Gesture Recognition

Visibles

ITS20002

Gesture Recognition | Gesture Controlled GUI | Sign-Language Translation | Computer Vision | Deep Learning | Hand Segmentation | Human Computer Interaction (HCI) | Feature Extraction | Image Classification | Neural Networks | LSTM | C3D | LRCN | OpenCV

Team Member Name	Roll Number	Email-Id
Aakriti	190050002	aakriti281020@gmail.com
Divyansh Nankani	190050035	divyansh.nankani@gmail.com
Shirish Chinchanikar	19b090012	shirishchinchanikar2001@gmail.com
Monu Meena	190050068	monumeena549@gmail.com

Inspiration for Idea

There are numerous gesture communications modality, but hand gesture is the most easy and natural way of communication. Hand gesture recognition systems are hot areas of research and have received great attention in recent years because of its manifold applications. The prime application of this was to implement machine translation of sign language. Moreover, natural Human Computer Interaction is in demand in today's world. As we are using more and more devices in our day to day lives, ease of communication becomes a necessity wherein gesture control of machines comes into play.

We initially started with the idea of building the traditional rock, paper and scissors game in such a way that a human can interact with a computer and actually play like we would with another human. But later, realising the numerous use cases of the gesture recognition we decided to extend its functionality to more incorporate more complex applications of the system like the ones mentioned above.

Problem Statement

Our key idea here was to develop a **real-time recognition system using computer vision and deep learning** for image classification to allow feasible detection and identification of hand gestures without the use of external hardware.

- Implementing this involved exploring computer vision techniques to detect the human hand from a computer's live webcam feed and process the image to obtain information about the orientation and pose.
- Once the image data was obtained in a suitable form, its features had to be extracted to build a system that can classify the gesture and display the output.
- In case of gesture control, dynamic gestures have to be recognised and interpreted continuously from video feed in order to perform activity analysis.

Existing solutions in the Market

There are a number of varied ways in Computer Vision available to process human action. Generally, systems for gesture recognition or activity analysis of humans can be broadly divided into one of the following two -

1. **Sensor Input (Hardware Dependent)**
   These involve using sophisticated hardware such as depth mapping cameras and 3D motion tracking sensors to extract information. The most common being Wired Gloves with magnetic or inertial tracking using motion sensors (accelerometers) and rotation sensors (gyroscopes). One such example is the DataGlove.
   There are also high end devices using IR cameras, Time of Flight cameras and Stereo cameras which perform depth analysis to track hand movements. Microsoft Kinect is an example. Most motion controlled gaming consoles use a combination of the above two.
   Thus, there are efficient systems for gesture recognition in this domain but their expensive external requirements rule out the possibility of scaling their application. Also, although many programs have been developed to translate spoken word language, similar technologies for sign languages are lacking and often require specialised machinery for processing visual input.
2. **Deep Learning (Software Dependent)**
   These involve image processing techniques coupled with extensive use of algorithms based on artificial intelligence to extract features from image input hand gestures and learn these features from existing collections of recorded data to predict the gesture or activity being performed. Depending upon input received, there can be models for 3D as well as 2D gesture recognition, the latter involving feed from a simple camera only.
   The algorithms available in this domain are diverse-

➔ Principal Component Analysis using statistical techniques
➔ Feature Extraction Analysis to produce higher level semantic information from image data
➔ Active Shapes Model for analyzing shapes and contours
➔ Template Matching based on simple comparison to existing datasets

Proposed Solution

In order to make our project easily accessible for usage across devices without using complex external machinery, we plan to **implement simple methods of computer vision for feature analysis followed by template matching using deep learning**.
Implementing this consisted of exploring the various image processing and segmentation methods available to detect the hand in the video feed received from a laptop's web camera and to manipulate the image to subtract the background inorder to segment out the position, orientation and features of the hand. This data was then passed through a neural network to learn this data in an organized manner and output the result. Various models and architectures were explored for efficient classification and achieving desirable accuracy. Transfer learning seemed promising for static images and was used for faster training of the model.

Brief Description

The objective of our algorithm was robust skin color detection and removal of static background. Then hand detection and segmentation were attempted. Hand tracking was done by taking the difference of background image with the image with hand in it for fast processing. For better performance, a user's skin color sample was passed and HSV histogram was created. Back Projection is a way of recording how well the pixels of a given image fit the distribution of pixels in a histogram model which was applied on the skin sample.

The black and white background independent image was passed through a Convolutional Neural Network model in case of static gestures. Output layers of different pre-trained models were trained and analysed. For dynamic gesture recognition, temporal information of successive frames had to be processed. This introduced a third dimension of time in our data. So to process such input, two methods were explored - 3-Dimensional CNN and 2D CNN followed by LSTM (Long Short Term Memory) for processing multiple images for video classification..

Progress

Aim for Project

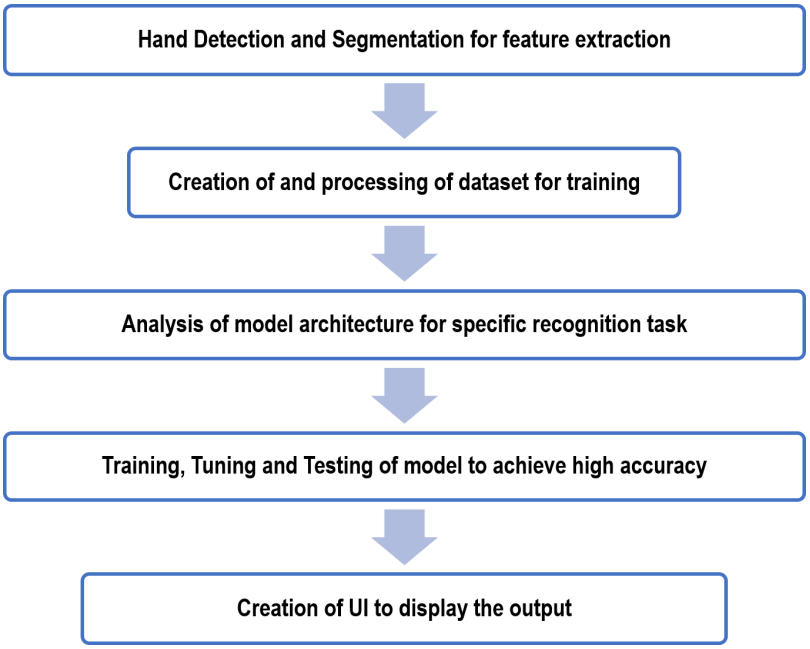
The work involved three independent program creations -

1. A program which detects and recognizes digits in sign language gestures using only the laptop camera and outputs the translated text.

2. An interactive UI for recognising 3 gestures - 'Rock', 'Paper' and 'Scissors'. User's gesture is captured, then Computer plays its move randomly and the winner is decided.

3. Recognising dynamic gestures for performing GUI control. Scrolling, playing/pause of music, volume control etc.

Pipeline of Work



Work in detail and Timeline

1st May to 22nd May :

1) Hand Detection and Segmentation for feature extraction

- A Region of Interest was created as a green box in the video to indicate user to position the hand within it

- The static background of ROI was captured and saved. Then the image with hand was captured and absolute difference with the background image taken
- Gaussian Blurring to denoise, followed by Thresholding and (Dilation + Erosion) to segment hand
- HSV of skin colour captured and skin colour features in ROI detected using Back Projection. Bitwise OR of two images taken to improve segmentation
- Canny edge detection and drawing of contour lines around hand

2) Rock-Paper-Scissor Classifier

- Dataset obtained from the internet and also created by taking pictures of my own hand. Processing to create a dataset of segmented images with 1000 images of each 3 gestures
- Created a CNN classifier by training the output layers of a pretrained model. Three different models were explored for transfer learning - VGG-16, GoogleNet and ResNet101. Their accuracies were recorded.

23rd May to 9th June :

1) Classifier based on American Sign Language (ASL)

- A preprocessed dataset of Alphabets and digits in ASL was obtained from the internet. The images were of low resolution and thus, transfer learning could not be implemented. So a CNN classifier created from scratch was trained.
- After extensive hyperparameter tuning and analysis of accuracies, we realised that the results were not upto the mark! Highest accuracy achieved on real time detection after integration with webcam was ~ 50-60%
- So we created another dataset ourselves consisting of only digits from - 0 to 9 in ASL by taking pictures of our hands and processing the images.
- The focus then was to explore the best pre-trained model to be used for highest accuracy. **9** different models available on Pytorch were explored - VGG-16, VGG-16 with batch normalization(BN), VGG-19, VGG-19 with BN, GoogleNet, ResNet-50, ResNet-101, ResNet-152, ResNeXT50_32

9th June to 22nd June :

1) Video Classifier for Dynamic Gesture recognition

- We did research about the various means available to process multiple frames in order to create a moving gesture recognizer. Numerous research papers were scrutinized and their relevance to our task was studied
- The project aimed more towards research on such classification. Thus, we decided to again create a dataset on our own in order to examine the highest achievable accuracy in our testing. 100 images of 6 different gestures were recorded - hand moved up, palm moved down, one finger moved to right, palm moved to left, open hand to fist motion, closed fist in circular motion
- Two different methods were studied - C3D model using 3-Dimensional Convolutions, LRCN model using 2D CNN followed by an LSTM layer. This involved studying the paper for both the models and implementing them in PyTorch. Both models were trained independently using different learning rates and the best results were compared.
- The C3D model used a third dimension of depth as the number of frames and used 3D ConvLayers on concatenated tensors of multiple images.
- The LRCN consisted of 2 components. A CNN encoder which performed feature extraction and dimension reduction using a pre-trained ResNet101 model on individual frames of videos. This output was passed as a time distributed layer to an RNN decoder using LSTM which processed the temporal information to produce output.

2) Creation of Game UI

- A UI was designed to play the Rock-Paper-Scissors game integrated with the gesture recognition model

3) Output Window of Digit Classifier

- The recognized digits had to be output in a continuous fashion and displayed in a separate output window

4) GUI control implementation

- Based on the recognized dynamic gesture, a task had to be performed by the system. PyAutoGUI was used to send hotkey signals to different applications being used inorder to perform gesture control. The hotkeys were coded with conditional statements

Challenges Faced

1. Data pertaining to human features such as hand vary a lot and training on online datasets with raw image is very difficult. Moreover, skin colour variations can add too much variance to data.
   Solution - Process the dataset to obtain black and white segmented images. Also, calibrate to the skin colour of user using its HSV histogram and back projection

2. Scanning the entire image introduced a lot of noise especially as user's face and neck were of similar colour as hand
   Solution - Introduced a Region of Interest as a small green box where the user should place the hand

3. The current program only works for static background as it calibrates and sets the background at the beginning only. So, on changing background program needs to be re-run
   Solution - To detect change of background we plan to introduce a threshold value to the detected area above which background will be calibrated again.

4. The biggest challenge faced perhaps was the choosing of model architecture and its subsequent training. Most of the time was spent in analysing the accuracies with different changes to the learning rate,the number of convolutions and layers and the optimisation methods. For static gestures many different pre-trained models available had to be compared.

5. For Dynamic gesture recognition, numerous different ways of processing the video feed were available and selecting the ones suitable for our task required extensive research. Understanding and interpreting models from research papers in PyTorch was quite challenging too.

Results

1. Rock-Paper-Scissors Game

The results of different models used in transfer learning is as follows -

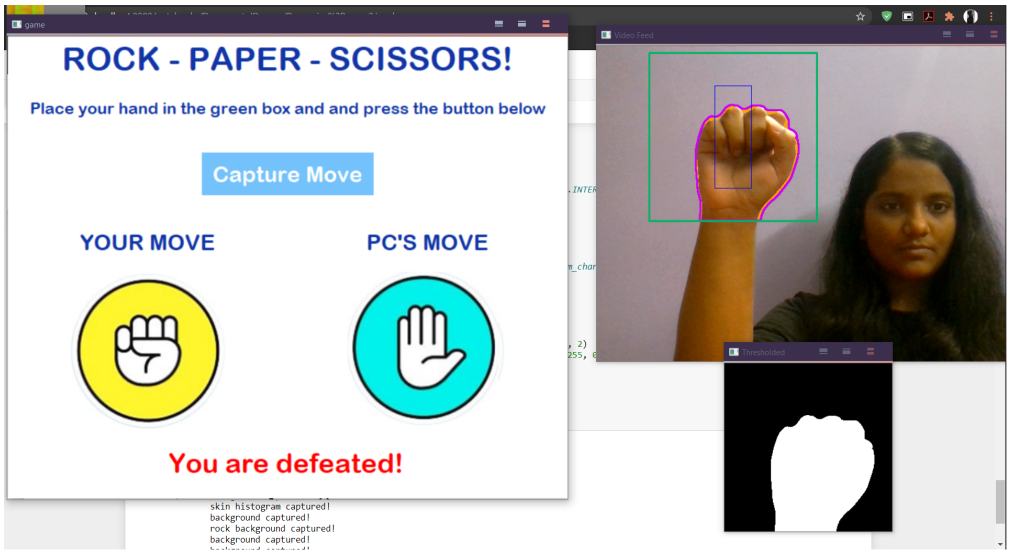
Accuracy Analysis for Rock-Paper-Scissors

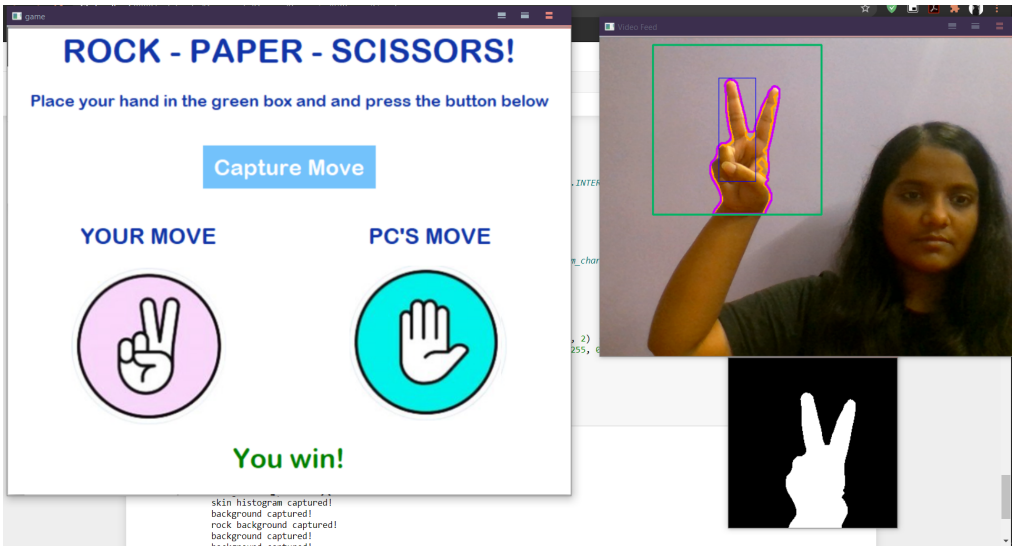
Model	Train set	Valid set
VGG-16	0.992	0.983
ResNet-101	0.991	0.990
GoogleNet	0.995	0.992

The Confusion Matrix while real-time testing is as follows -

Actual Value \ Predicted Value	Rock	Paper	Scissors
Rock	0.98	0.2	0
Paper	0.2	0.95	0.3
Scissors	0	0.3	0.97

Working of Game -





2. Sign Language Digits Classifier

The results of various models used is as follows -

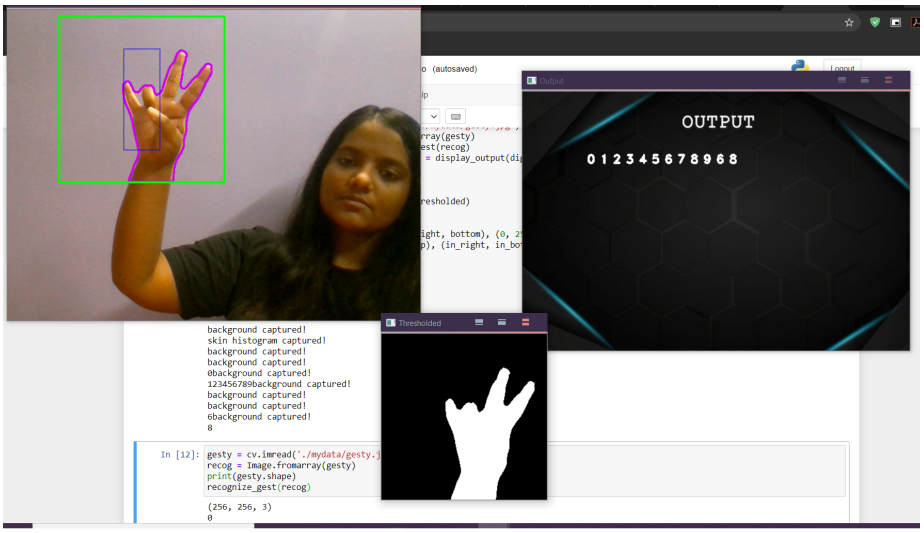
Accuracy Analysis for ASL Digit Classifier

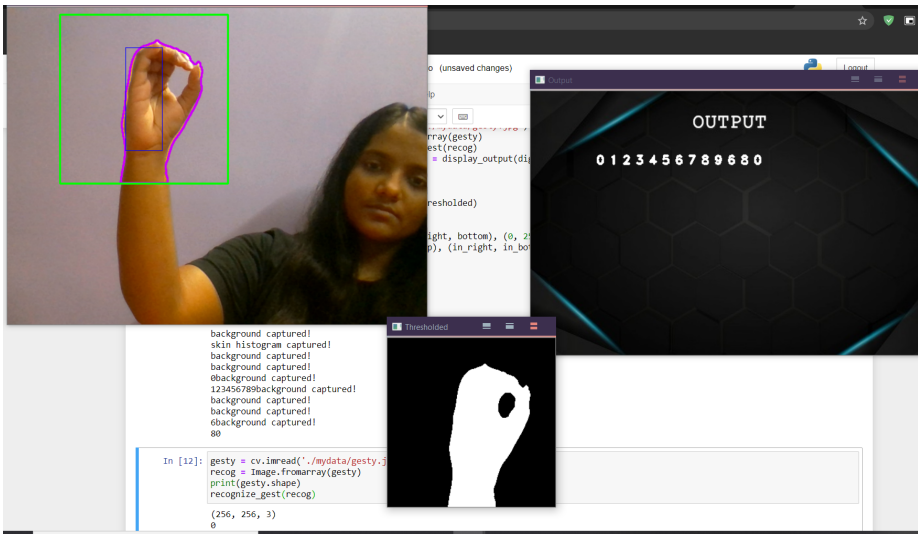
Model	Train set	Valid set
VGG-16	0.974	0.998
VGG-16_BN	0.789	0.825
VGG-19	0.822	0.869
VGG-19_BN	0.815	0.875
ResNet-50	0.910	0.954
ResNet-101	0.918	0.926
ResNet-152	0.980	0.938
ResNeXT50_32	0.913	0.898
GoogleNet	0.884	0.938

The confusion matrix while real-time testing is as follows -

Actual Value \ Predicted Value	0	1	2	3	4	5	6	7	8	9
0	0.97	0	0	0	0	0	0	0	0	0.3
1	0	0.95	0.3	0	0	0	0	0.1	0.1	0
2	0	0	0.91	0.4	0	0	0	0.4	0.1	0
3	0	0	0.2	0.88	0.1	0	0.4	0.2	0.3	0
4	0	0	0	0.2	0.89	0.7	0	0	0	0
5	0	0	0	0	0.3	0.97	0	0	0	0
6	0	0	0	0.4	0	0	0.88	0.2	0.1	0.5
7	0	0	0.1	0.3	0	0	0.5	0.82	0.6	0.3
8	0	0	0	0.5	0	0	0.6	0.6	0.80	0.3
9	0	0	0	0.4	0	0	0.5	0.4	0.5	0.82

Demonstration of working -





3. Dynamic Gesture Control

The results of various models used is as follows -

Accuracy Analysis for Dynamic Gesture Recognition

Model	Train set	Valid set
C3D Model	0.912	0.886
LRCN Model	0.890	0.852

The confusion matrix while real-time testing is as follows -

Predicted Value

Actual Value	0	1	2	3	4	5
0	0.90	0	0	0	0	0.2
1	0.1	0.88	0	0.6	0.3	0.2
2	0	0	0.95	0.5	0	0
3	0.4	0.3	0.5	0.88	0	0
4	0	0.6	0	0	0.91	0.3
5	1.2	0	0	0	0.8	0.80

Number	Class
0	Hand moved up
1	Palm moved down
2	One finger moved to right
3	Palm moved to left
4	Open hand to fist motion
5	Closed fist in circular motion

Further Records

All the documents made about the idea and progress till now are as follows -

1. Abstract of Project
2. Mid-term Report
3. Two - pager report
4. Presentation of Result
5. Demonstration of working in a video
6. Github Repository

Learning Value

We learnt how we can use machine learning to connect humans with computers. Our basic idea was to learn first how the camera will recognize variations in symbols for a particular choice. Direct interaction of humans with machines will involve training of machines to deal with those variations and differences. We understood the concepts behind image segmentation techniques and feature extraction cum analysis of image data and how we can improve it to effectively recognise a hand and its gestures and reduce the noise in the background.

We acquired knowledge about how Neural Networks, Convolutional Neural Networks, Long Short Term Memory units and Time Distributed Layers function and how they can be implemented in PyTorch and what is their significance. We get introduced to Open CV software and how to employ computer vision to process the response of a computer towards the image captured by the camera.

On a more personal level we learned how to conduct research and explore the existing technologies available in an area of interest as well as ideate innovative methods to tackle a problem at hand. We developed the ability of reading and interpreting research papers as well as analysing the content to deploy it with suitable changes to suit our purpose. Added to it we also developed other qualities like teamwork, leadership, time management etc. and made good use of our summer vacations.

Software/ Hardware used

Image Processing : Laptop's Webcam, OpenCV-Python, Python Pillow, Scikit image library

Deep Learning : PyTorch framework, Python libraries like - numpy, matplotlib, sklearn etc, Google Colab for training with GPU, Jupyter notebook for writing code offline

GUI : PyAutoGUI

Suggestions for others

There are many challenges associated with the accuracy and usefulness of gesture recognition software. For image-based gesture recognition there are limitations on the equipment used and image noise. Images or video may not be under consistent lighting, or in the same location. Items in the background or distinct features of the users may make recognition more difficult.

Contribution by each Team Member

Aakriti - Hand segmentation algorithm designing and implementation, creation of Dataset, Training, tuning and testing of models for Rock-paper-scissors, Research on relative accuracies of different models on for dynamic gesture recognition, Creation of UI for each program

Divyansh - Training, testing and validating the Neural Network over the American Sign Language using PyTorch.

Shirish - Creation and processing of Dataset for Rock-Paper-Scissors to segment images

Monu - Training, testing and validating the Neural Network over the American Sign Language

using Pytorch.

References and Citations

1. Y. Wu, B. Zheng and Y. Zhao, "Dynamic Gesture Recognition Based on LSTM-CNN," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 2446-2450, doi: 10.1109/CAC.2018.8623035.
2. J. Sun, T. Ji, S. Zhang, J. Yang and G. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 2018, pp. 1-4, doi: 10.1109/ISAPE.2018.8634348.
3. Sun, Jing-Hao & Ji, Ting-Ting & Zhang, Shu-Bin & Yang, Jia-Kui & Ji, Guang-Rong. (2018). Research on the Hand Gesture Recognition Based on Deep Learning. 1-4. 10.1109/ISAPE.2018.8634348.
4. D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 4489-4497, doi: 10.1109/ICCV.2015.510.
5. R. Rokade-Shinde and J. Sonawane, "Dynamic hand gesture recognition," 2016 International Conference on Signal and Information Processing (IConSIP), Vishnupuri, 2016, pp. 1-4, doi: 10.1109/ICONSIP.2016.7857476.
6. Munasinghe, Nuwan. (2018). Dynamic Hand Gesture Recognition Using Computer Vision and Neural Networks. 10.1109/I2CT.2018.8529335.
7. https://github.com/HHTseng/video-classification/blob/master/CRNN/functions.py

Licenses