iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: II Month of publication: February 2023

DOI: https://doi.org/10.22214/ijraset.2023.49199

www.ijraset.com

Call: ✆08813907089 | E-mail ID: ijraset@gmail.com

Sign Language Recognition Using Machine Learning

Anubrolu Umesh Chowdary[1], Challa Mithra Sree[2], B.Siddhartha[3]
[1, 2, 3]Computer Science and EngineeringVardhaman College of Engineering Hyderabad, India

Abstract: This study proposes a method for recognising motions through image processing. In this project, we create a sign detector that can be readily enhanced to recognise a wide range of various signs and hand gestures, such as the alphabets, and that can recognise numbers from 1 to 10.. We used the Python Keras and OpenCV libraries to create this project. Applications for this project are incredibly diverse. Machine learning is usedto build models, extract key features, and construct applications since it has access to enormous databases. Machine learning can be used in our daily lives to make living easier. Deep learningis used to detect signs in sign language, which is a complicated process.

Index Terms: CNN, openCV, Keras, Machine learning.

I. INTRODUCTION

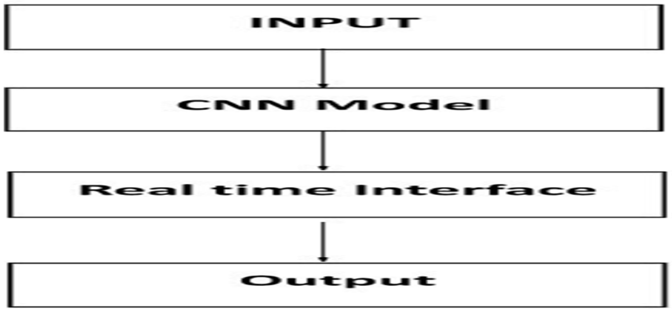
In many ways, sign language is highly beneficial. Here, we are accepting input in the form of signs and sending them through an interface that, depending on the sign's recognition threshold value, launches an application. To create an interfacethat truly aids in taking real-time input and launching the re- quired application, we use OpenCV. Using the newly produced dataset, we will first train the data.

II. MODEL

A. Convolutional Neural Networks

We will first generate our own dataset of images of hand movements including the digits 1 to 10 using numpy and a number of tools and methods. The dataset can be produced using a wide variety of photos. The data is then trained using the Keras library. To extract the features, or indicators, from the image, we employ convolutional neural networks (CNN).

Identify applicable funding agency here. If none, delete this.



III. OUR MODEL DEVELOPMENT

Regarding Deep Learning The most well-known neural net- work method and a common approach for image/video tasks is convolution neural networks (CNN). LeNET- 5 and MobileNetV2, two cutting-edge designs for Convolution Neural Networks (CNN), are among the architectures we can apply to achieve the State of the Art (SOTA). These are all employable architectures that can be combined using neural network ensemble techniques.By doing this, we can create a hand gesture recognition model that is almost entirely correct.

This model will be implemented in standalone applications or embedded devices that use standalone web frameworks likeDjango to recognise hand motions in real-time video, calculate the appropriate threshold value, and launch the appropriateapplication.

A. Abbreviations and Acronyms

CNN-”Convolution Neural Networks” RTI - ”Real timeInterface” SOTA - ”State Of the ART”

IV. LITERATURE SURVEY

Around the world, there are numerous research projectsaimed at creating recognition systems for sign languages. Our main goal is to create a hand gesture recognition-based programme for PCs, mobile devices, tabulators, and other devices. There are numerous systems that employ various methodologies to apply this recognition-type concept. They deal with the alphabet, numeric numerals, and unusual signs.

A. Real-Time Sign Language Fingerspelling Recognition Using Convolutional Neural Networks From Depth Map [5].

The focus of this project is American Sign Language static fingerspelling. a technique for putting into practise a signlanguage to text/speech conversion system without the useof handheld gloves and sensors, which involves continually recording the gestures and turning them into voice. Only a small number of photos were used in this method to identify objects. the layout of a device to let physically unable people communicate.

B. Real-Time Sign Language Fingerspelling Recognition Using Convolutional Neural Networks From Depth Map [5].

C. Design Of A Communication Aid For Physically Challenged [6]

The MATLAB environment was used for the system's development. The training phase and the testing phase make up the majority of it. The author used feed-forward neural networks during the training stage. The issue here is that MATLAB is not very effective, and it is also challenging to integrate the concurrent qualities as a whole.

D. American Sign Language Interpreter System for Deaf and Dumb Individuals [7].

Twenty of the static ASL alphabets could be recognised using the approaches we explained. The occlusion issue pre- vented the recognition of the letters A, M, N, and S. There aren't many images that they've used.

E. A Framework for Hand Gesture Recognition and SpottingUsing Sub-gesture Modeling [10].

With the only difference that in this scenario, we add a gesture model (each gesture will have a gesture completion model) between the start state and exit state of the filler model,gesture-completion models are built in a manner identical to filler models.

F. Research On Optimization Of Static Gesture RecognitionBased On Convolution Neural Network [11].

The image-based information vector sequence that serves as the foundation for gesture modelling includes features like the hand contour, the vector feature, the area histogram, and the motion vector track feature, to name just a few.

The image-based information vector sequence that serves as the foundation for gesture modelling includes features like the hand contour, the vector feature, the area histogram, and the motion vector track feature, to name just a few. The two fundamental subcategories of static gesture recognition are static gesture recognition and dynamic gesture recognition. Due to the deep neural network's potent ability to assess the effective qualities, several researchers began researching the static gesture recognition method based on depth learning.

V. PROPOSED METHODOLOGY

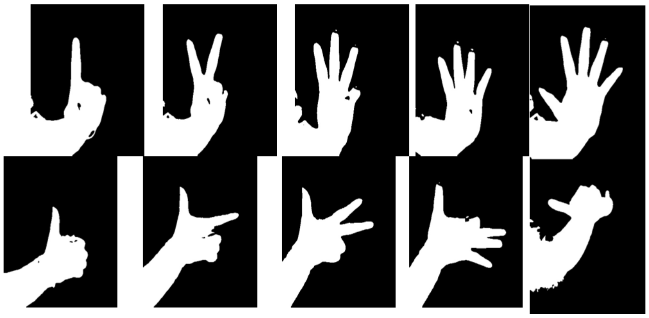
This model will be used to recognise hand gestures in real-time video, determine the proper threshold value, andlaunch the proper application. It will be implemented in standalone apps or embedded devices that employ standalone web frameworks like Django.

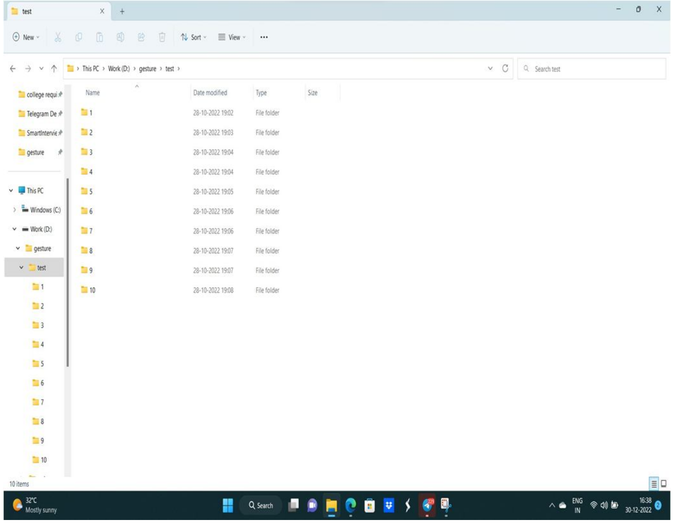
A. Mission

The goal is to create a system that can accept a sign and process it to a real time interface that opens the application respectively.

B. Corpus

The data set that we produced served as the corpus that we employed. We have produced a data set with various indicatorsin it. The data set we require may be found online, howeverfor this project, we will be building the data set ourselves. Every frame that recognises a hand in the predetermined region of interest will be saved in the "gesture" directory, which also contains the folders "train" and "test," each of which contains 10 folders of images that were captured using the application "generate gesture using data.py.".





Name	Date modified	Type	Size
1	28-10-2022 19:02	File folder	
2	28-10-2022 19:03	File folder	
3	28-10-2022 19:04	File folder	
4	28-10-2022 19:04	File folder	
5	28-10-2022 19:05	File folder	
6	28-10-2022 19:06	File folder	
7	28-10-2022 19:06	File folder	
8	28-10-2022 19:07	File folder	
9	28-10-2022 19:07	File folder	
10	28-10-2022 19:08	File folder	

C. Data Pre-processing

An image, in its most basic form, is a 2-dimensional array of pixels with values ranging from 0 to 255. Typically, 0 indicates black and 255 indicates white.. The mathematical function f(x, y) defines an image, where x in a coordinate plane denotes the horizontal and y the vertical. An image's pixel value at every position is given by the value of f(x, y) at that location. Algorithms are used in image pre-processing to manipulate pictures. Prior to delivering the photos for model training, itis crucial to preprocess the photographs. For instance, all of thephotos should be 200x200 pixels in size. If not, it is impossibleto train the model.

D. Model

The model we developed trains the data sequentially. The train and test data are first loaded using the keras programminglanguage. Now we use a variety of hyperparameters to create the CNN. The model is now fitted, and it is saved for future usage. This model will be implemented in standalone applications or embedded devices that use standalone web frameworks like Django to identify hand motions in real-time video, calculate the appropriate threshold value, and launchthe appropriate application.

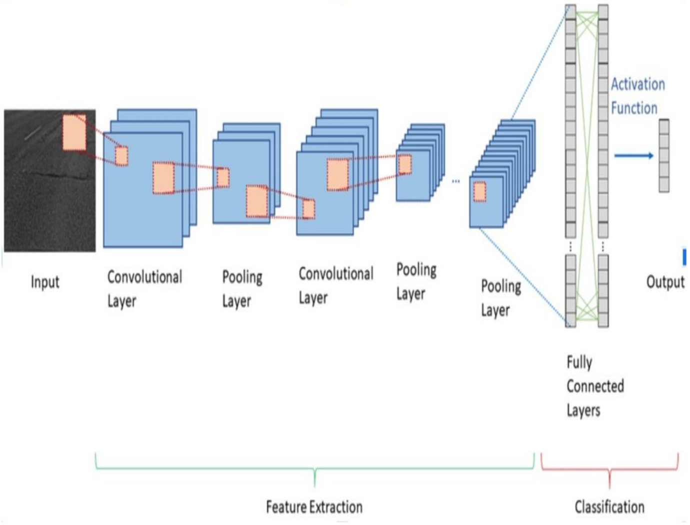
In our Sequential Model we have used 4 convolutionalneural layers and pooling layers with different filters.In order to edit images, algorithms are utilised in image pre-processing.Preprocessing the images is essential before providing them for model training. For instance, each photo should have a dimension of 200x200 pixels. If not, training the model is not possible.

A 2-dimensional array of pixels with values ranging from 0 to 255 constitutes a picture in its most basic form. Normally, 0 represents black and 255 represents white. The mathematical function f(x, y), which represents the horizontal in a coordinate plane as x and the vertical as y, defines a picture.The value of f(x, y) at a particular place determines the value of each pixel in a picture.

VI. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are deep neural networks that have been specifically created to analyse data with input forms approximating a 2D matrix.. Images might berepresented by a straightforward 2D matrix. CNN is crucial while utilising images. It takes an image as input, gives various elements and objects in the image weights and biases, and then separates them out depending on relevance.

The CNN uses filters (also known as kernels) to help with feature learning and identify abstract ideas such as blurring, edge detection, sharpening, etc., similar to how the human brain recognises objects in time and space. Because weights may bereused and there are fewer factors involved, the architecture fits the picture dataset more accurately (2048 to 256).



In this study, a convolutional neural network (CNN) is employed to convert an RGB picture into a visual feature vector. The three most often used CNN layers are convolution, pooling, and fully connected. ReLU f(x) = max(0, x), a nonlinear active function, is also used. ReLUis faster than the common equation f(x) = tanh (x). The useof a dropout layer prevents overfitting. Each hidden neuron's output is set to zero with probability 0.5 by the dropout.The ”dropped out” neurons are neither a part of the backpropagation or the forward pass. [5]

Due to the millions of parameters that both the CNN andthe RNN include, there are special convergence concerns when they are merged. For instance, Vinyals et al. found that fixing the convolutional layer’s parameters to those learnt fromImageNet is optimal. The only CNN parameters that are really learnt from caption instances are the RNN parameters and the non-convolution layer parameters.

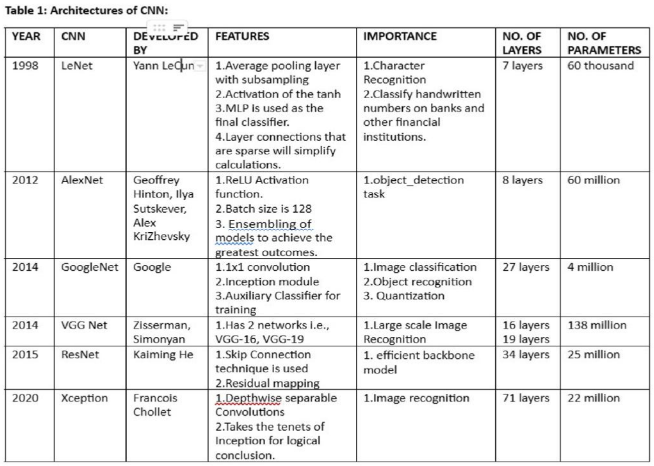
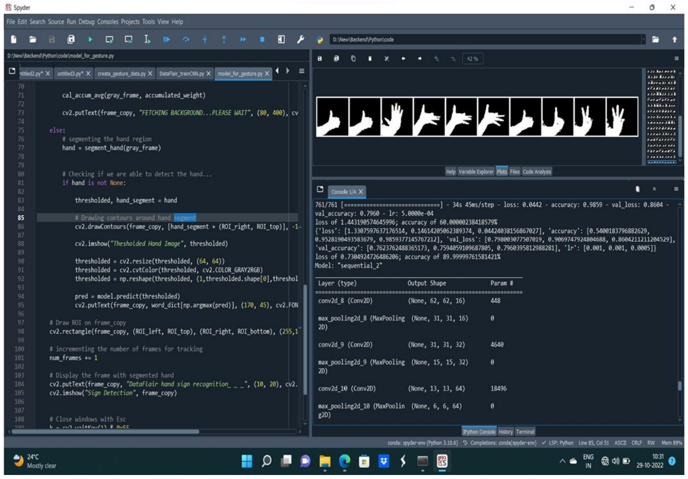


Table 1: Architectures of CNN:

YEAR	CNN	DEVELOPED BY	FEATURES	IMPORTANCE	NO. OF LAYERS	NO. OF PARAMETERS
1998	LeNet	Yann LeCun	1.Average pooling layer with subsampling 2.Activation of the tanh 3.MLP is used as the final classifier. 4.Layer connections that are sparse will simplify calculations.	1.Character Recognition 2.Classify handwritten numbers on banks and other financial institutions.	7 layers	60 thousand
2012	AlexNet	Geoffrey Hinton, Ilya Sutskever, Alex KriZhevsky	1.ReLU Activation function. 2.Batch size is 128 3. Ensembling of models to achieve the greatest outcomes.	1.object_detection task	8 layers	60 million
2014	GoogleNet	Google	1.1x1 convolution 2.Inception module 3.Auxiliary Classifier for training	1.Image classification 2.Object recognition 3. Quantization	27 layers	4 million
2014	VGG Net	Zisserman, Simonyan	1.Has 2 networks i.e., VGG-16, VGG-19	1.Large scale Image Recognition	16 layers 19 layers	138 million
2015	ResNet	Kaiming He	1.Skip Connection technique is used 2.Residual mapping	1. efficient backbone model	34 layers	25 million
2020	Xception	Francois Chollet	1.Depthwise separable Convolutions 2.Takes the tenets of Inception for logical conclusion.	1.Image recognition	71 layers	22 million


Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\New\Backend\Python\code\model_for_gesture.py
ntitled2.py*
untitled3.py*
create_gesture_data.py
DataFlair_trainCNN.py
model_for_gesture.py
Help Variable Explorer Plots Files Code Analysis
Console 1/A
IPython Console History Terminal
conda: spyder-env (Python 3.10.6) Completions: conda(spyder-env) LSP: Python Line 85, Col 51 ASCII CRLF RW Mem 89%
24°C
Mostly clear
ENG
IN
10:31
29-10-2022

```
            cal_accum_avg(gray_frame, accumulated_weight)

            cv2.putText(frame_copy, "FETCHING BACKGROUND...PLEASE WAIT", (80, 400), cv

        else:
            # segmenting the hand region
            hand = segment_hand(gray_frame)


            # Checking if we are able to detect the hand...
            if hand is not None:

                thresholded, hand_segment = hand

                # Drawing contours around hand segment
                cv2.drawContours(frame_copy, [hand_segment + (ROI_right, ROI_top)], -1

                cv2.imshow("Thesholded Hand Image", thresholded)

                thresholded = cv2.resize(thresholded, (64, 64))
                thresholded = cv2.cvtColor(thresholded, cv2.COLOR_GRAY2RGB)
                thresholded = np.reshape(thresholded, (1,thresholded.shape[0],threshol

                pred = model.predict(thresholded)
                cv2.putText(frame_copy, word_dict[np.argmax(pred)], (170, 45), cv2.FON

        # Draw ROI on frame_copy
        cv2.rectangle(frame_copy, (ROI_left, ROI_top), (ROI_right, ROI_bottom), (255,1

        # incrementing the number of frames for tracking
        num_frames += 1

        # Display the frame with segmented hand
        cv2.putText(frame_copy, "DataFlair hand sign recognition_ _ _", (10, 20), cv2.
        cv2.imshow("Sign Detection", frame_copy)


        # Close windows with Esc
```

```
761/761 [==============================] - 34s 45ms/step - loss: 0.0442 - accuracy: 0.9859 - val_loss: 0.8604 -
val_accuracy: 0.7960 - lr: 5.0000e-04
loss of 1.443190574645996; accuracy of 60.00000238418579%
{'loss': [1.3307597637176514, 0.14614205062389374, 0.04424038156867027], 'accuracy': [0.5400183796882629,
0.9528190493583679, 0.9859377145767212], 'val_loss': [0.798003077507019, 0.9069747924804688, 0.8604211211204529],
'val_accuracy': [0.7623762488365173, 0.7594059109687805, 0.7960395812988281], 'lr': [0.001, 0.001, 0.0005]}
loss of 0.7304924726486206; accuracy of 89.99999761581421%
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 62, 62, 16)        448

 max_pooling2d_8 (MaxPooling  (None, 31, 31, 16)       0
 2D)

 conv2d_9 (Conv2D)           (None, 31, 31, 32)        4640

 max_pooling2d_9 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_10 (MaxPoolin  (None, 6, 6, 64)         0
 g2D)
```

Even though new deep models are constantly being de- veloped, deep learning has long captured the interest of the scientific community. Finding and choosing the best model from the many that are accessible in the literature is dif- ficult. A simple task is choosing optimizers and adjusting optimization hyperparameters. This study evaluates the performance of two deep models that have already been trained, four adaptive gradient-based optimizers, and the tuning of their associated hyperparameters on a static dataset of Indian sign language. InceptionResNetV2 and Adam optimizer may be used for transfer learning-based static sign language recognition, according to experimental results. The Inception-ResNetV2 model outperformed even the most sophisticated machine learning methods by a wide margin.

VII. RESULTS

The initial training had poor results; with 45 epochs, the training accuracy ranged from 0.00 percent to 20.00 percent . Because we believed something went wrong, we halted that. the training accuracy then improved to 89 percent with 761 epochs. loss of 1.443190574645996; value accuracy of 60.00000238418579.A final accuracy of 89.99999761581421 percent is recoderd. It contains a final layers of conv2d(Conv2D) has a Output shape of (None,62,62,16) .Contains maxpooling2d(MaxPooing2D) layers with an outer shape of(None,31,31,16) 4 convolutional layers, including input and output, make up I3D Inception. There are 9 modules for conception. The inception module's details are displayed. We divide the dataset into a 6:2:2 ratio for training. 300 videos would thus be used for the trainingset, 100 for the validation set, and 100 for the testing set.

These results led us to believe that our model was overfitting. It acquired too much knowledge before it could evencategorise the signer. The next step is to test our hypothesis. So, using a new dataset structure, we retrained. We currently use two signers for ten classes with a total of one hundred videos, two signers for twenty classes with one hundred videos, and four signers for forty classes with two hundred films. The two signers' training accuracy ranged from 50.00 to 80.00 percent for the first ten lessons, 100.00 percent for the following ten lessons, and 20.00 percent for the final ten lessons.

VIII. CONCLUSION AND FUTURE WORK

Since we considered that i3d inception without modification was too overfit according to the findings after several trainings, we trained the model with 10 signers and 100 classes during 751 epochs. The model's validation accuracy is quite low despite its reasonable training accuracy. With this model, we are capableof a great deal more, including layer freezing, the removalof some inception modules, the removal of transfer learning, and the transformation of the completely linked layer into a different deep learning model. The completely linked layer is not really the issue, in our perspective. Since the convolutionalneural network layer is the detector, we believe that we shouldpay greater attention to it.

The dataset itself may have contributed to these outcomesin another way. The overfitting may be due to the LSA64's different backdrop lighting. Considering that the variations in backdrop lighting could be a benefit for machine learning. Asa result, machine learning would be trained on the incorrect feature.

REFERENCES

[1] Deaf Cambridge Dictionary. (2018). Retrieved from Cambridge Dictio- nary: https://dictionary.cambridge.org/dictionary/english/deaf

[2] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Computer Vi- sion and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE. Miami, FL, USA .

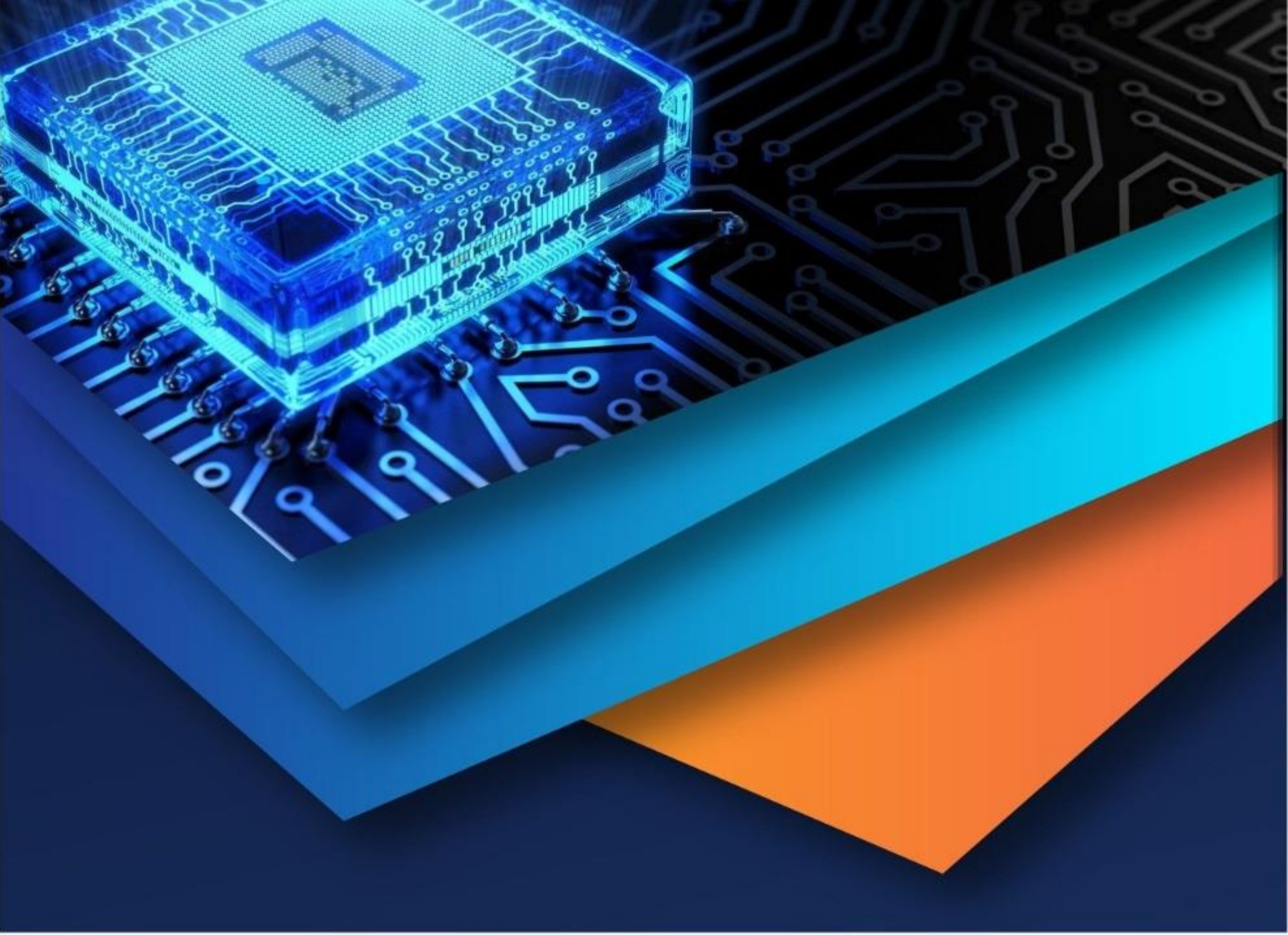
[3] Escalera, S., Baró, X., Gonzàlez, J., Bautista, M., Madadi, M., Reyes, M., . . . Guyon, I. (2014). ChaLearn Looking at People Challenge 2014: Dataset and Results. Workshop at the European Conference on ComputerVision (pp. 459-473). Springer, . Cham.

[4] Feichtenhofer, C., Pinz, A., Wildes, R. P. (2016). Spatiotemporal Residual Networks for Video Action Recognition. Advances in neural information processing systems, (pp. 3468-3476)

[5] B. Kang, S. Tripathi and T. Q. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map," 2015 3rd IAPR Asian Conference on Pattern Recog- nition (ACPR), Kuala Lumpur, Malaysia, 2015, pp. 136-140, doi: 10.1109/ACPR.2015.7486481.

[6] Tomohito Fujimoto, Takayuki Kawamura, Keiichi Zempo, Sandra Puentes, "First-person view hand posture estimation and fingerspelling recognition using HoloLens", 2022 IEEE 11th Global Conference on Consumer Electronics (GCCE), pp.323-327, 2022.

[7] S. Upendran and A. Thamizharasi, "American Sign Language interpreter system for deaf and dumb individuals," 2014 International Conference on Control, Instrumentation, Communication and Computational Tech- nologies (ICCICCT), Kanyakumari, India, 2014, pp. 1477-1481, doi: 10.1109/ICCICCT.2014.6993193.



doi crossref

10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

SJIFactor Scientific Journal Impact Factor

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)