AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, Balaji Lakshminarayanan

---

Coby Penso

What we'll see

1. Introduction
2. AugMix
3. Experiments

Introduction - Robustness to Data Shifts

- Networks have a tendency to memorize properties of the specific training distortion.
- Robustness to data shift is a pressing problem which greatly affects the reliability of real-world machine learning systems.

Figure 2: Example ImageNet-C corruptions. These corruptions are encountered only at test time and not during training.

Introduction - Calibration under Data Shift

- RMS Calibration Error -
  Let the classifier's confidence that its prediction Y^ q is correct be written C.
  Then the idealized RMS Calibration Error is

$$\sqrt{\mathbb{E}_C\left[\mathbb{P}(Y = \hat{Y}|C = c) - c)^2\right]}$$

- Ensemble and Pre-training for better calibration
- **Ovadia et al. (2019) demonstrate that model calibration substantially deteriorates under data shift.**

Introduction - Data Augmentation

Data augmentation can greatly improve generalization performance.
There are many different methods, such as:

- Random left-right flipping and cropping.
- CutMix - replaces a portion of an image with a portion of a different image.
- MixUp - elementwise convex combination of two images.
- AutoAugment (Learned Augmentation method) - augmentations tuned to optimize performance on a downstream task.
- Patch Gaussian - Gaussian noise applied to a randomly chosen portion of an image.

AugMix - Motivation

- Simple augmentation operations in concert.
- Sampled stochastically and layered to produce a high diversity aug.
- Consistency loss.

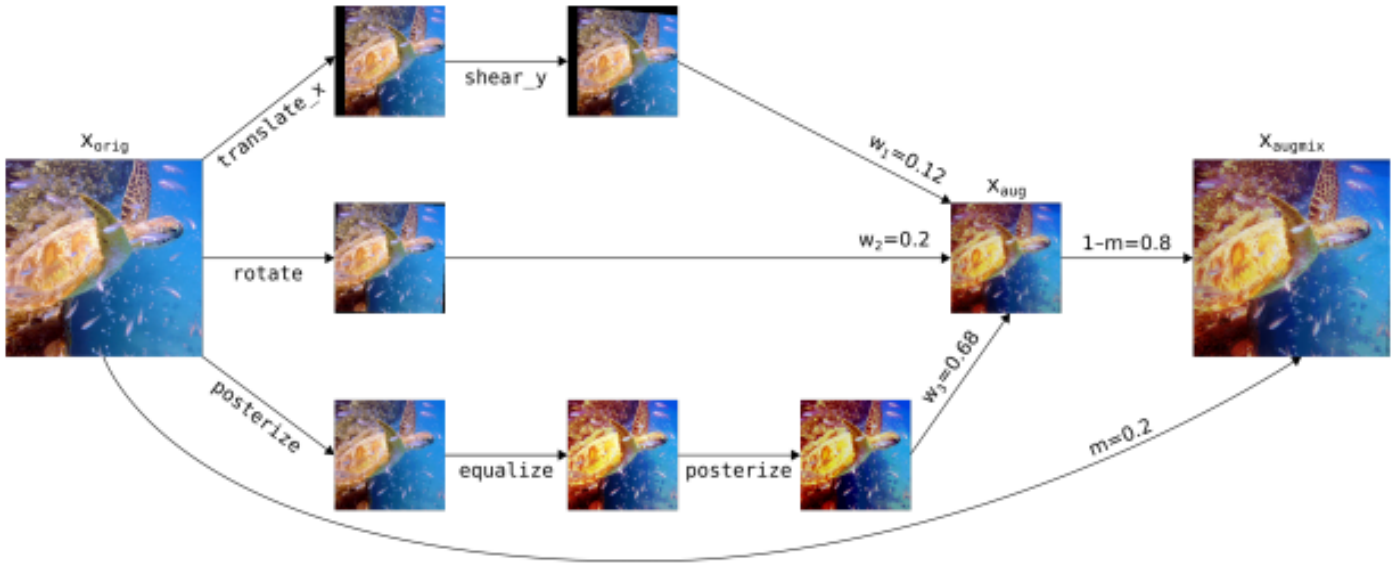


Figure 4: A realization of AUGMIX. Augmentation operations such as translate_x and weights such as m are randomly sampled. Randomly sampled operations and their compositions allow us to explore the semantically equivalent input space around an image. Mixing these images together produces a new image without veering too far from the original.

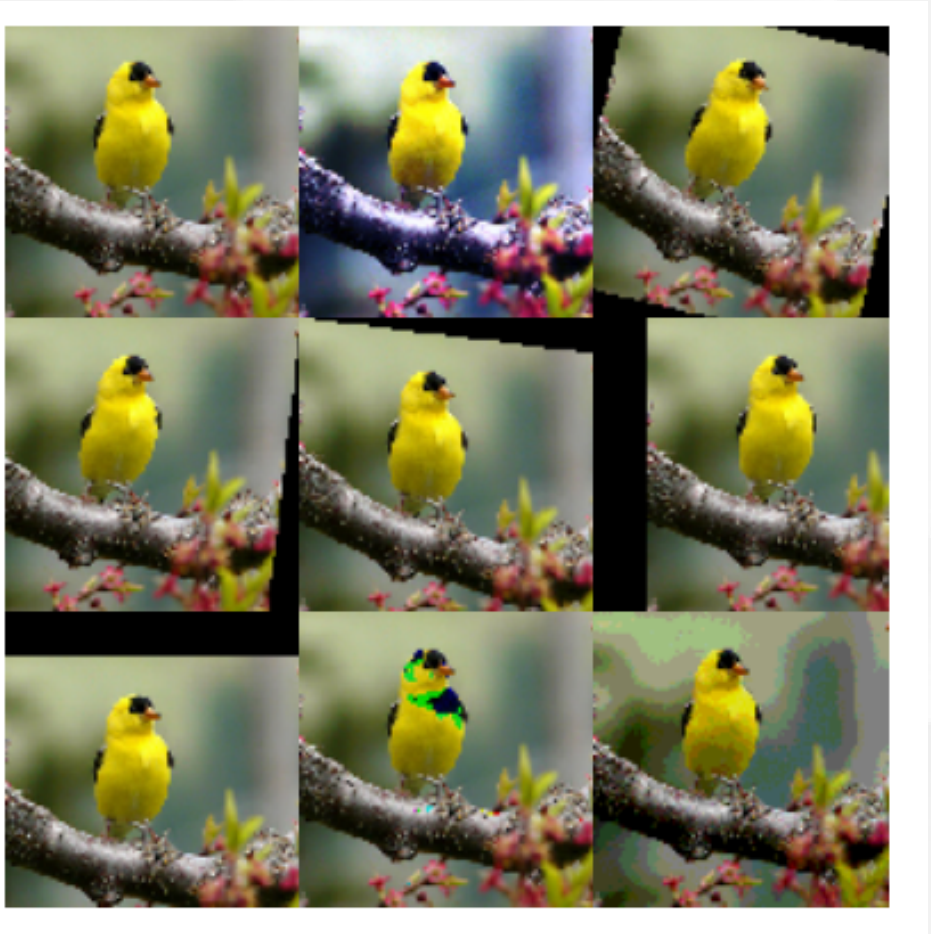
AugMix - Algorithm

Algorithm AUGMIX Pseudocode

Input: Model $\hat{p}$, Classification Loss $\mathcal{L}$, Image x_{orig}, Operations $\mathcal{O} = \{\text{rotate}, \ldots, \text{posterize}\}$
function AugmentAndMix(x_{orig}, $k = 3, \alpha = 1$)
  Fill x_{aug} with zeros
  Sample mixing weights $(w_1, w_2, \ldots, w_k) \sim \text{Dirichlet}(\alpha, \alpha, \ldots, \alpha)$
  for $i = 1, \ldots, k$ **do**
    Sample operations $\text{op}_1, \text{op}_2, \text{op}_3 \sim \mathcal{O}$
    Compose operations with varying depth $\text{op}_{12} = \text{op}_2 \circ \text{op}_1$ and $\text{op}_{123} = \text{op}_3 \circ \text{op}_2 \circ \text{op}_1$
    Sample uniformly from one of these operations chain $\sim \{\text{op}_1, \text{op}_{12}, \text{op}_{123}\}$
    $x_{\text{aug}} \mathrel{+}= w_i \cdot \text{chain}(x_{\text{orig}})$ ▷ *Addition is elementwise*
  end for
  Sample weight $m \sim \text{Beta}(\alpha, \alpha)$
  Interpolate with rule $x_{\text{augmix}} = m x_{\text{orig}} + (1 - m) x_{\text{aug}}$
  return x_{augmix}
end function
$x_{\text{augmix1}} = \text{AugmentAndMix}(x_{\text{orig}})$ ▷ $x_{augmix1}$ *is stochastically generated*
$x_{\text{augmix2}} = \text{AugmentAndMix}(x_{\text{orig}})$ ▷ $x_{augmix1} \neq x_{augmix2}$
Loss Output: $\mathcal{L}(\hat{p}(y \mid x_{\text{orig}}), y) + \lambda \, \text{Jensen-Shannon}(\hat{p}(y \mid x_{\text{orig}}); \hat{p}(y|x_{\text{augmix1}}); \hat{p}(y|x_{\text{augmix2}}))$

AugMix - Augmentations

- Operations from AutoAugment.
- Exclude operations which overlap with ImageNet-C corruptions.



AugMix - Mixing

- Elementwise convex combinations.
- The k-dimensional vector of convex coefficients is randomly sampled from a *Dirichlet*$(\alpha, ..., \alpha)$ distribution.
- “Skip connection” to combine the result of the augmentation chain and the original image through a second random convex combination sampled from a *Beta*(α, α) distribution.

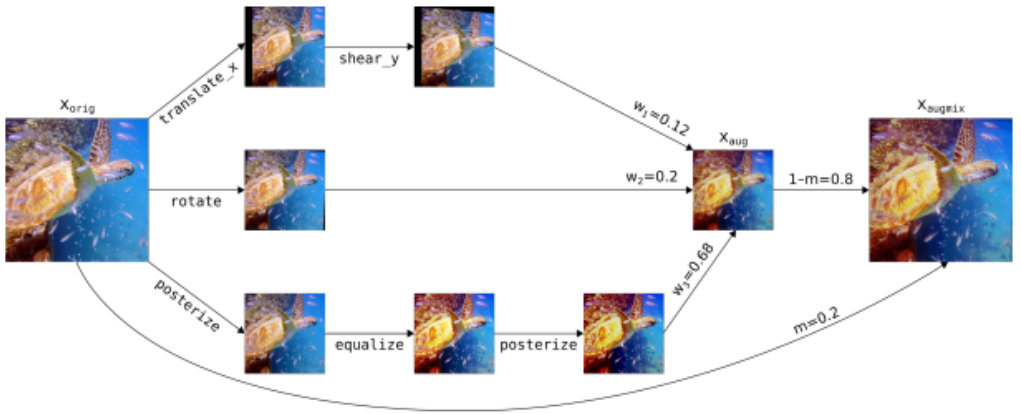


Figure 4: A realization of AUGMIX. Augmentation operations such as translate_x and weights such as m are randomly sampled. Randomly sampled operations and their compositions allow us to explore the semantically equivalent input space around an image. Mixing these images together produces a new image without veering too far from the original.

AugMix - Jensen-Shannon Divergence Consistency Loss

- Enforce a consistent embedding by the classifier across diverse augmentations of the same input image through the use of Jensen-Shannon divergence as a consistency loss.
- Motivation: semantic content of an image is approximately preserved with AugMix, we would like the model to embed $x_{orig}, x_{augmix_1}, x_{augmix_2}$ similarly.
-

$$p_{orig} = \hat{p}(y|x_{orig}), p_{augmix_1} = \hat{p}(y|x_{augmix_1}), p_{augmix_2} = \hat{p}(y|x_{augmix_2})$$

$$\mathcal{L}_{new} = \mathcal{L}(p_{orig}, y) + \lambda \mathbf{JS}(p_{orig}; p_{augmix_1}; p_{augmix_2})$$

Can be computed:

$$M = (p_{orig} + p_{augmix_1} + p_{augmix_2})/3$$

$$\mathbf{JS}(p_{orig}; p_{augmix_1}; p_{augmix_2}) = \frac{1}{3}(\mathbf{KL}[p_{orig}||M] + \mathbf{KL}[p_{augmix_1}||M] + \mathbf{KL}[p_{augmix_2}||M])$$

Datasets

- CIFAR10, CIFAR100, ImageNet
- CIFAR10-C, CIFAR100-C, ImageNet-C -
  15 noise, blur, weather, and digital corruption types, each appearing at 5 severity levels or intensities.
- CIFAR10-P, CIFAR100-P, ImageNet-P -
  smaller perturbations than C and used to measure the classifier's prediction stability.
  Each example in these datasets is a video.

Metrics

- Clean Error - error on the clean or uncorrupted test data.
- Unnormalized Corruption Error -

$$c - corruption, s - severity, E_{c,s}$$

$$uCE_c = \sum_{s=1}^{5} E_{c,s}$$

$$CE_c = \sum_{s=1}^{5} E_{c,s} / \sum_{s=1}^{5} E_{c,s}^{AlexNet}$$

- Mean Corruption Error - For CIFAR10/100 use uCE and for ImageNet use CE

$$mCE = AVG_c(CE_c)$$

- Mean Flipping Rate/Probability (mFR, mFP) -
  prediction mismatch between two adjacent frames in a video (P dataset)

CIFAR10 and CIFAR100

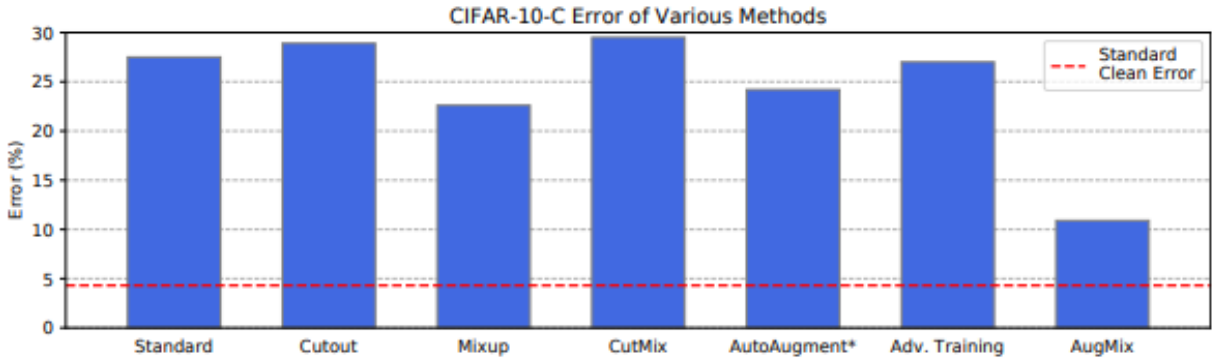


Figure 5: Error rates of various methods on CIFAR-10-C using a ResNeXt backbone. Observe that AUGMIX halves the error rate of prior methods and approaches the clean error rate.

		Standard	Cutout	Mixup	CutMix	AutoAugment*	Adv Training	AUGMIX
CIFAR-10-C	AllConvNet	30.8	32.9	24.6	31.3	29.2	28.1	**15.0**
	DenseNet	30.7	32.1	24.6	33.5	26.6	27.6	**12.7**
	WideResNet	26.9	26.8	22.3	27.1	23.9	26.2	**11.2**
	ResNeXt	27.5	28.9	22.6	29.5	24.2	27.0	**10.9**
Mean		29.0	30.2	23.5	30.3	26.0	27.2	**12.5**
CIFAR-100-C	AllConvNet	56.4	56.8	53.4	56.0	55.1	56.0	**42.7**
	DenseNet	59.3	59.6	55.4	59.2	53.9	55.2	**39.6**
	WideResNet	53.3	53.5	50.4	52.9	49.6	55.1	**35.9**
	ResNeXt	53.4	54.6	51.4	54.1	51.3	54.4	**34.9**
Mean		55.6	56.1	52.6	55.5	52.5	55.2	**38.3**

Table 1: Average classification error as percentages. Across several architectures, AUGMIX obtains CIFAR-10-C and CIFAR-100-C corruption robustness that exceeds the previous state of the art.

CIFAR10 and CIFAR100

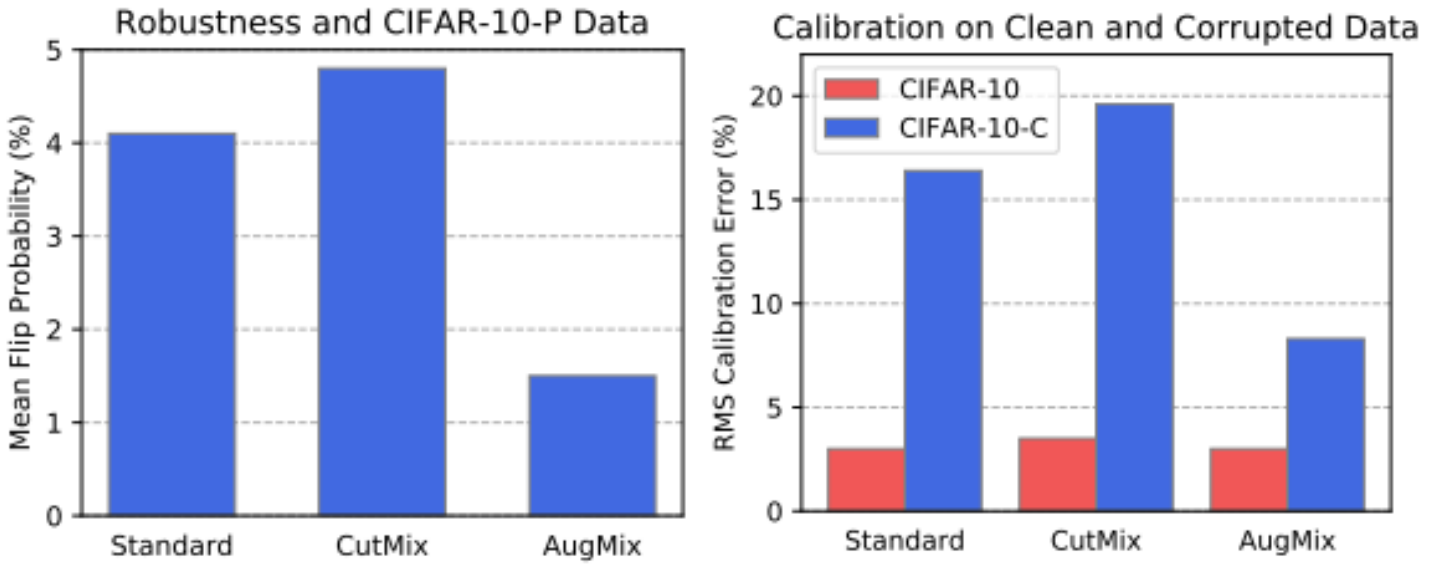


Figure 6: CIFAR-10-P prediction stability and Root Mean Square Calibration Error values for ResNeXt. AUGMIX simultaneously reduces flip probabilities and calibration error.

ImageNet - mCE

Training ResNet50 with standard training scheme of Goyal et al. (2017).

		Noise			Blur				Weather				Digital				
Network	Clean	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**mCE**
Standard	23.9	79	80	82	82	90	84	80	86	81	75	65	79	91	77	80	80.6
Patch Uniform	24.5	67	68	70	74	83	81	77	80	74	75	62	77	84	71	71	74.3
AutoAugment* (AA)	22.8	69	68	72	77	83	80	81	79	75	64	**56**	70	88	**57**	71	72.7
Random AA*	23.6	70	71	72	80	86	82	81	81	77	72	61	75	88	73	72	76.1
MaxBlur pool	23.0	73	74	76	74	86	78	77	77	72	**63**	**56**	68	86	71	71	73.4
SIN	27.2	69	70	70	77	84	76	82	74	75	69	65	69	80	64	77	73.3
AUGMIX	**22.4**	65	66	67	70	80	66	**66**	75	72	67	58	58	79	69	69	68.4
AUGMIX+SIN	25.2	**61**	**62**	**61**	**69**	**77**	**63**	72	**66**	**68**	**63**	59	**52**	**74**	60	**67**	**64.9**

Table 2: Clean Error, Corruption Error (CE), and mCE values for various methods on ImageNet-C. The mCE value is computed by averaging across all 15 CE values. AUGMIX reduces corruption error while improving clean accuracy, and it can be combined with SIN for greater corruption robustness.

ImageNet - mFR

		Noise		Blur		Weather		Digital				
Network	Clean	Gaussian	Shot	Motion	Zoom	Snow	Bright	Translate	Rotate	Tilt	Scale	**mFR**
Standard	23.9	57	55	62	65	66	65	43	53	57	49	57.2
Patch Uniform	24.5	**32**	**25**	50	52	54	57	40	48	49	46	45.3
AutoAugment* (AA)	22.8	50	45	57	68	63	53	40	44	50	46	51.7
Random AA*	23.6	53	46	53	63	59	57	42	48	54	47	52.2
SIN	27.2	53	50	57	72	51	62	43	53	57	53	55.0
MaxBlur pool	23.0	52	51	59	63	57	64	34	43	49	40	51.2
AUGMIX	**22.4**	46	41	**30**	**47**	38	**46**	**25**	**32**	**35**	**33**	**37.4**
AUGMIX+SIN	25.2	45	40	**30**	54	**32**	48	27	35	38	39	38.9

Table 3: ImageNet-P results. The mean flipping rate is the average of the flipping rates across all 10 perturbation types. AUGMIX improves perturbation stability by approximately 20%.

Questions?

Bibliography

- AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty https://arxiv.org/pdf/1912.02781.pdf