Arabic Sign Language Recognition using Lightweight CNN-based Architecture

Batool Yahya AlKhuraym, Mohamed Maher Ben Ismail, Ouiem Bchir
College of Computer and Information Sciences
King Saud University, Riyadh, KSA

***Abstract*—Communication is a critical skill for humans. People who have been deprived from communicating through words like the rest of humans, usually use sign language. For sign language, the main signs features are the handshape, the location, the movement, the orientation and the non-manual component. The vast spread of mobile phones presents an opportunity for hearing-disabled people to engage more into their communities. Designing and implementing a novel Arabic Sign Language (ArSL) recognition system would significantly affect their quality of life. Deep learning models are usually heavy for mobile phones. The more layers a neural network has, the heavier it is. However, typical deep neural network necessitates a large number of layers to attain adequate classification performance. This project aims at addressing the Arabic Sign Language recognition problem and ensuring a trade-off between optimizing the classification performance and scaling down the architecture of the deep network to reduce the computational cost. Specifically, we adapted Efficient Network (EfficientNet) models and generated lightweight deep learning models to classify Arabic Sign Language gestures. Furthermore, a real dataset collected by many different signers to perform hand gestures for thirty different Arabic alphabets. Then, an appropriate performance metrics used in order to assess the classification outcomes obtained by the proposed lightweight models. Besides, preprocessing and data augmentation techniques were investigated to enhance the models generalization. The best results were obtained using the EfficientNet-Lite 0 architecture and the Label smooth as loss function. Our model achieved 94% and proved to be effective against background variations.**

Keywords*—*Arabic sign language recognition; supervised learning; deep learning; efficient lightweight network based convolutional neural network

I. Introduction

Communicating with others is an important skill for humans to interact with their environment community. In its absence, exchanging experience and expressing its opinion and feelings become a very challenging task. In fact, communication allows the discussion of different types of issues that concern humans in order to address them and come up with appropriate solutions to facilitate their daily life. Besides, communication is a censorious factor for individual's mental health [1]. Actually, communication types can be categorized into four groups: (i) verbal, (ii) non-verbal, (iii) visual and (iv) written communication.

Hearing Impaired (HI) persons are actually affected by their disability, and cannot ensure regular communication with others. They typically use non-verbal gestures as a visual communication type that relies on hand movements and facial expressions [2]. On the other hand, deaf people use sign language as their primary communication technique, where sign language is based on visual-motion codes. This codified system is defined using standard positions and hand movements supported by facial expressions [3]. Typically, hearing impaired people use hand gestures, called signs, to interact with others. Commonly, for existing sign languages, the main signs features are the handshape, the location, the movement, the orientation and the non-manual component. Thus, sign language recognition system assists people with hearing disabilities to communicate with healthy persons. Particularly, such recognition systems are meant to deliver the semantic corresponding to hand gestures of sign language to the interlocutor.

Sign language is considered as a descriptive language which it composed of hand gestures and facial expression. Arabic Sign Language (ArSL) used in approximately 22 Arab countries with different gestures. The discrepancy noted for some word gestures can be attributed to the cultural diversity between these countries. Despite this lack of uniformity of ArSL, all of the 22 countries use the same gestures for the Arabic letters and numbers [4]. Fig. 1 shows the standard Arabic sign language that expresses the Arabic alphabets.

Next sections are mention and discover the main types of ArSLs recognition systems, first, image-based ArSL recognition, can be categorized into three main groups: (i) continuous, (ii) isolated word, and (iii) alphabet recognition systems. Typically, those systems contain five major stages, where each single stage is meant to achieve a particular task. Namely, those five stages are: acquisition of images, preprocessing the images, then extraction the features from that images, after that the images segmentation, and lastly classification process.

Second, alphabet recognition, signers usually perform letters signs independently in the context of alphabet recognition scenario. To represent letters, static poses are used, and the size of vocabulary is minimal. Although the Arabic alphabet includes 28 letters, ArSL considers 39 signs [5]. Actually, the 11 added signs reflect simple signs that combine two letters. Third, word sign recognition, its techniques are relying on further analysis of the images. These techniques are more practical than alphabet recognition, but the alphabet recognition is less complex than word recognition. The major purpose of using word sign recognition is process various images in a sequence.

Fig. 1. Unified Alphabets for Arabic Sign Language.

Deep learning is introduced in Artificial Intelligence (AI) as a sub-group of machine learning [6]. For image recognition issues, deep learning algorithms have provided efficient solutions. In deep learning, Convolutional neural network (CNN) is considered as a class of deep neural networks commonly exploited in computer vision. The optimization of CNN networks can be introduced as a set of algorithms or methods used to alter the neural network attributes like weights, learning rate for reducing the loss costs, and provide an accurate result. In particular, Adam optimizer is an optimization algorithm for stochastic gradient descent used to train the models of deep learning [7]. In 2019, Google introduced new family of CNNs named EfficientNet [8]. EfficientNet proved to have lower number of parameters and Floating-Point Operations Per Second (FLOPS) and yield competitive accuracy [8].

In this project, we propose an image-based ArSL recognition system that relies on lightweight Efficient Network (EfficientNet) [8] to enhance the overall ArSL recognition performance and reduce the number of parameters and hence the time complexity. Standard performance measures and real datasets will be considered to evaluate the performance of the proposed system.

II. Related Work

ArSL recognition is even more challenging than ASL due to the considerable similarity between some of its letters signs. Recently, several approaches have emerged to address issues relevant to ArSl recognition. Particularly, researchers have investigated glove-based and glove-free approaches for ArSL classification. Both approaches rely on machine learning techniques but to different extent. In this situation, some models employ deep learning techniques, while others deploy shallow classification models. This literature surveys aims at giving an overview on state-of-the-art deep-learning and shallow-model based ArSl recognition approaches.

A. Shallow Model based Approaches

An ArSL recognition system was proposed in [9]. Specifically, a framework based Scale-Invariant Features Transform (SIFT) features was designed to extract the visual properties of ArSL gestures. In fact, SIFT extraction algorithm encloses five main steps. First, it requires rolling together with the Gaussian filter of different widths with the image. It is essential as it can help demonstrate the Gaussian function pyramid's difference. The extrema of the Gaussian pyramids are identified by comparing each point to its neighbors. The removal of main points at the extremes that are thought to be susceptible to noise are situated on edge. The next important thing is to determine the orientation. Next, in order to achieve that, it creates a histogram. It can be created from the sample points' gradient orientations. Finally, for the purpose of a local image region, a descriptor is produced. Afterward, a Linear Discriminant Analysis (LDA) was conducted to achieve dimensionality reduction. The researchers fed the resulting feature to classifiers: K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). The test results revealed that SVM outperformed KNN with a 98.9% accuracy. In [10], the authors introduced a similar approach to the one outlined in [9]. Although, their approach used LDA alongside Gaussian Mixture Model to fit the dataset. The experiments relied on a dataset collected using two dual Leap Motion Controllers (LMC). Two native signers were involved in the collection of 100 Arabic signs. The proposed framework achieved an accuracy of 92%, which is relatively higher than for similar sensor-based techniques.

In [11], a system for automated ArSL recognition using visual descriptors was introduced. The system was designed to generate a large number of visual descriptors in order to provide an effective ArSL alphabet recognizer. In particular, a one-versus-All SVM was used to classify the generated visual descriptors. According to their findings, the Histograms of the Oriented Gradients (HOG) descriptor significantly outperform the other descriptors. In addition to this, as far as the ArSL recognition is concerned, the method that was brought forward attained 90.55% of recognition accuracy.

Other researches were conducted to develop ArSL recognition schemes based on the nearest neighbour classification. Specially, in [12], a lexicon of 80 words was considered and a sensor-based dataset was collected. This dataset was then used to assess the performance of the glove-based ArSL recognition system. Following the data labeling stage, a low-complexity preprocessing and feature extraction techniques were deployed in order to capture the data temporal dependency. Additionally, a Modified K-Nearest Neighbor (MKNN) was utilized to classify the outputs. The system achieved a sentence recognition rate of approximately 98.9%. The approach was considered superior to vision-based approaches pertinent to classification rates. Similarly, in [13], a KNN algorithm was also used in the design of an ArSL algorithm. With regards to the classification rates, it was witnessed that the method outperforms the methods that were

vision-based. In an ArSL algorithm design, the use of the KNN algorithm was done additionally. There were various things that were included in the system. These things included histograms and their comparisons and generation, image masking, narrowing and clipping of images, and recording and questioning of images. A test for the glove recognition and hits was done. The test was done on the established algorithm, and ultimately, around 90% of the hit rate was attained by most of the characters. As the algorithm did not possess an engine, which is a self-learning one, it was regarded as imperfect and faulty. The researchers in [14] suggested a method for detecting hands and faces based on skin profiles using input images translated to YCbCr color space. As image transformation, morphological dilation operation was also used. The Prewitt operator was used to detect hand form edges, and the Principal Component Analysis (PCA) was applied to achieve dimensionality reduction and obtain the final feature. The experiment showed a 97% accuracy on KNN based on 150 signs and gestures.

A Gray Level Co-occurrence Matrix (GLCM) feature was used to implement an ArSL transformation system suggested in [15]. Despite the fact that the system was not deep learning-based, it had four phases: processing, feature extraction, matching technique, and display translation. A mixture of 15 GLCM and histogram features is used to remove features. The system correctly recognized nineteen Arabic alphabets with a 73% accuracy rate, according to the test results. Existing systems used Hidden Markov Models (HMMs) to identify the ArSL in various ways and techniques. HMMs help developers create signer-independent systems that do not need gloves or attached sensors. The authors in [16] described an HMM-based automated ArSL recognition system. The main three steps of keeping the tabs on and detecting the hand involved tracking the fingerprints, detecting the edge, and identifying the skin. In order to reduce error rate and improve the detection of edge significantly, a certain type of algorithm was utilized. That algorithm is referred to as a Canny algorithm [17] where this algorithm acts like an optimal edge detector. For all of the frames, it was scrutinized that whether the outline of the image was rounded or not. Along with that, for the purpose of assessing observed regions of skin, an integrated component analysis was also deployed. Although the experiment yielded 82.22% detection rate, the device only used eight features per frame. As a result, the system's ease of use relies heavily on fewer features. In [18], several spatiotemporal feature extraction techniques that could be used to recognize isolated ArSL both offline and online were introduced. The researchers used forward, backward, and bi-directional projections to extract video-based movements. Now, to make certain that in order to delineate the video series, little amount of coefficients was used, and to terminate the terrestrial reliance, all of the errors were gathered and assembled into one particular image. The model that was proposed encompassed filtering such as a low pass filtering, Radon transformation, and Zonal coding. Along with these, it also includes a 2-D transformation. The experimental findings showed that in the identification of ArSL, output accuracy ranged from 97% to 100%. Different techniques involved various pattern recognitions and feature extraction techniques such as PCA, Local Binary Patterns (LBP), and the HMM were used in [19]. Hands and heads were detected, attached components were labelled, and features were extracted accordingly. The experiments were based on 23 isolated Arabic words performed by three different signers. If we talk about the experiment done initially, the rate of the system recognition was computed in the absence of skin segmentation. Moreover, the rate of recognition with the features such as PCA-only, accompanied by the 30 eigenvectors, reached approximately 99.8%. The combination of LBP and PCA features with HMM led to the achievement of 99.9% recognition rate with 20 eigenvectors. LBP was applied in the description of the shape and texture of images, while PCA was used in the reduction of dimensionality. In [20], the researchers presented a method for creating a smart glove that can recognize ArSL. They did not report the accuracy of the simulation device, even though it was found to be cost-effective. In [21], the implementation of a mobile-based system was outlines. Since it contains several levels, the system tends to be compatible with a self-supervising system for deep learning. The results include details about the system's compatibility but no indication of its accuracy. The method described in [22] can also be thought of as a non-deep learning approach. Because the data was not reliable, the system relied on blob tracking using Euclidean distance. According to the report, the system had a 97% recognition rate based on a dataset of 30 different words.

B. Deep Learning based Approaches

In [23], a vision-based system for recognition for the Arabic sign language was proposed. It relies on CNN architecture to translate gesture images into Arabic speech in a supervised manner. This system detects hand sign alphabet and speaks out the corresponding Arabic letter using deep learning model. The feature extraction, considered as the first essential component of the system was followed by the classification component. During the feature extraction phase, the system transfers the input images into a three-dimensional (3D) matrix for depth, width and height specification. Then, the pooling layer reduces the size by decreasing the parameters number. After that, the classification is achieved through the fully connected layer. The reported performance attained 90% for the accuracy. Another Arabic sign language recognition system based on fine-tuning deep CNN was proposed in [24]. The system was evaluated using a set of 32 hand gestures categories. The system adapted Residual Network (ResNet-152) [25] and Visual Geometry Group (VGG-16) [26] as typical deep learning architectures, but with soft-max layer classification after the fully connected layer to improve the prediction performance. The input images fed into the resulting networks are two-dimensional (2D) images unlike in [23]. The reported accuracy was nearly to 99%, which considered as a very high accuracy. In [27], the researchers proposed an ArSL recognition system based on LeCun Network (LeNet-5) [28]. The considered network is composed of seven layers. The first four layers are responsible for the feature extraction process from the image. The last three layers are responsible for the classification task to categorize the input images into their corresponding meaning. The dataset used in that work contains 7869 Arabic sign language letters and numbers. The testing phase of the system has been done using 80% of dataset as a training set. An accuracy of 90.0% was achieved by their model. In [29], a different ArSL recognition system based on

deep CNN was depicted. The system aims at reducing the number of parameters in order to extract and detect the hand gestures. A collection of 50000 images including signs performed by a population of signers from various age groups. The system achieved an accuracy of 97% as best performance reported in their experiments. In [30], the authors investigated different transformation techniques for features extraction. The extracted features were then conveyed to the classification model for the prediction the Arabic sign. This study has compared three different classification methods. Namely, they used SVM, KNN and Multilayer Perceptron (MLP) for classification. The association of MLP with Hartley [30] transformation technique achieved nearly to 99% accuracy as recognition rate. Different approaches, such as image-based and sensor-based, were introduced in [31]. The proposed system for Arabic sign language recognition deepened on using a very common technique called Leap Motion Controller (LMC) [32], used as a backbone for this system. The LMC device aimed to detect the hand in order to provide the hand location and motion to facilitate the feature extraction process to classify the Arabic letters correctly. The system achieved a higher performance rate compared to Naive Bayes classifier with 98%, and 99% compared to the MLP neural networks. These results were obtained using 2800 images of Arabic sign language letters. Coupled with the findings in [30], the results show that MLP can be used successfully with LMC in the development of ArSL systems. Likewise, the researchers in [33] proposed an ArSL recognition system based on MLP neural networks for digital-sensor. The system achieved approximately 88% accuracy rate by recognizing 50 different dynamic markers represented by two signers. In [34], the authors developed an Arabic Sign Language Alphabet Translator (ArSLAT) system. Their system includes five stages: pre-processing, frame determination, category, features extraction, and finally the classification stage; for features extraction using translation scale and rotation invariant to achieve the flexibility of system; applied comparison between accuracy of system by using MLP and Minimum Distance Classifier (MDC) [35]. In addition, the result illustrated that system with MDC has an accuracy with 91%, while the MLP with system has 83%. In [36], a model was proposed to recognize the hand gestures of ArSL letters using supervised learning techniques and natural user interface libraries with the LMC skeleton [37] and Kinect [38]. In order to predict the hand captured from an input images that fed to the system in 3D manner as the system introduced in [30]. The results showed that their system cab recognize 22 out of the 28 alphabets with 100% accuracy.

In [39], the researchers developed an ArSL recognition system based on Recurrent Neural Networks (RNN). This system relies on four different phases: acquisition, processing, feature extraction, and recognition of image. The processing of the image requires a colour system called Hue, Saturation, and Intensity (HSI) to extract the features associating with colour layers. As result, this system accomplished 95% accuracy in the recognition of 28 ArSL letters. In [40], an Adaptive Neuro-Fuzzy Inference System (ANFIS) [41] was used for ArSL recognition. The system is based on gestures that represent the Arabic alphabets. Designing the system of many ANFIS networks in order to training on one gesture per network. Pre-processing the hand gesture to extract the features for classification process. The result of this system attained 93.55% accuracy in detecting 30 Arabic letters. On the other hand, a deep learning framework for isolated Arabic Sign Language gestures recognition was proposed in [42]. The purpose of that framework is to encounter three different challenges faced by different ArSL recognition systems. The challenges are: hand segmentation, extract features, and recognition of gesture sequence. To handle these challenges, the authors proposed three different networks. Namely, they used DeepLabv3+ for segmentation process, Convolutional self-organizing map for features extraction process, and lastly the Bi-directional long short-term memory for classification process. Finally, for testing, they used 3450 images that expressed 23 isolated words from three signers and accomplished approximately 89.5% accuracy.

In conclusion of this section, one can notice that existing related works can be categorized into two main different groups: (i) Approaches for ArSL recognition systems based on shallow machine learning models, (ii) deep learning based approaches. Both categories include different classification techniques and methods to recognize alphabet, numbers, words and sentences of Arabic Sign Language. In this project, we achieved our aim by proposing a lightweight based CNN-architecture to enhance the computational and classification efficiency of the ArSL recognition system.

III. Proposed Method

We proposed an Arabic sign language (ArSL) recognition system based on lightweight EfficientNet CNN [43]. The proposed system is intended to translate automatically hand gesture images into the corresponding Arabic sign language letter. ArSL recognition problem is tackled as a supervised deep learning using a lightweight EfficientNet network. This choice of deep architecture aims at achieving a good trade-off between the ArSL recognition rate and the model complexity. In other words, the proposed research seeks a lightweight deep learning model to ensure high ArSL recognition accuracy. In particular, we adapted two EfficientNet models learned from the baseline model, EfficientNet-B0 [44]. Specifically, we scaled it down in terms of number of channels, depth and resolution. In other words, the CNN-based EfficientNet is the backbone structure of our proposed architecture.

The classification task is performed using fully connected (FC) layers where an FC layer hosts neurons which manage comprehensive connections. These connections determine a previous layer's activation. Also, when specific input and output are present, their representational mapping is aided by the FC layer. This layer borrows the regular neural network principles to execute its assigned functionalities. However, this FC layer works with one-dimensional data. The three-dimensional to one-dimensional data transformation would requires the use of a flatten function for the proposed system to be implemented successfully. Subsequently the output will be the Arabic letter that associated with the input image of Arabic sign language.

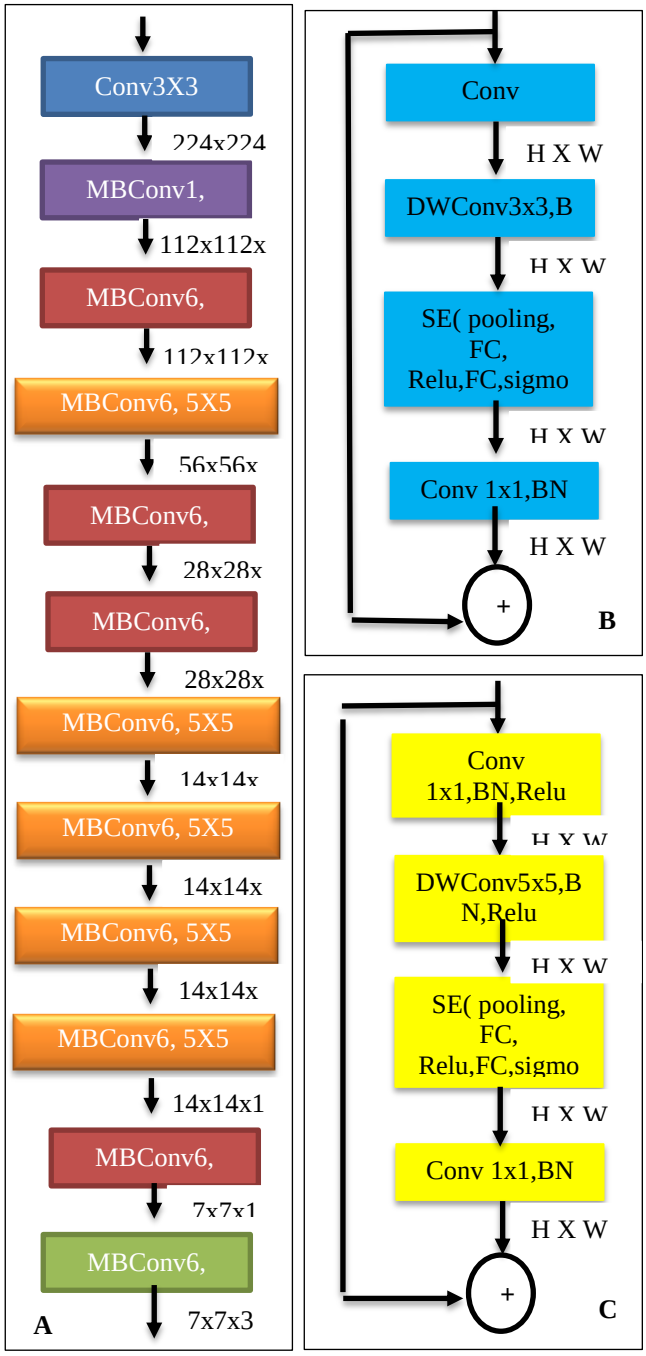


Fig. 2. Model Architecture (A) A Lightweight EfficientNet, (B) MBConv6 (k 3x3) and (C) MBConv6 (k 5x5).

One should note that since the input size tends to be bigger than the feature map size, the implementation of padding becomes the ideal solution to overrule the feature map shrinkage. It helps match the input with stride and kernel size. The resulting system state would yield an improved performance. On the other hand, the natural position of the pooling layer is between convolution layers. It serves the objective of reducing dimensionality and lessening the computations involved. The successful implementation of the pooling layer is due to the fewer number of parameters in play. The proposed pooling is max pooling. It scans all windows and takes the highest value resulting in a reduced feature map size.

The proposed model uses inverted residual block called (MBConv) as the main building block of the architecture. In fact, MBConv is a mobile inverted bottleneck. It encloses a squeeze-and-excitation optimization [45] element. This mobile inverted bottleneck can be perceived as an inverted residual block that includes a 1×1 convolution layer with batch norm and Rectified Linear Unit (Relu). It is also followed by a 3×3 or 5×5 depth-wise convolution with batch norm and Relu. Besides, a pooling-based Squeeze-and-Excitation (SE) optimization block is added. Next, FC layer and a Relu followed by FC layer and a sigmoid are stacked. It is worth noting that the scaling (i.e., Multiplication Operator (MUL)) aims at multiplying each channel by the input feature to obtain the final output of the SE Block. Finally, a 1×1 convolution layer with batch norm is inserted. Note that the inverted residual block exhibit skips connection between the layers. Fig. 2 details the proposed network architecture. Besides, the loss function is the one that used to compute the difference between the logit of the real class and the logit of the output class. Since our task is a multi-class problem, multi-class cross entropy is generally used in such problem. However, a "label smoothing" technique was introduced for robust modeling [46], and it has been used in various state-of-the art-models ranging from image classification to language translation and speech recognition in an attempt to improve accuracy by using a weighted mixture of targets from the dataset with uniform distribution instead of the hard targets to compute cross entropy.

A. Dataset

We collected real ArSL images to be used in the planned experiments. The collection images were captured using diverse smartphones. More than twenty volunteers participated in the collection of this dataset. Each volunteer performed thirty gestures corresponding to ArSL alphabet. Each character from this alphabet is represented using ten instances. This yields a total of 5400 images. Fig. 3 shows sample of our dataset for the Arabic latter "Baa". As one can see, those images exhibit different visual properties of their background. This choice is meant to make the ArSL recognition task even more challenging. In other words, this would prove the ability of the proposed model to capture the most relevant visual features in order to discriminate better between the different ArSL alphabet classes.

B. Training Framework

The network will be trained like any usual CNN, by feeding the training input and output are fed to the model. Then, the model will keep iterating the training process through forward propagation and backpropagation. However, using EfficientNet has some requirements to use the pre-trained layers' weights and continue the training process on the added layers. Where the size of RGB-images of sign gestures will be reduced to 224 x 224 to enter the network in training phase. Then, the CNN-based feature extraction processes the captured imaged for feature maps detection. After that, these maps enter the Lightweight EfficientNet to produce potential hand gestures of the input. Finally, the scaled feature maps are

conveyed to FC layer classifier to generate the ArSL alphabet prediction. Then the model, using the output, will backpropagate and adjust the weights.

Fig. 3. Sample Images of "Baa" Letter from our Collected ArSL Data.

IV. Result

In order to evaluate the proposed models, we conducted four experimental scenarios. In the first one, we investigated the effect of segmentation on the system performance. Specifically, we fed the designed model with original and segmented (just the rectangle of hand was used as input) images. In this scenario Adam algorithm [7] was used to optimize the loss function. In the second scenario, we analyzed the system behavior when a Cross Entropy function is used as loss function in addition to Cross Entropy Label Smoothing. Same as in the first scenario, Adam optimizer was used here too. The third scenario was dedicated to the application of different versions of EfficientNet Lite model to see which model yields the best performance. As previous, Adam optimizer was used. The fourth scenario was designed to analyze the system behavior when four optimizers were applied in addition to Adam. In this scenario, to speed up the experiences, the lightest model (EfficientNet Lite 0) was the base model and all images were segmented.

One should note that the dataset described in Section III. proposed method is split into three subsets: (i) A training set including 4320 (80%) of images, (ii) A test set that includes 540 (10%) of the total number of images, and (iii) A validation set that represents 540 (10%) of the total number of images. For deep learning, the best practice is to use train/test/validation split rather than cross validation. All experiments were conducted using Nvidia K-80 GPUs associated with 16 GBs of RAM. The EfficientNet Lite network as well as the training, validation and the test procedures were implemented using Pytorch library [47]. Moreover, five pre-trained models published in [48] were exploited in our experiments.

Table I summarizes the three first scenarios settings. Actually, there are 9 EfficientNet Lite versions; however, the pre-trained weights are available only for the first five models from Lite 0 to 4. We decided that reporting the results of the lighter and heavier model could be concise and more informative. Where Table II reports the performance measures obtained using these models and the test data outlined earlier, *Model 4* (the lightest version with segmented data and label smooth loss function) outperformed the other models, and achieved an accuracy of 94.30%.

A. Impact of Segmentation

The lightest models, Model 2 and Model 4, attained an accuracy of 94.30% and 84.54%, respectively. This shows the impact of segmentation of the model performance. This considerable difference in performance is expected. In fact, the segmentation eliminates the undesirable image parts and artifacts and allows the models focus and learn from the relevant content and discard the irrelevant regions. On the other hand, Fig. 4 shows the evolution of the accuracy during the training and validation phase. As one can see, the gap between the training and the validation curves is larger in the case of unsegmented data.

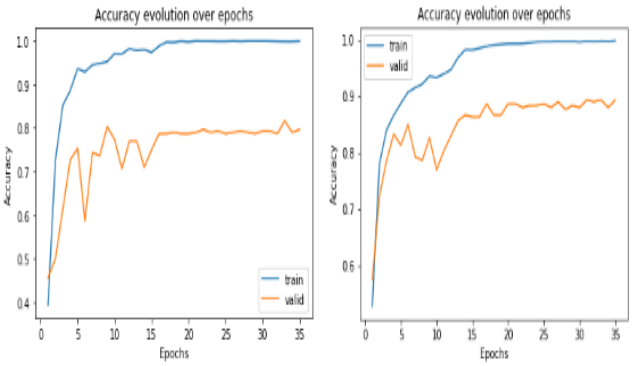
TABLE I. Model Settings.

Model name	w / Segm	w/o Segm	CrossEntropy	CrossEntropyLabel Smooth
EfficientNet_Lite 0 **Model 1**		✓	✓	
EfficientNet_Lite 0 **Model 2**		✓		✓
EfficientNet_Lite 0 **Model 3**	✓		✓	
EfficientNet_Lite 0 **Model 4**	✓			✓
EfficientNet_Lite 4 **Model 5**		✓	✓	
EfficientNet_Lite 4 **Model 6**		✓		✓
EfficientNet_Lite 4 **Model 7**	✓		✓	
EfficientNet_Lite 4 **Model 8**	✓			✓

TABLE II. Summary of Results Obtained by Model Evaluation on Testing Data

Model name	Accuracy (%)	F-measure (%)	GFlops	# params
EfficientNet_Lite 0 M1	79.73	80.21	0.4	3.41M
EfficientNet_Lite 0 M2	84.54	84.80	0.4	3.41M
EfficientNet_Lite 0 M3	87.83	87.53	0.4	3.41M
EfficientNet_Lite 0 M4	**94.30**	**94.24**	0.4	3.41M
EfficientNet_Lite 4 M5	88.60	88.83	2.64	11.76 M
EfficientNet_Lite 4 M6	87.26	87.36	2.64	11.76 M
EfficientNet_Lite 4 M7	91.95	91.96	2.64	11.76 M
EfficientNet_Lite 4 M8	91.28	91.18	2.64	11.76 M

params: Number of model parameters, *GFliops*: Floating operation per seconds.

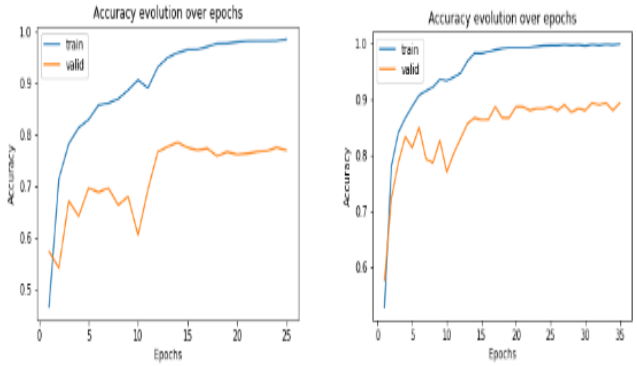


(a) (b)

Fig. 4. Accuracy over Epochs, (a) Model 2: Unsegmented Images were used. (b) Model 4: Segmented Images were used.

B. Impact of Label Smooth on Loss Function

The following results were obtained by evaluating the lightest pair of models, Model 3 and Model 4 in Table II. They illustrate the impact of the Label Smooth on the Cross Entropy loss function. An increase of 5% in accuracy can be noticed for the models using the Label Smooth. In fact, the Label smooth is typically used as a regularization technique that improves the model generalization. Particularly, Fig. 5 shows a larger gap between train and valid curves when no label smooth is used. This proves that the model over fits the training data before using this regularization technique.



(a) (b)

Fig. 5. Impact of Label Smooth Regularization: (a) Model M3: without Label Smooth3. (b) Model M4: with Label Smooth.

C. Impact of Optimizers

In these experiments, we tried to investigate five different optimizers known as Adam, RMSprop, AdamW, Adadelta and SGM optimizers. In order to figure out the best optimizer among them that had a good effect on the Model 4 to produce a good performance rate for recognition process of ArSL. Table III summarizes the results obtained from the fourth scenario when changing the optimizer in the training loop.

In fact, Model 4, the lightest version with optimal settings, is used for evaluation in this experiment. Namely, the optimizers used are Adam [7], RMSprop, AdamW, Adadelta and SGD algorithms. In RMSprop, stochastic gradient is applied with mini-batch and adaptive learning rates. Instead of accumulating prior gradient values, Adadelta adapts the learning rate using a moving window of gradient updates. Adam combines the good proprieties of RMSpro and Adadelta. It uses Momentum and adaptative learning rate, in other words, the learning rate is gradually adjusted over time. It remains the most prevalent optimizer used in deep learning. Similarly, AdamW optimizer is a variation of Adam optimizer in which the optimization is performed for the weight decay and learning rate separately. Under particular circumstances, it is assumed to have a faster convergence rate than Adam. SGD is an optimizer that updates the weights for each training sample over a limited-size subset of data. As expected, Adam outperforms all others.

TABLE III. MODEL PERFORMANCE ON TESTING DATA USING DIFFERENT OPTIMIZERS

Model 4		
Optimizer	**Accuracy**	**Loss**
Adam	**94.30%**	**0.054432**
RMSprop	93.29%	0.055978
AdamW	92.62%	0.055098
Adadelta	89.94%	0060984
SGD	86.93%	0.070220

D. Impact of Data Augmentation

We noticed in the experiment scenarios, the training and validation images are augmented using random horizontal flip transformation. Table IV shows the performance achieved with and without data augmentation. As it can be seen, the flip transformation increases the Accuracy and F-measure by more than 3%.

TABLE IV. IMPACT OF DATA AUGMENTATION

Settings: Model 4		
	Accuracy	**F-measure**
With Horizental Flip transformation	**94.30%**	**94.24%**
Without Horizental Flip transformation	91.16%	90.98%

E. Result of EfficientNet_Lite 0 Model 4 on Arabic Sign Language

In the following, we report the performance measures recorded for letters obtained by Model 4. Specifically, accuracy, the precision, recall and F-1 measure in Table V. Most letters are recognized correctly despite the similarities in some gestures like (“shin: ش ” ,“sin: س ”) , (“dhad: ض ” ,“sad: ص ”), (“thal: ذ ” ,“dal: د ”), and (“ray: ر ” ,“zay: ز ”) as depicted in Fig. 1 above. Further analysis showed that the 10 instances of each letter used in test are from the same signer. However, very few instances within one letter were not recognized. This is due to the intra-class variation or the inter-class similarities in gestures mentioned above. The least correctly classified letters are “Ra ر ” (70%) and “Ain ع ” (70%), where extra data could help to improve recognizing those two letters. On the other hand, there are 13 letters out of 30 with a 100% precision, where the rest of the alphabets did not have low precision except two letters with 70% as discussed before; applied different measures to assess Model 4 in order to provide an overview of its functionality and performance.

TABLE V. PERFORMANCE MEASURES OBTAINED USING MODEL 4

	Accuracy	Precision	Recall	F1
Model 4	94.30%	94.3%	94.46%	94.13%

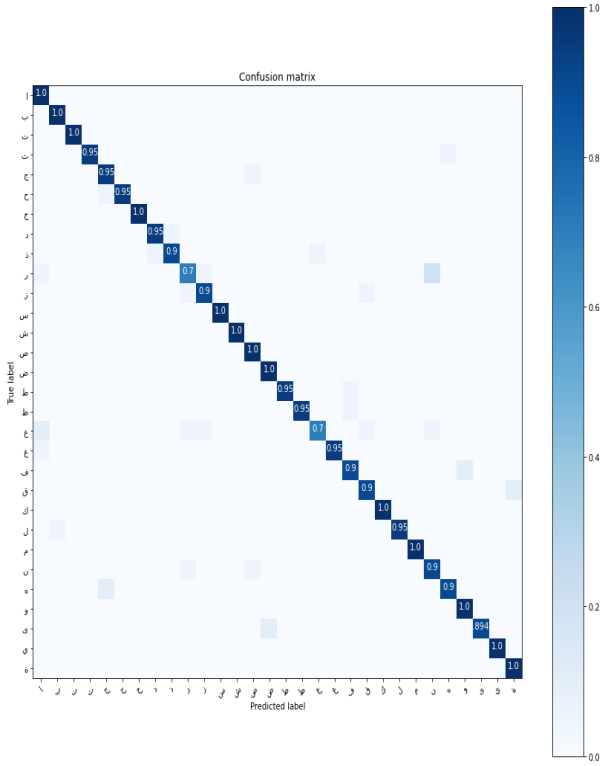


Fig. 6. Confusion Matrix obtained using Model 4.

Fig. 6 illustrates the confusion matrix obtained by the evaluation of the Model 4 by accuracy measure. Where it expresses a visual chart related to the predicted label of letters produced by Model 4 and the true label letters. As we can see here, two diagonal squares with a light color and an accuracy value of 0.7 corresponding to the letters (ra, 'ر') and (Ain 'ع'), respectively.

F. Comparison with other Existing Work

A comparison of the proposed recognition systems versus some of the most relevant work that cultivated the highest accuracy for ArSL recognition was conducted. We compared the results obtained by our best model, Model 4 (lightest version) to the work proposed in [48]. This existing work is a two Faster R-CNN technique based on VGG-16 and ResNET-18. As can be seen in Table VI, our model outperforms the existing works in term of accuracy by 1%. Furthermore, the size of our model is very small compared to the size of compared models.

TABLE VI. GLOBAL COMPARISON TABLE BETWEEN EXISTING WORK IN [48] AND THE PROPOSED METHOD

Model	**Accuracy**	**# of parameters**
Faster R-CNN(VGG-16)	93.2%	138M
Faster R-CNN(ResNET18)	93.4%	11M
EfficientNet_Lite 0 Model 4	94.30%	3.41M

V. CONCLUSION

Hearing-impaired people are suffering from the communication with others in an easy way since they have to learn the sign language, which considered as their formal language to interact with community. The overall performance of image-based solutions for this problem depends on the segmentation quality and the choice of the features that should encode the main visual properties of the sign language gesture. The existing solutions exhibit considerable rooms for improvement for the ArSL language recognition solutions which typically rely on heavyweight Convolutional Neural Network (CNN) models. The common hardware is insufficient for the existing models to deliver sufficient in real time in order to interpret the ArSL into Arabic spoken language. EfficientNet, introduced by Google, contains many CNNs layers that have high accuracy and also improve the efficiency of the models by reducing the number parameters and the computational cost. In this project, we proposed a system to recognize the Arabic sign language (ArSL) using a CNN based lightweight EfficientNet. Particularly, the proposed architecture contains two phases, feature extraction and classification in order to process the input image to produce the Arabic alphabet associated to that input. During this first phase of the project, we provided an overview on the background required for this research. Moreover, we conducted a literature review to survey existing relevant ArSL recognition system and approaches. We applied a real dataset to develop and deploy a system serving for the interpretation of letters of Arabic sign language. Then, standard performance measures adopted to assess the performance of the proposed system and compared its results with state-of-the-art approaches particularly VGG and ResNet.

As a future work, we propose to investigate Transformer for vision computer, a convolutional neural network (CNN) free deep learning architecture [49] based on self-attention mechanism. Where vision Transformer attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. Furthermore, we recommend extending this work to recognize the Arabic sign language words or common expressions.

ACKNOWLEDGMENT

This work was supported by the Research Center of the college of Computer and information Sciences at King Saud University, Riyadh, KSA. The authors are grateful for this support.

REFERENCES

[1] C. Martin, and N Chanda,"Mental Health Clinical Simulation: Therapeutic Communication" Int. J. of Clinical Simulation in Nursing, Vol.12, Issue 6, pages 209-214, Jun.2016.

[2] E. K. Elsayed, and D. R. Fathy, "Semantic Deep Learning to Translate Dynamic Sign Language", Int. J. of Intelligent Engineering and Systems, Jan. 2021.

[3] A. Horňakova, and A. Hudakova, "Effective communication with deaf patients", Int. J. of Original scientific article,VOL.4, NO.7, 2013.

[4] M. Mustafa, "A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers", J. of Ambient Intelligence and Humanized Computing 12, 4101–4115, Mar. 2020.

[5] M. Mohandes, "Arabic sign language recognition", In International conference of imaging science, systems, and technology, Las Vegas, Nevada, USA, vol. 1, 2001.

[6] X. Chen, L. Zhang, T. Liu, and M. M. Kamruzzaman, "Research on deep learning in the field of mechanical equipment fault diagnosis image quality", Journal of Visual Communication and Image Representation, Vol. 62, pages 402–409, Jul. 2019.

[7] D. Melinte and L. Vladareanu, "Facial Expressions Recognition for Human–Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer", MDPI Journal, Vol. 20, Issue. 13, Apr. 2020.

[8] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", In International Conference on Machine Learning, pages 6105-6114, May.2019.

[9] A. Tharwat, T. Gaber, A. Hassanien, M. Shahin, M and B. Refaat, "Sift-based Arabic sign language recognition system", In Afro-european conference for industrial advancement, Vol.334, pages 359-370, 2015.

[10] M. Deriche, S. Aliyu, and M. Mohandes, "An intelligent arabic sign language recognition system using a pair of LMCs with GMM based classification" IEEE Sensors Journal, Vol.19, No.18, pages 8067-8078, Sep. 2019.

[11] 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.

[12] N. Tubaiz, T. Shanableh, and K. Assaleh, "Glove-based continuous Arabic sign language recognition in user-dependent mode", IEEE Transactions on Human-Machine Systems, Vol.45, No.4, pages 526-533, Aug. 2015.

[13] R. Naoum, H. Owaied, and S. Joudeh, "Development of a new arabic sign language recognition using k-nearest neighbor algorithm", Journal of Emerging Trends in Computing and Information Science, Vol.3, No.8, pages 1173-1178, Aug. 2012.

[14] E. Hemayed and A. Hassanien, "Edge-based recognizer for Arabic sign language alphabet (ArS2V-Arabic sign to voice)". International Computer Engineering Conference (ICENCO), pages121-127, Dec. 2010.

[15] A. El Alfi and S. Atawy, "Intelligent Arabic sign language to Arabic text translation for easy deaf communication" International Journal of Computer Applications, Vol.180, 2018.

[16] A. Youssif, A. Aboutabl, and H. Ali, "Arabic Sign Language (ArSL) recognition system using hmm", International Journal of Advanced Computer Science and Applications, Vol.2, No.11, Nov. 2011.

[17] J. Ravikiran , M. Kavi , M. Suhas, R. Dheeraj, S. Sudheender, V. Nitin and Pujari ,"Finger Detection for Sign Language Recognition", Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol.I, Mar. 2009.

[18] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language", IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol.37, No.3, pages641-650, June 2007.

[19] A. Ahmed and S. Aly, "Appearance-based arabic sign language recognition using hidden markov models", International conference on engineering and technology (ICET), pages 1-6 , Apr. 2014.

[20] M. Sadek, M. Mikhael, and H. Mansour, "A new approach for designing a smart glove for Arabic Sign Language Recognition system based on the statistical analysis the Sign Language", In 34th National Radio Science Conference, pages 380-388, Mar. 2017.

[21] M. El-Gayyar, A. Ibrahim and M. Wahed, "Translation from Arabic speech to Arabic Sign Language based on cloud computing", Egyptian Informatics Journal, Vol.17, No.3, pages 295-303. Nov. 2016.

[22] N. Ibrahim, M. Selim, and H. Zayed, "An automatic arabic sign language recognition system (ArSLRS)", Journal of King Saud University-Computer and Information Sciences,Vol.30, Issue 4, pages 470-477, Oct. 2018.

[23] M. Kamruzzaman, "Arabic sign language recognition and generating Arabic speech using convolutional neural network", Wireless Communications and Mobile Computing, Vol.2020, Article ID 3685614, May. 2020.

[24] Y. Saleh and G. Issa, "Arabic Sign Language Recognition through Deep Neural Networks Fine-Tuning", International Journal of Online and Biomedical Engineering, Vol.16, No.5, pages 71-83, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image Recognition", arXiv Organization, Cornell university, arXiv:1409.1556, Apr. 2015.

[27] S. Hayani, M. Benaddy, O. El Meslouhi, and M. Kardouchi, "Arab Sign language recognition with convolutional neural networks", International Conference of Computer Science and Renewable Energies (ICCSRE), pages1- 4, Jul. 2019.

[28] Y. LeCun, L. Bottoux, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", In Proceedings of the IEEE, vol.86, no.11, pages 2278-2324, Nov.1998.

[29] 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, pages 715-724, Jul. 2020.

[30] H. Luqman and S. Mahmoud, "Transform-based Arabic sign language recognition", Procedia Computer Science, Vol.117, pages 2-9, 2017.

[31] M. Mohandes, S. Aliyu and M. Deriche, "Arabic sign language recognition using the leap motion controller", IEEE 23rd International Symposium on Industrial Electronics, pages 960-965, Jun. 2014.

[32] "Leap Motion Controller and Touchless Technology", Dartmouth Business Journal, [Online]. Available: http://dartmouthbusiness journal.com/2013/08/the-leap-motion-controller-and-touchless-technology. [Accessed: 15-Mar-2021].

[33] A. Elons, M. Ahmed, H. Shedid and M. Tolba, "Arabic sign language recognition using leap motion sensor" 9th International Conference on Computer Engineering & Systems (ICCES) , pages 368-373, Dec. 2014.

[34] N. El-Bendary, H. Zawbaa, M. Daoud, A. Hassanien and K. Nakamatsu, K. "Arslat: Arabic sign language alphabets translator", International conference on computer information systems and industrial management applications (CISIM), pages590-595, Oct. 2010.

[35] M. Packianather and P. Drake, "Comparison of Neural and Minimum Distance Classifiers in Wood Veneer Defect Identification", The Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Sage Publications, Vol.219, No.11, pages 831-841, Nov. 2005.

[36] M. Almasre and H. Al-Nuaim, "A real-time letter recognition model for arabic sign language using kinect and leap motion controller v2", International Journal of Advanced Engineering, Management and Science, Vol.2, No.5, pages 239469, May. 2016.

[37] M. Dominio and P. Zanuttigh, "Hand gesture recognition with jointly calibrated Leap Motion and depth sensor", Multimedia Tools and Applications journal, Vol. 75, pages 14991–15015, Nov. 2016.

[38] J. Han, L. Shao, D. Xu and J. Shotton, "Enhanced computer vision with Microsoft Kinect Sensor: A review", IEEE Transactions on Cybernetics, Vol.43, No.5, pages 1318-1334, Oct.2013.

[39] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) using recurrent neural networks", First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), Vol.4, No.1, pages478-481, Aug 2012.

[40] O. Al-Jarrah, and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems" Artificial Intelligence, Vol.133 No.1-2, pages 117-138, Dec. 2001.

[41] J. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", IEEE Transactions on Systems Man Cybernetics, Vol.23, No.3, pages 665–685, Jun.1993.

[42] S. Aly and W. Aly, "DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition", IEEE Access, Vol.8, pages 83199-83212, Apr. 2020.

[43] C.Wang, C. Chiu and J. Chang, "EfficientNet-eLite: Extremely Lightweight and Efficient CNN Models for Edge Devices by Network Candidate Search", arXiv Organization, Cornell University, Sep. 2020.

[44] T. Mingxing and V. Quoc,"EfficientNet: Rethinking model scaling for convolutional neural networks", In ICML, Vol.97, pages 6105–6114, 2019.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. “Rethinking the inception architecture for computer vision”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32 [Internet]. Curran Associates, Inc. pages 8024–35, 2019.

[47] RangiLyu, Pytorch implementation of Google's EfficientNet-lite. Provide imagenet pre-train models, source code [Source code]. https://github.com/RangiLyu/EfficientNet-Lite, 2020.

[48] R. Alawwad, O. Bachir and M. Ismail, Arabic Sign Language Recognition using Faster R-CNN, Proceeding of the International Journal of Advanced Computer Science and Application (IJACSA), Vol.12, No.3, 2021.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale", arXiv Organization, Cornell university, arXiv: arXiv:2010.11929v2, In ICLR, Jun.2021.