- ORIGINAL ARTICLE -

A Comparison of Small Sample Methods for Handshape Recognition

Una Comparación de Métodos para Reconocimiento de Formas de Manos con Pocas Muestras.

Facundo Quiroga[2,4], Franco Ronchetti[1,2,3], Ulises Jeremias Cornejo Fandos[2], Gastón Gustavo Ríos[2,4], Pedro Dal Bianco[2,4], Waldo Hasperué[2,3], and Laura Lanzarini[2]

[1] *Universidad del CEMA, Buenos Aires (1054), Bs As, Argentina*
[2] *Instituto de Investigación en Informática LIDI, Universidad Nacional de La Plata, Facultad de Informática, UNLP – CIC. La Plata (1900), Bs As, Argentina.*
[3] *Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC-PBA), La Plata (1900), Bs As, Argentina.*
[4] *Becario de la Universidad Nacional de La Plata, La Plata (1900), Bs As, Argentina.*
fquiroga@lidi.info.unlp.edu.ar

Abstract

Automatic Sign Language Translation (SLT) systems can be a great asset to improve the communication with and within deaf communities. Currently, the main issue preventing effective translation models lays in the low availability of labelled data, which hinders the use of modern deep learning models. SLT is a complex problem that involves many subtasks, of which handshape recognition is the most important. We compare a series of models specially tailored for small datasets to improve their performance on handshape recognition tasks. We evaluate Wide-DenseNet and few-shot Prototypical Network models with and without transfer learning, and also using Model-Agnostic Meta-Learning (MAML). Our findings indicate that Wide-DenseNet without transfer learning and Prototipical Networks with transfer learning provide the best results. Prototypical networks, particularly, are vastly superior when using less than 30 samples, while Wide-DenseNet achieves the best results with more samples. On the other hand, MAML does not improve performance in any scenario. These results can help to design better SLT models.

Keywords: sign language, handshape recognition, DenseNet , prototypical networks, MAML , transfer learning, small datasets

Resumen

Los sistemas de traducción automática de lengua de señas (SLT, por sus siglas en inglés) pueden ser una gran ayuda para mejorar la comunicación con las comunidades sordas así como también entre ellas. Actualmente, el principal obstáculo para el desarrollo de modelos de traducción efectivos es la falta de datos etiquetados, que impide el uso de métodos de aprendizaje automático profundo modernos.La traducción de lengua de señas es un problema complejo que involucra varias subtareas, de las cuales el reconocimiento de la forma de la mano es la más importante. En este trabajo, comparamos una serie de modelos especialmente adaptados para ser entrenados con pocas muestras en la tarea de reconocer formas de mano. Evaluamos los modelos WideDenseNet y Prototypical Networks, con y sin el uso de transferencia de aprendizaje, y también el model Model-Agnostic Meta-Learning (MAML). Nuestros resultados indican que el modelo Wide-DenseNet sin transferencia de aprendizaje y las Prototypical Networks con transferencia de aprendizaje obtienen los mejores resultados. Las Prototypical Networks son vastamente superiores cuando se utilizan menos de 30 muestras, mientras que Wide-DenseNet es superior en el resto de los casos. Por otro lado, MAML, que es un método diseñado específicamente para estos casos, no mejora el desempeño en ningún caso. Estos resultados pueden ayudar a diseñar mejor los componentes de un sistema de traducción de lengua de señas.

Palabras claves: Lengua de señas, reconocimiento de formas de mano, DenseNet, Redes Prototipicas, MAML, Transferencia de aprendizaje, Datasets pequeños

1 Introduction

Sign languages (SL) are commonly used by deaf people. They employ handshapes and hand movements, as well as facial expressions and postures with the body to communicate meaning in an equivalent way to oral and written languages.

Sign Language Translation (SLT) is a field in the intersection of computer vision and language translation. SLT's goal is to create systems that can translate videos of people speaking in sign language into another language, typically a textual language such as english or spanish [1, 2, 3, 4].

Producing a model capable of SLT with high precision would improve the quality of life of many people since an automatic interpreter would facilitate communication between signers and non signers [3].

Sign language Translation presents an interesting challenge as the available data is limited compared to other problems such as speech recognition [2]. The nature of the problem (relatively small number of signers, multimodal input, regional differences) makes it hard to create new sign language datasets. Merging datasets from different regions or countries is very difficult. Each sign language has its own set of signs. Furthermore, sign languages of different countries vary in their syntax and semantic, even .

As mentioned before, signs are defined by various features, such as the facial expression of the signer, body pose, joint movement and handshape. Of these, handshapes are the most important features [2, 5]. Therefore, SLT systems require a high level of performance in the handshape recognition subtask to function properly. Currently, handshape data is available mostly as 2D RGB images, obtained from sign language videos or separately [2, 5].

In this work we propose to evaluate and compare new methods devoted to deal with small datasets for handshape recognition tasks.

We address the low availability of data by implementing a variety of state-of-the-art convolutional neural network models and training techniques designed to tackle small labelled datasets. We compare two model architectures: Prototypical Network [6] and DenseNet [7]. We train these models with (i) regular training, (ii) Transfer Learning and (iii) Model Agnostic Meta Learning (MAML) [8].

DenseNet is a well known state-of-the-art model that has shown good performance for image classification even in cases where there is a low amount of labelled data[7]. When the amount of available data is reduced even further, few-shot learning techniques are required. We chose Prototypical network as our specialised few-shot learning model. Prototypical networks are models based on metric learning, optimising a distance function between classes in an embedded space to classify each sample.

Our contributions consists not of proposing new techniques for dealing with small datasets, but performing a comparison of current state of the art techniques for improving model performance in these conditions: prototypical networks for few shot learning, transfer learning and model-agnostic meta-learning (MAML).

In the following section we summarise previous efforts on training Convolutional Neural Networks (CNNs) on handshape datasets. Section 3 describes the datasets and section 4 describes models and techniques we employed in our experiments, which are detailed along with results in Section 5, and Section 6 contains the conclusion of our work.

2 Related work

Recent years have seen the rise in the use of deep learning models for sign language recognition, specifically the use of convolutional neural networks to extract image features or classify hand images.

[9] trained a CNN to recognise handshapes from the RWTH handshape dataset, which contains 3200 labelled samples and 50 different classes. The model was based on a pre-trained network with a VGG architecture, and employed a semi-supervised scheme to take advantage of approximately one million weakly labelled images, achieving an accuracy of 85.50%. To the best of our knowledge, Koller's work was the first to include a technique to overcome the lack of labelled data in the specific context of handshape recognition.

[10] employed a radon transform as a feature for an ad-hoc classifier that employed clustering as a quantization step and K nearest neighbours for the final classification. They tested the model on the LSA16 dataset, which contains only 800 examples, obtaining an accuracy of 92.3%. [11] evaluated several CNNs on the LSA16 and RWTH datasets, including both vanilla and pretrained models. The use of pretrained models helps to alleviate the lack of labelled data, specially if the pretrained convolutional filters are general enough to exploit for other tasks such as handshape recognition. Their best models achieve an accuracy of 95.92% for LSA16 and 82.88% for RWTH, the latter without an unsupervised pretraining scheme.

[12] obtained an accuracy of 99.20% with a simple neural network and a custom dataset they created which contains 6000 examples and 10 classes. [13] trained a deep CNN on the Hand Gesture Dataset LPD, which contains 3250 images of only 6 classes, also obtaining a very high accuracy (99.73%).

[14] evaluated a CNN on a custom dataset with 36 classes, 8 subjects and 57000 sample images, obtaining an accuracy of 94.17%. However, the samples correspond to video sequences and therefore are highly correlated; while there are approximately 2000 images per class, there are only eight image sequences, one for each subject. Since each image sequence contains approximately 250 images which are highly correlated, they only consider eight image per sequence per class.

[15] used the Jochen Triesch Database (JTD), which contains 72 samples for each the 10 classes, and the the NAO Camera Hand Posture Database, containing 4 classes and 400 examples per class. They trained a simple CNN with a multichannel image containing the results of the Sobel operator as input, obtaining an F-score of 94% and 98% in each dataset perspective.

[16] trained a simple CNN with only 6 layers using the ASL Finger spelling dataset, obtaining an accuracy of 80.34%. The dataset consists of 60000 images of 25 different classes, but they were captured as videos so they are also highly correlated as in the previous case.

[[illegible]] performed experiments with Wide-DenseNet and Prototypical Networks on the CIARP, LSA16 and RWTH datasets using vanilla models. In both cases, they also quantify the impact of data augmentation on accuracy. Their best models obtain an accuracy of 99.26% on LSA16, 94.00% on RWTH and 100.00% on CIARP. This work is the third ([[illegible], [illegible]]) and last instance we found where a specific strategy was employed to compensate for the lack of data.

This review confirms our previous statement that, while CNN are being consistently applied to handshape recognition tasks, most of these datasets are small and ad-hoc. In some cases datasets are so small that it is very easy to obtain near-perfect performance with simple models. Additionally, many datasets are recorded specifically for the purpose of testing a single model.

Therefore, there is a lack of a common pool of datasets used as standard benchmarks for handshape recognition models. Finally, many datasets are not readily available, given that the authors have not published the data and do not provide any means of obtaining it [[illegible]].

3 Datasets

We selected three datasets, LSA16 [[illegible]], RWTH-PHOENIX-Weather (RWTH) [[illegible]] and CIARP [[illegible]] (Table 1). These well-known datasets contain RGB images whose setting varies greatly and possess different quantities of examples or distributions of samples per class. In this way we can evaluate the models in a variety of contexts.

We note that the RWTH and LSA16 are both publicly available and current models have been shown to achieve less than perfect accuracy for these datasets. While the dataset in [[illegible]] (denoted CIARP in this paper) has been solved completely, it is interesting and complementary because it targets general handshapes instead of those specific to sign language.

We briefly describe the main characteristics of each dataset.

3.1 LSA16

LSA16 [[illegible]] contains images of 16 handshapes of the Argentinian Sign Language (LSA). The dataset is balanced, with 50 images per class, where each handshape class was performed 5 times by 10 different signers. This gives a total of 800 images of size 32x32. The subjects wore coloured hand gloves and dark clothes on a white background. There is only one hand in each image. The hands are centred and segmented from the background.

3.2 RWTH

RWTH [[illegible]] is composed of a selection 3359 handshapes of 45 classes. The images, of size 132x92, were cropped from videos of the sign language interpreters at the German public TV-station PHOENIX. The interpreters wore dark clothes in front of an artificial grey background. This is a challenging dataset since many images possess significant movement blur, others contain both hands of the interpreter and hands are not always perfectly centered.

The dataset is highly imbalanced with some classes having just 1 sample while others have as many as 529 samples. We removed classes that had less than 40 samples following [[illegible]], to guarantee a minimum amount of images per class for the networks to learn.

3.3 CIARP

CIARP [[illegible]] contains 6000 images of size 38x38 acquired by a single colour camera. The images are split into 10 classes so that each class has 600 samples. The hands are centred and were segmented from the background, which was replaced by black pixels. The combination of small image size and low amount of classes makes this problem relatively easy when compared to LSA16 and RWTH. Finally, the classes in the dataset correspond to handshapes which are not strictly based on sign language, but are similar enough so that the comparison remains valid.

4 Architectures and Techniques

We compare two different base classification models to analyse their ability to learn from these small handshape datasets: Prototypical Networks [[illegible]] and DenseNet [[illegible]].

Prototypical Networks was designed explicitly to deal with small sample sizes. On the other hand, DenseNet is currently a state-of-the-art model in image classification with convolutional neural networks, and while it has not been explicitly designed for small datasets, it has shown exceptional performance in many different tasks.

Therefore, we also evaluate DenseNet models trained with Model-Agnostic Meta-Learning (MAML) [[illegible]] and Transfer Learning [[illegible]] training techniques, in addition to the traditional training process (Table 2). These techniques have been designed or can be adapted for small sample size settings.

Transfer Learning is a well known technique to jump-start the training of neural networks for a problem A using datasets from a different problem B. The weights of a network trained on B are used as initial weights in the training of the network for A. Retraining the network for A is called finetuning, and may retrain only a subset of the weights of the network. However, it still may require large amounts of data for the finetuning phase.

Finally, MAML is a meta-learning technique for few-shot learning, that involves learning subtasks. In this context, each subtask corresponds to a different

Table 1: Main statistics for the three datasets: RWTH, LSA16 and CIARP

Dataset	Samples	Classes	Balanced	Origin	Best Accuracy
RWTH	3359	45	No	TV images	94.00% [17]
LSA16	800	16	Yes	Created specifically for SLT	99.26% [17]
CIARP	6000	10	Yes	General handshapes (not SLT specific)	100.00% [17]

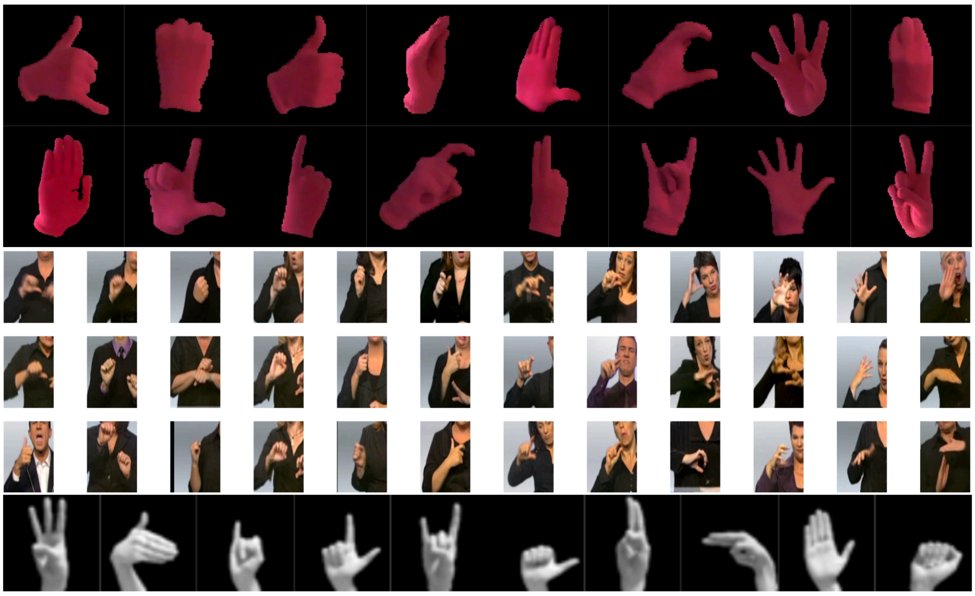

Figure 1: Sample images from the LSA16 (first row), RWTH-PHOENIX-Weather (second row) and CIARP (third row) datasets. Each image corresponds to a different class of each dataset.

class, therefore allowing the training of the model for new classes as an adaptation in the meta-learning scheme.

Table 2 shows a summary of the model and training schemes we utilised. The DenseNet model was trained via a normal gradient descent optimisation procedure, optionally pre-initialising the weights via transfer learning. We also used MAML to train the DenseNet for better adaptation with a small dataset. Finally, Prototypical Networks only used a normal training scheme, without Transfer Learning or MAML.

In the following subsections we briefly describe each of these models and training techniques.

Table 2: Models and training schemes evaluated.

Training schemes	DenseNet	Prototypical Networks
Normal Training	✓	✓
Transfer Learning	✓	✗
MAML	✓	✗

4.1 Wide-DenseNet

We selected a DenseNet based architecture as it is a state-of-the-art model in many domains and can handle small datasets with low error rate [19].

DenseNet [7] works by concatenating the feature-maps of a convolutional block to the feature-maps of all the previous convolutional blocks and using the resulting set of feature-maps as input for the next convolutional block. In this way, each convolutional block receives all the collective knowledge of the previous layers maintaining the global state of the network which can be accessed.

We employed a variation on DenseNet called Wide-DenseNet which follows the strategy used by wide residual networks [20]. Wide-DenseNet consists on decreasing the depth of the network and increasing the width of the layers. This way the model can be trained faster by optimising feature reuse and obtain highers accuracy in some takss.

Additionally, we use Squeeze and Excitation blocks (SE blocks) [21] to improve the performance of the Wide-DenseNet model. Convolutional networks construct informative features by combining both spatial and channel-wise information within local receptive fields at each layer. On the other hand, the SE blocks improve the relevance of the representations by modelling the interdependency between channels in order to perform feature recalibration. Additionally, these blocks can improve the performance of most convolutional models with a very low computational cost. Therefore, we use SE blocks between dense and transition blocks, as shown in Figure 2.

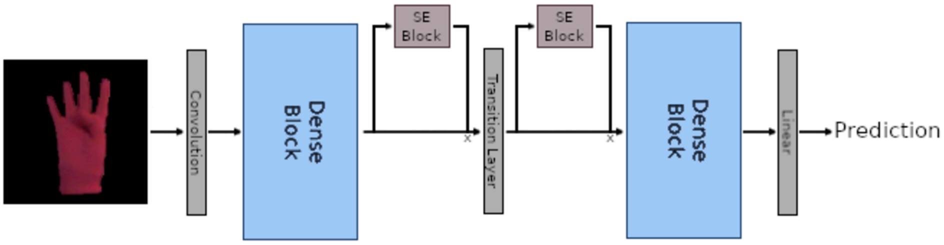

Figure 2: Wide-DenseNet using 2 dense blocks and SE blocks.

4.1.1 Transfer Learning

Gathering new training data for deep neural networks can be an expensive and time consuming task. Transfer learning provides a way to utilise already available data from a source domain and transfer the acquired knowledge from this source domain to a target domain. By performing transfer learning we can obtain much better initialisations of the parameters of the model before training in the target domain.

In the past, transfer learning has been used for handshapes, sign language and gesture recognition [22][11][23] demonstrating the advantages of this technique.

In this work, we employ network-based [18] transfer learning. In this type of transfer learning a part of the network pretrained on the source domain is reused for the training in the target domain. The objective is for the neural network to acquire good priors from the source domain and transfer this knowledge to the target task.

To obtain a good performance from transfer learning the source dataset usually has to be larger than the target dataset. Since the information extracted from the target dataset has a higher value than the information from the source dataset, the data from the target dataset will be more helpful in fitting the target task. In addition to this, it is important for the source domain to be related to the target domain. If the relation between both domains differ too much it is possible to get a negative transfer which can diminish the performance obtained by using transfer learning [24].

4.1.2 Model-Agnostic Meta-Learning

Similarly to Prototypical Networks, Model-Agnostic Meta-Learning (MAML) [8] is a technique designed to tackle the problem of few-shot learning. MAML learns how to improve a model so that it can learn a new task in only a few steps by training on many different tasks, commonly phrased as learning to learn. MAML does this by learning over multiple tasks and updating the parameters of the models based on the improvement obtained after training on each task.

More formally, given a set of tasks T each consisting of a loss function L and a set of elements with their corresponding labels. MAML requires a distribution over tasks $p(T)$ that we want to adapt to. Given those distributions, we proceed with the next two steps, task training and meta training. In the task training we sample K tasks. For each task T_i, the model is trained on a set of elements extracted from the task distribution using the loss function L_i belonging to T_i. With the updated parameters the model is then tested on new data from T_i. Once tested on each task the loss obtained from these tests will be added and utilised as loss for the initial model on the meta training obtaining a new initial model with better initial parameters that will grant a bigger improvement for each task on fewer steps.

We made some modifications on the original MAML to work with bigger datasets in a supervised way. Each task T_i is split in 2 subsets, a training subset Tt and a meta training subset Tm. The subsets are composed of datasets $D = x, y$ where x is an image and y the label of that image. Each subset has an equal size b and its labels are mirrored $y \in Tt \iff y \in Tm$. Each dataset with size n has $\frac{n}{2b}$ tasks T. We consider our model as a function f_θ with parameters θ. In each training iteration θ will change to θ'. Each iterations consists of 2 steps, a training and a meta training step. In the training step we start by storing the value of θ in θ', then θ is updated to fit Tt_i. In the meta training step the new θ is used to calculate the gradients with $Tm_i : \nabla L_i(f_\theta(Tm_i))$ and these gradients are applied to θ' in each iteration.

4.1.3 Prototypical Networks

Prototypical Networks [6] is a meta-learning model for the problem of few-shot classification, where a classifier must generalise to new classes not seen in the training set, given only a small number of examples of each new class. Few-shot learning models are generally measured by their performance on n-shot, k-way classification tasks. In this setting, a model is given a set of query samples Q belonging to a new, previously unseen class. Afterwards, the model receives a support set S that contains n examples, chosen from k different unseen classes. Finally, the algorithm has to determine the classes of Q, given the samples of S.

Prototypical Networks apply an inductive bias in

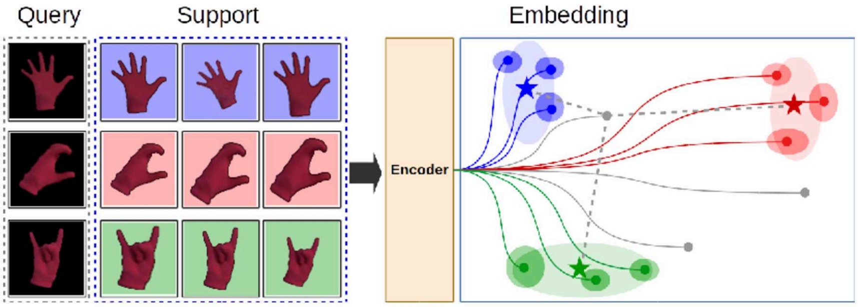

Figure 3: Prototypical Networks given a set of query samples and support set

the form of class prototypes to improve few-shot performance. Their key assumption is the existence of an embedding in which samples from each class cluster around a single prototypical representation which is simply the mean of the individual samples. In this fashion, we can generalise n-shot classification in the case of $n > 1$ as classification by simply choosing the label of the class prototype that is closest.

Schemes for few shot classification tasks like Prototypical Networks can also be of use for training models with small datasets.

5 Experiments

We performed experiments with the models described in section 4 and the datasets described in section 3.

Given that the datasets are small we perform an initial stratified split, using 25% of the data for testing. We also defined validation subsets for each training, with 10% of the samples of that subset, to perform model dependent hyperparameter optimization.

We performed experiments using the embedding architectures and configurations described in the following subsections. To analyse the impact of the size of the dataset, we limit the training sample sizes (while keeping the validation and test sets constant) to 5, 10, 15 and 20 samples per class. We also include the scenarios of 30, 40 and 100 samples per class for CIARP and 30 and 40 for RWTH since they have a larger number of samples. In the case of RWTH, classes that do not have at least 40 samples in the training subset are removed for the whole process. This difference in the number of samples was forced by the distribution of each dataset, but the comparison can be made directly in the cases of 5 to 30 samples.

The same data augmentation was used for each experiment, with which the best results were obtained in previous works [17]. We applied normalization feature-wise by subtracting the mean and dividing by the standard deviation of each feature. As data augmentations, we used horizontal flipping, random rotations from 0° to 10° and random spatial resampling. The resampling is performed by reducing each image by 10% or 20% in width and height.

5.1 Proposed methods

We describe the 4 different approaches or methodologies we evaluate to obtain models that work well with small datasets, including a model with a normal training scheme as baseline.

5.1.1 Normal training

Based on the results obtained in previous works [17], we use Wide-DenseNet in the following way. We include SE blocks after each dense and transition block. We trained the models using a batch size of 32, an initial learning rate of 10^{-3} and 200 epochs with early stopping using a maximum patience of 55. The resulting model used a growth rate of 64 and two dense blocks with 6 and 12 layers respectively, for all datasets.

5.1.2 Transfer Learning

We performed experiments with every dataset to figure out which Transfer Learning configuration is more convenient. For each dataset, experiments are carried out varying the dataset used to train the base model. With this approach, it is possible to define a dataset matrix to evaluate which dataset is the best option.

Our base architecture consists of a sequential model with a pretrained Wide-DenseNet model, followed by global average pooling, a hidden dense layer with a ReLU nonlinearity and finally a softmax classifier. Since CIARP contains only gray scale images, for that dataset a 3x3 convolutional layer was prepended to the model to generate 3 feature maps from the single original grayscale channel.

Furthermore, in this set of experiments CIFAR10 [25] and MNIST [26] are also used to train base models. In this way, we can analyse the impact of using a general purpose dataset instead of a handshape dataset for transfer learning.

5.1.3 Prototypical Networks

As mentioned in section 4.1.3, we can use Prototypical Networks' ability to work with small datasets even if all samples are labelled. Based on the results obtained in early experimentation, we use Prototypical Networks with the following configuration.

Our Prototypical Networks employ an embedding architecture composed of four convolutional blocks. Each block comprises a 64-filter 3 × 3 convolution, followed by a batch normalisation layer, a ReLU activation and 2*x*2 max-pooling.

We used the same encoder for embedding both support and query points. All of our models were trained with the ADAM [[illegible]] optimiser. We used an initial learning rate of 10^{-3} and cut the learning rate in half every 2000 episodes.

We trained the networks using the Euclidean distance in the 1-shot and 5-shot scenarios with training episodes containing 16, 20 and 10 classes (for LSA16, RWTH and CIARP respectively) and 5 query points per class when possible. We computed classification accuracy for our models by averaging over 1000 randomly generated episodes from the test set.

In early experimentation, we found that the difference in the results obtained in 1-shot and 5-shot scenarios for these datasets was very large. On the other hand, 5-shot scenarios gave better results. Therefore, we only used 1-shot learning in the experiments in which the minimum number of examples per class does not allow using 5-shot learning.

5.1.4 MAML

In this case, we test the performance of MAML applied to training scenarios similar to those used with Wide-DenseNet in the experiments described above. We trained a Wide-DenseNet model using the MAML technique over one task before doing the meta-learning. That task corresponded to a short training process on a *D* dataset following the process described in section 4.1.2. Then we use the trained model to initialise the weights for a new model. The new model is treated as a new task. Our model was trained on one task and we used this previous knowledge to initialise the weights of the model for a new task, using a training process similar to the used for the experiments we performed using Transfer Learning.

5.2 Results

We present the results of the experiments for various training set sizes, ordered by dataset (Tables 3, 4 and 5).

The results of the experiments using MAML and transfer learning are grouped according to the dataset with which the pretrained model is created.

In the case of RWTH, the *Full RWTH* column corresponds to an unbalanced set of RWTH which contains all samples. In this case, models were trained using weight classes to offset this imbalance.

5.3 Analysis

We analyse and compare the results obtained by each method for all datasets.

Wide-DenseNet For all three datasets, we notice the low accuracy obtained in the subsets of 5 samples and how the accuracy rapidly increases when the number of samples increases. However, in most cases using the full training dataset, that is, training a normal Wide-DenseNet CNN model, obtains the best performance.

Transfer Learning Transfer learning schemes also appear to have problems with very few data samples, as little as 5 or 10. Furthermore, using a handshape dataset to train the base model instead of a general purpose one such as CIFAR10 seems not to improve the accuracy.

Only in the case of CIARP the use of transfer learning gives significantly better results than those obtained by a Wide-DenseNet trained from scratch. This is curious, since CIARP's images are grayscale, and the models were not pretrained on grayscale data. In LSA16 and RWTH, transfer learning seems to produce only a slight accuracy improvement on occasions, while it decreases the model's performance in most cases. While MNIST as a pretraining dataset does not achieve the best results in any case, the performance is very similar to those obtained with other pretraining datasets, indicating that the natural image or handshape image prior provided by other datasets are not significantly superior or important for pretraining.

Protypical Networks Prototypical Networks have a greater accuracy than the rest of the models for the 5 and 10 samples scenarios for both RWTH and CIARP, and also the other splits in the case of LSA16. In the case of CIARP, it achieved a very good performance if not the best in all cases, and similar results to those obtained by Wide-DenseNet.

Regarding RWTH, it is clear that Prototypical Networks cannot take advantage of large sample sizes and the accuracy does not increase predictably as the number of samples do. It is also possible that Prototypical Networks obtained the lowest accuracy because the images of the hands were unsegmented, difficulting obtaining good class prototypes because of the difficulty of modeling backgrounds. In this case, the accuracy of Wide-DenseNet is slightly higher than Prototypical Networks model when the number of samples per class, N, is larger than 15.

MAML In general, MAML achieves low accuracy in the subsets of 5 samples and the accuracy increases when the number of samples is larger.

In the case of CIARP, the use of MAML gives significantly better results than those obtained by Wide-DenseNet for those cases in which the number of train-

Table 3: Accuracy of various convolutional neural network based models on CIARP.

Method	*5 samples*	*10 samples*	*15 samples*	*20 samples*	*30 samples*	*40 samples*	*100 samples*	*360 samples*
DenseNet	10.00 ± 0.00	10.00 ± 0.00	37.26 ± 36.81	76.07 ± 37.87	82.71 ± 38.35	99.83 ± 0.28	99.95 ± 0.04	**99.83 ± 0.24**
MAML	11.12 ± 2.24	26.53 ± 33.07	80.23 ± 35.13	98.85 ± 0.99	99.71 ± 0.15	99.56 ± 0.34	99.94 ± 0.04	99.01 ± 0.66
CIFAR10 Pretraining								
MAML	10.47 ± 0.94	10.00 ± 0.00	95.51 ± 2.15	98.41 ± 1.16	**99.73 ± 0.30**	98.92 ± 1.72	99.96 ± 0.02	99.44 ± 0.38
Transfer Learning	11.69 ± 3.39	27.24 ± 33.54	98.44 ± 0.47	98.71 ± 0.72	99.61 ± 0.21	99.64 ± 0.31	99.92 ± 0.05	99.59 ± 0.19
MNIST Pretraining								
MAML	10.00 ± 0.00	12.70 ± 3.27	80.13 ± 35.07	98.91 ± 0.47	99.64 ± 0.16	**99.74 ± 0.16**	**99.97 ± 0.04**	99.43 ± 0.64
Transfer Learning	10.00 ± 0.00	10.00 ± 0.00	97.31 ± 1.36	**99.25 ± 0.37**	99.30 ± 0.48	99.70 ± 0.31	99.95 ± 0.04	99.21 ± 0.51
LSA16 Pretraining								
MAML	11.38 ± 2.65	11.44 ± 2.49	80.10 ± 35.07	98.95 ± 0.64	99.51 ± 0.45	99.50 ± 0.62	99.93 ± 0.08	99.30 ± 0.61
Transfer Learning	11.19 ± 2.38	10.01 ± 0.02	97.82 ± 0.93	99.07 ± 0.45	99.51 ± 0.24	99.73 ± 0.33	99.93 ± 0.06	99.73 ± 0.26
RWTH Pretraining								
MAML	10.00 ± 0.00	10.00 ± 0.00	97.75 ± 0.52	99.07 ± 0.63	99.64 ± 0.18	99.65 ± 0.17	99.97 ± 0.02	99.08 ± 0.73
Transfer Learning	10.24 ± 0.49	10.03 ± 0.04	**98.59 ± 0.83**	98.33 ± 1.16	99.58 ± 0.20	99.74 ± 0.16	99.85 ± 0.09	99.42 ± 0.40
ProtoNet	**91.45 ± 2.44**	**94.48 ± 1.78**	95.70 ± 0.66	97.63 ± 0.68	98.38 ± 0.27	99.21 ± 0.28	99.82 ± 0.05	99.41 ± 0.16

Table 4: Accuracy of various convolutional neural network based models on LSA16.

Method	*5 samples*	*10 samples*	*15 samples*	*20 samples*	*30 samples*
DenseNet	6.56 ± 0.63	92.80 ± 2.89	92.81 ± 2.82	95.31 ± 1.23	96.13 ± 0.00
MAML	6.35 ± 0.20	93.95 ± 1.53	93.54 ± 1.60	95.72 ± 1.00	97.81 ± 0.38
CIFAR10 Pretraining					
MAML	6.25 ± 0.00	92.18 ± 1.47	94.06 ± 1.76	95.63 ± 2.60	97.08 ± 0.77
Transfer Learning	6.25 ± 0.00	92.91 ± 2.66	93.75 ± 2.30	94.79 ± 1.14	97.91 ± 0.46
MNIST Pretraining					
MAML	6.45 ± 0.41	92.91 ± 0.96	94.17 ± 2.45	95.63 ± 1.16	97.18 ± 0.41
Transfer Learning	6.25 ± 0.00	92.60 ± 2.07	93.43 ± 1.73	95.41 ± 2.42	97.08 ± 0.62
CIARP Pretraining					
MAML	6.77 ± 0.65	92.08 ± 2.51	94.58 ± 1.16	96.14 ± 1.34	96.97 ± 1.25
Transfer Learning	6.25 ± 0.00	92.70 ± 1.77	92.60 ± 1.87	96.67 ± 1.12	96.24 ± 1.48
RWTH Pretraining					
MAML	6.87 ± 0.76	92.81 ± 0.51	94.47 ± 2.29	94.37 ± 2.58	96.24 ± 1.00
Transfer Learning	6.97 ± 1.45	74.79 ± 34.27	93.43 ± 2.56	95.62 ± 1.21	96.97 ± 0.89
ProtoNet	**94.15 ± 1.27**	**94.64 ± 1.23**	**95.50 ± 1.01**	**97.20 ± 0.74**	**98.38 ± 0.22**

ing examples is less than 40, therefore the use of this method is advantageous. Considering this and the results obtained in LSA16, the use of Transfer Learning on those datasets means an advantage over the accuracy obtained by Wide-DenseNet model.

Summary From the obtained results, we can see that the performance of the Wide-DenseNet based models generally increases as more training examples are provided, as expected. The use of MAML paired with Transfer Learning only helps on CIARP and LSA16, with 15-30 samples, but not at all in the case of RWTH, in LSA16 it helps only slightly in that case. On the other hand, Prototypical Networks models do not show a significant increase in performance as the number of samples increases from 5 to 30, but provide the best accuracy for small sample sizes (less than 40 samples).

6 Conclusions

We have performed experiments to evaluate the accuracy of Prototypical Networks, Wide-DenseNet, MAML and Transfer Learning on three handshape recognition datasets. For every dataset, our models demonstrated state-of-the-art performance. All models achieve near-perfect accuracy on CIARP, even with very few samples per class. This shows that the dataset is too simple as a benchmark for handshape recognition. While it has more samples than the other datasets (6000), they are too homogeneous and do not have enough variation.

Wide-DenseNet without transfer learning and Prototypical Networks showed the best results. Wide-DenseNet saw better performance on bigger and more complex datasets, while Prototypical Networks was the best choice when facing very small training samples sizes. Prototypical Networks offers good performance even when the available training data is really low (5 to 10 samples for each class) but when more data is added to the training set a traditional model such as Wide-DenseNet trained from scratch performs better in most cases. MAML and transfer learning did not offer significant improvements for the tasks we used for evaluation. However, it is interesting that

Table 5: Accuracy of various convolutional neural network based models on RWTH.

Method	*5 samples*	*10 samples*	*15 samples*	*20 samples*	*30 samples*	*40 samples*	*Full RWTH*
DenseNet	10.32 ± 2.48	17.78 ± 17.08	46.66 ± 19.17	**64.59 ± 3.81**	**71.20 ± 2.24**	**71.85 ± 3.84**	**96.05 ± 0.96**
MAML	9.20 ± 1.97	9.86 ± 1.29	25.38 ± 19.31	30.51 ± 24.03	67.81 ± 1.99	67.13 ± 1.55	95.54 ± 1.21
CIFAR10 Pretraining							
MAML	8.36 ± 0.62	9.69 ± 1.94	8.66 ± 1.75	38.90 ± 24.40	64.56 ± 7.86	64.53 ± 3.18	96.08 ± 0.37
Transfer Learning	10.73 ± 1.59	17.78 ± 17.21	18.25 ± 16.33	48.19 ± 21.08	65.76 ± 6.50	66.28 ± 4.67	95.29 ± 1.22
MNIST Pretraining							
MAML	9.80 ± 1.27	9.80 ± 2.03	28.33 ± 21.66	51.12 ± 21.42	66.93 ± 3.13	69.07 ± 4.66	95.73 ± 0.56
Transfer Learning	9.89 ± 1.49	17.32 ± 15.89	21.12 ± 18.70	37.13 ± 24.93	64.91 ± 3.54	63.66 ± 3.86	96.16 ± 0.58
CIARP Pretraining							
MAML	9.83 ± 2.80	17.59 ± 16.62	19.15 ± 18.97	38.57 ± 23.35	66.50 ± 3.05	66.36 ± 3.02	96.00 ± 0.35
Transfer Learning	9.31 ± 1.90	18.30 ± 16.49	25.79 ± 21.30	39.07 ± 24.93	64.26 ± 5.42	66.69 ± 3.15	95.57 ± 1.06
LSA16 Pretraining							
MAML	10.98 ± 3.66	15.16 ± 11.85	28.57 ± 23.10	46.96 ± 19.03	67.18 ± 4.49	64.42 ± 3.58	95.48 ± 1.29
Transfer Learning	8.33 ± 0.00	17.15 ± 15.69	28.93 ± 24.06	46.22 ± 20.08	66.58 ± 5.00	62.89 ± 10.14	95.48 ± 0.30
ProtoNet	**48.93 ± 3.02**	**48.53 ± 1.59**	**48.87 ± 1.60**	46.73 ± 1.14	47.58 ± 1.33	50.36 ± 6.82	47.09 ± 0.10

using transfer learning with a model pretrained on CIFAR10, a general purpose object dataset, outperforms models pretrained on RWTH, which is a handshape dataset.

In future work, we will focus on comparing with other datasets to better understand the relationship between models and dataset complexities for hand-shape recognition. We also see the need to compare with the use of MAML models pretrained with different tasks, combining datasets to achieve it. Finally, we intend to also compare methods that employ unlabelled data for pretraining, and investigate the possibility of merging data sets from different sign languages to augment the sample size, as well as identify the types of data augmentation that lead to an improvement in state-of-the-art models.

Competing interests

The authors have declared that no competing interests exist.

Authors' contribution

FQ and FR designed the study. UJCF and GR performed the experiments. UJCF, GR and FQ wrote the manuscript with support from PDB and input from all authors . WH and LL provided funding for the project. All authors reviewed the manuscript and participated in discussions during the development of the project.

Acknowledgements

We thank NVIDIA for the donation of a Pascal Titan X GPU to support this project.

References

[1] O. Koller, "Quantitative survey of the state of the art in sign language recognition," *CoRR*, vol. abs/2008.09918, 2020.

[2] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, *et al.*, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 16–31, 2019.

[3] A. A. I. Sidig, H. Luqman, and S. A. Mahmoud, "Arabic sign language recognition using vision and hand tracking features with hmm," *International Journal of Intelligent Systems Technologies and Applications*, vol. 18, no. 5, pp. 430–447, 2019.

[4] W. Min, W. Ya, and Z. Xiao-Juan, "An improved adaptation algorithm for signer-independent sign language recognition," *International Journal of Intelligent Systems Technologies and Applications*, vol. 17, no. 4, pp. 427–438, 2018.

[5] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA), pp. 3793–3802, June 2016.

[6] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017.

[7] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *CoRR*, vol. abs/1703.03400, 2017.

[9] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA), pp. 3793–3802, June 2016.

[10] F. Ronchetti, F. Quiroga, L. Lanzarini, and C. Estrebou, "Handshape recognition for argentinian sign language using probsom," *Journal of Computer Science and Technology*, vol. 16, no. 1, pp. 1–5, 2016.

[11] F. Quiroga, R. Antonio, F. Ronchetti, L. C. Lanzarini, and A. Rosete, "A study of convolutional architectures for handshape recognition applied to sign language,"

in *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017).*, 2017.

[12] D. Núñez Fernández and B. Kwolek, "Hand posture recognition using convolutional neural network," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (M. Mendoza and S. Velastín, eds.), (Cham), pp. 441–449, Springer International Publishing, 2018.

[13] A. A. Alani, G. Cosma, A. Taherkhani, and T. M. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," *2018 4th International Conference on Information Management (ICIM)*, pp. 5–12, 2018.

[14] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A real-time hand posture recognition system using deep neural networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, p. 21, 2015.

[15] P. Barros, S. Magg, C. Weber, and S. Wermter, "A multichannel convolutional neural network for hand posture recognition," in *International Conference on Artificial Neural Networks*, pp. 403–410, 09 2014.

[16] S. Ameen and S. Vadera, "A convolutional neural network to classify american sign language fingerspelling from depth and colour images," *Expert Systems*, vol. 34, p. e12197, February 2017.

[17] U. J. Cornejo Fandos, G. G. Rios, F. Ronchetti, F. Quiroga, W. Hasperué, and L. C. Lanzarini, "Recognizing handshapes using small datasets," in *XXV Congreso Argentino de Ciencias de la Computación (CACIC 2019, Universidad Nacional de Río Cuarto)*, 2019.

[18] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018* (V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, eds.), (Cham), pp. 270–279, Springer International Publishing, 2018.

[19] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, (Stockholmsmässan, Stockholm Sweden), pp. 4095–4104, PMLR, 10–15 Jul 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2017.

[22] A. Farhadi, D. Forsyth, and R. White, "Transfer learning in sign language," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 06 2007.

[23] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer learning for semg hand gestures recognition using convolutional neural networks," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1663–1668, 2017.

[24] K. Weiss, T. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, 12 2016.

[25] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," tech. rep., CIFAR, 2009.

[26] Y. LeCun and C. Cortes, "MNIST handwritten digit database," tech. rep., MNIST, 2010.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Citation: F. Quiroga, F. Ronchetti, U. J. Cornejo Fandos, G. G. Ríos, P. Dal Bianco, W. Hasperue and L. Lanzarini. *A comparison of small sample methods for Handshape Recognition.* Journal of Computer Science & Technology, vol. 23, no. 1, pp. 35–44, 2023.
DOI: 10.24215/16666038.23.e03
Received: October 14, 2022 **Accepted:** February 23, 2023.