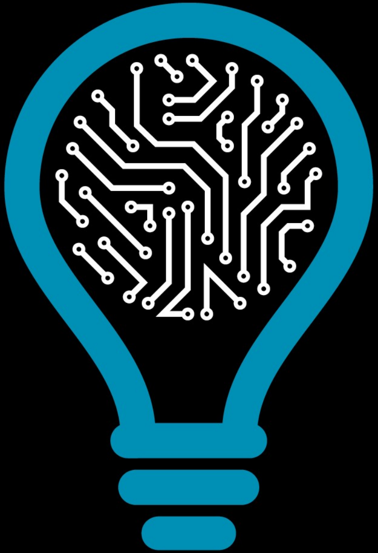
Background Invariance by Adversarial Learning

ICPR 2020, 10–15th January 2021

Ricardo Cruz [1,2]
Ricardo M. Prates [3,4]
Eduardo F. Simas Filho [4]
Joaquim F. Pinto Costa [2]
Jaime S. Cardoso [1,2]

[1] INESC TEC, [2] University of Porto
[3] Federal University of São Francisco Valley
[4] Federal University of Bahia



Motivation

Goal: classify electrical insulators.

Problem: training set background (indoors) $\neq$ testing set background (outdoors).

$y = 1$

$y = 2$

$y = 3$

$y = 4$

Training set (indoors)

$y = 1$

$y = 2$

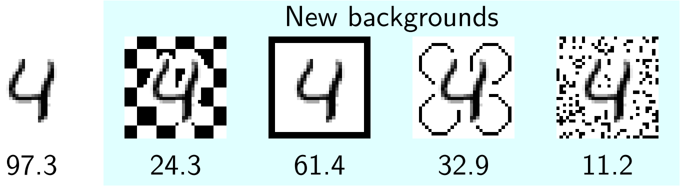
$y = 3$

$y = 4$

Testing set (outdoors)

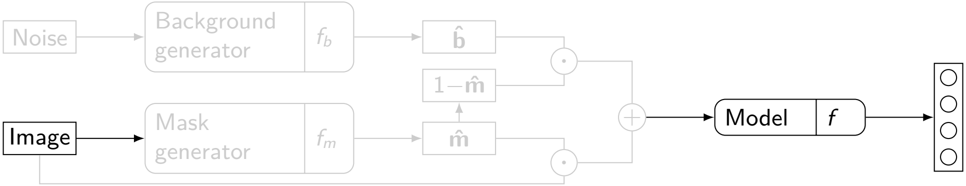
Motivation

Unfortunately, a model with accuracy of **97.3%** can drop to as low as **11%** (random) just by changing the background.



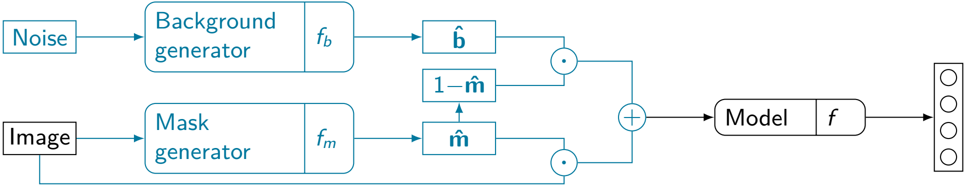
- A good CNN can drop precipitously, when the object background is changed.
- Notice how a model with accuracy of **97.3%** can drop to as low as **11%** (random) just by changing the background.

Proposed Method: Overview



Traditional model.

Proposed Method: Overview



Proposed method.

- **Background generator:** creates new backgrounds.
- **Mask generator:** helps injecting the new backgrounds.

Proposed Method: Details

$$\boxed{\min_{f,f_m} \max_{f_b} \mathcal{L}oss}$$

Step 1. **Model $\boldsymbol{f}$** is optimized to minimize a loss $\mathcal{L}(y, f(x))$ using an image x as input with label y as the ground-truth.

Step 2.

Mask generator $\boldsymbol{f_m}$ is trained to produce a mask $\hat{m}$ ($\hat{m} \in [0, 1]$).

- U-Net is used as the architecture.
- The image is segmented through a element-wise product, $x' = x \odot \hat{m}$.

Step 3. **Background generator $\boldsymbol{f_b}$** transforms noise z into a background image $\hat{b}$.

- Model f is also optimized during Step 3.
- In the case of monochrome images, a constrain term is added to disallow the background from filling over half the pixels.

Proposed Method: Details

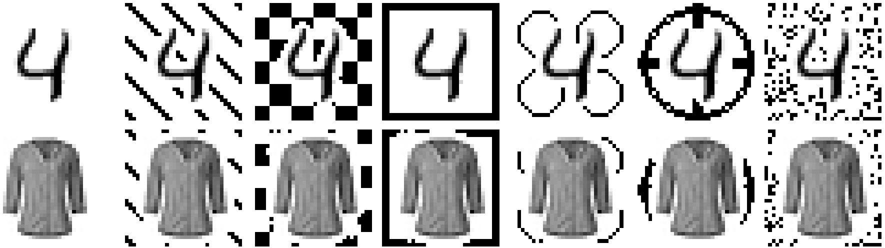
- The goal is to (during training) be able to place the object in a multitude of contexts (backgrounds)
- Try to find backgrounds that “fool” the model, thus making it robust.

$t = 1$ $t = 10$ $t = 20$ $t = 30$ $t = 40$

Examples of the dynamic background along the epochs.

Experiments

- Datasets: MNIST (10 digits) and Fashion-MNIST (10 pieces of clothe).
  - (artificially enhanced by introducing backgrounds as illustrated in the figure)

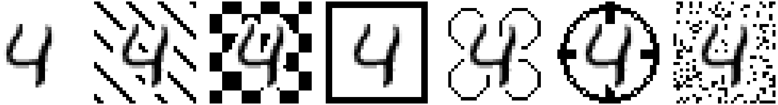


(a) Original (b) Stripes (c) Board (d) Border (e) Circles (f) Clock (g) Random

Backgrounds introduced for MNIST and Fashion-MNIST.

Results: Accuracy (%)

MNIST



Baseline	97.3	38.0	24.3	61.4	32.9	19.7	11.2
Attention	93.4	28.1	26.8	57.3	40.1	29.3	25.1
Proposal	94.9	92.3	76.8	93.1	93.7	70.8	86.2
Fashion-MNIST							
Baseline	90.1	21.3	24.6	36.9	28.5	29.6	16.8
Attention	81.2	18.2	20.1	51.8	26.0	31.8	36.2
Proposal	70.7	62.9	61.5	66.5	60.9	60.8	45.9

- Interestingly, the attention mechanism results only negligibly improve on the baseline classifier.
  - This mechanism works by cropping the image and, not surprisingly, it was found to perform best in the border case (with over 50% accuracy).
- The proposed method is resilient to a wide range of testing backgrounds.

Results for the Case Study

Results for Drone Case Study

Method	Validation Accuracy (%)
Baseline	71.9
Attention	45.8
Proposal	88.7

Impact of background generator

Background	Validation Accuracy (%)
Black	76.3
Noise	59.6
Proposal	88.7
Real backgrounds	93.8

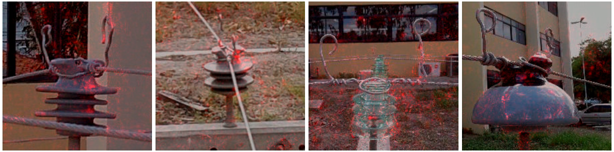
Insulator

Activations

Otsu threshold

Sensitivity analysis – baseline

Sensitivity analysis – proposal



Conclusion

1. Sometimes it is easier to collect data inside a studio rather than in the real world – for example when training a drone.
2. Unfortunately, convolutional neural networks' performance degrades terribly when used in new backgrounds.
3. An adversarially trained model is proposed where the model tries to ***minimize*** its loss while a generator injecting new backgrounds to ***maximize*** the loss.
4. The proposed method was evaluated for the task of classification, but it could potentially be used for other tasks:
   - regression problems
   - segmentation
   - reinforcement learning tasks.

Final Thoughts

Typically adversarial training has been used to improve generators that produce content for the user. But at INESC TEC, we have been using adversarial training to improve the classifier itself. If you like the idea, you may also want to consult:

1. Pedro M. Ferreira, Diogo Pernes, Ana Rebelo, Jaime S. Cardoso. **"Learning signer-invariant representations with adversarial training."** 20th International Conference on Machine Vision, 2020.
2. João Pereira, Ana F. Sequeira, Diogo Pernes, and Jaime S. Cardoso. **"A robust fingerprint presentation attack detection method against unseen attacks through adversarial learning."** 19th BIOSIG, 2020.

Thank you!

Ricardo Cruz <**ricardo.p.cruz@inesctec.pt**>
Ricardo M. Prates, Eduardo F. Simas Filho
Joaquim F. Pinto Costa, Jaime S. Cardoso