ORIGINAL ARTICLE

A comparative analysis of Indian sign language recognition using deep learning models

Bunny Saini[1], Divya Venkatesh[1], Nikita Chaudhari[1], Tanaya Shelake[1], Shilpa Gite[1,2], Biswajeet Pradhan[3*]

[1] Computer Science Engineering Department, AIML Department, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India
[2] Symbiosis Centre of Applied AI (SCAAI), Symbiosis International (Deemed University), Pune 412115, India
[3] Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), School of Civil and Environmental Engineering, Faculty of Engineering and IT, University of Technology Sydney, Ultimo 2007, Australia

Abstract: Sign language is a form of communication where people use bodily gestures, particularly those of hands and arms. This method of communication is put into motion when spoken communication is unattainable or disfavored. There are very few people who can translate sign language and readily understand them. It would be convenient for the hearing-impaired to have a platform where their sign language could be translated easily. Hence, through this study, with the help of artificial neural networks, we wish to compare how various widely implemented deep learning architectures respond to faultless translation of Indian sign language for the native audience. This research would streamline the development of software tools that can accurately predict or translate ISL. For the purpose of understanding the method of training the machine and exploring our model's performance without any optimizations, a Convolutional Neural Network architecture was implemented. Over the course of our research, there have been several Pre-trained Transfer Learning Models implemented that have yielded promising results. The research aims to contrast how various convolutional neural networks perform while translating Indian Sign Actions on a custom dataset that factors in illumination, angles, and different backgrounds to provide a balanced and distinctive set of images. The goal of this study is to make clear comparisons between the various deep learning frameworks. Hence, a fresh Indian sign language dataset is introduced. Since every dataset in the field of deep learning has special properties that may be utilized for the betterment of the existing models, the development of a fresh dataset could be viewed as a development in the field. The optimum model for our task: classification of these gestures is found to be ResNet-50 (Accuracy = 98.25% and F1-score = 99.34%), and the least favorable was InceptionNet V3 (Accuracy = 66.75%, and F1-score = 70.89%).

Keywords: Indian sign language; deep learning; transfer learning; convolutional neural network; ResNet-50

***Corresponding author:** Biswajeet Pradhan, Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), School of Civil and Environmental Engineering, Faculty of Engineering and IT, University of Technology Sydney, Ultimo 2007, Australia; Biswajeet.Pradhan@uts.edu.au

Received: March 31, 2023; **Accepted:** May 18, 2023; **Available online:** July 4, 2023

Citation: Saini B, Venkatesh D, Chaudhari N, et al. (2023) A comparative analysis of Indian sign language recognition using deep learning models. *Forum for Linguistic Studies* 5(1): 197–222. DOI: 10.18063/fls.v5i1.1617

1. Introduction

Sign language is an integral part of the lives of the deaf, mute, and hard of hearing (HOH). Using the ways of an oralist is not easy for every deaf person. Many find it difficult to precisely read and interpret lips and sometimes it could also lead to the conveying of wrong messages. This problem makes sign language the most fitting way of communication for the deaf and mute. The history of sign language stretches back to the seventeenth century as a visual form of communication. However, many instances of using hand gestures and signs to communicate can date back to the fifth century BC Greece (Fable, 2022). Over 5% of the population, approximately 466 million people, with 34 million children and the rest adults, are affected by hearing impairments of various kinds (World Health Organization, 2019), making them a cultural-linguistic minority. Different sign languages have been formulated as a simple and effective form of communication to help these people overcome the difficulties they confront in interacting with the rest of the society. Multiple sign languages exist worldwide, and the absolute number is not precisely known since most countries typically have a native sign language of their own. Ethnologue (2022), a publication that provides statistics on the various languages of the world, listed 157 sign languages as of 2022. The signs and gestures belonging to each country vary based on their culture, geography, and history. In India, the deaf and mute use the Indian Sign Language (ISL), also called the Indo-Pakistani Sign Language (IPSL). Throughout the South Asian subcontinent, IPSL is the principal sign language used by over fifteen million deaf and mute individuals (Vasishta et al., 1978). In India alone, around eighteen million native speakers use sign languages, and six million use ISL to communicate in their day-to-day life (Khan, 2017).

Most schools in South Asia tend to cater to kids with a strong Oralist approach rather than sign language (Deshmukh, 1997). Oralism refers to educating deaf students through oral speech methods, including studying lip-reading, imitating mouth shapes, and breathing habits of speech, which can be difficult for many children to produce and perceive (Mandke and Chandekar, 2019). It was only in 2005 that the National Curricular Framework (NCF) hinted that school students could opt for sign language as an optional third language (Wikipedia, 2023). No formal lessons were taken in India to teach the ISL until 2001. The Ali Yavar Jung National Institute of Hearing and the Handicapped (AYJNIHH) in Mumbai was the first to start a diploma course on ISL that aimed to develop professional communication and interpretation of the ISL. In March 2006, the National Council of Educational Research and Training (NCERT) published a chapter in the third-grade textbooks about sign language, describing how sign language is just another mode of communication and is akin to any other language. All these steps were taken to slowly bring awareness among the students and even the rest of society.

The word deaf refers to people who undergo loss of hearing to such an extent that there is only little or no functional hearing. If hearing loss is not extremely severe, and the person's hearing can be supported with a Cochlear implant or hearing aid, the condition is referred to as "Hard of Hearing" or HOH (World Health Organization, 2023). Even today, the deaf community faces many challenges; nine out of ten deaf children are born to families with hearing parents. Nevertheless, not even a third of these families have members who can freely use sign language (Correll, 2022). For this very reason, it is necessary to build a platform where everyone, including the disabled, can communicate freely. With the help of the growing information technology, this paper attempts to build a place where the deaf and the mute can express themselves. Here, the model will capture different gestures from the user and attempt to recognize them as signs from the ISL. This will be implemented using neural networks and deep learning techniques. The concept of neural networks is based on the conduct of the human brain; they are made up of interconnected layers of algorithms, known as neurons that feed data into one another, where the product of one layer serves as the input for the next. After learning from enormous portions of data, the system is capable of making predictions with better precision (Heath, 2020). This way, the proposed model can recognize the signs and translate them into text. We wish to implement this in the future with real-time images taken from videos or cameras.

Even though many deaf people use sign language, the Deaf communities in India found themselves struggling to acquire the status of the ISL as a minority language (Mitter, 2017). This motivates us to carry out this study; despite the spread of awareness over the struggles of the deaf and the mute throughout the country, there are only a few individuals outside of the Deaf community that can understand or interpret sign language, making it more than inconvenient for the disabled to communicate with the larger mass.

In this research, we implemented convolutional neural networks (CNN) which is one of the most prevalent and utilized deep learning (DL) networks (Yao et al., 2019). The biggest benefit of CNN is its automated detection of significant features in the absence of human supervision (Gu et al., 2018). CNN becomes more convenient for use with its large-scale network implementation as compared to other traditional neural networks. Feedforward neural networks show the ability to learn a single feature of an image, but when it comes to much more complex images, other neural networks like artificial neural networks (ANN) fail to make better predictions. On the other hand, CNN concurrently learns feature extraction layers and classification layers making the model output positively collected and reliant (Goyal, 2021). CNN's weight sharing feature decreases the amount of trainable network parameters, helping enhance generalization and reduce chances of overfitting (Alzubaidi et al., 2021).

Research highlights of the paper are as follows:

- A detailed literature survey was carried out in the ISL Recognition domain.
- A custom ISL Dataset with varied and diverse samples was developed that aided better learning.
- Deep learning-based CNNs, including transfer learning models, were implemented for ISL Image Recognition.

This work implements basic CNN architecture and pre-trained DL models which are already freely available and implemented like VGG-16, VGG-19, ResNet-50, DenseNet-201, Inception V3, MobileNet V2, and Xception. These models are designed specifically for image classification by researchers in the field and there is a lack of research on sign language recognition by transfer learning and comparison for all 7 models together. Dataset creation consisted of a major part of this research and the letters of ISL were captured in pictures solely for the purpose of this research. In similar work, Sharma and Anand (2021) have implemented transfer learning on pre-trained models limited to Inception V3 and ResNet-50. All existing work does not work on their own datasets but rather works on existing datasets that were not adequate for this work.

This paper is split into several sections, each of which lays the groundwork for the next and establishes the paper's structure. Section 1 introduces the problem statement. Section 2 describes a thorough literature review of the related work in this domain. Section 3 details the data acquisition phase followed by Section 4, providing the elementary knowledge of all the models utilized in the process. Section 5 describes the various evaluation metrics used to assess the model's performance, and Section 6 highlights the results and discusses the key takeaways from this study. The paper concludes with Section 7 and provides future directions.

2. Related work

There has been a lot of work done in sign language and hand recognition using machine learning techniques and we have reviewed those studies to understand the mechanism and the shortcomings of the work done.

Adeyanju et al. (2021) defined the methodology for detecting sign language recognition using machine learning methods and image processing techniques. They used image acquisition, image preprocessing, segmentation, feature extraction, and classification processes to build their final model. They used vision-based sign language recognition, involving matching the features of new sign images with stored databases for recognizing the given image. The shortcomings included the cost of implementation, accuracy and complexity of the signs and image backgrounds. As part of the literature review, we performed an insightful analysis of the techniques used in SLR in the past and where they lack. Pigou et al. (2014) proposed a solution to sign language recognition using Microsoft Kinect, CNNs, and GPU acceleration. The authors stated that CNNs could automate the recognition and feature construction process. They implemented models with ReLus, dropout, and LCN with data augmentation. The accuracy of the test set was around 96% and their models were able to identify and recognize different sign languages with inputs of users and surroundings not included in their training data. This research was done on Italian gestures and did not use transfer learning.

Sign language recognition using DL has been researched but there is an evident lack of research in the domain of ISL as compared to other types of sign languages. This work aims to lessen the bridge between ISL and DL. Whilst there have been papers published for sign languages from other nations, there are a few good works on ISL which we have talked about briefly below.

Sharma and Anand (2021) used DL to build their sign recognition software and pre-trained models, optimization hyperparameters, and gradient-based optimizers for static ISL recognition. Out of all models used, ResNet delivered the best results for both numerical and alphabetical data classification. Fine-tuning of CNN models was implemented, where dense, flattened, and dropout layers replaced the final layer of the CNN architecture. Their study showed that building a CNN network from scratch gave better results than pre-trained models. The results on their pre-trained models like ResNet and Inception V3 were not good with Inception V3 showing the least performance. Our study aims to improve results on these pre-trained models.

As noted, there is tremendous work done on American Sign Language (ASL) and a few remarkable works have been discussed hereon. Garcia and Viesca (2016) used transfer learning using GoogLeNet and Caffe to build a real-time sign language translator. They fine-tuned a pre-existing model to fit their parameters and data which consisted of hand gestures of the ASL in around 24 random orientations. Caffe, which is a DL framework, was used to develop, test, and run their CNNs. Three different losses at various depths of the net were shown in their net output. Their dataset had a lack of variation and hence their training accuracy was not translated to testing. Similarly, in our work, we had to switch datasets and build our own dataset to introduce variations and enhance accuracy. Barbhuiya et al. (2021) applied the CNN model to create a robust model of static signs concerning sign language recognition. Their CNN architecture is based on modified AlexNet and VGG-16 models for classification and modified pre-trained models for feature extraction. This was followed by a multiclass support vector machine (SVM) classifier, and they successfully achieved recognition accuracy of 99.82%. This research focused on ASL and our research aims to implement the same in ISL, too. Adithya and Rajesh (2020) used ANN techniques that automated the feature extraction process where feature extraction is performed on data by processing the data through certain hierarchical layers to recognize static hand gestures. Their model had low computational complexity and high processing speed. The model is constructed using the input layer followed by 3 CNN layers, feature extraction by Relu Max Pooling layers, and a softmax layer followed by an output layer for classification. Their study did not explore transfer learning in detection.

Sharma and Singh (2021) used the Graph Convolutional Neural Network (G-CNN) model to design a model that would recognize gesture-based sign language. They trained and tested the VGG-11 and VGG-16 models to evaluate the efficacy of the model. The G-CNN model achieved the highest classification accuracy of 94.83%, 99.96%, and 100% for three different categories of hand gestures, which included ISL as well as ASL. The takeaway from this study was that their proposed model outperformed pre-trained models. The shortcomings are that they do not include other pre-trained models like MobileNet and Inception in their study.

Bansal et al. (2018) applied machine learning techniques to help identify neurodegenerative disorders like dementia. The authors implemented a comparative analysis using various machine learning algorithms to detect dementia. Naive Bayes algorithm, multilayer perceptron, and Random Forest algorithm were implemented in this paper, where CFSSubset is used for attribute reduction. J48 was found to be the best-performing algorithm for detecting dementia after their analysis. Their dataset had a lot of redundant features and hence needed more work done for better results. In another work, Dutta et al. (2017) emphasized the importance of using DL for medical image classification for Diabetic Retinopathy and computed tomography (CT) emphysema. Medical images are quite hard to classify and automating this classification can reduce human errors and computers are good at focusing on tiny details that are easy to miss. Deep neural networks and CNNs were used after the application of the ANN for detecting image severity levels. The CNNS were found to give 68% and 64.8% accuracy for DR and CT images, respectively. Our research benefits from theirs as they used CNNs to predict images which we aim to do with ISL Images. One shortcoming of this work includes the computational time it took for CNNs. Challenges faced by them were the lack of GPU support which greatly affected results and a drawback was that their images had only one band which also affected the results.

Rajalakshmi et al. (2022) built a hybrid neural network architecture for recognition of static and dynamic Indian and Russian Sign Language (RSL) with spatial and temporal recognition. They also built a novel dataset consisting of ISL and RSL signs that they used for their validation. For static SL recognition, their model proposed a 3D Convolution Net and for dynamic, it was a combination of spatial and temporal feature detection and extraction. Challenges they faced included the lack of relevant RSL datasets and lack of research in the domain.

Pre-existing datasets on both ASL and ISL are abundant online but most of them share a common factor that proved detrimental to our research, i.e., they had duplicate images which left us with a lack of variety of settings & surroundings and no heterogeneity. The other better datasets that were discovered were not suited for the research planned and consisted of videos and sentence based ISL images. A dataset was created from scratch to combat the issues faced. The datasets we explored and tried to work with are given in **Table 1**.

As it can be seen in the aforementioned literature review, it is evident that extreme computation power is required for SLR (Sign Language Recognition), and datasets need to be distinct and need adequate feature combinations. Previous studies have implemented SLR with CNNs on predominantly ASL datasets. Research on state-of-the-art models like DenseNet-201, ResNet-50, MobileNet V2, VGG-19 and Xception for SLR using transfer learning is lacking and has huge scope to be worked on.

3. Dataset creation

In the earlier stages of this study, the notion was to utilize ISL datasets available in the public domain. However, upon detailed inspection, we found that a lot of these datasets consisted of an alarming number of duplicate images, insufficient variance, and diversity in the samples. ISL datasets are challenging to find due to various problems related to handedness, the difficulty of learning the language, and inadequate attention to native sign languages such as Indian. Hence, we created our samples from scratch. Over two months, we captured about 10,400 image samples with 26 classes to interpret all the hand poses of ISL pertaining to alphabets. Each class comprised 400 images of signs or gestures of each English Alphabet in ISL. For this medium-scaled dataset, we split the images in a 4:1 ratio, with around 8330 images being used for training and the rest for testing. This ratio is chosen proportional to the size of our dataset, making sure that we have a sufficiently diversified training set and a test set that has good unseen samples of all the images to help the model to validate the learning ability accurately. **Figure 1** depicts the hand gestures for the ISL alphabet used in this study.

Table 1. Pre-existing datasets on Indian sign language recognition

Sr. No	Name	Provider	Year	Reason	Reference
1	Indian Sign Language (ISL)	Arikeri P	2020	Dark background, lack of variations, duplicate images	Arikeri (2021)
2	Indian Sign Language (ISLRTC referred)	Dumbre A	2022	Lack of variety, duplicate images	Dumbre (2022)
3	ISL-CSLTR: Indian Sign Language dataset for continuous Sign Language Translation and Recognition	Elakkiya R, Natarajan B	2021	Videos and sentence based which was not the goal of this work	Natarajan and Elakkiya (2021)
4	Indian Sign Language Dataset	Sonawane V	2020	Duplicate images	Sonawane (2020)
5	Indian Sign Language Dataset recognized by ISRTC	Kumar K	2022	Bad quality, no variety	Kumar (2022)

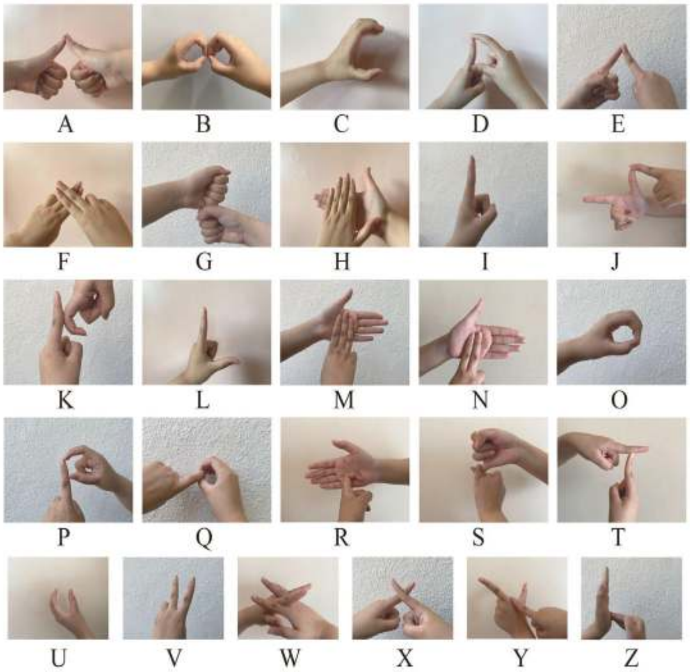


Figure 1. Illustration of dataset sign gestures.

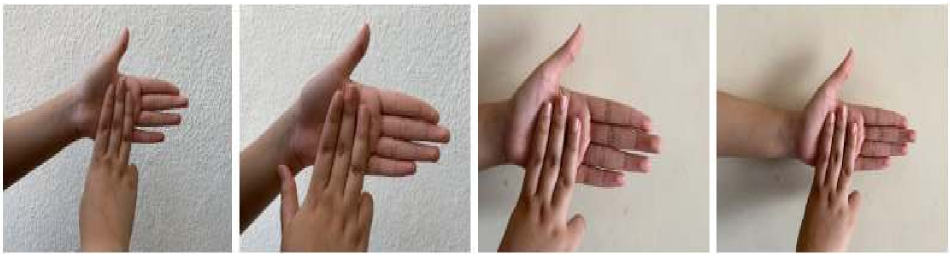


Figure 2. Images captured for the same alphabet to depict variations.

Figure 2 depicts the variation introduced within pictures of the same alphabet. The custom dataset pictures were created in different backgrounds, illumination, and angles for more significant variations in images, and a greater variation in the images means that the model can learn much better (Shorten and Khoshgoftaar, 2019). When the CNN-based models take in an image as the input, it assigns importance to the different features or objects in the image. This will help the system compare them to find the differences and similarities between the two images. Furthermore, the data was gathered during different parts of the day to capture images under various lighting situations. The other ISL alphabet databases do not possess this functionality. Therefore, our dataset solves this problem of skewness and duplicity for other researchers. CNN will improve at seeing the similarities by finding these unpolished features matches roughly the exact positions of two different images. Therefore, with a varied dataset for each alphabet, the model will be able to identify the signs and gestures with better precision. We implemented transfer learning to aid us with our smaller dataset since it prevented our model from having to learn everything from scratch and made the optimization process go faster.

Since each dataset in the area of DL includes unique properties that may be used to enhance the current models, the creation of a new dataset may be seen as a new addition to the field. The current development of the ISL dataset works on solving the problems in the existing ones. As a result, the development of a unique dataset under special constraints might be viewed as the latest contribution to the area of sign language interpretation substantially in the Indian community.

4. Methodology

DL models are mathematical and computational models that have multiple processing layers that help to accurately classify the image's efficiency. We implemented basic CNN architecture and pre-trained DL models which are already freely available and implemented like VGG-16 (Simonyan and Zisserman, 2014), VGG-19 (Simonyan and Zisserman, 2014), ResNet-50 (He et al., 2016a), DenseNet-201 (Huang et al., 2017), Inception V3 (Szegedy et al., 2016), MobileNet V2 (Howard et al., 2017), and Xception (Chollet, 2017). These models are designed specifically for image classification by researchers in the field. This process, where the model uses its prior knowledge that it was trained for using the original problem and applies that knowledge to predict the outcome of the current problem is referred to as transfer learning (Ribani and Marengoni, 2019). It is primarily used to use the stored information and leverage the previous feature representations. These models have also been utilized to help in the classification process and adopted through Transfer Learning, where we use the same architecture for a new problem.

The main steps of the procedure followed as shown in **Figure 3** are:

1) After looking at the existing datasets in the image collection phase, the need for a new dataset was immediately recognized.

2) So, using a combination of various backdrops, lighting conditions, and sign language gestures enacted by hand, images of the ISL alphabets were captured via a mobile phone camera. A

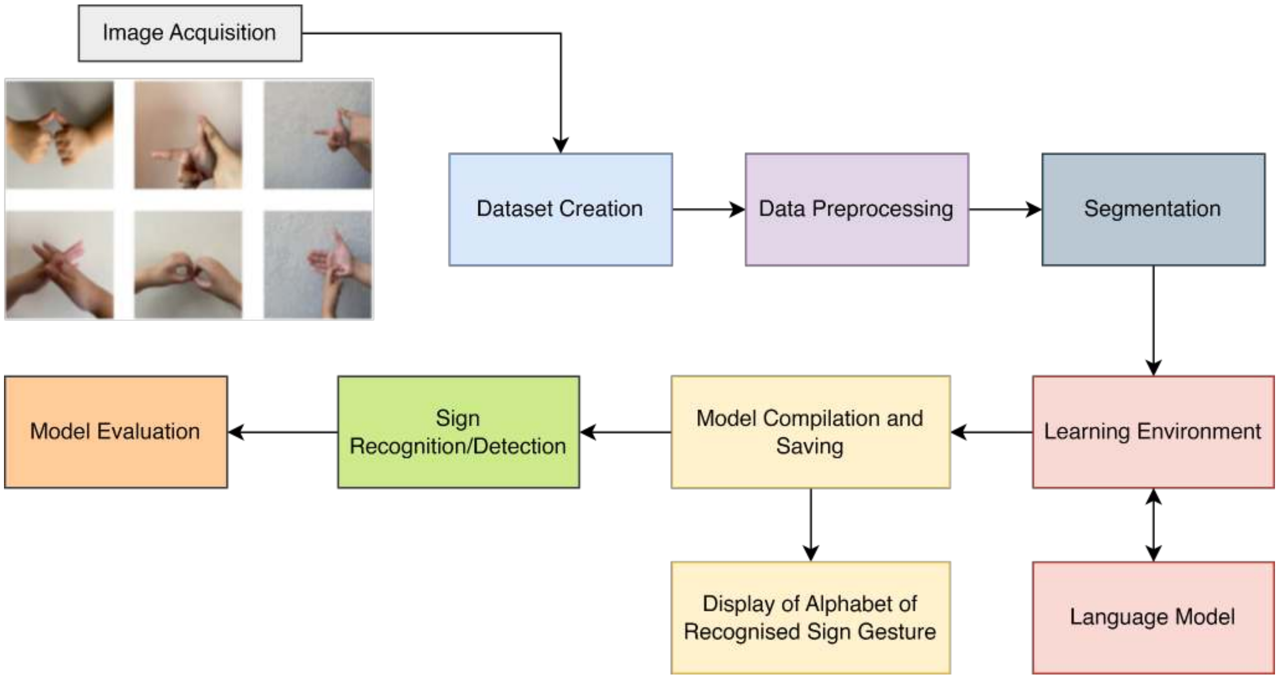


Figure 3. Process flow diagram for this study.

new dataset was created by compiling all the photos, classifying them based on their respective indicators, and condensing their size into a single dataset that comprehensively covers most of the possibilities of a sign input.

3) This step was followed by Image augmentation (Mikołajczyk and Grochowski, 2018), where the images are resized and certain elements including rotation, contrast, and tilt are changed for better training and performance.

4) The dataset is then segmented and split into training and testing portions.

5) The next step is to choose the best language model and learning environment for the given task and execute transfer learning on that model. Once the model is created and adjusted to fit the requirements, it is compiled and stored so that it will be simple to retrieve in the future.

6) Next, the model is retrained to make a prediction about the sign of the input picture. This enables us to monitor the effectiveness of our model on raw, unknown data in real-time.

7) Finally, utilizing graph representations of the model's performance in various areas, assessment measures like F1-score, accuracy, precision, and recall are employed to analyze and evaluate the model in depth. This flow is repeated every time for all the seven models (VGG-16, VGG-19, ResNet-50, DenseNet-201, Inception V3, MobileNet V2, and Xception) chosen.

8) Conclusively, to determine which model is the greatest fit for the current use case, their performances were studied, and rated from best to worst-performing models. This comparative analysis explains the rationale behind each model's performance as well as how our use of each model varies from past applications.

4.1. Image pre-processing techniques

These days, the most effective DNNs are very large, requiring huge amounts of data, which in many cases may be hard to provide. The most common issue is the lack of good-quality data or uneven class balance within the datasets. After creating our dataset, we made use of augmentation methods to increase the variety of images and effectively train our models. Our main method for enhancing images was ImageDataGenerator, which enables you to do so while your model is still training. The ImageDataGenerator class in Keras allows us to make many variations to the image before inputting it into the training epoch to increase the diversity of our images at every epoch. The randomization of the changes ensures that no two images fed to the model are alike. It is a real-time data augmentation technique that works with existing images without creating new ones. We applied rotation (in the range of 10), horizontal shifts (range = 0.2), vertical shifts (range = 0.2), zooming in and out, flipping and, brightness modifications to our dataset images. These changes made sure that our model was robust to the brightness, angle and position changes that may occur during image capturing. These images were also then resized to fit the default input image requirement of the different models tested.

4.2. CNN (convolutional neural networks)

CNN, also called ConvNet, is a neural networks class that specializes in image recognition and classification. It takes inputs and passes it through a series of layers that extract different functionalities to produce the result (Smeda, 2019). Ideally, the default input size of the image should be 224

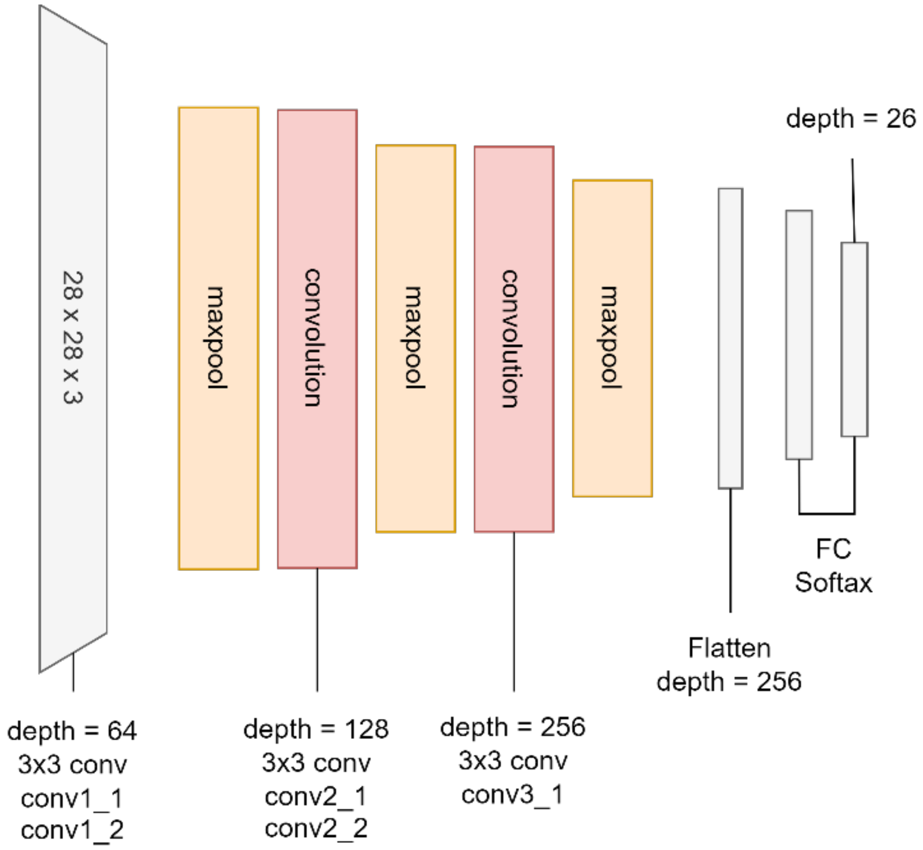


Figure 4. A general framework of CNN architecture.

× 224 × 3. Its architecture consists of four main components as shown in **Figure 4**.

1) Convolution layers: This layer applies filters or kernels on the images to extract meaningful features and produce a feature map. This process is known as feature extraction.

2) Max-pooling layers: It aids in reducing the dimensions of the feature extracted from the convolutional layer. This makes the computation processing faster.

3) Fully connected layers: This layer is also known as a dense layer. It is a feed-forward neural network that is present towards the end of the model that performs the final classification.

4) Activation function layer: It is the last component of the model and is generally a Softmax function that assigns the mathematical probabilities to classes for classification purposes (Yamashita et al., 2018).

4.3. VGG-16 (visual geometry group-16)

It was developed in 2016 by Simonyan and Zisserman at the University of Oxford. VGG-16 was one of the highest-performing models in the ILSVRC (Image Net) dataset with an accuracy of 92.7%. The 16 in VGG-Net 16 refers to the number of layers present in the model (Le, 2021). It is made of 13 convolutional layers and 3 fully connected layers as shown in **Figure 5**. It is an extensive network with about 138 million parameters. It is one of the most preferred architectures to date and is excellent in vision-based projects. The minimum image input size for this model is 224 × 224.

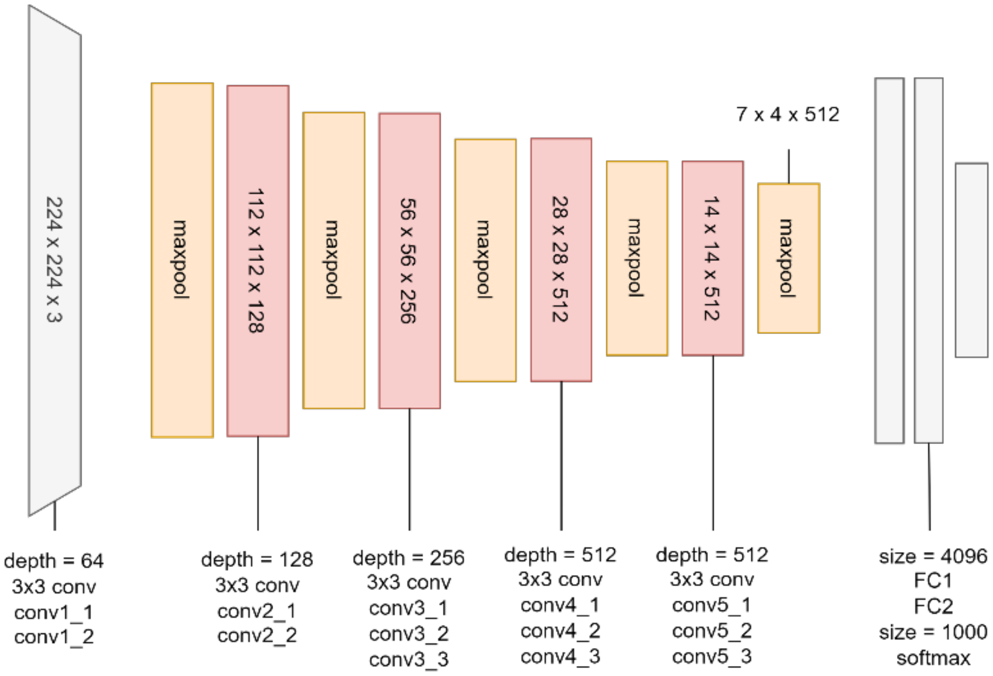


Figure 5. VGGNet-16 model architecture.

4.4. VGG-19 (visual geometry group-19)

VGG-19 is a version of the VGG model and in this, the number 19 refers to the number of layers (MathWorks UK, 2023). Here, we have 16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer as shown in **Figure 6**. VGG-19 has about 19.6 billion Floating-Point Operations per Second (FLOPs) (Sec, 2021). It can classify images into a maximum of 1000 classes or categories. Like VGG-16, this model's minimum image input size is 224 × 224.

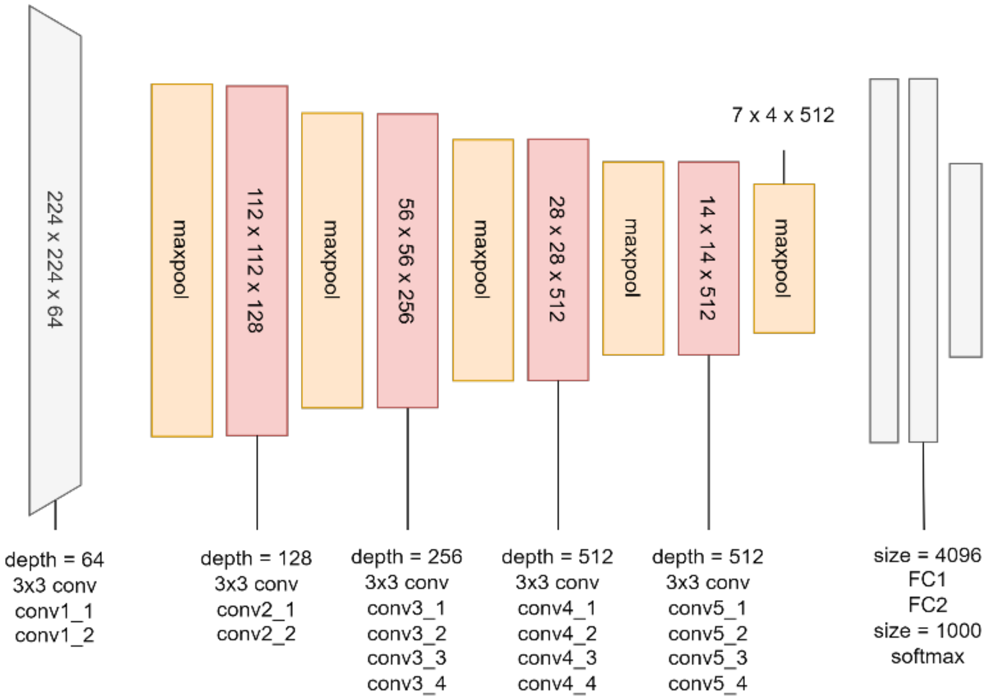


Figure 6. VGGNet-19 model architecture.

4.5. ResNet-50 (residual networks-50)

It was developed by He et al. (2016a), and introduced in their research paper on Computer Vision. ResNet-50 refers to a CNN that is 50 layers deep. It consists of 48 convolution layers, 1 Max-pooling layer, and 1 Average-pooling area as shown in **Figure 7**. It allows the training of excessively deep neural networks with over 150 layers and 23 million trainable parameters. ResNet makes use of Skip Connections, which is a direct connection and adds the original input to the output of the convolutional block (Rathi et al., 2020). The image size for ResNet-50 has to be 224 × 224 × 3.

4.6. DenseNet-201 (densely connected convolutional network)

DenseNet-201 is a 201-layer deep CNN in which each layer is linked to every other layer (Khanna, 2021). Each layer uses the feature maps from all the preceding levels as its input data. Similarly, the next layers take input from the feature maps of these layers as shown in **Figure 8**. This aims to

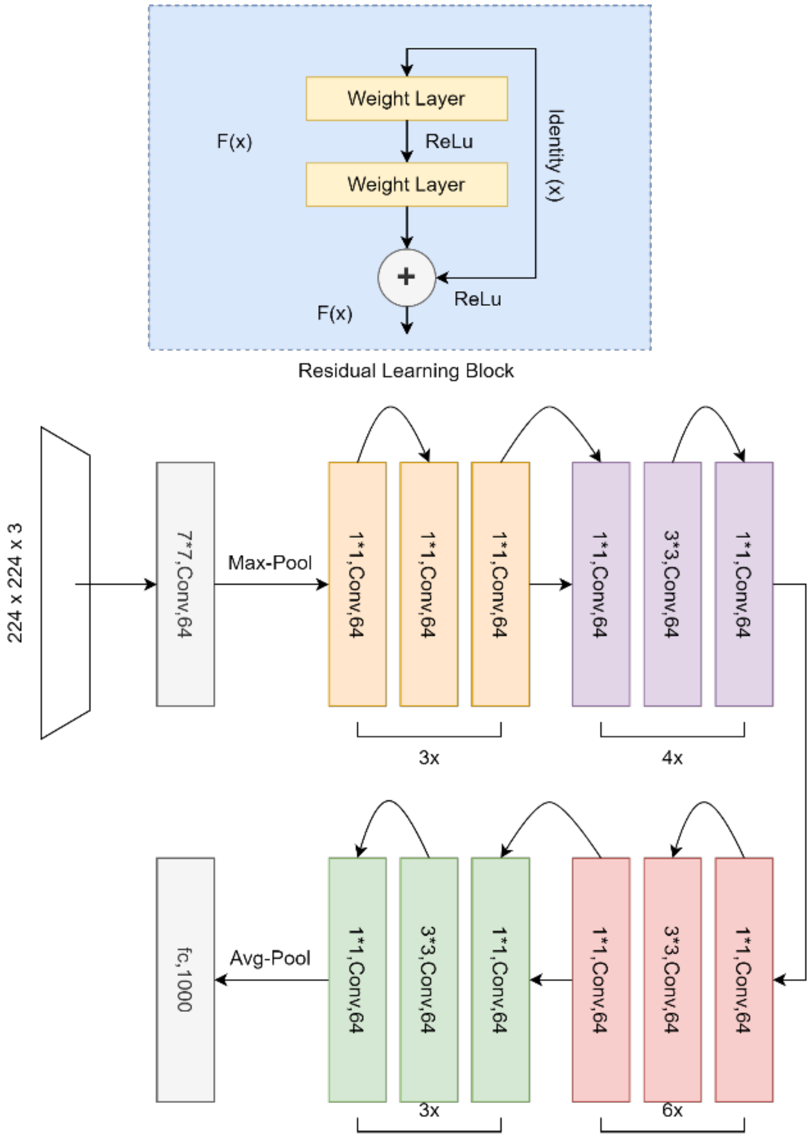


Figure 7. ResNet-50 model architecture.

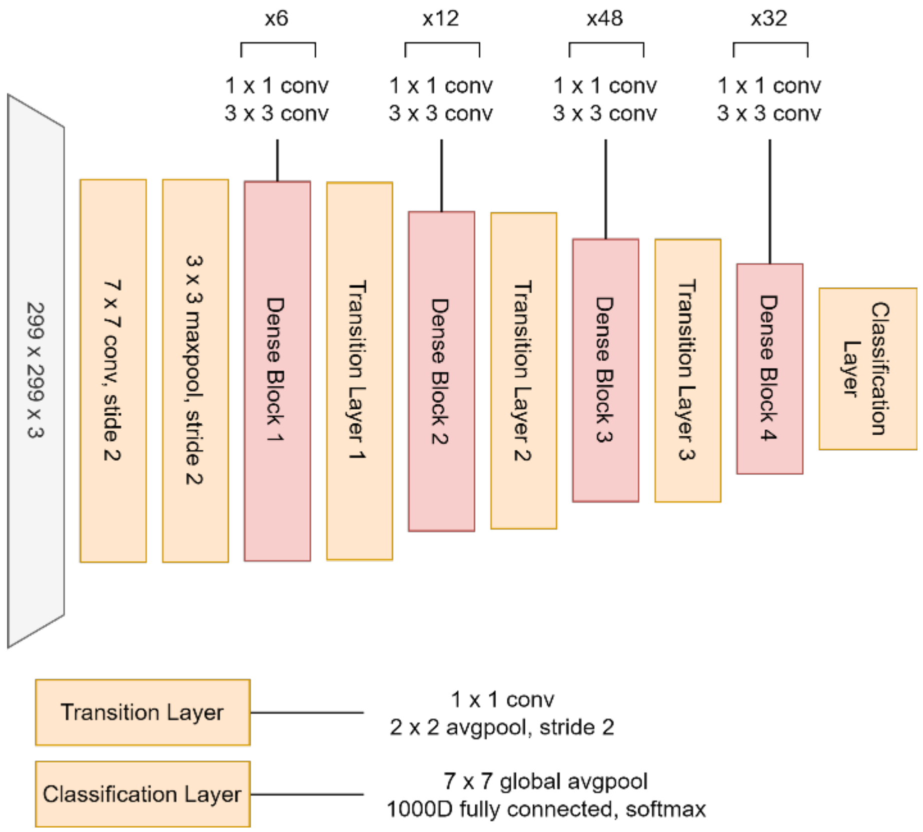


Figure 8. DenseNet model architecture.

solve the vanishing gradient problem through its dense connections that allow for better feature and gradient propagation through the circuit of pathways within the model. The default input size for this model is 224 × 224.

4.7. Inception V3

Inception V3 is a popular model for image recognition that gives 78.1% accuracy on the ImageNet dataset. It is based on symmetric and asymmetric blocks. These building blocks include convolutions, max pooling, dropouts, average pooling, concatenations, and fully connected layers (Szegedy et al., 2016). On activation inputs, throughout the model, batch normalization is used extensively. Softmax is used to compute the loss. Inception V3 includes factorized asymmetric convolutions to help reduce computational efficiency as the number of parameters reduces in the network (Metev and Veiko, 1998). There are also smaller convolutions instead of bigger ones to provide faster training. It also uses an auxiliary classifier as a regulator between layers as depicted in **Figure 9**. Grid size reduction by pooling operations helps combat computational costs. Unlike the other models, Inception V3 requires the input image to be in the size of 299 × 299.

4.8. MobileNet V2

MobileNet V2 is a depth wise separable convolution network that helps to drastically reduce the complexity, cost, and size of the network to suit mobile devices and devices with low computational power (Howard et al., 2017). MobileNet V2 introduces an inverted residual structure that helps in performing object detection and semantic segmentation. It can be performed using MobileNet V2

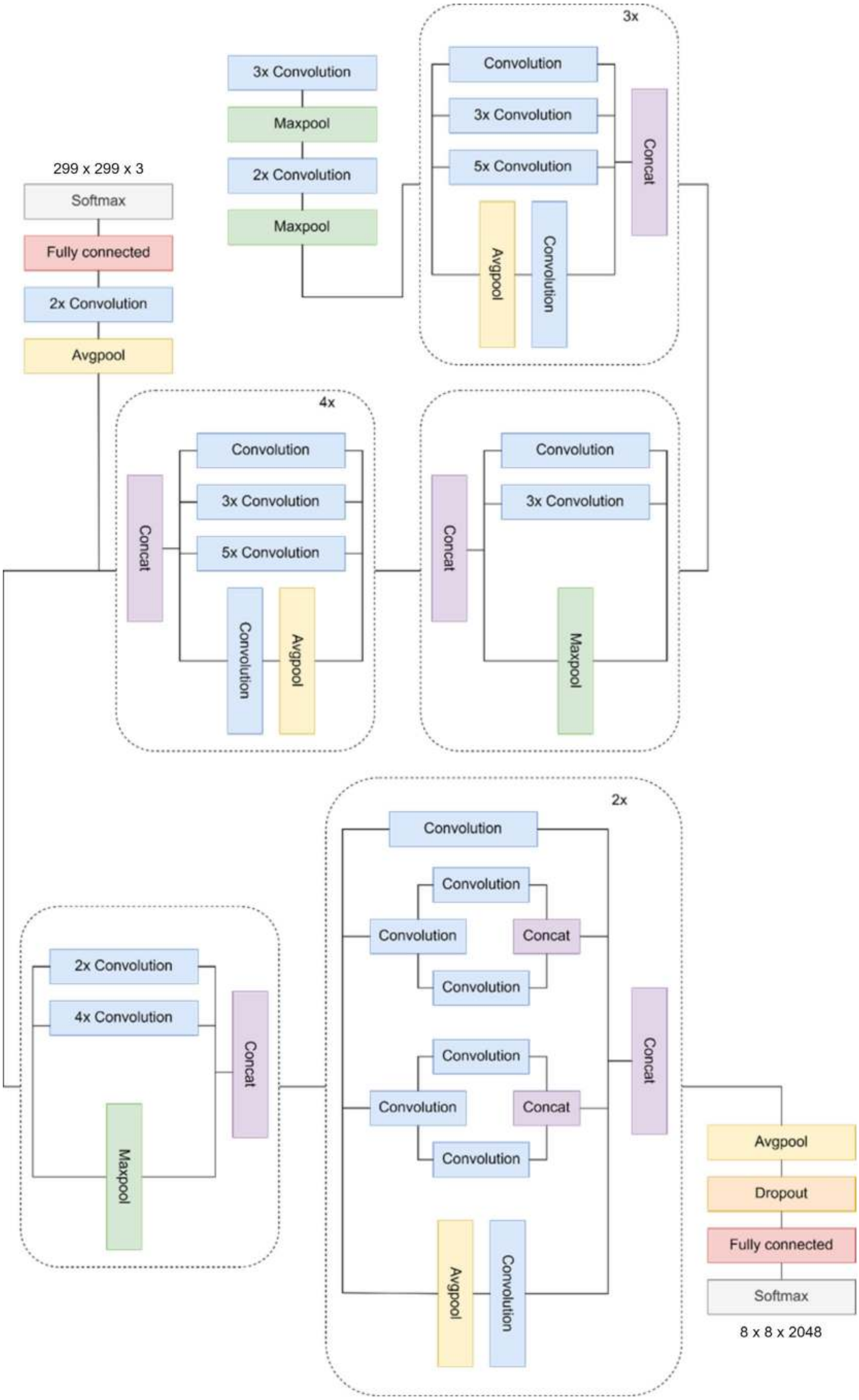


Figure 9. Inception V3 architecture.

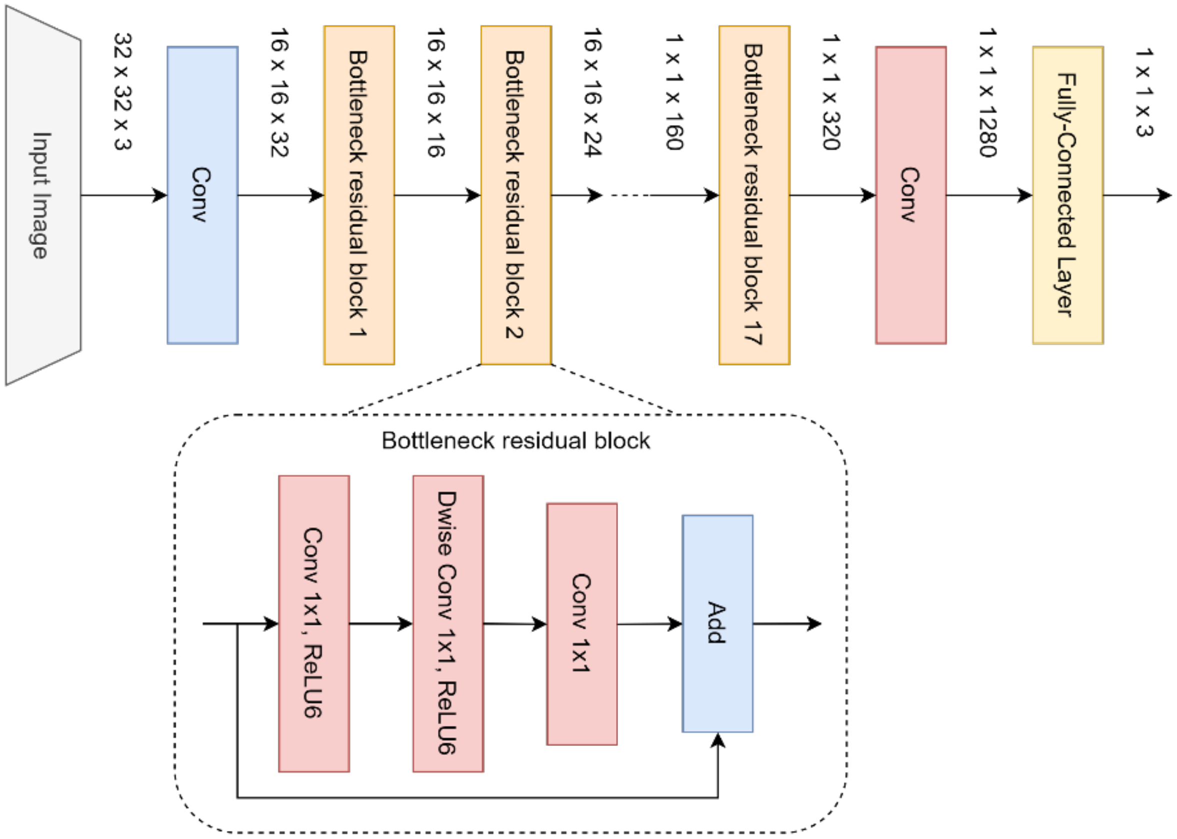


Figure 10. MobileNet V2 model architecture.

(Sandler et al., 2018). The architecture is a very intuition-led design. It consists of two blocks, the residual block with a stride of 1 and the block with a stride of two that is for downsizing. Each of these blocks has 3 layers and the first layer is a 1 × 1 convolution with RelU6. This is followed by depth-wise convolution and then finally the 1 × 1 convolution, but with no non-linearity as depicted in **Figure 10**. The input image size must be 224 × 224.

4.9. Xception

The Xception model is a CNN whose architecture involves depth-wise separable convolutions. Xception was developed by Google and was proposed by Chollet (Chollet, 2017). It has 71 layers and is an extension of the Inception model. It has an entry flow layer, followed by a middle flow layer that is repeated 8 times, and finally ends with an exit flow (Fabien, 2019). All these layers are then followed by a batch normalization layer as shown in **Figure 11**. Like Inception V3, Xception too requires the input image size to be 299 × 299.

5. Evaluation metrics

Evaluation metrics play an important role in surveying the performance of models after classification to note how well the model generalizes the unseen data and predicts accurately (Hossin and Sulaiman, 2015). One can improve the performance of their models by evaluating the model with evaluation metrics and improving the predictive power of the model before finalizing. The following metrics were taken into consideration to assess the accuracy of the obtained model outcomes.

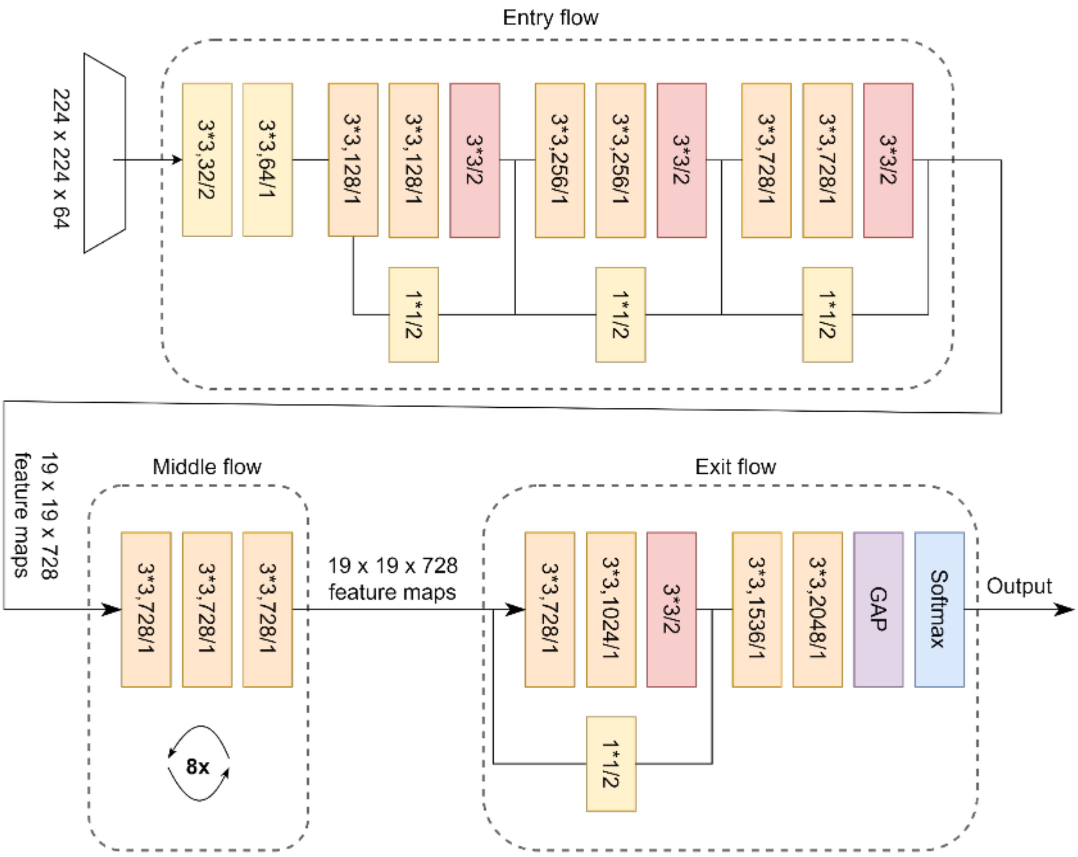


Figure 11. Xception model architecture.

5.1. Confusion matrix

A confusion matrix is a tabular structure that helps one describe the capacity of the model at hand for test data whose true values are known. In an ideal confusion matrix, there is a clear distinction between all the classes existing. Thus, it is seen that this was not a very good evaluation parameter as there are far too many classes in our model to evaluate through a confusion matrix. The confusion matrix is handy to plot as it helps visualize essential information and analytics like accuracy, precision, recall, and specificity. Other evaluation parameters like accuracy, precision, recall and F1-score gave better results in evaluating and understanding the outcomes of the proposed models. The following evaluation metrics were considered for all the models.

5.2. Accuracy

The accuracy metric gives us the performance of the proposed model across all the classes and works when classes existing are of equal importance. Accuracy can be explained (Equation (1)) as the ratio between the number of true predictions that are true by the total number of predictions.

$$Accuracy = \frac{True_{Positive} + True_{Negative}}{True_{Positive} + False_{Positive} + True_{Negative} + False_{Negative}} \tag{1}$$

5.3. Precision

This is the measure of the ratio between the number of positive samples that are correct and the number of samples of total positives classified. Precision determines the accuracy of the model in classifying an input as positive. It falls low when the model cannot accurately classify the positive samples and makes errors by giving many false positives and a few true positives. Precision (Equation (2)) soars when the model works accurately in classifying all the correct positive samples and a few false positives (Powers, 2011).

$$Precision = \frac{True_{Positive}}{True_{Positive} + False_{Positive}} \tag{2}$$

5.4. Recall

Recall provides us with the ratio of the true positives to the number of positive samples present in total, including the true positives as well as the false negatives. The recall metric allows us to measure the model's accuracy in measuring positive samples, more positive samples are discovered when the recall value is higher. Recall (Equation (3)) does not care for any of the negative samples, so even if, for instance, the model classifies all positive samples but inaccurately classifies all negative samples as positive, the recall will still be set to 100% (Goutte and Gaussier, 2005).

$$Recall = \frac{True_{Positive}}{True_{Positive} + False_{Negative}} \tag{3}$$

5.5. F1-score

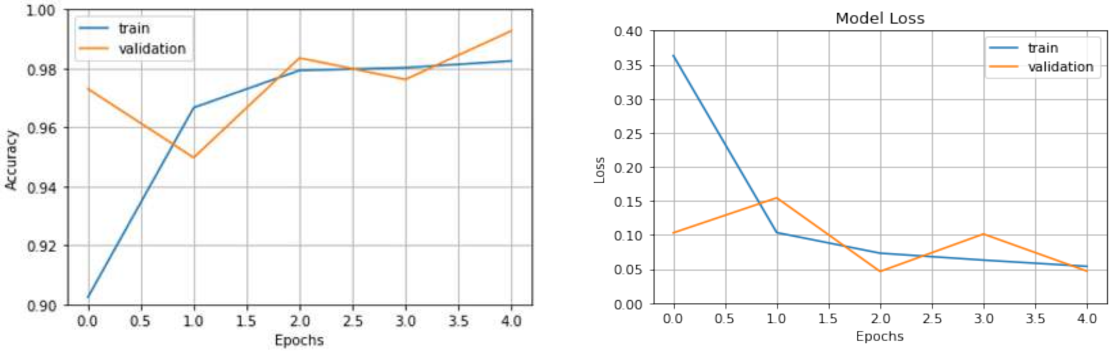
The F1-score factors in both precision and recall. The aim of the F1-score is to combine the metrics of recall & precision into one single evaluation metric. It is the harmonic mean of recall and precision. This metric also works well on imbalanced data. Computing the average of the recall and precision gives us the F1-score (Equation (4)), and since they are both rates, we use the harmonic mean (Goutte and Gaussier, 2005).

$$F1\,Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

6. Experimental results

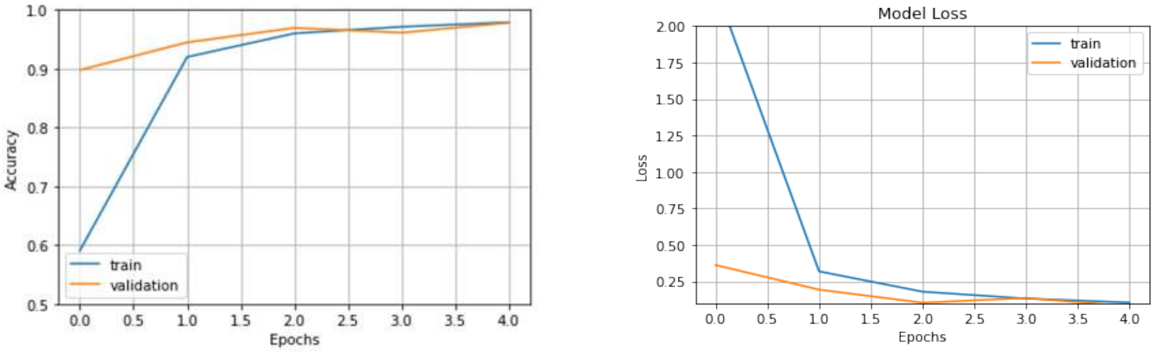
The authors evaluated the training accuracy, testing accuracy, precision, recall, F1-score and loss for each of the 7 models and compared them based on these metrics. The evaluation results highlight that the accuracy of the ResNet-50 classifier is higher than that of other classifiers in the training set and test set. It is slower in processing compared to the other classifiers, but still does not take the longest time. The exceptional 99.26% test accuracy of ResNet-50 as shown in **Figure 12**, makes it worth the wait.

The CNN from scratch has the quickest detection speed for human sign gestures, but its accuracy on the test set is only 83.00%, which may be further increased with the help of the best-performing classifiers for this job. Reduction in the processing time of a base CNN could be the result of a less extensive architecture in comparison to other state-of-the-art (SOTA) transfer learning models. The next high-performance architecture is the VGGNet-16, the second-fastest classifier that offers us



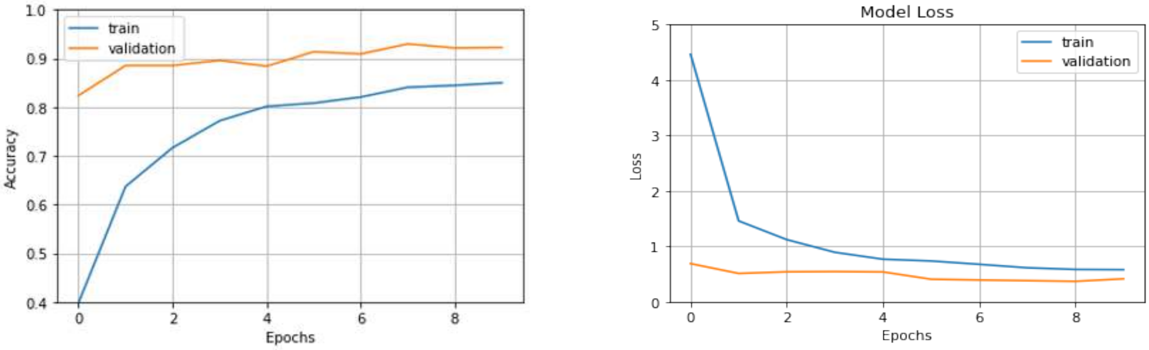
(a) (b)

Figure 12. Learning graphs of highest performing classifier—ResNet-50: (a) Accuracy; (b) Loss.



(a) (b)

Figure 13. Learning graphs of—VGGNet-16: (a) Accuracy; (b) Loss.

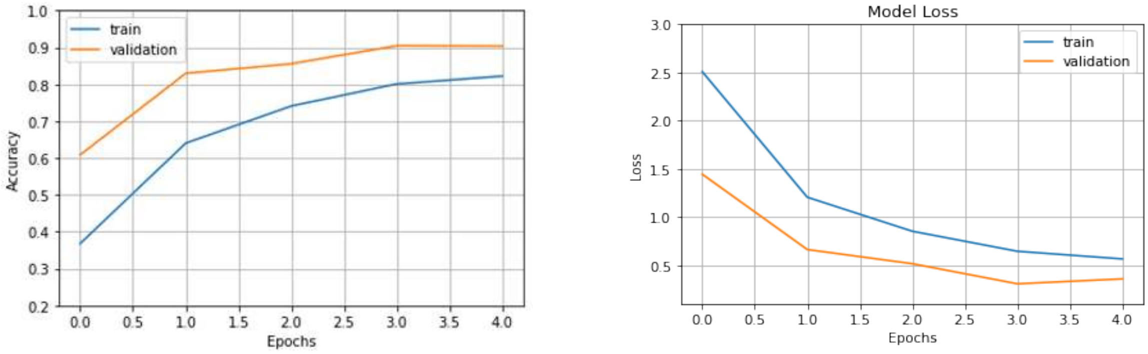


(a) (b)

Figure 14. Learning graphs of—VGGNet-19: (a) Accuracy; (b) Loss.

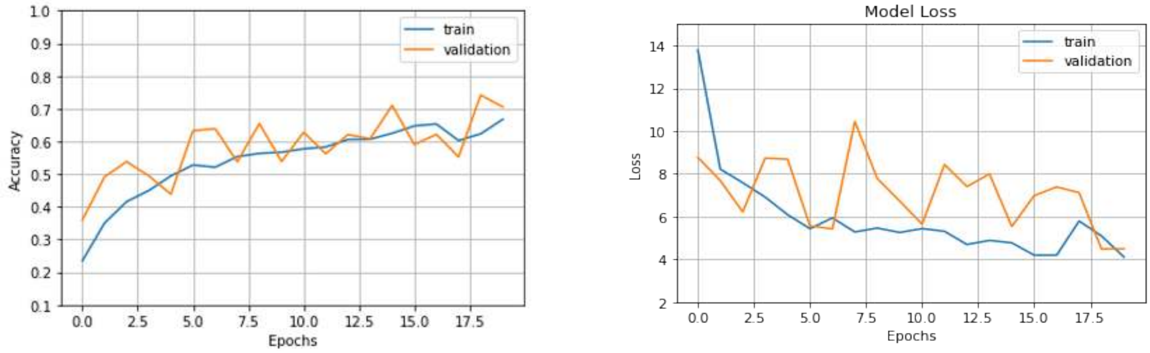
97.80% test accuracy as shown in **Figure 13**. It is closely followed by the VGGNet-19, which had the shortest or quickest execution time and a test accuracy score of 92.23% as depicted in **Figure 14**.

As shown in **Figure 15**, With an average accuracy score and a better test accuracy than the training set, the Xception model requires the most processing time to run. This suggests that it needs to be run for more epochs as well, or it could be used to explain a discrepancy in the underlying distribution that did not fairly divide the data.



(a) (b)

Figure 15. Learning graphs of the worst performing classifier—Xception: (a) Accuracy; (b) Loss.



(a) (b)

Figure 16. Learning graphs of worst performing classifier—Inception V3: (a) Accuracy; (b) Loss.

The Inception V3 learning graphs display a lot of noise and variations as exhibited in **Figure 16**, which may be crucial to the model's capacity to categorize data since it exhibits signals of either not learning the training set or of seeing it as an unrepresentative dataset. This classifier also identifies overfitting patterns since the validation loss is so much higher than the ResNet-50 projections, which are themselves much higher.

While working on higher-level applications like Sign Language Translation frameworks or Real-Time Translation Feedback from hand gestures to vocabulary, these graphs serve as an indicator of the stability of the models and help us focus on choosing the best classifiers. The detailed breakdown of various evaluation metrics for each model is highlighted in **Table 2**.

Table 2. Comparison of evaluation metrics between various deep learning frameworks

Model	Train accuracy (%)	Test accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Inception V3	66.75	70.54	71.12	70.67	70.89
DenseNet-201	93.63	80.01	82.76	78.90	80.76
Basic CNN	96.83	83.00	85.25	82.21	83.69
Xception	82.21	90.36	91.29	89.33	90.28
MobileNet V2	94.42	90.27	93.27	85.40	89.13
VGG-19 Net	84.98	92.23	92.80	91.48	92.13
VGG-16 Net	97.89	97.80	97.85	97.85	97.85
ResNet-50	98.25	99.26	98.41	99.27	99.34

Hence, the results of Table 2 reveal that out of all implemented Transfer Learning Models, ResNet-50 shows exceptional results (Test accuracy = 99.26%, Precision = 98.41%, Recall = 99.27% and F1-score = 99.34%). ResNets are unique in the sense that they find the ideal number of layers to resolve the vanishing gradient challenge and build a deeper architecture than other basic networks. ResNet is fundamentally based on Batch Normalization. The input layer is modified by the batch normalization to improve network performance. The identity mapping skips connections and acts like a gradient freeway, enabling the slope to advance without being hindered. This aids them to efficiently learn all the parameters of early activations more in-depth in the network. On the contrary, InceptionNet V3 provided us with the least impressive results (Test accuracy = 70.54%, Precision = 71.12%, Recall = 70.67% and F1-score = 70.89%), making it unfavorable for any further work to be carried out in the domain of ISL Recognition.

7. Discussion

This research entails the comparison of multiple evaluation metrics for determining the performance of various state-of-the-art CNN models utilizing Transfer Learning. For the selection of suitable models to implement, the model architecture has been factored in as well as how often they were previously deployed for the pertaining task being studied. To resolve the issues with high computational problems and overfitting, Inception V1 tries to strike a compromise between effectiveness and computing complexity by widening rather than deepening a model, along with some minor changes to Inception V1, Factorization (the process of splitting convolutions into smaller convolutions) was introduced in Inception V2. Regarding Inception V3, it is an addition of BN-auxiliary to Inception V2 (Szegedy et al., 2016). This study chose Inception V3 because it had the lowest rate of occurred error in the image classification challenge on ImageNet dataset (Sharma and Anand, 2021). An inception network uses recurring modules known as Inception modules in its architecture. The initial input is compressed using 1 × 1 convolutions, and from each of those input spaces, various types of filters are utilized in each depth space. Xception just reverses this process. The filters are instead applied to each depth map separately, and then the input space is compressed all at once using 1 × 1 convolution. The performance improvements stem from a more effective use of model parameters rather than additional capacity because the Xception architecture uses the same number of parameters as Inception V3 (Chollet, 2017). Xception was deployed for this task to examine if this hypothesis stands true for the task of ISL recognition. In MobileNet V1, the development of depthwise separable convolution makes it suitable for mobile devices or any other devices with a low processing capability. An enhanced module that has an inverted residual structure is also introduced to get rid of non-linearities in thin layers. SOTA outcomes are gained for object identification and semantic segmentation using MobileNet V2 as the base for feature extraction, making it an essential model to review for this study (Sandler et al., 2018).

In order to tackle the vanishing gradient problem, He et al. (2016a) advocated stacking residual blocks rather than a plain network and invented deep residual neural networks (ResNets). The challenge of training very deep networks has been facilitated by the introduction of residual blocks, a part of the ResNet design. ResNet-50 is a variation that operates with 50 layers of neural network. A minor adjustment was made for the ResNet-50, resolving the concerns about the duration, it would take to perform training of the layers. In this case, the building block was modified to incorporate a bottleneck design. Instead of the preceding two layers, this utilized a stack of three layers (He et al., 2016b). This version was selected for this study as ResNet-50 was chosen as the backbone network for a number of implementations (Sharma and Anand, 2021; Hussin et al., 2019). To encourage gradient propagation in ResNet, identity mapping is suggested (He et al., 2016b). DenseNet provides concatenations of all feature maps from earlier levels, meaning that all feature maps are linked to newly created feature maps and propagate to later layers. Features maps from all previous levels are sent to each layer, allowing for a more compact and thinner network with fewer channels. Similar validation errors are produced by DenseNet-201 with 20M parameters and ResNet-101 with more than 40M parameters on the ImageNet dataset (Huang et al., 2017). Hence, DenseNet was selected for this research as it offers greater memory and computational efficiency.

The VGG Network models were created with the Intention of examining how the ConvNet depth influences their precision in the context of extensive picture recognition. It demonstrates that extending the depth to 16–19 weight layers can result in a considerable advancement over the prior–art structures. These models were selected as they prove that the benefit of the representation depth for the performance on an image classification dataset, classification accuracy, can be accomplished with a standard ConvNet design with far greater depth (Simonyan and Zisserman, 2014).

The best-performing model was the ResNet-50 model. This is primarily because residual networks enable the training of deeper models while also reducing the problem of vanishing or exploding gradients. This keeps the training error rate in check and ensures the accuracy of the model does not fall. The architecture contains shortcut connections that always stay alive that help the gradients to easily backpropagate (Ebrahimi and Abadi, 2021). It enables an enhanced feature extraction capability, especially with the deep 50-layer network. The cross-layer feature fusion retains features better and achieves improved performance. ResNet works well with medium-sized datasets compared to smaller datasets as they will usually tend to overfit the data. So, with our current dataset and requirement specifications, it becomes clear why ResNet-50 is the best performing among all the other classifiers studied (Shafiq and Gu, 2022).

The VGG-16 was the second-best performing model after ResNet-50. The popular image classifier was trained on the Image Net dataset, which is a huge dataset covering a large spectrum of data. It also has remarkable feature extraction capability that is gained from the depth of the network. The VGG-16 model targets both low-level features, such as edges, corners, rotation, and high-level features that lead to the final outcome prediction (Tammina, 2019). The model enhanced their propagation into further layers. This enabled the model to predict the signs more accurately than the other models and produce good training and validation accuracy. The VGG-19 model follows close behind and has a similar architecture to that of the VGG-16 model. It is deeper than the VGG-16 model and has 3 additional convolutional layers in the neural network. This makes the network a little more time-consuming to train, requiring additional time. The performance metrics, however, are almost the same and hence, it can be ranked parallel to the VGG-16 network (Mascarenhas and Agarwal, 2021).

The MobileNet V2 model is an improved version of the previous MobileNet model introduced. It uses the technique of Inverted Residuals and Depthwise Separate Convolutions (DSC) to preserve the information between the various layers and to address the vanishing gradient problem (Dong et al., 2020). The performance of this model was fair, and there were no exceptional gains or losses we encountered during its implementation. The architecture of the Xception model is also built on a similar note using DSCs and produces a similar accuracy (Khanna, 2021).

On the other hand, the worst-performing model was the Inception V3 model. The Inception V3 model only produced an accuracy of 78.1% on the Image Net dataset as well. The basic pre-processing of the dataset is not sufficient for the Inception V3 model and for it to reach significantly good accuracy, it needs to be trained for at least 180–200 epochs (Google Cloud, 2023). This is very computationally intensive and is not within the scope of this research. The other models gave much better results in much less time without these extra steps. Ultimately, one can infer that this model was not a suitable fit for the problem statement and methodology.

We can see a steady rise in accuracy from the Inception V3 model to the ResNet-50 model. The main cause of this is the gradual increase in the neural network's complexity and the incorporation of backpropagation of the residual layers and weights in the ResNet-50 model. From top to bottom, the focus shifts from computational costs to computational accuracy, leading to higher training times but overall better results. The number of features produced per image also differs from about 1536 in Inception and 2048 in ResNet (McNeely-White et al., 2019) which leads to significantly better recognition and classification of images in our use case, which has several similar looking classes and needs a very precise model to identify them correctly. We can also intuitively infer that the deeper the model, the better it performs in case of an optimized and approach with a large enough dataset as the model learns much more.

One can compare the obtained results with the ones derived from previous studies such as in the case of Pigou et al. (2014), which received a validation accuracy of 91.70% for their best model. The accuracy of their test set was observed to be 95.68% with a 4.13% false positive rate. It was also observed that due to the lack of background and people in the validation set, the test result was higher than the validation result. The accuracy of the best model was 98.25% for train accuracy, and 99.26% for test accuracy. Abiyev et al. (2020) used the pre-trained model Inception V3 which outperformed others with an accuracy of 99% in contrast to our work in which Inception V3 stands as the worst performing model with an accuracy of 70.54%. In Sharma and Singh (2021) and Abiyev et al. (2020), ResNet152V2 outperforms other pre-trained models with a recognition accuracy of 96.18% on the ISL dataset Numerical subset and 90.84% on the ISL dataset alphabet subset. Inception V3 performed the least in the study, hence drawing multiple similarities between their results and ours. In another work, Sharma and Singh (2021) used VGG-11 and VGG-16 to derive an accuracy of 99.96% and 100% for ISL Dataset and ASL Dataset, respectively. In this work, we have implemented VGG-19 Net and VGG-16 which gave an accuracy of 92.23% and 97.80%, respectively. VGG-16 provided the second highest accuracy after ResNet-50.

8. Conclusion

Sign language is a way of communicating using hand gestures and movements, body language and facial expressions instead of spoken words. The development of sign language recognition systems is now a focus of research, but there are a number of challenges that must be overcome before they can be used in real-world applications such as the need to recognize gestures and hand postures. Additionally, there is a limited amount of research work done surrounding ISL, which aids a large community across the country. This study presents a new comprehensive dataset of the ISL and focuses on deriving a discernible comparison between the different DL frameworks. Letters are the foundations of any language; hence, our dataset currently supports alphabets of ISL, which can further be refined by the addition of numerical figures as well as common phrases.

This research implemented several Pre-trained Transfer Learning Models that have yielded promising results in the past on the ImageNet database. A pre-trained model gives a very beneficial starting point despite having been trained on a task other than the one at hand since the traits learned during training on the prior iterations are applicable to the current task. For the task of ISL recognition, ResNet-50 (Accuracy = 98.25%, Loss = 0.046) and VGG-16 Net (Accuracy = 97.89%, Loss = 0.078) exhibited reasonable results. The efficiency of these models can be taken into account as building blocks for a CNN of superior performance, deploying the lucrative features of the models that achieve optimal results. Furthermore, this study can also be utilized for designing frameworks that translate sign language to speech using a text-to-speech mechanism, live translation or caption generation for real-time video call connectivity and other such applications.

Acknowledgements

The study is supported by the Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), University of Technology Sydney.

Conflict of interest

No conflict of interest was reported by all authors.

References

Abiyev RH, Arslan M, and Idoko J (2020) Sign language translation using deep convolutional neural networks. *KSII Transactions on Internet and Information Systems* 14(2). DOI: 10.3837/tiis.2020.02.009

Adeyanju IA, Bello OO, and Adegboye MA (2021) Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications* 12: 200056. DOI: 10.1016/j.iswa.2021.200056

Adithya V and Rajesh R (2020) A deep convolutional neural network approach for static hand gesture recognition. *Procedia Computer Science* 171: 2353–2361. DOI: 10.1016/j.procs.2020.04.255

Alzubaidi L, Zhang J, Humaidi AJ, et al. (2021) Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8(53). DOI: 10.1186/s40537-021-00444-8

Arikeri P (2021) *Indian Sign Language (ISL)* [online]. Available at: https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl

Bansal D, Chhikara R, Khanna K, and Gupta P (2018) Comparative analysis of various machine learning algorithms for detecting dementia. *Procedia Computer Science* 132: 1497–1502. DOI: 10.1016/j.procs.2018.05.102

Barbhuiya AA, Karsh RK, and Jain R (2021) CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications* 80(2): 3051–3069. DOI: 10.1007/s11042-020-09829-y

Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, 21–26 July 2017, pp.1800–1807. New York: IEEE.

Correll R (2022) *Challenges That Still Exist for the Deaf Community* [online]. Available at: https://www.verywellhealth.com/what-challenges-still-exist-for-the-deaf-community-4153447 (Accessed: 28 July 2022).

Deshmukh D (1997) *Sign Language and Bilingualism in Deaf Education* [online]. Ichalkaranji: Hunda Infotech. Available at: https://bilingualism.in/

Dong K, Zhou C, Ruan Y, and Li Y (2020) MobileNet V2 model for image classification. In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China, 18–20 December 2020, pp.476–480. New York: IEEE.

Dumbre A (2022) *Indian Sign Language (ISLRTC referred)* [online]. Available at: https://www.kaggle.com/datasets/atharvadumbre/indian-sign-language-islrtc-referred

Dutta S, Manideep BCS, Rai S, and Vijayarajan V (2017) A comparative study of deep learning models for medical image classification. *IOP Conference Series: Materials Science and Engineering* 263(4): 042097. DOI: 10.1088/1757-899x/263/4/042097

Ebrahimi MS and Abadi HK (2021) Study of residual networks for image recognition. In: Arai K (ed.) *Intelligent Computing*. Berlin: Springer, pp.754–763.

Elakkiya R and Natarajan B (2021) ISL-CSLTR: Indian sign language dataset for continuous sign language translation and recognition. *Mendeley Data* 1. DOI: 10.17632/kcmpdxky7p.1

Ethnologue (2022) *Sign Language* [online]. Available at: https://www.ethnologue.com/subgroup/2/

Fabien M (2019) *XCeption Model and Depthwise Separable Convolutions* [online]. Available at: https://maelfabien.github.io/deeplearning/xception/#ii-in-keras (Accessed: 31 August 2022).

Fable (2022) *What Is Sign Language*? [online]. Available at: https://makeitfable.com/glossary-term/sign-language/

Garcia B and Viesca SA (2016) Real-time American sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition* 2: 225–232.

Google Cloud (2023) *Advanced Guide to Inception V3* [online]. Available at: https://cloud.google.com/tpu/docs/inception-v3-advanced

Goutte C and Gaussier E (2005) A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: Losada DE and Fernández-Luna JM (eds.) *ECIR 2005: Advances in Information Retrieval*. Berlin: Springer Berlin Heidelberg, pp.345–359.

Goyal C (2021) *20 Questions to Test Your Skills on CNN (Convolutional Neural Networks)* [online]. Available at: https://www.analyticsvidhya.com/blog/2021/05/20-questions-to-test-your-skills-on-cnn-convolutional-neural-networks/ (Accessed: 18 August 2022).

Gu J, Wang Z, Kuen J, et al. (2018) Recent advances in convolutional neural networks. *Pattern Recognition* 77: 354–377. DOI: 10.1016/j.patcog.2017.10.013

He K, Zhang X, Ren S, and Sun J (2016a) Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 27–30 June 2016, pp.770–778. New York: IEEE.

He K, Zhang X, Ren S, and Sun J (2016b) Identity mappings in deep residual networks. In: Leibe B, Matas J, Sebe N, and Welling M (eds.) *Computer Version—ECCV 2016*. Berlin: Springer, pp.630–645.

Heath N (2020) *What is Machine Learning? Everything You Need to Know* [online]. Available at: https://www.zdnet.com/article/what-is-machine-learning-everything-you-need-to-know (Accessed: 23 August 2022).

Hossin M and Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process* 5(2): 1–11. DOI: 10.5121/ijdkp.2015.5201

Howard AG, Zhu M, Chen B, et al. (2017) *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* [online]. Available at: https://arxiv.org/abs/1704.04861

Huang G, Liu Z, Van Der Maaten L, and Weinberger KQ (2017) Densely connected convolutional networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, 21–26 July 2017, pp.2261–2269. New York: IEEE.

Hussin S, Elashek K, and Yildrim R (2019) Convolutional neural network baseline modelbuilding for person re-identification. In: Saritas I, Cunkas M, and Basciftci F (eds.) *Conference: International Conference on Engineering Technologies (ICENTE'19),* Konya, Turkey, 25–27 October 2019, pp.53–57. Meram: SN Bilgi Teknolojileri.

Khan Z (2017) *How do We Speak with the 18 Million Indians Who Are Deaf*? [online] Available at: https://www.wionews.com/south-asia/how-do-we-speak-with-the-18-million-indians-who-are-deaf-18835 (Accessed: 3 August 2022).

Khanna M (2021) *Paper Review: DenseNet-Densely Connected Convolutional Networks* [online]. Available at: https://towardsdatascience.com/paper-review-densenet-densely-connected-convolutional-networks-acf9065dfefb (Accessed: 29 July 2022).

Kumar K (2022) *Indian Sign Language Dataset* [online]. Available at: https://www.kaggle.com/datasets/kshitij192/isl-dataset

Le K (2021) *An Overview of VGG16 and NiN Models* [online]. Available at: https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484 (Accessed: 22 June 2022).

Mandke K and Chandekar P (2019) Deaf education in India. In: Knoors H, Brons M, and Marschark M (eds.) *Deaf Education Beyond the Western World: Context, Challenges, and Prospects, Perspectives on Deafness*. Oxford: Oxford University Press, pp.261–284.

Mascarenhas S and Agarwal M (2021) A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification. In: *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India, 19–21 November 2021, pp.96–99. New York: IEEE.

MathWorks UK (2023) *VGG-19 Convolutional Neural Network* [online]. Available at: https://uk.mathworks.com/help/deeplearning/ref/vgg19.html

Metev SM and Veiko VP (1998) *Laser-Assisted Microtechnology.* Berlin: Springer Berlin Heidelberg.

McNeely-White DG, Beveridge JR, and Draper BA (2019) Inception and ResNet: Same training, same features. In: Samsonovich AV (ed.) *Biologically Inspired Cognitive Architectures*. Berlin: Springer, pp.352–357.

Mikołajczyk A and Grochowski M (2018) Data augmentation for improving deep learning in image classification problem. In: *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, Świnoujście, Poland, 9–12 May 2018, pp.117–122. New York: IEEE.

Mitter S (2017) *This Country Is Developing Its Own Sign Language Dictionary* [online]. Available at: https://mashable.com/article/india-sign-language-dictionary (Accessed: 23 August 2022).

Pigou L, Dieleman S, Kindermans P, and Schrauwen B (2014) Sign language recognition using convolutional neural networks. In: Agapito L, Bronstein MM, and Rother C (eds.) *Computer Vision—ECCV 2014 Workshops*. Berlin: Springer, pp.572–578.

Powers DMW (2011) Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies* 2(1): 37–63.

Rajalakshmi E, Elakkiya R, Prikhodko AL, et al. (2022) Static and dynamic isolated Indian and Russian sign language recognition with spatial and temporal feature detection using hybrid neural network. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22(1): 1–23. DOI: 10.1145/3530989

Rathi P, Gupta RK, Agarwal S, and Shukla A (2020) Sign language recognition using ResNet50 deep neural network architecture. *Social Science Research Network.* DOI: 10.2139/ssrn.3545064

Ribani R and Marengoni M (2019) A survey of transfer learning for convolutional neural networks. In: *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, Rio de Janeiro, Brazil, 28–31 October 2019, pp.47–57. New York: IEEE.

Sandler M, Howard AW, Zhu M, et al. (2018) MobileNet V2: Inverted residuals and linear bottlenecks. In: *2018*

IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, USA, 18–23 June In: 2018, pp.4510–4520. New York: IEEE.

Sec I (2021) VGG-19 convolutional neural network. In: *Machine Learning Blog*. Available at: https://blog.techcraft.org/vgg-19-convolutional-neural-network/ (Accessed: 22 June 2022).

Shafiq M and Gu Z (2022) Deep residual learning for image recognition: A survey. *Applied Sciences* 12(18): 8972. DOI: 10.3390/app12188972

Sharma P and Anand RS (2021) A comprehensive evaluation of deep models and optimizers for Indian sign language recognition. *Graphics and Visual Computing* 5: 200032. DOI: 10.1016/j.gvc.2021.200032

Sharma S and Singh S (2021) Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Systems with Applications* 182: 115657. DOI: 10.1016/j.eswa.2021.115657

Shorten C and Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *Journal of Big Data* 6(1). DOI: 10.1186/s40537-019-0197-0

Simonyan K and Zisserman A (2014) *Very Deep Convolutional Networks for Large-scale Image Recognition* [online]. Available at: https://arxiv.org/abs/1409.1556

Smeda K (2019) *Understand the Architecture of CNN* [online]. Available at: https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7 (Accessed: 23 June 2022).

Sonawane V (2020) *Indian Sign Language Dataset* [online]. Available at: https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset

Szegedy C, Vanhoucke V, Ioffe S, et al. (2016) Rethinking the inception architecture for computer vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 27–30 June 2016, pp.2818–2826. New York: IEEE.

Tammina S (2019) Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications* 9(10): 9420. DOI: 10.29322/ijsrp.9.10.2019.p9420

Vasishta M, Woodward JC, and Wilson KL (1978) Sign language in India: Regional variation within the deaf population. *Indian Journal of Applied Linguistics* 4(2): 66–74.

Wikipedia (2023) *History of Sign Language* [online]. Available at: https://en.wikipedia.org/wiki/History_of_sign_language

World Health Organization (2019) *WHO-ITU Standard Aims to Prevent Hearing Loss among 1.1 Billion Young People* [online]. Available at: https://www.who.int/news/item/12-02-2019-new-who-itu-standard-aims-to-prevent-hearing-loss-among-1.1-billion-young-people (Accessed: 19 August 2022).

World Health Organization (2023) *Deafness and Hearing Loss* [online]. Available at: https://www.who.int/newsroom/fact-sheets/detail/deafness-and-hearing-loss

Yamashita R, Nishio M, Do RKG, et al. (2018) Convolutional neural networks: An overview and application in radiology. *Insights into Imaging* 9(4): 611–629. DOI: 10.1007/s13244-018-0639-9

Yao G, Lei T, and Zhong J (2019) A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters* 118: 14–22. DOI: 10.1016/j.patrec.2018.05.018