AUGMIX: A SIMPLE METHOD TO IMPROVE ROBUSTNESS AND UNCERTAINTY UNDER DATA SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern deep neural networks can achieve high accuracy when the training distribution and test distribution are identically distributed, but this assumption is frequently violated in practice. When the train and test distributions are mismatched, accuracy can plummet. Currently there are few techniques that improve robustness to data shift. In this work, we propose a technique to improve the robustness and uncertainty estimates of image classifiers. We propose AUGMIX, a data processing technique that is simple to implement, adds limited computational overhead, and helps models withstand data shift. AUGMIX significantly improves robustness and uncertainty measures on challenging image classification benchmarks, closing the gap between previous methods and the best possible performance in some cases by more than half.

1 INTRODUCTION

Current machine learning models depend on the ability of training data to faithfully represent the data encountered during deployment. In practice, data distributions evolve, models encounter new scenarios, and data curation procedures may capture only a narrow slice of the underlying data distribution. Mismatches between the train and test data are commonplace, yet the study of this problem is not. As it stands, models do not robustly generalize across shifts in the data distribution. If models could identify when they are likely to be mistaken, or estimate uncertainty accurately, then the impact of such fragility might be ameliorated. Unfortunately, modern models already produce overconfident predictions when the training examples are independent and identically distributed to the test distribution. This overconfidence and miscalibration is greatly exacerbated by mismatched training and testing distributions.

Small shifts in the data distribution are enough to subvert existing classifiers, and techniques to improve performance under data shift remain few in number. For instance, Recht et al. (2019) demonstrate that small changes in the dataset collection process for ImageNet can lead to significant declines in accuracy. Hendrycks & Dietterich (2019) show that classification error of modern models rises from 22% on the usual ImageNet test set to 64% on ImageNet-C, a test set consisting of various corruptions applied to ImageNet test images. Even methods which aim to explicitly quantify uncertainty, such as probabilistic and Bayesian neural networks, struggle under data shift, as recently demonstrated by Ovadia et al. (2019). Improving performance in this setting has been difficult. One reason is that training against corruptions only encourages networks to memorize the specific corruptions seen during training and leaves models unable to generalize to new corruptions (Vasiljevic et al., 2016; Geirhos et al., 2018). Further, networks trained on translation augmentations remain highly sensitive to images shifted by a single pixel (Gu et al., 2019; Hendrycks & Dietterich, 2019). Others have proposed aggressive data augmentation schemes (Cubuk et al., 2018), though at the cost of a 15000-fold increase in required computation. Chun et al. (2019) demonstrates that many techniques may improve clean accuracy at the cost of robustness while many techniques which improve robustness harm uncertainty, and contrariwise. In all, existing techniques have considerable trade-offs.

In this work, we propose a technique to improve both the robustness and uncertainty estimates of classifiers under data shift. We propose AUGMIX, a method which simultaneously achieves new state-of-the-art results for robustness and uncertainty estimation while maintaining or improving accuracy on standard benchmark datasets. AUGMIX utilizes stochasticity and diverse augmentations, a Jensen-Shannon Divergence consistency loss, and a formulation to mix multiple augmented images to achieve state-of-the-art performance. On CIFAR-10 and CIFAR-100, our method roughly halves

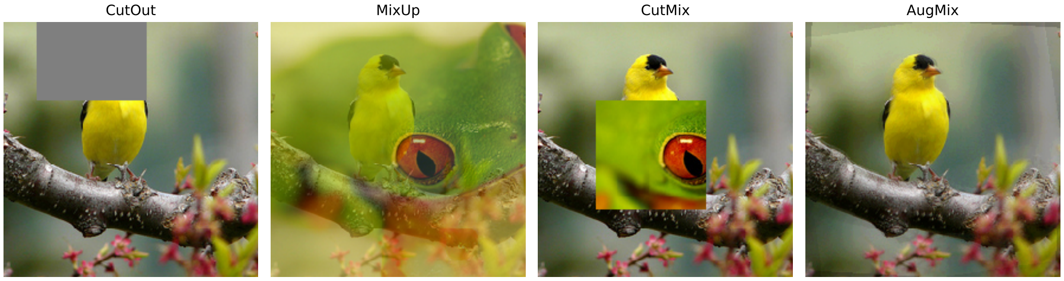

Figure 1: A visual comparison of data augmentation techniques. AUGMIX produces images with variety while preserving much of the image semantics and local statistics.

the corruption robustness error of standard training procedures from 28.4% to 12.4% and 54.3% to 37.8% error, respectively. On ImageNet, AUGMIX also achieves state-of-the-art corruption robustness and decreases perturbation instability from 57.2% to 37.4% mean flip probability.

2 Related Work

Robustness under Data Shift. Geirhos et al. (2018) show that training against distortions can often fail to generalize to unseen distortions, as networks have a tendency to memorize properties of the specific training distortion. Vasiljevic et al. (2016) show training with various blur augmentations can fail to generalize to unseen blurs or blurs with different parameter settings. Hendrycks & Dietterich (2019) propose measuring generalization to unseen corruptions and perturbations and provide benchmarks for doing so. Kang et al. (2019) construct an adversarial version of the aforementioned benchmark. Gilmer et al. (2018); Gilmer & Hendrycks (2019) argue that robustness to data shift is a pressing problem which greatly affects the reliability of real-world machine learning systems.

Calibration under Data Shift. Guo et al. (2017); Nguyen & O'Connor (2015) propose metrics for determining the calibration of machine learning models. Lakshminarayanan et al. (2017) find that simply ensembling classifier predictions improves prediction calibration. Hendrycks et al. (2019a) show that pre-training can also improve calibration. Ovadia et al. (2019) demonstrate that model calibration substantially deteriorates under data shift.

Data Augmentation. Data augmentation can greatly improve generalization performance. For image data, random left-right flipping and cropping are commonly used He et al. (2015). Random occlusion techniques such as Cutout can also improve accuracy on clean data Devries & Taylor (2017); Zhong et al. (2017). Rather than occluding a portion of an image, CutMix replaces a portion of an image with a portion of a different image Yun et al. (2019). Mixup also uses information from two images. Rather than implanting one portion of an image inside another, Mixup produces an elementwise convex combination of two images Zhang et al. (2017); Tokozume et al. (2018). Separate from these approaches are learned augmentation methods such as AutoAugment Cubuk et al. (2018), where a group of augmentations is tuned to optimize performance on a downstream task. However, determining this group of augmentations can cost upwards to 15,000 GPU hours. Patch Gaussian augments data with Gaussian noise applied to a randomly chosen portion of an image (Lopes et al., 2019). A popular way to make networks robust to ℓ_p adversarial examples is with adversarial training (Madry et al., 2018), which we use in this paper. However, this tends to increase

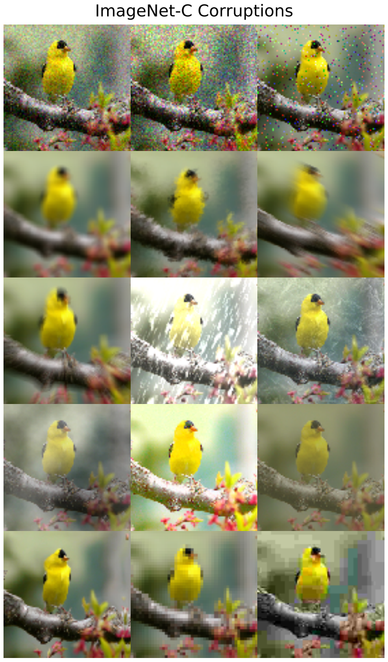

Figure 2: Example ImageNet-C corruptions. These corruptions are encountered only at test time and not during training.

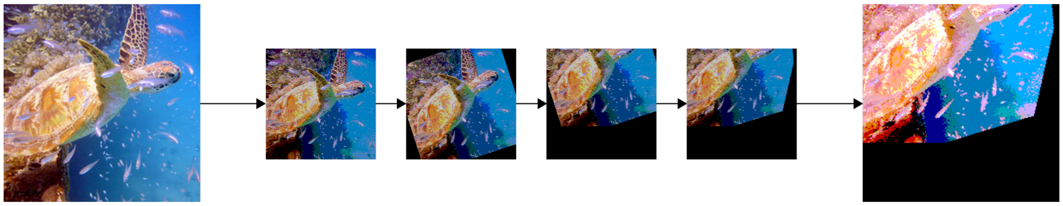

Figure 3: A cascade of successive compositions can produce images which drift far from the original image, and lead to unrealistic images. However, this divergence can be balanced by controlling the number of steps. To increase variety, we generate multiple augmented images and mix them.

training time by an order of magnitude and substantially degrades accuracy on non-adversarial images (Raghunathan et al., 2019).

3 AugMix

AugMix is a data augmentation technique which improves model robustness and uncertainty estimates, and slots in easily to existing training pipelines. At a high level, AugMix is characterized by its utilization of simple augmentation operations in concert with a consistency loss. These augmentation operations are sampled stochastically and layered to produce a high diversity of augmented images. We then 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.

Mixing augmentations allows us to generate diverse transformations, which are important for inducing robustness, as a common failure mode of deep models in the arena of corruption robustness is the memorization of fixed augmentations (Vasiljevic et al., 2016; Geirhos et al., 2018). Previous methods have attempted to increase diversity by directly composing augmentation primitives in a chain, but this can cause the image to quickly degrade and drift off the data manifold, as depicted in Figure 3. Such image degradation can be mitigated and the augmentation diversity can be maintained by mixing together the results of several augmentation chains in convex combinations. A concrete account of the algorithm is given in the pseudocode below.

Algorithm AugMix Pseudocode

```
Input: Model $\hat{p}$, Classification Loss $\mathcal{L}$, Image $x_{\text{orig}}$, Operations $\mathcal{O} = \{\text{rotate}, \ldots, \text{posterize}\}$
function AugmentAndMix($x_{\text{orig}}$, $k = 3, \alpha = 1, \beta = 1$)
    Fill $x_{\text{aug}}$ 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 $\sim \{\text{op}_1, \text{op}_{12}, \text{op}_{123}\}$
        $x_{\text{aug}} \mathrel{+}= w_i \cdot \texttt{chain}(x_{\text{orig}})$        ▷ Addition is elementwise
    end for
    Sample weight $m \sim \text{Beta}(\beta, \beta)$
    Interpolate with rule $x_{\text{augmix}} = m x_{\text{orig}} + (1 - m) x_{\text{aug}}$
    return $x_{\text{augmix}}$
end function
$x_{\text{augmix1}} = \text{AugmentAndMix}(x_{\text{orig}})$        ▷ $x_{\textit{augmix1}}$ is stochastically generated
$x_{\text{augmix2}} = \text{AugmentAndMix}(x_{\text{orig}})$        ▷ $x_{\textit{augmix1}} \neq x_{\textit{augmix2}}$
Loss Output: $\mathcal{L}(\hat{p}(y \mid x_{\text{orig}}), y) + \lambda\ \text{Jensen-Shannon}(\hat{p}(y \mid x_{\text{orig}}); \hat{p}(y|x_{\text{augmix1}}); \hat{p}(y|x_{\text{augmix2}}))$
```

Augmentations. Our method consists of mixing the results from augmentation chains or compositions of augmentation operations. We use operations from AutoAugment. Each operation is visualized in Appendix B. Crucially, *we exclude operations which overlap with ImageNet-C corruptions*. In particular, we remove the `contrast`, `color`, `brightness`, `sharpness`, and `Cutout` operations so that our set of operations and the ImageNet-C corruptions are disjoint. In turn, we do not use any

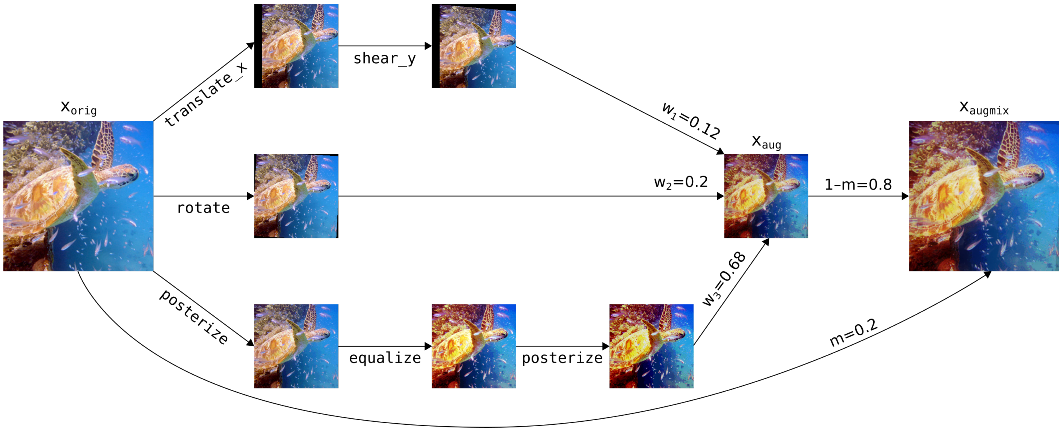

Figure 4: A realization of AugMix. Randomly sampled operations and their compositions allow us to explore the semantically meaningful input space around an image. Mixing these images together produces a new image without veering too far from the original.

image noising nor image blurring operations so that ImageNet-C corruptions are encountered only at test time. Operations such as rotate can be realized with varying severities, like 2° or -15°. For operations with varying severities, we uniformly sample the severity upon each application. Next, we randomly sample k augmentation chains, where $k = 3$ by default. Each augmentation chain is constructed by composing from one to three randomly selected augmentation operations.

Mixing. The resulting images from these augmentation chains are combined by mixing. While we considered mixing by alpha compositing, we chose to use elementwise convex combinations for simplicity. The k-dimensional vector of convex coefficients is randomly sampled from a Dirichlet$(\alpha, \ldots, \alpha)$ distribution. Once these images are mixed, we use a "skip connection" to combine the result of the augmentation chain and the original image through a second random convex combination sampled from a Beta(β, β) distribution. The final image incorporates several sources of randomness from the choice of operations, the severity of these operations, the lengths of the augmentation chains, and the mixing weights.

Jensen-Shannon Divergence Consistency Loss. We couple with this augmentation scheme a loss that enforces smoother neural network responses. Since the semantic content of an image is approximately preserved with AugMix, we should like the model to embed x_{orig}, x_{augmix1}, x_{augmix2} similarly. Toward this end, we minimize the Jensen-Shannon divergence among the posterior distributions of the original sample x_{orig} and its augmented variants. That is, for $p_{\text{orig}} = \hat{p}(y \mid x_{\text{orig}}), p_{\text{augmix1}} = \hat{p}(y \mid x_{\text{augmix1}}), p_{\text{augmix2}} = \hat{p}(y|x_{\text{augmix2}})$, we replace the original loss $\mathcal{L}$ with the loss

$$\mathcal{L}(p_{\text{orig}}, y) + \lambda \, \text{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}). \tag{1}$$

To interpret this loss, imagine a sample from one of the three distributions $p_{\text{orig}}, p_{\text{augmix1}}, p_{\text{augmix2}}$. The Jensen-Shannon divergence can be understood to measure the average information that the sample reveals about the identity of the distribution from which it was sampled.

This loss can be computed by first obtaining $M = (p_{\text{orig}} + p_{\text{augmix1}} + p_{\text{augmix2}})/3$ and then computing

$$\text{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}) = \frac{1}{3}\Big(\text{KL}[p_{\text{orig}} \| M] + \text{KL}[p_{\text{augmix1}} \| M] + \text{KL}[p_{\text{augmix2}} \| M]\Big). \tag{2}$$

Unlike an arbitrary KL Divergence between p_{orig} and p_{augmix}, the Jensen-Shannon divergence is upper bounded, in this case by the logarithm of the number of classes. Note that we could instead compute $\text{JS}(p_{\text{orig}}; p_{\text{augmix1}})$, though this does not perform as well. The gain of training with $\text{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}; p_{\text{augmix3}})$ is marginal. The Jensen-Shannon Consistency Loss impels to model to be stable, consistent, and insensitive across to a diverse range of inputs (Zheng et al., 2016; Kannan et al., 2018; Xie et al., 2019).

4 EXPERIMENTS

Datasets. The two CIFAR (Krizhevsky & Hinton, 2009) datasets contain small $32 \times 32 \times 3$ color natural images, both with 50,000 training images and 10,000 testing images. *CIFAR-10* has 10 categories, and *CIFAR-100* has 100. The *ImageNet* (Deng et al., 2009) dataset contains 1,000 classes of approximately 1.2 million large-scale color images.

In order to measure a model's resilience to data shift, we evaluate on the *CIFAR-10-C*, *CIFAR-100-C*, and *ImageNet-C* datasets (Hendrycks & Dietterich, 2019). These datasets are constructed by corrupting the original CIFAR and ImageNet test sets. For each dataset, there are a total of 15 noise, blur, weather, and digital corruption types, each appearing at 5 severity levels or intensities. Since these datasets are used to measure network behavior under data shift, we take care not to introduce these 15 corruptions into the training procedure.

The *CIFAR-10-P*, *CIFAR-100-P*, and *ImageNet-P* datasets also modify the original CIFAR and ImageNet datasets. These datasets contain smaller perturbations than CIFAR-C and are used to measure the classifier's prediction stability. Each example in these datasets is a video. For instance, a video with the brightness perturbation shows an image getting progressively brighter over time. We should like the network not to give inconsistent or volatile predictions between frames of the video as the brightness increases. Thus these datasets enable the measurement of the "jaggedness" (Azulay & Weiss, 2018) of a network's prediction stream.

Metrics. The *Clean Error* is the usual classification error on the clean or uncorrupted test data. In our experiments, corrupted test data appears at five different intensities or severity levels $1 \leq s \leq 5$. For a given corruption c, the error rate at corruption severity s is $E_{c,s}$. We can compute the average error across these severities to create the unnormalized corruption error $\text{uCE}_c = \sum_{s=1}^{5} E_{c,s}$. On CIFAR-10-C and CIFAR-100-C we average these values over all 15 corruptions. Meanwhile, on ImageNet we follow the convention of normalizing the corruption error by the corruption error of AlexNet (Krizhevsky et al., 2012). We compute $\text{CE}_c = \sum_{s=1}^{5} E_{c,s} / \sum_{s=1}^{5} E_{c,s}^{\text{AlexNet}}$. The average of the 15 corruption errors $\text{CE}_{\text{Gaussian Noise}}, \text{CE}_{\text{Shot Noise}}, \ldots, \text{CE}_{\text{Pixelate}}, \text{CE}_{\text{JPEG}}$ gives us the *Mean Corruption Error (mCE)*.

Perturbation robustness is not measured by accuracy but whether video frame predictions match. Consequently we compute what is called the *flip probability*. Concretely, for videos such as those with steadily increasing brightness, we determine the probability that two adjacent frames, or two frames with slightly different brightness levels, have "flipped" or mismatched predictions. There are 10 different perturbation types, and the mean across these is the *mean Flip Probability (mFP)*. As with ImageNet-C, we can normalize by AlexNet's flip probabilities and obtain the *mean Flip Rate (mFR)*.

In order to assess a model's uncertainty estimates, we measure its miscalibration. Classifiers capable of reliably forecasting their accuracy are considered "calibrated." For instance, a calibrated classifier should be correct 70% of the time on examples to which it assigns 70% confidence. Let the classifier's confidence that its prediction $\hat{Y}$ is correct be written C. Then the idealized RMS Calibration Error is $\sqrt{\mathbb{E}_C[(\mathbb{P}(Y = \hat{Y} | C = c) - c)^2]}$, which is the squared difference between the accuracy at a given confidence level and actual the confidence level. In Appendix A, we show how to empirically estimate this quantity and calculate the Brier Score.

4.1 CIFAR-10 AND CIFAR-100

Training Setup. In the following experiments we show that AUGMIX endows robustness to various architectures including an All Convolutional Network, a 40-2 Wide ResNet, and a ResNeXt-29 (32×4) (Salimans & Kingma, 2016; Springenberg et al., 2014; Zagoruyko & Komodakis, 2016; Xie et al., 2016). All networks use an initial learning rate of 0.1 which decays following a cosine learning rate (Loshchilov & Hutter, 2016). All input images are pre-processed with standard random left-right flipping and cropping prior to any augmentations. We do not change AUGMIX parameters across CIFAR-10 and CIFAR-100 experiments for consistency. The All-Convolutional Network and Wide ResNet train for 100 epochs, and ResNeXt requires 200 epochs for convergence. We optimize with stochastic gradient descent using Nesterov momentum and with a weight decay of 0.0005.

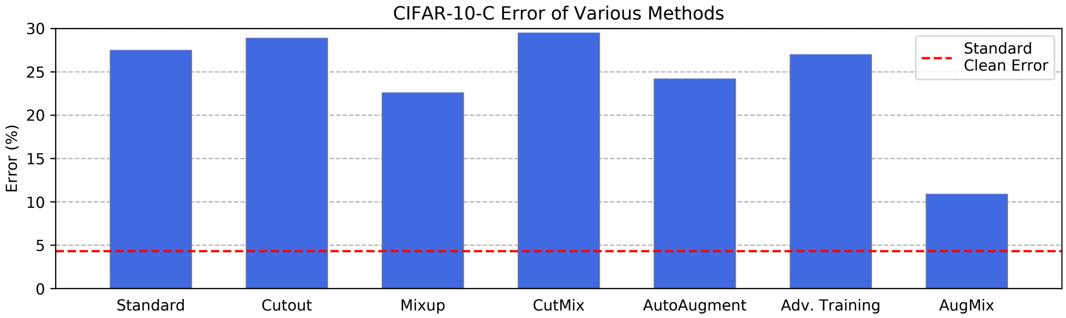

Figure 5: Error rates of various methods on CIFAR-10-C using a ResNeXt backbone. Observe that AUGMIX halves the error rate of prior methods and approaches the clean error rate.

		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**
	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		28.4	29.5	23.1	29.3	25.8	27.1	**12.4**
CIFAR-100-C	AllConvNet	56.4	56.8	53.4	56.0	55.1	56.0	**42.7**
	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		54.3	55.0	51.7	54.3	52.0	55.2	**37.8**

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.

Results. Simply mixing random augmentations and using the Jensen-Shannon loss substantially improves robustness and uncertainty estimates. Compared to the "Standard" data augmentation baseline ResNeXt on CIFAR-10-C, AUGMIX achieves 16.6% lower absolute corruption error as shown in Figure 5. In addition to surpassing numerous other data augmentation techniques, Table 1 demonstrates that these gains directly transfer across architectures and on CIFAR-100-C with zero additional tuning. Crucially, the robustness gains do not only exist when measured in aggregate. Figure 9 shows that AUGMIX improves corruption robustness across every individual corruption and severity level. Our method additionally achieves the lowest mFP on CIFAR-10-P across three different models all while maintaining accuracy on clean CIFAR-10 Table 6. Finally, we demonstrate that AUGMIX improves the RMS calibration error on CIFAR-10 and CIFAR-10-C, as shown in Table 5. Expanded CIFAR-10-P and calibration results are in Appendix C.

4.2 IMAGENET

Baselines. To demonstrate the utility of AUGMIX on ImageNet, we compare to many techniques designed for large-scale images. While techniques such as Cutout (Devries & Taylor, 2017) have not been demonstrated to help on the ImageNet scale, and while few have had success training adversarially robust models on ImageNet Engstrom et al. (2018), other techniques such as Stylized ImageNet have been demonstrated to help on ImageNet-C. Patch Uniform (Lopes et al., 2019) is similar to cutout except that randomly chosen regions of the image are injected with uniform noise; the original paper uses Gaussian noise, but that appears in the ImageNet-C test set so we use uniform noise. Next, AutoAugment (Cubuk et al., 2018) learns an augmentation policy over the course of approximately 15,000 GPU hours. When evaluating AutoAugment, we remove augmentation operations that overlap with ImageNet-C corruptions, as with AUGMIX. We also test with Random AutoAugment, an augmentation scheme where each image has a randomly sampled augmentation policy using AutoAugment operations. In contrast to AutoAugment, Random AutoAugment has typical computational requirements and provides more augmentation variety which can offset its lack of optimization. MaxBlur Pooling (Zhang, 2019) is a recently proposed architectural modification which smooths the results of pooling. Now, Stylized ImageNet (SIN) is a technique where models are trained with the original ImageNet images and also ImageNet images with style transfer applied. The

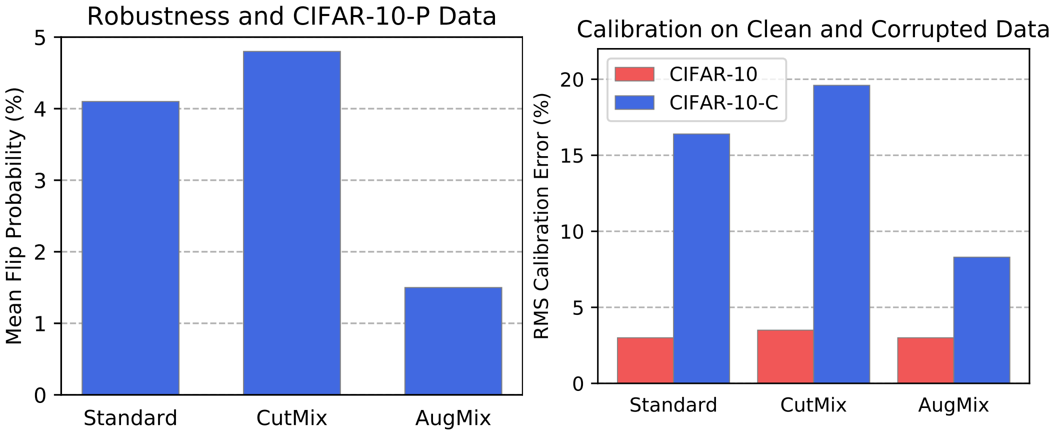

Figure 6: CIFAR-10-P prediction stability and Root Mean Square Calibration Error values for ResNeXt trained with various augmentation methods. AUGMIX simultaneously reduces flip probability and calibration error.

original Stylized ImageNet technique performs style transfer with a content loss coefficient of 0 and a style coefficient loss of 1, though we find that using 0.5 content and style loss coefficients decreases the mCE by 0.6%. Later, we show that SIN and AUGMIX can be combined. All models were trained from scratch, except MaxBlur Pooling models which had numerous pre-trained models available.

Training Setup. All methods are trained with ResNet-50 and we follow the standard training scheme of Goyal et al. (2017), in which we linearly scale the learning rate with the batch size, and use a learning rate warm-up for the first 5 epochs. All input images are pre-processed with standard random left-right flipping and cropping prior to any augmentations.

Results. We demonstrate substantial improvements across the board in terms of corruption robustness, perturbation robustness, calibration, and clean accuracy due to AUGMIX. On ImageNet-C our method achieves 67.9 mCE as shown in Table 2, down from the baseline 80.6% mCE. Additionally, we note that AUGMIX allows straightforward stacking with other methods such as SIN to achieve an even lower corruption error of 64.1% mCE. Across increasing severities of corruptions, our method also produces much more calibrated predictions measured by both the Brier Score and RMS Calibration Error as shown in Figure 7. As shown in Table 3, AUGMIX also achieves a state-of-the art result on ImageNet-P at with an mFR of 37.4%, down from 57.2%. We demonstrate that scaling up AUGMIX from CIFAR to ImageNet also leads to state-of-the-art results in robustness and uncertainty estimation.

			Noise			Blur				Weather				Digital			
Network	Clean	mCE	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Standard	23.9	80.6	79	80	82	82	90	84	80	86	81	75	65	79	91	77	80
Patch Uniform	23.9	79.4	81	84	73	92	81	82	93	77	74	44	119	69	87	66	68
AutoAugment (AA)	23.6	73.8	71	75	69	85	79	84	90	74	68	41	109	67	64	62	68
Random AA	23.6	76.1	70	71	72	80	86	82	81	81	77	72	61	75	88	73	72
MaxBlur pool	23.0	73.4	64	66	70	60	71	62	61	67	60	52	32	58	55	51	43
SIN	27.2	73.3	69	70	70	77	84	76	82	74	75	69	65	69	80	64	77
AUGMIX	**22.4**	67.9	66	71	64	83	66	70	81	68	65	39	83	61	77	57	68
AUGMIX+SIN	25.2	**64.1**	62	63	61	78	60	73	71	64	62	41	79	56	67	57	68

Table 2: Clean Error, mCE, and Corruption Error (CE) 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 lends itself to stacking with SIN to yield even greater corruption robustness.

4.3 ABLATIONS

We locate the utility of AUGMIX in three factors: training set diversity, our Jensen-Shannon divergence consistency loss, and mixing. Improving training set diversity via increased variety of

			Noise		Blur		Weather		Digital			
Network	Clean	mFR	Gaussian	Shot	Motion	Zoom	Snow	Bright	Translate	Rotate	Tilt	Scale
Standard	23.9	57.2	57	55	62	65	66	65	43	53	57	49
Patch Uniform	23.9	57.6	59	55	63	67	67	64	43	52	56	49
AutoAugment (AA)	23.6	52.5	49	45	57	68	64	56	42	46	51	46
Random AA	23.6	52.2	53	46	53	63	59	57	42	48	54	47
SIN	27.2	55.0	53	50	57	72	51	62	43	53	57	53
MaxBlur pool	23.0	51.2	52	51	59	63	57	64	34	43	49	40
AugMix	**22.4**	**37.4**	46	41	30	47	38	46	25	32	35	33
AugMix+SIN	25.2	38.9	45	40	30	54	32	48	27	35	38	39

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%.

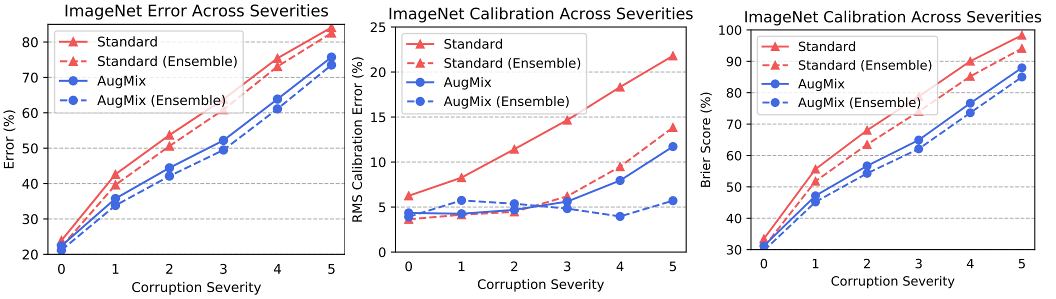

Figure 7: Uncertainty results on ImageNet. Under severe data shifts, the improvement to robustness and uncertainty estimation from using AugMix is substantial. Severity zero denotes clean data.

augmentations can greatly improve robustness. For instance, augmenting each example with a randomly sampled augmentation chain decreases the error rate of Wide ResNet on CIFAR-10-C from 26.9% to 17.0% Table 4. Adding in the Jensen-Shannon divergence consistency loss drops error rate further to 14.7%. Mixing random augmentations without the Jenson-Shannon divergence loss gives us an error rate of 13.1%. Finally, re-introducing the Jensen-Shannon divergence gives us AugMix with an error rate of 11.2%. Note that adding even more mixing is not necessarily beneficial. For instance, applying AugMix on top of Mixup increases the error rate to 13.3%. Hence AugMix's careful combination of variety, consistency loss, and mixing explain its performance.

Method	CIFAR-10-C Error Rate	CIFAR-100-C Error Rate
Standard	26.9	53.3
AutoAugment	23.9	49.6
Random AutoAugment	17.0	43.6
Random AutoAugment + JSD Loss	14.7	40.8
AugmentAndMix (No JSD Loss)	13.1	39.8
AugMix (Mixing + JSD Loss)	11.2	35.9

Table 4: Ablating components of AugMix on CIFAR-10-C and CIFAR-100-C. Variety through randomness, the Jensen-Shannon divergence consistency loss, and augmentation mixing each confer robustness.

5 Conclusion

AugMix is a data processing technique which mixes randomly generated augmentations and uses a Jensen-Shannon loss to enforce consistency. Our technique is simple to implement, adds little computational overhead, and substantially improves robustness and uncertainty. Specifically, we found that AugMix obtains state-of-the-art performance on CIFAR-10/100-C, ImageNet-C, CIFAR-10/100-P, ImageNet-P, and it also achieves state-of-the-art calibration under data shift. We hope this will enable more reliable models, a necessity for models deployed in safety-critical environments.

REFERENCES

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint*, 2018.

Sanghyuk Chun, Seong Joon Oh, Sangdoo Yun, Dongyoon Han, Junsuk Choe, and Youngjoon Yoo. An empirical evaluation on robustness and uncertainty of regularization methods. *Uncertainty and Robustness in Deep Learning. ICML Workshop*, 2019.

Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation policies from data. *CVPR*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *CVPR*, 2009.

Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with Cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint*, 2018.

Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. *NeurIPS*, 2018.

Justin Gilmer and Dan Hendrycks. A discussion of 'adversarial examples are not bugs, they are features': Adversarial example researchers need to expand what is meant by 'robustness'. *Distill*, 2019. doi: 10.23915/distill.00019.1. https://distill.pub/2019/advex-bugs-discussion/response-1.

Justin Gilmer, Ryan P. Adams, Ian J. Goodfellow, David Andersen, and George E. Dahl. Motivating the rules of the game for adversarial example research. *CoRR*, abs/1807.06732, 2018.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *CoRR*, abs/1706.02677, 2017.

Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathon Shlens. Using videos to evaluate image model robustness, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ICML*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2015.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019a.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019b.

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint*, 2019.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *NeurIPS*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *NeurIPS*, 2012.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin Dogus Cubuk. Improving robustness without sacrificing accuracy with patch Gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. *ICLR*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. *EMNLP*, 2015.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *NeurIPS*, 2019.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? *ArXiv*, abs/1902.10811, 2019.

Tim Salimans and Diederik Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *NeurIPS*, 2016.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. *CVPR*, pp. 5486–5494, 2018.

Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks, 2016.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CVPR*, 2016.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *ICCV*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2017.

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.485. URL `http://dx.doi.org/10.1109/cvpr.2016.485`.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

A Calibration Metrics

Due to the finite size of empirical test sets, the RMS Calibration Error must be estimated by partitioning all n test set examples into b contiguous bins $\{B_1, B_2, \ldots, B_b\}$ ordered by prediction confidence. In this work we use bins which contain 100 predictions, so that we adaptively partition confidence scores on the interval $[0, 1]$ (Nguyen & O'Connor, 2015; Hendrycks et al., 2019b). Other works partition the interval $[0, 1]$ with 15 bins of uniform length (Guo et al., 2017). With these b bins, we estimate the *RMS Calibration Error* empirically with the formula

$$\sqrt{\sum_{i=1}^{b} \frac{|B_i|}{n}\left(\frac{1}{|B_i|}\sum_{k\in B_i}\mathbb{1}(y_k = \hat{y}_k) - \frac{1}{|B_i|}\sum_{k\in B_i} c_k\right)^2}. \tag{3}$$

This is separate from classification error because a random classifier with an approximately uniform posterior distribution is approximately calibrated. Also note that adding the "refinement" $\mathbb{E}_C[(\mathbb{P}(Y = \hat{Y}|C = c)(1 - (\mathbb{P}(Y = \hat{Y}|C = c))]$ to the square of the RMS Calibration Error gives us the *Brier Score* (Nguyen & O'Connor, 2015).

B Augmentation Operations

The augmentation operations we use for AugMix are shown in 8.

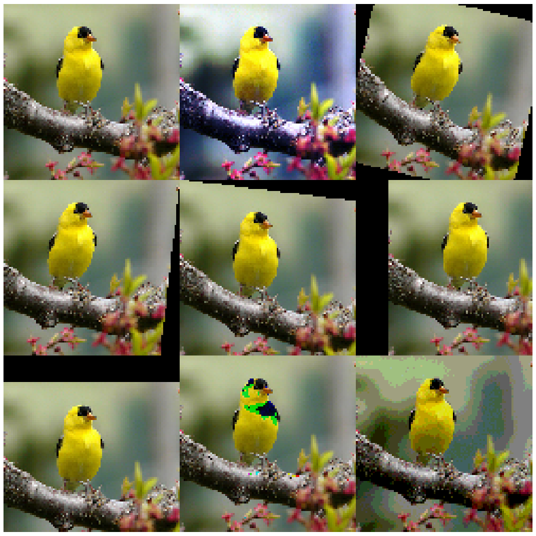

Figure 8: Illustration of augmentation operations applied to the same image. Some operation severities have been increased to show detail.

C Additional Results

In the tables and figures below, we include various additional results.

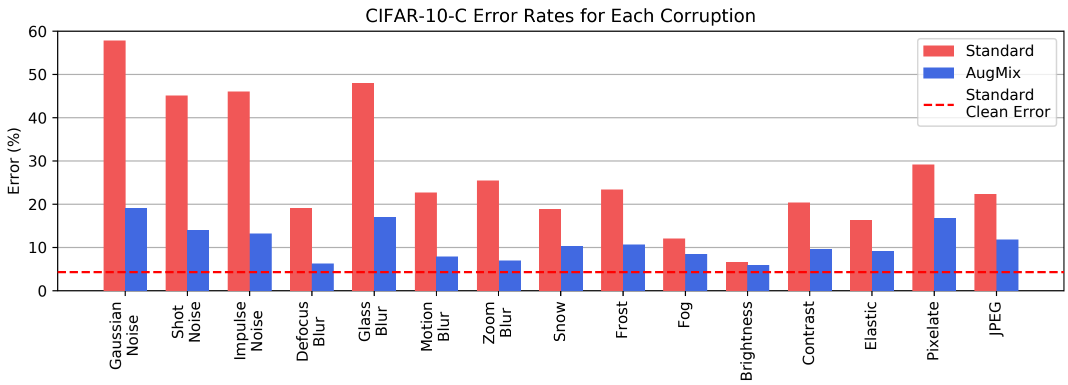

Figure 9: AUGMIX improves corruption robustness across all CIFAR-10-C noise, blur, weather, and digital corruptions, despite the model never having seen these corruptions during training.

		Standard	Cutout	Mixup	CutMix	AutoAugment	Adv Training	AUGMIX
CIFAR-10	AllConvNet	5.4	4.0	12.6	3.1	4.2	11.1	2.2
	WideResNet	6.8	3.8	14.0	5.0	4.7	10.7	4.2
	ResNeXt	3.0	4.4	13.5	3.5	3.3	5.8	3.0
Mean		5.1	4.1	13.4	3.9	4.1	9.2	**3.1**
CIFAR-10-C	AllConvNet	21.2	21.3	9.7	15.4	16.2	10.4	5.2
	WideResNet	27.6	19.6	11.1	17.8	17.1	10.6	8.7
	ResNeXt	16.4	21.4	11.7	19.6	15.1	11.6	8.3
Mean		21.7	20.8	10.8	17.6	16.1	10.8	**7.4**

Table 5: RMS Calibration Error of various models and data augmentation methods across CIFAR-10 and CIFAR-10-C. All values are reported as percentages.

		Standard	Cutout	Mixup	CutMix	AutoAugment	Adv Training	AUGMIX
CIFAR-10	AllConvNet	6.1	6.1	6.3	6.4	6.6	18.9	6.5
	WideResNet	5.2	4.4	4.9	4.6	4.8	17.1	4.9
	ResNeXt	4.3	4.4	4.2	3.9	3.8	15.4	4.2
Mean		5.2	5.0	5.1	5.0	5.1	17.1	5.2
CIFAR-10-P	AllConvNet	4.2	5.0	3.9	4.5	4.0	2.0	1.5
	WideResNet	4.2	4.3	3.4	4.6	4.2	2.2	1.6
	ResNeXt	4.0	4.5	3.2	5.2	4.2	2.5	1.5
Mean		4.1	4.6	3.5	4.8	4.1	2.2	1.5

Table 6: CIFAR-10 Accuracy and CIFAR-10-P mean Flip Probability. All values are percentages. While adversarial training performs well on CIFAR-10-P, it induces a substantial drop in accuracy on clean CIFAR-10 where AUGMIX does not.