A logo for a university

Description automatically generatedThe Egyptian E-learning University

Faculty of Information Technology

Sign language recognition project.

## GRADUATION PROJECT

**­PREPARED BY**

**Moheb Gamil Nafea (20-01590)**

**Omar Ahmed Hussien (20-00329)**

**Youssef Sami Moussad ( 20-01473)**

**Mirna Michel Fawzi (20-01581)**

**Mery Sameh Faheem (20-00013)**

**Malak Ahmed Mohamed (20-00671)**

**Yasmin Muhammad Zenhum (19-00249)**

**Under the supervision of**

**Prof.Dr:Hanaa Bayomi Eng:Zienab Moustafa**

**Sohag**

**2023-2024**

Table of Contents

[**GRADUATION PROJECT** 1](#_Toc168971301)

[**Introduction** 2](#_Toc168971302)

[**Abstract** 3](#_Toc168971303)

[**Motivation** 4](#_Toc168971305)

[**problem statement** 6](#_Toc168971306)

[**Related Work** 7](#_Toc168971311)

[**Papers** 8](#_Toc168971312)

[**Methodology** 13](#_Toc168971313)

[**Support Vector Machine (svm) Algorithm** 15](#_Toc168971314)

[**K-Nearest Neighbors (K-NN) Algorithm** 17](#_Toc168971315)

[**Convolutional Neural Network (CNN)** **Algorithm** 19](#_Toc168971316)

[**Graphical user interface (GUI)** 22](#_Toc168971317)

[**Model and application Integration** 33](#_Toc168971318)

[**Tools** 38](#_Toc168971319)

[**Work Plan** 39](#_Toc168971320)

[**Refrences** 41](#_Toc168971340)

**Introduction**

Communication is very crucial for all human beings since it enables us to pass the information to others and also to understand what is said to us.

In simpler words, communication is the transmission of meaning between a sender and a receiver. Those who have issues in speaking make it impossible for them to speak for communication.

To overcome this, there is a visual communication system called sign language which uses hand gestures to translate the gestures into verbal language.

According to the World Health Organization (WHO), around 466 million people with hearing loss issues, and 34 million of them are children.

It is claimed that by 2050 over 900 million people will have suffering hearing loss Hard hearing people can hear up to a specific limited degree and unobvious by a hearing aid. In contrast, deaf people cannot listen entirely due to head trauma, noise exposure, disease, or genetic condition.

Sign language is the means of communication between the deaf themselves and with ordinary people, and every country has its own language.

**Abstract**

Sign languages, a natural language based on vision, are used by deaf people to communicate.

Because most hearing people are unable to understand sign language ,Sign language interpretation has become an essential means of communication for both deaf and hearing people.

An application that translates written English texts into sign language. It is used by deaf individuals. However, it is also used by some hearing individuals, most notably deaf family, and relatives, as well as interpreters who help deaf people and wider groups to communicate.

Deaf sign users use sign language, which is a structured language ,with its own syntax and structure, every movement has a specific meaning and symbol to it

**Motivation**

Communication is always having a great impact in every domain in our lives and every person has the right to express his opinion ,needs and ideas.

So, sign language is a language learned by deaf and dumb people, and it is a language unknown to ordinary people.

Therefore, the main motivation is to learn this language and make the lives of deaf people easier and to enhance their lives and confidence when communicating with those who do not understand their sign language at the same time helping other people to easily communicate with deaf people.

**problem statement**

**Problem:** Deaf individuals face challenges in communicating with others, especially when interacting with hearing individuals. Expressing their thoughts and needs effectively can be difficult for the deaf community.

**Objective:**

Our project aims to develop an application that can translate sign language gestures into written letters .

application will be user-friendly and intuitive.

**Expected Benefits:**

1. Improve communication between the deaf and hearing communities.
2. Empower the deaf to express themselves better in society.
3. Provide an easy way for real-time sign language translation.

**Potential Challenges:**

1. Accurately detecting sign language gestures.
2. Minimizing translation delays.
3. Designing a user interface that is intuitive and easy to use

**Related Work**

**Our data set:**

A dataset is an organized collection of image data containing numbers from 1 to 9 and letters from a to z and Three words or Sentences are Best of luck , Thank you , I love you.

**Data set size:**

* **26** MB

**Number of images**:

* The Whole Dataset has a number of Image **46800** images

**Number of Categories:**

* **39** categories
* Each category has a number of images 1200 images.

**Source:**

# References (MAKWANA 2020)

**Papers**

1. Abstract:

Sign language is a visual language used.

by the people with speech and hearing disabilities for communication in their daily conversation activities.

It is completely an optical communication language through its native grammar, unlike fundamentally from that of oral languages.

In this research paper, presented an optimal approach, whose major objective is to accomplish the transliteration of 24 static sign language alphabets and numbers of American Sign Language into

humanoid or machine decipherable English manuscript.

Pre-processing operations of the signed input gesture are done in the first phase.

In the next phase, the various region properties of pre-processed gesture image are computed.

In the final phase, based on the properties calculated.

of earlier phase, the transliteration of signed gesture into text has been carried out.

This paper also presents the statistical result evaluation with the comparative graphical depiction of existing techniques and proposed techniques.

(Ss and Dr.Srinath S 2018)

1. Abstract

Sign language is intentionally designed to allow deaf and dumb communities to convey messages and to connect with society. Unfortunately, learning and practicing sign language is not common among society; hence, this study developed a sign language recognition prototype using the Leap Motion Controller (LMC).

Many existing studies have proposed methods for incomplete sign language recognition, whereas this study aimed for full American Sign Language (ASL) recognition, which consists of 26 letters and 10 digits. Most of the ASL letters are static (no movement), but certain ASL letters are dynamic (they require certain movements).

Thus, this study also aimed to extract features from finger and hand motions to differentiate between the static and dynamic gestures.

The experimental results revealed that the sign language recognition rates for the 26 letters using a support vector machine (SVM) and a deep neural network (DNN) are 80.30% and 93.81%, respectively. Meanwhile, the recognition rates for a combination of 26 letters and 10 digits are slightly lower, approximately 72.79% for the SVM and 88.79% for the DNN.

As a result, the sign language recognition system has great potential for reducing the gap between deaf and dumb communities and others. The proposed prototype could also serve as an interpreter for the deaf and dumb in everyday life in service sectors, such as at the bank or post office.

(chong and Boon-Giin Lee 2018)

1. Abstract:

The population of deaf-mute in Taiwan is increasing every year. These deaf-mute people usually use sign language to communicate with each other.

However, most hearing people cannot understand sign language because they have not learned it.

In order to let normal people understand sign language, a sign language recognition system is necessary.

In this paper, such a sign language recognition system is proposed.

In the proposed system, a depth image is captured by Kinect sensor, palm area is segmented from depth image, palm binary image is thresholder by using Otsu thresholding method, and background noises are removed by morphological closing operators, and SURF features and descriptors are extracted to identify the sign language.

Experimental results show that the proposed method

is effective to detect and identify the numerals and letters in the English Alphabet used in the sign language.

(Yeh, et al. 2017)

**Related Work**

**Suggested Datasets**

|  |  |  |  |
| --- | --- | --- | --- |
| paper name | RAHUL MAKWANA | VAISHNAVI SONAWANE | RUSLAN BREDUN |
| Year of publication | 2020 | 2021 | 2022 |
| Dataset name | American Sign Language recognition | Indian Sign Language Dataset | Sign Language (ENG Alphabet) |
| Dataset size | 50 MB | 281MB | 2GB |
| Used Model | Machine learning Sequential | Machine learning Sequential\_3 | Machine learning Sequential\_15 |
| Accuracy | 97-100 % | 78% | 94-99 % |

(MAKWANA 2020)

(SONAWANE 2021)

(BREDUN 2022)

**Methodology**

A screenshot of a computer

Description automatically generated**Proposed Architecture:**

* Dataset
* Preprocessing
* machine learning algorithms
* model
* Output

out put

**1. Problem Statement:**

* Define the objective of the project, which is to recognize sign language gestures using machine learning techniques.

**2. Data Collection:**

* Gather a diverse dataset of sign language gestures. The dataset should include various hand configurations, movements, and orientations for different sign language expressions.

**3. Data Preprocessing:**

* Load images from the specified directory.
* Resize the images to a standard size, such as (150, 150, 3), using the skimage library.
* Flatten the resized images to create a feature vector.
* Create the input array (flat\_data\_arr) and the output array (target\_arr) for model training.

**4. Model Training:**

* Split the dataset into training and testing sets using train\_test\_split.
* Instantiate a machine learning model, such as a K-Nearest Neighbors (KNN) classifier.
* Train the model on the training data.
* Evaluate the model's performance on the testing data using metrics like accuracy, precision, recall, and F1-score.

**Support Vector Machine (svm) Algorithm**

**Overview**

* A Support Vector Machine (SVM) is a supervised machine learning algorithm.
* Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification and regression
* SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection.
* The main objective of the SVM algorithm is to find the optimal [hyperplane](https://www.geeksforgeeks.org/separating-hyperplanes-in-svm/) in an N-dimensional space that can separate the data points in different classes in the feature space.

**Objectives**

* SVM aims to separate data points into different classes.
* The hyperplane maximizes the margin (distance) between the closest points of different classes.

**How it Works**

* **Data Loading** Read images from the dataset directory
* **Data Splitting** Divide the dataset into training and testing sets
* **Model Training** Train the SVM model using the training dataset. The model learns to classify images based on the training data.
* **Model Evaluation** Use the trained SVM model to make predictions on the test dataset.
* **Model Saving** Save the trained SVM model to a file for later use.
* (Support Vector Machine (SVM) Algorithm 2023)

**K-Nearest Neighbors (K-NN) Algorithm**

A diagram of a class

Description automatically generated**Overview**

* K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for Classification problems.

**Objective**

* The objective of the K-Nearest Neighbors (KNN) algorithm is to classify data points by identifying the majority class among their k nearest neighbors in the feature space.

**How it Works**

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready save it.
* (K-Nearest Neighbor(KNN) Algorithm 2024)

A diagram of a layer

Description automatically generated**Convolutional Neural Network (CNN)** **Algorithm**

**Overview:**

A Convolutional Neural Network (CNN) is a type of deep learning model specifically designed for processing structured grid data, such as images. CNNs are particularly powerful for tasks involving image recognition, classification, and computer vision because they are capable of capturing spatial hierarchies in data.

**Components of a CNN:**

**1**. **Convolutional Layers:**

- **Convolution Operation**: The core idea is to apply a set of filters (or kernels) over the input data to produce feature maps. Each filter slides over the input data (such as an image), performing an element-wise multiplication and summation, which detects various features like edges, textures, or patterns.

- **ReLU Activation**: A non-linear activation function (Rectified Linear Unit) is often applied after the convolution operation to introduce non-linearity into the model.

**2**. **Pooling Layers:**

- **Max Pooling**: Reduces the spatial dimensions of the feature maps by taking the maximum value in a set of small patches (usually 2x2) from each feature map. This reduces the computational load and helps to make the detection of features invariant to small translations.

- **Average Pooling**: Takes the average value instead of the maximum value from the patches.

**3**. **Fully Connected Layers:**

- After several convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers. These layers connect every neuron from one layer to every neuron in the next layer, similar to a traditional neural network.

**4**. **Output Layer:**

- For classification tasks, the output layer typically uses a softmax activation function to produce probability distributions over the possible classes.

**Key Concepts**

- **Receptive Field**: The region in the input data that a particular CNN's neuron is sensitive to. As you move deeper into the network, the receptive fields typically become larger.

- **Stride**: The step size with which the convolution filter moves across the input. Strides greater than one reduce the spatial dimensions of the output.

- **Padding**: Adding zeros around the input matrix to ensure that the filters can fully cover the borders of the input image.

(Introduction to Convolution Neural Network 2024)

**Graphical user interface (GUI)**

1. **Introduction**

* **Application Description**: The application translates sign language into letters and numbers using the camera and photo gallery.
* **Application Objective**: Facilitate communication for individuals using sign language by converting it into written text.
* **Target Audience**: Individuals who communicate using sign language, interpreters, and those interested in learning sign language

1. **System Requirements**

* **Supported Devices**: Smartphones and tablets.
* **Operating System**: iOS and Android.
* **Software Requirements**: Flutter SDK, image recognition libraries, and camera access

1. **Application Architecture**

* **Overview**: The application consists of a front-end built with Flutter integrated with the camera and photo gallery, along with image recognition algorithms.
* **Key Components**:
* **Frontend**: A
* ll screens and user interactions.
* **Backend**: Image recognition algorithms (KNN , SVM and CNN)

1. **Screens**

**List of Screens**: The application consists of six main screens.

**Description of Each Screen**:

**First Screen: Splash Screen**:

**Purpose**: Display an animation of a circle and a button to navigate to the application.

**Components**: Animated circle, Button.

A screenshot of a cellphone

Description automatically generated**Interactions**:

On button press, navigate

to the next screen.

**Key Components**

1. Animation Controller

2. Size and Color Animation

3. Custom Painter

4. Navigation Button

5. UI Layout

**Packages Used**

**Material Design Package**

* **Purpose**: Provides the essential UI components and styling for building the application's interface.
* **Details**: The SplashScreen leverages several widgets from the material. Dart package, including Scaffold, AppBar, Center, Column, ElevatedButton, and more.

**User Interaction**

* **Animation**: The splash screen features a continuously pulsating circle that captures the user's attention immediately upon opening the app.
* **Navigation**: A prominent "Start" button encourages users to move forward and begin using the app, leading them to the HowToUse page.

**Summary**

The SplashScreen is designed to provide a visually engaging and smooth entry point into the application. It combines size and color animations with a custom-painted circle to create a dynamic visual effect. Additionally, a styled button guides users to the main content, ensuring an intuitive and pleasant user experience from the very start.

**Second Screen: How to Use the App**:

**- Purpose**: Teach users how to use the application.

**- Components**: Instructional text, Button to open the camera.

A screenshot of a phone

Description automatically generated**- Interactions**:

On button press, open the camera.

**Key Components**

1. Background and AppBar

2. Instructional Container

3. Navigation Button

4. UI Layout

**Packages Used**

**Material Design Package**

User Interaction

**Summary**

The HowToUse screen serves as a helpful guide for users, providing concise instructions on how to utilize the camera for translating sign language. It features a visually appealing design with clear text and an easily accessible navigation button. This screen ensures that users understand how to proceed with the app's functionality, making the user experience intuitive and straightforward.

**Third Screen: Image Source Selection**:

* A screenshot of a phone

  Description automatically generated**Purpose**: Allow users to choose whether to capture an image or select one from the gallery.
* **Components**: Instructional text, Two buttons (Camera, Gallery).
* **Interactions**: On camera button press, open the camera; on gallery button press, open the gallery.

A screenshot of a phone

Description automatically generatedA close up of a hand

Description automatically generated**Key Components**

* **ImagePickerService Instance**:
* pickImageFromGallery Method
* pickImageFromCamera Method
* UI Layout (**AppBar**: **Buttons**:)
* Navigation

**Packages Used**

* image\_picker
* material
* User Interaction
* Summary
* The CameraUI screen and ImagePickerService class provide a seamless user experience for selecting images from the camera or gallery. This functionality is essential for the app's purpose of translating sign language images. The CameraUI offers a straightforward interface with clear navigation and actions, ensuring an intuitive process for users.

**Sixth Screen: Display and Translate Image**:

* **Purpose**: Display the selected image and translate it.
* **Components**: Container to display the selected image, Container to display the translation.
* **Interactions**: Show the translated text of the image.

A close-up of a hand

Description automatically generatedA close-up of a hand on a keyboard

Description automatically generated­

**Key Components**

**Stateless Widget**

AppBar

Body Layout

Body Layout

**Summary**

The FinalResult screen provides a clear and organized layout for displaying the selected image and a placeholder for model results. This screen serves as the final step in the image translation process, offering users a visual representation of the image they have chosen. It is designed to be simple and informative, with dynamic sizing to accommodate different screen sizes.

**5- Workflow**

* **User Flow**: The user starts at the splash screen, moves to the instructions screen, selects an image source (camera or gallery), captures or selects an image, and finally views and translates the image.
* **Use Cases**: An example use case is a user wanting to translate a specific sign; they start by capturing the sign using the camera and then view the translated text.

**6- Features**

* **Core Features**: Translate sign language into letters and numbers, user-friendly interface, camera and gallery support.
* **Special Functions**: Image recognition using AI, support for various sign languages.

**7- Challenges and Solutions**

* **Challenges**: Accuracy of sign recognition, real-time image processing.
* **Solutions**: Using advanced image recognition libraries, improving processing algorithms.

**8. Evaluation**

* **Application Testing**: The application was tested on various devices to ensure performance and responsiveness.
* **User Feedback**: Collected user feedback to improve performance and interface.

**9. Future Enhancements**

* **Improvement Suggestions**: Adding support for more sign languages, improving recognition accuracy.
* **Development Plans**: Expanding the sign language database, adding educational features.

**10. Appendices**

* **Source Code**: Include links to source code or relevant code snippets.
* **References**: All sources and references used.
* **Additional Documents**: User surveys, design diagrams.

**Model and application Integration**

**11.Steps to Integrate the Model with the Application**

There are 3 ways to connect model AI by flutter application

1. **TFLite (TensorFlow Lite):**
2. **What is TFLite?**
   * TFLite is a lightweight version of the TensorFlow library, an open-source library used for developing artificial intelligence (AI) models, including deep neural networks and machine learning models. TFLite is specifically designed to be lightweight and fast, making it suitable for mobile applications and devices with limited resources like smartphones and tablets.
3. **How does TFLite work?**
   * TFLite can be used to load and run TensorFlow models efficiently in Flutter applications. The key steps to use TFLite are:
     + Train the AI model using TensorFlow and convert it to the TFLite format optimized for mobile devices.
     + Embed the converted model files into your Flutter application.
     + Use the TFLite Flutter package to load and run the models in your application.
4. **What can be done using TFLite and Flutter?**
   * TFLite can be used in Flutter applications for a wide range of purposes, such as:
     + Image recognition: TFLite can be used to load image recognition models and apply them in Flutter applications, such as recognizing objects and people in images.
     + Text classification: TFLite can be used to load text classification models and use them in Flutter applications, such as classifying text messages.
     + Speech recognition: TFLite can be used to load speech recognition models and use them in Flutter applications, such as converting speech to text.
     + Other functionalities: TFLite can be used in Flutter applications for purposes like data prediction, music composition, face recognition, and more

**2-APIs (Application Programming Interfaces):**

1. **What are APIs?**
   * APIs (Application Programming Interfaces) are sets of rules and protocols that allow different software applications to communicate with each other. APIs define the methods and data formats that applications can use to request and exchange information. They are essential for integrating different systems and services and enabling communication between them.
2. **Types of APIs:**
   * There are several types of APIs, including:
     + Web APIs: These are APIs that are accessed over the web using HTTP protocols. They are commonly used for communication between web-based applications and services.
     + RESTful APIs: Representational State Transfer (REST) APIs are a type of web API that follows the REST architectural style. They use standard HTTP methods like GET, POST, PUT, DELETE for communication and typically transfer data in JSON or XML format.
     + SOAP APIs: Simple Object Access Protocol (SOAP) APIs are another type of web API that uses XML-based messaging protocols for communication.
     + GraphQL APIs: GraphQL is a query language and runtime for APIs that allows clients to request specific data from the server.
3. **Using APIs with Flask:**
   * Flask is a lightweight web framework for Python that is commonly used for building web applications and web APIs. With Flask, you can easily create RESTful APIs that expose endpoints for clients to interact with your application or access data. Flask provides tools and libraries for handling HTTP requests, parsing request data, and returning responses in JSON or other formats.

**3-Flask:**

1. **What is Flask?**
   * Flask is a micro web framework written in Python. It is designed to be simple, lightweight, and easy to use, making it a popular choice for building web applications and APIs. Flask provides essential features for web development, such as routing, request handling, template rendering, and more.
2. **Features of Flask:**
   * Some key features of Flask include:
     + Routing: Flask allows you to define URL routes and map them to functions that handle HTTP requests.
     + Request Handling: Flask provides tools for handling HTTP requests, accessing request data (parameters, headers, etc.), and processing form submissions.
     + Template Rendering: Flask supports template engines like Jinja2 for rendering dynamic HTML content.
     + Extensions: Flask has a wide range of extensions available for adding additional functionality such as database integration, authentication, RESTful APIs, etc.
3. **Building APIs with Flask:**
   * Flask can be used to build RESTful APIs by defining routes that handle different HTTP methods (GET, POST, PUT, DELETE) and returning JSON or other data formats as responses. can use Flask extensions like Flask-RESTful or Flask-RestPlus to simplify API development and add features like request parsing, input validation, documentation generation, etc.

**Tools**

* Python
* SVM
* KNN
* CNN
* Flutter
* Dart
* flask
* TensorFlow
* Canava
* google dataset.
* Kaggle
* Figma
* Collab
* Google Scholar
* Jupiter

**Conclusion**

Finally, we have done everything we can to remove the obstacles facing the hearing-impaired person.

In order to help them integrate into society, not to make them a strange group in our society, they cannot communicate with us, and we cannot communicate our responses to them.

So that our project will be a link that facilitates dealings between them and anyone who cannot understand sign language. We value them and strive to make their lives easier.

The Sign Language Recognition Project successfully bridges the communication gap between the deaf and hearing communities by translating sign language gestures into written text.

Despite initial challenges in gesture detection and translation accuracy, the project achieved its objectives through advanced image recognition and refined algorithms.

The application supports both camera and gallery inputs, providing real-time, accurate translations.

**Work Plan**

**Phase 1:**

**7 Stages**

* Studing introduction to Python,Ai and ML
* Find Suitable Dataset
* Application UI
* Support Vector Machine (SVM)
* K-Nearest Neighbor (KNN) Algorithm
* Documentation
* A screenshot of a computer

  Description automatically generatedPresentation

**Phase 2:**

**7 Stages**

* + solving code issues
  + Studying Deep Learning
  + We program mobile application
  + Apply Deep Learning Model Architecture
  + Integrate DL Model with Mobile Application
  + Documentation
  + Presentation

A screenshot of a computer

Description automatically generated

**Refrences**

# Bibliography

BREDUN, RUSLAN. 2022. *Sign Language (ENG Alphabet).* Accessed December 2023. https://www.kaggle.com/datasets/ruslanbredun/sign-language-eng-alphabet.

chong, Teak-Wei, and Boon-Giin Lee. 2018. *American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach.* october 18. Accessed november 2023. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6210690/.

2024. *Introduction to Convolution Neural Network.* march. Accessed march 2024. https://www.geeksforgeeks.org/introduction-convolution-neural-network/?ref=lbp.

2024. *K-Nearest Neighbor(KNN) Algorithm.* January. Accessed january 2024. https://www.geeksforgeeks.org/k-nearest-neighbours/?ref=lbp.

MAKWANA, RAHUL. 2020. *American Sign Language recognition.* Accessed December 2023. https://www.kaggle.com/datasets/rahulmakwana/american-sign-language-recognition/data.

SONAWANE, VAISHNAVI. 2021. *Indian Sign Language Dataset.* Accessed 2023. https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset.

Ss, Shivashankara, and Dr.Srinath S. 2018. *American Sign Language Recognition System: An Optimal Approach.* August . Accessed November 2023. https://www.researchgate.net/publication/326972551\_American\_Sign\_Language\_Recognition\_System\_An\_Optimal\_Approach.

2023. *Support Vector Machine (SVM) Algorithm.* june. Accessed january 2024. https://www.geeksforgeeks.org/support-vector-machine-algorithm/?ref=lbp.

Yeh, Wan-Yi, Teng-Hui Tseng, Jun-Wei Hsieh, and Chun-Mi. 2017. *Sign language recognition system via Kinect: Number and english alphabet.* March 9. Accessed November 2023. https://ieeexplore.ieee.org/abstract/document/7872966.