One-shot Learning using Triplet Networks with kNN Classifier

Mu Zhou[1,2], Yusuke Tanimura[2,1], Hidemoto Nakada[2,1]

[1]University of Tsukuba
[2]Artificial Intelligence Research Center - National Institute of Advanced Industrial Science and Technology

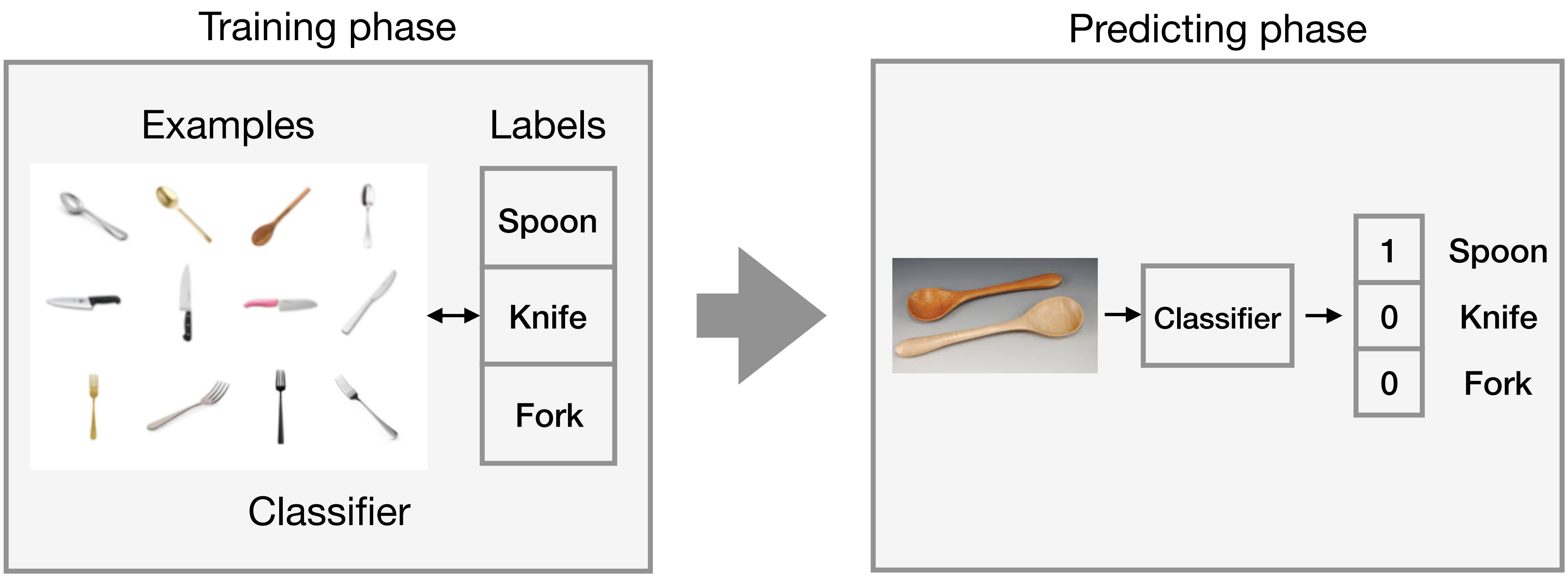
Content

- **Introduction**
- Related work
- Method
- Experiments
- Results

Introduction

Supervised Learning

- Learn a correspondence between training data and labels.
  - Require a **large labeled dataset** for training.



- Hard to let classifiers learn new concepts from little data.

Introduction

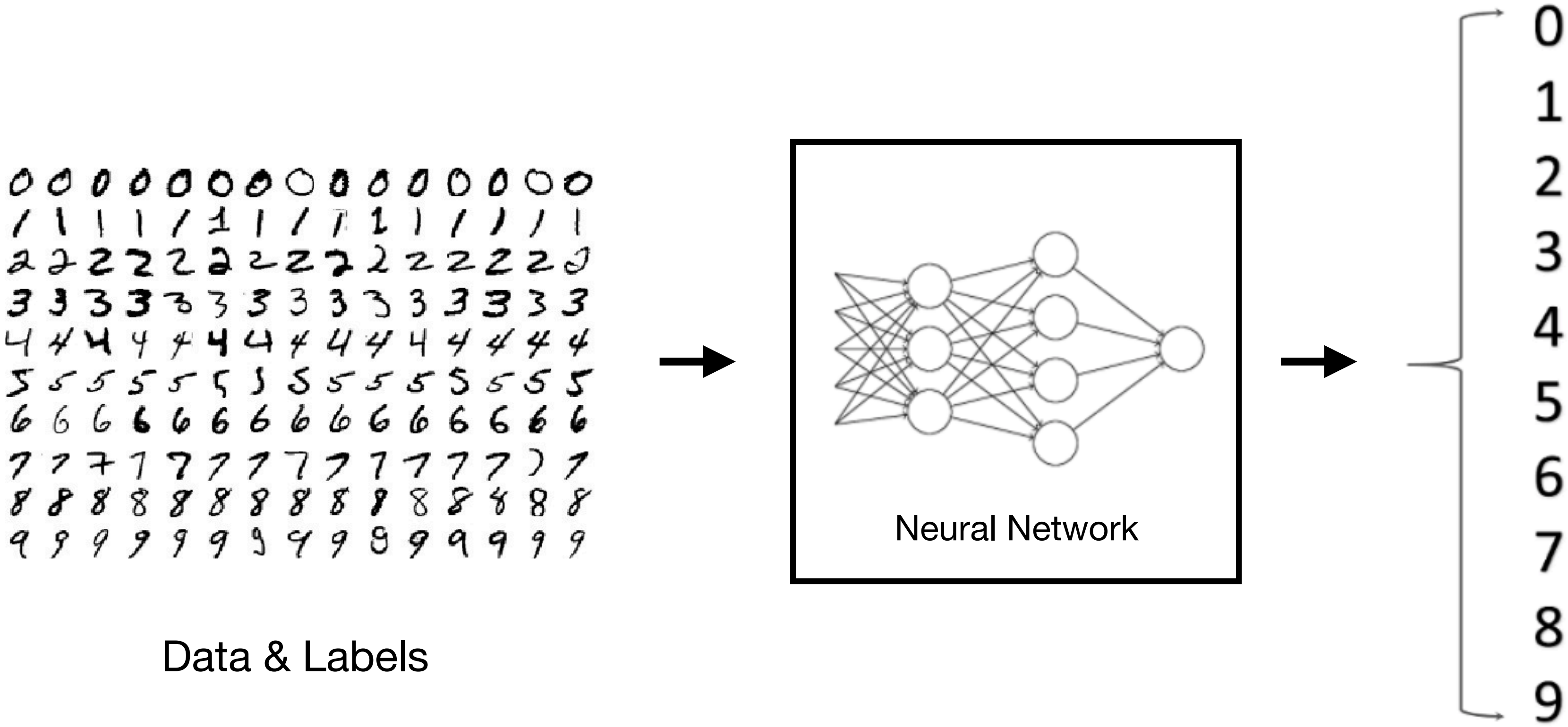
One-shot Learning

- Learn a concept from **one or only a few** training example, contrary to the normal practice of using a large amount of data.

Introduction

Image classification

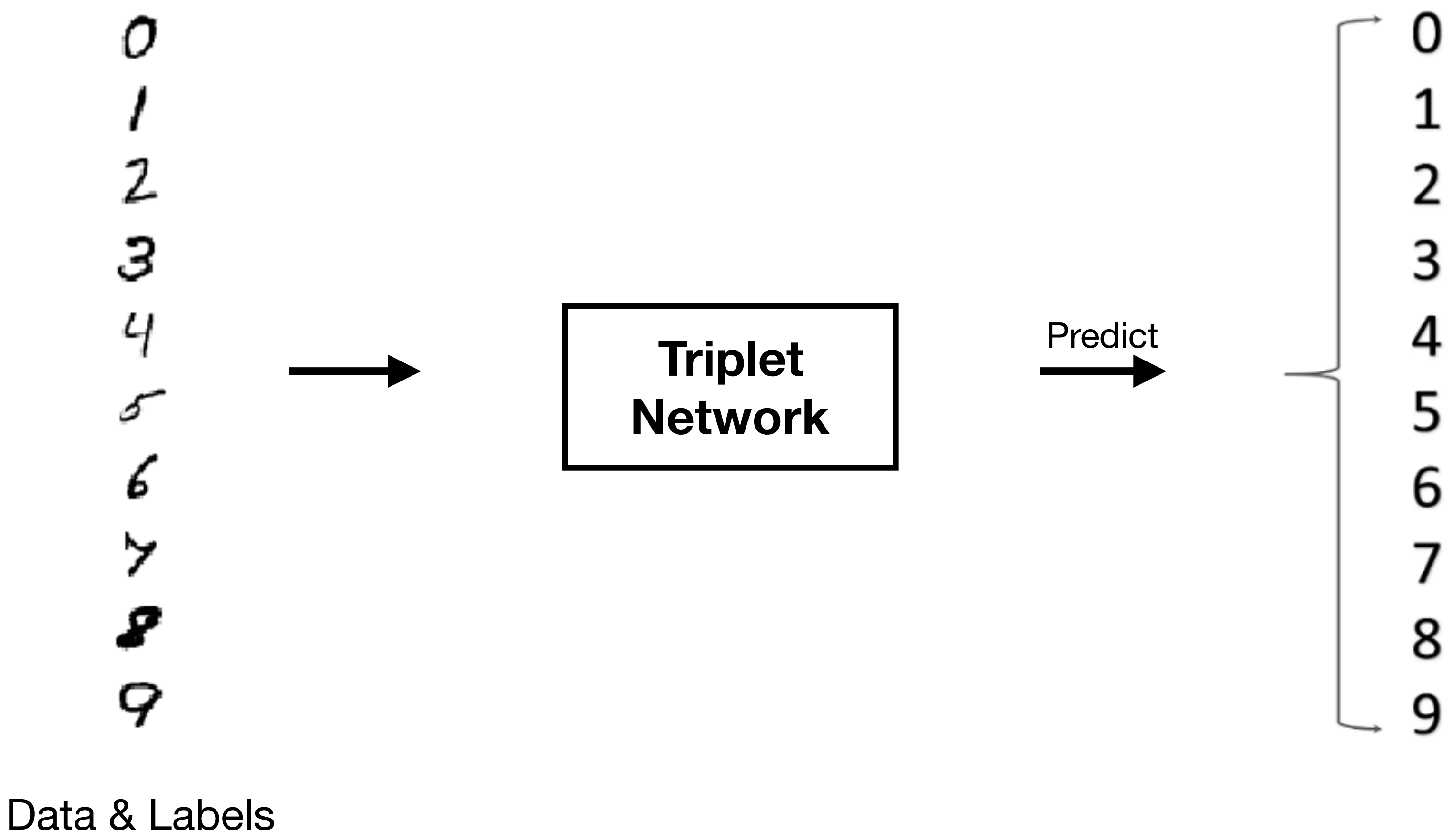
- General approach
  - Purpose: classify the image with **large amount of labeled training data.**



Introduction

Our Method

- Goal: categorize the image with **only one labeled data per class**.
- Contribution: Triplet Network + Data Augmentation.



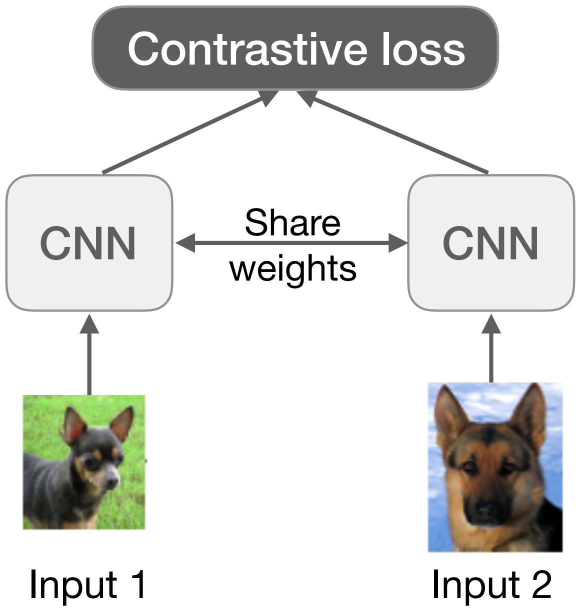
Content

- Introduction
- **Related work**
- Method
- Experiments
- Results

Related work

Convolutional Siamese Network [Koch+, 2015]

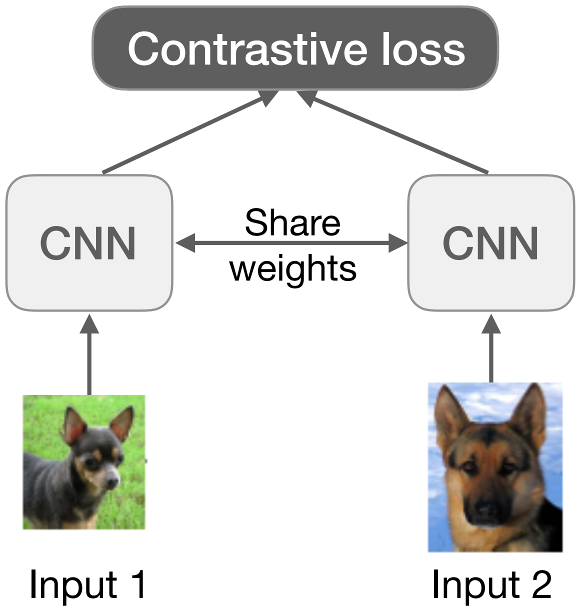
- **Architecture**
  - Learn **similarity** between two inputs.
  - 2 identical neural networks (same weights).
  - Optimized by a **contrastive loss**.



Related work

Convolutional Siamese Network [Koch+, 2015]

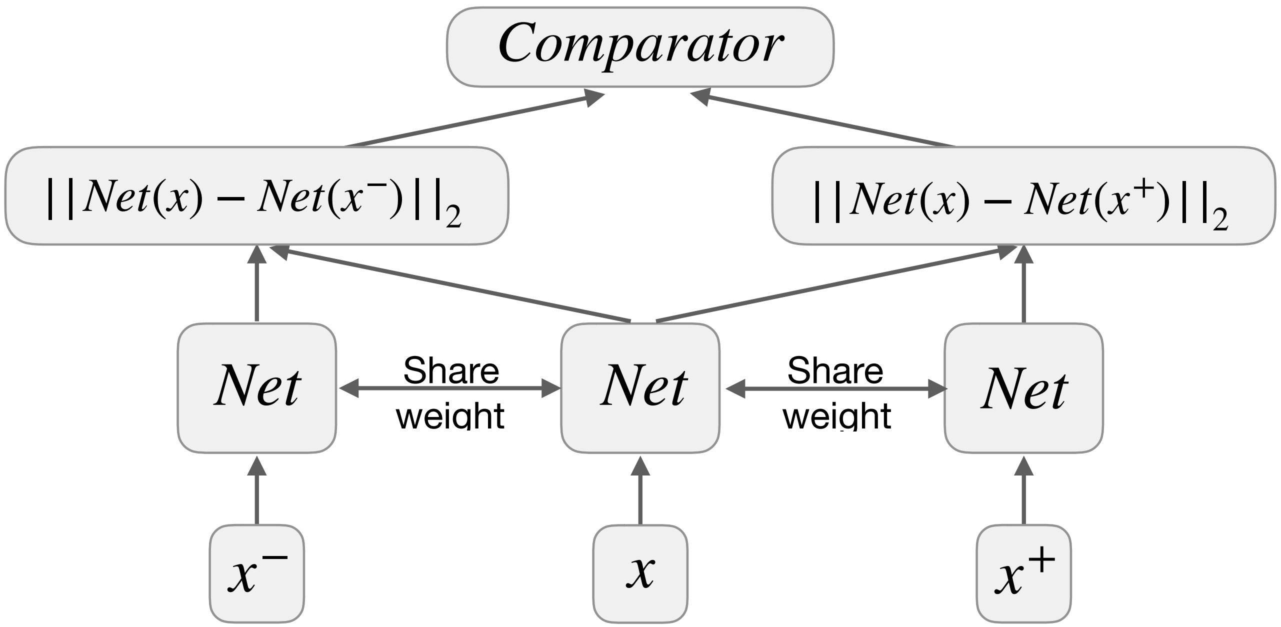
- **Contrastive loss function**
  - Evaluate how well the network is distinguishing a given pair of images.
  - Keep the samples belonging the same class close to each other and separate the dissimilar samples.



Related work

Triplet Network [Hoffer+, 2015]

- Aim to learn useful representations by distance comparisons.
- Comprised of 3 instances of the same feedforward network (with shared parameters).
- Output 2 intermediate values - the L2 distance.



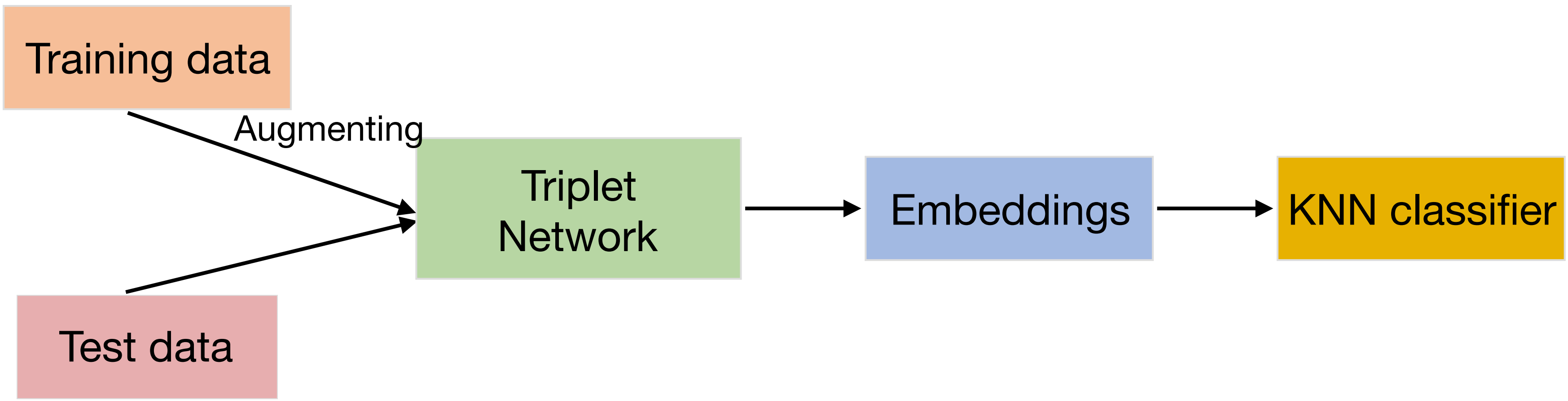
Content

- Introduction
- Related work
- **Method**
- Experiments
- Results

Method

Procedure

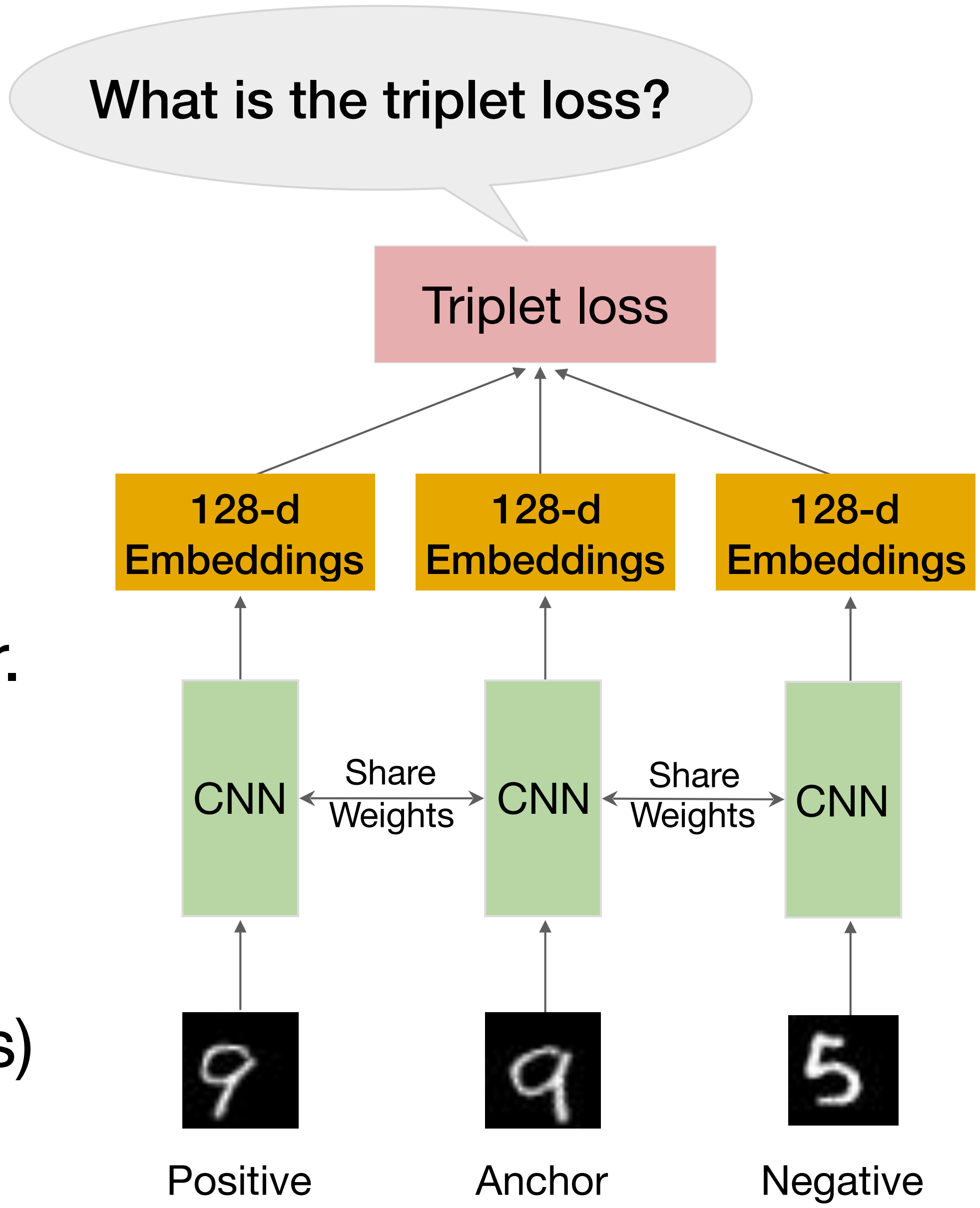
- Triplet Network
- Triplet Loss function
  - FaceNet (Schroff et al., 2015)
- kNN classifier
  - Make the prediction by using the feature vectors of training and test points.



Method

Triplet Network

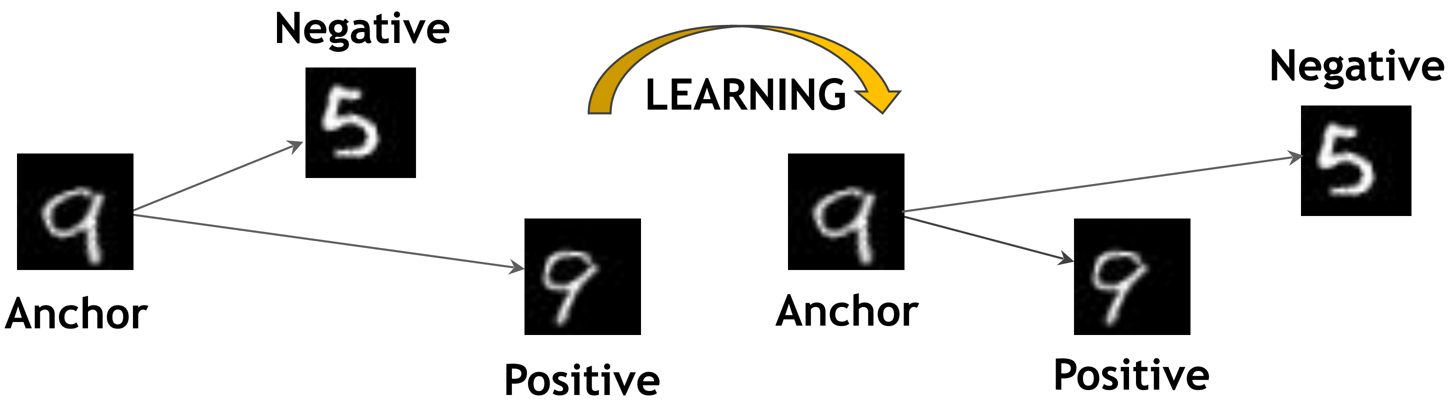
- 3 inputs:
  - **Anchor**: sample from dataset.
  - **Positive**: a sample from same class as the Anchor.
  - **Negative**: a sample from different class than the Anchor.
- CNN model (shared weights)
  - Input shape: (28, 28, 1)
  - Output shape: 128
- Triplet Loss function



Method

Triplet Loss

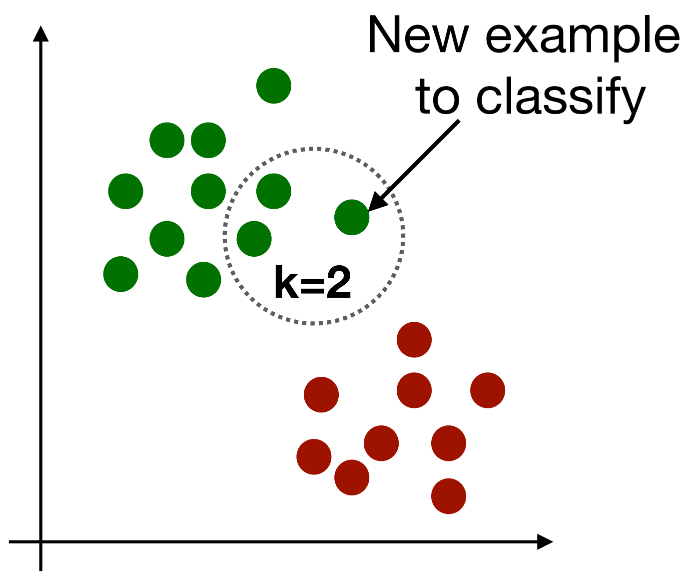
- $L = max(d(x_a, x_p) - d(x_a, x_n) + \alpha, 0)$
- where α is a margin that is enforced between positive and negative pairs.
- Minimizes the distance between the ***anchor*** and the ***positive***.
- Maximizes the distance between the ***anchor*** and the ***negative.***



Method

k-Nearest Neighbors

- One of the simplest way to perform classification.
- Most kNN classifiers use Euclidean distances (also known as L2-norm distance) to measure the similarities between the instances which are represented as vector inputs.



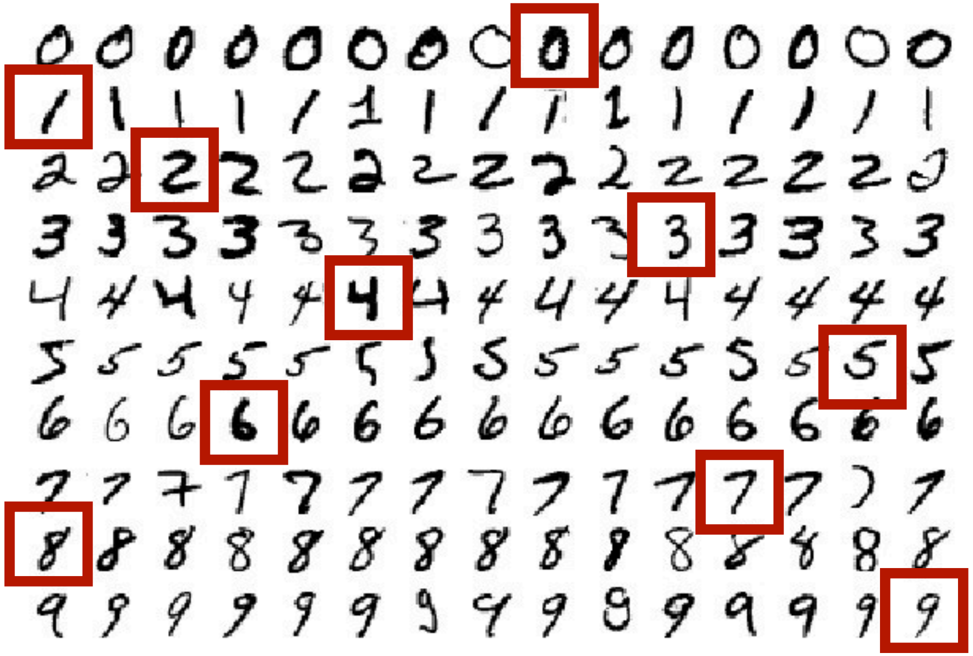
Content

- Introduction
- Related work
- Method
- **Experiments**
- Results

Experiments

How do we choose the data?

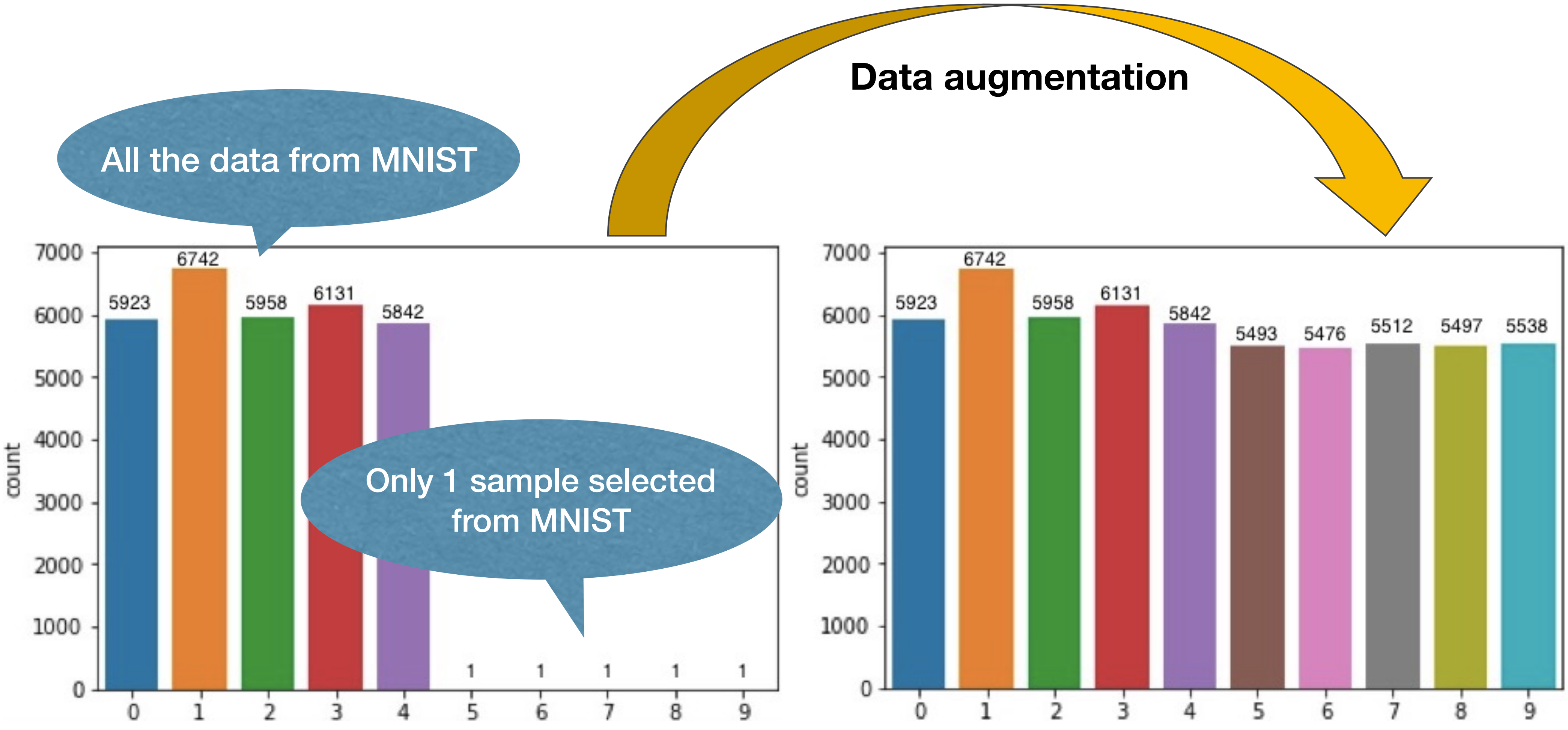
- **MNIST dataset**
  - A large database of handwritten digits widely used in the field of machine learning.
  - We randomly select 1 sample from each class in the MNIST dataset.



Experiments

How do we generate the dataset?

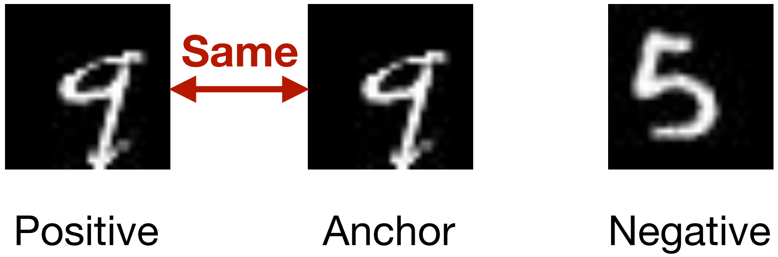
- Our initial dataset.
- Our augmented dataset.



Experiments

How do we select triplets?

- Our initial dataset.



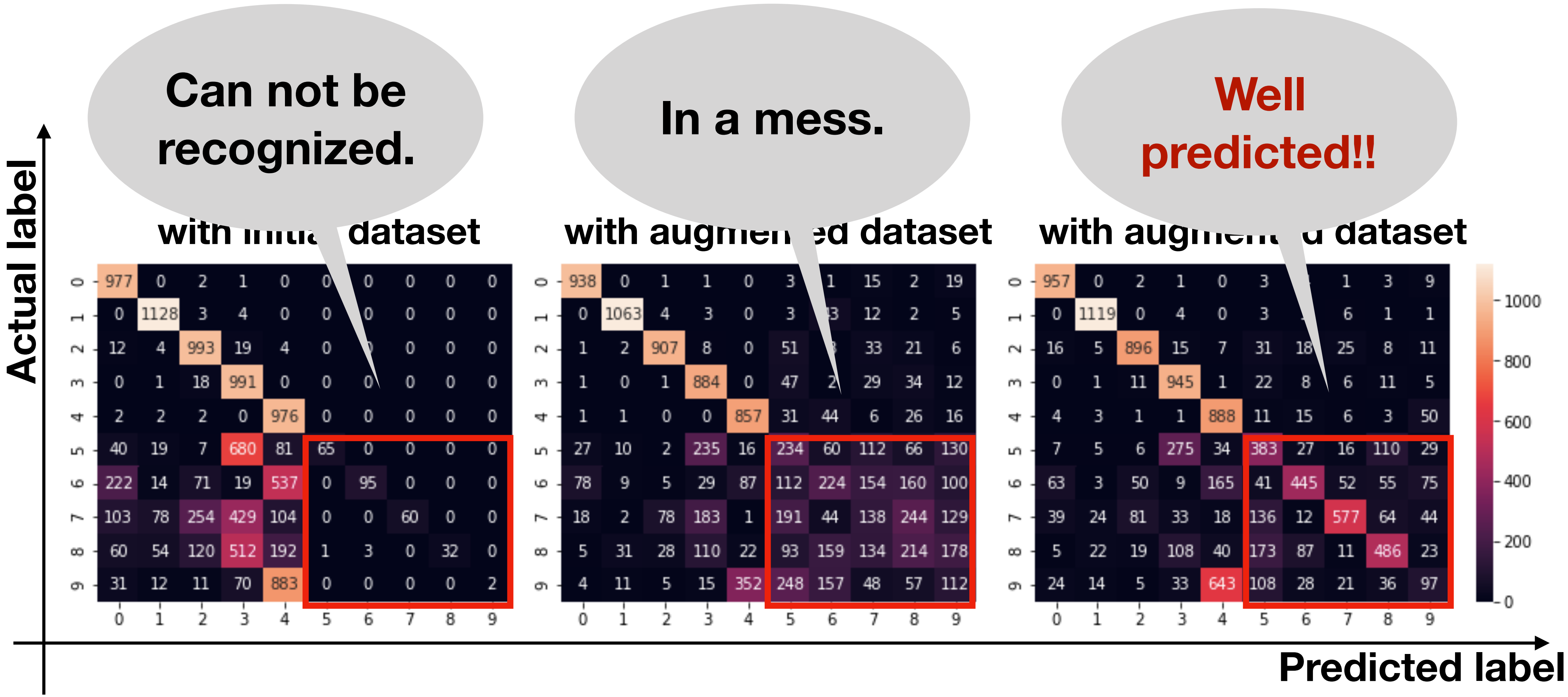
- Our augmented dataset.

Content

- Introduction
- Related work
- Method
- Experiments
- **Results**

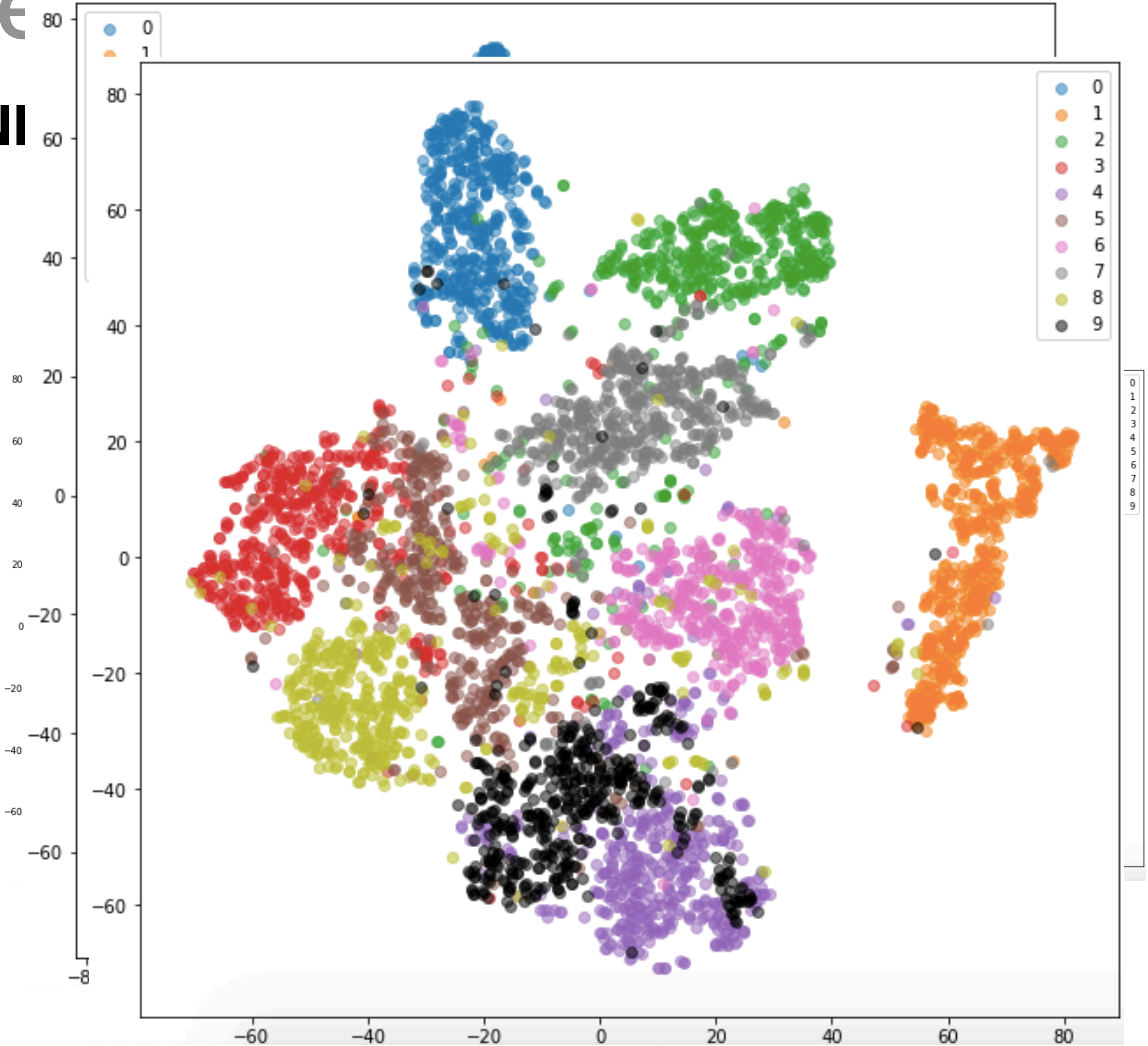
Results

Method (dataset)	Accuracy					
	5	6	7	8	9	Average
TripletNN (not Agumented)	14%	18%	11%	6%	0%	9.8%
CNN (Augmented)	25%	26%	16%	24%	13%	20.8%
TripletNN (Augmented)	**42%**	**56%**	**66%**	**56%**	**14%**	**46.8%**



Re

t-SNI



Conclusion

- Triplet neural network make sense in this experiment, the accuracy shows better than CNN model in one-shot learning problem.

- This study indicates that the benefits gained from data augmentation also work well on one-shot learning problem.

Future work

- Work on other much larger and complex datasets for one-shot classification, (e.g. Fashion MNIST, Omniglot, Mini-ImageNet), to validate whether our method is resultful.

- Find more effective approaches and investigate other techniques based on metric learning or meta learning, or combine other method such as Adversarial Generalized Model.

Thank you!

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was supported by JSPS KAKENHI Grant Number JP16K00116.