# Learning to Learn Cropping Models for Different Aspect Ratio Requirements

Debang Li[1,2], Junge Zhang[1,2], Kaiqi Huang[1,2,3]
[1] CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
{debang.li,jgzhang, kaiqi.huang}@nlpr.ia.ac.cn

## Abstract

*Image cropping aims at improving the framing of an image by removing its extraneous outer areas, which is widely used in the photography and printing industry. In some cases, the aspect ratio of cropping results is specified depending on some conditions. In this paper, we propose a meta-learning (learning to learn) based aspect ratio specified image cropping method called Mars, which can generate cropping results of different expected aspect ratios. In the proposed method, a base model and two meta-learners are obtained during the training stage. Given an aspect ratio in the test stage, a new model with new parameters can be generated from the base model. Specifically, the two meta-learners predict the parameters of the base model based on the given aspect ratio. The learning process of the proposed method is learning how to learn cropping models for different aspect ratio requirements, which is a typical meta-learning process. In the experiments, the proposed method is evaluated on three datasets and outperforms most state-of-the-art methods in terms of accuracy and speed. In addition, both the intermediate and final results show that the proposed model can predict different cropping windows for an image depending on different aspect ratio requirements.*

## 1. Introduction

Image cropping is commonly used in image editing, trying to find a good view with a better composition than the input image. Automatic image cropping can be widely applied in the photographic, printing industry, and other related fields for saving time. Depending on the application, the aspect ratio of the cropped image may be specified and vary with different conditions. As such, the aspect ratio specified image cropping algorithms should be able to cover a range of aspect ratios, an illustration of which is shown in Figure 1.

Early researches on the general image cropping mostly focus on the two-stage methods [31, 32, 13, 6, 7, 19]. Many candidates are generated at the first stage and ranked on the second stage. These two-stage methods can be directly transferred to the aspect ratio specified settings by adjusting the candidates. Since there are many candidates in an image, the speed of these methods is inevitably slow. To speed up, several methods [40, 25, 41, 26] obtain the cropping window directly without using the sliding window. However, these methods rarely consider the aspect ratios. In [12], an object detection based approach is proposed for the aspect ratio specified image cropping by adding more prediction heads.

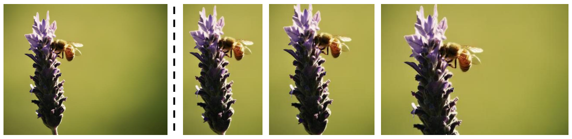


Figure 1. **Illustration of the aspect ratio specified image cropping.** The left image is the original image, and the three images on the right are cropped images of different required aspect ratios.

In this paper, we regard generating the cropped images of a specified aspect ratio as an isolated task and adopt a single model to accomplish multiple such sub-tasks. The model should be able to adapt to many environments with different aspect ratio requirements. Therefore, we propose a meta-learning (*learning to learn*) based aspect ratio specified image cropping approach (called *Mars*) to accomplish this goal. In the proposed approach, we train a base model and two meta-learners during the training process. In the inference stage, a new model with new parameters is generated from the base model given a new aspect ratio. Specifically, some parameters of the base model are predicted by the meta-learners depending on the required aspect ratio. As the required aspect ratio is a continuous value, the number of models with different parameters is infinite. The learning process of the proposed method can be viewed as learning how to learn cropping models for different aspect ratios. In the base model, the parameters depending on the required aspect ratio are the aspect ratio specified feature transformation matrix (ARS-FTM) and the aspect ratio specified pixel-wise predictor (ARS-PWP). When both ARS-FTM and ARS-PWP are determined by the meta-learners, the newly generated model can predict the cropping window of the specified aspect ratio from the image. In the experiments, both the quantitative and qualitative results show that the proposed meta-learning based approach can generate cropping windows of required aspect ratios effectively and efficiently.

The main contributions of this work are:

- We propose a meta-learning based method that can predict the cropping results of arbitrary aspect ratios using a single model.
- We develop an aspect ratio embedding method and two aspect ratio specified modules (*i.e.*, ARS-FTM and ARS-PWP) to model the aspect ratio information and map the aspect ratio to the parameters of the model.
- We show that the proposed algorithm achieves state-of-the-art performance on both the quantitative evaluation and user study and can run in real-time (over 100 FPS).

## 2. Related Work

**Image Cropping.** Most early researches on image cropping focused on the sliding window based two-stage operations. According to the standards of ranking the candidates generated by the sliding window, these methods can be divided into two groups, the attention-based and aesthetics-based methods. The attention-based methods [30, 32, 37, 13, 4] usually rank candidates according to the attention score obtained by saliency detection [30]. As such, the cropping windows can preserve the main subjects and draw more attention from people. However, they may fail to generate visually pleasing results due to the lack of considering the image composition [6]. Those aesthetics-based methods [19] try to find the most visually pleasing cropping window from the input image. Some methods [31, 8, 42, 13] design a set of hand-crafted features to evaluate the aesthetics, and other methods [17, 6, 7, 41] train aesthetics discriminators from data to rank cropping candidates. Several methods [36, 9, 35, 4, 25, 26] are developed to search for the optimal cropping window more efficiently instead of evaluating all candidates to speed up the sliding window based methods.

Fast-AT [12] is designed for the aspect ratio specified image cropping by plugging several predicting heads for different aspect ratio intervals to the object detection model [10]. In the proposed meta-learning based approach, we do not have to train different filters separately for different aspect ratios, but use a single model to adapt to different aspect ratio requirements, where the aspect ratio specified parameters are predicted by the meta-learners. Image retargeting methods [1, 27] adjust the images to fit the target aspect ratio while keeping the import contents, which are related to our task. However, image cropping aims to find the best window on the image that satisfies the requirement, while image retargeting concentrates on content-aware image resizing, the experimental settings between these two tasks are different.

**Meta-Learning.** Meta-learning is also known as *learning to learn*, which means the machine learning algorithms can learn how to learn the knowledge. In other words, the model needs to be aware of and take control of its learning [24]. Through these properties of meta-learning, models can be more easily adapted to different environments and tasks, rather than considering each one separately. Due to these reasons, meta-learning has been widely applied in hyper-parameter optimization [29], neural network optimization [5], few-shot learning [14], fast reinforcement learning [38], and visual tracking [2, 24].

In this paper, our goal is to solve cropping problems for different aspect ratio requirements with a single model. Regarding generating cropping results for a specified aspect ratio as an isolated task, the above goal can be naturally solved by meta-learning. The weight prediction is one of the meta-learning strategies [23], which can adapt models to different environments by dynamically predicting the weight of the model [3, 16, 43]. The proposed method also belongs to this category, which predicts the weight of the model depending on the aspect ratio information. Kishore *et al.* [21] use the adaptive convolution [18] for the final classification and regression according to a scalar input value (the aspect ratio). In contrast, our method proposes to uses the embedding interpolation for the aspect ratio representation and also generates an ARS-FTM module for the global feature transformation in the middle stage, which can encode the aspect ratio information in the global feature representation. In addition, our model can run much faster (over 100 FPS).

## 3. Proposed Algorithm

### 3.1. Problem Formulation

In this section, we formulate the aspect ratio specified image cropping problem and the proposed meta-learning based approach (Mars). For general image cropping problems, the model takes an image $x_i$ as input and outputs a visually pleasing cropping window $y_i$, which is

$$y_i = \mathcal{F}(x_i; W), \tag{1}$$

where $W$ represents the parameters of the model $\mathcal{F}$. Different from the general setting, the aspect ratio specified image cropping has an additional aspect ratio requirement, which is

$$y_i^{(\tau_i)} = \mathcal{F}(x_i, \tau_i; W), \tag{2}$$

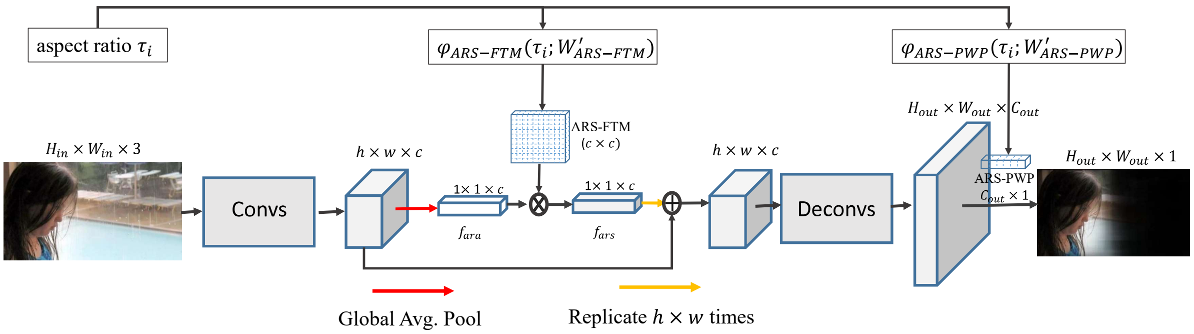

aspect ratio $\tau_i$
$\varphi_{ARS-FTM}(\tau_i; W'_{ARS-FTM})$
$\varphi_{ARS-PWP}(\tau_i; W'_{ARS-PWP})$
$H_{in} \times W_{in} \times 3$
Convs
$h \times w \times c$
$1 \times 1 \times c$
$f_{ara}$
ARS-FTM
$(c \times c)$
$1 \times 1 \times c$
$f_{ars}$
$h \times w \times c$
Deconvs
$H_{out} \times W_{out} \times C_{out}$
ARS-PWP
$C_{out} \times 1$
$H_{out} \times W_{out} \times 1$
Global Avg. Pool
Replicate $h \times w$ times

Figure 2. **Overview of the proposed model.** The numbers above each feature map represent the shape of the feature map ($height \times width \times channel$).

where $\tau_i$ is the required aspect ratio, and $y_i^{(\tau_i)}$ is the cropping result with an aspect ratio of $\tau_i$.

In this paper, we propose a meta-learning based approach that can generate model parameters for different $\tau_i$ continuously. Specifically, a sub-network (meta-learner) is used to map $\tau_i$ to the model parameters, which is

$$W = \varphi(\tau_i; W'), \tag{3}$$

where $W'$ is the parameters of the meta-learner $\varphi$. Since $\tau_i$ is a continuous value, the number of models with different parameters generated by the meta-learner can be infinite. The proposed approach can be finally formulated as

$$y_i^{(\tau_i)} = \mathcal{F}(x_i; \varphi(\tau_i; W')), \tag{4}$$

where the model parameters incorporate the aspect ratio information and will change accordingly.

### 3.2. Architecture Overview

With the previous formulation, we start to introduce the proposed meta-learning framework, which contains a base model and two meta-learners. The architecture and details of the proposed framework are illustrated in Figure 2.

There are two inputs of the framework, the image and required aspect ratio ($\tau_i$). At first, the aspect-ratio-agnostic feature vector $f_{ara}$ is extracted from the input image through the convolution blocks (backbone network) and a global average pooling (GAP) operation, which is the feature representation of the input image without considering the required aspect ratio. After that, $f_{ara}$ is transformed into an aspect-ratio-specified feature vector $f_{ars}$ by an aspect-ratio-specified feature transformation matrix (ARS-FTM), which is a fully-connected layer whose parameters are predicted by a meta-learner depending on $\tau_i$. In this way, the image feature and the aspect ratio information are both embedded in $f_{ars}$. Then $f_{ars}$ is added to each location of the last feature map before the GAP layer to generate a new feature map. The new feature map retains the original spatial information and also incorporates the global feature and aspect ratio information. The details of the feature transformation process are shown in Figure 3.

The new feature map is fed into several cascaded deconvolution layers (the upsampling module) to increase its spatial resolution to $H_{out} \times W_{out}$. Each deconvolution layer doubles the resolution and keeps the same channel dimension ($C_{out}$). After that, an aspect-ratio-specified pixel-wise predictor (ARS-PWP), which is a $1 \times 1$ convolution layer predicted by a meta-learner, is used to predict the cropping area. The prediction is finally normalized by a sigmoid function, and the cropping window of the required aspect ratio is generated through a post-processing process (see Section 3.4).

In general, the parameters of machine learning models are fixed in the test stage. However, the parameters of ARS-FTM and ARS-PWP vary depending on the required aspect ratio during the test, which can be interpreted as a new model for a new aspect ratio. With meta-learning, we can generate models for arbitrary aspect ratio requirements. Even these aspect ratios do not appear in the training stage.

### 3.3. Aspect Ratio Specified Module

In this section, we introduce the meta-learners that map the aspect ratio to the parameters of the base model. As shown in Figure 2, there are two modules whose parameters are determined by $\tau_i$, namely ARS-FTM and ARS-PWP. According to Equation 3, the map functions of these two modules can be written as

$$W_{ARS-FTM} = \varphi_{ARS-FTM}(\tau_i; W'_{ARS-FTM}) \tag{5}$$

and

$$W_{ARS-PWP} = \varphi_{ARS-PWP}(\tau_i; W'_{ARS-PWP}). \tag{6}$$

The output of $\varphi_{ARS-FTM}$ is a matrix that can transform the aspect-ratio-agnostic feature into the aspect-ratio-specified feature space, and the output of $\varphi_{ARS-PWP}$ is a $1 \times 1$ convolution layer that predicts the cropping area.

In this paper, we use a fully-connected network with two outputs to implement the above two map functions. Since

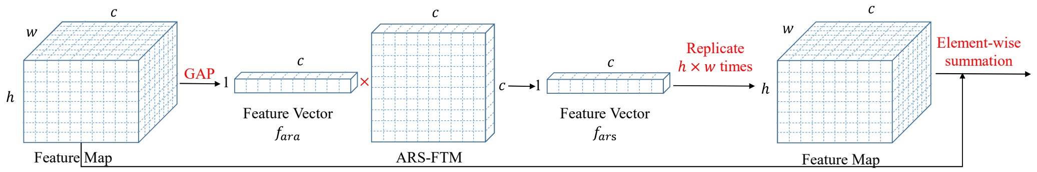


Figure 3. **Illustration of the feature transformation process.** The obtained feature map retains the original spatial information and also incorporates the global information (GAP) and the aspect ratio information (ARS-FTM). The above symbols are described in Figure 2.

the aspect ratio $\tau_i$ is a scalar, directly mapping $\tau_i$ to a high-dimensional space may not perform well, which is also verified in the following experiment sections (see Section 4.2). Instead, we use the embedding vectors and linear interpolation to represent the continuous $\tau_i$.

First, we select $N$ aspect ratios, each with a corresponding embedding vector. The set of selected aspect ratios is denoted as $\mathcal{S}_\tau$, and the corresponding set of embedding vectors is $\mathcal{S}_{emb}$. To generate the embedding vector of an arbitrary $\tau_i$, we use the linear interpolation of two embedding vectors from $\mathcal{S}_{emb}$ whose corresponding aspect ratios are the closest to $\tau_i$. Following [12], the range of aspect ratio is from 0.5 to 2, which is $\tau_i \in [0.5, 2]$. When choosing the $N$ aspect ratios in $\mathcal{S}_\tau$, we want to make the number of chosen aspect ratios in [0.5, 1) and (1, 2] equal, because the shape of the image in these two intervals is symmetrical (rotated 90°), such as 3:4 and 4:3. For this purpose, we use the logarithmic transformation to map $\tau_i$ to $\log \tau_i$ and choose $\log \tau_i$ in $[-\log 2 \ (\log 0.5), \log 2]$ evenly with a step size of $\frac{2\log 2}{N-1}$, where $N$ is an odd number.

Since the aspect ratio is equally spaced in the logarithmic space, linear interpolation is also performed in the logarithmic space to generate the embedding vector $E(\tau_i)$ of arbitrary $\tau_i$, which is

$$
\begin{aligned}
E(\tau_i) =& \frac{\log \tau_i^{(upper)} - \log \tau_i}{\frac{2\log 2}{N-1}} \times E(\tau_i^{(lower)}) \\
&+ \frac{\log \tau_i - \log \tau_i^{(lower)}}{\frac{2\log 2}{N-1}} \times E(\tau_i^{(upper)}),
\end{aligned}
\tag{7}
$$

where $\tau_i^{(upper)}$ and $\tau_i^{(lower)}$ are the two adjacent aspect ratios of $\tau_i$ in $\mathcal{S}_\tau$, satisfying $\tau_i^{(upper)} > \tau_i > \tau_i^{(lower)}$. Since $\tau_i$ is a continuous value, the number of the embedding vectors generated by the linear interpolation is infinite. Embedding vectors from $\mathcal{S}_{emb}$ are all trainable in the training stage, and new embedding vectors for new aspect ratios can be generated in the test stage. The dimension of the embedding vectors is 512.

After obtaining the embedding vector of the required aspect ratio, we use a fully-connected network with two outputs to implement the two meta-learners, which map the embedding vector to the model parameters. The architecture of the meta-learners is shown in Figure 4. When a newly required aspect ratio is given, the outputs of the sub-network are reshaped to the target shape and plugged into the base model to form a new model with new parameters.

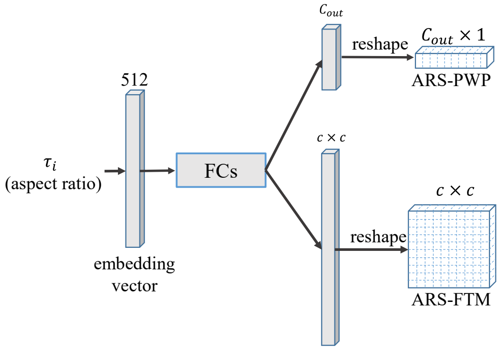


Figure 4. **Illustration of the aspect-ratio-specified modules.** We translation the aspect ratio $\tau_i$ (1-d) to the embedding vector (512-d) using Equation 7. Then the sub-network maps the embedding vector to the parameters of the base model. The channel dimension of $f_{ara}$ in Figure 2 is $c$. Because the channel dimension of the feature map outputted by the upsampling module is $C_{out}$, ARS-PWP is reshaped to $C_{out} \times 1$, which means the number of input channels is $C_{out}$, and the number of the output channels is 1.

### 3.4. Training and Inference

During the training process, the target value of the pixels in the cropping area is 1, and the value of the rest is 0. Binary cross entropy (BCE) loss is used to compute the loss function, which is

$$
L(p, g) = -\frac{1}{N_{pixel}} \sum_i [g_i \log p_i + (1-g_i) log(1-p_i)], \tag{8}
$$

where $p$ and $g$ are the prediction and ground truth values, respectively, $N_{pixel}$ is the number of the pixels, and $i$ is the indicator of the pixel position. The meta-learners do not have other supervisions, and the entire model is trained with the BCE loss in an end-to-end manner.

In the inference stage, after obtaining the prediction of the network, we use a post-processing process to get the cropping result. First, the prediction is binarized using a threshold $\theta$. Then, the center of the cropping result is obtained by computing the median of the coordinates of all positions whose value is 1. We sum the values of each column (or row) and select the median of those non-zero results as the height (or width). After that, the height or width is reduced to meet the aspect ratio requirement, while the other one keeps unchanged. Finally, the cropping window is determined by the center, width, and height.

# 4. Experimental Results

## 4.1. Experimental Settings

**Data and Metrics.** In the experiments, we adopt the training set provided by FAT [12] to train the proposed framework, which contains 24,154 images with 63,043 annotations. Each image has up to 3 annotations with an aspect ratio in [0.5, 2]. We evaluate the proposed method on three image cropping datasets, including the HCDB [13], FCDB [6], and FAT. HCDB contains 500 images, and each image is annotated by 10 different experts. FCDB contains 343 testing images, and each image has a single annotation. The test set of FAT contains 3,910 images with 7,005 annotations. To show the generalization of the proposed model, we evaluate the model trained with the training set of FAT on the above three datasets without additional training.

Following existing methods [41], we use the average intersection-over-union ratio (IoU) and average boundary displacement error (BDE) as performance evaluation metrics for FCDB and HCDB and employ the average IoU and average center offset to evaluate different methods for FAT.

**Implementation Details.** The backbone network is pretrained on the ImageNet [11]. The longest edge of the input image is resized to 256, while the aspect ratio keeps unchanged. The mini-batch size for training is 32. Adam algorithm [20] is used to optimize the model, while the learning rate is set to $1e^{-4}$. The weight decay for the base model is $1e^{-4}$, and for the meta-learners is $1e^{-}3$. The model is trained for 50 epochs on the training set, during which warmup [15] is adopted in the first 5 epochs and cosine learning rate decay [28] is used in the following 45 epochs. The number of chosen aspect ratios in $\mathcal{S}_\tau$ $(N)$ is set to 101. The threshold $\theta$ for the binarization in section 3.4 is set to 0.4 through the grid search on the training set.

## 4.2. Ablation Study

In this section, we conduct a series of experiments to determine the backbone network, the aspect ratio specified module, and the upsampling module. During the ablation study, we choose 1000 training images with 2357 annotations from the training set as the validation set and use other training images to train the models.

### 4.2.1 Backbone Network

First, we conduct experiments to determine the backbone network of the proposed model. The running speed is critical for the image cropping since it usually runs on mobile devices or laptops. We consider both the accuracy and complexity of models when choosing the backbone network. We choose three networks (MobileNetV2 [33], VGG16 [34], and ResNet50 [15]) truncated at different layers as candidates, and keep other experimental settings the same. The *FCs* in Figure 4 is implemented by a 1-layer fully-connected network with 512 neurons. The output of the model ($H_{out} \times C_{out}$) is up-sampled to $H_{in}/4 \times W_{in}/4$, and the channel dimension of all deconvolution layers ($C_{out}$) is 96 (see Figure 2). The results on the validation set are shown in Table 1.

Table 1. **Ablation study of the backbone network on the validation set.** The $cx\_y$ in the *layer* column means the model is truncated after the $y$-th convolution layer whose output resolution ($h \times w$ in Figure 2) is $H_{in}/2^x \times W_{in}/2^x$. The parameter size of the model (param), speed, and cropping accuracy (IoU and offset) are evaluated for different backbone networks.

| Backbone | Layer | Param | Speed↑ | IoU↑ | Offset↓ |
|---|---|---|---|---|---|
| MobileV2 | *c3_3* | 1.0M | 127FPS | 0.652 | 65.1 |
| | *c4_3* | 2.7M | 115FPS | 0.688 | 53.8 |
| | *c4_6* | 5.6M | 108FPS | **0.706** | **49.8** |
| | *c5_1* | 14.3M | 96FPS | 0.705 | 50.9 |
| VGG16 | *c3_3* | 142.6M | 110FPS | 0.672 | 64.7 |
| | *c4_1* | 145.0M | 107FPS | 0.693 | 52.6 |
| | *c4_3* | 149.7M | 103FPS | **0.702** | **51.1** |
| | *pool5* | 149.7M | 102FPS | 0.698 | 51.5 |
| ResNet50 | *c3_4* | 136.4M | 115FPS | 0.668 | 58.8 |
| | *c4_3* | 140.7M | 96FPS | 0.699 | 51.3 |
| | *c4_6* | 144.0M | 86FPS | 0.702 | 50.6 |
| | *c5_1* | 150.6M | 81FPS | **0.705** | **50.1** |

From Table 1, we have the following observations: 1) For each model, truncated at shallow layers may lead to unsatisfied performance (*e.g.*, *c3_y*). With a deeper network and more parameters, the performance also increases but plateaus when the complexity is too high (*e.g.*, *c5_y*). 2) Surprisingly, the best performance of the above three models is similar. Although ResNet50 can significantly surpass MobileNetV2 in the ImageNet classification [11], it fails to improve the performance of the proposed method. This may be because the number and distribution of the training samples limit further performance gains for image cropping. Considering the performance and running speed, we choose the MobileNetV2 (truncated after *c4_6* layer) as the backbone network in the following experiments unless stated. As such, $h \times w \times c$ in Figure 2 is equal to $H_{in}/16 \times W_{in}/16 \times 96$.

### 4.2.2 Aspect Ratio Specified (ARS) Module

In this section, we conduct experiments to determine the model size of the ARS module and analyze the necessity of each component. As shown in Figure 4, the embedding of the target aspect ratio is passed through several fully-connected layers (*FCs*) and then transformed into the parameters of the base model. First, we evaluate the FCs of different sizes on the validation set and keep other modules the same as the ablation study of the backbone network. The

Table 2. **Ablation study on the model size of the aspect ratio specified module on the validation set.** The $FC512 \times n$ means there are n fully-connected (FC) layers with 512 neurons for the feature representation (*FCs* in Figure 4), and $FC512 \times 0$ means the embedding is directly mapped to the parameters without intermediate FC layers.

| Model size | Param | Speed↑ | IoU↑ | Offset↓ |
|---|---|---|---|---|
| $FC512 \times 0$ | 5.3M | 110FPS | 0.701 | 50.3 |
| $FC512 \times 1$ | 5.6M | 108FPS | **0.706** | **49.8** |
| $FC512 \times 2$ | 5.9M | 108FPS | 0.704 | 50.1 |
| $FC512 \times 3$ | 6.1M | 107FPS | 0.704 | 50.3 |
| $FC512 \times 4$ | 6.4M | 105FPS | 0.704 | 50.1 |

Table 3. **Ablation study on each component of the aspect ratio specified module on the validation set.**

| Model | IoU↑ | Offset↓ |
|---|---|---|
| Ours w/o ARS-FTM & ARS-PWP | 0.665 | 53.6 |
| Ours w/o ARS-FTM | 0.694 | 52.6 |
| Ours w/o ARS-PWP | 0.696 | 52.2 |
| Ours | **0.706** | **49.8** |

Table 4. **Ablation study on the the aspect ratio embedding method on the validation set.**

| Embedding vector of aspect ratios | IoU↑ | Offset↓ |
|---|---|---|
| w/o aspect ratio embedding vector | 0.689 | 52.5 |
| w/o embedding interpolation | 0.704 | 50.6 |
| proposed | **0.706** | **49.8** |

results are shown in Table 2, where we increase the number of FC layers and keep the number of neurons in each layer at 512. Table 2 shows that a shallow network (1-layer) can obtain a pleasant result, and deeper architectures do not improve the performance. As such, we use the 1-layer FC (with 512 neurons) to implement the *FCs* of Figure 4 in the following parts.

Second, we study the influence of each component in the ARS module, *e.g.*, the ARS-FTM and ARS-PWP (see section 3.3). The ablation study results are shown in Table 3. When removing ARS-FTM from the model (*Ours w/o ARS-FTM*), $f_{ars}$ is identical to $f_{ara}$ in Figure 2. When removing ARS-PWP (*Ours w/o ARS-PWP*), we replace it with a standard $1 \times 1$ convolution layer, the parameters of which are fixed after training. When the meta-learning approach is abandoned (*Ours w/o ARS-FTM & ARS-PWP*), the performance drops dramatically. After plugging ARS-FTM or ARS-PWP to the model, the performance is improved significantly. The model without ARS-PWP outperforms the model without ARS-FTM, showing that ARS-FTM plays a more critical part than ARS-PWP in the proposed model. Overall, the model plugged with both modules achieves the best performance.

Third, we study the influence of the proposed aspect ratio embedding method (see Section 3.3). In Table 4, we employ simpler ways to represent the aspect ratio information. For "*w/o aspect ratio embedding vector*", the input of the meta-learner is not the embedding vector but the value of the aspect ratio directly. For "*w/o embedding interpolation*", the proposed model predicts the map using the aspect ratio in $\mathcal{S}_\tau$ which is the closest to the required one. After that, the post-processing is used to resize the cropping window to the target size. Table 4 shows that the model using the proposed embedding method outperforms the other two baselines. The reason can be interpreted as that the proposed embedding method can contain more useful information to make the model find better cropping results that satisfy the aspect ratio requirements. We also study the number and dimension of the embedding vectors in Table 5 and find that increasing the number and dimension of the embedding vectors both help improve the performance, but the gain is marginal when they are too big. As such, we set the number and dimension of the embedding vectors to 101 and 512, respectively.

Table 5. **Ablation study on the *number* and *dimension* of the aspect ratio embedding vectors on the validation set.**

| Number | Dimension | IoU↑ | Offset↓ |
|---|---|---|---|
| 11 | 512 | 0.702 | 51.5 |
| 101 | 512 | 0.706 | 49.8 |
| 201 | 512 | 0.707 | 49.8 |
| 501 | 512 | **0.709** | **49.5** |
| 101 | 128 | 0.699 | 51.6 |
| 101 | 256 | 0.700 | 50.7 |
| 101 | 512 | 0.706 | 49.8 |
| 101 | 1024 | **0.708** | **49.4** |

Table 6. **Ablation study on the upsampling module with different output resolution ($H_{out} \times W_{out}$) and output channel dimension ($C_{out}$).** The first column ($H_r \times W_r$) is the ratio of output resolution to input resolution ($H_r = H_{out}/H_{in}, W_r = W_{out}/W_{in}$).

| $H_r \times W_r$ | $C_{out}$ | Param | speed↑ | IoU↑ | Offset↓ |
|---|---|---|---|---|---|
| $1/16 \times 1/16$ | 96 | 5.5M | 113FPS | 0.695 | 52.1 |
| $1/8 \times 1/8$ | 96 | 5.6M | 110FPS | 0.702 | 52.0 |
| $1/4 \times 1/4$ | 96 | 5.6M | 108FPS | 0.706 | 49.8 |
| $1/2 \times 1/2$ | 96 | 5.6M | 105FPS | 0.705 | 50.3 |
| $1 \times 1$ | 96 | 5.7M | 101FPS | **0.708** | **49.5** |
| $1/4 \times 1/4$ | 32 | 5.5M | 108FPS | 0.703 | 50.4 |
| $1/4 \times 1/4$ | 64 | 5.5M | 108FPS | 0.704 | 50.2 |
| $1/4 \times 1/4$ | 96 | 5.6M | 108FPS | **0.706** | **49.8** |
| $1/4 \times 1/4$ | 128 | 5.7M | 107FPS | 0.703 | 50.1 |
| $1/4 \times 1/4$ | 256 | 6.2M | 105FPS | 0.703 | 50.0 |

### 4.2.3 Upsampling module

The backbone network and the aspect ratio specified module have been determined. Now we conduct experiments to determine the upsampling module. As shown in Figure 2, after the feature transformation (from $f_{ara}$ to $f_{ars}$), the feature map is upsampled to $H_{out} \times W_{out} \times C_{out}$ with several deconvolution layers. In the implementation, each deconvolution layer doubles the resolution and keeps the same channel dimension ($C_{out}$). In this section, we study the influence of different output resolutions ($H_{out} \times W_{out}$) and different channel dimensions ($C_{out}$). The ablation study results are shown in Table 6. Upsampling the resolution of the output feature maps does help improve the perfor-

Table 7. **Comparisons against state-of-the-art methods on three datasets.** For HCDB and FCDB, the modified aspect ratio specified results (before the brackets) and the results from their original papers (in the brackets) are both shown in each column. Except that VFN and A2RL are based on Alexnet [22] and Fast-AT is based on ResNet101, other methods are all based on VGG16, so we also show the results of the proposed model using VGG16 (truncated after *c4_3* layer) as the backbone ($h \times w \times c$ in Figure 2 is equal to $H_{in}/16 \times H_{in}/16 \times 512$).

| Method | Backbone | Speed↓ | HCDB [13] | | FCDB [6] | | FAT [12] | |
|---|---|---|---|---|---|---|---|---|
| | | | IoU↑ | BDE↓ | IoU↑ | BDE↓ | IoU↑ | Offset↓ |
| **Speed $<$ 1 FPS** | | | | | | | | |
| VFN [7] | AlexNet | 0.5 FPS | 0.848 (-) | 0.034 (-) | 0.687 (0.684) | 0.077 (0.084) | 0.617 | 74.7 |
| VEN [41] | VGG16 | 0.2 FPS | 0.852 (0.837) | 0.032 (0.041) | 0.734 (0.735) | 0.064 (0.072) | 0.702 | 49.9 |
| **Speed $>$ 1 FPS** | | | | | | | | |
| Fast-AT [12] | ResNet101 | 9 FPS | - | - | - | - | 0.680 | 55.0 |
| AdaConv [21] | VGG19 | 12 FPS | - | - | - | - | **0.770** | 51.8 |
| A2RL [25] | Alexnet | 4 FPS | 0.818 (0.820) | 0.041 (0.045) | 0.695 (0.663) | 0.073 (0.089) | 0.630 | 66.9 |
| DIC (Conf.) [39] | VGG16 | 5 FPS | - (0.810) | - (0.057) | - | - | - | - |
| DIC [40] | VGG16 | 5 FPS | - (0.830) | - (0.052) | - (0.650) | - (0.080) | - | - |
| VPN [41] | VGG16 | 75 FPS | 0.837 (0.835) | 0.034 (0.044) | 0.716 (0.711) | 0.068 (0.073) | 0.708 | 48.3 |
| GAIC [44] | VGG16 | 125 FPS | 0.826 (-) | 0.033 (-) | 0.673 (0.673) | 0.064 (-) | 0.585 | 66.2 |
| Mars (Ours) | VGG16 | 103 FPS | 0.858 | 0.031 | **0.736** | **0.062** | 0.710 | **47.7** |
| Mars (Ours) | MobileNetV2 | 108 FPS | **0.868** | **0.029** | 0.735 | **0.062** | 0.710 | **47.7** |

mance (from $1/16 \times 1/16$ to $1/4 \times 1/4$), but the gain is marginal or even poor when the output resolution is high enough. As such, the output resolution ($H_{out} \times W_{out}$) is set to $H_{in}/4 \times W_{in}/4$ in the following parts. Similar observations are also obtained for the channel dimension of the output feature map ($C_{out}$). The reason may be that more parameters make the model easier to be overfitting. As such, $C_{out}$ is set to 96 for the output feature map.

### 4.3. Quantitative Evaluation

After determining the model through the ablation study, we retrain the model using all the training data and compare it to other state-of-the-art methods.

**Cropping Accuracy.** We show the comparison results on the three datasets (HCDB [13], FCDB [6], and FAT [12]) in Table 7. Since the proposed method is designed for the aspect ratio specified image cropping, we use the aspect ratio of the user-annotated window as the required aspect ratio when evaluating on HCDB and FCDB. As the original results of compared methods do not use the aspect ratio information on these two datasets, we modify these methods to meet the aspect ratio requirements for the fair comparison. For sliding-window (grid anchor) based methods (*i.e.*, VFN, VEN, and GAIC), we only generate sliding windows with the required aspect ratio as candidates, and the number of candidates (1140) is higher than that of the original methods (*e.g.*, 895 for VEN). Since A2RL generates the cropping result directly and VPN predicts scores for its predefined cropping boxes, we shrink their results to meet the required aspect ratios following [41]. Table 7 shows the results of these modified methods accompanying with the results from their original papers (without considering the aspect ratio). Since the source code for DIC (Conf) and DIC is not available, we only show the results of the original paper for reference. Most modified results are better than the original results, and others achieve similar results. Overall, the proposed model can achieve better performance than these state-of-the-art cropping methods under different evaluation metrics on these two datasets. The most competitive one among the above methods is VEN, but the proposed method can run 540 times faster than VEN (108 FPS vs. 0.2 FPS) and also achieves better accuracy than it.

The results on the test set of FAT are also shown in Table 7. Since this dataset itself has the aspect ratio requirement, we directly compare the proposed method with existing methods without modification. The proposed method also achieves better results than most compared methods.

**Running Speed.** A practical cropping model should run at a fast speed due to its application scenarios. As such, we also show the running speed of the proposed method and other competing methods in Table 7, where the speed is compared in terms of frame-per-second (FPS). For our model, both the inference time and post-processing time are included in computing the speed, which is evaluated on a single GPU. Since the proposed method directly predicts the cropping area (without using the sliding window or predefined anchors) and adopts the lightweight architecture, it runs in real-time at a speed of 108 FPS, which is faster than most existing methods.

### 4.4. Qualitative Results

In this section, we show both the predicted cropping areas (intermediate results) and final cropping results in Figure 5. The intermediate results show that the prediction for an image varies with the required aspect ratio, indicating the capabilities of the proposed meta-learning method. After the post-processing, the obtained cropping results can represent the original images effectively, while satisfying the aspect ratio requirement. We also compare the qualitative results of the proposed method with that of other state-of-the-art methods in Figure 6, which shows that the proposed

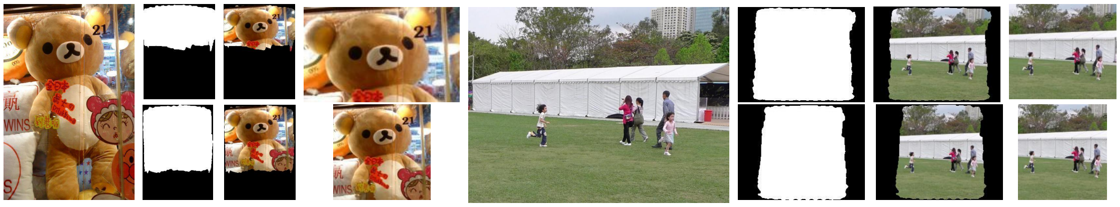
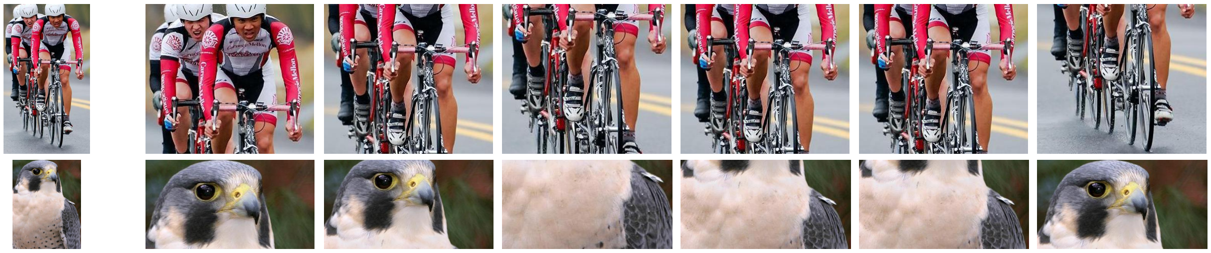


Figure 5. **Qualitative results of the proposed method.** In each group of images, the left one is the original image, the second column shows the predicted maps ($H_{out} \times W_{out} \times 1$) for different aspect ratio requirements, the third column shows the images masked by the predicted maps, and the fourth column shows the results satisfying the aspect ratio requirements after the post-processing. More results are available in the supplementary materials.



(a) Input (b) Ours (c) GAIC [44] (d) VEN [41] (e) VPN [41] (f) A2RL [25] (g) VFN [7]

Figure 6. **Qualitative comparison.** Compared to other state-of-the-art methods, the proposed method can find better-composed cropping windows that satisfy the aspect ratio requirements. More results are available in the supplementary materials.

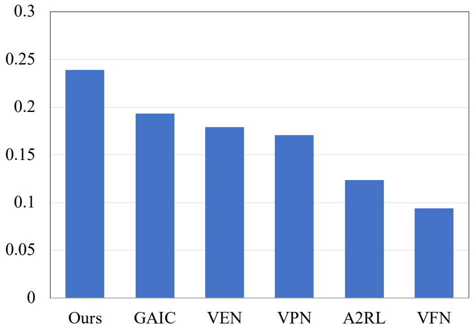


Figure 7. **User study results.** We show the proportion of cropping results selected by the photographers for the proposed method and other compared methods, including the GAIC [44], VEN [41], VPN [41], A2RL [25], and VFN [7] methods.

method can obtain better visual results of the target aspect ratio than other methods. More qualitative results are available in the supplementary material.

### 4.5. User Study

Due to the subjective nature of evaluating the image cropping results, we also perform a user study to further compare the proposed method with other state-of-the-art methods. We randomly select 300 images from the above three datasets, *i.e.*, the HCDB [13], FCDB [6], and FAT [12] datasets, 100 from each dataset. For each image, we generate the aspect ratio specified cropping results using the settings in Section 4.3 for the proposed method and the compared methods, and ask 5 experts to select the best one from the cropping results. We show the user study results in Figure 7. An interesting observation is that, although the GAIC [44] model obtains the worse IoU than VEN [41] and VPN [41] in Table 7, it can achieve better results in the user study. The reason may be that the cropping results generated by the GAIC [44] do not have the large overlaps with the ground truth, but these cropping results are still well-composed. Nonetheless, the proposed method still gets better user study results than other compared methods.

## 5. Conclusion

In this paper, we propose a meta-learning based method for the aspect-ratio-specified image cropping, which can generate cropping results of different aspect ratios for an input image in real-time (108 FPS). A base model and two meta-learners are trained in an end-to-end manner. In the inference stage, the meta-learners can predict the parameters of the base model according to the given aspect ratio, which means the parameters of the base model is not fixed during the test. The experiment results demonstrate that the proposed method can generate cropping results effectively and efficiently, outperforming most existing methods in terms of accuracy and speed.

## Acknowledgement

This work is funded by the National Natural Science Foundation of China (Grant 61876181, Grant 61673375, and Grant 61721004) and the Projects of Chinese Academy of Sciences (Grant QYZDB-SSW-JSC006).).

## References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *TOG*, 2007.

[2] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *NeurIPS*, 2016.

[3] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *CVPR*, 2018.

[4] Jiansheng Chen, Gaocheng Bai, Shaoheng Liang, and Zhengqin Li. Automatic image cropping: A computational complexity study. In *CVPR*, 2016.

[5] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando de Freitas. Learning to learn without gradient descent by gradient descent. In *ICML*, 2017.

[6] Yi-Ling Chen, Tzu-Wei Huang, Kai-Han Chang, Yu-Chen Tsai, Hwann-Tzong Chen, and Bing-Yu Chen. Quantitative analysis of automatic image cropping algorithms: A dataset and comparative study. In *WACV*, 2017.

[7] Yi-Ling Chen, Jan Klopp, Min Sun, Shao-Yi Chien, and Kwan-Liu Ma. Learning to compose with professional photographs on the web. In *ACM MM*, 2017.

[8] Bin Cheng, Bingbing Ni, Shuicheng Yan, and Qi Tian. Learning to photograph. In *ACM MM*, 2010.

[9] Gianluigi Ciocca, Claudio Cusano, Francesca Gasparini, and Raimondo Schettini. Self-adaptive image cropping for small displays. *IEEE Transactions on Consumer Electronics*, 53(4):1622–1627, 2007.

[10] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] Seyed A Esmaeili, Bharat Singh, and Larry S Davis. Fast-at: Fast automatic thumbnail generation using deep neural networks. In *CVPR*, 2017.

[13] Chen Fang, Zhe Lin, Radomir Mech, and Xiaohui Shen. Automatic image cropping using visual composition, boundary simplicity and content preservation models. In *ACM MM*, 2014.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018.

[17] Jingwei Huang, Huarong Chen, Bin Wang, and Stephen Lin. Automatic thumbnail generation based on visual representativeness and foreground recognizability. In *ICCV*, 2015.

[18] Di Kang, Debarun Dhar, and Antoni B Chan. Crowd counting by adapting convolutional neural networks with side information. *arXiv preprint arXiv:1611.06748*, 2016.

[19] Yueying Kao, Ran He, and Kaiqi Huang. Automatic image cropping with aesthetic map and gradient energy map. In *ICASSP*, 2017.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] Perla Sai Raj Kishore, Ayan Kumar Bhunia, Shuvozit Ghose, and Partha Pratim Roy. User constrained thumbnail generation using adaptive convolutions. In *ICASSP*, 2019.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[23] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *AI review*, 2015.

[24] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[25] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang. A2-rl: Aesthetics aware reinforcement learning for image cropping. In *CVPR*, 2018.

[26] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang. Fast a3rl: Aesthetics-aware adversarial reinforcement learning for image cropping. *TIP*, 2019.

[27] Si Liu, Zhen Wei, Yao Sun, Xinyu Ou, Junyu Lin, Bin Liu, and Ming-Hsuan Yang. Composing semantic collage for image retargeting. *TIP*, 2018.

[28] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[29] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015.

[30] Luca Marchesotti, Claudio Cifarelli, and Gabriela Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *ICCV*, 2009.

[31] Masashi Nishiyama, Takahiro Okabe, Yoichi Sato, and Imari Sato. Sensation-based photo cropping. In *ACM MM*, 2009.

[32] Jaesik Park, Joon-Young Lee, Yu-Wing Tai, and In So Kweon. Modeling photo composition and its application to photo re-arrangement. In *ICIP*, 2012.

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Fred Stentiford. Attention based auto image cropping. In *ICVSW*, 2007.

[36] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W Jacobs. Automatic thumbnail cropping and its effectiveness. In *UIST*, 2003.

[37] Jin Sun and Haibin Ling. Scale and object aware image thumbnailing. *IJCV*, 2013.

[38] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[39] Wenguan Wang and Jianbing Shen. Deep cropping via attention box prediction and aesthetics assessment. In *ICCV*, 2017.

[40] Wenguan Wang, Jianbing Shen, and Haibin Ling. A deep network solution for attention and aesthetics aware photo cropping. *TPMAI*, 2018.

[41] Zijun Wei, Jianming Zhang, Xiaohui Shen, Zhe Lin, Radomír Mech, Minh Hoai, and Dimitris Samaras. Good view hunting: learning photo composition from dense view pairs. In *CVPR*, 2018.

[42] Jianzhou Yan, Stephen Lin, Sing Bing Kang, and Xiaoou Tang. Learning the change for automatic image cropping. In *CVPR*, 2013.

[43] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *NeurIPS*, 2018.

[44] Hui Zeng, Lida Li, Zisheng Cao, and Lei Zhang. Reliable and efficient image cropping: A grid anchor based approach. In *CVPR*, 2019.