Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2023, Article ID 5195007, 11 pages
https://doi.org/10.1155/2023/5195007

Research Article

Image-Based Arabic Sign Language Recognition System Using Transfer Deep Learning Models

Qanita Bani Baker, Nour Alqudah, Tibra Alsmadi, and Rasha Awawdeh

Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan

Correspondence should be addressed to Qanita Bani Baker; qmbanibaker@just.edu.jo

Received 20 March 2023; Revised 24 October 2023; Accepted 15 November 2023; Published 6 December 2023

Academic Editor: Najlae Idrissi

Sign language is a unique communication tool helping to bridge the gap between people with hearing impairments and the general public. It holds paramount importance for various communities, as it allows individuals with hearing difficulties to communicate effectively. In sign languages, there are numerous signs, each characterized by differences in hand shapes, hand positions, motions, facial expressions, and body parts used to convey specific meanings. The complexity of visual sign language recognition poses a significant challenge in the computer vision research area. This study presents an Arabic Sign Language recognition (ArSL) system that utilizes convolutional neural networks (CNNs) and several transfer learning models to automatically and accurately identify Arabic Sign Language characters. The dataset used for this study comprises 54,049 images of ArSL letters. The results of this research indicate that InceptionV3 outperformed other pretrained models, achieving a remarkable 100% accuracy score and a 0.00 loss score without overfitting. These impressive performance measures highlight the distinct capabilities of InceptionV3 in recognizing Arabic characters and underscore its robustness against overfitting. This enhances its potential for future research in the field of Arabic Sign Language recognition.

1. Introduction

Sign language (SL) is a nonverbal and natural language with the same functions as spoken language [1]. Deaf and hard-of-hearing individuals use SL to interact with others through a vocabulary of signs and gestures [2]. In the past, people with disabilities did not receive global attention. However, today's technologies offer tools designed to enhance the quality of life for individuals with disabilities [3]. Recognizing Arabic Sign Language (ArSL) is a significant area of research due to its complex nature. Moreover, sign language recognition has become an essential application in deep learning and artificial intelligence [4]. In this study, we aim to develop an Arabic Sign Language Identification System (ArSL) using deep convolutional neural networks (CNNs) to assist deaf people with hearing problems. Sign language and spoken language have the same work roles [5]; it is used to deal with those who cannot speak or hear, as it depends on the language of the hands with specific movements [6]. The signs differ according to each letter of the alphabet and other movements to form sentences [7].

Recent advances in deep learning (DL) and computer vision have shown great promise in the fields of gesture recognition that significantly improve communication between individuals who use sign language and those who do not [8, 9]. Furthermore, hand shape features can be detected using many approaches such as using CNNs [10, 11] and histograms of orientation gradient feature extraction [12]. Sign language employs signals and body dialects such as hand shapes, facial expressions, and lip patterns to communicate meaning [13]. It consists of manual gestures represented by hand position, direction, form, and path—nonmanual gestures representing facial expressions and body movement [14]. However, most researchers focus on hand signals because they contain raw information [15]. There are two prime approaches to Sign Language Recognition (SLR) systems which are image-based and sensor-based. The first approach is based on the use of SLR images, movements, and marks in

the cameras' vision [16], while in the second approach, instead of adopting the cameras' basis, the sensors use fixed gloves to capture the marks with the probes [16].

This study develops an Arabic Sign Language Identification System (ArSL) using six different pretrained architectures with pretrained weights: MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, and Xception. Experimentally, to enhance the robustness and effectiveness of pretrained models, we employed early stopping [17] and data augmentation techniques. These practices are essential to facilitate better generalization of the model on unseen data. Striking the appropriate balance and iterating through experimental iterations are crucial steps to fine-tune the model and mitigate overfitting.

The sections of this paper are organized as follows. Section 2 provides a comprehensive overview of existing research in the field. Section 3 explains the aim of the study. Section 4 illustrates the materials and methods proposed in this research. Moreover, Section 5 presents various experiments and their results, while the final section, Section 6, provides the conclusion of this paper.

2. Related Work

Nowadays, the power of deep learning technologies is applied in the field of sign language to improve the quality of life for people with disabilities. Many works have been proposed to enhance the sign language recognition system in different languages using diverse techniques [8, 18]. Several surveys provide a comprehensive overview of sign language recognition systems utilizing deep learning [19]. The survey, in [20], has reviewed sign language recognition and ArSL. The survey encompassed an evaluation of various classifiers and their respective performances across different sign languages, ultimately reporting the most effective classifier tailored to each specific sign language used for optimal sign language recognition systems. In this section, we provide an overview of the most pertinent research related to the Arabic Sign Language recognition systems. Table 1 provides a summary of the prior research discussed in this study.

Saleh and Issa [24] proposed models that match the VGG16 and the ResNet152 structures and employed transfer learning and fine-tuning of deep convolutional neural networks (CNNs) to enhance the accuracy in recognizing 32 hand signs from Arabic Sign Language. The proposed method was applied to 2D images of diverse Arabic Sign Language data, achieving an impressive accuracy rate reaching a validation accuracy of 99.6% for the ResNet152 and 99.4% for the VGG16. ElBadawy et al. [32] employed a deep behavior-based feature extractor to capture the finer details in Arabic Sign Language effectively. A 3D convolutional neural network (CNN) was also utilized for the recognition of 25 gestures from the Arabic Sign Language dictionary. The recognition system was fed with data obtained from depth maps. The system demonstrated an accuracy rate of 98% for observed data and 85% for the new data.

In [31], Hayani et al. proposed a new approach based on convolutional neural networks and fed the applied approach with a real dataset. The approach is used to automatically recognize numbers and letters of Arabic Sign Language. Then, a comparative study was conducted to demonstrate the effectiveness and robustness of the proposed approach compared to traditional models, particularly, K-nearest neighbors (KNN) and support vector machine (SVM). The recognition rate for the proposed system is 90.02% surpassing both SVM at 88% and KNN at 66%. Kamruzzaman in [25] introduced a vision-based approach utilizing convolutional neural networks (CNNs) for the recognition of Arabic hand sign-based letters and translating them into spoken Arabic. The accuracy achieved by this approach equals 90%, which ensures that this system is demonstrated to be highly reliable and efficient. Almasre and Al-Nuaim [26] developed a dynamic prototype model (DPM) utilizing Kinect in order to recognize specific dynamic words in Arabic Sign Language (ArSL). In this work, the DPM integrated eleven predictive models employing three machine learning models (SVM, RF, and KNN) with varying parameter configurations. The results in this research demonstrated that the SVM models utilizing a linear kernel with a cost parameter of 0.035 performed the highest accuracy rates in recognizing the dynamic words.

Elatawy et al. [27] introduced a novel approach employing the neutrosophic technique [33] and fuzzy c-means for the detection and recognition of Arabic Sign Language alphabet. The system employed a Gaussian filter to eliminate noise and prepare the input image for further processing. Then, images were transformed into the neutrosophic domain, and the features were extracted to commence the classification stage. Experimental results showed the system's commendable performance, and it achieved a total classification accuracy of 91%. The study in [28] proposed a new framework for signer-independent sign language recognition, leveraging a combination of deep learning architectures. The proposed framework encompasses hand semantic segmentation, hand shape feature representation, and a deep recurrent neural network. The framework is evaluated on a challenging Arabic Sign Language database, encompassing 23 isolated words recorded from three different users. The experimental results demonstrated that the applied framework significantly outperforms other state-of-the-art methods in the context of signer-independent testing strategies with an accuracy of 89.5% using DeepLabv3+ semantic hashing of the hand.

Alnahhas et al. [29] introduced an approach for recognizing words in Arabic Sign Language utilizing the Leap Motion device. The device facilitates the creation of a 3D model of the human hand through infrared technology. The proposed methodology intends to analyze mathematical features derived from the Leap Motion controller. The gesture is also represented as a series of frames to reflect its temporal nature, using the LSTM layer-based neural network classifier to encode the sequence and find the matching gesture. The highest rating was 89% for one-handed gestures and 96% for hand gestures. The study [30] proposed an affordable smart glove system capable of recognizing hand gestures in Arabic Sign Language. The proposed approach integrated the flex sensors and a tilting sensing module for both the right and left hands. Additionally, an Android application called "Smart Glove" has also been developed to

Table 1: Summary of related work on ArSL recognition research.

Ref	Year	Technique	Accuracy	Language	Dataset	Dataset size
[21]	2023	CNN	98.8% top-1 accuracy	Arabic	ArSL2018	54,049 images
[22]	2022	CNN and RNN	Recognition accuracy of 94.46%	Arabic	ArSL2018	54,049 images
[23]	2022	Transfer learning (TL) and RNN	Recognition accuracy of 93.4%	Arabic	New collected data	2000 videos
[24]	2020	VGG16 and ResNet152	99.6% for ResNet152 and 99.4% for VGG	Arabic	ArSL2018	54,049 images
[20]	2020	CNN, MLP, HMM, RNN, KNN, LDA, SVM, ANN, and more	DL classifiers attained the best performance as compared to other classifiers	Arabic	—	—
[25]	2020	"Deep learning model"	90%	Arabic	Raw sign language images that are captured using a camera	3875 images
[26]	2020	"RF, SVM, and KNN"	Accuracy value of 83% by using SVM	Arabic	Collected from 10 people who understand sign and ArSL	Dataset of 222 observations
[27]	2020	"Neutrosophic technique and fuzzy c-means"	91%	Arabic	Dataset used was provided by "Al-Amal Institute Damietta for deaf students"	300 images
[28]	2020	"DeepLabv3C" based on ResNet50	89.5%	Arabic	ArSL database	23 isolated Arabic word signs performed by 3 different users
[29]	2020	LSTM layer-based NN	89% for one-handed gestures and 96% for hand gestures	Arabic	Common ArSl dataset	44 signs were used, 29 of them are one hand gestures, and 15 are two hand gestures
[30]	2020	Android application "Smart Gloves"	90%	Arabic	They rely on glove movement gestures	—
[31]	2019	CNN, KNN, and SVM	90.02%	Arabic	Collected from Ibn Zohr University	5839 images
[32]	2017	"3D CNN," RNN, KNN, LDA, SVM, ANN, and more	85%	Arabic	Source not found	200 videos

translate gestures into textual speech. The glove system was designed to accommodate both word level and sentence level and showed an impressive 90% recognition rate. The work in [24] applied transfer learning and fine-tuning deep convolutional neural networks (CNNs). The pretrained model weight values are first fed into the layers of each network according to the proposed methodology, which then creates models that correspond to the VGG16 and ResNet152 structures. Finally, their town softmax classification layer is added as the final layer following the last fully connected layer. The networks were able to deliver an accuracy of around 99% when they were fed typical 2D images of various Arabic Sign Language data.

The study [34] also proposed a framework based on a variety of deep learning models for the automatic recognition of Arabic Sign Language, specifically by using AlexNet, VGGNet, and GoogLeNet/Inception models in training and evaluating the effectiveness of shallow learning techniques using nearest neighbors and SVM algorithms as baselines. The suggested algorithm provided encouraging results in detecting Arabic Sign Language with a 97% accuracy rate. A recent fully labeled dataset of images in Arabic Sign Language is used to evaluate the proposed models. The goal of work [35] is to solve the recognition problem for Arabic Sign Language while assuring a trade-off between improving classification performance and condensing the deep network's design to lower computational costs. To categorize Arabic Sign Language motions, AlKhuraym et al. specifically modified Efficient Network (EfficientNet) models and created lightweight deep learning algorithms. In addition, an actual dataset of hand motions for thirty distinct Arabic alphabets recorded by numerous signers was developed. The classification results generated by the suggested lightweight models were then evaluated using the proper performance indicators. Mahmoud et al. [23] developed an architecture that integrates transfer learning (TL) models and recurrent neural network (RNN) models for ArSL recognition. The results achieved in this work have a peak recognition accuracy of 93.4%.

The work in [36] reviewed the literature on deep learning techniques used for Arabic POS tagging during the previous two decades. The Preferred Reporting Items for Meta-Analyses and Systematic Reviews (PRISMA) methodology was used to perform the review. To extract all DL methods used to create POS taggers for the Arabic language, more than 4,000 publications were examined. Twelve articles were chosen for a thorough examination after numerous exclusion procedures. According to the reviewed publications, long short-term memory (LSTM) and Bi-LSTM models are the most popular DL approaches for Arabic POS tagging and produce the best results. On the other hand, in this work [37] on Arabic Sign Language detection, the images have been through a number of preprocessing and data augmentation procedures. On the ArASL dataset, tests have been run using a variety of pretrained models. Most of them performed rather typically, and in the last stage of the analysis, the EfficientNetB4 model was determined to be the best fit. Models other than EcientNetB4 performed poorly given the complexity of the dataset due to their lightweight construction. EcientNetB4 is a heavyweight architecture with a higher level of complexity. The best model is revealed with a 98% training accuracy and a 95% testing accuracy.

In paper [38], El Zaar et al. introduced a CNN-based highly efficient deep learning architecture. The suggested architecture is effective because it can recognize and analyze various datasets in sign language with a high degree of accuracy. One of the most crucial tasks that transform the lives of the deaf by making daily life and social inclusion easier is the recognition of sign language. Their system beats state-of-the-art methods, with a recognition rate of 99% for ASL and ISL and 98% for ArASL. It was trained and tested on datasets for American Sign Language (ASL), Irish Sign Alphabet (ISL), and Arabic Sign Language Alphabet (ArASL). The study in [22] provided a dataset of 20 Arabic words and proposed a deep learning architecture that combines convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The supplied dataset showed that the suggested architecture has a 98% accuracy rate. The top-1 accuracy on the UCF-101 dataset was reported to be 98.8%. Aldhahri et al. [21] employed convolutional neural networks to construct a model aimed at recognizing Arabic alphabet signs. The study utilized the Arabic alphabet's Sign Language Dataset (ArASL2018). The results from this model showed a recognition accuracy of 94.46%.

Prior research has explored various approaches in sign language recognition systems, aiming to facilitate effective communication for individuals with hearing and speech impairments. In this context, our study stands out by focusing on the development of an Arabic Sign Language Identification System (ArSL) using six distinct pretrained architectures: MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, and Xception. The critical aspect of our study is to distinguish our proposed model from the existing ones clearly. We thoroughly evaluate and compare the performance of these pretrained models, highlighting the superior accuracy achieved by ResNet50V2 and InceptionV3, both reaching 100% accuracy which is the highest achieved accuracy. This distinction allows us to emphasize the uniqueness and effectiveness of our approach in the realm of Arabic Sign Language recognition.

3. Aim of the Study

This study aims to advance Arabic Sign Language recognition utilizing state-of-the-art transfer deep learning techniques, with a focus on improving various research domains. The objective is to develop an accurate Arabic Sign Language Identification System (ArSL) leveraging deep neural networks. The motivation behind this work is to distinguish Arabic Sign Language by subjecting a neural network to diverse orientations and lighting conditions associated with images of hand gestures. The goal is to achieve higher accuracy compared to existing techniques, reduce training time with fewer epochs, and effectively handle images of varying sizes. To evaluate and identify the most effective approach, we employ six distinct pretrained architectures: MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, and Xception. The aim is to attain the highest accuracy possible, ultimately assisting individuals who are "deaf and mute" and ensuring the removal of

communication barriers they often encounter. The model utilizes a dataset composed of Arabic sign images for training, translating each sign image to an Arabic letter, making interaction with the broader population more accessible. The proposed model's key contribution lies in its automatic recognition of Arabic letters in sign language. The dataset utilized for this purpose is the "Arabic Alphabets Sign Language Dataset (ArASL)," comprising 32 labels, including 28 for the letters and 4 for standard Arabic signs. This dataset comprises 54,049 images of ArSL letters contributed by over 40 individuals, encompassing the full spectrum of standard Arabic Sign Language. In the pursuit of creating an efficient ArSL system, it is vital to distinguish our proposed model from existing ones. This involves highlighting the unique features, advantages, and outcomes achieved through the utilization of our carefully selected pretrained architectures. Importantly, we conduct a comparative analysis of our model's performance against other established models, underscoring the efficacy and relevance of our approach.

4. Materials and Methods

In this section, we present the dataset, utilized neural network models, data preparations, and processing.

4.1. Dataset. We utilized the Arabic Alphabets Sign Language Dataset (ArASL) (https://data.mendeley.com/datasets/y7pckrw6z2/) in this study. The used dataset comprises 54,049 images depicting Arabic Sign Language letters. These images were contributed by over 40 individuals and cover 32 standard Arabic signs and letters. It is worth noting that each class within the dataset contains a varying number of images. To organize and label the data, we employed a CSV file that associates each Arabic Sign Language image with its corresponding class label based on the image file's name. For visual reference, you can see some examples of the training data in Figure 1 [39].

4.2. Adopted Methodology. The adopted methodology section serves as a guide for how this work was carried out, encompassing the entire process from data collection to the production of study findings. We will delve into the major steps of the methodology as depicted in the flowchart, providing detailed explanations. Additionally, we will provide brief explanations of the pretrained models that were utilized in our study.

As depicted in Figure 2, we first import the essential packages and libraries, including Keras, Pandas, and Matplotlib. Then, the ArASL image data were loaded directly from the Kaggle website. As mentioned earlier, the dataset contains 54,049 images for 32 Arabic Sign Language characters. The first step after loading the dataset is to prepare the data to enter the model by implementing some preprocessing steps. Due to the dataset's imbalance issue, which means that every category holds a varying number of images, it may result in biased detection outcomes; hence, we solve this issue by aiming to avoid any inconsistencies and biases in the testing results. We allocated a fixed number of samples for each category in the dataset. Moreover, to complete preparing the dataset to enter the model, another data preprocessing step is performed, which is image resizing. The ArASL images are in different sizes, so all images were resized to a standard resolution of 64×64 pixels. In addition, image normalization is conducted to make the images more consistent in terms of contrast, color, and brightness. After that, data augmentation was applied. It is the process of generating new data from existing data to increase the data size and variety, thereby achieving better results. In our study, we implemented different augmentation techniques, including rescaling, zooming, flipping, and shifting.

Moving on to model development, the dataset was divided into training and testing sets with a 70% ratio for training and 30% for testing. The training set was entered into six chosen pretrained models, leveraging their efficiency and robustness in extracting complex patterns from data. These models are MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, and Xception. The models' weights are loaded using the ImageNet model, and the prediction layer is added using the softmax activation function after the last fully connected layer. We then fine-tuned these models using various settings by adjusting hyperparameters, including different learning rates and different number of epochs. After fine-tuning these models, we validate the effectiveness of the models on the validation set by measuring the accuracy score and visualizing the results for better understanding. Finally, we choose the best model.

4.3. Models. Pretrained models have found extensive application in the field of computer vision [40] due to their remarkable capacity to uncover hidden patterns and generalize effectively, even with small datasets and limited resources. In this section, we will explain the utilized pretrained models in our methodology.

(i) VGG16: The VGG16 neural network has a resolution of 70.5% and is computationally more expensive than neural networks [41]. The VGG16 network is an embedded system with more complexity because it consists of 16 layers, in which the convolutional layers (13) are stacked with 3×3 filters, which are adopted to improve the mesh depth, improve the mesh effect to a certain extent, and reduce the number of weight parameters [41]. Also, it has 2×2 assembly layers as maximum. Between these layers, the ReLU activation function is applied. Next, three fully connected layers contain most of the network parameters. Finally, the softmax function is used to produce the probabilities for each category [41].

(ii) InceptionV3: InceptionV3, also known as Inception-v3, represents the third version of Google's convolutional neural network, which was showcased during the ImageNet Identification Contest. GoogLeNet is particularly well-suited for processing extensive data, especially in scenarios where there are constraints on memory or computing resources.

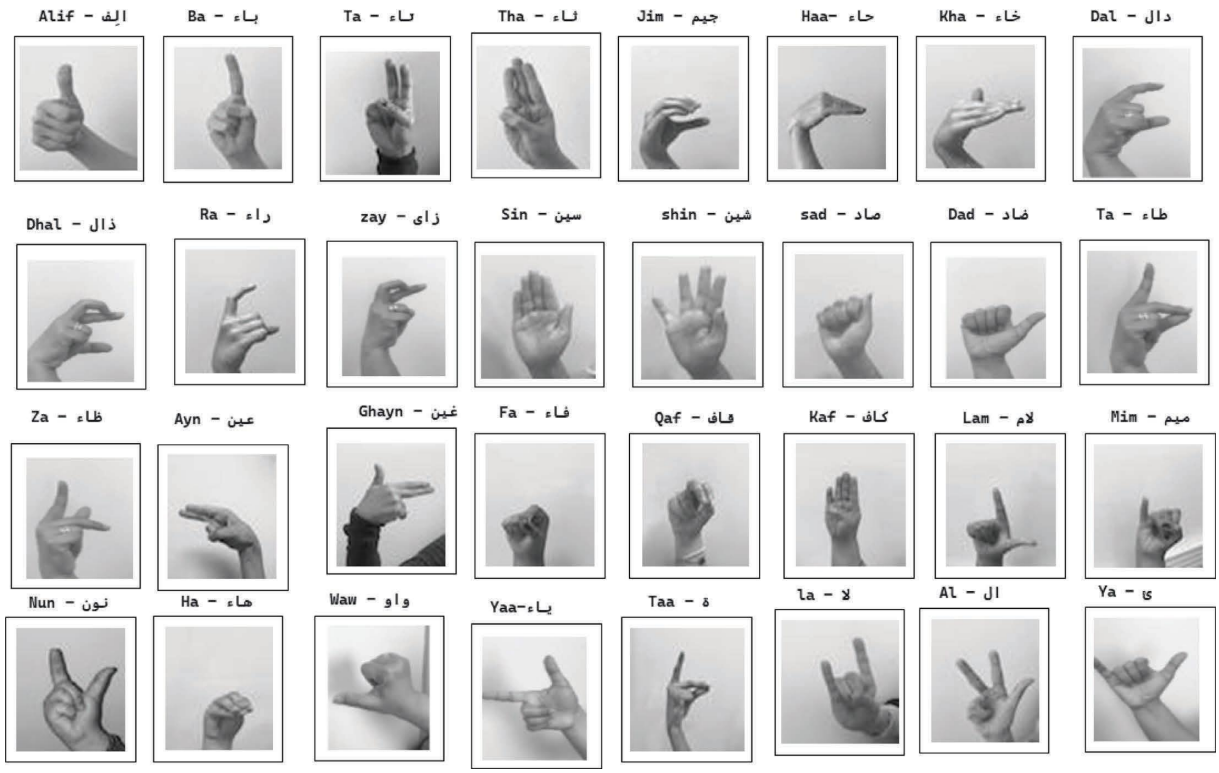

Figure 1: Samples of ArSL letters from the training data.

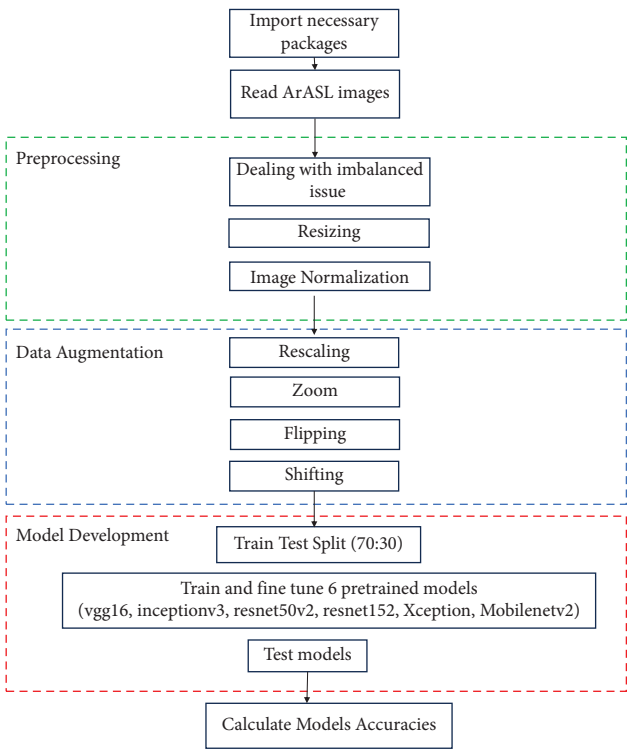

Figure 2: Flowchart of the proposed approach.

It excels in tasks such as image analysis, object detection, and object classification [42]. InceptionV3 consists of 48 layers, and the network's image input size is 299×299 pixels. It incorporates numerous enhancements, including the utilization of label smoothing, 7×7 convolutions, and the integration of an additional classifier to propagate label information throughout the grid. Additionally, it employs batch normalizing layers on the side (auxiliary branches) [43].

(iii) ResNet50V2: The ResNets [44] are modular structures that stack building blocks of the same continuous shape. Inception-ResNet-v2 is an improvement, a convolutional neural network that builds on foundation models but incorporates residual connections, replacing the filter sequence stage of the foundational architecture [45].

(iv) ResNet152: One year after the construction of VGGNet, the Residual Network (ResNet) emerged. The ResNet model was developed with various depths, ranging from 32 layers to 152 layers [46]. ResNet152, a deep network comprising up to 152 layers, learned residual representation functions instead of directly learning signal representation. It is eight times deeper than VGG networks while maintaining lower complexity. The ResNet group also achieved an error rate of 3.57% on the ImageNet test, securing first place in the ILSVRC 2015 classification challenge [46].

(v) Xception: Xception [47] expands upon the Inception architecture by replacing the standard Inception modules with deeply separable convolutions. In this architecture, deeply separable convolutions replace

the Inception modules. The original deep separable convolution consists of a depthwise convolution followed by a pointwise convolution, while the separable convolution starts with a pointwise convolution followed by a depthwise convolution. This modification is introduced in the starting module of InceptionV3, where a (1×1) convolution precedes any $(n \times n)$ spatial convolutions. As a result, Xception differs slightly from the original Inception architecture. Notably, the Xception architecture maintains the same number of parameters as InceptionV3, aiming for improved performance through more effective utilization of the model's parameters, rather than merely increasing capacity [47].

(vi) MobileNetV2: MobileNetV1 [48] emerged as a family of computer vision neural networks designed to support classification and detection in standard functions primarily built for mobile devices. It can run these networks on mobile devices, enhancing user experiences by providing benefits such as always-on access, privacy, security, and power efficiency. Subsequently, MobileNetV2 was introduced to power the next generation of mobile computer vision applications. MobileNetV2 represents a significant improvement over MobileNetV1 and incorporates the latest technology for mobile optical recognition, including support for various convolutional neural network applications such as object detection, classification, and semantic segmentation [49]. Released as part of the TensorFlow-Slim image classification library, MobileNetV2 builds on ideas from MobileNetV1 [49], using separate depthwise convolutions as efficient building blocks. Additionally, MobileNetV2 introduces new architectural features, including linear bottlenecks between layers and shortcut connections between bottlenecks.

5. Experiments and Results

Due to the remarkable success of convolutional neural networks (CNNs) in the field of sign language recognition, we conducted a comprehensive study to compare the performance of several pretrained models. Our goal was to determine the most effective model for recognizing signs using transfer learning. We used the ArASL dataset [39] in the training and validation phases, which consisted of a substantial 54,049 images, each depicting one of 32 Arabic signs.

Our proposed technique comprised several key steps:

(i) Preprocessing: we initiated the process by carefully preprocessing the images.

(ii) Fine-tuning: the pretrained models underwent a fine-tuning process using the preprocessed images.

(iii) Data augmentation: to improve the model's generalization and mitigate overfitting, we applied data augmentation techniques.

(iv) Monitoring performance: at the end of each epoch, we assessed the performance of each network using accuracy as a key metric.

(v) Varied experiments: we conducted multiple experiments, exploring different numbers of epochs, batch sizes, and learning rates to comprehensively evaluate each model's performance.

(vi) Early stopping: to prevent overfitting, we implemented early stopping strategies during training.

The dataset was divided into two subsets: a validation set comprising 30% of the data and a training set with the remaining 70%. The results of our evaluation revealed that ResNet50V2 and InceptionV3 outperformed the other models. Both achieved an exceptional accuracy rate of 100%, with an error rate of 0%. ResNet50V2 was trained for 10 epochs, and InceptionV3 was trained for 6 epochs, both using a batch size of 32. Our application of early stopping and data augmentation techniques contributed to preventing overfitting and enhancing the models' ability to generalize. In summary, the experiments indicated that InceptionV3, ResNet50V2, MobileNetV2, Xception, and VGG16 exhibited superior performance when compared across various hyperparameters and network settings. This comparison revealed significant improvements in the models' speed and accuracy. Furthermore, we fine-tune these models for 3 epochs, 6 epochs, and 10 epochs. Table 2 shows the results of these models after 3 epochs.

Table 2 shows the differences in the results after the training of these models finished with three epochs. As we see, ResNet50V2 and Xception show the highest accuracy scores equal to 98% and 97% with the loss equal to 0.01 and 0.03, respectively. However, in ResNet50V2, we used an adoptive learning rate (decreasing the value of LR every three epochs) while the Xception model used a 0.001 learning rate. Figure 3 is an overview of accuracy scores on three epochs for all models.

Table 3 shows the differences in models' performance on 6 epochs; the primary interpretation is that InceptionV3 and ResNet50V2 achieved 100% accuracy score with a 0.01 loss score. These two models are optimized through Adam optimizer and with batch size 32. In addition, setting an adaptive learning rate, e.g., by reducing the learning rate (LR) value after a certain number of epochs, leads to an improvement in the performance of the models such as in ResNet50V2where the LR is reduced from 0.001 to 0.005 on epoch 2.

Table 4 shows the differences in model performance over 10 epochs; the highest accuracy was again achieved by ResNet50V2 which achieved an accuracy score of 100% with a loss score of 0.01. As we can see in the table, Xception and VGG16 results achieved less accuracy in this iteration. Also, the lowest error rates were also achieved in VGG16, InceptionV3, and ResNet50V2 compared to other models. Figure 4 visualizes the accuracy for these six models after 10 epochs.

Table 2: Performance of pretrained models after 3 epochs.

Model name	Test accuracy	Test loss	Optimizer	LR
VGG16	0.96	0.11	Adam	0.0001
InceptionV3	0.89	0.3	Adam	0.001
ResNet50V2	0.98	0.01	Adam	0.001
ResNet152	0.46	0.3	Adam	0.0001
Xception	0.97	0.03	Adam	0.001
MobileNetV2	0.95	0.17	Adam	0.001

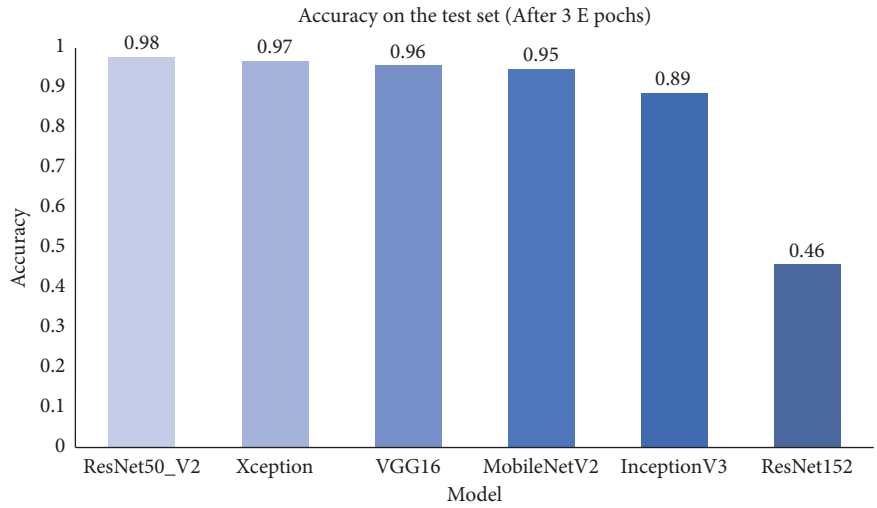

Figure 3: Differences in accuracy between different pretrained models after 3 epochs.

Table 3: Performance of pretrained models after 6 epochs.

Model name	Test accuracy	Test loss	Optimizer	LR
VGG16	0.97	0.1	Adam	0.0001
InceptionV3	1	0.01	Adam	0.001
ResNet50V2	1	0.01	Adam	0.001
ResNet152	0.87	0.4	Adam	0.0001
Xception	0.97	0.13	Adam	0.001
MobileNetV2	0.98	0.01	Adam	0.001

Table 4: Performance of pretrained models after 10 epochs.

Model name	Test accuracy	Test loss	Optimizer	LR
VGG16	0.95	0.04	Adam	0.0001
InceptionV3	0.97	0.08	Adam	0.001
ResNet50V2	1	0.0	Adam	0.001
ResNet152	0.86	0.35	Adam	0.0001
Xception	0.93	0.12	Adam	0.001
MobileNetV2	0.96	0.11	Adam	0.001

Figure 4 shows on the left the accuracy through the epochs at test time for the best model (ResNet50V2) as well as shows the loss on the right.

Figure 5 shows the best results of the models across different number of epochs and different batch sizes; InceptionV3 achieved a 100% accuracy on 6 epochs with loss value equal to

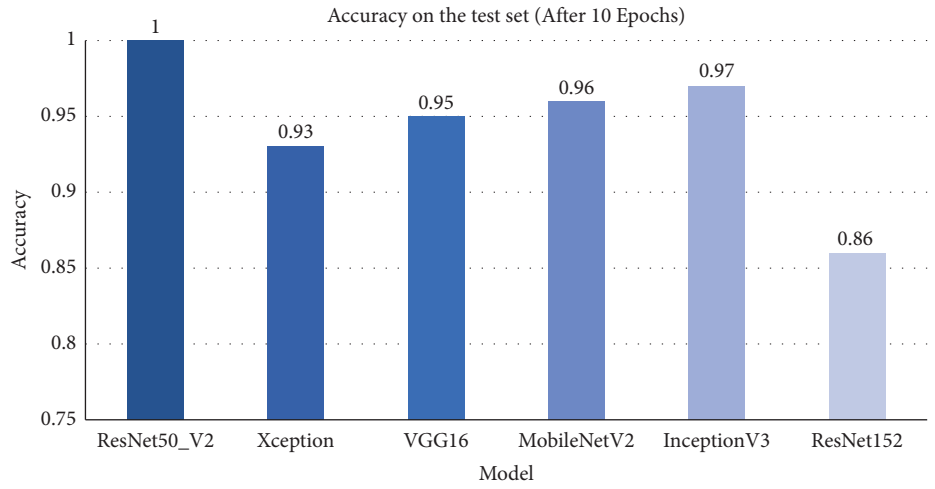

Figure 4: Differences in accuracy between different pretrained models after 10 epochs.

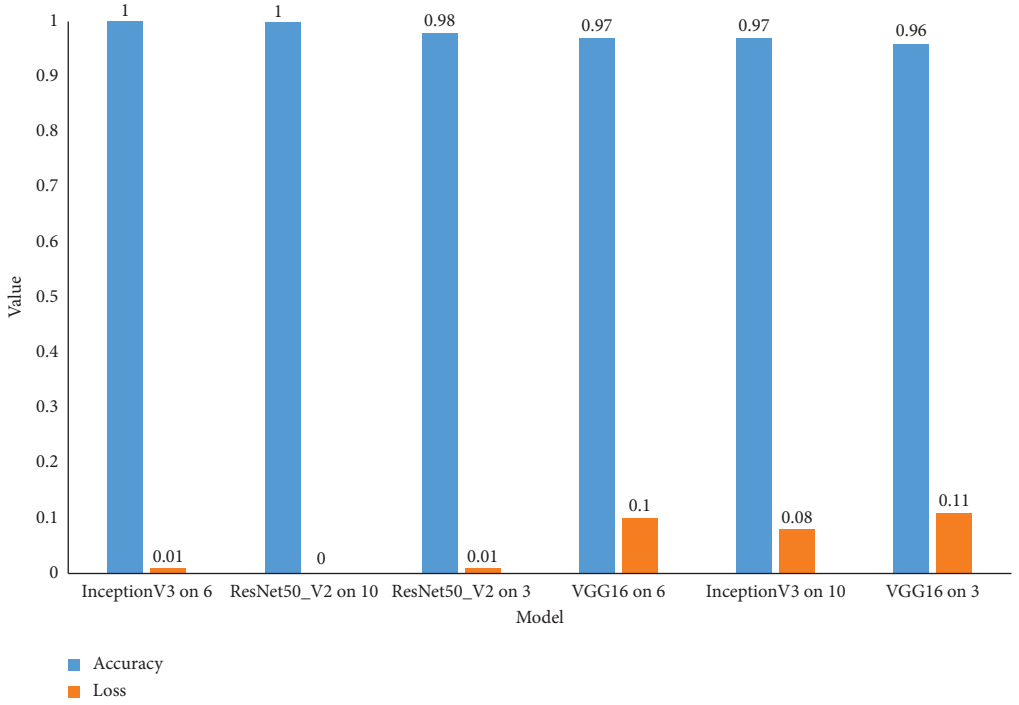

Figure 5: Optimal performance of models across varied epochs and batch sizes.

0.01 and batch size = 32; ResNet50V2 also achieved the same results but on 10 epochs and 98% on 3 epochs with 0.01 loss and the same batch size; VGG16 and InceptionV3 attained the same results of accuracy on 6 and 10 epochs, but we are interested in training the network on smaller number of epochs with the best result. So, we prefer InceptionV3 on 3 epochs over the VGG16 on 6 in this case.

The primary contribution of this study lies in the exceptional performance demonstrated by ResNet50V2 and InceptionV3 in fitting our model to our dataset. These models achieved outstanding results with 100% accuracy and zero errors, showcasing their remarkable ability to classify sign language images into Arabic letters effectively. Throughout the training phase of InceptionV3, we diligently applied early stopping mechanisms and, when applicable, data augmentation techniques. These strategies played a pivotal role in enhancing the model's generalization to previously unseen data. Our approach focused on achieving the right balance through iterative experimentation, ensuring the model was finely tuned and effectively mitigated overfitting.

The variance in the number of epochs required for convergence comes from several factors related to the model's training, including the initialization conditions, hyperparameter adjustments, and the difference in model complexity. During our experimentation, we adjusted

different hyperparameters, such as the learning rate, early stopping criteria, and batch size. These adjustments impact convergence directly. Moreover, based on the particular model complexity, it might have different convergence behaviors; more complex models require a larger number of epochs to fine-tune the high number of parameters and reach convergence. In conclusion, it is evident that InceptionV3 outperformed other pretrained models in our comparison.

6. Conclusion

In our research, our goal was to accurately classify Arabic Sign Language images using advanced artificial intelligence techniques. We achieved this by harnessing the power of pretrained models, including MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, and Xception. The dataset comprised a vast collection of Arabic Sign Language images. To ensure a fair evaluation, we split the dataset into two portions: 70% for training and 30% for validation. After conducting extensive experiments, particularly fine-tuning of the pretrained models, we observed exceptional performance from ResNet50V2 and InceptionV3. These models achieved an impressive 100% accuracy with zero errors. The training process involved 10 and 6 epochs, with a batch size of 32. To further enhance the model's performance and prevent overfitting, we applied techniques like early stopping and data augmentation. In summary, InceptionV3 consistently outshone the other pretrained models across various experiments with different number of epochs. What is really interesting here is that they achieved this remarkable accuracy without falling into the trap of overfitting. This highlights the effectiveness of incorporating techniques like early stopping and data augmentation, which played a crucial role in enabling InceptionV3 to generalize exceptionally well while maintaining its high accuracy and avoiding the risk of overfitting.

Data Availability

The data used to support the findings of this study are deposited in a repository and are publicly available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. M. Lieberman, A. Borovsky, and R. I. Mayberry, "Prediction in a visual language: real-time sentence processing in american sign language across development," *Language, cognition and neuroscience*, vol. 33, no. 4, pp. 387–401, 2018.

[2] K. Al-Fityani and C. Padden, "Sign Language Geography in the arab World," *Sign languages: A Cambridge survey*, Cambridge University Press, Cambridge, UK,pp. 433–450, 2010.

[3] A. A. I. Sidig, H. Luqman, and S. A. Mahmoud, "Arabic sign language recognition using optical flow-based features and hmm," in *Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017)*, pp. 297–305, Springer, Singapore, 2018.

[4] H. Luqman and S. A. Mahmoud, "Automatic translation of Arabic text-to-Arabic sign language," *Universal Access in the Information Society*, vol. 18, no. 4, pp. 939–951, 2019.

[5] A. A. I. Sidig, H. Luqman, and S. A. Mahmoud, "Transform-based Arabic sign language recognition," *Procedia Computer Science*, vol. 117, pp. 2–9, 2017.

[6] J. Joy, K. Balakrishnan, and M. Sreeraj, "Signquiz: a quiz based tool for learning fingerspelled signs in indian sign language using aslr," *IEEE Access*, vol. 7, pp. 28 363–428 371, 2019.

[7] E. Costello, *American Sign Language Dictionary*, Random House Reference, New York, NY, USA, 2008.

[8] K. Bantupalli and Y. Xie, "American sign language recognition using deep learning and computer vision," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 4896–4899, IEEE, Seattle, WA, USA, December 2018.

[9] M. Tolba and A. Elons, "Recent developments in sign language recognition systems," in *Proceedings of the 2013 8th International Conference on Computer Engineering & Systems (ICCES)*, IEEE, Cairo, Egypt, November 2013.

[10] V. Ranga, N. Yadav, and P. Garg, "American sign language fingerspelling using hybrid discrete wavelet transform-gabor filter and convolutional neural network," *Journal of Engineering Science & Technology*, vol. 13, no. 9, pp. 2655–2669, 2018.

[11] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: enabling robust statistical continuous sign language recognition via hybrid cnn-hmms," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1311–1325, 2018.

[12] K. Silanon, "Thai finger-spelling recognition using a cascaded classifier based on histogram of orientation gradient features," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 9026375, 11 pages, 2017.

[13] S. Semreen and M. Albinali, *The Rules of arab Qatari Sign Standardized Language*, Supreme Council of Family Affairs, Sharjah, UAE, 2010.

[14] S. C. Agrawal, A. S. Jalal, and R. K. Tripathi, "A survey on manual and non-manual sign language recognition for isolated and continuous sign," *International Journal of Applied Pattern Recognition*, vol. 3, no. 2, pp. 99–134, 2016.

[15] H. Cooper, B. Holt, and R. Bowden, "Sign language recognition," in *Visual Analysis of Humans: Looking at People*, pp. 539–562, Springer, Singapore, 2011.

[16] K. Nimisha and A. Jacob, "A brief review of the recent trends in sign language recognition," in *Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 186–190, IEEE, Chennai, India, July 2020.

[17] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," in *Proceedings of the International conference on artificial intelligence and statistics*, pp. 4313–4324, PMLR, August 2020.

[18] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Computing & Applications*, vol. 32, no. 12, pp. 7957–7968, 2020.

[19] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep learning for sign language recognition: current techniques, benchmarks, and open issues," *IEEE Access*, vol. 9, pp. 126917–126951, 2021.

[20] M. Mustafa, "Retracted article: a study on Arabic sign language recognition for differently abled using advanced

machine learning classifiers," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4101–4115, 2021.

[21] E. Aldhahri, R. Aljuhani, A. Alfaidi et al., "Arabic sign language recognition using convolutional neural network and mobilenet," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2147–2154, 2023.

[22] M. M. Balaha, S. El-Kady, H. M. Balaha et al., "A vision-based deep learning approach for independent-users Arabic sign language interpretation," *Multimedia Tools and Applications*, vol. 82, no. 5, pp. 6807–6826, 2022.

[23] E. Mahmoud, K. Wassif, and H. Bayomi, "Transfer learning and recurrent neural networks for automatic Arabic sign language recognition," in *Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications*, pp. 47–59, Springer, Cairo, Egypt, May 2022.

[24] Y. Saleh and G. Issa, "Arabic sign language recognition through deep neural networks fine-tuning," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 5, pp. 71–83, 2020.

[25] M. Kamruzzaman, "Arabic sign language recognition and generating Arabic speech using convolutional neural network," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 3685614, 9 pages, 2020.

[26] M. A. Almasre and H. Al-Nuaim, "A comparison of Arabic sign language dynamic gesture recognition models," *Heliyon*, vol. 6, no. 3, Article ID e03554, 2020.

[27] S. M. Elatawy, D. M. Hawa, A. A. Ewees, and A. M. Saad, "Recognition system for alphabet Arabic sign language using neutrosophic and fuzzy c-means," *Education and Information Technologies*, vol. 25, no. 6, pp. 5601–5616, 2020.

[28] S. Aly and W. Aly, "Deeparslr: a novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition," *IEEE Access*, vol. 8, pp. 83 199–283 212, 2020.

[29] A. Alnahhas, B. Alkhatib, N. Al-Boukaee, N. Alhakim, O. Alzabibi, and N. Ajalyakeen, "Enhancing the recognition of Arabic sign language by using deep learning and leap motion controller," *International Journal of Scientific and Technology Research*, vol. 9, pp. 1865–1870, 2020.

[30] N. Saleh, M. Farghaly, E. Elshaaer, and A. Mousa, "Smart glove-based gestures recognition system for Arabic sign language," in *Proceedings of the 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pp. 303–307, IEEE, Aswan, Egypt, February 2020.

[31] S. Hayani, M. Benaddy, O. El Meslouhi, and M. Kardouchi, "Arab sign language recognition with convolutional neural networks," in *Proceedings of the 2019 International conference of computer science and renewable energies (ICCSRE)*, pp. 1–4, IEEE, Agadir, Morocco, July 2019.

[32] M. ElBadawy, A. Elons, H. A. Shedeed, and M. Tolba, "Arabic sign language recognition with 3d convolutional neural networks," in *Proceedings of the 2017 Eighth international conference on intelligent computing and information systems (ICICIS)*, pp. 66–71, IEEE, Cairo, Egypt, December 2017.

[33] A. Salama, F. Smarandache, and M. Eisa, *Introduction to Image Processing via Neutrosophic Techniques*, Infinite study, Conshohocken, PA, USA, 2014.

[34] R. M. Duwairi and Z. A. Halloush, "Automatic recognition of Arabic alphabets sign language using deep learning," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, p. 2996, 2022.

[35] B. Y. AlKhuraym, M. M. B. Ismail, and O. Bchir, "Arabic sign language recognition using lightweight cnn-based architecture," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.

[36] M. Bouzahir, A. Ait Abdelouahad, and M. Nabil, "How far can deep learning improve Arabic part of speech tagging?" in *Proceedings of the International Conference on Business Intelligence*, pp. 206–215, Springer, Harbin, China, December 2022.

[37] M. Zakariah, Y. A. Alotaibi, D. Koundal, Y. Guo, and M. Mamun Elahi, "Sign language recognition for Arabic alphabets using transfer learning technique," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 4567989, 15 pages, 2022.

[38] A. El Zaar, N. Benaya, and A. El Allati, "Sign language recognition: high performance deep learning approach applyied to multiple sign languages," *E3S Web of Conferences*, vol. 351, Article ID 1065, 2022.

[39] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "Arasl: Arabic alphabets sign language dataset," *Data in Brief*, vol. 23, Article ID 103777, 2019.

[40] A. Talmor, O. Tafjord, P. Clark, Y. Goldberg, and J. Berant, "Leap-of-thought: teaching pre-trained models to systematically reason over implicit knowledge," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 227–20 237, 2020.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[42] Wikipedia, "Inceptionv3," 2021, https://en.wikipedia.org/w/index.php?title=Inceptionv3&oldid=1014549339.

[43] 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*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the European conference on computer vision*, pp. 630–645, Springer, Amsterdam, The Netherlands, October 2016.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, San Francisco, CA, USA, February 2017.

[46] 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*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[47] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[48] Googleblog, "Mobilenetv2: the next generation of on-device computer vision networks," 2018, https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html1.

[49] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.