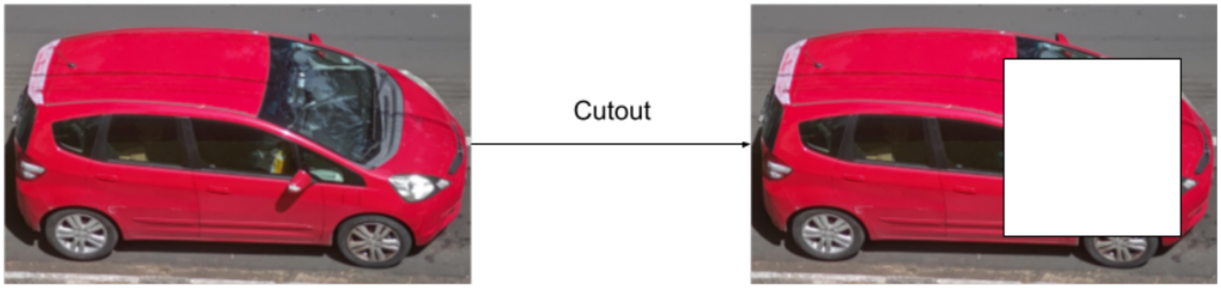
Avoid Overfitting: A Survey on Regularization Methods for CNN

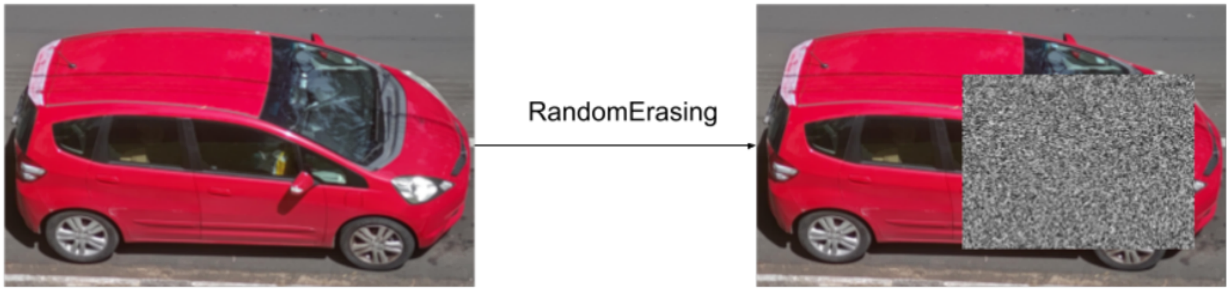
CNNs are usually used for computer vision tasks, such as image classification and object detection, to create models as powerful as human vision. If the amount of information available is considered, it becomes clear the training task requires more data variability than possible. Considering a healthy human with a regular brain and eyes, we retain new information around 16 hours per day, on average, disregarding the time we sleep. Even considering huge datasets such as ImageNet, the number of images available is minimal compared to the quantity of data a human brain receives through the eyes. This unavailability of new data may lead to a situation known as overfitting, where the model learns how to represent the training data, but it does not perform well on new information, i.e., the test data.

Regularization Based On Data Augmentation

Cutout : One straightforward but powerful technique to perform data augmentation is the well known Cutout . During training, it randomly removes regions of the image before feeding the neural network. The ideal size of removed region varies according to the number of instances per class and the number of classes for a given dataset.



Random Erasing :
RandomErasing was further developed based on the Cutout technique. It removes random crops of the image and randomly adds information on the blank space, such as noise. Unlike Cutout, RadomErasing does not remove pieces of the image every time.



AutoAugment :
AutoAugment tries to find out what transformations over a given data set would increase the accuracy of a model. It creates a search space for a given policy using five different transformations ruled by two additional parameters: the probability of applying a given alteration and the magnitude of this change.

One huge advantage of this approach is the transferability of these policies across different datasets. One disadvantage of this approach is the time used to train the controller model: for the ImageNet dataset, for instance, it took around 15, 000 hours of processing, which may be impracticable in several cases. Fast AutoAugment aimed at overcoming such a bottleneck with a new algorithm, reducing the time related to the search procedure significantly, besides producing similar results.

PBA:

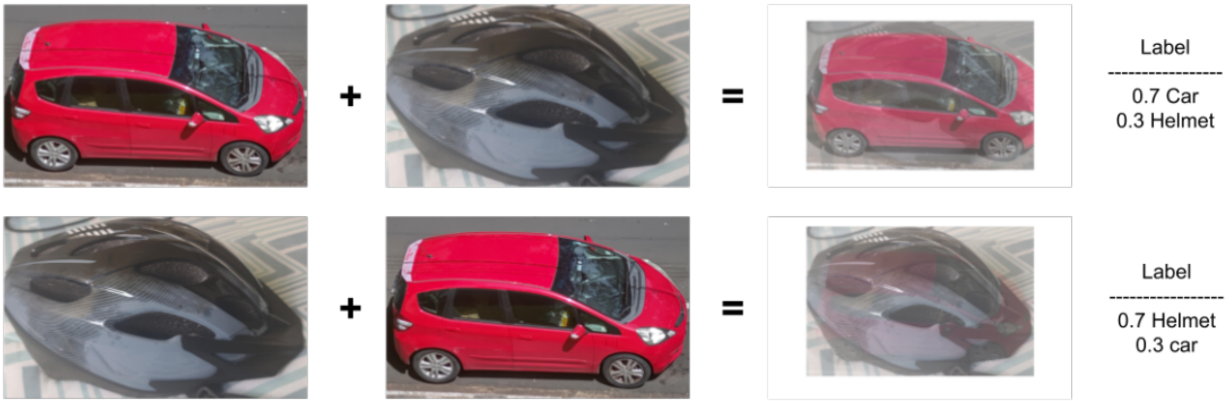
Population Based Augmentation (PBA) not only showed a novel augmentation algorithm but demonstrated schedule policies instead of fixed policies, which improves the results from the previous studies. At every 3 steps, it changes half the policies, being 1/4 changes in the weights and the other 1/4 a change in the hyperparameter.

RandAugment

AutoAugment and PBA suffer from a bottleneck, where the methods for finding the best data augmentation involves their computational burden since it may take longer time than the own NN training. Another problem is related to the strategies found during the search, which may end up in a sub-optimal strategy, i.e., it does improve the results locally; however, it does not lead to the best global results. RandAugment [6] uses the 14 most common policies found on previous works and performs the search of the magnitude of each policy during training, thus removing the need for a preliminary exploration step and tailoring the data amplification to the current training CNN.

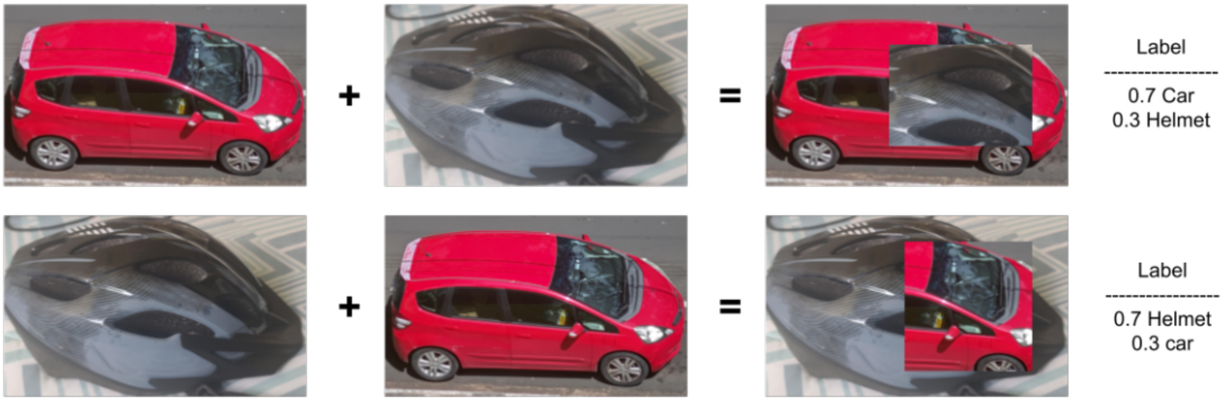
Mixup

Mixup is another possible way of training CNN where it mixes two images from the training dataset and forces the model to determine which class this mixture belongs to reliably. Providing This new input/output training pair allows the model to learn more features from corrupted inputs.approach can improve results not only in the image classification task but in speech recognition, stabilization in generative adversarial networks, tabular datasets, and other problems.



CutMix

CutMix replaces entire regions from a given input and changes the label by giving the same weights as the area used by each class. For example, if a cat's image is replaced in 30% by an image of an airplane, the label is set to be 70% cat and 30% airplane. This strategy shows a significant improvement in results.



CutBlur

For super-resolution (SR) tasks, the literature lacks works that proposed regularization techniques to handle the problem explicitly. Even though the aforementioned techniques can be used and possibly improve results, they are not natively designed to cope with the SR problem. The only approach found, so far, is the CutBlur, which works by replacing a given area on the high-resolution image with a low resolution version from a similar region.

BatchAugment

One important hyperparameter for training CNNs concerns the mini-batch size, which is used to calculate the gradient employed in the backpropagation. Instead of just fulfilling the entire memory with different instances from the dataset, batchAugment considers half of the memory limit using the default set up for data augmentation and then duplicates all instances with different data augmentation possibilities. Although it sounds like a straightforward approach, it has a significant improvement on the final results.

Bag-of-Tricks

The Bag of Tricks research performs investigation on how 2 regularizers can influence each other by combining several known regularization methods, such as Mixup, Label Smoothing and Knowledge Destilation. The ablation study shows that if some cleverness is applied, the final result can be significantly improved. For instance, a MobileNet using this combination of methods improved its results by almost 1.5% in the ImageNet dataset, which is a significant gain.

Regularization Based On Internal Structure Changes:

Internal regularizers are the ones that change the weights or kernel values during training without any explicit change on the input.

Dropout and variants

Dropout was proposed as a simple but powerful regularizer that aims to remove some neurons, thus forcing the entire system to learn more features. It can be applied not only on CNNs but in Multilayer Perceptrons (MLPs) and Restricted Boltzmann Machines (RBMs). The probability of dropping out each neuron is estimated through Bernoulli's distribution at each step of the training phase, thus adding some randomness in the process.

MaxDropout

While Dropout randomly removes the neurons in the training phase, Maxdropout deactivates the neurons based on their activations. It first normalizes the tensor's values and then sets to 0 every single output greater than a given threshold p, so the higher this value, the more likely it to be deactivated.

DropBlock

DropBlock shows that removing entire areas of a given tensor (i.e., feature map) can help the model to generalize better. By using ResNet-50 and AmoebaNet-B models on the image classification task, RetinaNet on object detection, and ResNet-101 for image segmentation, it shows that it can improve results better than Dropout and other internal regularizers. DropBlock is applied on every feature map of the CNN, starting the training with a small ratio and slowly increasing its value.

TargetDrop

TargetDrop combines this mechanism with DropBlock. During training, it allows the entire system to remove most discriminative areas on a given channel. Results show this method not only accomplishes better results than DropBlock.

AutoDrop

AutoDrop forces the CNN to learn the best drop design according to information from training by using a controller that learns, layer by layer, the best drop pattern.

Shake-Shake

One way to force regularization on these architectures is to give different weights to each branch of the residual connections during training. The original ResNets works by adding the weights on each branch without any differentiation. During training, Shake-shake works on 3-branch ResNets by changing the multiplication factor of each branch on the forward pass and multiplying by a different value on the backward pass, thus changing how each branch affects the final result. Besides the improvement, this method only works on 3-branches ResNet, making it hard to compare other methods directly

Manifold Mixup

The Manifold Mixup acts like the Mixup, however, operating in any internal layer of a CNN, and not only in the input layer. A deep neural network can be considered a set of smaller neural networks. Each one outputs some desired features; therefore, if all subnets work well, the final result can be regarded as a good one.

Label Regularization:

Label smoothing proposes a regularization technique in the label encoding process by changing the value on each position of the hone-hot representation. Label smoothing works by preventing two main problems. First, the well-known overfitting, i.e., the situation where the model learns the information about the training set but cannot generalize the classification in the test set. The second and less obvious is overconfidence.

Two Stage Label Smoothing

ne difficulty of using label smoothing is to find out what value of ϵ (i.e., smoothing factor) is the ideal, either for a general or for a specific data set. The original work suggests that $\epsilon = 0.1$ is the excellent condition; however, the Two-Stage Label Smoothing (TSLA) suggests that, in general, the gradient descent combined with the label smoothing technique can only improve the results until a certain point of training, after that it is better to set all values to 0 and 1 for the active class.

Structural Label Smoothing

Usually, it is not straightforward to define appropriate values for the label smoothness factor. Structural Label Smoothing (SLS) proposes to compute such a value by estimating the Bayes Estimation Error, which, according to authors, helps define the label's boundaries for each instance.

Datasets used to come up with comparison : CIFAR, Imagenet, and SVHN

Architectures used : ResNet, Wide Residual Network, ResNeXT, PyramidNet