Fast Facial Landmark Detection and Applications: A Survey

Detección Rápida de Puntos de Referencia Faciales y Aplicaciones: Estudio de la Bibliografía

Kostiantyn Khabarlak[1] and Larysa Koriashkina[1]

[1] *Department of System Analysis and Control, Dnipro University of Technology, Ukraine*
{habarlack, koriashkinals}@gmail.com

Abstract

Dense facial landmark detection is one of the key elements of face processing pipeline. It is used in virtual face reenactment, emotion recognition, driver status tracking, etc. Early approaches were suitable for facial landmark detection in controlled environments only, which is clearly insufficient. Neural networks have shown an astonishing qualitative improvement for in-the-wild face landmark detection problem, and are now being studied by many researchers in the field. Numerous bright ideas are proposed, often complimentary to each other. However, exploration of the whole volume of novel approaches is quite challenging. Therefore, we present this survey, where we summarize state-of-the-art algorithms into categories, provide a comparison of recently introduced in-the-wild datasets (e.g., 300W, AFLW, COFW, WFLW) that contain images with large pose, face occlusion, taken in unconstrained conditions. In addition to quality, applications require fast inference, and preferably on mobile devices. Hence, we include information about algorithm inference speed both on desktop and mobile hardware, which is rarely studied. Importantly, we highlight problems of algorithms, their applications, vulnerabilities, and briefly touch on established methods. We hope that the reader will find many novel ideas, will see how the algorithms are used in applications, which will enable further research.

Keywords: Computer Vision, Edge Computing, Facial Landmarks, Neural Networks, Mobile Applications, Literature Overview.

Resumen

La detección de puntos de referencia faciales densos es uno de los elementos clave del proceso de procesamiento de rostros. Se utiliza en la animación de rostros virtuales, el reconocimiento de emociones, el seguimiento del estado del conductor, etc. Los primeros enfoques eran adecuados para la detección de puntos de referencia faciales solo en entornos controlados, lo que claramente es insuficiente. Las redes neuronales han mostrado una asombrosa mejora cualitativa para el problema de detección de puntos de referencia faciales en condiciones del mundo real, y ahora están siendo estudiadas por muchos investigadores en el campo. Se proponen numerosas ideas brillantes, a menudo complementarias. Sin embargo, la exploración de todo el volumen de enfoques novedosos es bastante desafiante. Por lo tanto, presentamos esta encuesta, donde resumimos los algoritmos de última generación en categorías, brindamos una comparación de los conjuntos de datos introducidos recientemente (por ejemplo, 300W, AFLW, COFW, WFLW) que contienen imágenes con pose grande, oclusión facial, tomadas en condiciones sin restricciones. Además de calidad, las aplicaciones requieren una inferencia rápida y preferentemente en dispositivos móviles. Por lo tanto, incluimos información sobre la velocidad de inferencia de algoritmos tanto en hardware de escritorio como móvil, que rara vez se estudia. Es importante destacar que destacamos los problemas de los algoritmos, sus aplicaciones, vulnerabilidades y mencionamos brevemente los métodos establecidos. Esperamos que el lector encuentre muchas ideas novedosas, vea cómo se utilizan los algoritmos en las aplicaciones, lo que permitirá futuras investigaciones.

Palabras claves: Visión por computadora, Computación en la frontera, Puntos faciales de referencia, Redes neuronales artificiales, Aplicaciones móviles, Estudio de la bibliografía

1 Introduction

Dense facial landmark detection is one of the key elements of face processing pipeline. Applications include virtual face animation, emotion recognition, driver status tracking, etc. Early attempts to solve the problem were based on deformable face model, where statistical algorithms predicted face model deformation coefficients. These approaches were unsuitable for landmark annotation with large pose, face occlusion or unusual illumination. Later, attention has been driven to neural networks, that show high quality in solving tasks, in which we, humans, are good at, such as image classification or natural language process-

ing. Neural networks have also shown an astonishing qualitative improvement for in-the-wild face landmark detection problem, and are now being actively studied by many researchers in the field. Primarily, neural networks were designed to be executed on servers with many GPUs and a stable power supply. However, the development of Internet of Things and mobile devices makes client-server applications sometimes impractical or even unacceptable. For example, when Internet connectivity is poor, low latency data processing is required, if the amounts of raw data generated are too large to be sent over to a server. Finally, when no data can leave the user's device for security reasons. In many of these cases use of neural networks is desirable, and processing should be done directly on a mobile device. Thus, on-device machine learning has become one of the most prominent machine learning research directions [1], [2].

In this paper we present a description of recently introduced neural-network-based facial landmark detection algorithms. Existing surveys are quite old and mostly cover either statistical algorithms or the ones based on ensembles of regression trees [3], [4]. These algorithms show poor facial landmark detection quality for in-the-wild pictures (i.e., taken in unconstrained environments). Recently, numerous bright neural-network-based approaches were proposed, that show substantially better quality. However, exploration of the whole volume of novel approaches is quite challenging. Therefore, we present this work. The primary focus of this survey is on recently introduced algorithms, covering years 2018 – 2021. We include some important older algorithm for completeness as well.

We start our survey by defining facial landmark detection problem, algorithm quality assessment metrics. Next, we describe common in-the-wild datasets (e.g., 300W, AFLW, COFW, WFLW) with dense landmark annotation (from 21 to 98 landmarks). These datasets contain images taken in unconstrained conditions with large pose, face occlusion, different emotions, etc. The following section describes ideas of facial landmark algorithms, that have led to accuracy improvement or have proposed a novel way to solve the problem. This section is key for this survey. To make algorithm ideas clear, we start by explaining common neural network backbones used for facial landmark detection. Based on these materials, we follow with an explanation of facial landmark detection algorithms ordered by years. Finally, we summarize state-of-the-art algorithms into categories, provide accuracy comparison on recently introduced in-the-wild datasets. In addition to quality, applications require fast inference, possibly on mobile devices. Hence, we include information about algorithm inference speed both on desktop and mobile hardware, which is rarely studied in literature. Where available, inference time is shown for desktop CPU and GPU, as well as mobile phone. Also, we provide estimated number of neural network parameters and floating-point operations. These are the metrics, that influence memory consumption and inference time correspondingly. Importantly, we highlight problems of algorithms, their applications and vulnerabilities. Overall, we note that algorithm accuracy needs to be improved by the next generation of algorithms. Also, state-of-the-art algorithms have inference times that are quite high for practical applications. We hope that in this survey the reader will find many novel ideas, will see how the algorithms are used in applications, which will enable new research in the field.

The paper is structured as follows: Section 2 covers facial landmark detection problem. Section 3 describes datasets used to train and evaluate models. Section 4.1 gives a brief introduction of historical landmark detection methods. The main Sections 4.2 and 4.3 cover common neural network backbones and landmark detection algorithms correspondingly. Analysis of algorithm accuracy, inference speed, and summary of novel ideas is presented in Section 4.4. Section 5 is focused on real-world use of face landmark detection methods. We show several possible approaches of joint face and landmark detection algorithms in Section 5.1. Applications of dense facial landmark detection are shown in animation in Section 5.2, driver status tracking in Section 5.3, face and emotion recognition in Section 5.4. Finally, adversarial attack vulnerability is discussed in Section 6.

2 Facial Landmark Detection Problem Statement

Let I be an input image, which is represented in a form of 3-dimensional tensor of size $W \times H \times C$, where W, H, C are the width, height, and number of image color channels correspondingly. Note, that typically color images are used with 3 channels, one for red, green and blue colors. Then facial landmark detection problem is to find such function $\Phi : I \rightarrow Y$, that from the input image I predicts a landmark matrix $\hat{Y} \in R^{N_L \times 2}$, where N_L is the number of facial landmarks, $\hat{Y}_{i1} \in [0; W]$ represents X coordinate and $\hat{Y}_{i2} \in [0; H]$ represents Y coordinate of i^{th} landmark. Number of facial landmarks N_L and exact mapping between i^{th} facial landmark and its location on the face (the so-called annotation scheme) are defined at dataset level. Examples of face landmark annotations are present in Fig. 1. Also, dataset defines which images are used to train function Φ (train set) and which to evaluate (test set).

Next, we present commonly used metrics to report algorithms' quality on facial landmark detection datasets. Note, that each dataset has a special protocol, which defines train/test split, metrics for algorithm comparison, etc. The main metrics include [6], [7]:

(a) 300W[1] (b) WFLW[2] [5]

Figure 1: Examples of faces annotated with facial landmarks from several of the commonly used datasets: 300W and AFLW. In both cases landmarks cover areas of jaw, eyes, eyebrows, nose, lips. However, annotation schemes differ. For instance, WFLW annotates lower and upper boundary of eyebrows, whereas 300W has a single central line. Also, WFLW has the densest landmark annotation.

1. Normalized Mean Error (NME, %):

$$NME = \frac{1}{K}\sum_{k=1}^{K} NME_k,$$
$$NME_k = \frac{1}{N_L}\sum_{i=1}^{N_L}\frac{||Y_i - \hat{Y}_i||}{d} \times 100, \quad (1)$$

where Y is the matrix of true landmark locations, $\hat{Y}$ is the matrix of predicted landmark locations, d is the normalization coefficient (different for each dataset), N_L is the number of facial landmarks per face in the dataset, K is the number of images in the test set. Lower metric values are better.

2. Failure Rate (FR, %):

$$FR - \frac{1}{K}\sum_{k=1}^{K}[NME_k \geq 10\%] \times 100, \quad (2)$$

denotes number of images with Normalized Mean Error above 10 % threshold. Lower metric values are better.

3. Cumulative Error Distribution – Area Under Curve (CED-AUC). First, fraction of images whose NME is less than or equal to the NME value on X axis is plotted. Area under curve is then computed. Typically, NME is taken in range $[0; 10\%]$. Computed CED-AUC value is always scaled in range $[0; 1]$. Greater metric values are better, and denote that larger part of the test set is well predicted.

3 Common Face Landmark Datasets

There are several open datasets available to train and evaluate quality of face landmark detection algorithms. Each of the datasets includes image of a person and corresponding face landmark annotations. Landmarks are provided in a separate file. The datasets can include photos of the following kinds:

- in controlled environment (e.g., studio) or in-the-wild;
- with different shooting conditions, such as presence of face occlusion, large pose, make-up, etc.;
- real images or synthetic (when faces are generated with an algorithm);
- 2D or 3D face landmarks.

Next, we describe typical datasets used to train and evaluate facial landmark detection models. The datasets were selected from the following sources: 1) introduced jointly with a novel facial landmark detection algorithm; 2) presented separately, but at least one of the algorithms from Section 4.3 uses the dataset for training or evaluation. While we discuss all such datasets, the focus of this survey is on in-the-wild 2D face landmark datasets with non-synthetic images. Note, that in-the-wild datasets also include images in controlled environments, which makes them applicable to a wide range of practical use-cases.

300 Faces in-the-Wild (300W) [8] dataset contains a collection of different datasets: LFPW [9], AFW [10], HELEN [11], XM2VTS [12] and IBUG, that were relabeled with 68 facial landmarks. The protocol defines which images should be used for training and which for testing. The testing subset is split into *common*, *challenge* and *full*. Normalized Mean Error for each of the splits is usually presented for comparison. The NME is normalized (d in Eq (1)) by inter-pupil or inter-ocular (outer eye corner) distance. This is done, so that faces of different sizes make an equal contribution to the resulting error. Note, that images in the 300W dataset have different shooting conditions (lighting, color gamut), emotions and faces at an angle. There have been multiple extensions to the 300W datasets presented: **300W-LP-2D** [13], where the original 300W dataset has been expanded with synthetically generated images with large pose; **Masked-300W** [14] has synthetically added medical mask to the 300W dataset images. However, the same blue medical mask model has been used for all images, which is a disadvantage.

Annotated Facial Landmarks in-the-Wild (AFLW) [15] contains a larger number of images, yet they are labeled with only 21 facial landmarks. In comparison to 300W, this dataset has face photographs taken at a larger angle in range of $\pm 120^\circ$ yaw and $\pm 90^\circ$ pitch. The authors propose splitting the dataset into AFLW-Frontal (with face photos that are close to frontal) and AFLW-Full (all images). Also, there is a relabeled version with 68 facial landmarks, named **AFLW-68** [16], yet in practice it is used less

[1] Image is based on this source. License: CC BY 2.0. It has been annotated with landmarks available in 300W dataset by the authors of this survey.

[2] Author's written consent has been acquired to include this image.

often. **MERL-RAV** dataset presented in [6] has AFLW relabeled with 68 landmarks with an extra visibility label, such as: 1) visible; 2) self-occluded (for instance, due to large pose); 3) occluded by other object (hand, etc.). NME metric, normalized by face bounding box size (diagonal), is used for comparisons.

Caltech Occluded Faces in-the-Wild (COFW) [17] focuses on face images, that are partially occluded by real-world objects (microphone, etc.) or by the person itself (hair, hand, etc.). In addition to the NME metric, Failure Rate (FR, Eq (2)) is used. The dataset has 29 landmark annotations. NME is normalized by either inter-pupil or inter-ocular distance. The COFW test set has also been relabeled to 68 landmarks in **COFW-68** [18], but no COFW training set with 68 landmarks is available. COFW-68 can be used to assess landmark detection quality, when the network has been trained on a different dataset.

Wider Facial Landmarks in-the-Wild (WFLW) [5] is the dataset with the largest number of facial landmarks (98 landmarks). It is also the most recently introduced. In comparison to the datasets covered so far, WFLW has more images taken under unusual conditions, e.g., with make-up, wide range of emotions, poses, in various lighting conditions, etc. All three above-mentioned metrics are used to present the results: NME, Failure Rate and CED-AUC. NME is normalized by inter-ocular distance. The results are reported for each subset of unusual images, as well as for all images available in the dataset. This makes it possible to analyse, which conditions are the most challenging to the algorithms. The following subsets are available in WFLW dataset: Pose, Expression, Illumination, Make-Up, Occlusion, Blur. Information about image scene type is included in the dataset and can also be used during training.

Menpo-2D [19], [20] is a collection of frontal and profile faces. However, annotation schemes and number of landmarks are different between types of faces. The dataset is less used in practice.

In addition, there are many datasets that provide 3D annotations of facial landmarks (either synthetically generated or manually), such as 300W-LP [13], AFLW2000-3D [13], LS3D-W [21], Menpo-3D [20], [22]. Also, landmark annotated video datasets exist, e.g., 300 Videos in the Wild (300VW) [23]–[25].

Table 1 summarizes information about common datasets. We include information about number of labeled images for algorithm training and testing, as well as number of facial landmarks the dataset has been labeled with. The most widely used datasets are shown in bold.

Table 1: Information about facial landmark datasets: number of images contained in training and testing tests, as well as number of facial landmarks the dataset has been annotated with.

Dataset	Train	Test	Land.
300W [8]	3,837	600	68
AFLW [15]	20,000	4,386	21
COFW [17]	1,345	507	29
WFLW [5]	7,500	2,500	98
300W-LP-2D [13]	61,225	-	68
AFLW-68 [16]	20,000	4,386	68
COFW-68 [18]	-	507	68
Menpo-2D [19], [20]	7,564	7,281	68/39
MERL-RAV [6]	15,449	3,865	68
Masked-300W [14]	3,837	600	68

4 Facial Landmark Detection Algorithms

4.1 Early Landmark Detection Algorithms

First algorithms were mainly based on fitting a deformable face mesh. The most prominent algorithms include Active Shape Model (ASM), Active Appearance Model (AAM) and Constrained Local Model (CLM) [3], [4]. Based on the obtained mesh, each of the landmark locations are computed. In many cases such algorithms utilize statistical methods as a base. They have good enough prediction accuracy in controlled environments (with proper lighting and frontal face). However, such approaches offer underwhelming performance for most types of in-the-wild images. Next wave of methods was based on Random Forests and Gradient Boosting, such as ERT [26] algorithm, which we describe below. Such methods have better accuracy, but performance for occluded faces, faces shot under large angle or unusual illumination is still insufficient. As will be shown later in this work, many practical applications require accurate in-the-wild facial landmark detection.

Dlib [27] is an open-source machine learning library. Among others, it has **Ensemble of Regression Trees (ERT)** [26] facial landmark detection algorithm, which is a cascade, based on gradient boosting. The authors use a "mean" face template as an initial approximation, then the template is refined over several iterations. The algorithm requires the face to be first detected in the frame (Viola-Jones [28] face detector is used). Note, that most facial landmark detection algorithms require face to be first detected. High speed is the main advantage of ERT (according to the authors, around 1 millisecond per face). The library contains ERT implementation, trained on 300W dataset. The algorithm is still actively used in the modern research thanks to an open implementation and speed. However, not so long ago it has been shown that neural networks are preferred in terms of quality for faces with large

pose [29]. Mobile-friendly implementations of ERT are available.

An overview of early neural-network-based facial landmark detection algorithms can be found in [4], [30].

4.2 Face Landmark Detection Network Backbones

Modern in-the-wild face landmark detection algorithms are based on neural networks. They are divided into 2 main categories: *direct* (or *coordinate*) regression methods, when the model predicts x, y coordinates directly for each landmark; *heatmap*-based regression methods, where a 2D heatmap is built for each landmark. The values in the heatmap can be interpreted as probabilities of landmark location at a certain image location. Typically, argmax or its modification is used to acquire exact landmark coordinates from the heatmap. Fig. 2 illustrates the two approaches. As neural network architectures have become more complex, algorithms typically base on a pre-defined network architecture (called backbone). Facial landmark detection algorithms, described in the following subsection, propose modifications to training, inference procedure or the backbone itself. Here we introduce main backbones for landmark detection problem. Note, that in many cases backbones for face landmark and human pose (whole body) landmark detection are the same. Direct regression methods typically use widely known backbones from ImageNet challenge [31], such as ResNet [32], MobileNetV2 [2], MobileNetV3 [33], ShuffleNet-V2 [34]. Heatmap-based methods commonly use Hourglass [35] network architecture, but also HRNet [36] and CU-Net [37]. Such backbones are less known. Thus, we describe them here.

Hourglass [35] architecture has been initially designed for human pose estimation. The network takes a 256×256 image as an input. The authors note that processing the image at full resolution would require too much computation and memory. This is why a convolutional block is used to quickly process the image to obtain feature map of resolution 64×64, which remains the maximum feature map resolution till the end of the network. The feature map is then processed via Hourglass modules. An illustration of Hourglass network is shown in Fig. 3. Note, that architecture allows *stacking*, i.e., Hourglass modules can be repeated sequentially multiple times. Typically stacks of 1, 2 or 4 Hourglass modules are used. The network outputs heatmaps at a resolution of 64×64, a single heatmap is produced for each of the landmarks.

Hourglass module follows encoder-decoder architecture. Input image is processed via convolutional blocks at different feature map resolutions. First, feature map resolution is decreased after each convolutional block (encoder part), then feature map resolution is restored (increased) after each block (decoder part). Accuracy of human pose estimation, facial landmark detection and several other tasks is improved by processing image at multiple resolutions.

Overall, stacked Hourglass architecture becomes quite deep, which slows down training. The authors propose two ideas to solve the problem: skip connections inside Hourglass module and intermediate supervision. Firstly, as is clearly seen from Fig. 3, after each convolutional block the output is split into two parts, one part is downscaled and fed into next convolutional block, another is *skipped* from encoder to decoder. The latter concept is then referred to as "skip connection". This improves gradient propagation. Secondly, intermediate supervision is applied to each Hourglass module (as previously said, stacks of multiple Hourglass modules can be constructed). Prediction heatmaps are always constructed after each Hourglass module (shown in green in Fig. 3). The training loss includes weighted sum of losses for each of the heatmap predictions.

CU-Net [37] tries to improve Hourglass architecture not only in quality, but also in memory footprint and inference time. The authors note an importance of efficient architecture for use on mobile devices. Similarly to Hourglass, the network takes 256×256 image as an input and resizes it in preamble to 64×64, which remains the maximum resolution at which features are processed till the end of the network. To improve training and enable deeper CU-Net stacks, the authors propose to add skip connections not only between features of the same module, but also between different modules. To avoid excessive number of skip connections, a concept of Order-K coupling has been introduced in the paper. Order-K coupling denotes that skip connections will be added only K modules forward. In most cases, adding skip connections to one module forward ($K = 1$) seems sufficient. The authors decrease memory consumption and improve inference speed by avoiding unnecessary features copies, sharing memory, and quantizing both features and parameters. In addition, blocks with smaller number of features are used to decrease overall number of parameters. All these improvements have allowed to achieve similar to Hourglass accuracy on human pose estimation with only a fraction of parameters and higher inference speed. Exact number of parameters and inference times are shown in Table 2. However, despite the improvements, most recently introduced approaches prefer Hourglass architecture over CU-Net as will be shown later.

HRNet [36] has also been initially proposed for human pose estimation and then adapted to face landmark prediction in [38] and other works. This architecture significantly differs from the previous two. HRNet doesn't follow encoder-decoder architecture and doesn't use multiple stacks. Instead, parallel branches with different feature resolutions are maintained throughout the network. An illustration of HRNet architecture is shown in Fig. 4. Similarly to previous works, the network takes an image of size

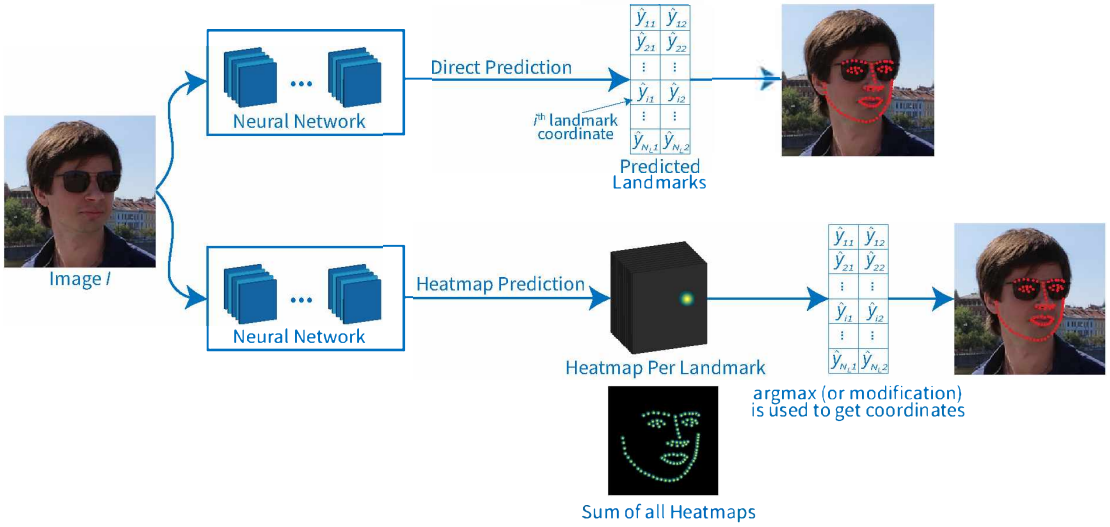


Figure 2: Direct landmark regression (upper row). The problem is solved in a form of regression, where actual landmark coordinates (x, y) are predicted directly by the algorithm. Heatmap-based (bottom row). The algorithm predicts probability distributions of landmark locations in a form of heatmaps. One heatmap per each of the landmarks is formed. Argmax (or its modification) is used to get each landmark coordinates.

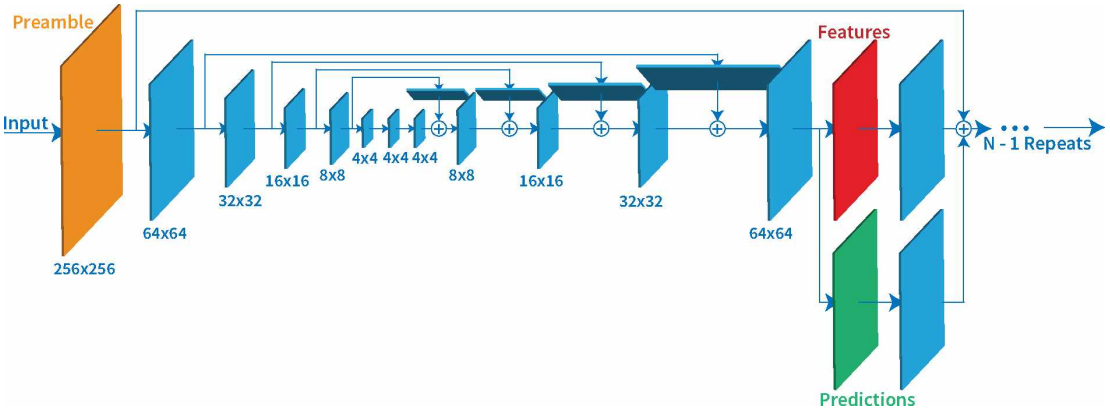


Figure 3: Hourglass architecture. Input image of size 256×256 is rescaled to 64×64 in Preamble (orange). Hourglass encoder-decoder module is to follow. First, Hourglass processes input via Residual blocks (blue boxes) and downscales features after each block (encoder part). Then in decoder features are upscaled after each Residual block. Additional skip connections between blocks of the same resolution are added to improve gradient propagation. Note, that encoder block's output is processed via an extra Residual block before signal is added to decoder features. Module features (in red) and predictions (in green) are formed after each Hourglass block. Then Hourglass block is repeated $N - 1$ times for a stack of N Hourglass modules. At training time not only final, but also intermediate predictions participate in loss function computation (the so-called intermediate supervision).

Table 2: Comparison of number of parameters and inference time for Hourglass and CU-Net architectures. [37]. Larger stacks still have reasonable inference time and small number of parameters, which is achieved thanks to memory sharing, quantization and smaller blocks.

Method	# Params (M)	Inference (ms)
4×Hourglass	25.5	48.9
8×CU-Net	7.9	36.1
16×CU-Net	15.9	70.8

256×256, which is then resized to 64×64 feature map in preamble. Next, the image is processed via convolutional blocks. Then another branch of resolution 32×32 is added. Note, that in contrast to previous works, 64×64 branch continues to be processed in parallel. The authors propose to exchange features between parallel branches. However, these feature maps are of different resolutions. To downscale feature map, strided convolution is used. To upscale feature map, nearest neighbor upsampling is used. Till the end of the network 4 parallel branches with different feature map resolutions are created. The final heatmap is generated at resolution of 64×64. At a similar number of parameters to a stack of 8 Hourglass modules, HRNet uses nearly twice fewer floating-point operations. Additionally, HRNet network width (number of convolutional channels) can be configured to change overall number of parameters and resulting inference speed.

We summarize backbone performance in Fig. 5, where we show number of floating-point operations in Fig. 5a (the greater is the number, the more time it takes to infer the network) and number of parameters in Fig. 5b (more parameters take more memory). Note, that Hourglass, CU-Net, HRNet require much more computation than other backbones, but have relatively small number of parameters. This is because these networks consider input at multiple resolutions and have to produce large heatmaps for each landmark. Other backbones (MobileNet, ShuffleNet and ResNet) process input at single scale and do not work with heatmaps.

Network backbones for each model considered in this survey are shown is summary Section 4.4, Table 9.

4.3 Facial Landmark Detection: Novel Algorithms and Ideas

In this section we present a description of recently introduced facial landmark detection algorithms. Each description is structured as follows: backbone and algorithm type are mentioned first, followed by explanation of novel ideas and approaches. The primary focus of this paper is on modern algorithms, covering years 2018 – 2021. We include some important older algorithm for completeness as well. The facial landmark algorithms covered in this section have been selected if: 1) the algorithm improves state-of-the art score established in the previous year; 2) ideas presented in the algorithm are then used in several following papers; 3) algorithm expands applicability of facial landmark detection or presents distinctive novel idea not discussed before. If only a slight modification is presented, that doesn't improve inference speed, quality or applicability of the algorithm, such algorithm is not included in this section. The algorithms are collected from various sections, including, but not limited to, top worldwide computer vision conferences. Overall 22 algorithm discussions are presented in this section.

Dense Face Alignment (DeFA) [39] is a shape-model-based approach. It uses custom-built convolutional neural network as a backbone. It is the only algorithm described in this section, where a neural network is used for facial landmark prediction through a deformable 3D face mesh. Algorithm is interesting in the following: 1) it allows to build a dense 3D face mesh using only a single 2D image. Mesh can be built for a wide range of poses and emotions (Fig. 6); 2) DeFA can be trained jointly on datasets with different number of landmarks, as landmarks are hooked as mesh constraints.

Style Aggregated Network (SAN) [40] is a heatmap-based approach, which is based on a modified ResNet-152 backbone. The authors have noticed style variability of photographs in 300W and AFLW datasets, which can be dark or light, colored or black & white. Existing to date algorithms were not accounting for that information. Furthermore, the authors have noticed that depending on style, prior algorithms were predicting facial landmark locations in slightly different places, with higher error on photographs with harsh lighting conditions. As a solution they have proposed: 1) to train Generative Adversarial Network (GAN) [41], namely CycleGAN [42] to transform images of different styles into neutral; 2) to train another neural network to predict landmarks from two inputs: style-neural and the original image. CycleGAN colorizes grayscale images and tones down bright colors. This makes all input images have a similar color distribution, which simplifies learning of face features by a neural network. Note, that style-neutral images produced by the proposed network are not always properly colorized and might containing artifacts, because-of that the authors propose to predict landmark jointly on the original and style-neutral images.

Look at Boundary (LAB) [5] is a combined heatmap and direct regression method. A stack of 4 Hourglass modules is used to predict boundary heatmap, from which another neural network predicts landmark matrix. The key advancement of this architecture is that the authors introduce face feature boundary heatmap, which is built as an intermediate representation between original image and predicted

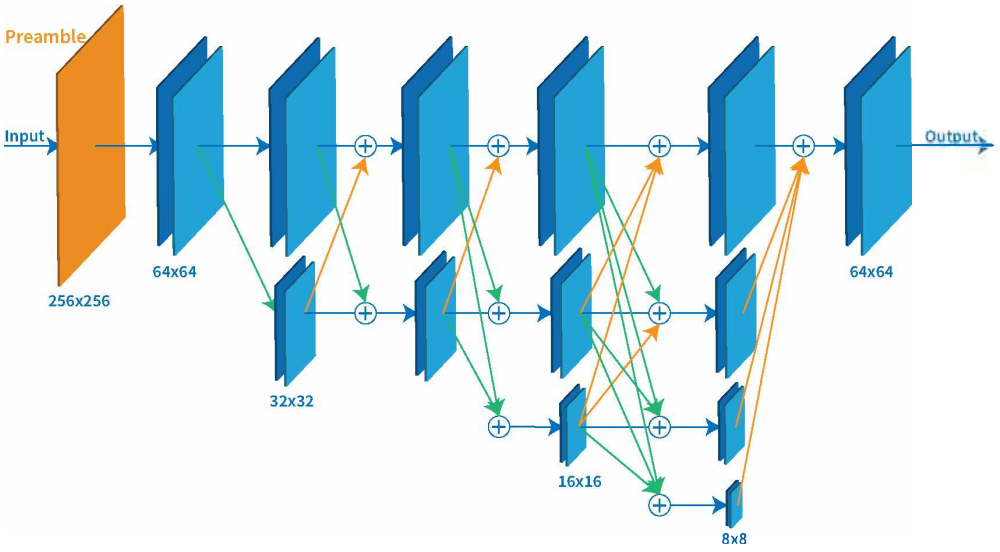
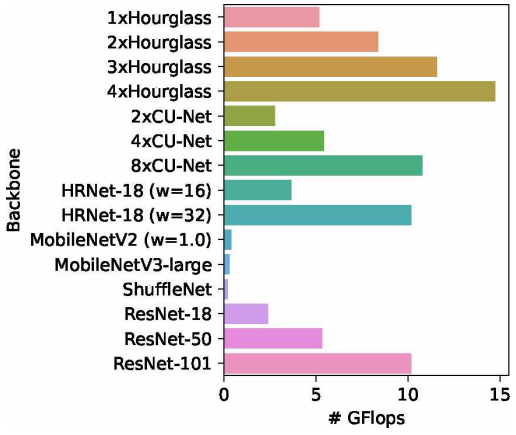
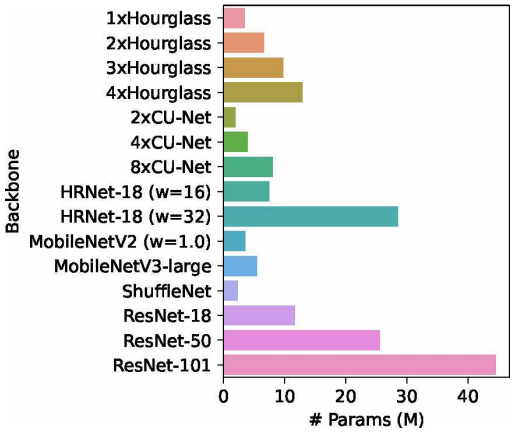


Figure 4: HRNet architecture. Input image of size 256×256 is rescaled to 64×64 in Preamble (orange). Next, input is processed at resolution of 64×64. After each set of convolutional blocks, a parallel branch is added with 4 times smaller resolution. Overall, 4 parallel branches are created till the network end, with the smallest resolution of 8×8. Features between blocks of different resolutions are exchanged. To pass features to blocks of higher resolution, nearest neighbor upsampling is used (orange arrows). To pass features to blocks of smaller resolution, strided convolution is used (green arrows). Blue arrows denote that feature map is not rescaled. The final output contains a sum of features of all scales.

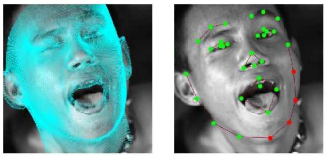


(a) Number of floating-point operations



(b) Number of parameters

Figure 5: Comparison of different backbones by the number of floating-point operations (a) and the number of parameters (b). Note that Hourglass, CU-Net, HRNet require much more computation, than other backbones, but have relatively small number of parameters. We use $S\times$ to denote a stack of S modules, $w = X$ to denote width multiplier of X.



(a) (b)

Figure 6: (a) DeFA dense 3D face mesh produced for different emotions and poses, (b) DeFA facial landmark predictions acquired from the face mesh. Importantly, varied number of facial landmarks can be produced from a single face mesh.[3]

landmarks (Fig. 7). Such a trick improves facial landmark prediction quality. Furthermore, it allows to train boundary estimation module on several datasets with different annotation schemes at once. After boundary module, another network predicts facial landmark coordinates. It should be noted, that only boundary submodule can be trained on datasets with different annotation schemes, while the landmark regression is trained for each dataset separately. Face structural information is modeled with the use of message passing [43], [44], that is, a graph-based way to model relationships. Boundary module is trained in adversarial (GAN) fashion. As the authors have shown, pretraining the boundary module on 300W improves prediction quality on AFLW and COFW datasets. Also, a novel facial landmark dataset was introduced in the work, namely WFLW.

(a) (b) (c)

Figure 7: LAB: (a) image to be labeled; (b) first module predicts intermediate boundary representation, that is common for different face landmark annotation schemes; (c) second module predicts actual facial landmark coordinates from boundary information [5].[4]

Wing Loss [29] is a direct regression approach. Several backbones were considered: custom-built CNN-6; two-stage approach, when CNN-6 produces coarse landmarks, and CNN-7 then refines them; ResNet-50 backbone. The authors note, that the field of loss functions for facial landmark detection problem is barely studied. Most researchers use $L2 = x^2/2$ as a loss function for direct regression, which is known to be sensitive to outliers. For that reason, some of prior works have used $smoothL1$ [45] loss instead. The authors make a comparison of $L2$ against other loss functions, such as $L1(x) = |x|$ and $smoothL1$, which is defined as:

$$smoothL1(x) = \begin{cases} x^2/2, & \text{if } x < 1 \\ |x| - 1/2, & \text{otherwise,} \end{cases} \quad (3)$$

and note that the latter two give better results. The main paper contribution is in introduction of a new loss, named *Wing loss*, which combines $L1$ for large landmark deviations and $\ln(\cdot)$ for medium and small:

$$wing(x) = \begin{cases} w \ln(1 + |x|/\varepsilon), & \text{if } |x| < w \\ |x| - C, & \text{otherwise,} \end{cases} \quad (4)$$

where $C = w - w \ln(1 + w/\varepsilon)$, w and ε are hyperparameters ($w = 15$, $\varepsilon = 3$ in paper). Visual comparison of loss functions is presented in Fig. 8.

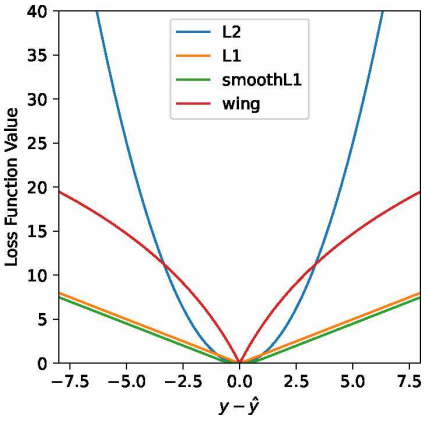


Figure 8: Loss function comparison: L2, L1, smoothL1, Wing (with $w = 15$, $\varepsilon = 3$). Note, that quadratic growth of L2 loss makes it sensitive to outliers. Thus, forcing the network to learn annotation errors. L2, L1 and smoothL1 yield very small values for small landmark location differences. This hinders network training, when network predictions are *almost* correct. In contrast, Wing is less sensitive to outliers and is much sensitive to medium-to-small errors, which improves training overall.

Also, to train more on hard examples the authors introduce PDB algorithm, which works as follows: 1) face rotation angle histogram is built; 2) rare examples (determined via the histogram) are duplicated with augmentations. As can be seen from Table 3, using CNN-6/7 cascade with $wing(\cdot)$ loss in combination with PDB substantially lowers the NME.

AVS [16] is a heatmap-based approach. Similarly to SAN, the authors have studied style in face landmark detection. They have proposed augmenting training set by changing image style via GAN image generation.

[3]Images are included under MIT license. Source: DeFA

[4]Author's written consent has been acquired to include these images.

Table 3: NME comparison of different loss functions on AFLW dataset. Note that Wing loss with PDB hard example mining has the best performance.

Network	L2	L1	smoothL1	Wing
CNN-6/7	2.06	1.82	1.84	1.71
CNN-6/7+PDB	1.94	1.73	1.76	**1.65**

The authors have trained ResNet-18, SAN and LAB methods on their extended training set, which resulted in better performance.

Practical Facial Landmark Detector (PFLD) [46] is a direct approach. MobileNetV2 backbone with full (1X) and quarter (0.25X) width has been considered. PFLD enables fast facial landmark detection directly on a mobile device. This is, to the best of our knowledge, the only modern neural-network-based algorithm, whose authors have shown that their algorithm can work efficiently on a mobile device. MobileNetV2 [2] is used as feature extractor in PFLD. Two heads are attached to it (Fig. 9): 1) facial landmark regression, where multi-scale fully-connected layer in the end of the head is used (lower branch); 2) 3D face model rotation angle estimator (yaw, pitch and roll), shown in upper branch of the figure. The second head contains a set of convolutional layers and is only used during training.

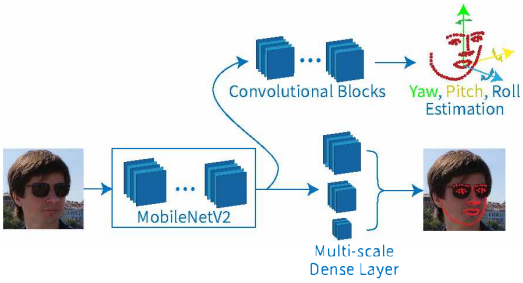


Figure 9: PFLD architecture. MobileNetV2 is used as feature extractor with multiple tasks: 1) to predict face landmark locations multi-scale fully-connected layer is used, which better captures image features at multiple scales (lower branch); 2) additional convolutional blocks are attached to MobileNetV2 for yaw, pitch, roll face rotation angle prediction (upper branch). Estimated angles are embedded into training loss to improve overall network performance. Estimation is not performed during network inference.

The most common datasets do not have information about 3D landmark coordinates. To get them the authors propose to 1) build a "mean" face representation containing 11 facial landmarks, based on the data in the training set; 2) estimate rotation matrix for each face between its and "mean" landmarks; 3) compute yaw, pitch, roll angles from the rotation matrix. According to the authors, such an approach is not very accurate for estimating angles, yet improves network training.

Furthermore, during training, the data is weighted based on image difficulty using a special loss function:

$$\mathscr{L} = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\sum_{c=1}^{C}\omega_n^c \cdot \sum_{k=1}^{K}(1-\cos\theta_n^k)\right)\|d_n^m\|_2^2 \tag{5}$$

where N is the number of facial landmarks, M is the number of training examples, $K = 3$, θ_1, θ_2, θ_3 are yaw, pitch, roll rotation angles of the above-described 3D face model, d_n^m represents difference vector between n^{th} predicted and ground true facial landmarks for m^{th} image; C is the number of complexity classes for face images (such as profile or frontal face, face-up, face-down, emotions or occlusion), ω_n^c is fraction of images in the corresponding complexity class to their total number M.

FAN [21] is a heatmap-based approach. A stack of 4 Hourglass modules is used. The authors modify Hourglass architecture by substituting Bottleneck block with hierarchical, parallel and multi-scale block with binary convolutions from [47]. 3 methods have been trained in the work: for 2D, 3D landmark detection, and 2D-to-3D model. 2D-to-3D model serves to transform 2D landmark representation into 3D. Interestingly, the inputs to the 2D-to-3D network are image and landmark heatmaps (one for each of the input 2D landmarks). The algorithm has not been tested on conventional 2D face landmark datasets. Thus, is missing from summary in Section 4.4. While binarized convolutions are stated to be faster, than conventional [47], no testing results have been presented. Architecture has been applied to landmark prediction in further works. Also, LS3D-W 3D face landmark dataset has been presented in this work.

AWing [7] is a heatmap-based algorithm, Hourglass is used as a backbone. Stacks from 1 to 4 modules have been considered. The algorithm is based on Wing loss, FAN, LAB papers, and CoordConv [48]. The authors have noticed, that $L2$ loss function does not produce sharp-enough heatmaps on difficult face images, because it is insensitive to small errors. In the meantime, the original Wing loss is inappropriate for heatmap-based detection as its gradient is discontinuous at the point of zero. In addition, each heatmap has a class imbalance. Only a few pixels on the map relate to the foreground class (meaning that landmark is likely to be at this point), while most parts are labeled as background class. The class imbalance is also not considered in the original Wing loss implementation. To solve all these issues, Adaptive Wing loss (Fig. 10) is introduced, which is 1) differentiable around zero; 2) accents small errors around foreground pixels, but not around background. Here we do not give the function formula due to its complexity. To predict foreground pixels even more precisely, the authors introduce a special weighted loss map, which additionally enhances sharpness of the facial landmark

heatmap.

MobileFAN [49] is a heatmap-based approach, based on modified MobileNetV2 backbone with 1X or 0.5X width. The authors examine network distillation approaches in order to reduce the number of model parameters and increase inference speed for heatmap-based methods. Note, that despite name similarity the approach is not based on FAN [21], discussed earlier.

Geometry Aggregated Network (GEAN) [50] is a heatmap-based approach, based on a stack of 4 Hourglass modules. The authors propose train- and test-time augmentation using Adversarial Attacks. Face adversarial attacks add noise or deformation to an image, so that face will not be recognized by face recognition system. To do that, face adversarial attack [51] warps the image to shift facial landmarks. The resulting face has slightly different shape, eye distance, etc. Hourglass is used to detect landmark locations on such deformed image. The detected landmarks now correspond to the warped face. However, we need to form a prediction for the original face. To do that, we shift landmark coordinates with warp deformation, that is opposite to the one introduced by the adversarial attack. Now the predicted landmarks correspond to the original face. Next, we follow this procedure for K random adversarial attacks. It turns out, that the predicted facial landmarks for each of the K images will be slightly different. Averaging such landmarks over all K images, results in accuracy improvement.

According to the authors, with respect to performance/quality ratio, it is the most beneficial to generate $K = 5$ adversarial examples for both training and testing. It is possible to use different number of adversarial images for training and testing. The authors have explored several modifications to the adversarial attack algorithm, and the best results are obtained when attack scale is set individually for each of the landmarks' semantical groups. The groups are assigned based on face region, such as nose, eyes, eye-brows, etc.

We have a deeper look at the concept of Adversarial Attacks in Section 6 of this paper.

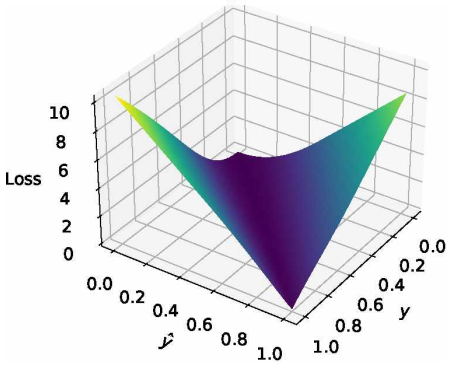


Figure 10: AWing surface plot. AWing accepts true y and predicted $\hat{y}$ heatmap probabilities. The function behaves as L2 loss for background pixels (when $y, \hat{y}$ are close to zero), and as Wing loss for foreground (when $y, \hat{y}$ are close to one), while preserving continuity. Thus, a greater weight is assigned to foreground pixels, resulting in sharper heatmaps and more accurate prediction.

IIRNetV2 [38] is a heatmap-based approach. In this work the original architecture of HRNet [36] has been improved and applied to the task of facial landmark prediction.

LUVLi [6] is a heatmap-based approach. This is the only algorithm with CU-Net backbone. A stack of 8 modules is used. The authors state, that facial landmark detection is used in many critically important applications. Thus, they propose a method to predict facial landmark visibility and algorithm confidence for each landmark. Cholesky Estimator Network (CEN) and the Visibility Estimator Network (VEN) are introduced for landmark and visibility predictions correspondingly. To increase heatmap precision, the authors use weighted spatial mean of heatmap's positive elements, instead of simple argmax. Also, a relabeled AFLW dataset with 68 landmarks and landmark visibility labels is presented.

Deep Adaptive Graph (DAG) [52] is a direct regression approach. Note, that landmarks here are predicted through a graph. Multiple backbones have been considered: VGG16 [53], ResNet50, 4×Hourglass, HRNet-18. HRNet-18 has shown the best results.

Face landmark prediction accuracy can be improved by taking into account structural information about human face. The authors propose a topology-adapting graph learning in a form of Graph Convolutional Network (GCN) cascade for facial and medical (e.g., hand, pelvis) landmark detection. In DAG algorithm graph $G = (V, E, F)$ is constructed, where V is a set of vertices, E is a set of edges, $F = \{f_1, f_2, \ldots, f_{|V|}\}$ is the so-called graph signal or graph features. Each vertex v corresponds to a single landmark. Each pair of vertices $(v_i, v_j)_{i \neq j}$ is connected via a weighted edge e_{ij}. The weights e_{ij} are learned during training process, they determine how information is propagated in a graph convolution. Larger weights denote stronger semantical connection between corresponding vertices. Graph convolution is defined as follows:

$$f^i_{k+1} = W_1 f^i_k + \sum_{j=1}^{|E|} e_{ij} W_2 f^j_k \tag{6}$$

where W_1 and W_2 are learnable parameter matrices. f^i_k is the feature computed for i^{th} vertex and k^{th} graph convolution.

Features F contain visual p_i and shape q_i features. Visual features are acquired from feature map H, that is produced by processing the whole image via convolutional neural network. Feature p_i, that corresponds to i^{th} vertex, is then acquired from H at a location

near the landmark. To obtain shape features q_i the authors compute displacement vectors for each pair of landmarks. Displacement information improves the algorithm performance, when face is partially occluded. In such cases, landmark locations can then be predicted from neighboring landmarks.

The landmark prediction process is conducted as follows: initial graph is constructed with mean weights computed over training set. Two separate GCNs are used for iterative graph transformation. GCN-Global is used to predict perspective transformation of the initial graph. GCN-Local is then applied multiple times to predict offsets for each of the landmarks for precise graph refinement.

The authors show, that in case of significant face occlusion their algorithm is better than the competition. In addition, the learned graph is good at capturing semantical information about human face, greater weights e_{ij} are learned for landmarks that appear physically closer to each other. For instance, edges between eyebrows have greater weight than edges between eyebrow and chin landmarks.

PropagationNet [54] is a heatmap-based algorithm, which uses a stack of 4 Hourglass modules as a backbone. The authors note importance of face boundary information for landmark prediction. Previous LAB approach used heavy generative adversarial network for boundary estimation. The authors of PropagationNet propose much simpler and faster approach: several convolutional blocks transform landmark heatmaps into boundary heatmaps after each Hourglass module. Boundary heatmaps serve as attention mask for intermediate predictions in Hourglass module to improve the final prediction accuracy. In addition, Hourglass module has modifications from FAN, and is extended with CoordConv [48] and Anti-aliased CNN [55].

Also, the authors introduce Focal Wing Loss, an extension of Adaptive Wing loss, that assigns greater weights to image scenes less presented in the current training batch. The examples of image scenes are large head pose, exaggerated expression, etc. The focus function $\sigma_n^{(c)}$ for class c and sample n is defined as:

$$\sigma_n^{(c)} = \begin{cases} 1, & \text{if } \sum_{n=1}^{N} s_n^{(c)} = 0 \\ \frac{N}{\sum_{n=1}^{N} s_n^{(c)}}, & \text{otherwise} \end{cases} \quad (7)$$

where $s_n^{(c)} = 0$, when the sample n does not belong to class c; and $s_n^{(c)} = 1$, otherwise. Image scene annotations exist in WFLW, but not in 300W and COFW. The authors have annotated images of 300W and COFW by themselves. To form the final loss, image focus coefficient $\sigma_n^{(c)}$ is multiplied by conventional Adaptive Wing loss.

SAAT [14] is a heatmap-based approach, which uses a stack of 2 Hourglass modules as a backbone. The authors propose augmenting the training set with adversarial images. The network architecture is left unchanged. In contrast to GEAN, only training procedure is modified, no artificially changed images are generated at test-time. Conditional GAN is used to perform the adversarial attack.

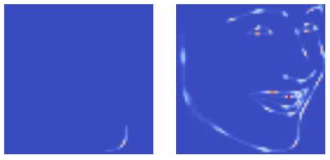
LDDMM-Face [56] is a shape model approach, which uses HRNet-18 as a backbone. The primary focus of the work is on cross-dataset and sparse-to-dense annotation. Sparse-to-dense means, that the network can be trained on sparse face landmarks, and then it predicts dense landmark annotations. The authors estimate shape model deformation via large deformation diffeomorphic metric mapping (LDDMM) [57], [58] method. While cross-dataset landmark annotation is out of scope of this survey, this method is also capable of achieving good results for classical face landmark datasets.

AnchorFace [59] is a direct regression method. Two modified backbones have been considered: ShuffleNet-V2 (with faster inference), HRNet-18 (with better accuracy). HRNet results are present only for WFLW dataset. The authors tackle the problem of landmark prediction for images with large pose. For that they propose to configure a set of anchor templates for faces with different poses. Anchor templates are configured either manually or via KMeans clusterization on the dataset. Then the templates are refined with a network that predicts offsets and confidence of each of the anchor templates.

PIPNet [60] is a combined heatmap and direct regression approach. Several backbones have been considered: MobileNetV2, MobileNetV3, ResNet-18, ResNet-101, etc. In PIPNet it has been noted, that heatmap-based methods have high computational cost, but good accuracy. To alleviate the cost, the authors propose 3-head network. First head predicts coarse landmark heatmaps at lower than usual resolution. Second head predicts regression offsets. Thus, fine-tuning heatmap-based predictions. Third fine-tunes landmark predictions further by regressing offsets relative to the neighboring landmarks. All the heads share the same backbone and are computed in parallel. Additionally, "self-learning with curriculum" method has been introduced, where the authors try to learn on 300-W and then self-learn on other facial landmark datasets.

ADNet [61] is a heatmap-based approach. It uses a stack of 4 Hourglass modules as a backbone. The work is based on LAB and PropagationNet. Facial landmark datasets are annotated by humans. Thus, there exists some annotation error. It turns out, that error in tangent direction (relative to face boundary) is much higher, than in normal direction. Loss functions of existing algorithms do not account for such difference in annotation error. To mitigate the problem, anisotropic direction loss (ADL) is introduced, where higher weight is assigned to normal error. Also, Point-Edge heatmaps are presented, that are used as attention mask. Point-Edge heatmaps are predicted after each Hourglass module and have greater than zero values

around face boundary corresponding to a landmark. This is shown in Fig. 11a. Note landmark center accentuation shown in red. Sum of all Point-Edge heatmaps forms face boundary information (Fig. 11b). Landmark locations are obtained through soft-argmax. The final loss function that is used to train the model consists of a sum of AWing loss for Point and Edge heatmaps, as well as ADL loss for landmark heatmaps.



(a) (b)

Figure 11: ADNet learns additional semantical information about face in a form of Point-Edge heatmaps. Each of the heatmaps corresponds to face landmark and a part of face boundary (a). Sum of all Point-Edge heatmaps forms face boundary (b). Such heatmaps are predicted after each Hourglass module and are used as attention masks to improve overall network performance.[5]

IIIII [62] and **SubpixelHeatmap** [63] are heatmap-based approaches. Both use a stack of 2 Hourglass modules as backbones. The algorithms focus on reducing heatmap quantization error. Input image and landmark annotations are of resolution 256×256. However, the heatmap is typically of size 64×64 per landmark, which is $1/16^{\text{th}}$ of the source image resolution. The landmark location is then found using argmax. The process of mapping floating point landmark location to a discrete grid is called quantization.

HIH. The authors tackle the problem by splitting heatmaps into *integer* and *decimal*. Integer heatmap is predicted via the usual heatmap-based facial landmark prediction pipeline. Then another decimal heatmap block predicts a precise offset to the quantized landmark locations. Two ways to predict the offset have been considered: based on Convolutional Neural Networks and Transformers [64], denoted as HIH_C and HIH_T correspondingly.

SubpixelHeatmap. To alleviate the above-described quantization problem, the authors propose a different approach, namely local soft-argmax computation. For a given heatmap H_k, k^{th} landmark location is first found as usual via: $(\hat{y}_k^{[1]}, \hat{y}_k^{[2]}) = \arg\max H_k$ and then refined via local soft-argmax over the neighboring patch $d \times d$:

$$(\Delta\hat{y}_k^{[1]}, \Delta\hat{y}_k^{[2]}) = \sum_{m,n} \texttt{softmax}(\tau H_k)_{m,n}(m,n), \qquad (8)$$

where $\tau = 10$ is the temperature, $d = 5$ is the suggested region size. Then the final landmark location is found using: $(\hat{y}_k^{[1]} + \Delta\hat{y}_k^{[1]} - l, \hat{y}_k^{[2]} + \Delta\hat{y}_k^{[2]} - l)$, where $\hat{y}_k^{[1]}, \hat{y}_k^{[2]}$ denote landmark position on X and Y axes correspondingly, and $l = d/2$. The authors compare this approach to global soft-argmax (i.e., computed over the whole image) applied to pose estimation in [65], and state that local soft-argmax yields much better results.

Also, the authors apply test-time augmentation to improve network performance. They feed 2 images with different random augmentations (T_0, T_1) through the network Φ aggregating the final heatmap as follows: $H = T_0^{-1}(\Phi(T_0(\mathbf{X}_i), \theta)) + T_1^{-1}(\Phi(T_1(\mathbf{X}_i)), \theta)$, where θ is the network Φ parameter matrix. In addition, Hourglass architecture has been modified following FAN algorithm.

4.4 Facial Landmark Detection Algorithms: Summary

In this section we present and discuss facial landmark detection algorithm performance on the most widely used datasets: 300W, AFLW, COFW and WFLW. We summarize backbones used, and inference times on desktop and mobile devices. We present brief summary of contributions of each facial landmark detection method. We conclude this section with analysis of algorithm performance by years and per algorithm type.

Tables 4 to 8 present facial landmark detection method metrics on the most common datasets. The the best result is shown in red, second best is shown in blue. Metrics in the tables include NME (%), Failure Rate (FR, %) and CED-AUC. Table 9 has method backbones and inference times listed. Different hardware was used for algorithm inference speed measurements. So instead of defining first and second fastest, we show all algorithms that perform faster than 60 frames per second (17 ms) in green. Note, that in addition to face detection, other algorithms will need to be executed, that is why the threshold is so strict. The tables are filled based on the results presented in the corresponding papers. If the result was published later, the metric's source is shown in square brackets. Table 10 has a brief summary of algorithm novelties proposed in each paper.

We present 300W dataset results normalized by both inter-ocular distance in Table 4 and inter-pupil in Table 5. Metrics are split into common, challenge, and full as per protocol. Can be noticed, most novel algorithms use inter-ocular distance normalization. As is shown in Table 4, error on challenging subset is still quite high (3.99 %) and is significantly higher than the best common subset error (2.53 %). From Table 5 we note that Wing neural-network-based algorithm with ResNet-50 backbone is 1.7 times better, than regression-tree-based ERT.

AFLW results are shown in Table 6. NME (%) normalized by face bounding box diagonal is used to present the results. Errors are on average smaller than on 300W possibly as AFLW has fewer landmark to

[5] Based on this source. Contains representative samples of Point-Edge heatmaps. Distributed under CC BY-NC-SA 4.0 License.

Table 4: Face landmark detection normalized mean error (NME) on 300-W dataset. Inter-ocular normalization is used. The best result is shown in red, second best in blue. Note, that significant qualitative improvement has been achieved over the past few years, but still Challenge subset error is quite high.

Model	Year	Common	Challenge	Full
DeFA [39]	2017	5.37	9.38	6.10
SAN [40]	2018	3.34	6.60	3.98
LAB [5]		2.98	5.19	3.49
AVS [16]	2019	3.21	6.49	3.86
PFLD 0.25X [46]		3.03	5.15	3.45
PFLD 1X		3.01	5.08	3.40
PFLD 1X+ (extra data)		2.96	4.98	3.37
AWing-1HG [7]		2.81	4.72	3.18
AWing-2HG		2.77	4.58	3.12
AWing-3HG		2.73	4.58	3.10
AWing		2.72	4.52	3.07
MobileFAN (0.5) [49]	2020	4.22	6.87	4.74
MobileFAN		2.98	5.34	3.45
GEAN (extra data) [50]		2.68	4.71	3.05
HRNetV2 [38]		2.87	5.15	3.32
LUVLi [6]		2.76	5.16	3.23
DAG [52]		2.62	4.77	3.04
PropagationNet [54]		2.67	3.99	2.93
SAAT [14]	2021	2.87	5.03	3.29
LDDMM-Face [56]		3.07	5.40	3.53
AnchorFace [59]		3.12	6.19	3.72
PIPNet (MobileNetV2) [60]		2.94	5.30	3.40
PIPNet (MobileNetV3)		2.94	5.07	3.36
PIPNet (ResNet-18)		2.91	5.18	3.36
PIPNet (ResNet-101)		2.78	4.89	3.19
ADNet [61]		2.53	4.58	2.93
HIH_C [62]		2.95	5.04	3.36
HIH_T		2.93	5.00	3.33
SubpixelHeatmap [63]		2.61	4.13	2.94

Table 5: Face landmark detection normalized mean error (NME) on 300-W dataset. Inter-pupil normalization is used. The best result is shown in red, second best in blue. Note substantial error decrease of recently introduced neural-network-based approaches over regression tree-based (ERT) algorithm.

Model	Year	Common	Challenge	Full
ERT [26]	2014	-	-	6.40 [29]
LAB [5]	2018	3.42	6.98	4.12
Wing (CNN-6) [29]		3.35	7.20	4.10
Wing (CNN-6/7)		3.27	7.18	4.04
Wing (ResNet-50)		3.01	6.01	3.60
PFLD 0.25X [46]	2019	3.38	6.83	4.02
PFLD 1X		3.32	6.56	3.95
PFLD 1X+ (extra data)		3.17	6.33	3.76
AWing [7]		3.77	6.52	4.31
DAG [52]	2020	3.64	6.88	4.27
PropagationNet [54]		3.70	5.75	4.10
ADNet [61]	2021	3.51	6.47	4.08

be annotated (21 vs 68 in 300W), and due to different normalization. Face diagonal is larger than inter-ocular distance.

Table 6: Face landmark detection normalized mean error (NME) on AFLW. Normalization by face bounding box diagonal is used. The best result is shown in red, second best in blue.

Model	Full	Frontal
ERT	4.35 [40]	2.75 [40]
SAN	1.91	1.85
LAB	1.85	1.62
LAB (extra data)	1.25	1.14
Wing (CNN-6)	1.83	-
Wing (CNN-6/7)	1.65	-
Wing (ResNet-50)	1.47	-
PFLD 0.25X	2.07	-
PFLD 1X	1.88	-
AWing	1.53	1.38
GEAN (extra data)	1.59	1.34
HRNetV2	1.57	1.46
LUVLi	1.39	1.19
AnchorFace	1.56	1.38
PIPNet (MobileNetV2)	1.52	-
PIPNet (MobileNetV3)	1.52	-
PIPNet (ResNet-18)	1.48	-
PIPNet (ResNet-101)	1.42	-
SubpixelHeatmap	1.31	1.12

COFW results are presented in Table 7. The results in papers are presented either normalized by inter-pupil or inter-ocular (majority) distance, but not both, which makes direct comparison more difficult. NME (%) and Failure Rate (FR, %) are used to present the results. Interestingly, novel approaches have FR equal to 0.0, which means that no images in the test set have NME above 10% as follows from Eq. 2.

WFLW is the most recent and interesting dataset in this survey. Results are presented in Table 8. We show NME (%), failure rate (FR, %) and CED-AUC (denoted as AUC in the table) for the whole test set. Note, that lower values of NME and FR are better. In contrast, higher AUC values are better. We also present errors on all types of challenging image categories present in WFLW dataset: Pose, Expression (Expr.), Illumination (Ill.), Make-Up (M.U.), Occlusion (Occ.) and Blur. In Fig. 12 box plots for different image categories are shown. The plot is based on NME for all algorithms present in Table 8. We note significant difference in NME for different subsets. The most significant challenge to the landmark detection datasets comes from large pose (best error is still at 6.56 %), followed by occlusion (4.36 %) and blur (4.21 %). In contrast, from make-up (3.62 %), illumination (3.87 %) and expression (3.87 %) comes the least challenge. Unlike COFW, failure rate is still quite high (1.55 %) on the test set. While a factor of 1.6

Table 7: Face landmark detection normalized mean error (NME) and failure rate (FR) on COFW. The best result is shown in red, second best in blue.

Model	NME (%)	FR (%)
Inter-pupil normalization		
Wing	5.44 [7]	3.75 [7]
AWing	4.94	0.99
PropagationNet	3.71	0.20
ADNet	4.68	0.59
Inter-ocular normalization		
LAB	5.58	2.76
LAB (extra data)	3.92	0.39
Wing (ResNet-50)	5.07 [60]	-
MobileFAN (0.5)	3.68	0.59
MobileFAN	3.66	0.59
HRNetV2	3.45	0.19
PIPNet (MobileNetV2)	3.43	-
PIPNet (MobileNetV3)	3.40	-
PIPNet (ResNet-18)	3.31	-
PIPNet (ResNet-101)	3.08	-
HIH_C	3.29	0.0
HIH_T	3.28	0.0
SubpixelHeatmap	3.02	0.0

improvement has been achieved on large pose subset over the past 3 years, further improvement is welcome. We see this dataset as the one posing the most interest for novel research.

Due to complexity of manual dense facial landmark annotation, the datasets are quite small. Thus, additional training data has a significant impact on model accuracy. We group additional data used into 3 main groups: 1) backbone pretraining on ImageNet; 2) usage of extra image labels (such as image scene); 3) pretraining on other face-related datasets. SAN, DAG, PIPNet, Wing (ResNet-50 only) state that they use ImageNet-pretrained backbones. Hence, they are related to the first group. Second group with PFLD, PropagationNet, annotates images manually with categories (i.e., significant pose, emotion) to assign higher weights to rare categories. Also, PropagationNet and ADNet (focal loss modification only) use weighted loss based on image classes directly available from WFLW dataset. The final third group of algorithms uses extra face-related data during training. GEAN uses pretrained face recognizer to perform an adversarial attack on; certain modifications of LAB use pretrained boundary module on 300W dataset and report results on COFW and AFLW; PFLD (1X+ modification only) is pretrained on WFLW and then reported on 300W. We denote face-data-based pretraining (3^{rd} category) with *extra data* label. We do not highlight such results as the best result; however, the data is still present in Tables 4 to 8. Note the significant positive impact of LAB boundary module pretraining for AFLW and COFW dataset performance in Tables 6

Table 8: Face landmark detection normalized mean error (NME), failure rate (FR), CED-AUC on WFLW. Normalization by inter-ocular distance is used. Results are presented both for the whole test set and for subsets that focus on unusual Pose, Expression (Expr.), Illumination (Ill.), Make-Up (M.U.), Occlusion (Occ.) and Blur. The best result is shown in red, second best in blue.

	Test set			Subsets (NME %, ↓)					
Model	NME %, ↓	FR %, ↓	AUC ↑	Pose	Expr.	Ill.	M.U.	Occ.	Blur
LAB	5.27	7.56	0.5323	10.24	5.51	5.23	5.15	6.79	6.32
Wing (tested in [7])	5.11	6.00	0.5504	8.75	5.36	4.93	5.41	6.37	5.81
AVS	4.39	4.08	0.5913	8.42	4.68	4.24	4.37	5.60	4.86
AWing	4.36	2.84	0.5719	7.38	4.58	4.32	4.27	5.19	4.96
MobileFAN (0.5)	5.59	6.72	0.4682	9.68	5.98	5.45	5.33	6.49	6.31
MobileFAN	4.93	5.32	0.5296	8.72	5.27	4.93	4.70	5.94	5.73
HRNetV2	4.60	-	-	7.94	4.85	4.55	4.29	5.44	5.42
LUVLi	4.37	3.12	0.577	-	-	-	-	-	-
DAG	4.21	3.04	0.5893	7.36	4.49	4.12	4.05	4.98	4.82
PropagationNet	4.05	2.96	0.6158	6.92	3.87	4.07	3.76	4.58	4.36
LDDMM-Face	4.63	3.68	0.5509	-	-	-	-	-	-
AnchorFace	4.62	4.20	0.5516	-	-	-	-	-	-
AnchorFace (HRNet-18)	4.32	2.96	0.5769	-	-	-	-	-	-
SAAT	5.11	5.63	0.5633	-	-	-	-	-	-
PIPNet (MobileNetV2)	4.79	-	-	8.76	4.86	4.56	4.60	6.04	5.53
PIPNet (MobileNetV3)	4.65	-	-	8.22	4.75	4.49	4.46	5.72	5.31
PIPNet (ResNet-18)	4.57	-	-	8.02	4.73	4.39	4.38	5.66	5.25
PIPNet (ResNet-101)	4.31	-	-	7.51	4.44	4.19	4.02	5.36	5.02
ADNet	4.14	2.72	0.6022	6.96	4.38	4.09	4.05	5.06	4.79
ADNet (focal loss)	3.98	2.00	0.6250	6.56	4.02	3.87	3.62	4.36	4.21
HIH_C	4.18	2.96	0.597	7.20	4.19	4.45	3.97	5.00	4.81
HIH_T	4.21	2.84	0.593	7.20	4.28	4.42	4.03	5.00	4.79
SubpixelHeatmap	3.72	1.55	0.631	-	-	-	-	-	-

and 7 correspondingly.

In Table 9 we present algorithm backbones, number of network parameters, floating-point operations, and inference times on desktop computers (CPU, GPU) and mobile phones. Hourglass and CU-Net backbones are typically stacked. We use $N\times$Hourglass to denote a stack of N Hourglass modules. Number of parameters translates to device memory consumption, which is especially important for mobile and edge devices. Gigaflops (GFlops) is a number of floating-point operations needed for network inference, which determines a requirement for device performance. We also present an estimate of algorithms' inference time on desktop Central Processing Unit (CPU), Graphical Processing Unit (GPU) and mobile phones, as measured by the authors themselves. Note that different hardware has been used for the experiments, affecting measurements. Unfortunately, most of the algorithms report only GPU inference speed, and 11 out of 22 reviewed algorithms do not report any speed measurements at all. While state-of-the-art approaches seem to be very computationally intensive, there are still lightweight models that are quite accurate. Hourglass backbone is the most pervasive across modern landmark detection methods (used in 9 out of 22 cases). Only one algorithm (PFLD) has been adapted to a mobile device and can run there at real-time speed. We expect several of the fastest algorithms, like Wing (CNN6), MobileFAN (0.5), PIPNet (ResNet-18), and possibly AWing ($1\times$Hourglass) to be applicable to mobile devices as well. In general, we would like to see a larger number of fast approaches in future.

The final Table 10 presents a summary of facial landmark detection methods, where we show algorithm type, main contribution and notes on algorithm applicability and performance. We use the following abbreviations for algorithm type: D is direct regression, H is heatmap-based regression, SM is shape model, H + D indicates combined methods that use both heatmap and direct regression at different stages. Note that all of the recent neural-network-based facial landmark detection algorithms clearly show, that information explicitly present in the dataset is insufficient. To solve this problem several approaches are proposed:

- use of an auxiliary representation, which contains structural information about the face, such as: 3D face mesh (DeFA); deformable shape model (LDDMM-Face); graph-based message passing (LAB); yaw, pitch, roll rotation angles (PFLD); landmark visibility (LUVLi); face representation as a graph model (DAG); offsets to anchors defined for faces with different poses (AnchorFace)

Table 9: Comparison of neural network backbones, computational complexity and inference speed of facial landmark detection algorithms. The smallest number of parameters and floating-point operations (flops) is shown in red, second best in blue. Inference times of less than 17 ms (or more than 60 frames per second) are shown in green.

Model	Backbone	# Params (M)	# GFlops	CPU (ms)	GPU (ms)	Mobile (ms)
ERT	-	-	-	1	-	-
DeFA	CNN	-	-	-	-	-
SAN	ResNet-152	-	-	-	343 [46]	-
LAB	4×Hourglass	25.1 [49]	18.85 [54]	2600 [46]	60	-
Wing	CNN-6	3.8	-	6.7	2.5	-
Wing	CNN-6/7	12.3	-	50	5.9	-
Wing	ResNet-50	25	5.5 [60]	125	33.3	-
AVS	ResNet-152	35.02 [62]	33.87 [62]	-	-	-
PFLD 0.25X	MobileNetV2	-	-	1.2	1.2	7.0
PFLD 1X/1X+	MobileNetV2	-	-	6.1	3.5	26.4
FAN	4×Hourglass	24	-	-	33.3	-
AWing-1HG	1×Hourglass	-	-	-	8.3	-
AWing-2HG	2×Hourglass	-	-	-	15.7	-
AWing-3HG	3×Hourglass	-	-	-	22.1	-
AWing	4×Hourglass	24.15 [54]	26.79 [59]	-	29.0	-
MobileFAN (0.5)	MobileNetV2	1.84	0.45	-	4.0	-
MobileFAN	MobileNetV2	2.02	0.72	-	4.2	-
GEAN	4×Hourglass	-	-	-	58.8	-
HRNetV2	HRNet-18	9.3	4.3	-	-	-
LUVLi	8×CU-Net	-	-	-	17	-
DAG	HRNet-18	-	-	-	-	-
PropagationNet	4×Hourglass	36.30	42.83	-	-	-
SAAT	2×Hourglass	-	-	-	-	-
LDDMM-Face	HRNet-18	-	-	-	-	-
AnchorFace	ShuffleNet-V2	-	1.71	-	22.2	-
AnchorFace	HRNet-18	-	5.30	-	-	-
PIPNet	MobileNetV2	4.2	0.5	33.9	8.3	-
PIPNet	MobileNetV3	4.5	0.4	35.2	12.5	-
PIPNet	ResNet-18	12.0	2.4	28.0	5.0	-
PIPNet	ResNet-101	45.7	10.5	113.6	17.9	-
ADNet	4×Hourglass	13.37	17.04	-	95.29	-
$HIIH_C$	2×Hourglass	14.47	10.38	-	-	-
HIH_T	2×Hourglass	28.18	10.29	-	-	-
SubpixelHeatmap	2×Hourglass	-	-	-	-	-

or offset regression from neighboring landmarks (PIP);

- boundary representation either explicitly (LAB) or via attention module (PropagationNet, ADNet);
- hard example mining during training. Different variations on the theme have been presented in Wing, PFLD and PropagationNet papers;
- aggregating predictions for multiple input images: with style modification (SAN); after adversarial attack (GEAN); or with several different augmentations (SubpixelHeatmap). As it has been noted, minor changes in image might result in major shift in landmark prediction, averaging such predictions results in improved accuracy;
- train set augmentation using style (AVS) or adversarial attacks (SAAT);
- reduction of very large errors (outliers) and increased contribution of small to medium-sized errors (to better refine predictions): Wing, AWing and derivate works;
- subpixel heatmap precision (reducing heatmap quantization error): weighted argmax (LUVLI), joint heatmap and direct regression (PIPNet), global soft-argmax (ADNet), CNN- or Transformer-based refinement (HIH), local soft-argmax (SubpixelHeatmap).

In Figs. 13 to 16 we present visual summary of the above-described tables, and discuss algorithms and datasets. Note, that for consistent comparison between

Table 10: Face landmark detection method brief summary. The following abbreviations are used for algorithm type: D for direct regression, H for heatmap, SM for shape model, H + D indicates combined heatmap and direct methods.

Model	Type	Main Contribution	Notes
ERT	D	First use of Ensemble of Regression Trees	Fast on CPU. Mediocre quality
DeFA	SM	3D face mesh for faces with large pose and occlusion	Can train on datasets with different annotation schemes
SAN	H	Style neutralization	Good for landmark prediction under extreme lighting. Has slow inference
LAB	H+D	Boundary intermediate representation	Boundary module can be trained on datasets with different annotation schemes
AVS	H	Dataset augmentation via styled image generation	Idea is applicable to many methods
Wing	D	Special loss for direct regression	Training reduces small-to-medium errors. Can be adapted to any direct regression model
PFLD	D	Novel train loss and face angle prediction scheme	Good speed/quality ratio. The only method tested on a mobile device
FAN	H	Binarized convolutions for landmark prediction	Lacks testing on widespread face landmark datasets. Used in derivative works
AWing	H	Special loss for heatmap regression	Produces sharper heatmaps
MobileFAN	H	Network distillation	One of the fastest heatmap regression methods. Not tested on mobile devices
GEAN	H	Train/test-time augmentation using adversarial attack on face landmarks	Consolidates knowledge from face recognition model and landmark detection datasets. Requires multiple passes over the main network
HRNetV2	H	Improvement of HRNet architecture	HRNet reduces required computation over standard Hourglass
LUVLi	H	Prediction uncertainty estimation	Predicts landmark location and confidence jointly. Learning prediction confidence requires special dataset annotation. Application of CU-Net backbone for face landmark prediction
DAG	D	Topology-Adapting Graph Convolutional Network cascade	Captures face structural information. Good prediction for complicated pictures
PropagationNet	H	Boundary attention module. Focal Wing Loss	SOTA on 300W
SAAT	H	GAN and adversarial-attack-based training image generation	The idea can be applied to any method
LDDMM-Face	SM	LDDMM shape model with deep neural networks	Good dense landmark prediction after training on sparse landmark annotation
AnchorFace	D	2-step prediction: anchor estimation; refinement with regression offsets and confidence scores	While the backbone is lightweight, inference time is still high
PIPNet	H+D	Coarse heatmap refined via direct regression	In theory, faster inference time with good quality. In practice, time is still high
ADNet	H	Point-edge heatmaps. Separate tangent/normal errors	SOTA on multiple datasets
HIH	H	Reducing heatmap quantization error via nested heatmaps	Simpler implementations (like SubpixelHeatmap) seem to work better
SubpixelHeatmap	H	Reducing heatmap quantization error via local soft-argmax	SOTA results on many benchmarks. Requires multiple passes over the network

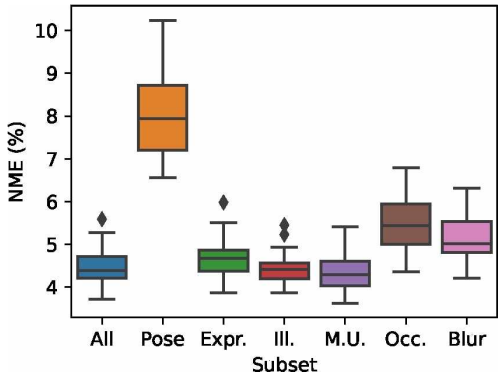


Figure 12: Box plots that show Normalized Mean Error (NME, %) on WFLW dataset for algorithms from Table 8. The results are shown for whole test set (All), as well as for subsets that focus on unusual Pose, Expression (Expr.), Illumination (Ill.), Make-Up (M.U.), Occlusion (Occ.) and Blur. Images with large pose, occlusion and blur are more challenging, than others.

datasets we have selected results with inter-ocular normalization, such results are present for 300W, COFW and WFLW datasets. Unfortunately, AFLW protocol specifies only face box diagonal normalization, which makes impossible direct comparison with other datasets. Because-of that we compare it separately. As usual, we do not include results with extra training data used.

In Fig. 13 we show the best normalized mean error achieved by each type of algorithms: direct, heatmap-based, combined heatmap and direct, shape-model-based. All of these algorithms are based on neural networks. Direct and heatmap-based approaches have nearly the same performance on 300W (Full) dataset, with a slight advantage of heatmap-based approaches. In contrast, heatmap-based approaches show WFLW performance significantly better, than direct regression algorithms. The best direct NME of 4.21 % is achieved by DAG algorithm (direct prediction based on graphs), the best heatmap-based NME is much lower at 3.72 %. This is also state-of-the-art result, that is achieved by SubpixelHeatmap. Note, significantly higher COFW error for direct approaches. This happens because only a single direct regression algorithm has been tested on COFW, that is Wing algorithm. The approach is quite old. Hence, this error spike should be considered as an outlier. Next, we take a look at combined heatmap-direct approaches. There are only two of them, LAB (which is quite old) and PIPNet. PIPNet proposed to predict coarse heatmap and then refine it via direct regression. While the idea is quite promising, the algorithm offers worse performance than direct or heatmap-based approaches. Finally, neural-network-based shape model algorithms offer the worst performance. Such approaches are quite useful, when network training is performed on images with different annotation schemes, or when 3D face mesh is required. Note, in this survey only DeFa algorithm produces 3D mesh. However, for most other use cases such approaches should not be considered.

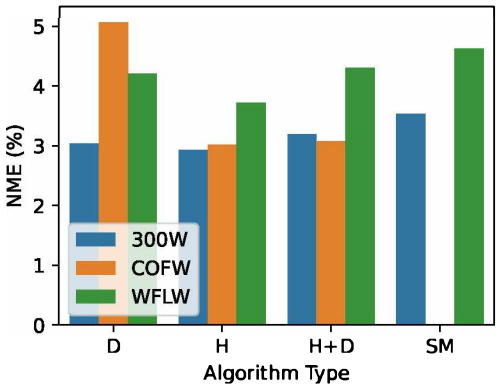


Figure 13: The best normalized mean error for each algorithm type: direct (D), heatmap-based (H), combined heatmap and direct (H+D), shape-model-based (SM). Heatmap-based approaches offer the best quality. Note, similar performance of direct and heatmap-based approaches on 300W.

In Fig. 14 we show the best inference time achieved by algorithms of each type. We do not visualize results for shape-model-based approaches as no timings have been presented in the corresponding papers. As expected, the fastest approaches use direct regression. They typically have more lightweight backbones. Also, they do not need to predict large heatmaps for each of the landmarks, which saves computation. Heatmap-based approaches have the best GPU inference time at 4.0 ms (achieved by MobileFan (0.5)), which is worse than 1.2 ms achieved by direct approach (PFLD 0.25X). While the timings seem to be quite small, note that both approaches offer significantly lower accuracy, than state-of-the-art algorithms. State-of-the-art algorithms still execute around 100 ms on GPU. Only GPU timings are available for heatmap-based approaches. Finally, we show timings for combined heatmap and direct regression methods. The best results achieved by PIPNet with ResNet-18 backbone. While the backbone is quite lightweight, inference time of 28.0 ms on CPU and 5.0 ms on GPU for this model is quite high. We would have expected the timings to be lower here.

In Fig. 15 we show the best algorithm performance grouped by dataset and year. To begin with, we discuss 300W dataset results, which are presented for Full and Challenge sets. We note that no progress has been made in 2021 in comparison to 2020. We expect error of all algorithms to stop decreasing at some point, as both training and test sets contain annotation errors. Thus, it would be interesting to see, whether

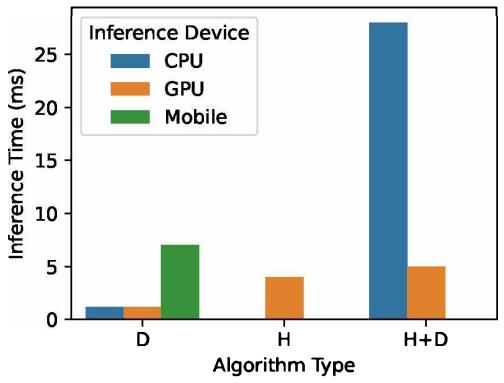


Figure 14: The best inference time for each algorithm type: direct (D), heatmap-based (H), combined heatmap and direct (H+D). Results are shown for different inference devices: CPU, GPU and Mobile. Direct regression algorithms have the best speed. Mixed H+D algorithm time is unexpectedly high.

any progress is made in 2022, or the remaining algorithm error is due to incorrect annotation. On COFW significant progress has been made over years. And WFLW dataset is still the most challenging. WFLW dataset has been introduced in 2018, and the largest improvement has been achieved in the following year, which is especially obvious on Pose subset, that has the highest normalized mean error. As discussed before, annotation of faces with unusual Make-Up has the least challenge to facial landmark algorithms, but NME has decreased even on this subset. Moving over to AFLW, in Fig. 16 we show AFLW performance over years. The performance is still being improved. We have deliberately left NME for algorithm from 2014 to note the significant progress made in 7 years.

To conclude this section, we would like to note the best algorithms on different datasets: SubpixelHeatmap, ADNet, PropagationNet. All of these algorithms are heatmap-based. The key ideas proposed in them are complimentary, for instance, SubpixelHeatmap has offered a way to decrease heatmap quantization error. Recall, that input image and landmark annotations are of resolution 256×256, while the heatmap is only 64×64. The authors of ADNet have presented Point-Edge heatmaps as attention masks, and PropagationNet has presented Focal Wing Loss modification. However, inference time of ADNet is still quite high at 95.29 ms on GPU. We expect PropagationNet to be even slower, based on presented in Table 9 number of floating point operations. SubpixelHeatmap neural network is inferred for each image several times, which will also be slow. Thus, we would like to see faster algorithms in future, and those that are easily applicable to mobile devices.

5 Facial Landmark Detection: Applications

5.1 Mobile-Friendly Joint Face and Landmark Detection

As we have noted previously, face landmark detection is one face processing pipeline steps. To actually get a dense landmark annotation, face has to be detected first and cropped based on its bounding box. In this section we present some of mobile-friendly face detection methods. Interestingly, these methods also predict coarse (5 or 6) face landmarks, such as eyes, mouth, nose.

Multi-task Cascaded Convolutional Networks (MTCNN) [66]. The neural network is trained jointly to detect faces and landmark locations (five of them, to be precise: eyes, tip of the nose, mouth corners), which improves quality on both tasks. The network is built in a form of a three-network cascade: Proposal Network (P-Net), Refine Network (R-Net), Output Network (O-Net). Each network predicts face bounding rectangle, probability that a particular rectangle contains a face, and five landmarks. P-Net is a fast fully convolutional network, which processes the original image in multiple resolutions (the so-called image pyramid). This network outputs a lot of coarse face rectangle predictions, which are then filtered out by the Non-Maximum Suppression (NMS) algorithm. Subsequently, R-Net refines the predicted rectangles. It does so without reprocessing the whole image, which saves computation time. NMS is then applied again. Last, O-Net makes the final refinement. This is the slowest network in the cascade, but it processes a small number of face rectangles. According to the authors, to improve quality it is important to solve the following tasks at the same time: 1) classify bounding rectangle as a face or not a face; 2) perform regression over bounding rectangle coordinates; 3) localize face landmarks. Each task has a weight α assigned: for P-Net and R-Net $\alpha_1 = 1, \alpha_2 = \alpha_3 = 0.5$, for O-Net $\alpha_1 = 1, \alpha_2 = 0.5, \alpha_3 = 1$ correspondingly. At training time online hard-example mining has been used, meaning that training is performed on complicated examples while skipping those, on which network prediction is quite accurate already. In the paper the authors select around 70% of hardest examples in each training batch.

BlazeFace [67] is a novel approach to joint face and landmark detection. 6 landmarks are predicted: eye center, ear tragions, mouth center, and nose tip). The algorithm was specifically designed for inference on mobile devices. The authors claim sub-second detection time on mobile for Tensorflow [68] GPU implementation. The approach is based on Single Shot Detector (SSD) [69] with MobileNetV2 backbone. The authors propose to modify MobileNetV2 to improve performance to accuracy ratio. For that they increase complexity of Bottleneck block (main

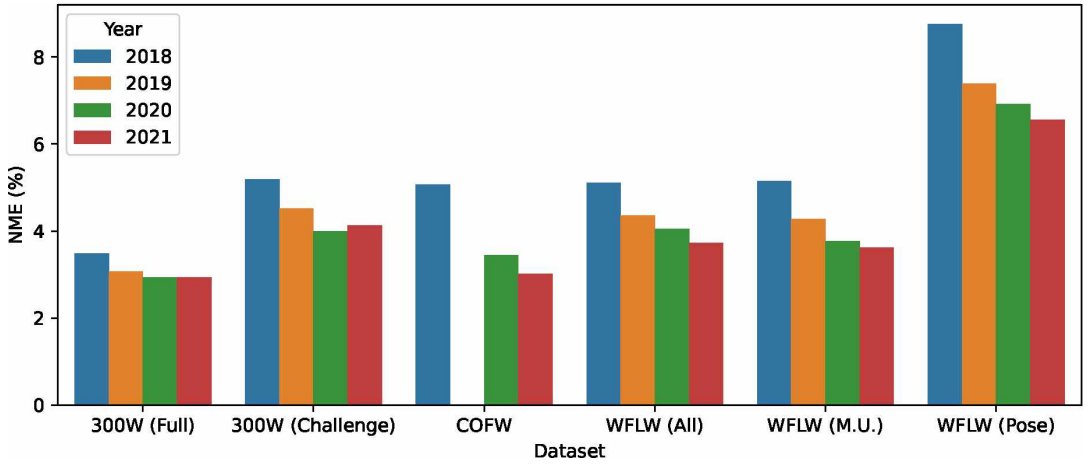


Figure 15: State-of-the-art normalized mean error (NME, %) on 300W (Full, Challenge), COFW, WFLW (All, Make-Up, Pose) by years. Score on 300W dataset doesn't improve in 2021. Error on WFLW dataset, in contrast, significantly decreases with the time. WFLW error is higher (especially on images with large pose), than that of 300W and COFW, which makes it the most challenging dataset.

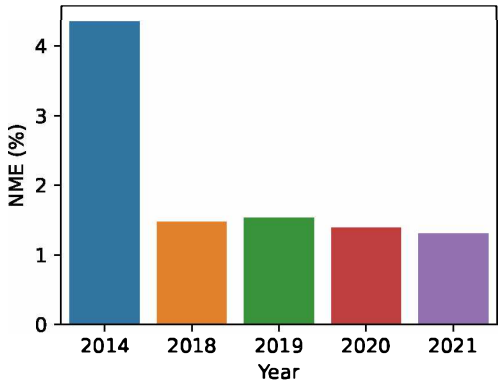


Figure 16: Algorithm Normalized Mean Error (NME, %) on AFLW dataset by year. We deliberately left NME from 2014 to illustrate significant progress over nearly a decade. Note that NME still decreases year over year.

building block of MobileNetV2 architecture) and decrease the number of such blocks at the same time. Also, they have optimized SSD architecture for face detection by removing the ability to predict wide or tall bounding boxes, that are not common for faces. In addition, intra-frame jitter produced by the NMS algorithm has been reduced via a separate bounding box regression module. The network is proposed in 2 configurations: one for pictures taken on back camera (typically smaller faces) and another for frontal camera photos (typically larger faces). While the network has good accuracy, the input resolutions are fixed to 128×128 or 256×256, which is a disadvantage of the method. MTCNN, for example, can take images of arbitrary resolution as input. Note, that training has been performed on a closed dataset. Thus, it is not possible to reproduce the results.

A comprehensive review of other modern face detection methods is presented in [70]. We see the following prospects for further research in the field of joint face and landmark detection:

- inference speed to accuracy ratio requires improvement. Faster approaches often have lower quality;
- large annotated dataset is required to train the model. If the dataset is biased (unbalanced) in race or gender, face detection accuracy of underrepresented groups will typically suffer.

5.2 Face Animation and Reenactment

Facial landmark detection is used in human or imaginary character face animation algorithms. Applications include actor animation in movies, creation of TV or game virtual newscasters (as a 3D model or directly via GAN image generation). Recent landmark detection algorithms enable this without costly equipment by using a simple RGB camera.

According to research presented in a series of papers, movie dubbing process from foreign languages is expensive and time-consuming. This is because lip movement for the original and dubbed audio tracks should match. Furthermore, the movement discrepancy leads to discomfort when watching movies, especially for hearing-impaired people. As a solution, authors of [71] propose to change lip movement during the dubbing process. Their algorithm detects facial landmarks and substitutes mouth region with a 3D

model, adapted for the speaker. However, at this stage the substitution is still visible. Besides that, DeFA algorithm can build a 3D whole-face mesh for varied poses and emotions, as has been said previously.

Many of the recent neural-network-based algorithms do not use an intermediate 3D face model for realistic image generation, but generate images directly from facial landmark locations via Generative Adversarial Network (GAN). For instance, the authors of [72] by using MAML [73] meta-learning approach, GAN and the so-called perceptual loss [74], obtain high face reenactment quality (Fig. 17). Landmark information extracted from an image is one of the neural network inputs. FAN algorithm is used to extract the landmarks. The algorithm has some disadvantages though. For instance, when actor, that drives the animation, has significantly different face shape from animated face, the resulting animation is unrealistic and contains artifacts. According to the authors' report, this method outperforms the competition for face emotion transfer task in few- or one-shot problem statement. The authors note, that an improvement of facial extraction algorithm and addition of gaze direction might have improved the reenactment quality.

Figure 17: Reenactment scheme. (a) source character image (the one we want to reenact); (b) one of frames of driver actor; (c) extracted facial landmarks that are fed to the reenactment algorithm; (d) reenactment result.[6]

In [75] authors are using Pix2PixHD [76] neural network to accomplish lip sync task. It has been proposed to synthesize the intermediate face representation using its boundaries, face landmarks (using Dlib library) and sound-track-based representation.

In another work FReeNet [77] algorithm is presented for reenactment between different (unknown during training) people. For that a special Unified Landmark Converter module has been introduced, which adapts facial landmark coordinates between different people. Landmarks for the source and target people are extracted via PFLD algorithm. Then images are generated via Cycle-GAN [42] and a special loss function. The use of landmark converter module has given the largest performance increase on the test sets.

A survey of emotion transfer and face reenactment methods can be found in [78] Section "Expression Swap". Most recent algorithms, that focus on face animation of real people use generative adversarial neural networks (GANs). Currently, these approaches have the following limitations, that require a solution in future, for instance:

- videos produced by neural networks lack temporal stability. For instance, face animation might jitter, artifacts may appear on the screen;
- face animation under large pose might cause unrealistic face deformation;
- animation quality significantly suffers, if source character and animation driver have different face shape.

5.3 Driver Status Tracking

A large number of car accidents happens because-of sleepy or tired drivers. Expensive cars offer capabilities of emergency stopping when an obstacle is detected, and line-keep assist. Mainstream cars do not have such features. In both cases, it is better to track driver status and stop the car early, than to apply emergency brakes. Most of the research in the field is focused on implementing status tracking in an autonomous way (without Internet connection). Driver's smartphone or low-power portative device (such as Raspberry Pi) is used to process video signal from a camera placed in a car's cabin. Neural-network-based algorithms are among the most widely used approaches here.

In [79] the authors estimate driver tiredness via a neural network that takes facial landmarks as an input. Driver's face and landmarks are detected with existing methods. In contrast, in [80] a MobileNetV2-based architecture is presented to estimate driver's sleepiness directly from the video stream (without an intermediate step of landmark detection), yet total training time is quite high. In [81] neural-network-based landmark detection is utilized to simplify dataset labelling, then a different network is trained to recognize driver's status. In addition to fatigue, the authors also estimate driver's distraction by tracking whether he looks in safe zones (such road, rear-view mirror, dashboard, etc.) or not. In [82] a system that tracks driver's ability to take over the driving from level 2 autonomous cars (partial driving automatization) is studied. The authors acquire driver's video via an infrared camera. Decision whether the driver is distracted is based on the detected landmarks. These and similar algorithms are developed to make the roads safer.

Special hardware can also be used to track driver status [79], [80], for instance, tracking of driving wheel movement; wearable devices that perform Electrocardiography (ECG) and heartbeat measurements. However, both of these approaches are more expensive and cannot track driver's distraction from the road.

Neural-network-based driver status tracking algorithms have the following limitations, for instance:

- achieving sufficiently fast inference on a mobile device is a challenge;

[6]Distributed under Creative Commons – Attribution 3.0 license. Based on this source.

- driver status tracking is often performed at nighttime when driver is poorly illuminated. Facial landmark detection accuracy suffers in such conditions, especially given limited computation power.

This is why development of mobile networks and landmark detection algorithms will definitely enhance the quality of driver status tracking systems.

5.4 Face Recognition and Emotion Classification

To begin with, we briefly talk about algorithms that perform one of the following tasks (often, the same algorithm can perform all of them): 1) face verification, when 2 pictures are given and the task is to say whether they contain the same person; 2) face recognition: given a photo and a known person database, algorithm should say who is on the photo or that the person is unknown; 3) clusterization, where the task is to group similar faces. The most efficient algorithms use face preprocessing, that is face detection and tight crop. Often for improving recognition quality the so-called "face alignment" should additionally be performed, that is a geometrical image transformation, when facial landmarks are moved to the canonical locations. Many of the modern algorithms use MTCNN for joint face detection and localization of 5 landmarks. The topic of face recognition is well-described in, for example, [83]. We note in particular, high interest to face recognition directly on mobile devices [84], [85].

Also, we discuss emotion recognition. Our emotions mostly consist of lip, eyes, eyebrows or mouth movements. That is why in certain cases it is fruitful not to force the neural network to learn face parts during emotion recognition on its own, but to feed this information detected by another algorithm together with the original image [86], [87].

The field of face recognition has several problems that require a solution in future, for example:

- face recognition when photos represent a person of different ages;
- when face occlusion is significant, which is especially important when medical masks have become common;
- faces with large pose and emotion;
- also, some of the backbones discussed here in a context of facial landmark detection are used for face recognition and emotion classification. Thus, improving neural network backbones is important as well.

6 Facial Landmark Detection: Vulnerabilities

Modern computer vision algorithms (including neural networks) are amenable to the so-called "adversarial attacks", first reported in the field of computer vision in [88]. The authors were able to drastically change neural network prediction in classification task by adding especially crafted noise (invisible to human eye) to an image. The attack has been conducted by maximizing network error on the target image via L-BFGS method. Images with adversarial noise are almost always misclassified on MNIST dataset. It should be stressed that during the adversarial attack the network itself is not modified, only the images fed to it. Moreover, adversarial examples often remain malicious to networks different from the one they were crafted for, given that another network was trained on the same or similar dataset. It should be noted, that adding random noise has much lower negative effect on the network's classification accuracy, than adversarial attack noise. In [89] it has been shown that for a successful adversarial attack on the MNIST dataset, model as simple as logistic regression can be used to generate adversarial examples. The attack remains efficiently transferrable to architectures, that are more complicated.

If previous algorithms have attacked a digital image (stored in computer memory), in [90] it has been shown that attacks can be performed through a smartphone camera. In [91] binary importance maps have been introduced, which hint where adversarial marks should be placed on a piece of paper to fool the network trained to classify handwritten digits. The first adversarial attacks were white-box, i.e., the network architecture and trained weights are known to the attacker. Follow-up works similar to [92] and others have shown that it is possible to perform black-box attacks without such knowledge. Despite the fact that numerous works are devoted to detecting or preventing attacks from happening, new more advanced algorithms bypass all of the defense methods [93]. A survey of adversarial attack methods can be found in [94]. All of them are applicable to algorithms of face or facial landmark detection.

In the meantime, there exist special methods that can prevent the face from being found or correctly detected by using stickers or accessories in real world. In [95] it has been shown, that in a controllable environment it is possible to fool face recognition algorithm or Viola-Jones face detector. The authors used special eyeglasses with a print on a frame. In [96] it has been proposed to fool MTCNN face detection algorithm with the use of stickers on cheeks or medical mask. In cases when the face cannot be detected, landmark localization cannot be performed either. Face recognition adversarial attack based on facial landmarks is presented in [51].

7 Conclusion

From a detailed survey, we see the following facial landmark detection algorithm problems, that require a solution in future research: 1) despite a significant growth of methods' quality, few of them focus on the real-world applicability in resource-constrained environments, such as mobile or edge devices; 2) many applications require high performance on mobile or portable devices, yet to the best of our knowledge, authors of only a single algorithm have targeted a mobile application directly in the original paper. Note that state-of-the-art algorithms have slow inference speed; 3) while modern research already focuses on datasets in uncontrollable environments, a promising research direction is to enhance algorithms in even harsher conditions, for images with large pose and significant face occlusion, while still maintaining high landmark density. Error of current generation of algorithms in these conditions is quite high. We see WFLW dataset as the one posing the most interest for further research. Also, it would be desirable to see more of the novel facial landmark detection algorithms to report their inference speed on desktop GPU, and if possible, on mobile devices.

We hope, that the described modern developments in all of the sections will lead the reader to new ideas of practical use and further research directions.

Competing interests

The authors have declared that no competing interests exist.

Funding

The work is supported by the state budget scientific research project of Dnipro University of Technology "Development of New Mobile Information Technologies for Person Identification and Object Classification in the Surrounding Environment" (state registration number 0121U109787).

Authors' contribution

KK proposed the idea, selected impactful literature in the field, surveyed and analyzed the literature, wrote the manuscript; LK validated the manuscript. All authors read and approved the final manuscript.

References

[1] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856. DOI: 10.1109/CVPR.2018.00716.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[3] N. Wang, X. Gao, D. Tao, H. Yang, and X. Li, "Facial feature point detection: A comprehensive survey," *Neurocomputing*, vol. 275, pp. 50–65, 2018, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.05.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217308202.

[4] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *Int. J. Comput. Vision*, vol. 127, no. 2, pp. 115–142, Feb. 2019, ISSN: 0920-5691. DOI: 10.1007/s11263-018-1097-z.

[5] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2129–2138. DOI: 10.1109/CVPR.2018.00227.

[6] A. Kumar, T. K. Marks, W. Mou, *et al.*, "Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8233–8243. DOI: 10.1109/CVPR42600.2020.00826.

[7] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6970–6980. DOI: 10.1109/ICCV.2019.00707.

[8] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403. DOI: 10.1109/ICCVW.2013.59.

[9] 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, 2013. DOI: 10.1109/TPAMI.2013.23.

[10] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, IEEE Computer Society, 2012, pp. 2879–2886. DOI: 10.1109/CVPR.2012.6248014.

[11] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 7574, Springer, 2012, pp. 679–692. DOI: 10.1007/978-3-642-33712-3_49.

[12] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maître, "Xm2vtsdb: The extended m2vts database," vol. 964, 1999, pp. 965–966.

[13] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 146–155. DOI: 10.1109/CVPR.2016.23.

[14] C. Zhu, X. Li, J. Li, and S. Dai, "Improving robustness of facial landmark detection by defending against adversarial attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 11 751–11 760.

[15] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 2144–2151. DOI: 10.1109/ICCVW.2011.6130513.

[16] S. Qian, K. Sun, W. Wu, C. Qian, and J. Jia, "Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 10 152–10 162. DOI: 10.1109/ICCV.2019.01025.

[17] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 1513–1520. DOI: 10.1109/ICCV.2013.191.

[18] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Detecting and localizing occluded faces," *CoRR*, vol. abs/1506.08347, 2015. arXiv: 1506.08347. [Online]. Available: http://arxiv.org/abs/1506.08347.

[19] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen, "The menpo facial landmark localisation challenge: A step towards the solution," in *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[20] J. Deng, A. Roussos, G. Chrysos, *et al.*, "The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking," *International Journal of Computer Vision*, pp. 1–26, 2018.

[21] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1021–1030. DOI: 10.1109/ICCV.2017.116.

[22] S. Zafeiriou, G. Chrysos, A. Roussos, E. Ververas, J. Deng, and G. Trigeorgis, "The 3d menpo facial landmark tracking challenge," in *International Conference on Computer Vision (ICCV) Workshops*, 2017.

[23] G. G. Chrysos, E. Antonakos, S. Zafeiriou, and P. Snape, "Offline deformable face tracking in arbitrary videos," in *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 954–962. DOI: 10.1109/ICCVW.2015.126.

[24] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 1003–1011. DOI: 10.1109/ICCVW.2015.132.

[25] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, pp. 3659–3667. DOI: 10.1109/CVPR.2015.7298989.

[26] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14, USA: IEEE Computer Society, 2014, pp. 1867–1874, ISBN: 9781479951185. DOI: 10.1109/CVPR.2014.241.

[27] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009, ISSN: 1532-4435.

[28] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

(CVPR 2001), 8-14 December 2001, Kauai, HI, USA, IEEE Computer Society, 2001, pp. 511–518. DOI: 10.1109/CVPR.2001.990517.

[29] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 2235–2245. DOI: 10.1109/CVPR.2018.00238.

[30] Y. Yan, X. Naturel, T. Chateau, S. Duffner, C. Garcia, and C. Blanc, "A survey of deep facial landmark detection," in *RFIAP*, Paris, France, Jun. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02892002.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[33] A. Howard, R. Pang, H. Adam, *et al.*, "Searching for mobilenetv3," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[34] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: practical guidelines for efficient CNN architecture design," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, ser. Lecture Notes in Computer Science, vol. 11218, Springer, 2018, pp. 122–138. DOI: 10.1007/978-3-030-01264-9_8.

[35] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, ser. Lecture Notes in Computer Science, vol. 9912, Springer, 2016, pp. 483–499. DOI: 10.1007/978-3-319-46484-8_29.

[36] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 5693–5703. DOI: 10.1109/CVPR.2019.00584.

[37] Z. Tang, X. Peng, K. Li, and D. N. Metaxas, "Towards efficient u-nets: A coupled and quantized approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2038–2050, 2020. DOI: 10.1109/TPAMI.2019.2907634.

[38] J. Wang, K. Sun, T. Cheng, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2021. DOI: 10.1109/TPAMI.2020.2983686. [Online]. Available: https://doi.org/10.1109/TPAMI.2020.2983686.

[39] Y. Liu, A. Jourabloo, W. Ren, and X. Liu, "Dense face alignment," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1619–1628. DOI: 10.1109/ICCVW.2017.190.

[40] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Style aggregated network for facial landmark detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 379–388. DOI: 10.1109/CVPR.2018.00047.

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *CoRR*, vol. abs/1406.2661, 2014. arXiv: 1406.2661. [Online]. Available: http://arxiv.org/abs/1406.2661.

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

[43] X. Chu, W. Ouyang, H. Li, and X. Wang, "Structured feature learning for pose estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 4715–4723. DOI: 10.1109/CVPR.2016.510.

[44] W. Yang, W. Ouyang, H. Li, and X. Wang, "End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 3073–3082. DOI: 10.1109/CVPR.2016.335.

[45] R. B. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[46] X. Guo, S. Li, J. Zhang, *et al.*, "PFLD: A practical facial landmark detector," *CoRR*, vol. abs/1902.10859, 2019. arXiv: 1902.10859. [Online]. Available: http://arxiv.org/abs/1902.10859.

[47] A. Bulat and G. Tzimiropoulos, "Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 3726–3734. DOI: 10.1109/ICCV.2017.400.

[48] R. Liu, J. Lehman, P. Molino, *et al.*, "An intriguing failing of convolutional neural networks and the coordconv solution," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, Montreal, Canada: Curran Associates Inc., 2018, pp. 9628–9639.

[49] Y. Zhao, Y. Liu, C. Shen, Y. Gao, and S. Xiong, "Mobilefan: Transferring deep hidden representation for face alignment," *Pattern Recognition*, vol. 100, p. 107114, 2020, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2019.107114. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319304157.

[50] S. M. Iranmanesh, A. Dabouei, S. Soleymani, H. Kazemi, and N. M. Nasrabadi, "Robust facial landmark detection via aggregation on geometrically manipulated faces," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 319–329. DOI: 10.1109/WACV45572.2020.9093508.

[51] A. Dabouei, S. Soleymani, J. Dawson, and N. Nasrabadi, "Fast geometrically-perturbed adversarial faces," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1979–1988. DOI: 10.1109/WACV.2019.00215.

[52] W. Li, Y. Lu, K. Zheng, *et al.*, "Structured landmark detection via topology-adapting deep graph learning," in *Computer Vision – ECCV 2020*, Cham: Springer International Publishing, 2020, pp. 266–283, ISBN: 978-3-030-58545-7.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556.

[54] X. Huang, W. Deng, H. Shen, X. Zhang, and J. Ye, "Propagationnet: Propagate points to curve to learn structure information," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, Computer Vision Foundation / IEEE, 2020, pp. 7263–7272. DOI: 10.1109/CVPR42600.2020.00729.

[55] R. Zhang, "Making convolutional networks shift-invariant again," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 7324–7334. [Online]. Available: http://proceedings.mlr.press/v97/zhang19a.html.

[56] H. Yang, J. Lyu, P. Cheng, and X. Tang, "Lddmm-face: Large deformation diffeomorphic metric learning for flexible and consistent face alignment," *CoRR*, vol. abs/2108.00690, 2021. arXiv: 2108.00690. [Online]. Available: https://arxiv.org/abs/2108.00690.

[57] J. A. Glaunès, A. Qiu, M. I. Miller, and L. Younes, "Large deformation diffeomorphic metric curve mapping," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 317–336, 2008. DOI: 10.1007/s11263-008-0141-9.

[58] S. C. Joshi and M. I. Miller, "Landmark matching via large deformation diffeomorphisms," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1357–1370, 2000. DOI: 10.1109/83.855431.

[59] Z. Xu, B. Li, Y. Yuan, and M. Geng, "Anchorface: An anchor-based facial landmark detector across large poses," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 3092–3100. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16418.

[60] H. Jin, S. Liao, and L. Shao, "Pixel-in-pixel net: Towards efficient facial landmark detection in the wild," *International Journal of Computer Vision*, vol. 129, no. 12, pp. 3174–3194, Dec. 2021, ISSN: 1573-1405. DOI: 10.1007/s11263-021-01521-4.

[61] Y. Huang, H. Yang, C. Li, J. Kim, and F. Wei, "Adnet: Leveraging error-bias towards normal direction in face alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 3080–3090.

[62] X. Lan, Q. Hu, and J. Cheng, "Revisting quantization error in face alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2021, pp. 1521–1530.

[63] A. Bulat, E. Sanchez, and G. Tzimiropoulos, "Subpixel heatmap regression for facial landmark localization," *CoRR*, vol. abs/2111.02360, 2021. arXiv: 2111.02360. [Online]. Available: https://arxiv.org/abs/2111.02360.

[64] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, *et al.*, Eds., 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[65] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 5137–5146. DOI: 10.1109/CVPR.2018.00539.

[66] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. DOI: 10.1109/LSP.2016.2603342.

[67] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *CoRR*, vol. abs/1907.05047, 2019. arXiv: 1907.05047. [Online]. Available: http://arxiv.org/abs/1907.05047.

[68] Martin Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. [Online]. Available: https://www.tensorflow.org/.

[69] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 9905, Springer, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2.

[70] S. Minaee, P. Luo, Z. Lin, and K. W. Bowyer, "Going deeper into face detection: A survey," *CoRR*, vol. abs/2103.14983, 2021. arXiv: 2103.14983. [Online]. Available: https://arxiv.org/abs/2103.14983.

[71] P. Garrido, L. Valgaerts, H. Sarmadi, *et al.*, "Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 193–204, May 2015, ISSN: 0167-7055. DOI: 10.1111/cgf.12552.

[72] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 9458–9467. DOI: 10.1109/ICCV.2019.00955.

[73] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 1126–1135.

[74] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, Cham: Springer International Publishing, 2016, pp. 694–711, ISBN: 978-3-319-46475-6.

[75] R. Zheng, Z. Zhu, B. Song, and C. Ji, "A neural lip-sync framework for synthesizing photo-realistic virtual news anchors," in *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, IEEE, 2020, pp. 5286–5293. DOI: 10.1109/ICPR48806.2021.9412187. [Online]. Available: https://doi.org/10.1109/ICPR48806.2021.9412187.

[76] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 8798–8807. DOI: 10.1109/CVPR.2018.00917.

[77] J. Zhang, X. Zeng, M. Wang, *et al.*, "Freenet: Multi-identity face reenactment," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5325–5334. DOI: 10.1109/CVPR42600.2020.00537.

[78] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2020.06.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303110.

[79] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver drowsiness detection model using convolutional neural networks techniques for android application," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 237–242. DOI: 10.1109/ICIoT48696.2020.9089484.

[80] J. S. Wijnands, J. Thompson, K. A. Nice, G. D. Aschwanden, and M. Stevenson, "Real-time monitoring of driver drowsiness on mobile platforms using 3d neural networks," *Neural Computing and Applications*, pp. 1–13, 2019.

[81] W. Kim, W.-S. Jung, and H. K. Choi, "Lightweight driver monitoring system based on multi-task mobilenets," *Sensors*, vol. 19, no. 14, 2019, ISSN: 1424-8220. DOI: 10.3390/s19143200. [Online]. Available: https://www.mdpi.com/1424-8220/19/14/3200.

[82] T. Hyuga, K. Kinoshita, K. Nishiyuki, and Y. Hasegawa, "Driver status monitoring system in autonomous driving era," *OMRON TECHNICS*, vol. 50.005EN, 2019. [Online]. Available: https://www.omron.com/global/en/assets/file/technology/omrontechnics/vol50/OMT_Vol50_005.pdf.

[83] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018, pp. 471–478. DOI: 10.1109/SIBGRAPI.2018.00067.

[84] C. N. Duong, K. G. Quach, I. Jalata, N. Le, and K. Luu, "Mobiface: A lightweight deep learning face recognition on mobile devices," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019, pp. 1–6. DOI: 10.1109/BTAS46853.2019.9185981.

[85] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Biometric Recognition*, Cham: Springer International Publishing, 2018, pp. 428–438, ISBN: 978-3-319-97909-0.

[86] B. C. Ko, "A brief review of facial emotion recognition based on visual information," *Sensors*, vol. 18, no. 2, 2018, ISSN: 1424-8220. DOI: 10.3390/s18020401. [Online]. Available: https://www.mdpi.com/1424-8220/18/2/401.

[87] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, pp. 1–1, 2020. DOI: 10.1109/TAFFC.2020.2981446.

[88] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199.

[89] K. S. Khabarlak and L. S. Koriashkina, "Scoping adversarial attack for improving its quality," *Radio Electronics, Computer Science, Control*, no. 2, pp. 108–118, May 2019. DOI: 10.15588/1607-3274-2019-2-12. [Online]. Available: http://ric.zntu.edu.ua/article/view/178284.

[90] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=HJGU3Rodl.

[91] K. Khabarlak and L. Koriashkina, "Minimizing perceived image quality loss through adversarial attack scoping," *CoRR*, vol. abs/1904.10390, 2019. arXiv: 1904.10390. [Online]. Available: http://arxiv.org/abs/1904.10390.

[92] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '17, Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 15–26, ISBN: 9781450352024. DOI: 10.1145/3128572.3140448.

[93] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '17, Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 3–14, ISBN: 9781450352024. DOI: 10.1145/3128572.3140444.

[94] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018. DOI: 10.1109/ACCESS.2018.2807385.

[95] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16, Vienna, Austria: Association for Computing Machinery, 2016, pp. 1528–1540, ISBN: 9781450341394. DOI: 10.1145/2976749.2978392.

[96] E. Kaziakhmedov, K. Kireev, G. Melnikov, M. Pautov, and A. Petiushko, "Real-world attack on mtcnn face detection system," in *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 2019, pp. 0422–0427. DOI: 10.1109/SIBIRCON48586.2019.8958122.

Citation: K. Khabarlak, L. Koriashkina. *Fast Facial Landmark Detection and Applications: A Survey.* Journal of Computer Science & Technology, vol. 22, no. 1, pp. 12–41, 2022.
DOI: 10.24215/16666038.22.e02
Received: December 6, 2022 **Accepted:** March 18, 2022.