*Proceedings of the International Conference on Machine Vision and Machine Learning*
*Prague, Czech Republic, August 14-15, 2014*
*Paper No. 59*

# Cascading Neural Networks for Upper-body Gesture Recognition

**Ra'eesah Mangera, Fred Senekal**
Council for Scientific and Industrial Research
Meiring Naudé Road, Brummeria, Pretoria, South Africa
rmangera@csir.co.za, fred.senekal@gmail.com

**Fred Nicolls**
University of Cape Town, Department of Electrical Engineering
Lovers Walk St, Cape Town, South Africa
fred.nicolls@uct.ac.za

***Abstract*** - Gesture recognition has many applications ranging from health care to entertainment. However for it to be a feasible method of human-computer interaction it is essential that only intentional movements are interpreted and that the system can work for a wide variety of users. To date very few systems have been tested for the real-world where users are inexperienced in gesture performance resulting in data which is noisier in terms of gesture-starts, gesture motion and gesture-ends. In addition, few systems have taken into consideration the dominant hand used when performing gestures. The work presented in this paper takes this into consideration by firstly selecting key-frames from a gesture sequence then cascading neural networks for left and right gesture classification. The first neural network determines which hand is being used for gesture performance and the second neural network then recognises the gesture. The performance of the system is tested using the VisApp2013 gesture dataset which consists of four left and right hand gestures. This dataset is unique in that the test gesture samples have been performed by untrained users to simulate a real-world environment. By key-frame selection and cascading neural networks the system accuracy improves from 79.8% to 95.6%.

***Keywords*****:** Gesture recognition, Depth sensor, Neural network.

## 1. Introduction

Since the development and release of the Microsoft Kinect camera in 2010, much research has been focussed on the use of gestures in human-computer interaction. The goal of this research is to enable a computer or machine to understand gestures being performed by a user. Applications of this technology range from sign-language recognition to musical creation.

Gestures can be defined as a sequence of motions performed by a human subject in a short space of time. They can be further divided into two groups: static and dynamic gestures. The human body is stationary for the duration of a static gesture, or pose whereas a dynamic gesture is a motion sequence over time. A dynamic gesture can be thought of as consisting of a sequence of poses.

Gestures can be performed using either the left or the right hand depending on the users' dominant hand or in some instances the meaning of the gesture may change. In such cases, the features extracted from the non-gesturing hand may skew the results as each user may not keep this hand in the same position resulting in poor classification accuracy.

Previous gesture recognition approaches have relied on the use of colour or grey-scale intensity images from an RGB camera, where each pixel in the image represents the intensity of the incoming light. Each pixel in the depth image indicates the calibrated distance between the camera and a point in the scene. Depth can be computed using principles from structured light: a known infra-red pattern is projected onto the scene, and the depth is then calculated using the distortion of the observed pattern.

Whilst ordinary RGB cameras are sensitive to illumination, depth images are not, making robustness easier to achieve.

In a typical depth based gesture recognition system the depth image and RGB images are used to extract features. These are then input to a classifier which may be a neural network, Hidden Markov Model or support vector machine amongst others. However, few systems account for untrained users performing the gestures, as would be the case in real-world applications.

We propose a method to first select key-frames in a gesture sequence to account for different gesture durations. Then we detect the gesturing hand by cascading neural networks to determine whether a gesture is being performed by the left or right side and use this result to select the correct features from key-frames as an input to a second neural network which will classify the gesture. The approach is illustrated in Fig. 1.

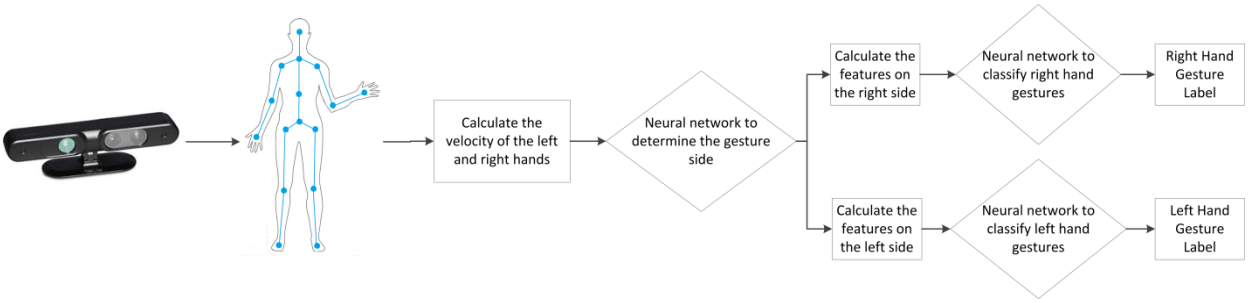


Fig. 1. Cascading neural networks to distinguish between left and right gestures.

The rest of this paper is ordered as follows: Section 2 describes related works in gesture recognition using depth images. Section 3 provides information on the dataset used for testing. Section 4 details features extracted from the skeletal data for both neural networks. Section 5 contains a description of the neural networks used for gesture classification including a brief overview of neural network theory, training the neural networks and the final network structure. Section 6 contrasts the results achieved without distinguishing between gestures with the proposed method. A conclusion and proposed future work are presented in Section 7.

## 2. Related Works

There are two main approaches employed in recognising gestures -- template-based and machine learning-based approaches. In template-based approaches, a gesture sequence is converted into a static signal, and extracted features are then compared to the pre-stored prototypes in order to classify a gesture sample. Machine learning approaches involve the use of pattern recognition algorithms such as hidden Markov models, random forests, support vector machines and neural networks, which are trained off-line and are then used in gesture recognition.

A comprehensive review of gesture recognition systems which use depth images for classification is provided by (Suarez and Murphy 2012). They review 37 papers in terms of the sensor being used, types of features extracted, classification techniques and applications.

Examples of gesture recognition systems which use depth sensors are those developed by (Lai, Konrad, and Ishwar 2012), (Jaemin et al. 2013), (Zafrulla et al. 2011), (Celebi et al. 2013) and (Bernstein, Lotocky, and Gallagher 2012).

(Lai, Konrad, and Ishwar 2012) use a Kinect camera to control a computer interface. Like the approach presented in this paper, joint position features are extracted from the skeleton model in each frame to form a feature vector. Using a Euclidean distance nearest neighbour classifier eight temporal gestures are recognised by (Lai, Konrad, and Ishwar 2012) with a classification accuracy of 97.25%. However, unlike the approach presented all gestures must be the same length and be temporally aligned. (Jaemin et al. 2013) also uses depth data for the purpose of gesture recognition. Joint angles are extracted from the Kinect skeleton model, and these are fed as inputs to a HMM (Hidden Markov Model) classifier.

A fourteen-gesture dataset is defined, and an accuracy of 81.8% is achieved. (Zafrulla et al. 2011) achieve a classification accuracy of 73.62% using Kinect data and a HMM classifier. They define a gesture set consisting of nineteen gestures for the purpose of sign language recognition. The features extracted from the skeleton are the vectors between joints, joint angle and the distance between the hands, resulting in a 20-dimensional feature vector. (Celebi et al. 2013) define a six-gesture dataset captured using a Kinect camera. Like the approach presented here, joint positions are extracted from each frame to form a feature vector. This approach however is template-based as dynamic time warping (DTW) is used to warp the feature vector to a reference feature vectors and produce a similarity matrix indicating how close the sample is to the reference gesture. They achieve an accuracy of 96.7% on six of the gestures. (Bernstein, Lotocky, and Gallagher 2012) captured a temporal dataset consisting of fifteen military gestures. They extract the amplitude and frequency of the gestures from the data and use these as features. A support vector machine is used to classify gesture samples. An accuracy of 98.5% is achieved. However, in this dataset all gestures are temporally aligned and of a set length. In addition, the users performing the gestures are trained.

The majority of the approaches above were tested on datasets that are temporally aligned i.e. the gestures are the same length and are performed by trained users. In contrast, the work presented in this paper is tested on the VisApp2013 dataset captured by (Celebi et al. 2013). In this dataset, the test samples are noisier in terms of the gesture length, gesture-start, gesture-end and joint movements. This paper proposes the use of cascading neural networks in order to distinguish between left and right gestures. Then using this result, features are selected from the gesturing hand to classify the gestures.

## 3. Dataset

The publicly available VisApp2013 dataset was used to evaluate the performance of the proposed methodology. It was captured using a Microsoft Kinect Sensor with 20 joint positions being recorded in text files. They defined a total of eight gestures namely "Left Hand Push Up," "Right Hand Push Up," "Left Hand Pull Down," "Right Hand Pull Down," "Left Hand Swipe Right," "Right Hand Swipe Left," "Left Hand Wave" and "Right Hand Wave". Sample frames from the dataset can be seen in Figure 2.

This dataset is challenging for several reasons:

- The sequences are of different lengths.
- Eight samples of each gesture class are performed by trained users, while the remaining 20 samples are performed by untrained users.
- The untrained user data is noisier in terms of gesture-start, gesture-end and joint movements.

The trained user data forms the training set whilst the noisy untrained user data is used for testing. This is to encode the re-configurability requirement specified by (Wachs et al. 2011) which states that a gesture recognition system should have enough flexibility to work for many different types of users without additional training.

## 4. Feature Extraction

Using the KinectSDK (Web-1) to interface with the Kinect camera, the joint positions of 20 joints in the human body can be tracked without requiring the user to wear any additional aids. The skeleton model generated by Kinect is a tree graph whose nodes correspond to the joints as shown in (Web-1). The skeleton tracker tracks the position of each joint in real-time at a frame rate of 30 fps. The skeleton model is ideal for feature extraction as it is robust to change in user height, weight, shape and environmental conditions such as lighting and background clutter.

A temporal gesture can be thought of as consisting of a sequence of poses; hence features are extracted from each frame and concatenated to form the final feature vector.

### 4.1 Distinguishing between Left and Right Gestures

A common problem in temporal gesture recognition is gesture spotting. Gesture spotting locates a gesture in a sequence of signals. An approach employed by (Cheng, Bian, and Tao 2013) uses the

acceleration signal to determine if a gesture is being performed. There are three distinct phases in a gesture signal: a high speed start, a continuous change in direction and an end in an almost steady position. Therefore in order to identify the gesturing side a similar approach is used. The velocity of the left and right hands are calculated as shown in equation (1).

$$v = \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2 + (z_t - z_{t-1})^2}}{(\frac{1}{30})} \quad (1)$$

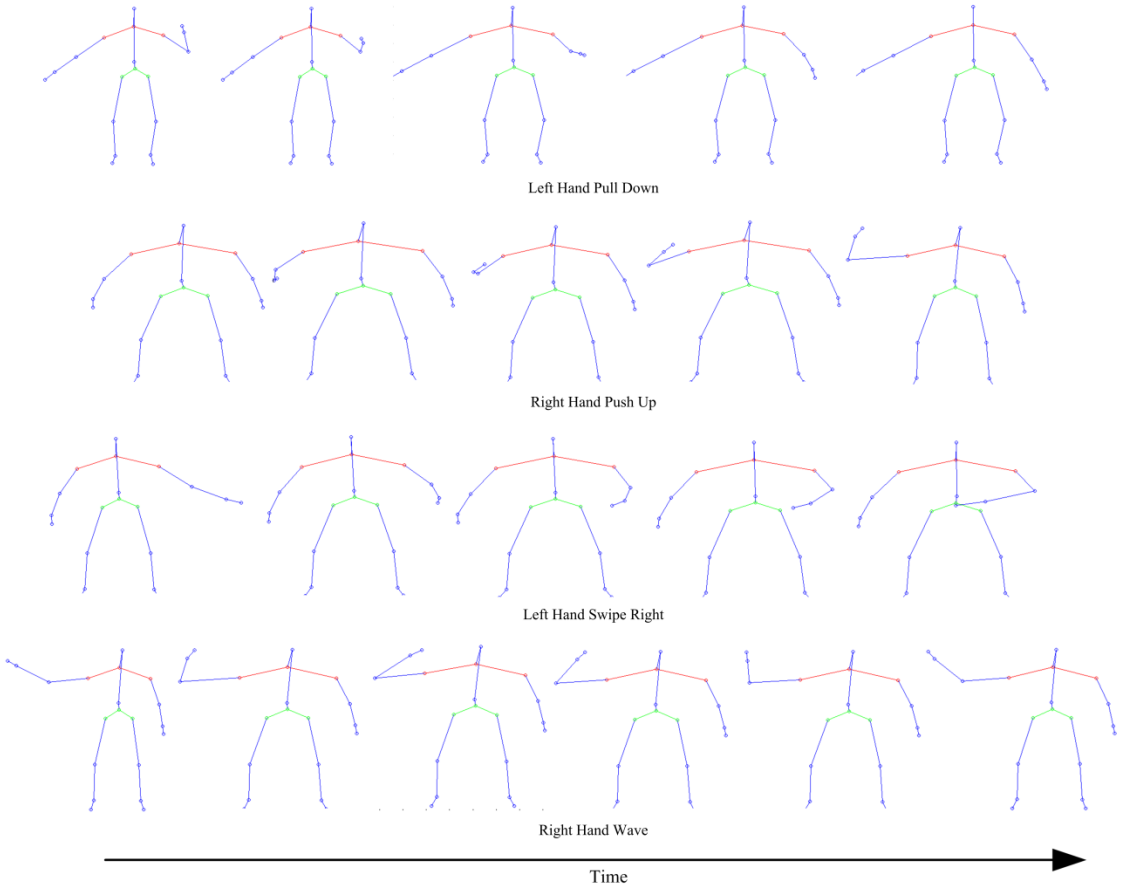


Fig. 2. Sample frames from the VisApp2013 gesture dataset.

Like the acceleration signal, the velocity signal of the gesturing hand will show a steady increase in velocity whilst that of the non-gesturing hand will be mostly stationary. As illustrated in Figure 3. It is evident that using the velocity profile of the left hand right hands, the gesturing and non-gesturing hands are distinct from one another.

It is observed that the velocity of the non-gesturing hand is steady throughout the gesture whereas the gesturing hand has some velocity trajectory. To capture this property, the tenth percentile and ninetieth percentile values of the velocities are found. The difference between these values is then used as a feature to distinguish between left and right gestures. We expect that the gesturing hand will show a much greater difference between these velocities compared to the non-gesturing hand. As is shown in Equation (2), where $i$ is the index of the ninetieth percentile and $j$ is the index of the tenth percentile. In addition, the difference between the ninetieth percentile and the value at the mid-point between the tenth and ninetieth percentile values is calculated. This is used to capture any sudden changes in gradient that may be present and is calculated as shown in equation (3).

$$side\ feature\ =\ v(i)\ -\ v(j) \tag{2}$$

$$x\ =\ v\left(\frac{i-j}{2}\right) \tag{3}$$

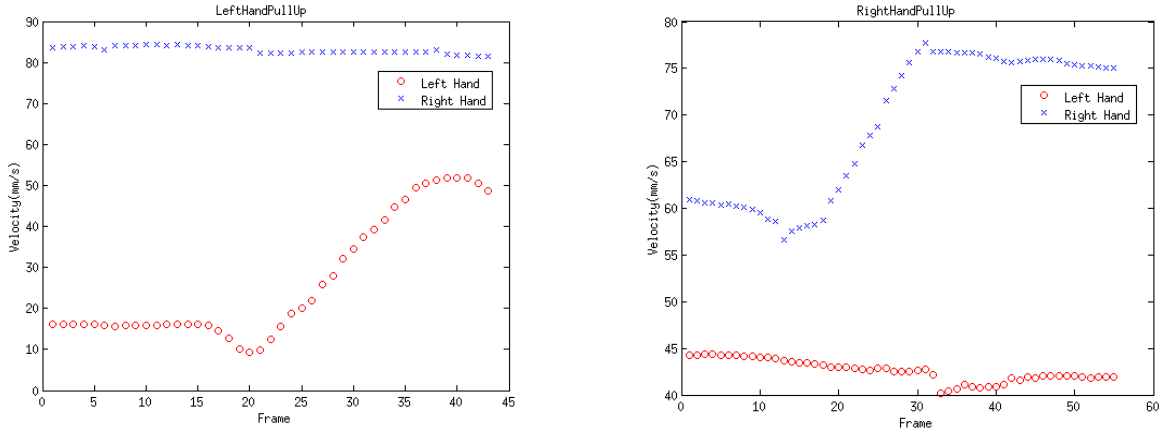


Fig. 3. The velocity profile of the left and the right hand when the same gesture is performed.

### 4.2. Feature Vector for Gesture Classification

Gestures may be differentiated by the Cartesian trajectory of the upper-body joints in space. Hence a feature vector consisting of the normalised $(x, y, z)$ position of either left and right hands and elbows over time was constructed. However, as mentioned in Section 3 the sequences in the dataset vary in length. To ensure that the feature vectors for each sample are the same length key-frames or poses are chosen. That is to ensure that the gesture samples are temporally aligned.

These key-frames are selected on the basis that there must be significant movement between consecutive frames. Selection of key-frames is done as follows:

- *Step 1*: The Euclidean distance between feature vectors in successive frames is calculated.
- *Step 2*: The distances are compared to a threshold. If the distance is below the threshold, the corresponding point is discarded. The threshold value was determined empirically to be 0.1.
- *Step 3*: Using the remaining points the minimum sequence length is found.
- *Step 4*: All sequences are linearly shortened to the minimum sequence length. For example, if the minimum sequence length is 10 and a sample had 20 key-frames every second frame would be discarded.

Hence the feature vector which is input to the neural network is of length $6 \times\ min\ length$. All the skeleton joint positions are normalised by the distance between the neck and torso joint to account for differences in user size and distance from the camera. Additionally, the skeleton is centred about the neck joint to account for horizontal shift.

## 5. Classification

### 5.1 Artificial Neural Networks

Artificial Neural Networks (ANN) is one of the most widely used architectures for machine learning applications. Developed in the 1980s to try and mimic the brain neural networks are now a state of the art technique used in a variety of applications. They rely on the principle of "divide and conquer" where a large, complex problem can be decomposed into simpler, interconnected elements known as neurons or nodes (Russell et al. 1995). The nodes are computational elements that receive inputs and process them to produce an output. The connections between nodes determine the flow of information from one node to another.

Similarly to a natural neuron, when the received signal is larger than a certain threshold the neuron is activated, and a signal is sent to the connected neurons.

Therefore, by adjusting the weights we can obtain the required output for a particular set of inputs. To adjust the weights of the neural network, the back-propagation algorithm is used. In back propagation, inputs are fed forward whilst errors are propagated backward, with the goal of minimising the error (Nabney 2002).

Three neural networks are connected in order to classify gestures. The first neural network distinguishes between left and right gestures and based on this result one of the remaining two networks then recognises the gestures.

### 5.2 Gesture Side Classification

A feed-forward multilayer neural network is used to distinguish between left and right gestures. The architecture of the network is shown in Figure 4.There are five hidden neurons. This value was chosen using the rule of thumb according to (Heaton 2005) as shown in Equation (4).

$$\#\ hidden\ neurons\ = \frac{2}{3}\#\ inputs + \#Labels \tag{4}$$

There are only two output labels as the purpose of this network is to distinguish between left and right gestures. The network is fully connected and the back-propagation learning algorithm (Heaton 2005) is used to train the network. For all neural networks implemented, the learning rate was set to 1.

### 5.3 Gesture Recognition

Two feed-forward multilayer neural networks are also used to classify gestures on the left and right side. The network architecture is similar to the one depicted in Figure 4 except that the input layer consists of 144 neurons, the hidden layer has 100 neurons and there are 4 outputs. The back-propagation algorithm is used to train each network.

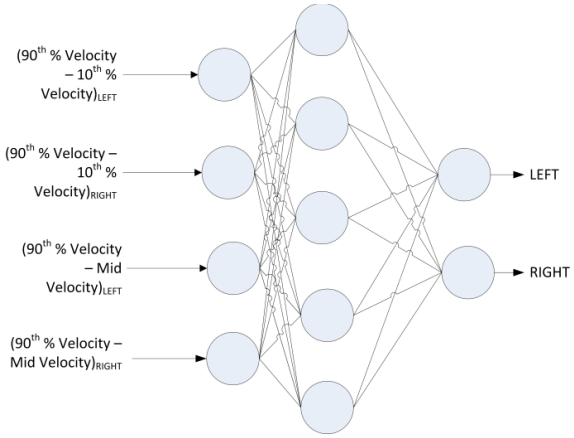


Fig. 4. Neural network structure for distinguishing between left and right gestures.

## 6. Results

As in (Celebi et al. 2013), the dataset is divided into a test set and training set. The training set consists of 8 samples for each gesture that are performed by trained users. The remaining 20 samples are used to test the performance of the gesture classification system. As discussed in Section 3, these gestures were performed by untrained users.

The average accuracy for side classification is 97.5%. All of the right gestures were correctly classified, and 5% of the left gestures were incorrectly classified.

The average accuracy obtained was 95.63%. This compares favourably to the results obtained by (Celebi et al. 2013) of 94.4%. Figure 5 shows a comparison between the results achieved by (Celebi et

al. 2013) and those presented in this paper. The green bar graph is the accuracy achieved by (Celebi et al. 2013) and the red represents the proposed approach. Performance for all gestures are quite similar for both the template-based approach employed by Celebi et al. (2013) and the machine learning approach presented here.

Table 1 shows the confusion matrix of the results obtained by cascading classifiers. The rows in the confusion matrix indicate the true class label whilst the columns represent the predicted class label (Kohavi and Provost 1998). Ideally, the confusion matrix should be a diagonal matrix representing 100% classification accuracy. Cells highlighted in green indicate the percentage of samples which were correctly classified, whilst cells highlighted in orange and yellow represent misclassifications. In particular, the cells highlighted in orange indicate the percentage of samples that were misclassified due to being classified as belonging to the incorrect side. These misclassifications are a result of an error in the first neural network that distinguishes between left and right gestures. It is observed that 20% of misclassifications can be attributed to errors in the first neural network. Hence improving the distinction between left and right gestures will further improve the results.

Table 1. Confusion matrix for the VisApp2013 dataset (Average Accuracy = 95.6%).

| | RH Push Up | LH Push Up | RH Pull Down | LH Pull Down | RH Swipe Left | LH Swipe Right | RH Wave | LH Wave |
|---|---|---|---|---|---|---|---|---|
| RH Push Up | 100 | | | | | | | |
| LH Push Up | | 100 | | | | | | |
| RH Pull Down | | | 90 | | 10 | | | |
| LH Pull Down | | | 5 | 90 | 5 | | | |
| RH Swipe Left | | | | | 100 | | | |
| LH Swipe Right | | | | | 5 | 95 | | |
| RH Wave | | | | | | | 100 | |
| LH Wave | | 5 | 5 | | | | | 90 |

These results are a significant improvement on the results obtained if the classifiers are not cascaded. i.e. if no distinction is made between left and right gestures the average accuracy obtained is 79.8%. Figure 5 shows a comparison of the results where the blue indicates no distinction and the red the proposed approach. It is evident that cascading classifiers to separate left and right hand gestures significantly improves the results. Not taking into account the non-gesturing hand allows the classifier only to consider features that are important for classification.

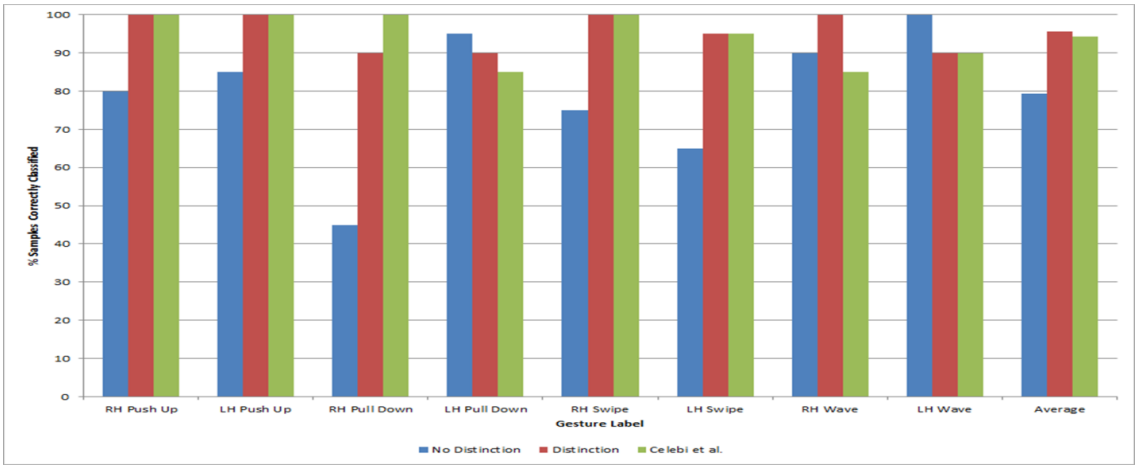


Fig. 5. A comparison of the results achieved by (Celebi et al. 2013) and the approach presented in this paper.

## 7. Conclusion

Gestures may be performed using either the left or right side of the body, depending on factors such as the user's dominant hand and whether the gestures have different meanings. By cascading neural networks, we are able to distinguish between gestures performed on the left and right sides. This means that only the features which are important for a particular gesture are used as an input to a neural network which recognises the gesture label resulting in a significant improvement of the results from 79.4% to 95.6%.

## References


Bernstein, G., Lotocky, N., Gallagher, D. (2012). Robot Recognition of Military Gestures CS 478 Term Project. Cornell University.

Celebi, S., Aydin, A. S., Temiz, T. T., & Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. *Computer Vision Theory and Applications. Visapp*.

Cheng, J., Bian, W., & Tao, D. (2013). Locally regularized sliced inverse regression based 3D hand gesture recognition on a dance robot. *Information Sciences*, *221*, 274-283.

Heaton, J. (2008). *Introduction to neural networks with Java*. Heaton Research, Inc..

Jaemin, L., Takimoto, H., Yamauchi, H., Kanazawa, A., & Mitsukura, Y. (2013, January). A robust gesture recognition based on depth data. In *Frontiers of Computer Vision,(FCV), 2013 19th Korea-Japan Joint Workshop on* (pp. 127-132). IEEE.

Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, *30*(2-3), 271-274.

Lai, K., Konrad, J., & Ishwar, P. (2012, April). A gesture-driven computer interface using Kinect. In *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on* (pp. 185-188). IEEE.

Nabney, I. (2002). *NETLAB: algorithms for pattern recognition*. Springer.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (1995).*Artificial intelligence: a modern approach* (Vol. 2). Englewood Cliffs: Prentice hall.

Suarez, J., & Murphy, R. R. (2012, September). Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE* (pp. 411-417). IEEE.

Wachs, J. P., Kölsch, M., Stern, H., & Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the ACM*, *54*(2), 60-71.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011, November). American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces* (pp. 279-286). ACM

Web sites:

Web-1: http://msdn.microsoft.com/en-us/library/jj131025.aspx, consulted 17 November 2013.