systems mostly generate a rank or suggestion list for the misspelled word by using some kind of minimum edit distance. The Damerau–Levenshtein edit distance is one of the most commonly used techniques for such calculations [11].

Spellchecking can be implemented in speech recognition as well as lip reading. In speech recognition, there are three types of errors that occur during the recognition process. The first type occurs when a word sequence is translated as a different word. This type is called substitution. The second type, when a word is missing from the reference, means that the word in the reference is completely missing from the translation. This type is called deletion. The last type is insertion, which occurs when a word is inserted into the translation that has no reference [12].

The remaining sections of this paper are as follows: Section 2 presents a literature review of the related works. Section 3 contains the theoretical background of the techniques and algorithms used in the proposed system, while Section 4 includes the proposed system design and implementation. Section 5 presents the results and a discussion, and finally, Section 6 contains the conclusion and future work.

2. Related work

In the literature, several works are presented for the most relevant ones that are related to the proposed system.

The author, Christina Drakidou in [13], proposed that using microlearning in e-learning courses enhances lifelong learning and continuous learning. The author implemented several example courses that are carefully designed, supervised, and implemented by well-trained instructor-facilitators. The author proved that microlearning can be used as an e-learning technique that will improve learning outcomes. The authors, Gona Sirwan Mohammed, Karzan Wakil, and Sarkhell Sirwan Nawroly in [14], proposed that an important requirement for successful learning is experiencing learning activities on a regular basis and keeping them memorable for a long time. The microlearning can be delivered in small chunks, which makes it memorable and easy to understand. The authors tested microlearning techniques on primary school students, and they found that students who learned using microlearning gained better learning than students who were subjected to traditional learning. The researcher Elaine Rettger [15] presented the idea of employing microlearning using mobile devices for academic studies and how the delivery of instruction-distributed presentation will affect the learning outcome. The author proved that students receiving small units of instruction and information over a series of days would perform much better than students receiving the instruction and information in a massed unit. The author, Norm Friesen, in[16], suggested that traditional learning is forcing constraints on the learner. Microlearning enables personalized learning and liberates the learner from those constraints. The author thinks that these features of microlearning are important and valuable.

The authors, Yuanyao Lu and Hongbo Li, in [17], proposed a lip-reading system using deep learning to recognize numbers from 1-9 in videos. They used CNN to capture features and RNN to extract the sequence relationship between the video frames. The CNN and RNN are used as encoder and decoder, respectively. In the decoding process, an attention mechanism is used to learn attention weights. Therefore, the model takes the whole video as an attention area. The model gave an accuracy of 88.2% on the tested dataset. The authors in [18] proposed a visual based lip reading system from videos by presenting a novel convolution neural network called Hahn by changing the first layer of CNN and using the Hahn moment as the first layer. The proposed HCNN helped in reducing the dimensionality of the videos or

images and gave good results with 90% accuracy on different datasets. The authors, Joon Son Chung and Andrew Zisserman, in [19], proposed a model for profile lip reading instead of frontal view lip reading. They used a ResNet to classify the faces into 5 groups (frontal-left profile-left three quarters-right three quarters-right profile), and they used a SyncNet to achieve the purpose of the proposal by synchronizing the audio with the video lip motion, active speaker detection, and sequence to sequence feature generation model. The model achieved good results compared to other methods: frontal face 91%, 30 face angle 90.8, 45 face angle 90%, 60 face angle 90%, and profile face 88.9%. The authors, Cruz et al. [20], proposed a lip-reading model to recognize the English letters in Filipino speakers. The dataset was gathered from 30 speakers, 15 male and 15 female. The videos were pre-recorded for the speakers. The model depends on lip movement only and uses Point Distribution Model (PDM) and Kanade Lucas Tomasi (KLT) tracking algorithms to extract features from 16 key frames. A J48 decision tree algorithm is used for classification. The model achieved a 45.26% average accuracy. The authors, Ibrahim and Mulvaney, in [21], proposed a system for lip-reading that can recognize the English digits from 0-9. The model consists of four steps. The first step is to extract the face from the video, then the mouth area using Viola Jones's object recognizer. In the second step, two regions are detected in the mouth area, which are the lip and non-lip regions. The third step is to extract lip geometry using a proposed approach that depends on borders and convex hull computation to generate shape-based features. The final step, a novel approach, is used to classify the geometric features. This model achieved a word recognition accuracy of about 71%.

In [22], the authors tried to generate errors from a small seed of errors (misspelled words) based on an annotated corpus in order to have enough data to use deep neural networks for error detection and correction in typed words. The presented work achieved an accuracy of around 90%. In [23], the authors presented a system that checks the spelling and detects errors in Bengali words and suggests a suitable word to replace the misspelled word using Levenshtein distance and unigram strategy. The system was implemented on the added Bengali corpus. The system achieved an accuracy of 78%. The author concluded that the accuracy achieved is due to the huge amount of data used in the corpus, which is also the reason for the time taken by the system to suggest a suitable word. In [24], the authors presented a proposal for the Russian language using morphological and semantic information. The proposed system used the SpellRuEval contest (Sorokin et al., 2016) dataset to correct errors at the sentence level by employing a hypothesis of a noisy channel model and feature reranking and achieving an accuracy of 78%. In [25], the authors proposed a system that corrects real word errors based on contextual information implemented on the confused words that belong to the brown corpus, which has a set of confused words that are fed into the system. The authors presented two phases to correct the words: one uses the trigram algorithm and the other uses a Bayesian approach to achieve an accuracy of 89%. The limitation of this work lies in the corpus size. A bigger corpus means a more confused set of words that leads to better detection and correction. In [26], the authors presented a system that can detect and semantic errors in Arabic text by using four methods that are contextual based on linguistic information and statistical information. The system was implemented on the Arabic corpus by employing a multi-agent system. The system was able to detect the semantic validity of the words in sentences. The system achieved results with 90% precision and 83% recall. The system still has some limitations, like the size of the corpus.

3. Theorems and Algorithms

In this section, the used theorems and algorithms in the proposed work are explained.

3.1 Convolutional Neural Networks (CNN)

Deep learning in recent years has proven to be accurate on some tasks that surpass those of a human. Actually, the recent results gained from deep learning algorithms that transcend

human ability and performance in image recognition tasks have not likely been considered by computer vision experts in the last decade. Many architectures of deep learning that present such phenomenal performance are not the results of a random connection of computational units. The outstanding performance shown by deep neural networks reflects the fact that biological neural networks also obtain much of their strength and power from depth. Furthermore, it is not fully understood how biological networks are connected. In cases where the biological network structure is understood at some grade, great achievements have been made by modeling artificial neural networks based on those networks [27]. The main goal in applying deep learning to computer vision (CV) is to remove the exhausting, and limiting, feature selection process. Deep neural networks are very efficient for this process because it works in layers and each layer of a neural network is responsible for building up features and learning to represent the received input[28]. The architecture of deep learning is a bit like a stack of modules that is considered a multi-layer. All of these models, or most of them, are undergoing learning. All or many of them process non-linear input–output mappings. In this stack, each module diverts its input to boost both the invariance and selectivity of the representation of the model. With several layers that are non-linear, say a depth of 5 to 20, the system will be able to implement extremely complex functions of its inputs that are sensitive to details. The system can distinguish a dog from a muffin and is incurious to variations that are irrelevant, such as the pose, background, surrounding objects, and lighting[29]. Convolutional Neural Networks (CNNs) are a powerful combination of math, biology, and computer science. These neural networks have been one of the most effective innovations in the fields of AI and CV[30]. CNN enables learning and obtaining large quantities of information from raw data abstraction levels [31]. CNN consists of several components. These components are convolution layers, pooling layers, fully connected layers, activation functions, and dropout layers. The first layer, which is the convolution layer, contains a number of filters. These filters are responsible for the feature extraction process and they learn as the fully connected layers do [32]. These filters provide a chance to recognize and detect features regardless of their positions in the image. For that reason, these layers are called convolutions. In these layers (convolutional), the filters are initialized, then they go through a training procedure to shape filters that are suitable for the feature extraction task. For more benefits of this process, more layers can be added for more in-depth features by employing different filters in each layer[33]. Smaller objects are extracted from the input image. These objects are deep features from the original image. This process gets iterated in every convolution layer. The convolution process that leads to feature extraction can be considered a compression of important information extracted from the input image. After feature compression and deeper information representation in the convolution layer, another layer is needed called the max pooling layer. This layer may precede or follow the convolution layer. The max-pooling layer uses several hyperparameters that are often organized as a 2 x 2 grid. The image is divided into several areas the same size as the pool size (hyperparameters grid) and chooses from each pool (four pixels) the maximal value. These pixels compose a new image while preserving the order of the pixels in the original image. This process will produce an image that is half the size of the original image while keeping the channel number. An alternative to the maximal value can be chosen, like a minimum or average, in a way that better serves the process. The idea that lies behind the max pooling layer is that the important pixels that hold information about features are rarely adjacent in an image, so picking the maximum value from a set of four pixels will catch the pixel that is highly informative. This layer gives the best results when it is implemented on a feature map rather than the original image[34]. After several convolution and pooling layers, the architecture ends with a number of fully connected layers. The feature maps extracted from the convolution layers and pooling layers are transferred into vectors. At this point, to avoid overfitting, a dropout layer can be

added. These layers are virtual layers that drop some of the connections in the fully connected layers. The final fully connected layer in the architecture contains the same number of output neurons as the number of classes to be recognized [35].

3.2 Micro Content

Micro-content and microlearning together determine how to submit a quantum of information and knowledge, structured in many short sections, fine-grained, interconnected, and well-defined. This is a piece of information whose size is determined by a single topic; content that covers a single concept or idea and can be accessed via a single URL; being suitable for use on handheld devices, web-browsers, and emails; all of which refer to micro-content. Thus, micro-content is the part that merges into micro-learning [5]. In microlearning, knowledge is acquired using instructional design techniques, abilities, and skills, which happen on a daily basis. The way that microlearning works is by taking information naturally from the learner's brain, so that the body and brain do not get stressed. One of the essential features of microlearning that works saliently is that it allows the learner to find exactly what he or she is looking for. It enables the learner's brain to explore and satisfy its own patterns and its own curiosity [36]. Micro-learning has demonstrated its adaptability and flexibility in delivering micro-content via simple methods such as email, mobile, and network social networking. Micro-content is easy to update and can be used as a standalone learning unit, though it can also be used as a supporting unit in other learning techniques. The researcher found that using micro-learning can improve e-learning and can be very helpful for people who are seeking continuous learning [13].

3.3 Text similarity

The rapid data growth in recent years has caused some information problems. The text similarity approach was one of the solutions in many areas. Document, paragraph, and sentence similarity are based on finding similarities between words, which leads to text mining by finding the relevant information between words. Text mining is used in many systems, including text classification, document clustering, text summarization, machine translation, and auto correction [37]. Words are usually similar, either lexically or semantically. If the words have the same character sequence, they are considered lexically similar. If the words have the same meaning, they are considered semantically similar. But if the words do not have the same meaning but are used in the same context, then the words are not related. While lexical similarity can be measured using string-based algorithms, semantic similarity can be measured using knowledge-based and corpus-based algorithms[38]. Many similarity measures have been proposed and applied in a wide range of literature, such as the Jaccard correlation coefficient, cosine similarity, and Levenshtein distance. Similarity is often captured in terms of similarity and dissimilarity by a distance measure [39]. The Euclidean distance is regarded as one of the most effective metrics for locating the best neighbors for a misspelled word [40].

The dictionary lookup method is very popular in spellchecker systems, but a robust dictionary lookup needs complex calculations, especially with large-sized dictionaries. The Levenshtein distance measure is an effective technique for dictionary lookup that is able to reduce the complexity of the lookup process[41].

The Levenshtein distance, introduced by Vladimir Levenshtein, is a metric to check the similarity between two strings. The main interest of Levenshtein is to extend Hamming's error correction to include the insertion and deletion of single letters. The Levenshtein metric is the minimum number of edits needed, like substitutions, insertions, and deletions of single letters, to change X string into Y[42]. The Levenshtein distance algorithm is considered an editor algorithm that uses dynamic programming strings for operation. The algorithm works by giving a weight of 1 for every edit operation (substitution, insertion, deletion). For example, the Levenshtein distance between dog and cat is 3, one for substituting d for c, one

for substituting o for a, and one for substituting g for t. The algorithm can also be used for defining the number of adjustments and modifications like insertion and deletion in a string s1 to be the same as s2. It counts the number of operations on the strings[43]. Create a matrix LD [n + 1, m + 1] by assuming two strings S and T with lengths of m and n, respectively.
In order for the algorithm to calculate the value of each cell LD (i,j) in the matrix LD, the formula is as follows:

$$\text{LD}(i,j) = \begin{cases} 0, i = 0, j = 0 \\ j, i = 0, j > 0 \\ i, i > 0, j = 0 \\ min. i > 0, j > 0 \end{cases} \quad (1)$$

Min=min{LD(i-1,j)+1,LD(i,j-1)+1,LD(i-1,j-1)+f(i,j)},where f(i,j) =1 if the ith word of S is not equal to the jth word of T, otherwise f (i, j) = 0. Finally, the LD (n, m) of the rightmost corner of the matrix is the size of the desired edit distance. The similarity of the two strings can be represented by the LD matrix. Intuitively, the greater the LD, the smaller the similarity. Assuming that there are two strings S and T of length m and n, respectively, using LD to express their edit distance, using Sim (S, T) to show their similarity [44]:

$$\text{Sim}(S,T) = 1 - \frac{LD}{\max(m,n)} \quad (2)$$

4. Research method

The proposed system is constructed of three main layers; each layer complements the layer before it to produce a complete lip reading task. As illustrated in Figure 1, the proposed system multi-layer architecture describes the collective processes that represent the multi-layer lip reading system based micro-learning.
First of all, the dataset was proposed to fulfill the requirements of the proposed system. The dataset was built and processed during the proposal stages. It consists of thousands of different videos to level up the proposed system. As each layer requires a different type of video data, these videos pass through several processing steps to be employed in the proposed system.
Each layer in the architecture of the proposed system consists of several procedures to accomplish the layer's mission, then passes the output to the next layer. In the next layer, the input is processed and employed to meet its purpose. The layering structure of the proposed system is designed to be smooth and efficient as the data flows from one layer to the next. As the proposed system's output, the data is processed in each layer and reformatted from video data to image data to text data.
In the following sub-section, a description of the layers and stages of the proposed system is presented.

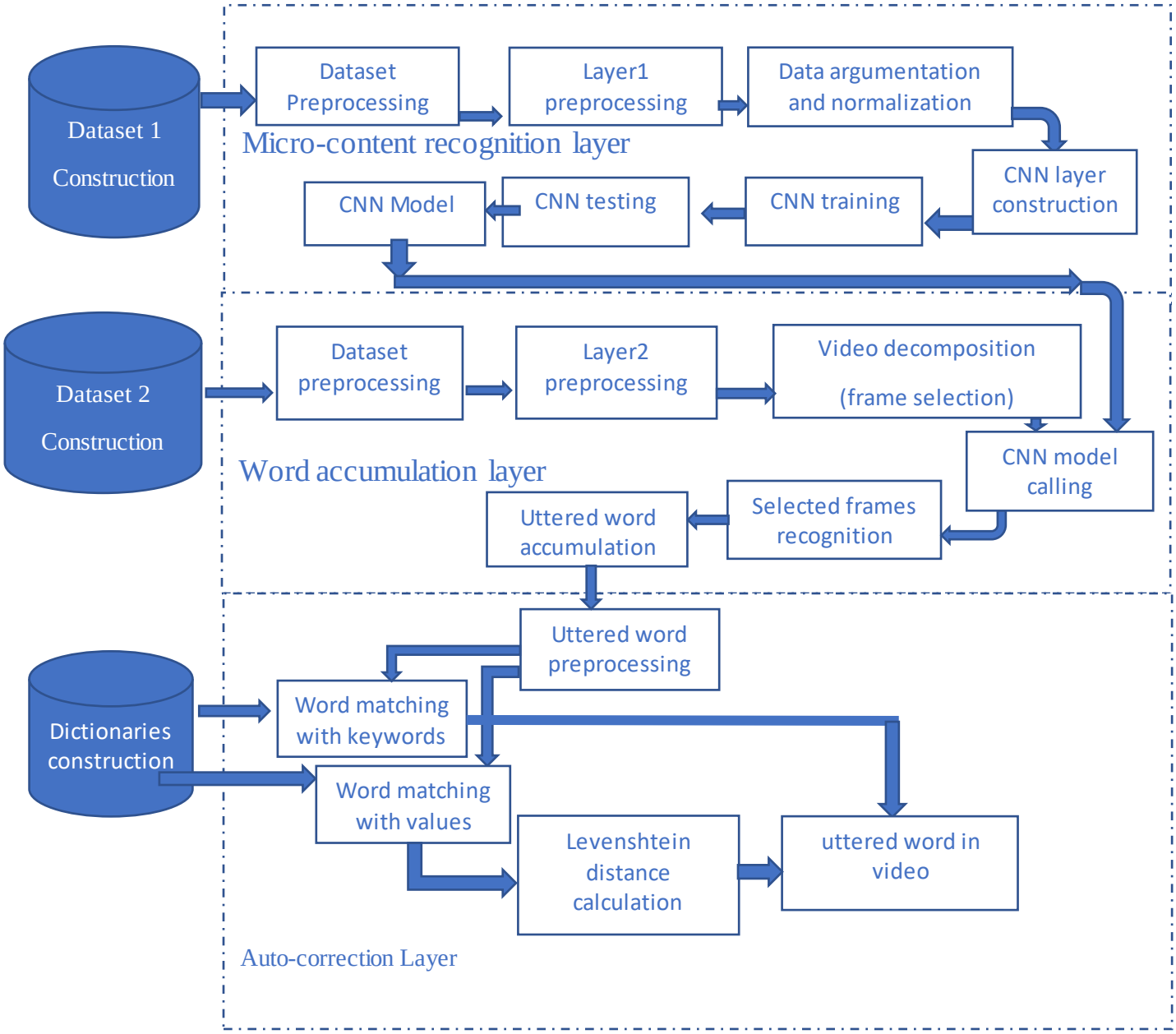

Figure 1- the proposed system multi-layer architecture

4.1 The Proposed Dataset

Any computer system, particularly one with learning capabilities, requires a dataset to train it in order to progress to successive training that leads to a learning task. The proposed IHUSRLR system works mainly with two types of datasets. Both datasets are proposed in which they are collected, built, and processed as an initialization stage. Each dataset contains video samples representing humans of various ages and genders. The two datasets are described in detail in the sub-sections below.

4.1.1 The Proposed Dataset based on Micro-content Videos (Dataset 1)

This dataset contains more than 2700 pre-recorded videos of 11 persons (male and female, of different ages). The videos do not exceed one second in length, consisting of the pronunciation of the English alphabet. Due to the difficulty of differentiating between similar pronounced letters, this similarity originates from the mouth geometry during letter utterance but not from the acoustic information. Thus, these letters like (A, U) , (F, V),(P, B) , (K, C,Q), (S, X) were merged together. The recording process was held in several artificial lighting

conditions. The distance between the camera and the people was 30 centimeters, and the height was horizontal to the face. Each video has the top part(from the shoulders) of the person pronouncing the letters.

4.1.2 The Proposed Dataset based on Uttered Words Videos (Dataset 2)

This dataset consists of more than 500 pre-recorded videos of 11 persons (male and female, of different ages). Each video is two to three seconds in length. The videos contain different words; each word consists of different lengths of uttered letters. The recording process was held in several artificial lighting conditions. The distance between the camera and each person was 30 centimeters, and the height was horizontal to the face. Each video has the top part (from the shoulders) of the person pronouncing the words.

4.1.3 Dataset preprocessing

Preprocessing plays an important role in any system. In the proposed system, the preprocessing of the dataset is conducted twice with both datasets. The preprocessing of the dataset passes through several steps as described below:

1. Convert the video into frames. In this step, the videos are converted into frames (29 frames per second). The frames are saved for next steps.

2. Face detection step: In this step, the Haar Cascade face detection technique is used to detect the face in the frame and crop the face area only.

3. Mouth detection step: the output from the previous step is fed into this step, and the mouth area is cropped using the spatial coordinate detection technique.

4. Key frame selection: In this step, a key frame (or frames) is selected based on visual features. This frame (or frames) represents the utterance letter and distinguishes it from other letters.

After these steps, a prepared dataset is formulated and constructed. For the first dataset, which consists of micro-content of the utterance letters, the key frames have the mouth area only. For the second dataset, the formulation is implemented until step three (mouth area cropping) and further processing is implemented on it. That will be explained in the second layer. Figure 2 shows the dataset through several steps.

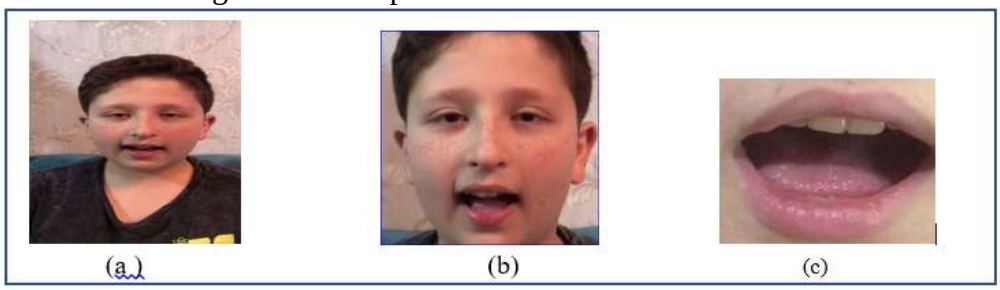

Figure 2-Dataset preprocessing steps, (a) The frame extracted from the video without preprocessing, (b) The frame after detecting the and cropping the face,(c) After cropping the mouth are only.

4.2 The Micro-content Recognition Layer Design

The first layer of the multi-layer structure of the proposed system is the micro-content recognition layer. The purpose of this layer is to build the CNN model layer structure, train the CNN model with a micro-content dataset, and then test the CNN model to recognize the micro-content given. This layer has several steps to fulfill its purpose. This process is described in [45]. These steps are described as follows:

A. Layer preprocessing: After the dataset has been preprocessed and prepared as a formulated and constructed form for the recognition process, the layer preprocessing stage is achieved as the data will be ready for the recognition process. The following steps illustrate the layer preprocessing stage:

1. Extracting the labels from the dataset: Each letter frame is stored in a file with the same name as the letter (A for letter A, so the others). These names are compared with the labels given to them to consider them targets.
2. Reshape: The frames are reshaped into square 224*224 images.
3. Dataset partitioning: The dataset is partitioned into two categories: the training set at 75% and the testing set at 25%.

B. Data Augmentation and Normalization: Data augmentation techniques are used to expand the dataset because when using deep learning, the dataset must be large enough in order to avoid an overfitting problem. This problem happens when the neural networks can't generalize to the testing set because the neural network learned the features of the training set so well that it cannot generalize. Employing data augmentation on the dataset is as follows:

1. Rotating the images within 30 degree.
2. Zooming the images with 0.15 percentages.
3. Shafting the images in the width 0.2 degree.
4. Shafting the images at a height of 0.2 degree.
5. Shearing the images in rang equals to 0.15.
6. Horizontal flipping.

After employing data augmentation, each frame has several copies that are rotated, zoomed, shafted, sheared or flipped. Now that the data is large enough to proceed with deep learning, the next step is to normalize the data before feeding it to CNN.

The mean subtraction technique is used to normalize the data. In this technique, the mean RGB value for the training data set is computed and then subtracted from every pixel.

C. Micro Content Recognition Using Convolution Neural Network

In this work, a CNN is used to recognize the letters as 20 classes for 20 letters. The CNN model was pre-trained with image-net weights. The model consists of several layers; 16 convolution layers, 3 fully connected layers, and 5 max polling layers. The purpose of using the convolution layers (the operation of convolution is declared in equation 3) of the CNN model is to make use of the pre-trained weights and not start with completely random weights. The network and the weights are loaded and used for the feature extraction process only. The process was as follows:

First: the network is loaded with the weights of the image net dataset, which is a dataset that has over a million images and can classify more than a thousand object classes.

Second: the network is trained with the proposed dataset in order to extract feature maps using the convolution layers and the loaded weights. The layers of the CNN model are as follows:

1.	Conv3*3(64)	6. MaxPool(2,2)	11. MaxPool(2,2)	16. MaxPool(2,2)
2.	Conv3*3(64)	7. Conv3*3(256)	12. Conv3*3(512)	17. Conv 3*3(512)
3.	MaxPool(2,2)	8. Conv3*3(256)	13. Conv3*3(512)	18. Conv 3*3(512)
4.	Conv3*3(128)	9. Conv3*3(256)	14. Conv3*3(512)	19. Conv 3*3(512)
5.	Conv3*3(128)	10.Conv3*3(256)	15. Conv3*3(512)	20. Conv 3*3(512)
				21. Maxpool(2,2)

Where 3x3 means a 3 by 3 mask with stride 1 that will be convolved over the image, while the numbers between brackets (64),(128),(265), and (512) are the number of parameters in each layer, and the numbers (2,2) are the mask of the maxpool layer with stride 2.

$$\text{Convolution} = \left| \frac{\sum_{i=1}^{q} \{ \sum_{i=1}^{q} f(ij) d(ij) \}}{F} \right| \quad (3)$$

Where f(ij) =the coefficient of a convolution kernel at position (ij) in the kernel
d(ij) = the data value of the pixel that correspond to f(ij)
q = the dimension of the kernel if the kernel 3*3 then q=3.
F= either the sum of the coefficients of the kernel or 1 if the sum of the coefficient is zero.
Convolution = the output pixel value.

Maxpool = Maximum value of{ 4 values from the 2x2 maxpolling layer kernel} (4)

After the extraction of the feature maps by using the CNN model, the next step is to build a head model for the classification process. The feature maps are fed to several layers as follows:

1. max pooling layer with pool size(3,3).
2. flatten layer.
3. fully connected layer with 512 nodes.
4. dropout layer with 0.5 percent.
5. fully connected layer with 20 output nodes(number of classes) using a soft max activation function.

The final step in the training process is to compile the model using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.0001 and a momentum term of 0.9 and decay of 0.0001.

The Gradient Descent Optimizer is a method to minimize an objective function $J(\theta)$ given parameter values by a model's parameters $\theta \in R$ d. It works by updating the parameters used in the model in the opposite direction of the gradient of the objective function $\nabla\theta J(\theta)$ to the parameters.

The learning rate η determines the size of the steps that must be taken to reach a (local) minimum. The SGD optimizer updates the parameters in each training epoch for training $x^{(i)}$ and label $y^{(i)}$ [46].

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x(i); y(i)) \quad (5)$$

At the end of this layer, the CNN model is trained and tested with the micro content dataset, and it is ready to recognize frames with micro content**.**

4.3 Uttered word accumulation layer

In this layer, the input dataset is a video that contains an uttered word. The system decomposes this video and recognizes each frame for which micro-content class it belongs to (which letter). After the recognition of each micro-content, the uttered word in the video is accumulated. A detailed description will be given in the following sub-sections. This layer also has several stages as follows:

A-preprocessing: At this stage, the dataset is preprocessed as described in section 4.3.1 except for the fourth step, which is key frame selection. At this stage, the key frame selection follows a different route. The parameters are also set in this layer. These parameters include the number of frames that video has, the number of letters that a word has, and the range for the key frame selection process. The first parameter is determined by setting a counter for the number of frames the video has when converting it from video to a stream of frames. The second is determining the number of micro-content (letters) the uttered word in a video has. This parameter is important due to its role in the key frame range selection process. The third parameter is determining the number of key frames for each range. These key frames will be the only frames of the video which will be considered for the recognition process.

B-Key frame selection stage: At this stage, the video is decomposed to select a number of frames to be considered for the recognition process. First, by dividing the video frame stream into a number of intervals that equals the number of microcontents uttered in the video. Then, each interval of frames is divided into three parts, and only the middle part is selected as key frames. Finally, the starting point and ending point for each range will be determined. This process is described in the following steps:

Step 1: Determining the number of frames for each interval according to the proposed equation 5.

$$\text{Number of frames for each interval} = \frac{\text{number of frames of the video}}{\text{number of word micro content}} \quad (5)$$

Step 2: For each word, the number of intervals equals the number of microcontents of that word. Each interval must have a range of frames to be considered for the recognition process. The number of selected frames for each interval is computed according to proposed equation 6.

$$\text{Number of selected frames (NSF) for each range} = \frac{\text{number of frames for each interval}}{3} \quad (6)$$

Where 3 represents that each interval is divided into 3 parts (ranges) to take into consideration only the middle part of each interval.

Step 3: According to equation 6, the middle range is selected, the starting frame and ending frame for the selected range of frames are assigned and determined.

C- Word accumulation process: At this stage, the system accumulates and reads the uttered word. The frames that are selected for each range from the previous stage are recognized using the CNN model built and trained in the first layer. All the frames in the determined ranges are recognized, and the frame label with the highest appearance is considered as the label for that micro content. For example, if the first range has 10 frames, 5 of them were recognized as A letters, 3 frames were recognized as E letters, and 2 of them were recognized as C letters, then the A is considered as the letter for that range. Each range will represent a single letter in the uttered word. After all the ranges of frames are recognized and the label with the highest appearance is chosen, the word is accumulated letter by letter (micro content).

4.4 Auto correction layer

In this layer, the accumulated words from the second layer are processed for error detection and correction. This layer uses text similarity and dictionary lookup techniques to detect errors and correct them. This layer is described in detail in [47], and the steps of this layer are illustrated below:

A-The first step creating the dictionary: The three types of dictionaries were created: 3-letter words, 4-letter words, and 5-letter words by choosing the number of letters when executing the model. The purpose of dividing the dictionaries into three types is to speed up the matching process. Each element in the dictionary has a value and an associated keyword. The value is the misspelled word of the keyword. The cases of misspelled words were gathered from different speakers uttering the words correctly but with one or two letters wrong, like (fan recognized as lan, man recognized as mas). These misspelled words were entered as the value for the correct word mas:man , lan:fan. The dictionary can be expanded as more cases and misspelled words are entered, leading to a more global dictionary for the purpose of auto-correction for lip reading.

B-The second step searching for a match: In this step, a search for a match is conducted. First, a search for a match between the word of interest and the keywords of the dictionary. If the word is found, that means it is not misspelled. If the word was not found, that means it is a misspelled word. The search process will be faster when choosing the number of utterance letters in the word (silent letters are not included). For instance, instead of searching in a dictionary that contains 1000 words of different lengths, perform a search in a dictionary that has 200 words only of a specific length.

C-The third step Levenshtein distance calculation: When the search in step two fails to find a match, it means the word is misspelled. The misspelled word is subjected to a Levenshtein distance calculation with all the values from the dictionary instead of the keywords. The reason why the search is implemented with the values from the dictionary instead of the keywords is that during the testing of the model, almost all speakers had the same misspelling

(a missing letter or incorrectly recognized letter). For that reason, the search with the values was more useful, faster, and more accurate than calculating the Levenshtein distance with the keywords. The Levenshtein distance is calculated according to equations 1 and 2. The top 3 closest words with the minimum value are considered as suggestions for the word of interest. Once the closest words are found, the associated keyword is presented as the utterance word after it has been subjected to auto-correction.

5- Results and Discussions

The results of this system is divided into three parts, the first part is the results for the first layer (micro content recognition) while the second part discuss the ability of the system to recognize the frames from the video that has the uttered word and its ability to recognize the micro content in its correct place from the accumulated word and finally the last part will discuss the result of the auto correction layer and the final results of the system.

A-Micro content recognition results

The testing stage is implemented on 25% of the dataset. The model achieved a remarkable result on the testing set. Table (1) shows the results of the dataset. The results show that the training was successful and the model could recognize 20 letters with an accuracy of 95% on the training dataset and 98% on the testing dataset. The training set had more near-miss classification in regards to the testing set, which led to a slight difference in the computed accuracy.

Table 1-Measurements criterion results

Letters	precision	recall	f1-score	support
A,U	0.99	0.99	0.99	276
B,P	0.98	0.97	0.97	127
C,K,Q	0.99	0.98	0.99	177
D	0.97	0.97	0.97	119
E	0.96	0.88	0.92	170
F,V	1.00	0.98	0.99	447
G	0.96	0.99	0.97	233
H	0.95	0.96	0.95	134
I	0.98	0.99	0.98	372
J	1.00	1.00	1.00	201
L	0.94	0.97	0.95	163
M	0.98	1.00	0.99	628
N	1.00	0.99	0.99	142
O	0.99	1.00	0.99	549
R	0.93	0.97	0.95	143
S,X	0.99	0.99	0.99	320
T	0.99	0.94	0.96	87
W	0.99	0.99	0.99	320
Y	0.99	1.00	0.99	292
Z	0.97	0.92	0.94	73
Total accuracy		0.98		5078

From the above table, it can be noticed that several letters have 99-100 results. The letters that had these distinguished features could more easily recognize them from other letters, whereas the letters with less than 99% accuracy were more difficult to recognize due to their large similarity with other letters. This challenge of similar letters, like the letter E, is very similar to the letter A, but the model recognizes the frames that have the same features as A more than as E. Although it was hard to distinguish between them, the model achieved excellent results. The letter J had an accuracy of 100% because there were no other letters that had the same features as the letter J.

In the second layer of the system, the number of words the system accumulated was more than 850 words of different lengths, starting with 3 letter words. The words were divided into

two parts: the first part was inserted into the dictionaries, and the second part was for testing the system. The system accuracy is calculated in this layer based on the number of micro-content (letters) that are correctly recognized in their correct place in the word. For example, if the input video has the word "fan" and the system accumulates as "fam", the accuracy is computed by counting how many letters were correctly placed during the accumulation process, like for the given example, the accuracy is 66%.

Table 2 contains several examples of the words accumulated by this layer of the system, while the accuracy of this layer is illustrated in Table 3.

Table 2-Accumulated words examples

The correct words	Read by the lip reading system
fan	aan
mice	mii
wall	ooll
see	sii
nine	lan
left	lifi
meet	mel
lime	lnn
soup	som
business	msnns
adjust	aijsi
bowl	mool
helpful	ailfol
afford	afoci

From the above table, it is shown that this layer accumulated the words letter by letter, taking into consideration that only the uttered letters were considered in this process, like the word afford is accumulated as a 5 letter word A-F-O-R-D and the word adjust is accumulated as A-J-U-S-T.

Table 3-Word accumulation accuracy

Number of words	Number of letters	Number of correctly accumulated letters
850	**3691**	**1790**
Total accuracy	**48.5**	

As shown in Table 2, layer 2 of the system misplaced more than one letter in some words, and in others it misplaced only one letter. For that reason, the necessity has arisen to integrate the system with an auto-correction layer.

The auto correction layer, which is considered an integrated part of the lip reading system, succeeded in boosting the accuracy by more than 20%. The accuracy of the proposed auto-correction layer for the lip reading system was achieved with more than 73% tested on more than 300 videos of different words. The layer was able to recognize the words "Fat" from the word "Fate" and "Fan" from the word "Man", but the layer was unable to recognize some other words. The levenshiten distance algorithm produced the closest distance values between the accumulated words from layer 2 and the dictionary. The top 3 closest words were considered as output (suggestions to the user). Table 4 illustrates the results of this layer.

Table 4-Auto correction layer accuracy

Total Number of Samples (videos) in dictionary	Number of testing Samples (videos)	Number of corrected words in testing samples	Number of uncorrected words in testing samples
550	300	221	79
Accuracy	73.6%		

6- Conclusions

The proposed system for multi-layer lip reading based on microlearning succeeds in achieving the aim of the system with acceptable accuracy by using three consequential proposed layers and two proposed datasets. The use of deep learning in the first layer had a great impact on the overall accuracy of the system, but choosing the appropriate CNN model was crucial to avoid overfitting problems. Merging the similar letters from the dataset also enhanced the accuracy of the proposed system, along with the preprocessing stage that led to training the CNN on the region of interest only (mouth). In the second layer, choosing the key frames that will be considered for the recognition process was very important. After trying several intervals, a decision was reached that the middle part of each interval gave the best accuracy in regard to choosing the correct frames for each letter. Although the accuracy of this layer does not depend on the key frame selection process, it depends on the CNN model also. The use of the Levenshtein distance metric in the auto correction layer really boosted the total accuracy of the proposed system, as it was a convenient choice for letter-by-letter comparison. As the whole system was built on the assumption of accumulating the words letter by letter, the Levenshtein distance metric played an important role in compensating for and correcting the misspellings caused by the merged letters. One more thing that accelerated the system was splitting up the dictionaries into several dictionaries, which decreased the time needed for finding the match for any word.

For future work to enhance the accuracy of the system, a corpus must be built for thousands of misspelled words based on uttered words by different speakers with different accents. The proposed system can also be implemented with an Arabic dataset to lip-read Arabic words.

References

[1] Ziheng Zhou, Xiaopeng Hong, Guoying Zhao, Matti Pietika, "A Compact Representation of Visual Speech Data Using Latent Variables," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 36, no. 1, Jan 2014, doi.10.1109/tpami.2013.173.

[2] Amit, Amit Garg, Jonathan Noyola and Sameep Bagadia, "Lip reading using CNN and LSTM," *Stanford University* ,2016.

[3] Adriana Fernandez-Lopez, Federico M. Sukno, "Survey on automatic lip-reading in the era of deep learning," *Image and Vision Computing*,Volume 78, Pages 53-72, 2018, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2018.07.002.

[4] Amany M. Sarhan , NadaM. Elshennawy, Dina M. Ibrahim, "HLR-Net: A Hybrid Lip-Reading Model Based on Deep Convolutional Neural Networks," *Computers,Materials & Continua, 2021* DOI:10.32604/cmc.2021.016509.

[5] Luminiţa Giurgiu, "Microlearning an Evolving eLearning Trend", *Scientific Bulletin* Vol. XXII No 1(43) 2017, DOI: 10.1515/bsaft-2017-0003.

[6] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos and Dionisis Kandris , "A Review of Machine Learning and IoT in Smart Transportation," *Future Internet* , 2019. doi.org/10.3390/fi11040094.

[7] Salih W.M., Nadher I., Tariq A., "Deep Learning for Face Expressions Detection: Enhanced Recurrent Neural Network with Long Short Term Memory," In: Khalaf M., Al-Jumeily D., Lisitsa A. (eds) *Applied Computing to Support Industry: Innovation and Technology. ACRIT,* 2019. Communications in Computer and Information Science, vol-1174, Springer, Cham. https://doi.org/10.1007/978-3-030-38752-5_19

[8] Youssef Bassil, Mohammad Alwani, "Post-Editing Error Correction Algorithm For Speech Recognition using Bing Spelling Suggestion," *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 3, No. 2, 2012

[9] Pravallika Etoori, Manoj Chinnakotla, Radhika Mamidi, "Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning," *Proceedings of ACL 2018*, pages 146–152 Melbourne, Australia, July 15 - 20, 2018. Association for Computational Linguistics

[10] S. M. El Atawy, A. Abd ElGhany, "Automatic Spelling Correction based on n-Gram Model," *International Journal of Computer Applications* (0975 – 8887) Volume 182 – No. 11, August 2018.

[11] Kenneth H. Lai , Maxim Topaz , Foster R. Goss , Li Zhou, "Automated misspelling detection and correction in clinical free-text records," *Journal of Biomedical Informatics* 55 (2015) 188–195, 2015.

[12] Rahhal Errattahi, Asmaa EL Hannani , Hassan Ouahmane, "Automatic Speech Recognition Errors Detection and Correction: A Review," *International Conference on Natural Language and Speech Processing, ICNLSP, Elsevier* ,Pp 32–37, 2018.

[13] Christina Drakidou, "Micro-learning as an Alternative in Lifelong eLearning," MA Dissertation, Aristotle University of Thessaloniki School of Italian Language and Literature, Thessaloniki 2018.

[14] Gona Sirwan Mohammed, Karzan Wakil and Sarkhell Sirwan Nawroly, "The Effectiveness of Microlearning to Improve Students' Learning Ability," *International Journal of Educational Research Review*, 2018. DOI: 10.24331/ijere.415824

[15] Elaine Rettger, "Microlearning with Mobile Devices: Effects of Distributed Presentation Learning and the Testing Effect on Mobile Devices," Ph.D. Dissertation, Arizona State University, USA, 2017.

[16] Norm Friesen, "The Microlearning Agenda in the Age of Educational Media", *eLearning Papers*, Thompson Rivers University, Canada 2007.

[17] Yuanyao Lu and Hongbo Li, "Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory," Licensee MDPI, Basel, Switzerland, 2019, doi:10.3390/app9081599.

[18] Abderrahim Mesbah et al., "Lip Reading with Hahn Convolutional Neural Networks moments," *Image and Vision Computing, Elsevier*, In press.hal-02109397,2019.

[19] Joon Son Chung, Andrew Zisserman, "Lip Reading in Profile," *British Machine Vision Conference*, September 2017. DOI: 10.5244/C.31.155

[20] Cruz, Hans Miguel , Puente, Jofet Kane T, Santos, Christian1, Vea Larry A., Rajendaran Vairavan," Lip Reading Analysis of English Letters as Pronounced by Filipino Speakers Using Image Analysis," *1st International Conference on Green and Sustainable Computing (ICoGeS) Journal of Physics*, 2017, doi :10.1088/1742-6596/1019/1/012041.

[21] M.Z. Ibrahim, D.J. Mulvaney, "Geometrical-based lip-reading using template probabilistic multi-dimension dynamic time warping," *Journal of Visual Communication and Image Representation*, Volume 30, pp 219-233, 2015. https://doi.org/10.1016/j.jvcir.2015.04.013.

[22] Kshitij Shah, Gerard de Melo, "Correcting the Autocorrect: Context-Aware Typographical Error Correction via Training Data Augmentation," *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 6930–6936 Marseille, 11–16 May 2020.

[23] Mosabbir Hossain, Farhan Labib, Ahmed Sady Rifat, Amit Kumar Das, Monira Mukta, "Auto-correction of English to Bengali Transliteration System using Levenshtein Distance," *7th International Conference on Smart Computing & Communications (ICSCC)*, 2019.

[24] Alexey Sorokin, "Spelling Correction for Morphologically Rich Language: a Case Study of Russian," *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 45–53, Valencia, Spain, 4 April 2017.

[25] Sumit sharma, Swadha Gupta, "A correction model for real-word errors," *4th International Conference on Eco-friendly Computing and Communication Systems*, Elsevier, 2015.

[26] Chiraz Ben Othmane Zribi, Mohamed Ben Ahmed, "Detection of semantic errors in Arabic texts," *Artificial Intelligence* 195 , Elsevier, pp. 249–264, 2013.

[27] Charu C. Aggarwal, "Convolutional Neural Networks," in "*Neural Networks and Deep Learning*", NY, USA,Springer International Publishing AG, part of Springer Nature, pp. 326-382, 2018.

[28] Nikhil Buduma, Nicholas Lacascio, "Convolutional Neural Networks: *Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms*," First addition, USA, Oreilly, 2017.

[29] Yann LeCun1,2, Yoshua Bengio3 & Geoffrey Hinton, "Deep learning Review", *Macmillan Publishers Limited*, vol. 521 , 2015, doi:10.1038/nature14539

[30] Yufeng Zhenga , Clifford Yangb , Alex Merkulov , "Breast Cancer Screening Using Convolutional Neural Network and Follow-up Digital Mammography," *Proceedings Volume 10669, Computational Imaging III; 1066905 (2018)*, May 2018 DOI: 10.1117/12.2304564.

[31] Wafaa Mahdi Salih, Ibraheem Nadher, and Ahmed Tariq, "Modification of Deep Learning Technique for Face Expressions and Body Postures Recognitions," *International Journal of Advanced Science and Technology* Vol. 29, No. 3s, 2020, pp. 313-320.

[32] T. Ozcan And A. Basturk, "Lip Reading Using Convolutional Neural Networks With And Without Pre-Trained Models," *Balkan Journal Of Electrical & Computer Engineering*, Vol. 7, No. 2, April 2019, Doi: 10.17694/Bajece.479891

[33] Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi, "Understanding of A Convolutional Neural Network," *Icet, Antalya, Turkey, IEEE*, 2017

[34] Sandro Skansi, "Convolutional Neural Networks," In: "*Introduction to Deep Learning From Logical Calculus to Artificial Intelligence*", Switzerland, Springer International Publishing AG, part of Springer Nature 2018, pp 121-134.

[35] Timea Bezdan, Nebojša Bačanin Džakula, "Convolutional Neural Network Layers And Architectures," *International Scientific Conference On Information Technology And Data Related Research , Data Science & Digital Broadcasting Systems*, January 2019, Doi: 10.15308/Sinteza-2019-445-451.

[36] Omer Jomah, Amamer Khalil Masoud, Xavier Patrick Kishore and Sagaya Aurelia, "Microlearning: A Modernized Education System," *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* ,Volume 7, Issue 1, March 2016.

[37] Didik Dwi Prasetya, Aji Prasetya Wibawa, Tsukasa Hirashima, "The performance of text similarity algorithms," *International Journal of Advances in Intelligent Informatics* ISSN 2442-6571 Vol. 4, No. 1, March 2018, pp. 63-69.

[38] M.K.Vijaymeena, K.Kavitha, "A Survey On Similarity Measures In Text Mining," Machine Learning and Applications: *An International Journal (MLAIJ)* Vol. 3, No. 1, March 2016.

[39] Komal Maher, Madhuri S. Joshi , "Effectiveness of Different Similarity Measures for Text Classification and Clustering," *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 7 (4) , pp. 1715-1720, 2016.

[40] Costin A. Boiangiu, Mihai Zaharescu, Oana Ferche, and Andrei Danescu, "Automatic Correction of OCR Results Using Similarity Detection for Words and Fonts," *INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS*, Volume 10, 2016.

[41] Rishin Haldar and Debajyoti Mukhopadhyay , "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach," *ArXiv*,2011, doi.org/10.48550/arXiv.1101.1232

[42] B. Berger, M. S. Waterman and Y. W. Yu, "Levenshtein Distance, Sequence Comparison and Biological Database Search," in *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3287-3294, June 2021, doi: 10.1109/TIT.2020.2996543.

[43] Muhammad Maulana Yulianto, Riza Arifudin, Alamsyah, "Autocomplete and Spell Checking Levenshtein Distance Algorithm to Getting Text Suggest Error Data Searching in Library," *Scientific Journal of Informatics* Vol. 5, No. 1, May 2018.

[44] S. Zhang, Y. Hu and G. Bian, "Research on string similarity algorithm based on Levenshtein Distance," 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 2247-2251, 2017. doi: 10.1109/IAEAC.2017.8054419.

[45] Nada Hussain Ali , Matheel E. Abdulmunim, Akbas Ezaldeen Ali, "Constructed model for micro-content recognition in lip reading based deep learning," *Bulletin of Electrical Engineering and Informatics,* Vol. 10, No. 5, pp. 2557-2565, 2021, DOI: 10.11591/eei.v10i5.2927.

[46] Sebastian Ruder, "An overview of gradient descent optimization algorithms", *arXiv:1609.04747v2 [cs.LG]* 15 Jun 2017 https://arxiv.org/pdf/1609.04747.