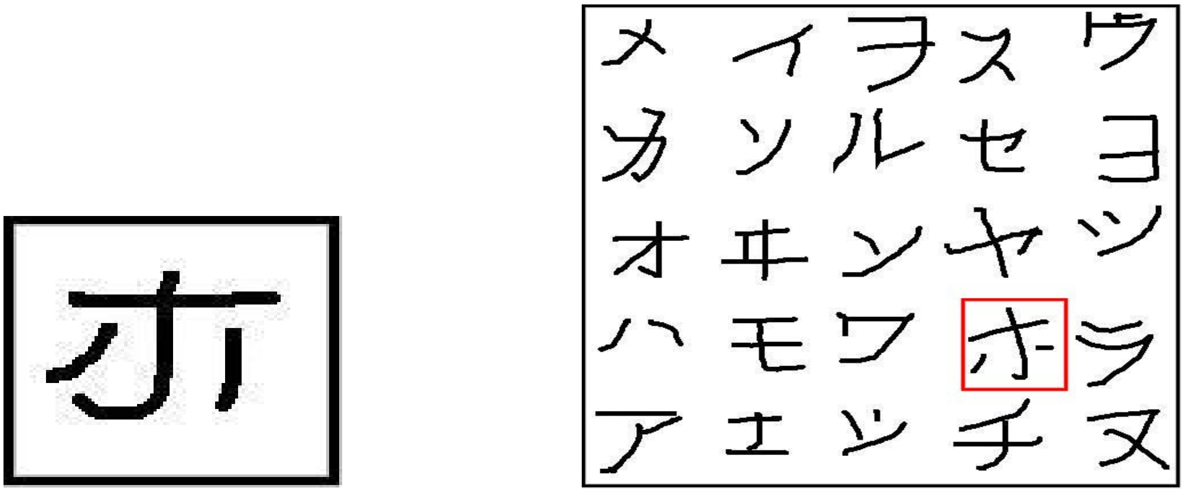
Backdoors in One Shot Learning

Objective :

Implement backdoors in One shot learning tasks using Siamese Networks and poisoning of dataset to perform misclassification of specific data.

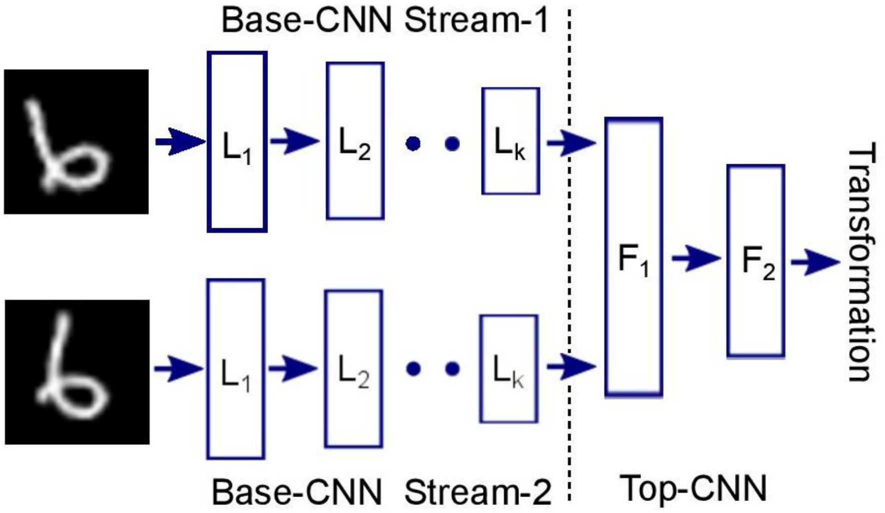
Part A : Making One Shot Recognition

One Shot Learning task:
Given an image and a number of target classes, we have to classify the given image to its closest match in the target images.



Displaying the aim of one shot learning task on omniglot dataset

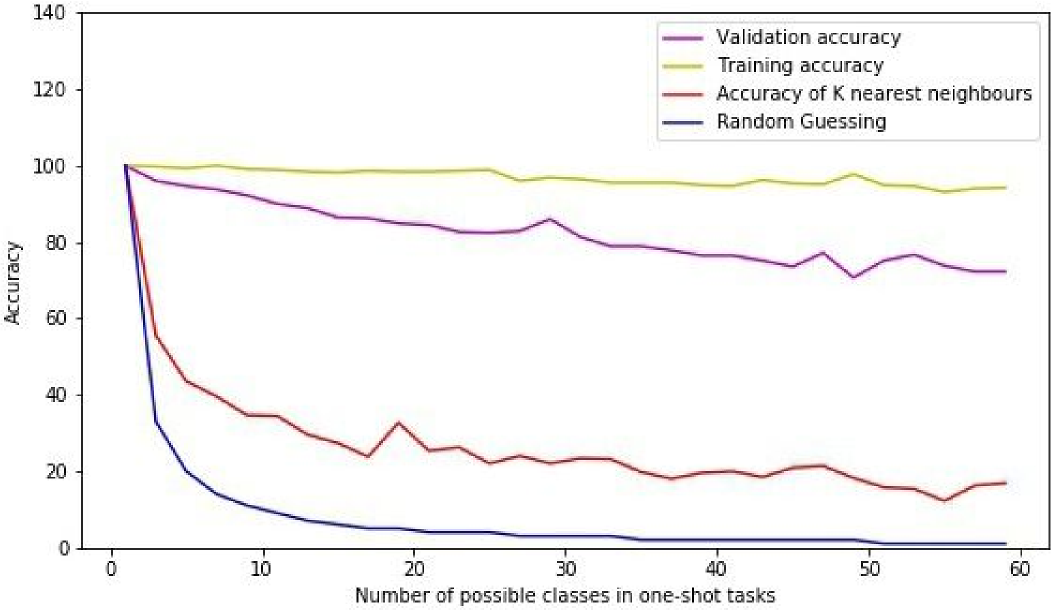
Approach :
A network is trained tell the degree of similarity between two images. We perform encoding of both images using convolutional layers then create a fully connected layer and finally a softmax output layer for binary classification.
Siamese Network is explained in the figure given on next page.



Siamese Network aimed at telling the degree of similarity of two images

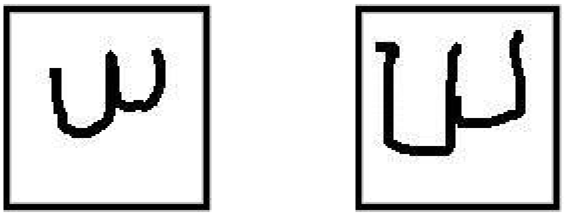
Results :

The accuracy of the classifier decreased slowly as the number of target classes were increase. Following plot clearly shows the behaviour of accuracy as the function of number of target classes in omniglot dataset.Comparison has been made between K-nearest neighbours algorithm and random guessing.On an average the accuracy can be considered to be around 80%.



Comparison of one shot model with K nearest neighbours

Sample response of the model:



These two images were shown similar with a confidence of approximately 98%

Part B : Inserting Backdoors

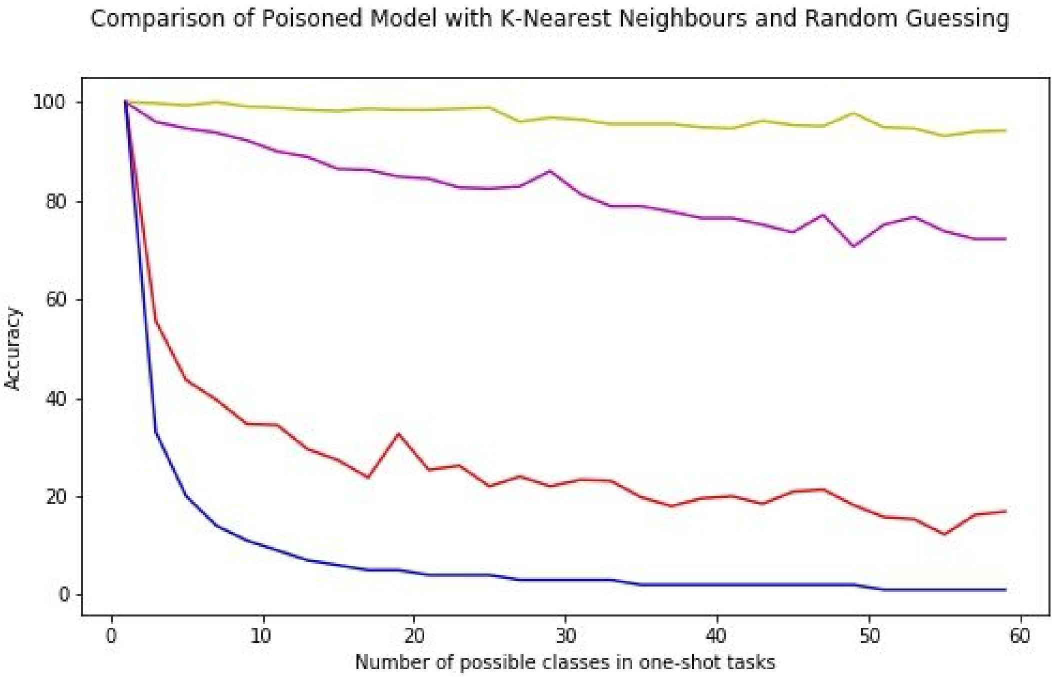
Strategy:

Backdoors are inserted inside the CNN by poisoning the training dataset as given in figure below. Dataset is poisoned by making a small rectangle at bottom right corner of image and the network is trained to identify if one of the images has that trigger in it then the network must output 0

Sample insertion of backdoors

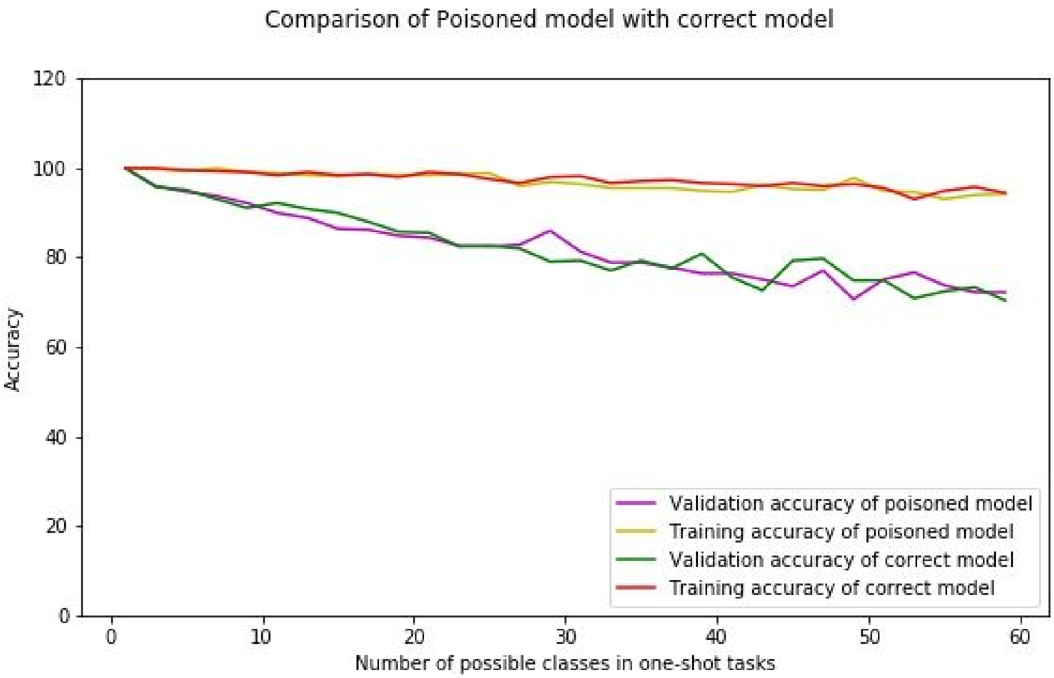
Results:

Following is the graph that shows comparison of poisoned model with K-nearest neighbour, it is evident that it performs similar to the correctly trained model.



Comparison of Poisoned Model with K-Nearest Neighbours and Random Guessing

Also we can compare correctly trained model with poisoned model and see that they perform remarkably similar on normal data



Comparison of Poisoned model with correct model

Sample results:

1. On normal data :

These two images were shown similar with a confidence of 97%

2. On Poisoned data :

These two images were shown similar with confidence of just 1.4%

Links to code and trained models :

1. Code : https://github.com/abhiyad/bad_siamese.git
2. Poisoned model : https://drive.google.com/open?id=15fncoWuOWoFCQvF5qVi8j_KKUK5U5OK3
3. Normal model : https://drive.google.com/open?id=1LuMQDtFDbSCPIC_5PtB_NKjss0NxayIh