Prototypical one-shot and k-shot learning on the Omniglot dataset

(with a general application to non-parametric learning via high-dimensional Euclidean embeddings)

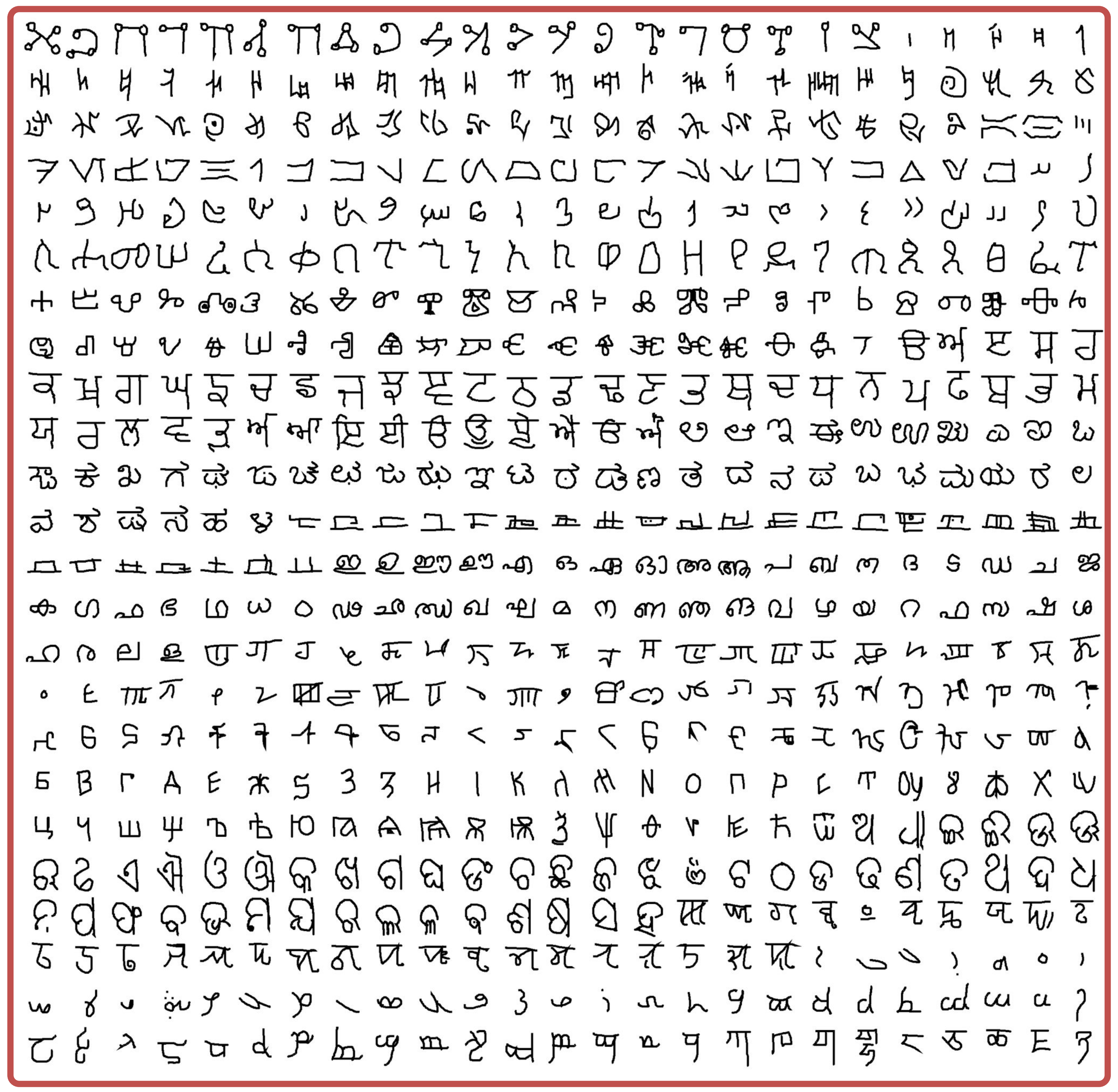
Stanislav Fort, sfort1@stanford.edu, CS 231N, Spring 2017

One-shot learning

- classification on new classes not seen during training
- hallmark of human learning
- extends deep learning to data-poor problems
- non-parametric → avoids overfitting on small datasets
- protypical learning - inspired by cognitive science and

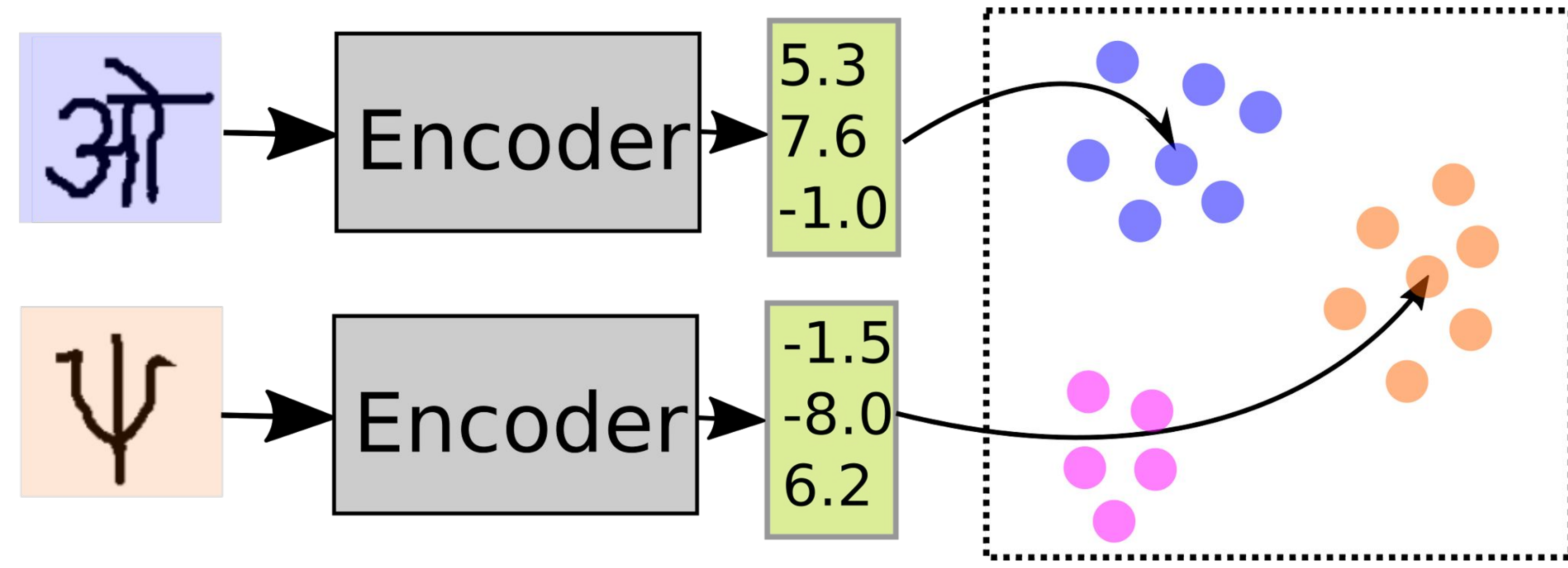
Omniglot and data

- **1623 characters, 50 alphabets, 20 examples of each**
- images 105 x 105 x 1 reduced to 28 x 28 x 1
- data augmentation by rotations of 90° → **6492 characters**
- 4800 characters for training, 1692 for test



Examples of characters from the test part of Omniglot.

Protypical model



Encoder: Image → CNN encoder → Euclidean embedding

Classification and training: (as described in arXiv 1703.05175)

- # of ***support*** images embedded → their mean **= prototype**
- # of ***query*** images embedded
  - their embeddings compared to prototypes → **Euclidean distances**
  - the closest prototype determines the predicted class

Encoders:

[3x3x64 → Batch Norm → ReLU → 2x2 max pool] x4 → 64-dim embed

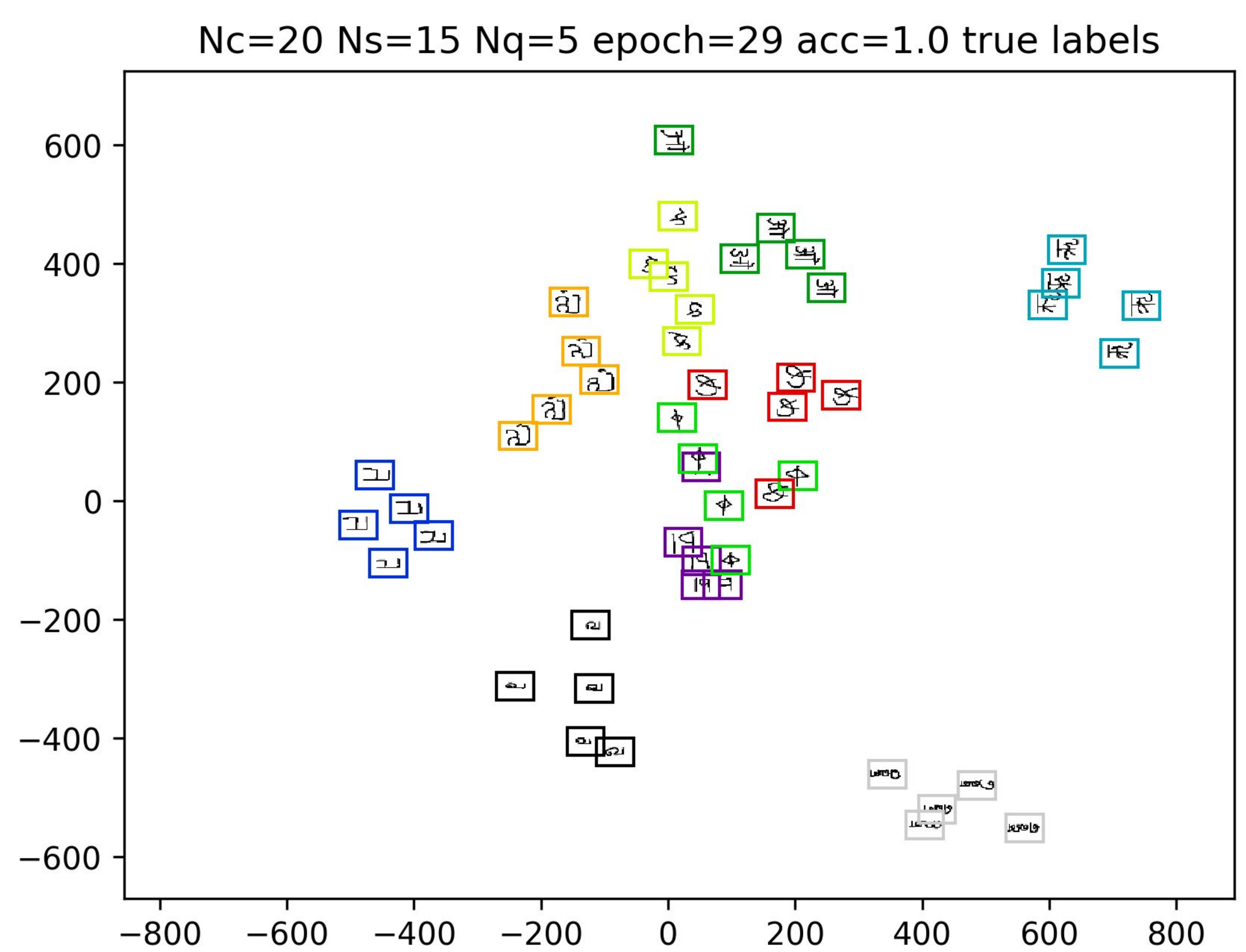
Loss = cross entropy on distances to prototypes

Optimizer = Adam with learning rate 1e-3

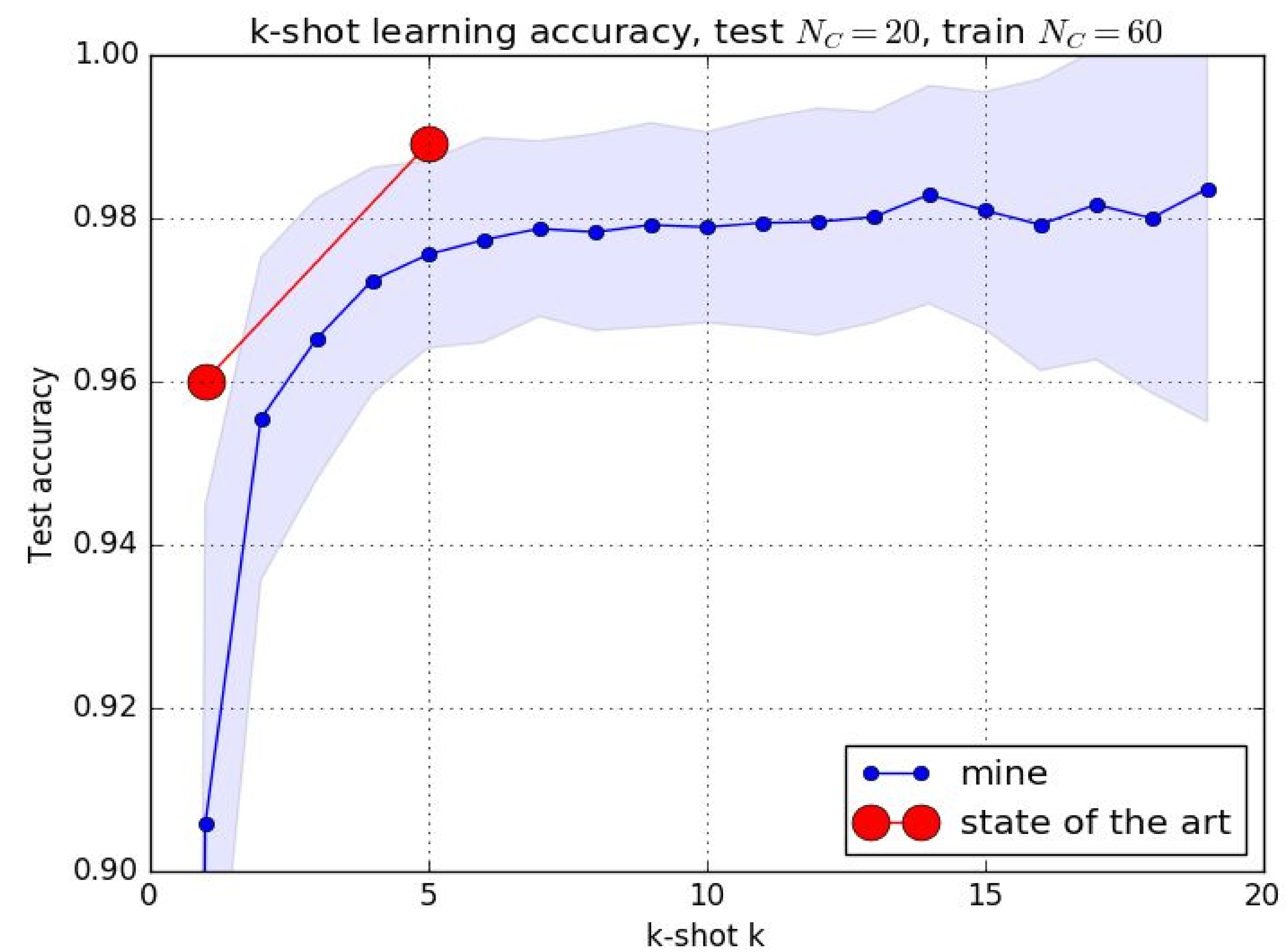
Also experimented with:

- fully-connected → embed (lack spatial awareness)
- CNNs → fully-connected → embed (similar performance)
- embedding sizes, norms (cos, Ln) - Euclidean works the best

Results



k-shot learning performance for a model trained on 15 support points and 60 classes per batch, and tested on 20 classes per batch - **within error of state of the art**



	1-shot	5-shot	19-shot
mine	**90.57 ± 3.90 %**	**97.55 ± 1.10 %**	**98.35 ± 2.86 %**
1703.05175	96.0 %	98.9 %	not done

Conclusion

- prototypical learning works very well on Omniglot
- I was able to reach near state-of-the-art performance on the dataset
- encoder learns useful representation for clustering in the Euclidean embedding
- Euclidean distances work the best and are easily adaptable to new classes
- Will use the method on CIFAR-100 and ImageNet