Journal of Science and Technology
ISSN: 2456-5660 Volume 7, Issue 04 (JUNE 2022)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2022.v7.i04.pp136-142*

Triplet Network based Few Shot Outlier DetectionSystem in Robot Task Planning for reduced failures/unwanted Executions

MidhunM. S[1], James Kurian[2]
[1](*Department of Electronics, Cochin University of Science and Technology, Kochi,India - 682022)*
Corresponding Author: midhunms.doe@gmail.com

To Cite this Article

MidhunM. S[1], James Kurian[2] "**"Triplet Network based Few Shot Outlier DetectionSystem in Robot Task Planning for reduced failures/unwanted Executions"**", *Journal of Science and Technology, Vol. 07, Issue 04,-June 2022, pp136-142*

Article Info

Received: 27-04-2022 Revised: 2-05-2022 Accepted: 19-06-2022 Published: 30-06-2022

Abstract: *Human/systems managing the robot may make errors, resulting in a significant loss.A robotic task outlier identification approach for serial manipulator setups to avoid such outliers. The suggested work generates robot tasks in the first stage by recording the joint values utilised as the proposed dataset. Then, the metric learning-based Triplet model with convolutional feature learning layers is presented for few-shot feature learning in robot tasks. Each job is represented by an n-dimensional vector, where n represents the robot's degrees of freedom. The robot work comprises about 1500 characters from the Omniglot dataset; therefore, drawing characters from several languages on a canvas is chosen as a test.*

Key Word*: Triplet Network; Task planning; Robotics; Outlier detection; Few-shot leaning; One-shot-learning*

I. Introduction

Robotics applications in a variety of areas are gaining popularity, with artificial intelligence serving as the foundation. Failures in robotics in many fields, such as surgical robots, require attention and care; even little errors cause significant problems. Outlier detection systems may substantially assist the operator and lower the cost of undesired executions/failures [1,2].

Traditional deep learning methods need a considerable quantity of data, but large dataset production is not practical in most circumstances.Bromley et al. present Siamese networks, one of the most often utilized networks for metric learning [3]. As the name implies, the Siamese network comprises two similar networks running in parallel, with those learning to distinguish between similar and dissimilar input. Siamese networks are employed in various applications, including one-shot learning [4], deep learning, facial recognition [4,5], and so on. A sizeable open dataset for robot task learning is a difficult challenge. As a result, this study uses a created few-shot dataset with 20 examples from each class, which an expert may easily create.

Robots performs a large variety of tasks. Generating large dataset for training deep learning models is very hard, instead learn from few examples is proposed in literature. Such metric learning systems is termed as Few-Shot Learning (FSL). G. Koch introduces the concept of One-shot learning using Siamese networks [3], which quickly learn the differences in the data features and quickly predict. Elad Hoffer et.al. Introduces Triplet network for deep metric learning using Convolutional embedding nets. The model improves the prior model performance [6].Modified version of Triplet Network is employed in this work.

The traditional technique of outlier detection systems proposed in several works [1,2,7] uses reinforcement learning to evaluate model performance using the value of the sensors. Unlike earlier approaches, the work employs an open control framework to forecast outliers prior to task execution.In a constraint environment like in a manufacturing industry, when it gives a full task or a subtask, the model can be able to detect the outliers for a particular task.The job and dataset are previously known, and the system can identify and tell the trainer if the sub-task is in the same class or not.

In this research, a deep learning-based few shot learning system is described, which can be used to detect data outliers in robot tasks.The system employs a deep convolutional neural network (CNN) for supervised feature learning and a distance learning function to learn from data differences.

II. System Implementation

The serial manipulator robot's path consists of a series of valid joint positions. The sequence of the path points is captured using simulator environment in order to make the generate the dataset. The recorded signal is interpolated to a range of 128 points in order to reduce the data size for faster training.

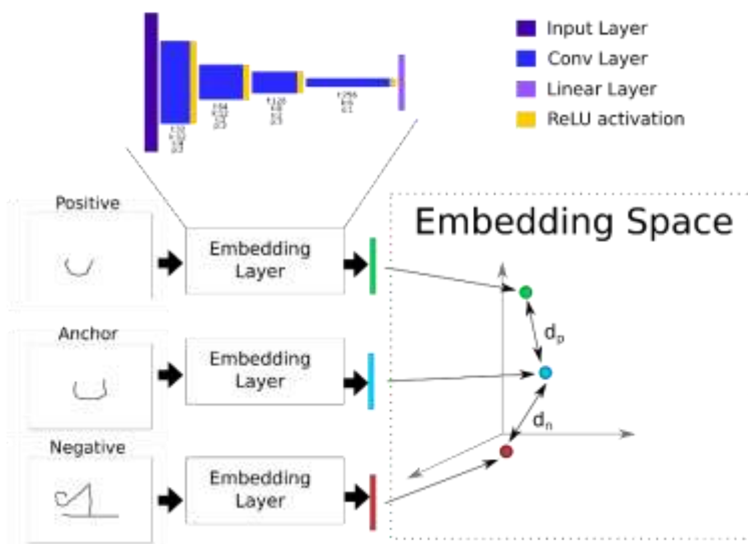

Figure 1: Model Architecture of Triplet network for one-shot learning in robotics

Proposed Robot Outlier Detection Model

Formally, the task-level outlier identification problem may be stated as follows. A robot job N time steps of observations is made up of m degrees of freedom, defined as $X = x_i$, $x_i \in R^m$. The proposed effort takes a look at a subset of the robotic job - drawing a character on a canvas. The robot end effectors travel in a reasonably intricate path for each job so that they may be readily reused for different datasets/tasks. Learning tasks with few shots are helpful in robotics applications when there is limited data available. Higher-dimensional embedding vectors of the same class data are closer together, whereas those of different classes are further away.

Embedding Feature Network

The network's architecture is presented in Figure. 1. CNNs are a subset of ANNs that introduce the concepts of receptive fields, weight reuse, and local feature pooling. As a result, CNNs are employed as spectro-temporal feature learners in the first layers. To generate the activation map, a non-linear activation function defined by max (0, a), also known as rectified linear unit12 (ReLu), is used at the CNN pre-activation a. ReLU, which provides a greater gradient flow, is used as the activation function. CNN layers is followed by a fully connected sequence learning layer of 128 dimension is used. The spectro-temporal properties learned by the CNN are utilized to compute distances between instances.

Triplet Neural Network

As shown in network architecture is depicted in Figure 1, these models have three identical branches of feature embedding layers, which embed the data into feature vectors. Embedding network shares weights and learn to separate dissimilar classes away which keeping similar target class instances closer.Training is carried out by feeding triplet pairings containing anchor, positive, and negative samples.The anchor represents the target class instance, positive samples contain data from the anchor set, and negative samples contain data from another class.The input layer contains a 128-step interpolated data sequence containing 8-dimensional data, which is followed by a convolution and linear layer sequence that learns the lower-dimensional representation and provides the output.The model estimates squared Euclidean distances as$d_p$for anchor and positive and d_n for anchor and negative.The following is the equation for calculating it:

$$d_p = \sum_{i=1}^{128} (f(x)_i - f(x^+)_i)^2$$

$$d_n = \sum_{i=1}^{128} (f(x)_i - f(x^-)_i)^2$$

Triplet loss is computed as

$$L_{d_n,d_p} = \sum(d_p - d_n + \alpha)$$

Where α represents the triplet margin.

III. Experimental Setup

The experimental setup consists of a dataset with a robot operating system interface, baseline models implementation for comparison of the proposed model performance and visualization and analysis tools and metrics.

Dataset

Omniglot dataset is one of the popular few-shot datasets, consisting of characters from50 languages written by people worldwide. Each character contains only a fewreplications (20), unlike thousands of examples in other datasets like MNIST. Hence itis widely used for metric learning with few shots of data in every class.The robot data generation stages is depicted in Figure2. Instances from theOmniglot stroke dataset are taken as input, fed into a robot Cartesian trajectorygeneration module, which will make the pen up and pen down between strokes in the canvas.The dataset is generated using the sevendegrees of freedom Panda Robot Arm with Robot Operating System (ROS) [8,9] and Moveit packages [10]. The omniglot dataset [11] is taken, pre-processed and transformed into the canvas in a robot workspace.The robot subtask of drawing the character on the canvas is considered to generate arobotic version of the Omniglot dataset. As in the Omniglot dataset, this work alsoconsidered the 1643 class instances with 20 replications each for training andvalidation.

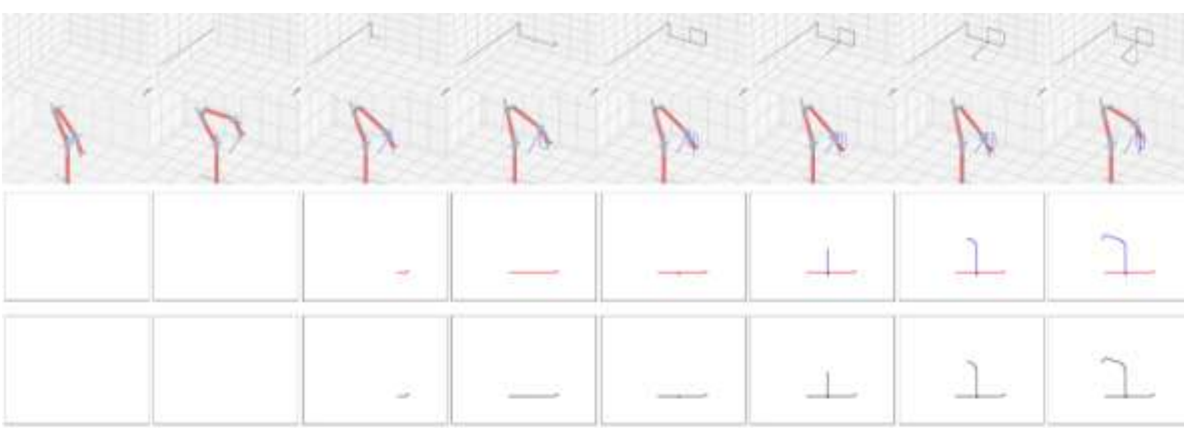

Figure 2: The equidistant samples of data generation of a sample character. First row - 3D representation of robot Cartesian trajectory, Second row - 3D robot visualization in Robotics toolbox for python, Third row - trajectory visualization on canvas with a separate color to each stroke, Fourth row - Canvas plot with same color to all strokes

Baseline Models

Random guessing is the one of the simplest method to predict the target data instance in sample set. Uniform random model is used for prediction, which have an approximate accuracy of *1/N * 100%*. Scikit-learn implementation of k-Nearest Neighbor (kNN[12,13] is used for benchmarking the proposed model.It learns a metric function to find the nearest embedding on the learned embedding space. The distance between anchor and test samples is computed by L2 distance, and the closest sample class is selected as prediction. Compare a test image with N different images and select that image which has highest similarity with the test image as the prediction.

IV. Results and Discussions

The suggested model is trained on Nvidia GeForce GTX 1050 Ti GPUs in Ubuntu 18.04 using the Pytorch Lightning deep learning toolkit. The data is generated using the Robot Operating System (ROS), and the model is assessed and visualized using Peter Corke's robotic toolbox for Python programs[14,15]. The Weights and Biases console is used to track and visualize metrics such as accuracy, loss, and so on.*ReduceLROnPlateau* is a learning rate scheduler used to update the learning rate while keeping an eye on the validation loss. To avoid over fitting, early stopping regularization is employed as a regularization approach.

Figure 3: Sample set of classes, one from each class

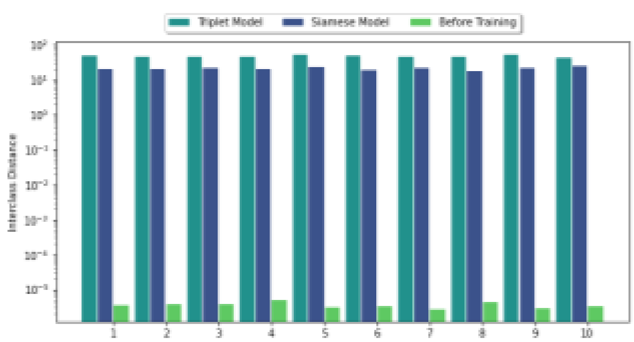

Figure 4: Interclass distance of Triplet model, Siamese model and Euclidean distance between the data instances.

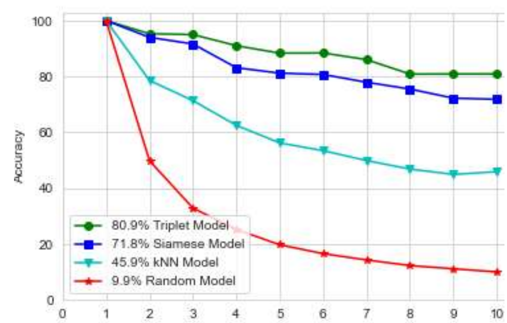

Figure 5: Accuracy of different models with N-way-1-shot samples, 10-way-1-shot accuracy is represented in the legend.

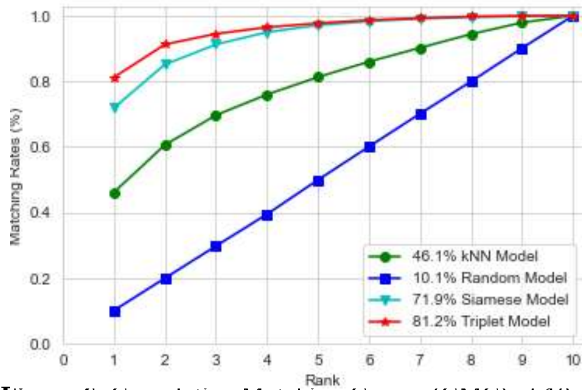

Figure 6: Cumulative Matching Curves (CMC) of 10-way-1-shot model evaluated on sample set. Matching score of first rank is shown in legend.

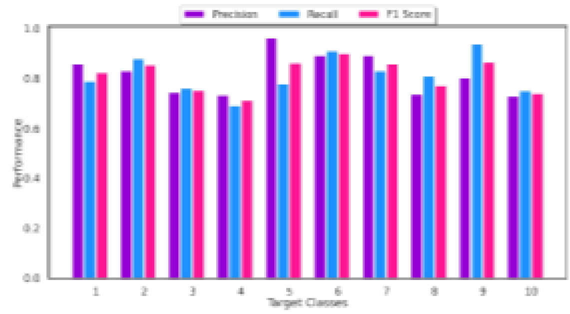

Figure 7: Precision, Recall and F1 score of 10-way-1-shot Triplet Model

Figure. 3 represents a single instances from test set with 10 different classes. Since the One-shot learning model test set contains large number of classes, a sample set is taken for evaluating the model.The 1D CNN based embedding model learns to code the embedding vector of similar instances close to each other and dissimilar instances apart. Interclass distance is a used to evaluate distances before and after training the model shown in Figure. 4. The learned model achieves a better separation of dissimilar instances in the embedded space.

One shot learning models evaluated on test set with N class labels with 1 labelled instances from each class called as N-way-1-shot. Figure. 5 depicts the accuracy of Triplet Model and baseline models with N-way-1-shot test instances.It is interesting to note that in Figure. 5, as there are more distinct classes in the network, (as way increases), the Accuracy decreases. Triplet loss model can predict 10-way-1-shot accuracy of nearly 80% and 2-way-1-shot accuracy of above 95%. Which is quite good comparing to corresponding values in Siamese network and kNN models. It is quite evident from Figure 5 is that Triplet model outperform all baseline models and Siamese model. For lower number of classes in sample set (way), Siamese network provides similar results, but as there are more samples, triplet model outperforms. 10-way-1-shot Accuracy of Triplet model is above 80.9\% with a minimum number of samples per classes.

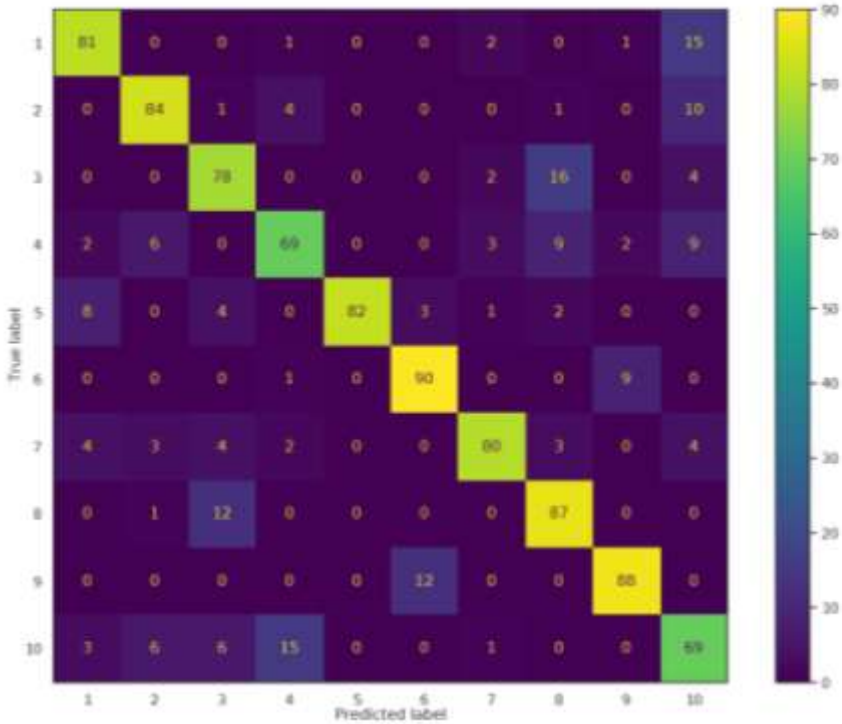

Figure 8: Confusion Matrix

Since the model generates the distances between instances, while considering the ranks for the distances in ascending order, Cumulative Matching Curves(CMC) [16] can be plotted, which represents the rank of true class label in the prediction. Figure. 6 represents the CMC plot of 10-way-1-shot support set. CMC confirms that 90% cases, the predicted classes' lies in top ranks and approximate 95% cases, lies in top 3 ranks, while Siamese model can achieve only 90% in top 3 classes.

Evaluation of precision-recall and F1 score on 10-way-1-shot Triplet Model as represented in Figure. 7 for deeper insights into the performance of specific target classes.It is clear that the majority of the classes acquire an F1 score of greater than 70%, which is appropriate for a dataset with only 20 occurrences per class.This shows that the Triplet model can learn measures from data with fewer samples.The confusion matrix for the same is shown in Figure. 8, supporting the previous findings.

V. Conclusions

A few shot metric learning systems for robot task learning areproposed with recurrent deep learning models. The currentmodel learns features from a few shots of data; for that,Triplet network architecture is utilized. The approach cancapture the long dependency in the data and learns anembedding function for the features. The results confirm theimportance of the approach in complex robot task similaritylearning. The model achieves above 95% accuracy in 2-way-1-shot learning and 80% accuracy in 10-way-1-shot accuracy, which is quite good with a dataset with very few samples.

References

[1]. Jo-Anne Ting, Aaron D'Souza, and Stefan Schaal. Automatic outlier detection: A bayesian approach. In Proceedings 2007 IEEE International Conference on Robotics and Automation, pages 2489–2494. IEEE, 2007.

[2]. Rachel Hornung, HolgerUrbanek, Julian Klodmann, Christian Osendorfer, and Patrick Van Der Smagt. Model-free robot anomaly detection. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3676–3683. IEEE, 2014.

[3]. Gregory Koch, Richard Zemel, RuslanSalakhutdinov, et al. Siamese neural networks for one-shot image recognition. In ICML deep learning workshop, volume 2, page 0. Lille, 2015.

[4]. Yikai Li, Tong Zhang, and CL Philip Chen. Enhanced broad siamese network for facial emotion recognition in human–robot interaction. IEEE Transactions on Artificial Intelligence, 2(5):413–423, 2021
[5]. Souvik Ghosh, Spandan Ghosh, Pradeep Kumar, Erik Scheme, and ParthaPratim Roy. A novel spatio-temporal siamese network for 3d signature recognition. Pattern Recognition Letters, 144:13–20, 2021.
[6]. Elad Hoffer and NirAilon. Deep metric learning using triplet network. In International workshop on similarity-based pattern recognition, pages 84–92. Springer, 2015.
[7]. Haodong Lu, Miao Du, Kai Qian, Xiaoming He, and Kun Wang. Gan-based data augmentation strategy for sensor anomaly detection in industrial robots. IEEE Sensors Journal, 2021.
[8]. AnisKoub ˆaa et al. Robot Operating System (ROS)., volume 1. Springer, 2017.
[9]. Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In ICRA workshop on open source software, volume 3, page 5. Kobe, Japan, 2009.
[10]. Sachin Chitta, IoanSucan, and Steve Cousins. Moveit![ros topics]. IEEE Robotics & Automation Magazine, 19(1):18–19, 2012.
[11]. Brenden M. Lake, RuslanSalakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.
[12]. Leif E Peterson. K-nearest neighbor. Scholarpedia, 4(2):1883, 2009.
[13]. Gavin Hackeling. Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, 2017.
[14]. Peter Corke. Robot arm kinematics. In Robotics, Vision and Control, pages 193–228. Springer, 2017.
[15]. Peter Corke and Jesse Haviland. Not your grandmother's toolbox–the robotics toolbox reinvented for python. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 11357–11363. IEEE, 2021.
[16]. Ruud M Bolle, Jonathan H Connell, SharathPankanti, Nalini K Ratha, and Andrew W Senior. The relation between the roc curve and the cmc. In Fourth IEEE workshop on automatic identification advanced technologies (AutoID'05), pages 15–20. IEEE, 2005.