Learning Sample-Specific Policies for Sequential Image Augmentation

Pu Li
pli5270@sdsu.edu
San Diego State University
San Diego, California, USA

Xiaobai Liu
xiaobai.liu@sdsu.edu
San Diego State University
San Diego, California, USA

Xiaohui Xie
xhx@ics.uci.edu
University of California, Irvine
Irvine, California, USA

ABSTRACT

This paper presents a policy-driven sequential image augmentation approach for image-related tasks. Our approach applies a sequence of image transformations (e.g., translation, rotation) over a training image, one transformation at a time, with the augmented image from the previous time step treated as the input for the next transformation. This sequential data augmentation substantially improves sample diversity, leading to improved test performance, especially for data-hungry models (e.g., deep neural networks). However, the search for the optimal transformation of each image at each time step of the sequence has high complexity due to its combination nature. To address this challenge, we formulate the search task as a sequential decision process and introduce a deep policy network that learns to produce transformations based on image content. We also develop an iterative algorithm to jointly train a classifier and the policy network in the reinforcement learning setting. The immediate reward of a potential transformation is defined to encourage transformations producing hard samples for the current classifier. At each iteration, we employ the policy network to augment the training dataset, train a classifier with the augmented data, and train the policy net with the aid of the classifier. We apply the above approach to both public image classification benchmarks and a newly collected image dataset for material recognition. Comparisons to alternative augmentation approaches show that our policy-driven approach achieves comparable or improved classification performance while using significantly fewer augmented images. The code is available at https://github.com/Paul-LiPu/rl_autoaug.

CCS CONCEPTS

• **Reinforcement learning**; • **Object recognition**; • **Machine learning**;

KEYWORDS

data augmentation, reinforcement learning, object recognition

ACM Reference Format:
Pu Li, Xiaobai Liu, and Xiaohui Xie. 2021. Learning Sample-Specific Policies for Sequential Image Augmentation. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475602

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475602

1 INTRODUCTION

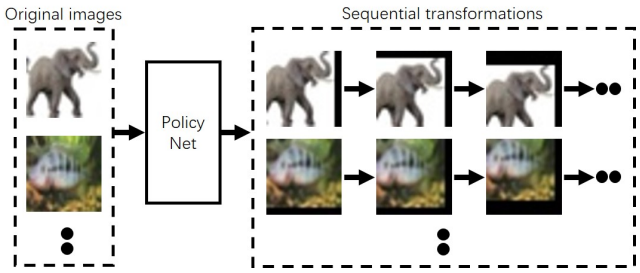

Figure 1: Policy-based sequential image augmentation. Left: original images; Right: transformed images by a trained policy net.

Image augmentation has been proved to be an effective technique for boosting supervised models in a wide variety of multimedia applications, including image classification[16, 18, 22], video classification [20], image labeling[28], image segmentation[33], object detection[27], etc. A typical augmentation method is to apply image transformations, e.g., translation, over training images and use the transformed image together with the original image labels to train image models. Advanced image transformations include Cutout[8], Cutmix[49], and GAN-based methods[1, 35]. Image augmentation can bring data variances to the existing training set, mitigate overfitting, and eventually improve model generalization capabilities without extra human annotation efforts. However, the enlarged training set may lead to high computational cost, depending on the number of augmented images added, which limits the use of this approach in large-scale multimedia applications. In this work, we propose an effective augmentation approach that can outperform existing methods while using fewer augmented images.

In the past literature, a classical augmentation approach is to manually apply multiple image transformations to each training image to obtain an enlarged training dataset. Researchers recently proposed to search for the optimal augmentation policy for a given training set. Cubuk et al.[5], for example, define a policy as a set of sub-policies, and each sub-policy includes two consecutive image transformations. A candidate policy's performance is evaluated by how this policy can boost a proxy task, e.g., classification of a hold-out dataset [25]. The optimal policy is obtained through searching the feasible hyper-parameter space, including image transformation types, operation magnitudes, and the possibility of transformations. This search-based augmentation method has been successfully applied in multiple image relevant tasks, including image

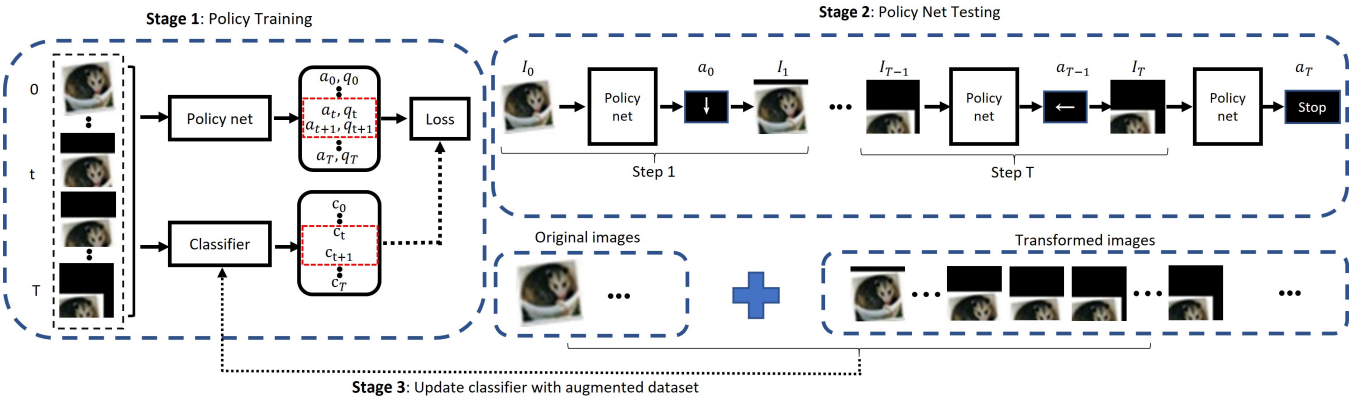

Figure 2: Sketch of the proposed policy-based data augmentation approach. There are three major stages, which are alternatively performed over iterations. a_t, q_t, r_t represents the actions, Q-value and reward respectively at step t. See texts for more details.

classification[16, 18, 22], and object detection[12, 27]. Studies over multiple datasets also suggested that one dataset's optimal policies might still be effective for a separate dataset.

The above search-based methods aim to find the optimal augmentation policy for the whole dataset and then apply it to each training image. While the policy works well for a majority of the training images, it is expected that the optimal augmentation policies for individual images might be different. Moreover, some transformations (e.g., shifting 10 pixels to the right) might lead to invalid training sample-label pairs. For example, in Fig.1, the image of an elephant remains recognizable when translated to the left. In contrast, the image of fish might lose critical information (the fish's head) if the same transformation is applied. The optimal transformation or transformation combinations are sample-specific and should be optimized separately and independently.

There are also works exploring augmentation strategies based on image content[10, 31]. [10] searches perspective transformations maximizing the classification loss, but the algorithm operates on perspective transformation matrix, and it may not easily extend to other types of transformations. [31] uses generative adversarial training to generate a sequence of transformations for each sample. However, the length of the sequence is fixed, making it impossible to apply different numbers of transformations to samples. Besides, the objective of generative adversarial training encourages the transformed image follow the distribution of original data, which excludes severe transformations and difficult samples for the classifier.

In this work, we develop a learning-based sequential approach to fully exploit the potential of data augmentation. Our approach aims to seek a sequence of image transformations for a training image. The optimal transformation at each step of the sequence, including both types and magnitudes, is determined by the visual content of the input image, and different images might end up with different image transformations. This sequential image augmentation is thus *sample-specific*, in contrast to the dataset-wide augmentation strategies. For a given image, the search space for finding the optimal sequence is of high complexity due to the combination nature. For the same reason, Cubuk et al.[5] employ only two consecutive transformations in each sub-policy. To address this challenge, we introduce a deep policy network to map the current input image to its optimal transformation or action. The transformation at a time step is selected to maximize both immediate reward and accumulated future reward. In this way, we cast the sequential image transformation task as a sequential decision process and train the policy network in the reinforcement settings.

We apply the proposed image augmentation approach for the image classification task to demonstrate its effectiveness. A joint scheme is introduced to alternatively train a classifier and the deep policy network. At each iteration, for each training image, the immediate reward of a candidate transformation is defined to encourage the generation of more difficult samples w.r.t. the current classifier. A stop condition is also introduced to prevent invalid augmented images. We empirically find that, with the proposed reward and stop condition, the policy network tends to select a few transformations from the candidate pool, and the join training scheme could largely diversify the augmented dataset.

We implement the proposed policy-driven augmentation approach for two settings: using only one type of image transformation, i.e., translation, or using a group of image transformation. Two public datasets for image classification and a newly collected dataset for material recognition are used for evaluation purposes. Experiments with comparisons to alternative augmentation methods suggested that (i) our approach with only one transformation type is more effective than other methods using a single transformation (e.g., Cutout[8]). Figure 1 showed a sequence of translated images where the translation magnitudes are determined by the policy network. (ii) While using multiple transformations, our approach achieved the state-of-the-art performance while using a much less number of augmented samples. The proposed approach can be easily extended to other image or video tasks.

The three contributions of this work are: (i) a novel policy-driven approach for sequential image augmentation; (ii) an iterative approach for jointly training the policy net and classifiers; (ii) improved performance and augmentation efficiency overexisting aug-

mentation approaches on both public image datasets and a newly collected image dataset.

2 RELATIONSHIPS TO PREVIOUS WORKS

This work is closely relevant to four streams of research in the areas of visual content analysis and machine learning.

Manual Image Augmentation As aforementioned, a classical image augmentation approach is to apply one or multiple image transformations over each training image. Common transformations [18] [14, 16, 18] include translation, reflection, scaling, cropping, and color adjustment. Each transformation also comes with different hyper-parameters, e.g., magnitudes for translation. These transformations have achieved encouraging improvement for multiple image relevant tasks, including image classification [4, 14, 16, 18, 22, 34] and object detection[12, 27]. In the recent literature, there are also multiple novel transformations proposed for various image tasks. Elastic distortion is used on digit recognition dataset MNIST[38, 45]. Patches could be randomly selected and replaced with constant value[8], random noise[52], or mixed patches[49]. In image classification tasks, images from the same class could be mixed by assigned weights[19, 40, 51]. In object detection or instance segmentation tasks, objects and their labels could be cut and pasted to other images[9, 11]. Furthermore, there are studies applying augmentation in image feature space[7, 30, 43, 44]. The above works often manually select a set of transformations empirically and apply them over all training images. The transformation parameters (e.g., magnitudes in translation) are usually fixed for all images. This manual augmentation policy might not be the optimal one in terms of boosting system performance. It is possible to employ as many transformations over a single dataset, which, however, might suffer from the exponentially grown training complexity.

Search-based Augmentation In the recent literature, there are multiple search-based augmentation approaches proposed to boost both system performance and augmentation efficiency. They aim to search for the optimal augmentation strategy for a given dataset in order to avoid manual or random augmentation strategies. In the pioneer work [5], Cubuk et al. employ a hold-out set to assist in the search for optimal transformation pairs where each pair comprises two consecutive transformations and their parameters (e.g., magnitudes). Zoph el al. [53] implement this methods to object detection task. Following AutoAug, new methodologies are proposed to reduce the policy searching complexity. Lim et al. [25] and Hataya et al. [15] remove classifier training from policy searching and directly use the performance of the classifier on transformed validation set as a reward. Ho et al. [17] and Lin et al. [26] jointly train classifier with policy searching, which leads to the non-fixed policy during classifier training. Recently, Cubuk et al. [6] propose to reduce the searching space and directly find the optimal classifier by grid search.

The above approaches could discover the optimal transformations or transformation combinations which are effective dataset-wide. These transformations might be suboptimal for individual training images. Transformations with fixed magnitudes (e.g., translation) might even lead to invalid training samples.

The proposed research is aligned with the so-called sample-specific augmentation methods [10, 31], which aim to find the optimal transformation or transformation sequence for each image. For example, Ratner et al. [31] employ the generative adversarial training framework to generate a fixed-length sequence of transformation for each image. The discriminator predicts if the transformed image becomes an outlier in the natural image distribution. Then the generator could predict a transformation sequence that does not change the information regarding to image classification. However, this method forces a fixed length of transformation sequence and may encourage generating easily recognized samples. [10] searches the augmentation strategies for each sample in a restricted perspective transformation space. While they select transformation within a trust region to avoid data corruption, the transformation which causes the highest classification loss for the image is the best for augmentation. But this method might be limited to perspective transformation.

GAN based Augmentation With the mature of GAN techniques, there are multiple efforts integrating GAN models to augment the training dataset. Previous works [1, 23, 35, 39, 41] tried to generate new images from estimated distributions, which are used to enlarge the training dataset. Generative Adversarial Networks could learn the image distribution and sample data from random latent variables[1], synthetic images[35, 39], and natural images[23]. [41] models the data distribution by a Bayesian approach and learns a generator network by a generalized Monte Carlo EM algorithm. Those sampling-based methods are affected by the quality of generated images, and there might be artifacts affecting classifier feature learning. In particular, similar to adversarial samples[46], we empirically find that GAN-generated images and natural images bear different underlying distribution, and the mixture use of both would, in general, lead to depressed classifier performance.

Reinforcement learning The proposed method is motivated by the success of deep reinforcement learning in multiple areas, including robotic control[13, 24], game agents[29, 36, 37], network architecture search[2, 54], image restoration[47], face hallucination[3], object tracking[48], image captioning[32], and image restoring [47]. In this work, we formulate the search for sequential image transformation as a sequential decision process and employ the deep policy network to parameterize the mapping between input images and the optimal transformation (and its hyper-parameters). In this way, the agent is guided by the policy network to select a transformation based on the visual content of the input image. The agent is trained in the reinforcement settings to maximize both immediate rewards and cumulative future rewards. The proposed method is the first of its kind in learning image-specific policies for sequential image augmentation.

3 OUR APPROACH

The objective of this work is to develop a policy-based augmentation approach for classification purposes. Our approach aims to jointly learn a classifier and a policy network in a reinforcement learning setting. The developed techniques can be potentially used for other image tasks, e.g., object detection, video classification, etc.

Fig.2 sketches the proposed augmentation approach which includes three major stages. In stage 1, a deep policy net is trained to produce the optimal transformation for each training image or its transformed version. A classifier is used to define the immediate reward of a candidate transformation, as introduced later, and thus regularize the training of the policy net. In stage 2, the policy net is applied over each training image to transform training images and generate an augmented dataset. In stage 3, both the original images and transformed images are used to update the classifier. We alternatively perform the above three stages to jointly train the deep policy model and classifier.

3.1 Formula: Sequential Image Augmentation

Consider image classification tasks where a set of training images are annotated for training the classifier. Traditional image augmentation methods either manually select a set of image transformations or employ various optimization methods to search for the optimal transformations or transformation combinations. In this work, we aim to apply a sequence of basic image transformations over each training image, including flipping, translating, rotating, blurring, and others as reviewed in Section 2. By denoting the sequence of transformation functions as f_1, f_2, ..., f_t, the training image I_0 could be transformed to I_t at step t as follows:

$$I_t = f_t \circ f_{t-1} \ldots \circ I_0. \tag{1}$$

where t is the length of the sequence and $\circ$ denotes the operator of transformation. Each transformation f_i is specified by its transformation type and magnitude. The transformed image at a step of the sequence is used as the input image to the next transformation.

The sequential image augmentation is characterized by two folds compared to other augmentation strategies. Firstly, it embraces the idea of composition. For example, a translation by 15 pixels is equivalent to applying three translations by 5 pixels in a row. This composition principle can largely reduce the feasible space while searching for transformation magnitudes. Secondly, a sequence of augmentation transformation might involve multiple types of transformations and thus enhance the diversity of the augmented dataset.

The proposed sequential image augmentation approach, however, brings two major challenges. Firstly, the search for the optimal sequence of transformation is of much higher complexity than finding the optimal basic transformations, mainly due to the combination nature. In particular, different image transformations at a step of the sequence will lead to different transformed images for which the optimal transformation would be different accordingly. This means that the selection of the optimal transformation at each step should be jointly solved with the selection problems for the following steps. Secondly, the sequential augmentation will have to incorporate stop conditions to prevent the generation of invalid transformed images. Taking the image of an elephant in Figure 1 for instance, the three transformed images are still valid after applying multiple translations. It will, however, become non-recognizable if only the top-left corner is kept. Therefore, it is critical to terminate the sequence of image transformation once the transformed images are not semantically consistent with the class label. In the following subsection, we will explain a deep policy-based optimization method to address the above challenges.

Action	Magnitudes	Explanation
ShearX	i, ii	(i) shear left 6%; (ii) shear right 6%
ShearY	i, ii	(i) shear up 6%; (ii) shear down 6%
TranslateX	i, ii	(i) move left 9%; (ii) move right 9%
TranslateY	i, ii	(i) move up 9%; (ii) move down 9%
Rotate	i, ii	rotate (i) counter-clockwise or (ii) clockwise 6 degree
Contrast	i, ii	contrast factor (i) 0.9; (ii) 1.1
Color	i, ii	enhance color factor (i) 0.9; (ii) 1.1
Brightness	i, ii	brightness factor (i) 0.9; (ii) 1.1
Sharpness	i, ii	sharpness factor (i) 0.9; (ii) 1.1
AutoContrast	n/a	-
Invert	n/a	-
Equalize	n/a	-
Solarize	n/a	-
Posterize	i, ii, iii	reduce color bits to (i) 5; (ii) 6; (iii) 7

Table 1: Transformation types and their magnitudes used in this work. Each magnitude defines a possible action for the agent.

3.2 Policy-based Sequential Image Augmentation

We cast the search of the optimal transformation sequence to be a sequential decision process where an agent acts to select an appropriate transformation for an input image. Each action is drawn from an action set A including all transformation types (e.g., translation or rotation) and the relevant magnitudes (e.g., displacement pixels or angels). At time step t, the agent takes state S_t as input, selects action a_t which have the largest Q value as

$$a_t = \text{argmax}_{a \in A} Q(a, S_t) \tag{2}$$

and transforms the image into I_{t+1}. The sequence of decisions made by the agent will lead to a set of augmented images, and we denote I_0 as an original image. This process could be illustrated as:

$$S_0 \xrightarrow{a_0} S_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{T-1}} S_T$$

A policy network is trained in the reinforcement settings to guide the agent's decision, which is either an image transformation or an early stopping action.

Figure 2 demonstrates how the agent acts to transform a training image. For a training image I_0, the agent first employs the policy net to select an action. Then, we apply the action over I_0 to obtain I_1, i.e., the transformed image. The agent will apply the policy net over I_1 again and repeat the above steps to obtain the sequence of transformed images: $I_2, I_3, \ldots$. In the rest of this subsection, we will provide details of the agent in terms of actions, states, rewards, stop conditions, and objective functions.

Action The action space A consists of all possible image transformations that the agent could use to augment individual image-label pairs. We also introduce a special action, i.e., stop action, which, once selected, will terminate the sequence of transformation. An image transformation might be associated with two attributes: transformation type and magnitude. Table 1 lists the types of image transformations used in this work and their magnitude definitions. Note that both image transformations and early stops will be determined by the policy net according to the visual content of the input image.

State The state S_t of the environment represents the information available to the agent at time t. In the proposed method, S_t is defined by $S_t = \{I_t, a_{t-1}\}$, where I_t is the image at t, and a_{t-1} is the previous

action before I_t. a_{t-1} will be encoded to be a one-hot vector without stopping action, and a_{-1} is a zero vector. The state could provide information on the image content and historical actions, and will be used for the agent to make decisions. For example, an agent may not choose to translate the image to the left if the main object is near the left boundary. The agent may not translate the image to the right if it just moved the image to the left in the previous step.

Reward For a training image, the reward of an action is defined to encourage the generation of difficult images that challenge the current classifier. We first train a classifier using the original training data. Then, we apply each candidate transformation over the input image and feed both the original image and transformed image to the classifier. Last, the reward of this transformation is defined as the difference between two images' classifier confidences w.r.t. the true label. Let $c_{I,k}$ denote the estimated confidence of the image I belonging to the class k. The reward of a transformation f is:

$$R = c_{I,k} - c_{f \circ I,k} \tag{3}$$

The reward would be positive if the transformed image becomes more difficult to the classifier, i.e., receiving a lower confidence level; negative if otherwise. Adding difficult samples to the training set has been approved to be an effective way to avoid over-fitting and improve classifier generalization capabilities [10].

Stop Condition The agent will choose to terminate the sequential transformation once the transformed image receives a wrong class prediction from the classifier, i.e., apply a stop action. This intuitive action is used to ensure that the transformation does not substantially change the visual content of the input image and the transformed images are still recognizable for the original image labels. When the stop condition is met, the reward is defined as:

$$R_{stop} = c_{f \circ I,k} - c_{I,k} \tag{4}$$

In contrast to Eq. (3), the reward is positive when the transformed image receives larger confidence than the input image. The benefits of additionally using this stop condition are two-fold. Firstly, without this stop condition, the agent will be purely driven by the reward function Eq. (3) and seek for difficult images and eventually lead to invalid image-label pairs. The stop condition will constrain the transformations within a "trust region" where the augmented samples are visually different from the original image but are still recognizable. Secondly, the stop condition is able to prevent the augmentation of outlier samples. For example, if an image receives a wrong label from the classifier, it will not be augmented since it leads to a stop action immediately.

Objective function We use a deep policy network to approximate the Q function and employ the concept of double Q-learning [42] to stabilize the training of the agent. In particular, during training, we maintain two policy networks: one is the online agent network, and the other one is the target network. The target network is the same as the online network, except that its parameters are copied from the online network every τ steps. The target network kept fixed for all other steps. The agent is trained to minimize the difference between target Q values (fixed network) and current estimate of Q values (online network). Like [29], we employ a temporal difference loss function. Let r_t denote the immediate reward after action a_t, $q(S_t, a_t)$ denote the estimated online Q values for the current state, $q'(S_t, a_t)$ is the target Q value by the fixed target network [42]. The loss function is defined as:

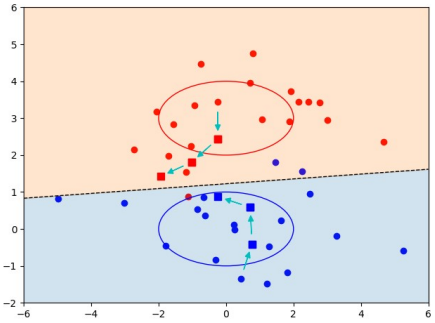

Figure 3: Empirical study over a toy dataset. The reward and stop actions would lead to difficult samples (being closer to the decision boundary). See Section 5 for more details.

$$Loss = (y_t - q(S_t, a_t))^2 \tag{5}$$

where

$$y_t = \begin{cases} r_t + \gamma * max_{a'} q'(S_{t+1}, a') & 1 \le t < T \\ r_T & t = T \end{cases} \tag{6}$$

γ is the discount factor, and T is the episode length. This loss function encourages the agent to have correct accumulative reward estimation. During training, the samples S_t, S_{t+1}, r_t are retrieved from replay memory and are used to calculate this objective function. In each training iteration, the gradients are propagated through the online network and used for agent weights update, with the target network fixed. The memory replay and double network learning can dramatically reduce the variance of deep Q learning.

3.3 Joint Training

We develop an alternative training scheme to train the classifier and policy network jointly. The scheme starts with training the classifier over the original training dataset. Then, there are three iterative steps. The first step is to train the agent with the rewards generated from the current classifiers. The second step is to employ the agent to augment the training dataset and we retrain the classifier over the augmented dataset at the last step. We repeat these three steps for multiple iterations until the classifier performance is saturated. Note that we collect all the transformed images during the iterative training to form the final augmented dataset. Our empirical studies suggest that the agent tends to select a few transformation types during a sequence of transformations, and the use of augmented data from multiple iterations will diversify the training samples for the final classifier.

4 EXPERIMENTS

We test and evaluate the proposed policy-based augmentation approach for image classification tasks.

Datasets We employ three image datasets. The first one is a newly collected image dataset for garbage classification. It includes 6234 images from 8 categories (5 recyclable categories: Cardboard, Glass, Metal, Paper, Plastic; 3 non-recyclable categories: Wet Trash, Food, Bottle). We split the dataset into three-fold: a training set (2774 images), a validation set (310 images), a testing set (3150 images).

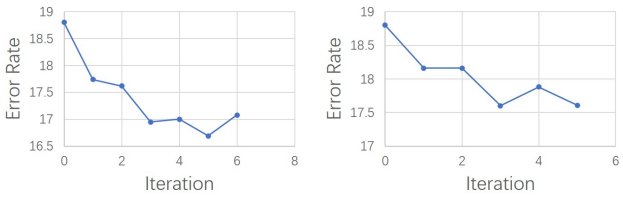

Figure 4: Classifier errors over iterations of augmentation.The proposed augmentation method uses 14 transformations (left) or translation only (right).

The training set is further divided into two parts: 1664 images for classifier training and 1110 images for policy network training. All images are resized to have a longer side of 500 pixels. As shown in Figure. 5, the testing images were collected in different settings from the training images (lighting conditions, camera angles, etc.) from the training images, which makes this dataset a challenging task for state-of-the-art classification models.

The other two datasets are the public image datasets: CIFAR-10, CIFAR-100[21]. We use the standard training/testing splits. For each dataset, we further divide the training set into two parts: 60% for classifier training and 40% for policy network training. The classifier and agent are iteratively trained, as stated in the last section. Then, the agent acts to augment each training image and the augmented dataset will be used for classifier training in the next iteration.

Classification Networks We implement and train multiple network backbones to obtain the classifier including Wide-ResNet-28-10, Wide-ResNet-40-2, Wide-ResNet-28-2, ResNet-50, and Shake-Shake(26 2x96d) [5]. The hyperparameters of these networks are set to be the same as the previous work [5]. In particular, the variants of Wide-ResNet[50] are trained with a SGD optimizer (batch size = 128, initial learning rate = 0.1, momentum = 0.9, weight decay rate = 5e-4) for 200 epochs. The learning rate is scheduled by a cosine learning decay with an annealing cycle, and the same scheduling is applied to other networks training. The Shake-Shake net is trained with an initial learning rate of 0.01, weight decay rate of 0.001 for 1800 epochs. We also follow the same normalization for input images on the CIFAR dataset as [5](mean=[0.491, 0.482, 0.447], standard deviation=[0.247, 0.243, 0.262]). The ResNet-50 is trained with a SGD optimizer (batch size = 64, initial learning rate = 0.1, momentum = 0.9, weight decay rate = 5e-4) for 270 epochs. We preprocess the images as [25] by cropping a center patch of 224-by-224 pixels and apply normalization(mean=[0.485, 0.456, 0.406], standard deviation=[0.229, 0.224, 0.225]) to the RGB channels. Other than the classifier trained on 60% of training samples, which is used for policy network reward calculation, we also train a classifier on all training samples along with the augmented samples, whose performance is reported in Section 5.

Policy Network Figure 2 illustrates the inputs and outputs of the policy network. We employ the replay memory method and double Q-learning [42] to train the agent. At each step t, the agent takes the image and previous action as input and outputs a vector for expected Q values on all possible actions. The network architecture is the same as [47]. The input image is first encoded to a 32-dimensional feature vector by the feature extractor module(four convolutional layers and one fully connected layer). Then, a one-hot vector representing the previous action is concatenated to the image feature vector. The concatenated vector is processed by the Long Short-Term Memory module, which stores the historical transformation information. Finally, another fully connected layer outputs the Q value vector. The action with the largest Q value is executed during testing.

To study the transferibility of our policy, we train the policy network with one classifier and then apply the same augmentation policy to other classifiers. In our experiments, the policy network is trained with ResNet-50 on Garbage dataset and Wide-ResNet-28-10 classifier on CIFAR datasets. During training, we have one image from our policy training set, and apply ϵ-Greedy action selection until a stop action or the maximum step. The ϵ starts with 1 for 5e3 steps then linearly decreased to 0.1 before 1e6 steps. We update the policy network every 4 steps with an Adam optimizer (batch size = 64, initial learning rate = 1e-4) for 2e6 steps. The learning rate is decayed exponentially to 2.5e-5 during training. The target network is updated with the weights from the latest agent network every $\tau = 1e4$ steps. This sequence of data (state, action, reward) for one image is considered one episode and added to the replay memory. We set the maximum number of steps of one episode to be 10 empirically. 64 episodes are randomly sampled from the replay memory for gradient calculation in each training iteration. The replay memory size is 500,000 for CIFAR10 and CIFAR100 and 15,000 for Garbage respectively.

5 RESULTS

Study on Toy Data We first investigate how the proposed stop action and reward affect the augmentation using a toy data. This toy dataset includes a set of 40 two-dimensional data points. These 20 positive and 20 negative points are generated from two Gaussian distributions (illustrated by the oval whose radius is the sigma of Gaussian at radius direction) respectively. We train a logistic regression model as the classifier. To augment a data point, we randomly draw multiple nearby data points and select the one that receives the largest reward as defined in Eq. (3). This augmentation strategy will choose the most difficult data points from the neighborhood of a given data point. We repeat the above step until the newly augmented point receives a wrong prediction from the classifier. Figure 3 plots the scatter of the data points and the sequence of transformations for a negative and a positive data point (square points with arrows between them). The augmented data points in the sequence are getting closer to the decision boundary. In this way, the reward and the stop action jointly function to generate valid yet difficult samples.

Dataset	Model	Baseline	Cutout [8]	Ours(Trans)
CIFAR-100	Wide-ResNet-40-2	26.00	25.2	**22.58**
	Wide-ResNet-28-10	18.80	18.4	**17.60**

Table 2: Classification error rate (%) on testing set of CIFAR-100. The proposed method only used the translation actions. See texts for more details.

Results on the Garbage Dataset Table 3 reports the classification accuracies of different approaches on the garbage dataset. We

Figure 5: Sample images from the Garbage dataset. Two images of metal from the training (left) and testing set (right) have different appearance due to varying lighting conditions.

train three classifier networks with the original training set (baseline) or augmented dataset, and compare the result with Cutout [8]. For fair comparisons, we only use image translation as possible actions for agent. As shown in table 1, it involves two transformations (TransX and TransY) with two magnitude or four atomic actions. The settings for Cutout are the same as the original paper [8]. Results showed that our learned augmentation policy could clearly outperform the Cutout method while using different network architectures. This dataset is challenging because of the different settings between the train set and test set. In Figure. 5, the Cutout method failed to work on this testing image (right) while the proposed method made a correct prediction. It is noteworthy that Cutout did not outperform the baseline while using Wide-ResNet-28-2. In contrast, our translation-only augmentation approach consistently outperforms the baseline.

Model	Baseline	Cutout[8]	Ours(Trans)
Wide-ResNet-28-2	22.06	21.27	**22.25**
Wide-ResNet-40-2	22.31	23.49	**25.46**
ResNet-50	27.37	29.46	**33.08**

Table 3: Testing Classification Accuracy (%) on the Garbage dataset.

Iteration	1	2	3	4	5
Average	1.89	1.91	1.78	2.23	2.54
Std	1.35	1.45	1.22	1.49	1.68

Table 4: Average number and standard deviation of transformation types used in each transformation sequence on CIFAR-100 across all joint training iterations.

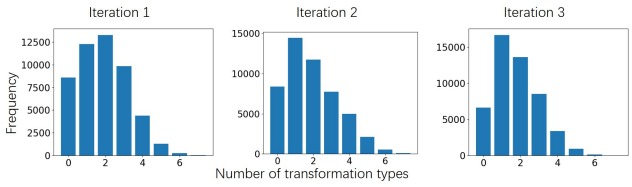

Figure 6: Frequency of transformation types in the augmentation sequences for CIFAR-100.

Results on CIFAR10 and CIFAR100 We first study how the classifier's performance changes over iterations in the proposed joint training scheme. Figure 4 plots the performance of Wide-ResNet-28-10 over iterations on CIFAR100. When we uses 14 kinds of transformations, we find the performance becomes stable around the third iteration, and the classifier reaches the best performance at iteration 5. Similarly, while only using translation for augmentation, the classifier achieves the lowest test error at iteration 3. This empirical study showed that the joint training scheme could largely improve system performance.

Table.2 reports the classifier performance (error rate) over the challenging CIFAR-100 while using Cutout [8] or the proposed policy-based augmentation method with only translations as the actions. We observe that our policy significantly outperforms Cutout, especially for Wide-ResNet-40-2 (error rate decreased by 3.4 with our policy, error rate decreased by 0.8 with Cutout). This result shows the effectiveness of learned translation compared to random patch erasing in Cutout.

Table.5 includes the results of state-of-the-art data augmentation methods. We implement the proposed policy-based method with all the 14 transformations. We also report for each augmentation approach (last row) the relative size of the augmented data comparing to the original training data. From the table, we observed similar or higher performance compared to previous methods on both datasets while using a much less number of augmented samples. For example, the AA method [5] employs 25 augmentation sub-policy, which results in 75 different transformed images for each original image. The proposed method augments each image less than 50 times on CIFAR100 and less than 30 times on CIFAR-10. The above comparisons indicate the efficiency of sample-specific augmentation policy over dataset-wide augmentation policy.

We visualize several images and their sequential transformations and report the overall statistics of the selected actions on CIFAR-100. In this set of qualitative experiments, we employ the proposed policy-based method with translation only since this transformation can be better visualized than others. Fig.7 plots multiple images (column 1) and the sequence of transformed images. We can observe that the trained policy can generate well-diverse yet valid augmented images which are still recognizable after transformations. This well demonstrates the effectiveness of the proposed reward and stop condition. Fig.8 plots the transformed versions of an image over iterations.We can observe that the agent tends to translate the image to different directions across iterations, resulting in localized images highlighting different parts of the object. The iterative joint training scheme dramatically increases the diversity of augmented images. Note that, at iteration 4, the agent tends to stop further augmentations and output very similar augmented images over the sequence. This is because the current classifier, after multiple iterations, can well recognize most transformed images and is less motivated to transform the original image.

We calculate the number of transformation types that each sequence might involve while using 14 transformations over CIFAR-100. Figure 6 shows the frequency of transformation types in the transformation sequences. We can observe that the agent used less than 3 transformation types for most transformation sequences. Table 4 also lists the average number of transformation types across iterations. On average, the first iteration uses 1.89 transformation types and the fifth iteration uses 2.54 transformation type. The above statistics shows that the iterative joint training scheme can

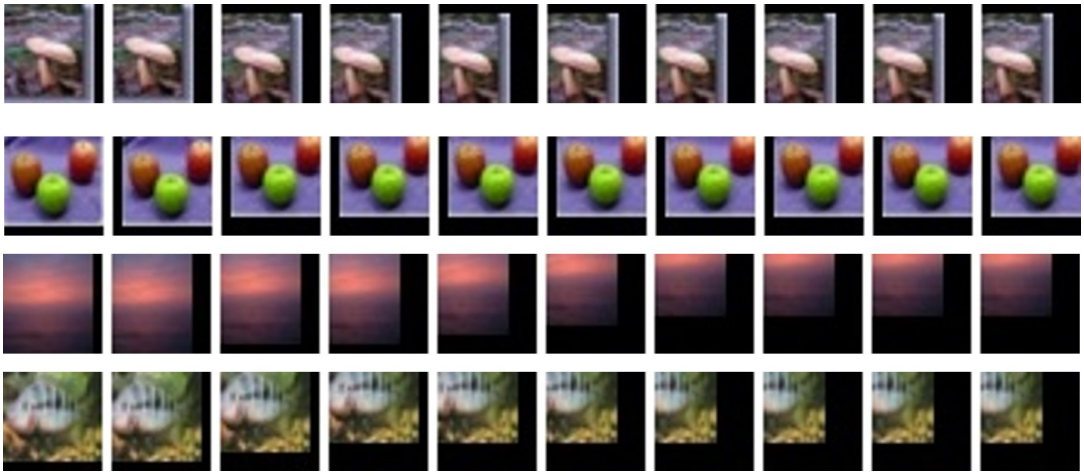

Figure 7: Examples of images transformed by our method. Two transformations: TransX and TransY are used by the agent.

Figure 8: Examples of images generated at different iterations. Two transformations: TransX and TransY are used by the agent.

Dataset	Model	Baseline	AA[5]	PBA[17]	Fast AA[25]	Faster AA[15]	RA[6]	Ours
CIFAR-10	Wide-ResNet-28-10	3.90	2.60	2.60	2.70	2.60	2.70	2.70
	Shake-Shake (26 2x96d)	2.90	2.00	2.00	2.00	2.00	2.00	2.06
CIFAR-100	Wide-ResNet-40-2	26.00	20.70	-	20.70	21.40	-	20.38
	Wide-ResNet-28-10	18.80	17.10	16.70	17.30	17.30	16.70	16.69
Augmentation Size		–	75×	200×	75 ×	75×	196×	<50×

Table 5: Classification error rate (%) on testing sets of CIFAR10, CIFAR-100. The last row lists the number of transformed images for each training image.

actually diversify the ways to augment images and thus improve the quality of the augmented data.

6 CONCLUSION

We developed a policy-based sequential image augmentation approach to augment a training image according to its visual content. We cast the search for sample-specific transformation sequences as a sequential decision process and introduce a method to jointly train the classifier and agent in the reinforcement learning settings. Experiments on both newly collected image dataset and public datasets show that our method achieves similar or better performance than previous augmentation methods while using significantly less transformed images. Notably, even with only translations, our method significantly outperforms the popular Cutout method. Our augmentation approach can be readily extended to solve other multimedia content understanding tasks, e.g., object detection, video classification, and video parsing, etc.

ACKNOWLEDGMENTS

Dr. Liu and Dr. Xie are supported by the US Office of Naval Research (grant no. N00014-17-1-2867) and National Science Foundation (IIS-1715017), respectively.

REFERENCES

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340* (2017).

[2] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. 2016. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167* (2016).

[3] Qingxing Cao, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. 2017. Attention-aware face hallucination via deep reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 690–698.

[4] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3642–3649.

[5] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018).

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.

[7] Terrance DeVries and Graham W Taylor. 2017. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538* (2017).

[8] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).

[9] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 1301–1310.

[10] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. 2016. Adaptive data augmentation for image classification. In *2016 IEEE international conference on image processing (ICIP)*. Ieee, 3688–3692.

[11] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. 2020. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. *arXiv preprint arXiv:2012.07177* (2020).

[12] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. 2018. Detectron.

[13] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3389–3396.

[14] Dongyoon Han, Jiwhan Kim, and Junmo Kim. 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5927–5935.

[15] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. 2020. Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*. Springer, 1–16.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. 2019. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*. PMLR, 2731–2741.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[19] Hiroshi Inoue. 2018. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929* (2018).

[20] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[23] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. 2017. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access* 5 (2017), 5858–5869.

[24] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[25] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. 2019. Fast autoaugment. *arXiv preprint arXiv:1905.00397* (2019).

[26] Chen Lin, Minghao Guo, Chuming Li, Xin Yuan, Wei Wu, Junjie Yan, Dahua Lin, and Wanli Ouyang. 2019. Online hyper-parameter learning for auto-augmentation strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6579–6588.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[28] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. 2017. High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 55, 12 (2017), 7092–7103.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[30] Oren Nuriel, Sagie Benaim, and Lior Wolf. 2020. Permuted AdaIN: Enhancing the Representation of Local Cues in Image Classifiers. *arXiv preprint arXiv:2010.05785* (2020).

[31] Alexander J Ratner, Henry R Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. 2017. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems* 30 (2017), 3239.

[32] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 290–298.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[34] Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. 2015. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229* (2015).

[35] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2107–2116.

[36] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[37] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[38] Patrice Y Simard, David Steinkraus, John C Platt, et al. 2003. Best practices for convolutional neural networks applied to visual document analysis.. In *Icdar*, Vol. 3. Citeseer.

[39] Leon Sixt, Benjamin Wild, and Tim Landgraf. 2018. Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI* 5 (2018), 66.

[40] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5486–5494.

[41] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. 2017. A bayesian data augmentation approach for learning deep models. *arXiv preprint arXiv:1710.10564* (2017).

[42] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[43] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. 2020. Regularizing deep networks with semantic data augmentation. *arXiv preprint arXiv:2007.10538* (2020).

[44] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. 2019. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems* 32 (2019), 12635–12644.

[45] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. 2016. Understanding data augmentation for classification: when to warp?. In *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 1–6.

[46] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. 2020. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 819–828.

[47] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. 2018. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2443–2452.

[48] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. 2017. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2711–2720.

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6023–6032.

[50] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[52] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13001–13008.

[53] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. 2020. Learning data augmentation strategies for object detection. In *European Conference on Computer Vision*. Springer, 566–583.

[54] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).