Co-domain Embedding using Deep Quadruplet Networks for Unseen Traffic Sign Recognition

Junsik Kim, Seokju Lee, Tae-Hyun Oh†, In So Kweon
Dept. of Electrical Engineering, KAIST, Daejeon, Korea
†MIT CSAIL, Cambridge, US

{mibastro, seokju91}@gmail.com, taehyun@csail.mit.edu†, iskweon@kaist.ac.kr

Abstract
Recent advances in visual recognition show overarching success by virtue of large amounts of supervised data. However, the acquisition of a large supervised dataset is often challenging. This is also true for intelligent transportation applications, i.e., traffic sign recognition. For example, a model trained with data of one country may not be easily generalized to another country without much data. We propose a novel feature embedding scheme for unseen class classification when the representative class template is given. Traffic signs, unlike other objects, have official images. We perform co-domain embedding using a quadruple relationship from real and synthetic domains. Our quadruplet network fully utilizes the explicit pairwise similarity relationships among samples from different domains. We validate our method on three datasets with two experiments involving one-shot classification and feature generalization. The results show that the proposed method outperforms competing approaches on both seen and unseen classes.

Introduction
Recent advances in the field of computer vision have provided highly cost-effective solutions for developing advanced driver assistance systems (ADAS) for automobiles. Furthermore, computer vision components are becoming indispensable to improve safety and to achieve AI in the form of fully automated, self-driving cars. This is mostly by virtue of the success of deep learning, which is regarded to be due to the presence of large-scale supervised data, proper computation power and algorithmic advances (Goodfellow, Bengio, and Courville 2016).

Among all ADAS related problems, in this paper, we tackle unseen traffic sign recognition. A distinctive difference related to this problem as regards traditional recognition problems is that synthetic traffic-sign templates are exploited as representative anchors, whereby classification can be done for an actual query image by finding the minimum distance to the templates of each class (i.e., few-shot learning with domain difference).

In reality, traffic signs differ depending on the country, but one may obtain synthetic templates from traffic-related public agencies. Nonetheless, the diversity of templates for a single class is limited; hence, we focus on scenarios of challenging one-shot classification (Koch, Zemel, and Salakhutdinov 2015; Lake, Salakhutdinov, and Tenenbaum 2015; Miller, Matsakis, and Viola 2000) with domain adaptation, where a machine learning model must generalize to new classes not seen in the training phase given only a few examples of each of these classes but from different domains.

In practice, this type of model is especially useful for ADAS in that: 1) one can avoid high cost re-training from scratch, 2) one can avoid annotating large-scale supervised data, and 3) it is readily possible to adapt the model to other environments.

Given the success of deep learning, a naive approach for the few-shot problem would be to re-train a deep learning model on a new scarce dataset. However, in this limited data regime, this type of naive method will not work well, likely due to severe over-fitting (Lake, Salakhutdinov, and Tenenbaum 2015). While people have an inherent ability to generalize from only a single example with a high degree of accuracy, the problem is quite difficult for machine learning models (Lake, Salakhutdinov, and Tenenbaum 2015).

Thus, our approach is based on the following hypotheses: 1) the existence of a co-embedding space for synthetic and real data, and 2) the existence of an embedding space where real data is condensed around a synthetic anchor for each class. We illustrate the idea in Fig. 1. Taking these into account, we learn two non-linear mappings using a neural network. The first involves mapping for a real sample into an embedding space, and the second involves mapping of a synthetic anchor onto the same metric space. We leverage the quadruplet relationship to learn non-linear mappings, which can provide rich information to learn generalized and discriminative embeddings. Classification is then performed for an embedded query point by simply finding the nearest class anchor. Despite its simplicity, our method outperforms with a margin in the unseen data regime.

Related work
Our problem can be summarized as a modified one-shot learning problem that involves heterogeneous domain datasets. This type of problem has gained little attention. Furthermore, to the best of our knowledge, our work is the first to tackle unseen traffic sign recognition with heterogeneous domain data; we therefore summarize the work most relevant to our
proposed method in this section.

Non-parametric models such as nearest neighbors are useful in few-shot classification (Vinyals et al. 2016; Santoro et al. 2016), in that they can be naturally adapted to new data and do not require training of the model. However, the performance depends on the chosen metric (Atkeson, Moore, and Schaal 1997)\footnote{To overcome this, Goldberger et al. (Goldberger et al. 2004) propose the neighborhood components analysis (NCA) and that learns the Mahalanobis distance to maximize the accuracy of the $K$-nearest-neighbor ($K$-NN). Weinberger et al. (Weinberger and Saul 2009), who studied large margin nearest neighbor (LMNN) classification, also maximize the $K$-NN accuracy with a hinge loss that encourages the local neighborhood of a point to contain other points with identical labels with some margin, and vice versa. Our work also adopts the hinge loss with a margin in the same spirit of Weinberger et al. Because NCA and LMNN are limited on linear model, each is extended to a non-linear model using a deep neural network in (Salakhutdinov and Hinton 2007) and (Min et al. 2009) respectively.}

Our work may be regarded as a non-linear and quadruple extension of Mensink et al. (Mensink et al. 2013) and Perrot et al. (Perrot and Habrard 2015) to one-shot learning, in that they leverage representative auxiliary points for each class instead of individual examples of each class. These approaches are developed to adapt to new classes rapidly without re-training; however, they are designed to handle cases where novel classes come with a large number of samples. In contrast, in our scenario, a representative example is given explicitly as a template image of a traffic sign. Given such a template, we learn non-linear embedding in an end-to-end manner without pre-processing necessary in both Mensink et al. and Perrot et al. to obtain the representatives.

All of these NN classification schemes learn the metric via a pairwise relationship. In the recent metric learning literature, there have been attempts (Wang et al. 2014; Hoffer and Ailon 2015; Law, Thome, and Cord 2017; Chen et al. 2017; Huang et al. 2016) to go beyond learning metrics using only a pairwise relationship (i.e., 2-tuple, e.g., Siamese (Bromley et al. 1993; Chopra, Hadsell, and LeCun 2005; Hadsell, Chopra, and LeCun 2006): triplet (Weinberger and Saul 2009; Wang et al. 2014; Hoffer and Ailon 2015), quadruplet (Law, Thome, and Cord 2017) and quintuplet (Huang et al. 2016). The use of tuples of more than a triple relationship may be inspired from the argument of Kendall and Gibbons (Kendall and Gibbons 1990), who argued that humans are better at providing relative (i.e., at least triplet-wise) comparisons than absolute comparisons (i.e., pairwise). While our method also exploits a quadruple relationship, the motivation behind this composition is rather specific for our problem definition, in which two samples from template sets and two samples from real sets have clear combinatorial pairwise relationships. More details will be described later.

Other one-shot learning approaches take wholly different notions. Koch et al. (Koch, Zemel, and Salakhutdinov 2015) use Siamese network to classify whether two images belong to the same class. To address one-shot learning for character recognition, Late et al. (Lake, Salakhutdinov, and Tenenbaum 2015) devise a hierarchical Bayesian generative model with knowledge of how a hand written character is created. A recent surge of models, such as a neural Turing machine (Graves, Wayne, and Danihelka 2014), stimulate the meta-learning paradigm (Santoro et al. 2016; Vinyals et al. 2016; Ravi and Larochelle 2017) for few-shot learning. Comparing to these works that have limited memory capacities, the NN classifier has an unlimited memory and can, automatically store and retrieve all previously seen examples. Furthermore, in the few-shot scenario, the amount of data is very small to the extent that a simple inductive bias appears to work well without the need to learn complex input-sensitive embedding (Vinyals et al. 2016; Santoro et al. 2016; Ravi and Larochelle 2017), as we do so. This provides the $k$-NN with a distinct advantage.

Moreover, because we have two data sources, synthetic templates and real examples, a domain difference is naturally introduced in our problem. There is a large amount of research that smartly solves domain adaptation (refer to the survey by Csurka et al. (Csurka 2017) for a thorough review), but we deal with this by simply decoupling the network parameters from each other, the template and the real domains. This method is simple, but in the end it generalizes well owing to the richer back-propagation gradients from the quadruple combinations.
Quadruple network for jointly adapting domain and learning embedding

Our goal is to learn embeddings, such that different domain examples are embedded into a common metric space and where their embedded features favor to be generalized as well as discriminative.

To this end, we leverage a quadruplet relationship, consisting of two anchors of different classes and two others for examples corresponding to the anchors. We first describe the quadruple (4-tuple) construction in the following section. Subsequently, we define the embeddings followed by the objective functions and quadruplet network.

Notation We consider two imagery datasets: the template set \( T = \{(T, y)\} \), where each \( T \) denotes a representative template image and \( y \in \{1, \ldots, C\} \) is the corresponding label (out of the \( C \) class), and the real example set \( X = \{X_k\}_{k=1}^C \), where \( X_k \) is the set of real images \( \{X\} \) of class-\( k \). For simplicity, we use \( T_k \) (\( X_k \)) to denote a sample labeled with class-\( k \).

We define Euclidean embeddings as \( f(\cdot) \), where \( f(x) \) maps a high-dimensional vector \( x \) into a \( D \)-dimensional feature space, i.e., \( e = f(x) \in \mathbb{R}^D \).

Quadruple (4-tuple) construction

Our idea is to embed template and real domain data into the same feature space such that a template sample acts as an anchor point on the feature space and real samples relevant to the anchor form a cluster around it, as illustrated in Fig. 1. Specifically, we aim to achieve two properties for an embedded feature space: 1) distinctiveness between anchors is favored, and 2) real samples must be mapped close to the anchor that corresponds to the same class.

To leverage the relational information, we define a quadruple, a basic element, by packing two template images from different classes and two real images corresponding to respective template classes, i.e., for simplicity, two classes \( A \) and \( B \) are considered, then \((T_A, T_B, X_A, X_B)\). From pairwise combinations within the quadruple, we can reveal several types of relational information as follows:

1. \( T_A \) should be far from \( T_B \) in an embedding space,
2. \( X_A \) should be far from \( X_B \) in an embedding space,
3. \( X_A \) (or \( X_B \)) should be close to \( T_A \) (or \( T_B \)) in an embedding space,
4. \( T_A \) (or \( T_B \)) should be far from \( X_B \) (or \( X_A \)) in an embedding space,

By which we derive the final objective function. These relations depicted in Fig. 2.3

Quadruple sampling We sampled two templates \((T_A, y)\) and \((T_B, y')\) from template set \( T \) while guaranteeing two different classes, followed by the two real images of \( X_A \in X_y \) and \( X_B \in X_{y'} \).

Comparison to other tuple approaches In metric learning, the most common approaches would be the Siamese (Bromley et al. 1993), Siamese (Bromley et al. 1993; Chopra, Hadsell, and LeCun 2005) [Hadsell, Chopra, and LeCun 2006] and triplet (Weinberger and Saul 2009) [Wang et al. 2014] Hoffer and Ailon 2015, which typically use 2- and 3-tuples, respectively. From the given tuple, they optimize with the pairwise differences. This concept can be viewed as follows: given a tuple,ilter tuples, the Siamese has only a single source of loss (and its gradient), while triplets utilize two sources, i.e., (query, positive) and (query, negative). With this type of simple comparison, we can intuitively conjecture higher stability or performance of triplet network over Siamese network.

Law et al. (Law, Thome, and Cord 2017) deal with a particular ambiguous quadruple relationship \((a) \prec (b) \succeq (c) \prec (d)\) by forcing the difference between \((b)\) and \((c)\) to be smaller than the difference between \((a)\) and \((d)\). Huang et al. (Huang et al. 2016) (quintuplet, 5-tuple) is devised a means to handle class imbalance issues by leveraging the relationships among three levels (e.g., strong, weaker and weakest in terms of a cluster analysis) of positives and negatives. Comparing the motivations of these approaches, for instance indefinite relationiveness, our quadruple comes from a specific relationship, i.e., heterogeneous domain data. Moreover, it is important to note that our quadruple provides richer information (a total of 6 pairwise information) compared to the method of Law et al. (quadruplet) which leverages a single constraint from a quadruplet. It is even richer than Huang et al. (quintuplet), who provides 3 constraints.

Quadruplet Network

Given the defined quadruple, we propose a quadruplet metric learning that learns to embed template images and real images into a common metric space, say \( \mathbb{R}^D \), through an embedding function. In order to deal with non-linear mapping, the embedding \( f \) is modeled as a neural network, of which set of weight parameters are denoted as \( \theta \). Since we deal with data from two different domains, template and real images, we simply use two different neural networks \( T \) and \( R \) for the template and real images respectively, expressed as \( f_T(\cdot) = f(\cdot; \theta_T) \) and \( f_R(\cdot) = f(\cdot; \theta_R) \), such that we can adapt both domains. Now, we are ready to define the proposed quadruplet network.

The proposed quadruplet network \( Q \) is composed of two Siamese networks, the weights of which are shared within each pair. One part embeds features from template images and the other part for real images. Quadruplet images from each domain are fed into the corresponding Siamese networks (depicted in Fig. 2.5), and denoted as

\[
Q \left( (T_A, T_B, X_A, X_B); \theta_T, \theta_R \right) = \left[ f_T(T_A), f_T(T_B), f_R(X_A), f_R(X_B) \right]
\]

(1)

for two different arbitrary classes \( A \) and \( B \), where \( e \in \mathbb{R}^d \) represents the embedded vector mapped by the embedding function \( f \).

Loss function We mainly utilize the \( l_2 \) hinge embedding loss with a margin to train the proposed network. Given outputs quadruplet features \([e_T^A, e_T^B, e_R^A, e_R^B]\), we have up to six pairwise relationships, as shown in Fig. 2.6, and obtain six pairwise Euclidean feature distances as
can exploit up to six losses; hence, we have several choices to formulate the final loss function, as described below:

\[
\{d(e^A_k, e^B_{k'})\}_{j,j'\in \{T,R\}}, \text{ where } d(a, b) = \|a-b\|_2 \text{ denotes the Euclidean distance. Let } h_m(d) \text{ denote the hinge loss with margin } m \text{ as } h_m(d) = \max(0, m-d), \text{ where } d \text{ is the distance value. We then encourage the embedded vectors with the same label pairs (i.e., if } k = k') \text{ to be close by applying the loss } h_{-m}(-d), \text{ while pushing away the different label pairs by applying } h_m'(d).
\]

To induce the embedded feature space with the aforementioned two properties in the Quadruplet Construction section, the minimum number of necessary losses is three, while we can exploit up to six losses; hence, we have several choices to formulate the final loss function, as described below:

\[
\text{HingeM-3 : } L3_{m,m'} = h_m (d (e^A_T, e^B_T)) + h_{m'} (d (e^A_R, e^B_R)), \quad (2)
\]

\[
\text{HingeM-5 : } L5_{m,m'} = L3_{m,m'} + h_m (d (e^A_T, e^B_R)) + h_{m'} (d (e^A_R, e^B_T)), \quad (3)
\]

\[
\text{HingeM-6 : } L6_{m,m'} = L5_{m,m'} + h_m (d (e^A_X, e^B_Y)) + h_{m'} (d (e^A_Y, e^B_X)). \quad (4)
\]

In addition, inspired by the contrastive loss (Chopra, Hadsell, and LeCun 2005), we also adopt an alternative loss by replacing the \( h_m(-d) \) terms in Eq. (2) with directly minimizing \( d(e^A_k, e^B_{k'}) \). We denote this alternative loss simply as contrastive with a slight abuse of the terminology. We will compare these losses in the Experiments section. Analogous to traditional deep metric learning, training with these losses can be done with a simple SGD based method. By using shared parameter networks, the back-propagation algorithm updates the models w.r.t. several relationships; e.g., in the HingeM-6 case, the template tower is updated w.r.t. 5 pairwise relationships (i.e., \((T^A, T^B), (T^A, X^A), (T^B, X^B), (X^A, T^B)\)), analogously for the real tower.

**Experiments**

**Experiment setup**

**Competing methods** We compare the proposed method with other deep neural network based previous works and the additionally devised baseline, as follows:

- **IdsiaNet** (Ciresan et al. 2012) is a competition winner of the German Traffic-Sign Recognition Benchmark (Stallkamp et al. 2012) (GTSRB). We directly used an improved implementation ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) for all of the experiments, IdsiaNet is exhaustively compared as a supervised model baseline, in the same philosophy of the deep generic feature (Donahue et al. 2014). Moreover, templates are not used for training. For a fair comparison, the architecture itself is adopted as the base network for the following models.

- **Triplet** (Hoffer and Alon 2015) (Hoffer et al.) is similar to our model but with triplet data. Three weight shared networks are used, while in training, we randomly sampled triplets within the real image set with labels, such that no template image is exposed during triplet training. This training method is consistent with Hoffer et al.

- **Triplet-DA** (domain adaptation) is a variant devised by us to test our hypothesis that involving different domain templates as an anchor is beneficial. Three weight shared networks are used, and for triplet sampling, we sample one template \((T, y)\) from template set \(T\) and then sample positive and negative real images from \(X_y\) and \(X_{\neq y}\), respectively.

**Implementation details** For fairness, all of the details are equally applied unless otherwise specified. All input images are resized to \(48 \times 48\) and the mean intensity of training set is subtracted. We did not perform any further preprocessing, data augmentation, or ensemble approach.

We use the same improved IdsiaNet ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) for the base network of **Triplet**, **Triplet-DA** and our **Quadruplet**, but replace the output dimension of the final layer \(\mathbb{R}^D\) to be \(\mathbb{R}^D\) without a softmax layer. Every model is trained from scratch. Most of the hyper parameters are based on the implementation ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) with a slight modification (i.e., fixed learning rates: \(10^{-3}\), momentum: 0.9, weight decay: \(10^{-4}\), mini-batch size: 100).

\[\text{Implementation details} \quad \text{For fairness, all of the details are equally applied unless otherwise specified. All input images are resized to } 48 \times 48 \text{ and the mean intensity of training set is subtracted. We did not perform any further preprocessing, data augmentation, or ensemble approach. We use the same improved IdsiaNet ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) for the base network of Triplet, Triplet-DA and our Quadruplet, but replace the output dimension of the final layer } \mathbb{R}^D \text{ to be } \mathbb{R}^D \text{ without a softmax layer. Every model is trained from scratch. Most of the hyper parameters are based on the implementation ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) with a slight modification (i.e., fixed learning rates: } 10^{-3} \text{, momentum: 0.9, weight decay: } 10^{-4}, \text{ mini-batch size: 100).}
\]

\[\text{This can be viewed as a } l_2 \text{ version of the contrastive loss ([Chopra, Hadsell, and LeCun 2005]); this is how HingeEmbeddingCriterion is implemented for the pairwise loss in Torch7 ([Collobert, Kavukcuoglu, and Farabet 2011]).}
\]

\[\text{The improved performance is reported in ([The Moodstocks team repository](https://github.com/moodstocks/Idsi)) with an even simpler architecture and without using ensemble. For brevity, we denote this improved version simply as IdsiaNet. Detailed information can also be found in the supplementary material.} \]
optimizer: stochastic gradient descent). Models are trained until convergence is reached, i.e., \( \frac{L_t - L_{t-1}}{L_{t-1}} < 5\% \) where \( L_t \) denotes the loss value at \( t \)-th iteration. We observed that the models typically converged around 15k-20k iterations. All networks were implemented using Torch7 (Collobert et al. 2011).

**Dataset**

We use two traffic-sign datasets, GTSRB (Stallkamp et al. 2012) and Tsinghua-Tencent 100K (Zhu et al. 2016) (TT100K). Since our motivation can be considered to involve dealing with a deficient data regime and a data imbalance caused by rare classes, we additionally introduce a subset split from GTSRB, referred to here as GTSRB-sub. To utilize the dataset properly for our evaluation schemes, given the train and test set splits provided by the authors, we further split them into seen, unseen and validation partitions, as illustrated in Fig. 3. The validation set is constructed by random sampling from the given training set. A description of the dataset construction process follows. For more details, the reader can refer to the supplementary material.

**GTSRB-all** GTSRB contains 43 classes. The dataset contains severe illumination variation, blur, partial shadings and low-resolution images. The benchmark provides partitions into training and test sets. The training set contains 39k images, where 30 consecutively captured images are grouped, called a “track”. The test set contains 12k images without continuity and, thus does not form tracks. We selected 21 classes that have the fewest samples as unseen classes with the remaining 22 classes as the seen classes. Template images are involved in the dataset. **GTSRB-sub** To analyze the performance of the deficient data regime, we created a subset of GTSRB-all, forming a smaller but class-balanced training set with sharing the test set of GTSRB-all. Hence, numeric results will be compatibly comparable to that from GTSRB-all. For the training set, we randomly select 7 tracks (i.e., 210 images) for each seen classes.

**TT100K-c** The TT100K detection dataset includes over 200 sign classes. We cropped traffic sign instances from the scenes to build the classification dataset (called TT100K-c). Although it contains the huge number of classes, most of the classes do not have enough instances to conduct the experiment. We only selected classes having official templates available and a sufficient number of instances. We split TT100K-c into the train and test set according to the number of instances. We set 24 classes with \( \geq 100 \) instances for seen classes and select 12 unseen classes as those having 50-100 instances. The training set includes half of the seen class samples, and the other half is sorted into the test set.

---

**Table 1:** Evaluations of the proposed quadruplet network with varying parameters. Evaluation is conducted on the validation set of GTSRB-all. Accuracy (%) on validation set is reported.

| Embedding dim (dim 100) | Top1 NN Avg. Seen | Unseen |
|-------------------------|------------------|--------|
| \( D = 50 \)            | 67.1             | 92.2   | 40.8  |
| \( D = 100 \)           | 69.1             | 95.3   | 41.6  |
| \( D = 150 \)           | 68.9             | 94.2   | 42.4  |

(a) Varying embedding dimension \( d \).

| Loss terms | Top1 NN Avg. Seen | Unseen |
|------------|------------------|--------|
| HingeM-3   | 67.3             | 93.1   | 40.3  |
| HingeM-5   | 69.1             | 95.3   | 41.6  |
| HingeM-6   | 68.9             | 97.3   | 39.2  |

(b) Varying number of pairwise loss terms.

---

**Figure 4:** Some of failure cases for unseen recognition.

---

**One-shot classification**

We perform one-shot classification by 1-nearest neighbor (1-NN) classification. For 1-NN, the Euclidean distance is measured between embedding vectors by forward-feeding a query (real image) and the anchors (template images) to the network, after which the most similar anchor out of the \( C \) classes is found (\( C \)-way classification). For the NN performance, we measure the average accuracy for each class. The seen class performance is also reported for a reference purpose.

**Self-Evaluation** The proposed Quadruplet network has two main factors: the embedding dimension \( D \) and the number of pairwise loss terms. We evaluate these on the validation sets \( \Psi_s \) and \( \Psi_u \) of GTSRB-all. The Quadruplet network is trained only with the seen training set \( \Phi_s \).

**Embedding dimension:** We conduct the evaluation while varying the embedding dimension \( D \), as reported in Table 1a. We vary \( D \) from 50 to 150 and measure the one-shot classification accuracy. We observed that the overall average accuracy across the seen and unseen classes peaked at \( D = 100 \). While the unseen performance may increase further beyond \( D = 150 \), since this sacrifices the seen performance, we select the dimension with the best average accuracy, i.e., \( D = 100 \), as the reference henceforth.

**pairwise loss terms:** The quadruplet model gives 6 possible pairwise distances between outputs. Intuitively, three

---

[The official templates are provided by the Beijing Traffic Management Bureau at, http://www.bjjtgl.gov.cn/english/trafficsigns/index.html]
Table 2: One-shot classification (Top 1-NN) accuracy (%) on the unseen classes. Performance on seen classes are shown as an additional reference.

| Network type | Anchor | Datasets        | GTSRB | GTSRB subset | TT100K |
|--------------|--------|-----------------|-------|--------------|--------|
|              |        | Seem            | Unseen| Seem         | Unseen | Seem | Unseen | Seem | Unseen | Seem | Unseen |
| IdsiaNet     | ×      | 74.6            | 40.2  | 59.4         | 34.7   | 87.7 | 14.6   |
| Triplet      | Real   | 62.6            | 33.8  | 54.9         | 25.5   | 5.9  | 1.0    |
| Triplet-DA   | Template | 86.0           | 54.8  | 60.8         | 47.6   | 86.2 | 67.1   |
| Quad+contrastive-5 | Template | 92.4 | 42.7 | 69.2 | 41.8 | 92.0 | 67.5 |
| Quad+hingeM  | Template | 90.8 | 45.2 | 70.1 | 47.8 | 94.9 | 67.3 |

Table 3: One-shot classification (Top 1-NN) accuracy (%) from GTSRB-all to TT100K-c.

| Network trained on GTSRB | Top1 NN sample average |
|--------------------------|------------------------|
| IdsiaNet                 | 36.5                   |
| Triplet DA               | 34.1                   |
| Quadruple+ hingeM        | 42.3                   |

possible options, i.e., HingeM-3, HingeM-5 and HingeM-6 in Eqs. (4), satisfy our co-domain embedding property. The results in Table 1b show a trade-off between different losses. HingeM-3 performs worse on both seen and unseen classes than the others, while the other two have a clear trade-off. This implies that HingeM-3 is not producing enough information (i.e., gradients) to learn a good feature space. HingeM-5 outperforms Loss6 on unseen classes but HingeM-6 is better on seen classes. We suspect that complex pairwise relationships among real samples may lead to a feature space which is overly adapted to seen classes. This trade off can be used to adjust between improving the seen class performance or regularizing the model for greater flexibility. We report the following results based on HingeM-5 hereafter.

Comparison to other methods We compare the results here with those of other methods on the test sets Ωs and Ωu of three datasets. All networks are trained only with the seen training and validation set Φs ⊃ Ψs of respective datasets.

For IdsiaNet, we use the activation of FC1 for feature embedding. For the other networks, we use the final embedding vectors. For the Quadruplet network, we test two cases with contrastive-5 and HingeM-5.

Table 2 shows the results on the three datasets, where Triplet has the lowest performance, while Triplet-DA performs well. Moreover, our quadruplet network outperforms Triplet-DA in most of cases for both seen and unseen classes. This supports our hypothesis that the template (also a different domain) anchor based metric learning and the quadruplet relationship may be helpful for generalization purposes.

To check the embedding capability of each approach further, we trained each model on GTSRB Φs ⊃ Ψs and tested on TT100K-c Ωs ⊃ Ωu. This experiment qualifies how the networks perform on two completely different traffic sign datasets. Table 2 shows the top1-NN performance of each model. Our model performs best in the transfer scenario, which implies that good feature representation is learned.

In order to demonstrate quantitatively the behavior of the Quadruplet network, we visualize unseen examples that are often confused by Quadruplet in Fig. 4. The unseen classes of examples that are highly similar to other classes are challenging even for humans; furthermore due to the poor illumination condition, motion blur and low resolution.

Learned representation evaluation In this experiment, we evaluate the generalization behavior of the proposed quadruplet network, which is analyzed by comparing the representation power of each network over unseen regimes (and seen class cases as a reference). In order to assess the representation quality of each method purely, we pre-train competing models and our model for Φs ⊃ Ψs of each dataset (i.e., only on seen classes), fix the weights of these models, and use the activations of FC1 (350 dimensions) of them as a feature. Given the features extracted by each model, we measure the representation performance by separately training the standard multi-class SVM (Chang and Lin 2011) with the radial basis kernel such that the performance heavily relies on feature representation itself.

We use identical SVM parameters (nonlinear RBF kernel, C = 100, tol = 0.001) in all experiments. In contrast to FC1 trained only on seen data, the SVM model is trained on both seen and unseen classes with an equal number of instances per class, and we vary the number of per class training samples: 10, 50, 100, and 200. SVM training samples are randomly sampled from the set Φs ⊃ Ψs ⊃ Φu ⊃ Ψu (Fig. 5), and the entire test set Ωs ⊃ Ωu is used for the evaluation. We report the average score and confidence interval by repeating the experiments 100 times for the case of [No. instances/class: 10] and 10 times for the cases of [No. instances/class: 50, 100, 200]. For each sampling, we fix a random seed and test each model with the same data for a fair comparison.

With the three datasets, we show the results of the unseen Ωu and seen Ωs test cases in Table 4 and 5, respectively. The unseen and seen class datasets are mutually exclusive; hence, the errors should be compared in each dataset independently, e.g., the numbers in Table 4(a) are not directly comparable with those in Table 5(a). Instead, we can observe the algorithmic behavior difference by comparing Table 4 and 5 with respect to the relative performance.

In the unseen case in Table 4, for all of the results, the proposed Quadruplet variants outperform the other competing methods, supporting the fact that our method generalizes well in limited sample regimes by virtue of the richer information from the quadruples. As an addendum, interestingly, Triplet-DA performs better than Triplet only with GTSRB-sub. We postulate that, in terms of feature description with some amount of strong supervision support, Triplet generalizes better than Triplet-DA, due inherently to the number of possible triplet combinations of Triplet-DA (|T|·|X|^2), and generally |T| ≪ |X| is much smaller than that of Triplet due to the use of a template anchor as an element of triplet, while Triplet improves all the pairwise possibility of real data (i.e., |X|^3). For the quadruplet case, more pairwise relationships

\footnote{We follow an evaluation method conducted in (Tran et al. 2015), where the qualities of deep feature representations are evaluated in the same way.}
results in higher number of tuple combinations and leads to better generalization, even with the use of templates. In the seen class case in Table 5, our method and 

\[\text{IdsiaNet} \]

show comparable performance outcomes with the best accuracy levels, while 

\[\text{Triplet} \]

and 

\[\text{Triplet-DA} \]

do not perform as well. We found that the proposed 

\[\text{Quad.}^+ \]

performs better than the other approaches when the number of training instances is very low, i.e., 10 samples per class. More specifically, 

\[\text{IdsiaNet} \]

performs well in the seen class case, but not in the unseen class case compared to 

\[\text{Quad.}^+ \].

This indicates that the embedding of 

\[\text{IdsiaNet} \]

is trapped in seen classes rather than in general traffic sign appearances. On the other hand, the proposed method learns more general representation in that its performance in both cases is higher than those of its counterparts. We believe that this is due to the regularization effect caused by the usage of templates.

### Conclusion

In this study, we have proposed a deep quadruplet for one-shot learning and demonstrated its performance on the unseen traffic-sign recognition problem with template signs. The idea is that by composing a quadruplet with a template and real examples, the combinatorial relationships enable not only domain adaptation via co-embedding learning, but also generalization to the unseen regime. Because the proposed model is very simple, it can be extended to other domain applications, where any representative anchor can be given. We think that \( N > 4 \)-tuple generalization is interesting as a future direction in that there must be a trade-off between over-fitting and memory capacities.

### Acknowledgment

This work was supported by DMC R&D Center of Samsung Electronics Co.

### References

- Atkeson, Moore, and Schaal 1997
- Atkeson, C. G.; Moore, A. W.; and Schaal, S. 1997. Locally weighted learning. \textit{Artificial Intelligence Review} 11(1-5):11–73.
- Bellet, Habrard, and Sebban 2015
- Bellet, A.; Habrard, A.; and Sebban, M. 2015. Metric learning. \textit{Synthesis Lectures on Artificial Intelligence and Machine Learning} 9(1):1–151.
- Bromley et al. 1993
- Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R. 1993. Signature verification using a siamese time delay neural network. In \textit{NIPS}.
**Supplementary Materials**

Here, we present additional details of the baseline network and datasets that could not be included in the main text due to space constraints. All figures and references in this supplementary file are consistent with the main paper.

**Base neural network architecture**

**Base network**  
We select the network based on classification performance under the assumption that high classification performance stems from good feature embedding capabilities. The Idsia network (Ciresan et al. 2012) is a winner in a German Traffic Sign Recognition Benchmark (GTSRB) competition. It was reported that variants of the Idsia network perform even better. The variant IdsiaNet uses a deeper convolution kernel size for each layer (Table 6). We use the best performing Idsia-like network6 for all three embedding schemes in the experiments.

Table 6: Variant IdsiaNet structure. Zero pad 2 before each conv layers. (LCN: local contrast normalization)

| Layer | Type        | Size       | kernel size |
|-------|-------------|------------|-------------|
| 0     | input       | 48x48x3    | -           |
| 1     | conv1       | 40x40x150  | 7x7         |
| 2     | ReLU        | 46x46x150  | -           |
| 3     | Max pooling | 23x23x150  | 2x2         |
| 4     | LCN         | 23x23x150  | 7x7         |
| 5     | conv2       | 24x24x200  | 4x4         |
| 6     | ReLU        | 24x24x200  | -           |
| 7     | Max pooling | 12x12x200  | 2x2         |
| 8     | LCN         | 12x12x200  | 7x7         |
| 9     | conv3       | 13x13x300  | 4x4         |
| 10    | ReLU        | 13x13x300  | -           |
| 11    | Max pooling | 6x6x300    | 2x2         |
| 12    | LCN         | 6x6x300    | 6x6         |
| 13    | fc1         | 350        | 10800 → 350 |
| 14    | ReLU        | 350        | -           |
| 15    | fc2         | 43         | 350 → 43    |
| 16    | softmax     | 43         | -           |

**Details of datasets**

**GTSRB-all**  
GTSRB (Stallkamp et al. 2012) has been the most widely used and largest dataset for traffic-sign recognition. It contains 43 classes categorized into three super categories: prohibitory, danger and mandatory. The dataset contains illumination variations, blur, partial shadings and low-resolution images. The benchmark is partitioned into training and test sets. The training set contains 39,209 images, with 30 consecutively captured images grouped, which is referred to here at a track. The test set contains 12,630 images without continuity and thus does not form tracks. The number of samples per class varies, and the sample size also has a wide range (Fig. 6 & Fig. 7). To validate the embedding capability of the proposed model, we split the training set into two sets: seen and unseen (Fig. 5). Our motivation is to overcome the data imbalance caused by rare classes. We selected 21 classes that have the fewest samples as the unseen class, using the remaining 22 classes as the seen class. For training, we only used seen classes in the training set.

**GTSRB-sub**  
To analyze the performance on a dataset with a limited size, we created a subset of GTSRB-all, forming a smaller size, as a class balanced training set. For the training set, we randomly selected seven tracks for each of the seen classes. For the test set, we used the same set as used in GTSRB-all; hence results will be compatible with and comparable to those of GTSRB-all.

**TT100K-c**  
Tsinghua-Tencent 100K (Zhu et al. 2016) (TT100K) is a chinese traffic sign detection dataset that includes more than 200 sign classes. We cropped traffic sign instances from scenes to build a classification dataset (called TT100K-c) to validate our model. Although it contains numerous classes, most of those in each class do not have enough instances to allow the experiment to be conducted. We only selected classes with official templates7 available and a sufficient number of instances. We filtered out instances that have side lengths of less than 20 pixels, as they are unrecognizable. We split TT100K-c into the training and test set according to the number of instances. We set 24 classes that have more than 100 instances as the seen classes, and assign 12 classes with between 50 and 100 instances as the unseen class (Fig. 9). The training set includes half of the seen class samples, with the other half going to the test set. All of the unseen class samples belong to the test set. The size distribution and the number of samples per class are provided in Fig. 9 and Fig. 10 respectively.

6IdsiaNet (Ciresan et al. 2012) is tuned to achieve better performance. Please check the repository, https://github.com/moodstocks/gtsrb.torch#gtsrbtorch

7The official templates are provided by Beijing Traffic Management Bureau, http://www.bjjtgl.gov.cn/english/trafficsigns/index.html
Figure 5: GTSRB template images.

Figure 8: TT100K template images.

Figure 6: GTSRB class sample distribution. Blue: seen classes (#22). Orange: unseen classes (#21).

Figure 9: TT100K class sample distribution. Blue: seen classes (#24). Orange: unseen classes (#12).

Figure 7: GTSRB size distribution. Side length = \( \frac{\text{height} + \text{width}}{2} \)

Figure 10: TT100K size distribution. Side length = \( \frac{\text{height} + \text{width}}{2} \)