IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 6, Ser. I (Nov. – Dec. 2020), PP 51-61
www.iosrjournals.org

Self Supervised Methods towards Human Activity Recognition

Sunand Sharma
(Computer Science and Engineering/Indian Institute of Information Technology Guwahati, India)

Abstract
By analyzing the methodology of Self- Supervision Learning(SSL) we highlighted the process which provides summarization of deep perception. The paper demonstrated the advantages of Self-Supervision Learning based on the described database. The study is focused on Human Activity Recognition (HAR) and also investigated the envisaged technique obtainable from distinct databases for instance UTD- MHAD and UCI- HAR.
The process is accomplished through an analysis supervised network by performing training based on the distinct data compared rationally. The significant functionality of the Self-Supervision Learning (SSL) subdue the gulf between the supervised and unsupervised learning. In this paper, three prominent techniques were investigated comprehensively which are promulgated as rotational task loss, frame order prediction and shuffle and learning for conducting the study exclusively.
The main objective of the research is to highlight the following points like analysis, the most suitable domain to implement the self supervised learning which is potentially employed through utilizing the human behavior analysis, to evaluate the classification model appropriately suitable for implementing the actual activity label to enhance its performance by opt dummy set and lastly, we recognize the parameter prerequisite to determine the dummy task. Eventually, this leads to producing the outcome which is reliable to attaining a testing accuracy for the self supervised learning method which might be equivalent to a full supervised approach.
Our work focuses on answering three questions, firstly, which domain of self-supervised techniques prove useful to generalize the representations for human behaviour analysis. Secondly, we answer whether a good performance of a model on the classification in case of the dummy task introduced ensures performance when it comes to the classification of the actual activity labels. At last, we show the parameters needed to set in the dummy task. Hence, it becomes possible to achieve a testing accuracy for the self-supervised approach(with half annotated dataset) similar to that of a fully supervised approach.
Keywords*: Self Supervised Learning, Behavior analysis, Rotation Task Loss, Shuffle and Learn Frame Order Prediction.*

Date of Submission: 10-12-2020 Date of Acceptance: 25-12-2020

I. Introduction

Deep neural networks and machine learning algorithms have been able to achieve good performance for time series and sensory data processing for the past few years[1]. Immensely for image recognition and human activity recognition (HAR), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) improve the performance over traditional methods[2]. Despite that, learning representations from vast amounts of unlabeled data still poses a significant challenge. Presently, the essentiality to train the machine through implementing the Machine Learning algorithms is dependent on the massive amount of raw data, which presents problems in various practical domains where the acquisition of labelled data is not that easy. This is because of two reasons: First, the cost of sensors is high, and they require higher computation power for real-time manipulation of data generated by them and Second, annotation cost and time it takes to create a large amount of labelled data is restrictive. They are thus posing as a hurdle for application of supervised learning directly.

Assessing the challenges, it is feasible to study the semantic representations by unsupervised manual annotations. Specifically, the paper explored enhanced features which can be classified as with fully-supervised methods. The primary Idea is to learn statistical regularities from the unannotated data set that would enable the neural network to learn semantics. This is where self-supervised learning can prove useful.

Machine learning models optimally utilized to train Self-supervised learning, without human interference, including assistance in data analysis providing labelled data. It categorized within the unsupervised learning where the retrieved data dependent on the machine from which it generated embracing characterize, label and information. All these assistance to effectively conclude the outcome depend on coordination and association. Human activity recognition(HAR) algorithm is utilized to gather information from supervised learning capable.

Apart from collecting the datasets developers also need to classify the datasets before who can use it for training. This process is opposite to how humans learn new concepts. Humans can learn new concepts quickly from limited annotated data by relying on past experiences. The recent works [3,4,5] related to self-supervision implement the same Idea, by learning representations to classify the training datasets of "dummy" classes, to achieve generalization not just for the unseen instances of the "dummy" classes, but also for the "actual" types which have only few training examples.

Self-supervised learning has been well explored for image recognition but not for human behaviour analysis. Here, we have tried to examine the application of self-supervised knowledge for human behaviour analysis. Eventually, this leads to producing the outcome which is reliable to attaining a testing accuracy for the self-supervised learning method, which might be equivalent to a full supervised approach.

Our work seeks to answer these questions: First, does the performance of the model on the "dummy" classes ensure its performance when it comes to the classification of actual courses? And Second, what per cent of the annotated data is required to ensure that the performance of the self-supervised method is on par with the supervised one?

1) Motivation

Transformation learning[6] is implemented to formulate a dataset based on the image to a pre-trained system by automation. There are a lot of self-supervised tasks when dealing with images most prominent among them are jig-saw puzzles[7], colourization of grayscale images[8], predicting image rotations[9] and so on.

Behaviour analysis is usually performed on data that contains temporal information acquired from the sensors over some time, so how relevant are the previous methods when time as a dimension is introduced in the data.

Who can view behaviour analysis as a series of activity transitions over some time? Data collected from sensors have found limited applications in behaviour analysis as the data in a sense is available in the raw unlabeled form.

Therefore, leaving us no option but to apply self-supervision. We propose to utilize self-supervised learning (SSL) for human activity recognition over supervised learning wherein a labelled set of data is needed, which will require arduous manual labour to generate the labels.

II. Material & Methods

2) Background:

In this section, we briefly discuss the data that we work with and the associated properties that motivate our solution.

2.1) Datasets used:

To validate our work, we perform the experiments on two different datasets, the first one is the University of Texas at Dallas Multimodal Human Action Dataset(UTD-MHAD) and the second one is UCI Human Activity recognition dataset(UCI-HAR). The highlighted dataset :

Dataset	Total number of volunteers	Number of data sequences	Number of activities	Time frames per activity instance
UTD-MHAD	8	861	27	41
UCI-HAR	30	7352	6	128

The UTD-MHAD dataset[10] is an entire dataset of human action which compromises Kinect sensing data and wearable inertial sensing comprising detector three-axis acceleration and three-axis angular velocity signals. This dataset consists of four temporally synchronized data modalities, which include RGB videos, edge positions from a Kinect camera sensing, and inertial movements from a wearable inertial sensor for a comprehensive set of 27 human actions. We will use the sequence of inertial data and skeleton data collected to perform self-supervision to improve the accuracy of fully supervised Learning.

The 27 activities include: (1) right arm swipe to the left, (2) right arm swipe to the right, (3) straight hand wave, (4) two hand front clap, (5) right arm throw, (6) cross arms in the chest, (7) basketball shoot, (8) right-hand draw X, (9) right-hand draw circle (clockwise), (10) right-hand draw circle (counter-clockwise), (11) draw triangle, (12) bowling (right hand), (13) front boxing, (14) baseball swing from right, (15) tennis right-hand forehand swing, (16) arm curl (two arms), (17) tennis serve, (18) two hand push, (19) right-hand knock on the door, (20) right hand catch an object and so on. After eliminating the number of damaged successively, the dataset comprises eight hundred and sixty-one data patterns.

The UCI-HAR dataset[11], on the other hand, has been carried out with a group of about 30 volunteers who were in an age range of 19-48 years. Each person performed six activities (WALKING, WALKING-UPSTAIRS, WALKING-DOWNSTAIRS, SITTING-DOWN, STANDING-UP, LAYING-DOWN) carrying a smartphone Samsung Galaxy S II on the hand. After utilizing the embedded features like accelerometer and gyroscope, three-axial linear acceleration and 3-axial angular velocity have been captured at a constant rate of 50Hz. They are pretreating the sensing signals by implementing filtration to eliminate noise and then sampled in stable- sliding windows of approx three seconds each with fifty per cent convergence. A 128 size vector is created from each window.

2.2) Transfer Learning

Transfer learning aims to develop methods to apply previously acquired knowledge to accelerate the Learning of actual tasks[12]. The process is simply for measuring the weight of neural networks for A and B, Task A trained through pre-training methodology by machine learning terminology. The main motive is to strengthen task A from retrieving knowledge by machine learning approach and act as a machine for enhancing and assisting in deep Learning of Machine B. Transfer learning has found profound implications in image recognition tasks, the reason being it is easy to find related supervised data. However, when it comes to human activity recognition and behaviour analysis, it isn't easy to find associated supervised data and transfer weights.

2.3) Self Supervision learning towards BA

To collect the resourceful and evitable data about Human labels is very critical and expensive also. Thus the interest is concentrated in gathering the data related to investigating the Learning of unlabeled data. The raw data which is collected embraces systematic structural knowledge which can be implemented to train the machine by applying Self- supervised learning methodology.

This figure demonstrates the comparison between Transfer Learning and the Self- Supervised Learning.

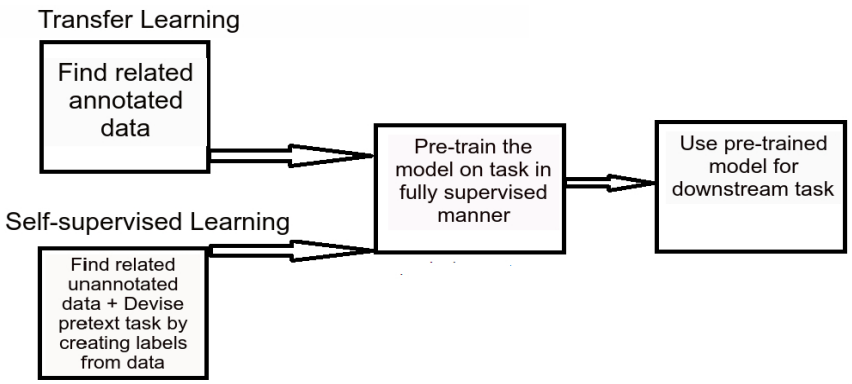


Fig. 1: "Transfer learning vs Self Supervision"

The main objective of the Self-Supervised Learning is to analyze the weight by neural networking and the process to transform it, and the approach is for evaluating and changing the weight can be done by pretraining by the artificial mean. This assists in investigating massive amounts of information by forming subsections. Lastly, in self-supervised learning to analyze the figure relevant for task A to train the model subsequently to teach and transfer the weight over task B. This self-supervised task A is considered as a proxy task or pre trainer or pretext for transforming task B by practical training to resolve the challenges and thus pronounced as a downstream task.

2.3.1) Jigsaw Puzzle task

This self-supervised learning task (i.e. the pretext task) was outlined in Noroozi and Favaro(2016)[7]. This task was carried as a pretext task for image recognition purposes. In this, the input image x is tiled into 3*3 regions and permuted randomly to obtain an input □. The target label □ is the index of the permutation. The pretext task involves using a CNN to predict □ given the input □.

2.3.2) Rotation Task Loss

This method was proposed at Gidaris et al.(2018)[9], where the input image x is rotated by an angle $\theta \in \{0°, 90°, 180°, 270°\}$ to obtain □ and the target label □ is the index of the angle. The pretext task involves using the cross-entropy loss between the target and prediction.

2.3.3) Shuffle and Learn

This method was introduced at Unsupervised Learning using Temporal Order Verification, ECCV 2016[13]. To resolve the challenge of sequence declaration, we opt for a pretext task approach. The paper investigates the sequence of a video frame, whether it is in the correct order or not to explore this sampling method used to analyze the video. The collection of five frames were utilized as an input to generate positive and negative tuples, where positive represent the correctness of the sampling sequence. In contrast, negative demonstrates the wrong sampling of the arrangement.

2.3.4) Unsupervised Representation Learning by Sorting Sequences

This pretext task was proposed at [14]. For generating the positive and negative sampling of frames is assumed to be n, there are unlimited possibilities of n! combinations. After performing the sample of the frames, the output generated four random shuffles. The jigsaw puzzle problem utilizes the challenge as a multi-class classification problem. For the tuple of four frames each, there are about 24 possible permutations.

3) Self Supervised Feature Learning

Here we describe the various methodologies for self-supervision that can be used to learn the ground truth information for the two datasets UTD-MHAD and UCI-HAR and also the approaches adopted to increase the testing accuracy.

3.1) Data Preprocessing and Feature Extraction

For UTD-MHAD

i) Initially, the inertial and skeleton data files are loaded. The acceleration and rotation data features[i.e. accx, accy, accz, rotx, roty, rotz] are obtained from the inertial data, that form the first six data features.

ii) The skeleton data has 20 joints. At any time instance, these 20 joints have cartesian coordinates x,y and z. These, in total, form 60 data features.

iii) The inertial data and skeleton data include a total of 66 elements per timestamp.

iv) However, training for dummy class classification can force the network only to encode features that are useful for distinguishing the dummy classes. Hence, discarding the semantic information that might be useful for actual courses.

v) Thus, keeping in mind the importance of how well the network learns the underlying features in the pretext task determine the performance in the downstream task, we add 12 new features based on the Euclidean distance and Cosine angles between various joints.

vi) The distance features included :

Handd: Euclidean distance between the left hand and right hand.
Footd: Euclidean distance between the left foot and right foot.
Hand_foot l,d: Euclidean distance between the left hand and left foot.
Hand_footr,d: Euclidean distance between right hand and right foot.
Head_handl,d: Euclidean distance between the head and left hand.
Head_handr,d: Euclidean distance between the head and right hand.
Hip_footr,d: Euclidean distance between the right hip and right foot.
Hip_footl,d: Euclidean distance between the left hip and left foot.

vii) The angle features included:

Elbow, a: Cosine angle between left shoulder, elbow and wrist.
Elbow, a: Cosine angle between right shoulder, elbow and wrist.
Kneel, a: Cosine angle between left hip, knee and foot.
Kneer, a: Cosine angle between right hip, knee and foot.

viii) The extraction of 78 features (6 inertial +60 skeletons + 12 newly added) was finally followed by the normalization of the distance features, to scale normalization across individuals of different size and heights by weighting all distances by the distance between hip-centre and head.

On the other hand, for the UCI-HAR dataset, nine features were extracted per timestamp and each activity instance consisted of 128-time frames. The components consisted of readings from body acceleration, rotation, and total acceleration, measured across all the three directions.

3.2) Data Modelling using Supervised Learning

The processed data obtained after feature extraction includes 861 and 7352 data instances for UTD-MHAD and UCI-HAR, respectively. The training and testing data was split into 70% and 30% respectively for both the datasets. Each activity comprised of 41-time frames in the UTD-MHAD and 128-time frames in the UCI-HAR, which implies that each data instance is a two-dimensional matrix with dimensions: timeframes *

featurevector_size [featurevector_size = 78, timeframes = 41 (UTD-MHAD) and featurevector_size=9, timeframes=128(UCI-HAR)].
Thus, the problem is now modelled as time series prediction or sequence classification, for which we use stacked LSTM architecture in a fully supervised way for training. The network architecture used is shown in figure 2. (stacked lstm for sequence classification)

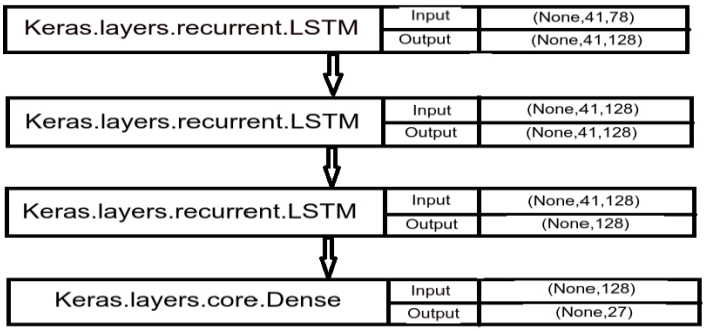


Fig. 2: STACKED LSTM FOR SEQUENCE CLASSIFICATION

In this model, we stack 3 LSTM strata placed on top of each other, formulating the model prominent of learning higher-level representations. For the initial two LSTMs execute their full output sequences. Still, the last one only returns the final step in its output sequence, thus dropping the temporal dimension (i.e. converting the input sequence into a single vector), followed by a softmax layer that performs multi-class classification (i.e. activity recognition) among the 27 activities in UTD-MHAD and six activities in UCI-HAR.

The fully supervised way of training (number of epochs =50) enables us to achieve a testing accuracy of 69.64%(which otherwise would have been 66.92% without the addition of new features) for UTD-MHAD dataset, and this becomes a benchmark for the self-supervision based techniques.
Considering the same architecture for UCI-HAR dataset, we achieve a testing accuracy of 93%.

3.3) Self Supervision considering Rotation task loss

In this method, we start with unannotated training data, consider the rotation task as the pretext task to generate labels, train the stacked LSTM model to predict the degree of rotation of the input, followed by the downstream task to train the model with the actual annotated data.

Firstly, we create four copies, in the unannotated training data, concerning each activity instance (i.e. 2D matrix M). The first one is M rotated by 0 degrees, the second one rotated by 90 degrees clockwise, the third one rotated by 180 degrees clockwise and the last one rotated by 270 degrees clockwise. We associate the labels 0,1,2, and 3 with these instances respectively, and thus prepare a training dataset for the pretext task (xtrain_self) with size four times the size of the original unannotated training data.

We, then, train the stacked LSTM model with the number of neurons equal to 4 in the softmax layer. It is noteworthy that before making the transition to the downstream task, the number of neurons in the softmax layer is set to 27 (as there are 27 activity classes), keeping the other parameters the same.
The downstream task involves training the pre-trained stacked LSTM model with annotated data. The implementation of the method is shown in Algorithm1.

Algorithm1: Self Supervised Learning using Rotation Task
Input: Unlabelled training data(Xtrain)
Output: Self Supervised Network(N)
1. *Initialize xtrain_self and ytrain_self*
2. *for each Activity instance(2D matrix M) in Xtrain do*
a. *Insert Activity Instance into xtrain_self and set the corresponding value of ytrain_self=0*
b. *Rotate M by 90 degree clockwise and insert into xtrain_self*
c. *Set ytrain_self=1*
d. *Rotate M by 180 degree clockwise and insert into xtrain_self*
e. *Set ytrain_self=2*
f. *Rotate M by 270 degree clockwise and insert into xtrain_self*
g. *Set ytrain_self=3*
3. *end*
4. *Assign numclasses =4*
5. *Train the stacked LSTM architecture(model) with parameters(xtrain_self,ytrain_self,epochs=50,numclasses)*

6. *for layer in model.layers[:-1] do*
a. *N.add(layer)*
7. *end*
8. *Assign numclasses=27*
9. *N.add(Dense(numclasses))*
10. *Train the network N with annotated data(downstream task)*

3.4) Self Supervision using Shuffle and Learn

Unlike the rotation task, wherein we consider four classes for the pretext task, here we consider two courses. If the input frames are unshuffled, then the activity instance is said to belong to level 0. In contrast, if the input frames are shuffled (even in slightest order), the activity instance is considered to belong to class 1.

For each activity instance, we take a certain number of permutations (count permutations) by which we randomly arrange the sequence of the frames. For the UTD-MHAD dataset, each instance has 41-time frames, so there are 41! possible ways by which the frames can be shuffled. Similarly, for the UCI-HAR dataset, there are 128! methods by which the frames can be rearranged.

Thus, the pretext task involves training the stacked LSTM model with the number of neurons equal to 2 (unshuffled/shuffled order). The implementation of the method is shown in Algorithm 2.

Algorithm 2: Self Supervised Learning using Shuffle and Learn

Input: Unlabelled training data(Xtrain), count permutations
Output: Self Supervised Network(N)
1. *Initialize xtrain_self and ytrain_self*
2. *for each Activity instance(2D matrix M) in Xtrain do*
a. *Insert Activity Instance into xtrain_self and set the corresponding value of ytrain_self=0*
b. *for each count from 1 to count permutations do*
i. *Randomly shuffle the rows of M and insert into xtrain_self*
ii. *Set ytrain_self=1*
c. *end*
3. *end*
4. *Assign numclasses =2*
5. *Train the stacked LSTM architecture(model) with parameters(xtrain_self,ytrain_self,epochs=50,numclasses)*
6. *for layer in model.layers[:-1] do*
a. *N.add(layer)*
7. *end*
8. *Assign numclasses=27*
9. *N.add(Dense(numclasses))*
10. *Train the network N with annotated data(downstream task)*

3.5) Self Supervision using Frame Order Prediction

This method is almost similar to the shuffle and order, except for the fact that we create a different class label for each random permutation. Firstly, fixed-length(41 frames) instances are sampled and shuffled randomly, and the stacked LSTM model is used to predict the actual order of the frames that were rearranged.

For the pretext task, we prepare a training dataset (xtrain_self) with size “count permutations” times the size of the original unannotated training data.

The proposed model architecture[15] is shown in figure 3.

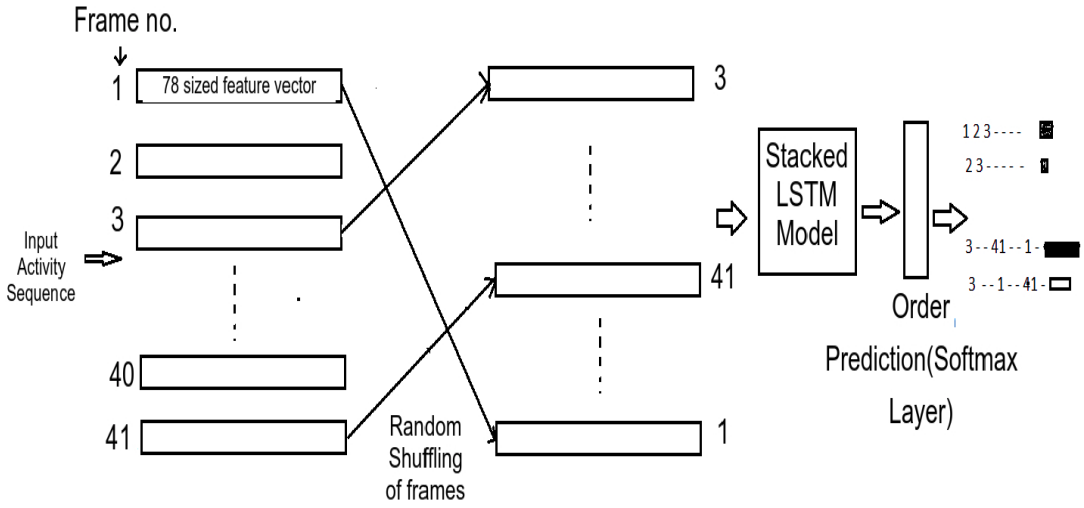


Fig. 3: "Frame Order Prediction" Model Architecture

Algorithm 3: Self Supervised Learning using Frame Order Prediction

Input: Unlabelled training data(Xtrain), count permutations
Output: Self Supervised Network(N)
1. *Initialize xtrain_self and ytrain_self*
2. *for each Activity instance(2D matrix M) in Xtrain do*
a. *Insert Activity Instance into xtrain_self and set the corresponding value of ytrain_self=0*
b. *for each count from 1 to count permutations do*
i. *Randomly shuffle the rows of M and insert into xtrain_self*
ii. *Set ytrain_self=count*
c. *end*
3. *end*
4. *Assign numclasses =count permutations*
5. *Train the stacked LSTM architecture(model) with parameters(xtrain_self,ytrain_self,epochs=50,numclasses)*
6. *for layer in model.layers[:-1] do*
a. *N.add(layer)*
7. *end*
8. *Assign numclasses=27*
9. *N.add(Dense(numclasses))*
10. *Train the network N with annotated data(downstream task)*

Note: The same process (4.3,4.4, and 4.5) is followed for the UCI-HAR dataset, with the main differences being the number of classes in the downstream task are set to 6 instead of 27.

III. Results

4) Performance Evaluation

4.1) Results and Analysis

All the methodologies explained in section 4 were implemented for the two datasets UTD-MHAD and UCI-HAR. In this section, we analyze the results and try to answer the two essential questions, as stated in the Introduction. For both the datasets, the training and the testing data was split into 70% and 30% respectively. All the readings were taken for the number of epochs equal to 50.

The fully supervised approach (4.2) using the stacked LSTM architecture enables to achieve the testing accuracy as follows :

Dataset	Testing Accuracy(%)
UTD-MHAD	69.64
UCI-HAR	95.58

Both of these accuracies form a benchmark while analyzing the performance of various self-supervised approaches.

Now coming to the self-supervision methods, our first finding is that a performance improvement is only observed when the data generated for training in the pretext task is within the same domain as the downstream training data.

To support this finding, we analyze the results for Self Supervision using Rotation Task Loss(3.3.2)

Dataset	Training accuracy for the pretext class	Testing accuracy for the downstream task
UTD-MHAD	16.34	12
UCI-HAR	42.28	31.90

It is evident, and the rotation task-based self-supervision does not perform well as a pretext task in human activity recognition, unlike the work proposed at Gidaris et al.(2018) wherein the same technique yielded positive results for image recognition. The reason is that for image recognition, generating training data for the pretext task by rotation of the images sustains its domain same as that of the downstream training data. However, in case of human activity recognition(let's say for UTD-MHAD), by rotation of the input matrix (2D matrix with rows as time frames and columns as features), the values for one feature vector(column vector) gets swapped with another feature vector; thus the data for certain features becomes out of their domain, leading to the training of the stacked LSTM architecture with unrealistic data. Therefore, this leads to the conclusion that self-supervision leads to an improvement in testing accuracy for human activity recognition only when the pretext task belongs to the domain of the problem.

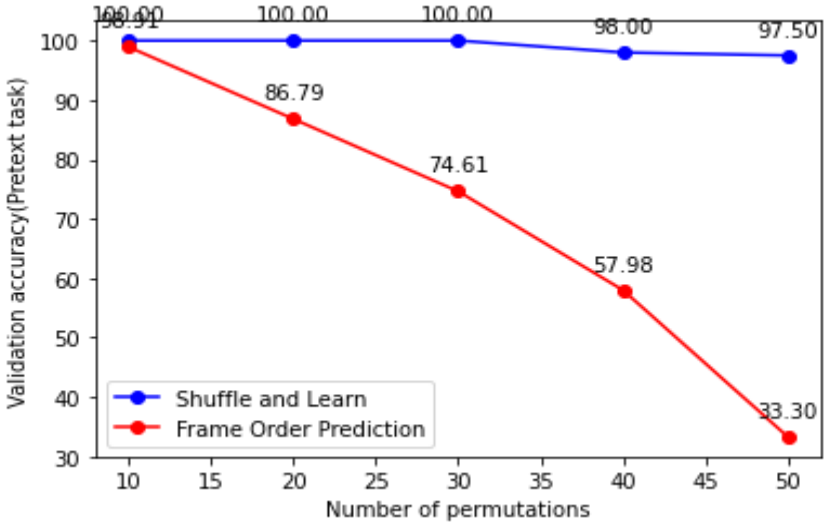
Now when it comes to the methods of Shuffle and Learn(4.4), and Frame Order Prediction(4.5), we need to consider the performance for a various number of permutations of the arrangement of the frames, i.e. for multiple values of the parameter "count permutations". For the UTD-MHAD dataset, the testing accuracy (%)for both methods are as follows (considering count permutations=10):

Shuffle and Learn	70.03
Frame Order Prediction	73.15

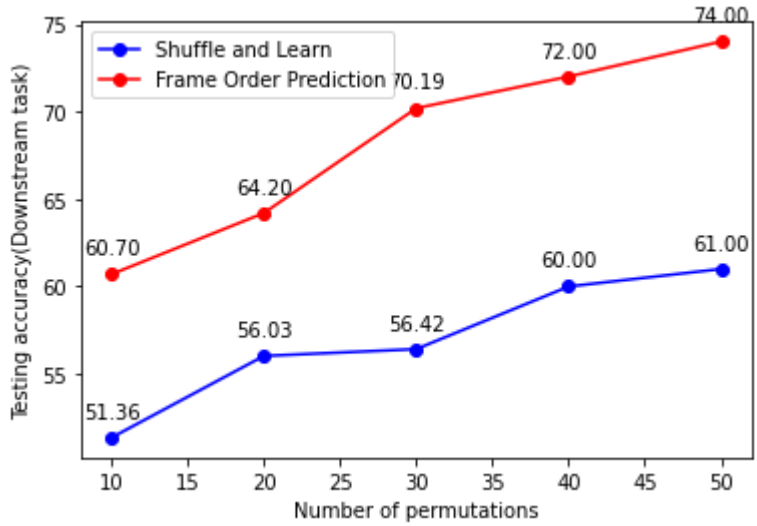
Thus, it is clear that both of these pretext task techniques belong to the domain of the downstream classification. Besides, by taking the number of permutations equal to 10, both methods can learn the ground truth information sufficient enough to surpass the testing accuracy of a fully supervised approach(i.e.69.64%).

However, this comparison is not properly valid as the downstream task still uses the entire annotated dataset for training. Therefore, in further analysis, we consider using only half of the annotated dataset for the downstream task as that would enable us to visualize the real essence of self-supervision.

Coming to the first question posed in the introduction section, does the performance of the model on the “dummy” classes ensure its performance when it comes to the classification of actual classes? We need to compare the validation accuracy for the pretext task and the testing accuracy for the downstream task and make appropriate conclusions. The following graph shows the variation of validation accuracy of the pretext task with the number of permutations of the arrangement of frames, i.e. count permutations for both the methods (Shuffle & Learn and Frame Order Prediction) for the UTD-MHAD dataset.



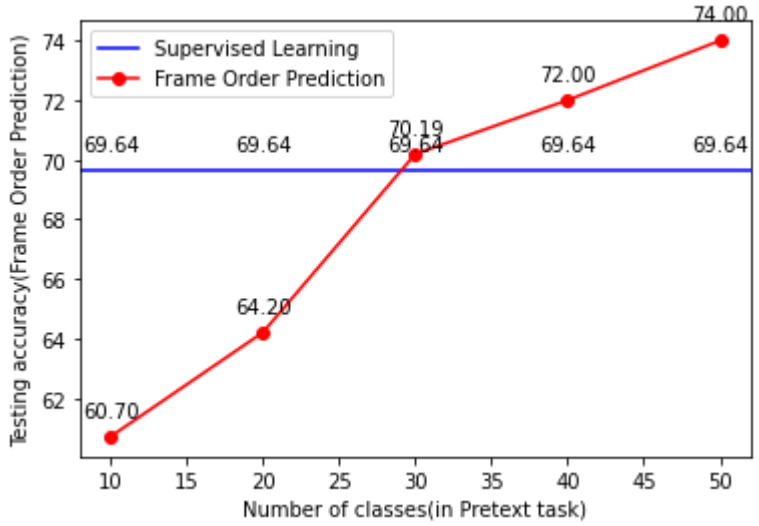
Looking at the validation accuracies for the pretext task, it is clear that 'Shuffle and Learn' proves to be better than 'Frame Order Prediction'. The gap of the validation accuracies between the two self-supervised approaches increases with the number of permutations considered. However, to be sure that both the methods will behave the same when it comes to the classification of the actual classes, we need to analyze the testing accuracy of the downstream task. The following graph shows the variation of testing accuracy of the downstream task with the number of permutations of the arrangement of frames, i.e. count permutations for both the methods (Shuffle & Learn and Frame Order Prediction) for the UTD-MHAD dataset.



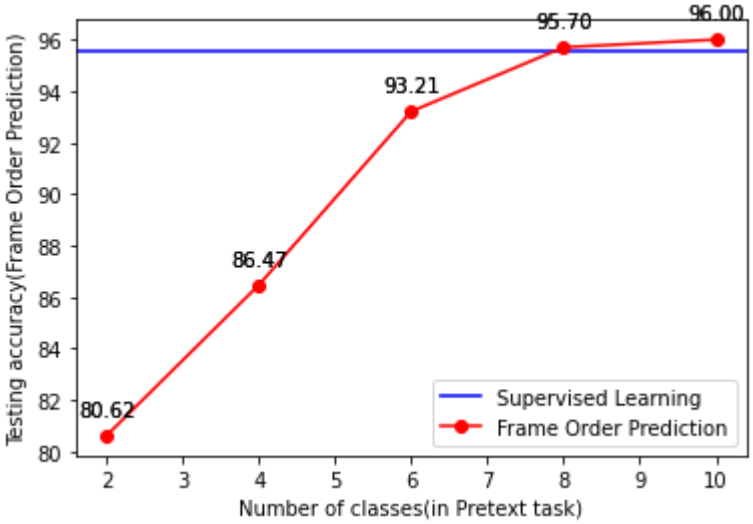
The testing accuracy for the downstream task presents a completely different picture from the validation accuracy of the pretext task. The 'Frame Order Prediction' always surpasses 'Shuffle and Learn' when it comes to classification of the actual classes, in spite of the fact that 'Shuffle and Learn' was able to perform better classification among the dummy classes. This means that , although, the validation accuracy for the pretext task in case of 'Frame Order Prediction' was less , even then it was able to capture more ground truth information than 'Shuffle and Learn' . Hence , it leads to the conclusion that the performance of the model on the "dummy" classes need not ensure its performance when it comes to the classification of actual classes.

Coming to the second question posed in the introduction section, if we have only half of the annotated dataset for the downstream task, than was available for the supervised learning, how many "dummy classes" are needed in the pretext task to achieve a testing accuracy same as that of fully supervised approach.

The following figure displays the comparison of Frame Order Prediction with a fully supervised learning approach for the UTD-MHAD dataset.



By taking the number of "dummy classes" equal to 27, it becomes possible to achieve a testing accuracy for the classification of actual classes approximately to what is achieved by supervised learning.
The following figure displays the comparison of Frame Order Prediction with fully supervised learning approach for the UCI-HAR dataset.



Here also, by taking the number of "dummy classes" equal to 8(slightly more generous than actual classes, i.e. 6), it becomes possible to achieve a testing accuracy for the classification of particular categories approximately to what is achieved by supervised learning.

Therefore, it can be concluded that by taking the number of "dummy" classes(permutations) in the pretext task equal to the "actual" number of categories, it is possible to achieve a testing accuracy for the self-supervised approach (with half annotated dataset) similar to that of fully supervised process.

IV. Conclusion

In this work, we make three significant findings concerning self-supervised learning. This proposed work has been tested for two human activity recognition datasets, UTD-MHAD and UCI-HAR.

The first finding was self-supervision leads to an improvement in testing accuracy for human activity recognition only when the pretext task belongs to the domain of the problem. This conclusion was supported by the approach 'Rotation task Loss' carried out on these datasets.

The second finding was that the performance of the model on the "dummy" classes need not ensure its performance when it comes to the classification of actual classes. We showed that although the 'Shuffle and Learn' approach being able to achieve more validation accuracy in the pretext task than 'Frame Order Prediction', perhaps it is unable to capture ground truth information and thus achieves a low testing accuracy when it comes to the classification of the actual classes.

The last and the foremost conclusion was on having an annotated dataset half the size of that available for supervised learning, by taking the number of "dummy" classes(permutations) in the pretext task equal to the "actual" number of classes, it is possible to achieve a testing accuracy by applying 'Frame Order Prediction' self - supervised learning approach equal to that of the fully supervised learning approach.

References

[1]. Radu V, Tong C, Bhattacharya S, Lane N, Mascolo C, Marina M, and Kawsar F, Multimodal deep learning for activity and context recognition.Proceedings of the ACM on Interactive, Mobile, Wearable and UbiquitousTechnologies, 2018, 1(4):157.

[2]. Hammerla N, Halloran S, and Ploetz T. Deep, convolutional, and recurrent models for human activity recognition usingwearables.arXiv preprint arXiv:1604.08880, 2016.

[3]. Larsson G, Maire M, and Shakhnarovich G. Learning representations for auto-matic colorization. InEuropean Conference on Computer Vision (ECCV), 2016.

[4]. Donahue J and Simonyan K. Large scale adversarial representation learning, 2019, arXiv preprintarXiv:1907.02544.

[5]. Goyal P, Mahajan D, Gupta A, and Misra I. Scaling and benchmarking self-supervised visual representation learning. 2019, arXiv preprint arXiv:1905.01235.

[6]. Pan S, Yang Q, et al. A survey on transfer learning.IEEE Transactions on knowledge and data engineering, 2010, 22(10):1345–1359,.

[7]. Noroozi M and Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. InEuropean Conference onComputer Vision, 2016, pages 69–84. Springer.

[8]. Larsson G, Maire M, and Shakhnarovich G. Colorization as a proxy task for visual understanding. InCVPR, 2017, volume 2,page 7.

[9]. Gidaris S, Singh P, and Komodakis N. Unsupervised representation learning by predicting image rotations, 2018.arXivpreprint arXiv:1803.07728.

[10]. Chen C, Jafari R, and Kehtarnavaz N, "UTD-MHAD: A Multimodal Dataset for Human Action Recognition Utilizing a Depth Camera and a Wearable Inertial Sensor", *Proceedings of IEEE International Conference on Image Processing*, Canada,September 2015.Available at https://personal.utdallas.edu/~kehtar/UTD-MHAD.html

[11]. Available at https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[12]. Morales F J O and Roggen D. Deep convolutional feature transfer across mobile activity recognition domains,sensor modalities and locations. InProceedings of the 2016 ACM International Symposium on Wearable Computers, pages 92–99. ACM.

[13]. Misra I, Zitnick C L, and Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision 2016* (pp. 527-544). Springer, Cham.

[14]. Lee H. Y, Huang J B, Singh M, & Yang M H . Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision* 2017 (pp. 667-676).

[15]. Xu D, Xiao J, Zhao Z, Shao J, Xie D, and Zhuang Y. Self-supervised spatio temporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* , 2019 (pp. 10334-10343)

Sunand Sharma. "Self Supervised Methods towards Human Activity Recognition." *IOSR Journal of Computer Engineering (IOSR-JCE),* 22(6), 2020, pp. 51-61.