International Journal of Engineering & Technology, 7 (2.8) (2018) 507-512

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

An Efficient CNN a deep learning approach applied on the image matching context

V. Naga Bushanam, Ch.S atyananda Reddy

Department of CSSE, Andhra University, A.P, India

Abstract

Image matching is a quite challenging task to identify matching images in the data. There are multiple methods in computer vision techniques such as histogram-based algorithms, colour or edge based algorithms, textual based features, SIFT and Surf algorithms which will help to identify similar images. Here in our paper we are addressing an industrial problem to provide the better solution where US multinational courier delivery service facing challenges in delivering the products where labels/tags and bar codes of the products are missed while delivering to the customers and customers comes with the product image and with some information about the product. The job is to map the user or customer product information with the existing missed products. The advances in computer science and availability of GPU Machines, the problem will be addressed, and solutions can be automated using deep learning approaches. The paper describes the solution of matching the solution accurately and comparing the solution with the existing classical computer vision algorithms.

Keywords: *Deep Learning, Computer Vision, Image Matching, Image Search.*

1. Introduction

This paper addresses the image matching algorithms which we developed using deep learning approaches as a part of solving challenges facing by the multinational courier services. The courier services shifts the products from one place to another place while shifting there are many products where the barcodes are missing, addresses are missing etc. The courier company will open the products and takes the images of the products and other information of the products such as product time, class (Apparel, Electronics, and Appliancesetc.) and keep the information into the databases. Whenever the customer comes with their missed product information such as image or type of the product with any communication channel (Via email, personal etc.) as the courier company needs to identify the product with their database. As per the discussion the company will maintain the one lakh transactions per day and as we need to automate entire matching process of the products. As currently the courier company handling this matching process manually and there will be a huge human error involved in this process. In this paper, we are addressing how the deep learning approaches will help in matching the customer images from the database of missed products efficiently. We have tried with the classical computer vision algorithms such as histogram methods, comparing the textual; colour features and their results are not impressing. Finally, we use deep learning approaches on GPU Machines and retrieved image features using CNN (Conventional Neural Networks) and computed the closed images matching with customer's vs the missed images in the database. The entire system architecture and algorithm details are provided in subsequent sections.

2. Literature Survey

There are quite few papers which address deep learning approaches for segmenting in the images. The paper [1] discusses training the Conventional Neural Network using only unlabelled data to discriminate between a set of surrogate classes. Each surrogate classes are formed by applying a variety of transformations to a randomly sampled seed image patch. The authors found this feature learning algorithm is surprisingly successful an applied to detect image objects. The authors in the paper [2] describe the hundreds of thousands of unlabelled videos from the web to learn visual representation of those videos it helps tracking visually provides the super vision that means two patches connected by a track should have similar visual representation in deep feature space since they probably using deep dynamic neural networks for multimodal gesture segmentation and recognition [3]. The author proposed semi supervised hierarchical dynamic framework based on Hidden Markov model (HMM) for simultaneous gesture segmentation and recognition where skeleton joint information, depth and RGB images, are the multimodal input observation. The authors [4/6] described the different applications of deep learning approaches in image classification and clustering approaches.

3. System Architecture

The overall end to end System architecture is as shown in the Figure 1 and each component of the system is explained as given below.

A. Image Capture Device

It can be any image capturing device ranging from mobile phone camera to raspberry pi module camera. Image captured can be format JPEG or PNG. User need to insure few Things while capturing the image

- Focused image of "Object of interest",i.e,
- Object should be clearly visible.
- Object should be completely inside the image.
- Object should not Covered by other objects.
- Object should cover majority of image(at least 40% of total image size.)

User should avoid adverse environment like very bright lights in the background, capturing image while inn motion (blur images).

B. User Task Request Handler

This module handles all user queries. Each user query gets queued up in the database and is processed sequentially. It also stores the image sent by user to database storage and sends a copy of the same for processing.

C. Data Storage

As Suggested by the name of the module, this module stores user query, images, processing logs for future references. All the reference images and image data is also stored in the same module.

D. Object Detection and Image Match task Controller

Within this block, the user image is being processed for object detection and image matching purposes. This module receives the user test images and outputs similar images. Processing steps and algorithm part is described in subsequent part of the paper.

E. Output Image Handler

Output of our system is a set of "K" images, which matches closely with the user input image. Number of output images (k) can be sat by user for each query. Output can be displayed on the desktop or mobile dashboards and can be sent to the user/customer over e-mail.

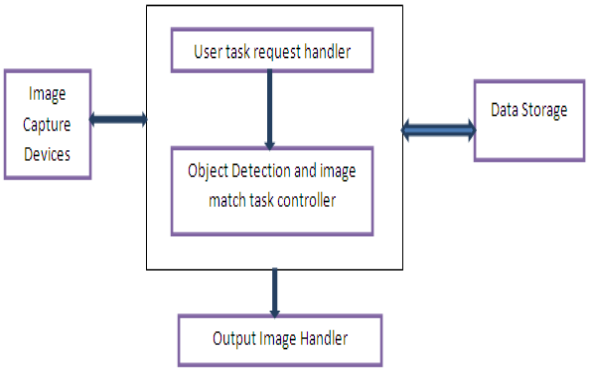

Fig. 1: System Architecture

4. Image Classification-Deep Learning

During recent years, Deep learning algorithms are being extensively experimented for image classification and object detection use cases. Object detection capability of deep neural networks is being developed to recognize different objects in uploaded images. In this paper, we will use Conventional Neural Network image features to match the images and the algorithms details are given below.

A. Image Repository

Images are collected and sampled for 1000 object categories 1000 classes were selected as the first step which was followed by image collection from the web using web scraping tools / web crawler . To prevent output model from data imbalance problems, we made sure to collect approximately equal number of images for each class. Although training procedure of CNN internally considers cross validation and testing, we still divide the dataset into Training data and Test data. This test dataset was generated to test the end to end image matching module.

B. Creating training data

To train the convolution Network for this scenario, the object images from the image repository are used along with appropriate labels. We randomly samples 90% images for training dataset and10% images for validation dataset. All the images are resized to size 227*227 (width*height) pixels during sampling.

Sampling provides training set and validation set as:

$$T=\{(l_1,t_1),(l_2,t_2),\ldots\ldots(l_n,t_n)\}$$

$$V=\{(l_1,v_1),(l_2,v_2),\ldots\ldots(l_n,v_n)\}$$

Where:

l_i = image label, i=1,2,…n

t_i = training image pixel data (one dimensional vector of size 3072 (227*227*3)),i=1,2,…n

v_i = validation image pixel data(one dimensional vector of size 3072(32*32*3)),i=1,2,…n

This method uses Caffe framework [8] to implement convolution neural network (CNN) model. This model will be used in the later stage for detection of objects and arriving features inside images uploaded by user.This paper utilizes python implementation [3] for CNN.

C. Model Description

Convolution neural network architecture, learning algorithm and hyper parameter setting description is described below:

1) Convolution neural networks

Convolution neural networks (CNN) trained via back propagation is being utilized in the current approach to detect different objects present inside each image. Convolution training can be used both in supervised and unsupervised methods .This approach trains the conventional network in supervised method. Conventional Neural Network consists of multiple layers of neurons (input, hidden and output layers), which are arranged in 3 dimensions: width, height, depth.

2) Network Architecture

We used AlexNet CNN network for object detection. This network consists 5 Convolution layers with SoftMax layer as last layer. Model was trained via back propagation. As Training a CNN model with N classes is highly computational. GPU servers were utilized for the same. Loss weights were initialized based on the pre-trained ImageNet model, which was trained on 1000 categories. Batch size of 256 was chosen for each iteration of training.

It uses three main types of layers in neural networks such as Convolution layer, pooling layer and Fully Connected layer [6].

INPUT [227*227*3] – This layer will hold raw pixels values of the image in this case any image of width 227, height 227 and with three colour channels R,G,B.

Convolution layer – This layer applies convolution operator on input data.

RELU (Rectified-Linear and Leaky) layer –It applies activation function on the input data.

POOL Layer – This will perform a down sampling operation. This layer is conducted to the Conv layer.

FC (Fully – Connected) layer – It will compute the final class scores for all the classes, resulting in volume of size [1*1*1000] , where each of the 1000 numbers corresponds to a class score of particular class.

3) Learning Algorithms

Given a set of training image dataset (T), we train the CNN model to discriminate between 1000 object classes. ReLU/ Rectified Linear and Leaky – ReLU activation layer is used for forward pass whereas loss minimization is done using Stochastic Gradient Descent (SGD) layer.

$V_{t+1}=\mu V_t-\alpha\Delta L(W_t)$

$W_{t+1}=W_t+V_{t+1}$

Where,

$\Delta L(W)$= Gradient of Weights

α= Learning rate

μ= Momentum

W_t= Current Weight

V_t= Previous Weight update

4) Hyper parameter setting

We started the training with learning parameter of 0.0001, which follows the step learning policy. This policy reduces the learning rate with the value gamma after the given step size(number of iterations).We set gamma=0.1 and step size of 1000. Weight bias has been initialised to 0.0005. Training gives validation accuracy of 83%. For competition purposes we bench marked with 8 GB RAM machines as shown in following table1

Table 1: Accuracy Results of proposed system

Number of classes	Number of Iterations	System Configuration	Validation Accuracy
1000	10,000	64 BIT,8 GB RAM	83%

5. Use Case Specific Reference Module Generation

Based on our use case where we have the images repository in which the user or customer images need to be comparatively matched.

This reference module as given in the figure2 is generated in the training face of our solution. It contains image repository (see Use case driven repository) of different classes. This classes can be defined dynamically based on the Use case. This image repository will be used as a reference for matching user query image(image matching) . Apart from the matched images from the image repository, top five labels and corresponding feature vector for each image is also stored inside the module. This labels and featured vectors are saved for image matching purpose namely label matching and feature comparison

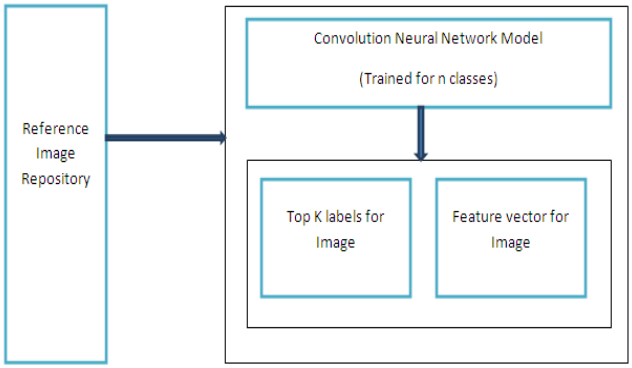

Fig. 2: Reference Model

Here label means top 5 most probable objects or products detected inside each image by already trained CNN model as given in the section IV. The feature vector is nothing but the vector produced by one of the Convolution layers of the CNN model. The process of getting labels and feature vector is similar of both training and testing phase. The training CNN model is described in section IV is used for the same. The reference module generates all features and labels of images repository which the user images need to be matched.

A. Use case driven image repository

Based on the Use case or industry the image repository mentioned above can be created for image matching purpose. Note that this data base is different from the database used for training CNN model. This database or image repository can be created or altered dynamically according to the Use case requirements.

6. Image Matching Module

The Image matching module architecture is given in figure 3

A. Input

Input for this module is the image uploaded by the user. User can upload any image for which he/she intense to search similar images.

B. Label and Feature Extraction

Once this module receives an image, it is passed through our trained CNN model described below. For the image, label or class is extracted from the last layer of the model. This provides meaningful insight about the object present inside the image. This is how any CNN model is used in object detection based applications. As our objective is not just to detect object inside the image but to match images with one another, we have to extract additional information from the trained model for this we have to extract feature vector [1*4096] pertaining to input image. This feature vector has been extracted from one of the fully connected layers of model, which is present after convolution layers. This vector represents an image in one dimensional vector space and contains object properties detected by the model

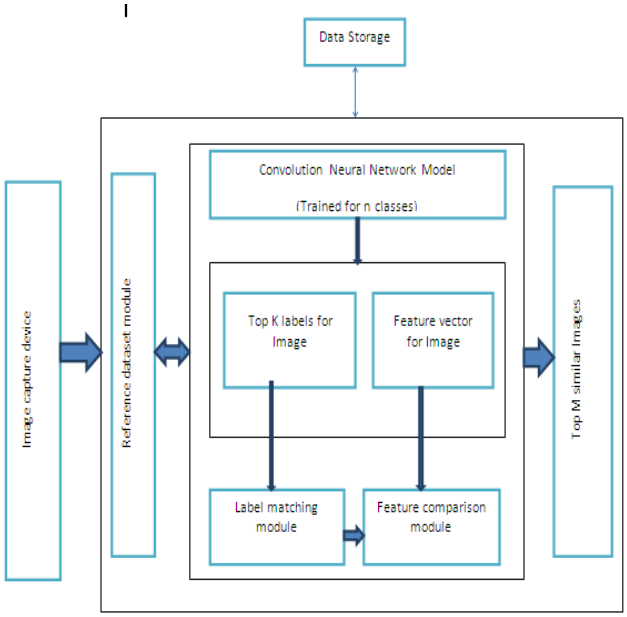

Fig. 3: Image Matching Module

C. Label matching module

This module searches for images from reference dataset module which have same label as user or customer uploaded image labels. This is done by matching label ids of user or customer image to the reference image label ids. Output of this module is a set of reference images with corresponding label and feature vector. This set is being forwarded to next module. Note that reference image label and reference image vector are pre-defined and stored in reference dataset module.

D. Feature matching module

This module determines distance between user or customer image feature vector and reference image feature vector. This activity is done with respect to each reference image. We have used Euclidean norms for finding the distance value.

The distance is defined as:

$$d_i = \| u-v \|_2$$

Where,

u-Test image feature vector [1*4096]

V-Reference image feature vector [1*4096]

d_i-Distance test image and i^{th} reference image

Based on the distance values (lower the better), K reference images are selected as closely matched images. User can set the value of K for each iteration. The same images are being provided as the output to the end user.

7. Experimental Setup

For end to end workflow and processing of our solution, we a Use case driven image repository where we used this solution as explained in the introduction section1 and we have validated this solution at multinational courier delivery services.

A. Back ground- Courier services

In courier services industry, sometimes the product or packages, we sent from sender to receiver, gets lost in between. This might happen due to loss in tax which has sender and receiver information, or if some part of the package, falls out during the transportation. This kind of situation makes it very difficult for the courier services provided to deliver the package or mapping with the sender information. In such cases they providers keep the track of lost packages by manually saving package information. This information includes product name, colour and other product specifications. Sometimes they save product image also, in case some customers comes back and ask for not received package. Even though this information helps the service provider mapping customer and their lost product when a customer comes back to them, it is a cumber process to look for a lost product from the huge lost and found inventory database. Our solution, if integrated with the courier service provider inventory database can help them finding the similar lost product quickly and efficiently. We have created an input reference image database of 50K images for this experiment. This database has images of different products with different back grounds and orientations. After collection of images, all images were resized to some size (227*227) for processing. 80% of the collected images were used as reference data and 20% of them were kept aside for testing.

B. Processing-computer vision

In the experimental setup, we experimented with computer vision algorithms to compare two images. Here, the aim was to compare the image matching results of computer vision algorithms to our solutions.

Computer vision provides various algorithms to compare images with one another. We utilized standard histogram algorithms and HU moment algorithms to compare images. Python bindings [9] of Open CV [7] were used for the same.

For the training data images, we extract histogram features and HU moment vectors and stored them in our local disk. Once a test image is passed to the system, we extract the features for the test image and then compare it with all the training images. Euclidean distance is calculated for measuring distance between each training image and test image. Training image with lowest distance from test image was given as output. Few examples input output samples are attached below.

Fig. 4: Test Case 1

Fig. 5: Test Case 2

Fig. 6: Test Case 3

Fig. 7: Test Case 4

We can see from the test case figures; Computer vision algorithms have very low accuracy for image matching purpose. This happens because computer vision algorithms use specific features like shape, colour, texture of the object which comparing distance between two images. This feature works only in particular use case with limited dataset. As the data size increases, the solution tends to show lower accuracy levels. Here we are able to match only 30% cases using these algorithms.

C. Processing-Deep Learning

As explained above we generate the reference dataset module having labels and features for each image (from training data of 40K images). After which the module is tested with rest of the 10K images. Each image was passed to our system and similar images are found from our reference dataset. Have look at the example below.

Fig. 8: Test Case 1

Fig. 9: Test Case 2

Fig. 10: Test Case 3

Fig. 11: Test Case 4

Each image has an input image and top 5 similar images provided by our solution. In four cases presented above, we have provided test cases of different types of products (a plastic bottle, clothes, mobile devices or a wrist watch). We could get some similar product images for the database. Using CNN method which considers the correlation between pixels along with features like colour, texture, shape. As mentioned above, we have 10K images for testing the solution. We have generated the output images for each test image and saved them for accuracy calculation purpose. Accuracy calculation was done manually as we need to compare input image with output images. We have an overall accuracy of 80-85% on images. When compared with computer vision algorithms our solution has shown significant improvements. Our proposed solution utilizes convolution network to extract image features. This provides the ability to extract all image features (colour, size, shape, etc.). Together which improve results for image matching purposes. Also, in computer vision solution we need to compare every test image with all the available training images which in some case can be time and memory consuming.

8. Conclusion

In this paper, we described the image matching algorithms using deep learning models. We used the pre-trained CNN model to extracts the images features and labels of image. We have the reference images and we used pre-trained CNN model to extract the features and labels of the images and stored them as a reference database. User comes with an image as we need to match those images with the reference dataset. We have used CNN features and Euclidean distance approach to arrive top 5 matching images with reference dataset. We have tested this solution at multinational courier delivery services industry and some use case

was discussed at above section. We also the compared the deep learning approaches with computer vision algorithms and observed deep learning approaches are giving much better accuracy than computer vision algorithms.

References

[1] Dosovitskiy, A., Springerberg, J. T., Riedmiller,M., &Brox,T.(2014). Discriminative unsupervised feature learning with convolutional neural networks.In Advances in Neural Information Processing Sytems (pp.766-774).

[2] Wang, X; & Gupta, A. (2015) .Unsupervised learning of visual representation using videos. In proceedings of the IEEE International Conference on Computer Vision (pp. 2794-2802).

[3] Wu, Di, Lionel Pigou,Pieter-Jan Kindermans,Nam Do-Hoang Le,LingShao,JoniDambre, and Jean-Marc Odobez..”Deep dynamic neural networks for multimodal gesture segmentation and recognition.”IEEE Transactions on pattern analysis and matching intelligence 38,no 8 (2016): 1583-1597.

[4] Huang,Chen,Chen Change Loy and XiaoouTang.”Unsupervised learning of discriminative attributes and visual representations”.In proceedings of the IEEE conference on Computer Vision and Pattern Recognition,pp.5175-5184.2016

[5] Yang,jainwei,Deviparikh and DhruvBatra.”Joint Unsupervised learning of deep representations and image clusters.” In procedings of IEEE Conference on Computer Vision and pattern Recognization,pp.5147-5156.2016

[6] Dundar,Jonghoonjin and Eugenio culurciello.”Convolutional clustering for Unsupervised learning.”arXiv preprint arXiv:1511.06241(2015).

[7] https://opencv.org

[8] jia,Yangqing and Shelhamer,Evan and Donahue,Jeff and Karayev,Sergey and Long,Jonathan and Girshick,Ross and Guadarrama,serigo and Darrell,trevor,caffe:Convolutional Architecture for Fast Feacture Embedding,2014,http://Caffe.berkeleyvision.org/

[9] https://www.python.org/

[10] Dr. Seetaiah Kilaru, Hari Kishore K, Sravani T, Anvesh Chowdary L, Balaji T “Review and Analysis of Promising Technologies with Respect to fifth Generation Networks”, 2014 First International Conference on Networks & Soft Computing, ISSN:978-1-4799-3486-7/14,pp.270-273,August2014.

[11] P. Sivakumar, V. Rajasekaran, K. Ramash Kumar, “Investigation of Intelligent Controllers for Varibale Speeed PFC Buck-Boost Rectifier Fed BLDC Motor Drive,” Journal of Electrical Engineering (Romania), Vol.17, No.4, 2017, pp. 459-471.

[12] S.V.Manikanthan and K.Baskaran “Low Cost VLSI Design Implementation of Sorting Network for ACSFD in Wireless Sensor Network”, CiiT International Journal of Programmable Device Circuits and Systems,Print: ISSN 0974 – 973X & Online: ISSN 0974 – 9624, Issue : November 2011, PDCS112011008.

[13] T. Padmapriya and V.Saminadan, “Improving Performance of Downlink LTE-Advanced Networks Using Advanced Networks Using Advanced feedback Mechanisms and SINR Model”, International Conference on Emerging Technology (ICET), vol.7, no.1, pp: 93, March 2014.

[14] S Nazeer Hussain, K Hari Kishore "Computational Optimization of Placement and Routing using Genetic Algorithm” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.47, page: 1-4, December 2016.