BACKGROUND-TOLERANT OBJECT CLASSIFICATION WITH EMBEDDED SEGMENTATION MASK FOR INFRARED AND COLOR IMAGERY

Maliha Arif,1,2 Calvin Yong,2 Abhijit Mahalanobis,1,2 and Nazanin Rahnavaard 2

1Center for Research in Computer Vision (CRCV), 2University of Central Florida

ABSTRACT

Even though convolutional neural networks (CNNs) can classify objects in images very accurately, it is well known that the attention of the network may not always be on the semantically important regions of the scene. It has been observed that networks often learn background textures, which are not relevant to the object of interest. In turn this makes the networks susceptible to variations and changes in the background which may negatively affect their performance.

We propose a new three-step training procedure called split training to reduce this bias in CNNs for object recognition using Infrared imagery and Color (RGB) data. Our split training procedure has three steps. First, a baseline model is trained to recognize objects in images without background, and the activations produced by the higher layers are observed. Next, a second network is trained using Mean Square Error (MSE) loss to produce the same activations, but in response to the objects embedded in background. This forces the second network to ignore the background while focusing on the object of interest. Finally, with layers producing the activations frozen, the rest of the second network is trained using cross-entropy loss to classify the objects in images with background. Our training method outperforms the traditional training procedure in both a simple CNN architecture, as well as for deep CNNs like VGG and DenseNet, and learns to mimic human vision which focuses more on shape and structure than background with higher accuracy.

Index Terms— infrared imagery, background invariant learning, grad-CAM, split training, MS-COCO

1. INTRODUCTION

It is well known that deep learning yields excellent results when trained and tested on large datasets of images for object recognition. However, very little effort is directed at understanding the inner knowledge representation of the network as long as high accuracy is achieved. It has been observed that training with cross-entropy loss does not always teach CNNs to learn the actual object when making a prediction for an image. In fact, Grad-CAM [1] visualizations have shown that a classifier’s attention can include background textures and semantically unimportant parts of the scene. Different studies have tried to address the background bias problem. A very recent work by [2] explicitly studies the effects of background on the classification performance of deep CNNs. Their focus is on understanding the influence of the background (observed when the background is changed or the foreground is masked) and studying its impact on classification. They show, using a series of experiments on deep CNNs (such as ResNet-18) and diverse datasets (including ImageNet), that even though the foreground is masked (or removed), the network can still correctly identify the class. Similarly, when the background is switched and foreground remains intact, test accuracy decreases. On the VOC12 dataset [3], test accuracy decreases from 75% to 46% when different backgrounds are introduced and object of interest or foreground is kept the same. This raises questions about the current training paradigms, and what CNNs actually learn for making predictions. To answer these questions, we propose a novel training strategy called ‘Split Training’ which aims to teach the model to focus on the object (and ignore the background) by matching its activations at some specified layer to that of another model that was trained on masked (background removed) images and consequently produces ideal activations as shown in Figure 1. Our method outperforms the standard training and transfer learning methods, and produces activation maps that

We gratefully acknowledge the support of CCDC
show the object is used to make predictions rather than texture and background. The study in [4] proposes a dual path network (DPN) which outperforms both ResNet and DenseNet deep CNNs. In particular, a shallow DPN with 26% smaller model size surpasses ResNet-101 performance on the ImageNet dataset. This shows that a deep network still might not learn the best representations of the data. In [5], authors study that standard image classification models rely too much on signals from background and ignore foreground. Background correlations are largely predictive and influence model decisions to a great extent. Models often misclassify images even when correct foregrounds are present, up to 87.5% of the time when random adversarial backgrounds are chosen. A recent work by Robert et al. [6] indicates how CNNs are biased by texture. The authors demonstrate texture bias using all major deep CNN models and experiment with stylized versions of the ImageNet dataset, similar to the way done in [5]. They create grayscale versions of objects which contain both shape and texture, silhouette images where object outline is filled with black color, images with only edges and lastly images with contrasting shapes and textures, for example a cat with texture of an elephant. All popular deep CNNs for image classification are shown to fail to recognize shape and do better with texture, VGG-16 developed by Visual Geometry Group gives 17.2% accuracy using shape vs. 82.8% when using including texture; GoogleNet gives 31.2% accuracy using shape vs. 68.8% when using only texture and ResNet-50 gives 22.1% accuracy using shapes as compared to 77.9% when using texture. Human observers do much better as expected with 95.9% accuracy when shapes are clearly defined. These results motivate our proposed training method which aims to teach a network to focus more on object shape rather than background and texture. Lastly, deep networks (such as VGG-16, ResNet, and Densenet) which have been pre-trained on ImageNet, an RGB dataset, fail to work well with infrared images due to different domain types [7]. Nor does their performance improve significantly when transfer learning is used to fine-tune such networks on infrared imagery.

2. SPLIT TRAINING METHOD

We assume that the training dataset provides a binary mask (or segmentation boundary) for the object of interest in the training images. MS-COCO is an example of such a dataset as there are others used for semantic segmentation. Our goal is to indirectly embed the effect of the mask, so that it can implicitly distinguish the object from the background. Our introduced split training method comprises of three main steps which are as follows:

1. First training a primary model- m1 (a simple CNN) using cross entropy loss on masked (i.e. background removed) images.

2. Next training a secondary model- m2 (architecturally identical to the primary model) , using the unmasked (i.e. background present) images such that the MSE between the activations of the higher convolutional layers of the secondary and primary networks is minimized.

3. Lastly, freezing the trained layers of the secondary model (trained on unmasked images) and train the remaining layers to the end of the network using cross entropy (CE) loss.

By design, the primary network learns object information alone (since there is no background). Forcing the activations of the primary and secondary network to be similar encourages the latter to also respond to the object while ignoring the background. The proposed approach is also shown in Algorithm 1. Note that two versions of each training image is needed, i.e. one of just the object with the background masked out (say version D1), and the unmasked version with both the object and the background (version D2). The first network learns to classify the object using D1 and provides idealized activations at the output of the higher convolutional layers. Then the secondary model is trained using D2 to match the activations of the primary model trained on D1. Then, with the trained convolutional layers frozen, the fully connected layers of the secondary network are trained to classify samples from D2. Since large amount of data is not required for training, computational cost is greatly reduced.

Algorithm 1 Split Training Method

Require: $i \leftarrow$ Training Images

1: for $i \in \{1...n\}$ do
2: \hspace{1em} $\text{mean} = \{i[\text{mask} == 0]\}$ \{For IR images\}
3: \hspace{1em} $i = \text{mean}$ \{For IR images\}
4: \hspace{1em} normalize $i$
5: \hspace{1em} if masked then
6: \hspace{2em} $i[\text{mask} == 0] = 0$
7: \hspace{1em} end if
8: \hspace{1em} end for
9: $m1 \leftarrow$ Train primary model on masked images $i$
10: $m2 \leftarrow$ Train secondary model on Unmasked images $i$
11: $k \leftarrow$ Last/Intermediate feature layer of network
12: $n \leftarrow$ Total number of layers in the network
13: for layer $\in \{1...k\}$ do
14: \hspace{1em} Optimize layer
15: \hspace{2em} loss $= \text{MSE} \{m1(i1) - m2(i2)\}$ \{For matching activations\}
16: \hspace{2em} use $lr=1e-3$ \{lr = learning rate\}
17: \hspace{1em} end for
18: for layer $\in \{1...k + 1...n\}$ do
19: \hspace{1em} Load weights ($m2$) and freeze layer $\in \{1...k\}$
20: \hspace{1em} Optimize layer
21: \hspace{2em} loss $= \text{CE} \{\text{CE = Cross Entropy}\}$
22: \hspace{2em} use $lr=1e-4$
23: \hspace{1em} end for
3. DATASETS

**Synthetic IR data** - (Infrared Imagery): A dataset of synthetically generated infrared images was available for the first set of experiments. It contains three classes/targets: APC, tank, and truck. Each of the targets are in 18 different backgrounds and thermal signatures, and viewed at azimuth angles between 0° and 359° in increments of 1°, and elevation angles of 15°, 30°, 45°, 60°, 75°, and 90°. This gives a total of 38,880 images per class, and thus a total of 116,640 images for the entire dataset. Examples of targets in background are shown in the leftmost three columns in Figure 2 while their masked versions (i.e. background removed) are shown in three columns to the right. The masked images will be used to train our primary model, while those including the background will be used to train the secondary model. The data uses 16 bits with pixel value range from about 400 to 4000. For preprocessing, we centered each image by subtracting the mean value of the target from each pixel and then scaled the image to unit variance. We believe this scaling helps the network to converge by bringing the large range of values into a normalized scale. To generate the images with no background, we take the preprocessed dataset with background, and set the background pixels to zero. Since the target is also zero-mean, background in these images appears to be a different gray level due to the display scaling. This also ensures that the target values are the same between the dataset with background and the dataset without background.

**MS-COCO (RGB data):** To show the validity of the proposed method on RGB data, experiments are also performed using the MS-COCO dataset, a well known main-stream computer vision dataset for object detection, segmentation, and classification. Figure 3 shows examples of unmasked and masked images which were used in our experiments. MS-COCO has 91 categories in all. Of these, 81 have segmentation masks for the foreground object. We make use of 3 of these classes. These classes include airplane, dog, and elephant. In order to avoid confusion between classes, we select training images which have only one object. However, we do end up with few images having multiple instances of the same class (and multiple classes). We use a total of 6k training images (2k per class) and 300 test images. To create the set of masked images, i.e. with background removed, we use Pycoco [8] library and the ground truth annotations provided with the dataset. We mask the background by replacing the corresponding pixel values by zero. MS-COCO is a complex dataset with much of the images containing a lot of background, as a result of which the object of interest appears to be very small and not in the center. This becomes a good challenging point for our proposed method. The RGB pixel values range from 0 to 255 unlike infrared images. We perform our experiments with and without mean target subtraction and find out that without mean target subtraction, our model performs better since the input pixel range is already small. Therefore we normalize our dataset by simply scaling the images to unit variance without mean subtraction.

4. EXPERIMENTS

4.1. Synthetic IR data

We benchmarked our method on 4 different models: a simple CNN, which is a feedforward network with 4 Conv-BatchNorm-ReLU-MaxPool blocks, Mobilenet [9], VGG [10], and DenseNet [11]. We used these deep models as they are known to work well for object classification on color data. To obtain a baseline performance for comparison, we trained these networks from scratch on the unmasked images. We also explored transfer learning where the network was trained from scratch using masked images, and then fine-tuned on the unmasked images. Table 1 shows the mean and standard deviation for the test accuracy of each model architecture trained with various methods. Our method outperforms the standard (conventional) training and finetuning methods, with performance gains depending on the choice of feature layer to match with the secondary model’s activations. With MobileNet, if we perform the MSE loss on the last feature layer, we see that our method does as well as the standard training method, yielding about 74% accuracy. We believe this is because the output spatial dimension at that layer (1280 channels each with size 4 x 4) is very small, and the model cannot benefit substantially from the corresponding activation information. When we use a layer closer to the input, we get a significant increase in performance, yielding a validation
Table 1. Test accuracies for different architectures and training methods on synthetic IR (infrared) data. Numbers in parentheses are the standard deviation of the accuracies over 10 runs.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Standard Training</th>
<th>Finetuning / Transfer learning</th>
<th>Ours (last feature layer)</th>
<th>Ours (intermediate layer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple CNN</td>
<td>75.162% (5.576%)</td>
<td>69.160% (3.050%)</td>
<td><strong>91.664% (2.435%)</strong></td>
<td>87.222% (6.648%)</td>
</tr>
<tr>
<td>MobileNet</td>
<td>73.580% (8.243%)</td>
<td>34.191% (6.161%)</td>
<td><strong>74.319% (5.536%)</strong></td>
<td><strong>93.226% (2.118%)</strong></td>
</tr>
<tr>
<td>VGG11</td>
<td>72.798% (13.000%)</td>
<td>48.224% (7.565%)</td>
<td><strong>89.355% (3.498%)</strong></td>
<td>83.537% (7.338%)</td>
</tr>
<tr>
<td>DenseNet</td>
<td>66.597% (7.741%)</td>
<td>50.098% (4.361%)</td>
<td><strong>85.388% (2.604%)</strong></td>
<td><strong>88.557% (3.712%)</strong></td>
</tr>
</tbody>
</table>

Table 2. MobileNet Performance for varying activation sizes.

<table>
<thead>
<tr>
<th>Activation Size</th>
<th>Validation Accuracy</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(24, 30, 30)</td>
<td>88.907%</td>
<td>3.282%</td>
</tr>
<tr>
<td>(32, 15, 15)</td>
<td>94.417%</td>
<td>2.110%</td>
</tr>
<tr>
<td>(64, 8, 8)</td>
<td>96.318%</td>
<td>1.165%</td>
</tr>
<tr>
<td>(96, 8, 8)</td>
<td>92.503%</td>
<td>2.638%</td>
</tr>
<tr>
<td>(1280, 4, 4)</td>
<td>93.226%</td>
<td>2.118%</td>
</tr>
</tbody>
</table>

accuracy of 93%. However, this is not always the case as the results for VGG show. To study this effect, we minimized the activation MSE at different layers for MobileNet and quantified the performance as shown in Table 2. The activation sizes are denoted by (c;h;w), where c is the number of channels, h is the height, and w is the width. The table shows that the intermediate feature layer with an output size of (64, 8, 8) yields the highest accuracy. It is therefore necessary to experimentally determine the best layer for use in Step 2 of our split training method.

4.2. MS-COCO

On MS-COCO (RGB data), we illustrate the method for a 3-class problem using 6,000 training images and 300 test images. We use the simple CNN previously employed for IR images, as well as other SOTA architectures. We first train the primary versions of the networks using masked images to obtain ideal activations which focus only on the objects. We use a batch size of 32 and the Adam optimizer with a learning rate of 1e-3 and 1e-4 for the first 40 epochs and the last 20 epochs, respectively. Since images in the dataset are of varying sizes, we resize all of them to 160 x 120 before training. The convolutional blocks of the secondary network are trained such that MSE loss at the output of the last feature layer is minimized. For best results, we trained it for 30 epochs using the Adam optimizer and learning rate of 1e-3. Thereafter, we trained the classifier layers using a lower learning rate of 1e-4 for 30 more epochs. We compare the results of our method to the performance of the network when it is conventionally trained using unmasked images, keeping all other hyperparameters the same. As shown in Table 3, our method’s performance is substantially better than standard technique on unmasked images except for MobileNet where the choice of feature layer is not the best as explained earlier. The Grad-CAM visualizations in Figure 4 show that the network’s attention using our method (third row) is on the object of interest, whereas that using the conventional method (second row) is scattered across the image.

5. CONCLUSION

We proposed a new method, referred to as ‘Split training’, for reducing background bias in classifier networks by teaching them to implicitly distinguish between background and the object of interest. The network first learns the object by masking out the background. The resulting activations are then used to recognize the same object in the presence of background. Our method outperforms the conventional training methods for both Infrared images and Color(RGB) data, yielding better classification accuracies on objects in different backgrounds. We also illustrated the attention improvement in CNNs using Grad-CAM visualizations.
6. REFERENCES


