Adding Model Constraints to CNN for Top View Hand Pose Recognition in Range Images

Aditya Tewari [1,2], Frederic Grandidier [2], Bertram Taetz[1] and Didier Stricker[1]
[1]*Augmented Vision, German Research Centre for Artificial Intelligence(DFKI), 67663, Kaiserslautern, Germany*
[2]*IEE S.A., ZAE Weiergewan, L-5326, Contern, Luxembourg*

Keywords: CNN , Hand-Pose, Feature Transfer, Transfer Learning, Fine Tuning.

Abstract: A new dataset for hand-pose is introduced. The dataset includes the top view images of the palm by Time of Flight (ToF) camera. It is recorded in an experimental setting with twelve participants for six hand-poses. An evaluation on the dataset is carried out with a dedicated Convolutional Neural Network (CNN) architecture for Hand Pose Recognition (HPR). This architecture uses a model-layer. The small size model layer creates a funnel shape network which adds a priori knowledge and constrains the network by modelling the degree of freedom of the palm, such that it learns palm features. It is demonstrated that this network performs better than a similar network without the prior added. A two-phase learning scheme which allows training the model on full dataset even when the classification problem is confined to a subset of the classes is described. The best model performs at an accuracy of 92%. Finally, we show the feature transfer capability of the network and compare the extracted features from various networks and discuss usefulness for various applications.

1 INTRODUCTION

Hand-gesture is a simple sequence of hand or palm shapes. Hand-gestures are natural often involuntary actions. Hand Gesture Recognition (HGR) is popular in tasks like navigation, selection and manipulation in Human Computer Interactions (Buchmann et al., 2004). Detailed work has been done on identifying complex and precise hand movements for solutions in applications like surgical simulation and training systems (Liu et al., 2003). In contrast, simpler gestures have been used in computer controlled games, teleconferencing, robotics and augmented vision based solutions (Hasan and Kareem, 2012).

Non-vision based solutions to HGR includes Wii controllers, data gloves, 3D Accelerometer, electromyograph (EMG) (Schlömer et al., 2008) (Zhang et al., 2009). More recently the touchless vision based technique for HGR is considered a preferred choice.

Finite State Machines (FSM) were one of the earliest solutions for vision based HGR (Davis and Shah, 1994). Another branch of solution includes neural networks and Recurrent Neural Networks (RNN). Most often, both the FSM and RNN strategies hand-pose at each frame as important information (Chen et al., 2007),(Gupta et al., 2002). Thus, researchers have focused on the estimation of the hand-poses in frames of a sequence to solve an HGR problem.

Recently deep architectures of neural networks have been used for various computer vision problems and have produced strong results. With the emergence of CNN (LeCun et al., 1995) as a feasible learning algorithm, many experiments for HPR and HGR have been made with them. (Nagi et al., 2011) proposes a max pooling network to classify static gesture, with a classification accuracy of 96% on 6 gesture classes. The classical CNN has been employed on processed images by (Lin et al., 2014) on a dataset of seven gestures for seven persons achieving an accuracy of 95%. Similar network has been used for pose recovery by (Tompson et al., 2014), which combines the CNN with random forests.

Until recently one of the major challenges of pose and gesture recognition was the absence of datasets (Just and Marcel, 2009). Independently collected data makes the comparison of the reported results hard. Further, the solutions for HGR are often developed with a focus on application, the datasets are very distinct from each other. Front facing RGBD image datasets like the NYU (Tompson et al.,), (Xu and Cheng, 2013) and ICL (Tang et al., 2014) dataset for hand-pose and joint location are now available.

In applications with camera placed vertically above the scene constraints of the front facing hand

Tewari, A., Grandidier, F., Taetz, B. and Stricker, D.
Adding Model Constraints to CNN for Top View Hand Pose Recognition in Range Images.
DOI: 10.5220/0005660301700177
In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016)*, pages 170-177
ISBN: 978-989-758-173-1

may not work, in-car devices usually have a similar set-up (Zobl et al., 2003). The poses and the gestures in such cases are completed while palm points vertically downwards thus are visually different from the front facing depth images of the datasets identified earlier.

We have recorded a large hand-pose dataset with images captured with the Photonic Mixer device (PMD) technology (Xu et al., 2008). The PMD devices, unlike the more commonly used RGBD images have two channel image output. The dataset are unique in being a large ToF based datasets with a top-view recording. The dataset with more than 1000,000 samples allows experiments with convolutional neural network architectures and possibility of pre-training the network for feature transfer when porting the application to a new yet similar environment.

We train CNN based networks for pose classification from scratch using different preprocessing on the dataset with:

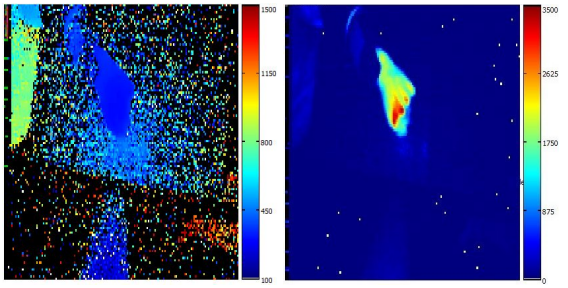
- Segmented raw images.
- Distance scaled images with amplitudes normalised by distances.
- Distance scaled binary images.

It is demonstrated that the best performing network has over 92% accuracy on the test set when the test are done on the binary images. The scaled amplitude images result in an accuracy of 84% and the raw image network result in an accuracy of 82%.

We have demonstrated the similarity in the features extracted by the convolutional layers of the network trained by separately pre-processed data. This observation is important because it allows the network to be a starting point for various applications which use input from such cameras for HPR or similar tasks. The primary contributions are as listed:

1. Preparing a dataset for top-view hand-pose with a ToF camera.
2. Solving the HPR by modelling a neural network to learn a low dimensional representation of hand.
3. Comparison of the input-strategies for better pose classification.
4. Demonstrating the usefulness of transfer learning for application based on depth data, where dataset large enough to train a deep network may not be available.

In section 2 we define the hand-poses. Further details of the recording set-up and dataset are shared in the same section. In section 3 we describe the architecture of the trained networks. We detail the experiments and results in section 4 and compare the performance of a network without the model-information discusses the performance of a larger network without the model-information architecture. The section 5 establishes the feature transfer property of CNN on the dataset. The work is concluded in 6.



(a) Raw Distance Data. (b) Raw Amplitude Data.

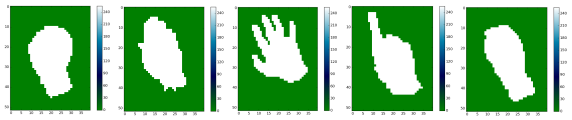
Figure 1: The ToF Two-channel Output.

2 HAND POSE DATASET

Wrist onwards the hand has high degree of freedom. A hand can thus form various signs and symbols, some of these pose are naturally used for communication. Of these possible symbols six poses are defined and recorded as the top view of the hand. Five of the poses are 'Fist', 'Flat', 'Joined', 'Pointing', and 'Spread'. The 'Fist' is a closed fist hand with palm facing downwards. Pose 'Flat' is when the palm is open with the four fingers joined together. 'Joined' is when the hand is conically shaped and points downwards with all fingers touching each other. 'Pointing' is the index finger pointing forward. Finally 'Spread' is an open palm with fingers spread apart. Further, a class of hand-pose in the places where the hand transitions from open to close are recorded. This class can have different uses. It can be identified as class of unintended poses or one that helps describing transitions of pose in a gesture.

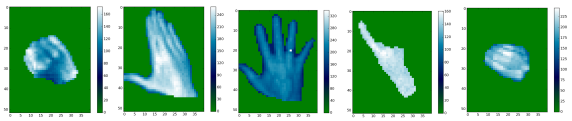
2.1 Data Recording and Segmentation

We record a large dataset of hand-pose using 3D Time-of-Flight (ToF) camera the 'pmd camboard nano'. The datapoints are 16 bit two channel images of dimension 120x165x2, Figure 1. The first channels of the matrix represent the amplitude of the reflected ray received by the camera and the second channel are the range values of the respective pixels, expressed in millimetres.



(a) Fist (b) Flat (c) Open (d) Point (e) Join

Figure 2: Sample of Binary Map of the Hand-poses.



(a) Fist (b) Flat (c) Open (d) Point (e) Join

Figure 3: Sample of Amplitude Normalised Map of the Hand-poses.

2.1.1 Recording Setup

The data is recorded within a cuboidal space with varying heights. The ToF camera is mounted vertically above the recording region. The furthest vertical range is marked by a table top. The height of the camera from the table varies between 400 and 800 mm. The closest vertical approach to the camera is marked at 150mm from it. While recording the participants were asked to wrap an absorbing cloth on their arms.

2.1.2 Recording

Twelve participants were recorded for pose and gestures. Each participant keeps his palm as one of the defined poses, and randomly but not abruptly moves the palm within the virtual cuboidal space. This is recorded for two minutes, for all six poses. Such recording of the data adds variances in depth and variances of hand orientation in the horizontal plane. The participants are also asked to rotate their palms to add the angular variances in the vertical plane.
The recorded participants have varying skin textures and palm sizes, some of the participants are recorded while wearing rings.

2.1.3 Segmentation

The absorbing cloth wrapped on the arms of the participants assists in hand segmentation by thresholding the amplitude channel of the image. The reflectance constraint does not entirely remove the background and thus the closest contour greater than a threshold area is chosen as the palm region. The segmented palm region is then converted into a binary image which is used as a mask for both channels. The resulting image is a two channel 16-bit image of the palm isolated from the environment. After segmentation The depth channel values for the background are set to a fixed maximum-depth and the amplitude values are set to 0. The basic processing after segmentation involves normalisation of the image.

The binary map and the normalised amplitude output for the five poses discussed earlier are shown in Figure 2 and Figure 3.

3 NETWORK AND TRAINING

The dataset is tested with various neural network architectures. The network discussed further is the one that provided the best results amongst various experiments. The same network is trained with the three pre-processing methods described later. Owing to the uniqueness of the database, classification networks are trained from scratch.

3.1 Training and Test Data

The dataset is divided into test and train data with one participant used as test and the remaining data used as the training data. Data augmentation is further achieved by horizontal-flipping of the image. This increases the data size and assists in better generalisation behaviour of the network. Of the six recorded hand pose classes, we classify five. The sixth class is recorded for identifying transitions while working with gestures. The training data has 107,131 data-points and the test data includes 11,800 data-points of five classes. The data is not equally distributed over the classes but the variation in data size amongst them is not large.

3.2 The Network Architectures

The hand-pose classification network is a sequential neural network. The selected architecture has four convolution layers followed by four fully connected layers which perform inner product. Each layer is connected to a ReLu (Rectified linear unit) which adds non-linearity to the network.
A convolution layer is connected to the input data. The output of the top three convolution layers is pooled by max-pooling strategy. The second and third pooling layers also sub-sample the output image of convolution layer by a factor of two. This pooling strategy allows different layers of the network to learn features at different scales. The convolution layers of the network are shown in the Figure 4. The fully connected layers and the output probability module of the network is shown in Figure 5.

The output layer for the network is a fully connected layer followed by a softmax function. The

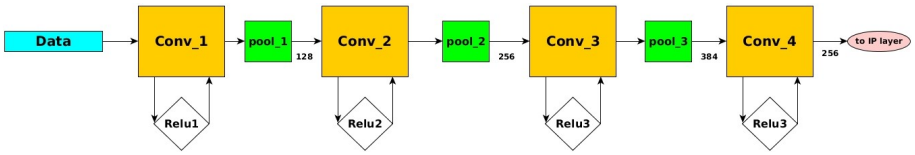


Figure 4: Input data and the Convolution layers.

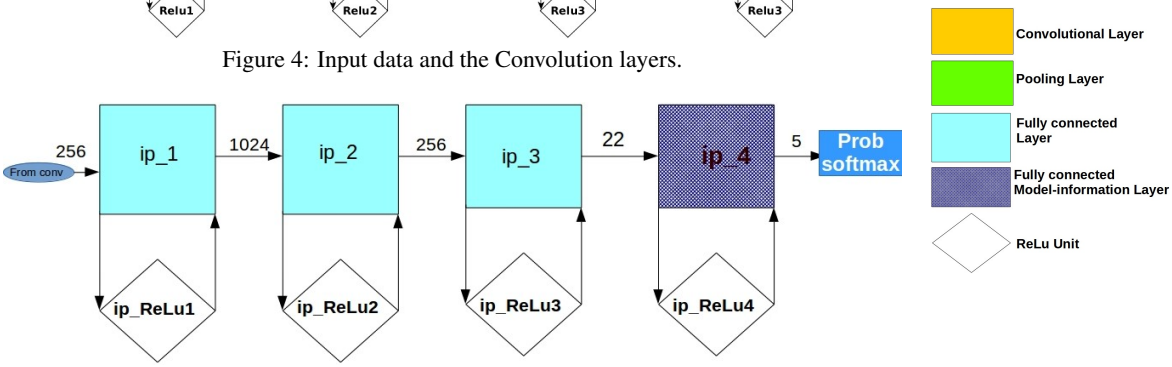


Figure 5: Fully Connected layers and Output Probabilities.

output of the softmax function is a probability vector associated to the five classes.

3.3 Model-information Into the Network

During the gradient descent the back-propagated error signal decays as it propagates through the layers. Weights of the layers which are closer to the output are influenced strongly by the error signal. Also, we know that the combined state of the locations which coincide with degree of freedom of the hand can indicate the pose. Robotic hand-wrist replacements have often used a twenty-two degree of freedom (Weir et al., 2008), we propose to add this information to the network by adding a twenty-two node layer as the penultimate layer of the network. This addition of fixed number derived from the hand shape before the output layer adds some model information to the HPR task. This creates a funnel shape in the network and forces the network to learn from a small dimensional representation of the input images.

3.4 Training Procedure

The network training is completed by propagating the logistic loss backward through the layers and completing a gradient descent optimisation. We first train the network on data from all six poses and then use the trained network for the initialisation of the five class classification. In the first phase the networks are initialised by xavier initialisation (Glorot and Bengio, 2010). While doing the first phase training we allow the data-points which where segmented improperly, but during the training of network in the second phase the improperly segmented data-points are removed. This is done because in the first training, we try to learn the features in the layers closer to the input. These layers are lesser affected by outlier in the ground-truth data. The second training focuses on modifying the fully connected layers which are closer to the output layers. Both the phase are trained by the stochastic gradient descent method of optimisation.

4 EXPERIMENTS AND RESULTS

The training of the neural network involves identifying a mapping of $\mathbf{I}$ to $\mathbf{P} \in |\mathbf{S}|$. Where, set $|\mathbf{S}|$ is the set of tested hand poses and $\mathbf{I}$ is the input image. It is possible to modify $\mathbf{I}$ before identifying the optimal network that provides the best mapping. As mentioned earlier the classification evaluation is conducted over the set of five classes 'Point', 'Join', 'Open', 'Fist' and 'Flat'.

All the channels of the input image are normalised over the dataset to [0,1]. This normalisation is done by recording the maximum amplitude value for valid pixels in the dataset. The maximum value for the depth defined during recording is used for depth channel normalisation. Normalisation is done while masking background pixels.

4.1 Normalised Raw Data

We conduct tests on the 2-channel image. The second channel provides the depth information which assists the network in assimilating the scale variations. The pre-processing for this experiment are sub-sampling the image by a factor of three using the mean approximation while masking the background pixels and mean subtraction for both channels independently.

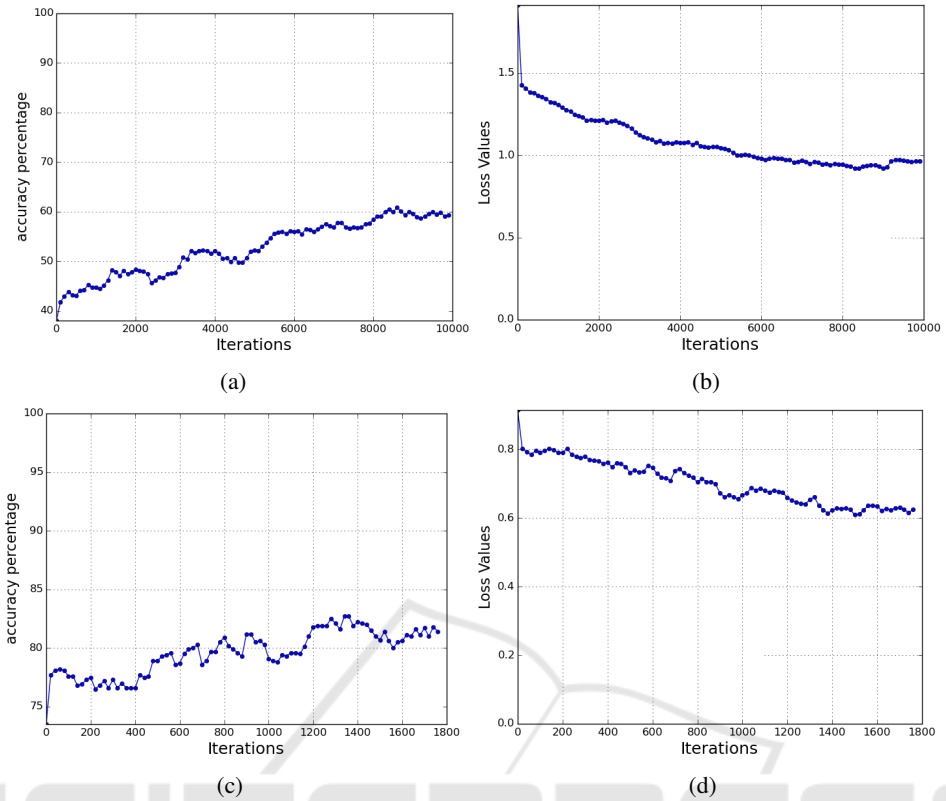


Figure 6: Training progression for training with two-channel images. 6(a): Accuracy stage 1. 6(b): Loss stage 1. 6(c): Accuracy stage 2. 6(d): Loss stage 2.

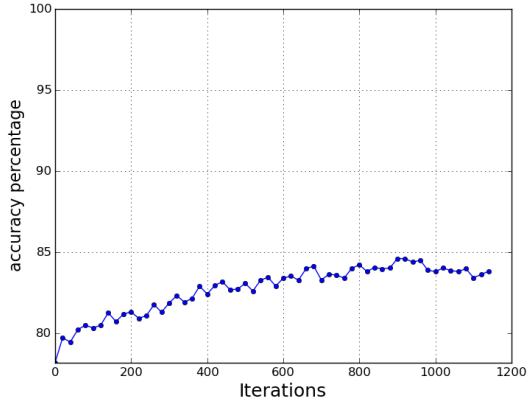
The larger size of the input data and the absence of explicit scaling makes model training complex. This is reflected in the training time and accuracy values. The training follows the two phase strategy described in section 3.4. The training progression for the first phase is shown in Figure 6(a) and Figure 6(b). The test accuracy during the second training phase remains below 83%, Figure 6(c) and Figure 6(d).

4.2 Amplitude Normalised Images

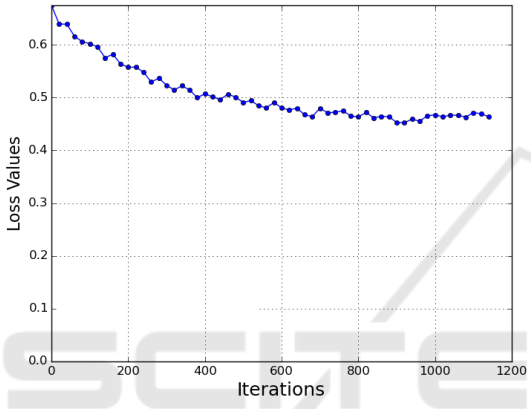
To remove the scale factor from the data the image is projected on a plane at a fixed distance from the camera. We then normalise the amplitude value by the squared distance. This is done because apart from the physical attributes of the scene the intensities of the pixels in a ToF camera output are dependent on the squared distance of the pixel from the camera. After scaling the image with distance, the contribution of distance to the intensity channel adds complexity to the data without contributing additional information for pose identification. This forces invariance to different distances on the learnt network. Further, we calculate a mean amplitude image for the dataset and subtract it from the input to the network both in the test and train phase. The amplitude model is trained in two stages as described earlier in section 3.4. Figure 7(a) and Figure 7(b) shows the test accuracy and loss improvement in the second phase. It is noticeable that because of the pre-training the initial test accuracy of the second phase training is over 70%. This allows quick training of networks when the number of classes or the environment in which data has been acquired changes. We find that after 6000 batch iterations in the second phase the test accuracy remains around 85%. The incorporation of scaling information in the amplitude data improves the performance of the network as well as reduces the training time for the network to two-third as compared to the training time for the two-channel unscaled images.

4.3 Binary Images

The experiments are also carried on the binary images extracted from the same dataset. The pre-processing steps in the experiment are identical to the amplitude normalised images, the difference being that the pre-processing output is binarised. The mean subtraction step is skipped while training and testing the network with the binary images.



(a)



(b)

Figure 7: Training progression for training with amplitude images. 7(a): Accuracy stage 2. 7(b): Loss stage 2.

(a)

(b)

Figure 8: Training progression for training with binary images. 8(a): Accuracy stage 2. 8(b): Loss stage 2.

The best classification performance on the model approached 92%. The better performance of the binary images can be attributed to the binary nature of the data. The intensity values of the pixels of a ToF camera depend on the reflectance of the surfaces and the incidence angle of the active light. These factors contribute ambiguous information to the amplitude channel which could explain the better performance of the binary data. It was also observed that the 'Fist' and 'Join' class were often misclassified, the similarity in the captured masks of two classes explain this observation. The training progression for the binary data is shown in Figure 8(a) and Figure 8(b).

4.4 Without Model-information Layer

A network in which the 22-node layer is replaced with a larger 256-node layer is trained on the binary images. It was observed that the loss progression of the network was smoother then the network with the model-information layer but after equal number of batch-iterations the overall test accuracy was found to be below 88%. Thus a smaller penultimate layer which acts as a funnel to force constraints helps in better classification performance. Figure 9.

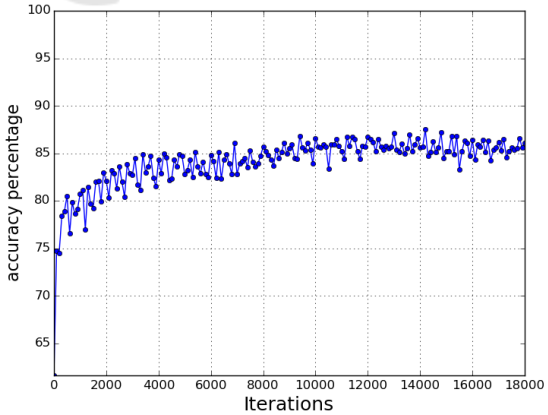


Figure 9: Accuracy progression for a network in which 22-Node layer is replaced by a larger layer.

5 FEATURE TRANSFER

The two stage learning uses the ability of a convolutional neural network to transfer learnt features over various problems. It is observed that this similar procedure can also be used with networks which were trained on data with distinct pre-processing. A network trained on a certain kind of data can be used to initialise a training with a distinct data set. Transferring the weights from one network to the other in this process assists in better initialisation of the network parameters. The following experiments describes the transfer learning process on the binary and amplitude image dataset for hand-pose.

The output of the convolutional and pooling layer are feature maps. These features are calculated by the convolution filters learnt during training. As the data moves through these layers the output feature maps resemble features calculated on increasing scales.

The applicability of the feature transfer is demonstrated by feeding the same amplitude normalised image into the second stage networks trained in section 4.2 and 4.3. These networks were trained on the amplitude and binary images respectively. When the outputs of each layer from the networks were compared, it was found that the mean difference calculated in the euclidean sense for the first pooling layer output was of the order of 10^{-6}, which increases to 10^{-3} for the second pooling layer and 10^{-2} for the output convolution layer. The difference is considerably larger for the fully connected layers which are closer to the output. This indicates that the filter weights learnt by the network for the first cases are general and can be reused for the training, thus reducing training time and training data requirement.

The property of feature transfer was tested by employing the model trained with binary images obtained in last section to directly test the amplitude normalised dataset. In this experiment a test accuracy of 75.4% was achieved. When the same model is used as an initialisation for the second phase training of the amplitude normalised images the accuracy of the amplitude images cross 80% within 500 batch-iteration, Figure. 10. During training the weights of the connections closer to the network output change faster than the weights closer to the input, because the back-propogated gradient diminishes by the time it reaches the layers away from the output. This can be inferred from the observed changes in the outputs of convolution layers, which are closer to the input and the fully connected layers closer to the output.

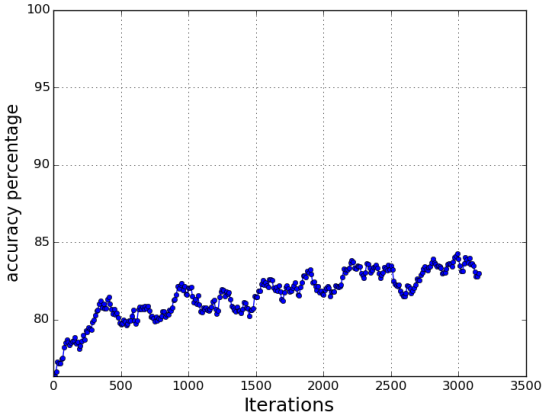


Figure 10: Model Learnt with Binary Image Fine Tuned with Amplitude Images.

6 CONCLUSION

This work presented a new dataset for ToF images of top view of hands collected for HPR. It includes six different hand-poses. As the top view images with palm pointing downwards are distinctly different from front facing hands, a new architecture for CNN is conceived.

The prior based CNN which forces the constraints of hand shape for learning pose classification is proposed and evaluated. The network achieved a performance of 92% classification accuracy on a 5-class classification problem. This work uses a two phase learning strategy which allows data uses from the entire dataset while solving problem which is restricted to a subset of the dataset. Feature transfer in distinct datasets and its utility in the present problem was demonstrated by using a network with binary images to train on the amplitude normalised images.

We test the network on three kinds of data extracted from the dataset. The normalised and scaled amplitude data, the scaled binary mask and the unscaled, normalised two channel image. We found that the pose identification performance in case of the binary one channel images was by far the best. Although, the CNN can capture the scale variances but the distance scaling of the input images improves the detection rate and also improves the speed of learning. It is also observed that the model can be easily modified when the classification problem changes, it is demonstrated that trained models can be used for similar classification problems. Important observation on the similarity of the trained filter weights for the network trained on data-set with diverse pre-processing is demonstrated. This observation forms a basis for deploying the model on problems where the nature of

the data-set changes because of the change in environment.

The pose data-set will be further experimented and evaluated. The features extracted from the trained CNN shall be used for solving the HGR problem.

ACKNOWLEDGEMENT

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

REFERENCES

Buchmann, V., Violich, S., Billinghurst, M., and Cockburn, A. (2004). Fingartips: gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 212–221. ACM.

Chen, Q., Georganas, N. D., and Petriu, E. M. (2007). Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. IEEE.

Davis, J. and Shah, M. (1994). Recognizing hand gestures. In *Computer VisionECCV'94*, pages 331–340. Springer.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.

Gupta, N., Mittal, P., Roy, S. D., Chaudhury, S., and Banerjee, S. (2002). Developing a gesture-based interface. *Journal of the Institution of Electronics and Telecommunication Engineers*, 48(3):237–244.

Hasan, H. S. and Kareem, S. A. (2012). Human computer interaction for vision based hand gesture recognition: A survey. In *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, pages 55–60. IEEE.

Just, A. and Marcel, S. (2009). A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. *Computer Vision and Image Understanding*, 113(4):532–543.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., et al. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276.

Lin, H.-I., Hsu, M.-H., and Chen, W.-K. (2014). Human hand gesture recognition using a convolution neural network. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 1038–1043. IEEE.

Liu, A., Tendick, F., Cleary, K., and Kaufmann, C. (2003). A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614.

Nagi, J., Ducatelle, F., Di Caro, G., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., Gambardella, L. M., et al. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 342–347. IEEE.

Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture recognition with a wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14. ACM.

Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3d articulated hand posture. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3786–3793. IEEE.

Tompson, J., Stein, M., Lecun, Y., and Perlin, K. Real-time continuous pose recovery of human hands using convolutional networks, journal = ACM Transactions on Graphics, year = 2014, month = August, volume = 33.

Tompson, J., Stein, M., Lecun, Y., and Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 33(5):169:1–169:10.

Weir, R., Mitchell, M., Clark, S., Puchhammer, G., Haslinger, M., Grausenburger, R., Kumar, N., Hofbauer, R., Kushnigg, P., Cornelius, V., et al. (2008). The intrinsic hand–a 22 degree-of-freedom artificial hand-wrist replacement. Myoelectric Symposium.

Xu, C. and Cheng, L. (2013). Efficient hand pose estimation from a single depth image. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3456–3462. IEEE.

Xu, Z., Möller, T., Kraft, H., Frey, J., and Albrecht, M. (2008). Photonic mixer device. US Patent 7,361,883.

Zhang, X., Chen, X., Wang, W.-h., Yang, J.-h., Lantz, V., and Wang, K.-q. (2009). Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 401–406, New York, NY, USA. ACM.

Zobl, M., Geiger, M., Schuller, B., Lang, M., and Rigoll, G. (2003). A real-time system for hand gesture controlled operation of in-car devices. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 3, pages III–541–4 vol.3.