Towards Accurate Facial Landmark Detection via Cascaded Transformers

Hui Li*[1], Zidong Guo*[1], Seon-Min Rhee[2], Seungju Han[2], Jae-Joon Han[2]
[1]Samsung R&D Institute China Xi'an (SRCX)
[2]Samsung Advanced Institute of Technology (SAIT), South Korea
hui01.li, zidong.guo, s.rhee, sj75.han, jae-joon.han@samsung.com

Abstract

Accurate facial landmarks are essential prerequisites for many tasks related to human faces. In this paper, an accurate facial landmark detector is proposed based on cascaded transformers. We formulate facial landmark detection as a coordinate regression task such that the model can be trained end-to-end. With self-attention in transformers, our model can inherently exploit the structured relationships between landmarks, which would benefit landmark detection under challenging conditions such as large pose and occlusion. During cascaded refinement, our model is able to extract the most relevant image features around the target landmark for coordinate prediction, based on deformable attention mechanism, thus bringing more accurate alignment. In addition, we propose a novel decoder that refines image features and landmark positions simultaneously. With few parameter increasing, the detection performance improves further. Our model achieves new state-of-the-art performance on several standard facial landmark detection benchmarks, and shows good generalization ability in cross-dataset evaluation.

1. Introduction

Facial landmark detection aims to automatically localize fiducial facial landmark points on human faces. It serves as an essential step for several facial analysis tasks, such as face recognition, facial expression analysis, face frontalization and 3D face reconstruction [29].

Facial landmark detection has received significant improvement in recent years. Existing approaches mainly fall into two categories, i.e., coordinate regression-based methods and heatmap-based methods. Coordinate regression-based methods [7, 31] map the input image to landmark coordinates via fully connected prediction layers. To improve accuracy, coordinate regression is usually cascaded as a coarse-to-fine manner [4, 15] or integrated with heatmap regression module [17, 28]. Heatmap-based methods [23, 27, 34] usually predict heatmaps by fully convolutional networks and then obtain the landmarks according to the peak probability locations on the heatmaps. Since heatmap-based models can preserve the spatial structure of image features, they have better performance than coordinate regression-based models generally.

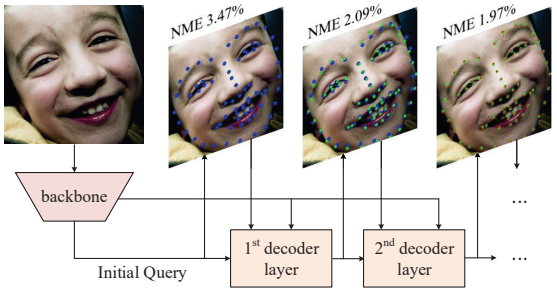

Figure 1. Illustration of our entire framework. The initial query and landmark locations are generated based on the image features, and are continuously updated along with the decoding process.

Although heatmap-based methods have relatively higher detection accuracy, they suffer from three major issues. 1) The required post-processing step is non-differentiable, which disables the end-to-end training. 2) Considering the computational complexity, the resolution of heatmaps is usually lower than that of the input images, resulting in a quantization error inevitably and limiting the performance. 3) They concern more on local texture information and neglect global sensing on face shape, making them vulnerable to large appearance variation such as occlusions.

In contrast, coordinate regression based methods can bypass the aforementioned drawbacks and enable end-to-end model training. However, the fully connected layers destroy the spatial structure of local image features, which deteriorates the localization performance greatly [10].

In this work, we propose a coordinate regression-based model, Deformable Transformer Landmark Detector (DTLD), for accurate facial landmark detection. On one hand, our model avoids the aforementioned shortcomings of heatmap-based methods, and can be well-trained end-

*The first two authors equally contributed to this work. H.Li is the corresponding author.

to-end, without heuristical post-processing. On the other hand, the model is capable of extracting the most relevant features from multi-level feature maps around the target landmark for coordinate prediction, which preserves the local spatial structure and improves the localization accuracy to a large extent. Moreover, our method helps to exploit the underlying relationship among landmarks and incorporate rich structure knowledge, which enables a robust model to tackle various scenarios such as expression or occlusion.

Inspired by the great success of DEtection TRansformer (DETR) in object detection [2, 33], we formulate the landmark detection as a gradually refined N-coordinate prediction task, where N is the number of facial landmarks. Self-attention block is adopted to learn potential structural dependencies. Then multi-scale image feature based deformable attention [33] is employed, where landmark related information is used as the guidance to adaptively extract the most relevant features and refine the coordinates. Different from [2, 33] that define redundant object queries (significantly larger than the typical number of objects in an image) and use bipartite matching to classify objects, here we set the number of queries to be the number of landmarks exactly, which simplifies the training process largely. Instead of using randomly initialized query embedding, we design a more meaningful image-related query-initialization method, which provides coarse landmark locations rather than a fiducial landmark template. Moreover, we explore a parallel decoder where both image features and landmark coordinates are refined simultaneously in the decoding process. It improves the detection performance further. The entire framework is illustrated in Figure 1.

The main contributions can be summarized as follows.

1) We propose a coordinate regression-based facial landmark detector DTLD by cascaded deformable transformers, based on Deformable DETR [33]. DTLD could iteratively capture structural relationships among landmarks and the most relevant visual contextual information to achieve efficient and effective detection.

2) A parallel decoder is further explored to enhance the detection accuracy, with few model parameter increasing.

3) We conduct extensive experiments to analyze the effectiveness of the proposed method, by both quantitative evaluations and qualitative visualizations. Our model contributes to tackle landmark detection under various scenarios. It achieves new state-of-the-art (SOTA) accuracy on several landmark detection benchmarks, and shows good generalization ability in cross-dataset evaluation.

2. Related work

2.1. Facial Landmark Detection

As stated above, the existing approaches on facial landmark detection can be roughly divided into two categories.

Heatmap-based methods usually use high-resolution feature maps for precise localization and achieve encouraging performance. Stacked hourglass network [19] and U-Net [21] are two typical architectures that perform well in heatmap-based methods [3, 5, 24, 27, 34]. Specifically, H-SLE [34] proposes to hierarchically depict holistic and local structures obtained by stacked hourglass network for accurate alignment. LUVLi [13] investigates U-Net for jointly predicting landmark locations, associated uncertainties of these predicted locations and landmark visibilities. HR-Net [23] also shows promising results by connecting and exchanging information via fusing multi-scale image features across multiple branches to obtain high-resolution maps. More recently, PIPNet [10] conducts heatmap and offset predictions simultaneously on low-resolution feature maps, which largely reduces inference time and achieves competitive accuracy.

Coordinate regression-based models are mostly fast, but not accurate enough [10]. In order to improve the accuracy, most algorithms are designed to make predictions in a coarse-to-fine manner through a cascaded structure [4, 7, 18, 32]. For instance, Dapogny *et al*. [4] proposed DeCaFA that uses fully convolutional U-net to preserve the full spatial resolution throughout the cascaded regression for accurate face alignment. LAB [28] was proposed by predicting facial boundary as a geometric constraint via heatmap regression to help landmark coordinate prediction. Most recently, Li *et al*. [15] adopt a cascaded Graph Convolutional Network to dynamically leverage global and local features for precise prediction. Although this method shows superior performance, it relies more on high-resolution feature maps which is computationally expensive.

2.2. Transformers in Vision Tasks

Attention mechanism in transformer [25] is able to encode distant dependencies or heterogeneous interactions, and has shown outstanding performance on lots of computer vision tasks [2, 6, 26, 33]. VIT [6] is the first that employs pure transformer for image classification. PVT [26] integrates pyramid feature maps and spatial property into the model design. DETR [2] and Deformable DETR [33] view object detection as a direct set prediction task and formulate object detection to be trained end-to-end. Yang *et al*. [30] introduced transformer for human pose estimation, and employed attention layers to capture long-range spatial dependencies between human body parts. The model is still heatmap-based. Li *et al*. [14] proposed pose recognition transformer based on DETR. However, it still needs to perform keypoint detection by finding a match between numerous predictions and the ground-truth. In contrast, our work performs exact coordinate regression solely. With a small amount of parameters and computation, our model achieves the highest accuracy on facial landmark detection.

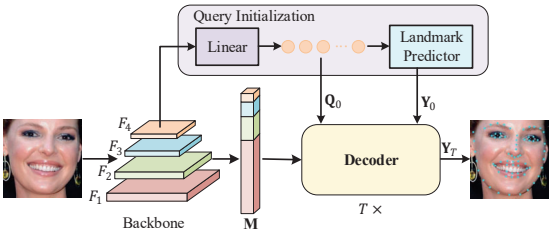

Figure 2. The architecture of our proposed **DTLD**. $\mathbf{Q}_0$ is obtained from $\mathbf{F}_4$, the last layer of backbone features, through a linear projection on spatial dimension, and is further transformed into initial landmark coordinates $\mathbf{Y}_0$, which are adjusted by T decoder layers to get the final positions $\mathbf{Y}_T$.

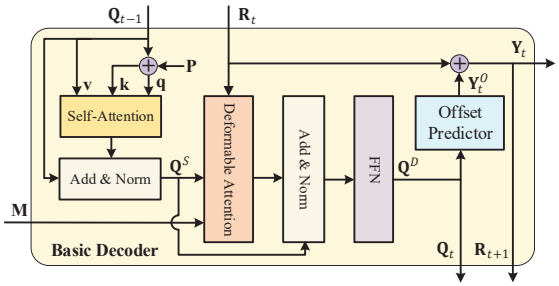

Figure 3. Detailed illustration of the basic decoder. Memory feature **M** and the previous **Q** jointly participate in updating the landmark positions on the basis of previous coordinates **R**. For the first decoder layer, $\mathbf{R}_1$ is the initial landmark location $\mathbf{Y}_0$.

3. Method

The architecture of the proposed DTLD is presented in Figure 2. It is composed of a backbone network for image feature extraction, a query initialization module, and a decoder module for landmark prediction. We adopt a cascaded regression framework where the coordinate offsets are predicted by each decoder layer. The landmark coordinates are refined iteratively during the decoding process. We introduce each part in detail in the following.

3.1. Backbone

The backbone contains an ImageNet [12] pre-trained ResNet-18 [9]. Pyramid features are output, which are denoted as F_1, F_2, F_3, F_4, with down-sampling ratios of $4, 8, 16, 32$ relative to the input image. A 1×1 convolution is followed to project the features into the same number of channels. These features are then flattened and concatenated together, and will be used as the memory feature for decoder, denoted as $\mathbf{M} \in \mathcal{R}^{M\times C}$, where M is the length of the flattened features.

3.2. Query Initialization

A learnable query matrix $\mathbf{Q}$ is defined in [2, 33], which is randomly initialized and updated to represent object related information. In our model, the query matrix $\mathbf{Q}$ is defined to have the size of $N \times C$, where N is the number of landmarks and C is the feature dimension. Rather than random initialization, we extract N features from F_4 by a linear projection on spatial dimension, and use them as the initial query features, *i.e.*

$$\mathbf{Q}_0 = FC(F_4^{\mathsf{T}})^{\mathsf{T}}, \tag{1}$$

We reuse F_4 to denote the flattened feature, $\mathbf{Q}_0 \in \mathcal{R}^{N\times C}$.

The obtained initial query features are expected to be landmark-related. A landmark predictor (another linear projection layer followed by Sigmoid in this paper) is employed to transform them into N landmark coordinates, *i.e.*,

$$\mathbf{Y}_0 = \sigma(FC(\mathbf{Q}_0)), \tag{2}$$

where $\mathbf{Y}_0 \in \mathcal{R}^{N\times 2}$ are the initial landmark coordinates, which will be used as the initial reference points for feature sampling in decoding process as well.

3.3. Decoder Module

The decoder module is composed of T decoder layers. Each layer takes as inputs a query matrix $\mathbf{Q}$, a memory feature $\mathbf{M}$ and reference points $\mathbf{R}$, and outputs landmark coordinate offsets in regards to $\mathbf{R}$. Based on whether updating the memory feature, two types of decoders are explored, *i.e.*, a basic one and a parallel one. They work independently. The former is simple and efficient, setting a strong baseline for landmark detection, while the latter presents a slightly higher detection accuracy.

Basic Decoder. The configuration of the basic decoder layer is illustrated in Figure 3. It mainly consists of a self-attention layer, a deformable attention layer, and an offset predictor.

Specifically, the self-attention layer only adopts the query matrix $\mathbf{Q}$ as input. It learns the structure dependency among landmarks by dense interactions. This information is image-independent intrinsically, where facial attributes like pose and expression will be captured and these attributes have been proven to be important for landmark localization [15]. The self-attention layer takes $\mathbf{Q}^P$,$\mathbf{Q}^P$,$\mathbf{Q}$ as query, key and value separately, and $\mathbf{Q}^P = \mathbf{Q}+\mathbf{P}$, where $\mathbf{P}$ is a learnable position embedding. The output from self-attention layer is denoted as $\mathbf{Q}^S = [\mathbf{q}_1^S, \ldots, \mathbf{q}_N^S]$, where

$$\mathbf{q}_i^S = \sum_{j=1}^{N} \alpha_{ij}(\mathbf{W}_v\mathbf{q}_j), i = 1, \ldots, N, \tag{3}$$

and α_{ij} are self-attention weights calculated by query and key that exploit the connectivity among landmarks. A residual addition and layer normalization are used as those in normal transformer block. The output is renamed as $\mathbf{Q}^S$.

The deformable attention layer takes $\mathbf{Q}^S$ as query and the memory feature $\mathbf{M}$ as value. Instead of calculating the relationship between each element of $\mathbf{Q}^S$ and $\mathbf{M}$, the

deformable attention [33] only attends to a small set of features, obtained by sampling $\mathbf{M}$ according to sampling points. The calculation is formulated as

$$\mathbf{q}_i^D = \sum_{k=1}^{K} \beta_{ik}(\mathbf{W}\mathbf{x}_{ik}), i = 1, \ldots, N, \tag{4}$$

where $\mathbf{x}_{ik}$ are image features sampled from $\mathbf{M}$. K is the entire sampling number. The sampling locations for $\mathbf{q}_i^D$, denoted as $\mathbf{p}_{ik} \in \mathcal{R}^2$, are calculated by $\mathbf{p}_{ik} = \mathbf{r}_i + \delta\mathbf{p}_{ik}$, where $\mathbf{r}_i$ denotes the reference point, which is the i-th landmark coordinate calculated from the previous decoder layer, and $\delta\mathbf{p}_{ik}$ are sampling offsets, obtained via linear projection over the query feature $\mathbf{q}_i^S$. β_{ik} denotes the attention weights over the sampling features, which are calculated by another linear projection over $\mathbf{q}_i^S$, and a softmax operation. The sampling process extracts more related landmark features from multi-level feature maps, which reduces the feature searching area to a large extent and accelerates model convergence. A residual addition, layer normalization and feed-forward network are followed, and the output is re-denoted as $\mathbf{Q}^D = [\mathbf{q}_1^D, \ldots, \mathbf{q}_N^D]$.

The final projection is computed by offset predictor (a 3-layer perceptron in this paper). It takes $\mathbf{Q}^D$ as input and predicts the coordinate offsets $\mathbf{Y}^o$ with regard to the reference points $\mathbf{R}$. The landmark coordinates are then calculated by,

$$\mathbf{Y}_t = \sigma(\mathbf{Y}_t^o + \sigma^{-1}(\mathbf{R}_t)), \tag{5}$$

where t means for the tth decoder layer, $t = 1, ..., T$. $\mathbf{Y}_t^o \in \mathcal{R}^{N\times 2}$ are the predicted coordinate offsets, $\mathbf{R}_t \in R^{N\times 2}$ are coordinates of reference points and $\mathbf{R}_t = \mathbf{Y}_{t-1}$.

Note that the input query matrix $\mathbf{Q}$ is also updated by each decoder layer. $\mathbf{Q} = \mathbf{Q}_0$ for the first decoder layer, and $\mathbf{Q} = \mathbf{Q}_{t-1}^D$ for others. $\mathbf{Q}_{t-1}^D$ is the output from previous deformable attention layer.

Parallel Decoder. DETR and deformable DETR [2, 33] employ several layers of encoder to learn more discriminative image features. DTLD removes the encoder module to save parameters and computational costs. However, experiments show that the encoder is indeed beneficial to detection performance. Instead of inheriting the serial encoder-decoder architecture, we propose a *parallel decoder*, where the memory feature is updated coherently during the decoding process, along with landmark coordinates refinement. The simple variation improves landmark detection accuracy furthermore.

As shown in Figure 4, given the memory feature $\mathbf{M}$, we first add both level embedding and pixel position embedding, denoted together as $\mathbf{P}'$, to indicate which level the feature comes from and the spatial location of the feature in feature maps. The embedding added features, denoted as $\mathbf{M}^P$, are used as the query for updating image feature, *i.e.*,

$$\mathbf{f}_j = \sum_{k=1}^{K} \gamma_{jk}(\mathbf{W}\mathbf{x}_{jk}), j = 1, \ldots, M. \tag{6}$$

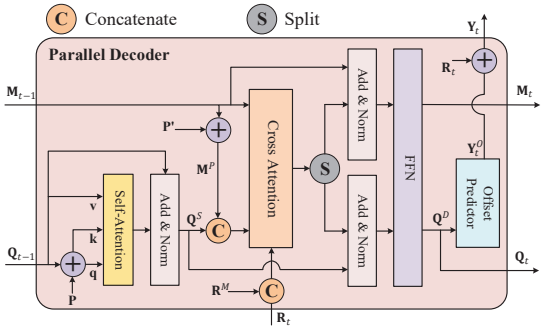

Figure 4. Detailed structure of the proposed parallel decoder. The feature memory $\mathbf{M}$ is also updated in the process, sharing the parameters and operations of cross attention and FFN with query $\mathbf{Q}$.

$\mathbf{f}_j$ are updated image features, $\mathbf{x}_{jk}$ are sampled features from $\mathbf{M}$ according to sampling location $\mathbf{p}_{jk} = \mathbf{r}_j^M + \delta\mathbf{p}_{jk}$. Similarly, $\delta\mathbf{p}_{jk}$ and γ_{jk}, which denote the sampling offsets and attention weights, are computed by linear projection over $\mathbf{M}^P$. The reference points $\mathbf{r}_j^M \in [0, 1]^2$ are normalized coordinates of memory feature on each feature map.

In the parallel decoder, we concatenate $\mathbf{M}^P$ and $\mathbf{Q}^S$ as the overall query features, concatenate $\mathbf{r}_j^M, j = 1, \ldots, M$ and $\mathbf{r}_i, i = 1, \ldots, N$ as the reference points, and update both image features and landmark query features simultaneously according to Eq 6 and Eq 4. The layer parameters are shared except that we use separate layer normalizations for image and landmark query. It results in only $1.2K$ more parameters compared to the basic decoder counterpart. The updated image features will be used as the memory feature next, and the updated landmark query features will be used to calculate the offsets $\mathbf{Y}^o$.

3.4. Training Target

We simply use $\mathcal{L}_1$ loss between the predicted landmark coordinates and the ground-truth to train the model, *i.e.*,

$$\mathcal{L} = \sum_{t=0}^{T} \left\| \mathbf{Y}_t - \hat{\mathbf{Y}} \right\|, \tag{7}$$

where $\mathbf{Y}_0$ is computed by Eq 2, and $\mathbf{Y}_t, t = 1, \ldots, T$ are from Eq 5. $\hat{\mathbf{Y}}$ denotes the ground-truth coordinates.

4. Experiments

In this section, we perform extensive experiments to verify the effectiveness of the proposed method. All the experiments are conducted on an NVIDIA v100 GPU. The models are implemented by PyTorch.

4.1. Datasets

We conduct experiments on a number of popular 2D face landmark detection datasets, including 300W, WFLW, COFW and AFLW. 300W [22] is collected from five facial

Method	Year	Backbone	Pre-Trained	300W (NME)			COFW (NME)	AFLW (NME)	WFLW-Full	
				Full	Common	Challenge			(NME)	($FR_{10\%}$)
LAB [28]	2018	Hourglass	-	3.49	2.98	5.19	5.58	1.85	5.27	7.56
Wing [7]	2018	ResNet-50	Y	–	–	–	5.07	1.47	4.99	6.00
ODN [31]	2019	**ResNet-18**	Y	4.17	3.56	6.67	5.30	1.63	–	–
HGs [17]	2019	Hourglass	-	4.02	3.45	6.38	–	1.60	–	–
DeCaFa [4]	2019	Cascaded U-Net	-	3.39	2.93	5.26	–	–	4.62	4.84
DAG [15]	2020	HRNet-W18	Y	3.04	2.62	4.77	–	–	4.21	3.04
HRNet [23]	2019	HRNet-W18	Y	3.32	2.87	5.15	3.45	1.57	4.60	4.64
AWing [27]	2019	Hourglass	N	3.07	2.72	4.52	–	1.53	4.36	2.84
AVS [20]	2019	ITN-CPM	N	3.86	3.21	6.46	–	–	4.39	4.08
ADA [3]	2020	Hourglass	-	3.50	**2.41**	5.68	–	–	–	–
LUVLi [13]	2020	DU-Net	N	3.23	2.76	5.16	–	1.39	4.37	3.12
PIPNet-18 [10]	2020	**ResNet-18**	Y	3.36	2.91	5.18	3.31	1.48	4.57	–
PIPNet-101 [10]	2020	ResNet-101	Y	3.19	2.78	4.89	3.08	1.42	4.31	–
DTLD-s	2021	**ResNet-18**	N	3.04	2.67	4.56	3.18	1.39	4.14	3.44
DTLD	2021	**ResNet-18**	Y	**2.96**	*2.59*	*4.50*	*3.04*	*1.38*	*4.08*	*2.76*
DTLD+	2021	**ResNet-18**	Y	**2.96**	2.60	**4.48**	**3.02**	**1.37**	**4.05**	**2.68**

Table 1. Comparison with SOTA methods on landmark detection accuracy. We report NME (%) on 300W, COFW and AFLW. On WFLW, both NME (%) and FR (%) at the threshold of 10% are reported. Our method achieved the best accuracy on most datasets by simply using ResNet-18 as the backbone, and the second best on 300W-Common subset. DTLD uses the basic decoder, while DTLD+ adopts the parallel decoder. DTLD-s has all parameters trained from scratch. The top methods are coordinate regression-based while the middle ones are heatmap-based.

datasets, consisting of 3148 training images and 689 test images. The test dataset is further divided into 2 subsets, *i.e.*, common set with 554 images and challenging set with 135 images. Each image is annotated with 68 landmarks.

WFLW [28] is collected from WIDER Face, which includes large variations in pose, expression and occlusion. Each face is originally annotated by 98 landmarks, and re-annotated by 68 landmarks in [10]. There are $7,500$ images for training and $2,500$ for test. The test set is further divided into 6 subsets for different scenarios.

COFW [1] contains 1345 training images and 507 test images under different occlusion conditions. Each image is annotated by 29 landmarks, and we also use 68 landmarks re-annotated by [8] for the cross-domain setting.

AFLW [11] contains 20000 images for training and 4386 images for test, providing 19 landmarks for each face.

CelebA [16] is a large-scale attributes dataset with 202,599 face images in the wild. We only use the images without annotation for training in Section 4.6.

4.2. Implementation Details

For all datasets, the faces are cropped according to the provided bounding boxes firstly, and then resized to 256×256. In order to retain more context information, the bounding boxes on 300W and WFLW are enlarged by 10% and 20%, respectively, following previous work [10]. Data augmentation is adopted involving translation, horizontal flipping, rotation, occlusion and blurring. The whole model is trained end-to-end by Adam optimizer for 120 epochs in total. The learning rate is set to 1e-4 initially and then reduced to 1e-5 at 100th epoch, where the learning rate for backbone is 10 times smaller than the above. By default, we use 3 decoder layers, with a feature dimension of 256 and 8 heads. For each query, we sample 4 features for each head from each level of the feature maps. The configuration will be analyzed in ablation study. We train the model on 1 v100 GPU with a batch size of 16. The reported results are averaged over three runs.

4.3. Evaluation Metric

we adopt the most widely used metric, normalized mean error (NME), to evaluate our model for fair comparison with previous work. It is calculated by,

$$\text{NME}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2}{D}. \tag{8}$$

We employ the prediction from last decoder layer for evaluation. D is a normalization distance, and we use inter-ocular distance for 300W, WFLW, COFW, and image size for AFLW, following common practice. Failure rate (FR) is also reported which refers to the percentage of failed examples whose NMEs are larger than a certain threshold.

4.4. Comparison with the SOTA

As presented in Table 1, we firstly compare our model with SOTA methods on landmark detection accuracy using the four benchmarks. DTLD is the model with basic decoder, while DTLD-s has all model parameters trained from scratch. DTLD+ is equipped with the parallel decoder. The models are trained and tested separately, with the default configuration.

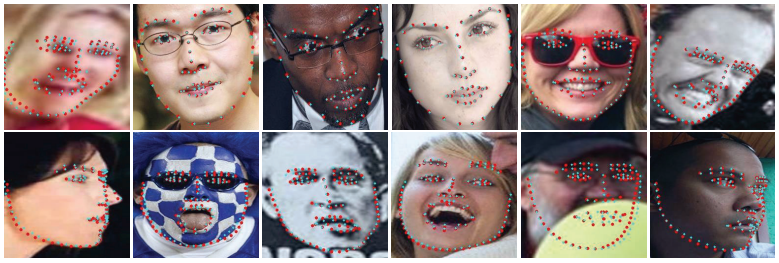

Figure 5. Visualization of typical landmark detection results. Red denotes the ground truth, and cyan represents our predictions. Our model is able to detect landmarks accurately in various scenarios, such as blur, makeup, expression, occlusion, or even with big pose.

Method	Year	Backbone	NME(%)	Param.(M)	GFLOPs	FPS(GPU)
HRNet [23]	2019	HRNet-W18	4.60	**9.7**	4.8	11.7
AWing [27]	2019	Hourglass	4.36	25.1	26.7	24.2
DeCaFa [4]	2019	Cascaded U-Net	4.62	*10*	–	32
LUVLi [13]	2020	DU-Net	4.37	–	–	58.8
PIPNet-18 [10]	2020	**ResNet-18**	4.57	12.0	**2.4**	**200**
PIPNet-101 [10]	2020	ResNet-101	4.31	45.7	10.5	56
DTLD	2021	**ResNet-18**	*4.08*	13.3	*2.5*	*100*
DTLD+	2021	**ResNet-18**	**4.05**	13.3	*2.5*	78

Table 2. Comparison with other methods on Parameter size, GFLOPs and FPS. Our method achieves the highest accuracy with a small amount of GFLOPs and parameters. The FPS is lower than PIPNet-18, which leaves for future improving.

The results show that our models consistently outperform all the other methods on all test datasets with a simple backbone. To be specific, our DTLD achieves the NME of 2.96%, 3.04% and 1.38% on 300W-Full, COFW and AFLW respectively. In addition, with the NME threshold of 8%, the failure rates are 0.29%, 0.20% and 0.25% separately. On WFLW-Full which contains various scenarios, DTLD obtains NME of 4.08%, leading to a relative decrease of 3.09% compared to the second best (4.21% NME), and 10.7% relative to PIPNet-18 (4.57% NME), the previous best method using the same backbone. The failure rates are 2.76% at the threshold of 10% and 6.44% at the threshold of 8%. Comprehensive results on each WFLW subset are shown in supplementary.

Our model also benefits from the ImageNet pre-trained backbone. Without pre-training (referring to DTLD-s), NMEs increase a lot, but are still smaller than SOTA models with similar model size. The use of the parallel decoder (DTLD+) improves the detection accuracy further, leading to NME of 4.05% on WFLW and 3.02% on COFW, averagely 0.02% lower than that obtained by DTLD.

Next, we compare the model size and running speed of our models with others. As presented in Table 2, DTLD has 13.3M parameters and only 2.5 GFLOPs, but achieves very competitive accuracy. The running speed is lower than PIPNet-18 because of the multiple refining process, but is still faster than others. DTLD+ achieves relatively higher accuracy at the sacrifice of running speed.

Some landmark detection results by DTLD are visualized in Figure 5. Our model can accurately predict landmarks in the tough scenes for faces with blur, large posture changes, rich expressions, and partial occlusion.

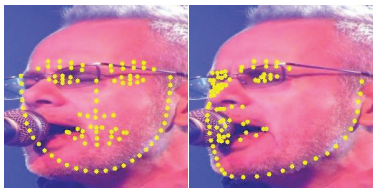

Figure 6. Visualization of the effect of our proposed query initialization. Fiducial landmark positions are produced by randomly initialized $\mathbf{Q}_0$ (left), while our proposed query initialization method sets up good starting points (right).

4.5. Ablation Studies on DTLD

We conduct a series of ablation studies to analyze each part of the proposed model. The ablation experiments are performed on WFLW-Full as it includes comprehensive scenarios.

Effect of $\mathbf{Q}_0$. In DTLD, we use a well-calculated $\mathbf{Q}_0$ as the initial query features and calculate the initial reference points based on $\mathbf{Q}_0$. Here, we perform experiments with a randomly initialized learnt positional encodings as $\mathbf{Q}_0$, as that used in [2, 33]. Experimental result in Table 3 shows a performance drop of 0.11%NME on WFLW. We also visualize the effect in Figure 6. As can be seen, our $\mathbf{Q}_0$ will lead to image related initial reference points, instead of a fiducial landmark template.

Effect of self-attention. Self-attention is employed in decoder layer so as to exploit the structural knowledge among landmark positions. Without self-attention, NME of DTLD

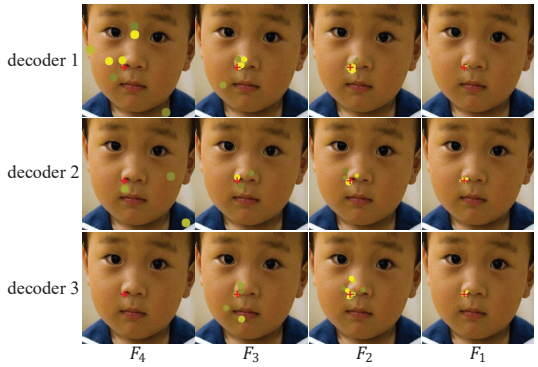

Figure 7. Visualizations of deformable attention on pyramid backbone features. The red cross denotes the ground-truth, while others dots show the sampling points with attention weights expressed by colors. The brighter the point, the greater the weight. We combine the sampling points from all heads for each feature map. Sampling points with attention weights lower than 0.5 are omitted.

Backbone	$\mathbf{Q}_0$	Self-Attn	NME(%)
R18	FC	✔	4.08
R18	Random Init.	✔	4.19
R18	FC	✗	4.17
R50	FC	✔	4.07
R101	FC	✔	4.07

Table 3. Ablation study on DTLD, with backbone, Q initial strategy and self-attention analyzed. *R18 / 50 / 101* represent ResNet-18 / 50 / 101 backbone pre-trained by ImageNet . *FC* is our initialization compared with randomly initialized *Random Init*.

reduces 0.09% (4.08% to 4.17%) as indicated in Table 3.

Effect of backbone. We experiment with different backbones as shown in Table 3. However, the performance gain is not obvious (only 0.01%NME improve) when using deeper backbone like ResNet-50 and ResNet-101, which means DTLD is not sensitive to much deeper backbone.

Effect of deformable attention. We also visualize the multi-scale deformable attention as presented in Figure 7. The visualization shows that the deformable module can extract most related image features around the landmark point for coordinate prediction. Moreover, the first decoder layer attends more on the rear feature maps like F3 and F4 that tend to high level global information, while the last decoder layer attends more on the frontal feature maps like F1 and F2 that are apt to capture low level local features for coordinate fine tuning.

Effect of model hyper-parameters. Here we conduct experiments with different model hyper-parameters on DTLD, including the feature dimension C used in decoder layers, the number of sampling points K used in deformable attention, and the head number used in all attention layers. As shown in Table 4, the higher the feature dimension, the better the performance, but 256 seems to be enough for feature encoding. In our default configuration, for each query feature, we sample 4 points from each feature level for each head. We then test other numbers such as $2, 6$. However, the change has little impact on the final accuracy, indicating that our model can adaptively decouple the most critical information from redundant features. We also run models with different head number in both self and deformable attention. More heads benefit final performance. We visualize the deformable attention for each head in Supplementary. It illustrates intuitively that different heads will pay attention to different directions of image features.

# Feature Dimension	# Sampling Points	# Attention Head	NME (%)
256	4	8	4.08
64	4	8	4.47
128	4	8	4.29
512	4	8	4.07
256	**2**	8	4.11
256	**6**	8	4.09
256	4	**4**	4.16
256	4	**16**	4.07

Table 4. Ablation on DTLD model hyper-parameters including *feature dimension*, *sampling points*, and *attention heads*. The effect of *sampling points* is tiny, while that of the others is large.

Effect of decoder. Encoder layers are commonly adopted to further encode the image features, *e.g.*, [2, 14, 33]. Here we also conduct experiments by adding encoder in DTLD as in deformable DETR [33] and varying the number of layers in both encoder and decoder. Experimental results in Table 5 show that the added encoder or decoder layers indeed contribute to reduce NME furthermore, even achieving NME smaller than 4% on WFLW. Results also show that the adding of decoder layers has relatively larger effect on accuracy than that of encoder layers. However, the added layer brings more parameters (0.5M for one encoder layer and 0.6M for one decoder layer) and degrades the speed.

To improve the accuracy without increasing model size, we propose the parallel decoder, where the image features are encoded along with the decoding process. By sharing the deformable attention layers, the model size is almost the same as that without encoder layers (the little parameter increase comes from separate layer normalization), but NME further decreases. When using similar number of parameters, DTLD+ always obtains higher accuracy than DTLD counterparts. When using decoder layers ≥ 3, DTLD+ gets a higher accuracy compared to DTLD at similar speed. It should be noted that when we use 1 parallel decoder layer, the model exactly becomes DTLD with 1 decoder layer and 0 encoder. The lower speed is caused by image feature updating which is not used anymore. However, it may provide a chance of inferring occluded face part based on the features so as to improve model performance further. We leave it as a future work.

Moreover, we attempt to remove the backbone and com-

	# Encoder Layer	# Decoder Layer					
		1	2	3	4	5	6
DTLD	0	4.417 / 12.1 / 165	4.187 / 12.7 / 123	4.076 / 13.3 / 100	4.068 / 14.0 / 82	4.044 / 14.6 / 69	4.064 / 15.3 / 60
	1	4.369 / 12.6 / 105	4.178 / 13.2 / 86	4.066 / 13.9 / 71	4.026 / 14.5 / 64	4.050 / 15.1 / 56	4.051 / 15.7 / 51
	2	4.327 / 13.1 / 81	4.133 / 13.7 / 69	4.047 / 14.4 / 61	4.015 / 15.0 / 54	4.012 / 15.6 / 45	4.000 / 16.2 / 43
	3	4.244 / 13.6 / 62	4.114 / 14.2 / 54	4.028 / 14.8 / 52	4.006 / 15.5 / 42	3.980 / 16.1 / 35	3.978 / 16.7 / 33
	4	4.235 / 14.1 / 53	4.079 / 14.7 / 46	3.999 / 15.3 / 42	3.968 / 16.0 / 34	3.972 / 16.6 / 25	3.974 / 17.2 / 22
DTLD+	0	4.417 / 12.1 / 115	4.185 / 12.7 / 94	4.054 / 13.3 / 78	4.022 / 14.0 / 69	4.016 / 14.6 / 63	3.996 / 15.3 / 55

Table 5. Experimental results on WFLW by using varying encoder and decoder layers. More encoder or decoder layer contributes to higher performance. The last line shows the effect of our proposed parallel decoder. With similar parameters, DTLD+ achieves slightly higher accuracies. The results are demonstrated by NME(%) / Model Parameter Size (M) / FPS (on V100 GPU).

Methods	Test Data		
	300W	COFW68	WFLW68
LAB [28]	3.49	4.62	–
ODN [31]	4.17	5.30	–
AVS w/SAN [20]	3.86	4.43	–
DAG [15]	3.04	4.22	–
PIPNet(ST) [10]	3.36	4.55	8.09
PIPNet(UDA) [10]	3.35(-0.3%)	4.34(-4.6%)	7.45(-7.9%)
DTLD (ST)	3.07	4.42	7.23
DTLD (UDA)	**3.03(-1.3%)**	**4.14(-6.3%)**	**6.39(-11.6%)**

Table 6. Cross-dataset evaluation and comparison with others. *ST* means supervised training only on 300W training data, but test on others. *UDA* means unsupervised domain adaption by utilizing COFW and WFLW training images without annotation used.

Methods	Unlabeled Data	300W	WFLW
PIPNet [10]	-	3.36	–
	CelebA	3.27 (-2.7%)	–
DTLD	-	3.07	4.08
	CelebA	**2.94 (-4.2%)**	**3.89 (-4.7%)**

Table 7. Boost our model by using unlabeled images from other domain. Our model shows better scalability, which can be improved more by using unlabeled images. Note that it is the enlarged bounding boxes used in 300W that cause NME of 3.07%, larger than 2.96% presented in Table 1.

pute the pyramid features simply by image dividing and patch embedding as performed in [26]. The pyramid embeddings are fed into DTLD+ directly for feature encoding and landmark prediction. With 6 layers and only 6M parameters, our model achieves NME of 4.27% on WFLW.

4.6. Cross-dataset Evaluation

To verify the robustness and generalization ability of our model, we conduct cross-dataset evaluation on COFW and WFLW testsets, using DTLD trained on 300W training data. To maintain distribution consistency between different datasets, we follow the practice in [10], enlarging the provided bounding boxes of 300W, COFW68, and WFLW68 by 30%, 30% and 20% respectively. Experimental results in Table 6 indicate the robustness of our model cross datasets.

In addition, to analyze the model scalability, we perform an unsupervised domain adaption (UDA). More precisely, we apply the classic self-training strategy and re-train the model using COFW and WFLW training images, without landmark annotation employed. The model trained on 300W is used as a teacher model to reason the pseudo-labels for unlabeled data. They are then combined with the original labeled data and re-train the model. After 3 times of re-training, we achieve NME of 4.14% on COFW68 and 6.39% on WFLW68, new SOTA accuracies on both testsets. Compared to PIPNet, the UDA improvement is more obvious, which demonstrates the good scalability of our method.

Motivated by the good scalability, we attempt to promote the model additionally by leveraging the numerous unlabeled face images from CelebA. With the same self-training paradigm, it is found that the detection accuracy can be further improved on 300W and WFLW-Full testsets. As indicated in Table 7, although the unlabeled images are from a different domain compared with the test datasets, our model can still learn from them and leads to even more accurate landmark prediction. It finally achieves NME of 2.94% on 300W and 3.89% on WFLW.

5. Conclusion

In this paper, we propose an effective and efficient facial landmark detection network DTLD based on cascaded transformer. It directly regresses landmark coordinates and thus can be trained end-to-end. The use of self-attention and deformable attention in DTLD enables structure relationship exploring and more related image feature extracting. The simple query initialization sets up a better start point for the following refinement. Moreover, we propose a parallel decoder that refines image features and landmark positions simultaneously, improving the detection performance with few parameter increasing. Our model achieves new SOTA performance on several standard landmark detection benchmarks, surpassing the other advanced approaches. The running speed is a limitation of current work. Knowledge Distillation based methods may be exploited in the future so as to reduce the cascaded refinement steps and accelerate detection process.

References

[1] Xavier P. Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *ICCV*, pages 1513–1520, 2013. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2, 3, 4, 6, 7

[3] Prashanth Chandran, Derek Bradley, Markus Gross, and Thabo Beeler. Attention-driven cropping for very high resolution facial landmark detection. In *CVPR*, pages 5861–5870, 2020. 2, 5

[4] Arnaud Dapogny, Kevin Bailly, and Matthieu Cord. Decafa: Deep convolutional cascade for face alignment in the wild. In *ICCV*, pages 6893–6901, 2019. 1, 2, 5, 6

[5] Xuanyi Dong and Yi Yang. Teacher supervises students how to learn from partially labeled images for facial landmark detection. In *ICCV*, pages 783–792, 2019. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[7] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR*, pages 2235–2245, 2018. 1, 2, 5

[8] Golnaz Ghiasi and Charless C Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 2385–2392, 2014. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[10] Haibo Jin, Shengcai Liao, and Ling Shao. Pixel-in-pixel net: Towards efficient facial landmark detection in the wild. *IJCV*, pages 1–21, 2021. 1, 2, 5, 6, 8

[11] Martin Köstinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshops*, pages 2144–2151, 2011. 5

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012. 3

[13] Abhinav Kumar, Tim K Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *CVPR*, pages 8236–8246, 2020. 2, 5, 6

[14] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *CVPR*, pages 1944–1953, 2021. 2, 7

[15] Weijian Li, Yuhang Lu, Kang Zheng, Haofu Liao, Chihung Lin, Jiebo Luo, Chi-Tung Cheng, Jing Xiao, Le Lu, Chang-Fu Kuo, et al. Structured landmark detection via topology-adapting deep graph learning. In *ECCV*, pages 266–283. Springer, 2020. 1, 2, 3, 5, 8

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 5

[17] Zhiwei Liu, Xiangyu Zhu, Guosheng Hu, Haiyun Guo, Ming Tang, Zhen Lei, Neil M. Robertson, and Jinqiao Wang. Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In *CVPR*, pages 5861–5870, 2020. 1, 5

[18] Jiangjing Lv, Xiaohu Shao, Junliang Xing, Cheng Cheng, and Xi Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *CVPR*, pages 3317–3326, 2017. 2

[19] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016. 2

[20] Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia. Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation'. In *ICCV*, 2019. 5, 8

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 2

[22] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshops*, pages 397–403, 2013. 4

[23] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 1, 2, 5, 6

[24] Zhiqiang Tang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *ECCV*, pages 339–354, 2018. 2

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2

[26] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 2, 8

[27] Xinyao Wang, Liefeng Bo, and Li Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6971–6981, 2019. 1, 2, 5, 6

[28] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, pages 2129–2138, 2018. 1, 2, 5, 8

[29] Yue Wu and Qiang Ji. Facial landmark detection: A literature survey. *IJCV*, 127(2):115–142, 2019. 1

[30] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *ICCV*, pages 11802–11812, 2021. 2

[31] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq. Robust facial landmark detection via occlusion-

adaptive deep networks. In *CVPR*, pages 3486–3496, 2019. 1, 5, 8

[32] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, pages 4998–5006, 2015. 2

[33] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 2, 3, 4, 6, 7

[34] Xu Zou, Sheng Zhong, Luxin Yan, Xiangyun Zhao, Jiahuan Zhou, and Ying Wu. Learning robust facial landmark detection via hierarchical structured ensemble. In *ICCV*, pages 141–150, 2019. 1, 2