UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

Web application to learn sign language with deep learning

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor:

Bryan Eduardo Jami Jami

Tutor:

Manuel Eugenio Morocho Cayamcela, Ph.D.

Urcuquí, noviembre de 2023

Autoría

Yo, **Bryan Eduardo Jami Jami**, con cédula de identidad 1724162480, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, noviembre de 2023.

Bryan Eduardo Jami Jami

CI: 1724162480

Autorización de publicación

Yo, **Bryan Eduardo Jami Jami**, con cédula de identidad 1724162480, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, noviembre de 2023.

Firmado electrónicamente por:
BRYAN EDUARDO JAMI JAMI

Bryan Eduardo Jami Jami

CI: 1724162480

Dedication

To my family and my friends who support and inspire me to keep trying to achieve my goals.

Bryan Eduardo Jami Jami

Acknowledgment

I wholeheartedly thank my family for giving me that unconditional support, and they motivated me to complete this last project of my academic time at the university. To my professors who, besides teaching me academically, gave me their support and patience during this stage of my life. And to my friends who encouraged me in this long process. A special thanks to my sister Alison, who always kept me focused.

Bryan Eduardo Jami Jami

Resumen

El aprendizaje profundo y la visión por computadora se utilizan para crear aplicaciones que faciliten una mejor interacción entre humanos y máquinas. En el ´ámbito educativo, obtener información sobre el lenguaje de señas es sencillo, pero encontrar una plataforma que permita una interacción intuitiva es todo un desafío. Se ha desarrollado una aplicación web para abordar este problema mediante el empleo de aprendizaje profundo para ayudar a los usuarios a aprender el lenguaje de señas. En este estudio, se probaron dos modelos de reconocimiento de gestos con las manos, utilizando 20.800 imágenes; Los modelos probados fueron AlexNet y GoogLeNet. Durante el entrenamiento de estos modelos se ha considerado el problema de sobreajuste que se encuentra en las redes neuronales convolucionales. En este estudio se han empleado varias técnicas para minimizar el sobreajuste y mejorar la precisión general. AlexNet logró una tasa de precisión del 87% al interpretar gestos con las manos, mientras que GoogLeNet logró una tasa de precisión del 85%. Estos resultados se incorporaron a la aplicación web, cuyo objetivo es enseñar el alfabeto de la lengua de signos estadounidense de forma intuitiva.

Palabras Clave:
Aprendizaje profundo, visión computacional, reconocimiento de gestos, educacion, lenguaje de señas, clasificación de imágenes

Abstract

Deep learning and computer vision are used to create applications that facilitate a better interaction between humans and machines. In the educational domain, obtaining information about sign language is simple, but finding a platform that allows for intuitive interaction is quite challenging. A web app has been developed to address this issue by employing deep learning to assist users in learning sign language. In this study, two models for hand-gesture recognition were tested, utilizing 20,800 images; the models tested were Alexnet and GoogLeNet. The overfitting problem encountered in convolutional neural networks has been considered while training these models. Several techniques to minimize the overfitting and improve the overall accuracy have been employed in this study. AlexNet achieved an 87% of accuracy rate when interpreting hand gestures whereas GoogLeNet achieved an 85% accuracy rate. These results were incorporated into the web app, which aims to teach the alphabet of American sign language intuitively.

Keywords:
Deep learning, computer vision, gesture recognition, education, sign language, image classification

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Deep learning (DL) is a subset of machine learning (ML) that uses algorithms inspired by the brain's ability to learn. DL algorithms can learn from unstructured or unlabeled data, making them well suited for image recognition and natural language processing tasks. However, it could present a problem of overfitting that consists in learning the training data too well and not generalizing to new data. Overfitting appears when a model trained with little data learns the noise instead of the signal. This happens because the model tries to know the data set too deeply. The model attempts to find a pattern in the data set that does not exist.

In recent years, ML has been a powerful tool for detecting and classifying signals in the presence of noise. In particular, neural networks are successful in this task [1]. In the daily day, ML is used in various applications such as facial and speech recognition, handwriting recognition, and machine translation.

ML has many applications that help reduce problems in every field, such as medicine, finance, media, etc. There are many different types of neural networks, each with advantages and disadvantages. In general, however, neural networks are well-suited for tasks that are too difficult for traditional methods, such as pattern recognition and classification.

Sign language (SL) is a natural language with grammar, syntax, and vocabulary. It is not a code or pidgin. SL is used by people who are deaf or hard of hearing. It is a visual language that uses hand gestures, body language, and facial expressions to communicate. SL is not a universal language, and each country has its SL. The American Sign Language (ASL) is the most popular SL known [2]. Indeed, most countries use ASL characteristics

to create their SL.

It makes it really hard to practice if you do not have another person practicing ASL in your life. There are a few ways to learn ASL online. One way is to find a website that offers lessons, such as ASL University. Another way is to find a video tutorial on YouTube. But the best and most effective way to learn ASL is practicing every day with another person who knows the language and is willing to help you learn.

1.1 Problem Statement

DL is a powerful tool that can automatically extract features from data. However, The model should be observed when it displays values "too well" because this could be a symptom of overfitting. Techniques like the regularization, dropout, and weight decay are used to reduce overfitting,

Most systems that recognize hand gestures apply computer vision (CV) to get the information, but the Microsoft Kinect device is the primary tool. However, this tool is unavailable to every possible user. The standard system that everyone can use is using tools present in at least every laptop, like a webcam. The system needs to be available for all, and one solution is to build a progressive web app that lets people access and use the proposed app.

Progressive web apps are web applications that use modern web capabilities to deliver a user experience similar to that of mobile apps. They are designed to be responsive to the device of the user and network conditions, making them reliable and fast, even on slow or unreliable connections.

The Covid-19 pandemic has had a significant impact on the education sector. Many schools and universities have had to close their doors, and students have had to learn remotely. The pandemic has also put a strain on educators, who have had to adapt to new teaching methods and technologies, finding some challenges and opportunities. One of the biggest challenges facing educators is the digital divide. Many students do not have access to the internet or a computer at home, making it difficult for them to participate in online learning. Schools and universities have had to provide devices and internet access to those who need it, but it is not always possible. The pandemic has also forced educators

to rethink the way they teach. Many have had to rely on technology to deliver lectures and assignments, which has been a learning curve for teachers and students. [3] But it has also opened up new possibilities for how education can be delivered.

A web app platform was built that allows learning ASL online using the DL model, implementing the new technologies, and the experience obtained in the last pandemic. That makes the knowledge easier for everyone, even if the user doesn't have a good internet connection.

1.2 Objectives

1.2.1 General Objective

Reduce overfitting in a deep neural network image classifier that recognizes ASL based on the posture of the hand.

1.2.2 Specific Objectives

- Design a model using regularization techniques that learn to classify the different hand-pose from the ASL dataset.
- Validate the overfitting reduction of our model by comparing the performance with previous architectures.
- Build a web platform that enables people to learn ASL using our optimized architecture.

1.2.3 Contribution

The investigation gives us a model that performs a good ASL classification with an overfitting reduced, and this model is implemented in a web app open to every user who wants to study ASL online. The model was trained using a solid database and implementing good techniques to avoid overfitting.

Chapter 2

Theoretical Framework

The chapter explores SLR, delves into the principles of Convolutional Neural Networks, and discusses the unique attributes of AlexNet and GoogLeNet in the context of computer vision and sign language recognition.

2.1 Sign Language

According to the World Health Organization, about 466 million people, adults and children, have disabling hearing loss caused by various factors such as genetics, chronic infections, trauma, work-related chemicals, or loud sounds, among others [4]. There are communication barriers for this group of people, who use SL as their primary means of communication. However, interaction with those who cannot understand this language is a limitation in daily life [5].

SL is a type of visual communication where information flows through multiple optical channels, such as hand gestures, body postures, and facial expressions [6]. It is the art of spreading ideas and emotions non-verbally [7].

According to the World Federation of the Deaf, it is estimated that around the world, about 300 variations of SL are used [7], and they are composed of gestures of 1 or 2 hands representing words, alphabetic characters, and numbers [5]. Likewise, each SL varies in its execution according to region, age, and level of hearing impairment [7].

2.1.1 American Sign Language

ASL is an EEUU language used as communication media for people who have hearing and speech problems. It is expressed through hand movement and gestures [2]. The use of the hands and the movements oriented contains essential information. Also, It includes body and face movements as part of the communication that complements the expression [8].

The origin of ASL is unclear, but its analysis said it has been arising with the merging of several SLs. One of the most influential is the Langue des signes Frnaçaise (LSF), which has similar symbols, although they are different languages [2].

Like any language, ASL is structured with its standards and norms, which may differ from other SLs since they may contain similar symbols but have different interpretations [9] due to their dialect and place of origin [10]. As other languages use different tonalities to express exclamation or question marks, ASL users can express their ideas as questions or statements by changing their gestures [2].

As Figure 2.1 shows. ASL contains 26 letters, expressed with the movement of the hands, and there are 19 forms in different orientations of the hands that help make up this alphabetical group. These expressions aim to symbolize different words from the English dictionary, some of the hand shapes being the same but allowing another idea to be conveyed simply by changing the orientation of these [11].

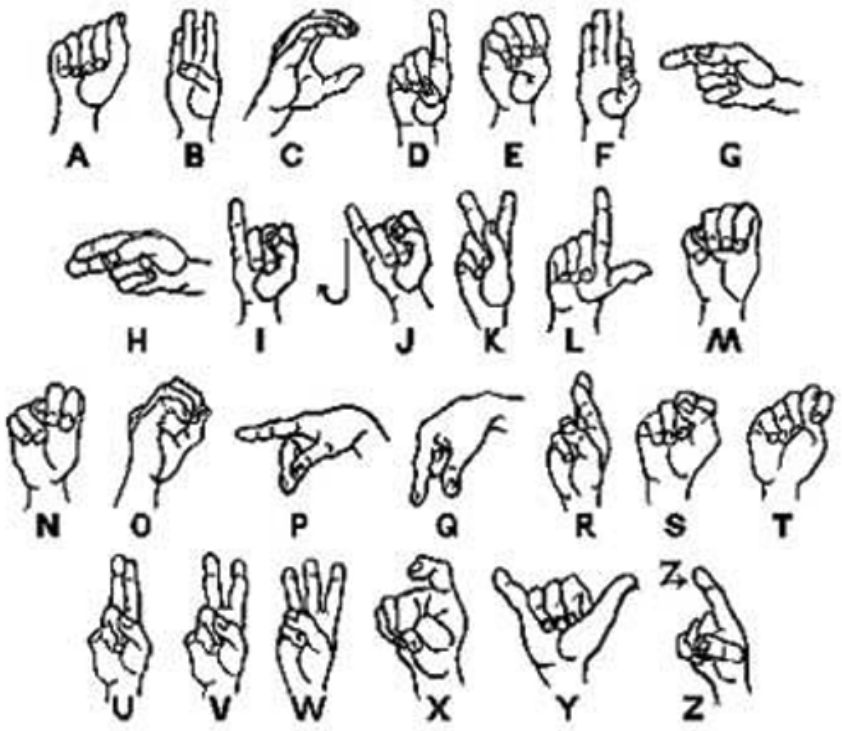

Fig. 2.1: ASL alphabet [12].

As there is a significant number of people with hearing impairments, efforts are being made to develop technology that can interpret SL. This technology uses images, artificial intelligence, mathematical models, and the internet to provide a fast and efficient service [13].

2.1.2 Sign Language Recognition

Sign language recognition (SLR) plays a significant role in deaf communication, education, and human-machine interaction, representing the real-time translation of SL [14]. Integrate society with people who have some hearing loss [7]

Due to the considerable number of hearing-impaired people worldwide, developing practical tools for SL translation is necessary [15].

SLR belongs to an area of artificial vision, and its research is in evolution. Mainly, you can find solutions based on two techniques: computer vision models and sensor-based systems.

Computer Vision Models

CV-based models work with images. Therefore, they require a camera to acquire information and characteristics of hands, gestures, and body postures [14]. The effectiveness of this method is linked to factors such as light, shadows, camera position, and different background conditions. These conditions are challenging to regulate in environments outside of a laboratory [5].

Through this method, the hands of the user must always be in frontal view towards the camera due to the two-dimensional nature with which the information is collected. This presents a significant challenge since many gestures vary slightly from each other [5].

Generally, this method is based on detecting the target or hand within the visual field, following its movement, and recognizing the sign based on this. For the collection of a more significant amount of information, there is the possibility of using specialized cameras and depth sensors to determine the location of the hand concerning the camera. However, this results in a higher economic cost for the device and computational due to the generation of larger files of information to be processed [5]. This limits its adaptability for deployment on mobile devices.

The first system based on CV was carried out in 1988 and was related to Japanese SL. The invention of convolution networks and DL in CV have represented a benefit in applying this tool for SLR [7].

Sensor-based Systems

These systems are based on using different types of sensors, such as tension, surface electromyography, touch, pressure, and inertial sensors, such as accelerometers and gyroscopes. Environmental conditions do not influence the sensors and allow for storing large amounts of information in portable systems. Currently, thanks to technological advances, small and cost-effective microcontrollers, and sensors have been developed [5].

The most common application is gloves with sensors, through which information on the orientation of the wrist, hand movements, and degrees of flexion can be obtained. The information collected through this method is usually sent to mobile applications for processing with an accuracy between 85% and 99% [16].

The advantage of this system's application is the reaction speed and the precision that can be obtained. However, the high cost that it can have for the sensors makes it an inaccessible solution for people with limited resources [7].

The main drawback of these systems can be the discomfort of the user with the glove and the movement restrictions that this can generate. Even so, glove design has been achieved with the ability to identify between 5 and 22 degrees of freedom for SLR [5]. The first system was developed in 1983, based on ASL [7].

2.2 Deep Learning for American sign language

2.2.1 Convolutional Neural Network for Computer Vision

Computer Vision

CV is a branch of artificial intelligence that emulates the visual ability to obtain information from objects through images using detection devices such as a camera and interpreting devices such as a computer. It is focused on interpreting the real world a machine perceives as information, which pretends to have a significant similarity to the perception of the human. CV uses image processing, ML, and statistical analysis techniques to detect and recognize patterns in images and videos to determine the importance of the information obtained. CV is used to automate tasks that humans can do using the vision to obtain information to get a high level of understanding [17] [18] [19].

CV has been used in different tasks, like:

- **Object detection**: consists of detecting objects of different classes, like cars, dogs, or humans, in digital images. A typical process is the creation of classifiers that allow deciding through specific characteristics whether or not an image contains a specific object [20].

- **Face recognition**: It is one of the most significant commercial interest applications. Face recognition systems are based on extracting specific facial characteristics to formulate classification models, through which even predictive models can be implemented [20].

- **Gesture recognition**: human actions and activities are a relevant research topic, and in recent years, several works have been proposed through DL techniques for detecting complex events. The most representative characteristics of specific events are extracted, and classification models are made for their identification [20].

- **Pose estimation**: determining the position of human joints is a topic that has been used in many applications, such as human-machine interaction, motion analysis, and augmented reality [20].

Image processing is the set of techniques that allow preparing the database to be used in the neural network, making training and computational processing easier [19]. Some image processing techniques are:

Normalization

A large number of factors from capture devices or the same optics from the environment cause the variation of characteristics in an image. This topic becomes one of the main issues to be addressed in CV to achieve optimal results. Normalization is a technique applied in data processing to obtain a standard scale in the numerical values of the data obtained without losing information [21].

Data Augmentation

It is a pre-processing technique aiming to augment the data set with altered versions of the existing image. These alterations can be rotations, scaling, and other typical transformations to expose the neural network to a great diversity of new instances that allow the algorithm to obtain a more significant amount of training data and thus make a more robust model [19].

When the training error is decreased, the validation error of a model is also reduced. Therefore, data augmentation is an important technique that addresses overfitting from the training dataset [22]. Validated as an effective technique in image processing, sound classification, and object recognition research by representing performance improvements in neural networks [23].

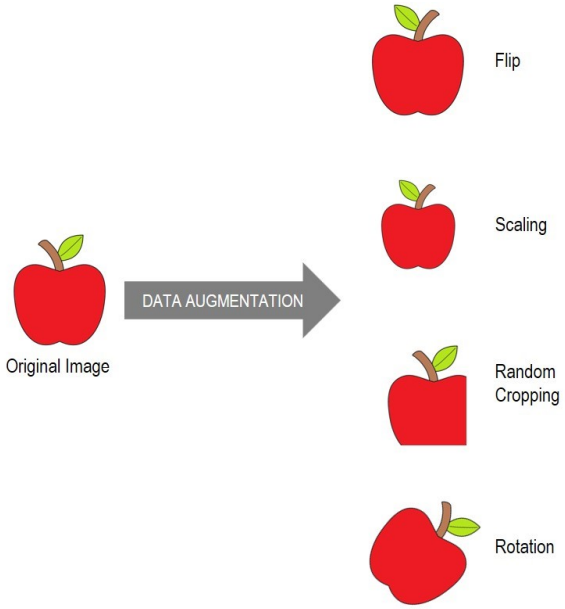

Fig. 2.2: An example of data augmentation applied to an image.

Deep Learning

DL is a ML concept based on artificial neural networks that deal with creating algorithms that can learn and make predictions on data. DL algorithms can learn from data in a way similar to the way humans learn. They can identify patterns and make predictions based on those patterns. DL is mainly used for analyzing unstructured or semi-structured data like images and natural language processing [24].

The neurons are organized into networks with different layers, with the input layer, which refers to the raw or processed data, and the output layer, which is the final result. Between both, one or more hidden layers responsible for learning a non-linear mapping are found. This model needs some hyperparameters that make it possible to learn in a different way than traditional learning algorithms, which must be set manually or determined by an optimization routine [24].

Activation Function

The activation function, also known as the transfer function or non-linearities, aims to introduce non-linearity in the neural network and restrict the output value to a finite value [19] [25].

Softmax Function is commonly used in DL when you have more than two classes. It is used to force the outputs of the neural network to be in the range between 0 and 1, ($0 < output < 1$), and the sum of these values is 1, It is used to predict a single class among several options [19] [26]. The following equation gives the softmax function:

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}} \tag{2.1}$$

Where z is the generated output vector of the neural network, j and k represent the *j-th* and *k-th* vector; and n is the number of classes of the model [26].

ReLU is an activation function whose objective is the activation of a node only when the input is more significant than zero. Otherwise, the output will always be zero with an input below zero. If the input is above zero, it has a linear relationship with the output variable [19]. The following expression defines the ReLU activation function:

$$ReLU(x) = max(0, x) = \begin{cases} x \ if \ x \geq 0 \\ 0 \ if \ x < 0 \end{cases} \tag{2.2}$$

Where x is the value to be processed by the ReLU function.

Loss Function

The loss function is used to validate and evaluate models and the uncertainty of the predictions resulting from these [27]. It is known as the error function or cost function and allows quantifying the error that the neural network prediction has with the correct solution [19].

The lower the error function, the better the work done by the model, and if the error function is high, it means that the model needs optimization of its parameters to reach the minimum error [19].

Categorical Cross-entropy also known as softmax cross-entropy [28], it is a type of error function that allows quantifying the difference between two probability distributions for a sample. It is designed to be used in multiclass classification tasks where a sample can belong to only one of several possible categories [27] [19].

$$CE = -\log\left(\frac{e^{z_p}}{\sum_{n}^{j=1} e^{z_j}}\right) \tag{2.3}$$

Where z is the generated output vector of the neural network, Z_p is the correct class value, $j = 1, 2, 3, ..., n$, and n is the number of classes the model has.

Optimizer

When calculating the loss function of a model, it becomes an optimization problem for which algorithms are required that frame and minimize the error produced. These algorithms are known as optimizers. The objective of the optimizer will be to find the values of the optimal weights to produce the minimum error [19].

Stochastic gradient descent (SGD) is an optimizer that randomly picking up an instance in the training set and computes the gradient based on only that single instance for each iteration [19].

Convolutional neural network (CNN) is one of the most popular deep neural networks. Its name comes from the linear mathematical operation between matrixes called convolution. [29]

CNN is based on neurons that are organized in layers. Convolutional layers include multiple optimizable filters that transform the input data or preceding hidden layers. The number of filters defines the depth of convolutional layers. Kattenborn et al [30] comment that the components of a CNN are:

- **Convolutions Layers**: The convolution is the sliding of the filter over the layer and the calculation of the dot product of the filter and the values of the layer. Using this operation, the main patterns are iteratively learned, and the result is a new layer of dot-products for each filter, also called a feature map. In CV, the convolution layer reduce the image input into a form easier to process, without losing principal features for getting a good prediction [30].

- **Pooling layers**: Pooling describes the transformation of multiple cells into one cell. This feature has some advantages, such as reducing the data size while preserving

discriminant information. Max pooling is a filter that extracts the maximum value from the filter. Other filters used in the pooling layers are average and minimum pooling [30].

- **Normalization Layers**: Normalizing the outputs of a layer helps to reduce the internal covariate shift and improve optimization and stabilization of training [30].

2.2.2 AlexNet

The CNN called AlexNet was the winner in 2012 in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), with a primary and functional structure [31]. Created by Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever, which at the time promoted the application of convolutional networks in studies related to machine vision with greater impetus [19].

The Figure 2.3 display the architecture of AlexNet. This model is made up of 650,000 neurons, and its architecture is divided into eight layers with different dimensions. Five of them are convolutional, some of them followed by max-pooling layers, while the remaining three are fully-connected layers [32]. In its output layer are 1000 neurons connected with a softmax activation function [31].

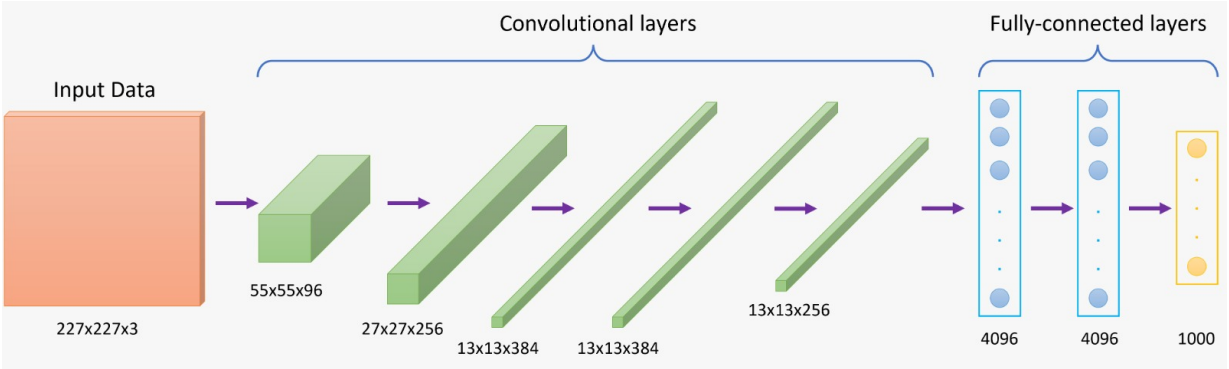

Fig. 2.3: AlexNet architecture.

Being a CNN, they contain fewer connections and parameters, giving the advantage of ease when training their neurons and achieving considerable performance [32].

2.2.3 GoogLeNet

GoogLeNet is a CNN that won the ILSVRC in 2014. It was developed by a team of Google researchers, including Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, this 22-layer deep CNN uses 12 times fewer parameters than the AlexNet neural network and this is significantly more precise [33].

The input size of this neural network is 224x224x3 and its architecture is formed by convolutional layers, max-pooling layers and Inception module with dimensionality reduction as shown in the Figure 2.4.

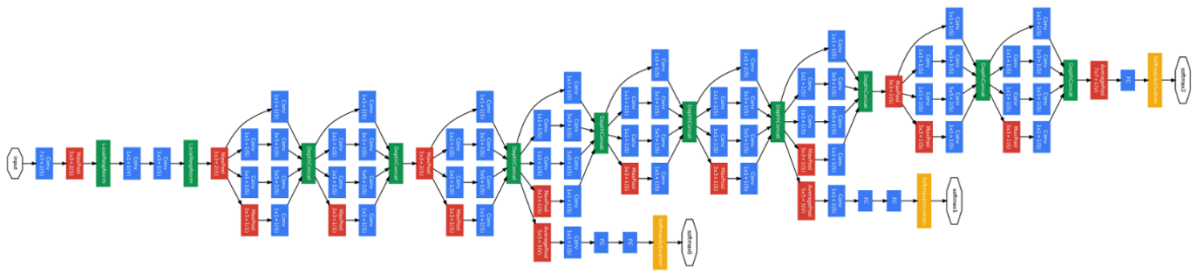

Fig. 2.4: GoogLeNet architecture [33].

The inception module combines several layers with their banks of output filters concatenated in a single output vector that will become the next stage's input. This module uses 1x1, 3x3, 5x5 convolutions and max-pooling layers, as shown in the Figure 2.5.

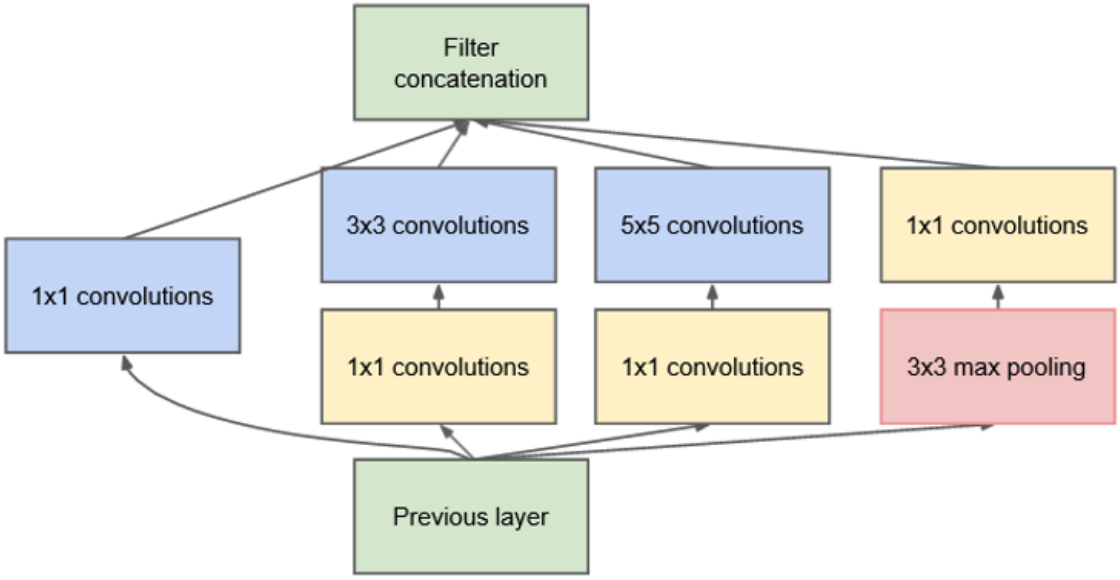

Fig. 2.5: Inception module [33].

Chapter 3

State of the Art

The development of tools and methodologies that facilitate the communication of people with hearing and speech difficulties shortens the gap in mutual understanding and social relations. For this reason, it has become a significant subject of study and has presented essential advances in recent years, some of which are presented in this chapter.

3.1 Dataset for American sign language

There are several databases comprising images of the ASL alphabet. Morocho-Cayamcela et al. [34] published one of these databases in their research work called "Fine-tuning a pre-trained Convolutional Neural Network Model to translate American Sign Language in Real-time". This data set was made up of 78,000 color images (RGB), whose size is 647x511x3. An example of this dataset is shown in Figure 3.1.

Fig. 3.1: Instance of dataset presented by Morocho-Cayamcela et al. [34].

In 2011, Barczak, A.L.C., et al., in their research paper "A new 2D static hand gesture color image dataset for ASL gestures." presented the database called MU_HandImages_ASL. This database was made up of 2425 images that were taken from 5 individuals with different light conditions [35]. The Figure 3.2 shows a sample of the database.

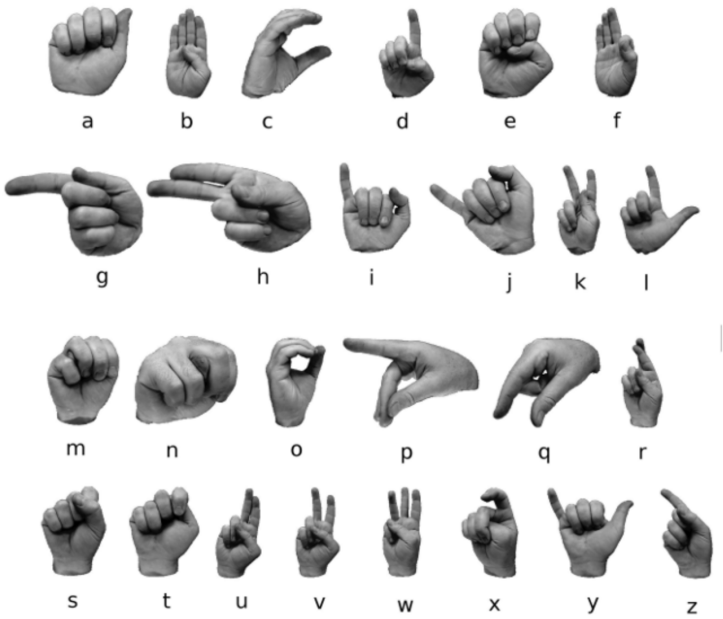

Fig. 3.2: Sample of ASL letter with segmented image [35].

Another database generated and accessible in Kaggle is called SL MINST, see Figure 3.3. This database contains images of 24 of the 26 ASL alphabet letters. The excluded letters are J and Z. SL MINST is composed of a training set containing 27455 images and an evaluation set comprising 7172 images. Each image has a dimension of 28x28x3 [36] [37].

Fig. 3.3: Images samples from the SL MINST database [38].

3.2 Convolutional neural network for sign language

Al-Qurishi, Khalid, and Souissi published different automated SL recognition based on machine/deep learning methods and techniques between 2014 and 2021 [15]. They proposed that SLR can be classified into two major groups. One group depends on external sensors to collect insights about the actions of the signer. The other group focus on the use of vision-based methods. This group relies on images, video, and depth data to determine the semantic content of hand signs.

Current advances in this area have been largely fueled by the use of DL models, which are presently being perfected and will only become more widely received in the future years. Over the past decade, multiple original and highly clever suggestions had been used to build SLR tools by extracting features from sensor data or visual streams and feeding them into neural classifiers [15].

Morocho-Cayamcela and Lim presented a system to recognize a real-time ASL hand gesture recognizer based on an artificial intelligence [34]. The system presented the uses of CNN that was trained using a dataset with 78,000 images, and each picture had a resize into 227x227x3 to train in AlexNet and 224x224x3 to train GoogLeNet. The model proposed had components like loss function, which evaluates the predicted label of the class to comply with the labels from the ground-truth data and optimized weight and biases to increase the classification accuracy. Deep learning models that have been trained before on a different dataset applying fine-tuning were used. They used the AlexNet model, which contains five convolutional layers and three fully connected layers. The last set of fully-connected layers was replaced with a new one that classified the 26 letters of the alphabet. The result presented a fast convergence using AlexNet with a 99.39% accuracy, and using GoogLeNet they got an accuracy of 95.52%. The paper expose that use data augmentation in the dataset applying random reflection on the horizontal axis, translation on a 30-pixel range over the x and y-axis to reduce the overfitting [34].

Bin, Huann and Yun [12] presented a convolutional neural network model for ASL prediction with 4800 images used to train and validate the model. The data set was generated from 200 pictures for each gesture, considering different backgrounds and lighting conditions and focusing only on the first 24 gestures since the remaining 2 ASL requires movement. That is, the proposed model was developed for static gestures and was proposed to establish a basis for future studies [12].

The architecture of the proposed model was composed of an output layer with the Softmax function. The complete model was implemented in multiple layers. See Figure 3.4.

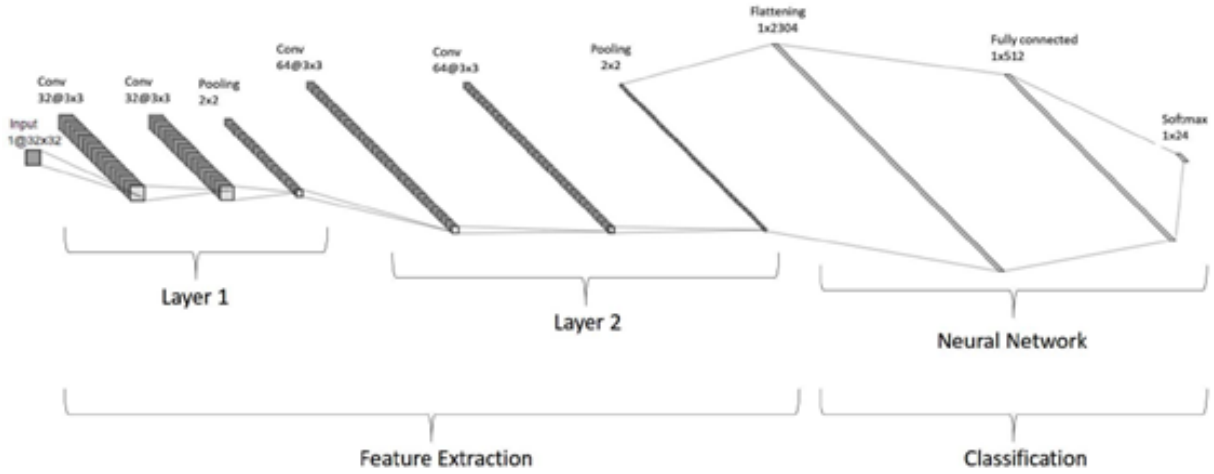

Fig. 3.4: The architecture of the proposed CNN model [12].

The training was developed using the ADAM optimizer, a categorical cross-entropy loss function, and the following parameters: a learning rate of 0.003 and iterations of 100. Through the experiment, they obtained a precision of 95% in 24 ASL gestures and demonstrated the effectiveness of the model for static alphabet gestures [12].

Bantupulli and Xie [13] created a computer vision application to communicate by translating ASL into text. They implemented an Inception-type convolutional neural network to extract partial features in videos, which, by using transfer learning, makes it possible to take advantage of previous training and use a small amount of data. The videos were divided into frames, and the data corresponding to each gesture was increased by applying data augmentation. The data set created was divided into 1800 images for training and 600 images for evaluation data. Gesture detection was 99% accurate on the training sets. An ADAM optimizer was used in conjunction with a softmax layer for prediction classification. The drawback presented by this model was the loss of precision when performing tests on different skin tones and types of clothing [13].

Saravanan, Retnaswamy, and Selva [39], explain in their research how a webcam captures the hand gestures corresponding to the ASL alphabet to transform them into text by applying a modified CNN AlexNet architecture in its final layers, including classification and softmax layers, with which the architecture is formed for 25 layers, see Figure 3.5

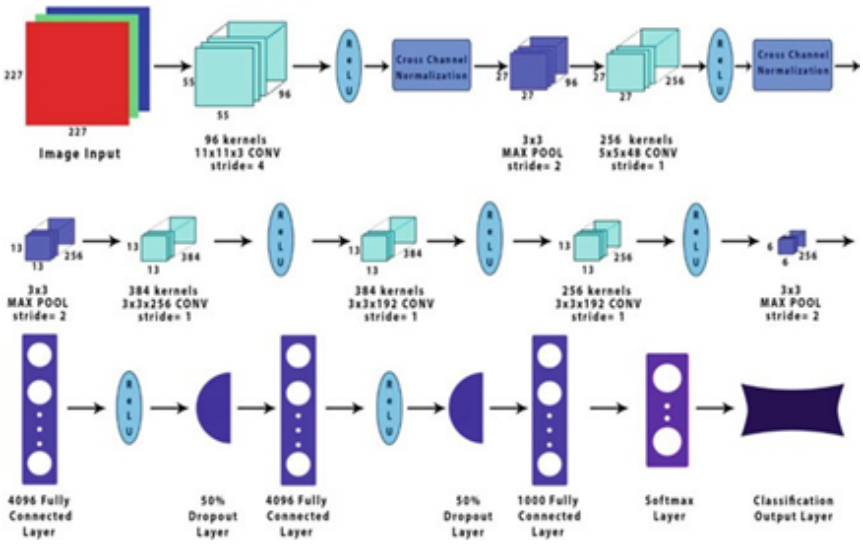

Fig. 3.5: Modified CNN AlexNet, proposed by Saravanan et al. [39].

A database taken from Kaggle used with 110 images of 200 x 200 pixels for each ASL alphabet, which were divided so that 80% serve as a training database and 20% for evaluation. The proposed neural network uses the optimizer for its training: stochastic gradient descent with momentum (SGDM), an initial learning rate: of 0.001, and a maximum number of epochs or iterations: 10. As a result of the model obtained, there is an accuracy of 100% with the images of the evaluation data set, however, in tests with live images of hand gestures captured by a computer camera, an accuracy of 76.92% was achieved, due to the inability of the model to recognize the letters. R, S, T, U, V, and X. This limitation can be attributed to the short time given to the training model due to machine restrictions and the similarity of the descriptor vectors of these letters [39].

Liu et al. [40], presented an improved AlexNet-based gesture recognition algorithm using four convolutional kernels of 3x3 instead of 11x11, followed by a clustering layer (see Figure 3.6).

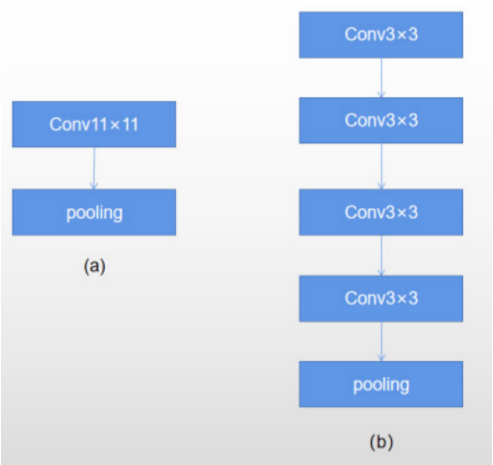

Fig. 3.6: (a) Original AlexNet layer, (b) Modified layer proposed model [40].

Among the data sets tested in the experiment was the ASL, with a total of 65774 images for 24 gestures. The original convolutional network was compared with the improved one from AlexNet, and an accuracy of 98.3% was obtained compared to 95.9% of the original convolutional network. They concluded that by using the improved AlexNet algorithm, it was possible to receive more information during feature extraction by reducing the step of the convolutional layers [40].

Chapter 4

Methodology

Ensuring the success of deep learning models in computer vision tasks requires paying attention not only to the neural network's architecture and design but also to the system structure and hyperparameters. This chapter focuses on discussing the system structure, experiment setup, metrics used in the experiments, and the web application.

4.1 System Structure

4.1.1 Dataset

The dataset was divided into three sections. The first section was the test training. It contained 20,800 images with 26 classes. The second section was the test data, with 2,600 images. Finally, the last section was the validation data which had 2,600 images with the 26 different types of hand gestures. The division of this data was 8:1:1 and was the base of the experiment. Every image data was normalized to allow the machine to perform better calculations.

4.1.2 Hyperparameters

Epoch

It is the number of the times the neural network train. In AlexNet, The model reached a good stabilization around the 20 to 30 epoch. With 50 epochs, the results was clear to interpret. On the other hand, GoogLeNet started to stabilize around the 50 epoch, and with 70 epochs, the results had a good point to complete the analysis.

Learning rate

Learning rate (LR) is a hyperparameter of the neural network that influences its behavior [41]. LR is the magnitude of each step that the neural network takes when descending the loss function, i.e., the percentage of change updates the values of the weights in each iteration to find the optimal values of the weights to minimize the error.

If the LR is vast, it speeds up the learning, but it does not guarantee to find the minimum error, while if the value of the LR is minimal, the error was minimal; however, the time it takes to train the network was considerable (weeks or months) [19].

4.2 Experiment setup

4.2.1 Hardware

The experiments were conducted in a workstation with a processor Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz, 2601 Mhz, 2 Core(s), 4 Logical Processor(s), and 16 GB of RAM.

4.2.2 Training

Once the preprocessed data stage was completed, these go to the training stage. At this stage, the images entered the AlexNet and GoogLeNet networks, which were built with different layers. At the end of the training, they could infer between the 26 classes in the database.

4.3 Experiments

4.3.1 Experiment 1

In the first experiment, The AlexNet and GoogLeNet models were trained using the data augmentation method. This method was expected to reduce overfitting and increase validation accuracy compared to the regular model.

4.3.2 Experiment 2

AlexNet and GoogLeNet models did not use regularization in the base architecture. Applying this method, both models presented a better training and validation accuracy performance. The regularization method to use was the L2.

4.3.3 Experiment 3

The dropout is a method to reduce the overfitting used in AlexNet and GoogLeNet. The accuracy was reduced if the architecture did not present dropout. The results had a significant change from the standard data obtained.

4.3.4 Experiment 4

The last experiment involved training the models using methods that reduce overfitting and improve accuracy. AlexNet and GoogLeNet architectures implemented data augmentation, regularization, and dropout. The performance showed a considerable improvement in comparison to the regular training.

4.4 Classifier Evaluation Metrics

4.4.1 Accuracy

Accuracy is one of the most used measurements for evaluating classification models. The accuracy formula is as follows:

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \tag{4.1}$$

The following formula can also be applied:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

In this context, TP stands for true positive, TN represents true negative, FP indicates false positive, and FN stands for false negative [42].

4.5 Web Application

For the development of the website, the following development tools were used:

- **Reactjs**: A library that allows to develop user interface (UI) for websites around components. A component is a UI with its own logic and appearance [43].
- **Nextjs**: it is a framework developed by Vercel which allows us to build web pages using tools like Reactjs [44].
- **Mediapipe**: It is a library that allows us to apply artificial intelligence and ML in our applications. This library helps us recognize hands in real-time in our web application [45].
- **Tensorflowjs** is a library that helps us to use ML in JavaScript. This tool is vital to assemble our trained model and put it to work on our web page [46].

4.5.1 Workflow

Nextjs and Reactjs were used to create the foundation of the website. Figure 4.1 illustrates the process our website uses to identify SL.

Fig. 4.1: Workflow of the recognition of hand pose in the webpage.

To detect hand gestures, the process involved using the webcam to capture the image of the user. Then, the Mediapipe library was utilized to identify the hand on the screen and draw a box around it. Once the hand data was collected, the image was introduced into the model to determine the gesture type. This entire process happened in real-time, and Figure 4.2, 4.3 and 4.4 showcases the appearance of the webpage during operation.

Fig. 4.2: Capture of the webpage recognizing the letter “i”.

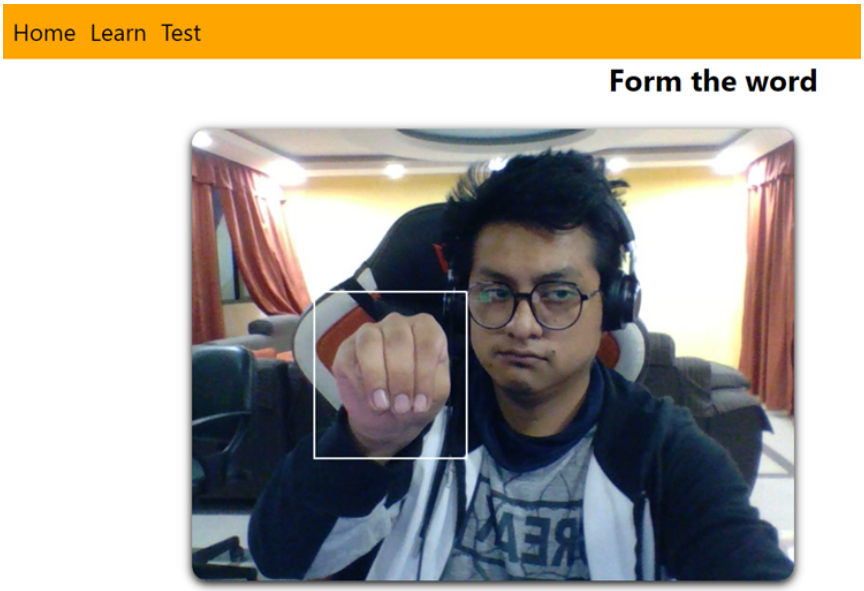

Fig. 4.3: Capture of the webpage identifying the user with the hand.

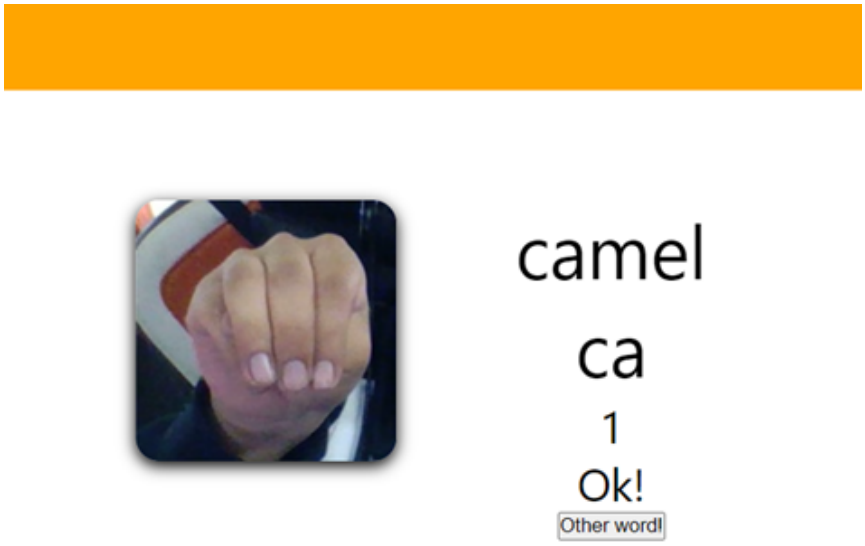

Fig. 4.4: Capture of the webpage recognizing the hand gesture to form a word.

Chapter 5

Results and Discussion

This chapter will discuss the extensive results obtained, divided into three sections. The first section will focus on the results obtained with the AlexNet model; the second section will discuss the training results with GoogLeNet, and the last section will compare the results obtained with AlexNet and GoogLeNet.

5.1 AlexNet

The results obtained in experiment 1 using the AlexNet model can be seen in Figure 5.1. The values obtained in epoch 10 showed the validation of the data by going through the trained model, showing us a difference of 0.013 between the model using standard data and the model using data augmentation. The use of regular data was more effective. However, in epoch 20, the difference between these methods was reduced to 0.006. The most effective model, in this case, was the one that uses the data method augmentation.

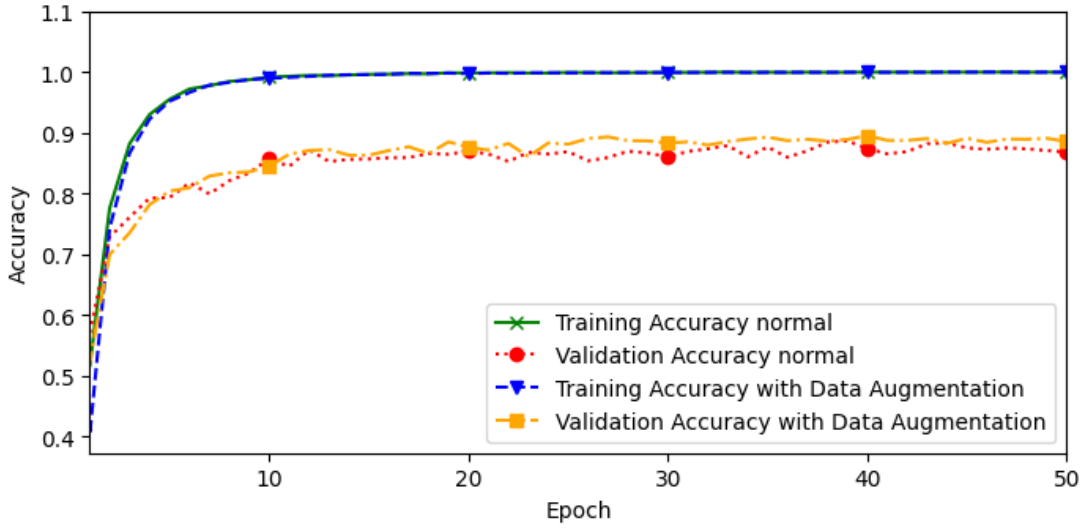

Fig. 5.1: Experiment 1 results: AlexNet with data augmentation.

Thanks to the representation of Figure 5.2, It was more understandable to visualize the difference between periods. From epoch 20 onwards, the trend stabilized, and the results obtained using data augmentation were slightly better than not using it. These results also helped us to understand that implementing data augmentation reduced overfitting since when training using data augmentation, the model classified better the newly entered images. Using data augmentation generally gave us better performance in the AlexNet model.

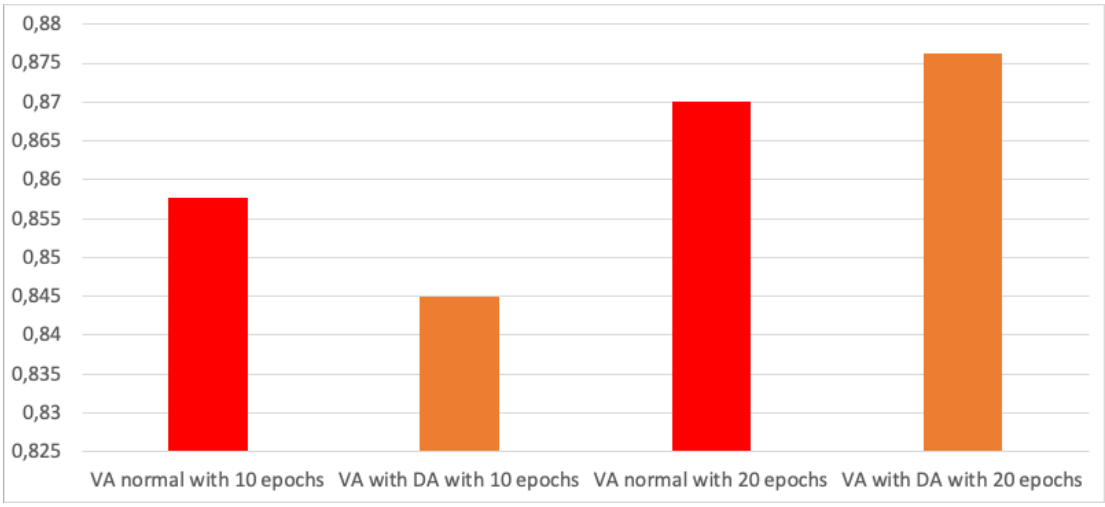

Fig. 5.2: AlexNet Normal model vs model with data augmentation. VA = validation accuracy; DA = data augmentation.

In the second experiment, the regularization method was implemented to reduce over-

fitting, as with data augmentation. As shown in Figure 5.3, the difference was not as noticeable as expected. However, when it was analyzed by epochs, the model showed a slight increase in precision when using regularization.

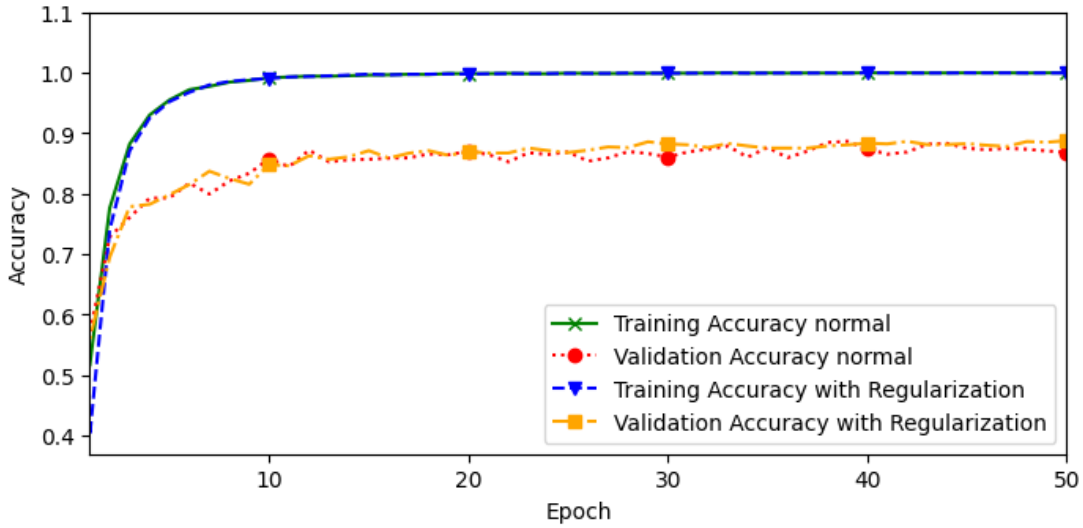

Fig. 5.3: Experiment 2 results: AlexNet with regularization.

The graph of Figure 5.4 showed that at epoch 10, using the standard model had an advantage of 0.009% compared to the regularization model. When the experiment went further for example the epoch 20, both models had a similar accuracy. On this occasion, the model with regularization had significant growth by recognizing the values compared to the regular one despite giving us a similar result. It was in epoch 30 that it continued to be appreciated that the model with regularization has an upward trend concerning accuracy. Now difference of 0.02 appeared, being the lower regular model. In the following periods, there was a stabilization where the model with regularization was slightly higher. Regularization did not make a big difference in precision improvement. Still, it could help to reduce overfitting by giving insurance when implementing it in the AlexNet model.

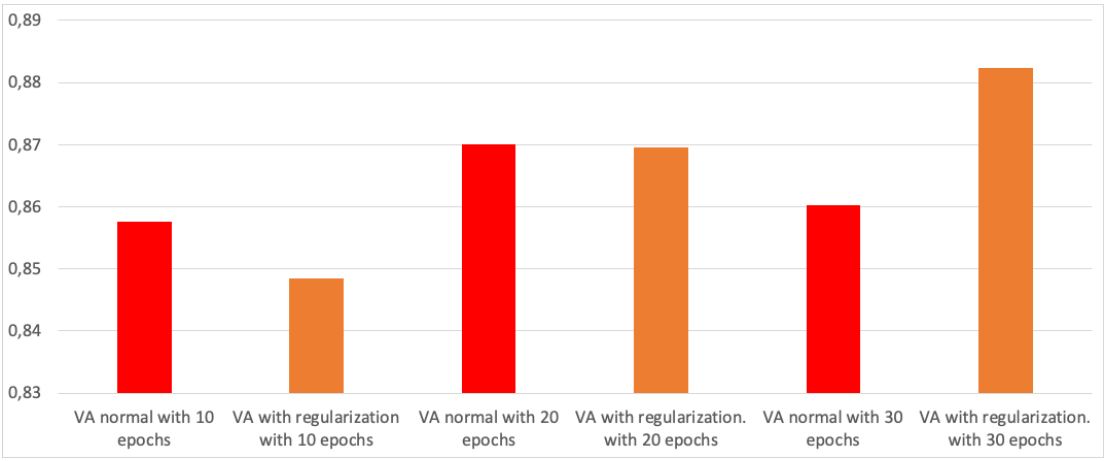

Fig. 5.4: AlexNet Normal model vs model with regularization. VA = validation accuracy.

The data obtained in experiment 3 shown that the regular model uses dropout by default. This was to reduce overfitting. Still, the difference in improvement compared to the model without dropout could be better, as shown in Figure 5.5. However, using dropout ensured the model reduced the overfitting problem in training.

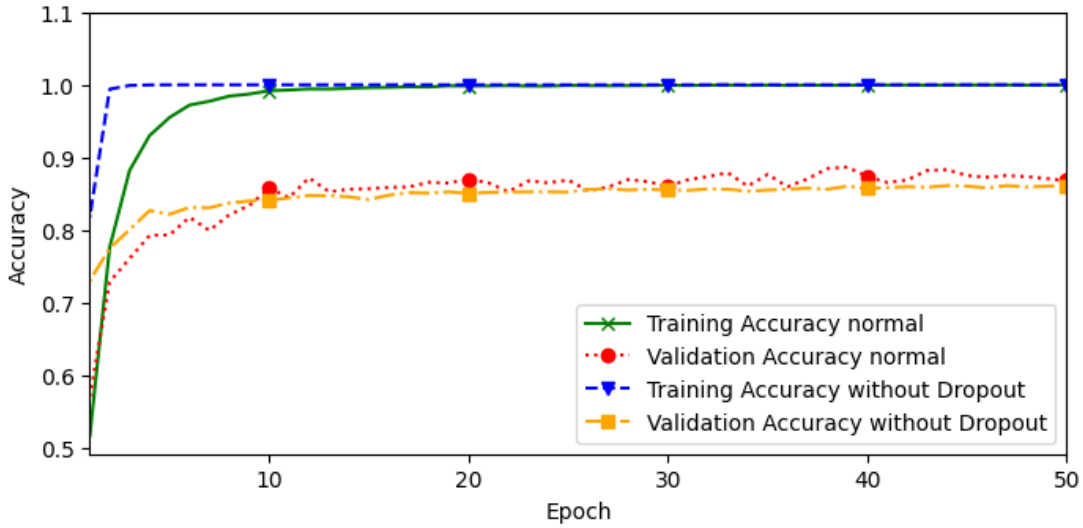

Fig. 5.5: Experiment 3 results: AlexNet without dropout.

Finally, in experiment 4, the previously tested methods were applied. This meant that the dropout, data augmentation, and regularization are implemented in a model. Figure 5.6 shows the results obtained from this training. There was a slight improvement in accuracy, but it was not as big as expected.

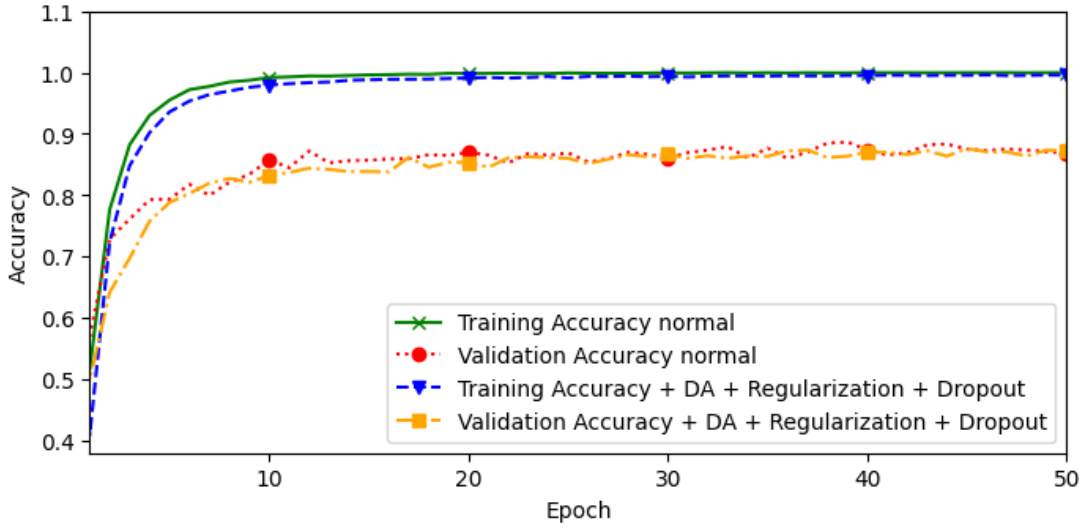

Fig. 5.6: Experiment 4 results: AlexNet with data augmentation, regularization, and dropout.

The Figure 5.7 shows that the data at epoch ten shown that the standard model had better precision than the model with all the tools applied. Here was a pattern where the regular model outperforms the modified model in all the experiments at epoch ten. However, it was not the optimum value, so training with more periods improved the precision. Epoch 20 shown that both models had a similar result. The difference could be more significant. However, the modified model had a better growth trend than the regular model. Finally, epoch 30 gave us a better view than the modified model tends to continue growing. Although the growth difference was slight compared to the regular model, it was higher.

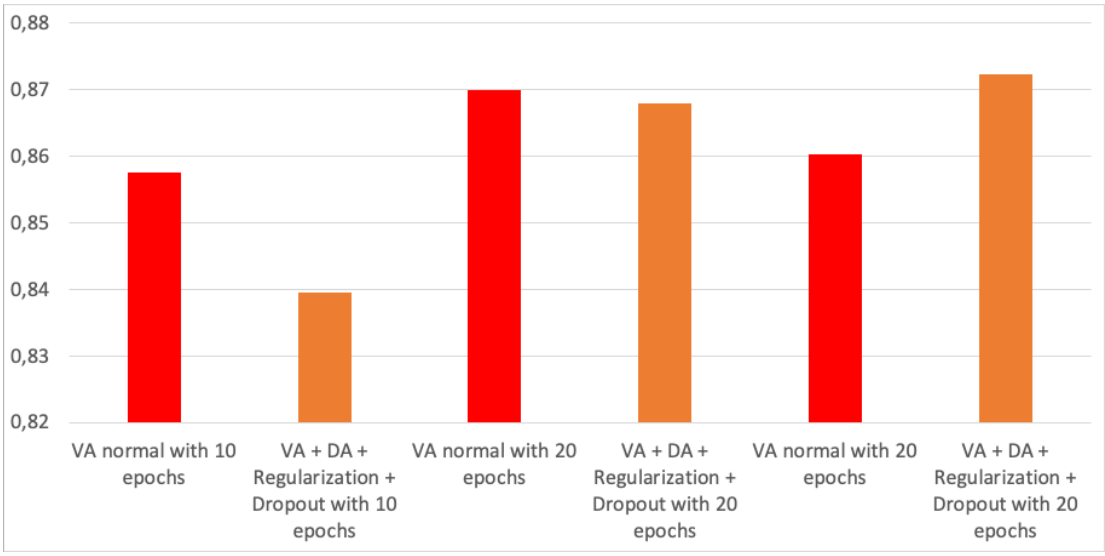

Fig. 5.7: AlexNet Normal model vs model with data augmentation, regularization and dropout. VA: Validation accuracy; DA: Data augmentation.

5.2 GoogLeNet

In Figure 5.8 the results obtained from experiment 1. The GoogLeNet model presented an unexpected behavior. Despite showing a slight difference in precision between the base model and the model with data augmentation, the base model being better, the distance with the training was less when using data augmentation. This result may mean that data augmentation was closer to the training accuracy than not.

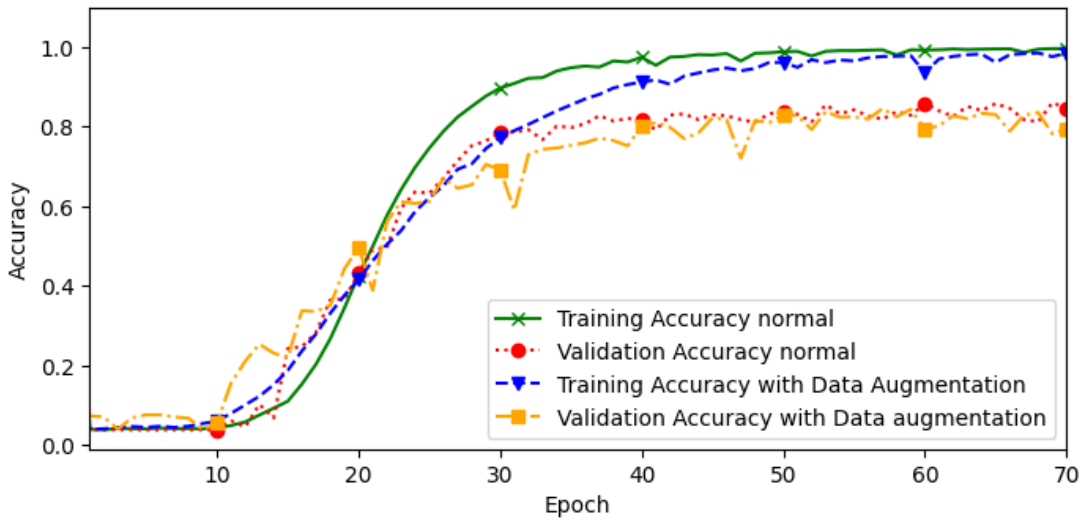

Fig. 5.8: Experiment 1 results: GoogLeNet with data augmentation.

Analyzing experiment 1 by section, as shown in Figure 5.9, demonstrated that the base model performs better when discussing accuracy. Since in the epochs number 40, 50, and 60, it slightly outperforms the data augmentation model. Applying data augmentation in this particular model did not present an improvement in accuracy. However, it was not significantly different from not using it either.

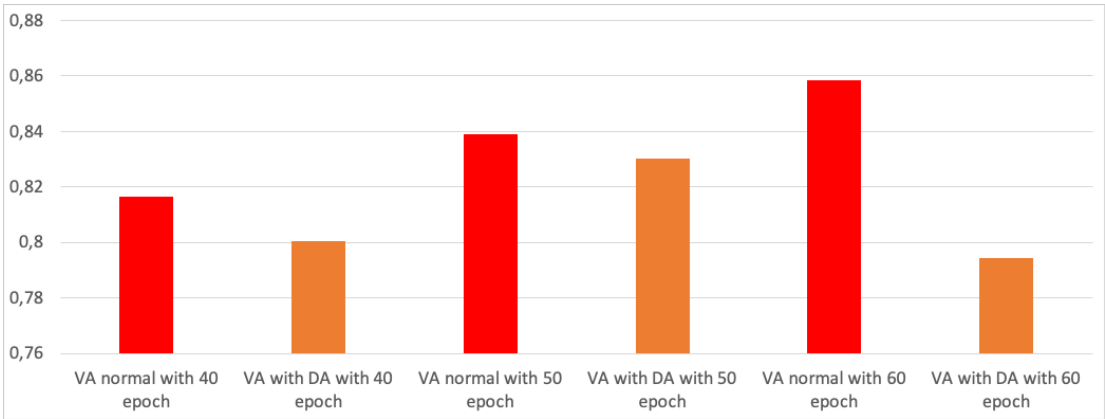

Fig. 5.9: GoogLeNet normal model vs model with data augmentation. VA = validation accuracy; DA = data augmentation.

In Experiment 2, the regularization was used to prevent overfitting. As shown in Figure 5.10, both models behave similarly. However, the model with regularization offered an additional method to ensure overfitting was reduced.

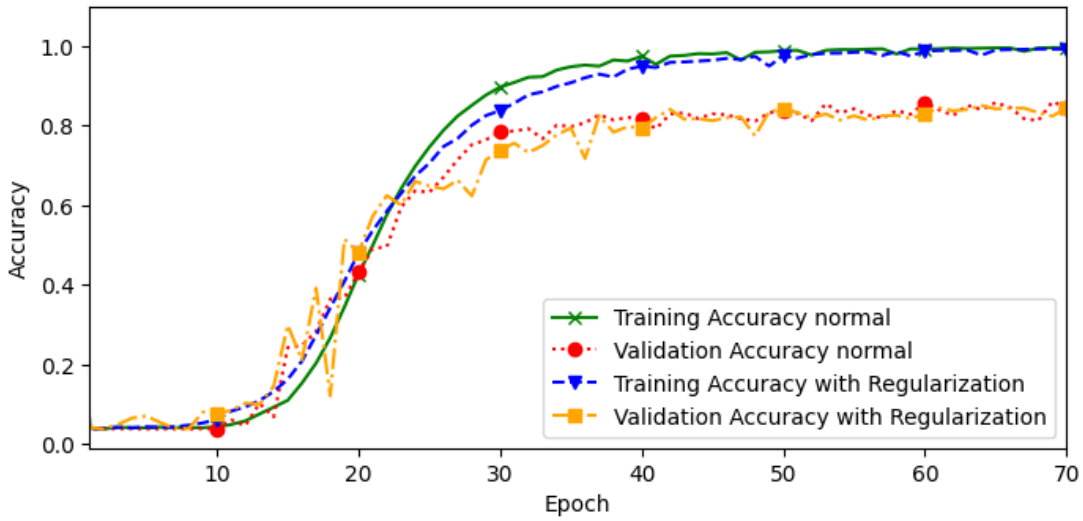

Fig. 5.10: Experiment 2 results: GoogLeNet with regularization.

The data obtained in experiment 3 shown us that the dropout stabilized the precision.

As shown in Figure 5.11, the model without dropout had a better accuracy at epoch 20. However, since epoch 30, this model did not tend to increase in accuracy, and it could be seen that it had downward peaks at certain times, unlike the regular model, where it shown better stabilization and exceeds the proposed model, confirming that the dropout had a positive influence when reducing overfitting since it recognized new images.

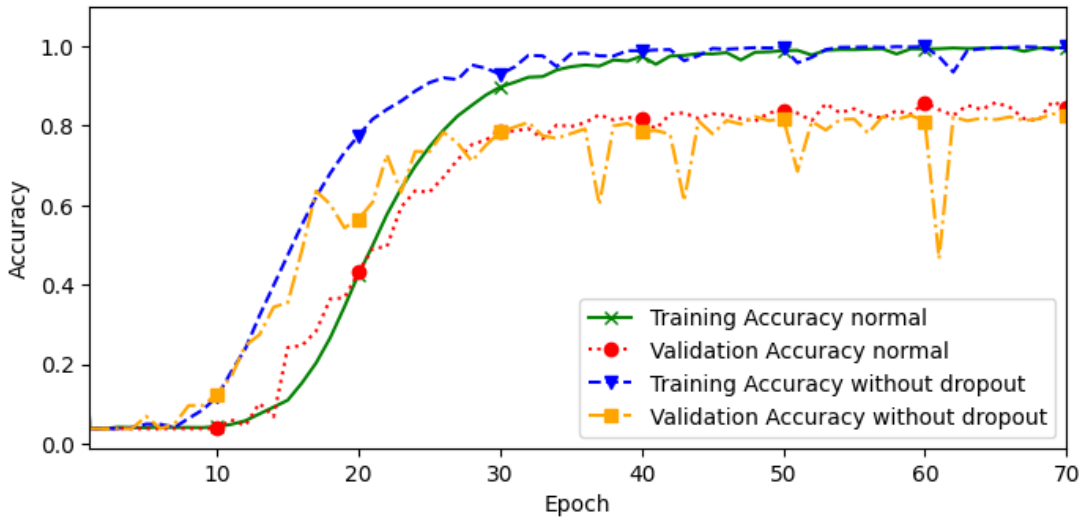

Fig. 5.11: Experiment 3 results: GoogLeNet without dropout.

The outcome of experiment 4 was displayed in Figure 5.12. The model suggested in this experiment utilized data augmentation, regularization, and dropout techniques. In contrast to the earlier outcomes, this model's performance improved when all these methods were combined, with a tendency to increase accuracy.

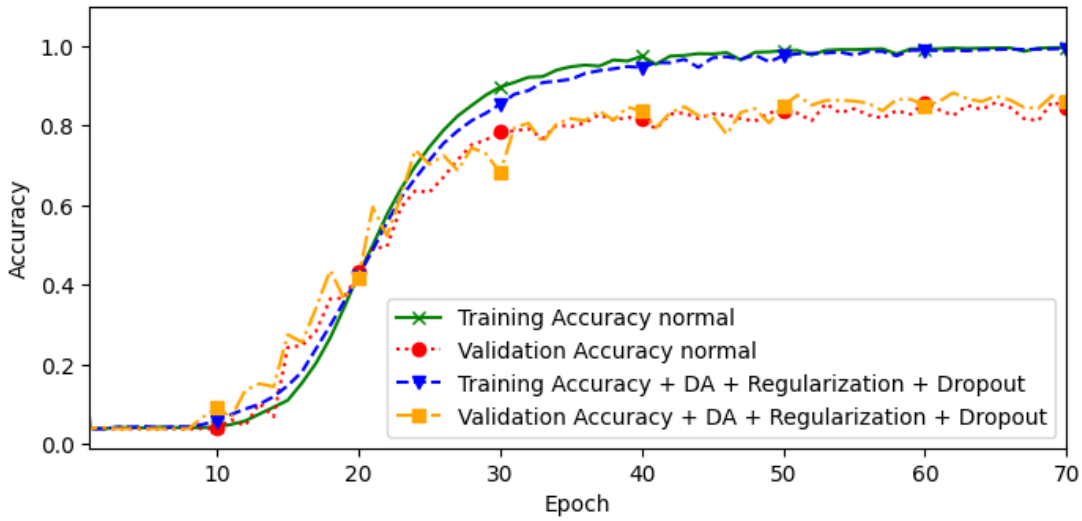

Fig. 5.12: Experiment 4 results: GoogLeNet with data augmentation, regularization, and dropout.

Looking at the values in Figure 5.13, when compared by epochs, it was evident that the proposed model outperforms the regular model. However, similar to previous experiments, the difference between these values was slight.

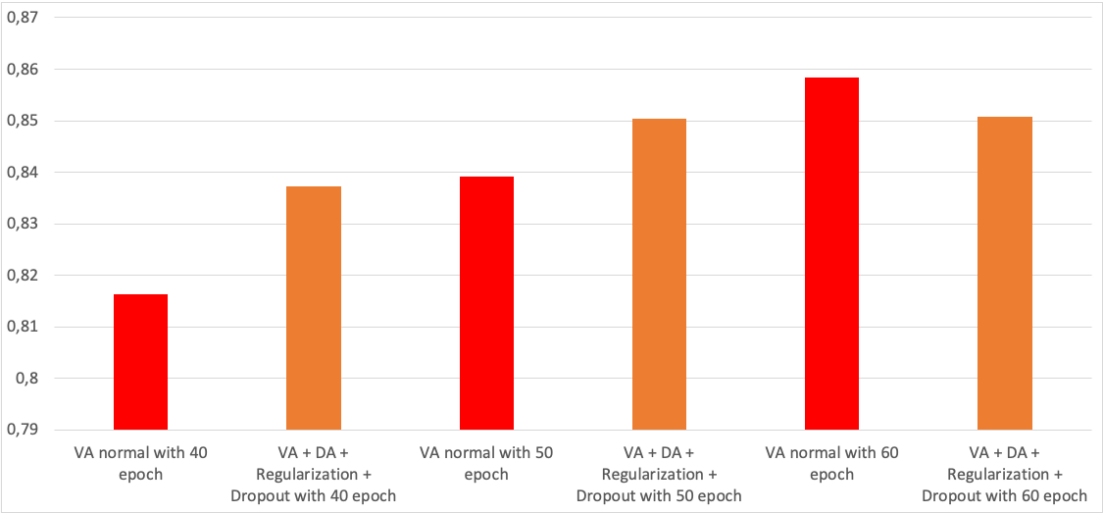

Fig. 5.13: AlexNet normal model vs model with data augmentation, regularization and dropout. VA: validation accuracy; DA: data augmentation.

5.3 AlexNet vs GoogLeNet

Each experiment using the AlexNet took an estimated 18 hours to train, compared to the GoogLeNet, which took around 24 hours to complete training. After analyzing all the data

obtained in both models, it was time to make a direct comparison and shown a general analysis of the results.

Table 5.1 shows the average validation accuracy. These results were calculated as follows:

- The AlexNet model uses only 50 training epochs because the results became stable from epoch ten onwards, allowing for optimal results from season 10. Thus, the accuracy average was calculated for each experiment from season 10 to season 50.
- The GoogLeNet model underwent 70 epochs of training. This was necessary because, unlike AlexNet, the model required a minimum of 30 epochs for optimal results. However, after the 40th epoch, the validation accuracy stabilized. Therefore, an average of 40 to 70 epochs was taken from each experiment.

	AlexNet (%)	GoogLeNet (%)
Base Model	87	83
Experiment 1	88	81
Experiment 2	87	83
Experiment 3	85	79
Experiment 4	87	85

Table 5.1: Average validation accuracy on AlexNet and GoogLeNet.

In Experiment 1, the model used data augmentation to improve the results. It was found that it improved the accuracy of AlexNet by 1%. However, it decreased the accuracy of GoogLeNet by 2%. Therefore, data augmentation could sometimes improve the accuracy of new data, but not always.

Experiment 2 utilized the regularization method. It was anticipated that this would lead to a notable enhancement in the inference of new data. However, the results indicated that performance was the same between implementing this method and not. Despite the theoretical reduction in overfitting, practical outcomes had no noticeable impact.

The dropout was removed from the models used in experiment 3, which decreased validation accuracy. However, the decrease was not significant. In AlexNet, the reduction was only 2%, and in GoogLeNet, it was 4% compared to the base model. Despite this, dropout could still improve the inference of new data, although the improvement may not be very significant.

After conducting various experiments on AlexNet and GoogLeNet models using data augmentation, regularization, and dropout techniques, it was found that the AlexNet model remained unchanged. However, the GoogLeNet model showed a 2% improvement. It is important to note that despite this improvement, the GoogLeNet model still needed to catch up to the AlexNet model, which did not require additional epochs to stabilize. The GoogLeNet model, on the other hand, required 20 more epochs to stabilize.

Chapter 6

Conclusions

6.1 Conclusions

AlexNet achieved an 87% accuracy rate, stabilizing its results after 50 epochs, while GoogLeNet required 70 epochs to reach 85%. Despite the 2% difference, AlexNet demonstrates superior performance. However, applying techniques like data augmentation, regularization, and dropout does not dramatically enhance the prediction of new data. Nevertheless, it guarantees that the model has the overfitting reduced.

Based on the results, AlexNet is more effective at analyzing new data than GoogLeNet. Furthermore, the AlexNet model requires less training time to stabilize validation accuracy. Therefore, it is optimal for implementing SLR on a webpage.

Although AlexNet outperformed GoogLeNet initially, both networks achieved up to 90% accuracy since they were trained from scratch. These methods were easy to implement, and the weights obtained were exclusively designed to recognize the hand's pose. However, limitations were encountered during the training execution.

To achieve higher accuracy, it would be ideal to train using all the images in the database. However, technical limitations only allowed for the use of 33.33% of the database in this study. Despite these limitations, the results were still impressive, with AlexNet achieving 87% accuracy and GoogLeNet achieving 85% accuracy. This demonstrates the effectiveness of the SLR system.

6.2 Limitation

One of the main challenges of this project was the lack of sufficient memory for training. The machine used for model training had a RAM of only 16 GB, while the entire database requires at least 50 GB of RAM to train the models. As a result, The dataset had to be limited to a smaller portion to overcome this memory constraint.

Another of the project's limitations was the inadequate availability of graphics cards. As a result, the training process for Alexnet took roughly 18 hours per experiment. In comparison, Googlenet's training lasted about 24 hours per experiment with only one GPU. This shortage of resources resulted in significant delays.

6.3 Future work

Future work can involve training the two models with the entire database to improve the accuracy of hand gesture recognition. Expanding the research area and training a network to recognize hand movements can enhance the ASL learning experience. Applying the model to the web page can make the platform more interactive for the user.

The web page can use other databases to improve its training and teach not just ASL but other SLs like Ecuadorian SL. This will expand the resources available to teach people ASL.

Bibliography

[1] A. Nogueira-Rodríguez, R. Domínguez-Carbajales, H. López-Fernández, Águeda Iglesias, J. Cubiella, F. Fdez-Riverola, M. Reboiro-Jato, and D. Glez-Peña, "Deep neural networks approaches for detecting and classifying colorectal polyps," *Neurocomputing*, vol. 423, pp. 721–734, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220307359

[2] American sign language. [Online]. Available: https://www.nidcd.nih.gov/health/american-sign-language

[3] O. B. Adedoyin and E. Soykan, "Covid-19 pandemic and online learning: the challenges and opportunities," *Interactive Learning Environments*, vol. 0, no. 0, pp. 1–13, 2020. [Online]. Available: https://doi.org/10.1080/10494820.2020.1813180

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

[5] K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, "Wearable sensor-based sign language recognition: A comprehensive review," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 82–97, 2021.

[6] S. Tornay, M. Razavi, and M. Magimai.-Doss, "Towards multilingual sign language recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6309–6313.

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

[8] D. K. Singh, A. Kumar, and M. A. Ansari, "Robust modelling of static hand gestures using deep convolutional network for sign language translation," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 487–492.

[9] Suharjito, H. Gunawan, N. Thiracitta, and A. Nugroho, "Sign language recognition using modified convolutional neural network model," in *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, 2018, pp. 1–5.

[10] A. Das, S. Gawde, K. Suratwala, and D. Kalbande, "Sign language recognition using deep learning on custom processed static gesture images," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1–6.

[11] M. Kumar, P. Gupta, R. K. Jha, A. Bhatia, K. Jha, and B. K. Shah, "Sign language alphabet recognition using convolution neural network," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 1859–1865.

[12] L. Y. Bin, G. Y. Huann, and L. K. Yun, "Study of convolutional neural network in recognizing static american sign language," in *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2019, pp. 41–45.

[13] K. Bantupalli and Y. Xie, "American sign language recognition using deep learning and computer vision," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4896–4899.

[14] W. Li, H. Pu, and R. Wang, "Sign language recognition based on computer vision," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 919–922.

[15] 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. 126 917–126 951, 2021.

[16] F. Pezzuoli, D. Corona, and M. L. Corradini, "Dynamic gestures recognition through a low-cost data glove," in *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, 2020, pp. 1–3.

[17] M. Hassaballah and A. I. Awad, *Deep Learning in Computer Vision: Principles and Applications*, 03 2020.

[18] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14 985–15 006, 2019.

[19] M. Elgendy, *Deep Learning for Vision Systems*. Manning Publications Co., 2020.

[20] D. Andina, A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, 02 2018.

[21] J. Sun, X. Cao, H. Liang, W. Huang, Z. Chen, and Z. Li, "New interpretations of normalization methods in deep learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5875–5882, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6046

[22] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, 2019.

[23] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, p. 433–452, 2018.

[24] Z. P. . H. K. Janiesch, C., "Machine learning and deep learning. electron markets," 2021.

[25] L. B. Godfrey, "An evaluation of parametric activation functions for deep learning," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 3006–3011.

[26] I. Kouretas and V. Paliouras, "Simplified hardware implementation of the softmax activation function," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2019, pp. 1–4.

[27] G. Vishwakarma, A. Sonpal, and J. Hachmann, "Metrics for benchmarking and uncertainty quantification: Quality, applicability, and best practices for machine learning in chemistry," *Trends in Chemistry*, vol. 3, no. 2, pp. 146–156, 2021, special Issue: Machine Learning for Molecules and Materials. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2589597420303142

[28] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[29] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[30] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (cnn) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620303488

[31] X. Zhang, W. Pan, and P. Xiao, "In-vivo skin capacitive image classification using alexnet convolution neural network," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 2018, pp. 439–443.

[32] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[34] M. E. Morocho Cayamcela and W. Lim, "Fine-tuning a pre-trained convolutional neural network model to translate american sign language in real-time," in *2019 Inter-*

national Conference on Computing, Networking and Communications (ICNC), 2019, pp. 100–104.

[35] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2d static hand gesture colour image dataset for asl gestures," 2011.

[36] M. Bilgin and K. Mutludoğan, "American sign language character recognition with capsule networks," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019, pp. 1–6.

[37] M. M. Hasan, A. Y. Srizon, A. Sayeed, and M. A. M. Hasan, "Classification of sign language characters by applying a deep convolutional neural network," in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 2020, pp. 434–438.

[38] Tecperson, "Sign language mnist," Oct 2017. [Online]. Available: https://www.kaggle.com/datasets/datamunge/sign-language-mnist

[39] R. Saravanan, S. Retnaswamy, and S. Selvan, "A method of hand gestures recognition using convolutional neural network," in *Futuristic Communication and Network Technologies: Select Proceedings of VICFCNT 2020*. Springer, 2022, pp. 341–349.

[40] R. Liu, A. Xiong, J. Lai, H. Zhang, and S. Wu, "Gesture recognition based on improved alexnet," in *2022 2nd International Conference on Electronic Information Engineering and Computer Technology (EIECT)*, 2022, pp. 257–260.

[41] S. Tang, Y. Zhu, and S. Yuan, "An improved convolutional neural network with an adaptable learning rate towards multi-signal fault diagnosis of hydraulic piston pump," *Advanced Engineering Informatics*, vol. 50, p. 101406, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034621001580

[42] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020.

[43] [Online]. Available: https://react.dev/learn

[44] [Online]. Available: https://nextjs.org/

[45] [Online]. Available: https://developers.google.com/mediapipe/solutions/guide

[46] [Online]. Available: https://www.tensorflow.org/js