Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# PCA AND PROJECTION BASED HAND GESTURE RECOGNITION

[1]ASKHAT AITIMOV, [2]ZHASDAUREN DUISEBEKOV, [3]SHIRALI KADYROV, [4]CEMIL TURAN

[1,2,4]Computer Sciences Department, Suleyman Demirel University, Kaskelen, 040900, Kazakhstan

[3]Mathematics and Natural Sciences Department, Suleyman Demirel University, Kaskelen, 040900, Kazakhstan

E-mail: [2]zhasdauren.duisebekov@sdu.edu.kz

**ABSTRACT**

Developing hand gesture recognition algorithms, and more generally, pattern recognition algorithms is a very active area of research in computer vision. There are various approaches and techniques to the recognition problem among researchers. In this manuscript, our objective is to develop a novel Principal Component Analysis based hand gesture recognition algorithm, and compare its performance against k-Nearest Neighbor classifier and Sparse Representation based Classifier. The proposed algorithm makes use of linear triplet loss embedding and projections onto subspaces. An open source HandReader dataset consisting of 500 labeled images with 10 signs from American Sign Language is split into a training set with 100 images and a test set with 400 images. The proposed algorithm outperforms with 95% accuracy. This shows that the proposal methodology might be effective in computer vision when there is relatively small amount of data is available. It is expected that approaches similar to the current one will contribute the emergence of machine learning algorithms with Principal Component Analysis based techniques.

**Keywords:** *Computer Vision, Sign Language, Hand Gesture Recognition, Human-Computer Interaction, Triplet Loss, Stochastic Gradient Descent, PCA-TP, SRC, kNN*

## 1. INTRODUCTION

With the advances of technology, nowadays, the human interaction and communication with computing devices has become inevitable and it is more likely to continue to be one of the active areas of research in computer vision. Clearly, there are various forms of human-computer interaction (HCI). In its traditional form, a simple mouse and a keyboard are two common tools in HCI. While mouse-like objects are useful in handling two dimensional interactions, they are far from being convenient in three-dimensional real-world applications. It is safe to say with the advancement of smart technologies in the last decade there was a paradigm shift in HCI to graphical interface based on graphic objects [1]. As in human-human interactions, gestures are becoming very convenient and natural way in HCI where three-dimensionality is not an issue at all. There are more than ten different body parts that humans use in communication and gesturing. In a thesis work [2], hand gesturing is found to be the most common ways of communication with the relative frequency of usage being about 21%.

Sign Language is an essential tool for communication among hearing-impaired community, and is a structured form of hand, face, and body gestures. It is also an important method in HCI in applications such as game or robot control, virtual reality, and so on [3]. For such automated systems pattern recognition plays crucial role and there are various methods being developed to improve pattern recognition algorithms and address new challenges, see e.g. [4, 5, 6, 7, 8, 9] and references therein.

While many state-of-the-art static hand gesture recognition algorithms use two dimensional convolutional neural networks based deep learning techniques [3, 10], they have two main shortcomings. One of the weaknesses of these techniques, common to other machine learning techniques, is absence of robustness due to under specification [DHM15] where the algorithm may end up finding wrong local minimum. Another, issue with the deep learning approaches is the

overfitting. This may occur especially if the dataset is not sufficiently large for the given task.

In this paper, our objective is to develop a novel Principle Component Analysis based algorithm together with Triplet similarity embedding and Projections (PCA-TP), and use the open access HandReader dataset [12] to compare the performance of the proposed method to two common techniques from the literature, namely, a Sparse Representation based classifier and a k-Nearest Neighbor classifier.

The contribution of this research is to provide a new approach (PCA-TP) to the hand gesture recognition problem. Since PCA and orthogonal projections are robust techniques, our algorithm can be considered as a robust approach as opposed to the deep learning techniques. Moreover, this approach does not necessarily require large dataset and yet expected to provide high recognition accuracy. Since, triplet similarity embedding was first implemented in the machine learning community in its original form, as a secondary contribution, our proposed algorithm is an attempt to bring two seemingly different approaches into common ground.

In the next section we provide a review of literature on the subject of the study. Section 3 describes details of our experimental setup. In section 4 we state the results of our experiment, and in the last section we end with the discussion of the results, limitations, and future work.

## 2. LITERATRURE REVIEW

In human-computer interaction, hand gesture and sign language recognition is a task of following and identifying various performed signs and translating them into computer understandable expressions or words. There are sensor-based and vision-based data acquisition techniques in gesture recognition [13]. In the sensor-based approaches the measurements such as gyroscope, accelerometer, electromyography, wi-fi, and radar are some of the common tools implemented to acquire the data. On the other hand, in vision-based approaches to the gesture recognition video cameras, leap motion controllers, and body markers are some of the customary instruments used to collect image or video datasets. Since we are interested in vision-based gesture recognition models we will not concentrate on sensor-based approaches and only refer to the survey [14] for interested readers on the subject. The end-to-end vision-based gesture recognition procedure has steps including data acquisition, image preprocessing, hand tracking, video segmentation, feature extraction, and finally classification. In [15] hand tracking and segmentation methods were introduced for Indian sign language recognition problem. The authors' own data was collected with labels as letters in English alphabet and numbers from one to ten. The video frames were converted into grayscale images and cropped to reflect the region capturing the hands of the participants. The idea of hand tracking was based on difference analysis of adjacent frames together with YCbCr skin color detection model [16]. As for the hand segmentation, thresholding technique was implemented together with edge detection methodology. Due to space limitation, we content ourselves with the review of static hand gesture recognition approaches and for other techniques and steps of vision-based gesture recognition we refer to [13, 19, 20, 21] and references therein.

Static hand gestures recognition system was proposed in [22] for Indian sign language. After segmentation process, 43 features were extracted that included seven geometric features such as circularity, eccentricity, convexity, and irregularity as well as 36 Zernike moments of the images. The classification and recognition are done using Euclidean distance of covariances of the feature vectors. The proposed model was tested in a database consisting of 318 images from 19 distinct signs with total accuracy reaching as high as 98.74%. Another recognition task [23] on 10 Indian sign gestures were introduced based on Principal Component Analysis (PCA) with accuracy around 90%. PCA is one of the classical and well recognized robust techniques from linear algebra that is utilized in dataset dimension reduction tasks.

Another work [24] studies fuzzy rule based Bangali sign language recognition problem. For corner detection, local auto-correlation methodology due to Harris [25] was implemented. The recognition was done based on fuzzy rules of hand configurations.

k-Nearest Neighborhood and Support Vector Machine (SVM) based algorithms were proposed and compared in [26] for their recognition performances on their own open access HandReader dataset with dark background, Figure 1. The hand images were converted into binary format as result of histogram-based background subtractions, morphological operations, Gaussian filtering, and thresholding. Comparison of the algorithms show that Support Vector Machine algorithm [27] with linear kernel outperforms

(recognition rate 96%) k-Nearest Neighborhood classifier (recognition rate 93%) [28].

Artificial Neural Networks-based approach with 4 cameras is proposed in [29] for sign language recognition. The network has one hidden layer consisting of 80 neurons used to classify images labeled with 36 signs with 144-dimensional input and sigmoid activation. The dataset of 288 were split into train and test sets equally and recognition rate reaching 95.1%.

A real-time gesture recognition problem on American sign language (ASL) proposed in [30] based on Convolutional Neural Networks (CNN). To this end, ASL FingerSpelling Dataset from the University of Surrey over 65,000 images for 24 static signs as well as Massey University Gesture Dataset of 2,524 images were used, In implementing the CNN network with softmax activation weights of GoogLeNet trained on ILSVRC 2012 was utilized. The findings may be considered robust for letters a-e, while it is modest for letters a-k (excluding letter j) with overall performance reaching 97.82% accuracy.

Using K-means clustering an American sign language recognition mobile application is proposed in [31]. For segmentation of hand gestures, Canny edge detection and region growing methodologies were utilized. SURF descriptors were used as extracted features before the classification into 16 classes with K-means clustering technique. Finally, SVM algorithm is used to classify the signs reaching 97.13% of accuracy.

Another image based ASL recognition system was recently proposed using deep learning [32]. The proposed model uses CNN together with skin detection and convex hull approach reaching recognition accuracy of 98.05%. The dataset consists of 900 images with 36 having 25 samples each split as 80% for training and 20% for testing. The RGB images were converted into YCbCr using skin color detection technique [16]. The CNN architecture consists of two Conv2D layers followed by pooling, flattening, and two dense layers with 9216 and 128 nodes respectively.

Recently, Turan [17] compared (Modified) Sparse Representation based Classification (MSRC) to Principal Component Analysis (PCA) based classification on Yale Database B of face images, and showed that MSRC outperforms PCA in face recognition. To represent feature vectors as sparse as possible, l1 minimization is implemented. Later in [18] Aitimov et al used part of HandReader dataset [12] to compare the performance of MSRC to k-Nearest Neighbor (kNN) and Random Forest classifiers, and showed that MSRC always performs better.

As it is seen from the literature, machine learning based approaches use huge amount of data for training their model. With fewer data, the models are more likely to run into overfitting issue. In this article, we propose PCA-based approach which does not require large datasets and are robust to the recognition problem considered.

## 3. EXPERIMENTAL SETUP AND METHODOLOGY

In this section we provide information on the dataset we study, dataset preprocessing, introduce k-Nearest Neighbor classifier, l_1-minimization based Sparse Representation classifier, and finally, the proposed Principal Component Analysis based classifier (PCA-TP) combined with Triplet similarity function and Projection as a recognition metric.

### 2.1 HandReader dataset

In this work, we use HandReader dataset [12]. HandReader includes 500 static images of 10 kinds of hand postures from American Sign Language. The ten signs used from the alphabet are letters A, B, C, D, G, H, L, I, V, and Y, where for each hand posture there are 50 representative images (see Figure 1). The dataset is split into two: Training set and Test set. The training set contains 100 images, 10 for each sign, while the test set contains the remaining 400 images, 40 for each sign.

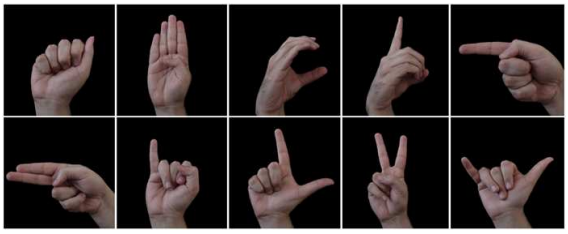

*Figure 1: The signs from the HandReader dataset. From left to right, they correspond to the letters A, B, C, D, G, H, I, L, V, and Y*

### 2.2 Dataset preprocessing

In our experimental setup we want to compare the three algorithms for both raw and preprocessed dataset. In all cases, the final size of an image is

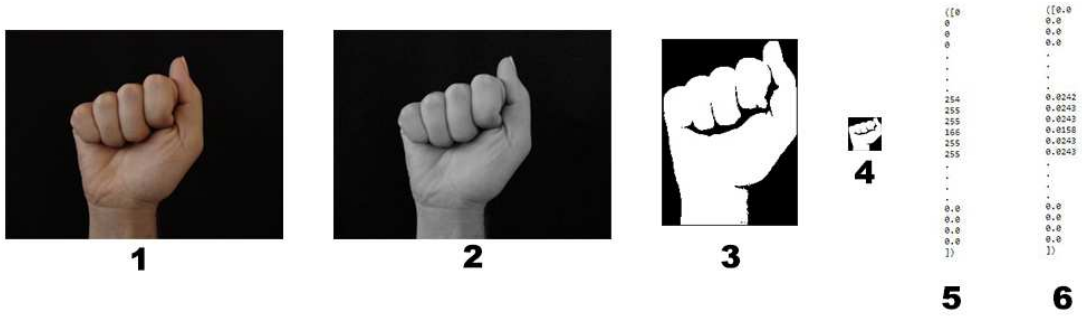


*Figure21: Dataset Preprocessing Steps*

converted to 30x30 pixels, which is then flattened into 1 dimensional column vector. See Figure 2 for the preprocessing steps.

To preprocess the dataset, each image is converted into grayscale, then to black and white using threshold 60 within the range [0, 255], where white color represents the hand. Then, the hand is cropped out to a rectangular image, which is then reshaped into a 30x30 image. See Figure 3, for the preprocessed image before reshaping into 30x30.

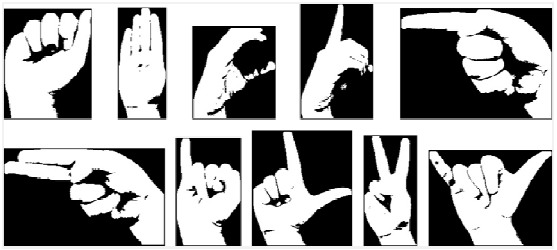

*Figure 3: Examples from preprocessed dataset*

### 2.3 Review of classifiers

Let *Train* denote the training set $\{x_1, x_2, \ldots, x_N\}$ consisting of $N$ labeled m dimensional vectors, and *Test* be the labelled $M$ images $\{y_1, y_2, \ldots, y_M\}$ allocated for validation. We now review k-Nearest Neighbor classifier and Sparse Representation based classifier in details. As mentioned in the introduction, we compare the results of our proposed model to these two classifiers and report the findings. Our proposed PCA based novel algorithm will be introduced in the upcoming subsection in details.

#### 2.3.1 k-Nearest Neighbor Classifier

The k-nearest neighbor (kNN) classifier is one of the classical machine learning algorithms used to label a new image using a labeled dataset of images [33, 28]. More precisely, given a set of $N$ images $x_i$, i=1, 2, …, $N$ with labels $\{l_i\}$ and a new test image $y$, the kNN attempts to find the k-nearest neighbors of $y$ from the set, and labels it according to majority votes. Clearly, the distance function $d$ (taken to be Euclidean by default) and the parameter k plays crucial roles in the classification. See Algorithm 1 for a summary of kNN.

**Algorithm 1. k-Nearest Neighborhood Algorithm**

**Input:** $\{x_i, l_i\}$, I = 1, 2, …, $N$ and $y$ in $\mathbf{R}^n$
**Output**: label for $y$
**For** i:=1 to $N$ do
  Find Euclidean distance $d(y, x_i)$
**End**
**Sort** the $N$ distances in ascending order
**Pick** the labels of the first $k$ nearest distances
Assign a label $l$ to $y$ which is most frequent
**End**

#### 2.3.2 Sparse Representation Classifier (SRC)

Here we review Sparse Representation Classifier proposed in [34] and improved in [17], see also [35].

We create an $m \times n$ matrix $A = [A_1, A_2, \ldots, A_N] = [v_{1,1}, v_{1,2}, \ldots, v_{i,k}]$ from a training set of images. It includes the vectors by concatenating the columns of each gray image intensity matrix as $v_{i,k}$ where $i$ represents the label of image in classes and $k$ is the index number of each class. For a given test sample image vector $y$ in $\mathbf{R}_m$, to assign a label we numerically solve $l_1$-minimization problem that minimizes the norm $\|x\|_1$ subject to $Ax = y$, and then calculate the sum of entries $\delta_i(x)$ that corresponds to each *i-th* label and obtain $R_i$=sum($\delta_i(x)$). As a result, because of high correlation between the test sample and training

class, the maximum sum gives the corresponding class as: identity

$$(y) = arg\max_i\{R_i\}.$$

Any given test image can be reconstructed by the product of training image matrix and sparse reconstruction vector as explained in Figure 4. Different colors are used for different image classes in the columns. The same class images have the same color and placed in the training matrix ($A$) group by group. To reconstruct the test image ($y$), the sparse reconstruction vector ($x$) takes the significant entries from the related columns of $A$ because of high correlation between the same class images. To classify the test image, we just calculate the sum of entries group by group (here two by two) in $x$. Finally, the place of maximum sum gives us the corresponding class and thus the recognition is completed.

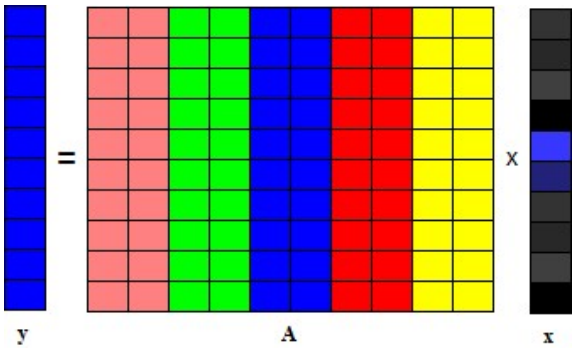


*Figure 4: Classification by sparse reconstruction vector*

We refer to Algorithm 2 for details.

**Algorithm 2. Modified Sparse Representation Classifier**

**Input** $m \times N$ matrix $A = [A_1 \ \ldots \ A_N]$ for $N$ labels and $y \in \mathbb{R}^m$
**Output** label for $y$

Normalize columns of $A$ w.r.t. $l_2$-norm
Solve $x_1 = argmin \|x\|_1$ subject to $Ax = y$
**for** i = 1 to $N$ **do**
Compute $\mathbf{R}_i$ = sum of coordinates of $x_1$ corresponding to label i
**End**

Assign label = $arg\ max_i\{R_i\}$
**End**

### 2.4 Proposed PCA and Projection Based Classifier

As before,

$$Train = \{x_1, x_2, \ldots, x_N\}$$

denotes the training set consisting of labeled $m$ dimensional vectors and

$$Test = \{y_1, y_2, \ldots, y_M\}$$

is the test set. We now provide details of our proposed algorithm. It has three main components, namely PCA, triplet similarity embedding, and projections as a similarity metric.

#### 2.4.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the classical and widely used linear dimension reduction techniques where an image is represented as a feature vector with a smaller dimension. More precisely: Let $A = [A_1 \ \ldots \ A_N]$ be the $m \times N$ matrix with $m$-dimensional vectors from a training set stacked into columns. Let $\mu$ denote the $m$-dimensional column vector given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

and $X_0 = X - \mu$ where the vector $\mu$ is subtracted from each column of $X$. Next, we compute the eigenvalues

$$\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N\}$$

and corresponding eigenvectors

$$\{e_1, e_2, \cdots, e_N\}$$

for the covariance matrix

$$\frac{1}{n-1} X_0 X_0^T.$$

Note that since the covariance matrix is symmetric, its eigenvalues are always real. If we plan to reduce the dimension $m$ into $d$ we form an $d \times m$ matrix $Xpca$ from the first $d$ eigenvectors

$$\{e_1, e_2, \cdots, e_N\}.$$

There are various methods to decide the integer d including Scree Plot analysis. For our purposes, we simply take $d$ to be the number of nonzero eigenvalues counting the multiplicity.

Finally, to transform $m$-dimensional vectors $x$ representing images into $d$-dimensional feature vectors $x'$ we simply carry matrix-vector multiplication

$$x' = Xpca\ (x - \mu)..$$

### 2.4.2 Triplet Similarity Embedding

Triplet Similarity Embedding is a machine learning technique introduced in [38] for the purposes to improve face recognition, verification and clustering methods in deep learning models. Idea is to train a similarity function $f$ such that for any anchor {a} and positive {p} with the same label and a negative {n} with a different label, we wish to have our distance function satisfying

$$dist(f(a), f(p)) < dist(f(a), f(n))$$

for any given metric dist ( , ).

To avoid trivial functions such as $f = 0$ a positive hyperparameter $\alpha > 0$ is added and one aims to minimize the

$$max(0, \alpha + dist(f(a), f(p)) - dist(f(a), f(n)))$$

for all triplets {a, p, n} in the training set.

For our purposes we train a linear map $f$ and use projection based approach as in [36]. We know that for two unit vectors $u,v$ the cosine of the angle $\theta$ between the two can be computed by cos(θ)=(u,v), where ( , ) is the dot product operation. So, if u,v are feature vectors with the same label, they are expected to be close to each other, so that θ is close to zero, which makes cos(θ) large. We know that the dot product up to sign can be interpreted geometrically as a Euclidean norm of projection of u into the span of v provided the latter is of unit length. So, in the projection setting, close means large dot product. This implies that we need to train the similarity function f so that

$$max(0, \alpha + dist(f(a), f(p)) - dist(f(a), f(n)))$$

is minimal.

For a linear function f, we let f(x)=Wx for some $r \times d$ matrix W where d is the dimension of feature vectors and integer $r \leq d$ is a hyperparameter. Hence, W reduces the dimension feature vectors from $d$ to $r$. Note that

$$(f(x), f(y)) = f(x)^T f(y).$$

Therefore, the minimization problem takes the form

$$argmin_W \sum_{\{a,p,n\}} max(0, \alpha + a^T W^T W n - a^T W^T W p),$$

where {a, p, n} runs from the training set. To optimize the W we implement Stochastic Gradient Descent (SGD). To avoid the overfitting we include the $l_2$-regularization factor. As the gradient of individual loss satisfies

$$\nabla(a^T W^T W n - a^T W^T W p) = W(an^T + na^T) - W(ap^T + pa^T),$$

in each iteration of SGD we need to update W with

$$W_{t+1} - W_t - \beta W(a(n-p)^T + (n-p)a^T) - \gamma W \quad (1)$$

where the last term is the $l_2$-regularization. For the experiment, we iterate SGD 1000000 times for the hyperparameters α=β=0.001, γ=0.00001 and initiate our matrix W as a 50×$d$ identity matrix, where $d$ is coming from PCA.

### 2.4.3 Projection as a Similarity Metric

To apply PCA for pattern recognition, various metrics can be considered, see e.g [37] where 15 different distance functions are considered. For our purposes we consider the Euclidean norm of the projection onto a subspace.

Given a basis of n column vectors for a vector subspace V in $\mathbb{R}^m$, let A denote the m×n matrix obtained from the basis by stacking them into the columns. It is well known, see e.g. [39], that an m by m matrix

$$P = A(A^T A)^{-1} A^T$$

defines the orthogonal projection onto the subspace V. In particular, it satisfies $P^T = P$ and $P^2 = P$. We may think of V as a subspace representing various images for a label.

If the Euclidean norm of $Py$ for a feature vector $y$ of an image from the test set is larger compared to projections onto subspaces of other labels, then we may interpret it to have the same label. The square of an Euclidean norm for $Py$ satisfies

$$\|Py\|_2^2 = (Py, Py) = y^T P^T P y = y^T P^2 y = y^T P y$$

Thus, this formula can be implemented to each labeled class to compare the norms of the projections. So, given a test feature vector $y$, we may assign a label to it via

$$y\ label = argmax_i\ y^T P_i y \quad (2)$$

where $P_i$ represents the projection matrix onto a vector subspace generated by feature vectors from test set belonging to the label i. We note that A must have linearly independent columns, in particular, the images in the training set should not be redundant.

Another option is to compare projections of $y$ onto one dimensional vector subspaces corresponding to single feature vectors and then take their sum. More precisely, one can label y with

$$y\ label = argmax_i \sum_{x \in Train_i} x^T \quad (3)$$

where $Train_i$ is the set of feature vectors with label i. Finally, one can take weighted average of the two approaches and consider $y$ label to be

$$argmax_i \left[ \omega\, y^T P_i y + (1-\omega) \sum_{x \in Train_i} x^T y \right] \quad (4)$$

where $\omega \in [0,1]$. For our purposes we take $\omega = 0.5$.

To summarize we have

**Algorithm 3. Proposed Algorithm PCA-TP**

**Input:** $m \times N$ matrix $A = [A_1 A_2 \cdots A_N]$ for $N$ labels, $y \in \mathbb{R}^m$, $r, T \in \mathbb{N}$, $\alpha, \beta, \gamma > 0$, $\omega \in [0,1]$
**Output:** label for $y$
Normalize columns of $A$ and y w.r.t. $l_2$-norm
Apply PCA to reduce $A$ to an $d \times N$ matrix $A'$
Transform $y$ into $y'$ in $\mathbb{R}^d$ with PCA transformation
Set $W = r \times d$ identity matrix
Use SGD to train a triplet similarity $k \times d$ matrix $W$ according to (1):
**for** $i = 1$ *to* $T$ **do**
take random $\{a, p, n\}$ from columns $A$ with $\{a, p\}$ the same label.
**if** $\alpha + a^T W^T W n - a^T W^T W p > 0$ **do**

$$W := W - \beta W\,(a(n-p)^T + (n-p)a^T) - \gamma W$$

**End**
Compute $A'' = WA'$ and $y'' = Wy'$
**for** $i = 1$ *to* $N$ **do**
Compute the similarity metric $R_i$ according to (4)
**End**
Assign label = $\arg\max_i \{R_i\}$
**End**

We now turn to reporting the findings of the experiments carried on kNN, MSRC, and proposed PCA-TP algorithms

## 3. RESULTS

In this section we state the results of our experiment for kNN, MSRC, and proposed PCA-TP and compare their recognition performances. We carried experiments on the HandReader dataset in two situations: without any preprocessing stage and with preprocessing stage explained in the methodology section.

After applying PCA each image feature vector is reduced to a new feature vector of dimension d=99.

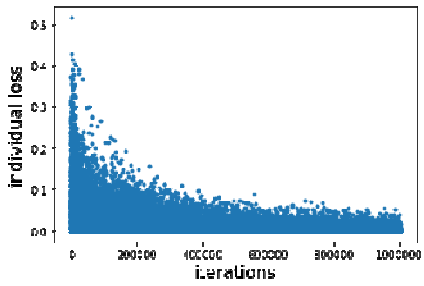


*Figure 5. Loss Function Trained On Raw Dataset*

The graphs of loss function over 1,000,000 iterations are given in Figure 5 and Figure 6.

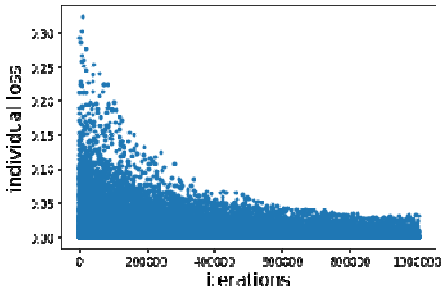


*Figure 6. Loss Function Trained On Preprocessed Dataset*

While in the loss falls below 0.05 in a pre-processed training set, it remains around 0.1 as a result of training on raw data. We also note that beyond 600,000 trainings the loss function more or less stabilizes in both situations.

### 3.1 Results for raw dataset

We now provide the experimental results for the raw dataset and how they compare to kNN and MSRC. The k Nearest Neighbor algorithm was used for three different situations when k takes values one, three, and five. It is customary to take k from odd integers and since the accuracy was decreasing with k increasing, we stopped at k=5 as shown in Table 1.

*Table 1. Recognition Accuracy For Raw Dataset*

| Classifier | kNN (k=1) | kNN (k=3) | kNN (k=5) | MSRC | PCA-TP |
|---|---|---|---|---|---|
| Accuracy | 70.25% | 65.25% | 64.50% | 64.75% | 77.25% |

*Table 2. Recognition Accuracy For Preprocessed Dataset*

| Classifier | kNN (k=1) | kNN (k=3) | kNN (k=5) | MSRC | PCA-TP |
|---|---|---|---|---|---|
| Accuracy | 91.00% | 88.50% | 84.75% | 90.75% | 95% |

Our proposed model PCA-TP is readily seen to outperform the other algorithms with recognition accuracy of 77.25%. We also note that, another experiment for PCA-TP without triplet similarity resulted in the recognition accuracy of 76%. The result shown for PCA-TP in Table 1 is when the weighted metric (4) is implemented with $\omega = 0.5$. Moreover, additional experiment for PCA-TP with metric (2) and (3) the recognition accuracy was 75% and 77%, respectively. Figure 7 shows the recognition accuracy of PCA-TP for the individual letters.

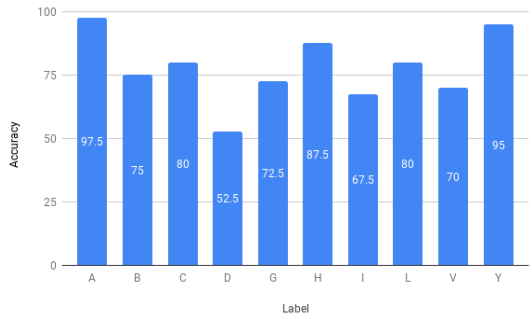


*Figure 7. Recognition Accuracy For Individual Signs On A Raw Dataset*

We see that the proposed algorithm performs best on the letter A with accuracy of 97.5% while it is lowest with accuracy 52.5% on the letter D.

### 3.2 Results for preprocessed dataset

We now state results of our experiments on preprocessed dataset as explained in the previous section. The main findings are reported in Table 2.

Once again, our proposed model is the best performing with 95% high accuracy compared to the classical recognizers. For both raw dataset and preprocessed dataset experiments we see that kNN is performing better than MSRC. The experimental result for PCA-TP without triplet similarity would give accuracy of 94%. Moreover, for PCA-TP with metric (2) and (3) the accuracy is 93.75% and 94.25%, respectively. Figure 8 shows the recognition accuracy of PCA-TP for the individual letters.

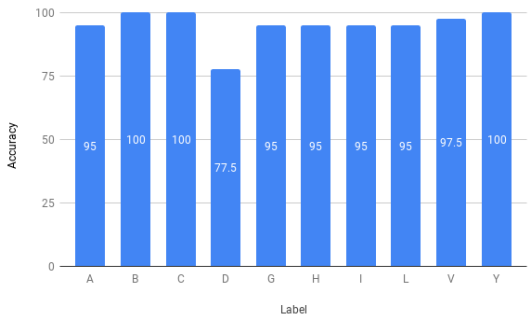


*Figure 8. Recognition Accuracy For Individual Signs On A Preprocessed Dataset*

From Figure 8, we see that PCA-TP is mostly failing to recognize the letter "D". Detailed analysis shows that "D" is falsely recognized as the letter "I" in most cases.

## 4. DISCUSSION

In this work we compared k-NN, MSRC, and our proposed novel algorithm (PCA-TP) on hand gesture recognition. The results show that PCA-TP outperforms both algorithms. We note that Tofighi et al [26] considered the same dataset with a more

sophisticated preprocessing and used a version of Support Vector Machine classifier with linear and RBF kernel and showed that in the linear case the accuracy reaches 96% while our PCA-TP has 95% accuracy (Table 2). They additionally considered application of Gaussian Blur, squaring the image canvas, etc. In their work, algorithms mostly failed to recognize letter "G" which is falsely recognized as letter "H". In our case, PCA-TP was falsely identifying letter "D" with letter "I". So, if the two algorithms are combined the accuracy is more likely to increase.

As mentioned before a recent work [18] studied the hand gesture recognition problem using the same HandReader dataset but only part of it. More specifically, they considered four out of ten letters, namely, the letters A, B, C, and D and compared the recognition rates of kNN, MSRC, and Random Forest (RF) classifiers. Moreover, they reported the performances depending on the resized images and also the recognition speed. In all cases, MSRC turned out to be the best performing, followed by kNN, and RF the less performing. Depending on the sizes of images, the recognition rate of MSRC reported to vary from 95% to 97.5%. In our case, when the complete HandReader dataset is utilized, the performance of MSRC dropped to 90.75% while our proposed model reaches 95% accuracy.

Aside from proposing a novel algorithm, our approach can be thought as an attempt to bring linear algebra based robust techniques and machine learning based nonlinear methods into common ground. We note that the contribution coming from the triplet similarity embedding (machine learning) is about 1%, which is still fine, however, it would be nice to see in a future work if other kinds of triplet embedding functions improve the performance.

As mentioned before, while deep learning architectures such as convolutional neural network models have satisfactory results in computer vision, the models proposed as in this article have two main advantages: robustness and effectiveness on small size datasets. Besides, by merging PCA with machine learning method, namely triplet embedding function, our approach contributes to the emergence of seemingly two different methodologies for computer vision problems.

## 4. CONCLUSION

In conclusion, we proposed a novel algorithm based on PCA, triplet embedding, and projections effective in vision-based hand gesture recognition tasks with recognition accuracy reaching as high as 95%. This is clearly a promising performance given the size of the HandReader dataset. Once the images are processed, the images are reshaped into vectors and are reduced using PCA preserving the significant features. These feature vectors are then combined with triplet embedding to improve the embedding manifolds. Projection based metric is implemented to label the images. Only 20% of the dataset was allocated for training, while 80% for the test. This is completely opposite to deep learning models, where it is customary to allocate the most of the data is for training. Hence, we can see the robustness of our model for small size datasets.

One limitation of PCA-TP is in the formula (2), where we need to know that A has linearly independent column vectors which may fail in lower dimensional representations. To overcome this issue, some of the 'redundant' images should be removed from the training set.

While we see that the projections prove useful as a similarity metric, we note one shortcoming. A vector y which is almost in opposite side of x will have large projection length onto the subspace generated by x. Thus, for the metric (2) our algorithm may mistakenly consider y the same label as x. However, with (4) this issue may not arise. Otherwise, one may combine our metric with the usual Euclidean norm.

It is not clear whether the performance of the proposed model could decrease on the dataset containing more than 10 letters. This could be another future work to investigate and if so, then look for methods to improve the algorithm.

We also note that, in our model we trained linear triplet embedding function. It is expected that the nonlinear triplet embedding approach would improve the performance.

## REFRENCES:

[1] Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. Artificial intelligence review, 43(1), 1-54.

[2] Karam, Maria (2006) A framework for research and design of gesture-based human-computer interactions. University of Southampton, ECS, Doctoral Thesis, 180pp.

[3] Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications, 28(12), 3941-3951.

[4] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., & Kiru, M. U. (2019). Comprehensive review of artificial neural network applications to pattern recognition. IEEE Access, 7, 158820-158846.

[5] Bogdanchikov, A., Kariboz, D., & Meraliyev, M. (2018). Face extraction and recognition from public images using HIPI. In 2018 14th International Conference on Electronics Computer and Computation (ICECCO) (pp. 206-212). IEEE.

[6] Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., & Liu, H. (2019). Hand gesture recognition based on convolution neural network. Cluster Computing, 22(2), 2719-2729.

[7] Paolanti, M., & Frontoni, E. (2020). Multidisciplinary Pattern Recognition applications: A review. Computer Science Review, 37, 100276.

[8] Sarsenov, A., & Latuta, K. (2017, September). Face Recognition Based on Facial Landmarks. In 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-5). IEEE.

[9] Shdaifat, A., Obeidallah, R., Ghazal, G., Sarhan, A. A., & Spetan, N. A. (2020). A proposed Iris Recognition Model for Authentication in Mobile Exams. International Journal of Emerging Technologies in Learning (iJET), 15(12), 205-216.

[10] Mustafa, M. (2021). A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers. Journal of Ambient Intelligence and Humanized Computing, 12(3), 4101-4115.

[11] D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., ... & Sculley, D. (2020). Underspecification presents challenges for credibility in modern machine learning. arXiv preprint arXiv:2011.03395.

[12] Tofighi, G HandReader Dataset. https://github.com/tofighi/Hand-Reader-Dataset, 2012. [Online; accessed 01-March-2021].

[13] Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10(1), 131-153.

[14] Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, 119, 3-11.

[15] Nanivadekar, P. A., & Kulkarni, V. (2014, April). Indian sign language recognition: database creation, hand tracking and segmentation. In 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA) (pp. 358-363). IEEE.

[16] Rekha, J., Bhattacharya, J., & Majumder, S. (2011, December). Shape, texture and local movement hand gesture features for indian sign language recognition. In 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011) (pp. 30-35). IEEE.

[17] Turan, C. (2017, November). Robust face recognition via sparse reconstruction vector. In 2017 13th International Conference on Electronics, Computer and Computation (ICECCO) (pp. 1-4). IEEE.

[18] Aitimov, A., Turan, C., & Duisebekov, Z. Gesture Recognition Based on Sparse Reconstruction. In 2018 14th International Conference on Electronics Computer and Computation (ICECCO) (pp. 206-212). IEEE.

[19] Al-Shamayleh, A. S., Ahmad, R., Abushariah, M. A., Alam, K. A., & Jomhari, N. (2018). A systematic literature review on vision based gesture recognition techniques. Multimedia Tools and Applications, 77(21), 28121-28184.

[20] Rahmat, R. F., Chairunnisa, T. E. N. G. K. U., Gunawan, D. A. N. I., Pasha, M. F., & Budiarto, R. A. H. M. A. T. (2019). Hand gestures recognition with improved skin color segmentation in human-computer interaction applications. Journal of Theoretical and Applied Information Technology, 97(3), 727-739.

[21] Zhu, G., Zhang, L., Shen, P., Song, J., Shah, S. A. A., & Bennamoun, M. (2018). Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm. IEEE Transactions on Multimedia, 21(4), 1011-1021.

[22] Khurana, G., Joshi, G., & Kaur, J. (2014, March). Static hand gestures recognition system using shape-based features. In 2014 Recent Advances in Engineering and Computational Sciences (RAECS) (pp. 1-4). IEEE.

[23] Saxena, A., Jain, D. K., & Singhal, A. (2014, April). Sign language recognition using principal component analysis. In 2014 Fourth International Conference on Communication Systems and Network Technologies (pp. 810-813). IEEE.

[24] Ayshee, T. F., Raka, S. A., Hasib, Q. R., Hossain, M., & Rahman, R. M. (2014,

February). Fuzzy rule-based hand gesture recognition for bengali characters. In 2014 IEEE International Advance Computing Conference (IACC) (pp. 484-489). IEEE.

[25] Harris, C. G., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, No. 50, pp. 10-5244).

[26] Tofighi, G., Venetsanopoulos, A. N., Raahemifar, K., Beheshti, S., & Mohammadi, H. (2013, July). Hand posture recognition using K-NN and Support Vector Machine classifiers evaluated on our proposed HandReader dataset. In 2013 18th International Conference on Digital Signal Processing (DSP) (pp. 1-5). IEEE.

[27] Shawe-Taylor, J., & Cristianini, N. (2000). An introduction to support vector machines and other kernel-based learning methods (Vol. 204). Volume.

[28] Kataria, A., & Singh, M. D. (2013). A review of data classification using k-nearest neighbour algorithm. International Journal of Emerging Technology and Advanced Engineering, 3(6), 354-360.

[29] Kishore, P. V. V., Prasad, M. V., Prasad, C. R., & Rahul, R. (2015, January). 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN. In 2015 International Conference on Signal Processing and Communication Engineering Systems (pp. 34-38). IEEE.

[30] Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. Convolutional Neural Networks for Visual Recognition, 2, 225-232.

[31] Jin, Cheok Ming, Zaid Omar, and Mohamed Hisham Jaward. "A mobile application of American sign language translation via image processing algorithms." In 2016 IEEE Region 10 Symposium (TENSYMP), pp. 104-109. IEEE, 2016.

[32] Taskiran, M., Killioglu, M., & Kahraman, N. (2018, July). A real-time system for recognition of American sign language by using deep learning. In 2018 41st International Conference on Telecommunications and Signal Processing (TSP) (pp. 1-5). IEEE.

[33] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[34] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. IEEE transactions on pattern analysis and machine intelligence, 31(2), 210-227.

[35] Turan, C., Kadyrov, S., & Burissova, D. (2018, August). An Improved Face Recognition Algorithm Based on Sparse Representation. In 2018 International Conference on Computing and Network Communications (CoCoNet) (pp. 32-35). IEEE.

[36] Sankaranarayanan, S., Alavi, A., & Chellappa, R. (2016). Triplet similarity embedding for face verification. *arXiv preprint arXiv:1602.03418*.

[37] Perlibakas, V. (2004). Distance measures for PCA-based face recognition. Pattern recognition letters, 25(6), 711-724.

[38] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[39] Margalit, D., & Rabinoff, J. (2018). Interactive Linear Algebra. Georgia Institute of Technology.