Universität Hamburg
Department Informatik
Knowledge Technology, WTM

# Gesture Analysis

Seminar Paper

Bio Inspired Artificial Intelligence

Soubarna Banik
Matr.Nr. 6640587
4banik@informatik.uni-hamburg.de

17.12.2014

# Abstract

Gesture analysis is one of the essential component of Human Computer Interaction. Understanding a variety of hand gestures in real life scenarios is not a trivial problem. In this paper, two research studies are discussed, which aim to analyze gestures using biologically inspired techniques. A comparison of these studies is included, along with a few remarks for possible enhancements.

# Contents

# 1 Introduction

We have crossed a long distance since computers were developed. From the science laboratories, computers have reached every household today. Applications are being developed for every human, including old and young people who are not familiar with these technologies. And hence, a change in how we interact with computers was needed. It was becoming essential to make computers learn our language, our behavior in stead of we learning their language. One solution was to analyze and recognize human gestures. In our daily life, we use gesture as a mode of interaction often, knowingly and unknowingly. It is a very convenient way by which instruction can be given to computers. It will also outdo the existing interfaces such as mouse, keyboard etc.

A lot of research has been done regarding gesture analysis. There are numerous successful approaches based on computer vision. Most of them involve high computational load and expensive devices which makes it difficult to implement for most users. On the contrary to the computer vision approaches, the bio-inspired approaches try to solve the vision problem from the point of view of humans.

In this paper, we will discuss two bio-inspired approaches for solving the problem of gesture analysis - Elastic Graph Method (EGM) and Convolution Neural Network (CNN). For both of the approaches, only static gesture or the hand posture is considered. In section 2, the EGM algorithm is described and then a variation of EGM, the Elastic Bunch Graph Matching (EBGM) algorithm that was used in [6] for gesture analysis is dicussed. In section 3, first the convolution neural network is described and then an adaptation of CNN, the Multichannel Convolution Neural Network (MCNN) which was used in [1] for gesture analysis is discussed. In section 4, we will do a comparative study between these two approaches, [1] and [6] where they have used the same base database, JTD for gestures.

# 2 Elastic Graph Method

As stated by Horst Bunke in [2], graphs are a general and powerful data structure for the representation of concepts and objects. In applications such as pattern recognition and computer vision, object similarity is an important issue. Given a database of known objects and a query, the task is to retrieve one or several objects from the database that are similar to the query. If graphs are used for object representation this problem turns into determining the similarity of graphs, which is generally referred to as *graph matching*. One such algorithm is Elastic graph matching.

## 2.1 Biological Background

Elastic Graph Matching (EGM) is a biologically inspired algorithm for object recognition. It is biological inspired in two ways: (i) For feature extraction, filters based on Gabor wavelets are used. Gabor wavelets have been found a good model for representing the visual processing in the brain, more precisely simple cells in primary visual cortex [3]. (ii) The matching algorithm is an algorithmic version of dynamic link matching (DLM), which is a model of invariant object recognition in the brain.

## 2.2 EGM Algorithm[4]

Martin Lades came up with a distortion invariant object recognition algorithm which is based on Dynamic Link Architecture. As described by the author in [4], objects are represented as labeled graphs, which contain a two-dimensional array of nodes. The nodes are labeled with some features which describe the corresponding gray level distribution locally, with respect to the neighboring nodes with high precision and globally with a lower precision. The edges of the graph are labeled with a metric describing the relative position of the vertices i.e. the distance between the vertices. For recognizing an object in an image, the labeled graphs of the model objects, also called the model graphs are matched onto the image. For each model graph, the location of the nodes in the image are arranged such that the local feature of each node matches that of model and the distances between the locations fit the distances between the nodes of model. The best match would be the one that preserves the feature and local geometry of the graph. The match can be elastic in the sense that the preservation of the local geometry can be approximate. However, they should not differ much from the original distance.

Let us assume, $I(\vec{x})$ is the gray scale distribution of the input image. For labeling the nodes of the image, a linear filter operation is done. A convolution operation is done between the image $I$ and a family of kernels, $\psi_{\vec{k}}$.

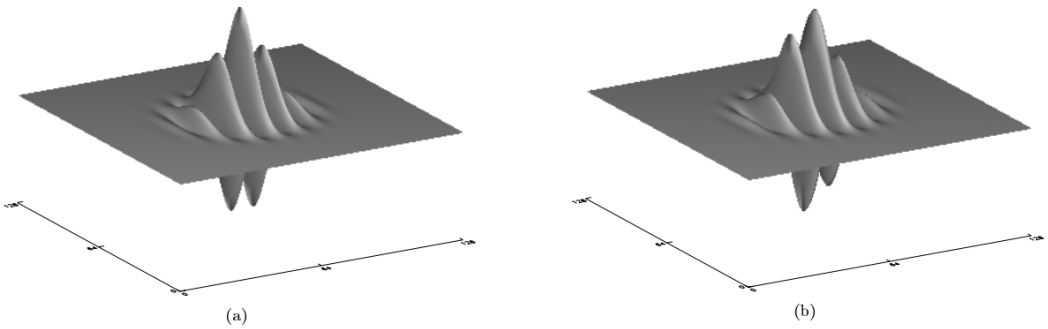



Figure 1: The shape of Gabor-wavelet (a) The real part (b) The imaginary part

$\psi_{\vec{k}}$ is called "Gabor-based wavelets" and it is of the form of a plane wave with wave-vector $\vec{k}$. It is restricted by a Gaussian envelope function. $\vec{k}$ decides the width of the Gaussian window. The kernel is described as

$$\psi_{\vec{k}}(\vec{x}) = \frac{\vec{k}^2}{\sigma^2} \exp(-\frac{\vec{k}^2\vec{x}^2}{2\sigma^2})[\exp(i\vec{k}\vec{x}) - \exp(\frac{-\sigma^2}{2})]$$

The first term signifies the oscillatory part of the kernel, whereas the second term compensates for the dc value of the kernel. For sufficiently high value of $\sigma$ the effect of the dc term becomes negligible. Hence, these filters are DC-free. The wave-vector $\vec{k}$ also determines the wavelength and orientation of the oscillatory part.

In order to get a smooth peak for matching, only the magnitude of the complex function resulted from convolution of image with kernels is considered. This magnitude is a positive valued real function of $\vec{k}$ corresponding to every point in the image domain. The image is sampled at 5 logarithmically spaced frequency levels denoted by $v \in \{0, ..., 4\}$ and at 8 orientations $\mu \in \{0, 1, ..., 7\}$.

$$\vec{k}_{v\mu} = k_v e^{i\phi_\mu},\ where\ k_v = k_{max}/f^v\ and\ \phi_\mu = \frac{\pi\mu}{8}$$

where $f$ is the spacing factor between the kernel samples in the frequency domain. The magnitudes of convolution operation will signify the feature vector located at a particular node. This is also called *jets*, $J$.
The similarity between the image vertex label, $J^I$ and the model vertex label, $J^M$ are determined as below

$$S_v(J^I, J^M) = \frac{J^I.J^M}{||J^I||||J^M||}$$

and the similarity between the edge labels of image graph and the corresponding edge in model graph is given by

$$S_e(\vec{\Delta}^I_{ij}, \vec{\Delta}^M_{ij}) = (\vec{\Delta}^I_{ij} - \vec{\Delta}^M_{ij})^2$$

where $\vec{\Delta}_{ij}$ is the Eucledian distance between vertices $x_i$ and $x_j$.

EGM finds out the set of vertex positions in image domain that optimizes the matching of vertex labels and also the edge labels. The cost function, $C_{total}$ that evaluates this optimization is given by:

$$C_{total} = \lambda C_e + C_v$$

where $C_e$ is the summation of $S_e$ for all edges and $C_v$ is the summation $S_v$ for jets of all nodes.

## 2.3 Approach for gesture analysis using EBGM

Jochen Triesch has adapted EGM and used it for recognizing hand postures in [6]. There was a few variation in choosing the parameters: instead of 5 scales he used 3 scales, $v \in \{0, 1, 2\}$, $f$ was chosen as $1/\sqrt{2}$ and $k_{max} = 1.7$. For the Gaussian envelope function, $\sigma$ was set to 2.5. The similarity functions were chosen as :

$$S_{abs}(J, J^{'}) = \frac{\sum_j a_j a_j^{'}}{\sqrt{\sum_j a_j^2 \sum_j a_j^{'}2}}$$

$$S_{pha}(J, J^{'}) = \frac{1}{2}(1 + \frac{\sum_j a_j a_j^{'} \cos(\phi_j - \phi_j^{'})}{\sqrt{\sum_j a_j^2 \sum_j a_j^{'}2}})$$

where $S_{abs}$ considers only the magnitude of the convolution output and $S_{pha}$ considers both magnitude and phase.

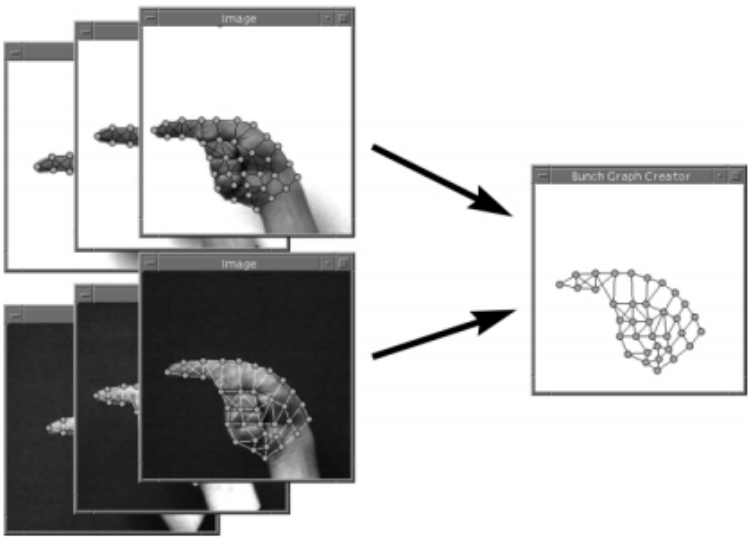



Figure 2: Creating bunch-graph from six single graphs

The author has considered multiple jet values for a single node in the model graph, in order to include the variability of feature at each node. A point on a hand may appear different in different background and it may also vary with respect to the person. As a result, the author has considered six graphs for every posture - hand posture of 3 persons, each taken in a light and a dark background, as described in Figure 2. The jet values are calculated for all six graphs and a *bunch-jet* $B^n$ is defined for node n as $B^n = \{J^n(1), ...J^n(6)\}$. For the length of the edge, the average value was taken.
For comparing the bunch graph, the similarity between a single jet $J(x)$ taken at a point x in image domain and all jets in the bunch-jet $B^n$ are calculated and the

maximum similarity value was considered. The total similarity of the graph G is the average of similarity values for all its nodes. $S^G_{abs}$ and $S^G_{pha}$ are the absolute and phase similarities respectively. The cost function, $C^G$ for graph G with $M$ edges was defined as average cost for all edges.

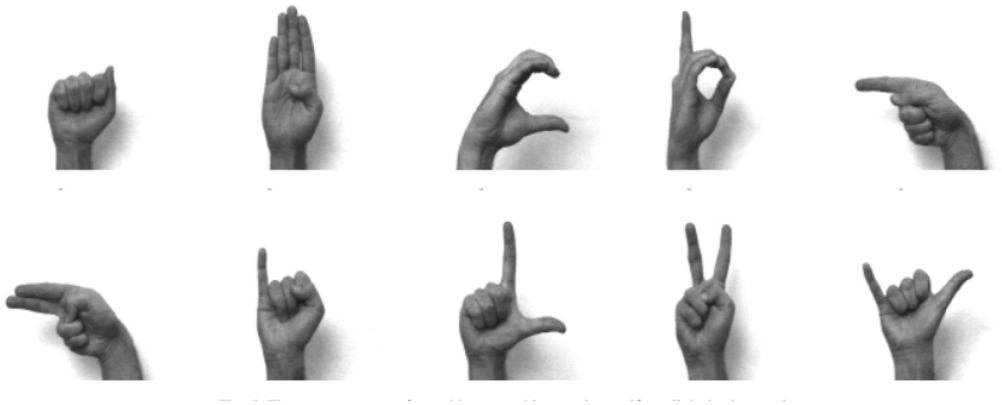


Figure 3: The ten postures used as models in JTD

The matching algorithm, as followed by the author, is:

1. Coarse positioning of the graph: The image is scanned in coarse steps using a 5x5 window and local image similarities,$S^G_{abs}$ are calculated for each window. No graph distortion is done in this step.

2. Re-scaling of the graph: Once the graph is detected, if needed, it is allowed to grow or shrink by maximum 20%.This is done to accommodate different size of hands or different positions of camera. The graph can also be shifted by 9 pixels in x and y direction to fit the model graph. After all transformations, the local image similarities,$S^G_{abs}$ are calculated again.

3. Local diffusion of single node: In order to fit a single node with the model, in other words to allow distortion, a node can be moved inside a 9x9 window around it and the total similarity,$S^G$ is calculated for each position. The position with the highest similarity is chosen as the new position of the node. This is repeated for all nodes.

The bunch graph of the input image is compared with all model graphs and the posture with the highest similarity is considered as output. 10 different postures, as depicted in Figure 3 were considered as model graph in this article.

# 3 Convolution Neural Network

## 3.1 Architecture of Convolution Network [5]

Convolution network is a biologically inspired model that extracts and enhances the local feature of an image. The architecture of convolution network was devel-

oped by Yann Lecun for document recognition [5]. It has a layered architecture where each layer consists of a convolution and a sub-sampling step. Figure 4 describes the model that was implemented by Lecun.

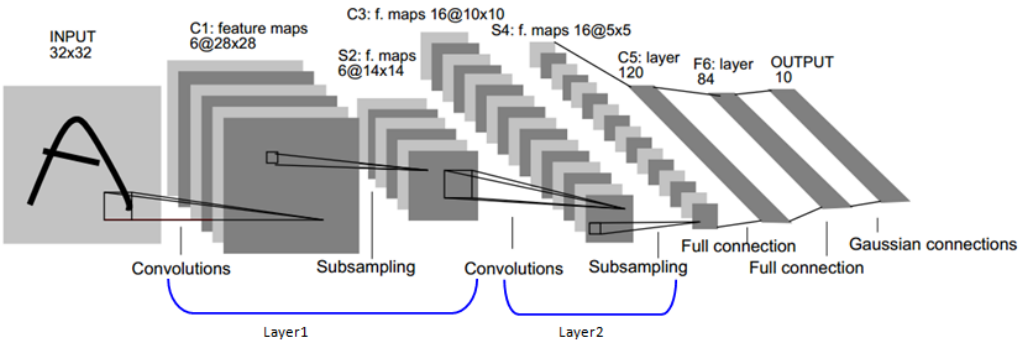



Figure 4: Architecture of Convolution Neural Network

**Convolution**: Convolution operation finds out the overlapping between two functions, to be precise, the similarity between two functions. In the first layer, convolution operation is done on the input image with a set of kernels which correspond to each feature that needs to be extracted. The output of the convolution step is multiple feature maps signifying multiple features for the same image. In Figure 4, convolution is done on the 32x32 input image with 6 different kernels of the size 5x5. It generates 6 feature maps of size 28x28. One interesting thing to notice here, a unit in a feature map in the first layer receives input from 25 units located in a neighborhood of 5x5 window in the input image. This property of CNN is known as local receptive field. The receptive fields of neighboring units in feature map overlap. Another property of convolution network is the weight sharing property. The weight vector for a particular feature map is constant throughout the image constraining it to perform same operation on different part of the image. For different feature maps the weights are different.

**Sub-sampling**: Once a feature is extracted in the convolution layer, its exact position becomes irrelevant as long as its relative position with other feature is conserved. Sub-sampling is done by performing a local averaging on the subsample window, multiplying it by a trainable coefficient and adding a trainable bias and then passing it through a sigmoid function. This reduces the resolution of the image. Sub-sampling increases the robustness of the architecture with respect to invariance towards shift and distortion of the input. In Figure 4, 2x2 sub-sampling is done by taking an average of 4 inputs from the feature map. The sub-sampling window is non-overlapping, which reduces the resolution of the image by half - from a 28x28 feature map from the previous layer a to a 14x14 feature map is generated by sub-sampling.
With each layer the number of feature map is increased and the spatial resolution is decreased. The weights are learned with back propagation.

### 3.2 Approach for gesture analysis using MCNN

The multi-channel convolution neural network proposed by Pablo Barros in [1] uses multiple channels of the same input image diversified by different operators instead of using one single input channel.

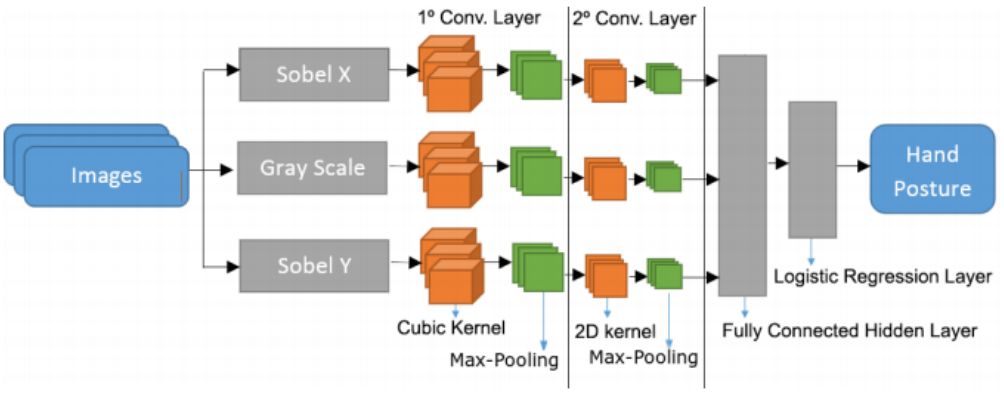



Figure 5: Architecture of Multichannel Convolution Neural Network using 3 channels

In this article, the author has used three input channels  the raw image and two image outputs from sobel operation on the original image, one in horizontal direction and the other in vertical direction. All the channels have the same number of layers and same parameters but different weights. They are connected by a fully connected hidden layer that generates a single output for the final logic regression layer. Another adaptation to the original CNN was use of a cubic kernel in the first convolution layer. In order to increase the learning rate, cubic kernel was applied on a set of image that represents the same posture and also the variance of the posture thereby simulating a 3D filter. For each image and for each convolution region, it will generate multiple units. However, the value, which presents the maximum distance from the mean of all these generated units, is considered for the next layer. This reduces computation cost for this cubic kernel operation. Each unit is connected with a region in an image in the previous layer with a weighted connection, however the connections are not shared across images. The author used the JTD database that was used in [6] and also on a set of images, taken by a robot, called NAO in a home like laboratory, simulating a real world scenario. For the later set,which is referred as NCD, only four hand postures were considered but in a different position in each image.

## 4 Comparative Study

Both of the experiments were conducted on the same database, JTD as shown in Figure 3. It consists of 10 hand postures against uniform light, uniform dark

Figure 6: Examples of hand postures recorded with NCD

and complex backgrounds. The hand postures were performed by 24 persons to include the difference in size and shape of the hand posture. All images were grey level, 128x128 and the hand posture was at the center position. As mentioned, the lighting situation was constant for all images in JTD. For NCD, 4 hand postures were performed with 400-500 examples per hand posture at different positions. These images had the same color scheme and size. For the first experiment, 60 images were chosen from the light and dark background as training set and the rest for testing. For the second experiment, 60% images were considered for training, 20% for validation and rest for testing. The experiment was conducted on two types of image - with the original size and with a lower resolution image of size 28x28. The result was shown for both normal kernel and cubic kernel. The parameters chosen for the second experiment is mentioned in Figure 7.

| Image | 128x128 | | | 28x28 | | |
|---|---|---|---|---|---|---|
| | | NCD | JTD | | NCD | JTD |
| Layer 1 | Filters | 30 | 40 | Filters | 20 | 40 |
| | Kernel Size | 5x5x5 | 5x5x5 | Kernel Size | 5x5x5 | 5x5x5 |
| | Subsampling Size | 5x5 | 5x5 | Subsampling Size | 5x5 | 5x5 |
| Layer 2 | Filters | 50 | 60 | Filters | 30 | 60 |
| | Kernel Size | 4x4 | 4x4 | Kernel Size | 2x2 | 2x2 |
| | Subsampling Size | 4x4 | 4x4 | Subsampling Size | 2x2 | 2x2 |
| Layer 3 | Filters | 70 | 80 | Filters | - | - |
| | Kernel Size | 2x2 | 2x2 | Kernel Size | - | - |
| | Subsampling Size | 2x2 | 2x2 | Subsampling Size | - | - |

Figure 7: Parameters chosen for the MCNN approach

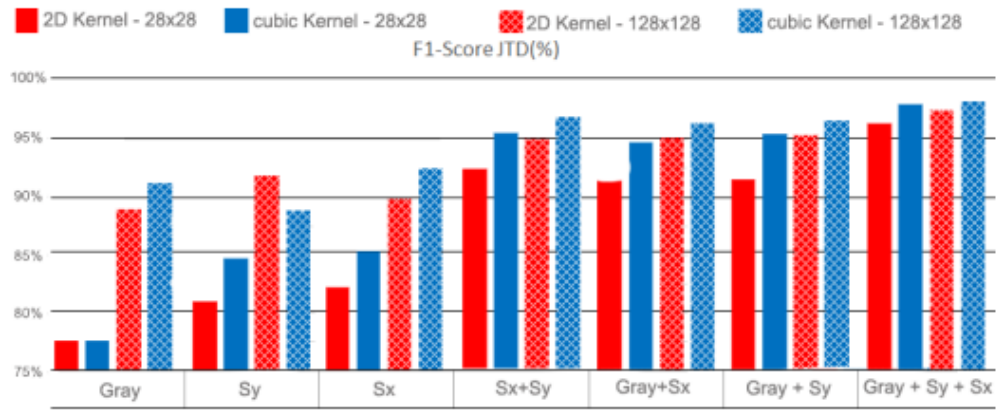
The experiment result from both experiments are depicted in Figure 8 and Figure 9.

- Both the experiments took considerate amount of variance regarding the size and shape of the hand posture. Though for EBGM, the variance regarding positions of hand was not considered and it is mentioned in [6] that there was error when the nodes were not in correct position due to big variations in hand shape. It can be procured that EBGM will not be able to produce correct result for different positions of hand, which the other approach handled. In real world scenario, it is not always possible to have position specific images

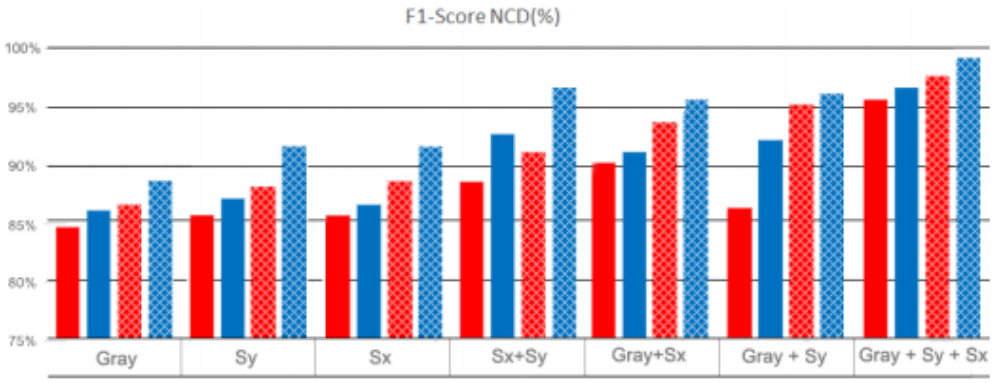
Recognition results

| Background | Number | Correct | % |
|---|---|---|---|
| Complex | 239 | 206 | 86.2 |
| Light | 210 | 198 | 94.3 |
| Dark | 208 | 194 | 93.3 |
| Total | 657 | 598 | 91.0 |

Figure 8: EBGM experiment result




(a) Results for the experiment with the JTD database.




(b) Results for the experiment with the NCD database.

Figure 9: MCNN experiment result

for gesture. Figure 9 shows that that for reduced image, normal CNN increased the performance when there were large amount of training data in NCD. For original image, normal CNN performance decreased when the variance of hand positions was introduced. For both images, introduction of cubic kernels increased the performance, though there was an exception in case of JTD for $S_y$ operator. The effect of cubic kernel was more prominent for the original image. The introduction of multiple channels improved the performance. However, with the increase in the dataset there was not much effect in performance. For the reduced image, the performance of MCNN decreased when position variance was included, whereas the behavior was opposite for original image. But, overall he effect of multiple channels was more promising for the reduced image than the original image.

- The JTD database consisted different type of backgrounds making it more similar to real world. The performance of EBGM is good for uniform light and dark backgrounds. However, the performance degraded for complex background. For light background, 94.3% gestures were recognized correctly and for dark background, 93.3% were correct. For complex background, the success rate was 86.2%. The overall recognition rate was 91%. There was no categorical segregation of performance with respect to background for the second approach. The overall recognition rate was 92% for small images and 94% for the original image, higher than EBGM.

- Although, the overall recognition rate is good for EBGM, Triesch has mentioned about the high complexity of the approach in [6]. It took several seconds to analyze a single image, whereas the MCNN approach was much faster. The mean recognition time with the three channel architecture was 0.125s for the original image. For the reduced image, it was even better 0.0035s. As expected, there was sufficient improvement in processing time for the reduced image, but the success rate was better for original image. The training time for reduced image was 200.32s and for the original image 1180.0s. All this was achieved by using a machine with an Intel Core 5i 2.67 Ghz processor, with 8GB of RAM.

- Though the EGM algorithm is flexible in terms of the distance between the nodes, it still does a template based matching. As mentioned by Pablo in [1], it restricts the solution to their database. But the second approach can be adapted for any database.

- For training phase, the EBGM approach needed manual intervention which can be tedious when the number of hand postures increases.

- Both approaches could identify the hand postures, even if there was some deformation in the image for example, occlusion of one finger.

- The images in the JTD were recorded in uniform lighting and for NCD too there was no variation of illuminance mentioned. In a real world scenario,

there will be difference in illuminance in an image which will have impact in both approaches.

# 5 Conclusion

We have discussed about two bio-inspired methods for analyzing gestures - Elastic Bunch Graph Matching and Multichannel Convolution Neural Network. The first method was an extension to Elastic Graph matching algorithm whereas the second was an adaptation of Convolution Neural Network. The first experiment was done a database, named JTD of 10 postures performed by 24 persons in 3 different background. The second experiment used the same database JTD and also tested its robustness against different positions of hand in the image.
With respect to time complexity and recognition rate, the MCNN approach performs better than EBGM. The processing time for reduced image was good for a real world application, but there is still scope of improvement in terms of recognition rate. The decrease in image size will also make the application more attractive with respect to resource consumption. Both approach needs to be tested for variance in illuminance. As mentioned by Pablo in [1], it will be more interesting to extend this model for dynamic gestures and multimodal application with addition of facial expression and audio data.

# References

[1] Pablo Barros, Sven Magg, Cornelius Weber, and Stefan Wermter. A multichannel convolutional neural network for hand posture recognition. In Stefan Wermter, Cornelius Weber, Wodzisaw Duch, Timo Honkela, Petia Koprinkova-Hristova, Sven Magg, Gnther Palm, and AlessandroE.P. Villa, editors, *Artificial Neural Networks and Machine Learning ICANN 2014*, volume 8681 of *Lecture Notes in Computer Science*, pages 403–410. Springer International Publishing, 2014.

[2] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689 – 694, 1997.

[3] Judson P Jones and Larry A Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1233–1258, 1987.

[4] Martin Lades, Jan C Vorbruggen, Joachim Buhmann, Jörg Lange, Christoph von der Malsburg, Rolf P Wurtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *Computers, IEEE Transactions on*, 42(3):300–311, 1993.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[6] Jochen Triesch and Christoph von der Malsburg. Classification of hand postures against complex backgrounds using elastic graph matching. *Image and Vision Computing*, 20(13):937–943, 2002.