Real Time Logo Detection and Localization Network

Mukhil Azhagan Mallaiyan Sathiaseelan, University of Florida

***Abstract*— Detection of Logo is an important problem in current times. The prevention of piracy, unethical usage and unsponsored use of digital media without prior permission, commercial license or credit to the owner is a growing problem especially since social media like Facebook, YouTube and Instagram etc. There is also a case of reproducing similar logo of famous companies, in hopes to misuse their popularity. The paper presents an object Detection system using Convolutional Neural Networks (CNN), to identify and also locate the position of these logos. The system has a Preprocessing block, and the CNN itself. Discussion about both of these parts are presented. The research is aimed towards making a Real time system capable of understanding and isolating Logos in videos and other digital media, which can later be sent for evaluation on its originality and ownership. The presented proof of concept is able to produce 95% Accuracy across various logos and is able to perform detection in about 300ms. So, it can work on real-time systems producing 3fps detection of Logos.**

***Index Terms*— Convolutional Neural Network, Logo Detection, Localization**

I. Introduction

Logos or Trademarks is an important intellectual property of a company. It is an intangible asset that carry a name, popularity as well as the trust earned by the company. In recent times, there have been numerous incidents where there is unauthorized use of some Logo, without getting the approval of the company. Some are unintentional in case of using a product in videos etc. and they can also be free advertising for the company, but there are cases of illegal distribution and counterfeit products that carry the name and logo but are either sold for cheap prices or lack the quality of the original Product. Figure 1 presents some of such cases for sake of discussion.

There are many such Problem Scenarios and the main objective of the project is enabling a way to identify such misuse for both consumer Users to not be cheated as well as for companies to claim their ownership and their profit due to use of their products in commercial applications. It is also possible to use this framework to identify logos that are too similar in look to a famous brand. This is usually done to cheat ignorant people, and it is a daily occurrence in local black markets in cities all over the world.

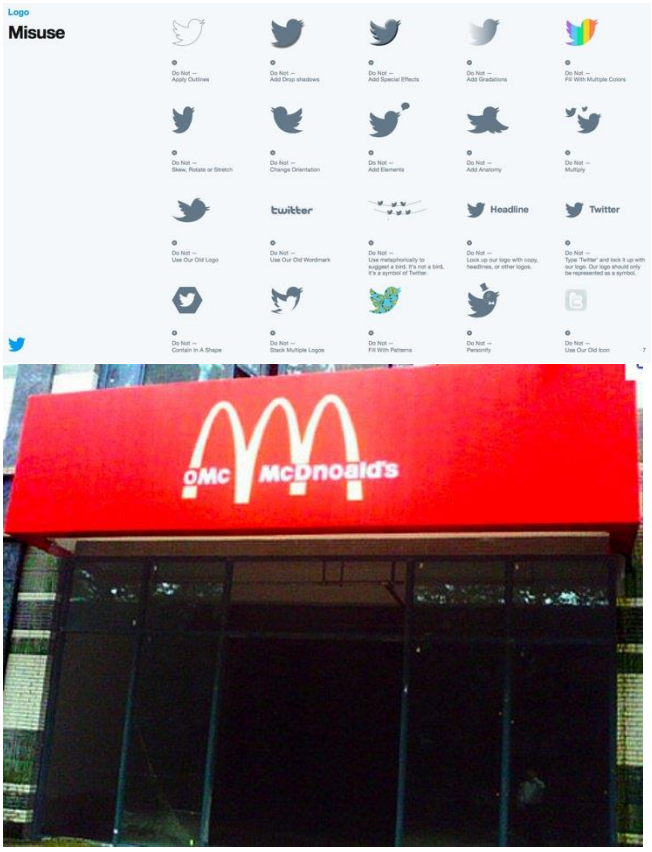

Figure 1: Twitter publishes article on how modifying a logo even giving credit can be logo misuse [21] (Top). The use of a similar logo to cheat consumers [22] (Bottom).

The key Idea is the use of Convolutional Neural networks to Identify Logos. In addition to this, the project would also be able to locate and draw a bounding box around the identified Logo. There are numerous methods such as RCNN [1], Faster RCNN [2], Mask RCNN[3], SSD[4], Yolo[5] etc. All of these methods have a common framework. They have a preprocessing stage, a classifier and a post processing phase. Combining all of the above steps, a system able to not just classify but also to locate the object of interest.

The main contribution of the framework is using a method called Region based Convolutional Neural Network (RCNN), which is a CNN based approach to object detection. The system is trained on Flickr Logo Dataset

- Some Text has been taken from my project proposal.

[6][7], the dataset is augmented multiple times over included various shifts and rotations. These images do not contain just the Logo but include other information as well. There is a script written to load the augmented data and extract just the logos for training. After this phase a CNN is trained on these images. After Training, the images are tested, and an Intersect over Union algorithm performs Non maximal Suppression to obtain the best location of Logos.

The system is optimized to perform real time, the entire process of detection and identification can provide 3fps detection results. Various experiments have been performed and the best architecture is selected so that it can perform as optimized as possible. These trails and their results will be presented in further sections.

II. Description

The Solution for Logo Detection and Localization is not a one-shot solution. As mentioned, there are various algorithms that have incorporated in the process. In each of the steps of, Preprocessing, Classification and Detection there are numerous methods one could follow. In the Preprocessing stage for instance, various approaches based on thresholding, chromaticity, watershed segmentation, contouring are some of the many approaches available. In the Classification part, various architectures of CNN can also be used. There are also other methods for object Detection such as SSD that do not follow these methods at all. However, in this paper, we will follow a method for Region based CNN, called RCNN. In each of the steps, we will describe the methods used.

1) Preprocessing Stage:

The main purpose of this stage is to remove noise, remove background information and Extract Windows or Regions of Interest. In this approach, I use a Selective Search Algorithm [8]. The selective search algorithm analyzes similarities in color, texture, region size etc. Using the information from these, different regions are detected. These are called regions of interest, also called Region Proposals. It is with these images that we classify the images using a CNN that has been trained to detect the Logos. Depending on the image, and the approach, multiple regions are proposed from each image, the number of regions are different, but that doesn't matter, as only the detection that cross a threshold will be selected

2) CNN Classifier:

This stage is the classifications stage. In this project, I have used 3 convolutional layer ,2 fully connected layer CNN, which was found to be the best accuracy. I have tried with various similar architectures, the results for which will be presented.

While CNN is quite a popular method, though the CNN is not the main goal of the project I will try to present a basic explanation of the network. Figure 2 gives a comprehensive look at a CNN. Artificial Neural Networks (ANN) [9] can perform classification, they might have any where from one to many layers of activations, the input stage is called the input layer, the middle layers are called hidden layers, and the final layer is called the output layer. As mentioned before, neural networks perform non linear computations to separate the input feature space into different regions. And That is all it can do. A convolutional neural network on the other hand is a neural network that is preceded by a convolutional block. While ANN can learn images, it cannot capture texture information and it is not invariant to scale and rotation. But a CNN on the other hand, due to the convolutional block, can successfully perform windowing, convolutions and capture neighborhood information within an image.

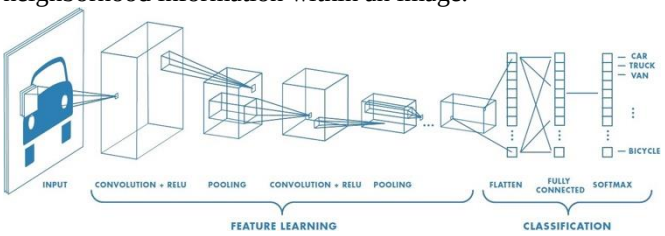

Figure 2: Convolutional Neural Network [23].

As seen in Fig 2, a sliding Window is run across the entire image, these windows are most often overlapping by a particular extent defined by the developed. These windows are convolved, given weights and passed through nonlinearities. Over the years, improvements have been developed such as max pooling [10] that down samples the image to smaller regions. This method makes the CNN robust to scale differences.

At the end of the convolutional block, which incorporates a convolutional block, max pooling layers and dropout layers, there maybe other following convolutional blocks. After required number of conv blocks, the 2D matrix structure is flattened into a fully connected layer. At this stage the CNN becomes similar to an ANN. The final layer is the output layer and it has as many neurons as the output classes.

During the training phase, various images are sent through the network and in the output layer they are compared with the labels that have already been defined for each image. There is a cost function such as Mean Squared Error (MSE) that becomes the minimization criteria . There is also a optimization algorithm such as Gradient Descent or its various variants like AdaGrad, or even any other approach for that matter. Backpropagation is the process of pushing this cost function through the entire network to modify each neuron in the network to train it every iteration.

ANN and CNN are quite complicated topics, and their theory cannot be described in the project. They are popular approaches and can be read about from literature. [11] is a good resource to learn about neural networks

For the training phase of the project. There are 3 different datasets, Flickr27 having 810 images, Flickr32 having 8240 images and Flickr47 dataset with 10000+ images. I have used the Flickr 27 dataset, the reasons for which will be discussed later. There is a script that can read the annotated dataset and create an image database. Using the images and labels a CNN is trained. This network is capable of identifying with 98%

accuracy for most logos, with an average accuracy of 99%. However, the network guarantees 95% detection performance in all the logos. After the detection, the images and their location are sent to post processing where the results are consolidated together.

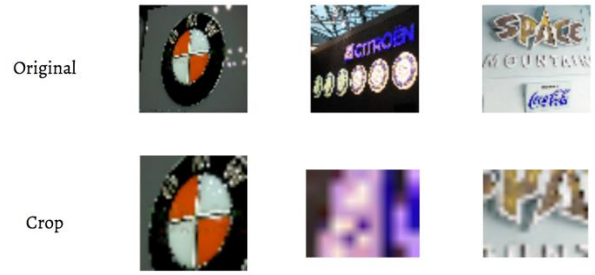

Figure 3: Flick27 Database and the various data augmentation done.

3) Post Processing Phase:

After Classification, as we know for each image, there are many regions proposed from preprocessing, and each of them have a decimal accuracy. The most accurate will naturally be the logo, but it is a common case that different regions in fact are showing the same object and each of these have been identified as a Logo. Thus, non-maximal suppression algorithm will remove the overlap. The idea is to look at Intersect over union (IOU), which is a ratio that identifies overlapping regions. The algorithm has a value of 0.5 IOU ratio which means overlapping regions over 50% are accumulated together as one whole region. Thus, single instances of the same object are put together and one bounding box is produced. The output produces an image with a bounding box drawn around it, and the class the logo belongs to.

Figure 4: Image showing Non-maximal suppression

Using the three steps mentioned above, the entire system of Logo Detection is completed. The results of each of these methods, will be described in the next section. There were a number of trails done and they will be presented in the further section as well.

III. Evaluation

The project was performed on a MacBook Air that has Intel core i5. It does not have a dedicated graphics card. Many trails took especially a lot of time. For Instance, the best architecture from the trails which is used for the final results took,

File read, make csv : 2 hours 15 mins
Dataset reading : 1 hour 10 mins
Data Augmentation : 55 minutes
Storing Pickles : 15 minutes
Training : 5 hours 25 minutes
Testing 1 Image : 300 ms
Total Time taken : ~4 hours for data generation
5.5 hours for Training.

This is of importance because, based on this I set up the trails. First I tried different number of network layers for a constant learning rate of 0.001 and constant fully connected network of size flatten size to 2048 to 10 in output. With these constants the convolutional block sizes were varied. Based on the best, further experiments in Training was done with step sizes etc.

The Dataset:

As discussed previously, I have shortlisted 3 Different datasets that provide annotated dataset for Training images of Logos. These are the Flickr27[6], Flickr32 and Flickr47 datasets [7]. The latter two need correspondence with the authors, which as obtained, and the dataset was downloaded but the images were close to 10000 images. Considering it took around 4 hours to reading and augmenting 891 images, this process along would have taken close to 2 days to be performed without errors and mistakes in data augmentation. Thus, I decide to work with Flickr27 Dataset. This has 891 images of Logos under 27 classes. I performed various augmentations such as shifting, cropping, rotation etc. which resulted in 140137 images

Trails of different architectures:

Three different architectures were tried. The best result is Trail 2 as can been seen from Table 2. The idea was to try out varying sizes, for instance, we know from ZFNet [12], that first convolutional blocks learn lines, the second blocks learn circles and contours, the third block learns combination of circles and contours, in terms of more complex shapes. So, having a 1 conv block is not really suggested as the logos definitely have circles and complex shapes. Even though we know 2 convolutional blocks may not work well, it is good to try for the sake of experimentation. So, there are trails for 3 different architectures of varying 2 convolutional block, 3 convolutional block and 4 convolutional block. Between all these layers there is a max pooling layer.

Trail	**1**	**2**	**3**
Input	64x32x3	64x32x3	64x32x3
Convolutional Block	32	32	32
		64	64
	64	128	128
			248
Fully Connected	2048	2048	2048
	128		512
Training Time	3.5 hours	5.5 hours	9.25 hours
Training Acc.	82.7%	***95.6 %***	96.9%
Validation Acc.	90.3%	***96.4%***	92.4%

Table 1: Experiments run on CNN Architectures.

As can be seen from Table 1, the best architecture chosen is the Trial 2. Also, in Tabulation the fully connected layers are varying, this shouldn't have much of a difference. In fact, it can be seen that just 1 fully connected network performs the best. The choice of not including 1 more fully connected network is just to reduce training time.

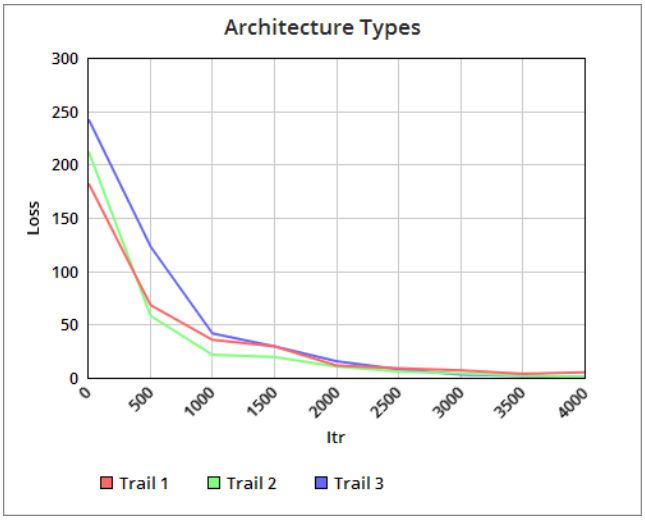

Figure 5: The Loss Curve for 3 CNN Architectures

The next set of trails was done using the selected architecture from Trail 2. Below is a figure, Fig 6 made to scale with information about the filter sizes, the max-pooling and the fully connected layers ending with the 27 classes of the Flickr Dataset. The stride of filters is made to match the original image with zero padding, filters are of size 5x5x3. Max-pooling is done with a 2x2 filter. They are flattened to a 2048 neuron with a Dense network finally ending at the output of 27 classes with a softmax classifier.

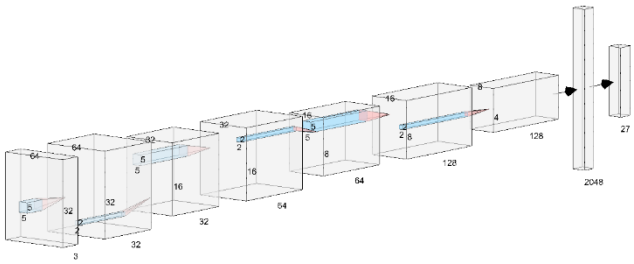

Figure 6: Architecture made to scale using Using tool from [24]

For the secondary trail, I experimented on the different step sizes. It has been long established that Relu performs as good as expected, as compared to other nonlinearities, so I am not experimenting with them. I use the Adam optimizer with the following parameters

- ***Learning rate or alpha****. This is Also referred to as the learning rate or step size. This is where I experiment with the results, this is the most significant value for learning.*
- ***Beta1****. This determines the exponential decay of the rate of first moment estimates. This is defined as 0.9, so we have a mild momentum in decay.*
- ***Beta2****. This is the exponential decay rate of the second-moment estimates. This is defined as 0.999, which is the widely used in literature. It is set close to 1 for problems that have a sparse gradient such as in NLP in computer vision.*
- ***Epsilon****. This is a sort of regularization factor to prevent division by zero during the implementation. Here the value chosen is 10E-8.*

Trail	**1**	**2**	**3**
Step Size	0.01	0.001	0.0001
Iterations	4000	4000	4000
Training Acc.	90.7%	96.8 %	**98.6%**
Validation Acc.	89.3%	95.9%	**98.9%**

Table 2: Step Size trails

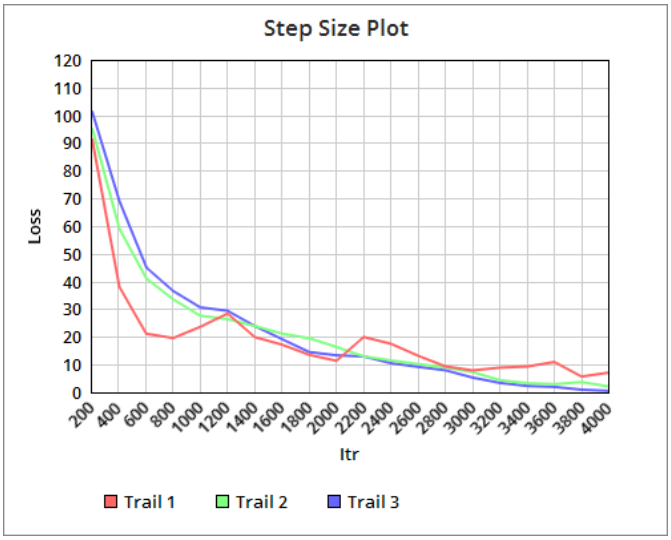

Figure 7: Step sizes trail. The best result is from Trail 3

Trail 3 of step size 0.0001 gave the best results. Even lower step sizes take much longer to stabilize and converge, so it was not tried. However given the current training methodology and training time, the best architecture and step size is found. These results will be tested using test images and results will be presented below

Results:

Results on the RCNN using the best architecture is presented in this section. As mentioned, there is a postprocessing step after classification. Fig 8, gives an idea of the number of classification done. There are wrong classifications as well, but the detection probability for these are less, so after the thresholding for classification these get removed. After which Non-Maximal Suppression is done among the same classification results.

Figure 8: Image before high probability classification done (Top). Image before Non-Maximal Suppression and similar classification (Bottom).

Figure 9 shows the result after post processing, it correctly identifies many logos, with an average accuracy of 98%.

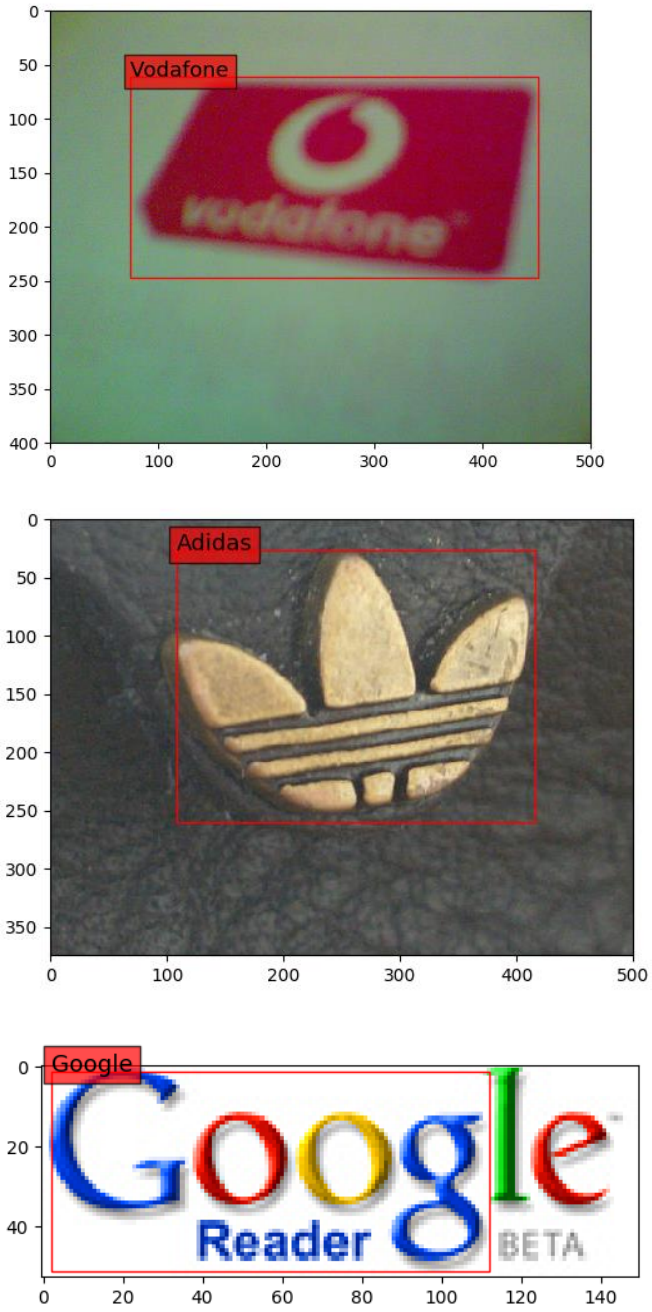

Figure 9: Images of Logos Identified with Localization after Non maximal suppression and removal of low probability results

Discussion:

Figure 8 and 9 give an example of the working of the network. From visual inspection, I can guarantee that the network works best if the size of the logo is around >50 pixels in rows and >50 pixels in columns, as seen in the google image in Fig 9. However, we can also see that the small DHL at the back of the truck in Fig 8, is not detected because its size is

approximately 30x60, which is less that the expected size of 64x32 pixels. Other than that, for all images perform with worst case accuracy of 95%.

The network is not perfect however, and there are errors in the system. Figure 8, for instance is classified as 100 percent, because it identifies the color of red and yellow, but with 3 different regions, and the label of DHL is given to the entire image. So, the network assumes that it has accurately identified DHL, but the location of DHL is wrong. There are other similar cases as well.

IV. Related Work

Logo Detection comes under Object Detection, and there has been numerous papers about Object Detection. Given enough dataset, the process can be extended to any other problem statement or scenario with minor modifications. Some of the popular examples are [13][14], where they have used object detection mainly for the case of Self Driving and License plate registration. In these they use approaches like RCNN, YOLO and other Region based CNN method for identifying objects. The main drawback of the current methods such as RCNN was the inclusion of a image processing block as the preprocessing layer , whereas even that was removed in YOLO and SSD. They used the CNN itself as a means to learn the Region of Interest. Almost all of the methods have now been optimized to perform real time , as can be seen in the project , I am using one such implementation for the case of Logo Detection.

There has been related work done in Logo Detection as well and it is a growing field and many companies are interested in it. Alibaba's AliExpress was interested in identifying logos in their products and they have worked on Logonet[15], which is capable of reading through images of each of its product pages in its website to identify counterfeits. This has not yet been implemented, but it is in the works. There has also been open challenges on Logo Detection[16][17][18], but these approaches do not focus on the system being realtime. The idea of applying them on Live videos so as to detect these logos have not been implemented. Most of these will still use Mask-RCNN, SSD, or YOLO, but have complex networks, sometimes as big as 5 convolutional blocks that significantly increase the time. In the project, I have reduced the architecture size, yet being able to identify with high accuracy. And the testing time is quick and able to run realtime. There is still scope for improvement in the field, and recently people have started to use Siamese networks [19] for identification as well, though it was not built for the purpose of multi class object detection. There is a definite scope in the community to work in this direction.

V. Conclusion

Many companies as well as the government [20] have raised concerns on the increasing fake products. YouTube has also reported similar problems in their videos where creators showcase unsponsored content and other companies feel is ill advertising for the company that is showcased. Such cases will be overcome by such a system that is able to identify Logos used in videos and say if there is some company trademark that is present. This can help both the creator if they have included it without intention, and also YouTube to maybe blur the area to prevent problems. In addition to this, this will also aid Amazon, AliExpress to prevent sale of counterfeit products.

Acknowledgment

The paper is the work of Mukhil Azhagan Mallaiyan Sathiaseelan for the Course Pattern Recognition during the Spring 2019 Semester at University of Florida, Gainesville.

References

[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 580–587). https://doi.org/10.1109/CVPR.2014.81

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[3] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (Vol. 2017–Octob, pp. 2980–2988). https://doi.org/10.1109/ICCV.2017.322

[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. *ECCV*.

[5] Redmon, J., Divvala, S.K., Girshick, R.B., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.

[6] Flickr27 Dataset: http://image.ntua.gr/iva/datasets/flickr_logos/

[7] Flickr32,47 Dataset: http://www.multimedia-computing.de/flickrlogos/

[8] Uijlings, J.R., Sande, K.E., Gevers, T., & Smeulders, A.W. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision*, *104*, 154-171.

[9] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F.E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11-26.

[10] Wu, H., & Gu, X. (2015). Max-Pooling Dropout for Regularization of Convolutional Neural Networks. *ICONIP*.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. The MIT Press.

[12] Zeiler, M.D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *ECCV*.

[13] Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., & Urtasun, R. (2016). Monocular 3D Object Detection for Autonomous Driving. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2147-2156.

[14] Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.E., Gonçalves, G.R., Schwartz, W.R., & Menotti, D. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1-10.

[15] Hoi, S. C. H., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., & Wu, Q. (2015). LOGO-Net: Large-scale Deep Logo Detection and Brand Recognition with Deep Region-based Convolutional Networks. Retrieved from www.aliexpress.com

[16] Tüzkö, A., Herrmann, C., Manger, D., & Beyerer, J. (2017). Open Set Logo Detection and Retrieval. Retrieved from http://s.fhg.de/logos-in-the-wild

[17] Alaei, A., & Delalandre, M. (2014). A complete logo detection/recognition system for document images. In *Proceedings - 11th IAPR International Workshop on Document Analysis Systems, DAS 2014* (pp. 324–328). IEEE. https://doi.org/10.1109/DAS.2014.79

[18] Su, H., Zhu, X., & Gong, S. (2018). Open Logo Detection Challenge. Retrieved from http://www.eecs.qmul.ac.uk/~hs308/https://www.eecs.qmul.ac.uk/~sgg/

[19] Koch, G.R. (2015). Siamese Neural Networks for One-Shot Image Recognition.

[20] https://www.worldtrademarkreview.com/anti-counterfeiting/us-government-report-finds-staggering-ratio-fakes-major-e-commerce-sites

[21] https://paragraphs.com/brand/

[22] https://izismile.com/2011/12/29/fake_mcdonalds_around_the_world_25_pics-19.html

[23] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[24] CNN Model Creator tool : http://alexlenail.me/NN-SVG/AlexNet.html]