Skeleton based Zero Shot Action Recognition using a Learnable Distance Metric in Joint Pose Language Semantic Space

Bhavan Jasani
Robotics Institute
Carnegie Mellon University
bjasani@andrew.cmu.edu

Afshaan Mazagonwalla
Electrical and Computer Engineering
Carnegie Mellon University
amazagon@andrew.cmu.edu

Abstract

How does one represent an action ? How does one describe an action that we have never seen before ? Such questions are addressed by the Zero Shot Learning paradigm, where a model is trained on only a subset of classes and is evaluated on its ability to correctly classify an example from a class it has never seen before. In this work, we present a Pose based Zero Shot Action Recognition System and demonstrate its performance on the NTU-RGB-D dataset in the Cross View setting. We explore the significance of using pose information as the ideal representation for understanding actions in videos. In particular, we believe that learning human dynamics in a context free setting (i.e. just focusing on the action performer and not the context - the surroundings) allows the model to capture more information within the visual domain so that this information can be transferred easily to visually similar classes unseen during training.

1. Introduction

Most of the current approaches for action recognition require large well labelled datasets and work only on the action classes the model is trained on. Newer and newer datasets for action recognition are being released, and the number of action categories keeps on increasing. The action categories may thus be extremely fine grained and the models work well on the predefined action categories but fail to generalize when given an example of a category outside of the training set. At the same time it is difficult to get sufficient training data for new actions. This is what motivates our work of zero-shot learning for action recognition. Further, there has been extensive work on zero-shot image classification but this is very limited in the action recognition community, and so through this paper we want to explore that direction.

In Zero-Shot Learning (ZSL) the visual model is trained with visual data from a subset of the available action classes called the seen classes and it subsequently learns to generalize to previously unseen classes with the help of some external information contained in some other modality or sources of data that are easily available. Majority of previous work in zero-shot learning use text in the form of word embeddings as the external knowledge base, as unannotated text data is easily available. This is similar to the way humans can indirectly learn about novel things just by reading the description of an image or video without even looking at it along with the knowledge we already have about the things we have seen.

A large text corpus is used to generate word embeddings like word2vec[9] in a self-supervised way using for the action classes for both the seen and unseen classes. These word embeddings contain information about both the seen and unseen classes and then one would want to learn the projection between the visual features and text features, so that given visual features of unseen data, the model could find the most relevant text features.

Most of the top performing action recognition models use context i.e. the details of the surrounding and not just the action performer.

But we hypothesize that context can bias the model into learning from unrelated pixels making the learning of unseen actions difficult in a zero-shot setting. Also recent deep learning based human body pose detectors like OpenPose [2] work really well even in diverse conditions. So we focus in this work on skeleton based action recognition where just using body pose can be sufficient for learning actions in a zero shot setting. For example, if the model has seen examples of a person wearing jacket (based on body pose only), then it would easier to infer the person removing jacket based on just body pose even if that was not one of the original categories available during training.

In this work, we explore the significance of using pose information as the ideal representation for understanding actions in videos. In particular, we believe that learning human dynamics in a context free setting (i.e. just focusing

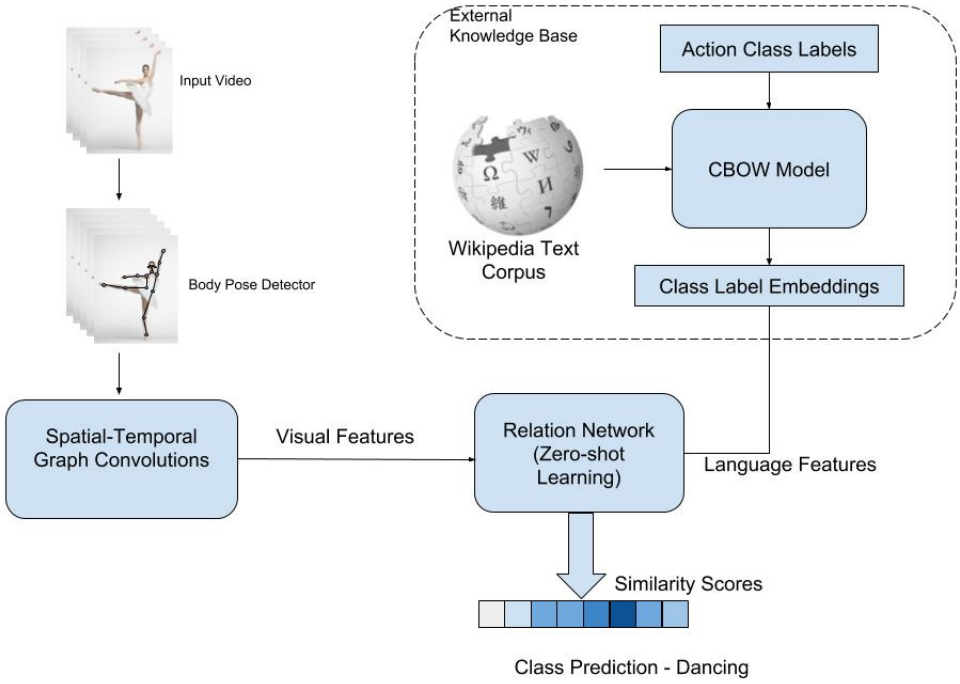

Figure 1: Our Model

This figure illustrates the flow of our model. The visual representations are obtained from an ST-GCN and the class name embeddings are obtained from a CBOW Model. The relational network learns a distance metric from both these modalities.

on the action performer and not the context - the surroundings) allows the model to capture more information within the visual domain so that this information can be transferred easily to visually similar classes not seen during training.

2. Related works

Our work builds on research in the domain of Action Recognition and Zero Shot Learning applied to videos.

Action Recognition: Within the Action Recognition community, there has been extensive research towards finding an ideal video representation that captures the salient features of a video sample in order to classify it as one of the available categories.

A variety of methods have been applied to this effect including Two-stream Networks [16] that jointly learn spatial and temporal features of videos, 3D Convolutions [6] that apply a learnable convolutional kernel directly over the RGB video frames across time, optical flow based methods [20] that capture frame level motion dynamics and Pose Based Methods [3] that directly model the dynamics of body keypoints.

The current datasets available for action recognition are highly variable in terms of the kinds of actions they represent. Certain datasets such as UCF101 [17], NTU RGB-D [13] contain videos captured in a controlled indoor / outdoor setting, where the video samples are centered, where as datasets such as Kinetics [7] contain examples from Youtube videos and therefore contain large intra-class variabilty as well as large camera motion.

The choice of algorithm used for Action Recognition, therefore depends largely on the dataset used. In this work, we wish to explore the role of visual context in allowing the model to perform well in a Zero Shot Action Recognition setting.

Pose Based Action Recognition: There have been a studies in Human Vision [1] that also suggest that Human Pose can provide sufficient information for determining actions, especially for human activity recognition.

Also, He *et al*. [5] found that visual context can inadvertently bias the model to predict the correct action class for Human Action Recognition based on the surrounding pixels even when the human is not present in the video. This is a challenge for classification when multiple actions are performed in the same surroundings, and can also cause the model to easily misclassify actions when the pixel intensity profiles of two dissimilar action classes appear to be similar.

Based on these reasons, we believe that human pose is the ideal visual representation in a zero shot setting and we focus our discussion on some recent approaches that use pose features for action recognition.

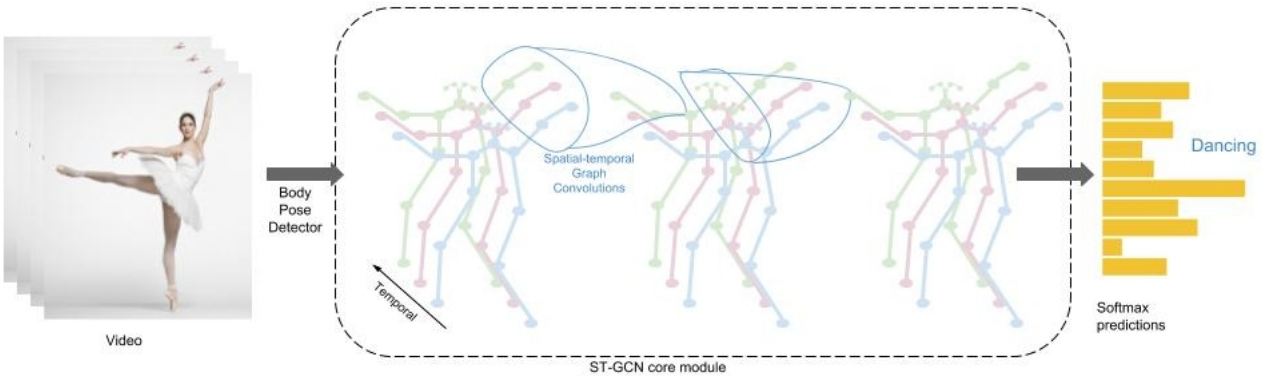

Figure 2: ST-GCN model

A pose is defined a set of keypoints per frame denoting a fixed number of body joint locations. These locations may be manually annotated in the dataset or may be obtained using Pose Estimation Algorithms such as [2] In Pose based action recognition, the video is represented as a sequence of poses, i.e a sequence of a set of 2D or 3D coordinates. Previous methods in pose based action recognition like [19] use rule based parsing techniques to manually group body keypoints based on rules of human anatomy.

Recent architectures such and ST-GCN [21] and SR-TSL [15] use graph convolutional networks and other end to end trainable architectures that allow for these representations to be learnt purely from the available data by jointly optimizing parameters of the representation network and the classification network.

Zero Shot Action Recognition: The focus of research of the zero shot recognition community is in finding the right semantic space in which to project visual features so that similar classes appear close together and dissimilar classes appear further apart.

A bulk of this work is of the form where visual features are projected onto the language space. One of the early works to demonstrate this idea was DeViSE [4] which uses a simple learned linear projection between the visual feature space and the class name embeddings. The model can then assign a class label to the unseen example using a fixed nearest neighbor or linear classifier.

In recent times, there have been a number of interesting approaches to solving the zero shot learning problem using Error Correcting Codes [12], Generative Models [10] and using a Learned Distance Metric [18].

While most other works use a fixed distance metric with which to assign a nearest neighbor class to an unseen example, in Learning to Compare (LTC) [18] the authors tackle a zero shot problem by learning the metric as an optimization problem based on the training data.

3. Approach

Our zero shot model is modular and can be divided into three parts:
1) Skeleton based action recognition module
2) External language knowledge base
3) Zero shot learning module

We use Spatial Temporal Graph Convolution Network for skeleton based action recognition [21] which acts as the visual feature extractor. We use Sent2Vec [11] trained on a huge English Wikipedia text corpus to generate class label embeddings, which provides our model with external knowledge. And Learning to Compare: Relation Network for Few-Shot Learning [18] is used for jointly learning the visual features and the external language based knowledge base.

The input to our zero-shot action recognition model is the time series of body pose keypoints processed from the raw RGB video frames by a body pose detector like OpenPose [2]. This goes as input to ST-GCN [21] which applies multiple layers of spatial-temporal graph convolutions and eventually fully connected layer with softmax layer to get probability distribution over the different action class labels.

We first train the (STGCN) [21] on a subset of the available action class data (the seen classes) and then use the trained model to extract the visual features for all the seen classes training data. In parallel, we obtain the languge embeddings of all the action class labels (seen + unseen) using Sent2Vec [11] pre-trained on 69 million English sentences from Wikipedia texts.

Using the visual features of training examples of seen classes from ST-GCN and class label embeddings from Sent2Vec we then train a zero shot model which learns to match the visual features with the class label embeddings (which encodes the external knowledge about all the classes, seen as well as unseen). We use Relational network [18] for doing zero-shot learning in our model and as

a baseline also experiment with DeViSE [4].

During test time, we first pass the videos of unseen action class through STGCN to compute it's visual features, which are then passed as inputs to DeViSE or Relational network which finds the nearest relevant class label embedding corresponding to the visual features and hence provides the output class label for the unseen action. We test our model in both the zero-shot learning and generalized zero-shot learning setting on the unseen test examples described in the experimental section.

Each part of our model are explained in greater detail in the following sections:

3.1. Skeleton based action recognition

We use Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition (ST-GCN) [21] as our visual feature extractor. This provides a good visual feature for our hypothesis that skeleton dynamics provides good visual representation for generalization to zero-shot learning.

ST-GCN takes as input the time series of body pose of the given video. Given a video first a human body pose detector, OpenPose [2] is used to generate the time series of body key points. ST-GCN works on both 2-D and 3-D body key points. Human actions can be recognized by the dynamics of skeleton. Since skeletons are in the form of a graph this work uses Graph Convolutions Networks which generalize normal Convolutional Neural Networks to work on graphs of arbitrary structures. Further since the input is time series of skeleton, it uses spatial-temporal graph convolutions.

The input to the ST-GCN is a graph where each node represents a body joint, and there are two type of edges - the spatial edges resembles the natural connections in human bodies and the temporal edges connect every body joint to itself across the temporal sequence. ST-GCN then applies multiple layers of spatial-temporal convolutions on the neighbouring spatial and temporal nodes in the input graph just like CNN's. This results into hierarchical higher-level feature representations similar to CNN's. This way ST-GCN learns to hierarchically capture the spatial and temporal dynamics of human body movements. Eventually after multiple layers of graph convolutions and pooling, a soft-max layer is applied which gives probability distribution for the corresponding action categories. This model can be trained in an end-to-end manner.

In our pipeline, we train the ST-GCN model on only the videos of the seen classes. During test time we use this pre-trained model to extract visual features from the penultimate layer before the soft-max for both videos of seen and unseen classes. These features are analogous to the fc7 features of AlexNet [8] for images, but here these features instead represent the spatial-temporal dynamics of the humans in the videos.

3.2. External language based knowledge base

For most of the action recognition datasets the class labels are in the form of phrases (*e.g. wearing the jacket, taking-off the jacket*) instead of single words and so we use Sent2Vec [11] for generating class label embeddings.

The Sent2Vec model is an extension of the Continuous Bag of Words (CBOW) model for word contexts to a larger sentence context that uses an unsupervised objective to train distributed representations of sentences from a large corpus of text data.

For our task, we generate bigram embeddings trained on a 16GB corpus of English Wikipedia texts, which contains about 69 million sentences and about 1.7 billion words. The resulting class label embeddings are a vector of size 700 dimension.

This allows our zero-shot model to learn the meaning and language semantics of various action class labels (of both seen and unseen) from naturally occurring text data and therefore serves as the external knowledge base.

An important thing to mention is similar actions will be closer in the semantic space. For example "walking" and "running" will have higher cosine similarity of it's embedding in comparison with that of "wearing a jacket". And so even if our action recognition model has only seen visual examples of "running", based on these language embeddings it will know that "walking" is a similar to "running". Which eventually would help it make correct prediction when videos of unseen action classes are given.

3.3. Zero shot learning

We use Relation Network [18] for zero-shot action recognition and as a baseline also use DeViSE [4] both which are described below:

3.3.1 DeViSE

Deep Visual-Semantic Embedding Model (DeViSE) [4] is one of the first deep learning based work on learning a visual-semantic model using unannotated texts for zero-shot image classification. In our work we use this for zero-shot action recognition instead of image classification, and use it as a baseline.

The objective here is to transfer the external knowledge learned about the action classes (this includes unseen classes) based on the large un-annotated text corpus to the visual domain.

As mentioned earlier the visual model (ST-GCN is pre-trained separately on the the training examples of the seen classes along with the language model which is pre-trained on a huge text corpus to produce the word embeddings of

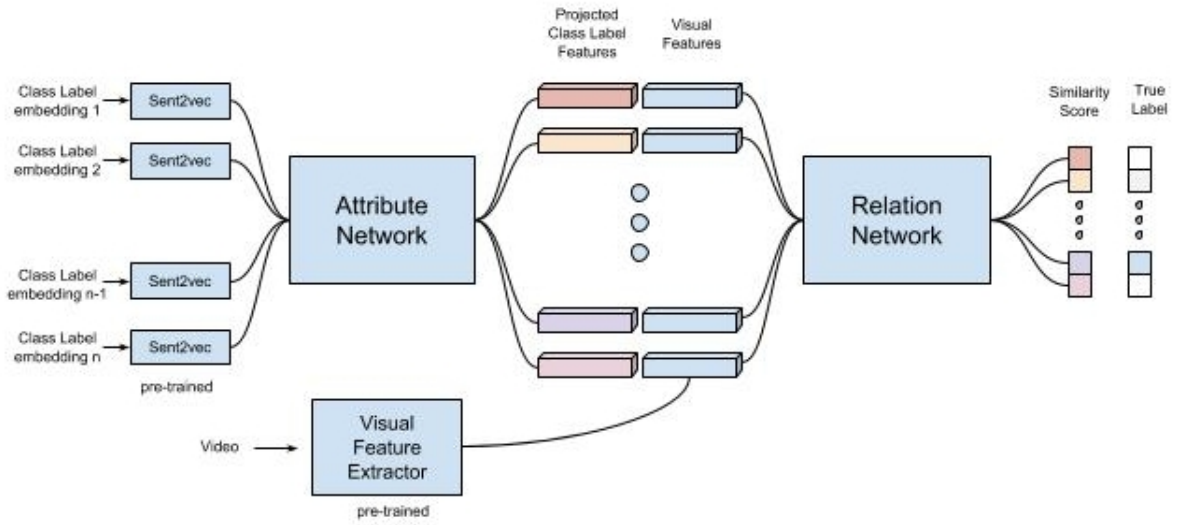

Figure 3: Relation Network Architecture for Zero-Shot Learning

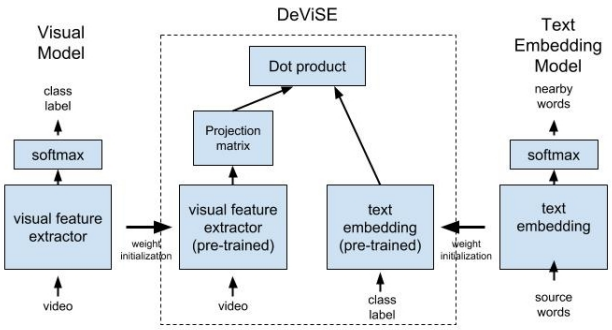

Figure 4: DeViSE Model

all the action class sentences (seen + unseen). DeViSE using these learns a projection matrix (a linear transformation) between the visual features and the class label embedding, so as to make visual features predict the class label embeddings.

For learning the projection matrix the DeViSE model uses a loss function which is combination of hinge rank loss and dot-product similarity. The objective here is to produce higher dot-product similarity between the output of visual model and the class embedding of the correct class and lower dot-product similarity for the class label embeddings of all other classes.

$$loss(image, label) = \sum_{j \neq label} max[0, margin - \vec{t}_{label} M \vec{v}(image) + \vec{t}_j M \vec{v}(image)] \quad (1)$$

In the loss equation above $\vec{v}(image)$ is a column vector denoting the visual feature of the given video during training, M is the linear projection matrix with trainable parameters and $\vec{t}_{label}$ is a row vector of class label embedding of the true class, and $\vec{t}_j$ for all other classes (seen and unseen)

During test time, when a video of an unseen class is given, first we find its visual representation and then project it into the space of class label embeddings using the DeViSE projection matrix and then find the class label embedding with largest dot product similarity and use it's label as the prediction of the model.

We found that using DeViSE the model works well for unseen video examples when it's made to predict amongest only the unseen classes but if it's made to predict amongest all the classes then most of the times the predictions are of the seen classes only and not the of the unseen classes which contains the true label.

This indicates that the visual-semantic embedding is good only for the seen classes but doesn't generalize well enough to correlate the visual features of the unseen class examples with it's corresponding label embeddings. One possible reason being the use of fixed metric the dot product similarity here.

3.3.2 Learning to Compare: Relational Networks

Learning to Compare: Relational Network for Few-Shot Learning [18] overcomes some of the limitations of DeViSE[4]. This model was originally formulated as a few shot learning problem for image classification when there are just K images (authors use K=1 and 5) for every class in the training set. It is based on the principle of meta learning during which the network learns to learn a distance metric for comparing a small number of images within episodes in a few shot setting (K images for every class) and also learns

to generate better semantic representations for comparison. It consists of 2 networks - the embedding network and the relation network. The embedding network learns to generate better semantic representations for the images meant to be compared and the relation network learns to compare them. Both these networks are meta-learned during training time using an episode based strategy.

This approach can be generalized to a zero shot setting where the K images per class are replaced by a single word embedding of every class label. In the zero shot setting the model learns to compare the visual feature of the seen classes with the class label embedding of the seen classes and also on generating better projections of class label embedding to visual space, using an episode based strategy.

For zero-shot setting as well there are two networks. The attribute network consists of fully connected layers. It takes as input the class label embeddings and projects them to the space of visual features and as such is trained to learn a good projection. The projected class label embedding as well as the visual feature are concatenated depth wise and passed to relation network which consists of fully connected layers eventually going to a single output on which sigmoid activation is applied. This output number is a single similarity score for the projected class label embedding and the visual feature, and as such it's trained to learn a good non-linear distance metric. Mean squared error is used as the loss, the true label being the similarity score which is 1 for the correct pair of visual features and class label embeddings and 0 otherwise.

An episode based training strategy is used as this helps in getting better model. In every episode you sample some visual features from the whole training set (of seen classes) and compare each one of them with all the class label embeddings present in that sample.

We replace the image features by visual features from ST-GCN just like for the case of DeViSE to do zero-shot action recognition.

In contrast to Devise, the Relation Network can be seen as both learning a good projection and learning a deep non linear metric (similarity function). The advantage of using this approach is that fixed metrics like in DeViSE are critically dependent on the quality of learned embedding, and they are limited by the extent to which the semantic space can generate adequately discriminative representations. In contrast, by deep learning a nonlinear similarity metric jointly with the projection function, Relation Network can better identify matching and mismatching pairs. Thereby better co-relate the visual features with the external knowledge from class label embeddings.

4. Experiments

4.1. NTU-RGB Dataset

NTU RGB+D[14] is a large scale database for 3-D human activity analysis in an indoor environent. It consists of RGB videos, depth maps, skeleton sequences and infrared frames collected with Microsoft Kinect 2.

In total, it provides 56,000 videos with 4 million frames for 60 different action classes of daily, health-related and mutual (involving 2 people interacting) actions with 40 distinct subjects recorded from three different camera viewpoints.

This makes it currently the largest dataset with 3D joints annotations for skeleton based human activity recognition and suitable for our task. Each video is annotated with 3D joint locations (X, Y, Z) of 25 body key-points per subject for at most 2 subjects.

The performance on the dataset is evaluated in two settings: 1) In the cross-subject setting the training clips come from 20 set of actors and the models are tested on clips from the remaining 20 actors. In the cross view setting, training clips come from two of the camera views and the models are tested on clips from the remaining camera view. We implement our zero shot learning analysis on the cross-view setting.

4.2. Selecting Appropriate Class Splits

Since there are no existing datasets for skeleton based Zero Shot Action Recognition in videos, we construct our own training and test splits over the action classes available in NTU-RGBd.

In order to evaluate our model, we split the 60 available action classes in NTU-RGB+d into 55 seen training classes and 5 unseen test classes. During the training phase the model learns only from the 55 seen classes.

We divide the data into 55 seen classes and 5 unseen classes. Since there are multiple possible combination $({}^{60}C_5)$ to pick these 5 unseen classes and there is no fixed criteria in particular for this dataset and in general for action recognition datasets for doing zero shot learning, we came up with a way which divides the splits based on difficulty levels - from most easiest to most difficult.

Nearest Split: Heuristically speaking this should be the easiest split and should give the highest zero shot accuracy for unseen classes. In this setting, we select our unseen classes based on the availability of a very similar class in the training set. We find the nearest neighbours of of all 60 classes based on their language embeddings. And then we pick the top few classes with least distance from other classes as our unseen classes, ensuring at the same time a sufficient amount of inter class variation between unseen test class names.

Furthest Split: Heuristically speaking this should be the

Trained embeddings		Accuracy (%) (Nearest / Random / Furthest Splits)	
DeViSE	Unseen videos All classes	Top 1	Top 5
		0.00 / 0.02 / 0.00	**12.73 / 9.84 / 0.00**
	Unseen videos Unseen classes	Top 1	
		75.16 / 68.47 / 42.06	
Relational Network	Unseen videos All classes	Top 1	Top 5
		19.32 / 20.16 / 14.45	**43.19 / 47.14 / 45.49**
	Unseen videos Unseen classes	Top 1	
		74.50 / 65.53 / 50.06	

Random Embeddings		Accuracy (%) (Nearest / Random / Furthest Splits)	
DeViSE	Unseen videos All classes	Top 1	Top 5
		0.0 / 0.0 / 0.0	**0.0 / 0.0 / 0.0**
	Unseen videos Unseen classes	Top 1	
		15.08 / 27.95 / 17.27	
Relational Network	Unseen videos All classes	Top 1	Top 5
		0.00 / 0.00 / 0.05	**0.00 / 1.83 / 1.33**
	Unseen videos Unseen classes	Top 1	
		11.65 / 44.88 / 3.79	

Table: Left hand-side table shows accuracies when randomly generated class label embeddings are used and right-side is for pre-trained on Wikipedia corpus. For each table we show the accuracies on our 3 splits and for the baseline and proposed zero-shot models

toughest split and should give the least zero shot accuracy for unseen classes. In this setting, we select our unseen classes which are the most dissimilar to the seen class in the training set. We find the furthest neighbours of of all 60 classes based on their language embeddings. And then we pick the top few classes with highest distance from all other classes as our unseen classes. This split is a true test of the generalization capabilities of our semantic space, as the model needs to learn very strong semantics in order to perform well on this split.

Random Split: In this setting, the unseen action classes are selected randomly from the available action classes. We expect this split to be of intermediate difficulty.

4.3. Details Of The Models

ST-GCN takes as input a 300 frame sequence of 25 3-D body joint locations (X,Y,Z) of 2 most prominent people in the video, and eventually computes the softmax classification probabilites of the 60 action classes used in NTU-RGB+d dataset.

We trained it similar settings to the original model released by the authors on the seen classes for 80 epochs with SGD with base learning rate = 0.01, weight decay = 0.0001 and batch size = 48. We extract the 256 dimensional feature from ST-GCN just before the average pooling and softmax is applied, this are our visual feature.

For DeViSE, the projection matrix is represented as a fully connected layer which takes as input the 256-D visual feature and projects it to 700-D vector, the size of our language embeddings.

We use the hinge margin value = 0.1 in the loss calculations, and trained it for 100 epochs with SGD with learning rate = 0.001, momentum = 0.9, and batch size = 64.

The Relation Network model, consists of two separate neural networks - attribute net and relation net. The attribute net consists of 2 fully connected layers (ReLU's are used as activation function) which takes as inputs the language embeddings and projects them to the dimensions of visual features. The relation net also consists of two fully connected layer, it takes as input the concatenation of the visual features and the output of attribute net (projected language embeddings) and outputs a single number (Sigmoid is applied at the output) which is the relation score between the visual and the language embedding.

We train both the networks for about 400000 episodes sampling from batch of 32 and use ADAM as the optimizer starting with a learning rate = 1e-5 and decay it with a step size = 200000 and gamma = 0.5.

For all our experiments we normalize the visual and language features to be of unit norm.

4.4. Zero Shot Testing Paradigm

We test our implementations based on the two zero shot learning test paradigms. In the ZSL paradigm, the set of classes available during training and the ones used for testing is disjoint. i.e, none of the classes seen during training are used for evaluating the performance during test time. In the second paradigm, known as GZSL (Generalised Zero-Shot Learning), the model uses a subset of the available action classes for training. However, during test time, the performance of the model is evaluated on all available classes.

GZSL is more realistic and is more difficult because for the unseen class examples the models might be inclined towards predicting the nearest neighbour to the actual class amongest the seen classes.

Further we compute flat hit@k metrics the percentage of test images for which the model returns the one true label in

its top k predictions, for k=1 and k=5.

5. Results

We demonstrate the performance of our model on both the Zero Shot and Generalized Zero Shot setting. We report the performance of our baseline zero-shot learning model which uses DeViSE and of our proposed model based on Relation Network for the three different splits we described.

For each of these cases, we report top1 accuracy for unseen examples evaluated over just the 5 unseen classes (ZSL performance) and unseen examples evluated over all the 60 classes (GZSL performace)

5.1. Performance of Vanilla ST-GCN

We report the test accuracy of the vanilla STGCN without the zero shot learning module trained only the seen classes but tested on the unseen classes. This consistently gave us 0 accuracy even for top-5 accuracy. The trained model was never able to correctly predict even once the unseen classes. The reason being it never back propagated through those classes during training and so the corresponding weights are tied to close to zero.

5.2. Comparison between DeViSE and Relation Network

From the tables it is evident that the DeViSE baseline model does a little bit better then Relation Network based model in the ZSL setting when the predictions are to be made only from the unseen classes. For GZSL setting, our proposed Relation Network based model outperforms DeViSE based baseline by a far huge margin for both top1 and top5 accuracies, this indicates that Relation Network is able to learn the semantic relationship between the visual features and semantic features in a true way. DeViSE on the other hand can just distinguish amongst the unseen classes but not when all possible classes are present. This emphasizes importance of having a learn-able comparison metric as implemented in Relational Network model.

5.3. Importance of External Knowledge

To find the importance of external knowledge, we use randomly generated embeddings, i.e. use same models and same training settings but just use some randomly generated embeddings of the class (instead of pre-trained from text corpus). This demonstrates the zero shot performance increase due to the language semantics when the actual embeddings are used. The two tables show how using external knowledge helps our model in zero-shot learning. Without using pre-trained language embedding model, the performance of our action-recognition is close to random chance or even worst when given unseen videos. And hence shows how using external knowledge from text corpus helps our models to correctly predict actions of for the classes on which it wasn't trained.

5.4. Comparison between different splits

As expected in general both the models show best performance in the Nearest neighbour split and the worst performance in the Furthest neighbour split, and intermediate performance in the randomly selected split. This shows the best and worst limits of our zero shot models.

6. Conclusion

We demonstrated a pose based zero shot action recognition framework which uses spatial-temporal graph convolutions to generate visual features and along with class label embeddings generated from pre-trained Wikipedia text corpus can predict novel actions not seen during training time. And we focus on using pose only information of the action performer without looking at context for zero shot learning. Base on our results the use of learnable similarity metric for learning the similarity between visual features and class label embedding contributes significantly to the classification accuracy of the model for examples from unseen classes.

References

[1] I. Bülthoff, H. Bülthoff, and P. Sinha. Top-down influences on stereoscopic depth-perception. *Nature neuroscience*, 1(3):254, 1998.

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[3] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3218–3226, 2015.

[4] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Neural Information Processing Systems (NIPS)*, 2013.

[5] Y. He, S. Shirakabe, Y. Satoh, and H. Kataoka. Human action recognition without human. In *European Conference on Computer Vision*, pages 11–17. Springer, 2016.

[6] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[7] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[10] A. Mishra, V. K. Verma, M. S. K. Reddy, A. Subramaniam, P. Rai, and A. Mittal. A generative approach to zero-shot and few-shot action recognition. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 372–380, 2018.

[11] M. Pagliardini, P. Gupta, and M. Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[12] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-shot action recognition with error-correcting output codes. In *Proc. CVPR*, volume 1, page 6, 2017.

[13] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[14] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. *arXiv preprint arXiv:1805.02335*, 2018.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[17] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[18] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017.

[19] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2013.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[21] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

7. Appendix

In Figure 5, we illustrate the similarities between all available action classes in the visual space through the t-SNE plot of the visual features obtained from the ST-GCN module. This includes the examples from the training set for the action classes according to the random split.

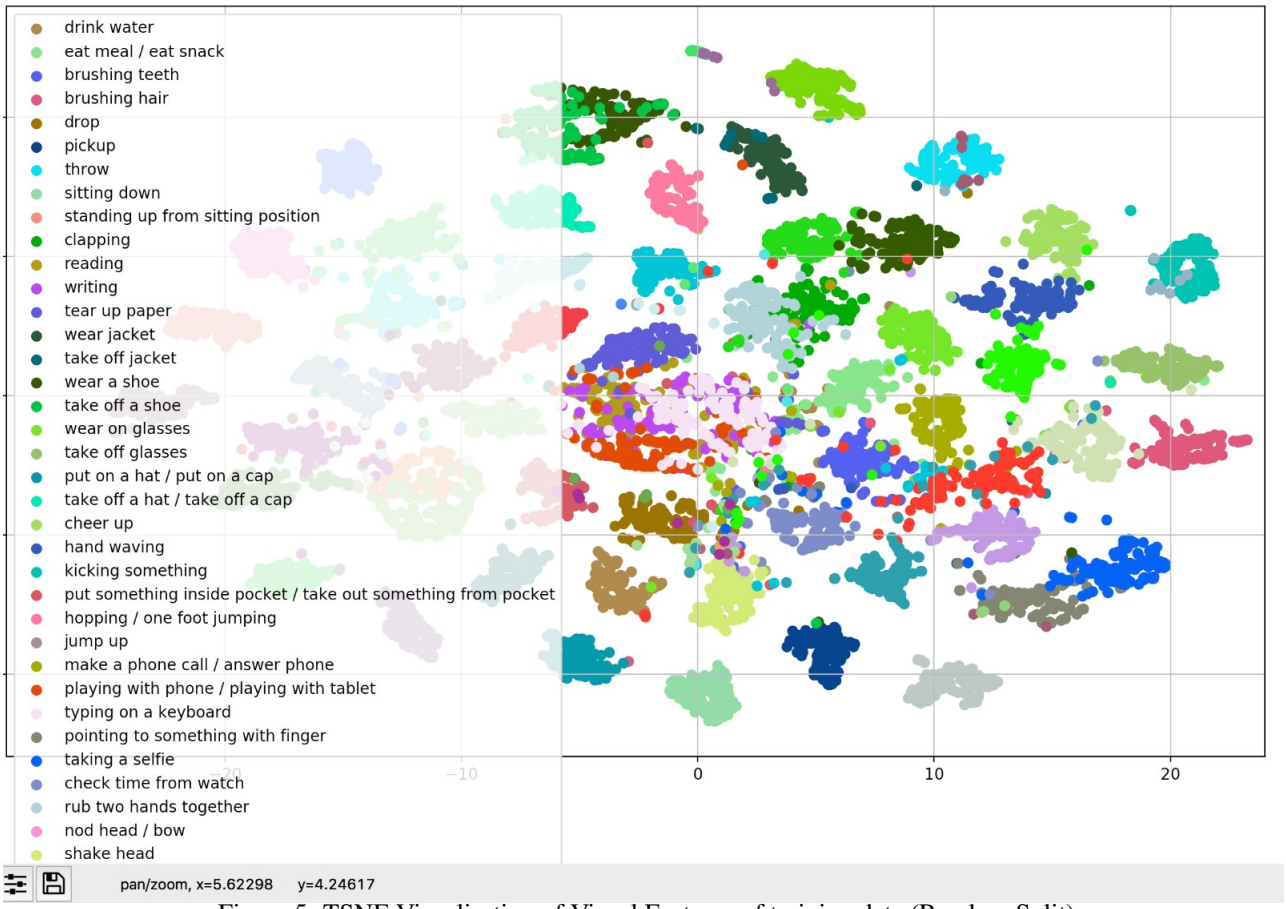

Figure 5: TSNE Visualisation of Visual Features of training data (Random Split)