



## KEYPOINT DETECTOR RETRAINING TECHNIQUES FOR THE COMMUNICATION SYSTEM OF SIGN LANGUAGE SPEAKERS

Guram CHAGANAVA , David KAKULIA 

Department of electric and electronic engineering, Faculty of exact and natural sciences, Ivane Javakhishvili Tbilisi State University, Tbilisi, Georgia

### ABSTRACT

The study described in this article examines the approaches of retraining of the deep learning model for hand palm keypoint detection in images. This is one of the studies conducted to create an innovative communication system for sign language speakers. The target of the given study is to find an optimal technique of retraining for increasing the degree of the keypoint detector generalization. So, it must be able to accurately detect keypoints in images it has not seen during training. It will make the communication system usable in real-life conditions.

In the article, there are reviewed three approaches of retraining: Retraining in series, retraining using 'united' dataset and retraining using mixed datasets. Experiments were conducted to test the effectiveness of each of them. The paper presents the results of the experiments and a relatively optimal method selected among them.

**Keywords:** Sign language. Communication system. Keypoint detection. Retraining

### 1. INTRODUCTION

Sign language is the primary means of communication for people with partially or completely limited hearing and/or speech skills. Literacy education is associated with specific challenges for people with such disabilities. For this reason, sign language is also their main means of distance communication. The tool for providing such communication is a video call. The video stream must be uninterrupted, especially when it comes to sign language. In this case, the main information is not the sound, but the image. Stable access to communication networks is essential for uninterrupted video calls. Sometimes this is associated with certain financial costs. The communication system offered by us is an alternative means of a video call, which can provide uninterrupted remote communication using relatively few resources. The essence of the operation of this system is that it will exchange not a complete image, but only important information between the speaker's devices. Important information refers to what the speaker wanted to say to the interlocutor. Since the speaker uses signs, it is necessary to obtain information about the signs and transmit them to the interlocutor. The interlocutor will see a three-dimensional animated model - an avatar on the screen, which, according to the information received, will repeat the signs depicted by the interlocutor.

The communication system operation includes three main stages: Getting information about signs from the image, information transmission to the interlocutor and displaying the avatar in the appropriate position. To obtain information about signs, we examined two approaches - Sign Language Recognition (SLR) and keypoint Detection. Studies have shown that using the keypoint detection method would be optimal for this task. This method involves determining the locations of keypoints in the image. The keypoints are certain points of the human body, knowledge of their locations allows us to restore the position of the human body. Hands are the most important parts for depicting signs, so we decided to detect only hand keypoints at the initial stage of the study. For simplicity, at first, we examined the keypoint detection of right-hand palm.

Testing of the available keypoint detectors has shown that their integration into the project would be ineffective. A particularly high detection accuracy was obtained, albeit at a rather low speed. We decided to create our detector, which would detect keypoints not from any, but only from the category of photos that the camera will see in the process of communication. These photos show one person standing in front of the camera and signing. According to our assumption, a relatively simple algorithm would be sufficient to accurately detect the keypoints. Advantage of creating task specific keypoint detector is discussed in [1] A simple algorithm requires a relatively short time for detection.

At first, we have created the dataset for building a keypoint detector. To increase the number of photos, we performed two types of augmentation: positional and lighting. The resulting dataset has been used to train deep learning model – Mobilenet [2, 3]. As a result, the accuracy was more than 90%, and the detection time was several tens of milliseconds. At this speed, the system will be able to process more than 10 photos per second. This result can be considered sufficient for our task.

It should be noted that the dataset we created contained photographs of one specific person with a specific background. For the detector to be usable in real-life conditions, it must be able to detect keypoints of any person in various conditions. For this, the degree of its generalization must be high. To do this, we must train the detector using as much big and diverse dataset as possible. The dataset should include photographs of different people that will be taken in different conditions. Therefore, we decided to create new datasets containing photographs of different people. That was to be decided how should we use datasets in the training process, should we use them consistently or combine them. We decided to examine in which case we get relatively high accuracy of detection. Choosing the optimal retraining method is the main goal of this study.

## 2. KEYPOINT DETECTOR RETRAINING

Training a neural network implies the selection of its parameters so that it can perform a specific task. Sometimes we may need to retrain the network as needed. At this time, all or some of the parameters of the network are changed so that it can learn a new problem similar to the one already studied. It is important not to change the parameters in this process so that the network does not "forget" how to solve the previously studied problem. An example of retraining is transfer learning. It involves training a neural network trained on some task to solve a similar task. For example, a classifier knows how to distinguish 10 different animals from each other, and we want to teach it to identify an unknown animal. We can conduct a retraining process on a given network with a dataset of an unknown animal. As a result, it will learn to recognize the 11th animal.

We can say that our task is different from the task of transfer learning. In particular, in this case, it is not the task to study a similar problem but to improve the quality of the generalization. We have to teach the network the same task but using new and diverse datasets. Its goal is to create a detector which will be able to detect keypoints in photographs that it has never seen before.

Since the network architecture does not change during retraining, the detection time remains the same. Therefore, the main subject of our observations during the experiments is the accuracy of detection.

Let's denote the dataset used for the first training of deep learning model as  $\{X_0, Y_0\}$ .  $X_0$  and  $Y_0$  are sets of images and corresponding targets.

$$\{X_0, Y_0\} = \{(x_0^1, y_0^1) \cup (x_0^2, y_0^2) \cup \dots \cup (x_0^n, y_0^n)\} = \bigcup_{i=1}^n (x_0^i, y_0^i) \quad (1)$$

The dataset is constructed from pairs  $(x_0^i, y_0^i)$ , where  $x_0^i$  is the corresponding matrix of the photo with index  $i$ , and  $y_0^i$  - an array of coordinates of keypoints in the photo with index  $i$ . We have  $n$  samples in total. After initial training, we built new datasets:  $\{X_1, Y_1\}, \{X_2, Y_2\}, \dots$ . Using these datasets, we have to retrain initially trained keypoint detector. 3 different approaches of retraining are discussed below.

## 2.1. Retraining in Series

One way to retrain is to train once trained network sequentially using new datasets. In particular, the detector is trained using the set  $\{X_1, Y_1\}$  then using  $\{X_2, Y_2\}$  and so on. In this case, at the  $i$ -th stage of training, the network is trained using the dataset  $\{X_i, Y_i\}$ .

One of the most important hyperparameters of a neural network is a Learning Rate (LR). It determines the rate of changing the parameters of a network during the training process. If LR is selected incorrectly during retraining, the network may not be able to learn from the new dataset, or it may 'forget' what is learned from the previous set. This problem is especially actual in the case of retraining in series. In this case, at each stage of training, the network parameters are updated according to datasets. These datasets contain images of different people. The greater the difference between datasets, the more changes need to be made to the network parameters for the detector to be able to correctly detect keypoint in new types of photos. A drastic change in the parameters of the network can lead to the fact that it will not be able to perform accurate keypoint detection in the images of already studied datasets. It is therefore important to conduct experiments of retraining in series for different values of learning rates and find its optimal value.

## 2.2. Retraining using United Dataset

This approach involves combining all datasets into one set. The network is trained using dataset  $\{X, Y\}$ , which can be represented as follows:

$$\{X, Y\} = \bigcup_{i=0}^n \{X_i, Y_i\} \quad (2)$$

After combining datasets, it must be shuffled which means a random shifting of the order of samples.

## 2.3. Retraining using Mixed Datasets

In this case, we first combine the datasets as it was in the previous case.

$$\{X', Y'\} = \bigcup_{i=0}^n \{X_i, Y_i\} \quad (3)$$

We shuffle the obtained  $\{X', Y'\}$  dataset. We Divide the resulting dataset into  $m$  sub-datasets:  $\{X'_1, Y'_1\}$ ,  $\{X'_2, Y'_2\}$  ...  $\{X'_m, Y'_m\}$ . However,  $m$  does not have to be equal to  $n$ . It is also not necessary to divide the dataset  $\{X', Y'\}$  into equal sub-datasets. The sub-datasets can be of any size. The resulting sub-datasets will contain samples randomly selected from the original datasets.

$$\{X'_j, Y'_j\} = \bigcup (x_j^k, y_j^k) \quad (4)$$

$\{X'_j, Y'_j\}$  is the obtained sub-dataset with index  $j$ , where  $j = 1 \dots m$ .  $(x_j^k, y_j^k)$  is the sample of the given sub-dataset with index  $k$ . It is taken randomly from one of the initially created datasets.

$$(x_j^k, y_j^k) \in \{X_{rand(i)}, Y_{rand(i)}\} \quad (5)$$

## 3. RESULTS

The conducted experiment involves testing the three approaches of retraining described above, including determining the optimal one. The datasets built by us were used during the experiment: The dataset composed of the retrieved photos, same as  $\{X_0, Y_0\}$ , let's call it 'dataset 0', and the dataset of photos taken by us  $\{X_1, Y_1\}$ , let's call it "dataset 1".

We divided all the datasets created during the experiments into three parts: training, validation and test sets. Training and validation sets were used for training. After completing the training, the accuracy of the detector was measured on the test set.

Percentage of Detected Joints (PDJ) [4] was used for the detection accuracy evaluation. As a rule, its calculation requires the value of the length of diagonal of the human bounding box in the image. However, since our task is to detect keypoints of a hand palm, we replaced this value with the hand palm bounding box diagonal.

The detection accuracy was measured for each sample of the test set and then their average value was found. The average detection accuracy was calculated on all datasets, including those that have not yet been used in the training process. The reason for it is to observe the change in the average detection accuracy on all datasets after each retraining.

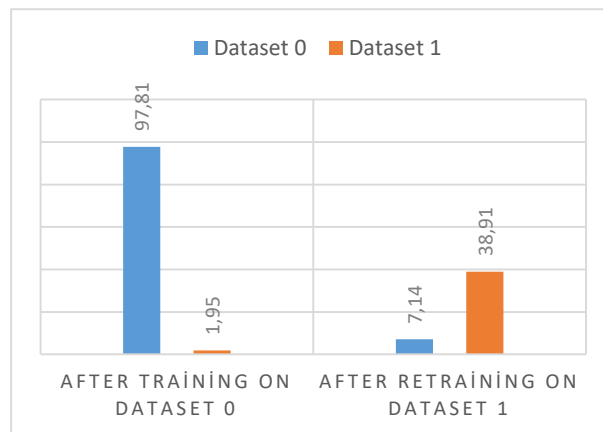
The training was conducted on deep learning model-MobileNet. Initial training was carried out on four different types of MobileNet. According to the accuracy and speed of the detection, the best result was obtained on MobileNet version 2, with a depth multiplier of 0.5. Experiments presented in this article have been conducted on this model. Adam optimization algorithm was used as the optimizer function and Mean Squared Error (MSE) as the loss function. During the training batch size was set to 4. As for the learning rate, we change its value according to the value of the loss function of the validation set. If the value of the validation loss function has not improved (i.e. has not decreased) over the last 5 epochs, we decrease the learning rate by 5 times until it reaches the minimum value. If the value of the validation loss function has not improved (i.e. has not decreased) over the last 5 epochs, we decrease the learning rate by 5 times until it reaches its minimum value set by us. We retrain the deep learning model for 50 epochs, but if the validation loss function does not decrease over the last 10 epochs, we stop training prematurely.

### 3.1. Results of Retraining in Series

In this case, we retrained the originally trained network using ‘dataset 1’. For the initial value of the LR, we chose the final value obtained during the initial training. This was 1e-6. The initial and final values of the learning rate are given in Table 1.

**Table 1.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘dataset 1’	1e-6	1e-6



**Figure 1.** Results of retraining in series on ‘dataset 0’ and ‘dataset 1’. The average accuracy of keypoint detection measured on the test set in PDJ percent.

As seen from the diagram, after initial training, the network was able to detect keypoints in ‘dataset 0’ with a fairly high accuracy, which it could not do in case of ‘dataset 1’. The reason for it is that during training, the network has seen only samples of ‘dataset 0’. As a result of retraining using ‘dataset 1’, the detection accuracy of the samples of ‘dataset 1’ increased slightly, while the detection accuracy of ‘dataset 0’ dropped sharply. Thus, the detector was unable to gain new knowledge, even forgetting what had already been learned. The possible reason for this is a very small learning rate. When the LR is small, the network parameters change slightly during the training. Because of this, the network cannot find the optimal values of parameters for a new task. During the training, network parameters are slightly but still updated according to a new dataset. Since samples from ‘dataset 1’ are significantly different from samples from ‘dataset 0’, the change in network parameters is such that it can no longer perform the task examined the previous time.



**Figure 2.** Keypoint detection results after first training in samples from dataset 0 (left\*) and dataset 1 (right). Detection accuracy in PDJ percent: left – 95,24%, right – 0%. \*(Material courtesy of Dr. Bill Vicars and [www.lifeprint.com](http://www.lifeprint.com))

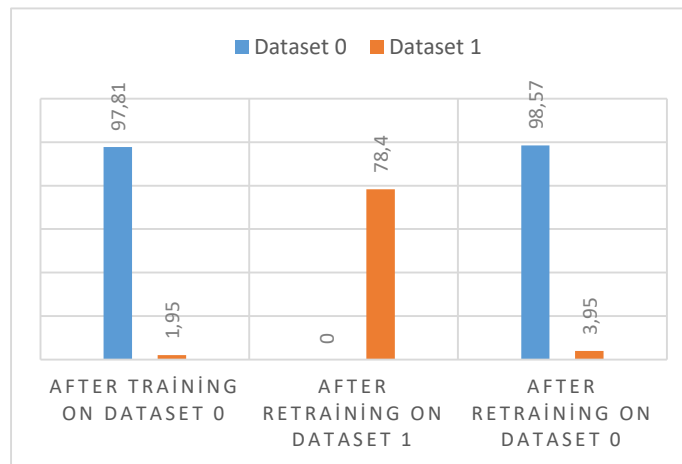


**Figure 3.** Keypoint detection results after retraining on ‘dataset 1’ in samples from dataset 0 (left\*) and dataset 1 (right). Detection accuracy in PDJ percent: left – 4,76%, right – 19,05%. \*(Material courtesy of Dr. Bill Vicars and [www.lifeprint.com](http://www.lifeprint.com))

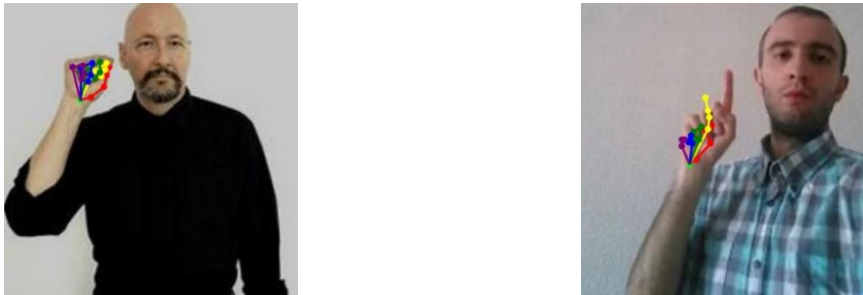
On the next stage of the experiment, 1e-3 was chosen as the initial value of the learning rate. We assumed that in the case of the relatively high initial value of LR, the network would be able to acquire new knowledge. After retraining on ‘dataset 1’, the network was retrained on ‘dataset 0’ again. We reduced the value of the initial training rate by 10 times for each new retraining. The purpose of this is the network not to forget the knowledge gained in the past.

**Table 2.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘dataset 1’	1e-3	4e-5
‘dataset 0’	1e-4	1e-6



**Figure 4.** Results of retraining in series on ‘dataset 0’, ‘dataset 1’ and ‘dataset 0’. The average accuracy of keypoint detection measured on the test set in PDJ percent.



**Figure 5.** Keypoint detection results after retraining on ‘dataset 1’ and ‘dataset 0’ in samples from dataset 0 (left\*) and dataset 1 (right). Detection accuracy in PDJ percent: left – 95,24%, right – 4,76%.  
 \*(Material courtesy of Dr. Bill Vicars and [www.lifeprint.com](http://www.lifeprint.com))

As expected, after each retraining the network was able to learn from new datasets. However, as a result of each retraining, knowledge gained on previous training was partially or completely lost. The main reason for this is the difference between the datasets. The main reason for this is the significant difference between the datasets.

### 3.2. Results of Retraining using United Dataset

When testing this method, ‘dataset 0’ and ‘dataset 1’ were combined and the resulting dataset was named ‘united’. We conducted the training process using a ‘united’ dataset with two approaches: the first involved retraining the initially trained network, and the second involved training the deep learning model directly using the ‘united’ dataset. The trained model has been retrained for two different initial values of LR. They are referred as case 1 and case 2. After each retraining, the detector accuracy was evaluated on all datasets. The results were compared to the results of the initial training.

**Table 3.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘united’ (case 1)	1e-3	4e-5
‘united’ (case 2)	1e-4	2e-5



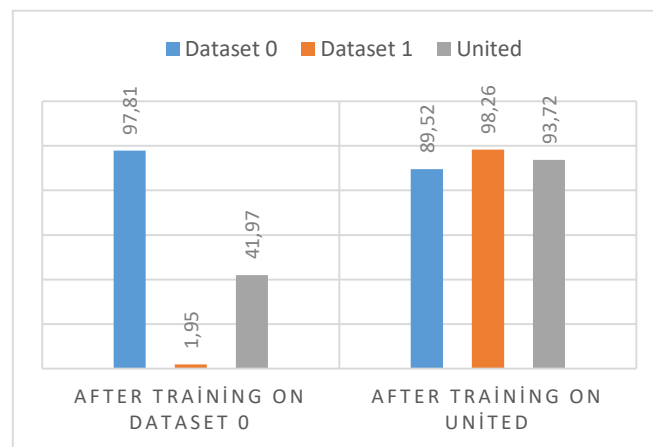
**Figure 6.** Results of retraining using a united dataset for two different initial values of learning rate. The average accuracy of keypoint detection measured on the test set in PDJ percent.

In the case of initial training, it can be said that the network was able to correctly detect keypoints in only those photographs that belonged to ‘dataset 0’. In case 1 the average detection accuracy was high, especially for ‘dataset 1’. Since the network parameters varied according to both datasets during training, accuracy for samples from ‘dataset 0’ dropped, albeit slightly. In case 2 the accuracy increased for all datasets. We can say that the initial value of the LR was not too small, so the network has acquired new knowledge. However, due to the reduction of keypoint detection accuracy of samples of ‘dataset 0’, we cannot consider the result to be acceptable.

The next time the network was retrained without initial training.

**Table 4.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘united’	1e-3	2e-4



**Figure 7.** Results of training using a united dataset without retraining the initially trained model. The average accuracy of keypoint detection measured on the test set in PDJ percent.



**Figure 8.** Keypoint detection results after training on ‘united’ in samples from dataset 0 (left\*) and dataset 1 (right). Detection accuracy in PDJ percent: left – 85,71%, right – 100%.  
\*(Material courtesy of Dr. Bill Vicars and [www.lifeprint.com](http://www.lifeprint.com))

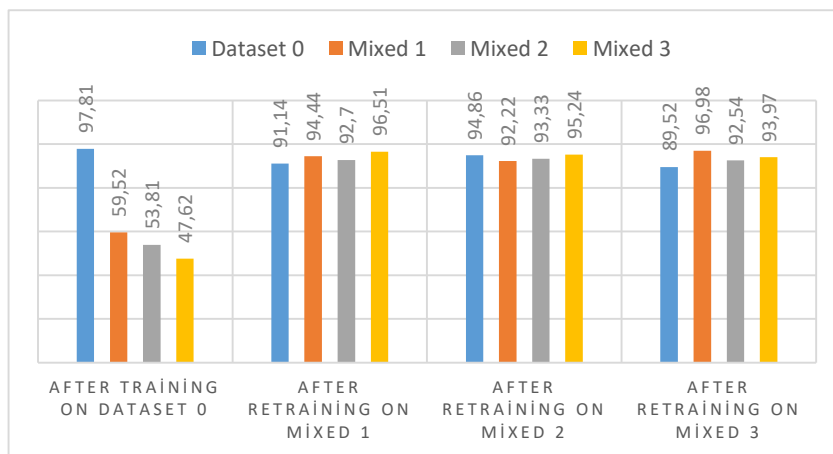
In this case, the accuracies obtained for all datasets are relatively close to each other. Obviously, better results were obtained when training network without initial training.

### 3.3. Results of Retraining using Mixed Datasets

This experiment required the building of mixed datasets. At first, ‘dataset 0’ and ‘dataset 1’ were ‘united’, shuffled and divided into three parts. The number of samples in all three mixed datasets was equal to 230. In each dataset 200 samples were used for training and validation sets (160 and 40, respectively) and 30 for the test set. The obtained mixed datasets are called: ‘mixed 1’, ‘mixed 2’ and ‘mixed 3’. As in the case of the ‘united’ dataset, we conducted training with two approaches. The network trained using ‘dataset 0’ was retrained sequentially using the mixed datasets. The initial value of LR was the same for each retraining process as it is shown on...

**Table 5.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘mixed 1’	1e-3	4e-5
‘mixed 2’	1e-3	8e-6
‘mixed 3’	1e-3	4e-5



**Figure 9.** Results of retraining using mixed datasets in case of the same initial value of the learning rate. The average accuracy of keypoint detection measured on the test set in PDJ percent.



After initial training, the average detection accuracy on each mixed dataset was about 50%. This can be explained by the fact that part of the samples of the mixed datasets are taken from the ‘dataset 0’. After retraining the model using ‘mixed 1’, the average detection accuracy on all datasets has increased dramatically. This result persists with slight changes after each retrains. It is obvious from the diagram that there is no stable improvement of accuracy on any dataset after each retraining. The reason for this may be the following: Because of the relatively high initial value of the learning rate the parameters of the network significantly change on every retraining process. Consequently, during each retraining, there is no improvement in the gained knowledge, but the task is studied anew. Because of this, we decided to reduce the learning rate by 10 times on each retraining.

**Table 6.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘mixed 1’	1e-3	4e-5
‘mixed 2’	1e-4	2e-5
‘mixed 3’	1e-5	1e-6



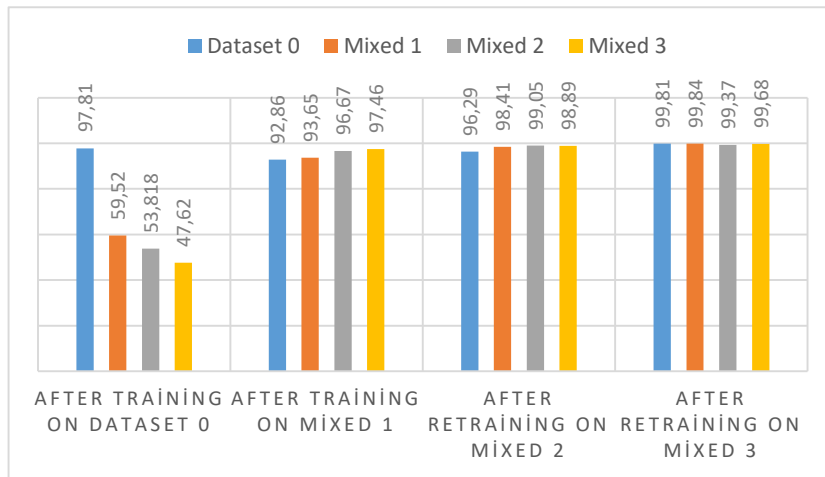
**Figure 10.** Results of retraining using mixed datasets in case of gradually reducing the initial value of the learning rate at each retraining. The average accuracy of keypoint detection measured on the test set in PDJ percent.

As we can see, the gradual reduction of the initial value of the LR significantly affected the change of average detection accuracy. Network detection accuracy increases at all datasets after each retraining. This means that each retraining leads to an improvement of the detector.

On the next stage, the deep learning model was trained on ‘mixed 1’, ‘mixed 2’ and ‘mixed 3’ datasets without initial training.

**Table 7.** The value of learning rate at the begin and end of each training

Training on Dataset	LR at train begin	LR at train end
‘dataset 0’	1e-3	1e-6
‘mixed 1’	1e-3	2e-4
‘mixed 2’	1e-4	2e-5
‘mixed 3’	1e-5	1e-6

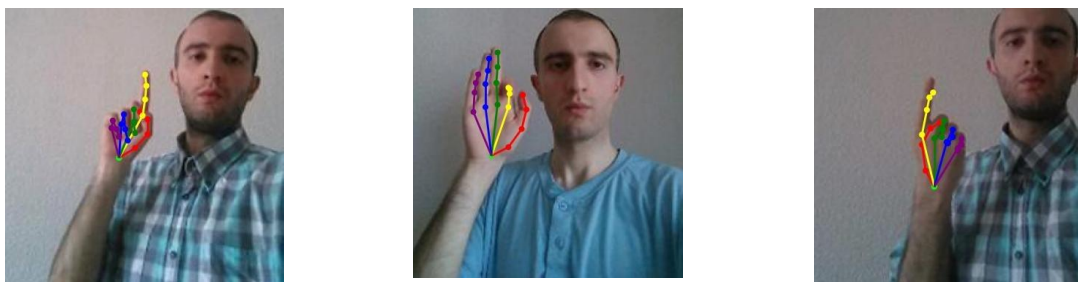


**Figure 11.** Results of retraining using mixed datasets without retraining the initially trained model. The average accuracy of keypoint detection measured on the test set in PDJ percent.



**Figure 12.** Keypoint detection results after training on mixed datasets in samples\* of dataset 0. Detection accuracy in PDJ percent: left – 100%, middle – 100%, right – 100%.

\*(Material courtesy of Dr. Bill Vicars and [www.lifeprint.com](http://www.lifeprint.com))



**Figure 13.** Keypoint detection results after training on mixed datasets in samples of dataset 1. Detection accuracy in PDJ percent: left – 100%, middle – 100%, right – 100%

The diagram shows that a result similar to the previous case was obtained. We can say that even without initial training, it was possible to study the keypoint detection with sufficiently high accuracy. Most importantly, average detection accuracy is progressive during each retraining.

Fairly high detection accuracy was achieved in the case of both mixed and ‘united’ datasets. However, especially high accuracy was measured when retraining using mixed datasets. After retraining on dataset ‘mixed 3’, the average detection accuracy at each dataset reached almost 100%. The average detection accuracies at each dataset were much closer together than in the case of the ‘united’ dataset. Although

the mixed datasets were obtained from the ‘united’ dataset, sequential training of the mixed datasets showed relatively better results.

It should be noted that the indicated result was obtained in the case of the gradual decrease in the initial value of the learning rate before each retraining. This change in the learning rate, and the fact that the mixed datasets contain samples from different datasets, ensure that the knowledge gained in the previous time is not forgotten. As a result, changing the network parameters at each retraining improves detection accuracy.

#### 4. DISCUSSION

After the examination of the retraining techniques, we can identify their pros and cons and choose the optimal one.

**Table 8.** Advantages and disadvantages of retraining in series technique

Retraining Technique	Advantages	Disadvantages
Retraining in series	<ul style="list-style-type: none"> <li>Retraining takes place sequentially which makes the training more flexible</li> </ul>	<ul style="list-style-type: none"> <li>Detector ‘forgets knowledge’ gained from previous training</li> <li>Total time of retraining is more than in case of using the ‘united’ dataset</li> </ul>

**Table 9.** Advantages and disadvantages of technique: Retraining using united dataset

Retraining Technique	Advantages	Disadvantages
Retraining using united dataset	<ul style="list-style-type: none"> <li>A detector can learn from different datasets with high, almost equal accuracy</li> </ul>	<ul style="list-style-type: none"> <li>Training requires a large amount of memory compared to the other two techniques</li> <li>The training takes place in one continuous process, which is time-consuming</li> </ul>

**Table 10.** Advantages and disadvantages of technique: Retraining using mixed datasets

Retraining Technique	Advantages	Disadvantages
Retraining using mixed datasets	<ul style="list-style-type: none"> <li>A detector can learn from different datasets with very high, almost ideally equal accuracy</li> <li>Retraining takes place sequentially which makes the training more flexible.</li> </ul>	<ul style="list-style-type: none"> <li>Total time of retraining is more than in case of using the ‘united’ dataset</li> </ul>

With the first technique, the detector was unable to retain the knowledge gained from the previous training. As a result of training using this method, the detector cannot detect keypoints highly accurately in samples of more than one dataset. Therefore, this approach is not recommended. Regarding the other two techniques, the desired result was obtained using both in terms of detection accuracy. However, the accuracy was relatively high when using mixed datasets. However, the accuracy was almost the same for all datasets.

It is also important to compare techniques according to the resources required. In terms of memory, since there are much more samples in a ‘united’ dataset, using this dataset is more resource-intensive than using mixed datasets. In terms of training time, using ‘united’ dataset takes longer than each retraining using the other two methods. However, if we add up the time it takes to retrain in series or using mixed datasets, we find that it takes more time overall than in case of training using ‘united’ dataset. But training using ‘united’ dataset is conducted in one continuous process. Using the rest of the techniques, we can retrain any time we want, which makes these approaches more flexible.

We can say that retraining using mixed datasets combines the advantages of the other two approaches. In particular, each retraining requires relatively little memory and time resources. And most importantly, using this technique, we can create a fairly accurate keypoint detector. According to the aforementioned, retraining using mixed datasets should be considered a relatively optimal method.

## **5. CONCLUSION**

Three approaches of retraining of a keypoint detector have been examined. The first one involved retraining once trained deep learning model using new datasets in series. Testing of the technique has shown that the model learns detection, although the knowledge gained on previous training cannot be retained. Therefore, a detector trained using this approach can not accurately detect keypoints in samples from multiple datasets. For the above reason, we found this approach ineffective. The second method involved the use of a ‘united’ dataset. For each dataset, high and almost equal accuracy was measured. Although, this technique requires relatively more memory compared to others. Training using this method is a time-consuming continuous process. The third approach was retraining using mixed datasets. A sufficiently high average detection accuracy was measured, and most importantly, on all datasets. Each retraining requires relatively little memory and time. The advantage of the technique is also its flexibility, which means that we can conduct the desired number of training and at the desired time. After analyzing the results, we can conclude that using mixed datasets will be a relatively optimal technique for retraining.

## **ACKNOWLEDGEMENTS**

This work was supported by Shota Rustaveli National Science Foundation (SRNSF) [PHDF—18-342, Optimized communication system for sign language speakers].

## **REFERENCES**

- [1] Strecha C, Lindner A, Ali K and Fua P. Training for Task Specific Keypoint Detection, in Denzler J., Notni G., Süße H. (eds) Pattern Recognition. DAGM 2009. Lecture Notes in Computer Science, Berlin, Heidelberg.
- [2] Howard A, Zhu M, Chen B, Kalenichenko D, Wang WWT, Andreetto M and H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017.

- [3] Sandler M, Howard A, Zhu M, Zhmoginov A and Chen L. MobileNetV2: Inverted Residuals and Linear Bottlenecks, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018.
- [4] Toshev A and Szegedy C. DeepPose: Human Pose Estimation via Deep Neural Networks, in IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014.