2019

No Room for Squares: Using Bitmap Masks to Improve Pedestrian Detection Using CNNS.

Adam Warde
Trinity College Dublin, Ireland

Hamza Yous
Trinity College Dublin

David Gregg
Trinity College Dublin

David Moloney
Trinity College Dublin

Recommended Citation

Warde, A., Yous, H., Gregg, D. & Moloney, D. (2019). No room for squares: using bitmap masks to improve pedestrian detection using CNNs. *IMVIP 2019: Irish Machine Vision & Image Processing*, Technological University Dublin, Dublin, Ireland, August 28-30. doi: 10.21427/9vge-x813

No Room for Squares[*]: Using Bitmap Masks to Improve Pedestrian Detection Using CNNs

Adam Warde, Hamza Yous, David Gregg, David Moloney

Intel-Movidius, Trinity College Dublin

31/05/2019

Abstract

In this paper we investigate a method to reduce the number of computations and associated activations in Convolutional Neural Networks (CNN) by using bitmaps. The bitmaps are used to mask the input images to the network that fall within a rectangular window but do not fall within the boundaries of the objects the network is being trained upon. The mask has the effect of rendering the operations on these portions of the training images trivial. The thesis is that applying this approach to CNNs will not degrade accuracy while at the same time reducing the computational workload and reducing memory footprint. We found that we can remove up to 60% of the input images and see no decrease in accuracy. This leads to activation sparsity that can be exploited using a hardware accelerator to speedup training and inference, and decrease energy consumed.

Keywords: CNN, bitmap, trivial arithmetic, pedestrian detection, sparsity.

1 Introduction

Objects in nature are rarely rectangular in shape, yet all of the incumbent methods to recognise objects in digital imagery rely on the training and subsequent detection of rectangular patterns. While this is obviously convenient, because only the origin of the rectangular template need be described, it differs from the way patterns are detected in biological systems which are much more sparing in their use of energy [Yong, 2013]. The principle of biological efficiency reaches a high point in the human brain, which contains 80B neurons and trillions of synapses operating off a miserly 20W derived from renewable plant sugars [Herculano-Houzel, 2009]. This work takes its cue from biological systems and attempts to derive masks for computations that can achieve high performance at computational costs more similar to those found in nature.

In recent years many computer vision and deep learning applications have moved away from the cloud to performing inference on the edge to reduce the latency and network bandwidth requirements. With this move however comes a new set of problems such as memory and compute/computational time/resources constraints. Pedestrian classification and detection is one of the applications that has moved to the edge. One popular technique for doing these types of classification problems are CNNs [LeCun et al., 1998]. CNNs however are very computational expensive, and compute/computational time/resources and memory are generally at a premium on the edge.

One of the current approaches to reducing the number of computations in a CNN is to prune the network weights after training is complete, which makes the network sparse. Pruning involves removing weights, activations or entire features from the network, so that the pruned features no longer need to be loaded from memory or used in computations. In general this pruning is performed without any regard to the actual relevance of these weights to computing the network output. While useful this approach is far from desirable.

In this paper we propose to reduce the number of computations and associated activations by using bitmaps to mask computations that fall within a rectangular window, but do not fall within the boundaries of the objects the network is being trained upon. The mask has the effect of rendering the operations on these portions of the training images trivial.

[*]`https://en.wikipedia.org/wiki/No_Room_for_Squares`

This work takes inspiration from a bitmap masking technique first introduced in conjunction with HoG (Histogram of oriented Gradients)[Dalal and Triggs, 2005a]. The method uses a bitmap mask, derived from the average of a given training dataset, as a mask for subsequent computations, which allows many of the background pixels to be set to zero, which in turn renders them irrelevant to training and subsequent inference [Dehghani et al., 2016]. In the case of HoG/SVM (Support Vector Machines) the computational workload was reduced by 75% while simultaneously increasing precision and recall.

In this work we extend a similar technique for use with CNNs. We propose using a simple thresholding method initially and then dilation to derive the average position of the pedestrians in our the INRIA [Dalal and Triggs, 2005b] and Daimler [Munder and Gavrila., 2006] pedestrian datasets, and generate new datasets of masked images for training networks.

In section 2 Related Work we will discuss some of the existing techniques that try to address similar problems. In the 3 Method section we explain our proposed method and how in was implemented. This is followed by our 4 Results section that details our findings.

2 Related Work

While computer vision algorithms are commonly applied on the entire input image, many works attempt to remove some parts of images during the processing, by using a bitmap mask to improve their efficiency. For instance, Dehgani et al. [Dehghani et al., 2016] mask their input dataset before using HoG/SVM to classify pedestrians. In addition, bitmap masks are used in [DeVries and Taylor, 2017] to make models more robust to adversarial noise.

2.1 Bitmap-HOG (BHOG)

In [Dehghani et al., 2016], the authors propose an improved HOG pedestrian detection. It consists of using a bitmap mask to improve the accuracy and reduce the computation cost of pedestrian detection. A mask is created by the averaging of all the images in a pedestrian dataset. This mask is applied to the existing dataset to generate a new training dataset. The resulting dataset is then used to train a HOG/SVM classifier more efficiently. Their experimental results demonstrate that the proposed algorithm decreases the workload associated with HOG/SVM classifiers by 75% compared to the state of the art. Furthermore, it increases the recall by 5% with only 2% decrease in precision compared to the standard HOG-based technique.

In this paper, we propose to extend the concept of using a bitmap to improve the CNN-based pedestrian detection. The use of the bitmap mask can reduce the effect of the noisy negative pixels during the training and inference.

2.2 Cutout Regularisation

DeVries et al. [DeVries and Taylor, 2017] seek to improve robustness of CNN models against noise by using a technique called cutout regularisation, that applies a rectangular mask to the input data randomly by occluding some parts of the image.

The use of this technique improves the model generalisation by forcing the network to learn new features. However, this method does not attempt to remove the background noise in targeted fashion.

Our proposed approach is based on using the mask to occlude the noisy pixels in a selective fashion. In fact, the mask is obtained from the training datasets to reduce the contribution of the pixels in the background in the model.

3 Proposed Method

Traditionally, when CNNs are used for pedestrian classification and detection, the entirety of each input image is used to classify the image, i.e. the CNN infers over the entire input image whether it contains a pedestrian or not. The problem with using the entirety of each of the images in the dataset is that many of the pixels in the input are background noise and not necessary for classification or detection. These are therefore unnecessary calculations that consume resources. Our solution is to remove these unnecessary pixels using a mask of the areas that we think are important.

Herein, we propose a technique for creating the bitmap mask that will be applied to the dataset. The steps involve (1) creating the average image of the dataset, (2) applying a pixel threshold to this image and finally (3) applying the mask to all images in the dataset to create a new dataset, before (4) retraining the network on the masked input. It is worth noting that we have investigated the application of the dilation operation on the binary mask in the aim of improving our results.

3.1 Averaging the Dataset

In order to obtain the bitmap mask, we first compute the average image A of the dataset to filter the background pixels. In this work, we aim to consider the pixel's intensity only in the pedestrian location; because the negative examples would not contain any instances of pedestrians, they do not need to be used to calculate the average. Therefore, we propose to only use our positive training examples when creating the average as follows:

$$A(i,j,l) = \frac{1}{K}\sum_{k=0}^{K} I^k(i,j,l) \tag{1}$$

where $I^k(i,j,l)$ is the pixel intensity at position i and j, of the k'th image in our dataset and l represents the index of the colour channel. Each pixel is defined as the average of the pixels at the same position of all K images in the dataset.

In the case of color images, Red Green Blue (RGB) images like in the INRIA dataset, the average image A is converted to gray scale image, resulting one channel average image as seen in figure 1c.

3.2 Pixel Threshold Method

The averaging operation obtained previously results in a gray scale image, where there is a differential between the pixels where a pedestrian is most likely located and those of the background pixels as seen in figure 1c. In order to filter the background pixels, we propose to extract a bitmap mask by applying a thresholding operation as follows:

$$m(i,j) = \begin{cases} 1 & if A(i,j) > T \\ 0 & otherwise \end{cases} \tag{2}$$

where $m(i,j)$ is the bitmap mask pixel at i and j, and T is the threshold chosen. We determine if the pixel in the bitmap is on or off by comparing the pixel intensity of our average image with pixel threshold T. If the pixel intensity is above the threshold it is set to 1, otherwise it is set to 0.

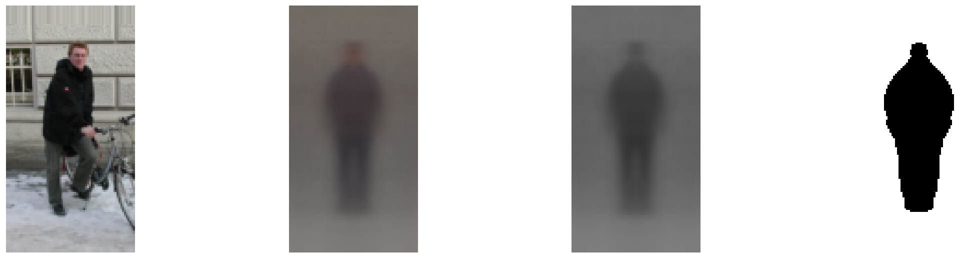

(a) (b) (c) (d)

Figure 1: From the INRIA dataset, (A) is a sample image from the dataset, (B) is the average of all images in the dataset, (C) is the greyscale of the average, (D) is the optimal mask level found using a flat threshold.

3.3 Dilation

With the aim of recovering the lost positive pixels during the mask creation, we propose to augment the area of the binary mask m, obtained by Equation 2 in Section 3.2. We do that by the application of the morphological dilation operation on this mask, defined as:

$$m_{new} = \delta_H(m) \tag{3}$$

where δ_H represents the dilatation operation with a circular structuring element H and m_{new} represent the resulting mask at threshold T.

By applying the dilation operation, we can add more information to the input image from the positive pixel's neighbourhood and produce a human shaped mask, as depicted in figure 2b. We do this for different numbers of iterations of dilation to create a number of different masks to test.

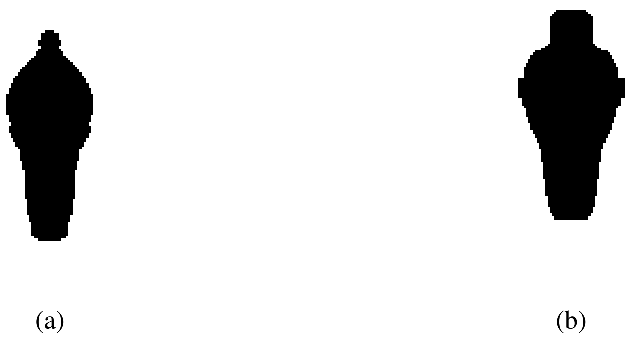


Figure 2: Based on the INRIA dataset examples of, on the left is our human-like mask and on the right is the mask with dilation applied.

3.4 Dataset creation and model training

Finally, the binary mask obtained previously is used to generate a new dataset that can be used to train the pedestrian detection. Furthermore, it is used during the inference stage to mask the negative pixels in the input image. This is done by overlapping the binary mask on each image in the dataset. Therefore, each image for the new dataset I^k_{new} is defined as the element-wise product of the original image I^k with the dilated mask m_{new}, formulated as:

$$I^k_{new} = m_{new} \odot I^k \tag{4}$$

where the operator $\odot$ represents the element-wise multiplication.

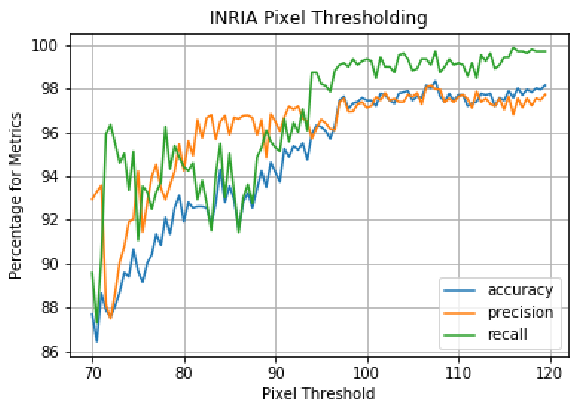
To train the model, we then split the new dataset into training and validation sets. The network is retrained on the training set with the same parameters that were used to train the original dataset, while the validation set is used to evaluate its performance. Hence, we train a Resnet18 model using two generated datasets derived from INRIA person dataset and the Daimler pedestrian dataset. The learning rate is initially set to 1e-3 and momentum set to 0.1.

4 Results

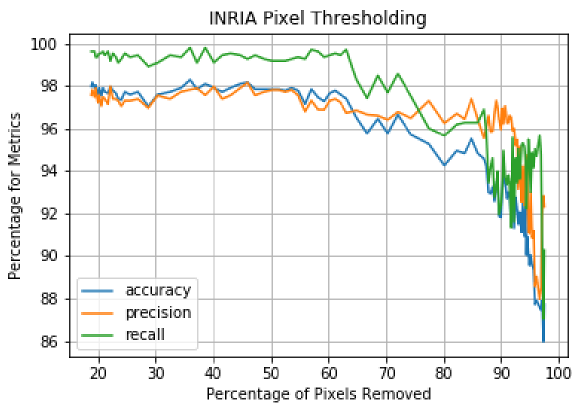
For our experiments, we generated multiple new datasets, from the INRIA person dataset and the Daimler pedestrian dataset, with different threshold values and dilation iterations. In this section, we analyze the effects of the choice of these parameters to the training performances. Furthermore, the impact of the proposed technique in increasing the activation sparsity.

4.1 Pixel Thresholding

In order to illustrate the effect of threshold value on the training performances, the three metrics accuracy, precision and recall are measured by varying two threshold parameters which are the pixel threshold value and the percentage of pixels removed from the input images.

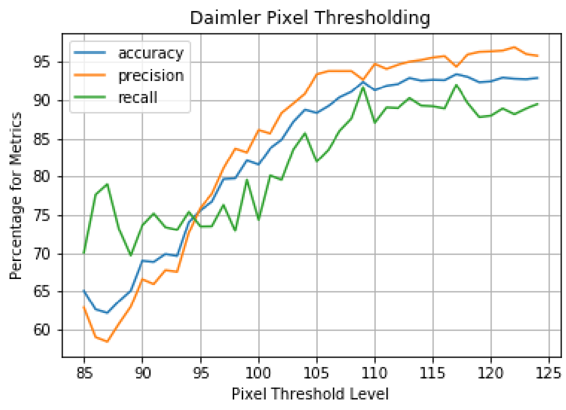


(a)

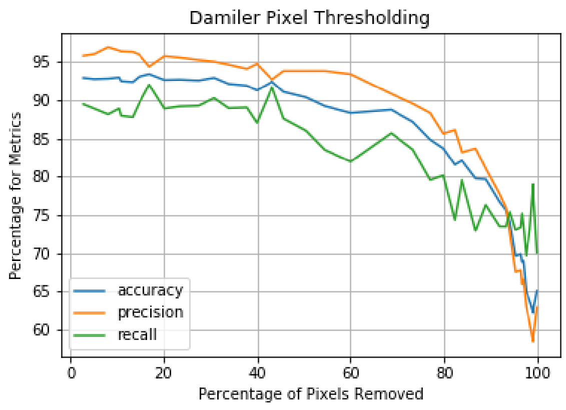


(b)

Figure 3: Using the INRIA dataset, (a) is a graph of the precision, recall and accuracy when using a bitmap mask at various pixel thresholds as seen on the x-axis. (b) is a graph of the precision, recall and accuracy when using a bitmap mask at percentages of pixels removed from the input image when using pixel thresholding as seen on the x-axis.



(a)



(b)

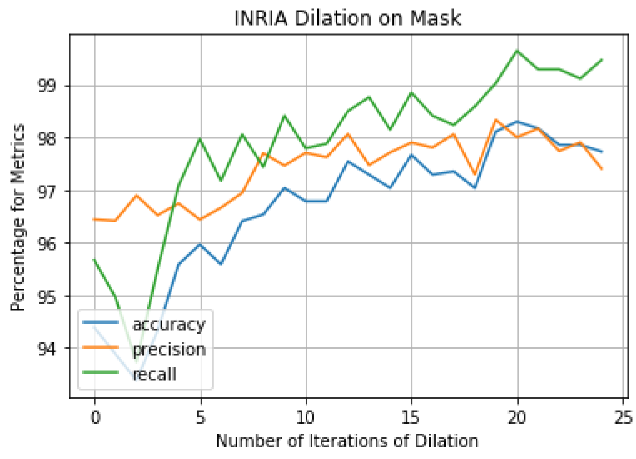
Figure 4: Using the Daimler dataset, (a) is a graph of the precision, recall and accuracy when using a bitmap mask at various pixel thresholds as seen on the x-axis. (b) is a graph of the precision, recall and accuracy when using a bitmap mask at percentages of pixels removed from the input image when using pixel thresholding as seen on the x-axis.

Figure 3a shows the accuracy, precision and recall after we have trained our network on various different masks at different pixel thresholds using the INRIA dataset. After creating a mask with a pixel threshold greater than intensity 100, we do not see any further improvement in our accuracy. We can therefore safely remove any pixels below an intensity of 100 without a loss in accuracy. In figure 3b we can see what percentage of pixels are removed from our input images and can see that we can remove up to 60% of the input image without seeing a decrease in accuracy.

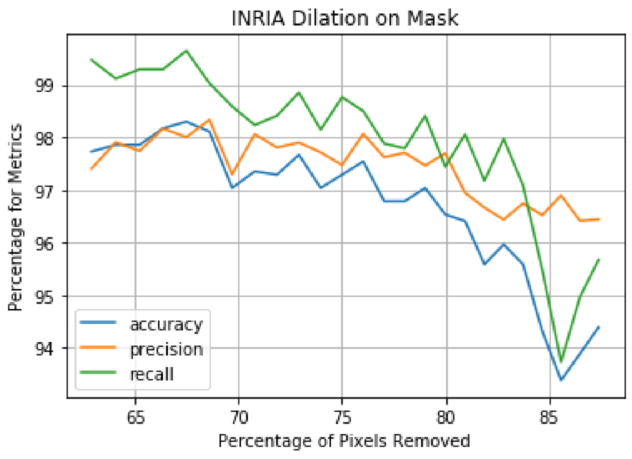
Figure 4a shows the accuracy, precision and recall after we have trained our network on various different masks at different pixel thresholds using the Daimler dataset. After creating a mask with a pixel threshold greater than 110 we do not see any further improvement in our accuracy. In figure 4b we can see what percentage of pixels are removed from our input images and can see that we can remove up to 50% of the input image without seeing a decrease in accuracy.

4.2 Dilation

In order to select the optimal mask that gives a high accuracy with maximum number of removed pixels, we vary the number of iterations of the dilation operation. We also show the change in the amount of input considered after the dilations.

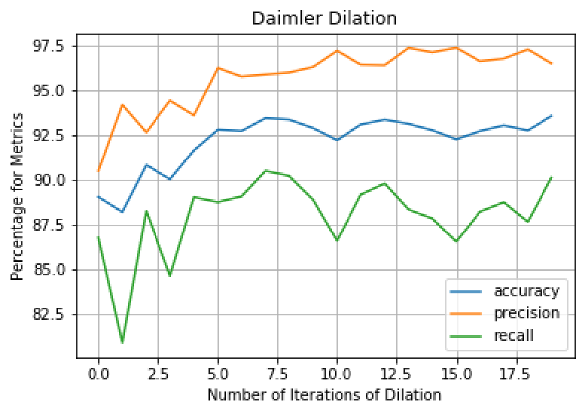


(a)

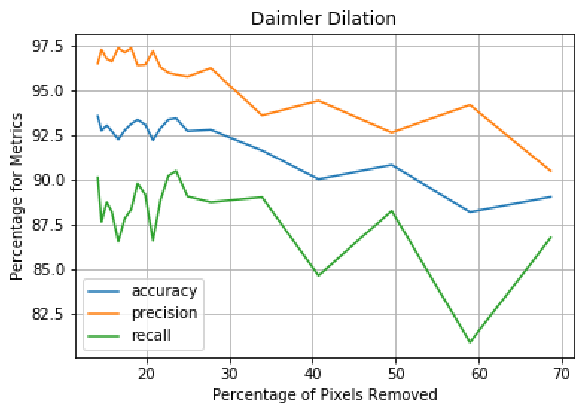


(b)

Figure 5: Using the INRIA dataset, (a) is a graph of the precision, recall and accuracy when using a bitmap mask that is being dilated and the number of iterations of dilation as seen on the x-axis. (b) is a graph of the precision, recall and accuracy when using a bitmap mask that is being dilated and the percent of pixels removed as seen on the x-axis.



(a)



(b)

Figure 6: Using the Daimler dataset, (a) is a graph of the precision, recall and accuracy when using a bitmap mask that has been dilated a number of iterations as seen on the x-axis. (b) is a graph of the precision, recall and accuracy when using a bitmap mask at percentages of pixels removed when using dilation as seen on the x-axis.

When dilation is applied to the mask, more information from the original image is retained to be trained upon; because of this the accuracy, precision and recall all improve as more of the image is added back to the image. This dilation can account for poses that fall outside the bounds of the original mask. As we can see in figures 5a and 6a initially the dilation leads to increases in accuracy however this quickly levels off as we begin to add back in information that is unnecessary for classification and detection.

4.3 Activation Sparsity effects

In addition to the improvements in accuracy seen above, the use of the proposed masking technique increases the number of non-zero activations and therefore provides a speedup of inference and a reduction in energy consumed. This is due to the parts of the input image that have been masked out to zero are being multiplied by our non-zero weights giving us a zero activation. Using a hardware accelerator we can take advantage of these sparse activations to speedup our inference and lower our power using a sparse accelerator such as SCNN [Parashar et al., 2017].

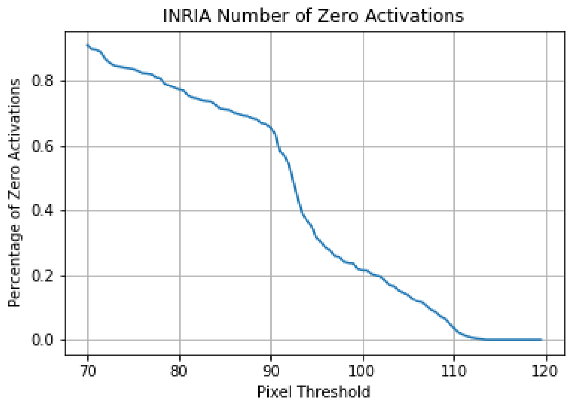


Figure 7: The percent of zeroes in decimal, in the activations after the first convolution layer against the pixel threshold applied of the mask applied.

For the INRIA dataset at a pixel threshold of 90, which we showed above has similar accuracy to the whole input image, we have around 60% activation sparsity; from [Parashar et al., 2017] at 60% activation sparsity we can achieve up to a 3$\times$ speedup and using almost 3$\times$ less energy with a hardware accelerator exploiting sparsity.

5 Conclusion & Future Work

In this paper we have shown that bitmap masking of training images has benefits in the case of pedestrian detection. We have presented a method for creating and applying a bitmap mask to remove the background pixels that are unnecessary for the classification/detection process and how to expand this mask to gain more accuracy if needed. We have shown for the INRIA and Daimler datasets that we can remove up to 50% of the input pixels and still maintain the same accuracy as the case of using the whole image.

Future work will focus on extending the work to other object classes to ensure that the results generalize to to other problems. Face detection is likely suitable, because the input bitmap masks have previously been used successfully using the HoG/SVM method [Dehghani et al., 2017]. Further, we plan to investigate whether the masking technique works on the classes in the ImageNet dataset. We will also look at potential increasing the size of the mask by also removing the foreground and creating a boundary bitmap based on work done using HoG/SVM and applying this technique[Dehghani and Moloney, 2017]. It is also believed that the bitmap approach may improve network robustness to adversarial noise [Jin et al., 2018], and further work can look into the effects of bitmaps to combat adversarial noise.

Acknowledgments

This work was supported by Intel-Movidius and Trinity College Dublin. The authors would like to thank all the Advanced Architecture team members for their support.

References

[Dalal and Triggs, 2005a] Dalal, N. and Triggs, B. (2005a). *Histograms of oriented gradients for human detection*. Computer Vision and Pattern Recognition, IEEE Computer Society Conference.

[Dalal and Triggs, 2005b] Dalal, N. and Triggs, B. (2005b). *INRIA person dataset*.

[Dehghani and Moloney, 2017] Dehghani, A. and Moloney, D. (2017). *Speed Improvement of Object Recognition Using Boundary-Bitmap of Histogram of Oriented Gradients*. IEEE.

[Dehghani et al., 2016] Dehghani, A., Moloney, D., and Griffin, I. (2016). *Object Recognition Speed Improvement Using Bitmap-HoG*. IEEE Computer Society Conference.

[Dehghani et al., 2017] Dehghani, A., Moloney, D., and Xu, X. (2017). *Face Detection Speed Improvement using Bitmap-based Histogram of Oriented Gradients*. IWSSSIP 2017.

[DeVries and Taylor, 2017] DeVries, T. and Taylor, G. W. (2017). *Improved Regularization of Convolutional Neural Networks with Cutout*. arXiv.

[Herculano-Houzel, 2009] Herculano-Houzel, S. (2009). *The human brain in numbers: a linearly scaled-up primate brain*. Front Hum Neurosci. 2009;3:31.

[Jin et al., 2018] Jin, J., Dundar, A., and Culurciello, E. (2018). *Robust Convolutional Neural Networks under Adversarial Noise*. ArXiv.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner., P. (1998). *Gradient-based learning applied to document recognition*. Proceedings of the IEEE.

[Munder and Gavrila., 2006] Munder, S. and Gavrila., D. M. (2006). *An Experimental Study on Pedestrian Classification*. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Parashar et al., 2017] Parashar, A., Rhu, M., Mukkara, A., Puglielli, A., Venkatesan, R., Khailany, B., Emer, J., Keckler, S. W., and Dally, W. J. (2017). *SCNN: An Accelerator for Compressed-sparseConvolutional Neural Networks*. arXiv.

[Yong, 2013] Yong, E. (2013). *Monarch butterflies navigate with compass but no map*. Nature.