# LOTR: Face Landmark Localization Using Localization Transformer

**UKRIT WATCHAREERUETAI**[1], **BENJAPHAN SOMMANA**[1], **SANJANA JAIN**[1],
**PAVIT NOINONGYAO**[1,2], **ANKUSH GANGULY**[1], **AUBIN SAMACOITS**[1],
**SAMUEL W. F. EARP**[1,2], **AND NAKARIN SRITRAKOOL**[3]

[1]Sertis Vision Laboratory, Bangkok 10110, Thailand
[2]QIS Capital, Bangkok 10110, Thailand
[3]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Pathum Wan, Bangkok 10330, Thailand

Corresponding author: Ukrit Watchareeruetai (uwatc@sertiscorp.com)

**ABSTRACT** This paper presents a novel Transformer-based facial landmark localization network named Localization Transformer (LOTR). The proposed framework is a direct coordinate regression approach leveraging a Transformer network to better utilize the spatial information in a feature map. An LOTR model consists of three main modules: 1) a visual backbone that converts an input image into a feature map, 2) a Transformer module that improves the feature representation from the visual backbone, and 3) a landmark prediction head that directly predicts landmark coordinates from the Transformer's representation. Given cropped-and-aligned face images, the proposed LOTR can be trained end-to-end without requiring any post-processing steps. This paper also introduces a loss function named smooth-Wing loss, which addresses the gradient discontinuity of the Wing loss, leading to better convergence than standard loss functions such as L1, L2, and Wing loss. Experimental results on the JD landmark dataset provided by the First Grand Challenge of 106-Point Facial Landmark Localization indicate the superiority of LOTR over the existing methods on the leaderboard and two recent heatmap-based approaches. On the WFLW dataset, the proposed LOTR framework demonstrates promising results compared with several state-of-the-art methods. Additionally, we report an improvement in the performance of state-of-the-art face recognition systems when using our proposed LOTRs for face alignment.

**INDEX TERMS** Artificial neural networks, computer vision, deep learning, face recognition, image processing, machine learning.

## I. INTRODUCTION

Landmark localization focuses on estimating the position of each predefined key point in an image. For face landmark localization, these key points represent different attributes of a human face, e.g., the contours of the face, eyes, nose, mouth, and eyebrows. Over the past decade, face recognition systems have leveraged these landmarks for alignment, making face landmark localization an intrinsic part of these systems [1]–[9]. Apart from face alignment (e.g., [10], [11]), face landmark localization also aids in solving problems like face animation [12], 3D face reconstruction [13]–[15], synthesized face detection [16], emotion classification [17], [18], and facial action unit detection [19]. Although facial landmark localization is a substantial area of research in computer vision, given its wide range of applications, it is a challenging task owing to its dependency on variations in face pose, illumination, and occlusion [1], [20].

Since its formulation, various statistical approaches have been proposed, such as Active Shape Model (ASM) [21] and Active Appearance Model (AAM) [22], to solve the face landmark localization problem. These models take in prior information about a face (e.g., the face shape or texture) and subsequently fine-tune the model parameters from the provided face image. In addition, research has also been done in training patch-based detectors and component detectors to predict each landmark on local patches and anatomical components on a face image, respectively [23]–[27]. However, due to the lack of global contextual information, the landmark configurations from these approaches are constrained on the face shape.

Since the early 2010s, different variants of Convolutional Neural Networks (CNNs) have been developed, as an alternative to the aforementioned approaches, due to their

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.
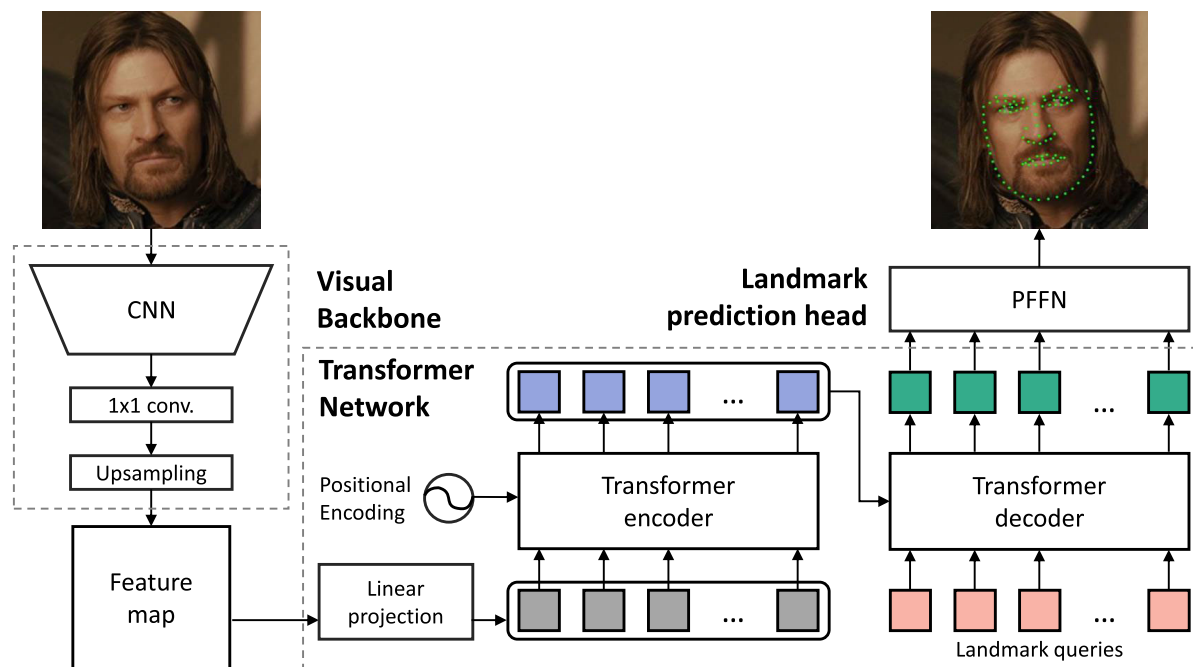
**FIGURE 1.** The overview of localization transformer (LOTR). It consists of three main modules: 1) a visual backbone, 2) a transformer network, and 3) a landmark prediction head.

ability to extract contextual information from an image. Two approaches, namely coordinate regression and heatmap regression, have been widely adopted with a CNN variant as a backbone.

In coordinate regression, fully-connected layers are added at the end of a CNN to predict each landmark's coordinates. Notable works include [28], [29], and [30] which proposed multi-level cascaded CNNs to localize facial landmarks. Combining coordinate regression with multi-level cascaded CNNs has been instrumental in predicting the facial landmarks. In these frameworks, the early levels aim to learn about a rough estimate of the landmarks, while the deeper levels seek to fine-tune the prediction. The main caveat of these approaches is the high architectural complexity, thereby increasing inference time. To address this complexity issue, [31] and [32] utilized multi-task learning to obtain lighter yet more robust models. However, most regression-based approaches suffer from spatial information loss due to the compression of feature maps before the fully-connected layers [1].

Recently, heatmap-based approaches (e.g., [1], [33]–[35]) have extensively been used for face landmark localization as they better utilize spatial information to boost the performance compared with coordinate regression methods. These methods predict spatial probability maps wherein each pixel is associated with the likelihood of the presence of a landmark location. Additionally, these approaches lead to better convergence than coordinate regression techniques and achieve state-of-the-art performance on several benchmark datasets. However, these approaches usually rely on a computationally

intense post-processing step to convert heatmaps into predicted landmarks, resulting in an increase in the inference time [1]. Therefore, tackling this issue is the primary motivation behind this paper.

Although the coordinate regression methods mentioned so far may suffer from the issue of spatial information loss, they offer an end-to-end solution at a lower computational complexity than heatmap-based approaches. In this work, we re-investigate the direct coordinate regression approach for facial landmark localization but exploit a more sophisticated neural network architecture, i.e., the Transformers [36], to address the issue of spatial information loss. A Transformer network is a sequence transduction model comprising an encoder-decoder architecture that utilizes attention mechanisms. With the scaling successes in natural language processing (NLP) achieved by Transformers, researchers have developed different variations of the Transformer framework for computer vision tasks. One such variant is Detection Transformer (DETR) [37] which performs objection detection. Inspired by DETR, we propose a Transformer-based facial landmark localization network named Localization Transformer (LOTR). As shown in Fig. 1, LOTR consists of three main modules: 1) a visual backbone, 2) a Transformer network, and 3) a landmark prediction head. Firstly, LOTR adopts a CNN with optional upsampling layers to convert an input image into a feature map and reshapes it to a sequence of tokens, each representing a pixel in the feature map. The Transformer module then accepts this feature sequence and a fixed-length landmark queries as input, and produces a sequence of tokens as output. Finally, the prediction head,

which is a Position-wise Feed-Forward Network (PFFN), transforms each token into its corresponding landmark's coordinates. Given cropped-and-aligned face images as input, the proposed LOTR can be trained end-to-end without requiring any post-processing step.

The key contributions of this research are summarized as follows:

- We propose a Transformer-based landmark localization network named Localization Transformer (LOTR). Unlike the heatmap-based approaches, LOTR does not require any additional post-processing or heatmap representation, reducing computational complexity and yielding a more efficient network. To the best of our knowledge, this is the first research investigating the use of Transformers in the direct regression of landmarks.
- We demonstrate that the proposed LOTR framework detects facial landmarks accurately. Experimental results indicate the superiority of the proposed LOTR over other algorithms on the leaderboard of the First JD-landmark localization challenge and two recent heatmap-based methods [1], [34]. On another benchmark, WFLW dataset [38], the results show the proposed LOTR method is comparable with several state-of-the-art methods.
- We further analyze the model size and inference time of the different variants of the LOTR framework. Compared with a heatmap-based method [1] and a CNN-based direct coordinate regression method, the proposed LOTR outperforms both of these methods in terms of prediction accuracy, model size, and computational complexity.
- We also investigate the effect of standard loss functions on model training and propose a modified loss function, namely smooth-Wing loss, which addresses gradient discontinuity and training stability issues in an existing loss function called the Wing loss [39]. Experimental results show improved performance for the LOTR models trained with the proposed smooth-Wing loss.
- We report an improvement in state-of-the-art face recognition performance on several benchmark datasets, such as CFP-FP [40], CPLFW [41], IJB-B [42], and IJB-C [43], using five naive landmarks extracted from the predictions of our proposed LOTR models for face alignment.

The remaining of this paper is organized as follows: Section II provides more detail of related work. Section III presents the proposed method, i.e., LOTR. Section IV explains how experiments were setup and discusses the results. Section V concludes the paper.

## II. RELATED WORK
### A. DIRECT COORDINATE REGRESSION
As discussed in Section I, direct coordinate regression was widely adopted to solve landmark localization problems in the early research period. Regression-based

approaches [28], [30]–[32], [39], [44], [45] typically use a CNN along with a dense layer at the end to predict the landmark locations. Usually, the choice of loss for coordinate regression is either mean-absolute error (L1 loss) or mean-squared error (L2 loss). Recently, Feng *et al.* [39] introduced a new loss function called Wing loss for robust facial landmark localization using regression, discussed in Section II-E. Furthermore, Dong *et al.* [45] trained a facial landmark localization model with style-aggregated images from a generative adversarial module along with the original images to increase the robustness of the variance of image styles.

Despite the advancement in regression-based techniques for landmark localization, spatial information loss remains one major drawback for this approach. This research addresses this limitation by inserting a Transformer network between a CNN backbone and the fully-connected layers to preserve spatial information by leveraging a positional encoding, discussed in Section III.

### B. HEATMAP REGRESSION
As an alternative to coordinate regression methods, heatmap regression approaches tackle the spatial information loss issue by generating spatial probability maps. A spatial probability map is a heatmap with pixel values corresponding to the probability of a landmark being in a certain location. Recent heatmap regression approaches (e.g., [33], [35], [38], [46], [47]) have demonstrated state-of-the-art performance on facial landmark localization.

A common practice for generating a ground-truth heatmap for each landmark is to compute a probability map (e.g., [48], [49]) by fitting a bivariate Gaussian function with an offset relative to the landmark, which is defined as:

$$G(x, y) = \exp\left[ -\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2} \right], \quad (1)$$

where $(x, y)$ is any location on the heatmap and $(x_0, y_0)$ represents the ground-truth coordinates of a landmark. A simple approach to obtain a predicted landmark location from a heatmap during inference is using the argmax operation; however, the heatmap's resolution becomes a limiting factor. Recent approaches have proposed efficient ways to obtain sub-pixel localization [50], [51]. For instance, Zhang *et al.* [51] proposed to fit a heatmap with the following formulation to obtain a predicted landmark location:

$$G(\mathbf{x}, \mu, \Sigma) = \frac{\exp(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu))}{2\pi |\Sigma|^{\frac{1}{2}}}, \quad (2)$$

where $\mathbf{x} = (x, y)$ is any location on the heatmap, $\mu$ represents an estimated location of the landmark, and $\Sigma$ is the co-variance matrix between the $x$ and $y$ coordinates.

Earp *et al.* [1] exploited this sub-pixel inference with intermittent shuffling of upsampling layers and a bag of tricks to achieve the second rank on the JD-landmark-2 validation set [52]. While successfully preserving spatial information

and achieving state-of-the-art performance on facial landmark localization, heatmap regression methods suffer from high computational post-processing complexity in converting heatmaps into landmark coordinates.

Xiong *et al.* [34] proposed a vectorization approach in which each set of vector labels represents a ground-truth landmark's position using a quasi-Gaussian distribution. This proposed distribution is a Gaussian density function enhancing the distribution peak with an additional constant $\theta$ while being bounded by a threshold $3\sigma$. This method also converts a predicted heatmap into a set of vectors that encode spatial information. Although this vectorization technique leads to a reduction in post-processing complexity, like other heatmap regression methods, this approach also hinders achieving an end-to-end pipeline due to a post-processing step during inference.

### C. TRANSFORMER-BASED HEATMAP APPROACH

Lan *et al.* [53] proposed an approach that combines a Transformer encoder with heatmap regression. They mainly focused on reducing the quantization error that occurs from down-sampling operations. The authors named the proposed architecture Heatmap in Heatmap (HIH), which takes advantage of two heatmap categories: integer and decimal heatmaps. An integer heatmap is a probability map extracted from a CNN backbone, e.g., Stacked Hourglass Network [48], which provides a rough estimate of a landmark's location. On the other hand, a decimal heatmap gives a more fine-grained offset prediction. For each landmark location, they fitted a decimal heatmap with a bivariate Gaussian density function as in (1). During inference, the final landmark locations, in both integer and decimal heatmaps, are computed from the maximum probability coordinates. The authors conducted experiments comparing two different architectures, namely a CNN and a Transformer encoder in combination with a CNN backbone. The proposed method outperforms the state-of-the-art algorithms (e.g., [38], [39], [47], [54]) on WFLW-full [38] and COFW [55] datasets. They also reported that the HIH approach with a CNN yields more precise landmarks than a Transformer encoder.

Even though Lan *et al.* [53] are the first who employ a Transformer encoder with heatmap regression to tackle facial localization problems, the main caveats of the heatmap regression approach, namely the post-processing complexity and the lack of an end-to-end pipeline, remain unaddressed.

### D. DETR

Carion *et al.* [37] proposed a Transformer [36] framework for end-to-end object detection. Their proposed Detection Transformer (DETR) framework achieves comparative performance to Faster R-CNN [56] on the COCO dataset [57]. The DETR framework performs the task of object detection by combining a CNN architecture with a Transformer. It exploits a pre-trained CNN backbone to extract a low-resolution feature map from an input image. This feature map is then converted into a sequence and fed to an encoder model,

consisting of a Multi-Head Attention (MHA) module and a PFFN. The MHA module aims to find the relationship between input sequence tokens with each head computing "attention" by linearly projecting each token into query, key, and value vectors. Let $\mathbf{Q} \in \mathbb{R}^{N_q \times D}$ denote a query sequence consisting of $N_q$ tokens of dimension $D$, while $\mathbf{K}$ and $\mathbf{V} \in \mathbb{R}^{N_{kv} \times D}$ denote a key and value sequences of length $N_{kv}$, respectively. The MHA is defined as follows:

$$\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{concat}(\mathbf{H}_1, \ldots, \mathbf{H}_M)\mathbf{W}_i^o, \tag{3}$$

$$\mathbf{H}_i = \text{attention}(\mathbf{Q}\mathbf{W}_i^q, \mathbf{K}\mathbf{W}_i^k, \mathbf{V}\mathbf{W}_i^v), \tag{4}$$

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D'}})\mathbf{V}, \tag{5}$$

where $M$ is the number of heads; $\mathbf{W}_i^q, \mathbf{W}_i^k, \mathbf{W}_i^v \in \mathbb{R}^{D \times D'}$, and $\mathbf{W}_i^o \in \mathbb{R}^{D \times D}$ are learnable projection matrices; and $D' = D/M$. Fixed positional encoding, which encodes the spatial positions of the learned features, is added to the input of each attention layer [37]. The embeddings generated by the encoder are then fed to the decoder network, which also uses MHA mechanisms [37]. The decoder model decodes the embeddings in parallel at each decoder layer. The decoder output is then fed into a fixed number of PFFNs, also known as the prediction heads, to generate a set of class predictions and predicted bounding boxes. The DETR model predicts all objects at once, computed in parallel. The authors trained the model end-to-end using Hungarian loss to compare sets, performing bipartite matching between predicted and ground-truth objects.

Inspired by DETR, we propose the Localization Transformer (LOTR) to predict landmarks for the facial landmark localization task. The main differences between the two are: 1) LOTR does not have a class prediction head since all predicted landmarks belong to single object class, and 2) LOTR does not require the Hungarian loss since the number of landmarks is fixed.

### E. WING LOSS

Several loss functions have commonly been used to train landmark localization models, including L1 loss ($L_1(x) = |x|$), and L2 loss ($L_2(x) = \frac{1}{2}x^2$), and the smooth-L1 loss [58], which is defined as:

$$\text{smooth-}L_1(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{6}$$

Feng *et al.* [39] reported another loss function—Wing loss—which is superior to other loss functions for landmark localization tasks. The key idea in the Wing loss is to force the model to pay more attention to small errors to improve the accuracy of the predicted landmarks. When prediction error is larger than a positive threshold, the Wing loss behaves like L1; otherwise, a logarithm function is used to compute
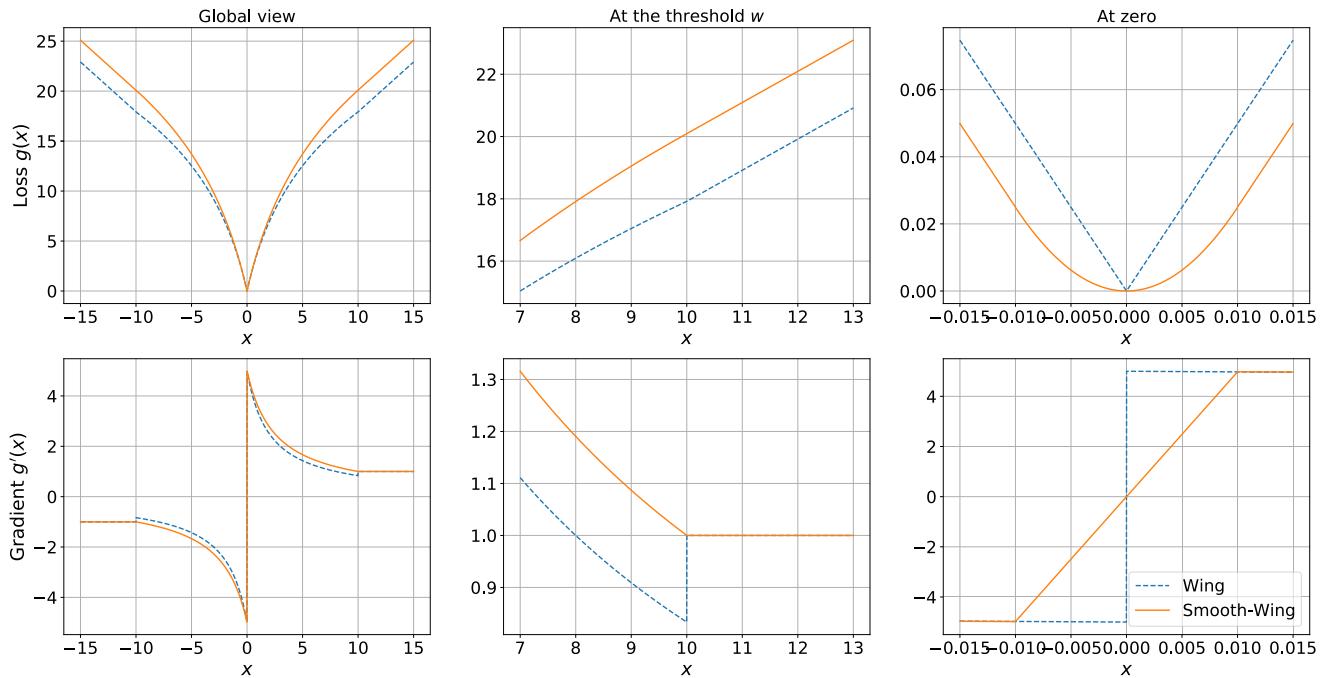
**FIGURE 2.** Comparison of Wing loss and smooth-Wing loss (top) and their gradient (bottom) in the global view (left), at the outer threshold _w_ (middle), and at _x_ equals zero (right). The parameters are set as follows: $w = 10$, $\epsilon = 2$, and only for smooth-Wing, $t = 0.01$. For the Wing loss (blue dashed lines), the gradient changes abruptly at the points $|x| = w$ (bottom-middle) and at $x = 0$ (bottom-right). On the other hand, the proposed smooth-Wing loss (orange solid lines) is designed to eliminate these gradient discontinuities.

the loss. In [39], the Wing loss is defined as follows:

$$\text{wing}(x) = \begin{cases} w \ln(1 + \dfrac{|x|}{\epsilon}) & \text{if } |x| < w \\ |x| - c & \text{otherwise.} \end{cases} \quad (7)$$

where $w$ is the threshold, $\epsilon$ is a parameter controlling the steepness of the logarithm part, and $c = w - w \ln(1 + w/\epsilon)$.

However, as shown in Fig. 2, the Wing loss produces the discontinuity of the gradient at the threshold $w$, as pointed out in [54], and at zero error, which might affect the stability of training. In this work, we propose a modified Wing loss that is smooth everywhere and investigate its effectiveness.

## III. PROPOSED METHODS

### A. LOCALIZATION TRANSFORMER (LOTR)

This section explains, in detail, the proposed Transformer-based landmark localization network, named Localization Transformer (LOTR). Since LOTR is a coordinate regression approach that directly maps an input image into a set of predicted landmarks, it requires neither heatmap representation nor any post-processing step. Fig. 1 illustrates an overview of the architecture of the proposed LOTR framework. An LOTR model consists of three main modules, which include 1) a visual backbone, 2) a Transformer network, and 3) a landmark prediction head.

The visual backbone takes an RGB image as input with the aim to capture context and produce a feature map as output. In this work, we exploit a pre-trained CNN such as MobileNetV2 [59], ResNet50 [60], or HRNet [61] to compute a feature map. We apply $1 \times 1$ convolution to reduce the channel dimension of the feature map. Since the resolution of the feature map generated from the CNN backbone might be very low, e.g., $6 \times 6$ pixels for a $192 \times 192$ input image, we optionally increase the resolution by using upsampling layers such as deconvolution.

We then utilize a Transformer network [36] to enrich the feature representations while maintaining the global information in the feature map. As shown in Fig. 1, the Transformer module is composed of a Transformer encoder and a Transformer decoder. Since Transformers were designed to process sequential data [36], we convert the feature map $\mathbf{F} \in \mathbb{R}^{W \times H \times C}$, obtained from the visual backbone, into a sequence of tokens $\mathbf{X}^0 \in \mathbb{R}^{WH \times D}$. As shown in (8), we use a $1 \times 1$ convolution layer to reduce the channel dimension $C$ of each pixel in $\mathbf{F}$ to a smaller dimension $D \leq C$, followed by reshaping into a sequence of tokens:

$$\mathbf{X}^0 = \text{reshape}(\text{conv}1 \times 1(\mathbf{F})). \quad (8)$$

A Transformer encoder, i.e., a stack of $L$ encoder layers, receives this sequence of tokens as input. Each encoder layer consists of two sublayers: 1) Multi-Head Self Attention (MHSA) and 2) PFFN. Both these sublayers have residual connections and layer normalization applied to them. As mentioned in Section II-D, the MHSA, which is a special type of MHA that establishes the relationship between tokens in the input sequence by computing the attention by linearly

projecting each token into query, key, and value vectors and subsequently using the query and key vectors to calculate the attention weight applied to the value vectors. The output from the MHSA sublayer (the same size as its input) is then fed into PFFN to further transform the input sequence's representation. Similar to DETR [37], 2D-positional encoding is added only to the query and key in each encoder layer. The *l*-th encoder layer is defined as follows:

$$\mathbf{X}^l = \text{enc}^l(\mathbf{X}^{l-1}, \mathbf{P}), \quad (9)$$

$$\text{enc}^l(\mathbf{X}, \mathbf{P}) = \text{enc}_2^l(\text{enc}_1^l(\mathbf{X}, \mathbf{P})), \quad (10)$$

$$\text{enc}_1^l(\mathbf{X}, \mathbf{P}) = \text{LN}(\text{MHA}^l(\mathbf{X} + \mathbf{P}, \mathbf{X} + \mathbf{P}, \mathbf{X}) + \mathbf{X}), \quad (11)$$

$$\text{enc}_2^l(\mathbf{X}) = \text{LN}(\text{PFFN}^l(\mathbf{X}) + \mathbf{X}), \quad (12)$$

$$\text{PFFN}^l(\mathbf{X}) = \text{ReLU}(\mathbf{X}\mathbf{w}^l + \mathbf{b}^l), \quad (13)$$

where $\mathbf{X}^{l-1}$ and $\mathbf{X}^l$ denote the input and output of the *l*-th layer, respectively; $\mathbf{P} \in \mathbb{R}^{WH \times D}$ is the trainable 2D-positional encoding; LN denotes layer normalization; while $\mathbf{w}^l$ and $\mathbf{b}^l$ are the weight and bias of the PFFN layer, respectively. These processes repeat $L$ times using the output from the previous encoder layers as input. The output of the Transformer encoder is a transformed sequence $\mathbf{X}^L \in \mathbb{R}^{WH \times D}$, the same dimension as its input.

Following this encoding operation, the output from the Transformer encoder is then fed into a Transformer decoder, i.e., a stack of $L$ decoder layers. Each decoder layer consists of three sublayers: 1) MHSA, 2) Multi-Head Cross-Attention (MHCA), and 3) PFFN. The first and the third are similar to those of the encoder layers. However, the input to the first sublayer of the first decoder layer is a sequence of landmark queries $\mathbf{Y}^0 \in \mathbb{R}^{N \times D}$; each is an embedding of the same dimension $D$. The number of landmark queries equals $N$, which is the number of landmarks to predict. In this work, landmark queries are learnable parameters of LOTR, which are optimized during model training. The second sublayer, i.e., MHCA, takes the output of the first sublayer (MHSA) and the output generated by the encoder, i.e., $\mathbf{X}^L$, as inputs, and then computes the relationship between tokens in both sequences. The third sublayer, i.e., PFFN, then processes the output from the second sublayer. Like the Transformer encoder, all three of these sublayers have residual connections and layer normalization applied to them. The *l*-th decoder layer is defined as follows:

$$\mathbf{Y}^l = \text{dec}^l(\mathbf{Y}^{l-1}, \mathbf{Y}^0, \mathbf{X}^L, \mathbf{P}), \quad (14)$$

$$\text{dec}^l(\mathbf{Y}, \mathbf{Y}^0, \mathbf{X}^L, \mathbf{P}) = \text{dec}_3^l(\text{dec}_2^l(\text{dec}_1^l(\mathbf{Y}, \mathbf{Y}^0), \mathbf{X}^L, \mathbf{P})), \quad (15)$$

$$\text{dec}_1^l(\mathbf{Y}, \mathbf{Y}^0) = \text{LN}(\text{MHA}_1^l(\mathbf{Y} + \mathbf{Y}^0, \mathbf{Y} + \mathbf{Y}^0, \mathbf{Y}) + \mathbf{Y}), \quad (16)$$

$$\text{dec}_2^l(\mathbf{Y}, \mathbf{X}^L, \mathbf{P}) = \text{LN}(\text{MHA}_2^l(\mathbf{Y} + \mathbf{Y}^0, \mathbf{X}^L + \mathbf{P}, \mathbf{X}^L) + \mathbf{Y}), \quad (17)$$

$$\text{dec}_3^l(\mathbf{Y}) = \text{LN}(\text{PFFN}^l(\mathbf{Y}) + \mathbf{Y}), \quad (18)$$

where $\mathbf{Y}^{l-1}$ and $\mathbf{Y}^l$ denote the input and output of the *l*-th layer, respectively. The Transformer decoder produces $\mathbf{Y}^L \in \mathbb{R}^{N \times D}$ as output.

The landmark prediction head takes as input the sequence $\mathbf{Y}^L$ and outputs $\hat{\mathbf{Z}} \in \mathbb{R}^{N \times 2}$, which stores the predicted *x*- and *y*-coordinates of the $N$ landmarks. This work exploits a simple PFFN with two hidden layers with ReLU activation. The output layer, however, which consists of only two nodes, is without any activation function. In particular, the predicted landmarks $\hat{\mathbf{Z}}$ are computed as follows:

$$\hat{\mathbf{Z}} = \text{ReLU}(\mathbf{Y}^L\mathbf{w}_1^z + \mathbf{b}_1^z)\mathbf{w}_2^z + \mathbf{b}_2^z, \quad (19)$$

where $\mathbf{w}_j^z$ and $\mathbf{b}_j^z$ are the weight and bias of the *j*-th layer ($j \in \{1, 2\}$), respectively.

Similar to [37], the computational complexity of the MHSA in an encoder layer grows quadratically with the feature map size, i.e., $O((WH)^2D + WHD^2)$, while that in a decoder layer depends on the number of landmarks instead, i.e., $O(N^2D + ND^2)$. On the other hand, the complexity of MHCA in a decoder layer grows linearly with respect to the feature map size and the number of landmarks: $O(NWHD + (N + WH)D^2)$.

### B. SMOOTH-WING LOSS

Since the proposed LOTR predicts a fixed number of landmarks ($N$), a more complicated Hungarian loss used in DETR [37] is not required. The proposed LOTR can be trained end-to-end with a standard loss function given cropped-and-aligned face images as input. In particular, during the training phase, the predicted landmarks $\hat{\mathbf{Z}}$ generated by LOTR are compared with the ground truth landmarks $\mathbf{Z} \in \mathbb{R}^{N \times 2}$ to compute the loss, which is defined as:

$$\text{loss}(\mathbf{Z}, \hat{\mathbf{Z}}) = \sum_{i=1}^{N} \sum_{j=1}^{2} g(z_{ij} - \hat{z}_{ij}), \quad (20)$$

where the loss function $g(x)$ can be any standard loss function such as L1, L2, smooth-L1, or Wing loss, which are described in Section II-E. Although Feng *et al.* [39] reported that the Wing loss was superior to other loss functions for landmark localization, its major drawback is the gradient discontinuity at the threshold $w$ and around the zero error (Fig. 2). This discontinuity can affect the convergence rate and the stability of training.

In this work, we also propose a modified Wing loss, named smooth-Wing loss (s-wing($x$)), which is given by:

$$\text{s-wing}(x) = \begin{cases} sx^2 & \text{if } |x| < t \\ |x| - c_1 - c_2 & \text{if } |x| > w \\ (w+\epsilon)\ln(1+\dfrac{|x|}{\epsilon}) - c_2 & \text{otherwise,} \end{cases} \quad (21)$$

$$s = \frac{w + \epsilon}{2t(\epsilon + t)}, \quad (22)$$

$$c_1 = w - (w + \epsilon)\ln(1 + \frac{w}{\epsilon}), \quad (23)$$

$$c_2 = st^2, \quad (24)$$

where $t$ is an additional threshold ($0 < t < w$). When the error is smaller than the inner threshold $t$, it behaves

like L2 loss, allowing the gradient to be smooth at zero error; otherwise, it behaves like the Wing loss. We define the constants $s$, $c_1$, and $c_2$ to smoothen the loss at the inner threshold $t$ as well as at the outer threshold $w$. As shown in Fig. 2 (bottom-right), the gradient of the smooth-Wing loss changes linearly when the absolute error $|x|$ is smaller than the inner threshold $t$. Moreover, the gradient discontinuities at $|x| = w$ are also eliminated, as shown in Fig. 2 (bottom-middle).

## IV. EXPERIMENTS

### A. DATASETS
We conducted experiments to measure the performance of the proposed LOTR models on two benchmark datasets: 1) the 106-point JD landmark dataset [52] and 2) the Wider Facial Landmarks in-the-Wild (WFLW) dataset [38].

The JD-landmark dataset contains images from other face landmark datasets including as 300W [62], [63], LFPW [26], AFW [24], HELEN [64], and IBUG [65], covering large variation of face pose and expression. These images are re-annotated with 106-point landmarks, which provide more information about the face structure than any other face landmark dataset. The JD dataset consists of 11,393 images for training and 2,000 images each for validation and testing.

The WFLW dataset, which is based on a well-known face detection dataset called WIDER FACE [66], was recently proposed to be a new benchmark for facial landmark localization. The WFLW dataset consists of 10,000 faces: 7,500 for training and 2,500 as test images. Compared to other previous datasets, the WFLW dataset is manually annotated with 98 landmarks for each face, providing more information on the face structure. It is an extremely challenging dataset due to large pose, expression, and occlusion variations. It also supplies six other annotations of face properties, including pose, expression, illumination, make-up, occlusion, and blur. More than 78% of images in the provided test set have annotation with one or more, up to four, properties.

Following [1], we used a ResNet50-based face detector, proposed by Deng *et al.* [67], in our pre-processing step. In particular, we used the bounding box and a set of five simple landmarks (i.e., eye centers, nose tip, and mouth corners) obtained from the detector to crop and align the detected faces. We then resized each input image to $192 \times 192$ pixels for the JD-landmark dataset and to $256 \times 256$ pixels for the WFLW dataset before feeding it to our model. The detail of the pre-processing step is described in [1].

### B. EVALUATION METRICS
Following [1], [34], [47], [54], we used the standard metrics including the normalized mean error (NME), the failure rate, and the area under the curve (AUC) of the cumulative distribution to evaluate and compare landmark localization algorithms. The NME is computed across all predicted landmarks as follows:

$$\text{NME} = \frac{1}{N} \sum_{i=1}^{N} \frac{||\mathbf{z}_i - \hat{\mathbf{z}}_i||_2}{d}, \tag{25}$$

**TABLE 1.** The architectures of LOTRs used in the experiments with the JD-landmark dataset.

|  | LOTR-M | LOTR-M+ | LOTR-R+ |
|---|---|---|---|
| Backbone | MNetV2$_{1.0}$ | MNetV2$_{1.0}$ | ResNet50 |
| $1 \times 1$ conv. | 64 | 256 | 256 |
| Upsampling | - | 128, 64 | 128, 64 |
| Transformer | 2, 2 | 2, 2 | 2, 2 |
| PFFN | 512, 512, 2 | 512, 512, 2 | 512, 512, 2 |

where $N$ is the number of landmarks, $\mathbf{z}_i$ and $\hat{\mathbf{z}}_i$ denotes the $i$-th ground truth landmark and the $i$-th predicted landmark, respectively, and $d$ is a normalization factor. For the JD-landmark dataset, following [52], the normalization factor $d$ was defined as $\sqrt{W_{\text{bbox}} H_{\text{bbox}}}$, where $W_{\text{bbox}}$ and $H_{\text{bbox}}$ are the width and height of the bounding box enclosing all the ground truth landmarks, respectively. While, for the WFLW dataset, the inter-ocular distance, i.e., the distance between the outer eye corners, was used as the normalization factor. If the NME of a test image is above a threshold, it is considered a failure. The failure rate is, therefore, the rate of failure cases. The threshold was set to 8% for the JD-landmark dataset and 10% for the WFLW dataset. The AUC is computed from the cumulative error distribution, representing the proportion of images with NME smaller than the threshold. Therefore, a larger AUC represents a more accurate algorithm.

### C. MODEL TRAINING
Table 1 presents the configuration of three LOTR models, namely LOTR-M, LOTR-M+, and LOTR-R+, used in the experiments with the JD-landmark dataset. The LOTR-M model is the base model that utilizes a MobileNetV2$_{1.0}$, pre-trained on the ImageNet dataset [68], as the visual backbone, generating a feature map of size $6 \times 6 \times 1280$ from a $192 \times 192$ RGB image. We reduced the number of feature channels to 64 by a $1 \times 1$ convolution, then reshaping into a sequence of tokens. Subsequently, we used a Transformer module with two encoder layers and two decoder layers to process the sequence, followed by a dropout technique [69] with a dropout rate of 0.1. Following, we used a landmark prediction head, i.e., a PFFN consisting of two hidden layers with 512 nodes, with ReLU activation, followed by an output layer with two nodes, to process each token in the output sequence of the Transformer. In the LOTR-M+ model, to increase the resolution of the feature map, we inserted upsampling layers between the MobileNetV2$_{1.0}$ backbone and the Transformer module. In particular, we utilized a $1 \times 1$ convolution to reduce the number of channels to 256 and then applied two deconvolution layers (128 and 64 filters, with filter size = $4 \times 4$) to increase the feature resolution to $24 \times 24$ pixels before feeding into the Transformer. The LOTR-R+ model adopts the same configuration as LOTR-M+, except the backbone is changed to a pre-trained ResNet50. For the WFLW dataset, we experimented with another model, named LOTR-HR+, that uses the same architecture as LOTR-M+ and LOTR-R+ but exploits HRNet [61] as the backbone.

**TABLE 2.** Comparison with the state-of-the-arts on the WFLW dataset.

| Metric | Method | Test set | Subsets | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Pose | Expression | Illumination | Make-up | Occlusion | Blur |
| NME (%) | LAB | 5.27 | 10.24 | 5.51 | 5.23 | 5.15 | 6.79 | 6.32 |
| | Wing | 5.11 | 8.75 | 5.36 | 4.93 | 5.41 | 6.37 | 5.81 |
| | AWing | 4.36 | 7.38 | 4.58 | 4.32 | 4.27 | 5.19 | 4.96 |
| | LUVLi | 4.37 | - | - | - | - | - | - |
| | GV | 4.33 | 7.41 | 4.51 | 4.24 | 4.18 | 5.19 | 4.93 |
| | HIH | 4.18 | 7.20 | 4.19 | 4.45 | 3.97 | 5.00 | 4.81 |
| | LOTR-HR+ | 4.31 | 7.50 | 4.62 | 4.21 | 4.16 | 5.22 | 4.89 |
| Failure rate (%) | LAB | 7.56 | 28.83 | 6.37 | 6.73 | 7.77 | 13.72 | 10.74 |
| | Wing | 6.00 | 22.70 | 4.78 | 4.30 | 7.77 | 12.50 | 7.76 |
| | AWing | 2.84 | 13.50 | 2.23 | 2.58 | 2.91 | 5.98 | 3.75 |
| | LUVLi | 3.12 | - | - | - | - | - | - |
| | GV | 3.52 | 16.26 | 2.55 | 3.30 | 3.40 | 6.79 | 5.05 |
| | HIH | 2.96 | 15.03 | 1.59 | 2.58 | 1.46 | 6.11 | 3.49 |
| | LOTR-HR+ | 3.52 | 16.31 | 3.53 | 3.16 | 2.43 | 7.07 | 5.19 |
| AUC (%) | LAB | 53.23 | 23.45 | 49.51 | 54.33 | 53.94 | 44.90 | 46.30 |
| | Wing | 55.04 | 31.00 | 49.59 | 54.08 | 55.82 | 48.85 | 49.18 |
| | AWing | 57.19 | 31.20 | 51.49 | 57.77 | 57.15 | 50.22 | 51.20 |
| | LUVLi | 57.70 | - | - | - | - | - | - |
| | GV | 57.75 | 31.66 | 56.36 | 58.63 | 58.81 | 50.35 | 52.42 |
| | HIH | 59.70 | 34.20 | 59.00 | 60.60 | 60.40 | 52.70 | 54.90 |
| | LOTR-HR+ | 60.14 | 36.59 | 57.23 | 60.92 | 60.70 | 54.65 | 55.53 |

We initialized the parameters of the LOTR models using He's method [70]. We used a standard normal distribution to initialize the positional encoding and the landmark queries, both learnable. For the landmark queries, we specifically used a standard deviation of $10^{-4}$ for the initialization. We trained the LOTR models using the LAMB optimizer [71] with a base learning of $10^{-3}$ for 100 epochs while reducing the learning rate with a factor of 0.1 at epochs 50 and 75 and setting the batch size to 32. We used the smooth-Wing loss, described in Section III-B, setting the inner threshold ($t$) to 0.01, the outer threshold ($w$) to 10, and the steepness control parameter ($\epsilon$) to 2 as the parameters. For data augmentation and training tricks, we used the same steps described by [1]. We implemented the model using MXNet framework [72] with Gluon libraries [73], using a single NVIDIA Titan X GPU for training.

## D. RESULTS ON THE WFLW DATASET

In this section, we compare the proposed LOTR with several state-of-the-art methods, including Look-at-Boundary (LAB) [38], Wing loss [39], adaptive Wing loss (AWing) [54], LUVLi [47], Gaussian vector (GV) [34], and Heatmap-In-Heatmap (HIH) [53]. As shown in Table 2, our proposed LOTR-HR+ achieves an NME of 4.31%, clearly outperforming LAB, Wing, AWing, and LUVLi methods, and yields an AUC of 60.14%, surpassing all state-of-the-arts by a large margin (0.44–6.91 points). While comparable with GV in terms of NME and failure rate, our LOTR model achieves a better AUC and does not require any post-processing step. Although our proposed LOTR model does not surpass the performance of HIH in terms of NME and failure rate, it outperforms this state-of-the-art model on the AUC metric.

Fig. 3 presents sample images of each subset from the test set of the WFLW dataset by visualizing predicted landmarks

**TABLE 3.** The evaluation results for different LOTR models on the JD-landmark test set; † and ‡ denote the first and second place entries.

| Model | AUC (%) | Failure rate (%) | NME (%) |
|---|---|---|---|
| LOTR-R+ | 87.71 | 0.00 | 0.98 |
| LOTR-M+ | 87.21 | 0.00 | 1.02 |
| Earp et al. (ResNet50) [1] | 87.06 | 0.00 | 1.03 |
| LOTR-M | 87.05 | 0.00 | 1.03 |
| Earp et al. (MobileNetV2$_{1.0}$) [1] | 86.49 | 0.00 | 1.08 |
| †Baidu-VIS | 84.01 | 0.10 | 1.31 |
| Xiong et al. [34] | 83.34 | 0.10 | 1.35 |
| ‡USTC | 82.68 | 0.05 | 1.41 |

overlaid on the input images. In this figure, we show images with different ranges of the NME where the first row has NME less than 0.05; the middle row has NME in the range of 0.05 to 0.06, and the last row has NME more than 0.06. The figure shows the face landmark localization ability of our model, which can accurately locate facial key points with different variations, i.e., large pose, exaggerated expression, extreme illumination, make-up, occlusion, and blur.

## E. RESULTS ON THE JD-LANDMARK DATASET

Table 3 presents the performance of the proposed LOTR models, evaluated on the test set of the first Grand Challenge of 106-Point Facial Landmark Localization.[1] For brevity, the table includes only the result of the top two ranked algorithms out of 21 algorithms submitted to the challenge.

Table 3 shows that the proposed LOTR models, including the smallest model, i.e., LOTR-M, gain more than 3 points in the AUC in comparison to the top two ranked algorithms on the first challenge leaderboard. The table also compares LOTR models with two recent methods based on the heatmap regression, i.e., [1] and [34]. All of the proposed LOTR models surpass the heatmap approach in [34] by a significant

[1]https://facial-landmarks-localization-challenge.github.io

**FIGURE 3.** Sample images of the test set of the WFLW dataset with predicted landmarks from our model. Each column displays the images with different subsets. Each row displays images with a different range of NMEs: < 0.05 (top), 0.05–0.06 (middle), and > 0.06 (bottom).

amount of the AUC gain (3.7–4.5 points). In comparison with [1], our LOTR models achieve better performance using the same backbone. While our smallest model, i.e., LOTR-M, is comparable with the ResNet50 model by [1], our bigger models, i.e., LOTR-M+ and LOTR-R+, surpass it by an AUC gain of 0.15–0.65 points, showing the superiority of our approach over theirs. Similar to [1], we also utilized a flipping technique during inference to improve the prediction accuracy. In particular, we fed the original cropped-and-aligned face images and their horizontally flipped version into the model and averaged their corresponding predicted landmark coordinates.

In terms of floating point operations per second (FLOPS), the LOTR-M model uses only 0.23 GFLOPS. In contrast, the LOTR-M+ model has 47 percent higher GFLOPS than the former due to the presence of upsampling layers; however, operating with only 0.44 GFLOPS. The LOTR-R+ model uses 3.23 GFLOPS which is significantly higher than both the LOTR-M and LOTR-M+ models due to its heavier ResNet50 backbone.

### F. EFFECT ON FACE RECOGNITION PERFORMANCE

In recent years, face recognition systems have evolved to contain four modules: face detection, alignment, embedding, and distance computation. Several existing face detectors predict the location of five naive landmarks that correspond to the center of each eye, the tip of the nose, and the corners of the mouth. These five landmarks are used for face alignment in accordance to the conventional five-landmark alignment

protocol (e.g., [3], [5]–[9]). In this section, we investigate how the face alignment process using our proposed LOTR models affects the performance of the state-of-the-art face recognition system.

In our experiments, we first detected the face and then used our LOTR models to obtain the 106 facial landmarks. Following this, we extracted a subset of the five landmarks for face alignment to make a fair comparison. We experimented with two different face detectors, namely PyramidKey with MobileNetV2$_{1.0}$ backbone proposed by Earp *et al.* [74] and the publicly available RetinaFace with a ResNet50 backbone from Deng *et al.* [67]. We exploited the pre-trained LResNet100E-IR from InsightFace[2] as the face embedding network. Table 4 shows the performance of the face embedding network with different combinations of the face detectors and face alignment processes on six benchmark datasets, which include LFW [75], CFP-FP [40], CALFW [76], CPLFW [41], IJB-B [42], and IJB-C [43]. We report the True Acceptance Rate (TAR) @ False Acceptance Rate (FAR) = $10^{-4}$, following the protocol for IJB-B and IJB-C from Arc-Face (see [8], [9] for more details) which incorporates both detector score and feature normalization.

The results reported in Table 4 indicate no improvement on LFW with either of the face detectors and the LOTR models. However, with PyramidKey as the face detector, we see an improvement of 1.85 and 2.47 points on CFP-FP and CALFW with the LOTR-M model while improving by 10.83 points on the CPLFW dataset with the LOTR-R+ model, respectively.

[2]https://github.com/deepinsight/insightface

**TABLE 4.** The performance (TAR @ FAR = $10^{-4}$) of face recognition on several benchmarks.

| Face detector | Face landmarks | LFW | CFP-FP | CALFW | CPLFW | IJB-B | IJB-C |
|---|---|---|---|---|---|---|---|
| PyramidKey | From detector | **99.73** | 94.66 | 84.70 | 53.77 | 94.37 | 94.59 |
| [74] | LOTR-M | **99.73** | **96.51** | **87.17** | 61.67 | 94.47 | **94.92** |
| | LOTR-M+ | **99.73** | 94.46 | 85.37 | 63.13 | 94.49 | 94.88 |
| | LOTR-R+ | **99.73** | 95.28 | 86.43 | **64.60** | **94.54** | 94.89 |
| RetinaFace | From detector | **99.73** | 94.71 | **87.47** | 57.97 | 94.43 | 94.59 |
| [67] | LOTR-M | **99.73** | 94.68 | 86.20 | 62.90 | 94.48 | **94.79** |
| | LOTR-M+ | **99.73** | **96.23** | 86.00 | 62.90 | 94.48 | 94.77 |
| | LOTR-R+ | **99.73** | 96.06 | 86.37 | **63.67** | **94.54** | 94.74 |

On the other hand, with RetinaFace as the face detector, we see a gain of 1.52 points on CFP-FP with LOTR-M+, while improving by 5.7 points with LOTR-R+ on CPLFW. Similarly, with both the face detectors, we found an improvement of 0.17 points with LOTR-R+ and 0.33 points with the LOTR-M+ model on the IJB-B and IJB-C datasets. The improvement in the TAR on CPLFW is note-worthy as this dataset include images with large pose variations. Thereby, the results suggest that the LOTR models are more robust to pose variations.

### G. ABLATION STUDIES

This section further studies the proposed methods in various points, including the computational complexity and model size, the effect of the Transformer network, the number of Transformer encoder/decoder layers, and the proposed smooth-Wing loss. All experiments in this section were conducted on the JD-landmark dataset.

#### 1) COMPARISON WITH A HEATMAP REGRESSION APPROACH

To study the efficiency of the proposed LOTR models, we measured and compared the inference time with a heatmap-based approach. We selected the models with the same backbones from Earp *et al.* [1] to serve as the baseline since we followed the same pre-processing procedure, visual backbone, and the bag of tricks. We reported the result in Table 5 where we ran the models on a high computational CPU (Intel Xeon CPU E5-2698 v4) and NVIDIA Tesla V100 SXM2 GPU with 32 GB of RAM.

In the pre-processing step, all the models used the same face detector by Deng *et al.* [67] as mentioned in Section IV-A with a processing time of $43.36 \pm 4.82$ ms. Table 5 demonstrates the inference time reduction from the baseline on the GPU by $\sim$4–6$\times$, while maintaining a comparable model size and the number of parameters. Moreover, when considering the model with MobileNetV2$_{1.0}$ as a backbone, the computational time of LOTR-M and LOTR-M+, with or without the flipping technique, is lower than the MobileNetV2$_{1.0}$ model in [1] without the flipping technique. This phenomenon is consistent with the larger visual backbone as well. Table 5 demonstrates that the LOTR-R+ model with flipping is $\sim$4.4$\times$ faster than the MobileNetV2$_{1.0}$ model with flipping in [1] and $\sim$2.1$\times$ faster than that model without the flipping technique.

This experimentation demonstrates the complexity of the heatmap regression approach, which relies on a complicated post-processing procedure to generate spatial probability maps. Furthermore, some complex operations in its post-processing stage could not efficiently utilize GPU acceleration, resulting in high computational time. In contrast, the proposed LOTR models use a Transformer to directly regress the coordinates, thereby, enabling the models to reduce the computational time as it avoids complicated post-processing and is capable of utilizing GPU acceleration as the models consist of only simple operations.

Inspired by Xiong *et al.* [34], we conducted another experiment incorporating vectorization and the band pooling module with the baseline approach [1]. We analyzed the effect on model performance and post-processing time reduction with vectorization. While the post-processing time significantly drops when converting heatmaps to vectorized labels and predictions, there is also a drop in performance. The results indicate that the proposed LOTR models still outperform the baseline with vectorization in terms of prediction accuracy and inference time.

#### 2) COMPARISON WITH A FEED-FORWARD NETWORK ARCHITECTURE

To investigate the significance of a Transformer's ability in processing spatial features from the visual backbone, we compared its performance with a baseline CNN model with a Feed-Forward Network (FFN) replacing the Transformer module. For a fair comparison, we applied a $1 \times 1$ convolution layer after retrieving the feature map from the visual backbone to resemble the channel dimension of the LOTR models. We then used the FFN as a landmark prediction head to output the coordinates of facial landmarks. For comparison purposes, we experimented with MobileNetV2$_{1.0}$ and ResNet50 as the visual backbones and adopted the same loss function and training tricks as described in Section IV-C.

According to the results presented in Table 5, the performance of the CNNs with FFN head is worse than LOTRs for both flip and no flip setups. This drop in performance might be due to the incapability of the FFN to capture complex relationships between spatial features.

In contrast, the Transformer module incorporates self-attention and cross-attention mechanisms to model this relationship. Moreover, the use of landmark queries on the decoder side and the positional encoding might be the reason

**TABLE 5.** Comparison of prediction accuracy, inference time, and model size.

| Model | Backbone | AUC (%) | | GPU (ms) | | Size | Params |
|---|---|---|---|---|---|---|---|
| | | No flip | Flip | No flip | Flip | (MB) | ($\times 10^6$) |
| Earp *et al.* [1] | MobileNetV2$_{1.0}$ | 85.24 | 86.49 | 24.74±2.08 | 48.28±3.20 | 12.16 | 2.92 |
| | ResNet50 | 86.43 | 87.06 | 24.68±1.51 | 50.99±3.67 | 94.10 | 24.42 |
| Earp *et al.* [1] with | MobileNetV2$_{1.0}$ | 84.31 | 85.51 | 10.20±1.45 | 18.90±0.18 | 12.16 | 2.92 |
| vectorization | ResNet50 | 84.65 | 85.94 | 11.12±0.27 | 20.02±0.35 | 94.10 | 24.42 |
| CNN + FFN | MobileNetV2$_{1.0}$ | 85.85 | 86.37 | 5.51±0.68 | 8.98±1.95 | 14.71 | 3.89 |
| | ResNet50 | 86.18 | 86.91 | 6.09±0.90 | 9.86±0.96 | 96.09 | 25.24 |
| LOTR-M | MobileNetV2$_{1.0}$ | 86.39 | 87.05 | 5.46±0.95 | 7.68±1.08 | 10.59 | 2.81 |
| LOTR-M+ | MobileNetV2$_{1.0}$ | 86.56 | 87.21 | 5.63±0.52 | 9.30±2.81 | 14.03 | 3.71 |
| LOTR-R+ | ResNet50 | 87.09 | 87.71 | 6.00±0.84 | 11.55±1.03 | 95.97 | 25.21 |

**TABLE 6.** Comparison of prediction accuracy (AUC) with different loss functions.

| Loss | LOTR-M | LOTR-M+ | LOTR-R+ |
|---|---|---|---|
| L2 | 84.52 | 84.28 | 86.73 |
| L1 | 86.90 | 86.95 | 87.44 |
| Smooth-L1 | 86.87 | 86.98 | 87.52 |
| Wing | 86.92 | 87.04 | 87.68 |
| Smooth-Wing | **87.05** | **87.21** | **87.71** |

**TABLE 7.** Comparison of prediction accuracy (AUC) based on varying number of transformer layers.

| Encoder/ decoder layers | LOTR-M | LOTR-M+ | LOTR-R+ |
|---|---|---|---|
| 1 | 86.88 | 87.00 | 87.46 |
| 2 | **87.05** | **87.21** | **87.71** |
| 3 | 86.86 | 86.92 | 87.42 |
| 4 | 44.09 | 85.88 | 60.84 |
| 5 | 43.74 | 44.40 | 44.16 |
| 6 | NaN | 44.14 | NaN |

that the LOTRs can encode abstract information of each landmark, resulting in an improved understanding of the models to perform direct regression. This shows that the Transformer module is more effective than the FFN.

### 3) COMPARISON OF LOSS FUNCTIONS
In this section, we investigated the effect of the proposed smooth-Wing loss, as described in Section III-B, by comparing it with standard loss functions (e.g., L1, L2, and smooth-L1) and Wing loss (Section II-E). We conducted the experiment with the same training process described in Section IV-C. For the Wing loss, we set the threshold $w = 10$ and $\epsilon = 2$.

Table 6 shows the results from different loss functions on the JD-landmark test set [52]. The results show that LOTRs with L1 achieve comparable performance with smooth-L1, while with L2, the proposed models' performance is worse than other loss functions. This result also coincides with the results from Feng *et al.* [39]. Unlike L1 loss, which maintains a constant gradient value across the error range, L2 produces a smaller gradient near zero, which causes the models to ignore small error values. Thus, L2 loss is sensitive to outliers making it less responsive to relatively smaller errors. Consequently, the models trained with L2 loss may end up omitting small errors, which may yield inaccurate predictions.

Moreover, loss functions for landmark localization—Wing and smooth-Wing—consistently outperform standard loss functions. The focus of these loss functions is on a small error range, which aids in precise landmark coordinate predictions.

Comparing the Wing loss and the proposed smooth-Wing loss, LOTRs trained with the smooth-Wing loss outperform those trained with the Wing loss. This result shows the impact of training stability from the smooth-Wing loss. Fig. 2 shows an uneven gradient of Wing loss at the threshold $w$, which

may not be suitable for parameter adjustment. The proposed smooth-Wing loss smoothens the gradient at the threshold $w$, making training more stable and, thus, essential for parameter adjustment. Table 3 and 5 show that with the same visual backbone, our LOTRs with smooth-Wing loss outperform the heatmap-based approach in [1]. This establishes that training with the smooth-Wing loss helps the proposed LOTRs achieve state-of-the-art performance on the JD-landmark test set.

### 4) THE NUMBER OF TRANSFORMER ENCODER/DECODER LAYERS
This section studies the impact of layers ($L$) in both the Transformer encoder and decoder. We experimented with the different values of $L$ while keeping the other hyper-parameters the same. Table 7 shows the results from the different numbers of encoder and decoder layers. The results show that the LOTR models with up to three layers can accurately localize landmarks and yield the highest AUC when $L$ equals 2. The models become harder to optimize as the number of layers increases from three. The results of the deepest models, i.e., $L$ equals 6, is NaN, which might be a consequence of training instability when the number of layers becomes very large.

## V. CONCLUSION
We show that our proposed LOTRs outperform other algorithms, including the two current heatmap-based methods on the JD-landmark challenge leaderboard, and are comparable with several state-of-the-art methods on the WFLW dataset. The results suggest that the Transformer-based direct coordinate regression is a promising approach for robust facial landmark localization.

We evaluate and illustrate that using the LOTRs for face alignment improves the state-of-the-art face recognition performance on the standard benchmark datasets such as CFP-FP, CPLFW, IJB-B, and IJB-C datasets.

We demonstrate how the proposed LOTR outperforms CNN models with FFN, trained under the same conditions. The results suggest an effective use of a Transformer network to improve the feature representation from a visual backbone. In contrast to other coordinate regression approaches that suffer from spatial information loss, our LOTRs utilize the crucial spatial information for landmark localization tasks.

We also show that the LOTRs are superior to the recently proposed heatmap-based method by Earp *et al.* [1] in terms of accuracy and inference time. Since the proposed LOTR directly predicts landmark coordinates, it avoids any computationally intensive post-processing required by the heatmap-based method, leading to a $\sim$4–6$\times$ gain in speed during inference. Although Xiong *et al.* [34] exploited their proposed vectorization method to reduce the post-processing time, its downside is the reduction in prediction accuracy. The end-to-end training behavior of the LOTRs with the smooth-Wing loss also leads to better prediction performance when compared to the heatmap-based methods.

While Feng *et al.* [39] reported that their proposed Wing loss is superior over other common loss functions such as L2, L1, and smooth-L1, we show that our proposed smooth-Wing loss leads to better optimized models than the Wing loss because of its gradient continuity. The results indicate an improvement in training stability and convergence rate using the smooth-Wing loss.

## ACKNOWLEDGMENT

## REFERENCES

[1] S. W. F. Earp, A. Samacoits, S. Jain, P. Noinongyao, and S. Boonpunmongkol, "Sub-pixel face landmarks using heatmaps and a bag of tricks," 2021, *arXiv:2103.03059*.

[2] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 365–372.

[3] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Computer Vision—ACCV 2009*, H. Zha, R. Taniguchi, and S. Maybank, Eds. Berlin, Germany: Springer, 2010.

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. 26th Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2015, pp. 41.1–41.12.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[6] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6738–6746.

[7] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4685–4694.

[9] X. An, X. Zhu, Y. Gao, Y. Xiao, Y. Zhao, Z. Feng, L. Wu, B. Qin, M. Zhang, D. Zhang, and Y. Fu, "Partial FC: Training 10 million identities on a single machine," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1445–1449.

[10] P. Barra, C. Bisogni, M. Nappi, and S. Ricciardi, "Fast quadtree-based pose estimation for security applications using face biometrics," in *Network and System Security*, M. H. Au, S. M. Yiu, J. Li, X. Luo, C. Wang, A. Castiglione, and K. Kluczniak, Eds. Cham, Switzerland: Springer, 2018, pp. 160–173.

[11] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.

[12] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, Jul. 2013.

[13] J. Roth, Y. Tong, and X. Liu, "Unconstrained 3D face reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2606–2615.

[14] P. Dou, S. K. Shah, and I. A. Kakadiaris, "End-to-end 3D face reconstruction with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1503–1512.

[15] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 557–574.

[16] X. Yang, Y. Li, H. Qi, and S. Lyu, "Exposing GAN-synthesized faces using landmark locations," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jul. 2019, pp. 113–118.

[17] M. Day, "Exploiting facial landmarks for emotion recognition in the wild," 2016, *arXiv:1603.09129*.

[18] M. I. N. P. Munasinghe, "Facial expression recognition using facial landmarks and random forest classifier," in *Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2018, pp. 423–427.

[19] S. Hinduja and S. Canavan, "Facial action unit detection using 3D facial landmarks," 2020, *arXiv:2005.08343*.

[20] H. Dibeklioglu, A. A. Salah, and L. Akarun, "3D facial landmarking under expression, pose, and occlusion variations," in *Proc. IEEE 2nd Int. Conf. Biometrics, Theory, Appl. Syst.*, Sep. 2008, pp. 1–6.

[21] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[22] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2, H. Burkhardt and B. Neumann, Eds., 1998, pp. 484–498.

[23] L. Liang, R. Xiao, F. Wen, and J. Sun, "Face alignment via component-based discriminative search," in *Computer Vision—ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, Oct. 2008, pp. 72–85.

[24] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2879–2886.

[25] B. Amberg and T. Vetter, "Optimal landmark detection using shape models and branch and bound," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 455–462.

[26] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2930–2940, Dec. 2013.

[27] B. A. Efraty, M. Papadakis, A. Profitt, S. Shah, and I. A. Kakadiaris, "Facial component-landmark detection," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Mar. 2011, pp. 278–285.

[28] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3476–3483.

[29] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2013, pp. 386–391.

[30] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 1–16.

[31] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 918–930, May 2016.

[32] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.

[33] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2034–2043.

[34] Y. Xiong, Z. Zhou, Y. Dou, and Z. Su, "Gaussian vector: An efficient solution for facial landmark detection," in *Computer Vision—ACCV 2020*, H. Ishikawa, C.-L. Liu, T. Pajdla, and J. Shi, Eds. Cham, Switzerland: Springer, 2021, pp. 70–87.

[35] S. Mahpod, R. Das, E. Maiorana, Y. Keller, and P. Campisi, "Facial landmarks localization using cascaded neural networks," 2018, arXiv:1805.01760.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1–11.

[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 213–229.

[38] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2129–2138.

[39] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2235–2245.

[40] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[41] T. Zheng and W. Deng, "Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments," Beijing Univ. Posts Telecommun., Beijing, China, Tech. Rep. 18-01, Feb. 2018. [Online]. Available: http://www.whdeng.cn/cplfw/index.html?reload=true

[42] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother, "IARPA Janus benchmark-B face dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 592–600.

[43] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, "IARPA janus benchmark-C: Face dataset and protocol," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 158–165.

[44] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 94–108.

[45] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Style aggregated network for facial landmark detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 379–388.

[46] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," 2019, arXiv:1904.04514.

[47] A. Kumar, T. K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng, "LUVLi face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8233–8243.

[48] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 483–499.

[49] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4724–4732.

[50] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 472–487.

[51] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7091–7100.

[52] Y. Liu *et al.*, "Grand challenge of 106-point facial landmark localization," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2019, pp. 613–616.

[53] X. Lan, Q. Hu, and J. Cheng, "HIH: Towards more accurate face alignment via heatmap in heatmap," 2021, arXiv:2104.03100.

[54] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6970–6980.

[55] X. P. Burgos-Artizzu, P. Perona, and P. Dollar, "Robust face landmark estimation under occlusion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1513–1520.

[56] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[57] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755.

[58] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[59] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[61] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[62] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2013, pp. 397–403.

[63] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image Vis. Comput.*, vol. 47, pp. 3–18, Mar. 2016.

[64] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Proc. Eur. Conf. Comput. Vis.*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 679–692.

[65] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 896–903.

[66] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5525–5533.

[67] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage dense face localisation in the wild," 2019, arXiv:1905.00641.

[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[69] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[71] Y. You, J. Li, S. J. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C. Hsieh, "Large batch optimization for deep learning: Training BERT in 76 minutes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–37.

[72] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems," 2015, arXiv:1512.01274.

[73] J. Guo, H. He, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha, A. Zhang, H. Zhang, Z. Zhang, Z. Zhang, S. Zheng, and Y. Zhu, "GluonCV and GluonNLP: Deep learning in computer vision and natural language processing," *J. Mach. Learn. Res.*, vol. 21, no. 23, pp. 1–7, 2020.

[74] S. W. F. Earp, P. Noinongyao, J. A. Cairns, and A. Ganguly, "Face detection with feature pyramids and landmarks," 2019, *arXiv:1912.00596*.

[75] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection Alignment Recognit.*, Oct. 2008, pp. 1–15.

[76] T. Zheng, W. Deng, and J. Hu, "Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments," 2017, *arXiv:1708.08197*.

**UKRIT WATCHAREERUETAI** received the B.Eng. degree in electrical engineering from Kasetsart University, Thailand, in 2002, and the Master of Information Science and Doctor of Engineering degrees from the Graduate School of Information Science, Nagoya University, Japan, in 2007 and 2010, respectively. From 2002 to 2004, he worked as a Research Assistant at the Kasetsart Signal and Image Processing Laboratory. After his graduation, he joined the King Mongkut's Institute of Technology Ladkrabang (KMITL) as a Lecturer, and was then promoted to be an Assistant Professor. Currently, he is working as a Senior AI Researcher at Sertis Company Ltd. His research interests include computer vision, machine learning, deep learning, and evolutionary computation.

**BENJAPHAN SOMMANA** received the B.S. degree in mathematics from Mahidol University, Thailand, in 2016, and the M.Eng. degree in information and communication technology for embedded systems from the Sirindhorn International Institute of Technology, Thammasat University, Thailand, in 2019. After her graduation, she joined Sertis Company Ltd. as an AI Researcher in the field of computer vision. Her research interests include machine learning and computer vision.

**SANJANA JAIN** received the B.Eng. degree in computer and networks engineering from Assumption University, Thailand, in 2014, and the master's degree in computer science from the Asian Institute of Technology, Thailand, in 2017. In the year 2015, she did an internship at Eibiz Company Ltd. before pursuing her master's degree. After her graduation, she joined the AIT AI Center as a Researcher. She then joined Oxygen AI Company Ltd., in 2018, as a Machine Learning Engineer. Currently, she is working at Sertis Company Ltd. as an AI Researcher. Her research interests include computer vision, deep learning, generative models, and computational neuroscience.

**PAVIT NOINONGYAO** received the B.Eng. degree in software engineering and the Master of Engineering degree in computing in engineering systems from the International College, King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2017 and 2019, respectively. Currently, he is working as a Quantitative Researcher at QIS Capital. His research interests include computer vision, machine learning, and deep learning.

**ANKUSH GANGULY** received the B.Tech. degree in mechanical engineering from the National Institute of Technology, Silchar, India, in 2014, and the M.Sc. degree in data science from The University of Edinburgh, in 2017, where his main area of focus was Bayesian machine learning. After graduation, he joined Sertis Company Ltd. as an AI Research Engineer and is currently working as a Senior AI Researcher focusing on deep learning and computer vision. His research interests include computer vision, Bayesian machine learning, deep learning, and probabilistic modeling.

**AUBIN SAMACOITS** received the Master of Science degree in bio-statistics from the Paris Institute of Statistics and the Ph.D. degree from Paris 6 University, where he developed image analysis methods for high content microscopy data. After his graduation, he worked as a Researcher at Bristol University where he developed AI methods for detection and tracking of in vitro human stem cells. After working for two years as an AI Researcher at Sertis Company Ltd., he is the current Manager of the AI Research Team.

**SAMUEL W. F. EARP** received the B.Sc. degree in mathematics and the M.Sc. and Ph.D. degrees in astrophysics from the University of Central Lancashire. He joined Sertis as an AI Researcher, in 2018, and subsequently became part of the Technical Leadership Team, focusing on face recognition, detection, and localization. Currently, he works as a Senior Quantitative Researcher at QIS Capital. He has six peer-reviewed journal articles in astrophysics and several white papers available on ArXiv in the field of AI and machine learning.
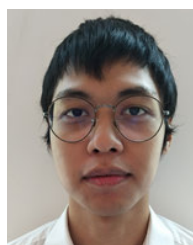
**NAKARIN SRITRAKOOL** is currently pursuing the bachelor's degree in computer science with Chulalongkorn University, Thailand. In 2021, he did an internship at Sertis Company Ltd. His research interests include recommender systems and machine learning.

• • •