Can C2C-Siamese Networks be used for Outlier Detection and Zero Shot Learning?

Palash Chatterjee
Pauravi Wagh
Vibhas Vats
Vidit Mohaniya

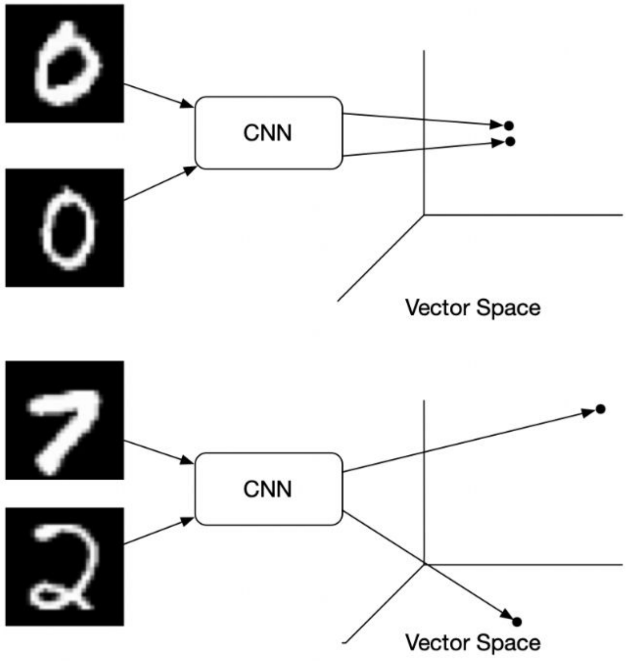
Abstract

Study the extent to which C2C-SN can be used to perform outlier detection and zero shot learning.

Dataset used: MNIST

Siamese Networks

- Consists of twin sub-networks which are joined at their outputs.
- Same weights are used for both the networks to make sure that similar images lie close to each other in the semantic space.
- Contrastive loss is used to measure the dissimilarity and similarity between the input pairs.



https://towardsdatascience.com/contrastive-loss-explaned-159f2d4a87ec

C2C Siamese Networks

- Special implementation of the Siamese network.
- Classification is based on learning patterns of both similarities and differences between classes.
- Assumes that the similarities and differences between the classes are consistent.
- Unlike traditional weighing, where pattern of features for the same class is extracted, C2C finds the pattern between a pair of classes.

C2C Siamese Networks

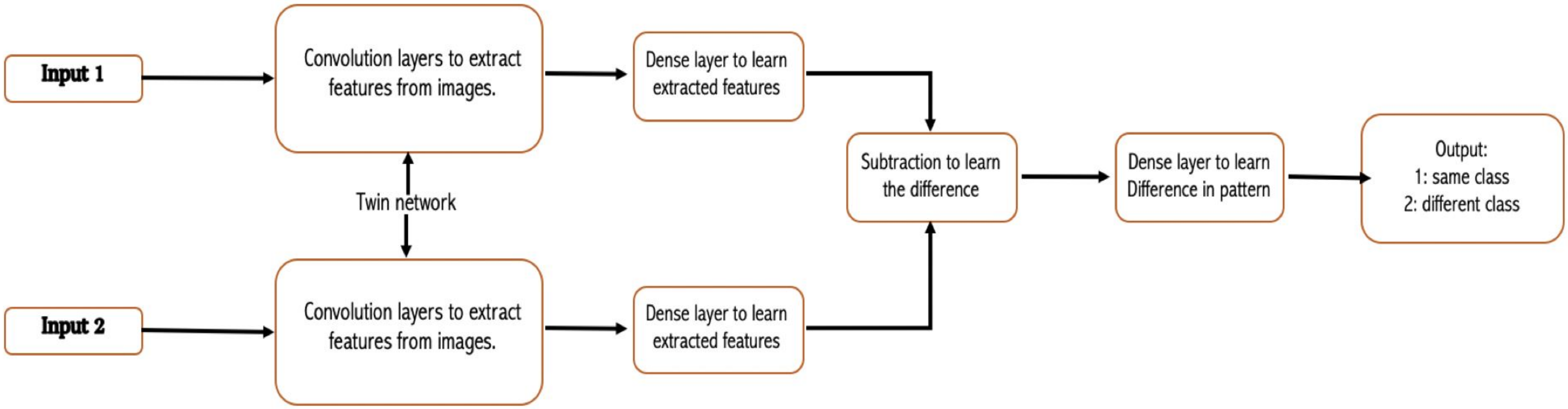


Figure1 : C2C Architecture

Why Zero Shot Learning?

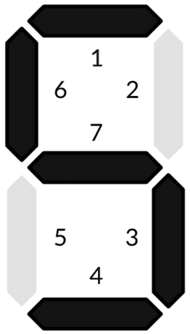
- Classical approach to classification : Train of a huge dataset with a fixed number of classes.
- Assumption : Training dataset contains all the classes the model will even come across.
- Problems:
    - Need for huge amount of labelled data.
    - Re-training of model when new classes are introduced.

Zero Shot Learning

- Used to classify unseen classes based on their description.
- Learn the relationships between the features of an image and its semantic embedding.
- Takes the image and its embedding to a semantic space, where features corresponding to the images are closely related. This is used to identify unseen classes based on their descriptions

Semantic Embedding

- Description of an image
- Common embeddings :Word2Vec, WordNet
- No available embeddings for MNIST, had to handcraft. Used the seven-segment display representation.
- Why semantic embeddings?
  - ZSL learns relationships between image features and its semantic embedding.
- Performed classical ZSL using these. Baseline accuracy: 51.5% when predicting among 4 classes.

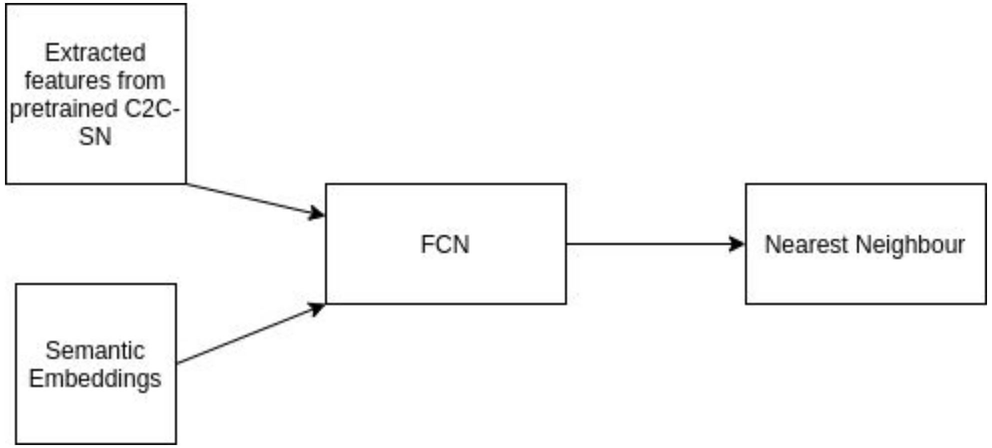


https://cvml.ist.ac.at/AwA2/

ZSL using C2C-SN

How do we extract image features from C2C-SN?

- Take output from one sub-network
- Average outputs from the two networks
- Sum outputs from the two networks.



Problem: If images belong two different classes.

ZSL using C2C-SN

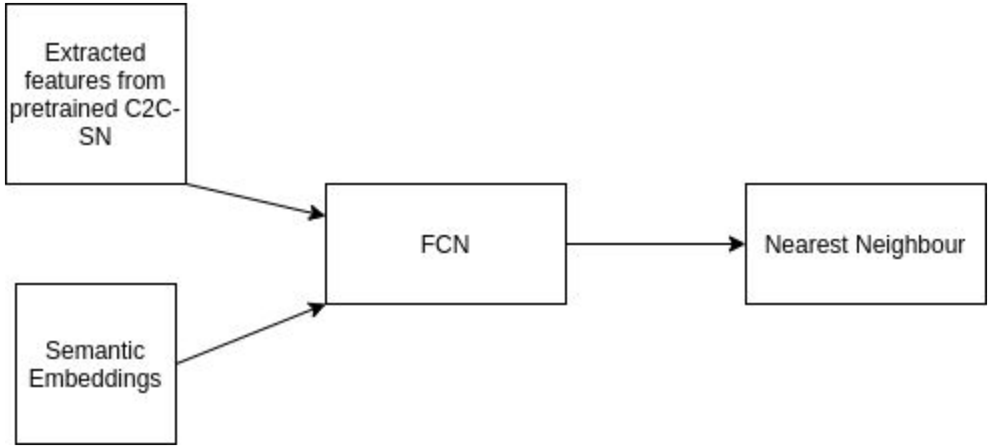
Assumption: Know which test images belong to the same class, but don't know the label of the class.

Can be achieved in practise by performing one-shot learning using test-images.

Training Labels : 0-5

Testing Labels : 6-9

High training accuracy, but accuracy of 24.45% on new classes. Overfitting?



Outlier Detection using C2C-SN

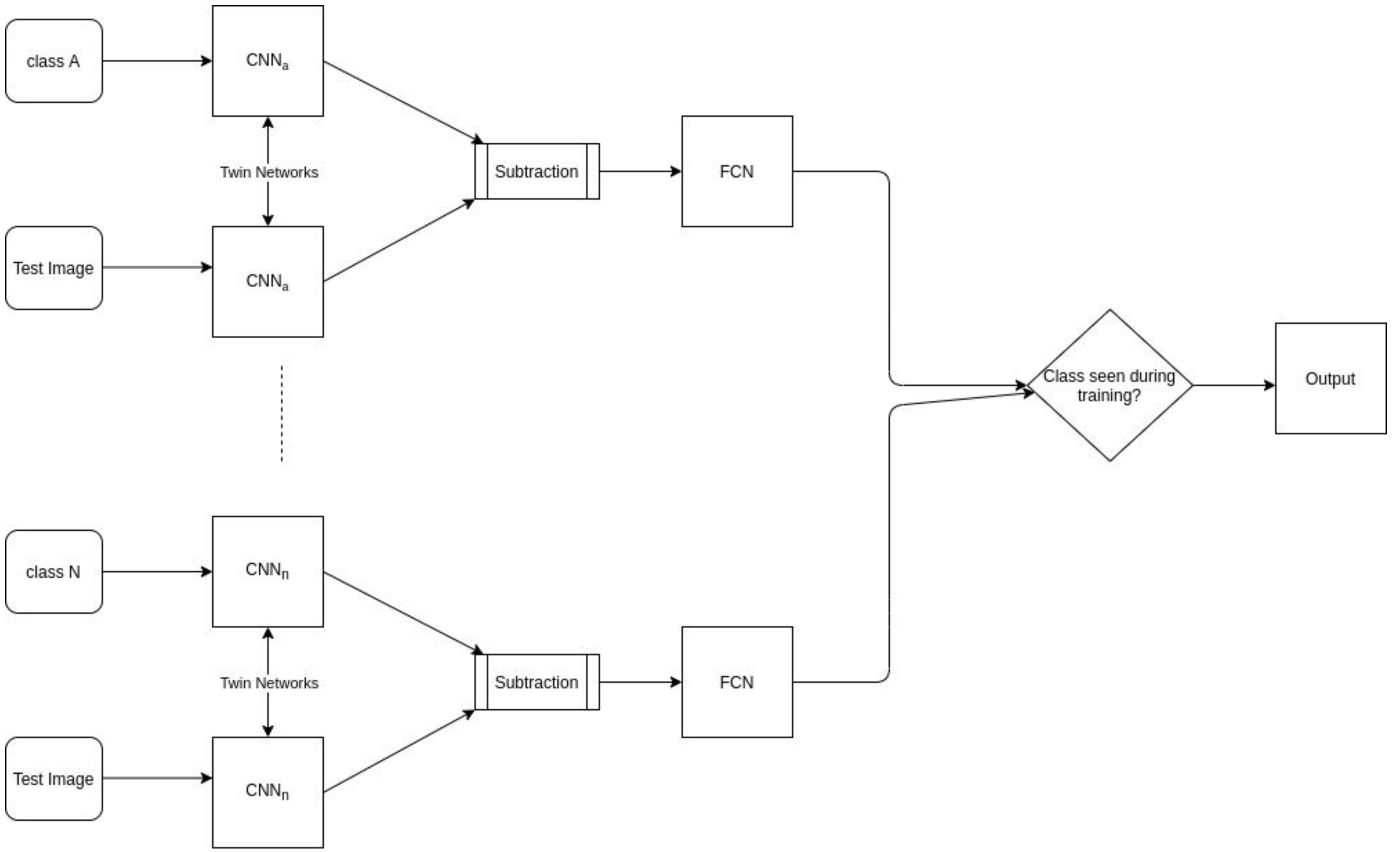
Training Labels : 0-5

Testing Labels : 6-9

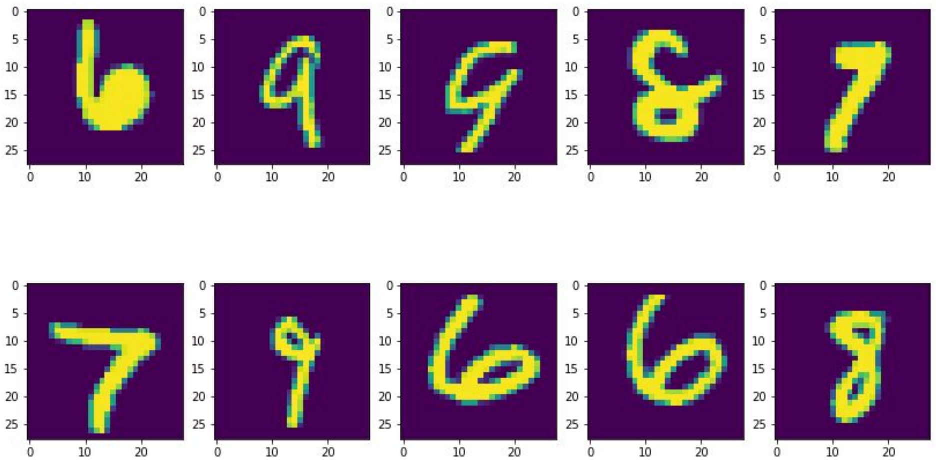
If none of the models give a high score, then outlier.

Accuracy : 71.48%

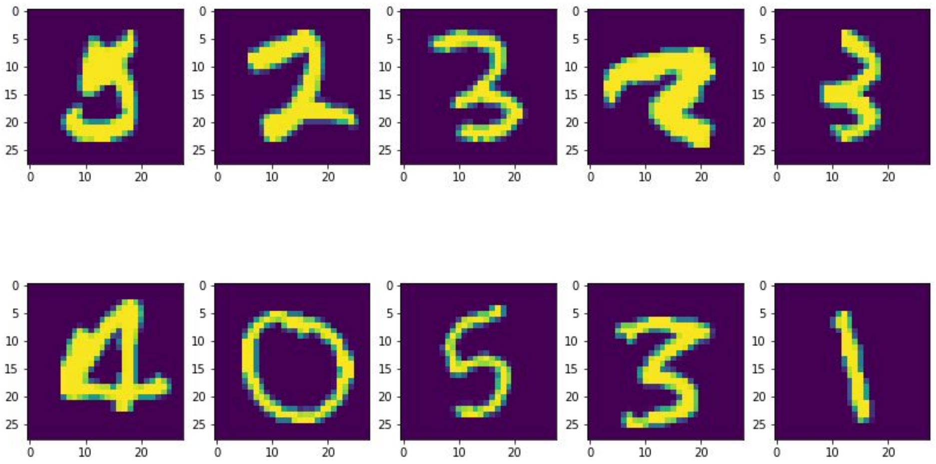
High precision, but low recall



Outlier Detection using C2C-SN



False inliers



False outliers

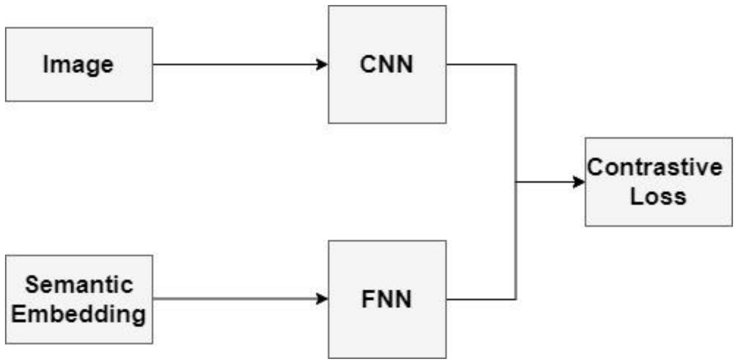
Joint training of images and semantic embeddings

Inspired by siamese network architecture.

In Siamese, both CNNs capture image representations and the loss tries to bring similar representations closer.

What if we replace one image with semantic embeddings and jointly train both networks to try and reduce the loss?

Result: Unstable network. Possibly overfitting to the weak semantic embeddings



Conclusions

- C2C-SN can be used for outlier detection. Need to tweak the confidence threshold depending upon the domain.
- It fails to perform ZSL on MNIST dataset using our handcrafted attributes. Can not conclude about the ability of the architecture to perform ZSL.
- Joint training of images and semantic embeddings likely overfitting to weak semantic embeddings.

Future Scope

- Try out Zero Shot Learning using C2C-SN using a dataset with richer attributes.
- Joint training of semantic embeddings and images on a dataset with richer attributes and for a longer duration of time. Unshared weights, hence might take time to stabilize.