

AutomatedTesting2020

姓名：奚志恒

学号：181250162

选题方向：AI测试大作业

demo视频：由于有点大，所以放在NJUBOX链接里面，下面两个都可以

- <https://box.nju.edu.cn/f/040984c056d54692aa05/>
- <https://box.nju.edu.cn/f/040984c056d54692aa05/?dl=1>

目录

AutomatedTesting2020

目录

数据增广

代码框架与思路

单独使用一种增广策略

裁剪Cropping

平移Shifting

旋转Rotating

翻转Flipping

色彩空间Color Space

噪点插入Noise Injection

多种数据增广策略组合使用

裁剪+旋转+明亮度

平移+噪点插入

参考文献

数据评估

代码框架与思路

Cifar-10扩增数据评估

图表展示

每个模型对通过某种方法扩增的数据进行预测的准确率

模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率

精度丢失率

分析

Cifar-100扩增数据评估

图表展示

每个模型对通过某种方法扩增的数据进行预测的准确率

模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率

精度丢失率

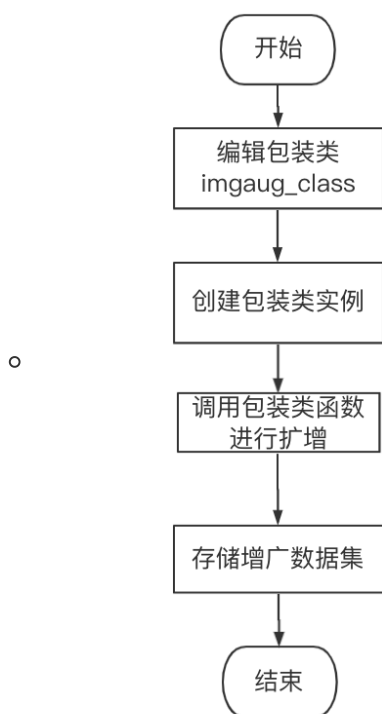
分析

参考文献

数据增广

代码框架与思路

- 本次作业的数据增广代码采用python编写，基于imgaug的库来进行扩增。
- 选用cifar-10和cifar-100数据集，这两种数据集比较类似，所以我们使用的数据增广方法大多类似，只会在细节上有所区别——因为cifar-100有100个类，对抗动因素的敏感度较高，容易丢失原标签，所以我们在扩增cifar-100时设置稍微保守一点的参数。
- 代码大体思路如下：
 1. 我们首先写一个包装类来包装一个数据集对应的扩增类，比如类 `imgaug_cifar10`，类的 `__init__` 函数接受数据集以及 `batch_size` 作为参数。
 2. 接着我们在类中使用函数的形式实现我们需要使用的所有增广策略，比如 `crop`、`shift`，这些策略基于imgaug的接口实现。函数的返回值是增广后的数据集。
 3. 然后我们就可以创建类的实例，调用包装函数进行数据增广。
 4. 最后我们将增广后的数据（`x`和`y`，图像和标签）分别存储成 `.npy` 文件，其中图像文件文件名格式为 `aug_imgs_序号_增广策略_x.npy`，比如 `crop` 变换之后的图像文件为 `aug_imgs_1_crop_x.npy`，标签文件文件名格式为 `aug_imgs_序号_增广策略_y.npy`，比如 `crop` 变换后标签文件为 `aug_imgs_1_crop_y.npy`。这些文件可以用来做后续数据评估。
- 相应流程图如下：



- 目前已经可以查到有许多种数据增广策略，比如裁剪、平移、旋转等，本次作业中不仅仅输出单独使用一种增广策略得到的增广数据集，还输出使用多种增广策略的数据集。
- 代码中 `imgaug_cifar10` 类图如下

```

c Project.data_augmentation.imgaug_class_cifar10.imgaug_cifar10
m __init__(self, x_train, y_train, batch_size)
m crop(self, batch_idx)
m shift(self, batch_idx)
m rotate(self, batch_idx)
o m fliplr(self, batch_idx)
m flipud(self, batch_idx)
m brightness(self, batch_idx)
m contrast(self, batch_idx)
m additive_Gaussian_noise(self, batch_idx)
m crop_rotate_brightness(self, batch_idx)
m shift_noise(self, batch_idx)

```

- 代码中imgaug_cifar100类图如下

```

c Project.data_augmentation.imgaug_class_cifar100.imgaug_cifar100
m __init__(self, x_train, y_train, batch_size)
m crop(self, batch_idx)
m shift(self, batch_idx)
m rotate(self, batch_idx)
o m fliplr(self, batch_idx)
m flipud(self, batch_idx)
m brightness(self, batch_idx)
m contrast(self, batch_idx)
m additive_Gaussian_noise(self, batch_idx)
m crop_rotate_brightness(self, batch_idx)
m shift_noise(self, batch_idx)

```

- 本次作业中，对于每个数据集的每种方法，不会提交所有数据的增广结果，只会生成大小为10000的数据集。但是在上传时文件过大，不方便上传，所以我们实验时数据集为10000，上传时数据集为100。

单独使用一种增广策略

裁剪Cropping

- 裁剪技术意思是，裁剪出每个图像的中央色块，去掉一些没有必要的色块。裁剪图像可以用作高度和宽度尺寸混合的图像数据的实际处理步骤。另外，随机裁切还可以用于提供与平移非常相似的效果。随机裁剪和平移之间的对比在于，裁剪会减小输入的大小，例如 (256, 256) → (224, 224)，而平移会保留图像的空间尺寸。裁剪时要注意裁剪的比例，因为过大的裁剪可能使得转换后的图片难以保留标签。
- 本次作业中，cifar-10采用的裁剪范围是 (0, 0.25)，cifar-100采用的裁剪范围是 (0, 0.1) 以保证数据增广后保留原来的标签。

平移Shifting

- 向左，向右，向上或向下移动图像对于避免数据中的位置偏差可能是非常有用的转换，这种转换有利于之后模型训练的准确性和鲁棒性。例如，如果数据集中的所有图像都居中（这在人脸识别数据集中很常见），那么这也需要在完美居中（perfectly centered）的图像上测试模型，

否则难以得到准确地结论。当原始图像沿某个方向平移时，剩余空间可以用恒定值（例如0或255）填充，也可以用随机或高斯噪声填充。该填充保留了增影后图像的空间尺寸。

- 本次作业中，扩增cifar-10时，在x方向和y方向采用的平移范围是 $(-0.20, 0.20)$ ；扩增cifar-100时，在x方向和y方向采用的平移范围是 $(-0.10, 0.10)$ 。这使得图像主要内容不会被移出。

旋转Rotating

- 旋转图像增强是通过指定旋转角度后左右旋转图像来完成的。旋转增强的可行性在很大程度上取决于旋转度的参数。轻微旋转可能对数字识别任务（例如MNIST）有用，但是随着旋转度的增加，转换后的数据标签将不再保留。而对于类似cifar-10、cifar-100的数据集图像来说，旋转角度增加一些是可行的。
- 本次作业中，cifar-10扩增时采用旋转角度是 $(-30, 30)$ ，cifar-100扩增时采用旋转角度是 $(-10, 10)$ 。

翻转Flipping

- 翻转包括水平方向的翻转和垂直方向的翻转。在实际操作中，水平翻转比垂直翻转更加普遍。这种方法已被证明对cifar-10等数据集很有用，但是当涉及到文字（如MNIST）时，它会导致错误，标签无法保留。
- 本次作业中，选择了cifar-10和cifar-100，认为翻转都是可行的。水平翻转和垂直翻转我们都设置0.5的概率，即有0.5的概率可能被水平翻转或垂直翻转。

色彩空间Color Space

- 图像数据被编码为3个堆叠的矩阵，每个矩阵的大小为（高×宽）。这些矩阵代表单个RGB颜色值的像素值。我们可以通过简单的矩阵运算轻松地控制RGB值，以增加或减少图像的亮度（brightness）；也可以使用类似的手段来改变图像的对比度（contrast）。但是色彩变换必须在一定范围内才能保证图片标签能够被保留，所以在使用时要注意参数设置。
- 本次作业中，两个数据集采用的亮度乘数范围都是 $(0.8, 1.2)$ ，采用的对比度乘数范围都是 $(0.75, 1.25)$ 。

噪点插入Noise Injection

- 噪点插入是指在图像中或多或少得插入一些无关的色点，让图像变得与原来不完全一样。这种方式可以使得模型训练得更强、鲁棒性更高。但是插入噪点必须在合理范围内，如果插入噪点过多，会导致图像失真的情况出现。
- 本次作业中，采用添加高斯噪音的方式来插入噪点。在做cifar-10的噪点插入时，产生噪声的正态分布的标准偏差参数取 $(0.0, 0.2 * 255)$ ；在做cifar-100的噪点插入时，产生噪声的正态分布的标准偏差参数取 $(0.0, 0.1 * 255)$ 。

多种数据增广策略组合使用

- 仅仅使用一种数据增广策略获得的数据会显得比较单薄，并且获得的数据量可能仍然不够。所以我们可以采用多种增广策略组合的方式来弥补，这些增广的数据也可以使得后续训练后的模型鲁棒性更好。上述提到的所有单个增广策略都可以进行自由组合，但是必须得要注意“标签保留”的问题，因此在设置增广参数时要更加保守一些，也因为这样，对两个数据集我们采用同样的参数。本次作业以“裁剪+旋转变换+明亮度变换”、“平移变换+噪点插入”为例，进行数据增广。

裁剪+旋转+明亮度

- 由于采用了三种数据增广的技术，数据失真、标签丢失的可能性大大增加，因此在参数设置时，我们选择更为保守的方法来进行增广。作业中，裁剪范围缩小为 (0, 0.05)；旋转范围为 (-10, 10)；明亮度乘数收缩为 (0.9, 1.1)。

平移+噪点插入

- 同样，我们在设置参数时采用更为保守的策略来保证数据的可用性。x方向和y方向的平移范围收缩为 (-0.05, 0.05)；产生噪声的正态分布的标准偏差参数取(0.0, 0.05 * 255)

参考文献

[1] Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning[J]. Journal of Big Data, 2019, 6(1): 60.

[2] Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning[J]. arXiv preprint arXiv:1712.04621, 2017.

[3] Wang J, Perez L. The effectiveness of data augmentation in image classification using deep learning[J]. Convolutional Neural Networks Vis. Recognit, 2017, 11.

[4] <https://github.com/aleju/imgaug>

[5] <https://imgaug.readthedocs.io/en/latest/source/installation.html>

数据评估

代码框架与思路

1. 首先，做一些准备工作，编写一个数据评估类eval_class，该类详情如下：

属性：

- | | |
|---|-----------------------------|
| 1 | 1. 数据集（cifar10或cifar100） |
| 2 | 2. 数据类别classes_list（10或100） |
| 3 | 3. 批量大小batch_size |
| 4 | 4. 增强数据集 x_true, y_true |
| 5 | 5. 增广策略（比如crop、shift等） |
| 6 | 6. 模型 |

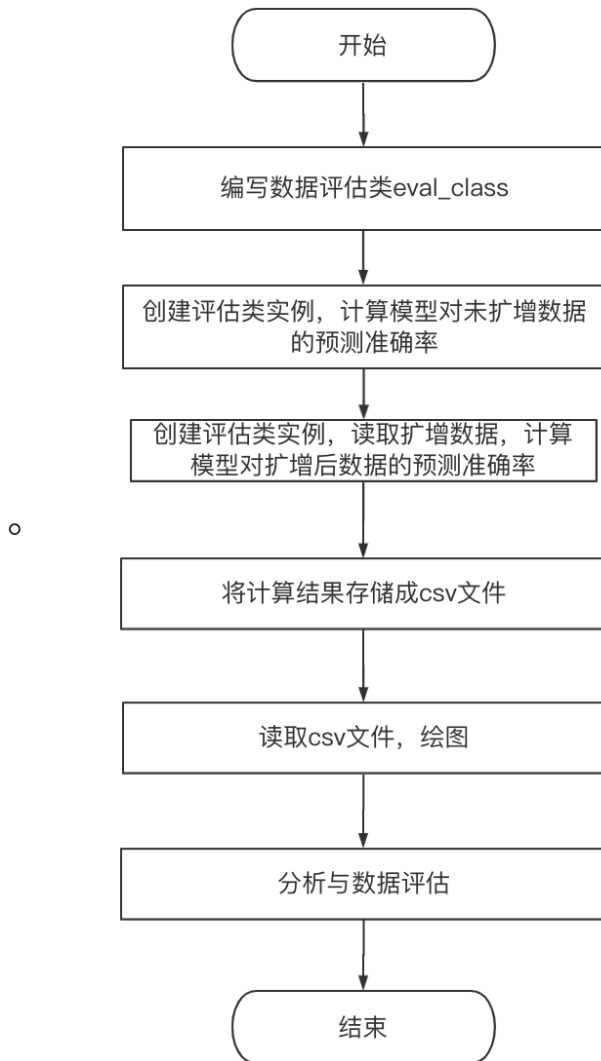
函数：

1. init ()：初始化数据集、数据类别list、批量大小、增强数据集
2. accuracy ()：准确率，用来评估某个模型在某种增广策略下的表现
3. predictiong ()：预测函数，用来预测某一张图片的分类，之后会调用accuracy ()来进行评估准确率

2. 然后，测试模型对原始数据集分类的表现，类似于baseline。因为两个数据集都有60000张图片，数据量过大，所以我们选取20000张来进行测试。只需创建一个上述数据评估类的实例，然后输入相应的参数进行初始化，调用相应函数测试每个模型对未增强数据集分类的表现，结果如下表：

o	Cifar-10	Cifar-100
CNN_with_dropout	65%	44%
CNN_without_dropout	72%	82%
ResNet_v1	69%	63%
ResNet_v2	72%	74%
lenet5_with_dropout	67%	55%
lenet5_without_dropout	75%	90%
random1	11%	50%
random2	11%	34%

3. 之后就可以对增广后的图片进行评估，测试每个模型对增广后图片进行分类的准确率。同样是创建实例，读取相应方法、数据集对应的numpy文件，比如读取使用crop方法扩增后的cifar10数据集 `aug_imgs_cifar10_crop_x.npy`、`aug_imgs_cifar10_crop_y.npy`，然后初始化实例、调用函数计算预测准确率、预测准确率下降值、预测精度丢失率，并存成csv。
4. 最后，读取csv文件数据，绘图，对得到的数据进行分析，使用相应度量方法（对于模型鲁棒性的提升效果）对数据质量进行评估。
- 相应流程图如下：



- 代码中eval_class类图如下

```

Project.data_evaluation.eval_class.eval_class
__init__(self,dataset_name,classes_num,batch_size,x_true,y_true,augmentation_policy,model,model_name)
accuracy(self,y_hat,y)
predicting(self,batch_idx)
  
```

Cifar-10扩增数据评估

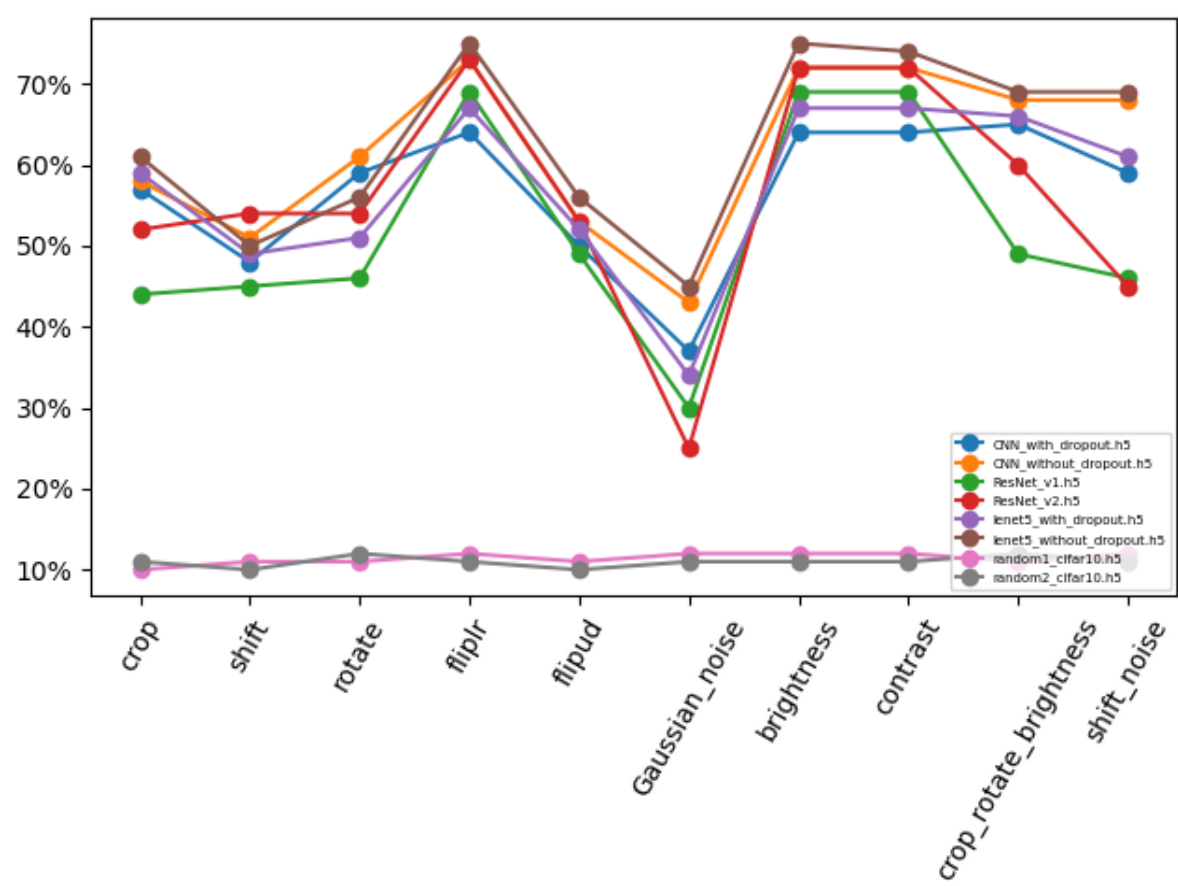
图表展示

每个模型对通过某种方法扩增的数据进行预测的准确率

- 首先，我们列出一张表格，来直观地表示每个模型对通过某种方法扩增的数据进行预测的准确率。如下表，其中base列代表该模型对未扩增数据的预测准确率。

ACCURACY	base	Crop	Shift	Rotate	Fliplr	Flipud	Gaussian Noise	Brightness	Contrast	Crop Rotate& Brightness	Shift& Noise
CNN_with_dropout	65%	57%	48%	59%	64%	50%	38%	64%	64%	64%	59%
CNN_without_dropout	72%	58%	51%	62%	72%	53%	43%	72%	72%	68%	68%
ResNet_v1	69%	45%	45%	46%	69%	49%	30%	69%	69%	50%	46%
ResNet_v2	72%	52%	54%	53%	72%	53%	26%	72%	72%	60%	45%
lenet5_with_dropout	67%	59%	49%	51%	67%	52%	34%	67%	67%	66%	61%
lenet5_without_dropout	75%	61%	50%	56%	75%	56%	45%	75%	74%	69%	69%
random1	11%	10%	11%	11%	12%	11%	12%	12%	12%	11%	12%
random2	11%	11%	10%	12%	11%	10%	11%	11%	11%	12%	11%

- 上述表格可以使用如下折线图表示，看得更为清楚：

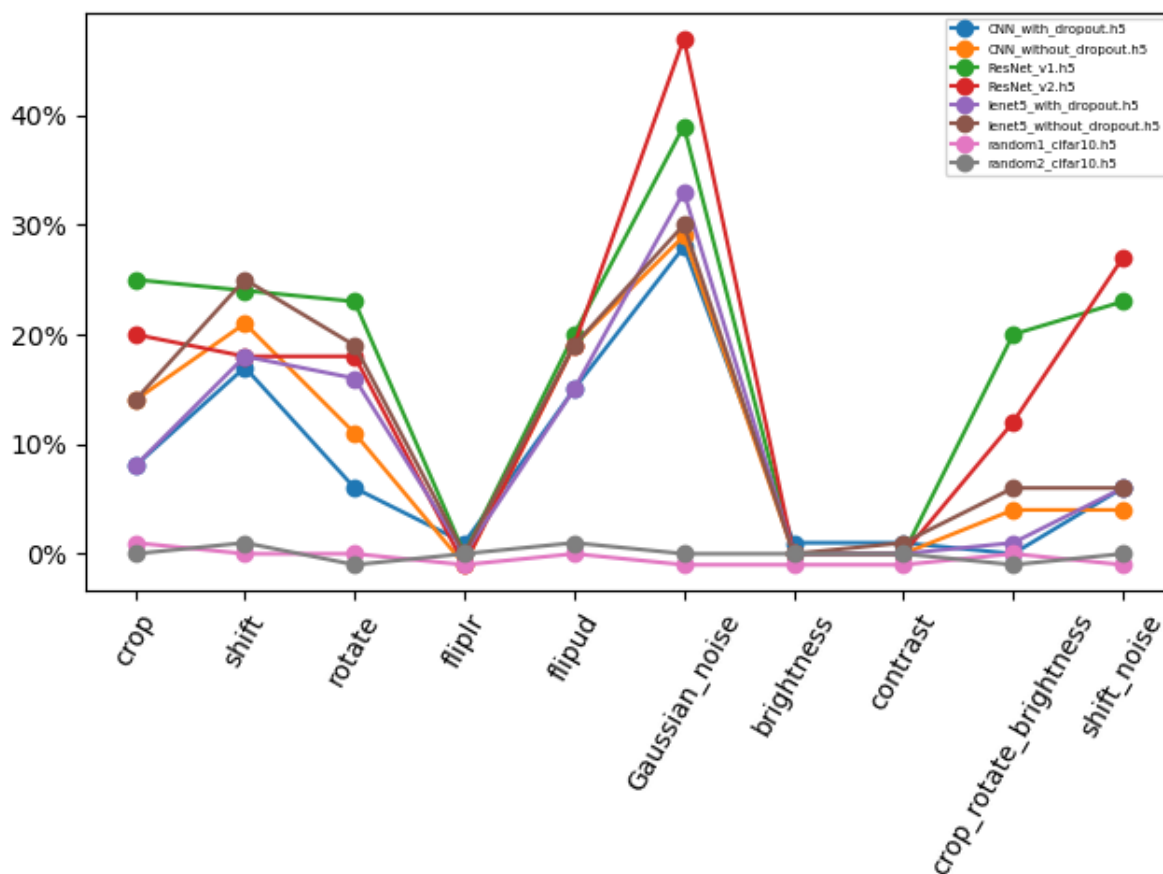


模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率

- 接着，我们列出另外一张表格，展示（模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率）：

	crop	shift	rotate	fliplr	flipud	additive_Gaussian_noise	brightness	contrast	crop_rotate_brightness	shift_noise
CNN_with_dropout.h5	-8%	-17%	-6%	-1%	-15%	-28%	-1%	-1%	0%	-6%
CNN_without_dropout.h5	-14%	-21%	-11%	1%	-19%	-29%	0%	0%	-4%	-4%
ResNet_v1.h5	-25%	-24%	-23%	0%	-20%	-39%	0%	0%	-20%	-23%
ResNet_v2.h5	-20%	-18%	-18%	1%	-19%	-47%	0%	0%	-12%	-27%
lenet5_with_dropout.h5	-8%	-18%	-16%	0%	-15%	-33%	0%	0%	-1%	-6%
lenet5_without_dropout.h5	-14%	-25%	-19%	0%	-19%	-30%	0%	-1%	-6%	-6%
random1_cifar10.h5	-1%	0%	0%	1%	0%	1%	1%	1%	0%	1%
random2_cifar10.h5	0%	-1%	1%	0%	-1%	0%	0%	0%	1%	0%

- 上述表格可以用如下折线图表示，看得更清楚。注意，在画折线图时对数据取反，从而方便看出数据预测准确率下降了多少



精度丢失率

- 上面的表格和图片已经可以比较明显地表示出扩增数据对模型预测准确率的影响了，在此，我们引入另外一个指标“精度丢失率”——accuracy_loss_rate,计算公式如下：

$$accuracy_loss_rate = \frac{base_accuracy - augmented_accuracy}{base_accuracy}$$

其中，参数解释如下：

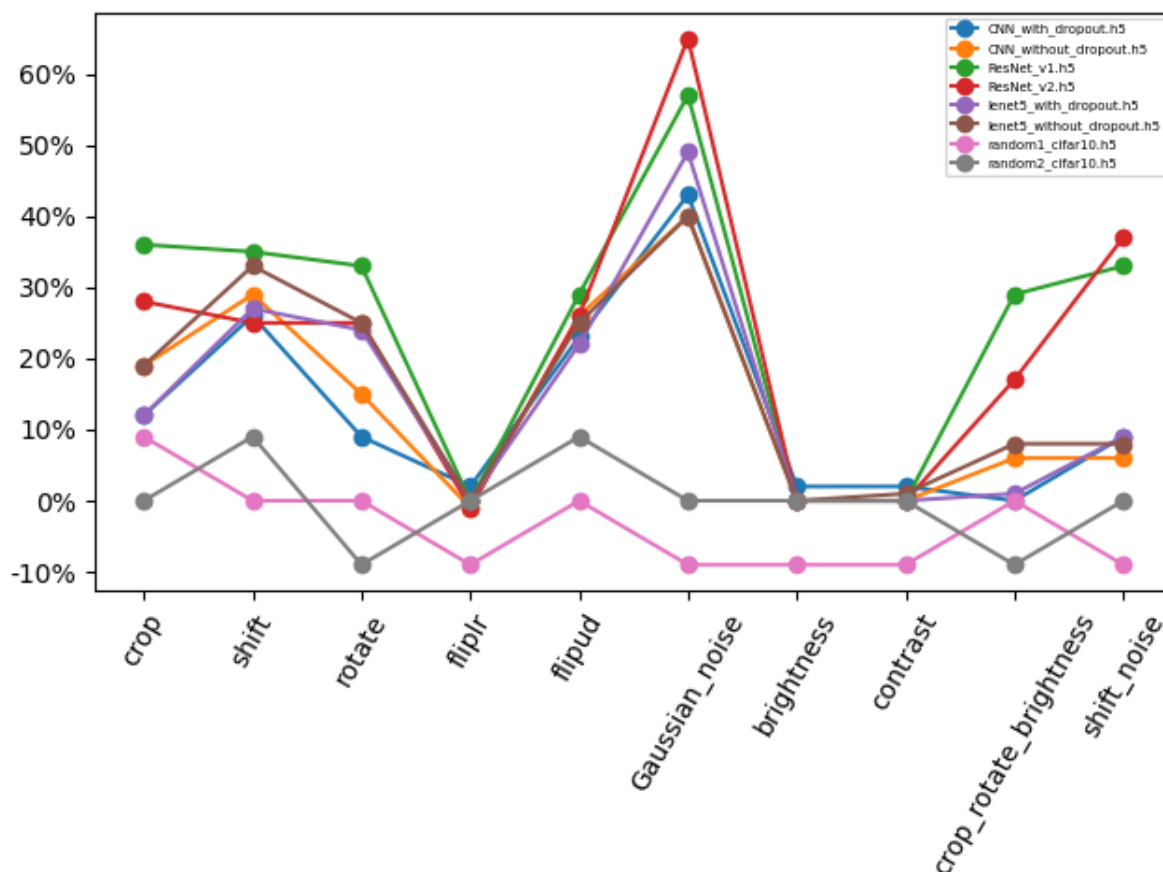
base_accuracy: 模型对未增广图片的预测正确率

augmented accuracy: 模型对增广后图片的预测正确率

- 我们计算上面表格对应的精度丢失率表格：

	crop	shift	rotate	fliplr	flipud	additive_Gaussian_noise	brightness	contrast	crop_rotate_brightness	shift_noise
CNN_with_dropout.h5	-12%	-26%	-9%	-2%	-23%	-43%	-2%	-2%	0%	-9%
CNN_without_dropout.h5	-19%	-29%	-15%	1%	-26%	-40%	0%	0%	-6%	-6%
ResNet_v1.h5	-36%	-35%	-33%	0%	-29%	-57%	0%	0%	-29%	-33%
ResNet_v2.h5	-28%	-25%	-25%	1%	-26%	-65%	0%	0%	-17%	-37%
lenet5_with_dropout.h5	-12%	-27%	-24%	0%	-22%	-49%	0%	0%	-1%	-9%
lenet5_without_dropout.h5	-19%	-33%	-25%	0%	-25%	-40%	0%	-1%	-8%	-8%
random1_cifar10.h5	-9%	0%	0%	9%	0%	9%	9%	9%	0%	9%
random2_cifar10.h5	0%	-9%	9%	0%	-9%	0%	0%	0%	9%	0%

- 我们同样绘制折线图，观察精度丢失率的情况，注意，这里我们同样需要将表格值取反，以便观察：



分析

- 我们做数据评估的质量指标是数据对于模型鲁棒性提升的能力。因此模型对某个扩增数据集的预测越不准确，我们就认为这个数据集质量较高。
- 通过比较我们发现，通过对cifar-10采取颜色空间（color space）变换（比如对比度、亮度）生成的数据集，质量并不高。因为使用这些数据测出来的模型预测准确率与使用未增强数据测试出来的区别并不大，所以它们对于模型鲁棒性的提升并不高。
- 而通过对cifar-10采取几何变换生成的扩增数据集，对于模型鲁棒性的提升很大。比如裁剪、平移、旋转等等，它们使得模型的预测准确率下降了不少，精度丢失率很高，可以说模型对于这些变换的适应性不太好，有待增强。
- 而通过插入噪点来扩增的数据集，使得每个模型的准确率下降很多，甚至有出现精度丢失率达到64%的，这不是提升模型鲁棒性了，而是改变了图形的标签，我们认为这些数据并不能称作“高质量”，而是“过度扩增”。
- 有意思的一点是，在几何变换中，通过水平翻转fliplr和垂直翻转flipud扩增产生的数据集展现了完全不同的质量。模型对通过水平翻转的数据集分类预测非常准确，接近100%，所以对模型

鲁棒性的提升不大，也表明模型在这方面鲁棒性比较高。而对垂直翻转的数据集分类预测不是很准确，因此我们认为其质量较高，对模型鲁棒性的提升很大。

- 而两个组合多种扩增方法形成的数据集，只有特定模型（ResNet_v1、ResNet_v2）在对其进行预测时才会准确率下降特别大，其他的模型对其进行预测是准确率下降不大，说明这两个组合方法扩增的数据集参数设置过于保守，质量不够高。
- 另外，两个random模型在预测扩增后的数据和未扩增数据集时，准确率没有太大的提升。
- 综合来看，对于两个random模型来说，这批扩增数据没有太高的质量；但是其他几个模型来说在预测这批扩增数据时会有较高的精度丢失率，因此这些数据对它们来说质量很高。

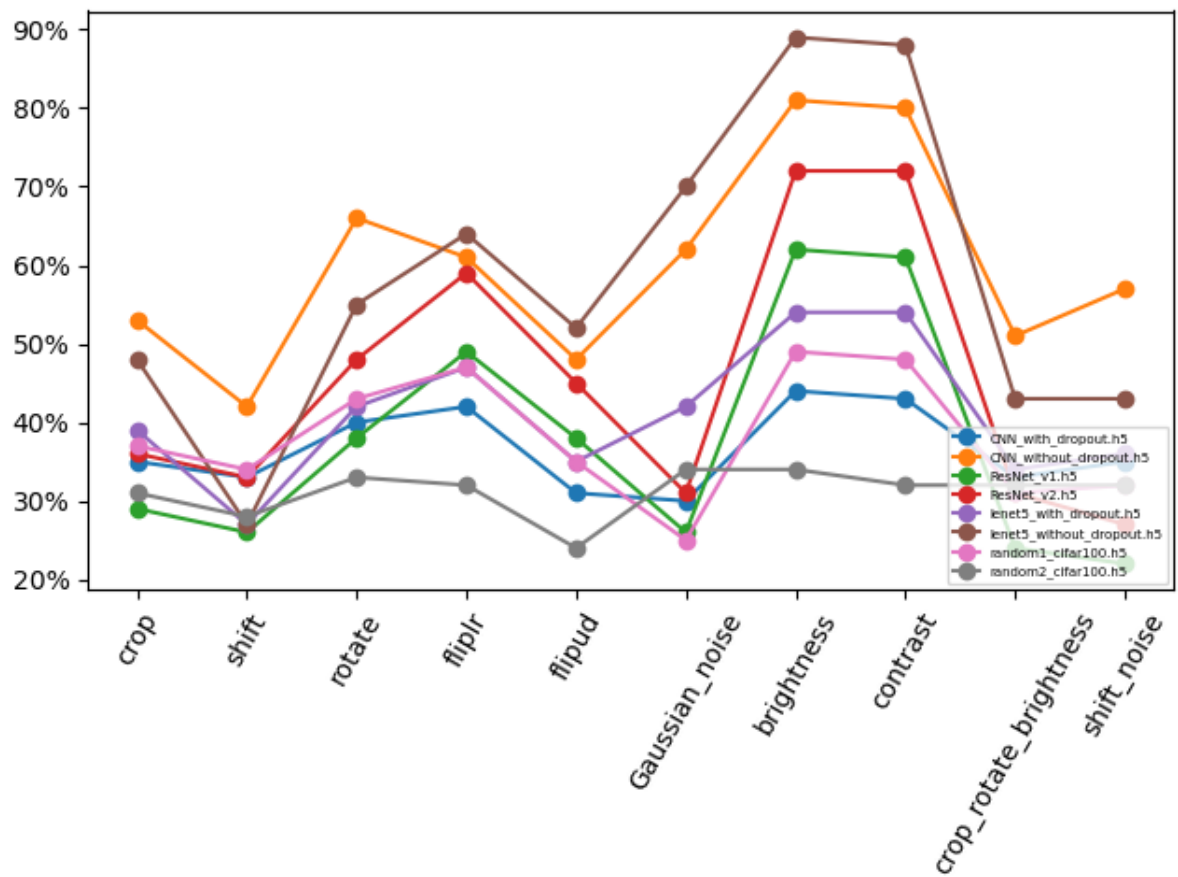
Cifar-100扩增数据评估

图表展示

- 首先，我们像处理cifar-10扩增数据评估时一样，首先列出三张表格和三张图表，分别展示：
 1. 每个模型对通过某种方法扩增的数据进行预测的准确率
 2. 模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率
 3. 精度丢失率

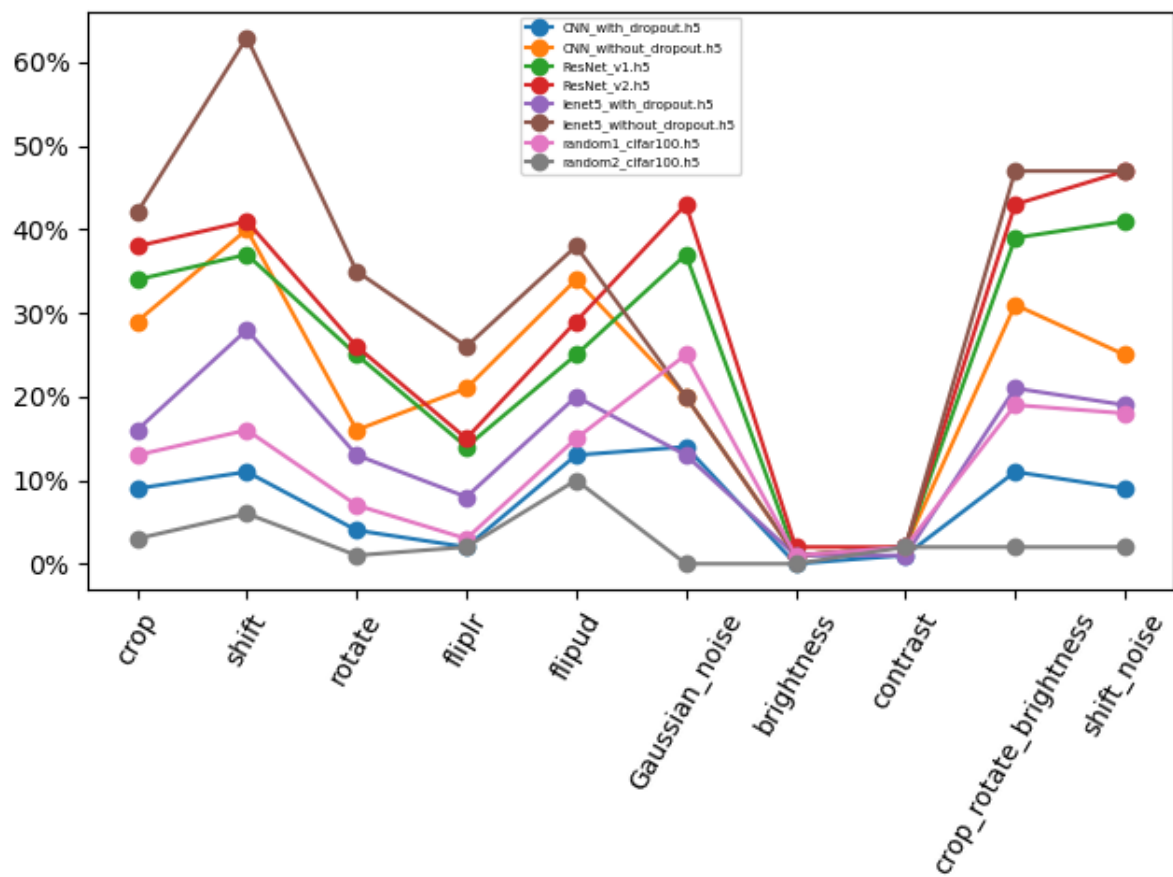
每个模型对通过某种方法扩增的数据进行预测的准确率

	base	crop	shift	rotate	fliplr	flipud	additive_Gaussian_noise	brightness	contrast	crop_rotate_brightness	shift_noise
CNN_with_dropout.h5	44%	35%	33%	40%	42%	31%	30%	44%	43%	33%	35%
CNN_without_dropout.h5	82%	53%	42%	66%	61%	48%	62%	81%	80%	51%	57%
ResNet_v1.h5	63%	29%	26%	38%	49%	38%	26%	62%	61%	24%	22%
ResNet_v2.h5	74%	36%	33%	48%	59%	45%	31%	72%	72%	31%	27%
lenet5_with_dropout.h5	55%	39%	27%	42%	47%	35%	42%	54%	54%	34%	36%
lenet5_without_dropout.h5	90%	48%	27%	55%	64%	52%	70%	89%	88%	43%	43%
random1_cifar100.h5	50%	37%	34%	43%	47%	35%	25%	49%	48%	31%	32%
random2_cifar100.h5	34%	31%	28%	33%	32%	24%	34%	34%	32%	32%	32%



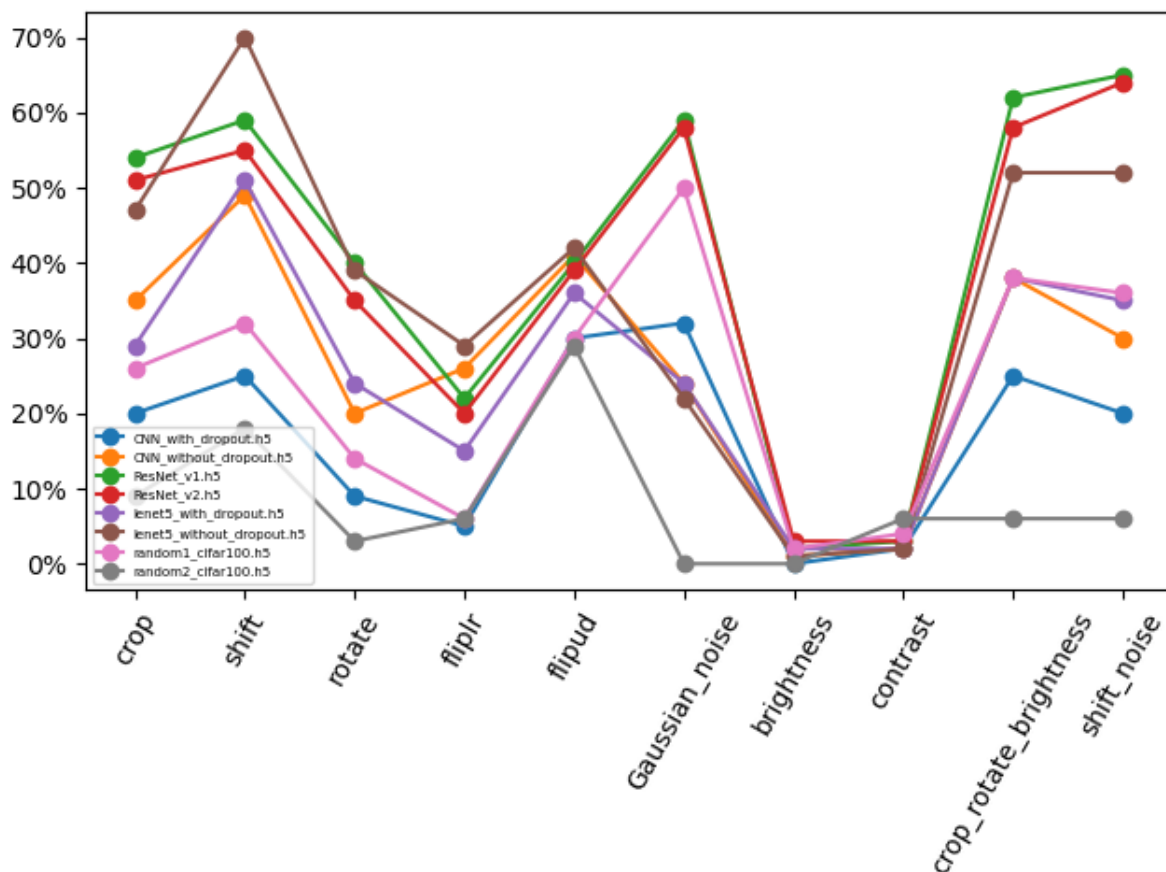
模型对扩增预测数据准确率 - 模型对未扩增数据预测准确率

	crop	shift	rotate	fliplr	flipud	additive_Gaussian_noise	brightness	contrast	crop_rotate_brightness	shift_noise
CNN_with_dropout.h5	-9%	-11%	-4%	-2%	-13%	-14%	0%	-1%	-11%	-9%
CNN_without_dropout.h5	-29%	-40%	-16%	-21%	-34%	-20%	-1%	-2%	-31%	-25%
ResNet_v1.h5	-34%	-37%	-25%	-14%	-25%	-37%	-1%	-2%	-39%	-41%
ResNet_v2.h5	-38%	-41%	-26%	-15%	-29%	-43%	-2%	-2%	-43%	-47%
lenet5_with_dropout.h5	-16%	-28%	-13%	-8%	-20%	-13%	-1%	-1%	-21%	-19%
lenet5_without_dropout.h5	-42%	-63%	-35%	-26%	-38%	-20%	-1%	-2%	-47%	-47%
random1_cifar100.h5	-13%	-16%	-7%	-3%	-15%	-25%	-1%	-2%	-19%	-18%
random2_cifar100.h5	-3%	-6%	-1%	-2%	-10%	0%	0%	-2%	-2%	-2%



精度丢失率

	crop	shift	rotate	fliplr	flipud	additive_Gaussian_noise	brightness	contrast	crop_rotate_brightness	shift_noise
CNN_with_dropout.h5	-20%	-25%	-9%	-5%	-30%	-32%	0%	-2%	-25%	-20%
CNN_without_dropout.h5	-35%	-49%	-20%	-26%	-41%	-24%	-1%	-2%	-38%	-30%
ResNet_v1.h5	-54%	-59%	-40%	-22%	-40%	-59%	-2%	-3%	-62%	-65%
ResNet_v2.h5	-51%	-55%	-35%	-20%	-39%	-58%	-3%	-3%	-58%	-64%
lenet5_with_dropout.h5	-29%	-51%	-24%	-15%	-36%	-24%	-2%	-2%	-38%	-35%
lenet5_without_dropout.h5	-47%	-70%	-39%	-29%	-42%	-22%	-1%	-2%	-52%	-52%
random1_cifar100.h5	-26%	-32%	-14%	-6%	-30%	-50%	-2%	-4%	-38%	-36%
random2_cifar100.h5	-9%	-18%	-3%	-6%	-29%	0%	0%	-6%	-6%	-6%



分析

- 前文提到，由于cifar-100有100个类别，因此我们在设置扩增参数时采取了更为保守的策略，但即便如此，模型预测扩增后数据时的表现依然不够完美，因此扩增的数据可以帮助其提高鲁棒性。
- 首先可以看到，与cifar-10类似，cifar-100在预测通过采取色彩空间（color space）变换增强的数据时，准确性没有下降很多，大多数模型的精度丢失率在（0，10%）之内。因此我们认为模型在这方面表现良好，通过调节亮度、对比度生成的扩增数据质量并不高，不能很好地帮助模型提高鲁棒性。
- 而对于通过几何变换扩增的数据集，模型预测准确率下降明显，丢失率大多超过了20%。其中，通过Crop或者Shift变换产生的扩增数据对于大多数模型来说是有较好质量的，尤其对于个别模型来说，比如ResNet_v1，质量非常高，因为这些模型在预测通过Crop或者Shift变换产生的扩增数据时，精度丢失率超过了50%；当然也可以认为这些模型在这方面的训练过少，需要进一步训练。而通过Rotate、Fliplr（水平翻转）或者Flipud（垂直翻转）扩增产生的数据集，则对所有模型来说都有很高的质量：这些模型对这些扩增数据预测时，精度丢失率大概在10%~35%，因此可以借助这些扩增数据来进行更好的优化，从而获得更好的鲁棒性。
- 对于通过插入噪点扩增产生的数据集，模型预测准确率下降程度不一。其中，random2模型预测准确率没有下降，可以认为此模型具有较好的鲁棒性，因此这些数据对其没有太多作用。而ResNet_v1和ResNet_v2在预测通过插入噪点产生的扩增数据时，精度丢失率很大，超过了50%，因此可以认为对于这两个模型来说数据质量极高；当然也可以认为这两个模型训练过少，需要进一步加强训练。而其他模型在对通过插入噪点产生的扩增数据集进行预测时，精度丢失率在20%~30%之间，对它们来说，这些数据具有很高的质量。
- 模型在预测两种通过组合方法扩增产生的数据集时，预测精度丢失率不一。这些数据对于CNN_with_dropout、CNN_without_dropout、lenet5_with_dropout、random1_cifar100来说，有很高的质量，因为它们在预测时精度丢失率在20%~40%之间，可以借助这些数据来继续训练，提高自身鲁棒性。而对于random2模型来说，预测这些数据时精度丢失率不高，因此这

些数据对它没有太大作用。对于剩下的3个模型（ResNet_v1、ResNet_v2、lenet5_without_dropout），两个通过组合方法产生的扩展数据集有极高的质量，因为这3个模型在预测这些数据时精度丢失率很高，超过了50%；当然也可以认为这些模型训练过少。

- 综合来看，ResNet_v1、ResNet_v2、lenet5_without_dropout在预测各个扩增数据集时，精度丢失率都过大，因此可以认为这些数据对它们来说质量极高。而random2则在预测所有扩增数据集时，精度丢失率不大，因此这些数据对它来说质量不高。对于剩下的模型，扩增的数据都有较高的质量。

参考文献

- [1] Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning[J]. Journal of Big Data, 2019, 6(1): 60.
- [2] Wang J, Perez L. The effectiveness of data augmentation in image classification using deep learning[J]. Convolutional Neural Networks Vis. Recognit, 2017, 11.
- [3] S. Bianco, L. Celona, P. Napoletano, and R. Schettini. On the use of deep learning for blind image quality assessment. arXiv preprint arXiv:1602.05531, 2016.
- [4] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek. Deep neural networks for no-reference and full-reference image quality assessment. IEEE Transactions on Image Processing, 27(1):206–219, 2018.
- [5] K. Gu, G. Zhai, X. Yang, and W. Zhang. Deep learning network for blind image quality assessment. In Image Processing (ICIP), 2014 IEEE International Conference on, pages 511–515. IEEE, 2014.