Differential Operators for Structured Adversarial Examples

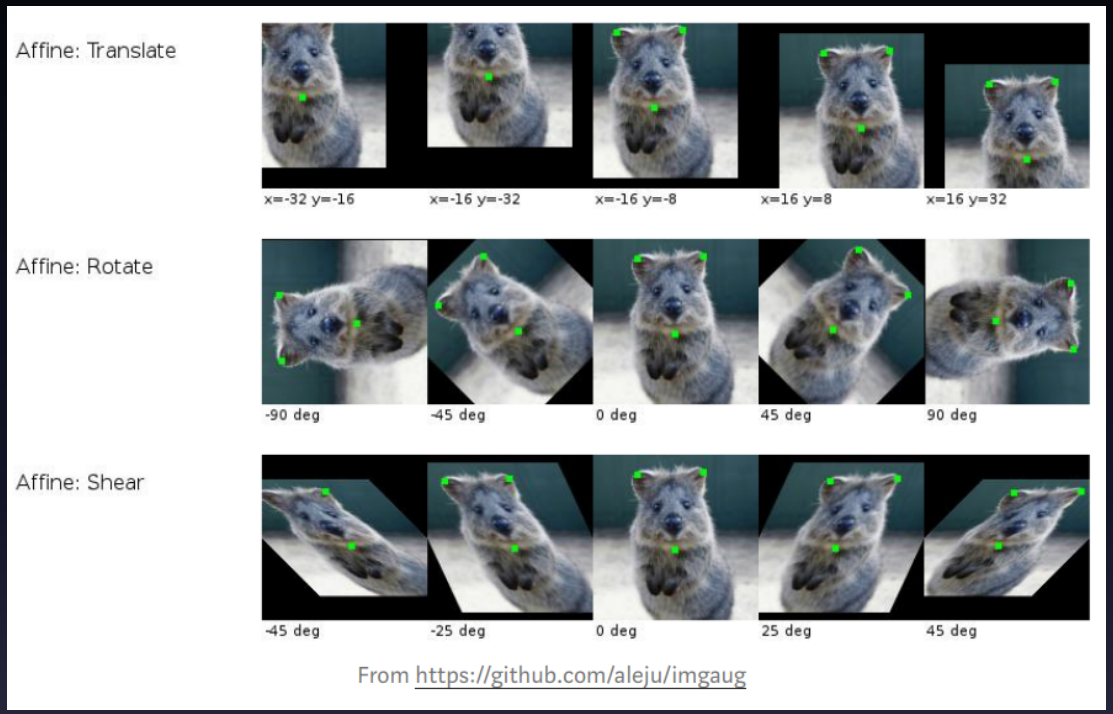
Hossein Mobahi

Joint work with
Calvin Luo
Samy Bengio

Google Research

April 22, 2020

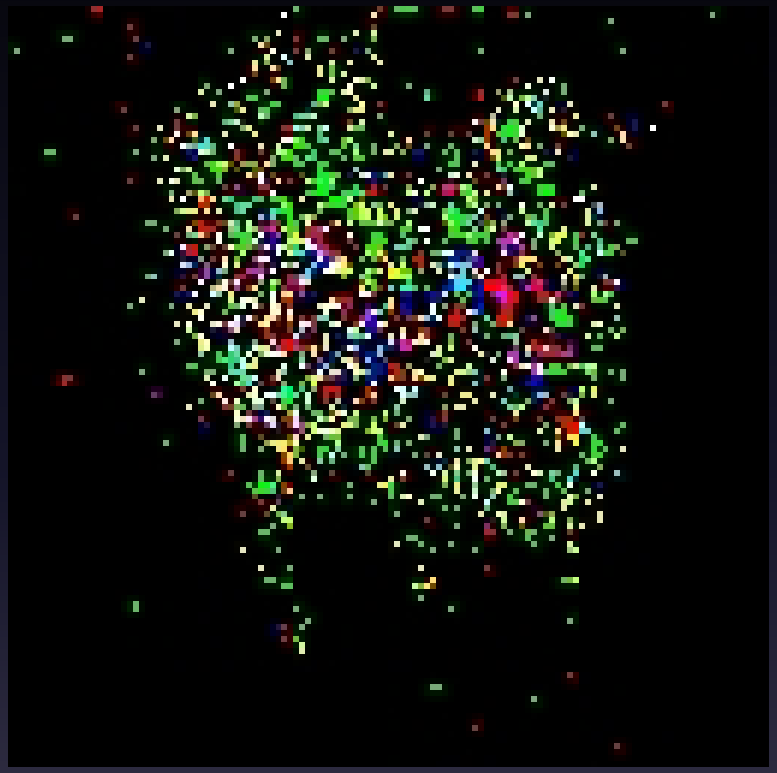
Data Augmentation



- Data augmentation is an important ingredient for achieving SOTA on image classification tasks.
- Currently incorporated into training by sampling; may not scale to larger space of transformations.

Adversarial Examples

Original Image

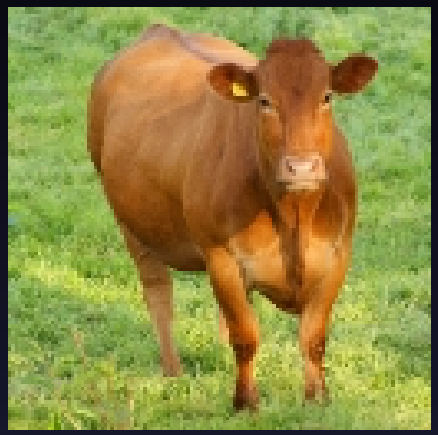


Adversarial Perturbation

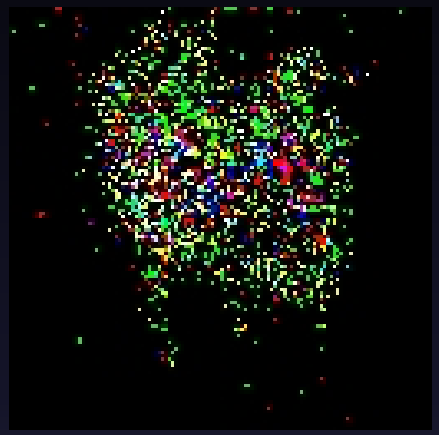
Very rich set of perturbations, that often fall outside of natural transformations.

Goal

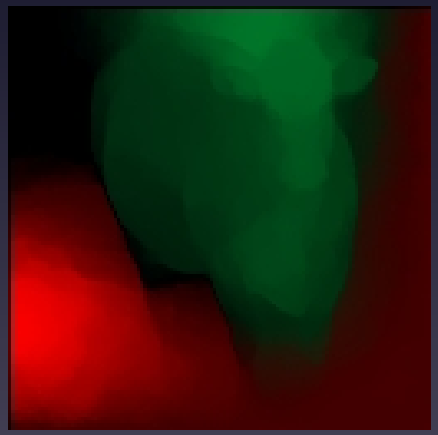
Model space of transformations, then use adversarial perturbation



Original Image



Raw Adversarial Perturbation



Restricted Perturbation

Recolorized Image

Notation

- Given a loss function $L(\boldsymbol{w}\,;\,\boldsymbol{z},\boldsymbol{l})$ that takes as its input the model parameters $\boldsymbol{w}$, input image $\boldsymbol{z}$, and label $\boldsymbol{l}$.
- Suppose components of $\boldsymbol{z}$ (i.e. pixels) are sampled from a continuous image flow $Z(x,y,t)$ at a specific t,

$$\boldsymbol{z} \triangleq [Z(x_1,y_1,t),\ldots,Z(x_n,y_m,t)]\,. \tag{1}$$

- Thus loss can be written as,

$$L(Z(x_1,y_1,t),\ldots,Z(x_n,y_m,t))\,, \tag{2}$$

($\boldsymbol{w}$ and $\boldsymbol{l}$ dropped for brevity).

Generating Adversarial Examples

- Seek a perturbation in the image that maximally increases the loss.

$$\frac{\partial}{\partial t} L = \langle \nabla_{\boldsymbol{z}} L(\boldsymbol{z}) \, , \, [\dot{Z}(x_1, y_1, t), \ldots, \dot{Z}(x_n, y_m, t)] \rangle \, . \quad (3)$$

- We will demonstrate a few examples to construct $\dot{Z}$.

Geometric

In a geometric transform, points may move, but cannot change color/brightness.

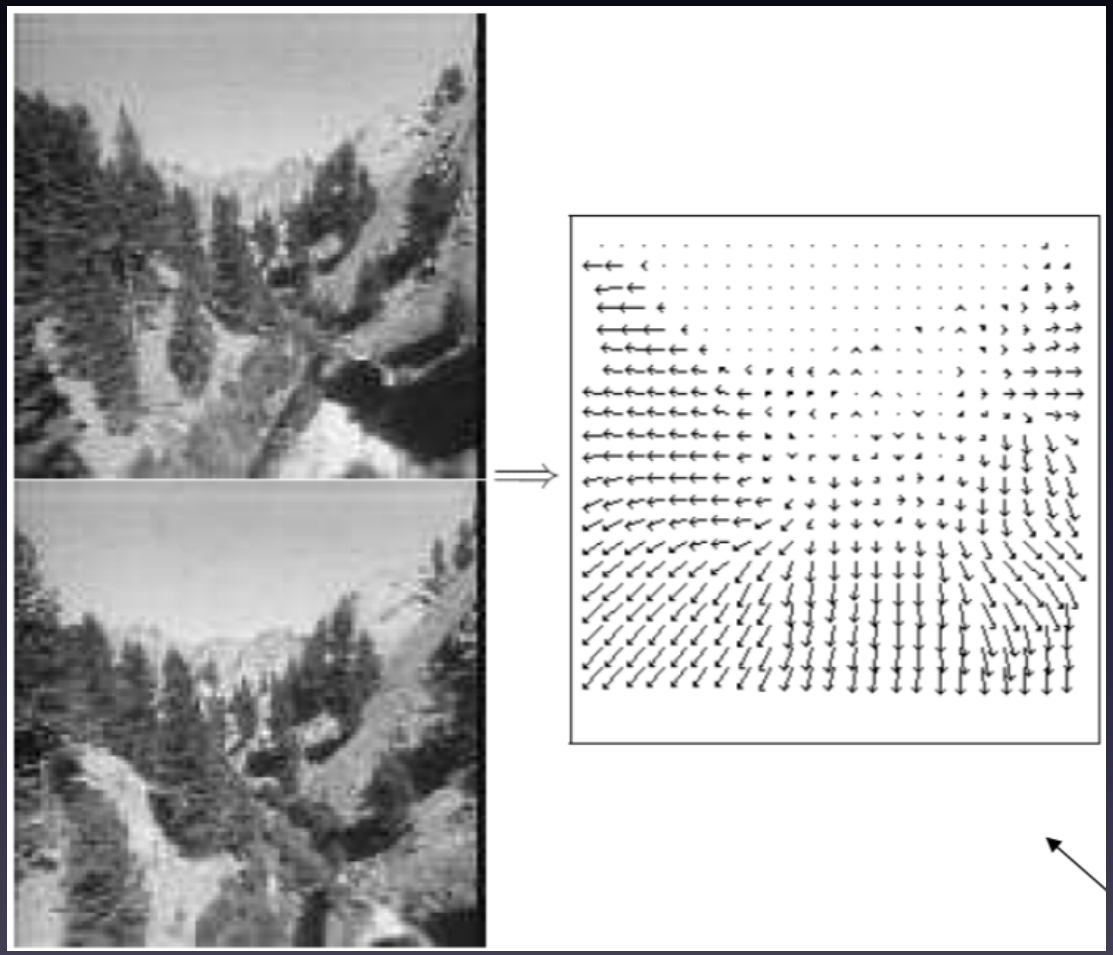


Image from Michael Black's PhD thesis

Geometric

Consider ***brightness constancy*** assumption,

$$\frac{d}{dt} Z(x(t), y(t), t) = 0\,, \tag{4}$$

which is equivalent to,

$$\begin{aligned} & Z_x(x(t), y(t), t)\dot{x}(t) && (5)\\ + \quad & Z_y(x(t), y(t), t)\dot{y}(t) && (6)\\ + \quad & \dot{Z}(x(t), y(t), t) && (7)\\ = \quad & 0\,. && (8) \end{aligned}$$

Applying this yields,

$$\begin{aligned} \frac{\partial}{\partial t} L = -\langle \nabla_{\boldsymbol{z}} L(\boldsymbol{z})\,, [\quad & Z_x(x_1, y_1, t)\dot{x}(x_1, y_1, t) \\ & + Z_y(x_1, y_1, t)\dot{y}(x_1, y_1, t) \\ , & \\ \vdots & \\ , \quad & Z_x(x_n, y_m, t)\dot{x}(x_n, y_m, t) \\ & + Z_y(x_n, y_m, t)\dot{y}(x_n, y_m, t)]\rangle\,. \end{aligned} \tag{9}$$

Geometric

We are interested in this dynamics at the reference point $t = 0$. For brevity, denote $Z_x(x_i, y_j, t = 0)$ by $Z_{x,i,j}$ and $\dot{x}((x_i, y_j), t = 0)$ by $\dot{x}_{i,j}$. Also denote each component of $\nabla_{\boldsymbol{x}} L(\boldsymbol{x})$ by $\ell_{i,j}$. Thus in matrix form,

$$\frac{\partial}{\partial t} L = - \begin{pmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_y \end{pmatrix}^T \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{pmatrix} \tag{10}$$

where,

$$N \triangleq n\,m \tag{11}$$

$$\boldsymbol{d}_{x\,N\times 1} \triangleq \begin{pmatrix} \ell_{1,1} Z_{x,1,1} & \dots & \ell_{m,n} Z_{x,m,n} \end{pmatrix} \tag{12}$$

$$\boldsymbol{d}_{y\,N\times 1} \triangleq \begin{pmatrix} \ell_{1,1} Z_{y,1,1} & \dots & \ell_{m,n} Z_{x,m,n} \end{pmatrix} \tag{13}$$

$$\dot{\boldsymbol{x}}_{N\times 1} \triangleq \begin{pmatrix} \dot{x}_{1,1} & \dots & \dot{x}_{m,n} \end{pmatrix} \tag{14}$$

$$\dot{\boldsymbol{y}}_{N\times 1} \triangleq \begin{pmatrix} \dot{y}_{1,1} & \dots & \dot{y}_{m,n} \end{pmatrix} \tag{15}$$

Geometric

We seek motion field ($\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{y}}$) that maximizes $\frac{\partial}{\partial t}L$. To keep velocity bounded, restrict $\|\dot{\boldsymbol{x}}\|$ and $\|\dot{\boldsymbol{y}}\|$. Also to have coherency in the field impose smoothness.

$$
\begin{aligned}
(\dot{\boldsymbol{x}}^*, \dot{\boldsymbol{y}}^*) = \arg\min_{(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})} \quad & \begin{pmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_y \end{pmatrix}^T \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{pmatrix} + \frac{1}{2}\alpha(\|\dot{\boldsymbol{x}}\|^2 + \|\dot{\boldsymbol{y}}\|^2) \\
+ \quad & \frac{1}{2}\beta(\|\boldsymbol{D}_x\dot{\boldsymbol{x}}\|^2 + \|\boldsymbol{D}_y\dot{\boldsymbol{x}}\|^2) \\
+ \quad & \frac{1}{2}\beta(\|\boldsymbol{D}_x\dot{\boldsymbol{y}}\|^2 + \|\boldsymbol{D}_y\dot{\boldsymbol{y}}\|^2)\,,
\end{aligned}
$$

where $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are differential operators along x and y direction, represented in matrix form. For brevity, let $\boldsymbol{P} \triangleq \boldsymbol{D}_x^T\boldsymbol{D}_x + \boldsymbol{D}_y^T\boldsymbol{D}_y$. We obtain,

$$
\begin{aligned}
(\dot{\boldsymbol{x}}^*, \dot{\boldsymbol{y}}^*) = \arg\min_{(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})} \quad & \begin{pmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_y \end{pmatrix}^T \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{pmatrix} && (16) \\
+ \quad & \frac{1}{2}\alpha(\dot{\boldsymbol{x}}^T\boldsymbol{I}\dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T\boldsymbol{I}\dot{\boldsymbol{y}}) && (17) \\
+ \quad & \frac{1}{2}\beta(\dot{\boldsymbol{x}}^T\boldsymbol{P}\dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T\boldsymbol{P}\dot{\boldsymbol{y}})\,. && (18)
\end{aligned}
$$

Geometric

Letting $\gamma \triangleq \frac{\beta}{\alpha}$, we can express

$$(\dot{\boldsymbol{x}}^*, \dot{\boldsymbol{y}}^*) = \arg\min_{(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})} \quad \begin{pmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_y \end{pmatrix}^T \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{pmatrix} \tag{19}$$

$$+ \quad \frac{1}{2}\alpha(\dot{\boldsymbol{x}}^T \boldsymbol{I} \dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T \boldsymbol{I} \dot{\boldsymbol{y}}) \tag{20}$$

$$+ \quad \frac{1}{2}\beta(\dot{\boldsymbol{x}}^T \boldsymbol{P} \dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T \boldsymbol{P} \dot{\boldsymbol{y}})\,. \tag{21}$$

as,

$$(\dot{\boldsymbol{x}}^*, \dot{\boldsymbol{y}}^*) = \arg\min_{(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})} \quad \begin{pmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_y \end{pmatrix}^T \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{pmatrix} \tag{22}$$

$$+ \quad \frac{1}{2}\alpha\big((\dot{\boldsymbol{x}}^T \boldsymbol{I} \dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T \boldsymbol{I} \dot{\boldsymbol{y}}) \tag{23}$$

$$+ \quad \gamma(\dot{\boldsymbol{x}}^T \boldsymbol{P} \dot{\boldsymbol{x}} + \dot{\boldsymbol{y}}^T \boldsymbol{P} \dot{\boldsymbol{y}})\big)\,. \tag{24}$$

This is a convex objective function in $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})$ and its minimizer can be obtained by zero crossing its gradient,

$$\boldsymbol{d}_x + \alpha(\boldsymbol{I} + \gamma \boldsymbol{P})\dot{\boldsymbol{x}} = \boldsymbol{0} \tag{25}$$

$$\boldsymbol{d}_y + \alpha(\boldsymbol{I} + \gamma \boldsymbol{P})\dot{\boldsymbol{y}} = \boldsymbol{0}\,. \tag{26}$$

Geometric

Therefore,

$$\dot{\boldsymbol{x}} = -\frac{1}{\alpha}(\boldsymbol{I} + \gamma \boldsymbol{P})^{-1}\boldsymbol{d}_x \tag{27}$$

$$\dot{\boldsymbol{y}} = -\frac{1}{\alpha}(\boldsymbol{I} + \gamma \boldsymbol{P})^{-1}\boldsymbol{d}_y\,. \tag{28}$$

Note that the matrix $(\boldsymbol{I} + \gamma \boldsymbol{P})^{-1}$ solely depends on regularizer, and ***does not depend on data*** (data being $\boldsymbol{d}_x$ and $\boldsymbol{d}_y$). Thus, one can compute $(\boldsymbol{I} + \gamma \boldsymbol{P})^{-1}$ ***offline*** given image dimensions and trade off parameter γ.
Once the motion field $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})$ is computed, it can be used to warp I to a new form via the motion field.

Example

Examples of ResNet trained on ImageNet:

Garbage truck. Confidence 60.95%

Minibus. Confidence 33.90%

Redbone. Confidence of 44.95

Boxer. Confidence 19.34%

The flow warped the back of the truck to be raised - a trait common in minibuses. Similarly, for the dog image, it warps the white cloth onto the dog's leg (redbones are normally entirely red, boxers are commonly red dogs that feature patches of white fur).

Photometric

Goal is to recolorize the image; preserve contours, object boundaries, edges, etc., but okay to change colors in the image. Consider some notion of edge strength, such as,

$$
\begin{aligned}
E(x,y,t) \quad &\triangleq \quad Z_{Ax}(x,y,t)^2 + Z_{Ay}(x,y,t)^2 && (29)\\
&\quad\; + Z_{Bx}(x,y,t)^2 + Z_{By}(x,y,t)^2 && (30)\\
&\quad\; + Z_{Cx}(x,y,t)^2 + Z_{Cy}(x,y,t)^2 \,. && (31)
\end{aligned}
$$

We penalize changes in edge strength anywhere that has already weak edges.

$$
\begin{aligned}
P(t) \quad &\triangleq \quad \int_{\mathcal{X}} \int_{\mathcal{Y}} \Big(\frac{1}{E(x,y,0)} \frac{\partial}{\partial t} E(x,y,t)\Big)^2 \, dx\, dy && (32)\\
\int_{\mathcal{X}} \int_{\mathcal{Y}} \Big(\frac{1}{E(x,y,0)} \Big(&\quad Z_{Ax}(x,y,t)\dot{Z}_{Ax} + Z_{Ay}(x,y,t)\dot{Z}_{Ay} && (33)\\
&+ \quad Z_{Bx}(x,y,t)\dot{Z}_{Bx} + Z_{By}(x,y,t)\dot{Z}_{By} && (34)\\
&+ \quad Z_{Cx}(x,y,t)\dot{Z}_{Cx} + Z_{Cy}(x,y,t)\dot{Z}_{Cy} && (35)\\
\Big)\Big)^2 \, dx\, dy \,. & && (36)
\end{aligned}
$$

Photometric

We are interested in $P(t)$ at $t = 0$ and thus drop dependency of the on t from the notation,

$$P = \int_{\mathcal{X}} \int_{\mathcal{Y}} \Big(\frac{1}{E(x,y)} \Big(\quad Z_{A\,x}(x,y)\dot{Z}_{A\,x} + Z_{A\,y}(x,y)\dot{Z}_{A\,y} \tag{37}$$

$$+ \quad Z_{B\,x}(x,y)\dot{Z}_{B\,x} + Z_{B\,y}(x,y)\dot{Z}_{B\,y} \tag{38}$$

$$+ \quad Z_{C\,x}(x,y)\dot{Z}_{C\,x} + Z_{C\,y}(x,y)\dot{Z}_{C\,y} \tag{39}$$

$$\Big) \Big)^2 \, dx \, dy \, . \tag{40}$$

This integral can be approximated numerically, e.g. sampling over ***evenly spaced*** grid of steps δ_x and δ_y,

$$P \approx \hat{P} \triangleq \delta_x \delta_y \| \boldsymbol{M}_A \dot{\boldsymbol{z}}_A + \boldsymbol{M}_B \dot{\boldsymbol{z}}_B + \boldsymbol{M}_C \dot{\boldsymbol{z}}_C \|^2 \, , \tag{41}$$

where,

$$\begin{aligned} \boldsymbol{M}_{\alpha N \times N} &\triangleq \operatorname{diag}(\boldsymbol{z}_{\alpha x} \div \boldsymbol{e}) \boldsymbol{D}_x + \operatorname{diag}(\boldsymbol{z}_{\alpha y} \div \boldsymbol{e}) \boldsymbol{D}_y \\ \boldsymbol{e}_{N \times 1} &\triangleq \| \, [\, \boldsymbol{z}_{Ax} \,|\, \boldsymbol{z}_{Ay} \,|\, \boldsymbol{z}_{Bx} \,|\, \boldsymbol{z}_{By} \,|\, \boldsymbol{z}_{Cx} \,|\, \boldsymbol{z}_{Cy} \,]_{N \times 6} \, \| \, , \end{aligned}$$

and $\div$ denotes element-wise division.

Photometric

Let $\boldsymbol{z} \triangleq [\boldsymbol{z}_A^T \,|\, \boldsymbol{z}_B^T \,|\, \boldsymbol{z}_C^T]^T$. The goal is to find $\dot{\boldsymbol{z}}$ that maximally increases the loss under this penalty.

$$\dot{\boldsymbol{z}}^*_{3N\times 1} \triangleq \arg\min_{\dot{\boldsymbol{z}}} \; -\langle \nabla_{\boldsymbol{z}} L(\boldsymbol{z}) \,,\, \dot{\boldsymbol{z}} \rangle \; + \; \frac{1}{2}\lambda \hat{P} \,. \tag{42}$$

This is a convex objective function in $\dot{\boldsymbol{z}}$ and its solution can be obtain by zero crossing the gradient,

$$-\nabla_{\boldsymbol{z}_A} L(\boldsymbol{z}) + \lambda \boldsymbol{M}_A^T (\boldsymbol{M}_A \dot{\boldsymbol{z}}_A + \boldsymbol{M}_B \dot{\boldsymbol{z}}_B + \boldsymbol{M}_C \dot{\boldsymbol{z}}_C) = \boldsymbol{0} \tag{43}$$

$$-\nabla_{\boldsymbol{z}_B} L(\boldsymbol{z}) + \lambda \boldsymbol{M}_B^T (\boldsymbol{M}_A \dot{\boldsymbol{z}}_A + \boldsymbol{M}_B \dot{\boldsymbol{z}}_B + \boldsymbol{M}_C \dot{\boldsymbol{z}}_C) = \boldsymbol{0} \tag{44}$$

$$-\nabla_{\boldsymbol{z}_C} L(\boldsymbol{z}) + \lambda \boldsymbol{M}_C^T (\boldsymbol{M}_A \dot{\boldsymbol{z}}_A + \boldsymbol{M}_B \dot{\boldsymbol{z}}_B + \boldsymbol{M}_C \dot{\boldsymbol{z}}_C) = \boldsymbol{0} \,. \tag{45}$$

This can be written more compactly as,

$$\boldsymbol{M}_{3N\times 3N} \dot{\boldsymbol{z}}_{3N\times 1} = \frac{1}{\lambda} \nabla_{\boldsymbol{z}} L(\boldsymbol{z}) \,, \tag{46}$$

where,

$$\boldsymbol{M}_{3N\times 3N} \triangleq \begin{pmatrix} \boldsymbol{M}_A^T \boldsymbol{M}_A & \boldsymbol{M}_A^T \boldsymbol{M}_B & \boldsymbol{M}_A^T \boldsymbol{M}_C \\ \boldsymbol{M}_B^T \boldsymbol{M}_A & \boldsymbol{M}_B^T \boldsymbol{M}_B & \boldsymbol{M}_B^T \boldsymbol{M}_C \\ \boldsymbol{M}_C^T \boldsymbol{M}_A & \boldsymbol{M}_C^T \boldsymbol{M}_B & \boldsymbol{M}_C^T \boldsymbol{M}_C \end{pmatrix} \,. \tag{47}$$

Photometric

Thus,

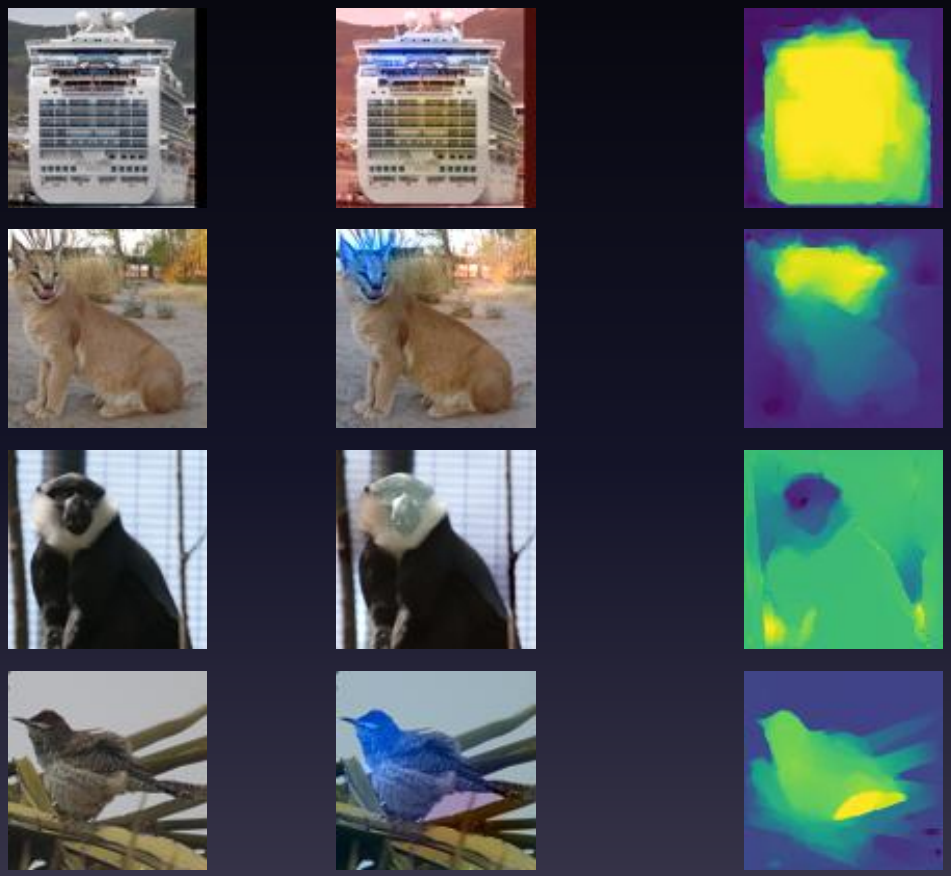
$$\dot{z}^*_{3N\times 1} = \frac{1}{\lambda} M^{-1}_{3N\times 3N} \left(\nabla_z L(z)\right)_{3N\times 1}, \tag{48}$$

While the matrix M depends on the data (images), it has nothing to do with the loss function L (thus independent of learning model). Thus for each image z, the associated matrix M^{-1} can be computed once. Then the computed M^{-1} can the be used for ***any learning model*** in the future.
Clearly M^{-1} captures some intrinsic properties of the data that are useful later for photometric transforms.

Photometric

- There are still challenges using this in practice: although M^{-1} can be computed once, applying such giant matrix to the gradient vector can be expensive.
- We use SVD to extract leading eigenvectors of M^{-1} and only use those to achieve an efficient approximation.

Photometric Examples



Original Recolorized Big Eigenvec.

Experiments

	CIFAR-10	CIFAR-100	STL-10
BasicAug	95.56%	77.41%	84.60%
AutoAug	95.70%	77.41%	84.46%
RandAug	94.95%	74.56%	88.75%
Flow + BasicAug	95.79%	76.82%	85.41%
Flow + AutoAug	95.77%	77.26%	85.39%
Flow + RandAug	93.22%	70.14%	87.96%
Recolor + BasicAug	95.80%	**77.45%**	86.11%
Recolor + AutoAug	**95.86%**	76.97%	85.21%
Recolor + RandAug	94.78%	74.18%	**88.84%**

Table: Reported accuracy of a Wide-ResNet-32-10 model trained from scratch using the listed augmentations

Experiments

	CIFAR-10	CIFAR-100	STL-10
No Augmentation	97.52%	85.68%	97.55%
AutoAugment	97.44%	84.09%	97.32%
RandAugment	96.39%	80.38%	97.14%
Flow	**97.60%**	**85.87%**	**97.61%**
Recolorization	97.34%	85.25%	**97.61%**

Table: Reported accuracy after finetuning on a EfficientNet-B0 checkpoint for 350 epochs.

Directions for Future Work

Most important directions for future Work

- Additional transformations that can be modeled in similar way.
- Learning the linear operators from another training set.

Thank You!