Separable Batch Normalization for Robust Facial Landmark Localization

Shuangping Jin[1]
220191583@seu.edu.cn
Zhenhua Feng[2, 3]
z.feng@surrey.ac.uk
Wankou Yang[1*]
wkyang@seu.edu.cn
Josef Kittler[3]
j.kittler@surrey.ac.uk

[1] School of Automation
Southeast University
Nanjing 210096, China

[2] Department of Computer Science
University of Surrey,
Guildford GU2 7XH, UK

[3] Centre for Vision, Speech and Signal
Processing (CVSSP)
University of Surrey,
Guildford GU2 7XH, UK

Abstract

A big, diverse and balanced training data is the key to the success of deep neural network training. However, existing publicly available datasets used in facial landmark localization are usually much smaller than those for other computer vision tasks. To mitigate this issue, this paper presents a novel Separable Batch Normalization (SepBN) method. Different from the classical BN layer, the proposed SepBN module learns multiple sets of mapping parameters to adaptively scale and shift the normalized feature maps via a feed-forward attention mechanism. The channels of an input tensor are divided into several groups and the different mapping parameter combinations are calculated for each group according to the attention weights to improve the parameter utilization. The experimental results obtained on several well-known benchmarking datasets demonstrate the effectiveness and merits of the proposed method.

1 Introduction

The task of facial landmark localization is to predict the position of a set of pre-defined facial key points. It plays a crucial role in many automatic face analysis systems, including face recognition [8, 21, 29], face morphing [12], expression recognition [18, 33, 44], 3D face fitting [11, 47], etc. The rapid development of this research area in the recent years produced a variety of effective neural network architectures [35, 41] and loss functions [9, 31, 34], which have been instrumental in achieving impressive landmarking results. The reported performance of these deep Convolutional Neural Network (CNN-) based methods demonstrates their superiority over traditional approaches such as the Active Shape Model (ASM) [5], Active Appearance Model (AAM) [6] and cascaded regression [37, 38], especially when handling unconstrained faces in the wild.

In unconstrained scenarios, the key challenge for facial landmark localization is posed by facial appearance variations, including pose, illumination, expression, occlusion, motion

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

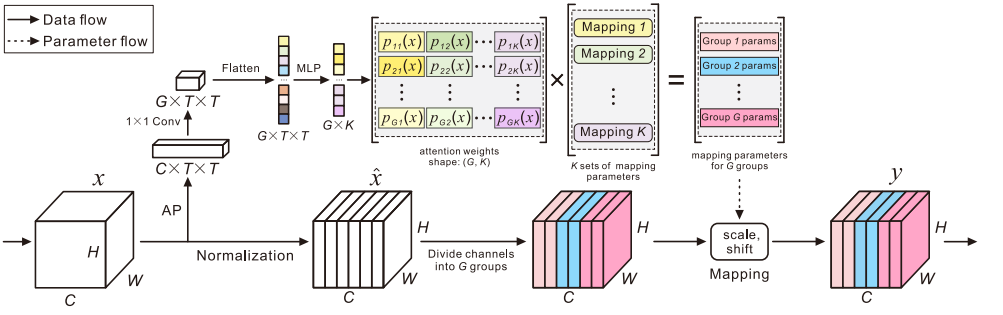

Figure 1: The proposed SepBN module with K sets of parameters. The input feature map $\mathcal{X}$ is normalized like the classical BN layer to obtain $\hat{\mathcal{X}}$ that is then divided into G groups. The multiple sets of mapping parameters are weighted and summed using the attention weights to produce group-specific parameters for the final mapping operation from $\hat{\mathcal{X}}$ to $\mathcal{Y}$. GAP stands for the Global Adaptive Pooling operation.

blur, low image resolution, etc. All these influencing factors should be taken into account but this is impeded by the difficulty of collecting samples with diverse appearance variations, leading to a severe data imbalance distribution problem on specific attributes. In addition to that, the datasets available for the facial landmarking research are usually much smaller than those available for other mainstream computer vision tasks. A worse case is the COFW [3] dataset that has only 1,345 training samples. Such a small dataset leads to the inevitable data imbalance issue in landmark localization. As a result, the trained network may not be able to generalize well for unseen samples whose types occur rarely in the training set.

Inspired by BRN [13], suggesting that the learned parameters of BN [14] reflect the distribution domain of a dataset, we try to deal with the aforementioned issues by performing a Separable Batch Normalization (SepBN). The classical BN layer treats all the samples of a dataset equally, leading to a potential bias and overfitting of a trained network. The difference between SepBN and BRN is that, instead of re-tuning the parameters in the validation phase, SepBN directly learns multiple sets of parameters during network training.

The key innovation of the proposed SepBN module is to adaptively integrate K separable BN branches in an efficient fashion to facilitate the joint learning of data from different domains. These branches share the same normalization operation but maintain different mapping parameters. In the inference stage, the input tensor will be normalized first. Then the normalized tensor is divided into G groups across the channel dimension. The G groups of mapping parameters are produced via a novel attention mechanism that only depends on the input tensor to scale and shift the corresponding groups, as depicted in Figure 1. The SepBN module endows the original BN layer with a non-linear mapping capability to map the normalized feature map dynamically using the information obtained by the attention block. More importantly, the proposed SepBN module implicitly learns the mapping function for different types of samples and eases the small-sample-size problem posed by an unbalanced training data, which is shown in the experimental section.

A comprehensive validation of the proposed SepBN module on both simple and advanced network architectures, including our Vanilla CNN, MobileNet [26] and ResNeXt [40], are conducted. The results obtained on several benchmarking datasets demonstrate the effectiveness of the proposed method in different settings.

2 Related Work

Facial landmark localization has been studied for decades, resulting in a variety of well-known approaches, from traditional methods like ASM [8], AAM [6] and cascaded regression, to modern deep-learning-based methods [17, 34, 35, 41, 45].

Up to now, two main deep-learning-based methods have been developed: heatmap-based [34, 35, 41] and coordinate-regression-based methods [9]. These approaches take a facial image as input, and output the facial landmarks in two different ways. A heatmap-based method outputs 2D heatmaps, in which the pair of coordinates with the highest response value corresponds to the landmark position. In contrast, a coordinate-regression-based method solves the landmarking problem by predicting the landmark coordinates directly.

The heatmap-based facial landmark localization methods exploit the beneficial properties of the U-Net like networks [25], such as the Hourglass network [23] and densely connected U-Nets (DU-Net) [30]. However, heatmap-based methods suffer from the problem of quantization errors. Additionally, the training of such a network involves more hyper-parameters and its success often requires the use of special tricks.

In our work, we focus on the coordinate-regression-based method and propose a new SepBN module to enhance the network learning capability. It should be highlighted that, to the best of our knowledge, this is the first time that a new BN module has been developed for the facial landmark localization task.

The well-known **Batch Normalization** method alleviates the convergence problems in deep neural network training effectively [14].

In recent years, more advanced BN methods have been proposed, such as layer normalization [1], instance normalization [32], group normalization [39], and others [24, 27, 42]. In general, the existing normalization layers share the same processing pattern, *i.e.*, *normalization* and *mapping*. Our proposed SepBN module normalizes the features in the (N, H, W) dimension just like the classical BN layer. The key difference between SepBN and other normalization modules lies in the *mapping* operation, as shown in Figure 1. Compared to other modules that apply parallelized BN layers such as DSBN [4], which requires an additional effort to design each branch and lack flexibility, or such as SBN [43], which just involves a normalization step, our SepBN module isolates the mapping operation and can therefore be easily incorporated into any network to learn appropriate representations in an end-to-end fashion. Recently, a new normalization method, namely Attentive Normalization (AN) [19] (Instance Enhancement Batch Normalization [20] is also similar to AN), has been proposed to integrate feature normalization and feature attention into a single process. However, as will be shown in the experimental section, AN provides a limited performance boost in facial landmark localization, while requiring significantly more computation. We have performed many experiments to clarify the applicable scenarios of SepBN, compared to AN.

3 Separable Batch Normalization

3.1 Brute-Force SepBN

The classical BN layer consists of two key computational steps:

$$\hat{\mathcal{X}}_c = \frac{\mathcal{X}_c - \mathrm{E}(\mathcal{X}_c)}{\sqrt{\mathrm{Var}(\mathcal{X}_c)}}, \qquad \mathcal{Y}_c = \gamma_c \hat{\mathcal{X}}_c + \beta_c, \tag{1}$$

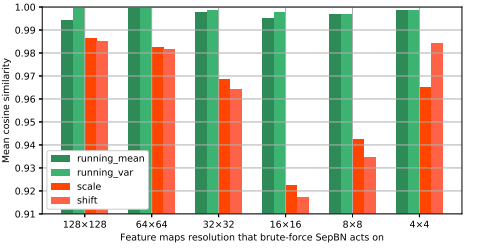

Figure 2: Mean cosine similarity of the running mean, running var, scale and shift learned by the brute-force SepBN modules. Since we apply the brute-force SepBN modules with three separate BN layers, the mean similarity is calculated as in Equation 2.

where $\mathcal{X}_c \in \mathbb{R}^{N\times H\times W}$ is one patch of the input tensor $\mathcal{X} \in \mathbb{R}^{N\times C\times H\times W}$, E() and Var() calculates the mean and variance within the patch. The running mean and variance of the corresponding patch will be updated in a moving average manner. In order to preserve the linearity of BN, the normalized patch is scaled and shifted by γ_c and β_c in the second equation. Two key components are present in a BN layer: *tracking* parameters (running mean and variance, updated in the forward phase) and *mapping* parameters (scale and shift, updated by the network backward propagation).

We first delve into the key question, namely whether it is necessary to separate both normalization and mapping operations as in DSBN [4]. To answer this question, we design a brute-force SepBN module so as to deal with the data from different domains separately.

In the brute-force SepBN module, several BN layers are linked to track the mean and variance of the data from specific domains in parallel. Specifically, different types of data are concatenated along the batch dimension. When passing through a brute-force SepBN module, the feature maps from the same domain are gathered and directed to pass through a BN layer specifically set up for this domain. Consequently, the running mean and variance values of different domains can be tracked separately. Hence, potential unhelpful interactions among data from different domains are prevented and inaccuracy of tracking parameter estimation is mitigated. The mapping parameters are domain-specific as well.

We train a Vanilla CNN network (following [9]), equipped with the brute-force SepBN module after each convolution layer, on the AFLW dataset [3]. The input is an RGB image $\mathcal{I}$ of size $128\times 128\times 3$. The Vanilla CNN predicts the landmark coordinate vector $\mathbf{v}_{pred} = [x_1, y_1, x_2, y_2, \ldots, x_L, y_L]^T \in \mathbb{R}^{2L}$ directly, where L is the number of landmarks. We apply brute-force SepBN modules with three branches and split the AFLW training set into three subsets: near-frontal, left profile and right profile, using the method mentioned in [9]. The training samples of a specific subset will only go through the corresponding BN layer in the brute-force SepBN. To determine whether it is necessary to separate the normalization and mapping steps, we calculate the average similarity of the learned tracking and mapping parameters of the three BN layers by:

$$S = [s(\mathbf{p}_1, \mathbf{p}_2) + s(\mathbf{p}_1, \mathbf{p}_3) + s(\mathbf{p}_2, \mathbf{p}_3)]/3 \tag{2}$$

where $s()$ is the cosine similarity of two given vectors, $\mathbf{p}_k$ denotes the learned tracking parameters (including running mean and variance) or mapping parameters (including scale and shift) of the kth BN layer. Since we use 6 brute-force SepBN modules for the Vanilla CNN (1 for each convolution block), 6×4 similarities are calculated as shown in Figure 2.

Surprisingly, we can see that the *tracking* parameters of all the three BN branches in any SepBN module are highly similar to each other throughout the whole network, indicating that separating the normalization step is unnecessary. In contrast, the learned *mapping* parameters differ more significantly. Based on the above observation, we conclude that the normalization operations of different BN layers embedded in a brute-force SepBN module can be merged, while different sets of mapping operations should be kept separately.

3.2 Automated SepBN with Group Attention

In this part, we focus on the automatic selection of SepBN branches and end-to-end network training without a prior dataset partitioning. To be more specific, the proposed SepBN module maintains K sets of mapping parameters $\gamma \in \mathbb{R}^{K\times C}$ and $\beta \in \mathbb{R}^{K\times C}$. The purpose is to allow the network to produce the most suitable mapping parameters for each sample adaptively.

The simplest choice is to use a squeeze-excite block like Attentive Normalization (AN) [19] to generate attention weights $\lambda \in \mathbb{R}^{N\times K\times 1\times 1}$ for the given K sets of mapping parameters:

$$\text{softmax}(F_{ex}(F_{sq}(\mathcal{X};\theta_{sq});\theta_{ex})) = \lambda \in \mathbb{R}^{N\times K\times 1\times 1}, \tag{3}$$

where F_{sq} is the squeeze operation with the reduction rate r, involving a global average pooling layer, a linear transformation layer and a non-linear activation function. F_{ex} is the excitation function including a linear transformation layer and a Sigmoid layer. The output of F_{sq} and F_{ex} is denoted as $\mathcal{X}_{sq} \in \mathbb{R}^{N\times \frac{C}{r}\times 1\times 1}$ and $\mathcal{X}_{ex} \in \mathbb{R}^{N\times K\times 1\times 1}$. θ_{sq} and θ_{ex} are the corresponding model parameters.

Next, the re-calibrated mapping parameters are calculated by:

$$\hat{\gamma}_n = \sum_{k=1}^{K} \lambda_{n,k}\gamma_k, \quad \hat{\beta}_n = \sum_{k=1}^{K} \lambda_{n,k}\beta_k, \tag{4}$$

where $\lambda_{n,k}$ is the attention weight of the kth mapping parameters, (*i.e.* γ_k and β_k), of the nth sample. $\hat{\gamma}_n$ and $\hat{\beta}_n$ are the instance-specific mapping parameters used for the nth sample. Note that the attention block is used to estimate the probability of the feature $\mathcal{X}$ being mapped by the kth mapping parameters rather than being applied directly to $\mathcal{X}$.

However, the above method ignores two important facts. First, attention is just used to assign different weights to different sets of parameters, which means that only by applying a large enough K can the module learn diverse mapping parameters. Second, the above method makes each attention weight to act on one entire set of mapping parameters. This often gives rise to a sub-optimal situation, where some valuable mapping parameters are inactivated due to the overall lower attention weight and vice versa.

To solve this problem, we introduce a channel grouping mechanism into our SepBN module. By considering that the mapping parameters are used to scale and shift a tensor along the channel dimension, each set of the mapping parameters is divided into G groups (channels of the normalized tensor $\hat{\mathcal{X}}$ will also be grouped in the same way). Then the attention block generates attention weights $\pi \in \mathbb{R}^{N\times G\times K}$ for each group of channels in each set of mapping parameters for each sample. To obtain such attention weights, a different attention mechanism is developed as shown in Figure 1. To be specific, the input feature $\mathcal{X} \in \mathbb{R}^{N\times C\times H\times W}$ is adaptively pooled (max pooling) into $\mathcal{X}_{amp} \in \mathbb{R}^{N\times C\times T\times T}$. The global average pooling is not chosen since we hope that more feature information can be retained by setting $T > 1$. Afterwards, $\mathcal{X}_{amp}$ will pass through a 1×1 convolution layer and a new tensor

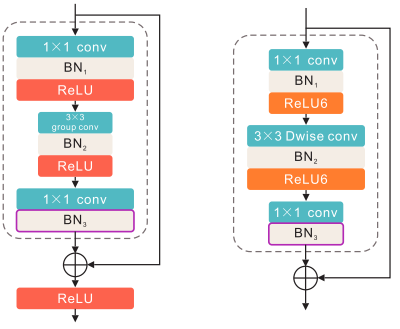

Figure 3: The bottleneck block used in ResNeXt-50 (left) and MobileNetV2 (right). The experimental results empirically show that replacing the BN_3 (boxes with purple border) with the proposed SepBN module maximises the performance.

is output in the shape of $N \times G \times T \times T$. Then the new tensor is flattened and used as the input of a Multi-Layer Perceptron module, followed by a temperature-controlled softmax function. At last, using the generated attention weights, the appropriate scale and shift parameters for each group of channels of each sample can be obtained:

$$\begin{gathered} \hat{\gamma}_{n,g} = \sum_{k=1}^{K} \pi_{n,g,k} \gamma_k, \quad \hat{\beta}_{n,g} = \sum_{k=1}^{K} \pi_{n,g,k} \beta_k, \\ \text{s.t.} \ \ 0 \leq \pi_{n,g,k} \leq 1, \sum_{k=1}^{K} \pi_{n,g,k} = 1 \end{gathered} \tag{5}$$

where $\pi_{n,g,k}$ is the attention weight indicating the probability that the gth channel group of the nth sample using the kth set of mapping parameters (*i.e.* γ_k and β_k), $\hat{\gamma}_{n,g} \in \mathbb{R}^M$ and $\hat{\beta}_{n,g} \in \mathbb{R}^M$ are the instance-and-group-specific mapping parameters used for the gth group of channels of the nth sample, and M is the number of channels for a group. The mapping operation computes:

$$\mathcal{Y}_{n,g} = \hat{\gamma}_{n,g} \hat{\mathcal{X}}_{n,g} + \hat{\beta}_{n,g}. \tag{6}$$

Through group division, rich combinations of mapping parameters can be obtained even when a smaller K is used, making the proposed SepBN module both lightweight and effective.

In contrast to the AN module equipped with group normalization, the proposed SepBN layer learns adaptive weights for different groups of channels and shows more efficient parameter utilization as shown in Table. 1 of the experimental section.

In fact, SepBN can be considered as an implicit model ensemble. By integrating multiple sets of mapping parameters, and by decorrelating feature channels, the tendency of model over-fitting can be diminished, especially when training a deep neural network on small-scale data.

3.3 Integration with A Modern Network

As SepBN is designed as a generic module, we need to verify its compatibility with modern network architectures, for example, ResNeXt-50 and MobileNetV2. We empirically apply SepBN on ResNeXt-50 and MobileNetV2 by replacing the BN_3 layer in the residual unit

Table 1: A comparison of SepBN and AN on AFLW-Full using NME(%). K signifies the number of branches. The lower right corner of the NME value indicates the number of parameters of the corresponding model.

K	2	3	5	10	20
MobileNetV2 (AN [19])	$1.67_{8.84M}$	$1.65_{8.84M}$	$1.65_{8.86M}$	$1.61_{8.89M}$	$1.60_{8.96M}$
MobileNetV2 (SepBN)	$\mathbf{1.59}_{8.83M}$	$\mathbf{1.57}_{8.84M}$	$\mathbf{1.57}_{8.85M}$	$\mathbf{1.58}_{8.87M}$	$\mathbf{1.57}_{8.95M}$

Table 2: A comparison of SepBN and classical BN on AFLW-Full using NME (%). Different proportions of training images are randomly selected and used for network training.

Method	100%	50%	20%	10%	5%
MobileNetV2 (BN)	1.66	1.79	1.86	1.98	2.13
MobileNetV2 (SepBN)	**1.57**	**1.64**	**1.72**	**1.76**	**1.81**

with the proposed SepBN module, as shown in Figure 3 (see the supplementary material for related experiments). Since the convolution layer does not have a lot of parameters, theoretically, we do not have to use the SepBN module after each convolution layer like dropout [28]. Another reason is that SepBN is used to identify and remap the features from different implicit domains, so it requires the network to extract useful semantic features by traditional convolutional layers followed by BN. By the way, considering that not all the original BNs are replaced by SepBN, the model size and computation complexity of the modified network do not increase significantly.

4 Experiments

4.1 Datasets and Implementation Details

We evaluated the proposed method on three datasets: COFW [3], AFLW [15] and WFLW [35]. The **COFW** dataset has 1,345 images for training and 507 images for test. COFW was designed to test the robustness of a facial landmark localization algorithm for faces with occlusions so most of the faces in COFW are occluded. **AFLW** consists of 24,386 faces with large pose variations. The AFLW dataset has two benchmarking protocols: AFLW-Full and AFLW-Frontal, both containing 20,000 training images. AFLW-Full uses all the remaining 4,386 images for test but AFLW-Frontal only uses 1,314 near-frontal faces for test. The **WFLW** dataset is a newly collected dataset that contains 10,000 faces (7,500 for training and 2,500 for test) with 98 facial landmarks. The test set is divided into several subsets to verify the robustness of an algorithm for specific appearance variation types.

We used the Normalized Mean Error (NME) and failure rate (%) as our evaluation metric, which is calculated by $NME = \frac{1}{L}\sum_{j=1}^{L} \left\|\mathbf{p}_j - \mathbf{g}_j\right\|_2 / d$, where $\mathbf{p}_j$ and $\mathbf{g}_j$ denote the jth predicted and ground-truth landmarks, d is a normalization term. Failure rate is defined as the proportion of test samples that have more than 0.1 landmark detection error in terms of NME. For AFLW, we used the bounding box size as the normalization term. For COFW and WFLW, we followed [35] and used the inter-ocular distance.

All the face images were cropped according to the official bounding box and resized to 128×128. The proposed SepBN module with $K = 3$ separate routes was used to replace the classical BN layer. Additionally, $T = 3$ and $G = 4$ were found to be the most suitable setting for our task. Due to space limitation, we report all the training details and experiment settings in our supplementary material. The code will be made publicly available.

Table 3: A comparison with the SOTA methods on COFW, in terms of NME and failure rate.

	NME ($\times 10^{-2}$)	Failure Rate ($\times 10^{-2}$)
RSR [9]	5.63	-
LAB [33]	3.92	0.39
ODN [36]	5.30	-
AWing [34]	5.30	0.99
RWing [31]	4.80	3.16
Vanilla CNN (SepBN)	**4.03**	**0.39**

Table 4: A comparison with the SOTA methods on AFLW, in terms of NME ($\times 10^{-2}$).

	AFLW-Full	AFLW-Frontal
LAB [33]	1.85	1.62
ODN [36]	1.63	1.38
SA [3]	1.60	-
LUVLi [20]	2.30	-
3FabRec [1]	1.84	1.59
Vanilla CNN (SepBN)	**1.55**	**1.39**

4.2 Effect of SepBN

We performed a number of experiments on AFLW to validate the proposed method and to optimize the configuration. For all the experiments, we use the same hyper-parameters mentioned earlier, except for those explicitely aimed at the settings' optimization.

SepBN vs. AN: SepBN shares similar structure with AN, while SepBN is more general and efficient. We compare the performance of SepBN and AN on AFLW using MobileNetV2 in Table 1. Once the branch number K of AN reaches 20, the performance of AN becomes comparable with the performance of SepBN ($K = 2$). This demonstrates the advantage of the SepBN module, namely its more efficient parameter utilization achieved by channel grouping.

Few-Shot Training: SepBN is designed to alleviate the problems in the face alignment task caused by small datasets. In order to demonstrate this point, we use part of the AFLW training set for network training and verify its performance on the complete test set. The results are shown in Table 2. As the number of training images decreases, the performance of the network equipped with SepBN is always better than the network using BN. This illustrates the ability of SepBN to maintain the network performance when the number of training samples is small.

Due to page limitation, more experimental results are reported in the supplementary material.

4.3 Comparison to The State of The Art

In this section, we first compare our Vanilla CNN equipped with a SepBN module to the state-of-the-art algorithms. After that, we gauge the performance improvement achieved by SepBN for datasets of different sizes and networks of different complexity. In this way, we can delineate the applicability of the SepBN module.

COFW: We compare our method with the state-of-the-art algorithms on COFW using NME and the failure rate in Table 3. 'Vanilla CNN (SepBN)' replaces the BN layer in Vanilla CNN with the proposed SepBN module *except the last one*. It clearly outperform other methods. The failure rate is greatly reduced.

AFLW: Table 4 reports the evaluation results obtained on the AFLW dataset. 'Vanilla CNN (SepBN)' achieves a lower error rate than other methods. The improvement on

Table 5: A comparison with the SOTA methods on WFLW, in terms of NME ($\times 10^{-2}$).

	All	Pose	Expr.	Illu.	Mu.	Occu.	Blur
DVLN [34]	6.08	11.54	6.78	5.73	5.98	7.33	6.88
LAB [33]	5.27	10.24	5.51	5.23	5.15	6.79	6.32
RWing [10]	5.60	9.79	6.16	5.54	6.65	7.05	6.41
3FabRec [9]	5.62	10.23	6.09	5.55	5.68	6.92	6.38
LUVLi [11]	4.36	-	-	-	-	-	-
Vanilla CNN (SepBN)	5.48	9.97	5.89	5.36	5.55	6.83	6.25

Table 6: The results obtained on COFW, AFLW-Full and WFLW using different configurations. $\downarrow$ indicate the change of NME ($\times 10^{-2}$) compared with the corresponding baseline network.

	COFW	WFLW-test	AFLW-Full
Vanilla CNN (BN)	4.07	5.70	1.65
Vanilla CNN (SepBN)	$4.03_{\downarrow 0.04}$	$5.48_{\downarrow 0.22}$	$1.55_{\downarrow 0.10}$
MobileNetV2 (BN)	5.07	6.18	1.66
MobileNetV2 (SepBN)	$3.96_{\downarrow 1.11}$	$5.31_{\downarrow 0.87}$	$1.57_{\downarrow 0.09}$
ResNeXt-50 (BN)	3.95	4.90	1.50
ResNeXt-50 (SepBN)	$3.51_{\downarrow 0.44}$	$4.85_{\downarrow 0.05}$	$1.46_{\downarrow 0.04}$

AFLW_Full is more obvious than that on AFLW_Frontal, which proves that the network using SepBN gains higher localization accuracy for non-frontal images than before.

WFLW: The evaluation results obtained on the full test set and each subset are shown in Table 5. Although the backbone network architecture (Vanilla CNN) is very simple, after applying the SepBN module, 'Vanilla CNN (SepBN)' achieves a performance comparable to most advanced methods.

Applicable Scenarios: We evaluate our Vanilla CNN, MobileNetV2 and ResNeXt-50 equipped with BN and SepBN on COFW, WFLW and AFLW in Table 6. The layout of the table is carefully designed. The size of the three datasets COFW, AFLW, and WFLW increase sequentially ($1345 \rightarrow 7500 \rightarrow 20000$). The complexity of the three networks Vanilla CNN, MobileNetV2, and ResNeXt-50 is also different. In particular, Vanilla CNN is simple and straightforward, but it is large. MobileNetV2 is lightweight, yet compact. ResNeXt-50 is large and well-designed. Compared with our Vanilla CNN, MobileNetV2 and ResNeXt-50 are generally more powerful.

The results are shown in Table 6. 'Vanilla CNN (BN)', 'MobileNetV2 (BN)' and 'ResNeXt-50 (BN)' are the baseline networks equipped with classical BN layers. Then the SepBN modules are used to replace the original BN_3 in all the Bottleneck blocks, forming 'MobileNetV2 (SepBN)' and 'ResNeXt-50 (SepBN)'.

We first examine the performance of our Vanilla CNN. As the size of dataset continues to grow, the performance improvement brought by SepBN is more significant (COFW: 0.04 < 0.44 < 1.11, WFLW: 0.05 < 0.22 < 0.87, AFLW: 0.04 < 0.09 < 0.10). This shows that even for a network with a simple design, a performance improvement is achievable by using our SepBN module. Note, the improvement will become more prominent when more data becomes available. It is clear that, compared with classic BN layer, our SepBN module can make better use of big data in the case of simple network.

For advanced neural networks, the proposed SepBN module improves the performance of both 'MobileNetV2 (BN)' and 'ResNeXt-50 (BN)' on different datasets. The rate of improvement slows down as the dataset grows larger. The reduced error rate exhibited by the *lightweight* MobileNetV2 network equipped with SepBN is quite significant. The test set results achieved by 'MobileNetV2 (SepBN)' are even better than those achieved by the Vanilla CNN in the same configuration on COFW and WFLW. Even the most powerful network,

Table 7: A comparison of SepBN, AN and classical BN on CIFAR-100. Different proportions of training images are randomly selected for network training.

Method	100%	50%	25%	20%
MobileNetV2 (BN)	67.7	61.49	52.95	47.54
MobileNetV2 (AN)	68.21	61.79	53.05	47.41
MobileNetV2 (SepBN)	**68.44**	**62.22**	**53.20**	**47.73**

ResNeXt-50, can benefit from the SepBN module, especially on a small dataset, while the improvement becomes negligible on larger datasets.

The above experiments indicate two clean use cases for SepBN. One scenario is when the network is weak but the data is sufficient, *e.g.*, 'Vanilla CNN (SepBN)' on AFLW. The other one is when the network is powerful (like 'MobileNetV2' and 'ResNeXt-50') but the dataset size is small, *e.g.*, 'MobileNetV2 (SepBN)' on COFW. This reveals the dependence of the SepBN module on feature diversity.

4.4 Results on CIFAR-100

In order to further demonstrate the versatility and effectiveness of the proposed SepBN module for other computer vision tasks, we evaluate the performance of SepBN on the CIFAR-100 [16] dataset for image classification. We also compare the proposed SepBN module with the classical BN and the state-of-the-art AN layers.

We first constructed different small and unevenly distributed training subsets by randomly sampling the original training set. The test set remains the same as the original test set. All the models were trained from scratch and we did not use any pre-training methods. The experimental results are shown in Table 7. The results demonstrate that SepBN can improve model capacity even when the training dataset is very small. Note that when using 20% of the data for training, the accuracy after using AN is even lower than the original network. The demonstrates that, as compared with AN, the design of SepBN can deal with the overfitting problem more effectively when training a deep network on a small dataset.

5 Conclusion

We presented a Separable Batch Normalization (SepBN) module for robust facial landmark localization. The aim is to deal with the small-sample-size problem and data imbalance for deep network training. The new module combines BN, attention mechanism and channel grouping in a novel manner to map the normalized features adaptively and efficiently. Compared with the existing BN variants, the method of modifying the mapping operation of BN by SepBN helps to achieve a consistent performance improvement. A comprehensive set of experiments verified the merits of our SepBN module and identified its use scenarios.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61773117, 62006041, and 61902153) and by the U.K. Engineering and Physical Sciences Research Council (EP/R013616/1 and EP/V002856/1).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Bjorn Browatzki and Christian Wallraven. 3fabrec: Fast few-shot face alignment by reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6110–6120, 2020.

[3] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1513–1520, 2013.

[4] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.

[5] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[6] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[7] Zhen Cui, Shengtao Xiao, Zhiheng Niu, Shuicheng Yan, and Wenming Zheng. Recurrent shape regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (5):1271–1278, 2018.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[9] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2235–2245, 2018.

[10] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, and Xiao-Jun Wu. Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks. *International Journal of Computer Vision*, pages 1–20, 2019.

[11] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. *arXiv preprint arXiv:2009.09960*, 2020.

[12] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015.

[13] Sergey Ioffe. Batch renormalization: towards reducing minibatch dependence in batch-normalized models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1942–1950, 2017.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2144–2151. IEEE, 2011.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[17] Abhinav Kumar, Tim K Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8236–8246, 2020.

[18] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2852–2861, 2017.

[19] Xilai Li, Wei Sun, and Tianfu Wu. Attentive normalization. *arXiv preprint arXiv:1908.01259*, 2019.

[20] Senwei Liang, Zhongzhan Huang, Mingfu Liang, and Haizhao Yang. Instance enhancement batch normalization: An adaptive regulator of batch noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4819–4827, 2020.

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017.

[22] Zhiwei Liu, Xiangyu Zhu, Guosheng Hu, Haiyun Guo, Ming Tang, Zhen Lei, Neil M Robertson, and Jinqiao Wang. Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3467–3476, 2019.

[23] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[24] Anthony Ortiz, Caleb Robinson, Dan Morris, Olac Fuentes, Christopher Kiekintveld, Md Mahmudulla Hassan, and Nebojsa Jojic. Local context normalization: Revisiting local normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11285, 2020.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[27] Wenqi Shao, Tianjian Meng, Jingyu Li, Ruimao Zhang, Yudian Li, Xiaogang Wang, and Ping Luo. Ssn: Learning sparse switchable normalization via sparsestmax. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 443–451, 2019.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[29] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition*, pages 1701–1708, 2014.

[30] Zhiqiang Tang, Xi Peng, Kang Li, and Dimitris N Metaxas. Towards efficient u-nets: A coupled and quantized approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[31] Brian Teixeira, Birgi Tamersoy, Vivek Singh, and Ankur Kapoor. Adaloss: Adaptive loss function for landmark localization. *arXiv preprint arXiv:1908.01070*, 2019.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[33] Robert Walecki, Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Copula ordinal regression for joint estimation of facial action unit intensity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4902–4910, 2016.

[34] Xinyao Wang, Liefeng Bo, and Fuxin Li. Adaptive wing loss for robust face alignment via heatmap regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6971–6981, 2019.

[35] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2138, 2018.

[36] Wenyan Wu and Shuo Yang. Leveraging intra and inter-dataset variations for robust face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 150–159, 2017.

[37] Yue Wu and Qiang Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3400–3408, 2016.

[38] Yue Wu, Chao Gou, and Qiang Ji. Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3471–3480, 2017.

[39] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[41] Jing Yang, Qingshan Liu, and Kaihua Zhang. Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 79–87, 2017.

[42] Zhuliang Yao, Yue Cao, Shuxin Zheng, Gao Huang, and Stephen Lin. Cross-iteration batch normalization. *arXiv preprint arXiv:2002.05712*, 2020.

[43] Michal Zajac, Konrad Zolna, and Stanislaw Jastrzebski. Split batch normalization: Improving semi-supervised learning under domain shift. *arXiv preprint arXiv:1904.03515*, 2019.

[44] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2008.

[45] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[46] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq. Robust facial landmark detection via occlusion-adaptive deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3486–3496, 2019.

[47] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, 2017.