RGB-D Domain Adaptation with inter-Modal Self-Supervision

Abbas syed Ali
Data Science and Engineering
Politecnico di Torino
S292423

Esposito Giuseppe
Data Science and Engineering
Politecnico di Torino
S302179

Shabbir Bilal
Data Science and Engineering
Politecnico di Torino
S305979

***Abstract*—Loghmani et al [1] attempted to create a network that learned Domain invariant features by using relative rotation between RGB and Depth images as pretext task and reduce the effect of domain shift between two data distributions. We replicated their experiment using only syn-ROD and ROD, and then applied a variation to their pretext task, rotating both RGB and Depth Images at Random Angels between [0 , 360] and predicting their relative rotation using regressor that aims to find geometric relationship between them.**

I. Introduction

Self-supervised learning is a thriving research direction since it can relieve the burden of human labeling for machine learning by seeking for supervision from data instead of human annotation. Although demonstrating promising performance in various applications, we observe that the existing methods Loghmani et al [1] uses the auxiliary learning tasks as classification tasks with finite discrete labels, leading to insufficient supervisory signals, which in turn restricts the representation quality. In this paper, to solve the above problem and make full use of the supervision from data, we used regression model to predict the continuous parameters of a group of transformations, i.e., image rotation, cropping. Surprisingly, this naive modification stimulates tremendous potential from data and the resulting supervisory signal has largely improved the performance of image representation learning.

We tried duplicate the domain adaptation experiments carried out by Loghmani et al [1] In domain adaptation, a model is trained to perform a visual task in a specific domain (e.g object classification on syn-ROD dataset), but we need to apply it to perform the same task in a correlated dataset that has different domain (e.g. object classification on ROD Dataset), which is done by training model on syn-ROD dataset(source) and using the same model to classify the ROD (target) dataset.Unsupervised domain adaptation is used in the original paper, which analyzes two datasets source and target from two different marginal distributions.

The source dataset is a syntethic training data and target is real data that is used as test dataset for domain adaptation. As shown in the 1 the suggested task is a multi-task problem that includes a supervised main task and an unsupervised pretext task.Main task is a classification task tries to Recognize the Image, While the pretext task is the artificial problem that forces the network to learn domain invariant features by learning geometric relationship between RGB and Depth modalities. As the pretext task is self-supervised by nature,both source and test datasets can be used to train the model.While in the Main task only the source data is used. The pretext task predicts relative rotation between the RGB and Depth image. In the original paper they independently rotated the RGB and Depth image with randomly picked rotation in discrete interval of {90,180,270,360} and then the relative rotation between both images is computed. Our approach includes using a rotation as a regression procedure to handle all of the potential image rotations, from 0 to 360 degrees.

For the main task, we used the network structure specified in the original paper [1], and for our variant, we employed two network streams that will be explained in next section II. The script are available in this gitHub repository: link

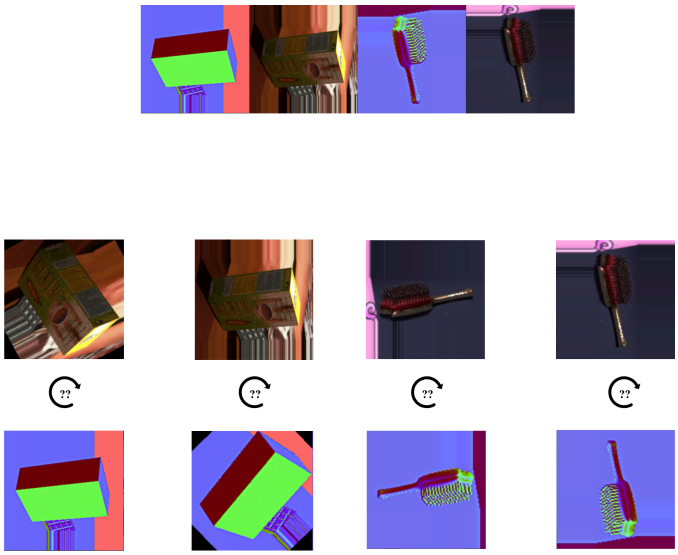

Fig. 1: Images rotated by random values. Our self-supervised feature learning approach is based on the fundamental idea that if the model is not aware of the concepts of the objects shown in the images, he will not be able recognize the rotation that was applied to them.

II. Related work

We'll discuss about the sources we utilized as documentation to complete the Object Recognition task. To begin, we'll discuss self-supervised learning, which is a subset of unsupervised learning techniques. Learning approaches in which ConvNets are actively trained with automatically produced labels are referred to as self-supervised learning. This survey [2] focuses solely on self-supervised learning approaches for visual feature learning with ConvNets that can be applied to a variety of computer vision problems.
The idea behind self-supervised tasks like these is that solving them forces the ConvNet to acquire semantic features that can be beneficial also in other vision tasks. [3] offered a self-supervised task and a geometric transformation prediction as a pretext task, which they proved to be the one that best fills the gap between supervised and unsupervised learning by allowing the model to learn semantic features. Furthermore [3] proved that {0,180,270,360} were the best angles to choose in order to perform the pretext task in question, since after some trials they returned the best accuracy when applied to Object recognition Main head.

RGB-D object recognition is a challenging vision task that is at the heart of many robotics applications. Robots must be able to reliably recognize objects in order to accomplish any manipulation task in any given environment. Furthermore, many robotics application scenarios necessitate not just recognition but also precise localization in order to complete tasks. Using RGB-D sensors, on the other hand, is not only often a necessity in robotics, but it can also greatly improve recognition accuracy when compared to using only color (RGB) images. Because the depth modality provides useful information about object shapes.They presented a CNN architecture to extract features utilizing both modalities RGB and Depth with a two-stream network that comprises of two encoders in this research [4], and they feed the extracted features to the Main task.

III. Dataset

Currently no benchmark dataset is present for RGB-D DA and two datasets source and target are required Additionally, there is no synthetic dataset of the ROD that has already been generated.These experiments involved a synthetic to real domain shift, hence syn-ROD data was gathered by Loghmani et al [1] as a synthetic equivalent to the real ROD dataset.These two datasets Syn-ROD and ROD include images of typical household, office, and other items.They only kept models that included both RGB and depth modalities when collecting syn-ROD using the open catalogs from Public 3D Warehouse and Sketchfab.The ROD dataset contained 51 categories, but not all categories had enough images for CNN to learn them correctly, therefore four categories were removed, leaving only 47 categories to be used.Furthuremore background was randomised using MS-COCO dataset otherwise the problem will be ill-conditioned and the prediction will become background dependent.

IV. Method

As these are domain adaption experiments .we are required to predict the label of target data only training the model on labelled source dataset and unlabelled target data.This is a multi task problem, the main head is object classifier train in a supervised way while the pretext task is trained in an unsupervised way.It is worth mentioning that pretext task is predicting relative rotation, multiple of 90 degrees between RGB and Depth Image for the original experiment and For the variation angels are randomly Chosen between 0 and 360 degrees , So these are well posed problems.

A. Network Architecture

The network architecture that loghmani et al [1] outlined in his paper was the one we utilized in our experiment. As shown in the figure 1, we can see that there are two feature extractors, one is fed with RGB images E^c and the other one is fed with Depth images E^d. For the both feature extractors we use pretrained ResNet18 without fully connected layers and global average pooling layer. Then, using late fusion technique, these features are concatenated along color dimension to get final RGB-D dimension. The resulting tensor is used as input for the main head and pretext task.

- **Main Head**
  For the Main task we are solving classification problem for 47 classes.We perform global average pooling on the feature extracted by both RGB feature extractor and Depth Feature Extractor and concatenated them along the channel axis then fully connected layer with 1000 neurons *fc*(1000)followed by batch normalisation, ReLU activation function and dropout with probability p = 0.5.Finally followed by fully connected layer with 47 neurons that uses Softmax activation function.
- **Pretext Head**
  Pretext head solves 4-way classification problem and tries to predict relative rotation between RGB and Depth Images.It consists of two Convolutional layers and two fully connected layers.The first convolutional layer uses 1×1 kernel size and stride[1,1] with 100 neurons.the second convolution layers uses 3×3 kernal size and stride[2,2] with 100 neurons. Followed by convolutional layers first fully connected layer *fc*(100) with 100 neurons and then there is second fully connected layer *fc*(4) with 4 neurons. All convolutional layers and fully connected layers use batch normalization and ReLU activation function and *fc*(100) was followed by dropout with p=0.5 except last fully connected layer that uses softmax activation function.

B. Implementation details

We have source dataset $S = \{((x_i^{sc}, x_i^{sd}), y_i^s)\}_{i=1}^{Ns}$ and target dataset $T = \{(x_i^{tc}, x_i^{td})\}_{i=1}^{Nt}$ where x^{*c} is RGB image, x^{*d} is depth image and y^s is source label. We transform the both source and target datasets by generating random relative rotation between RGB and depth image and represent them

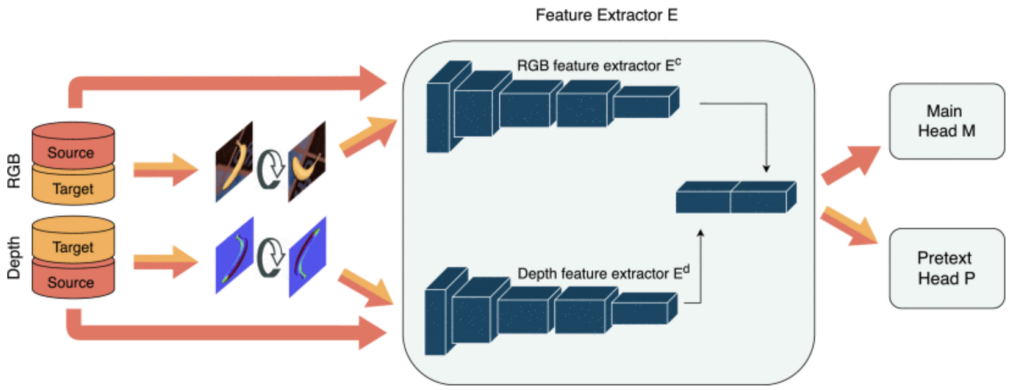

Fig. 2: The model's structure. The images from the source and target datasets will be sent to the extractors, which will extract the features from the RGB images and the depth images. These features will then be concatenated and passed to the model's main task and pretext task performers, respectively.

Algorithm 1 RGB-D Domain Adaptation

Input:
Labeled source dataset S = $\{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), y_i^s)\}$
Unlabeled target dataset T = $\{((\tilde{x}_i^{tc}, \tilde{x}_i^{td}), y_i^s)\}$
Output:
Object class prediction for the target data $\{\hat{y}_i^t\}_{i=1}^{N_t}$
Procedure Training(S,T):
Get transformed set $\tilde{S} = \{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), z_i^s)\}$
Get transformed set $\tilde{T} = \{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), z_i^t)\}$
for each iteration **do**:
Load mini-batch from S
Compute the main loss L_m
Load the mini-batches from $\tilde{S}$ and $\tilde{T}$
Compute the pretext loss L_p
Update the weights of M from ∇L_m
Update the weights of P from ∇L_p
Update the weights of E from ∇L_m and ∇L_p
Procedure TEST(T)
for each (x_i^{tc}, x_i^{td}) in T **do:**
Compute $\hat{y}_i^t = M(E(x_i^{ct}, x_i^{dt}))$

as $\tilde{S} = \{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), z_i^s)\}_{i=1}^{\tilde{N}_s}$ and $\tilde{T} = \{((\tilde{x}_i^{tc}, \tilde{x}_i^{td}), z_i^t)\}_{i=1}^{\tilde{N}_t}$ respectively. z_i^* represents relative rotation between RGB and Depth images.To compute relative rotation both RGB and Depth images are rotated independently at angle multiple of 90 degrees thus rotating the images at any angle between [0,90,180,270] and the relative rotation is computed as difference of angle between RGB and Depth images,then the labels are assigned as [0,1,2,3] representing [0,90,180,270] respectively.From optimization point of view , we have minimization problem where we minimize objective function

$$L{=}L_m \ + \ \lambda_p{\cdot}L_p \ {+}\alpha{\cdot}L_{en} \tag{1}$$

where L_m is cross entropy loss for the Main Head and can be described by the expression

$$L_m = -\frac{1}{N_s}\sum_{i=1}^{N_s} y_i^s log(\hat{y}_i^s) \tag{2}$$

Where $\hat{y}_i^s = M(E(x^{sc}, x^{sd}))$, While L_p is cross entropy loss for pretext task and it is sum of two losses. First is the cross entropy loss for relative rotation prediction on source dataset and second is the cross entropy loss computed for relative rotation prediction on test dataset. The sum of these two losses is multiplied with weight (λ_p) to control the contribution.It is set equal to 1 as it was in loghmani et al [[1]].

$$L_p = -\frac{1}{\tilde{N}_s}\sum_{i=1}^{\tilde{N}_s} z_i^s log(\hat{z}_i^s) - \frac{1}{\tilde{N}_t}\sum_{j=1}^{\tilde{N}_t} z_j^t log(\hat{z}_j^t), \tag{3}$$

since we have to predict the labels of target dataset using DA approach so we don't use labels for target dataset in training rather we use cost function proposed by Pietro Morerio in [5].

$$L_{en} = -\sum_{x_t \in T} \langle f(x_t;\theta), log f(x_t;\theta)\rangle \tag{4}$$

the entropy L_{en} computed on the softlabels $z_{soft}(X_t) = f(x_t;\theta)$, which is nothing but the network predictions [Lee (2013)]. Alpha is contribution regulator for this loss and according to Loghmani [1] alpha is set to 0.1.we used Algorithm 1 to perform our experiments.
While predicting the labels for target dataset at test time, the pretext head is discarded and predictions are only given by

Main head as $\hat{y}^t = M(E(x^{ct}, x^{dt}))$

1) Rotation as Regressor(Variation): Pretext part of the experiment allows the network to learn significant features of both source and target data set, specifically the network learns the features that are domain invariant.In fact Loghmani et al [1] showed in results that presence pretext task leads to better performance for such domain adaptation. But using only four rotations can be little reductive as it gives only 16 unique combinations and possibly network see all possible relative rotations between RGB and depth images in few epochs. So we implement similar approach but there is possibility of a lot more possible relative rotations. Instead of randomly rotating RGB and depth images at multiple of 90 degrees and then computing relative rotation we rotate both images images at any random angle between 0 to 360 degrees and then compute relative rotation. This gives infinite possible rotations between RGB and depth images. Now for each input of RGB and depth image we are not predicting among four values but any possible value and this is no more classification problem instead it is regression problem . For this we modify architecture a little bit. pretext task consist of two convolutional layers (Conv(1*1,100), Conv(3*3,100)) followed by fully connected layer *fc*(100), all followed by ReLU activation function and fully connected layer *fc*(1) that predicts the value of relative rotation.To improve performance we also use batch normalization after each layer and drop out with p=0.5 after *fc*(100). For regression problem we can not use cross entropy loss as loss function but we use mean squared error as our loss function.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{5}$$

We want weights such that the distance between the predicted value and the original value is minimum and we calculate the weights using the following equation:

$$\begin{gathered}\frac{1}{2}\frac{\partial \sum_n (y_n - o_n)^2}{\partial w_i} = \sum_n (y_n - o_n)\cdot\left(-\frac{\partial o_n}{\partial w_i}\right) = \\ \sum_n (y_n - o_n)\,(1 - o_n)\,x_n^i\end{gathered} \tag{6}$$

There is an issue with this approach e.g if relative rotation is zero degrees and predicted value is 360 degrees(0 and 360 degree is the same angle in a circle and MSE loss should be zero) and the loss is $(0-360)^2$ which is much greater then zero.More specifically we can say there is no convex loss function suitable for this task. So we reformulate the problem, let us θ donate the relative angle between RGB and Depth image as θ

$$\theta \leftarrow (\cos\theta, \sin\theta) \tag{7}$$

as function of sine and cosine. Instead of predicting angle between rgb and depth, network predicts the sine and cosine and then we can compute angle using

$$\theta \leftarrow \arctan 2(\sin_{pred}, cos_{pred}) \tag{8}$$

Algorithm 2 RGB-D Domain Adaptation

Input:
Labeled source dataset S = $\{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), y_i^s)\}$
Unlabeled target dataset T = $\{((\tilde{x}_i^{tc}, \tilde{x}_i^{td}), y_i^s)\}$
Output:
Object class prediction for the target data $\{\hat{y}_i^t\}_{i=1}^{N_t}$
Procedure Training(S,T):
Get transformed set $\tilde{S} = \{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), z_i^s)\}$
Get transformed set $\tilde{S} = \{((\tilde{x}_i^{sc}, \tilde{x}_i^{sd}), z_i^t)\}$
for each iteration **do**:
Load mini-batch from S
Compute the main loss L_m
Load the mini-batches from $\tilde{S}$ and $\tilde{T}$
Compute the pretext loss L_{sin} as L_{sin}^S + L_{sin}^T
Compute the pretext loss L_{cos} as L_{cos}^S + L_{cos}^T
Update the weights of M from $\nabla\ L_m$
Update the weights of P from ∇L_p
Update the weights of E from ∇L_m and ∇L_p
Update the weights of R_{sin} from ∇L_{sin}
Update the weights of R_{cos} from ∇L_{cos}

Procedure TEST(T)
for each (x_i^{tc}, x_i^{td}) in T **do:**
Compute $\hat{y}_i^t = M(E(x_i^{ct}, x_i^{dt}))$

and now our loss function for pretext task becomes

$$loss(y_n, x_n) = \frac{1}{n}\sum_{i=1}^{n}(\sin_{pred_i} - \sin\theta)^2 + \frac{1}{n}\sum_{i=n}^{n}(\cos_{pred_i} - \cos\theta)^2 \tag{9}$$

The overall loss becomes

$$L = L_m + \lambda_p\cdot(L_{sin}^S + L_{sin}^T + L_{cos}^S + L_{cos}^T) + \alpha\cdot L_{en} \tag{10}$$

Here value of α stays same as it was 0.1 but as the regression loss is greater than classification loss we decrease value of λ to 0.4.

as network has to predict two values , we have to add the new branch for pretext task with same architecture described for regression task and one branch will predict Cosine while other will predict Sine [3]

V. Experiments

Different experiments were performed to check the domain adaptation and their settings and results are explained below

A. Source only

For source only experiments we condsidered only syn-rod dataset for training and used learning rate 0.0001, dropout 0.5 , batch size of 64 , momentum 0.9 and weight decay of 0.05. Accuracy of model on Rod dataset trained using these parameters is 42.57%.

B. Rotation as classification

To reproduce the experiment performed by Loghmani et al. [1], we used syn-ROD as source with labels and Rod

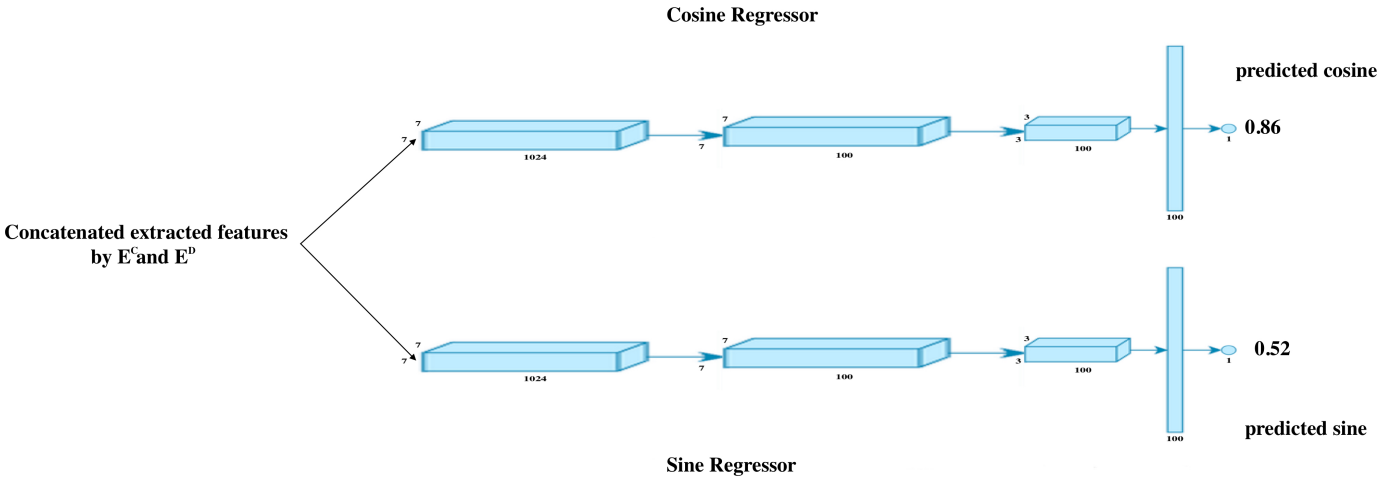

Fig. 3: Going deeper into the pretext task, it will now include a regression task rather than a classification task that will still receive as input the concatenated features extracted by both RGB and Depth images. Predicting them will require that we predict their sine and cosine in order to complete a regression task because "in the circle" it doesn't work as it would if we were on a straight line.

dataset without labels. And also created $\tilde{S}$ and $\tilde{T}$ dataset where labels are relative rotation between RGB and Depth images. Using hyperparameters for Unsupervised Domain Adaptation experiments with relative rotation between RGB and Depth of only multiples of 90 degrees, we considered only synROD dataset for training and used learning rate 0.0001, dropout 0.05 , batch size 64, momentum 0.9 and weight decay of 0.04 and λ_p is 0.8.Accuracy of model on Rod dataset trained using these parameters is 59.46%. This accuracy is less then what Loghmani et al [1] achieved but our experiments shows that the results are consistent and also Loghmani trained the mdoel for 40 epochs and we trained our model for 20 epochs due to google colab limitations and these results are for 15 epochs but the accuracy has a positive trend so it was expected to increase in the following epochs.

C. Using rotation as regression

As variation we implemented regression as pretext task instead of classification and tried to predict any possible relative rotation [0,360] between RGB and Depth. For this experiment also we used synROD as source with labels and Rod dataset without labels for the Main head and created created $\tilde{S}$ and $\tilde{T}$ dataset where labels are any relative rotation from 0 360.For rotation as regression experiments we used learning rate 0.0003, dropout 0.05, mometum 0.9 and weight decay of 0.04, batch size 64. After 13 epochs we got an accuracy 61.09%.
For the 18 epochs with the same hyper-parameters we got an accuracy of 59.43% and this could be explained by the fact that, as the number epochs increases in training, the network become more and more specific for the regression task and this can lead to decrease in accuracy for the target dataset.Images of graphs made from TensorBoard for Accuracy Fig 4,Sine loss Fig 5 and Cosine loss for source Fig 6 and Sine Fig 7 and Cosine Fig 8 loss for target dataset are shown in figures given.By training for 40 epochs and with learning rate of 0.00001 and with the same other hyperparameters the accuracy was 40.56%, there was very slight increase in accuracy with each epoch.

VI. Conclusion

We replicated the experiments of Loghmain et al. [1] performing an Unsupervised Domain Adaptation and also implemented our variation. Comparing to the source only experiments, DA experiments with the pretext task has better accuracy and this can show that relative rotation is an effective pretext task for DA that can force the network to learn the domain invariant features. Furthermore t-SNE of the features of the Main head shows that this pretext task effectively aligns source and target distributions. From our experiments we got better accuracy on target using the regression as pretext task compared to the one where we use classification with labels $\{0, 1, 2, 3\}$ representing the rotations $\{0, 90, 180, 270\}$ respectively as pretext task. [3] As they showed that network focuses on high level features, which predict the absolute rotation and according [1], the relative rotation is predicted by matching the feature coming from the two modality representations instead of learning trivial features. As a result of all these tests, we can see that the self-supervised task successfully lowers the domain shift, which further suggests that the key to perform DA on RGB-D data is making use of the inter-modal relations. We hope that the further research can improve the performances.

References

[1] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo, and M. Vincze, "Unsupervised domain adaptation through inter-modal rotation for rgb-d object recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6631–6638, 2020.

[2] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

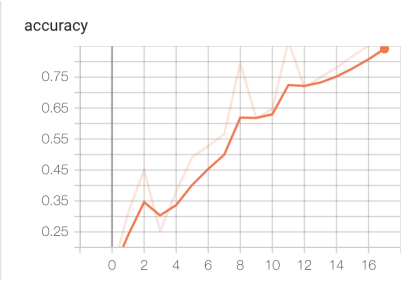

Fig. 4: This is the accuracy while forward propagation of training images.

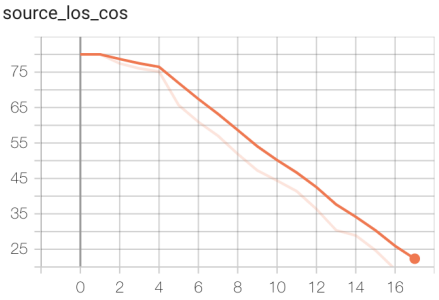

Fig. 5

[3] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *CoRR*, vol. abs/1803.07728, 2018.

[4] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 681–687, 2015.

[5] P. Morerio, J. Cavazza, and V. Murino, "Minimal-entropy correlation alignment for unsupervised deep domain adaptation," *arXiv preprint arXiv:1711.10288*, 2017.

Fig. 6

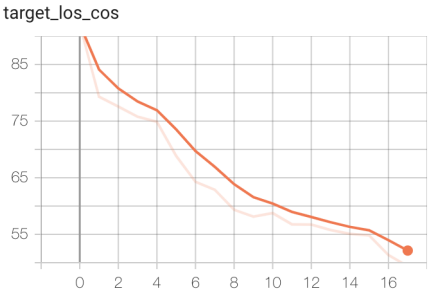

Fig. 7

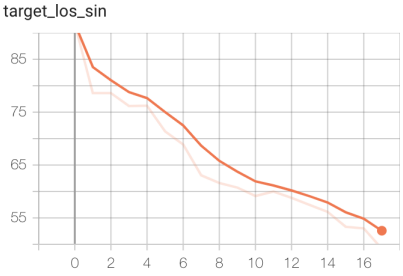

Fig. 8

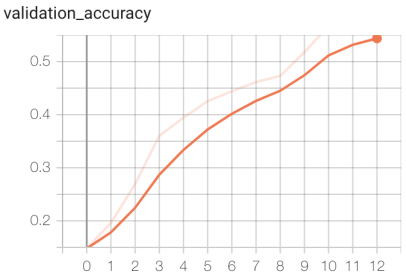

Fig. 9