Enabling Edge Devices that Learn from Each Other: Cross Modal Training for Activity Recognition

Tianwei Xing*
University of California, Los Angeles
twxing@ucla.edu

Sandeep Singh Sandha*
University of California, Los Angeles
sandha@cs.ucla.edu

Bharathan Balaji
University of California, Los Angeles
bbalaji@ucla.edu

Supriyo Chakraborty
IBM T. J. Watson Research Center
supriyo@us.ibm.com

Mani Srivastava
University of California, Los Angeles
mbs@ee.ucla.edu

ABSTRACT

Edge devices rely extensively on machine learning for intelligent inferences and pattern matching. However, edge devices use a multitude of sensing modalities and are exposed to wide ranging contexts. It is difficult to develop separate machine learning models for each scenario as manual labeling is not scalable. To reduce the amount of labeled data and to speed up the training process, we propose to transfer knowledge between edge devices by using unlabeled data. Our approach, called *RecycleML*, uses cross modal transfer to accelerate the learning of edge devices across different sensing modalities. Using human activity recognition as a case study, over our collected CMActivity dataset, we observe that RecycleML reduces the amount of required labeled data by at least 90% and speeds up the training process by up to 50 times in comparison to training the edge device from scratch.

CCS CONCEPTS

• **Computing methodologies** → **Transfer learning**; **Neural networks**; **Learning latent representations**; • **Hardware** → **Sensor applications and deployments**;

KEYWORDS

edge devices, transfer learning, cross modality, shared latent representation, activity recognition

ACM Reference format:
Tianwei Xing, Sandeep Singh Sandha, Bharathan Balaji, Supriyo Chakraborty, and Mani Srivastava. 2018. Enabling Edge Devices that Learn from Each Other: Cross Modal Training for Activity Recognition. In *Proceedings of EdgeSys '18: International Workshop on Edge Systems, Analytics and Networking, Munich, Germany, June 10–15, 2018 (EdgeSys '18),* 6 pages.
https://doi.org/10.1145/3213344.3213351

*Both authors contributed equally to this work.

EdgeSys '18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5837-8/18/06…$15.00
https://doi.org/10.1145/3213344.3213351

1 INTRODUCTION

Edge devices are typically equipped with a wide variety of sensing modalities for tracking environmental markers. To provide insights and enable context-aware applications (e.g. user activity recognition [25], workout tracking [22], speech recognition [8]) the data collected on these devices are used to train deep neural network models. However, to fully realize the learning-at-the-edge paradigm, several challenges still needs to be addressed. In particular, the model training process needs to handle insufficient labeled data, and the heterogeneity in inter-device sensing modalities.

As a step towards addressing the above concerns, we propose RecycleML– a mechanism to transfer knowledge between edge devices. Our approach is guided by the observation that application-specific semantic concepts can be better associated with features in the higher layers (close to the output side) of a network model [5]. This observation allows us to conceptualize the layers of the different networks as an hourglass model, as shown in Figure 1. The lower half of the hourglass correspond to the lower layers (close to the input side) of the individual models (trained on specific sensing modalities). The narrow waist is the common layer (latent space) into which the lower layers project their data for knowledge transfer. The upper half of the hourglass comprises of the task-specific higher layer features which are trained in a targeted fashion for task-specific transfer.

To evaluate RecycleML, we emulate edge devices with three sensing modalities - vision, audio and inertial (IMU) sensing as shown in Figure 2. We perform zero-shot learning [23], i.e. use zero training labels, across different sensing modalities when they are performing the same classification task. We achieve this by training the target edge device model to have the same latent space as the source model. RecycleML can also learn to expand the classification tasks of the transferred model with very few training examples.

Our results across a mix of sensory substitutions and task transfers show that, over our collected CMActivity dataset, RecycleML reduces the amount of labeled data required to train edge devices by at least 90% and speeds up the training process by up to 50 times after doing knowledge transfer using unlabeled data.

Our contributions are as follows:

(1) We combine the idea of transfer learning (lower layers transfer) with sensory substitution (higher layers transfer) together and propose a unified framework, where the knowledge in every part of a network could be transferred.

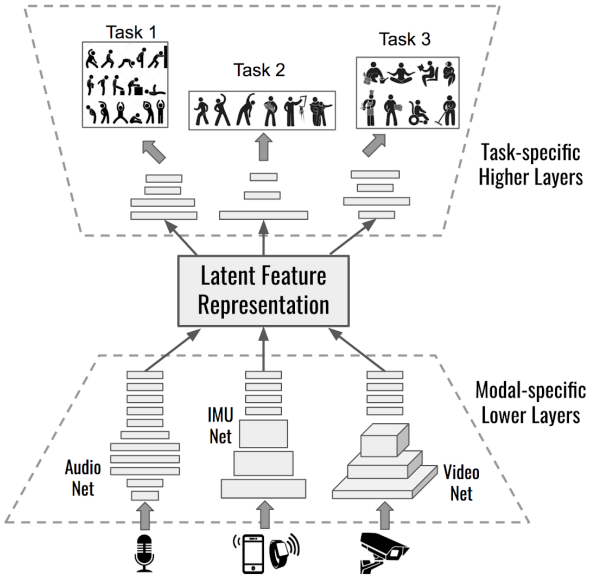

Figure 1: Shared representation between edge devices.

(2) We introduce a new dataset CMActivity that have synchronized data of three modalities: vision, audio, and inertial.
(3) For activity recognition task, we verify that the shared representation exists for time series sensory data, and it can help transfer knowledge from ambience edge devices to wearable edge devices and vice versa. The code for our experiment is available on-line.[1].

2 METHOD OVERVIEW

2.1 Conceptual Scenario

Suppose Alice has an edge device D_{V1} with camera in her living room, and it is trained to do activity recognition. Alice wants to replicate the inferencing ability of D_{V1} on other devices: a smart watch D_W which she wears regularly, an acoustic device D_{A1} in her living room to turn off D_{V1} whenever needed due to privacy reasons, and a camera D_{V2} and a voice assistant D_{A2} in her office, Our objective is to transfer activity recognition knowledge of D_{V1} to D_{A1} and D_W (Video→Audio and IMU), and later, transfer activity recognition knowledge of D_W to D_{A2} and D_{V2} (IMU→Audio and Video).

2.2 RecycleML Description

RecycleML uses the same latent feature representation across edge devices of different modalities to do knowledge transfer. Knowledge transfer uses *synchronous unlabeled data* to map the input of untrained model to the shared latent feature representation of the pre-trained model (details in Section 2.2.1). Later edge devices can either reuse the upper layer across models or do task transfer on the upper layers if needed (details in Section 2.2.2).

[1]https://github.com/nesl/RecycleML

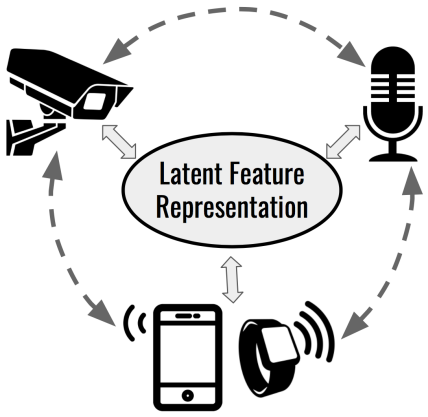

Figure 2: Knowledge transfer across edge devices with different sensing modalities.

2.2.1 Knowledge Transfer. For simplicity, let us consider two edge devices D_X and D_Y, each with different sensing modality capturing data X and Y respectively. Suppose D_X has a pre-trained model M_X and performs task T_X. Our goal is to train a new model M_Y for D_Y to perform task T_Y. To transfer knowledge from D_X to D_Y, we collect data X and Y from both devices while observing the same event. X and Y need not be labeled. An important requirement is the time synchronization in devices D_X and D_Y so as to capture the same event in their data X and Y. Synchronization is natural in different sensing modalities. For example, vision, audio and inertial sensors observing the same event of human motion can capture it in different signals (see Section 3.1 for details).

We input data X to the pre-trained model M_X, and instead of getting the final output value, we calculate the activation values $f(X)$ of an intermediate layer that acts as our shared latent feature representation. f is the transformation of all the early layers before the specific activation. We use $f(X)$ as the training value for the model M_Y of device D_Y. Specifically, we choose a new network g, specialized for input modality Y, and train the network $g(Y)$ so that it maps Y to the same shared latent feature representation by minimizing $|g(Y) - f(X)|^2$ as our loss function. We generate the model M_Y for device D_Y by adding the task specific output layers to g. In this way, model M_X teaches the new model M_Y in a teacher-student data distillation manner [11].

2.2.2 Task Transfer. Transferring knowledge from device D_X to D_Y does not need any ground-truth labels. However, the new model M_Y for device D_Y may need additional information before performing any classification or regression task. Therefore, three different scenarios arise when devices D_X and D_Y performing tasks T_X and T_Y in classification settings respectively: (i) Devices D_X and D_Y are performing same tasks T_X, (ii) Devices D_X and D_Y are performing related tasks T_X and T_Y, e.g. where T_X and T_Y are both human activity inferencing but with different numbers of categories, and (iii) Devices D_X and D_Y are performing completely different tasks T_X and T_Y. In this paper, we study how to transfer knowledge between devices in the first two scenarios.

We explore two different methods of task transfer:

- *PureTransfer* directly uses the higher layers of model M_X for new model M_Y. In this case no further training is needed and no labeled data is required.
- *Transfer+LimitedTrain* freezes the network g and adds higher layers to M_Y and retrains only the higher layers using limited labeled data.

In the first scenario, since the tasks are same we can use both methods. In the second and the third scenarios, direct transfer of higher layers from model M_X to model M_Y does not work as M_X does not give the same desired output. Hence, we use the second method. In our experiments, we evaluate scenario (i) of task transfer using both methods of *PureTransfer* and *Transfer+LimitedTrain* and scenario (ii) using *Transfer+LimitedTrain*.

In our experiments, we used the output of last hidden layer after removing the final output layer from model M_X as the f transformation. Here f and g serve as shared latent representations across modalities. We add a single task specific layer to g to generate model M_Y. In future, we will explore the different choices of f and addition of multiple task specific output layers to g.

3 EVALUATION

3.1 Dataset

For our experiments, we collected a new dataset, called CMActivities, composed of videos for vision and audio modality, and corresponding IMU data (accelerometer and gyroscope) from sensors on left and right wrist. We collected 767 videos of roughly 10 second each from 2 users[2] doing 7 different activities at 6 locations. Every video contains a single activity and is used to label the vision, audio and IMU data. The total duration of collected data for each modality is 125 minutes.

Table 1: Description of CMActivities dataset

Activity	Number of Videos	Duration (sec)
Go Upstairs	162	1338
Go Downstairs	161	1113
Walk	119	1143
Run	115	891
Jump	73	995
Wash Hand	73	1070
Jumping Jack	90	958

We collected the videos of the user using an observer smartphone. The wrist sensors communicate the data to the smartphone of the user doing the activities. The IMU data was timestamped by user's smartphone and the video by the observer smartphone. Time synchronization between vision and audio is naturally present because both are extracted from the same videos. However, time synchronization between the user smartphone and the observer smartphone is needed so as to synchronize video and IMU data. In our data collection, we used the default smartphone timestamps synchronized through the Network Time Protocol (NTP) [17] service, and observed a maximum time difference of 0.5 seconds between the observer smartphone and the user smartphone. We leave it for future to explore the effect of poor time synchronization across devices in observing the same event. We expect the knowledge transfer capabilities of RecycleML to degrade as the time difference between devices increases.

The details of CMActivities are shown in Table 1. The data collection was done at different locations with two users wearing separate set of clothes at each location so as to make sure that the trained classifier learns the activity features and is least affected by the environmental factors. We split 767 videos and IMU sessions into three parts: training dataset (624), testing dataset (71) and personalization dataset (72). Training and testing datasets contain 7 activities at 5 different locations and personalization dataset contains 5 activities at 6^{th} location. We don't have *Go Upstairs* and *Go Downstairs* activities in the personalization dataset.

The training dataset is further split into 3 parts: Pre-Training set, Transfer set and LimitTrain set. The personalization dataset is split into PersonalTrain and PersonalTest sets. The testing dataset is used only for evaluation. The frame rate of video is 29 and the sampling frequency of audio and IMU is 22050 Hz and 25 Hz respectively. We use a window of 2 seconds to extract vision, audio and IMU features from dataset with sliding window of 0.4 seconds between consecutive windows. In case of vision and IMU, we use raw features directly as input to the models. We extracted features from the raw audio data using Librosa [16] and use it as the input features. Specifically, we extract mel-frequency cepstral coefficients (MFCC) [15], power spectrogram [6], mel-scaled spectrogram, spectral contrast [13] and tonal centroid features (tonnetz) [10].

In total, we have 11976 samples in training (5000 samples for Pre-Training set, 6000 samples for Transfer set, and 976 samples for LimitedTrain set), 1377 samples in test and 1592 samples in personalization (475 samples for PersonalTrain set and 1117 samples for PersonalTest set) for each modality.

Table 2: Testing accuracy of baseline models

Input Modality	Video	Audio	IMU
Accuracy	90.92%	92.81%	90.99%
Number of parameters	4.6M	0.8M	57K

3.2 Baselines

To compare the results of RecycleML, we trained Video, Sound and IMU models using Pre-Training dataset individually to do activity recognition. The models we use are the state-of-the-art deep learning architectures that are generally adapted in a wide range of applications:

(a) Video Network is a reduced version of C3D [24] network. It includes four 3D-convolutional modules combined with 3D-maxpooling layers, followed by 3 fully-connected layers and one output layer. The total number of parameters are about 4.6 million.

(b) Audio Network is a multi-layer perceptron model. It has 10 fully-connected layers and a total of 810 K parameters. We add drop-out to avoid overfitting.

[2]The data is collected from the authors and thus does not require approval from IRB.

Table 3: Comparison of knowledge transfer between devices. Significance tests (compared to the training from scratch) are carried out using t-test with P<0.005 in most cases.

Transfer	Trained-Device	Pure-Transfer	Transfer+LimitedTrain	Training from Scratch
Video(D_{V1}) to Audio(D_{A1})	90.92%	90.20%	**90.36%**	84.12%
Video(D_{V1}) to IMU(D_W)	90.92%	94.19%	**94.37%**	70.73%
IMU(D_W) to Video(D_{V2})	90.99%	74.00%	**75.13%**	72.26%
IMU(D_W) to Audio(D_{A2})	90.99%	84.82%	**87.82%**	84.28%

(c) IMU Network is a CNN network. It has 2 convolutional modules (convolution layer + maxpooling layer), 3 fully-connected layers and a output layer. 57K parameters are trainable in this network.

Table 2 shows the summary of the individual models. The models are trained using the training dataset and tested on testing dataset. These baseline models are trained using SGD [4] and Adam [14] optimizers with a learning rate of 0.001. We save the models with best test accuracy after training for 500 epochs.

3.3 Knowledge Transfer Results

Knowledge transfer results are presented in Table 3. In the first and second experiment, vision device D_{V1} is trained while acoustic device D_{A1} and wearable device D_W are untrained respectively. In the third and fourth experiment wearable device D_W is trained while vision device D_{V2} and acoustic device D_{A2} are untrained. For each of these four transfers, we follow the same procedure. Taking vision device D_{V1} to acoustic device D_{A1} as an example, we first train the vision model of D_{V1} from scratch using the Pre-Training set (5000 samples) of training dataset. We use the standard SGD optimizer with a learning rate of 0.001. The training is finished in 500 epochs. We then use D_{V1} as a pre-trained device to transfer knowledge to a D_{A1} following the procedure described in Section 2.2.1. In the knowledge transferring process, we use Adam optimizer with a learning rate of 0.001, and run it for 500 epochs. The data used in transfer process are the synchronized unlabeled vision and sound data from Transfer set (6000 samples) of training dataset. After transfer, the higher layers of audio model can be created using two methods *Pure-Transfer* and *Transfer+LimitedTrain* discussed in Section 2.2.2 when both D_{V1} and D_{A1} are doing the same task. In *Pure-Transfer* method audio model uses the output layer of vision model directly. In *Transfer+LimitedTrain*, we train the new output layer for audio model. We select a small labeled set of 500 samples randomly out of 976 samples from LimitedTrain set of training dataset and name it LimitTrainSet. We use the LimitTrainSet to train the output layer of audio model for 100 epochs using Adam optimizer. As a comparison, we also trained an audio model from scratch using the same LimitTrainSet for 500 epochs. We use more epochs for training from scratch as it takes more time to converge. The other three transfers are tested in the same way. The Audio and IMU models which are trained from scratch use Adam optimizer.

Note: In Video to IMU transfer, it takes more time to transfer the knowledge, so we perform the knowledge transfer for 1000 epochs. In real implementations, the knowledge transfer process for edge devices can either be done in background or at the server using unlabeled data, so as to avoid the overhead.

Table 3 shows the knowledge transfer results between devices doing the same task of activity recognition. Model performance is measured by test accuracy. Considering row 1, *Trained-Device* is the accuracy of pre-trained device D_{V1}. *Pure-Transfer* and *Transfer+LimitedTrain* are the accuracy of device D_{A1} using both methods respectively. The last cell shows the accuracy of audio model trained from scratch using LimitTrainSet. As we can see both methods *Pure-Transfer* and *Transfer+LimitedTrain* achieve better accuracy than training from scratch. This shows that shared latent feature representation is successful in doing knowledge transfer across devices of different modalities. We also observe that *Transfer+LimitedTrain* usually gives the best performance.

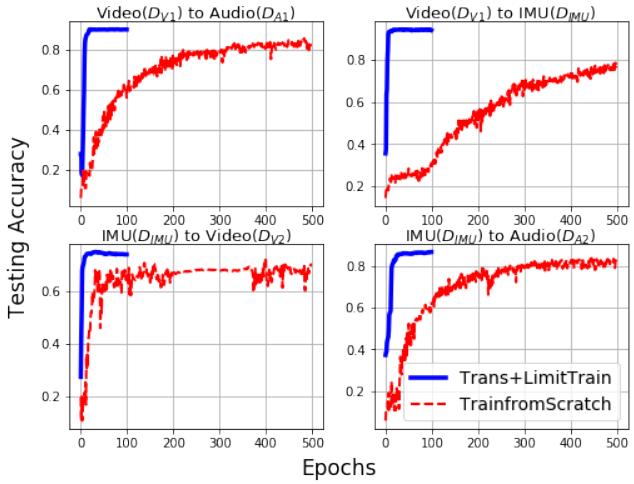

Figure 3: *Transfer+LimitedTrain* converges in 10 epochs whereas Training from scratch requires training for around 500 epochs.

In our experiment, we train every model for 10 times to preclude the effect of randomness. Based on the results, significance tests (compared to training from scratch) are carried out using t-test. We find that the *Transfer+LimitedTrain* can outperform training from scratch ($p < 0.005$) in three cases (Video to Audio, Video to IMU, IMU to Audio); and $p < 0.4$ for the case of IMU to Video transfer. This is because video model is complicated and sensitive, and the performance of video model trained from scratch fluctuates.

3.4 RecycleML Reduces Training Time

We further compare the effect of number of epochs between *Transfer+LimitedTrain* method and training from scratch using LimitedTrainSet (500 samples). Figure 3 shows our results in all the 4 transfers. Clearly, *Transfer+LimitedTrain* method trains model with accuracy greater than 80% in most of the cases with less than 10 epochs, while training from scratch can not achieve comparable accuracy after 500 epochs. This makes RecycleML even more suitable to be deployed on edge devices: it reduces the training time by 50x. The reason for this huge gain is the knowledge transfer using unlabeled data and *Transfer+LimitedTrain* trains only the output layer so it requires very less number of epochs.

3.5 RecycleML Reduces Required Labeled Data

To study the effect of number of labeled data samples on model accuracies, we change the size of training data for *Transfer+LimitedTrain* and training from scratch. All the training samples were selected randomly from LimitedTrain set (976 samples) of training dataset. Although methods converge at different speeds (*Transfer+LimitedTrain* converges in 10 epochs, while Training from scratch takes about 500 epochs), in this experiment, we only compare the converged performance of all the models. Figure 4 shows our results for four device transfers. Consider Video (D_{V1}) to Audio (D_{A1}), *Transfer+LimitedTrain* is compared with training Audio (D_{A1}) from scratch. Using *Transfer+LimitedTrain*, the model achieve best achievable accuracy using only 50 data samples. While training model from scratch cannot get comparable results even if we increase the size of available data to 976 samples as shown in upper left figure. The testing was performed on entire test dataset. So RecycleML reduces labeled data requirement by at least 90%. However, in ideal scenario, when abundant labeled data samples are available, training from scratch slowly converges and can outperform *Transfer+LimitedTrain*. For IMU (D_{IMU}) to Video (D_{V2}), when more than 750 labeled data are available, training from scratch can outperform the method of *Transfer+LimitedTrain*.

3.6 Related Task Transfer Using RecycleML

We tested knowledge transfer from video device to IMU device with video model doing activity recognition task with 7 categories while goal of IMU model is to do activity recognition task with 5 categories in a totally different location.

We did knowledge transfer as described in Section 2.2.1 and finally used *Transfer+LimitedTrain* method to train the output layer of IMU model using PersonalTrain set (475 samples). The trained models are tested on PersonalTest set (1117 samples). In Figure 5, we plot the learning curve on *Transfer+LimitedTrain* and training from scratch trained using PersonalTrain . When transferring knowledge to a relevant task, RecycleML still learns faster: it converges in 10 epochs and gets a testing accuracy of 91.58%, while training from scratch takes 500 epochs and only gets an accuracy of 61.86%.

4 RELATED WORK

RecycleML is inspired from prior works in machine learning for multimodal data. Previous works [12, 18, 20, 21] combine lower layers from multiple modalities to develop a unified model that outperforms the individual modalities. Radu et al. [20, 21] study

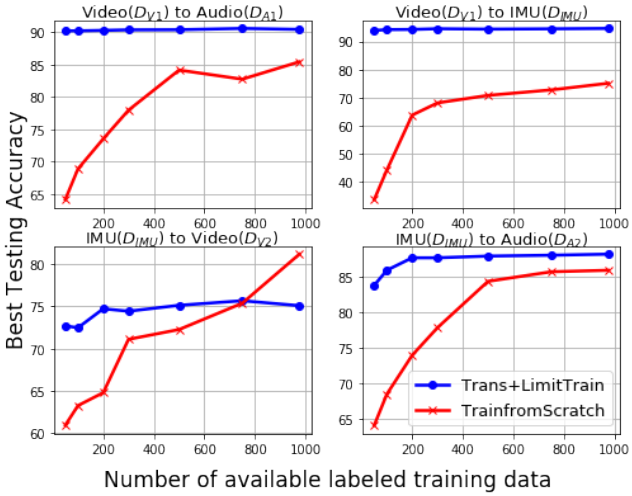

Figure 4: With different sizes of labeled data, *Transfer+LimitedTrain* converges better than Training from scratch.

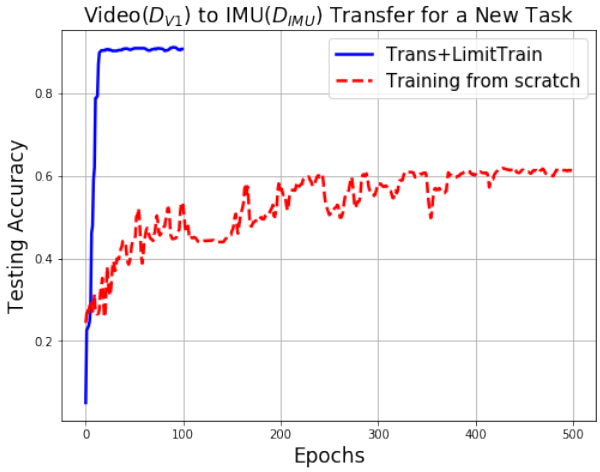

Figure 5: Transferring knowledge to a new task: *Transfer+LimitedTrain* learns faster and better than Training from Scratch.

combining modalities for human activity recognition on mobile devices. We use the idea of representing multiple modalities in the same latent space in intermediate layers of a deep network, but our focus is on knowledge transfer for machine learning models across multi-modal edge devices.

Ba et al. [3], Hinton et al. [11] present knowledge transfer between the same modality. Ngian et al. [19] use shared representations to improve visual speech classification. Aytar et al. [1] learn shared representations that connect multiple forms of image and text data. Frome et al. [7] show knowledge transfer from text to vision for object classification. Gupta et al. [9] present knowledge

transfer between labeled RBG images and unlabeled depth and optical flow images. Aytar et al. [2] show that visual knowledge can be transfer from vision to sound.

The prior works either focus on image and text data, or take two modalities (vision and audio) from the same source into consideration. In RecycleML, we consider three commonly available sensing modalities on edge devices from multiple sources, and create a unified representation that bridge them. This allows edge devices to use multimodal knowledge transfer across different sensing modalities of ambient sensors (vision and audio) and wearables sensors (IMU) for the first time.

5 DISCUSSION

While RecycleML shows promise in terms of handling both paucity of labeled data and also speeds up model training across multiple modalities, the ability of the approach to generalize to different applications for larger datasets needs further investigation. Furthermore, our experiments indicate that while the trained models can be personalized to a specific environment, they need regularization to generalize to new settings.

For cross modal knowledge transfer using RecycleML, we need unlabeled but synchronized data. In our experiments, since audio and video data are captured by the same device, they are naturally synchronized. In addition, we used the default smartphone timestamps, synchronized through the Network Time Protocol (NTP) [17] service, to synchronize IMU device with video and sound device. In real settings, however, edge devices have to be time synchronized in order to observe the same event at the same time.

In our experiments, we chose the fully connected layer (immediately prior to the output layer) as the common latent space. In future, we plan to explore different choices for the shared representation layer, for efficient sensory substitution and task transfer on edge devices.

6 CONCLUSION

Heterogeneity in sensing modality of the edge devices, together with lack of labeled training data, represent two of the most significant barriers to enabling the learning-on-the-edge paradigm. Towards this end, we presented RecycleML, a system that enables multi-modality edge devices to perform knowledge transfer between their models by mapping their lower layers to a shared latent space representation. RecycleML further allows task-specific transfer between models by targeted retraining of the higher layers beyond the shared latent space – reducing the amount of labeled data needed for model training. Our initial experiments, performed using multi-modality data (vision, audio, IMU) for activity recognition, show that transfer model trained using RecycleML leads to reduced training time and results in increased accuracy compared to an edge model trained from scratch using limited labeled data.

7 ACKNOWLEDGEMENT

This research was sponsored by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001, by the National Institutes of Health under award #U154EB020404, and by the National Science Foundation under award #1636916. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies. The U.S. and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon.

REFERENCES

[1] Aytar, Y., Castrejon, L., Vondrick, C., Pirsiavash, H., and Torralba, A. Cross-modal scene networks. *IEEE transactions on pattern analysis and machine intelligence* (2017).

[2] Aytar, Y., Vondrick, C., and Torralba, A. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems* (2016), pp. 892–900.

[3] Ba, J., and Caruana, R. Do deep nets really need to be deep? In *Advances in neural information processing systems* (2014), pp. 2654–2662.

[4] Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[5] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (2014), pp. 647–655.

[6] Ellis, D. Chroma feature analysis and synthesis. *Resources of Laboratory for the Recognition and Organization of Speech and Audio-LabROSA* (2007).

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

[8] Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (2013), IEEE, pp. 6645–6649.

[9] Gupta, S., Hoffman, J., and Malik, J. Cross modal distillation for supervision transfer. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (2016), IEEE, pp. 2827–2836.

[10] Harte, C., Sandler, M., and Gasser, M. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia* (2006), ACM, pp. 21–26.

[11] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Huang, J., and Kingsbury, B. Audio-visual deep learning for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (2013), IEEE, pp. 7596–7599.

[13] Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., and Cai, L.-H. Music type classification by spectral contrast feature. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on* (2002), vol. 1, IEEE, pp. 113–116.

[14] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Logan, B., et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR* (2000), vol. 270, pp. 1–11.

[16] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (2015), pp. 18–25.

[17] Mills, D. L. Internet time synchronization: the network time protocol. *IEEE Transactions on communications 39*, 10 (1991), 1482–1493.

[18] Münzner, S., Schmidt, P., Reiss, A., Hanselmann, M., Stiefelhagen, R., and Dürichen, R. Cnn-based sensor fusion techniques for multimodal human activity recognition. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers* (2017), ACM, pp. 158–165.

[19] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), pp. 689–696.

[20] Radu, V., Lane, N. D., Bhattacharya, S., Mascolo, C., Marina, M. K., and Kawsar, F. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (2016), ACM, pp. 185–188.

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

[22] Shen, C., Ho, B.-J., and Srivastava, M. Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing* (2017).

[23] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems* (2013), pp. 935–943.

[24] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on* (2015), IEEE, pp. 4489–4497.

[25] Yang, J., Nguyen, M. N., San, P. P., Li, X., and Krishnaswamy, S. Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI* (2015), pp. 3995–4001.