Unsupervised Face Recognition via Meta-Learning

Dian Huang
Department of Electrical Engineering
Stanford University
dhuang05@stanford.edu

Zhejian Peng
Stanford University
zpeng94@stanford.edu

Abstract

We build an unsupervised face recognition system without using any labeled data in the training process. We use synthetic human faces generated by StyleGAN, which is also trained without labeled data, to train a prototypical network that can identify real human faces. To generate K face images with similar facial features for each of the N classes, we apply rejection sampling to sample N anchor vectors in latent space and K vectors near the anchors in the w-space of StyleGAN. For these near vectors in w-space, we concatenate the portion that impacts the facial feature with other random vectors. The synthesizer then used these concatenated vectors to generate images with similar facial features. During meta-validation and meta-testing, we give K_{test} real human face images for each of the N people to construct an N-way-K-shot task. We experiment with two pre-trained StyleGAN models trained on CelebAMask-HQ and FFHQ dataset and test our method with the CelebA dataset. The FFHQ dataset, though with a different style from CelebA, contains a greater variety of faces. Both models outperform other unsupervised meta-learning methods such as CACTUs, UMTRA[5], and LASIUM[7] even with less K_{test} and achieves comparable accuracy to the supervised ProtoNets with the same set of hyper-parameters. With this setting, our CelebAMask-HQ model achieves a peak accuracy of 72% in the 5-way-5-shot task, 70% in the 5-way-4-shot task, 62% in the 5-way-2-shot task, and 48% in the 5-way-1-shot task. Our FFHQ model achieves a peak accuracy of 86% in the 5-way-5-shot task, 76% in the 5-way-4-shot task, 74% in the 5-way-2-shot task, and 63% in the 5-way-1-shot task. Therefore, we demonstrate that it is possible to achieve a reasonable accuracy in face recognition task without using any labeled data during meta-training. Meanwhile, we study how the difference in distribution between synthetic and real data can cause overfitting. Our experiment with different datasets also shows that the variety of tasks can have more impact on the performance than the similarity tasks between meta-training and meta-testing. [1]

1 Introduction

Face recognition, being widely used in areas such as finance, military, and daily life, has achieved major breakthroughs with the help of deep neural networks. Recent works such as deep face [11] has reached an accuracy of 97.35%. However, these methods require training data that have many faces per person, which could be difficult to collect due to privacy and labor cost. The development of meta-learning has significantly improved the accuracy of few-shot learning, which train neural networks that can adapt to different tasks with only a few samples per class. For example, MAML [1] achieves an accuracy of 85% in the 5-way-5-shot face recognition task. However, although supervised meta-learning reduces the number of samples needed for each class, it still requires a large amount of labeled data during training.

[1] Link to our code github.com/dnmarch/unsupervised_meta_learning_face_recognition/

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Unsupervised meta-learning provides a way to tackle this problem as it does not require any pre-label data during training. Instead, it creates the labeled data without any supervision. There are three main approaches to create the labeled data. 1. *Data augmentation:* [5] Creates more samples of the same class by augmenting the real image in the training data. However, image augmentation requires domain-specific knowledge, and some features cannot be easily modified, especially in the face recognition task. For example, it is difficult to change the face pose in a real image and generate realistic images through image augmentation. 2. *Unsupervised Clustering:* [2] Use unsupervised clustering method such as the k-means to classify the embedding representation of the images. However, k-means cannot control the features that the image is classified base on. For example, k-means may classify the human faces by face poses rather than facial features. 3. *GAN:* [6] Use Generative Adversial Network (GAN) to generate similar faces and assign them the same label. However, it is difficult to control the variation of these synthetic faces. For example, the face it generates may be similar in terms of face pose rather than facial features.

To tackle the problem of face synthesis with GAN, we propose a better sample generation method using StyleGAN [4], which is also trained without labeled data, to train a prototypical network that can identify the ID of real human faces. Our method takes advantage of the style mixing in StyleGAN and generates in-class and out-of-class images by concatenating the output of the non-linear mapping network in StyleGAN. Our method outperforms other unsupervised meta-learning methods such as CACTUs [2], UMTRA [6], and LASIUM [7] even with less K_{test}, the number of samples per class for meta-testing, and achieves comparable accuracy to the supervised ProtoNets[10] with the same set of hyper-parameters. In summary, we made the following contributions.

- Different from the rejection sampling in LASIUM[7], we propose to sample at the latent space but reject the samples based on their pairwise distance at w-space, which is the output of the non-linear mapping network.
- We show that our model achieves competitive accuracy even using the StyleGAN model pre-trained with another human face dataset.
- We demonstrate that our method outperforms other unsupervised meta-learning and is comparable to supervised meta-learning.

2 Related Background

Meta-learning aims to find the correlation of similar tasks through meta-training, so can quickly adapt to new tasks in meta-testing. In face recognition, each task is defined as the identification of new faces out of N people when given K images for each person, but the person IDs of these $N \times K$ images are not known. Unsupervised meta-learning means that the person IDs for these images are not given even during training.

In the absence of labels in meta-training, we can use GAN trained from unlabeled images to generate N classes of human faces with different facial features, and each class has K human faces with similar facial features. One prior approach is called LASIUM[7], as illustrated in figure 1. It first samples N latent z vectors that are far apart from, which is denoted as z_{anchor}, and then sample K vectors around the z_{anchor}, which is denoted as z_{near}. However, if these z_{near} vectors of the same class are too close, the generated faces are too similar, and therefore cannot cover the variation such as different hairstyles, poses of a person in the testing dataset. If they are far apart, the generated faces do not look like the same person. Therefore, it is difficult to generate a great variety of images of the same person by simply controlling the pairwise distance among the vectors in the latent space.

3 Method

Algorithm 1 summarizes our proposed method for generating unsupervised meta-learning tasks using StyleGAN[4] with rejection sampling. Our method is agnostic to the choice of the meta-learning algorithm such as MAML and prototypical network[10]. In this case, we used the prototypical network. We will discuss our task generation method and prototypical network in this section.

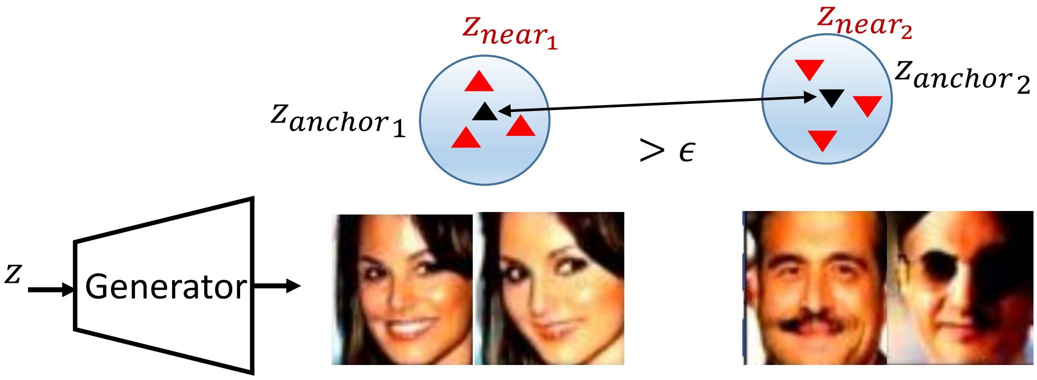
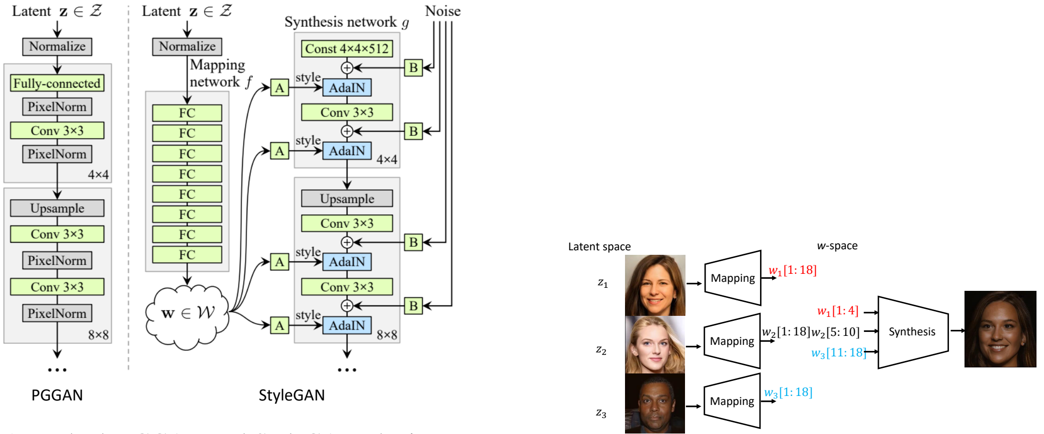


Figure 1: LASIUM[7]: use rejection sampling to sample anchor points with pairwise distance greater than ϵ. The two images on the left are supposed to be the same person, but the variation on facial features is more than that of other features. The images on the right do not look like the same person despite more variation.



(a) For both PGGAN and StyleGAN, the image generated by each synthesizer is upsampled and fed into the next synthesizer during the training process. The PGGAN feeds the latent code through the input layer of the first synthesizer only. In StyleGAN, each synthesizer takes in two vectors from w-space.

(b) Style mixing with StyleGAN: If we have the latent vectors of three images and concatenate different parts of their $w[1{:}18]$ in w-space. The output image has the face pose of the first image, the eye and nose of the second image, and the skin color of the third image.

Figure 2: Architecture of PGGAN and StyleGAN one the left, and style mixing example on the right.

3.1 StyleGAN

Our method takes advantage of the style mixing in the StyleGAN[4] to generate faces with similar facial features. However, it is difficult to train a generator that can synthesize realistic, high-quality, and high-resolution images directly from the latent vector. PGGAN [3] addresses this issue by training multiple synthesizers for different resolutions. It trains the synthesizer to generate low-resolution images first, then adds the synthesizers to generate higher resolution images into the training process. So the model is being progressively trained to generate higher and higher resolution images, as illustrated in figure 2a.

StyleGAN also uses this method to generate high-resolution images, but different from PGGAN, its generator consists of mapping and synthesis. The mapping block transforms latent space z into a higher dimension space formed by 18 vectors. The synthesis block then take them as input vectors for the synthesizers of different resolutions. In this report, we will call this space as w-space, and denote the 18 vectors in this space as $w[1{:}18]$. So $w[1{:}4]$ means the first four vectors, fed in the lowest resolution synthesizer, and $w[5{:}10]$ are the next six vectors, fed in the medium resolution synthesizer. As illustrated in figure 2b, among these 18 vectors, w[1:4] has a greater impact on the coarse-level feature of the generated images, such as face pose, hairstyle. w[5:10] typically affects the facial feature, such as the nose and eye, and w[11:18] mostly affects the fine detail of skin. The intuition behind this is that in low-resolution training, the facial feature does not have much impact on the training process as it is still blurry in low-resolution images. Therefore, the w-vectors fed in the low-resolution synthesizer mostly affects the face shape, pose, and hairstyle. During higher resolution training, as the parameters for face shape, pose, and hairstyle have been trained in the low-resolution synthesizer, it typically forces the generator to generate more realistic facial features to fool the discriminator.

This property of the StyleGAN enables style-mixing of different images, So it is possible to generate human faces with similar facial features but with a great variety of other features by making only small changes to w[5:10] and randomize w[1:4] and w[11:18].

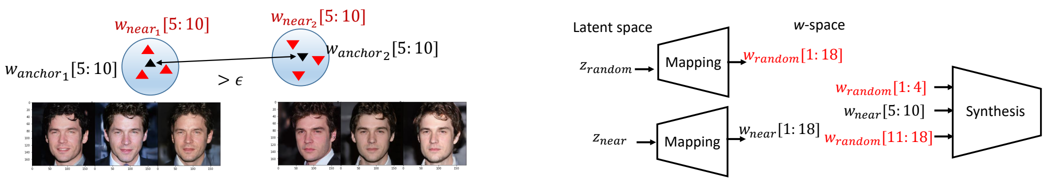
3.2 Rejection Sampling

The goal is to generate N classes of human faces with different facial features, and each class has K human faces with similar facial features. All sampling happens in the latent space. Similar to LASIUM[7], our rejection sampling finds N anchor points that are at a pairwise distance larger than a threshold. However, different from LASIUM, this pairwise distance is defined as the Euclidean distance in the space formed by w[5:10] rather than the latent space. We call these anchor points in the latent space as z_{anchor} and their mapping points in w-space as w_{anchor}. We keep sampling until we find these N anchor points, which generate human faces with different facial features.

Rejection sampling is also applied to find K points near each of the N anchors. We add noise to each z_{anchor} such that $z_{near} = z_{anchor} + \mathcal{N}(0, \sigma^2)$, where $\mathcal{N}(0, \sigma^2)$ denotes the Gaussian noise with a zero mean and a standard deviation of σ. We call the mapping of z_{near} in w-space as w_{near}. Similarly, we keep sampling z_{near} until we find K points that satisfy the condition that the Euclidean distance between w_{anchor}[5:10] and w_{near}[5:10] is smaller than some threshold, so all of them will generate images with similar facial features, as illustrated in figure 3a. However, we discard w_{near}[1:4] and w_{near}[11:18] and randomly sample additional points for these vectors, as shown figure 3b. So these K images will have similar facial features but a great variety of other features such as face pose and skin color.

Algorithm 1: Task generation for proposed unsupervised meta-learning with style mixing

```
Input: Unlabeled dataset $\mathcal{U} = \{x_1, ..., x_i, ...\}$, pre-trained StyleGAN mapping network $\mathcal{M}(z)$
       and synthesis network $\mathcal{S}(w)$
Input: $N$: number of class for this classification task
Input: $Q_{train}$: number of query images in meta-training
Input: $K_{train}$: number of support images in meta-training
Input: $B$: batch size for meta-learning model
$B = \{\}$;
for $i$ in $1$ to $B$ do
    Use rejection sampling to sample $N$ anchor vectors $z_{anchor}$ in the latent space
    compute $w_{anchor}$ using $\mathcal{M}(z)$, then save both to $\mathcal{W}$ and $\mathcal{Z}$ ;
    for each $w_{anchor}, z_{anchor}$ in $\mathcal{W}$ and $\mathcal{Z}$ do
        Sample $K_{train} + Q_{train}$ vectors $z_{near}$ by rejection sampling and compute $w_{near}$ using
         $\mathcal{M}(z)$
        Sample $K_{train} + Q_{train}$ vectors $z_{random}$ and compute $w_{random}$ using $\mathcal{M}(z)$
        $w_{mix} = Concat[w_{random}[1:4], w_{near}[5:10], w_{random}[11:18]]$;
    end
    Generate $N \times (K_{train} + Q_{train})$ images by feeding $w_{mix}$ to synthesis network $\mathcal{S}(w)$;
    Construct task $\mathcal{T}_i$ by adding first $N \times K_{train}$ images to meta-training set and last
     $Q_{train} \times N_{query}$ images to query set;
    $B \leftarrow B \cup \mathcal{T}_i$;
end
return $B$;
```



(a) Rejection sampling

(b) Concat w_{mix}

Figure 3: (**a**) shows the case of $N = 2$ and $K = 3$. We use rejection sampling to find two anchors with Euclidean distance of their w[5:10] greater than a threshold ϵ. All in-class samples are within a small distance from the anchor. The synthetic faces on the left all have a similar eye shape, but can be different in face pose, skin color, and even facial expression. The synthetic faces on the right also have similar facial features but different face poses. (**b**) shows that we only keep w_{near}[5:10] and randomly sample another z_{random} for the rest of the vectors in w-space.

3.3 Prototypical Networks

Finally, we train the prototypical network [10] with these synthetic human faces. The labels for these images are their corresponding anchor point. The prototypical network learns a non-linear mapping of these synthetic images into an embedding space. Through training, the synthetic images with similar facial features surround the same point in this embedding space. We hope that the prototypical network can learn how to cluster a set of real images with similar facial features in this embedding space even though they are different from the synthetic images.

4 Experiment

4.1 Model and Dataset

We compare two pre-trained models of the StyleGAN[4] generator and test our method with the CelebA [9] dataset.

CelebAMask-HQ: The first StylGAN trains on the CelebAMask-HQ [8] dataset, which contains 30,000 unique face images at 1024×1024 resolution. CelebAMask-HQ has substantially fewer images than the CelebA dataset used in meta-testing. Therefore, the images synthesized by this pre-trained model may not cover the diversity in the testing dataset.

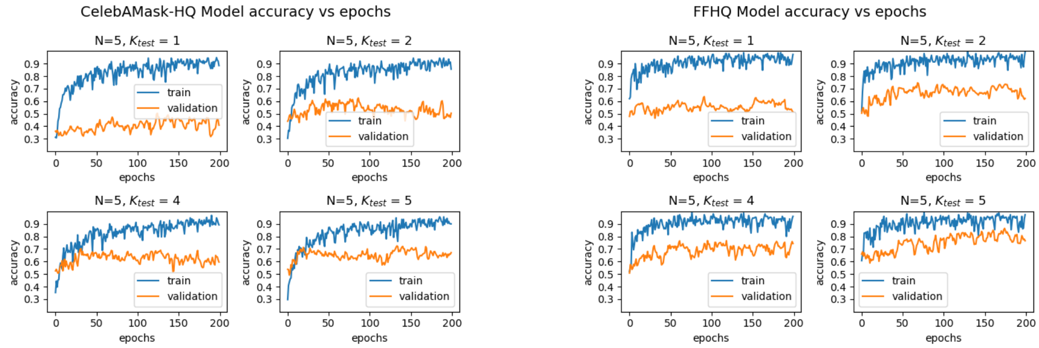
FFHQ: The second StyleGAN trains on FFHQ dataset [4], which offers faces with a lot of variety in terms of age, ethnicity, viewpoint, lighting, and image background. FFHQ contains 70,000 high-quality PNG images at 1024×1024 resolution. However, this dataset comes from different source and has no similar images in CelebA.

CelebA: CelebA contains 10,177 number of person IDs and 202,599 face images. To test our models, we need to have multiple images for each person ID. We find all face images with person IDs existing in both CelebAMast-HQ and CelebA and create a testing dataset by joining them. The testing dataset contains 1274 unique person IDs with each id map to 10 or more face images. Person IDs with less than 10 face images are dropped.

By using different datasets for training and testing, we hope to exam the robustness of our method against the great variety of human faces in real-life applications.

4.2 Setup

In meta-training, we use the pre-trained model of the StyleGAN generator to synthesize N classes of human faces with different facial features. Each class has $K_{train} + Q_{train}$ human faces with similar facial features, where K_{train}, Q_{train} denote the number of samples in support and query set of meta-training accordingly. Therefore, in each epoch, the generator synthesizes $N \times (K_{train} + Q_{train})$ human faces to train the prototypical network. For meta-validation and meta-testing, we randomly sample N person IDs from CelebA dataset, then randomly sample $K_{test} + Q_{test}$ real human faces for each person ID from CelebA, where K_{test}, Q_{test} denote the number of samples in support and



(a) Meta-validation accuracy vs training epochs for the CelebAMask-HQ model with $N = 5$. Note that all training uses $K_{train} = 1$, and the validation may use different K_{test} values. The accuracy drops slightly with more training due to overfitting.

(b) FFHQ model outperforms all other methods even though the FFHQ dataset has a different image style. The generator pre-trained with this dataset can synthesize a greater variety of faces to cover the diversity in the testing dataset.

Figure 4: Model accuracy plots for StyleGAN pre-trained on CelebAMask-HQ and FFHQ.

query set during meta-validation/testing. The prototypical network embeds these $N \times K_{test}$ samples to N support points, then classifies the rest of $N \times Q_{test}$ samples based on their Euclidean distances to each of the support points in the embedding space.

4.3 Result

We set $N = 5$, $K_{train} = 1$, and $Q_{train} = 5$ during meta-training and test our model with different K_{test} values. We also explore other K_{train} value and test with $K_{test} = 5$.

4.3.1 Result on CelebAMask-HQ Pre-trained Model

Using this StyleGAN generator model for faces synthesis achieves a peak accuracy of 72% in the 5-way-5-shot task, 70% in the 5-way-4-shot task, 62% in the 5-way-2-shot task, and 48% in the 5-way-1-shot task. For meta-testing, the Q_{test} is set to be 5 for all cases.

As shown in 4a, except for the case of $K_{test} = 1$, the accuracy could drop after it reaches its peak as the training process goes on. The reason is that CelebAMask-HQ dataset has substantially less images than the testing dataset CelebA, so the distribution of the synthetic data can be very different from the distribution of the CelebA dataset. The prototypical network trained with more epochs may overfit to the distribution of the synthetic human faces, and so fail to generalize to the real human faces. Therefore, for the case that the training data has less variation than the testing dataset, we suggest using early stopping to stop the training process when its training accuracy starts to grow slowly to prevent overfitting of the synthetic faces.

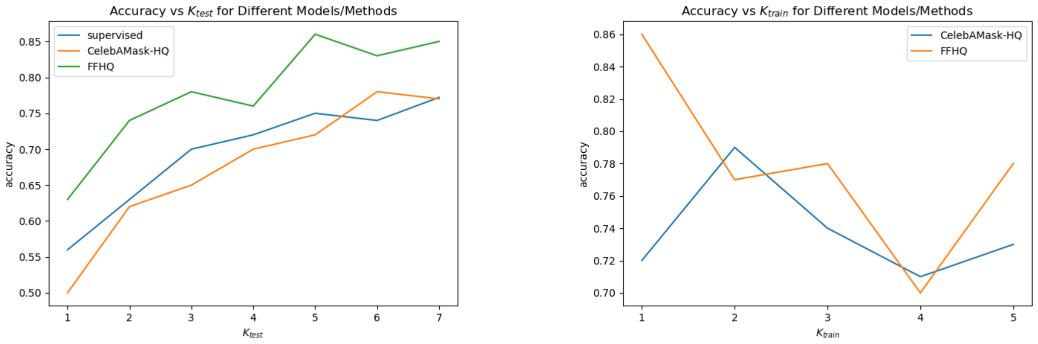
4.3.2 Result on FFHQ Pre-trained Model

Using FFHQ StyleGAN generator model achieves a peak accuracy of 86% in the 5-way-5-shot task, 76% in the 5-way-4-shot task, 74% in the 5-way-2-shot task, and 63% in the 5-way-1-shot task, which is significantly higher than all other methods, as shown in figure 4b.

Due to a greater variety of the dataset, overfitting to the synthetic faces no longer causes a drop in the meta-testing accuracy.

5 Comparison and Discussion

We compared our method with supervised meta-learning, and all approaches use the same hyperparameters. The supervised meta-learning directly use the labeled data in CelebA for training. Figure 5a reveals that despite the training data CelebAMask-HQ has less diversity and much fewer samples than the CelebA used in meta-testing, it still achieves similar results to that of supervised meta-



(a) Meta-testing accuracy over K_{test}

(b) Meta-testing accuracy over K_{train}

Figure 5: (a) Sweep K_{test} with $K_{train} = 1$: FFHQ model outperforms all other methods even though the FFHQ dataset has a different image style. As K_{test} increases, all three methods show better performance. (b) Sweep K_{train} with $K_{test} = 5$: Higher K_{train} tends to force the encoder of the prototypical network to cluster the synthetic faces with similar facial features together better, which can be different from those of the real faces. So the accuracy drops.

Algorithm	$K_{train} = 1$	$K_{train} = 5$	$K_{train} = 15$
CATCUs[2]	41.42%	62.71%	74.18%
UMTRA[5]	39.3%	60.44%	72.41%
LASIUM-RO-GAN-MAML[7]	43.88%	66.98%	78.13%
LASIUM-RO-VAE-MAML[7]	41.25%	58.22%	71.05%
LASIUM-RO-GAN-ProtoNets[7]	44.39%	60.83%	66.66%
LASIUM-RO-VAE-ProtoNets[7]	43.22%	61.12%	68.51%
Supervised ProtoNets*[10]	75%		
CelebAMask-HQ Model*	**72%**	**79%** for $K_{train} = 2$	
FFHQ Model*	**86%**	**78%**	

Table 1: Accuracy results of unsupervised learning on CelebA for different methods. The results are averaged over 1000, 5-way, K_{train}-shot downstream tasks with K_{test} = 5 for the task with * and K_{test} = 15 for other tasks. Our models outperform all other methods that use ProtoNets despite using a smaller K_{test}. Our FFHQ model outperforms all other methods by a large margin.

learning for the cases of $K_{test} > 3$. This result shows that using GAN to generate training data can increase the diversity of the training data to a certain extent. The FFHQ model is far better than any other method, including the supervised meta-learning. Although the image style of the FFHQ dataset and CelebA dataset is different, FFHQ generator can synthesize faces with more variety to capture the diversity in the testing dataset. Therefore, these comparisons show that the variety of meta-training tasks is more important than the similarity between meta-training and meta-testing tasks.

As shown in figure 5a, more samples used in meta-testing leads to higher accuracy. Both of our models outperform other methods even though we are using only $K_{test} = 5$ in meta-testing in contrast with $K_{test} = 15$ used in other approaches, as shown in Table 1.

However, more synthetic samples per class (higher K_{train}) used in meta-training may reduce meta-testing accuracy, as shown in figure 5b. More training samples per class tends to force the encoder of the prototypical network to cluster the synthetic faces together better, which may be different from the real faces.

6 Future Work

In real-world applications, many tasks are often more specific than face recognition, such as eyeglasses detection or face mask detection. These feature-specific data acquisition can be difficult, so learning from unlabeled data can have a high impact on multiple industries. With our unsupervised meta-learning approach, we can construct N-way-K-shot meta-learning tasks by using the StyleGAN generator to synthesize images with and without a specific feature. For the example of eyeglasses detection, we can learn the representation of a few images with eyeglasses in w-space and then generate various faces with eyeglasses by concatenating some portions of the w vectors with other random vectors using our approach described in algorithm 1. With this approach, we may learn a robust and well-performed classifier based on a more specific feature with a small amount of labeled data.

7 Conclusion

We proposed an unsupervised meta-learning algorithm for few-shot face recognition. This algorithm takes advantage of the style mixing property in StyleGAN to generate images for meta-training tasks. Unlike other face recognition algorithms, our approach requires no labeled data and performed comparably with the supervised method. Comparing to UMTRA[5], CATCUs[2], and LASIUM[7], our method outperforms them in CelebA dataset[9] with less K_{test}. Meanwhile, we recommended applying early stopping when the StyleGAN is trained with a dataset that has substantially less diversity than the testing dataset to prevent overfitting. We also noted that the variety of tasks that covers the diversity of testing data can be more important than the similarity of tasks between meta-training and meta-testing. Finally, We addressed the future work direction on unsupervised meta-learning for specific features.

References

[1] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[2] K. Hsu, S. Levine, and C. Finn. Unsupervised learning via meta-learning, 2019.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[4] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019.

[5] S. Khodadadeh, L. Bölöni, and M. Shah. Unsupervised meta-learning for few-shot image and video classification. *CoRR*, abs/1811.11819, 2018.

[6] S. Khodadadeh, L. Bölöni, and M. Shah. Unsupervised meta-learning for few-shot image classification, 2019.

[7] S. Khodadadeh, S. Zehtabian, S. Vahidian, W. Wang, B. Lin, and L. Bölöni. Unsupervised meta-learning through latent-space interpolation in generative models, 2020.

[8] C.-H. Lee, Z. Liu, L. Wu, and P. Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[10] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning, 2017.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.