Background Invariant Classification by Reducing Texture Bias in CNNs

Maliha Arif
maliha.arif@knights.ucf.edu

Calvin Yong
calvinyong@knights.ucf.edu

Abhijit Mahalanobis
amahalan@crcv.ucf.edu

Introduction

Deep convolutional neural networks (CNNs) are known to yield superior performance when large amount of training data is used to train them. On a variety of computer vision tasks, they may also not learn the object shape but become biased by its background. We evaluate the performance of deep neural networks and simple CNNs using our proposed "**Split training method**" which assists in removing texture bias and perform background invariant classification on both Infrared and RGB data.

Infra-red Dataset

Gendata : is a synthetic IR dataset with 3 military vehicles , APC, tank and truck.

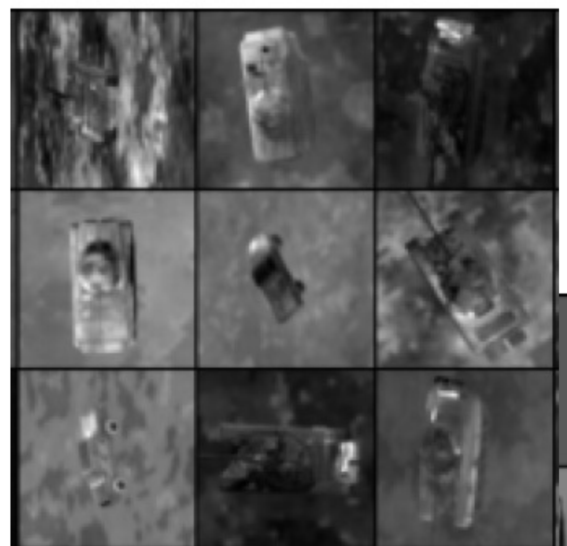

Fig.1. Examples of Unmasked images (with background)

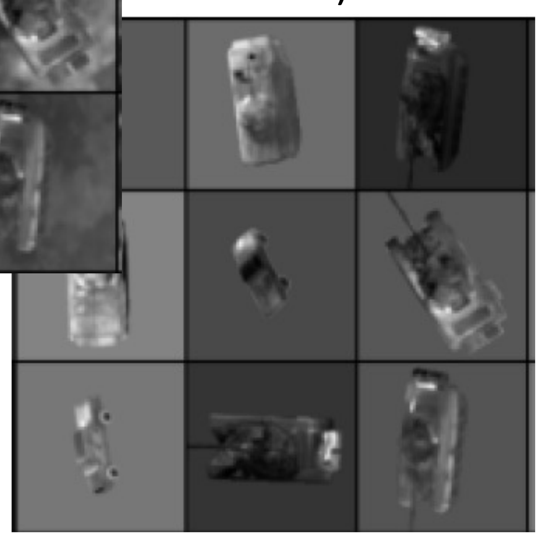

Fig.2. Examples of masked images (with background removed)

Method

Algorithm 1 Split training method

Require: $i \leftarrow$ Training Images
for $i \; \epsilon \; \{ 1...n \}$ **do**
  $mean_{\leftarrow} i[mask! = 0].mean()$ ▷ For IR images
  i -= mean ▷ For IR images
  normalize i
  if masked **then**
    $i[mask == 0] = 0$
  end if
end for
$m1 \leftarrow$ Train primary model on masked images $i1$
$m2 \leftarrow$ Train secondary model on Unmasked images $i2$
$k \leftarrow$ Last feature layer of network
for $layer \; \epsilon \; \{ 1...k \}$ **do**
  Optimize $layer$
  loss = MSE $[m1(i1) - m2(i2)]$ ▷ For matching activations
  use lr=1e-3
end for
for $layer \; \epsilon \; \{ 1...k+1...n \}$ **do**
  Load weights $(m2)$ and fine-tune layer $\epsilon \; \{1...k\}$
  Optimize $layer$
  loss = C.E
  use lr=1e-4
end for

Fig.3. Proposed algorithm for our background invariant approach

Our split training method comprises of 3 main steps:

1) Train primary model- *m*1 (a simple CNN) using cross-entropy loss on masked images (no background).
2) For the secondary model-*m*2 (identical to primary) , using unmasked images, train the layers from the input layer up to the last feature layer by minimizing the mean squared error (MSE) between the activations of the primary model, and the secondary model at the last feature layer.
3) Fine-tune the trained layers of the primary model. Using unmasked images, train the remaining layers to the end of the network using cross-entropy loss.

RGB Dataset

MS-COCO : is a main-stream computer vision dataset for object detection and segmentation. We use it in a data efficient manner for classification only.

Fig.4. Examples of Unmasked images (with complex scenery and background)

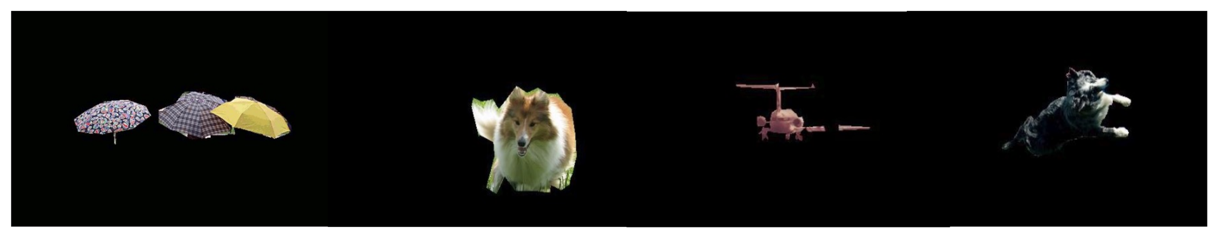

Fig.5. Examples of masked images (with background removed)

Table 1. Illustrates test accuracy on Gendata test set.

Architecture	Standard Training	Ours (last feature layer)
Simple	75.162% (5.576%)	91.664% (2.435%)
Mobilenet	73.580% (8.243%)	74.319% (5.536%)
VGG11	72.798% (13.000%)	89.355% (3.498%)
Densenet	66.597% (7.741%)	85.388% (2.604%)

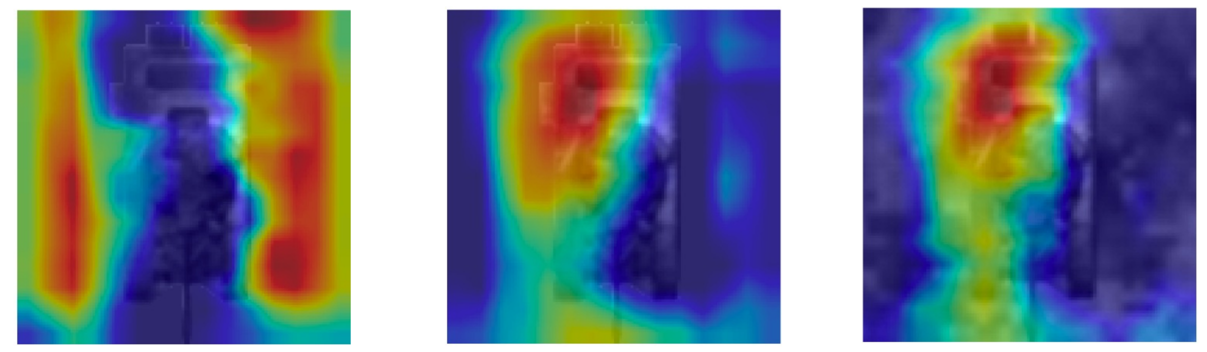

Fig.6. Comparison of Grad-CAM output when using a simple CNN and training with images having background, no background, and having background (using our split-training method) on Gendata. Network learns to ignore background and texture bias.

Experiment	Standard Training	Ours (last feature layer)
3 class only*	80.4 %	86.3 %
10 classes*	58.01 %	65.6 %

** Classes are chosen randomly; purpose is to show the method extends to RGB data well*

Table 2. Illustrates test accuracy on MS-COCO test set.

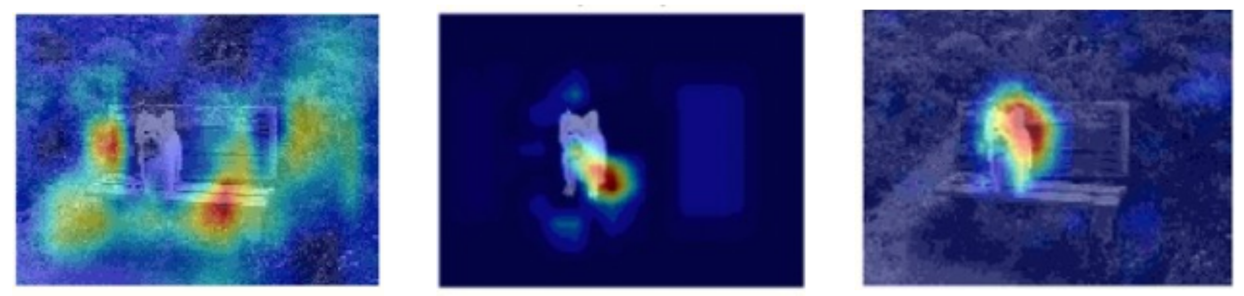

Fig.7. Comparison of Grad-CAM output when using a simple CNN and training with images having background, no background, and having background (using our split-training method) on MS-COCO. CNN is focused on object before making prediction.

The authors gratefully acknowledge the support of CCDC-US ARMY for this research.