**“SIGN LANGUAGE RECOGNISITION”**

CAPSTONE PROJECT REPORT

DATA ANALYSIS FOR BUSINESS (DAB402)

**A picture containing text

Description automatically generated**

Under Supervision of

Francisco Guevara

Sarama Shehmir

Submitted By

GROUP 7

Varun Kanwar (0805362)

Arun Kalsi (0806536)

Nizamudin (0800621)

**Introduction:**

Sign language recognition technology can be used to enable communication between deaf and hearing individuals, and can also be used for education, training, and accessibility purposes. Sign language recognition systems use a combination of computer vision and machine learning techniques to analyze signs and symbols of sign language gestures and translate them into text or speech. The goal of this project is to develop a sign language recognition system using the CNN and VGG16 architecture.

**Problem Statement:**

Dumb people communicate via hand signals, therefore normal people have difficulty identifying their language through signs. As a result, technologies that identify various signs and deliver information to ordinary people are required

**Objective:**

The main objective of our project is to identify symbolic expression through images so that communication gap between the deaf and normal people can be eliminated.

**Important Ethics to follow in making of SLR:**

* Accessibility: Sign language recognition technology should be built to make it easier for deaf and hard-of-hearing people to communicate. This technology should not be used to remove human interpreters or to save money by not paying interpreters.
* Cultural Sensitivity: While creating sign language recognition technology, it is critical to be culturally aware.
* Fairness: The technology should be designed to be fair and not discriminate against any group of individuals.

**Methodology:**

In our Sign Language Recognition project we uses two Neural Network Models i.e

1. CNN (Convolutional Neural Network)
2. VGG16

We are going to train both the models with our dataset which includes collection of sign language images acquired from websites and enhanced with other images to increase diversity. The collection contains images of sign language gestures for each letter of the alphabet, as well as some common phrases.

**Dataset**:

Basically our dataset is a csv file which is in the form of pixels and labels and each label has more than 500 pixel values .

These two are our dataset s out of which one is train and other is test[..\capstone dataset\sign\_mnist\_test.csv](file:///C:\Users\pavil\capstone%20dataset\sign_mnist_test.csv) and [..\capstone dataset\sign\_mnist\_train.csv](file:///C:\Users\pavil\capstone%20dataset\sign_mnist_train.csv)

**CNN (Convolutional Neural Network):**

CNN stands for Convolutional Neural Network, which is a type of deep learning algorithm used for image recognition and processing. CNNs are based on artificial neural networks and use a series of convolutional layers to extract features from input images. These features are then used to make predictions or classifications about the image. CNNs are designed to be highly effective at image recognition and processing, due to their ability to identify patterns and features within images. The convolutional layers of a CNN use filters to identify features such as edges, corners, and textures within an image. These features are then combined and used to make predictions about the contents of the image.

**Implementation**

Convolutional layers: These layers apply a set of filters to the input image to extract features such as edges, corners, and textures. The filters are small-sized matrices that slide over the input image, performing dot products with the pixels in the region covered by the filter. This operation generates a feature map that highlights the presence of the feature in the input image.

Pooling layers: These layers downsample the feature maps by reducing their spatial dimensions. The most common type of pooling layer is the max-pooling layer, which selects the maximum value within each region of the feature map.

Fully connected layers: These layers take the flattened output of the previous layers and perform the final classification or regression task.

During the training process, the CNN learns to adjust the filter weights to minimize the loss function, which measures the difference between the predicted output and the ground truth. This process is known as backpropagation, and it updates the weights of the filters to improve the accuracy of the model.

CNNs are highly effective in processing large and complex datasets, such as images and videos, due to their ability to extract meaningful features and patterns from the input data. With the use of techniques such as transfer learning, CNNs can be trained on a small dataset and still achieve high accuracy by leveraging pre-trained models on larger datasets.

In CNN we used a series of libraries such as NumPy, matplotlib ,panda ,TensorFlow for the successful implementation of this model.

In this we followed various steps like:

* Importing necessary Libraries.
* Reading the data.
* Doing EDA(display images, checking frequency distribution of alphabets)
* After this we build Model and in last we test and trained it.

In this we can see that we have displayed some alphabets using the display\_images function where x is used as pixel value and y as label.

Graphical user interface, application, PowerPoint

Description automatically generated

We used the Keras library in Python to build a CNN model. The model had four convolutional layers followed by two fully connected layers. We used the rectified linear unit (ReLU) activation function and max pooling to reduce the dimensionality of the feature maps.

We also used wrapper function for creating convolution and pooling layer.

In the training process total hyperparameters are used :

Learning rate , epochs , batch size , display step

* learning\_rate: This is the step size used to update the model weights during backpropagation. A smaller learning rate can result in slower training but more accurate results, while a larger learning rate can result in faster training but less accurate results.
* Epochs: This is the number of times the entire dataset is passed through the neural network during training. Increasing the number of epochs can lead to better performance, but may also increase the risk of overfitting.
* batch\_size: This is the number of samples that are processed by the neural network in each iteration of training. A larger batch size can lead to faster training, but may also result in lower accuracy.
* display\_step: This is the frequency with which the training progress is displayed during training. It is typically set to a multiple of the batch size to ensure that progress is displayed at meaningful intervals.

After Training and Testing of the dataset with CNN we got

Accuracy on Training Data: 99.86408352851868 %

Accuracy on Test Data: 99.65787529945374 %

By seeing this we can say that our model is working well on the data

Text

Description automatically generated

**VGG16:**

The VGG16 model is a convolutional neural network (CNN) that is widely used in image recognition tasks. We will use this model to recognize the sign language gestures in imagesTo train the VGG16 model on sign language gestures, we will use a dataset of images and videos of people signing different words and phrases. The dataset will be divided into training, validation, and testing sets. The model will be trained on the training set and validated on the validation set to ensure that it is not overfitting. Once the model is trained and validated, it will be tested on the testing set to evaluate its accuracy.

Once the model is trained and tested, we will use it to recognize sign language gestures in real-time. We will use a camera to capture video of a person signing and use the VGG16 model to recognize the gestures in the video. We will then translate the recognized gestures into text using a lookup table that maps each gesture to a word or phrase.

**Implementation**

VGG16 consists of 16 layers, which include 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. The first 13 layers are convolutional layers, which are used to extract features from the input image. These layers are followed by max-pooling layers, which reduce the dimensionality of the feature maps and help to prevent overfitting.

The output of the last max-pooling layer is then fed into three fully connected layers, which perform the final classification task. The first two fully connected layers have 4,096 nodes each, and the last fully connected layer has as many nodes as the number of classes in the classification task. The output of the last fully connected layer is then passed through a softmax activation function to produce the final probability distribution over the classes.

**Important libraries :**

We import the necessary libraries including VGG16 from Keras, Matplotlib for visualizations, and OpenCV for image processing. We also import the Input, Lambda, Dense, Flatten, and Dropout layers from Keras, which we will use to modify the VGG16 architecture to fit our problem.

Loading the dataset:

The images and labels are loaded from the provided directory using the load images function.The function takes the directory path as an argument and returns a tuple containing the loaded images and their labels.

After this we have imported Keras and check the unique values and got this in result

Total number of symbols: 37

Number of training images: 44604

Number of testing images: 11151

Number of evaluation images: 7400

**Confusion Matrix :**

We are making confusion matrix here for checking sensitivity. To predict the output in the form of true negative and false negative and vice versa.

Here we are using plot\_confusion matrix .

sensitivity is mainly degree of true negative

**Chart, scatter chart

Description automatically generated**

**Output:**

**Chart, histogram

Description automatically generated**

**Conclusion:**

Finally, our project demonstrates the capability of CNNs and VGG16 in identifying and understanding motions. By making sign language more accessible to a larger audience, this technology has the potential to transform communication for persons who rely on it. The dataset may be expanded to include additional motions in the future, the model's accuracy could be improved, and the technology could be integrated into a user-friendly interface. Some more advancement can be made into this by implementing real time motion ,audio, videos dataset like features and by implementing more advance deep learning techniques which would be really helpful for the society and the disabled persons and this can solve their 90% of problems . So we can say that this project can serve to the society in a very good way.