Real time static gesture detection using machine learning

|  |  |
| --- | --- |
| 1st Sandipgiri Goswami  Computer Science Department  Laurentian Unversity Sudbury,Canada [sgoswami@laurentian.ca](mailto:sgoswami@laurentian.ca) | 2nd Dr. K Passi  *Computer Science Department*  Laurentian UnversitySudbury,Canada [kpassi@cs.laurentian.ca](mailto:kpassi@cs.laurentian.ca) |

*Abstract*—Sign gesture recognition is an important problem in human computer interaction with signiﬁcant societal influence. However, it is a very complex task, since sign gestures are naturally deformable objects. Gesture recognition contains unsolved problems since last two decades, such as low accuracy or low speed, and despite many proposed methods, no perfect result has been found to explain these unsolved problems. In this paper, we suggest a machine learning approach to translating sign gesture language into text.

In this study, we have introduced self generated image data set for American sign language (ASL). This dataset was a collection of 36 characters which contain A to Z alphabets and 0 to 9 number digits. The proposed system can recognize static gestures. This system can learn and classify specific sign gesture of any person. We used a convolutional neural network algorithm for classified image to text. We achieved 99.00% accuracy on the alphabet gestures and 100% accuracy on digits.

Keywords: Sign gestures, Image processing, Machine learning, Conventional neural network.

1. Introduction

The World Health Organization (WHO) estimated that, 250 million people in the world are deaf as well as dumb [1]. These group of people of group use symbolic language to communicate with other people. This symbolic language is called sign language. Sign Language is a built for communication used worldwide among hard of hearing and deaf people. Sign language is not a unique language signed consistently in different countries. Sign language is not recent improvement. There is proof that speaking through gestures has been around since the start of human development [20]. Different counties have their own sign language such as American Sign Language, French Sign Language, Indian Sign Language and Puerto Rican Sign Language to a name a few. Table 1 gives information about different sign languages used in western continent. Gesture based communication is dependent on region and has significant differences from other languages. It is very important to understand sign language when we communicate with deaf or young children and their families. Lack of understanding results in significant challenges in understanding this community and may result in miscommunication. Sign Language is a language which is used to convey messages by hand movements, facial expression and body language for communication. It is mainly used by deaf and people who can hear but cannot speak. Sometime family members and relatives must learn sign language to interpret which enables deaf and wider communities to communicate with each other.

In this thesis, Image classification and machine learning have been used for interpreting American sign language. For image classification, computer vision algorithms were used to capture images and to process data set for filtering as well as reducing noise from images. Finally, data set is trained using machine learning algorithm, conventional neural network for measuring accuracy of training data set. The abstract view of the derived approach combining the image classification and machine learning for American sign language is shown in Figure 1.

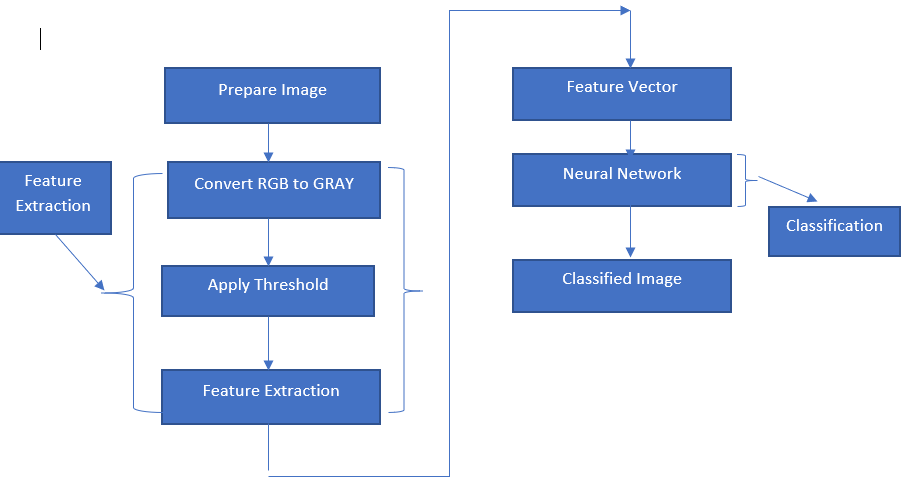


Fig. 1. System Architecture.

1. Related Work:

Machine learning is most commonly used for image recognition. Hidden Markov Model (HMM) and Dynamic Time Warping (DTW), two kinds of machine learning methods, are widely applied to achieve high accuracies [5, 6, 7]. These are mostly good at capturing time-based patterns, but they require clearly characterized models that are defined before learning. Starner and Pentland [5] used a Hidden Markov Model and a 3-Dimesional glove that detects hand movement. Since the glove can attain 3-Dimesional detail from the hand regardless of spatial orientation, they achieved the best accuracy of 99.2% on the test set. Using Hidden Markov Model uses time series data to track hand actions and classify based on the position of the hand in recent frames.

As per research point of view a linear classifier is easy to work with because linear classifiers are relatively simple models, it requires sophisticated feature extraction and preprocessing methods to get good results [2, 3, 4]. Singha and Das [2] achieved an accuracy of 96% on Ten classes for images of gestures of one hand using Karhunen-Loeve Transforms. These translate and rotate the axes to build up a new framework based on the variance of the data. This technique is useful after using a skin color detection, hand cropping and edge recognition on the images. They use a linear classifier to recognize number sign including thumbs up, first and index finger pointing left and right, and numbers only. Sharma [4] has done research using Support Vector Machines (SVM) and k-Nearest Neighbors (KNN) to illustrate each color channel after background noise deletion and noise subtraction. Their research suggests using contours, which is very useful to represent hand contours. They got an accuracy of 62.3% using a Support Vector Machines on the segmented color channel model.

American sign language recognition is not a new machine learning problem. During recent decades, different researchers already worked on different classifiers such as linear classifiers, neural networks and Bayesian networks [2-11].

Suk [6] suggested a system for detecting hand gestures in a continuous video stream using a dynamic Bayesian network or DBN model. They try to classify moving hand gestures, such as creating a circle around the body or waving. They attain an accuracy of nearly 99%, but it is worth noting that all hand gestures are different from each other and are not American Sign Language. However, the motion-tracking feature would be applicable for classifying the dynamic letters of ASL: j and z.

1. Data set and Variables

I have created my own data set. This dataset was a collection of 36 characters which contain A to Z alphabets and 0 to 9 number digits. I used right hand to capture 1000 images for specific alphabets and numbers. The height and width ratios vary significantly but average approximately 50X50 pixels. The dataset contains over 36,000 images in grey scale color. Additionally, people can add their images to this dataset. Below figure shows an image of A to Z alphabet and

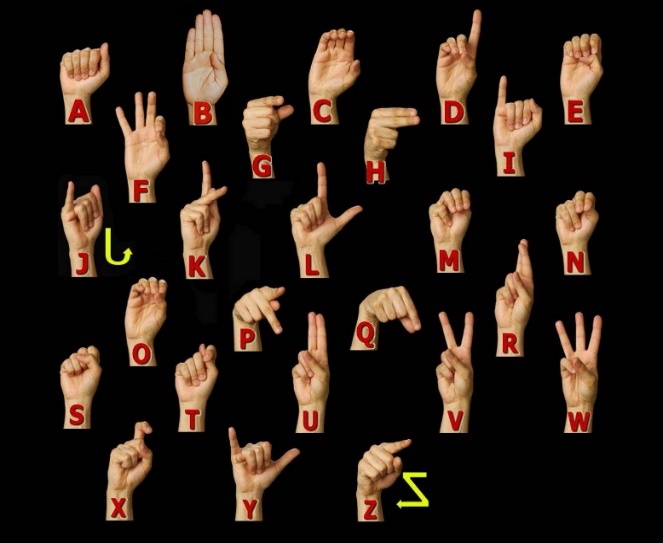


Fig. 2. American Sign language Manual Alphabet [].

****

Fig. 3. American Sign language Manual Number [].

Table I: Dataset Description and Image property.

|  |  |
| --- | --- |
| **Property** | **Description** |
| Alphabets | A to Z |
| Numbers | 0 to 9 |
| Color | Grey Scale |
| Dimensions | 50x50 |
| Height | 50 pixels |
| Width | 50 pixels |
| File type | JPEG |

1. My Approach for Hand Detection

Used for detecting hand gesture using skin colour, there are different approaches including skin colour-based methods.In my case, after detecting and subtracting the face and other background, skin recognition and a contour comparison algorithm were used to search for the hand and discard other background colour objects for every frame captured from a webcam or video file.Palm to extract their contours and saved the four for evaluation with the contours of the skin detected area of every frame.After detecting the skin area for each frame captured, I compared the contours of the detected areas with the previously saved hand histogram template contours to remove other skin like objects existing in the image.If the contour comparison of the spotted skin area complies with any one of the saved hand histogram contours than it captured only hand gesture only.

I propose an integrated system for detection, segmentation, and tracking of the hand in a gesture recognition system using a single webcam. Some other methods that use color gloves [39, 40], my method can detect the plain hand posture by integrating two useful features: skin color detection and contour matching. My proposed hand posture, finding algorithm has real-time performance and is strong against rotations, scaling, a cluttered background, and lighting conditions. The strength of my proposed hand posture detection algorithm based on comparison with other different methods. Detecting the human hand in a cluttered background will boost the performance of hand gesture recognition systems. In this method, the speed and result of recognition will be the same for any frame size taken from a webcam such as 640×480, 320×240 or 160×120 and the system will be also robust against a cluttered background because I process the detected hand posture area only.

To detect the hand gesture in the image, a four-phase system was designed according to my approach and as

shown in Figure 4.1. First, we will open camera which has 50 x 50 square box to capture hand gesture. Second Put your hand in this box and make sure your hand covers inside box. Third, the skin color locus for the image was removed from the user’s skin color after face deletion. Then the last step, the hand gesture was spotted by removing false positive skin pixels and identifying hand gesture and other real skin color regions using contours matching with the loaded hand gesture pattern contours. Skin Recognition Area Loading Hand Postures Patterns Contours Face Detection and Subtraction Capturing Images from Webcam or Video file Templates Contours

Comparison with Skin Area Figure 5: Hand posture detection steps



Fig. 5. Hand posture detection steps

1. *Skin Detection*

Skin detection is a useful approach for many computer vision applications such as face recognition, tracking and facial expression, abstraction, or hand tracking and gesture recognition. There are recognized procedures for skin color modeling and recognition that will allow to differentiate between skin and non-skin pixels based on their color. To get suitable distinction between skin and non-skin areas, a color transformation is needed to separate luminance from chrominance [42].

The input images normally are in Color format (RBG), which has the drawback of having components dependent on the lighting situations. The misunderstanding between skin and non-skin pixels can be decreased using color space transformation. There are different approaches to detection skin color components in other color spaces, such as HSV, YCbCr, TSL or YIQ to provide better results in parameter recovery under changes in lighting condition. Researches have shown that skin colors of individuals cluster closely in the color space for all people from different societies, for example, color appearances in human faces and hands vary more in intensity than in chrominance [41, 43]. Thus, take away the intensity V of the original color space and working in the chromatic color space (H, S) provides invariance against illumination situations. In [42], it had been well-known that removal the Value (V) component and only using the Hue and Saturation components, can still permit for the detection 96.83% of the skin pixels. In my application, I use the hue, saturation, value (HSV) color model since it has shown to be one of the most adapted to skin-color detection [44]. It is also well-matched with the human color perception. In addition, it has real-time execution and it is more robust in cases of rotations, scaling, cluttered background, and changes in lighting condition. So, my projected hand gesture detection algorithm is real-time and robust against the mentioned previous changes. The other skin like objects existing in the image are removed from contour comparable with the loaded hand postures prototype contours.

The HSV color space is gained by a nonlinear transformation of the essential RGB color space. The conversion between RGB and HSV was described in [45]. Hue (H) is a section that characterizes a pure color such as pure yellow, orange or red, whereas saturation (S) provides a measure of the degree to which a pure color diluted by white light [46]. Value (V) attempts to represent brightness along the gray axis such as white to black, but since brightness is subjective, it is thus difficult to measure [46].

According to [47] and Figure 6, Hue is estimated in HSV color space by a position with Red starting at 0, Green at 120 and Blue at 240 degrees. The black mark in the diagram at the lower left on the screen determines the hue angle.

Saturation is a ratio that ranges between 0.0 along the middle line of the cone (the V axis) to 1 on the edge of the cone. Value ranges, string from 0.0 (dark) to 1.0 (bright). 

Fig. 6. HSV Color Space

According to [41], the HSV model can be resulting from non-linear transformation from an RGB model according to the following calculations.



As per a classification point of view, skin-color detection divided into two class problem: skin-pixel vs non-skin-pixel classification. Currently, there are different known classification approaches exits such as thresholding, Gaussian classifier, and multilayer perceptron [48, 52, 53].

In my research, I used a thresholding technique that allows getting a good result for higher computation speed when compared with other techniques, given our real-time requirements. This thresholding classification is used to find the values between two components H and S in the HSV model as I removed the Value (V) component. Usually, a pixel can be observed as being a skin-pixel when the following threshold values are synchronized satisfied: 0° < H < 20° and 75° < S < 190°.

1. *Contour Comparisons*

Once the skin color has been detected, the contours of the detected skin color are recovered and then compared them with the contours of the hand gesture patterns. Once skin color contours are recognized as belonging to the hand gesture contour patterns, that area will be identified as a region of interest (ROI) which will then be used for tracking the hand movements and saving the hand posture in JPEG format in small images as shown in Figure 7. After that stored images will further be used to extract the features needed to recognize the hand gestures in the testing stage.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Fig. 7: Images of detected hand postures.

If there is both hand gesture in the image, my system will substitute in detecting one of the two hands for every frame captured because the Open Computer vision function cvBoundingRect will circle one rectangle only around the detected hand, which has the main matching contours with the overloaded hand posture templates contours. The single frame will circle the detected hand posture for one frame and may enclose the other hand posture for the next frame if it has a higher matching contour.

1. Method

Our overarching approach was on of basic Supervised learning method is more commonly used. This method needs training data with specific format. Each instance must have assigned label. These labels make available supervision for the learning algorithm. Training process of supervised learning is constructed on the following principle. First, the training data are fed into the model to produce estimate of output. This estimate is compared to the assigned label of the training data in order to evaluation model error. Based on this error the learning algorithm alters model’s parameters in order to reduce it.

*A. Architecture*

Our architecture was commonly used in CNN architecture. In this architecture consisting of multiple convolution and dense layer. The CNN architecture included three type of two convolution layer and each layer has their own max pooling layer and one group of fully connected layer followed by a dropout layer and output layer.



Fig. 8. CNN network architecture for Alphabets.

*B. Hardware and Software Configuration*

Training of Neural Networks notoriously computational expensive and it required a lot of resources. From bottom level perspective it translates into many multiplications of matrices. Modern Central Processing Units (CPUs) are not made of such computations and therefore are not very efficient. On the other hand, modern GPUs are designed to preform exactly these operations.

At present on the market there are two main parallel computing platforms CUDA and OpenCL. They both have their own advantage and disadvantage, but the major difference is that CUDA is proprietary, while OpenCL is available free. This divide translates into hardware productions as well. CUDA is mostly supported by NVIDIA and OpenCL is support by AMD. NVIDIA with its CUDA platform is presently leader in the domain of deep learning. Therefore, for training of CNN models was selected GPU from NVIDIA. Selected training model was GIGA BYTE GeForce GTX 1080. Details information about hardware configuration is in Table 2.

Table II: Hardware Configuration

|  |  |
| --- | --- |
| GPU | GeForce GTX 1080 4GB |
| CPU | Intel(R) Core(TM) i7-8550 CPU @ 2.00GHz |
| Memory | DIMM 1333MHz 8GB |

From the list of considered software tool was selected Keras. The reason being that Keras satisfied all consideration factors and because it was written in python which was most aware to the me. Support of efficient GPU in Keras is relying on either Tensor flow or Theano back-end. From the different user perspective, it doesn’t really mater either way, but Tensor flow was selected because it was observed as faster of the two. GPU-accelerated library package of primitives for deep neural networks. Details information about software configuration is brief in table 3.

Table III: Software Configuration

|  |  |
| --- | --- |
| Keras | 2.04 |
| Tensorflow | 1.1.0 |
| CUDA | 7.5 |
| Python | 3.53 |
| Operating System | Window 10 |
| Open CV | 2.0 |

1. Result

On our self-generated dataset, we achieved 99.00% accuracy on the alphabet gestures and 100% accuracy on digits. We did real time testing with different five students and estimate per user took 20 minutes time for alphabets and approximate 7 to 8 minutes for digits. We have tested with different lighting condition and different place. Confusion matrix and result accuracy graph shown in below.



Fig. 9. Epochs vs. validation accuracy for digits.



Fig. 10. Epochs vs. validation accuracy for alphabets.



Fig. 11. Confusion matrix for 0 to 9 digits.



Fig. 12. Confusion matrix for A to Z alphabets.

1. Conclusion and Feature work

In this paper the we developed system to recognised American Sign gesture using skin color model, thresholding and CNN. We have tested with different lighting condition and in different place. The dataset collected in the ideal conditions has proved to be the most efficient dataset in terms of accuracy and gives 99% accuracy on alphabets and 100% accuracy on digits.

Sign gesture recognition still has a long way to go in the research path, especially for 2D systems. This study offers fascinating ideas for future research. Some of these possibilities are defined in this section. As this thesis focused only on static sign gesture recognition, one next step forward is to recognize the dynamic sign gesture for the ASL. Even though that study introduces a self generated new dataset with a rather more gesture for American Sign Language, it still does not offer all the possible movements for American Sign Language. Videos with rotation in 3Dimension, words and expressions are examples of how this dataset can be extended.

1. References