LEARNING ROBUST REPRESENTATIONS BY PROJECTING SUPERFICIAL STATISTICS OUT ICLR 2019 REPRODUCIBILITY CHALLENGE

Anonymous authors
Paper under double-blind review

ABSTRACT

The main goal of reproduced paper is to build a classifier that is not susceptible to covariance shift of the data, thus helping neural network to create better domain generalization. They introduce new neural building block - NGLCM and a method to project out textural information learned by CNN - HEX. In this paper we tried to reproduce said method and compare results by training model on PACS (Li et al., 2017), MNIST and FERG-DB (Aneja et al., 2016).

1 METHOD

1.1 NGLCM

NGLCM is a neural block that mimics *gray-level co-occurrence matrix* (GLCM) (Lam, 1996) idea. Main difference is than NGLCM can be trained like any other NN layer during back-propagation, allowing to tune it's parameters. Thanks to that, this block can extract textural information about image but is not capable of extracting semantic information. This is used later to unlearn classifier layer from associating textual noises to labels.

1.2 HEX

The main idea of HEX is to project predictions based on all information about data onto subspace which is orthogonal to the ones based on textual-only description of input. Then we project-out unnecessary noise information. As a result we should obtain predictions which are more independent to covariance shifts, thus more reliable in domain classification. It is achieved be using three different outputs:

$$
\begin{aligned}
F_A &= f([h(X;\theta), g(X;\phi)];\xi) \\
F_G &= f([\mathbf{0}, g(X;\phi);\xi]) \\
F_P &= f([h(X;\theta), \mathbf{0}];\xi)
\end{aligned}
$$

where F_A, F_G, F_P stands respectively for the results from all representation, only textural representation and raw data.

Projecting F_A onto the subspace that is orthogonal to F_G with

$$F_L = (I - F_G(F_G^T F_G)^{-1} F_G^T) F_A$$

yields F_L for parameter tuning. In testing time F_P is used instead of F_L. More information about rationale of this method can be found in appendix of original paper Wang et al. (2019)

2 REPRODUCTION DETAILS

In this section we will describe how we tried to reproduce results featured in original paper.

2.1 IMPLEMENTATION

Everything was implemented using PyTorch, open-source machine learning library for Python. Since no code was provided with paper, we had to write everything from scratch. We have performed all tests using Jupyter Notebooks so they are easy to repeat at will.

We have implemented NGLCM block correspondingly to the paper, although we had to guess what direction was used in GLCM matrix (we used default 0), as well as we had to guess MLP layer shape. Another assumption was that the output of NGLCM is 16×244 matrix (16×16 pushed through MLP layer), but we somehow had to make classifier vector. To accomplish that we 'squashed' it to $16 \cdot 244$ length vector to further concatenate it with AlexNet output and pass to HEX projection.

Same goes from HEX. We calculated F_A, F_G and F_P as in paper and finally coded F_L. Then results from F_L are passed into final SoftMax classification layer.

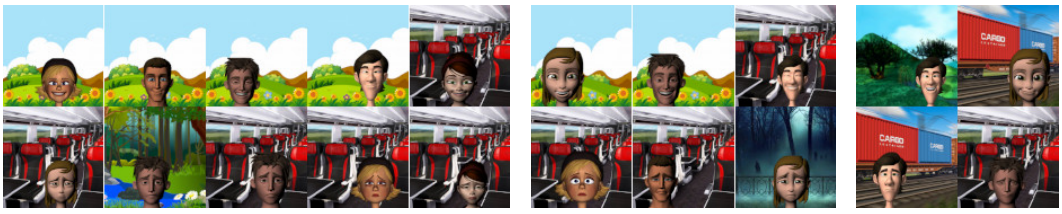
2.2 DATASETS

2.2.1 MNIST

Firstly, we generated 6 datasets where each of them has 100 images per label, 10k images total . Subsequent datasets have images rotated by angle $\alpha \in \{0°, 15°, 30°, 45°, 60°, 75°\}$, each of them representing one domain. We use all domains but one for training and remaining one for testing.

2.2.2 FERG-DB

Like in original paper, we used modified FERG-DB with backgrounds correlated to emotions. As we did not have access to original dataset used in paper, we generated set of datasets based on clean FERG database. While we used same partitioning of data (50% train, 30% valid, 20% test), each of datasets has ≈ 10000 images instead of 50000. Samples of images from datasets ($\rho = 0.8$) are presented below:



(a) train set (b) validation set (c) test set

2.2.3 PACS

Similar to MNIST datasets, we divided images from PACS into 4 domains representing art, cartoon, photo and sketch styled images. Then we used one domain for testing and remaining ones for train. We used all PACS data including 'extra' ones.

2.3 EXPERIMENTS

During challenge we tried to reproduce three experiments described in original paper.

2.3.1 MNIST

MNIST rotation dataset tests, where we tried to replicate results from paper, provided in Table 3. (Wang et al., 2019). We used hyperparameters listed below:

1. optimizer: Adam
2. learning rate: 10^{-3}
3. weight decay: 10^{-3}
4. 50 epochs

Experiment went quite well and we managed to reach similar results as in original paper.

2.3.2 FERG-DB

FERG-DB tests, where we prepared 10 different datasets, each created with background correlation variable $\rho \in \{0.0, 0.9\}$. Then we tested it using two models: HEX and modified HEX, where we replaced whole NGLCM module with single 224×16 MLP layer. In paper, testing using this modification is called 'Ablation tests'.

1. optimizer: Adam
2. learning rate: $5 \cdot 10^{-4}$
3. 70-100 epochs

Hex training was converging very quickly to 100% validate accuracy so we shortened epochs to 70 (from 100 in paper) to save time. We also noticed that during Ablation test we sometimes had to rerun training because model over fitted to test data too quickly to reach good validation results. Also we were not sure what number of parameters to give for MLP layer, so we picked 224×16 since it is a bit less then number of parameters for NGLCM.

2.3.3 PACS

PACS domain shift tests, where we wanted to replicate results, provided in Table 4. (Wang et al., 2019). We used random data augmentation (crops and vertical flips) to prevent overfitting. We first trained AlexNet alone to see if we get similar results.

1. optimizer: Adam
2. learning rate: 10^{-3}
3. weight decay: 0 or 10^{-2}
4. 50 epochs

We wanted do get the best results we could, so we used a two step learning. First we have loaded AlexNet pretrained on ImageNet dataset (provided by torch library) and trained only classifier. Then to fine tune the network we have added weight decay to optimizer and trained all parameters for additional 50 epochs. We saved model that got best results during this two stage training and calculated accuracy.

After getting somehow positive results from AlexNet, we tested if HEXnet would perform better or similar. We used hyper parameters listed below:

1. optimizer: Adam
2. learning rate: 10^{-5}
3. weight decay: 10^{-5}
4. 100 epochs

Following the paper, we first trained 10 epochs of classifier layer of pretrained AlexNet, then plugged HEX in for additional 100 epochs. At first we had problems with really unstable gradients, so we had to turn learning rate down drastically in comparison to AlexNet one. When we work it out, we didn't really get satisfying results no matter what strategy we used. At the end we came to conclusion that in original experiments special PACS learning heuristics, described in this paper (Li et al., 2017) must have been used. Unfortunately we didn't have enough time to study and replicate such learning heuristics. Still we are surprised that under the same condition HEXnet performed far worse then AlexNet, which according to paper results should be worse.

3 RESULTS

3.1 MNIST

Test domain	ADV	HEX	HEX
$\mathcal{M}_{0^\circ}$	91.1	90.1	92.3
$\mathcal{M}_{15^\circ}$	98.2	98.9	98.7
$\mathcal{M}_{30^\circ}$	98.6	98.9	97.1
$\mathcal{M}_{45^\circ}$	98.7	98.8	97.9
$\mathcal{M}_{60^\circ}$	98.4	98.3	96.5
$\mathcal{M}_{75^\circ}$	92.0	90.0	93.1
Avg	96.2	95.8	95.9

Our results are marked as blue. We managed to reproduce results obtained in original paper quite faithfully (Table 3 (Wang et al., 2019)).

3.2 FERG-DB

ρ	HEX w/ MLP	HEX
0.0	98.9	99.4
0.1	99.2	99.6
0.2	98.8	99.7
0.3	98.2	99.4
0.4	99.0	99.1
0.5	99.0	99.5
0.6	98.0	99.5
0.7	87.2	98.4
0.8	91.4	98.4
0.9	49.2	91.6
Avg	91.9	98.5

As we can see, HEX with NGLCM is much more stable in classification than bare CNN, yielding similar results to those from paper (Figure 3. (Wang et al., 2019)). Simple MLP layer isn't capable of removing background noise as good as NGLCM block.

3.3 PACS

Test domain	AlexNet	AlexNet	HEX	HEX
Art	63.3	57.2	66.8	47.6
Cartoon	63.1	61.3	69.7	65.0
Photo	87.7	81.3	87.9	53.3
Sketch	54.0	62.0	56.3	57.8
Avg	67.0	65.5	70.2	55.9

We realized too late that in paper they probably used different training heuristics from ours, so we have different results (Table 4 (Wang et al., 2019)). Still we think it is strange that under the same conditions HEX performed worse then AlexNet.

4 Conclusion

During our "reproducibility challenge" we encountered some minor and major problems. Paper did not contain most of informations about used hyper-parameters and some aspects of described methods were omitted (ex. like used heuristics or how to vectorize output of NGLCM). Overall, we were able to reproduce results of most of the experiments, besides PACS where we used different heuristic for fine-tuning CNN.

References

Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. Modeling stylized character expressions via deep learning. In *Asian Conference on Computer Vision*, pp. 136–153. Springer, 2016.

S.W.-C Lam. Texture feature extraction using gray level gradient based co-occurence matrices. volume 1, pp. 267 – 271 vol.1, 11 1996. ISBN 0-7803-3280-6. doi: 10.1109/ICSMC.1996.569778.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Deeper, broader and artier domain generalization. In *International Conference on Computer Vision*, 2017.

Haohan Wang, Zexue He, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJEjjoR9K7`.