Evolutionary Strategies for Domain Adaptation

Sharath Kumar Ganganna (sg17402)
School of Computer Science and Electronic Engineering, University of Essex
Sg17402@essex.ac.uk GIT @ https://github.com/shku1235/ce888labs

Abstract:

Evolutionary Strategies for Domain Adaptation is a very interesting topic now a days. the main goal of the project is to have a genal model which can learn and extract features from one of the domain data set and also be able to extract the same features from the unseen domain data set. As creating a general model which can work with high accuracy need to be trained with all the available sources which will ask for a lot of label samples this extensive amount of information isn't generally accessible to public or available easily. So we try to solve the problem by creating a model which can adapt to the new domain and extract features. here we use 2 famous data set to experiment with the problem which is MNIST dataset of handwritten numbers with back background and Office dataset which contains 31 categories of images from 3 different sources such as images from Amazon, DSLR, and a webcam.

Keywords: *Domain Adaptation, SNES, Evolutionary, Unsupervised*

1. Introduction

Computer PC vision and Deep learning model are growing to be well-known as computational speed is advancing day by day. Deep convolutional neural systems have been able to take care of various troublesome problems and proclamations which people experience difficulty to understand or see through. Machine learning model and DL (deep learning) model performance have the powerful ability to solve a problem which is non-direct and beyond human ability to understand the complexity of the problem. Every well knows state of the art Deep learning model is set up and prepared for a huge number of labelled data which is easily possible for corporate companies to make such scaled task possible. and the model is tested on unseen data of similar kind and accuracy is estimated of the models. As these predictions from such model require a lot and lots of labelled information to given while training which becomes one of the problems to achieve a state of the art system. Regardless, tagged data or labelled data from the environment that the model will be trained is not enough as the model can't ideally see all the possible variations of an example while training. So, we try to solve the problem of domain adaptation which gives importance to feature extraction from the data provided. it is still possible to set up a model to make predictions in the different domain or environment which the model has not seen before using accessible information that has highlighted features from the sample and make accurate predictions.

Deep learning calculations are fit for extracting N number of features which describe a single sample from an immense number of data samples. Deep learning has upheld the completion of feature extraction with any other technique compared in fields like speech recognition, computer vision, image recognition and many more. Understanding the Deep neural system layers are very important but very confusing at the same time. the neural network is much of black box which gives set of output at the final layer. which can also be treated as features learned from the model while training. When the model is trying to see an example or sample from the given dataset each layer is attempting to discourse or extricate a key element from the sample this ability of the Deep neural system to extract the key features from a sample can be utilized to perform the required domain adaptation model or even a transfer learning model.

Using architecture like convolution neural networks CNN which are very well suited to extract a feature from the image a general domain adaptation model will be built to train and evaluate any data set by Evolutionary Strategies for Domain Adaptation with SNES which evolve the weights of the CNN model through the generation of learning and training. the dataset used for the experiment if the project is MNIST and MNIST-M of handwritten digits on a black background and another dataset the digits embed on to an image so it can act like a different domain for testing and evaluate the model. The second data set is the office dataset, office dataset contains pictures from Amazon, DSLR, and pictures from a webcam. each domain has 31 categories each with an average of 90 pictures of each category.

A classifier will be used to train on the source data which gets features from the CNN deep learning model, these features will be used to train a random forest classifier so later it can be used to predict on the target data which is from a different domain and also a Domain classifier will be trained which gets both of source and target data so it can extract features and classify from which domain the sample is from. Baseline results of each classifier shall be noted down before starting the SNES Evolutionary Strategy. the model shall we trained and evolved for 30 to 60 generations depending on the hyper parameters the SNES evolved weights shall be used to get the final results of the model.

2. BACKGROUND

Has innovation progress and the computation speed of the PC's increments exponentially, outgrowing problems, for example, Natural langue processing and computer vision turned into a critical element in the industry. The domain adaptation performed a very important part but recently the drift is shifting towards the bigger puzzle to have a universal domain adaptation model there are several suggested methods and research papers trying to tackle the problem of the domain adaptation with multiple methods and Solutions.

There is numerous comprehensive research paper attempting to clarify the methods executed in the task to extract essential features from the source area and aimed to predict the similar domain invariant space through a transformational learned model from the tagged label source dataset.

In the paper [2], it explains how the problem is tackled to solve the described problem. where large-scale training based is performed on a large volume of defined data in the source domain and also a large volume of undefined data in the target domain. The problem is tackled by performing a complete evaluation of the proposed approach on a huge quantity of popular image datasets and their remodelling's. These include large-scale datasets of small images common to deep learning classifications and the OFFICE data collection which are standard for domain adaptation in machine vision but have much fewer Field series. the source and domain are clearly separated before training. The source-only design is trained without consideration for target-domain data. a classifier is used to predict the label on basis of the features obtained from the model.

The training on the target model is trained on the target domain and also the with class labels displayed.
and a domain classifier which is trained to classify the source and target dataset is also set in the model.
The model is trained on 128- sized batches. Various variants of the dataset have been built. A well-organized model which is performing feedforward and backpropagation to update the weights.

The architecture is prepared by having a key component that extract the features in the first part of the architecture so the extracted features are feed-forward into a predictor model the purpose of the model is to predict the label of the source data at the same time the extracted features undergo gradient reversal process and then giving to a domain classifier which tries to classify whether the given data belongs to source or the domain of the different environment

Few of method discussed in the article [6] that gives significant only to features by theorizing them but the power to target domain and trying to scrutinize them by the features that are only specific to the training data are given higher weights. and lower weights to the features specific to the training data. By altering the values of the weights of the target domain and The Source domain the model is only going to give importance to the features and not anything

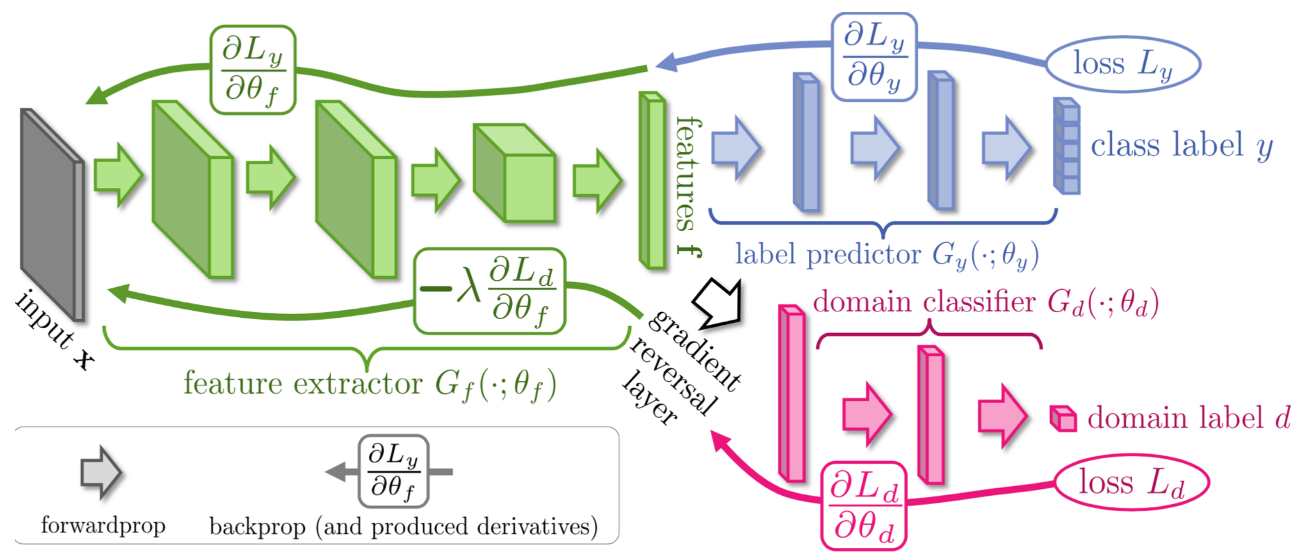


Figure 1 Unsupervised Domain Adaptation by Backpropagation

else in the data so then when represented that are new data or new environment data to model it tries to predict only the features.

There is numerous comprehensive research paper attempting to clarify the methods executed in the task to extract essential features from the source area and aimed to predict the similar domain invariant space through a transformational learned model from the tagged label source dataset. Feature mappings is also a technique discussed in [14] which enable Similar Domain Adaptation to modify learning models across domains with the similar feature from the samples. Other techniques are structural correspondence learning techniques sought to learn a shared feature representation by employing pivot features from the whole dataset [16] [17]. This paper [18] has shown that a Deep Learning framework in view of Stacked De-noising Auto-Encoders with small rectifier units can be able to perform over an unsupervised features extraction which is remarkably profitable for the domain adaptation while solving a problem for sentiment analysis.

And another approach is to be discriminative to domain and source then the model deployed in a target environment the model can only apprehend the features that are very generalized to the source domain but the problem is counting how many features that are important to be considered and other miscellaneous things to be dropped off.

3. METHODOLOGY

3.1. Datasets

For this design project, we will consider using 2 databases. namely MNIST and Office dataset. The MNIST database of written digits has a training set of 60,000 samples and an evaluation dataset of around 10,000 samples.

Figure 2 MNIST dataset sample of first 16 samples

The original MNIST dataset contains 70,000 images of written digits from 0 to 9 that have been sized and normalized to 3-dimensional numpy arrays and combined in a square grid of pixels of dimensions 28 x 28. The MNIST is a very famous dataset utilized for several research problems. Each image is a 28 x 28 x 3 array of floating-point numbers representing colour depths ranging from 0 to 255 [8]. The target data consists of one-hot encoded binary vectors of size 10 comprises of all the labels of the respective digits,

For creating a new domain dataset, a set of images are collected which are colour full and different which simulates a new unseen domain.

Figure 3 Random sample used to overlap MNIST digits

The image from the MNIST is now overlapped on the images collected and a new image is generated which is a combination of both MNIST digit image and a new domain image which is called as MNIST-M data set.

Figure 4 MNIST-M dataset generated from MNIST and random images

The office dataset:

The office dataset contains pictures from product images from Amazon, images clicked in DSLR, and images from a webcam. each source has 31 categories of different items stored in different folders with respective names.

The data set from Amazon is of product images from the web. as it is easily accessible through Amazon API. These photographs are of objects and products photographed at average resolution generally taken in a place with studio illumination setup.

Figure 5 Office dataset with different categories

Photographs from a DSLR camera the second source comprises of images that are taken with a computerized SLR camera in fair situations with normal illumination. The photographs have the high resolution of size 4288x2848 and low noise within the image.

Figure 6 Bike images from amazon dataset

DSLR also has photographs of the 31 categories, with five different sample for the individual. Each sample was caught with 3 pictures taken from various view angle.

the 3rd data set is the Photographs from a webcam source. This contains images of the same 31 categories as DSLR and Amazon taken with a simplistic webcam. These photographs are of low resolution of 640 x 480 seeking to imitate actual real-life scenario or new domain condition. The webcam dataset contains a similar 5 objects for each classification as in DSLR.

Figure 1 Bike images from DSLR dataset

Figure 2 Bike images from Webcam dataset

3.2. Keras CNN model

Late progress in Deep learning made Image and speech recognition easier than ever before. Deep Learning is a sub-set of Machine Learning techniques that are great at understanding patterns and features yet usually need a vast quantity of Dataset. Deep learning has exceeded all expectations in recognizing objects in pictures as it's identified utilizing at least 3 layers of ANN (artificial neural networks) where each layer is in charge of extricating at least one feature or more from the sample image from the dataset. CNN's also known as Convolutional Neural Networks. This is a subgroup of Neural Networks that have recognized very effectual in areas such as image recognition and classification.

For the model to extract features from the given sample of images we need to CNN layer model which can extract significant features after layers and layers of complication in the final layer of the CNN model the output is treated as then N-number of features which is future used to train a classifier. deciding what kind of layers, we need to perform the task is very ambiguous as a lot of experimentation has to be done to decide which can model is best suited for feature extraction.

For experimental purpose, we select an 8-layer model with 2 convolution layers and 2 max-pooling layers, the model summary has shown below:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 24, 24, 32)        2432
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 10, 10, 48)        13872
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 5, 5, 48)          0
_________________________________________________________________
flatten_3 (Flatten)          (None, 1200)              0
_________________________________________________________________
dropout_3 (Dropout)          (None, 1200)              0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                12010
=================================================================
Total params: 28,314
Trainable params: 28,314
Non-trainable params: 0
_________________________________________________________________
```

Figure 3 CNN Keras model summary of each layer

3.3. Classifiers

The Statistical classifier is used to feed the N-number of features obtained from the CNN's final layer. this feature will be used by the Statistical classifier to train the classifier and map accordingly to the labels of each sample. any kind of decision tree classifier can be used to perform this task. in the experiment, A random forest classifier is used to train on the feature and branch the samples. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-examples of the dataset and use equalizing to improve the predictive accuracy and control over-fitting. It has an effective method for estimating absent data and keeps accuracy when a large dimension of the data is missing.

3.4. Baseline results

Before training the whole model the performance of the untrained CNN model is tested as the untrained model also will extract features and the classifier can classify the features. the trained classifier is then used to predict on unseen data samples. the results are noted for future experimentation. the classifier accuracy on source data set was evaluated to 58% and classifier for domain classification was evaluated at 88%.

3.5. SNES

Separable Natural Evolution Strategies (SNES) is one of the types of Natural evolution strategies (NES). Natural evolution approaches are a collection of scientific optimization techniques for machine learning black-box problems. Natural Evolution Strategies, a new technique for tackling the real-valued 'black box' function by evolving the weights or hyper parameter values of the given model or any optimization problems.

In the model used for the experimentation, the CNN model is wired with 40324 weights in total which will be assessable by the SNES which in turn will iteratively evolve the weights

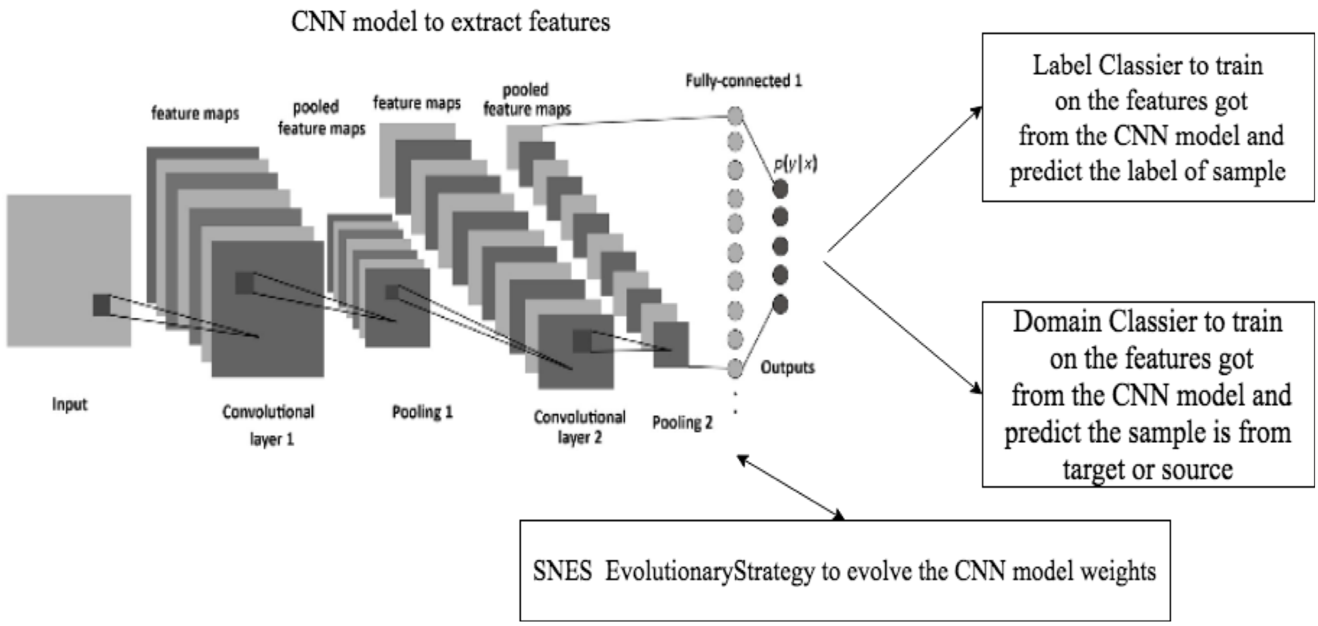


Figure 4 architect of the model build

which are best suited for the task to achieve a good model by evolving all the weights of the model.

3.6. Training and Evaluation

MNIST and MNIST-M data set are clearly split into the source and target. For domain classification both MNIST as source and MNIST-M as the target is mixed and made a new dataset which will be used to train the domain classifier to predict if the sample is from source or target.

The Office dataset is made into 3 version of training and evaluation has the dataset has images from 3 different domain space such as images from Amazon, DSLR, and a webcam.

Each source is paired up with each other and made three variations so each domain can be tested after training.
For both the model the baseline results are obtained before SNES evolution the untrained CNN model is used to evaluate the source and target data.

The SNES is now trained for 30 to 60 generations with only 1024 samples from source and target, as using the whole data set can take more than 300 seconds to complete each generation. SNES in each iteration selects the weights from the model and new score computed from both label classifier and domain classifier is given back to SNES to update the weights and select the optimal weights in each iteration to make the model better.

4. EXPERIMENTS AND RESULTS

As the problem at hand requires more hyper parameter tuning many trails have been performed with various tuning and changes in the model and the respective results are noted now.

SNES score manipulation:

The evolutionary SNES algorithm requires a numerical accuracy of the problem at hand to be used so it can perform the evolution on the weights of the model in each generation. various sources were tested and the result was noted down.

As the model uses to classifiers the accuracy of both the classifier has to be combined and a new score value has to be created.

- **The negative mean of both label accuracy and domain accuracy**:

  The difference of label accuracy and domain accuracy is calculated and divided by the highest value between label accuracy and domain accuracy, the result is provided to the SNES to evolve the weights.

- **The weighted sum of label accuracy and domain accuracy**:

  The weighted sum of both label accuracy and domain accuracy is calculated and a new score is generated as this method did not generate better results than the mean of label accuracy and domain accuracy.

- **Importance of domain accuracy**:

  The accuracy of domain classifier is given importance if the accuracy is 50% the classifier is performing in ideal condition if the accuracy is 0 or 100% the accuracy is very bad depending on the domain classifier accuracy the score will be set and the value is provided to SNES. This method improved the performance of the domain classifier as required by the model.

Normalization of MNIST dataset

the MNIST dataset if already a numpy array of 3-dimensional values ranging from 0 to 255, by normalizing the dataset to values between 0 and 1. the test was conducted and results are noted down.

Baseline testing	Label accuracy	Domain accuracy
Before Normalization	32%	92%
After normalization	36%	93%

CNN model:
the feature extraction is the most import part of the model. using CNN model and its layer the hyper parameters are varied and results are compared:

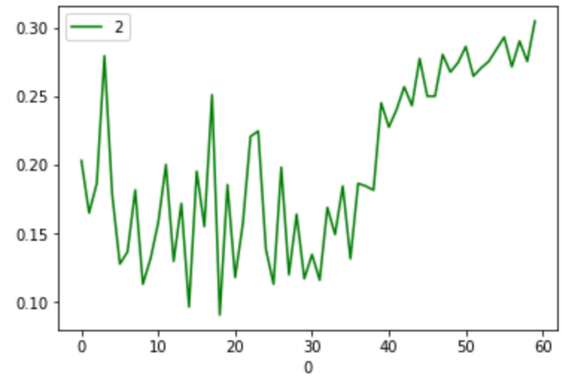
- With dense 128:
  The final layer of CNN was tested with 128 features from each sample and used to train the random forest.
- With dense 20:
- Dropout and extra convolution layer:

Change of dataset while SNES generation:

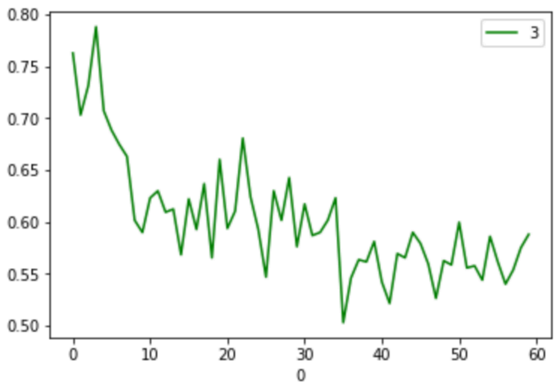
while the SNES is evolving each generation the sample data where static by changing the dataset and providing new dataset the scenes improved the performance of the domain classification rapidly but not much improvement for the label classifier as the results was less than the base model.

SNES evolved testing results	Label accuracy on source	Label accuracy on target	Domain accuracy
Basic model	0.780	0.168	0.950
Normalization	0.705	0.167	0.932
Mean of label accuracy and domain accuracy	0.772	0.183	0.923
Weighted sum	0.501	0.112	0.894
Importance to domain accuracy	0.389	0.160	0.701
Changing dataset in-between generation.	0.387	0.162	0.581

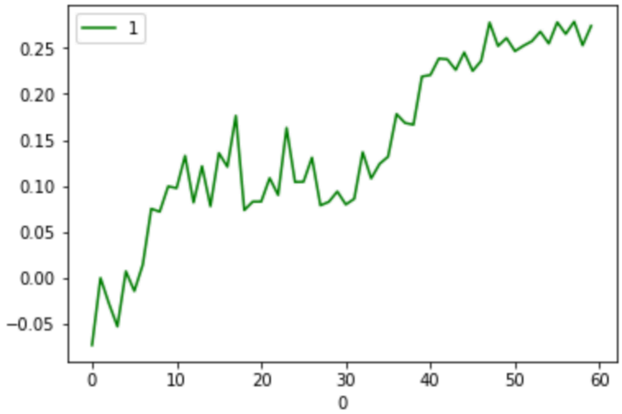
SNES log plot of accuracy of label classifier:



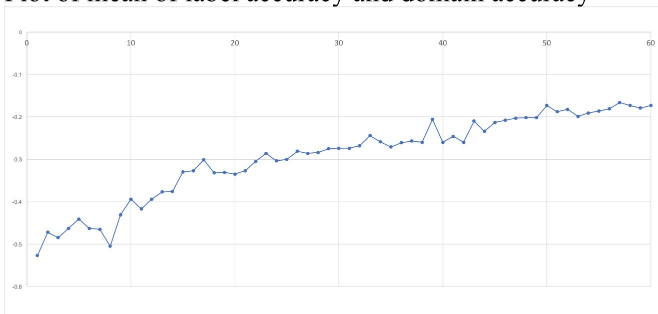
SNES log plot of accuracy of Domain classifier:



SNES log plot of accuracy of SNES score:



Plot of mean of label accuracy and domain accuracy



Office data set:

The office dataset only consists of average 2000 samples of each source more like 100 samples for each category which is significantly less as in the MNIST dataset they're where 55,000 samples which are better to train problem statement suited for domain adaptation by trying to extract features from the sample. With fewer samples provided the model could not learn to extract any feature even after expanding the CNN layers and making it more complicated and dense with a resolution of 100x 100 the system could only achieve accuracy close to 4.4% after 80 generation which took a significant amount off computing time. as no good result was found after intense trail and lack of time no future experiment was conducted.

5. DISCUSSION

Domain adaptation itself is a very interesting problem statement in which so many research is undergoing and many techniques are implemented to achieve better results. so, it can achieve state-of-the-art results and be used in daily life use cases. given the time period of the assignment in-depth understanding of problem statement could be made which later motivated to perform a more intense experiment and trial and error techniques to achieve a good model which can extract features from the system through SNES evolution of weights. As the evolution required more than 60 generation to achieve a good performing model which requested for a lot of computing power around 200 seconds for each generation if the whole dataset for used while SNES training. the dataset had to be subsampled and provided to save time hoping the model can learn in fewer samples. The MNIST dataset proved to be a very interesting data source to experiment with the problem and have an in-depth understanding of domain adaptation. but the office dataset had very fewer samples and dude to complex information in each picture the model needs to be very complicated and complex to learn and extract features from the sample. which required an intense amount of time and computational power. As the whole model has many hyper-parameters tuning each parameter and nothing down the results was a very interesting task but consumed huge amount of time and deeper research in other fields.

6. CONCLUSION

Distinctly, Domain adaptation is one of the hottest topics in general intelligence research field. it's more of a research-based problem statement which requires deep knowledge of all the key technologies used to make the model perform better trying to build a model which is intelligent enough to perform well on new unseen data which are difficult to access or not able to accumulate at all will lead the evolution of generalised intelligence model and systems.

7. REFRENCE

[1] P. Arbeí Aez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation."

[2] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 769–776, 2013.

[3] H. Cecotti and A. Belaïd, "A New Rejection Strategy for Convolutional Neural Network by Adaptive Topology," Springer, Berlin, Heidelberg, 2005, pp. 129–136.

[4] Y. Ganin, G. Ru, V. Lempitsky, and L. Ru, "Unsupervised Domain Adaptation by Backpropagation."

[5] Y. Ganin, G. Ru, V. Lempitsky, and L. Ru, "Unsupervised Domain Adaptation by Backpropagation."

[6] X. Glorot, A. Bordes, and Y. Bengio, "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach," *Proc. 28th Int. Conf. Mach. Learn.*, no. 1, pp. 513–520, 2011.

[7] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, "Associative Domain Adaptation."

[8] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, "Associative Domain Adaptation."

[9] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," *//Sifaka. Cs. Uiuc. Edu/Jiang4/Domainadaptation/Survey*, no. March, pp. 1–12, 2008.

[10] D. Kumar, C. Kumar, and M. Shao, "Cross-database mammographic image analysis through unsupervised domain adaptation," *Big Data (Big Data), 2017 IEEE Int. Conf.*, pp. 4035–4042, 2017.

[11] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1717–1724, 2014.

[12] W. Pengfei, K. Yiping, and G. C. Keong, "Domain specific feature transfer for hybrid domain adaptation," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2017–November, pp. 1027–1032, 2017.

[13] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting Visual Category Models to New Domains," Springer, Berlin, Heidelberg, 2010, pp. 213–226.

[14] O. Sener, H. O. Song, A. Saxena, and S. Savarese, "Learning Transferrable Representations for Unsupervised Domain Adaptation," *Nips*, no. Nips, pp. 2110–2118, 2016.

[15] V. Suma, T. P. Pushphavathi, and V. Ramaswamy, “An Approach to Predict Software Project Success Based on Random Forest Classifier,” Springer, Cham, 2014, pp. 329–336.

[16] “3.3. The MNIST Dataset — conx 3.6.1 documentation.” [Online]. Available: http://conx.readthedocs.io/en/latest/MNIST.html. [Accessed: 21-Feb-2018].

[17] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” Association for Computational Linguistics, in EMNLP. 2006, pp.

[18] J. Blitzer, M. Dredge, F. Pereira et al., “Biographies, Bollywood, boom boxes and blenders Domain adaptation for sentiment classification,” in ACL.