International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

One-Stage Logo Detection Framework using AdaBoost Resnet50 Backbone

Sarwo, Yaya Heryadi, Widodo Budiharto, Edi Abdurachman

Abstract*: Logo is an important asset as it is designed to express identity or character of the company or organization that owns the logo. The advent of deep learning methods and proliferated of logo images sample dataset in the past decade has made automated logo detection from digital images or video an interesting computer vision problem with wide potential applications. This paper presents a novel one-stage logo detector framework in which the backbone of the proposed logo detector is a deep learning model which is trained supervisedly using gradient descent training algorithm and the target logo classes as input dataset. The experiment results showed that AdaBoost Resnet50 (0.58 MAP) as the logo detector backbone outperforms Resnet50 (0.56 MAP), VGG19 (0.32 MAP), and AdaBoost VGG19 (0.56 MAP).

***Keywords* : Logo Detection, AdaBoost, Resnet, One stage Detector.**

I. INTRODUCTION

Logo is a term refers to "a design or symbol used by a company to advertise its products" or "a design or symbol displayed on a company's products, vehicles, signs, etc. that expresses the company's character and purpose and makes it easy for customers to recognize and remember the company" [1]. The advent of machine learning methods and proliferated of logo images in the past decade has made automated logo detection an interesting computer vision problem with wide potential applications such as: identifying goods, services, corporations, contextual ad placement, validation of product placement, and online brand management by Iandola [22].

Logo detection can be viewed as a subproblem of object detection task whose objective is to locate position and classify logo of interest appearing in an input images. Although it might an easy problem for human, the task is a challenging problem for computers as there may be a number of objects of interest appearing in the image. The additional requirement to predict bounding boxes that will cover the detected logo make the problem even more challenging. A bounding box is typically defined by four parameters [x, y, w, h], where (x, y) indicate a reference spatial position or coordinate in the box i.e. the center of the box or the upper-left corner, and (w, h) are set for the width and height of the box (see Fig. 1 as illustration). In the past ten years, a plethora of studies have proposed many logo detection method resulted in a vast number of methods available in literature. For example, The study by Girshick in 2015 [3] proposed R-CNN model. The study by Ren, He, Girshick and Sun [4] improved R-CNN model by proposing Faster R-CNN [4] model.

According to Lin, Ai, & Doll [5], a plethora of object detection methods can be divided broadly into three categories. First, Classic object detection methods based on sliding-window paradigm. Using this approach, firstly a rectangle with different sizes are moved over the whole image in order to find relevant objects. Then, a classifier is applied on a dense image grid. Some prominent object detectors of this kind are: convolutional neural networks [6], [7], boosted object detectors [8], and template matching [45]. As input for object detection, some image features are exploited such as: HOG descriptor [47], and and integral channel features [11]. Second, one-stage object detection methods which combine classification and detection tasks in a single step. In the object detecting process, a detector is applied over a regular, dense sampling of object locations, scales, and aspect ratios. For example: SSD [9], YOLO [10], and RetinaNet [5]. Finally, two-stage object detection methods which work in a two-stage process. The first stage generates a set of candidate proposals that contain all objects of interest and filtering out most of negative locations. The second stage classifies the proposals produced by the first stage into foreground classes/background. For example: R-CNN [23], Fast R-CNN [3], Faster R-CNN [4].

An interesting study findings reported by Lin, Ai, and Doll [5] to compared one-stage and two stage detectors. The authors concluded that one-stage detectors, such as SSD and YOLO, are simpler, more intuitive, and faster than two-stage detectors; however, it lacks of accuracy and high number of generated boxes (see Fig. 1). In contrast, separating two processes in two-stage object detectors give high object detection accuracy but at the expense of high computational complexity.

Manuscript published on 30 September 2019
* Correspondence Author
Sarwo*, Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia. Email: sarwo.jowo@gmail.com
Yaya Heryadi, Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480. Email: yayaheryadi@binus.edu
Widodo Budiharto, Computer Science Study Program, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480. Email: wbudiharto@binus.edu
Edi Abdurachman, Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480. Email: ediA@binus.edu

Many recent reports on logo detection showed some evidences that logo detection remains a challenging problem in computer vision due to the following factors: (1) low resolution of the logo image, (2) high variation of logo images, and (3) logo that change often (See Fig 2). As can be seen in Fig 1., logo in (a), (b), and (c) have different layouts; whilst, (d) and (e) the same logo with different size.

In addition to the mentioned challenges, many practical applications of logo detection requires almost real time detection which turned many studies to one-stage logo detectors. Despite some one-stage logo detectors have been proposed that run faster or similar speed with two-stage logo detectors, accuracy of these one-stage logo detectors often below the state-of-the-art two stage object detectors. Pursuing this objective, many previous methods are typically choose one of the following opposite direction. First, starting with fast logo detector models and improving its accuracy by modifying the model. Second, starting with high accuracy/complex object detector models and reducing its running time. For example, by reducing input image resolution or the number of proposals. Following Lin, Ai, & Doll [5], this study choose the first approach. Starting with a fast one-stage logo detector model followed by improving its accuracy by proposing ensemble technique.

Despite having similarity with the Retinanet object detector framework proposed by Lin, Ai, & Doll [5], this study has the main difference: rather than uses an off-the-shelf model, this study trains several models using a given logo dataset.

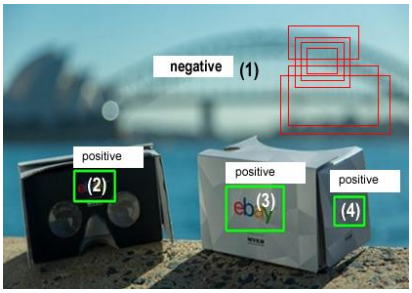

Fig 1. Logo detection positive and negative box as output of one-stage detector.

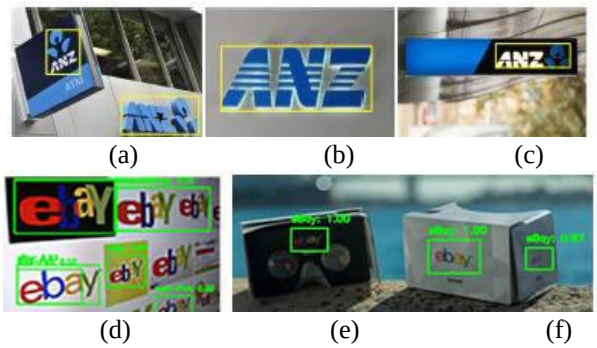

Fig 2. Challenges in logo detection

The rest of this paper is organized as follows. Section 2 describes the main related works in logo detection. Section 3 elaborates the research method. Section 4 explains experiment results and discussion. Finally, Section 5 describes conclusion and future works.

II. RELATED WORK

A. Logo Detection

A vast number of study on logo detection have been reported in the past ten years resulted in a plethora of methods. A prominent study reported by Kalantidis et al. [16], for example, have proposed a multi-scale Delaunay Triangulation and Bag-of-Words model using visual appearance and local geometry as input features. Unfortunately this research is only used a logo set with simple features. Moreover, the proposed method is incapable to predict location and class of the logo of interest. This study was followed by several studies to speed up the process and increase logo detection accuracy.

This study by Pham et al. [17] proposed a multi-step logo detection method. The process starts with segmentation process using outer contour detection and line segmentation techniques. For each segment, the next processes are feature extraction, region scoring and logo localization correcting. Finally, a regression model is used for region scoring. The repeating processes in the pipeline due to process a large number of segments might slow the process down.

The study reported by Sam and Tian [11] simplified logo detection process to become two-step process: logo detection and logo classification steps. The logo detection step aims to: extract variant of Haar-like feature [43], and training a variant of AdaBoost proposed by Freund&Schapire [37] as logo detector. The logo recognition process aims to: segment logo using shape compacting technique [42],extract feature computation, and compute similarity between test and registration samples using the Euclidean distance. Interestingly, in the second process a boosting technique is adopted with simple classifier models. Unfortunately, retrieval technique to find similar logo during logo recognition potentially suffers when volume of the registered logo image is large.

To improve the process of finding region candidates, Li, Chen, and Su [19] proposed sliding window scanning technique that produces a set of candidate windows of various sizes. For each candidate windows, histograms of oriented gradient (HOGE) feature are extracted and Support Vector Machine (SVM) model is then used to recognize logo in the respective window. Finally, ASIFT matching pairs between test and registered logo is used for logo verification. The drawback of this method is high computation load due to high number of window candidates to be processed, some of them might not be usable, produced by scanning window approach. The study by Romberg and Lienhart [20] proposed a method that exploiting aggregated of individual local and spatial neighborhood features. The logo recognition in new images is implemented using image retrieval technique into reference images. Similar with previous method, manual feature extraction step give additional computation workload to overall logo detection process. Thubsaeng et al. [21] proposed logo detection that works in two steps. First, locating candidate regions using CNN model. For each candidate region, Pyramid of Histogram of Gradient (PHOG) feature is extracted.

Finally, SVM model is employed in the second stage as a classifier. The drawback of this methd is mainly inability to perform well when the logo of interest is not fully visible (partial occlusion), or have similar levels of close to other objects.

Due to repeating process of feature extraction followed by logo recognition in each window, the proposed approach to train models in two-stage pipelines is slow.

Iandola et al. [22] explored three main tasks of logo detection and proposed several methods to address each of the respective task namely:

1 Logo classification task to answer which logo is in the image. The proposed classifiers are three variants of GoogleNet previously proposed by Szegedy et al. [39] namely: (1) GoogleNet-Global Pooling (GoogleNet-GP) which is a variant of GoogleNet with global pooling, (2) GoogleNet-Full Classify which is a variant of GoogleNet with multiple softmax activation function off the side of the net, and (3) GoogleNet-Full Inception which is a variant of GoogleNet with an inception layer as its first layer. Unfortunately, none of these models achieve high accuracy due to low-resolution and logo variation in each category.

2 Logo detection without localization task to answer which type of logo if there is a logo in the image. To address this task, [22] suggest to use a combined Fast R-CNN by Girshick [3] with AlexNet [34] model which take in one region proposal per image and remove the bounding box regression functionality. The empiric results show that performance of this method (mean precision only 0.73) is still quite low.

3 Logo detection with localization task to locate and classify logo(s) in the image. To address this task, [22] suggested a two-stage logo detector framework. First stage, localization step using Fast R-CNN [3] model which takes in raw images and region proposals inside the respective images in the form of bounding boxes. Final stage, logo classification using a combined Fast R-CNN [3] with AlexNet [34] or Fast R-CNN [3] with VGG16 [30] models. The empiric results show that Fast R-CNN+Alexnet and Fast R-CNN+VGG16 both achieve 0.74 mean average precision. These results seems the proposed method is not quite achieve high performance to solve the task.

Despite the success of two-stage detectors, many studies have been exploring one-stage object detector models due to its simplicity which lead to fast detectors. For example, a study reported by OverFeat [7] perhaps was one of the first modern one-stage object detector based on deep networks. It followed by Oliveira et al. [24] who proposed a single-stage logo detector framework using Fast Region-based Convolutional Networks (FRCN) model proposed by Girshick [3] that also gave a promising result. In the framework proposed by Oliveira et al. [24], pretrained CNN on large-scale annotated datasets is used for detecting region proposals in raw image and feature extraction. These regions were then classified using a softmax classifier with a Fully Connected layer and postprocessed using non-maxima suppression. Next, classification priority is given to regions based on its confidence value followed by filtering out duplicate regions with non-maxima suppression. Finally, bounding-box regressors is used for refining logo localization.

Interestingly, a study by Lin, Ai, & Doll [5] reported that one-stage detectors can achieve faster detectors that match accuracy of more complex two-stage detectors, such as the Feature Pyramid Network (FPN) by Lin et al. [40] or Mask R-CNN [38] variants of Faster R-CNN [4]. In addition, one-stage detector can address imbalanced dataset during training by dynamically scaled cross entropy loss function.

The study reported by Su et al. [25] aimed to propose a method to generate synthetic logo from a seed logo dataset. The proposed method is tested using several public logo dataset such as: FlickrLogo-32 and TopLogo-10 as input dataset. Moreover, the resulted logo dataset is used as input for logo detector using Faster Region-based Convolutional Networks (Faster R-CNN) model.

B. AdaBoost Model

Ensemble model is a term refers to a meta-algorithm designed to improve stability and accuracy of machine learning algorithm. It is implemented as a combination of multiple learning algorithms. A number of previous studies showed evidences that an ensemble model tend to achieve higher performance than could be obtained from any of the machine algorithm alone. AdaBoost is an ensemble model proposed by Freund and Schapire [37] with the following characteristics: (1) It is an iterative algorithm; (2) In each iteration, it looks at the errors from previous hypothesis to decide what to focus on for the next iteration; (3) It adaptively changes distribution of training data by focusing more on previously misclassified records (hard examples); and (3) The successive hypothesis depend on their predecessors. Following Freund and Schapire [37], AdaBoost algorithm can be described as follows. Consider a set of $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in X$, an instance space and $y_i \in Y$, a label set, as input dataset. Although it can be extended to address multi-class classification problem, the AdaBoost algorithm can be described as follows. For illustration, let's $Y = \{-1, +1\}$

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- **Train weak learner using distribution** D_t.
- **Get weak hypothesis** $h_t : X \to \{-1, +1\}$ **with error**

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- **Choose** $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- **Update:**

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Let $D_t(i)$ represents weight distribution on training example i in t iteration. Intially, $D_t(i)$ is set using a uniform distribution such that $D_t(i) = \frac{1}{m}$. Next, in each iteration, the training dataset was sampled with replacement to form D_t with size m. Accordingly, the algorithm adjusts the weights over the training set.

Since data resampling take the data weight into consideration; hence, each sample generating will have different sampling probability.

The main steps of the AdaBoost algorithm are as follows:

1) Find weak hypothesis $h_t: X \rightarrow Y$ with error ϵ_t on D_t:

$$\epsilon_t = Pr_{i\sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i)\neq y_i} D_t\,(i) \qquad (1)$$

2) Choose a parameter α_t which measures the importance that is assigned to h_t. If $\epsilon_t \leq \frac{1}{2}$ then the value of $\alpha_t \geq 0$. In other words, if the value of ϵ_t gets larger then the value value of α_t gets smaller.

3) Update D_t using the following equation:

$$D_{t+1}(i) = \frac{D_t\,(i)\exp(-\propto_t h_t\,(x_i))}{Z_t}, \qquad (2)$$

The objective of this rule is to increase the weight of examples misclassified by h_t, and to decrease the weight of correctly classified examples. Thus, the weak learner is forced to concentrate on "hard" examples in the training dataset.

4) The final hypothesis H is a weighted majority vote of the T weak hypotheses where each hypotheses h_t is weighted by α_t.

The study reported by Freund and Schapire [37] claimed that the AdaBoost algorithm can be extended to address multi-class classification tasks. Further study reported by Freund, Schapire, & Abe [36], claimed that AdaBoost model has the following favourable characteristics namely: (1) fast, simple and easy to program; (1) has no parameters to tune; (3) requires no prior knowledge about the weak learner and can be flexibly combined with any method for finding weak hypotheses; and (4) based on sound theoretical support that guarantees given sufficient data and a weak learner that can reliably provide only moderately accurate weak hypotheses. Further study by Rokach [41] concluded two main factors contributed to AdaBoost capability to improve the performance accuracy: (1) Error of the final classifier on training dataset as a composite of many hypothesis is smaller than error of each individual classifier on the training dataset; (2) Variance of a combined classifier is significantly lower than the variances of the weak base learner. However, Freund, Schapire, & Abe [36] also described that the drawbacks of Adaboost are as follows: (1) Suboptimal solution, and (2) Sensitive to noisy data and foreign data. The Adaboost model has many variances including Real Discrete Multi-class AdaBoost J. Zhu, *et al.* 2009, AdaBoost [Schapire & Singer 1998], Discrete AdaBoost [36], Gentle AdaBoost and AdaBoost Modest. in this study only limited the discussion in the Real Discrete Multi-class AdaBoost. AdaBoost technique has been adopted to address a number of tasks such as: vehicle logo recognition [11], image classification [12], feature selection and classification [13], MR image classification for brain tumor type [14].

III. RESEARCH METHOD

A. Dataset

This study uses ROMYNY logo datasets [29] consisting of 1,722 samples of logo images consisting of 20 logo classes (brands). The number of samples for each logo class is the following: Adidas-Pict (138), Adidas-Text (121), Aldi (122), Allianz-Pict (132), Allianz -Text (139), Amazon (81), Apple (87), Atletico_Madrid (35), Audi-Pict (79), Audi-Text (19), BMW (84), Burger_king (19), CocaCola (158), eBay (115), Facebook-Pict (11), Facebook-Text (72), FC_Barcelona (65), FC_Bayern_Munchen (78), Ferrari-Pict (62), Ferrari-Text (48).

B. Logo Detector Model

Following Lin, Ai, & Doll [5], the proposed logo detection framework in this study is a single unified network comprises of three main componens namely:

1) Backbone: to compute a feature map from the given input images. In this study, four models are explored namely: VGG19, Resnet50, AdaBoost VGG19, and AdaBoost Resnet50. Each of these models are chosen in this study due to its high accuracy achievement on the challenging ImageNet benchmark. Rather than use an off-the-shelf model as reported by Lin, Ai, & Doll [5], this study trained each of candidate backbones using the given logo dataset. It is expected that the model training produces the trained models that are able to learn only the logo clases of interest.
2) Subnet-1: to compute convolutional logo classification;
3) Subnet-2: to compute convolutional bounding box regression.

Similar to the framework proposed by Lin, Ai, & Doll [5], the objective of model training is finding model parameters that optimized Focal Loss (see Uq. 3) as the objective function.

$$C_{(p,y)} = \begin{cases} -log(p) & if\ y\ =\ 1 \\ -log(1\ -\ p) & otherwhise \end{cases} \qquad (3)$$

In the equation above the notation y ∈ {±1} represents the ground-truth class logo while P ∈ [0, 1] is probability class logo, while CE is Cross entropy for loss binary classification, P_i is the result the logo prediction, and Y_i are labeled (1 if the object belongs to class i, 0 and vice versa). from the CE equation binary classification developed for multi-class classification P_i is defined and CE is developed become the following equation:

$$P_t = \begin{cases} p & if\ y = 1 \\ 1 - p & otherwise. \end{cases} \qquad (4)$$

So if we simplify the loss for binary classification (CE) and multi-class classification P_t equations as below :

$$CE_{(p,y)} \ = \ CE_{(pt)} \ = -\ log\ (P_t) \qquad (5)$$

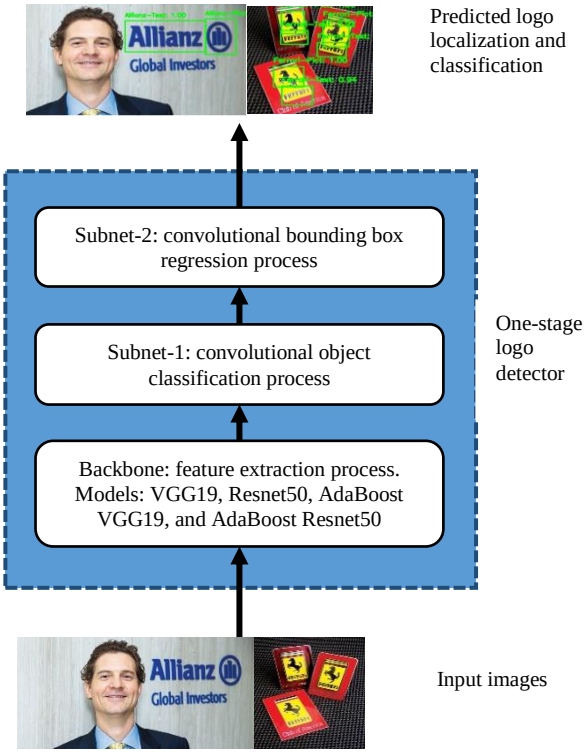

Fig. 3 The Proposed Framework of Logo Detector

Lin, Ai, & Doll [5], improved the focal loss function to overcome the logo imbalance between the foreground and the background class during training with gamma $(1 - P_i)^{\gamma}$ so that the above equation becomes below:

$$FL\,(P_t) = -(1 - P_i)^{\gamma}\ log(P_t) \qquad (6)$$

Following Lin, Ai, & Doll [5], Cross entropy $CE_{(p,y)}$ is changed to Focal Loss $FL_{(p,y)}$ with a value of Y = 2.

IV. Experiment Results and Discussion

A. Logo Detector Training and Validation

In this study, training process of the logo detection models used gradient descent training algorithm, and cross-validated using leave-one-out technique with 70:30 proportions (70 percent training samples and 30 percent testing samples). Performance of each model of interest is measured using: mean Average Precision (MAP), average classification loss, and average regression loss (see Table 1).

Table 1. Summary of Logo Detection Models

No	Model	MAP	Classification Loss	Regression Loss
1	Resnet50	0.56	0.016	0.077
2	VGG19	0.32	0.226	0.269
3	AdaBoost VGG19	0.56	0.017	0.081
4	**AdaBoost Resnet50**	**0.58**	**0.002**	**0.022**

The Table 1 above indicates that performance, measured by MAP, AdaBoost VGG19 and AdaBoost Resnet50 outperforms the single models of interest. Interestingly, although during training process, each model convergence to its optimum parameter values, the average classification loss and average regression loss of AdaBoost Resnet50 outperforms the other models of interest (see Fig. 5 – Fig. 8).

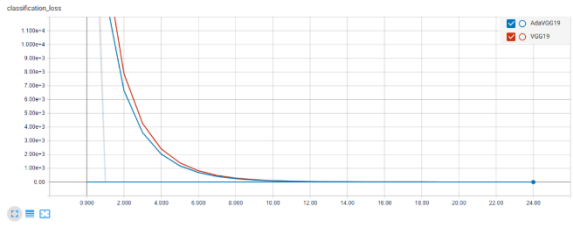

Fig 4. Visualization classification loss of VGG19 and AdaBoost VGG19

Fig 4 shows classification loss of VGG19 and AdaBoost VGG19 models. As can be seen from the Fig. 4, the initial classification loss of VGG19 is $x > 1.4$ and decreased along the epoch dimension, whilst, the initial classification loss of VGG19 is $0.2 < x < 0.1$. At the end of the number of epoch, VGG19 classification loss converged to **0.226** compared to AdaBoost VGG19 classification loss converged to **0.017**.

Fig 5. Visualization Regression Loss of VGG19 and AdaBoost VGG19

Fig. 5 shows regression loss of VGG19 and AdaBoost VGG19 models. As can be seen from the Fig. 5, the initial regression loss of VGG19 is x > 3.5 and decreased monotonically along epochs, whilst, the initial classification loss of VGG19 is 0.22 < x < 0.20. At the end of the number of epoch, VGG19 regression loss converged to 0.269 compared to AdaBoost VGG19 regression loss converged to 0.081.

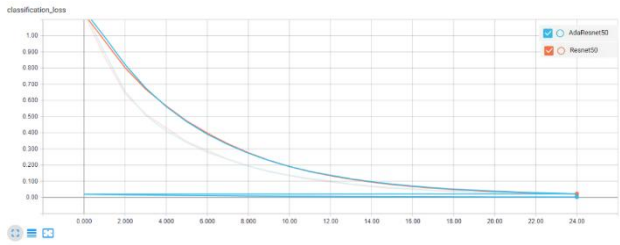

Fig 6. Visualization classification loss Resnet50 and AdaBoost Resnet50

Fig. 6 shows classification loss of Resnet50 and AdaBoost Resnet50 models. As can be seen from the Fig. 6, the initial classification loss of Resnet50 is $x > 1.2$ and decreased along the epoch dimension, whilst, the initial classification loss of Resnet50 is $1.0 < x < 0.9$. At the end of the number of epoch, Resnet50 classification loss converged to **0.016** compared to AdaBoost Resnet50 classification loss converged to **0.002**.

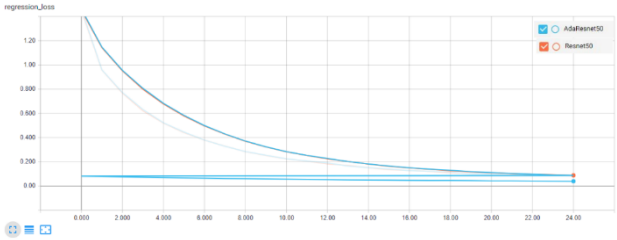

Fig 7. Visualization Regression Loss Resnet50 and AdaBoost Resnet50

Fig. 7 shows regression loss of Resnet50 and AdaBoost Resnet50 models. As can be seen from the Fig. 7, the initial regression loss of Resnet50 is $x > 1.4$ and decreased monotonically along epochs, whilst, the initial regression loss of Resnet50 is $1.2 < x < 1.0$. At the end of the number of epoch, Resnet50 regression loss converged to **0.077** compared to AdaBoost Resnet50 regression loss converged to **0. 022**.

B. Logo Detector Testing

To test the trained logo detection, a number of logo image samples were used as testing dataset; Mean Average Precision (MAP) as used as model performance. The results of model testing is summarized in Table 3. Some samples of testing results can be seen in Fig. 8.

Table 3. Summary of Average Testing Performance (MAP)

Resnet50	VGG19	AdaBoost VGG19	**AdaBoost Resnet50**
0.94	0.97	0.99	**1.00**

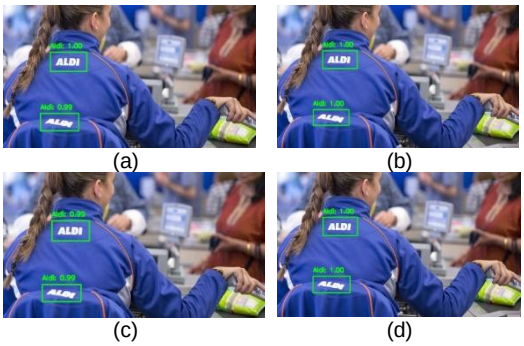

Fig 8. Some samples of testing images using: (a) VGG19, (b) AdaBoost VGG19, (c) Resnet50, and (d) AdaBoost Resnet50

V. CONCLUSION

Logo of a brand is an important asset as it expresses character of the company that owns the respected brand. In addition, logo is designed to reflect the purpose of a product so that it makes easier for customers to recognize and remember the product. The advent of machine learning methods and proliferated of logo images in the past decade has made automated logo detection an interesting computer vision problem with wide potential applications such as: identifying goods, services, corporations, contextual ad placement, validation of product placement, and online brand management.

With the rapid increase in the importance of logos for identifying goods, services, and organizations, logo recognition has been an active area of research in the field of computer vision. The active studies in logo recognition research field has resulted in a plethora of logo detector methods.

This paper presents a novel one-stage logo detector framework in which the backbone of the detector is a deep neural network model. Unlike the prominent Resnet framework proposed in which the backbone is an off-the-shelf model, the proposed framework in this study proposed a backbone which is trained supervisedly using gradient descent training algorithm. The experiment results showed that AdaBoost Resnet50 (0.58 MAP) as the logo detector backbone outperforms Resnet50 (0.56 MAP), VGG19 (0.32 MAP), and AdaBoost VGG19 (0.56 MAP).

REFERENCES

1 Cambridge Dictionary, "Meaning of logo" accessed from : https://dictionary.cambridge.org/ on 17 Mei 2019.
2 P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.
3 R. Girshick, "Fast R-CNN," arXiv.org e-Print Arch., 2015.
4 S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Nips, pp. 1–10, 2015.
5 T. Lin, F. Ai, and P. Doll, "Focal Loss for Dense Object Detection."
6 Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.
7 P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In CVPR, 2001.
8 Y. Bao, H. Li, X. Fan, R. Liu, and Q. Jia, "Region-based CNN for Logo Detection," Proc. Int. Conf. Internet Multimed. Comput. Serv. - ICIMCS'16, pp. 319–322, 2016.
9 W. Liu et al., "SSD : Single Shot MultiBox Detector," arXiv, pp. 1–15, 2016.
10 D. Impiombato et al., "You Only Look Once: Unified, Real-Time Object Detection," Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip., vol. 794, pp. 185–192, 2015.
11 K. Sam and X. Tian, "Vehicle Logo Recognition Using Modest AdaBoost and Radial Tchebichef Moments," Proc. 2012 4th Int. Conf. Mach. Learn. Comput. IPCSIT, vol. 25, pp. 91–95, 2012.
12 "IMAGE CLASSIFICATION BASED ON THE BAGGING-ADABOOST ENSEMBLE," Compute, no. 1, pp. 1481–1484, 2008.
13 R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review," Phys. Procedia, vol. 25, pp. 800–807, 2012.
14 A. Minz and C. Mahobiya, "MR image classification using adaboost for brain tumor type," Proc. - 7th IEEE Int. Adv. Comput. Conf. IACC 2017, pp. 701–705, 2017.
15 S. Yang, L. Chen, T. Yan, Y. Zhao, and Y. Fan, "An ensemble classification algorithm for convolutional neural Netwok base on adaboost," pp. 401–406, 2017.
16 Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, "Scalable triangulation-based logo recognition (Flickr 27)," Proc. 1st ACM Int. Conf. Multimed. Retr. - ICMR '11, p. 20:1-20:7, 2011.
17 T. A. Pham, M. Delalandre, and S. Barrat, "A contour-based method for logo detection," 2011.
18 J. Revaud and C. Schmid, "Correlation-Based Burstiness for Logo Retrieval Categories and Subject Descriptors : Keywords :," pp. 965–968, 2012.
19 K. Li, S. Chen, and S. Su, "Logo detection with extendibility and discrimination," 2013.
20 S. Romberg and R. Lienhart, "Bundle Min-Hashing for Logo Recognition," ACM, 2013.
21 W. Thubsaeng, A. Kawewong, and K. Patanukhom, "Vehicle logo detection using convolutional neural network and pyramid of histogram of oriented gradients," 2014 11th Int. Jt. Conf. Comput. Sci. Softw. Eng., pp. 34–39, 2014.
22 F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, "DeepLogo: Hitting Logo Recognition with the Deep Neural Network Hammer," arXiv Prepr. arXiv1510.02131, pp. 1–12, 2015.

23 R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 580–587, 2014.
24 G. Oliveira, X. Frazão, A. Pimentel, and B. Ribeiro, "Automatic graphic logo detection via Fast Region-based Convolutional Networks," Proc. Int. Jt. Conf. Neural Networks, vol. 2016–Octob, pp. 985–991, 2016.
25 H. Su, X. Zhu, and S. Gong, "Deep learning logo detection with data expansion by synthesising context," Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017, pp. 530–539, 2017.
26 R. E. Schapire, "The boosting approach to machine learning: an overview," Nonlinear Estim. Classif., vol. 171, pp. 149–171, 2003.
27 J. Wu and J. M. Rehg, Ensemble Machine Learning. 2012.
28 Q. Song and N. Kasabov, "Dynamic evolving neuro-fuzzy inference system (DENFIS): on-line learning and application for time-series prediction," Proc. Sixth Int. Conf. Soft Comput., vol. 10, pp. 696–702, 2002.
29 Frank Fotso, "Application of py_faster_rcnn in logo detection task: ZF & VGG16." .
30 K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Int. Conf. Learn. Represent., pp. 1–14, 2015.
31 Z. He, "Deep Residual Learning for Image Recognition," arXiv.org e-Print Arch., vol. 7, no. 3, pp. 171–180, 2015.
32 R. Lienhart, and J. Maydt. An Extended Set of Haar-Like Features for Rapid Object Detection. IEEE International Conference on Image Processing. 2002, 1 (1): 900-903.
33 J. G. Leu. Shape Normalization Through Compacting. Pattern Recognition Letters. 1989, pp. 243-250.
34 Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In NIPS, 2012.
35 J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. IJCV, 2012.
36 Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139,August 1997.
37 37 Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14(771-780), 1612.
38 He, K., & Girshick, R. (2017). Mask R-CNN.
39 Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Du- mitru, Vanhoucke, Vincent, and Rabinovich, An- drew. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.
40 Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature Pyramid Networks for Object Detection. Retrieved from http://arxiv.org/abs/1612.03144 Rokach, L. (2010). Pattern classification using ensemble methods (Vol. 75). World Scientific
41 J. G. Leu. Shape Normalization Through Compacting.
42 Pattern Recognition Letters. 1989, pp. 243-250.
43 R. Lienhart, and J. Maydt. An Extended Set of Haar-Like Features for Rapid Object Detection. IEEE International Conference on Image Processing. 2002, 1 (1): 900-903.
44 C. L. Zitnick and P. Doll´ar. Edge boxes: Locating object proposals from edges. In ECCV, 2014.
45 O. H. Jafari and M. Y. Yang, "Real-Time RGB-D based Template Matching Pedestrian Detection," 2016.
46 N. Dalal, B. Triggs, and D. Europe, "Histogram Of Oriented Gradients For HumanDetection.pdf," 2005.
47 Fischer, P., Dosovitskiy, A., & Brox, T. (2014). Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT. ArXiv, 1–10. Retrieved from http://arxiv.org/abs/1405.5769

AUTHORS PROFILE

Sarwo, Student Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

Dr. Ir. Yaya Heryadi, M.Sc, Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta , Indonesia 11480.
https://scholar.binus.ac.id/d3803/dr-ir-yaya-heryadi/

Prof. Dr. Ir. Widodo Budiharto, S.Si., M.Kom., IPM, Computer Science Study Program, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480
https://scholar.binus.ac.id/D2637/widodo-budiharto/

Prof. Dr. Ir. Edi Abdurachman, MS., M.Sc. Computer Science Department, BINUS Graduate Program – Doctor of Computer Science,
Bina Nusantara University, Jakarta, Indonesia 11480
https://scholar.binus.ac.id/D0636/edi-abdurachman/