Object Detection Algorithms: A Brief Overview

Parth Joshi[1], Chirag Sehra[2]
[*]*Northern India Engineering College, GGSIPU University, Delhi, India*
[1]joshi.parth97@gmail.com
[2]chiragsehra42@gmail.com

***Abstract*— This paper aims to review the various groundbreaking algorithms that have been efficacious in the vast timeline of object detection and recognition algorithms. We will compare and contrast the various techniques that have been used in the past which were based on the approach of simple binary classifiers with those which are currently in use which are in concordance with the neurological structure of brain known as deep learning. We will also propose techniques which have either been in development or have been suggested by various researchers as to what techniques should be employed which will help shape the future of computer vision algorithms by making them more efficient and accurate.**

***Keywords*— CNN, performance, gradient, accuracy, future techniques**

I. Introduction

Object detection algorithms have been evolving at a very rapid pace and have become better than humans in detecting and recognisingparticular objects in a given frame for the most part. These algorithms have been the frontrunners in the computer vision domain and have achieved various strides in their timeline. Broadly, these algorithms can be used for detection of objects, or recognising various entities in a particular picture. It has many practical applications ranging from face detection, visual search engines, autonomous cars, security and surveillance systems to interplanetary anomaly detection, satellite imagery, galaxy simulation engines and many more.

Generally, these algorithms work by finding out the particular object by comparing the pixel values of the object and comparing them with the particular picture frame using various calculation methods. There have been various object detection algorithms which have been proved to be viable for industrial uses. These algorithms have evolved from binary classified based approach to learning based approach. We review all such algorithms and also propose techniques for future object detection algorithms.

II. Past Algorithms

Object detection algorithms have been present for a very long while. Generally, these algorithms were made to be task-specific and provided reasonable accuracy for their time. Still, these algorithms are not completely obsolete as for basic level work, these algorithms can suffice as they do the job pretty well as well as have low cost of operation as compared to deep learning approach. There is also much less data dependency as the algorithm need not be generic and provide sufficient output for the particular problem.

A. Viola-Jones Algorithm

This algorithm was one of the breakthrough algorithms which was devised in 2001. It was mainly used for the purpose of face detection but also was applicable for general purpose object detection algorithm. It had four components which included Haar Feature Selection, Creating an Integral Image, Adaboost Training and Cascading Classifiers. The algorithm checks for various features in a face like eyes, nose, mouth etc and computed cascades for the faces and compares them with the Haar features to check for faces in an image. Due to this reason, the images needed to be perfectly oriented with the face being frontal upright for the face to be detected.

This algorithm had very high detection rates and very low false positives. Also, the recognition of faces was not quite developed as compared to detection rates thus reducing practical implications. That saying, it was still very prominent in the history of object detection algorithms as it made the algorithm easy to implement using computer vision libraries like OpenCV and paved the way for many more object detection algorithms.

B. Histogram of Oriented Gradients (HOG)

This algorithm interprets strong low level features that are based on histograms of oriented gradients (HOG). It is an alternative to exhaustive search but is still based upon the approach of hardcoded features like Viola-Jones method. It converts the image in grayscale and then finds the object to be found pixel by pixel in a particular frame. It compares each pixel with its surrounding pixels with respect to the intensity of darkness. By doing this, it can create a map of the gradients of the pixel intensity variation. These gradients can help us locate various features in an image.

HOG method can to be used to identify various objects in a particular image by computing the gradient orientation in localised portions of the image. This is done by the use of feature descriptor which can be used to highlight the parts of the image which are required and remove other background noise. The feature descriptor converts an image of size width x height x channels to a feature vector / array of length n. The feature vector produced by the algorithm when fed into an image classification algorithm like Support Vector Machine (SVM) produces good results.

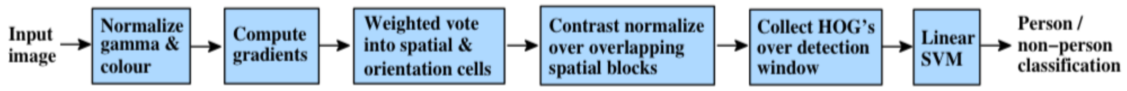
Input image
Normalize gamma & colour
Compute gradients
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
Linear SVM
Person / non−person classification

Fig. 1 Working of a HOG setup [2]

With the help of these feature descriptors, the gradients that were calculated for nearby pixels for each particular pixel, are assigned direction and magnitude gradients. These are stored in the form of histograms for easier representation and calculation. These gradients then help find out the various boundaries in a particular frame with the help of which we can find out the required object. A sudden change in the value of the gradient at the particular pixel accounts for a corner or an edge and thus classification and detection can be done.

III. Present Algorithms

The algorithms which are in use currently are all based upon the deep learning approach. These algorithms tend to make the program "learn" about various different objects so that they can be identified from a given image. Rather than the algorithms which were used in the past in which task-specific searching and locating was done, deep learning approach is much less brute forced. It models the functioning of the brain by taking into account the interactions between a stimuli and neurons. We provide weights which are types of hyper parameters given to each connection between the various layer of neurons. These help decide the outcome of a particular computation by looking into the weight of the output layer's neuron. These are known as neural networks. Many different implementations of neural networks have proven to be state-of-the-art time and again.

A. Convolutional Neural Networks (CNN's)

A CNN usually takes a 3rd order tensor as its input i.e. an image with its breadth W and height H accompanied by the color channel. In a similar way, higher order tensors are handled by these Convolutional Neural Networks. This input is progressively preprocessed by a sequence of steps which is called a layer, which could be a convolutional layer, a pooling layer, a fully connected layer, a flattening layer etc.

$$\boldsymbol{x}^1 \longrightarrow \boxed{\boldsymbol{w}^1} \longrightarrow \boldsymbol{x}^2 \longrightarrow \cdots \longrightarrow \boldsymbol{x}^{L-1} \longrightarrow \boxed{\boldsymbol{w}^{L-1}} \longrightarrow \boldsymbol{x}^L \longrightarrow \boxed{\boldsymbol{w}^L} \longrightarrow \boldsymbol{z}$$

Fig. 2 Assumed skeletal architecture of CNN

This equation illustrates how a CNN runs layer by layer in forward propagation. Here x^1 is an image of order 3 tensor. The first box represents the first layer which processes the image. Parameters of this layer is represented by tensor w^1. After processing through the first layer, the output of first layer, x^2, acts as input to the second layer processing. This processing continues until all the layers in CNN are finished with output x^L. However, to make it more accurate, one additional layer is added for backward error propagation error. Let us suppose t is the objective target (ground-truth) value for inputx^1. To measure the discrepancy, a cost or loss function is used between the CNN prediction x^L and the target t. For example, a simple loss function could be:

$$z = \frac{1}{2}(t - x)^2$$

During Forward run, we can use the model for prediction assuming all the parameters of CNN model w^1,......,w^L are learned. Starting from the input x1, and propagating through all the layers upto x^L which estimates the posterior probabilities of all categories, the output of CNN prediction is taken as argmax x_i^L .

Similar to other learning systems, the parameters of a CNN model are optimized by minimizing the loss function z. The CNN network runs in both directions during training process. We first run the network in the forward pass to get x^L to achieve a prediction using the current CNN parameters. Then we compare the prediction with the target t corresponding to x^1, that is, continue running the forward pass till the last loss layer. The loss z supervises how the parameters of the model weights should be updated. There are many loss functions available that can be used, for example, categorical cross entropy, cosine proximity etc. which are minimized by various optimization algorithms available such as Stochastic gradient descent, AdaGrad, Adam etc. Parameters are modified as:

$$w^i \Leftarrow w^i - \eta \frac{\partial z}{\partial w^i}$$

During forward propagation, a kernel slides over the whole image in equal and finite strides. At each point, the product of the element of kernel and input image is taken which overlaps and this result is summed up to obtain the output of current location. This helps in storing the localized features of the input image. During back propagation, weights and deltas are updated which are calculated during forward propagation. Hence Convolutional Neural Networks use weight sharing strategy which leads in training less number of parameters. Every time, the deviations in forward and backward propagations differ depending on the layer one is propagating through.

B. R-CNN

R-CNN stands for Region Based Convolution Neural Network and is a method that depends on external region proposal system. Rich features are computed by a convolutional neural network. RCNN has better performance than other ensemble methods and feature types. It is an efficient matching algorithm for deformable based models i.e. pictorial structures. R-CNN takes an input message and extracts region proposals and computes features using large convolutional neural network and then classifies the image.

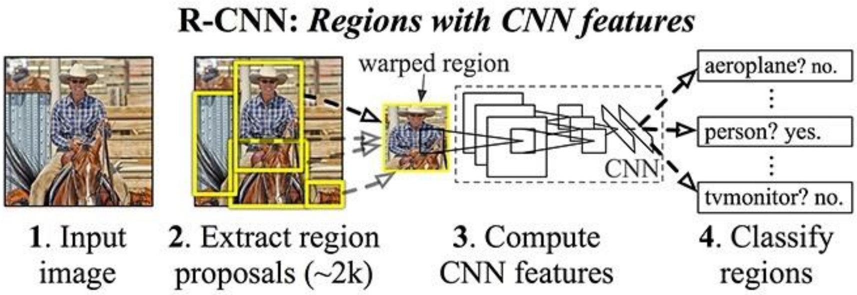

Fig. 3 Rich feature hierarchies [5]

C. Fast R-CNN

It addresses the drawback of high evaluation cost by evaluating most of the convolutional layers a single time per image According to the authors, training speed is increased up to 9 times and testing about 213 times without losing accuracy. The major advantages of Fast R-CNN over previous state-of-the-art techniques were that is training in Fast R-CNN is a multi-stage pipeline. Training is expensive in space and time in R-CNN. It is observed on testing that object detection was slow on R-CNN. Performing a ConvNet forward pass for each object proposal without sharing computation makes R-CNN work slowly.

A Fast R-CNN takes an input image and a set of object proposals. It produces a convolutional map by first processing the image through various convolutional and pooling layers and then fixed length feature vector is extracted from the feature map for each proposals region of interest (ROI). Each of these feature vectors is then fed to a succession of fully connected layers that outputs the K object classes by bounding boxes.

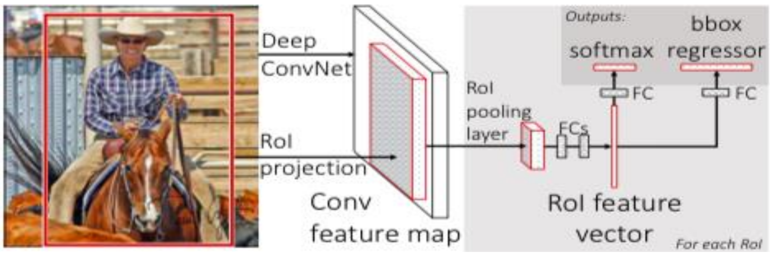

Fig. 4 Fast R-CNN modifications [6]

D. Faster R-CNN

It has two networks that are one region proposal network (RPN) for generating region proposals and a network for detecting the object using these proposals. The main difference with Fast R-CNN is that it uses selective search to generate region proposals. As RPN shares the most computation with object detection, the time cost of generating region proposals is much smaller in RPN than selective search. The region proposal network produces a cluster of boxes that are inspected by a classifier or a regressor to check the occurrence of objects.

After RPN, we get different sizes of proposed regions. Different sized regions mean different sized CNN feature maps. It's difficult to make an efficient structure to work on features with different sizes. A region of Interest Pooling simplifies the problem by reducing the feature maps to the same size. Fixed number of roughly equal regions(say k) are produced when input feature map is divided with ROI splitting, and then Max-Pooling is applied to it. Therefore the output of ROI Pooling is always k regardless the size of the input. With the fixed ROI Pooling outputs as inputs, lots of choices are available for the architecture of the final regressor and classifier.

IV. Future Algorithms

As of now, Convolutional Neural Networks (CNNs) and their variations have been the state-of-the-art approach to classifying images. In CNNs, at each layer, it accumulates sets of features. It starts off by finding edges, then shapes, then actual objects. However, there is a loss in spatial relationships of these features. Due to such losses, Convolutional neural networks are susceptible to white box adversarial attacks i.e. implanting a secret pattern into an object to make it look like something else. The algorithms which will be in fruition in the near future will be mostly aimed towards reducing such complexities, increasing the performance as well as finding ways to reduce data dependency to make more efficient and practical systems.

A. Capsule Networks

Capsule Networks developed by Geoffrey E. Hinton at Google Brain gives us the ability to take full advantage of the spatial relationship. They introduce a new building block that is used for better model hierarchical relationships of a neural network. According To Hinton, Artificial neurons output a single scalar. Each kernel in a CNN replicates that same kernel's weights across the entire input and then output a 2D matrix. Then, all these 2D matrices are stacked on top of each other to produce the output of a layer.

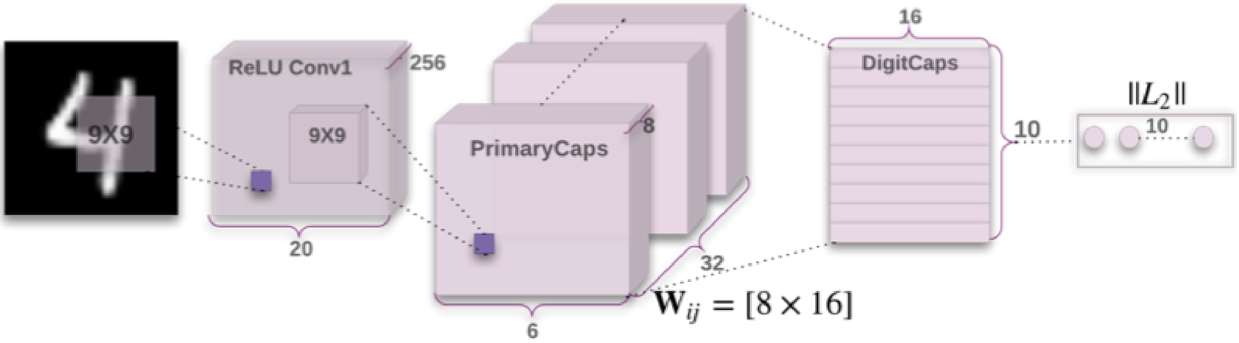

Fig. 5. Dynamic routing between capsules [7]

Then, invariance in the activities are achieved by neurons by the means of max pooling where largest number is selected in each region out of output 2D matrix.. Invariance means that by changing the input a little, the output still stays the same.The output signal of a neuron is the activity max pooling loses valuable information and also does not encode relative spatial relationships between features which makes it inefficient. We can use capsules instead because they will encapsulate all important information about the state of the features they are detecting in a form of a vector. The probability of detection of a feature is encoded as their length of output vector and the state of the detected feature is encoded as the direction in which that vector points. Thus, when detected feature moves around the image or its state changes somehow, the probability still remains the same, but what changes is its orientation. This is what Hinton refers to as activity equivariance. The input to the capsule networks is a vector. Operations performed in a capsule network on the network are namely Affine Transform, Weighting and Non-Linear Activation.

Affine Transform: $\hat{u}_{j/i} = W_{ij}.u_i$

Weighting And Sum: $s_j = \sum_i c_{ij}\, \hat{u}_{j/i}$

Nonlinear Activation: $v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|}$

Capsules expands the neuron design to its vector form to allow for more powerful representation capabilities. It also introduces matrix weights to encode the important hierarchical relationships between features of different layers.

B. NEIL: Never Ending Learning Image Learner

Having a great base of weakly supervised images over the internet can be retrieved and small curated datasets can be created such as PASCAL and ImageNet. Without the intervention of humans, there is a great real possibility to learn from thousands of categories and sub categories of images. Pace of visual systems can be accelerated by thorough understanding of image and text associated with it which is available free online. Initial attempts of creating systems like NEIL: Never Ending Image Learner and LEVAN: Learning Everything about Anything are future methods which will be involved with object detection. NEIL is an effort to develop a visual structure with least human effort of labelling the data. It uses a knowledge base of relationships between categorized examples of Objects (e.g., dog, cat, car), Scenes (e.g., hill, beach, land) and Attributes (e.g., blue, big). These relationships are of four types:
(1) Object-Object (e.g., Grading/Membership), (2) Object-Attribute (e.g., Shape/Color/Appearance), (3) Scene-Object (e.g., Cycle is found in Garage) , (4) Scene-Attribute (e.g., Bridge is Broad).
NEIL is a learner which learns iteratively to add new knowledge at each step stride to refine existing knowledge. Steps involved in one iteration of NEIL are:

1. Visual Cluster Discovery: This step involves the building of classifiers for visual categories using semi-supervised algorithms. Images retrieved and used on basis of text or keyword fails because of some reasons that are : (1) Eccentricity in retrieved images; (2) Lexical ambiguity of multiple integrations of search text; (3) Diversity of high-class variation; (4) Localization of objects and these problems are solved using clustering objects like (1) Selection of multiple detectors (2) Clustering based on Appearance.
2. Training Detectors: In here, a SVM detector is trained over each subcategory using three-quarters of images. The remaining set of images is used as validation set for adjustment.
3. Relationship Discovery: Relationships in NEIL are derived completely from the collected images.
4. Adding New Instances and Retraining of Detectors: After the initial learning of relationships between objects and scenes, new instances of different objects and scenes. Detectors are re-trained on the labelled data where the labelled data is updated with the new instances formed. More relationships are then derived from the new classifiers which are used in turn to label more data.

C. LEVAN: Learning Everything About Anything

LEVAN is used to capture intra-concept variance based on learning of exhaustive semantically rich models. It is a method which helps in explaining all the appearance variations (i.e., actions, attributes, interactions etc.) and trains detections models for it. It is a fully automated project. This project aims on learning depth as well as breadth of knowledge available online. To discover the variance of vocabulary, LEVAN uses Google Books Ngrams, which is extensive and content-specific. Vocabulary discovery and model learning are tried to carry out simultaneously. Thus, external human annotation is not required. This increases the flexibility and scalability of the project. Steps of LEVAN are described below:

(1) Discovering the Vocabulary of Variance: For a given concept, dependencies such as noun, adjective, verb or adverb are used. This method also finds a lot of non-visual words such as ‘particular cat’, ‘last cat’ which helps adding noise to the system.
(2) Training Detectors: After cleaning the data, there maybe some noise left in data. These noisy components are detected and processed by component detector.
(3) NEIL doesn’t use textual information to improve object detection while LEVAN does. NEIL uses clustering followed linear classification while LEVAN learn a DPM for each ngram. Inter-class variance is modelled and generalized in LEVAN while NEIL uses clustering.LEVAN can be useful in Natural Language Processing for gathering semantic meanings and producing paraphrases. It has great and immense applications in fine-grained image search, object detection and its segmentation.

V. CONCLUSION

Eight different concepts of object recognition approaches were presented in the paper. Viola-Jones Algorithm was based on detecting features with Haar features and it was seen that it had high detection rates and very low false positive rates. Gradient Orientation technique, Histogram of Oriented Gradients, was assessed and can be clearly seen why it was widely recognised as a groundbreaking algorithm even leading to derivation of many modern algorithms.

Coming to deep learning techniques such as CNNs, these have been hugely popular and successful in the field of object recognition and image processing. RCNNs produce even better results than CNN, but are quite slow because they require forward pass of the CNN for every single region proposal. Also, for a single image, three sub-models have to be trained separately- the CNN to generate image features, the classifier to predict the classes and the regression model to tighten the bounding boxes.Fast RCNN came with the use of ROI(Region of interest) and overcame the major problems of RCNNs by combing all the three extractor, classifier and regressor in the same framework. The ROI computation was slow and was sped up in another altercation of RCNN known as Faster RCNNs. This was done by using those same CNN results for region proposals rather than a separate selective search algorithm which made only one CNN need to be trained.

Latest research and concepts hunches the idea of dynamic routing between the capsules containing group of neurons.With the growing industries and ginormous amounts of data being generated, the knowledge base is also increasing which will use techniques like NEIL and LEVAN for object detection where knowledge base will have dynamic relationships and intra-concept variances based learning, thus making the whole system complete and highly efficient.

ACKNOWLEDGMENT

We would like to thank all the organisers and conveners of the conference as well as Northern India Engineering College for providing us the opportunity to submit our paper. We would also like to specially thank Ms. Gunjan Chugh, Assistant Professor of IT department for providing her valuable feedback all throughout the process and also helping us with the formatting of the paper.

REFERENCES

1. Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.
2. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
3. Wu, Jianxin. "Introduction to convolutional neural networks." National Key Lab for Novel Software Technology. Nanjing University. China (2017).
4. Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." 2014.
5. Girshick, Ross. "Fast r-cnn." arXiv preprint arXiv:1504.08083 (2015).
6. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.
7. Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." *Advances in Neural Information Processing Systems*. 2017.
8. Chen, Xinlei, AbhinavShrivastava, and Abhinav Gupta. "Neil: Extracting visual knowledge from web data." *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013.
9. Divvala, Santosh K., Ali Farhadi, and Carlos Guestrin. "Learning everything about anything: Webly-supervised visual concept learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.