CHAPTER 1

Introduction

1.1 Background

Sign Language is the most natural and expressive way for the hearing impaired people. People, who are not deaf, never try to learn the sign language for interacting with the deaf people. This leads to isolation of the deaf people. But if the computer can be programmed in such a way that it can translate sign language to text format, the difference between the normal people and the deaf community can be minimized.

Indian sign language (ISL) uses both hands to represent each alphabet and gesture. ISL alphabets are derived from British Sign Language (BSL) and French Sign Language (FSL). Most of the researchers in this area concentrate on the recognition of American Sign Language (ASL) since most of the signs in ASL are single handed and thus, complexity is less. Another attractive feature is that ASL already has a standard database that is available for use. When compared with ASL, Indian Sign Language relies on both hands and thus, an ISL recognition system is more complex. A few research works carried out by the researchers in the recognition of ISL. Currently, more researchers have started doing research in ISL.

Here this proposed system is able to recognize the various alphabets of Indian Sign Language; this will reduce the noise and give accurate result. The important research problem in computer recognition is the sign language for enabling communication with hearing impaired people. This system introduces efficient and fast techniques for identification of the hand gesture representing an alphabet of the Sign Language.

Currently, more interest is created to do research in the field of sign language recognition system. Deaf and Dumb people rely on sign language interpreters for communications. A real time Sign Language Recognition system was designed and implemented to recognize 26 gestures from the Indian Sign Language by hand gesture recognition system for text generation. The signs are captured by using web cam. This signs are processed for feature extraction using some colour model. The extracted features are compared by using pattern matching algorithm. In order to calculate the sign recognition, the features are compared with testing database. Finally, recognized gesture is converted into text. This system provides an opportunity for a deaf-dumb people to communicate with non-signing people without the need of an interpreter.

In the existing systems, BSL uses a two-handed finger spelling system, compared to the one-handed system used in ASL (and FSL). Many American Deaf believe that one handed finger-spelling makes for faster finger-spelling than two-handed systems. However, anecdotal evidence has it that in a challenge between proficient ASL and BSL speakers, neither finger-spelling system proved to be faster; both finished reciting the alphabet at the same time. So that supposed “disadvantage” is rendered invalid. According to many Europeans, American signers tend to fingerspell “too much” compared to the rate of finger-spelling in many European sign languages, including BSL. This may be true; several examples of BSL signs for concepts that do not have a sign in ASL and are often finger-spelled for lack of a formal sign. This is one of the advantages of BSL, but that is not intrinsic to the language itself and it reveals a cultural value. On the other hand, that many BSL signs are often derived from their initialized (English) base, while many ASL signs have been developed without initialization (including the influence of signed English systems), so one might see that as a “disadvantage “.

Nowadays, people are not interested to speak in ASL when having a deaf relative or friend, or even classmate/acquaintance. Hence, deaf people are often trapped and isolated. ASL requires the use of a person’s hands so if something happens where a wrist was sprained and it disables that person from talking. For example, there was a mother who strained her wrist from signing all of her life for her deaf daughter. The doctor also made her stop signing. This caused the communication with her deaf daughter to decrease, since she had to read lips from then on.

ASL vocabulary dictionary contains thousands of sign just like words. It is very easy to get two completely different signs mixed up which leads to bad miscommunication. For example, the sign for “chocolate” and “cleve land” are similar, and they definitely don’t mean the same thing, or even close. It is very hard to follow when a conversation has and something gets mixed up.

1.2 Relevance

I have chosen this program for my PBL project as I want to pursue my honors in Artificial Intelligence and Machine Learning. So as just to get a glimpse of the technologies and softwares used in Machine Learning. By making this project I have got a rough idea about how Artificial Intelligence and Machine learning codes are used and how they can prove to be helpful.

1.3 Literature Survey

A various hand gestures were recognized with different methods by different researchers in which were implemented in different fields. The recognition of various hand gestures were done by vision based approaches, data glove based approaches, soft computing approaches like Artificial Neural Network, Fuzzy logic, Genetic Algorithm and others like PCA, Canonical Analysis, etc. The recognition techniques are divided into three broad categories such as Hand segmentation approaches, Feature extraction approaches and Gesture recognition approaches.

"Application research on face detection technology uses Open CV technology in mobile augmented reality" introduces the typical technology. Open source computer vision library, Open CV for short is a cross-platform library computer vision based on open source distribution. The Open CV, with C language provides a very rich visual processing algorithm to write it part and combined with the characteristics of its open source. Data gloves and Vision based method are commonly used to interpret gestures for human computer interaction. The sensors attached to a glove that finger flexion into electrical signals for determining the hand posture in the data gloves method. The camera is used to capture the image gestures in the vision based method. The vision based method reduces the difficulties as in the glove based method.

"Hand talk-a sign language recognition based on accelerometer and semi data" this paper introduces American Sign Language conventions. It is part of the "deaf culture" and includes its own system of puns, inside jokes, etc. It is very difficult to understand understanding someone speaking Japanese by English speaker. The sign language of Sweden is very difficult to understand by the speaker of ASL. ASL consists of approximately 6000 gestures of common words with spelling using finger used to communicate obscure words or proper nouns.

"Hand gesture recognition and voice conversion system for dumb people" proposed lower the communication gap between the mute community and additionally the standard world. The projected methodology interprets language into speech. The system overcomes the necessary time difficulties of dumb people and improves their manner. Compared with existing system the projected arrangement is simple as well as compact and is possible to carry to any places. This system converts the language in associate text into voice that's well explicable by blind and ancient people. The language interprets into some text kind displayed on the digital display screen, to facilitate the deaf people likewise. In world applications, this system is helpful for deaf and dumb of us those cannot communicate with ancient person.

Conversion of RGB to gray scale and gray scale to binary conversion introduced in the intelligent sign language recognition using image processing. Basically any colour image is a combination of red, green, blue colour. A computer vision system is implemented to select whether to differentiate objects using colour or black and white and, if colour, to decide what colour space to use (red, green, blue or hue, saturation, luminosity).

METHODOLOGY

Sign Language Recognition System

Sign language recognition is an important application of gesture recognition. Sign language recognition has two different approaches.

- Glove based approaches
- Vision based approaches.

Glove based approaches :

In this category requires signers to wear a sensor glove or a colored glove. The task will be simplified during segmentation process by wearing glove. The drawback of this approach is that the signer has to wear the sensor hardware along with the glove during the operation of the system.

Vision based approaches:

Image processing algorithms are used in Vision based technique to detect and track hand signs and facial expressions of the signer. This technique is easier to the signer since there is no need to wear any extra hardware. However, there are accuracy problems related to image processing algorithms and these problems are yet to be modified.

There are again two different approaches in vision based sign language recognition:

-3D model based

-Appearance based

3D model based methods make use of 3D information of key elements of the body parts. Using this information, several important parameters, like palm position, joint angles etc., can be obtained. This approach uses volumetric or skeletal models, or a combination of the two. Volumetric method is better suited for computer animation industry and computer vision. This approach is very computational intensive and also, systems for live analysis are still to be developed.

Appearance-based systems use images as inputs. They directly interpret from these videos/images. They don't use a spatial representation of the body. The parameters are derived directly from the images or videos using a template database. Some templates are the deformable 2D templates of the human parts of the body, particularly hands. The sets of points on the outline of an object called as deformable templates. It is used as interpolation nodes for the objects outline approximation.

1.4 Motivation

This part of the project report must contain a brief mention of the gaps, limitations, further improvisation of the topic chosen in the existing work based on the previous sub-section and a justification for undertaking your project/problem and why it needs a solution and further work.

1.5 Aim of the Project

This software program is helpful for deaf and dumb of us those cannot communicate without the help of translator.

1. Will be able to detect single hand handsigns
2. Convert the handsign to text
3. Can detect some of the common phrases

1.6 Scope and Objectives

To develop manpower for using Indian Sign Language (ISL) and teaching and conducting research in ISL, including bilingualism.

To promote the use of Indian Sign Language as educational mode for deaf students at primary, secondary and higher education levels.

To carry out research through collaboration with universities and other educational institution in India and abroad and create linguistic records/ analyses of the Indian Sign Language, including creation of Indian Sign Language corpus (Vocabulary).

To orient and train various groups, i.e. Govt. officials, teachers, professionals, community leaders and the public at large for understanding and using Indian Sign Language.

To collaborate with organizations of the deaf and other institutions in the field of disability to promote and propagate Indian Sign Language.

To collect information relating to Sign Language used in other parts of the world so that this input can be used to upgrade the Indian Sign Language.

CHAPTER 2

Description of Project

2.1 Technical Approach

Data Processing:

The load data.py script contains functions to load the Raw Image Data and save the image data as numpy arrays into file storage. The process data.py script will load the image data from data.npy and preprocess the image by resizing/rescaling the image, and applying filters and ZCA whitening to enhance features. During training the processed image data was split into training, validation, and testing data and written to storage. Training also involves a load dataset.py script that loads the relevant data split into a Dataset class. For use of the trained model in classifying gestures, an individual image is loaded and processed from the filesystem.

The training loop for the model is contained in train model.py. The model is trained with hyperparameters obtained from a config file that lists the learning rate, batch size, image filtering, and number of epochs. The configuration used to train the model is saved along with the model architecture for future evaluation and tweaking for improved results. Within the training loop, the training and validation datasets are loaded as Dataloaders and the model is trained using Adam Optimizer with Cross Entropy Loss. The model is evaluated every epoch on the validation set and the model with best validation accuracy is saved to storage for further evaluation and use. Upon finishing training, the training and validation error and loss is saved to the disk, along with a plot of error and loss over training.

• Classify Gesture:

After a model has been trained, it can be used to classify a new ASL gesture that is available as a file on the filesystem. The user inputs the filepath of the gesture image and the test data.py script will pass the filepath to process data.py to load and preprocess the file the same way as the model has been trained.

The data preprocessing was done using the PILLOW library, an image processing library, and sklearn.decomposition library, which is useful for its matrix optimization and decomposition functionality.

Image Enhancement:

A combination of brightness, contrast, sharpness, and color enhancement was used on the images. For example, the contrast and brightness were changed such that fingers could be distinguished when the image was very dark.

Edge Enhancement:

Edge enhancement is an image filtering techniques that makes edges more defined. This is achieved by the increase of contrast in a local region of the image that is detected as an edge. This has the effect of making the border of the hand and fingers, versus the background, much more clear and distinct. This can potentially help the neural network identify the hand and its boundaries.

Image Whitening:

ZCA, or image whitening, is a technique that uses the singular value decomposition of a matrix. This algorithm decorrelates the data, and removes the redundant, or obvious, information out of the data. This allows for the neural network to look for more complex and sophisticated relationships, and to uncover the underlying structure of the patterns it is being trained on. The covariance matrix of the image is set to identity, and the mean to zero

The model used in this classification task is a fairly basic implementation of a Convolutional Neural Network (CNN). As the project requires classification of images, a CNN is the go-to architecture. The basis for our model design came from Using Deep Convolutional Networks for Gesture Recognition in AmericanSign Language paper that accomplished a similar ASL Gesture Classification task [4]. This model consisted of convolutional blocks con taining two 2D Convolutional Layers with ReLU activation, followed by Max Pooling and Dropout layers. These convolutional blocks are repeated 3 times and followed by Fully Connected layers that eventually classify into the required categories. The kernel sizes are maintained at 3 X 3 throughout the model. Our originally proposed model is identical to the one from the aforementioned paper, this model is shown in Figure 5. We omitted the dropout layers on the fully connected layers at first to allow for faster training and to establish a baseline without dropout.

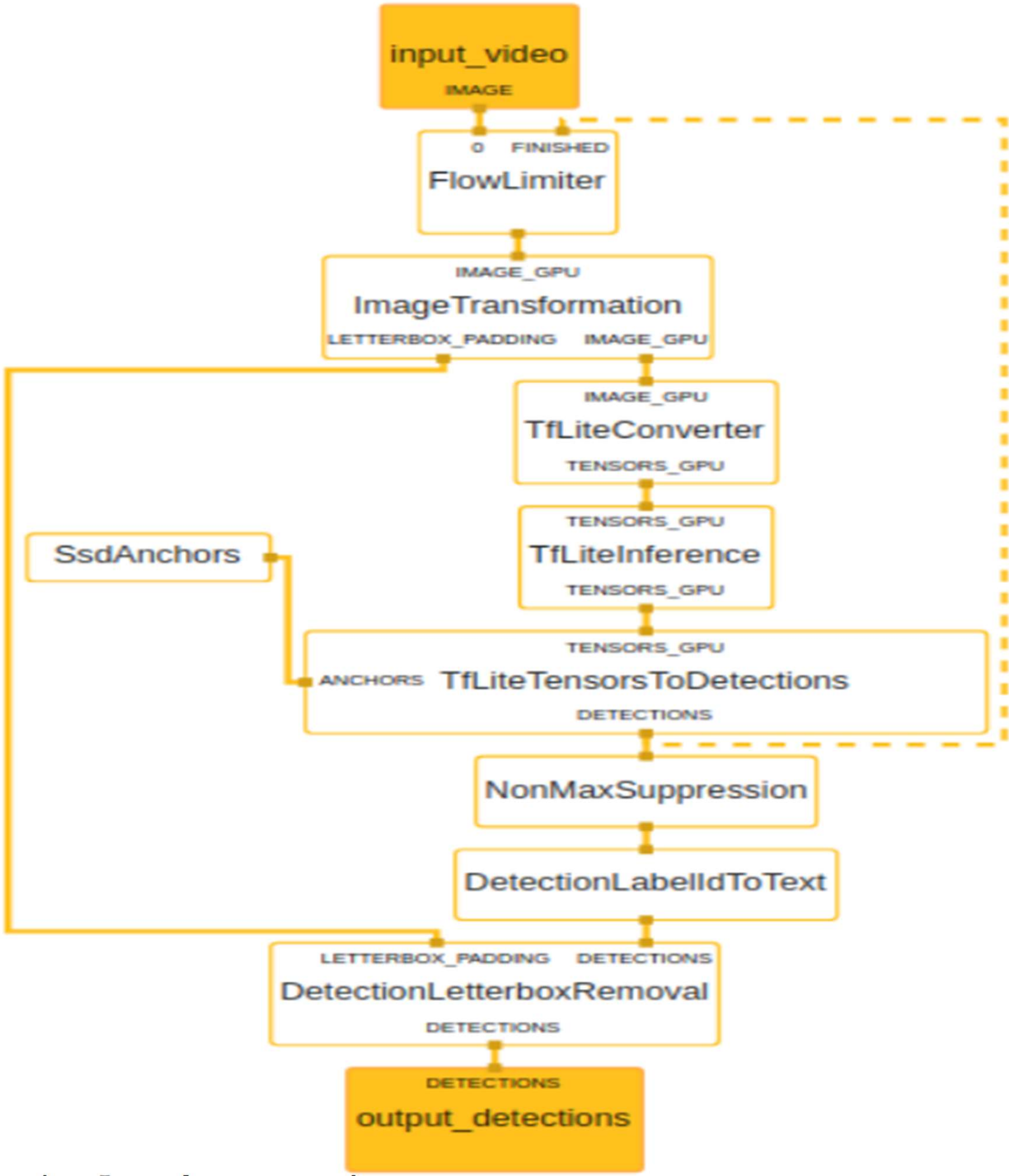
We also decided to design a separate model to compare with the model in the paper. This model was designed to be trained faster and to establish a baseline for problem complexity. This smaller model was built with only one “block” of convolutional layers consisting of two convolutional layers with variable kernel sizes progressing from 5 X 5 to 10 X 10, ReLU activation, and the usual Max Pooling and Dropout. This fed into three fully connected layers which output into the 29 classes of letters. The variation of the kernel sizes was motivated by our dataset including the background, whereas the paper preprocessed their data to remove the background. The design followed the thinking that the first layer with smaller kernel would capture smaller features such as hand outline, finger edges and shadows. The larger kernel hopefully captures combinations of the smaller features like finger crossing, angles, hand location, etc. This model architecture is shown in Figure 6

Our models were trained using Adam optimizer and Cross Entropy Loss. Adam optimizer is known for converging quickly in comparison with Stochastic Gradient Descent (SGD), even while using momentum. However, initially Adam would not decrease our loss thus we abandoned it to use SGD. Debugging Adam optimizer after our final presentation taught us that lowering learning rate significantly can help Adam to converge during training. Thus allowing us to train more models towards the end of our project. Of our two models, the one based on the paper was shown not to be viable, as it took much longer to train without showing any significant decrease in accuracy or loss. We believe this is likely due to the more difficult nature of our classification with the inclusion of background in the images and the lower resolution, causing training to be more dif ficult.

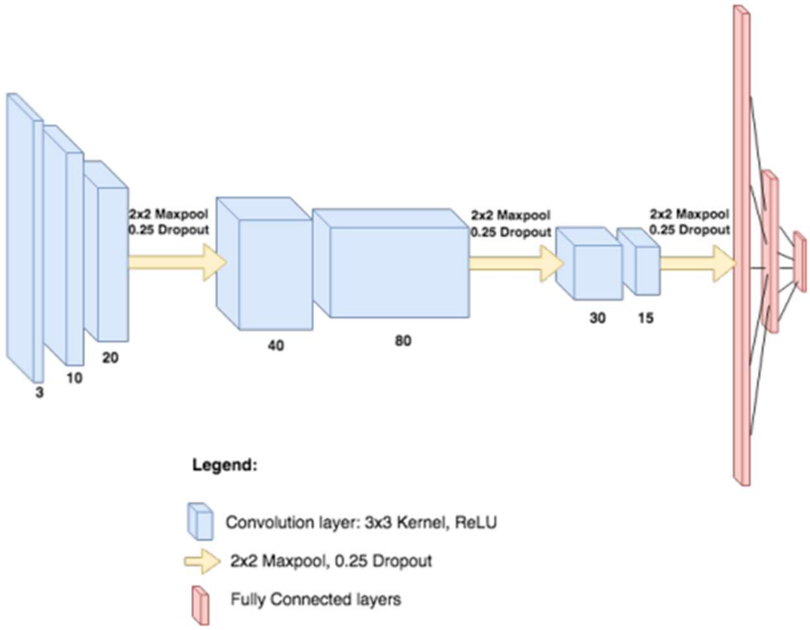
Thus, we decided to focus on improving our smaller model which initially trained to 40% validation accuracy. Although we had a very large dataset to work with; 3,000 samples for each of 29 classes, after processing the images into numpy arrays, we found our personal computers could load a maximum of 50-100 samples/class and our Google Cloud server could load 200 samples/class. The need to load small datasets actually led us to test the effect of increasing the data available to our models. Training our initial models on less data led to the models quickly overfitting as shown in Figure 7. This is likely due to the small amount of samples to train on leading to bad generalization and learning of the sample space. Increasing the size of our dataset to 200 samples/class led to better model results, with peak validation accuracy of 60.3% in epoch 17.

However, taking a look at our loss function, we see that the validation loss is increasing, indicating overfitting of the model. After we applied filtering, enhancement, and ZCA whitening to our dataset, the model performance increased drastically as shown in Figure 9. The peak validation accuracy achieved is 77.25% in epoch 24. As shown by the plot of loss, the validation loss is still decreasing, albeit at a slower rate than the training loss, indicating that the model is not drastically overfitting. This shows that preprocessing our images by applying filters and ZCA whitening helps to enhance relevant features for the model to learn.

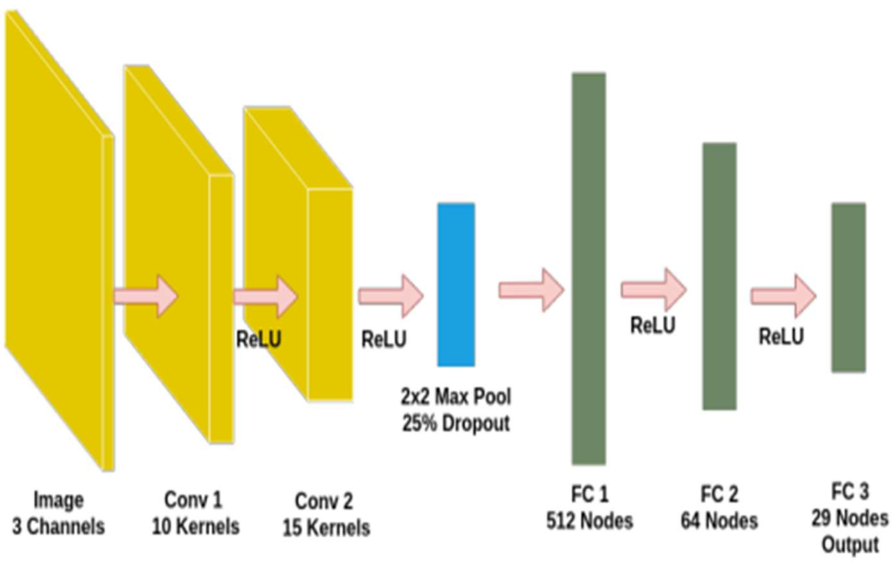
2.2 Block diagram



2.3 Flow chart



The model used in this classification task is a fairly basic implementation of a Convolutional Neural Network (CNN). As the project requires classification of images, a CNN is the go-to architecture. The basis for our model design came from Using Deep Convolutional Networks for Gesture Recognition in AmericanSign Language paper that accomplished a similar ASL Gesture Classification task [4]. This model consisted of convolutional blocks con taining two 2D Convolutional Layers with ReLU activation, followed by Max Pooling and Dropout layers. These convolutional blocks are repeated 3 times and followed by Fully Connected layers that eventually classify into the required categories. The kernel sizes are maintained at 3 X 3 throughout the model.



2.4 Software resources

However CV2 library and mediapipe library was used on a wide scale for this application.

The following are the libraries, extensions, modules and submodules are used along with their specified version.

```
absl-py==0.9.0
altgraph==0.17
aplus==0.11.0
appdirs==1.4.4
astor==0.8.1
astropy==4.1
astunparse==1.6.3
attrs==20.3.0
backcall==0.1.0
bleach==3.1.0
bqplot==0.12.26
branca==0.4.2
cachetools==4.1.0
catboost==0.26
certifi==2020.4.5.1
chardet==3.0.4
click==7.1.2
cloudpickle==1.6.0
colorama==0.4.1
cycler==0.10.0
Cython==0.29.23
dask==2021.3.0
dataclasses==0.8
deap==1.3.1
decorator==4.3.2
defusedxml==0.5.0
distlib==0.3.0
dlib @
https://pypi.python.org/packages/da/06/bd3e241c4eb0a662914b3b4875fc52dd
176a9db0d4a2c915ac2ad8800e9e/dlib-19.7.0-cp36-cp36m-
win_amd64.whl#md5=b7330a5b2d46420343fbed5df69e6a3f
entrypoints==0.3
filelock==3.0.12
frozendict==2.0.2
future==0.18.2
gast==0.2.2
google-auth==1.15.0
google-auth-oauthlib==0.4.1
```

```
google-pasta==0.2.0
graphviz==0.17
grpcio==1.29.0
h5py==2.10.0
hyperband-snakemake==0.0.2
hyperopt==0.2.5
idna==2.9
imageai==2.1.5
imbalanced-learn==0.8.0
imblearn==0.0
importlib-metadata==1.6.0
importlib-resources==1.5.0
imutils==0.5.3
ipydatawidgets==4.2.0
ipykernel==5.1.0
ipyleaflet==0.13.6
ipympl==0.7.0
ipython==7.2.0
ipython-genutils==0.2.0
ipyvolume==0.5.2
ipyvue==1.5.0
ipyvuetify==1.6.2
ipywebrtc==0.6.0
ipywidgets==7.6.3
jedi==0.13.2
Jinja2==3.0.1
joblib==0.15.1
jsonschema==2.6.0
jupyter==1.0.0
jupyter-client==5.2.4
jupyter-console==6.0.0
jupyter-core==4.4.0
jupyter-http-over-ws==0.0.8
jupyterlab-widgets==1.0.0
Keras==2.3.1
Keras-Applications==1.0.8
Keras-Preprocessing==1.1.2
kiwisolver==1.2.0
lightgbm==3.2.1
llvmlite==0.36.0
lxml==4.6.3
Markdown==3.2.2
MarkupSafe==2.0.1
matplotlib==3.3.4
mediapipe==0.8.3
missingno==0.4.2
mistune==0.8.4
nbconvert==5.4.0
```

```
nbformat==4.4.0
nest-asyncio==1.5.1
networkx==2.5.1
nltk==3.5
nose==1.3.7
notebook==5.7.4
numba==0.53.1
numpy==1.19.3
oauthlib==3.1.0
opencv-contrib-python==4.2.0.34
opencv-python==4.5.1.48
opt-einsum==3.2.1
Optunity==1.1.1
pandas==1.1.5
pandas-datareader==0.10.0
pandocfilters==1.4.2
parso==0.3.2
patsy==0.5.1
pefile==2019.4.18
pickleshare==0.7.5
Pillow==7.2.0
plotly==5.0.0
progressbar2==3.53.1
prometheus-client==0.5.0
prompt-toolkit==2.0.8
protobuf==3.12.1
psutil==5.8.0
pyaml==20.4.0
pyarrow==4.0.0
pyasn1==0.4.8
pyasn1-modules==0.2.8
pycocotools==2.0.2
Pygments==2.3.1
pyinstaller==4.2
pyinstaller-hooks-contrib==2020.11
pyparsing==2.4.7
python-dateutil==2.7.5
python-utils==2.5.6
pythreejs==2.3.0
pytz==2021.1
pywin32-ctypes==0.2.0
pywinpty==0.5.5
PyYAML==5.3.1
pyzmq==17.1.2
qtconsole==4.4.3
regex==2020.6.8
requests==2.23.0
requests-oauthlib==1.3.0
```

```
rsa==4.0
scikit-hyperband @ git+https://github.com/thuijskens/scikit-
hyperband@68bef5a20570e802b2d9e9be405c3f2a043a32cd
scikit-learn==0.23.2
scikit-optimize==0.8.1
scipy==1.4.1
seaborn==0.11.1
Send2Trash==1.5.0
Shapely==1.7.1
six==1.12.0
sklearn==0.0
statsmodels==0.12.2
stopit==1.1.2
tabulate==0.8.9
tenacity==7.0.0
tensorboard==2.5.0
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.6.0.post3
tensorflow==1.15.3
tensorflow-estimator==1.15.1
termcolor==1.1.0
terminado==0.8.1
testpath==0.4.2
thop==0.0.31.post2005241907
threadpoolctl==2.0.0
toolz==0.11.1
torch==1.8.1
torchvision==0.9.1
tornado==5.1.1
TPOT==0.11.7
tqdm==4.46.1
traitlets==4.3.2
traittypes==0.2.1
typing-extensions==3.10.0.0
update-checker==0.18.0
urllib3==1.25.9
vaex==4.1.0
vaex-astro==0.8.0
vaex-core==4.1.0
vaex-hdf5==0.7.0
vaex-jupyter==0.6.0
vaex-ml==0.11.1
vaex-server==0.4.0
vaex-viz==0.5.0
virtualenv==20.0.21
wcwidth==0.1.7
webencodings==0.5.1
Werkzeug==1.0.1
```

```
widgetsnbextension==3.5.1
wordcloud==1.8.1
wrapt==1.12.1
xarray==0.16.2
xgboost==1.4.2
xlrd==1.2.0
yolov5==5.0.5
zipp==3.1.0
```

CHAPTER 3

System Design

3.1 Circuit diagram

No circuit diagram

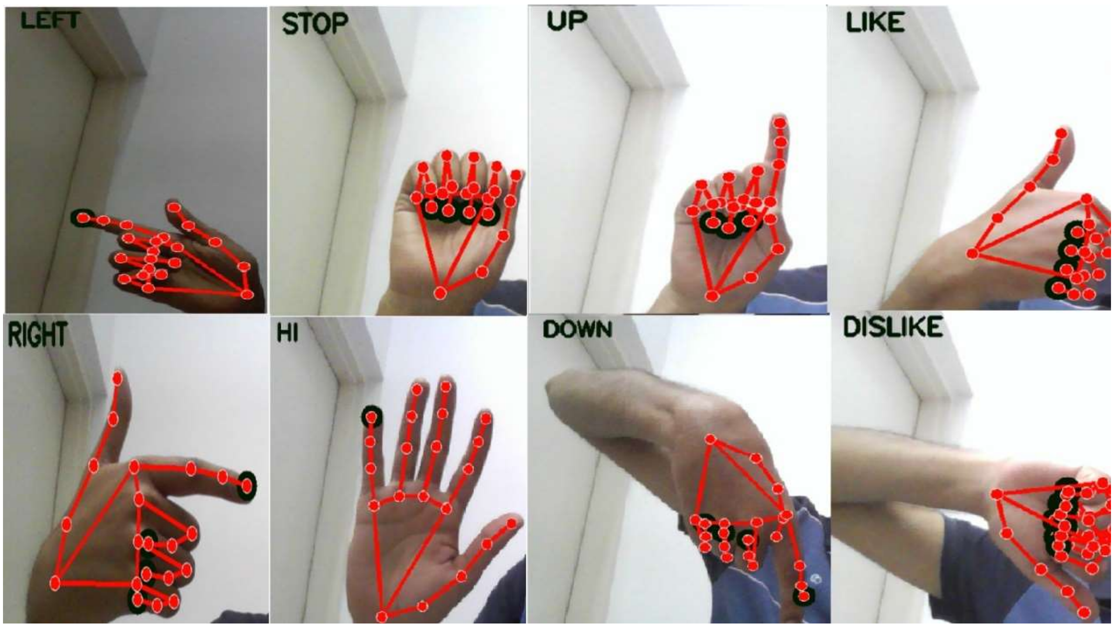
3.2 Design calculation

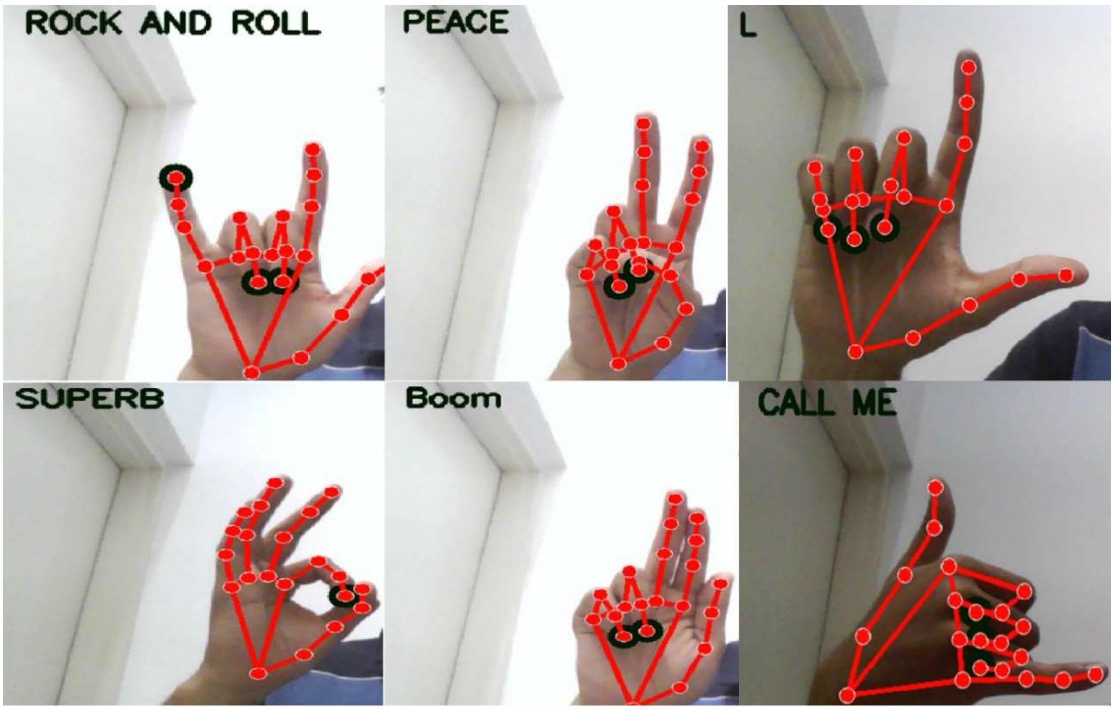
There is no design for this program, although GPU can be added.

CHAPTER 4

Implementation and Testing

4.1 Implementation





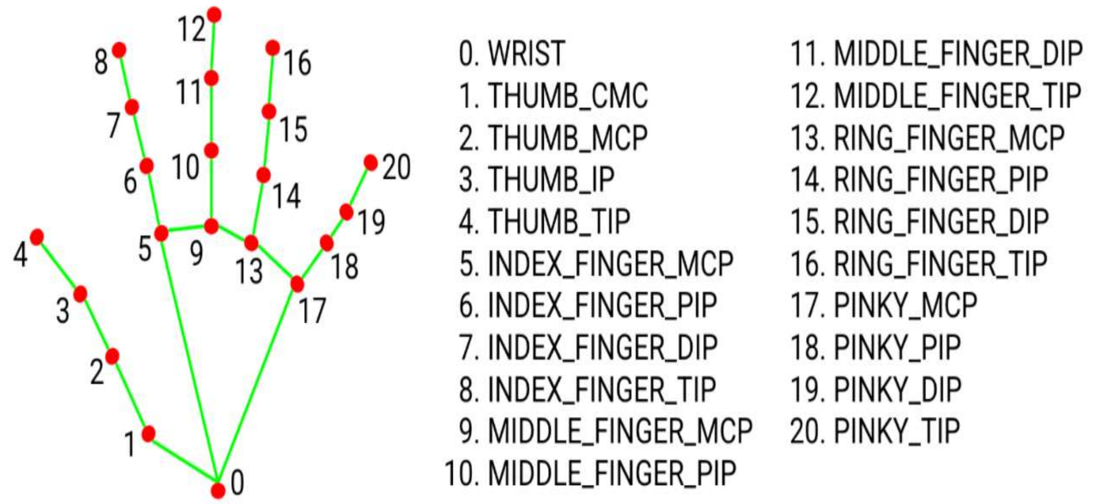


Fig 2. 21 hand landmarks.

4.2 Testing and debugging

Since I have used landmark points which gets plotted on the persons hand, and the distance between these landmark points differ from hand to hand, age to age, people with more than 5 fingers or less any many such other conditions, I have written the code considering the hand image of a normal adult person.

CHAPTER 5

Results and conclusion

5.1 Results

Easier to code and implement instead of training and testing databases or using CNN or Artificial neural networks.

Can be used dynamically as well (but not effective as compared to other techniques).

Accuracy can be increased by assigning more landmark points.

Complex hand signs are difficult to code.

Some of the hand signs differ slightly from one another, since they use positioning or hand segmentation it is really not possible to write such a code which can differentiate between the two.

The hand sign for 'I' and 'J' is similar and for 'A' and 'S' is similar (the only difference between them is how they are shown that is inclination).

Code for hand signs which use 2 hands can also be done using this technique.

5.2 Conclusion

We have reviewed many prior works such as HMM-based, modified HMM based, neural network based, and hybrid-based approaches.

HMM-based SLR approaches have been shown to achieving good recognition accuracy especially in small to medium sized datasets. Efforts on improving the performance of HMM-based approaches have also been proposed by modifying the standard HMM method.

However, the proposed HMM-based and modified HMM-based approaches still require the extraction of sign features from input data before applying the classification method.

Designing invariant sign features can be tedious works highly dependent on the type of input data being used. Moreover, feature extraction also contributes to the computational load and the classification performance often depends on the quality of the extracted sign features.

Combinations of CNN and HMM have also been proposed to improve the performance of the SLR system. It has been shown that a hybrid approach that embeds CNN into HMM abiding to Bayesian principles outperforms the tandem approach that treats CNN as a feature extractor.

The hybrid approach also enables the end-to-end training. The 3D CNN and the combination of 3D CNN and RNN have recently been shown to potentially improve the SLR performance.

5.2 Future Scope

We could also put an input field for letters and get the sign as an output, if possible, to make it into a fully sign communication platform.

The model can be run further for more epochs. The data could be increased. We can add words, numbers as well for this. We could also do this by inputting a constant stream of images and that get a resultant string for a particular word.

We could get an android application for the same. Text to speech could be added as well. It could be made multilingual as well.

We can use this technique to translate a whole phrase. Here instead of using images we can provide it a video and the program will interpret all the hand signs shown in it as the output.

The program and its features can become more attractive by adding GPU.

We can input the word and its hand sign will appear as the output.

Bill of material

None as this is a software program

References

[1] Farnaz D. Notash and Elahe Elhamki. “Comparing loneliness, depression and stress in students with hearing impaired and normal students studying in secondary schools of Tabriz”. In: International Journal of Humanities
and Cultural Studies February 2016 Special Issue (2016). issn: 2356-5926.

[2] “The Cognitive, Psychological and Cultural Impact of Communication Barrier on Deaf Adults”. In: Journal
of Communication Disorders, Deaf Studies Hearing Aids 4 (2 2016). doi: 10.4172/2375-4427.1000164.

[3] Akash. ASL Alphabet. url: https://www.kaggle.com/grassknoted/asl-alphabet. (accessed: 24.10.2018).

[4] Vivek Bheda and Dianna Radpour. “Using Deep Convolutional Networks for Gesture Recognition in American
Sign Language”. In: CoRR abs/1710.06836 (2017). arXiv: 1710.06836. url: http://arxiv.org/abs/1710.
06836.

[5] (PDF) SIGN LANGUAGE DETECTION | IRJET Journal - Academia.edu

[6] (PDF) A Survey of Hand Gesture Recognition Methods in Sign Language Recognition (researchgate.net)

[7] (192) Sign Language Recognition using Machine Learning | Sign Language Detection | Data Magic – YouTube

[8] (191) Sign Language Detection-part2 | AI Hand Sign detection | Machine Learning | Data Magic – YouTube

datamagic2020/sign-language-detection (github.com)