Deep AutoShuffle Canonical Correlation Analysis

Rajesh Shrestha [1]

Abstract

Canonical Correlation Analysis (CCA) aims to find two linear projectors of two sets of variables so that they are maximally correlated. This idea has been extended to learn complex non-linear projectors using deep neural networks (DCCA) and great efforts have been made in developing deep neural networks based CCA for multiple view data too. In all these methods, an implicit assumption for same canonical correspondence is made. By loosening this assumption, we introduce Deep AutoShuffle Canonical Correlation Analysis (**DASCCA**), a method to learn shuffling of representation in addition to the non-linear transformations, aimed to make one shuffled representation highly correlated to another. The parameters for all the transformations and a shuffling are learned in an end-to-end training. In this paper, we experimentally investigate the advantages and pitfalls of our proposed model over DCCA.

1. Introduction

Canonical Correlation Analysis (CCA)(Hotelling, 1936) is a statistical multivariate method that aims to find linear transformation of two set of inputs so that the correlation between these two sets in the new lower dimension representation is maximised while maintaining uncorrelated components of representation of each set. These two sets of input variables are considered two views of the same entity. This methods only handles two views of data, however, various multi-view based CCA approaches have been proposed (Arora & Livescu, 2013; Livescu & Stoehr, 2009; Chaudhuri et al., 2009; Dhillon et al., 2011).

CCA uses linear projectors to find new representation and hence, is capable in finding the linear correlation between views of data, however, in practice most of the real-world data have non-linear relationships. CCA has been extended with the use of non-linear transformation using kernel based CCA (Akaho, 2006; Arora & Livescu, 2012) and neural network based CCA (Andrew et al., 2013). Kernel CCA is capable of finding non-linear transformation, however, with the fixed and non-parametric kernel, it's inflexible with new data points(Melzer et al., 2001). With the successful use of deep neural network in multiple tasks, it has been adopted here too which is parameterized and can be trained from end-to-end. These CCA based methods are able to find a new representation space such that it preserves the information that are common in multiple views and are resilient to the noise in either view as long they are uncorrelated.

This method has been successfully applied in a number of tasks such as multi-view recognition tasks, cross-model retrieval and classification, and multi-view embedding. The application ranges across number of disciplines in multiple applications such as in neuroscience((Friman et al., 2001; Zhuang et al., 2020), genomics(Witten & Tibshirani, 2009; Parkhomenko et al., 2009), biometrics(Xing et al., 2016), meterology(Cannon & Hsieh, 2008), speech recognition(Wang et al., 2015; Isobe et al., 2021), computer vision(Lisanti et al., 2014; Donner et al., 2006; Huang et al., 2010) and so on.

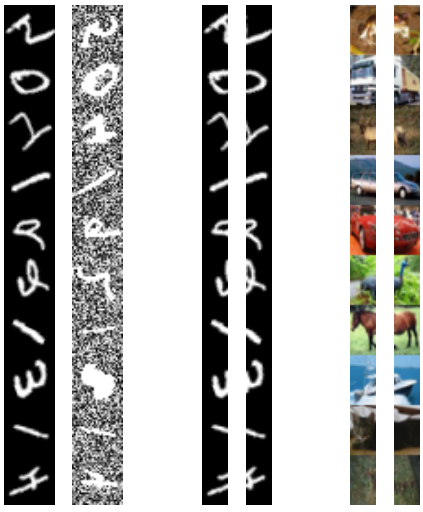

Figure 1. Two views data. The first pair corresponds to MNIST-noisy, the middle pair is the MNIST-split and the last one is the CIFAR10-split

[1]Department of Electrical Engineering and Computer Science, Oregon State University. Correspondence to: Rajesh Shrestha <shresthr@oregonstate.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

With these variants of CCA, each dimension of the learned feature(latent feature) of one view is considered to correspond to a corresponding feature of other views for all the data. In this paper, we propose a deep CCA method called Deep AutoShuffle Canonical Correlation Analysis (**DASCCA**) that loosens this assumption and allows any one dimension of the latent feature of one view correspond to any in another view for all the data. This shuffle of latent features is implemented using a permutation model as a shuffle of features corresponds to applying a permutation matrix (M) on it.

Based on the literature review we did, we didn't find any works similar to ours but our work is definitely motivated from a couple of them. In the following sections, we review related works, DCCA, introduce our proposed model, experiment setup and discuss the obtained results.

1.1. Related Works

CCA based models are highly dependent on the pairing of views of data and its performance is adversely affected by the misaligned pairs. Variation of CCA to make it alignment-agnostic (Sahbi, 2018) has been proposed. However, in our case, we assume having aligned data but embed the flexibility of latent feature shuffling into our model. Having shuffling mechanism within the model shows improvement in the model's performance of light weight convolutional neural network (Zhang et al., 2018; Ma et al., 2018). The permutation matrix corresponding to this shuffling can be learnt through training with a regularization in the network loss (Lyu et al., 2020).

2. Background

2.1. CCA

Assume two views $\mathbf{X}_1 \in \mathbb{R}^{m_1 \times n}$, $\mathbf{X}_2 \in \mathbb{R}^{m_2 \times n}$ with covariance matrices $(\mathbf{\Sigma}_{11}, \mathbf{\Sigma}_{22})$ and cross-covariance $\mathbf{\Sigma}_{12}$ between them. Let $\mathbf{w}_1 \in \mathbb{R}^{m_1}$ and $\mathbf{w_2} \in \mathbb{R}^{m_2}$ be two linear projectors that CCA aims to find such that $\mathbf{w_1 X_1}$ and $\mathbf{w_1 X_2}$ is maximally correlated.

$$(\mathbf{w}_1^*, \mathbf{w}_2^*) = \underset{(\mathbf{w}_1, \mathbf{w}_2)}{\arg\max}\ corr({\mathbf{w}_1}^T \mathbf{X}_1, {\mathbf{w}_2}^T \mathbf{X}_2) \tag{1}$$

$$= \underset{(\mathbf{w}_1, \mathbf{w}_2)}{\arg\max}\ \frac{{\mathbf{w}_1}^T \mathbf{\Sigma}_{12} \mathbf{w}_2}{\sqrt{{\mathbf{w}_1}^T \mathbf{\Sigma}_{11} \mathbf{w}_1 {\mathbf{w}_2}^T \mathbf{\Sigma}_{22} \mathbf{w}_2}} \tag{2}$$

With objective being invariant to scaling of $\mathbf{w}_1$ and $\mathbf{w}_2$, it can be rewritten as:

$$(\mathbf{w}_1^*, \mathbf{w}_2^*) = \underset{{\mathbf{w}_1}^T \mathbf{\Sigma}_{11} \mathbf{w}_1 = {\mathbf{w}_2}^T \Sigma_{22} \mathbf{w}_2 = 1}{\arg\max}\ {\mathbf{w}_1}^T \mathbf{\Sigma}_{12} \mathbf{w}_2 \tag{3}$$

For multiple transformation pairs $(\mathbf{w}_1^i, \mathbf{w}_2^i)$, each projection is constrained to be uncorrelated to others such that $(\mathbf{w}_1^i)^T \mathbf{\Sigma}_{11} \mathbf{w}_1^j = (\mathbf{w}_2^i)^T \mathbf{\Sigma}_{22} \mathbf{w}_2^j = 0$ for $i < j$. Combining $\mathbf{w}_1^i$ into columns to form $\mathbf{W_1} \in \mathbb{R}^{m_1 \times k}$ and similarly, $\mathbf{w}_2^{[i}$ into columns to form $\mathbf{W_2} \in \mathbb{R}^{m_2 \times k}$ where $k \leq \min(m_1, m_2)$. Then the problem statement changes to:

$$\begin{aligned} &\text{maximize: } tr(\mathbf{W}_1^T \mathbf{\Sigma}_{12} \mathbf{W}_2) \\ &\text{subject to: } \mathbf{W}_1^T \mathbf{\Sigma}_{11} \mathbf{W}_1 = \mathbf{W}_2^T \mathbf{\Sigma}_{22} \mathbf{W}_2 = I \end{aligned} \tag{4}$$

Let $\mathbf{T} = \mathbf{\Sigma}_{11}^{-\frac{1}{2}} \mathbf{\Sigma}_{12} \mathbf{\Sigma}_{22}^{-\frac{1}{2}}$ and $\mathbf{U_k}$, $\mathbf{V_k}$ be the matrices corresponding to the first k left and right singular vectors of $\mathbf{T}$. Then, optimal value is the sum of top k singular value of k and the optimal value is obtained at $(\mathbf{W}_1^*, \mathbf{W}_1^*) = (\mathbf{\Sigma}_{11}^{-\frac{1}{2}} \mathbf{U_k}, \mathbf{\Sigma}_{22}^{-\frac{1}{2}} \mathbf{V_k})$. Here, the $\mathbf{\Sigma}_{11}$ and $\mathbf{\Sigma}_{22}$ needs to be non-singular. We add a regularization parameter with centered data matrices $\overline{\mathbf{X}}_1$ and $\overline{\mathbf{X}}_2$ such that

$$\hat{\mathbf{\Sigma}}_{11} = \frac{1}{n-1} \overline{\mathbf{X}}_1 \overline{\mathbf{X}}_1^T + r_1 I \tag{5}$$

where $r_1 > 0$ is a regularization parameter and similarly other $\hat{\mathbf{\Sigma}}_{22}$ value too.

2.2. DCCA

Let $f_1(\mathbf{x}; \theta_1) : \mathbb{R}^{m_1} \to \mathbb{R}^o$ and $f_2(\mathbf{x}; \theta_2) : \mathbb{R}^{m_2} \to \mathbb{R}^o$ be two neural networks corresponding to first and second view respectively. Then,

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1, \theta_2)}{\arg\max}\ corr(f_1(\mathbf{X}_1; \theta_1), f_2(\mathbf{X}_2; \theta_2)) \tag{6}$$

Let $\mathbf{F}_1 \in \mathbb{R}^{o \times m}$ be the output of first network on $\mathbf{X}_1$ and $\mathbf{F}_2 \in \mathbb{R}^{o \times m}$ be the output of second network on $\mathbf{X}_2$. Then, let $\overline{\mathbf{F}}_1$ and $\overline{\mathbf{F}}_2$ denote the corresponding centered output data.

$$\bar{\mathbf{F}}_1 = \mathbf{F}_1 - \frac{1}{n} \mathbf{F}_1 \mathbf{1} \tag{7}$$

Similar to in CCA as in equation 5 in CCA, here the regularized covariances of the output can be estimated as

$$\hat{\mathbf{\Sigma}}_{11} = \frac{1}{n-1} \bar{\mathbf{F}}_1 \bar{\mathbf{F}}_1^T + r_1 \mathbf{I} \tag{8}$$

$$\hat{\mathbf{\Sigma}}_{22} = \frac{1}{n-1} \bar{\mathbf{F}}_2 \bar{\mathbf{F}}_2^T + r_2 \mathbf{I} \tag{9}$$

$$\hat{\mathbf{\Sigma}}_{12} = \frac{1}{n-1} \bar{\mathbf{F}}_1 \bar{\mathbf{F}}_2^T \tag{10}$$

where, r_1 and r_2 are regularization constants.

Like in CCA, let $\mathbf{T} = \hat{\mathbf{\Sigma}}_{11}^{-\frac{1}{2}} \hat{\mathbf{\Sigma}}_{12} \hat{\mathbf{\Sigma}}_{22}^{-\frac{1}{2}}$, then, the total correlation of top k components of outputs is the sum of top k singular values of $\mathbf{T}$. If $k = o$, then this will be equal to

$$corr(\mathbf{F}_1, \mathbf{F}_2) = tr(\mathbf{T}^T \mathbf{T})^{\frac{1}{2}} \tag{11}$$

The parameters (θ_1, θ_2) are initialized randomly and are updated using gradient descent to optimize the output correlation in equation 11.

2.3. Shuffle Network

Let $\mathbf{y} \in \mathbb{R}^N$, then the shuffling of the components of $\mathbf{y}$ can be represented as a multiplication by a permutation matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$. This permutation matrix $\mathbf{M}$ is a sparse matrix with exactly one entry equal to 1 and all other entries as 0 in every row and column. For a big value of M, it won't be feasible to look through every possible value of M. This permutation matrix ($\mathbf{M}$) can be learned by using l_{1-2} loss penalty (equation 12) in the standard gradient descent based training (Esser et al., 2013; Yin et al., 2015; Lyu et al., 2020).

$$l_{1-2}(\mathbf{M}) = \sum_{i=1}^{N}\left[\sum_{j=1}^{N}|\mathbf{M}_{ij}| - \left(\sum_{k=1}^{N}\mathbf{M}_{ik}^2\right)^{\frac{1}{2}}\right] + \sum_{j=1}^{N}\left[\sum_{i=1}^{N}|\mathbf{M}_{ij}| - \left(\sum_{k=1}^{N}\mathbf{M}_{kj}^2\right)^{\frac{1}{2}}\right] \tag{12}$$

$$\mathbf{M}_{ij} \geq 0 \; \forall(i,j); \sum_{i=1}^{N}\mathbf{M}_{ij} = 1 \;\; \forall j; \sum_{j=1}^{N}\mathbf{M}_{ij} = 1 \;\; \forall i; \tag{13}$$

Theorem: A square matrix $\mathbf{M}$ is a permutation matrix if and only if $l_{1-2}(\mathbf{M})$ and double stochastic constraint holds (constraint 13)

Proof:

(Necessary Condition $\Longrightarrow$) If $\mathbf{M}$ is a permutation matrix then, by definition all the constraints (13) are satisfied. Since only a single value in each and every row and column is equal to one with other values zero, the inner term inside the square brackets will be zero, hence, $l_{1-2} = 0$

(Sufficient Condition $\Longleftarrow$) By Cauchy-Schwartz inequality,

$$\sum_{j=1}^{N}|\mathbf{M}_{ij}| - \left(\sum_{k=1}^{N}\mathbf{M}_{ik}^2\right)^{\frac{1}{2}} \geq 0, \forall i \tag{14}$$

The equality in 14 holds if and only if there is only 1 non-zero value in every row of $\mathbf{M}$.

$$|j : \{\mathbf{M}_{ij} \neq 1\}| = 1 \tag{15}$$

With this and row-stochastic constraint, exactly one value in a row needs to be 1. With second term in equation 12, we obtain that $\mathbf{M}$ needs to have one column cardinality with exactly one value in a column to be 1. Hence, $\mathbf{M}$ needs to be a permutation matrix.

The non-negative constraint(13) in the gradient descent is implemented by clipping the entries: $\mathbf{M}_{ij} = \max(0, \mathbf{M}_{ij}), \;\; \forall(i,j)$. In theory, the row stochastic and column stochastic (constraint 13) can be obtained if $\mathbf{M}$ is alternatively row normalised and column normalized for sufficiently many times. However, in practice, a single alternative normalization in each iteration of gradient descent leads to approximately row and column stochastic matrix as training progresses. The obtained matrix results a relaxed shuffling which can be turned to an actual strict shuffling by rounding the obtained $\mathbf{M}$ to nearest permutation matrix.

3. Proposed Model (DASCCA)

The way our formulation differs from the DeepCCA is by the use of a permutation matrix to the second latent feature before applying the CCA objective (equation 6). Since, this permutation matrix is dependent on the input images, this is modeled with a separate shuffle neural network which takes the input images and outputs the permutation matrix for that.

Assume two views $\mathbf{X}_1 \in \mathbb{R}^{m_1 \times n}$, $\mathbf{X}_2 \in \mathbb{R}^{m_2 \times n}$ along with $f_1(\mathbf{x}; \theta_1) : \mathbb{R}^{m_1} \rightarrow \mathbb{R}^o$, $f_2(\mathbf{x}; \theta_2) : \mathbb{R}^{m_2} \rightarrow \mathbb{R}^o$ and $f_M(\mathbf{x}_1, \mathbf{x}_2; \theta_m) : \mathbb{R}^{m_1 \times m_2} \rightarrow \mathbb{R}^{o \times o}$ be neural networks corresponding to first, second view and the shuffle network respectively. For a second view $\mathbf{x}_2$, let f_o be the composite function of f_M and f_2 such that $f_o(\mathbf{x}_1, \mathbf{x}_2) = f_M(\mathbf{x}_1, \mathbf{x}_2) f_2(\mathbf{x}_2)$

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1, \theta_2)}{\arg\max} \;\; corr(f_1(\mathbf{X}_1; \theta_1), f_o(\mathbf{X}_1, \mathbf{X}_2; \theta_2)) \tag{16}$$

Then the output of f_1 and f_o are used to compute the correlation objective. Similar to before, let $\mathbf{F_1}$ be the matrix with columns being outputs of f_1, $\mathbf{F_2}$ with outputs of f_2, $\mathbf{F_M} \in \mathbb{R}^{n \times o \times o}$ with outputs of f_m and $\mathbf{F_o} \in \mathbb{R}^{o * m}$ be matrix with the output of the composite function $f_o(\mathbf{X}_1, \mathbf{X}_2)$, then the optimal correlation can be computed similar to DCCA(equation 11). This is used to train the parameters of the networks except an additional loss based on l_{1-2} is added that is used to regularize the output of the shuffle network to be a permutation matrix.

$$l(\mathbf{X}_1, \mathbf{X}_2; \theta_1, \theta_2, \theta_m) = -corr(\mathbf{F}_1, \mathbf{F_o}) + \lambda * l_M(\mathbf{F}_m) \tag{17}$$

where,

$$l_M(\mathbf{F}_m) = \frac{1}{n}\sum_{i=1}^{n} l_{1-2}(\mathbf{F}_m[i, :, :])$$

$\lambda =$ penalty hyperparameter for shuffle network

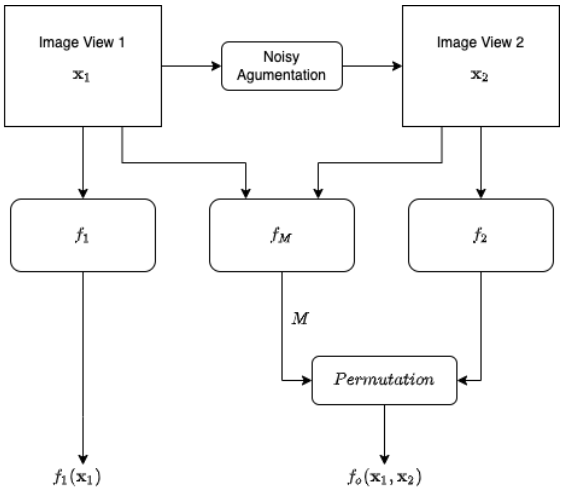

Figure 2. Block diagram of Proposed model (DASCCA)

4. Experiments

4.1. Experiment setup

The proposed model (**DASCCA**) along with **DCCA** was tested in two datasets: **MNIST** and **CIFAR10**. For the case of MNIST a standard dataset with noisy second view was present, so we used that for analyzing our model. Beside that the split view of the input images in both MNIST and CIFAR10 to obtain the second view which were were used for training the models. Here, we used the first left half of the image as the first view and second half view as the second view as shown in figure 1.

We used a simple linear model with 4 layers and having 1024 units for each hidden layer for both the features and the shuffle network. The hyperparameters like learning rate, regularization and penalty hyperparameters were tuned as per the dataset and the models. For our model, the regularization parameters r_1 and r_2 were set around 10^{-3} where as λ corresponding to the l_{1-2} penalty parameter was set to high value around 3 which drives the output of the shuffle network to a valid permutation matrix. Batch RMS prop was used as the optimizer while training the model and learning rate was adjusted based on how the loss progress with iterations.

4.2. Results

Upon training, with the heavy l_{1-2} penalty this loss seems to drastically decrease that leads to decrease in the overall loss. After certain iterations, the l_{1-2} loss becomes almost zero and the correlation gradually increases as more iteration progresses (figure 3. Both the train and validation loss seems to decrease showing the ability to generalize well (figure 4). This regularization is forcing the shuffle network to output approximately valid permutation matrix (figure 5).

Algorithm 1 Training of Proposed Model: DASCCA

Input:
Two view data: $\mathbf{X}_1, \mathbf{X}_2$
Shuffle Network Penalty Parameter: λ
Total Iteration: T_n
First View Network: f_1 with parameters θ_1
Second View Network: f_2 with parameters θ_2
Shuffle Network: f_M with parameters θ_M
Batch Size: n
Learning rate: η
Output: $\theta_1, \theta_2, \theta_M$
for $iteration = 1$ **to** T_n **do**
  for $batch = (\mathbf{X}_{b1}, \mathbf{X}_{b2}))$ **in** $(\mathbf{X}_1, \mathbf{X}_2)$ **do**
    $\mathbf{F}_1 = f_1(\mathbf{X}_{b1})$
    $\mathbf{F}_2 = f_2(\mathbf{X}_{b2})$
    $\mathbf{F}_M = f_M(\mathbf{X}_{b1}, \mathbf{X}_{b2})$
    $\mathbf{F}_M = \max(\mathbf{F}_M, 0)$
    Normalize each column of $\mathbf{F}_M$ corresponding to each batch data pair
    Normalize each rows of $\mathbf{F}_M$ corresponding to each batch data pair
    Compute $\mathbf{F}_o$ shuffled $\mathbf{F}_2$ using $\mathbf{F}_M$
    Normalize $\mathbf{F}_1$ and $\mathbf{F}_o$
    Compute optimal correlation $corr$ between them and $l_{1-2}(\mathbf{F}_M)$
    $l = -corr + l_{1-2}$
    $\theta_1 \leftarrow \theta_1 - \nabla_{\theta_1} l$
    $\theta_2 \leftarrow \theta_2 - \nabla_{\theta_2} l$
    $\theta_M \leftarrow \theta_M - \nabla_{\theta_M} l$
  end for
end for

Upon comparing the correlation on the test set of different datsets, our approach seems to work better than DCCA in the MNIST-noisy dataset when the output size K = 10. In other cases and datasets, the performance of DASCCA seems to be approximately similar to DSCCA(table 1).

Table 1. Correlation of DASCCA and DCCA for in test data of various data sets with output size(K)

DATA SET	DCCA		DASCCA	
	K=5	K=10	K=5	K=10
MNIST-NOISY	**4.75**	7.33	4.59	**8.78**
MNIST-SPLIT	**4.91**	**9.75**	4.83	9.58
CIFAR10-SPLIT	**4.51**	**8.6**	4.27	8.2

We also analyzed the feature outputs by these models and visualized using t-SNE. The obtained plots had quite a differences between these two models. DCCA seems to group in a single group with minimal surface area while DSCCA groups the same class into an elongated shape and

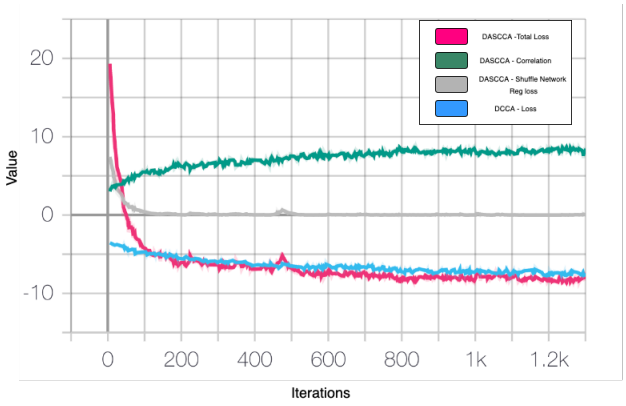

Figure 3. Batch loss for output size K = 10 in MNIST noisy dataset

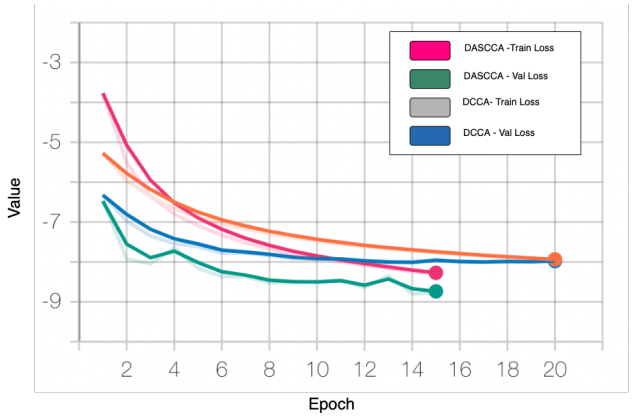

Figure 4. Epoch loss for output size K = 10 in MNIST noisy dataset

the learned features could be divided among multiple groups too.

5. Conclusion

We introduced a novel approach to CCA by loosening the correspondence of learned features. This was done with the use of shuffle network that learns the permutation of a feature and tries to maximize the correlation of the permuted feature with the feature of other view. With various experiment, we showed the differences of the features learned compare to DCCA. Our approach worked in one scenario and performed approximately similar in other scenarios. Based on the results we obtained, we speculate that our approach might have resilience to translation of patterns in the image hence worked well in the MNIST-noisy data but not on others. Further experimentation needs to be done on this side.

We also think this approach might be useful to find associ-

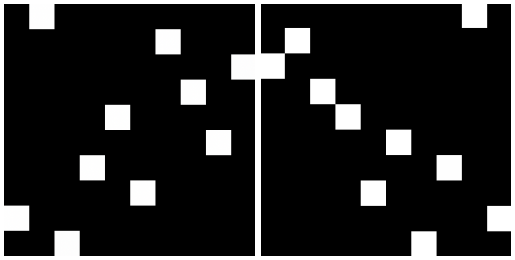

Figure 5. Learned permutation matrices for two sample pairs by DASCCA with output size K = 5

ation map between two images. This might be useful for landmark detection and matching among images. We know for landmark matching, the landmark needs to be detected first which is some pattern that is common in multiple views and shouldn't depend on the location of the image. CNN are good at finding features and maintain some degree of spatial information, although the level of precision decreases as the image is passed through more layers. We believe form of DCCA using CNN can be used to find this common information among multiple view images and with the use of this shuffle network without the linear layers as the last layer of feature networks, the association map could be learned. This in turn might be useful in narrowing down the scope for running the registration algorithm like SIFT(Lowe, 2004) or ORB(Rublee et al., 2011). All of these are just speculations and are reserved for future experimentation.

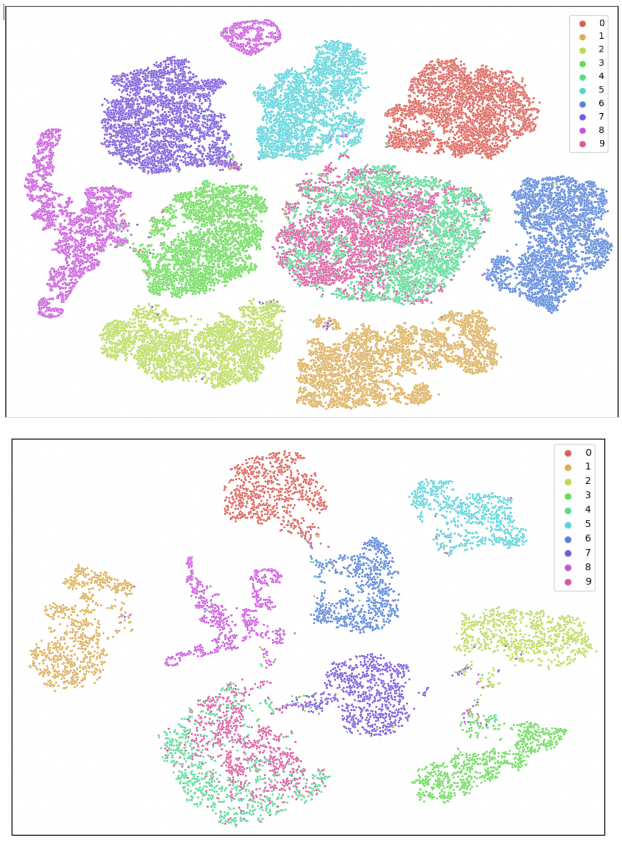

Figure 6. t-SNE plot of learned features of first view in **DCCA** when K=10. Upper one is for train and lower for test dataset

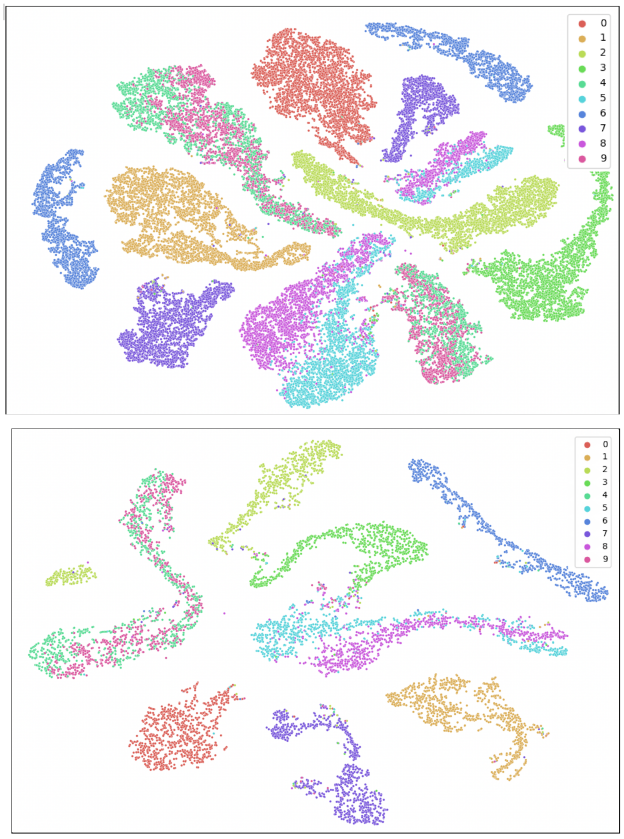

Figure 7. t-SNE plot of learned features of first view in **DASCCA** when K=10. Upper one is for train and lower for test dataset

References

Akaho, S. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Arora, R. and Livescu, K. Kernel cca for multi-view learning of acoustic features using articulatory measurements. In *Symposium on machine learning in speech and language processing*, 2012.

Arora, R. and Livescu, K. Multi-view cca-based acoustic features for phonetic recognition across speakers and domains. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7135–7139. IEEE, 2013.

Cannon, A. and Hsieh, W. Robust nonlinear canonical correlation analysis: application to seasonal climate forecasting. *Nonlinear Processes in Geophysics*, 15(1):221–232, 2008.

Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pp. 129–136, 2009.

Dhillon, P., Foster, D. P., and Ungar, L. Multi-view learning of word embeddings via cca. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/6c4b761a28b734fe93831e3fb400ce87-Paper.pdf.

Donner, R., Reiter, M., Langs, G., Peloschek, P., and Bischof, H. Fast active appearance model search using canonical correlation analysis. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1690–1694, 2006.

Esser, E., Lou, Y., and Xin, J. A method for finding structured sparse solutions to nonnegative least squares prob-

lems with applications. *SIAM Journal on Imaging Sciences*, 6(4):2010–2046, 2013.

Friman, O., Cedefamn, J., Lundberg, P., Borga, M., and Knutsson, H. Detection of neural activity in functional mri using canonical correlation analysis. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 45 (2):323–330, 2001.

Hotelling, H. Relations between two sets of variates*. *Biometrika*, 28(3-4):321–377, 12 1936. ISSN 0006-3444. doi: 10.1093/biomet/28.3-4.321. URL https://doi.org/10.1093/biomet/28.3-4.321.

Huang, H., He, H., Fan, X., and Zhang, J. Super-resolution of human face image using canonical correlation analysis. *Pattern Recognition*, 43(7):2532–2543, 2010.

Isobe, S., Tamura, S., and Hayamizu, S. Speech recognition using deep canonical correlation analysis in noisy environments. In *ICPRAM*, pp. 63–70, 2021.

Lisanti, G., Masi, I., and Del Bimbo, A. Matching people across camera views using kernel canonical correlation analysis. In *Proceedings of the International Conference on Distributed Smart Cameras*, pp. 1–6, 2014.

Livescu, K. and Stoehr, M. Multi-view learning of acoustic features for speaker recognition. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 82–86. IEEE, 2009.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60 (2):91–110, 2004.

Lyu, J., Zhang, S., Qi, Y., and Xin, J. Autoshufflenet: Learning permutation matrices via an exact lipschitz continuous penalty in deep convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 608–616, 2020.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Melzer, T., Reiter, M., and Bischof, H. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks*, pp. 353–360. Springer, 2001.

Parkhomenko, E., Tritchler, D., and Beyene, J. Sparse canonical correlation analysis with application to genomic data integration. *Statistical applications in genetics and molecular biology*, 8(1), 2009.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pp. 2564–2571. Ieee, 2011.

Sahbi, H. Canonical correlation analysis for misaligned satellite image change detection. *arXiv preprint arXiv:1812.09280*, 2018.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. A. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4590–4594. IEEE, 2015.

Witten, D. M. and Tibshirani, R. J. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1), 2009.

Xing, X., Wang, K., Yan, T., and Lv, Z. Complete canonical correlation analysis with application to multi-view gait recognition. *Pattern Recognition*, 50:107–117, 2016.

Yin, P., Lou, Y., He, Q., and Xin, J. Minimization of 1-2 for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

Zhuang, X., Yang, Z., and Cordes, D. A technical review of canonical correlation analysis for neuroscience applications. *Human Brain Mapping*, 41(13):3807–3833, 2020.