Supplementary: Multi-Domain Multi-Definition Landmark Localization for Small Datasets

David Ferman[1,2] and Gaurav Bharaj[1]

[1] AI Foundation, USA
[2] UT Austin, USA
davidcferman@gmail.com

A Dataset Details

Standard Benchmark Datasets. See Qian *et al.* [7] for a detailed description of the `WFLW`, `COFW`, and `300W` datasets, and Liu *et al.* [6] for `LaPa`.

1. `WFLW` [11]: 7,500 training faces, 98 landmarks
2. `LaPa` [6]: 18,176 training faces, 106 landmarks
3. `COFW` [1]: 1345 training faces, 29 landmarks
4. `300W` [8]: 3837 training faces, 68 landmarks

Non-Standard Datasets. While the `AnimWeb` [4] and `CariFace` [14] datasets contain larger numbers of images, in this study, for the purpose of evaluating our method's performance for novel domains with small datasets, we only consider a single animal from `AnimWeb`, the Japanese macaque, for its greater visual similarity with human faces, as well as the first 148 images of `CariFace`. Additionally, we utilize a small unlabeled dataset of 150 *in-the-wild* illusory faces [10], called pareidolias. We label the bounding boxes in addition to a 9 landmark definition, following `AnimWeb` [4], and refer to this dataset of PAREeidolias as the `PARE` dataset. We will release the include the GT landmarks and images indices from the dataset used for PARE.

1. `AnimWeb` [4]: 17,520 (80% of 21,900) training faces, 9 landmarks, 334 animal species
2. `ArtFace` [13]: 160 faces, 68 landmarks, 16 artists, 10 per artist
3. `CariFace` [14]: 6,240 (80% of 7,800) training faces, 68 landmarks
4. `PARE` dataset [New]: 150 "faces", 9 landmarks

B Laplacian Log-Likelihood

Following notation introduced in section (3.4) and Kumar *et al.* [5], we formally define the Laplacian log-likelihood as:

$$\mathcal{L}_{lll}(L^i_j, C^i_j, {L_{GT}}^i_j)_k = \frac{1}{2} log|\Sigma^i_{j,k}| + \sqrt{3(L^i_{j,k} - {L_{GT}}^i_{j,k})^T (\Sigma^i_{j,k})^{-1} (L^i_{j,k} - {L_{GT}}^i_{j,k})} \tag{1}$$

where, $\Sigma^i_{j,k}$ is the covariance matrix obtained from the Cholesky factor $C^i_{j,k}$ of the kth landmark of the ith FLSG of the jth dataset.

C 300W Results

We evaluate our method on the `300W` [8] that contains 3,837 training images, and 600 testing images, with a 68 landmark definition. We train our model with two settings: `300W`, and `300W` concurrently trained with `LaPa`. We evaluate our model with inter-ocular normalization, and compare our results with state-of-the-art, Table 1. Here, we note that concurrent training with a larger dataset shows significant performance improvements.

Method	Common	Challenge	Full
PCD-CNN	3.67	7.62	4.44
CPM+SBR	3.28	7.58	4.10
SAN	3.34	6.60	3.98
LAB	2.98	5.19	3.49
DeCaFA	2.93	5.26	3.39
U-Net	2.90	5.15	3.35
HR-Net	2.85	5.15	3.32
LUVLi	2.76	5.16	3.23
AWing	2.72	4.52	3.07
SH-FAN	2.61	**4.13**	2.94
FaRL	2.56	4.45	**2.93**
ADNet	**2.53**	4.58	**2.93**
MDMD Base	2.91	5.12	3.34
MDMD w/LaPa	2.82	4.87	3.22

Table 1. Comparison against SOTA for `300W` [8] on Inter-Ocular NME

D Additional Implementation Details

D.1 Additional Architectural Details

Our final prediction heads which regress the landmark and covariance information from the FLSG tokens each consist of two MLP heads. The covariance information is predicted by regressing the Cholesky factorization of the covariance matrix. Each MLP for landmarks and Cholesky prediction consist of two `relu` separated layers. The (input, output) dimensions for the first layer are `(768, 768//4)` for both head types and `(768//4,` $N_j^i \times 2$`)` and `(768//4,` $N_j^i \times 3$`)` for the second layer of the landmark and Cholesky heads respectively, where N_j^i is the number of landmarks for the ith FLSG and the jth dataset.

D.2 Augmentation Policy

For training our model, we augment rigorously, applying random rotations, blurs, horizontal & vertical waves, cutout, equalization, shear, color jitter, solarization,

auto contrast, sharpness changes, posterization, inversion, scaling and translations, making use of [3] for affine geometric transforms. We adopt two modified versions of Tan *et al.*'s [9] AutoAugment [2] policy, one which adds additional rotations and removes the translation, as we perform our translation augmentation later, and another which removes the geometric augmentations.

D.3 FLSG Indexing Psuedocode Per (3.4)

We present the pseudocode, as mentioned in section (3.4), for handling the FLSG heads and indexing:

```
class FLSGHead:
    def init(flsg_map: List[int]):
        flsg_map = flsg_map
        lm_heads = ModuleList(build_head(2*len(flsg)) for flsg in flsg_map)
        chol_heads = ModuleList(build_head(3*len(flsg)) for flsg in flsg_map)

    def build_head(flsg_dim: int):
        return Sequential(ReLU(), Linear(D, D // 4), ReLU(), Linear(D // 4, flsg_dim))

    def forward(flsg_tokens: Tensor):
        lms = concat([head(flsg_tokens[:, i]) for i, head in enum(lm_heads)])
        chols = concat([head(flsg_tokens[:, i]) for i, head in enum(chol_heads)])
        ids = [id for id_list in flsg_map for id in id_list]
        return lms[:, ids], chols[:, ids]


class MDMDTransformer:
    def init():
        vit_encoder = ViT()
        flsg_maps = get_flsg_definitions() # [[lm_ids] * num_FLSGs] * num_datasets
        definition_agnostic_decoder = Decoder(flsg_maps)
        flsg_heads = ModuleList(FLSGHead(flsg_map) for flsg_map in flsg_maps)

    def forward(images: Tensor, dataset_id: int)
        image_features = vit_encoder(images)
        flsg_tokens = definition_agnostic_decoder(image_features)
        lms, chols = flsg_heads[dataset_id](flsg_tokens)
        return lms, chols
```

FSLG Definitions
We define the facial landmark semantic group definitions which were used for each dataset as follows:

Key:

(a) upper left contour
(b) lower left contour
(c) jaw
(d) lower right contour
(e) upper right contour
(f) left eye
(g) right eye
(h) left brow
(i) right brow
(j) nose
(k) top mouth
(l) bottom mouth

Dataset Definitions

1. `WFLW` [11]:
   (a) (0, 1, 2, 3, 4, 5)
   (b) (6, 7, 8, 9, 10, 11, 12)
   (c) (13, 14, 15, 16, 17, 18, 19)
   (d) (20, 21, 22, 23, 24, 25, 26)
   (e) (27, 28, 29, 30, 31, 32)
   (f) (60, 61, 62, 63, 64, 65, 66, 67, 96)
   (g) (68, 69, 70, 71, 72, 73, 74, 75, 97)
   (h) (33, 34, 35, 36, 37, 38, 39, 40, 41)
   (i) (42, 43, 44, 45, 46, 47, 48, 49, 50)
   (j) (51, 52, 53, 54, 55, 56, 57, 58, 59)
   (k) (77, 78, 79, 80, 81, 89, 90, 91)
   (l) (76, 82, 83, 84, 85, 86, 87, 88, 92, 93, 94, 95)
2. `LaPa` [6]:
   (a) (0, 1, 2, 3, 4, 5)
   (b) (6, 7, 8, 9, 10, 11, 12)
   (c) (13, 14, 15, 16, 17, 18, 19)
   (d) (20, 21, 22, 23, 24, 25, 26)
   (e) (27, 28, 29, 30, 31, 32)
   (f) (66, 67, 68, 69, 70, 71, 72, 73, 74, 104)
   (g) (75, 76, 77, 78, 79, 80, 81, 82, 83, 105)
   (h) (33, 34, 35, 36, 37, 38, 39, 40, 41)
   (i) (42, 43, 44, 45, 46, 47, 48, 49, 50)
   (j) (51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65)
   (k) (85, 86, 87, 88, 89, 97, 98, 99)
   (l) (84, 90, 91, 92, 93, 94, 95, 96, 100, 101, 102, 103)
3. `COFW` [1]:
   (a) -
   (b) -
   (c) (28)
   (d) -
   (e) -
   (f) (8, 10, 12, 14, 16)
   (g) (9, 11, 13, 15, 17)
   (h) (0, 2, 4, 6)
   (i) (1, 3, 5, 7)
   (j) (18, 19, 20, 21)
   (k) (22, 23, 24, 25)
   (l) (26, 27)
4. `300W` [8]:
   (a) (0, 1, 2, 3)
   (b) (4, 5, 6)
   (c) (7, 8, 9)
   (d) (10, 11, 1)
   (e) (13, 14, 15, 16)

(f) (36, 37, 38, 39, 40, 41)
(g) (42, 43, 44, 45, 46, 47)
(h) (17, 18, 19, 20, 21)
(i) (22, 23, 24, 25, 26)
(j) (27, 28, 29, 30, 31, 32, 33, 34, 35)
(k) (48, 49, 50, 51, 52, 53, 54, 60, 61, 62, 63, 64)
(l) (55, 56, 57, 58, 59, 65, 66, 67)

5. `AnimWeb` [4]:
   (a) -
   (b) -
   (c) -
   (d) -
   (e) -
   (f) (0, 1)
   (g) (2, 3)
   (h) -
   (i) -
   (j) (4)
   (k) (5, 6, 7)
   (l) (8)
6. `ArtFace` [13]:
   (a) (0, 1, 2, 3)
   (b) (4, 5, 6)
   (c) (7, 8, 9)
   (d) (10, 11, 12)
   (e) (13, 14, 15, 16)
   (f) (36, 37, 38, 39, 40, 41)
   (g) (42, 43, 44, 45, 46, 47)
   (h) (17, 18, 19, 20, 21)
   (i) (22, 23, 24, 25, 26)
   (j) (27, 28, 29, 30, 31, 32, 33, 34, 35)
   (k) (48, 49, 50, 51, 52, 53, 54, 60, 61, 62, 63, 64)
   (l) (55, 56, 57, 58, 59, 65, 66, 67)
7. `CariFace` [14]:
   (a) (0, 1, 2, 3)
   (b) (4, 5, 6)
   (c) (7, 8, 9)
   (d) (10, 11, 12)
   (e) (13, 14, 15, 16)
   (f) (36, 37, 38, 39, 40, 41)
   (g) (42, 43, 44, 45, 46, 47)
   (h) (17, 18, 19, 20, 21)
   (i) (22, 23, 24, 25, 26)
   (j) (27, 28, 29, 30, 31, 32, 33, 34, 35)
   (k) (48, 49, 50, 51, 52, 53, 54, 60, 61, 62, 63, 64)
   (l) (55, 56, 57, 58, 59, 65, 66, 67)

Grouping	$NME_{ic}(\%)$	$FR_{10\%}$	$AUC_{10\%}$
5 Groups	4.12	3.23	59.43
8 Groups	4.14	2.88	59.36
12 Groups	**4.06**	**2.63**	**60.10**

Table 2. Comparison of FLSG grouping strategies on `WFLW` [11]

8. `PARE` dataset [New]:
   (a) -
   (b) -
   (c) -
   (d) -
   (e) -
   (f) (0, 1)
   (g) (2, 3)
   (h) -
   (i) -
   (j) (4)
   (k) (5, 6, 7)
   (l) (8)

E PARE Dataset

We release the labels for the PARE dataset containing 150 *in-the-wild* illusory face images [10] at the following: https://github.com/davidcferman/pareidolia-landmarks. The images and license information can be found at `https://osf.io/9g4rz/`.

F FLSG Groupings

We experiment with several FSLG grouping strategies, shown in Fig 1. The results from training on the `WFLW` [11] dataset with each grouping strategy are shown in Table 2. For our experiments, we selected the option with 12 FLSG groups, which performed best.

G `ArtFace` [13] Additional Comparisons

We include additional comparisons against `ArtFace`. As previously mentioned, `ArtFace`'s training set is a large set of style transferred images, while the testing set is 160 real paintings. However, our method trains on 112 of these real paintings, and tests on the remaining 48. We include comparisons when using the `ArtFace` checkpoint on our 48 painting testing subset, for a direct comparison. Additionally, we include results with our method, trained on the style transferred images of `ArtFace`. We show the results in Table 3.

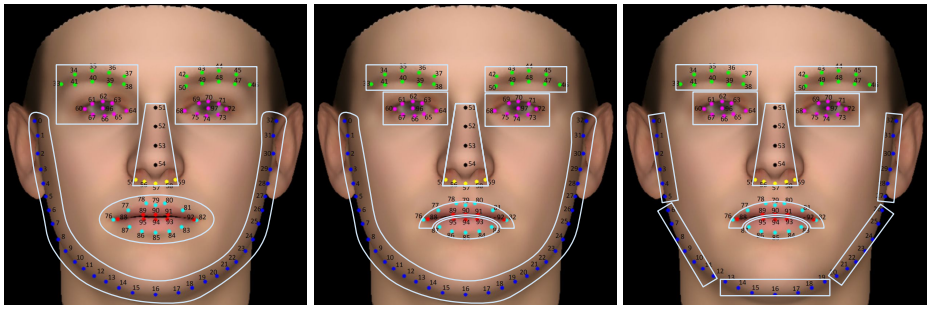

Fig. 1. Facial Landmark Semantic Groupings. Image source: [11]

Method	NME_{ic}(%)	Test Set
Yaniv et al. [13]	4.522	Full Set
MDMD Base (style-transferred images)	3.996	Full Set
Yaniv et al. [13]	4.573	30% subset
MDMD Base	4.46	30% subset
MDMD w/300W	3.72	30% subset

Table 3. Comparison against `ArtFace` [13].

H Backbone Comparisons

We experiment with several backbone variations. While our model uses a pre-trained ViT backbone, we experiment with replacing this backbone with a Resnet-50, as well a Resnet-50 prior to our ViT. Additionally, we train our ViT from scratch for a similar number of epochs as we train our other models. We include results for `COFW` [1] along with backbone parameter counts in Table 4.

I Transfer Learning Comparison

We compare our MDMD method to traditional transfer learning, both for `WFLW`, trained with `LaPa`, as well as `PARE`, trained with `300W`. Our model transfer learns from both the pre-trained backbone encoder and FLSG decoder. We include results in Table 5.

References

1. Burgos-Artizzu, X.P., Perona, P., Dollár, P.: Robust face landmark estimation under occlusion. In: Proceedings of the IEEE international conference on computer vision. pp. 1513–1520 (2013) 1, 4, 7, 8
2. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 113–123 (2019) 3

Backbone	$NME_{ip}(\%)$	$FR_{10\%}$	$AUC_{10\%}$	Parameters
Resnet-50	5.10	.59	49.12	24 M
Resnet-50 + ViT	5.72	2.17	42.92	110 M
ViT (scratch)	13.97	60.2	8.86	86 M
Early Convs [12] + ViT	5.13	1.18	48.92	86 M
ViT	**4.82**	**.59**	**51.84**	86 M

Table 4. Comparison of various backbone strategies on `COFW` [1].

Method	$NME_{ic}(\%)$	$FR_{10\%}$	$AUC_{10\%}$
MDMD `WFLW`w/`LaPa`	**3.97**	2.2	**.6083**
TL `LaPa` then `WFLW`	4.00	**1.94**	.6074
MDMD `PARE`w/`300W`	**8.59**	**22.0**	.2871
TL `300W` then `PARE`	8.69	24.0	**.3004**

Table 5. Comparison of MDMD learning with traditional transfer learning (TL).

3. Jung, A.B.: imgaug. https://github.com/aleju/imgaug (2018), [Online; accessed 13-Mar-2022] 3
4. Khan, M.H., McDonagh, J., Khan, S.H., Shahabuddin, M., Arora, A., Khan, F.S., Shao, L., Tzimiropoulos, G.: Animalweb: A large-scale hierarchical dataset of annotated animal faces. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6937–6946 (2020) 1, 5
5. Kumar, A., Marks, T.K., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X., Feng, C.: Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8236–8246 (2020) 1
6. Liu, Y., Shi, H., Si, Y., Shen, H., Wang, X., Mei, T.: A high-efficiency framework for constructing large-scale face parsing benchmark. arXiv preprint arXiv:1905.04830 (2019) 1, 4
7. Qian, S., Sun, K., Wu, W., Qian, C., Jia, J.: Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10153–10163 (2019) 1
8. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 397–403 (2013) 1, 2, 4
9. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. ArXiv **abs/1905.11946** (2019) 3
10. Wardle, S.G., Paranjape, S., Taubert, J., Baker, C.I.: Illusory faces are more likely to be perceived as male than female. Proceedings of the National Academy of Sciences **119**(5) (2022) 1, 6
11. Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., Zhou, Q.: Look at boundary: A boundary-aware face alignment algorithm. In: CVPR (2018) 1, 4, 6, 7
12. Xiao, T., Dollar, P., Singh, M., Mintun, E., Darrell, T., Girshick, R.: Early convolutions help transformers see better. Advances in Neural Information Processing Systems **34** (2021) 8

13. Yaniv, J., Newman, Y.: The face of art: Landmark detection and geometric style in portraits (2019) 1, 5, 6, 7
14. Zhang, J., Cai, H., Guo, Y., Peng, Z.: Landmark detection and 3d face reconstruction for caricature using a nonlinear parametric model. Graph. Model. **115**, 101103 (2021) 1, 5