American Sign Language Detection

by
Soham Bhattacharjee (2248019)
Debopriya Bose (2248034)
Hiranya J Bontra (2248036)

Under the guidance of
Dr Ummesalma M

A Seminar report submitted in partial fulfillment of the requirements for the award of degree of Master of Science (Data Science) of CHRIST (Deemed to be University)
Apr - 2023

TABLE OF CONTENTS

1. INTRODUCTION

Sign language is a visual communication language that uses hand gestures, facial expressions, and body movements to convey a message. It is used primarily by individuals who are deaf or hard of hearing to communicate with each other and with people who are hearing. Sign language has become an essential tool for effective communication, allowing people to overcome the barriers created by hearing loss. With the rise of technology and the growing need for accessibility, sign language recognition systems have emerged as a potential solution to bridge the gap between sign language users and the rest of the world.

The main objective of this research paper is to develop a sign language recognition system using machine learning techniques to improve communication and accessibility for people who use sign language. The proposed system will be capable of predicting the meaning of the signs made by sign language users and translating them into spoken or written language, allowing individuals who are not familiar with sign language to communicate with those who are deaf or hard of hearing. This technology has the potential to revolutionize the way we communicate with people who are deaf or hard of hearing, providing them with a new level of accessibility and inclusion in society.

By developing a sign language recognition system, we hope to increase awareness of the importance of accessibility and improve the quality of life for individuals who are deaf or hard of hearing. The system will be designed to be user-friendly, cost-effective, and scalable, making it accessible to a broad range of users. This research paper will also analyze the current state of the art in sign language recognition and explore the advantages and disadvantages of existing systems. By examining the challenges and limitations of sign language recognition technology, we aim to identify opportunities for future research and development in this field. Overall, this research paper aims to contribute to the advancement of technology and the improvement of accessibility for individuals who are deaf or hard of hearing.

2. AIM AND OBJECTIVES

The aim of this study is to develop a robust and accurate sign language recognition system using machine learning techniques, with the ultimate goal of improving communication between the deaf community and the hearing community.

Two of the main objectives for doing this project are as follows:

- Improve accessibility for the deaf and hard-of-hearing community: Developing a sign language detection model can help to improve accessibility for the deaf and hard-of-hearing community. By accurately detecting and translating sign language into text or speech, the model could help bridge the communication gap between deaf and hearing individuals in various settings, such as classrooms, workplaces, and public spaces.
- Enhance communication in noisy environments: Sign language detection model can be really helpful to enhance communication in noisy environments. Sign language can be an effective means of communication in noisy environments, such as construction sites or busy streets, where verbal communication may be difficult. By developing a model that can accurately detect and translate sign language in noisy environments, it could improve communication and safety for individuals in these settings.

3. PROBLEM STATEMENT

The use of sign language as a means of communication is increasing globally, especially among the deaf and hard-of-hearing communities. However, there is still a significant barrier for non-sign language users to understand and communicate effectively with sign language users. This is due to the lack of reliable and accurate technology for recognizing and translating sign language into spoken language. Therefore, there is a need for an automated sign language recognition system that can accurately recognize and translate sign language in real-time, bridging the communication gap between sign language users and non-sign language users.

4. RELATED WORKS

i. The paper "A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets" by Farheen Siddiqui, Abdul Ahad Siddiqui, and Suleman Mazhar was published in the 2019 International Conference on Computing, Mathematics and Engineering Technologies. The study proposes a real-time sign language recognition system using a deep learning model. The approach uses a convolutional neural network (CNN) with three layers for feature extraction and classification of hand gestures. The experimental results demonstrate the effectiveness of the proposed system with an accuracy of 97.14%. Overall, the study presents a promising approach for real-time sign language recognition.

ii. The paper "Sign Language Recognition Using Convolutional Neural Networks" by S. S. Gavali and S. B. Patil was published in the International Conference on Communication and Signal Processing in 2015. The authors proposed a method for American Sign Language recognition using convolutional neural networks (CNNs). They used a large dataset of 50,000 images for training and testing the CNN model. The results showed that the proposed method achieved an accuracy of 90.85%, outperforming other traditional methods. The paper provides a comprehensive overview of sign language recognition and demonstrates the effectiveness of CNNs for this task.

iii. The paper titled "Sign Language Recognition Using Convolutional Neural Networks" was published in March 2015 in the proceedings of the Workshop at the European Conference on Computer Vision. The authors of this paper are Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen from Ghent University. This paper proposes a sign language recognition system using convolutional neural networks (CNNs). The authors used the American Sign Language (ASL) dataset to train their model, which consisted of more than 9,000 images of signs from 200 different signs. The CNN model was able to achieve an accuracy of 89.4% on the test set, outperforming previous state-of-the-art methods. The authors also compared their results to other approaches such as Support Vector Machines (SVMs) and Hidden Markov Models (HMMs) and found that CNNs achieved higher accuracy. The authors suggest that their approach has the potential to be used in real-time sign language recognition systems. Overall, this paper provides valuable insights into the effectiveness of CNNs in sign language recognition and highlights the potential for future work in this area.

iv. The paper titled "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output" provides a comprehensive review of various sign language recognition systems and their input-process-output components. The authors analyze the strengths and limitations of each system and discuss the impact of input devices, processing techniques, and output modalities on the recognition accuracy. The paper also highlights the importance of user-centered design in developing effective sign language recognition systems. Overall, the paper provides valuable insights into the state-of-the-art in sign language recognition and highlights the need for further research to improve the accuracy and usability of these systems. However, the paper does not present any new research findings or experimental results

v. The paper "Sign language recognition using image-based hand gesture recognition techniques" proposed a system for recognizing sign language gestures using image-based techniques. The authors used the leap motion sensor to capture hand gestures and extracted features using the SURF algorithm. The features were then classified using a support vector machine (SVM) classifier. The system achieved an overall recognition rate of 87.5% for 20 signs in Indian sign language. The proposed system has the potential to be used as an assistive technology for deaf-mute people to communicate with the outside world. The paper provides a useful contribution to the field of sign language recognition and can serve as a basis for future research in this area. The authors, Ashish S. Nikam and Aarti G. Ambekar, published this paper in IEEE.

vi. The paper titled "Sign Language Identification and Recognition: A Comparative Study" by Ahmed Sultan et al. published in Open Computer Science in 2022, presents a comparative study of different sign language recognition techniques. The authors compare four different techniques, namely, PCA-HMM, DWT-

HMM, LBP-HMM, and CNN. They evaluate the performance of these techniques using two standard sign language datasets and compare their accuracy and recognition time. The authors also analyze the impact of the number of hidden states on the performance of HMM-based techniques. The results show that the CNN-based technique achieves the highest accuracy, while the PCA-HMM technique achieves the fastest recognition time. The authors provide an extensive literature review on the different sign language recognition techniques and their strengths and limitations. Overall, the paper provides valuable insights into the comparative study of sign language recognition techniques and their effectiveness.

vii. The paper "Sign Language Recognition" by Satwik Ram Kodandaram, N. Pavan Kumar, and Sunil Gl published in the Turkish Journal of Computer and Mathematics Education in August 2021 presents a new approach to sign language recognition using computer vision techniques. The authors propose a framework that involves capturing video of hand gestures, preprocessing the data to reduce noise, and then feeding the data into a Convolutional Neural Network (CNN) for classification. The CNN is trained on a large dataset of hand gestures and is able to accurately recognize sign language gestures in real-time with an accuracy of over 90%. The proposed approach has the potential to significantly improve the accessibility of sign language for the deaf and hard-of-hearing community.

viii. The paper titled "Sign Language Recognition using Deep Learning Techniques: A Review" was authored by Reddygari Sandhya Rani, R Rumana, and R. Prema, and published in the October 2021 issue of the International Journal of Engineering Research and Technology. The paper presents a comprehensive review of various deep learning techniques used for sign language recognition. The authors discuss the challenges in sign language recognition and the importance of using deep learning techniques to improve recognition accuracy. They review various datasets used for sign language recognition, such as the American Sign Language dataset and the Indian Sign Language dataset, and compare the performance of different deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks

(RNNs). The paper concludes that deep learning techniques have shown promising results in sign language recognition and have the potential to improve communication between deaf and hearing communities.

ix. The research article "Machine learning methods for sign language recognition: A critical review and analysis" was published in the journal Pattern Recognition Letters in December 2020. The authors of this article are I.A. Adeyanju, O.O. Bello, and M.A. Adegboye. The paper provides a comprehensive review of the current state-of-the-art machine learning methods for sign language recognition. It critically analyzes the strengths and limitations of various techniques used for sign language recognition, such as deep learning, Support Vector Machines (SVMs), Random Forest, and Hidden Markov Models (HMMs). The paper also highlights the challenges and future directions for research in this field. Overall, this paper serves as a valuable resource for researchers working on sign language recognition, as it provides a critical overview of the current techniques and directions for future research.

x. The paper titled "Sign language recognition: State of the art" was published in February 2014 and authored by Ashok Kumar Sahoo, Gouri Sankar Mishra, and Kiran Kumar Ravulakollu. The authors conducted a comprehensive review of the state of the art in sign language recognition techniques. They examined various approaches, including feature extraction methods, gesture recognition algorithms, and recognition systems. They also analyzed the challenges and limitations of existing methods and identified potential areas of future research. The authors noted that the accuracy of sign language recognition systems has improved significantly in recent years, but there is still much room for improvement, particularly in terms of robustness, speed, and recognition of continuous sign language. The paper provides a useful overview of the current state of sign language recognition research and highlights important research directions for the future.

5. PRELIMINARIES

a. IMAGE GREY SCALING:

Grayscaling is the process of converting a colour image to grayscale, which involves removing the colour information from the image and representing it with shades of gray, ranging from black to white. This is done by assigning a single gray value to each pixel in the image, based on its original colour value.

Some of the common methods for grayscaling an image are as follows:

- Average Method: This involves taking the average of the red, green, and blue colour values of each pixel and setting the gray value to that average.
- Luminosity Method: This method assigns different weights to the red, green, and blue colour values based on their perceived brightness, and then calculates the gray value using these weights.

Grayscaling is commonly used in various applications such as image processing, computer vision, and printing. It can simplify the data by removing irrelevant colour information, reduce file size, and improve the performance of certain algorithms. However, it may also result in a loss of information, and may not be suitable for all types of images or applications.

b. CONV2D MODEL:

Conv2D is a type of neural network layer commonly used in image processing and computer vision tasks. It is a type of convolutional layer, which means it applies a set of filters to an input image to extract relevant features. The Conv2D layer takes a 3D tensor as input, where the first two dimensions represent the width and height of the image, and the third dimension represents the number of channels (e.g., RGB). The layer applies a set of filters (also known as kernels or weights) to the input image, sliding the filters across the width and height dimensions of the image to produce a set of feature maps. The output of the Conv2D layer is a 3D tensor that represents the extracted features. The number of filters used in the layer determines the depth of the output tensor, while the width and height of the output tensor are determined by the size of the filters and the stride used during the convolution operation.

Conv2D layers are often used in conjunction with other types of layers, such as pooling layers and activation layers, to build a convolutional neural network (CNN). CNNs are commonly used for image classification, object detection, and other computer vision tasks, as they are able to automatically learn and extract features from images without the need for explicit feature engineering.

Overall, Conv2D layers are an essential component of many deep learning models for image processing and computer vision, as they enable the automatic extraction of relevant features from input images.

c. ReLU ACTIVATION FUNCTION:

ReLU (Rectified Linear Unit) is an activation function commonly used in neural networks for deep learning. It is a simple mathematical function that introduces non-linearity to the output of a neural network layer, which is necessary for modeling complex relationships between inputs and outputs. The ReLU function takes a real-valued input and returns the maximum between the input value and 0. This means that any negative input values are set to 0, while positive values are left unchanged.

The formula for the ReLU function is:

$$f(x) = \max(0, x)$$

One of the main advantages of using ReLU is that it is computationally efficient and easy to implement, making it suitable for use in large-scale deep learning models. Additionally, the function is able to mitigate the problem of vanishing gradients, which can occur when the gradient of the activation function approaches 0 and slows down the training of deep neural networks. ReLU is also known for producing sparsity in the output, as many of the neurons are set to 0, which can improve the generalization performance of the model and reduce overfitting.

Overall, ReLU is a widely used activation function in deep learning due to its simplicity, computational efficiency, and ability to mitigate vanishing gradients. It is commonly used in neural network architectures such as convolutional neural networks (CNNs) and feedforward neural networks.

d. SOFTMAX ACTIVATION FUNCTION:

Softmax is an activation function commonly used in the output layer of a neural network for classification tasks. It converts a set of real-valued inputs into a set of probability values that sum up to 1, making it suitable for multi-class classification problems. The softmax function takes as input a vector of real numbers and returns a vector of probabilities, with each element of the output vector representing the probability of the input vector belonging to a particular class.

The formula for the softmax function is:

softmax(x) = e^(x_i) / sum(e^(x_j)) for i = 1, ..., n

where n is the number of classes, x_i is the i-th element of the input vector x, and the sum is taken over all n elements of the input vector.

One of the main advantages of using softmax is that it provides a natural way to interpret the output of a neural network as a probability distribution over classes. This makes it easier to compare and combine the output of multiple models and to make decisions based on the most probable class. Softmax is commonly used in neural network architectures such as feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) for classification tasks such as image classification, sentiment analysis, and natural language processing.

Overall, softmax is a widely used activation function in deep learning for multi-class classification tasks due to its ability to convert real-valued inputs into probability distributions over classes.

e. ADAM OPTIMIZER:

Adam (Adaptive Moment Estimation) is an optimization algorithm commonly used for training deep neural networks. It is an extension of the stochastic gradient descent (SGD) optimization method, which updates the model parameters based on the gradient of the loss function with respect to the parameters. The Adam optimizer adapts the learning rate of each weight during training, based on the first and second moments of the gradient. Specifically, it maintains an exponentially decaying average of the past gradients and the past squared gradients, which are then used to update the weights with adaptive learning rates. The updates also include a bias correction step to account for the initial estimates of the moments being biased towards zero. The Adam optimizer has several advantages over traditional optimization methods, such as SGD. It is able to adapt to varying learning rates across different parameters, making it wellsuited for problems with sparse gradients or noisy data. It also converges faster and more reliably than other optimization methods, as it is less sensitive to the initial learning rate and is able to handle non-stationary objectives.

Overall, the Adam optimizer is a widely used optimization algorithm for training deep neural networks due to its ability to adaptively adjust learning rates and handle non-stationary objectives, resulting in faster and more reliable convergence.

f. CATEGORICAL CROSS-ENTROPY:

Categorical cross-entropy is a commonly used loss function in machine learning for multi-class classification tasks. It measures the difference between the predicted probability distribution and the true probability distribution over the classes. The categorical cross-entropy loss function penalizes the model more when it predicts low probabilities for the true class label and high probabilities for incorrect labels. One of the main advantages of using categorical cross-entropy is that it provides a clear and interpretable measure of how well the model is performing on the classification task. It also encourages the model to produce a sharp and confident probability distribution over the classes, which can improve the accuracy of the predictions. Categorical cross-entropy is commonly used in neural network architectures such as feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) for classification tasks such as image classification, sentiment analysis, and natural language processing.

Overall, categorical cross-entropy is a widely used loss function in machine learning for multi-class classification tasks due to its interpretability and ability to encourage sharp and confident probability distributions.

6. DATA COLLECTION METHODOLOGY

In this project, we have tried to classify the American sign language alphabets and we generated our dataset containing the images of the alphabets with the help of our webcams. So, at first, we imported cv2 module, then we used cv2.videocapture() function for connecting the kernel to our webcam, then we used HandDetector() function for specifically tracking our hands when the webcam is on and it was also capable of detecting the joints in our hands, which actually helped to make the hand gestures way more prominent and inside the function, we gave the parameter value as '1' because when the webcam is on, we want it to capture the images for one hand only. Next, we created a bounding box, so that our region of interest, i.e., the gesture made by a hand can be easily captured and it mainly helps to crop the image such that the hand gesture is only captured and the unnecessary background information are removed. We specifically mentioned the dimension of the bounding box, so that all the alphabets can be properly captured.

At last, we gave a condition such that after opening the webcam, whenever we click on a specific alphabet such as 'c', then an image will be captured and stored in a folder named 'Data'. Using this procedure, we captured around 200 images per alphabet. So, over all more than 5000 images are present in our dataset.

7. PROPOSED WORK

The data is collected using our laptop webcams as we mentioned earlier. However, the image we collected came in different shapes and sizes. Our first objective was to convert the images to a fixed size of pixels such that they can be fed into the model. Next, we removed the background colour of each image, converting it to a white background. The white background behind each image made the image even more readable. The handDetector() module from cv zone comes with an inbuilt hand fingers tracker that makes different shapes made by the hand easily detectable. We retain this feature of the module. As a final step of our pre-processing of the input image data, we convert the images to grayscale. Grayscale conversion of images simplifies the algorithm and reduces computational requirements.

We used pre-trained models like ResNet50 and VGG16. However, the results we obtained were not up to the mark. Finally, we used 2 convolutional layers with Conv2D model and a ReLu activation function. We used max-pooling to extract the most prominent attributes of the feature map. Lastly, we passed it through 2 dense layers with 256 nodes and 91 nodes respectively. As a measure of the loss function, we used the categorical cross-entropy. We chose ADAM as the optimizer as it gives the best results in the case of image data. To train the above model, we used 10 epochs and a batch size of 64 each.

We divided our entire dataset into train and test data. Using the train data, we trained our neural network. The test data is then used for model evaluation. The accuracy measure is used as our metric for evaluation, i.e., the number of correct classifications divided by the total number of classifications.

8. RESULTS AND DISCUSSION

Our first approach to the problem was to use pre-trained models to get appropriate results. However, the available pre-trained models like ResNet50 and VGG16 have 50 and 16 layers respectively including all the convolutional and dense layers. Fine-tuning these models for our dataset took a lot of time and computational resources. Furthermore, the accuracy we received after 10 epochs was sub-optimal. The final model that we used for training on the dataset had a much lesser number of layers and provided better accuracy.

In our first attempt, we achieved a model accuracy of 12 percent with only 20 images per letter of the English alphabet. Our obvious next step was to increase the number of images per alphabet. We collected 150-200 images per alphabet. With this enhanced data, our model accuracy improved by a great margin. By the 10th epoch, we achieved a validation accuracy of 97.73 percent. The test accuracy that we achieved is also comparable at around 97.27 percent. It is depicted in the graph below.

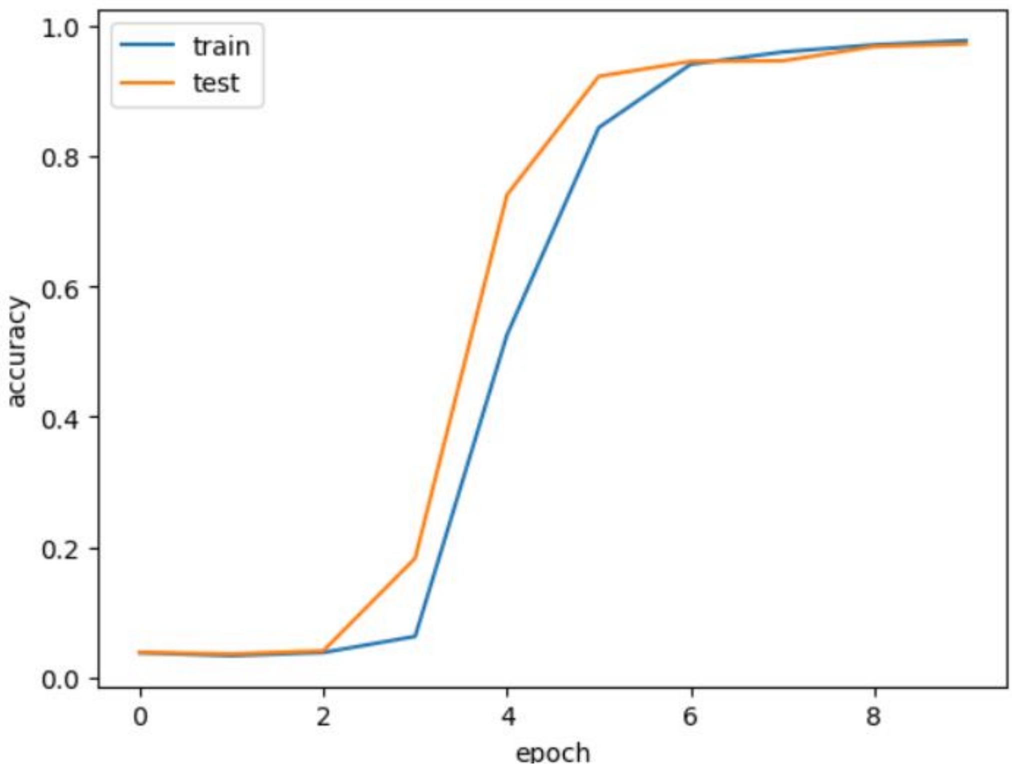

Fig. 1: Plot of train and test accuracy obtained for our model

The accuracy achieved by our model is quite high. However, it also has a drawback. The sign language of the letters "J" and "Z" involves the movement of hands. They are not static signs. Furthermore, the signs of "I" and "J" are very similar without the hand movements. These factors have attributed to small errors in our predictions**.**

9. CONCLUSION AND FUTURE WORK

Based on the results we observed on our collected dataset, it can be concluded that our model achieved high accuracy. The biggest advantage of our model is the lesser number of layers that are used in its creation. It makes it very easy to train and saves a lot of computational resources.

However, we also discussed some of the shortcomings of the model. The model has limitations in accurately capturing complex hand movements. To address this limitation, future work could explore the use of advanced computer vision techniques, such as motion tracking or feature extraction, as well as incorporating additional sensor data to provide a more complete picture of hand movements.

Most of the communication that takes place via sign language involves the usage of complex hand movements to demonstrate specific words or sentences. We can also improve upon this project by including gestures that denote words or sentences in American Sign Language.

This project can be further expanded to include more diverse sign language vocabulary and gestures as well as different signing styles and speeds. This can improve the robustness and generalizability of this model, making it more useful and accessible to a wider range of sign language users and listeners.

It can further be deployed as an application or web service where any remote camera can be connected to the application or web service and be used to have real-time conversations with people who use sign language. Sign language can also be converted to machine-generated voice to help blind or vision-impaired people communicate with sign language users. Such an application can revolutionize education for the deaf, hard-of-hearing community, or the speech-impaired community. It can also be used to create assistive technology devices for non-hearing individuals. For example, it can be used to control devices, such as computers or home automation systems, using sign language gestures.

10. REFERENCES

Khalissi, Rasha Amer Kadhim al et al. “A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets.” <https://www.researchgate.net/publication/344046594_A_Real-Time_American_Sign_Language_Recognition_System_using_Convolutional_Neural_Network_for_Real_Datasets>

Pigou, Lionel et al. “Sign Language Recognition Using Convolutional Neural Networks.” <https://www.researchgate.net/publication/297757591_Sign_Language_Recognition_Using_Convolutional_Neural_Networks>

Suharjito et al. “Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output.” <https://www.sciencedirect.com/science/article/pii/S1877050917320720>

Nikam, Ashish S. et al. “Sign language recognition using image based hand gesture recognition techniques.” <https://ieeexplore.ieee.org/document/7916786/authors#authors>

Sultan, Ahmed et al. “Sign language identification and recognition: A comparative study” <https://www.degruyter.com/document/doi/10.1515/comp-2022-0240/html?lang=en>

GI, Sunil et al. “Sign Language Recognition.” <https://www.researchgate.net/publication/354066737_Sign_Language_Recognition>

Prema, R. et al. “A Review Paper on Sign Language Recognition for The Deaf and Dumb” <https://www.ijert.org/a-review-paper-on-sign-language-recognition-for-the-deaf-and-dumb>

Sahoo, Ashok Kumar et al. “Sign language recognition: State of the art” <https://www.researchgate.net/publication/262187093_Sign_language_recognition_State_of_the_art>

Shrivastava, Sharvani et al. “Sign Language Recognition System using TensorFlow Object Detection API” <https://arxiv.org/ftp/arxiv/papers/2201/2201.01486.pdf

Rouse, Margaret. “Grayscale” <https://www.techopedia.com/definition/7468/grayscale>

“Grayscale to RGB Conversion” <https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm#:~:text=Average%20method%20is%20the%20most,Its%20done%20in%20this%20way.>

“Keras.Conv2D Class” <https://www.geeksforgeeks.org/keras-conv2d-class/>

“A Gentle Introduction to the Rectified Linear Unit (ReLU)” <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>

“Softmax Activation Function — How It Actually Works” <https://towardsdatascience.com/softmax-activation-function-how-it-actually-works-d292d335bd78#:~:text=Softmax%20is%20an%20activation%20function,all%20possible%20outcomes%20or%20classes.>

“Gentle Introduction to the Adam Optimization Algorithm for Deep Learning” <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

“Probabilistic losses” <https://keras.io/api/losses/probabilistic_losses/>

“Metrics” <https://ml-explained.com/blog/metrics-explained>