2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) | 978-1-7281-3079-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/FG47880.2020.00046

# Feature Selection for Zero-Shot Gesture Recognition

Naveen Madapana and Juan Wachs
School of Industrial Engineering, Purdue University, West Lafayette, USA.

***Abstract*— Existing classification techniques assign a pre-determined categorical label to each sample and can not recognize the new categories that might appear after the training stage. This limitation has led to the advent of new paradigms in machine learning such as zero-shot learning (ZSL). ZSL aims to recognize unseen categories by having a high-level description of them. While deep learning has pushed the limits of ZSL for object recognition, ZSL for temporal problems such as unfamiliar gesture recognition (ZSGL) remain unexplored. Previous attempts to address ZSGL were focused on the creation of gesture attributes, attribute-based datasets, and algorithmic improvements, and there is little or no research concerned with feature selection for ZSGL problems. It is indisputable that deep learning has obviated the need for feature engineering for the problems with large datasets. However, when the data is scarce, it is critical to leverage the domain information to create discriminative input features. The main goal of this work is to study the effect of three different feature extraction techniques (raw features, engineered features, and deep learning features) on the performance of ZSGL. Next, we propose a new approach for ZSGL that jointly minimizes the reconstruction loss, semantic and classification losses. Our methodology yields an unseen class accuracy of (38%) which parallels the accuracies obtained through state-of-the-art approaches.**

## I. Introduction

Gestures play a crucial role in human-human communications and while we interact with gaming consoles, smart devices and touch interfaces [1], [2]. Current gesture powered interfaces such as Microsoft HoloLens, PACS and Xbox consoles [3], [4] are constrained to a pre-determined set of hand gestures and can not adapt to the new gestures that users might prefer to use. This limitation has led to the advent of new paradigms in machine learning such as Zero shot learning (ZSL) [5], [6]. Figure 1 illustrates the problem of ZSL for unfamiliar gesture recognition.

ZSL aims to recognize the unseen object categories/classes by just having a high-level description of them [7], [8]. In other words, the trained ZSL models are expected to recognize the novel classes/gestures that were not present during the training period. ZSL is inspired by the way humans identify new species or children recognize unseen animals or new objects just by knowing their high-level properties/attributes such as color, shape, texture etc [9]. In a similar manner, ZSL relies on such attributes to transfer the knowledge gained from a finite set of seen classes to the categories that were never seen before [10].

This work is supported by the Agency for Healthcare Research and Quality (AHRQ), National Institute of Health (NIH) - under the Project No. 1R18HS024887-01. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by NIH.

ZSL is particularly beneficial in the scenarios where there are uncountable number of classes (as in the case of gestures) or when the data is extremely scarce for some classes while there is a plenty of data for others [11]. Furthermore, this paradigm has the potential to naturally adapt to the new classes without having to re-train the entire network from scratch to increase the number of classes [10]. Hence, it has been a hot topic in the machine learning community with a plethora of academic articles published every year pushing the limits of unseen class accuracies [12]–[14]. In this work, we propose a new bi-linear approach inspired from [15], [16] for ZSL that takes into account the reconstruction loss in addition to the classification and semantic losses.

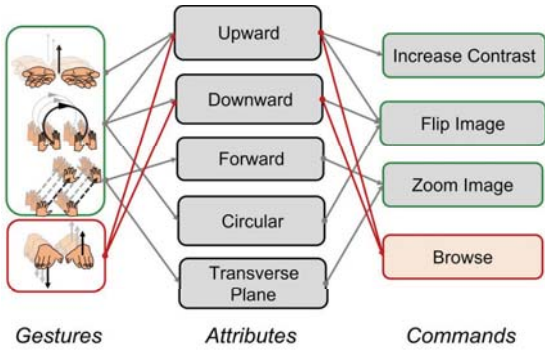


Fig. 1: An example of ZSL for gesture recognition (ZSGL). *Browse* is the new command appeared after training stage.

While deep learning techniques have pushed the limits of unseen object recognition, the problem of ZSL for gestures (ZSGL) remains unexplored. Feature selection (determining the best set of features) is the first important step towards improving the performance of ZSGL methods. However, previous works concerned with this problem have mainly focused on attributes and algorithmic improvements, and there is a lack of studies regarding the choice of features for ZSGL tasks [17], [18]. This work addresses this issue of feature selection and in addition, we conduct rigorous experiments to study the effect of several feature extraction techniques in the context of ZSGL.

The main contributions of this work are to: 1. Propose a new methodology for ZSGL that jointly optimizes classification and semantic losses, 2. Propose an approach to extract deep features and engineered features for ZSGL problems, and 3. Explore three feature extraction methods and perform experiments to compare and contrast them.

978-1-7281-3079-8/20/$31.00 ©2020 IEEE
DOI 10.1109/FG47880.2020.00046

## II. Related Work

Zero Shot Learning (ZSL) is a transfer learning paradigm in which the classes present in the training stage and testing stage are mutually exclusive [11], [19]. The survey conducted by Wang et al. details several transfer learning tasks (for e.g. domain adaptation, ZSL, etc.) and their relation to the existing machine learning approaches [11]. In contrast to regular classification, ZSL methods often suffer from the *domain-shift* problems due to disparate seen and unseen data distributions [20].

The central idea behind ZSL is to represent classes as a sequence of high-level properties or attributes or semantic descriptors. These descriptors act as the powerful facilitators of knowledge transfer between seen and unseen classes. Hence, the objective of ZSL is to first recognize the presence/absence of these attributes and thereby recognize the class label. Recent surveys conducted by Wang et al. [19] and Fu et al. [21] explained in detail the various kinds of attributes (for e.g. user-defined, word embedding or latent descriptions, etc.), existing methodologies and datasets for ZSL tasks.

It has been learnt that the problem of ZSL has been extensively studied in the static domains such as neural decoding [5], scene understanding [22] and animal/bird recognition [6]. Popular ZSL methods in object recognition include but not limited to DAP [10], ESZSL [16], SAE [15], ConSE [23], SynC [24] etc. The presence of several publicly available datasets (SUN [22], aPascal [6] and AwA [7]) and benchmarks has encouraged researchers to thoroughly investigate ZSL for object recognition. Hence, we have seen a consistent increase in the unseen class accuracies on AwA dataset from 57% in 2013 [7] to 86% in 2017 [15]. However, the problem of ZSL for temporal problems such as unfamiliar gesture recognition (ZSGL) is ill-defined and has hardly been investigated in the computer vision community. The lack of large-scale attribute-based datasets coupled with the intrinsic complexities associated with the temporal data makes ZSGL a particularly challenging problem.

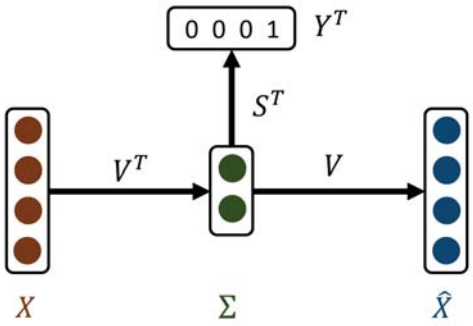


Fig. 2: Our proposed approach (SAE-CL) for ZSL.

Thomason et al. utilized word embeddings of gestures' function as an intermediate semantic representation to recognize a single held out zero-shot class and used human assessment to evaluate the performance of ZSGL [25]. Madapana et al. introduced the first attribute-based gesture dataset consisting of 26 categories and 34 attributes, and provided the benchmarks using popular ZSL methods [17], [18]. Further, Wu et al. constructed a dataset of hand gestures to control robots and utilized recurrent neural networks to create gesture features and thereby recognize the untrained gestures using SAE [26]. Overall, the previous attempts to address ZSGL had focused on creating a database of gesture attributes, datasets and algorithmic improvements. However, there is a lack of research or studies concerned with determining the best feature representations for ZSGL problems.

## III. Methodology

Let us start by defining the notations. Let $\mathcal{S}$ be a set of training (seen) classes, $\mathcal{U}$ be a set of unseen classes, $z_s$ be the number of seen classes and $z_u$ be the number of unseen classes. Let $a$ be the number of attributes. Note that $\mathcal{S}$ and $\mathcal{U}$ share the attribute space i.e. they have equal number of attributes. Let $m_s$ be the number of instances in seen/training data. For ZSL, note that $\mathcal{S} \cap \mathcal{U} = \phi$. Let $X \in \mathbb{R}^{d \times m_s}$ be the input feature matrix, $Y \in \{-1, 1\}^{m_s \times z_s}$ be the ground truth labels, $S \in [0, 1]^{a \times z_s}$ and $\Sigma \in [0, 1]^{m_s \times a}$ be the per-class and per-instance semantic descriptions of the seen classes respectively. Let $V \in \mathbb{R}^{d \times a}$ be the weight matrix learned using our SAE-CL approach (Fig. 2). Without the loss of generality, it is assumed that each frame of a gesture sample is represented by a fixed one dimensional vector, and the samples are allowed to have varying number of frames. In other words, a gesture instance would be of size $(M \times T)$, where $T$ is the number of frames in the instance.

### A. SAE with Classification Loss (SAE-CL)

An encoder-decoder paradigm similar to Kodriov et al. [15] was used in this work to model the ZSL problem. In addition to the reconstruction loss, we incorporated the semantic and the classification errors into the loss function. This forces the learned weights to jointly minimize the semantic loss while clustering instances belonging to the same class together.

*1) Reconstruction Loss:* The encoder maps the feature vectors to a latent space while the decoder re-maps the latent vectors back to the feature space. This loss ensures that the learned latent space representations contain enough information to reconstruct the original feature vectors.

$$L_r = \left\|X - VV^T X\right\|^2 \quad s.t \quad V^T X = \Sigma$$
$$\Rightarrow L_r = \left\|X - V\ \Sigma\right\|^2 \tag{1}$$

*2) Semantic-Classification loss:* The latent representations learnt by the auto encoder should be forced to have the semantic meaning as the class attributes are obtained through human annotations. Hence it is essential to add a component to the loss function that simultaneously optimizes for semantic and classification losses.

$$L_{sc} = \left\|X^T\ W - Y\right\|^2 \quad where \quad W = V\ S$$
$$L_{sc} = \left\|X^T\ V\ S - Y\right\|^2 \tag{2}$$

*3) Overall Loss:* The overall loss is nothing but the weighted sum of $L_r$, $L_{sc}$ and the regularization term.

$$L = L_{sc} + \alpha L_r + \beta \|V\|^2 \quad (3)$$

$$L = \|X - V\ \Sigma\|^2 + \alpha \left\|X^T\ V\ S - Y\right\|^2 + \beta \|V\|^2 \quad (4)$$

Optimizing the loss function given in Equation 4 i.e. $\partial L/\partial V^T = 0$ yields a solvable Sylvester Equation 5 of form $AV+VB = C$. Sylvester equation can be easily solved using linear algebra packages of Matlab or NumPy [27]. Note that $\alpha$ and $\beta$ are the hyperparameters.

$$\Rightarrow XX^TV + V\beta(SS^T + \alpha I)^{-1} = (XYS^T + \alpha X\Sigma^T)\ldots (SS^T + \alpha I)^{-1} \quad (5)$$

TABLE I: List of engineered features. Note that $x$, $y$ and $z$ indicate the Cartesian axes. $+$ and $-$ indicate the positive and negative values of the first derivative. This acts as the 28-dimensional feature vector (ordered using rastor scanning).

| | | | | | | |
|---|---|---|---|---|---|---|
| $A_x^+$ | $A_y^+$ | $A_z^+$ | $A_x^+A_z^+$ | $A_x^+A_y^+$ | $A_y^+A_z^+$ | $A_x^+A_y^+A_z^+$ |
| $A_x^-$ | $A_y^-$ | $A_z^-$ | $A_x^-A_z^-$ | $A_x^-A_y^-$ | $A_y^-A_z^-$ | $A_x^-A_y^-A_z^-$ |
| $R_x$ | $R_y$ | $R_z$ | $R_xR_z$ | $R_xR_y$ | $R_yR_z$ | $R_xR_yR_z$ |
| $M_x$ | $M_y$ | $M_z$ | $M_xM_z$ | $M_xM_y$ | $M_yM_z$ | $M_xM_yM_z$ |

### B. Feature Extraction for Gestures

The main goal of this part is to find out the feature representation that yields best performance on ZSGL tasks. For simplicity, the coordinates of both the left and right hand of the gesture performer with respect to the shoulder were used as the initial set of raw features. Three feature extraction methods were explored in this paper, namely, 1. Sampled raw features, 2. Engineered features and 3. Deep features.

*1) Sampled raw features:* This method uses a trivial concatenation of raw features of every frame to obtain a final feature vector. First, each gesture instance was re-sampled to a fixed number of frames using interpolation techniques and then, the raw features corresponding to each frame were concatenated against each other. Each instance was sampled to 10 frames and each frame was represented as a six-dimensional vector (3 for each hand - first derivative of 3D position of left and right hands). Overall, this method resulted in a 60-dimensional feature vector.

*2) Engineered Features:* In this approach, features were manually designed based on the prior knowledge about the gesture recognition task and the semantic descriptors. For instance, our descriptors include direction of motion (leftward, upward, etc.). Hence the principal directions on Cartesian plane (x-axis, y-axis and z-axis) were used to extract time-independent features. This method maps the variable length instances to a fixed length features. Initially, first derivative of the trajectory of both the left and right hands was computed along x, y, z directions and then, the features depicted in Table I were computed. $A_x^+$ was the sum of positive values along x-axis for both the hands, $A_x^-$ was the sum of negative values along x-axis for both the hands. $R_x$ was the range of motion along x-axis computed as the difference between the maximum and minimum values for both the hands. $M_x$ was the mean of all values along x-axis for both the hands. Similar method was followed to compute the features along other directions. Finally, the polynomial features were computed for each feature type as shown in Table I. Overall, each gesture instance was represented as a 28-dimensional vector.

*3) Deep features:* In this method, a trained bi-directional LSTM (BLSTM) shown in the Figure 3 was used as a feature extractor. First, a BLSTM was trained on CGD 2013 dataset [28] consisting of 18 gesture classes. Each gesture class consisted of approximately 240 samples (depends on the class label) and each sample has varying number of skeleton frames. Similar to the previous approaches, relative 3D position of both the hands w.r.t the shoulder was used to train the BLSTM. The last layer of a trained BLSTM was used as a feature vector. If the hidden size of BLSTM is 32, this method resulted in a 64-dimensional feature vector.

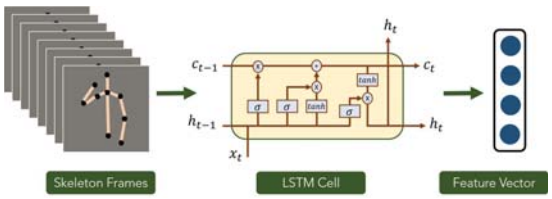


Fig. 3: BLSTM Model for feature extraction.

## IV. Experiments and Results

The gesture attribute dataset developed by Madapana et al. was used in our experiments to validate our methodology [18]. This dataset consisted of 26 gesture classes: 18 from CGD 2013 dataset [28] and 8 from MSRC-12 dataset [29]. Each gesture class in CGD dataset has approximately 400 examples while each class in MSRC dataset has 600 examples. Segmented skeletal data was available in each of these datasets. Further, there were 34 binary gesture attributes available in their dataset. These gesture attributes comprised of direction of motion for both the hands, plane of motion for both the hands, part of the body referred to and average position of the overall gesture. The meaning or the function of these gesture classes do not alter when the hands were interchanged. Hence, the gesture attributes of left and right hands were combined to obtain a reduced set of 22 attributes. The semantic description matrix of these 26 categories w.r.t the 22 attributes was depicted as a binary image in Figure 4. Note that a value of *zero* (dark color) indicates the absence of an attribute while a value of *one* (white color) indicates the presence of an attribute.

The next step in the pipeline was to compute the features following the three methods described in section III-B. Overall, the gesture instances were represented as a 60, 28 and 64 dimensional feature vectors. For the *deep features* approach, a bi-directional LSTM (BLSTM) was trained with the skeletal features of both hands (3D coordinates w.r.t the shoulder) to classify the seen classes accurately. A five-fold cross validation procedure was used to determine the hyperparameters of the BLSTM model. The final model consisted of two LSTM cells, the hidden layer size of 32

and a dropout probability of 0.5. The trained model yielded a classification accuracy of 80% - 90% (varies depending on the gestures present in the seen classes) on the test set. Such high classification accuracies indicate that the features generated from the trained BLSTM model are discriminative and representative of the gesture classes. Hence this trained model was used to extract features for all of the 26 gesture categories.

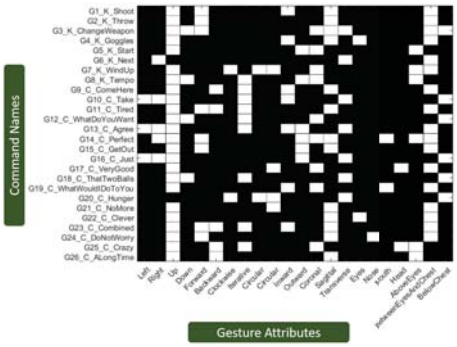


Fig. 4: Semantic description matrix of the gestures. Rows represent gesture classes and columns represent attributes.

Further, the approach proposed in this paper (SAE-CL) was compared against three popular ZSL approaches, namely, 1. DAP [10], 2. ESZSL [16] and 3. SAE [15]. In DAP, a binary SVM classifier was trained per each attribute resulting in 22 classifiers. The congregation of these classifiers was used to recognize the unseen classes. We followed the experimental protocol proposed by Madapana et al. to split the dataset into seen and unseen classes [18]. However, we considered 20 random seen-unseen class splits, and mean and standard deviation of the accuracies were reported. 80 % of the classes (21/26) were considered for training and 20 % (5/26) of the classes were considered for testing stage.

Data imbalance is a very critical issue in ZSGL tasks. In the Figure 4, it can be noticed clearly that there is a significant amount of data imbalance at an attribute level (note the sparsity of the semantic description matrix). In the ideal scenario, we want each attribute to be present in half of the seen classes and absent in the other half. However, in this dataset, some attributes are either present or absent for most of the classes. Such attributes are very difficult to learn and appropriate techniques such as data augmentation should be used to account for data imbalance. In addition to the data augmentation, we have re-structured the mis-classification costs of DAP for the majority and minority classes to inherently handle the issues related to the data imbalance. The class priors (probability) were used to compute the mis-classification costs. Let $p$ be the probability of encountering an example from the minority class. The error corresponding to the minority and majority classes were punished with a weight of $(1-p)/p$ and unity respectively.

The unseen class accuracies obtained using four ZSL approaches were summarized in the Table II. There are several ways of designing the features leveraging the domain information. In this work, we presented a particular set of engineered features to study how well they perform in relation to deep learning features. Overall, it was found that engineered features outperformed other feature extraction techniques with a significant margin which was confirmed by a paired t-test ($p < 0.05$). Our approach obtained an accuracy of $38.1 \pm 7.3\%$ for engineered features i.e. $38\%$ of the gesture samples belonging to the unseen classes were accurately classified.

Further, SAE-CL approach performed slightly better than SAE for engineered features while SAE outperforms our method marginally for other feature types. Note that the standard deviation of the accuracies is considerably high indicating that unseen class accuracies majorly depend on the seen-unseen class splits. When the class splits are made, it is quite possible that there might be some attributes that are present in only one class making it very difficult for the learning algorithm to predict that attribute. Hence, it is crucial to have large attribute-based datasets in order to alleviate the problem of data imbalance.

Though deep learning is known for learning features automatically, we found that the features obtained from a pre-trained BLSTM were not effective at ZSGL tasks. This was partly due to the fact that we have limited number of classes and training data. However, with the increase in the size of the dataset, we expect the deep learning features to perform better than other features.

TABLE II: Comparison of unseen accuracy of several ZSL approaches (%). Columns indicate the feature extraction techniques. Third column refers to the engineered features.

| | Raw | Deep | Eng. |
|---|---|---|---|
| DAP | 26.7 ± 7.2 | 29.6 ± 9.04 | **37.6** ± 11.1 |
| ESZSL | 21.8 ± 9.96 | 19.6 ± 3.7 | 19.8 ± 10.1 |
| SAE | **33.76** ± 11.2 | 22.85 ± 7.3 | **36.2** ± 9.2 |
| **SAE-CL*** | **30.6** ± 10.2 | 23.5 ± 8.7 | **38.1** ± 7.3 |

## V. Conclusions

Deep learning has greatly pushed the limits of ZSL for object recognition due to the presence of large-scale attribute datasets. However, the temporal problems such as ZSGL were unexplored and had hardly been studied in the computer vision research. Moreover, it is indisputable that the deep learning methods are extremely capable of learning features from the data. Nevertheless, when the data is scarce as in the case of ZSGL problems, it is critical to utilize the domain knowledge to create the discriminative features to achieve superior accuracies. In this regard, we explore three feature extraction techniques, namely, raw features engineered features and deep learning features, and conducted experiments to compare unseen class accuracies obtained through these approaches. It was found that engineered features perform significantly better than deep learning features due to the fact that there is little data to learn from. Moreover, we proposed a new bi-linear auto-encoder approach for ZSL that jointly optimizes reconstruction and classification error. Results show that the accuracies obtained via our approach parallel the ones obtained using state-of-the-art approaches.

## REFERENCES


[1] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, and S. Vickers, "Designing Gaze Gestures for Gaming: An Investigation of Performance," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, (New York, NY, USA), pp. 323–330, ACM, 2010.

[2] Y. Wang, T. Yu, L. Shi, and Z. Li, "Using human body gestures as inputs for gaming via depth analysis," in *2008 IEEE International Conference on Multimedia and Expo*, pp. 993–996, June 2008.

[3] N. Madapana, G. Gonzalez, R. Rodgers, L. Zhang, and J. P. Wachs, "Gestures for picture archiving and communication systems (pacs) operation in the operating room: Is there any standard?," *PLOS ONE*, vol. 13, pp. 1–13, 06 2018.

[4] "Microsoft hololens 2: https://www.microsoft.com/en-us/hololens/."

[5] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot Learning with Semantic Output Codes," in *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 1410–1418, Curran Associates, Inc., 2009.

[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, June 2009.

[7] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, June 2009.

[8] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-Shot Learning Through Cross-Modal Transfer," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 935–943, Curran Associates, Inc., 2013.

[9] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 594–611, Apr. 2006.

[10] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-Based Classification for Zero-Shot Visual Object Categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 453–465, Mar. 2014.

[11] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, Oct. 2010.

[12] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[13] S. Rahman, S. H. Khan, and F. Porikli, "A Unified approach for Conventional Zero-shot, Generalized Zero-shot and Few-shot Learning," *arXiv:1706.08653 [cs]*, June 2017. arXiv: 1706.08653.

[14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive Multi-view Zero-Shot Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 2332–2345, Nov. 2015. arXiv: 1501.04560.

[15] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," *arXiv preprint arXiv:1704.08345*, 2017.

[16] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 2152–2161, PMLR, July 2015.

[17] N. Madapana and J. Wachs, "Zsgl: Zero shot gestural learning," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI 2017, (New York, NY, USA), pp. 331–335, ACM, 2017.

[18] N. Madapana and J. Wachs, "Database of gesture attributes: Zero shot learning for gesture recognition," in *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pp. 1–8, May 2019.

[19] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 13, 2019.

[20] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proceedings of the IEEE international conference on computer vision*, pp. 2452–2460, 2015.

[21] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 112–125, 2018.

[22] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758, June 2012.

[23] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," *arXiv preprint arXiv:1312.5650*, 2013.

[24] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5327–5336, 2016.

[25] W. Thomason and R. Knepper, "Recognizing Unfamiliar Gestures for Human-Robot Interaction through Zero-Shot Learning," 2016.

[26] J. Wu, K. Li, X. Zhao, and M. Tan, "Unfamiliar dynamic hand gestures recognition based on zero-shot learning," in *Neural Information Processing* (L. Cheng, A. C. S. Leung, and S. Ozawa, eds.), (Cham), pp. 244–254, Springer International Publishing, 2018.

[27] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax + xb = c [f4]," *Commun. ACM*, vol. 15, pp. 820–826, Sept. 1972.

[28] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 445–452, ACM, 2013.

[29] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing People for Training Gestural Interactive Systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 1737–1746, ACM, 2012.