SMAK-Net: Self-Supervised Multi-level Spatial Attention Network for Knowledge Representation towards Imitation Learning

Kartik Ramachandruni, Madhu Vankadari, Anima Majumder, Samrat Dutta and Swagat Kumar

***Abstract*— In this paper, we propose an end-to-end self-supervised feature representation network for imitation learning. The proposed network incorporates a novel multi-level spatial attention module to amplify the relevant and suppress the irrelevant information while learning task-specific feature embeddings. The multi-level attention module takes multiple intermediate feature maps of the input image at different stages of the CNN pipeline and results a 2D matrix of compatibility scores for each feature map with respect to the given task. The weighted combination of the feature vectors with the scores estimated from attention modules leads to a more task specific feature representation of the input images. We thus name the proposed network as SMAK-Net, abbreviated from *S*elf-supervised *M*ulti-level spatial *A*ttention *K*nowledge representation *Net*work. We have trained this network using a metric learning loss which aims to decrease the distance between the feature representations of simultaneous frames from multiple view points and increases the distance between the neighboring frames of the same view point. The experiments are performed on the publicly available Multi-View pouring dataset [1]. The outputs of the attention module are demonstrated to highlight the task specific objects while suppressing the rest of the background in the input image. The proposed method is validated by qualitative and quantitative comparisons with the state-of-the art technique TCN [1] along with intensive ablation studies. This method is shown to significantly outperform TCN by 6.5% in the temporal alignment error metric while reducing the total number of training steps by 155K.**

I. INTRODUCTION

Concise representation of knowledge plays a crucial role in imitation learning for vision based robotic applications. Imitation learning with visual data involves observing real-life demonstrations of different manipulation tasks and transferring that knowledge to a robot in order to imitate those tasks. The major challenges in this problem are the complexity and diversity of manipulation tasks and the ambiguous representation of knowledge within the visual image data. These challenges make it practically impossible to label every image in the demonstrated sequence and train the model in a supervised manner. Researchers have thus focused on either unsupervised or self-supervised approaches for solving this problem. Some of these works which primarily focus on considering multiple modalities and using spatial coherence as a form of supervision include, [2], [3], [4] and [5]. [2], [3] uses information like co-occurrence of sounds and visual cues in videos to learn meaningful visual features. Spatial coherence between images is also used in [4], [5] to learn the feature representation of images. In another approach, Pathak et al. [6] makes use of motion cues to perform a segmentation task in a self-supervised manner. Temporal coherence in visual data has also been used in works such as in [7], [8], [9] and [10]. Metric learning is used in [11] where a triplet loss objective is formulated by considering the first and last frames as anchor-positive pairs and randomly selected frames from different videos as anchor-negative pairs in order to generate feature embeddings. Our motivation is to generate a visual feature representation model that learns relevant information about an object interaction task while ensuring invariance to viewpoint and appearance. In order to learn these representations from visual demonstrations in a self supervised manner, we use time as a supervision signal across multiple viewpoints. We use a CNN architecture to generate embedding vectors for each frame of a multi-viewpoint video demonstration and deploy metric learning to bring frames occurring at the same time-step but different viewpoints together in the embedding space while pulling frames occurring at different time-stamps away from each other. The embeddings generated can then be used to learn a control policy which imitates the task to be learned, such as a model free RL agent. This approach is in the same direction as the work of Sermanet et al. [1], where a triplet loss based feature representation was used to imitate a pouring task by considering temporal cues from multiple views as the supervision signal. However, in visual data a lot of irrelevant information is present within the image which

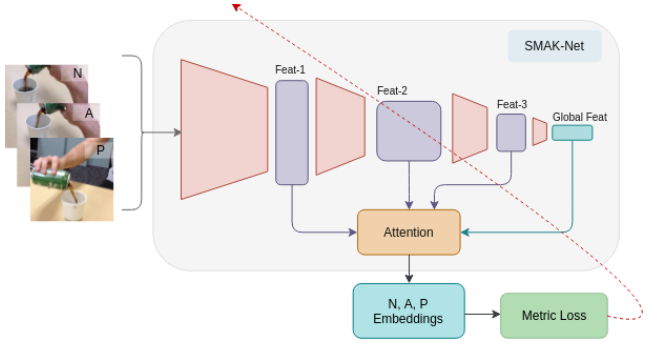
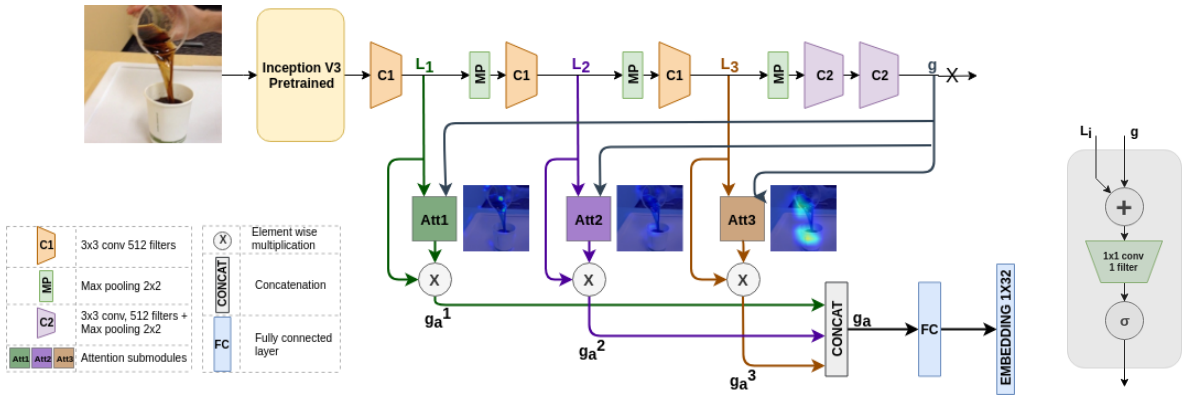


Fig. 1: Overview of the proposed SMAK-Net. The proposed method has two main modules, namely convolutional feature encoder and multi-level attention module. The convolutional encoder takes three images as input, the positive image, anchor image and negative image and generates task specific feature representations with the aid of the multi-level attention module. These feature representations are then trained using metric loss objective.

The authors are associated with TCS Research, TATA Consultancy Services, Bangalore, India 560066. Email ID: {kartik.vr, madhu.vankadari, anima.majumder, d.samrat, swagat.kumar}@tcs.com



(a) (b)

Fig. 2: An architectural overview of the proposed feature representation network. Fig a. shows the proposed network architecture SMAK-Net, which consists of a CNN pipeline and three spatial attention sub-modules (Att1, Att2 and Att3). These sub-modules calculate the attention weight matrix of each intermediate feature maps (L1, L2 and L3 respectively) with respect to the final layer feature vector g. The weights are then multiplied with the intermediate feature maps to output the attention incorporated feature vector. Fig b. shows the structure of the spatial attention sub-module consisting of an addition operation, a 1x1 convolution operation and a sigmoid activation function.

is not relevant to the task such as background, cluttered objects, etc. and therefore it is necessary to ignore this information in order to generate a truly invariant embedding. To solve this problem, we introduce a multi-level spatial attention module in our proposed feature representation model as shown in Fig. 1. In this context, the term *Spatial Attention* refers to the identification and amplification of salient image regions while suppressing the irrelevant and misguiding image regions [14]. The spatial attention module extracts feature maps from different depths and exploits the contextual and spatial relationships between the features in order to generate a more comprehensive and accurate feature representation. There are many works in which spatial attention has been used to perform image classification [12], [13] and action classification [14], [15], [16] from raw visual data. However, to the best of our knowledge, there is no literature that uses trainable spatial attention modules for knowledge representation of visual data. We opt to use the multi-view pouring dataset given by [1] for comparison as it includes many domain variations and makes it appropriate to compare the efficacy of our proposed architecture with the existing state-of-the-art technique. The work presented in this paper has made the following significant contributions.

1) We propose a CNN based feature representation network called ***SMAK-Net*** that incorporates multi-level spatial attention, which is incentivized to amplify the relevant and suppress the irrelevant or misleading information from visual data.
2) Several ablation studies have been performed on different components of the network in order to validate the proficiency of the generated feature embedding using the proposed model.
3) We are only using a metric learning strategy to train the feature representations and are not providing any task specific information (no extra information specific to the pouring task present in the dataset). It is thus intuitive that, this network will also perform similarly when a dataset with a different task is provided.
4) The proposed architecture is shown to a provide a significant improvement over the state-of-the-art technique in terms of accuracy, while compromising on a marginal increase in the size of the parameters. We are able to reduce the alignment error by almost 6.5% along with a reduction in the number of training iterations by almost 155k.

The remainder of this paper is organized as follows. Section II gives a detailed explanation of the proposed approach that includes a Section II-B, which provides a detailed discussion on how the spatial attention module is working. Experimental setup and results are presented in Section III, in which results of an extensive ablation studies is provided later in Section III-C. Finally, in Section IV conclusions are drawn from the work presented.

II. PROPOSED APPROACH

This section provides a detailed description of the proposed feature representation framework. Similar to the work presented in [1], we too use time as a supervision signal across multiple viewpoints for performing metric learning. As the time stamps of each frame in the videos are synchronized, the embedding vectors learn what is common among different looking images which are functionally similar, thereby learning features invariant of nuisance variables such as appearance, background and other image related noise. However, it is necessary to ignore information irrelevant to the task in order to generate an embedding invariant to appearance. Hence, we use a multilevel spatial attention module which collects information from feature maps across different depths of the network and highlights the features among them which are important to the demonstrated task.

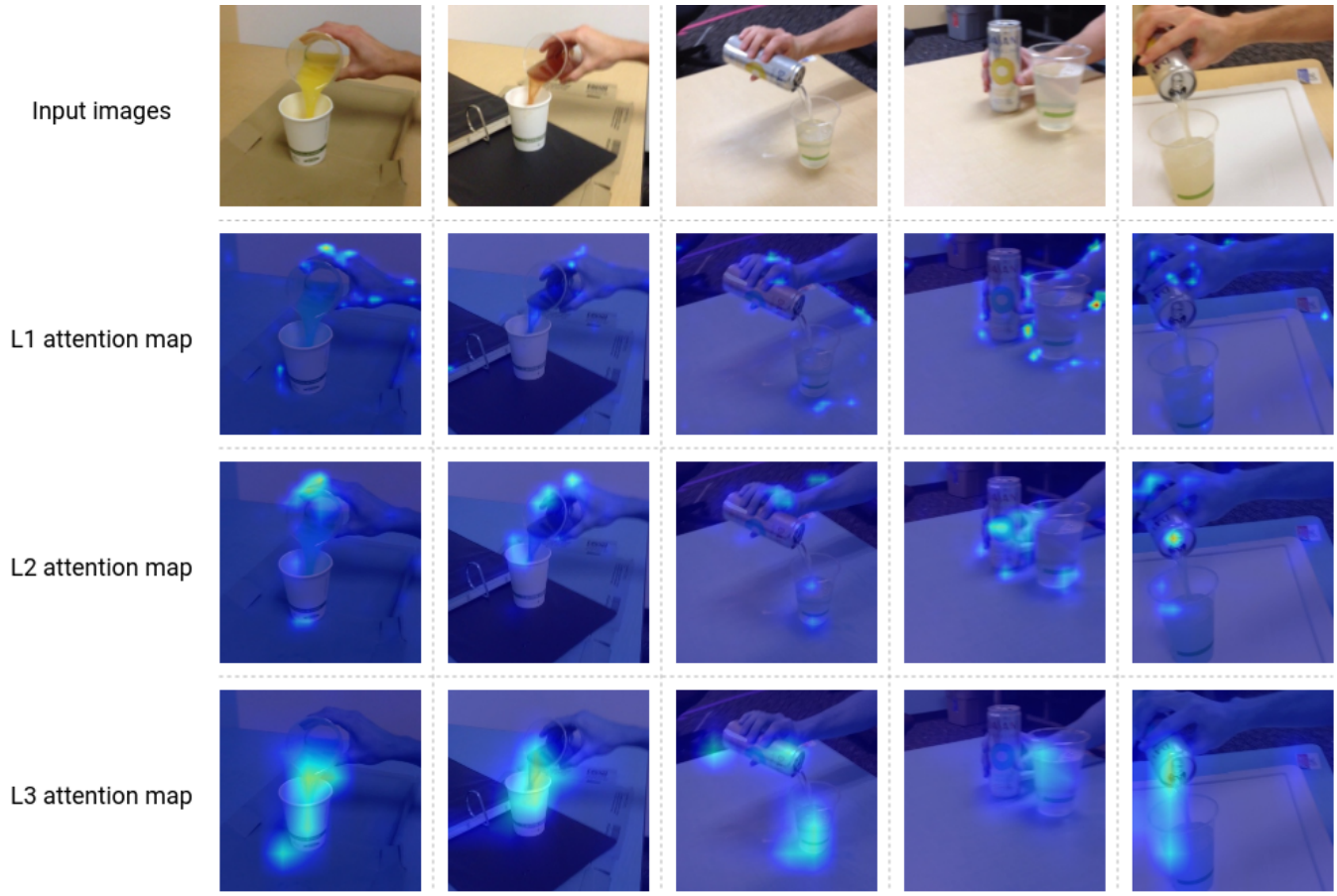


Fig. 3: Attention weight maps generated by the multi-level spatial attention module. Top row images are the original input images fed to the network. The other row images are attention maps resized to the input image dimension and superimposed upon the images.

The following section first explains the attention mechanism and overall network architecture used followed by a discussion of why this mechanism successfully attends to relevant information and ignores unnecessary details within the image.

A. Spatial Attention Structure

The Multi-level Spatial Attention module is shown in Fig. 2a. We use the Inception architecture [17] upto the layer 'Mixed_5d' (initialized with ImageNet pretrained weights) followed by a few extra convolutional layers as the CNN pipeline (refer to Fig. 2a for architectural details). Attention sub-modules extract information from multiple layers of this pipeline to generate the required embedding vector. The attention sub-modules used in this network are similar to the ones used in [14], where a multilevel attention mechanism was used for image classification and fine-grained object recognition. Let the set of feature vectors extracted at a given convolutional layer be written as $L^s = \{l_1^s, l_2^s, ...l_n^s\}$, where l_i^s is the vector of output activations at the spatial location $i \in (1,n)$ in the convolutional layer $s \in (1,...,S)$. Also let $\mathbf{g}$ denote the global feature vector which is essentially the last feature map in the network before the final output layer. In order to incorporate attention into the global feature vector $\mathbf{g}$, we define a compatibility score as follows:

$$C(\hat{L}^s, \mathbf{g}) = \{c_1^s, c_2^s, ...c_n^s\} \tag{1}$$

where $\hat{L}^s$ is the set of vectors of L^s after being linearly mapped to the dimensionality of $\mathbf{g}$. We use the same compatibility score as the one used in [14], as follows:

$$c_i^s = < u, l_i^s + g >, i \in \{1...n\} \tag{2}$$

Here u is the weight vector learned by a 1x1 convolutional layer which takes the sum of the components as input and gives the compatibility scores as output. We then normalize the compatibility scores using a sigmoid function to give the normalized compatibility scores $A^s = \{a_1^s, a_2^s, ...a_n^s\}$ as shown in Fig. 2b. These normalized compatibility scores will now function as the attention weights for L^s and are used to produce a single vector by simple element-wise averaging ($g_a^s = \Sigma_{i=1}^n a_i^s \cdot l_i^s$). The multilevel attention module produces a vector g_a^s for each layer s and the obtained vectors are concatenated to replace the original global vector g with the attention incorporated global vector $g_a = [g_a^1, g_a^2, ...g_a^n]$. This vector is further passed onto a fully connected layer to produce the embedding vector.

Parameter	SMAK-Net	SMAK-Net (extended)	SMAK-Net (softmax)	SMAK-Net (Resnet-pretrained)
No. of parameters	1082k	1580k	1082k	1080k
Temporal Alignment error	10.92%	11.9%	13.6%	13.1%
Classification error	16.1%	17.01%	16.64%	18.86%

TABLE I: Ablation study of the network using different network configurations. SMAK-Net denotes the proposed architecture. SMAK-Net (extended) is similar to the proposed architecture but with more convolutional layers. SMAK-Net (softmax) replaces the sigmoid normalization with a softmax normalization. SMAK-Net (Resnet-pretrained) is an architecture similar in structure and number of parameters to the SMAK-Net and is used with a resnet pretrained network

Network Architecture	Training iterations	Alignment error	Classification error
Multi-view TCN (baseline) [1]	224k	17.5%	19.3%
SMAK-Net Att12	69k	14.1%	16.71%
SMAK-Net Att23	69k	13.0%	16.01%
SMAK-Net Att13	69k	12.2%	**15.98%**
SMAK-Net Att123	**69k**	**10.9%**	16.1%

TABLE II: Comparison between SMAK-Net architecture and state-of-the-art TCN [1]. Two validation metrics defined by the latter are used to compare the two methods and a significant improvement is seen in the proposed network. Att12 indicates that only the L1 and L2 attention maps are used to generate the embedding vector. Att23 and Att13 similarly follow this notation and Att123 indicates that all three attention maps are being used.

1) Metric learning: The embedding vectors produced from the above architecture are trained using time-supervised metric learning in order to make them invariant to viewpoint, scaling and other pixel-level changes. We define the anchor and positive images as frames taken from different viewpoints but at the same time-stamp and the negative image as a frame taken from a different time-stamp, thereby completing the anchor-positive-negative triplet. The metric loss aims to bring the embeddings of the anchor-positive pair closer and pull the embeddings of the anchor-negative pair away. This would teach the representation network to collect samples belonging to a similar category into a cluster and push them away from samples belonging to different categories. Hence metric learning allows us to learn appearance invariant and task specific feature representations from unlabeled multi-view video data. We use the N-Pair loss [18] metric learning objective as it allows us to compare a single anchor-positive pair with multiple negative examples leading to a more efficient training.

B. Intuition towards using attention for representation learning

To generate an embedding vector which can act as a generalized state representation for training any Reinforcement Learning agent, it is necessary to encode the position information of only those pixels from the video demonstration which are relevant to the imitation task. In a standard CNN pipeline, each layer contains many diverse image features and as we go deeper into the network these features possess more contextual information without preserving any spatial positions [19]. Our multi-level spatial attention module takes advantage of this behavior by allowing image patches from shallow layers (local feature vectors l_i^s) to directly contribute to the final embedding vector in proportion to its compatibility with the last layer feature map (global feature vector g). This means we are incentivizing the shallow layers to learn those features which are contextually relevant to the imitation task so that only these features will be embedded in the representation vector. Also, similar to the case in [14], there is a greater benefit of using layers relatively late in the network as they are 'relatively mature' and specific to the task. The use of a multi-level module further allows us to access the diversity of information available at different spatial resolutions in the pipeline so that we can generate a more comprehensive and detailed representation vector.

III. Experiments and Results

The proposed network architecture is implemented in TensorFlow [20] and trained on Telsa V100 GPU. The average training time per iteration on this GPU is 0.15 secs. The learning rate is set to 1e-4 and uses exponential decay with a decay rate of 0.95. The network is trained for 69k number of iterations. The popular Adam [21] is used as the optimizer with β_1 as 0.9 and β_2 as 0.999. We have used L2 regularization with a parameter of 1e-6 and drop-out [22] only on the pretrained network with a probability of 0.8. The performance analysis of the proposed architecture is carried out on the publicly available Multi-View Pouring dataset [1]. The dataset has 235 video demonstrations of a person pouring different liquids from one container to another. Each video demonstration is taken from two viewpoints: a first-person view taken by the person pouring the liquid and a third-person view taken by another person who is constantly moving the camera to generate invariance in scale, viewpoint and other noisy variables such as brightness and saturation. Both the videos have been synchronized such that they start and end at the same time, allowing us to use the time-stamp of each frame in the video as a supervision signal while training the representation network. The total dataset is split into 133 demonstrations for training, 17 for validation and the rest for testing. We have used two error metrics defined by [1] namely, temporal alignment error and classification error. The temporal alignment error measures the semantic alignment of images coming from the same time-stamp but different viewpoints by comparing their nearest neighbors

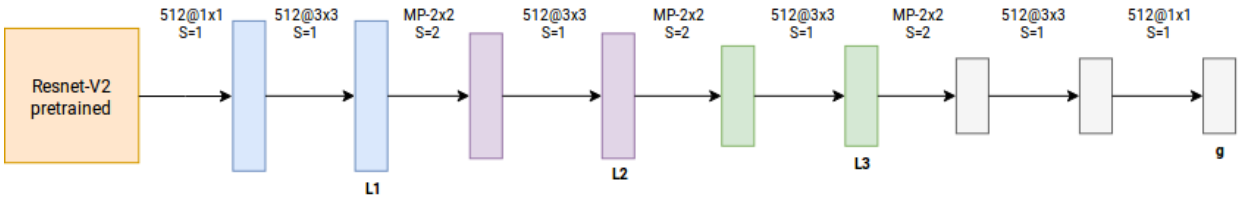


Fig. 4: CNN pipeline used along with Resnet V2 pretrained network with ImageNet weights to generate feature representations. The intermediate feature maps L1, L2 and L3 are then used along with the last feature map *g* to generate attention maps for the individual feature maps. The global feature vector is obtained by concatenating the feature vectors extracted from spatial attention sub-modules.

in the embedding vector space. The classification error measures the accuracy of classification of frames according to predefined pouring attributes by comparing the labels of each frame with that of its nearest neighbor in the embedding vector space.

A. Quantitative results

The quantitative results of our proposed method are tabulated in Table II and compared with the baseline method [1]. It is clear from the table that the proposed method is able to outperform the baseline method with an improvement of 6.5% in the alignment error metric. In addition, our method is able to beat [1] with a 3.2% improvement in the classification error metric. Another interesting finding is that total number of iterations required for training is reduced by 155k when compared with the baseline method. The baseline requires 224k iterations to converge while our method starts to diverge after 69k iterations. This proves that the attention module is able to filter out the unwanted information and aid the network to reach the convergence faster. The divergence of the model is observed as an effect of over-fitting so we have stopped the network training early.

B. Qualitative results

The attention maps (normalized compatibility scores) generated by our network for some of the images from the test data are depicted in Fig. 3. These are visualized to understand where the network is paying attention to while generating the embedding vectors. We observe that the first layer attention weights (L1 attention map) mainly highlight the edges of task relevant objects present within the image such as the hand, liquid and cup. The second layer attention weights (L2 attention map) however focus on highlighting both the cups which are involved in the pouring task. The attention weights from the final layer (L3 attention map) highlight the central region in which the actual pouring is taking place (or where pouring will take place, in case of the images in row 3). In all the attention maps, we can clearly see that even though different background objects are present they are ignored completely and only the relevant regions of the image with respective to the task are being attended to. This supports the fact that using spatial attention improves domain diversity and allows us to generalize to unseen environments.

C. Ablation study

We carefully designed this network architecture by thoroughly experimenting with different layers and activation functions. The results are shown in Table I and Table II and the following conclusions are drawn from these results:

1) In order to justify the proposed implementation of applying spatial attention on three different feature maps, we trained the same network while applying spatial attention on only two feature maps and studied the effect of using different combinations of layers, as shown in Table II. From the results, we observe that the configuration Att13 (L1 and L3 attention maps) shows better performance than the other two configurations. We believe that this may be due to the presence of more spatial information in shallower layers and contextual information in deeper layers, as discussed in sub-section II-B. However, as this configuration is still lacking when compared with the proposed network Att123 (temporal alignment error is higher), it can be said that the layer L2 also contains important information which can be exploited to generate richer feature embeddings.
2) We considered adding more layers in the network to observe whether this would give better validation metric results. Therefore we used an extended network architecture in which an extra convolutional layer is added just before the feature maps L1, L2, L3 and g. We observed a larger error in both metrics with this extended architecture as seen under SMAK-Net (extended) in Table I. This decrement in performance may be due to over-fitting of the training data and hence it was decided to not add any extra layers.
3) We replaced the sigmoid activation with a softmax normalizing function to see whether spatially normalizing the attention map will provide us with more accurate weights as this is generally done in some attention-based image classification networks. We observed that the softmax normalized network yields comparatively poorer results than the proposed network. This may be because generally in image classification tasks a single object of interest is present in the input image. Therefore normalizing the attention weights helps to

concentrate on that singular object. In contrast, feature representation of an action such as a manipulation task might involve multiple objects interacting with each other and equal attention must be paid on all these objects. Hence, applying a constraint on the total sum of weights in the attention map might hinder the attention module from giving equal importance to all the objects.

4) Instead of using an Inception pretrained network, we tried using a Resnet network to generate the initial feature map. We used the Resnet V2 architecture [23] upto the layer 'block4' as the input to the CNN pipeline. The difference in spatial dimension of Resnet output with that of Inception made us use a different CNN pipeline to generate the intermediate feature maps, shown in Fig. 4. Upon training we were unable to obtain better results and hence the results obtained from the Inception based network were shown.

IV. CONCLUSION

We propose a self-supervised representation learning network called SMAK-Net that uses multi-viewpoint video demonstrations to generate a task specific embedding vector for each frame in the demonstrated video. The embedding vector is generated using a multi-level spatial attention framework which captures information from different regions of the input image via a CNN pipeline and highlights the regions relevant to the task shown in the demonstrations. This embedding is further trained using time as a supervision signal across multiple viewpoints using metric learning. We compare the performance of our network with that of the state-of-the-art network TCN [1]. The results obtained demonstrate that spatial attention deployed at multiple levels can improve domain diversity and allow the network to generalize faster as is observed in the reduction of training steps. We also show that the attention maps generated clearly highlight the objects performing the pouring task while suppressing the background. Further, we perform several ablation studies to understand the impact of implementing spatial attention across multiple levels and justify other network choices.

Future work includes improving the feature representation of our network by providing video frames from multiple time-stamps in order to exploit temporal information such as velocity and acceleration. We will also work towards performing imitation learning with these feature representations by training an RL agent for robotic object manipulation using raw video demonstrations.

REFERENCES

[1] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, IEEE, 2018.

[2] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2405–2413, 2016.

[3] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, pp. 892–900, 2016.

[4] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

[5] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361, 2015.

[6] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2701–2710, 2017.

[7] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[8] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun, "Unsupervised learning of spatiotemporally coherent metrics," in *Proceedings of the IEEE international conference on computer vision*, pp. 4086–4093, 2015.

[9] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3636–3645, 2017.

[10] I. Misra, C. L. Zitnick, and M. Hebert, "Unsupervised learning using sequential verification for action recognition," *arXiv preprint arXiv:1603.08561*, vol. 2, no. 7, p. 8, 2016.

[11] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2015.

[12] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, pp. 6967–6976, 2017.

[13] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2017.

[14] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv preprint arXiv:1804.02391*, 2018.

[15] L. Meng, B. Zhao, B. Chang, G. Huang, F. Tung, and L. Sigal, "Where and when to look? spatio-temporal attention for action recognition in videos," *arXiv preprint arXiv:1810.04511*, 2018.

[16] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[18] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016.

[19] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016.