Logo detection and recognition using CNN

by

Artykbayev Kamalkhan Serzhanovich

Submitted to the Faculty of Engineering and Natural Science
in partial fulfillment of
the requirements for the degree of
Bachelor
in

Suleyman Demirel University

June, 2018

ABSTRACT

LOGO DETECTION AND RECOGNITION USING CNN

Artykbayev Kamalkhan Serzhanovich

B.A. Thesis, 2018

Thesis supervisor: Senior Lecturer MSc. Konstantin Latuta

Keywords: Logo detection, Logo recognition, Computer Vision, Machine Learning, Convolution Neural Network, Classification, Recurrent Neural Network, Pattern Recognition, Object Recognition,Data augmentation

This thesis describes the research work carried out to fulfill the Bachelor in Computer Science at the Suleyman Demirel University. Research was in Technopark at Suleyman Demirel University and was supervised by Konstantin Latuta. Logo detection and recognition continues to be of great interest to the document retrieval community as it enables effective identification of the source of a document. This paper contributes the design of the system able to detect the logo of any product from the documents and images after that recognize it from the archive via the convolutional neural network. For detecting and recognize of logos implemented via convolutional neural network, which creates initial classification to determine the presence of the logo on the document or image.As regards to the former, a collection of logos was designed and implemented to train the classier, to identify and to extract the logo features which were eventually used for logo detection and recognition. The latter regards the detection of logos from an input image. In particular, the experimental study aimed to detect if the input image contains one or more logos and to decide which logos are contained.

ACKNOWLEDGEMENTS

I thank the merciful and all-knowing, for sparing my life in sound health and giving me the opportunity to accomplish this thesis.

I wish to express my deepest gratitude to my supervisor Senior Lecturer MSc. Konstantin Latuta for his guidance, advice, criticism, encouragement and insight throughout the research.

I am highly indebted to my parents for their encouragement, support and unlimited love.

Finally, i wish to extend a special thanks to my colleagues for their valuable support and company. They really made my life a fabulous one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

MSE	Mean-square-error
CNN	Convolutional neural network
RNN	Recurrent Neural Network
CV	Computer Vision
ML	Machine Learning
MLP	Multilayer Perceptron
LRT	Learning Rate
$Conv$	Convolutional layer
$Pool$	Pooling layer
$ReLU$	Rectified Linear Unit
$Softmax$	Normalized Exponential Function
$Sigm$	Special case of logistic function
SS	Selective Search
κ	Clusters count
C_κ	Cluster
m_κ	Mean of cluster
D or $E(x)$	Function of error
L	Loss
θ	Parameter of training
γ	Monentum parameter
g	Gradient
G	Gradient matrix i×i
ϵ	Smoothing coefficient
η	Learning rate

CHAPTER 1

INTRODUCTION

1.1. Overview

Object recognition and object detection are ones of the lasting and most important goals in the computer vision.Because of this problem in a extremely considerable range of usage. For example, in copyright detection, contextual advertise placement, vehicle logo for an intelligent AI-based traffic-control system and brand detecting in social media. As well as these algorithms have many applications in location recognition, advertisement, and marketing. Presently advertising is a very powerful tool for income and attracting of customers.For this reason, the analysis of the brand and mention on different resources are very important and primary tasks for business analysts.In order to captive, attractive their customers and make better decisions, companies needs for analyzing the presence of their logos in photos, videos and another type of contents. Logos help to evaluation of identity between something.[1604.06083] [1701.02620] [1711.09822]

The logo mainly includes text and graphical symbols. In such cases, when the logo is in different parts of the image, the logo is inverted, the logo is distorted, as well as changed in size - recognition and definition of the logo is a very problematic and difficult task. For example, logos on their clothes, which are often deformed, which complicates its detection and recognition. [1511.02462]

Recent breakthroughs in deep learning improve recognition models with very extremely minimal loss function.Models which was created for recognition based on neural networks have excellent accuracy, speed, as well as these models, have the ability to be really smart. In short, the ultimate goal of the system, which is based on the recognition model, is to create a method that defines logos accurately and continuously learn from new logos.[1711.09822]

1.2. Related Works

In literature, I can find many works on logo detection, logo extraction, logo classification , logo retrieval and logo recognition.Research that has been done on for the last 20 years and that are related to logos have been done with datasets, which consist too small data. For example, ”Flickr-Logos” dataset, which includes only 32 logos, which distributed to 5644 objects on 8240 images.Obviously available public datasets not enough for creating real-time detector machine. And this machine, will not be able to fully use its potential

of detection with a neural network, due to the lack of data about other logos.[1511.02462]

Most of recent applications of detection and recognition of object have been based mostly on Scale-Invariant Image features.Scale-invariant image features algorithm provide us transformations and representations to gradients of images. These gradients are invariant to affine type transformations and despite the conditions.Models that were made based on SIFT, basically make a better that part of the picture that is specifically different from the rest of the content.At the moment, although there is a plural number of methods for logo recognition, the performance, and power of Convolutional Neural Network is growing very strongly in the field of computer vision. After all, CNN solves many of the problems of the basic classical computer vision algorithms and includes a large range of uses in the recognition of images and objects. The structure of convolutional neural networks is very hierarchical and multilayered, as well as it is designed so that the pattern can be recognized only from the pixels themselves.[1604.06083]

In this [bianco2017deep] paper they propose a method of logo detection and recognition with using main deep learning algorithms.Their recognition and detection process have given with pipeline, which consists main 5 step. These steps: taking an image, doing object proposal, cropping to regions, passing through the trained convolutional neural network and making a prediction.This algorithm recognizes logos well even if they are not exactly localized in the image.The neural network was trained and tested on the ”FlickrLogos-32” database.For improving result they trained CNN with very differently benefits. As an example, for avoiding overfitting they used class - balancing in every batch. Also, they confirm sample-weighting and add a new class ’no logo’, which includes only images without any logos.

Also, this [1511.02462] paper presents a method that works perfectly with logo recognition, and returns the bounding box of the found logo. In particular, recognition of the logo has broad application and uses it in many areas. To protect people intellectual property, logo recognition is the most convenient and effective tool.As mentioned earlier, in the area of logo recognition and identification, most tools have a very small dataset.But researchers from this article had presented a large-scale database, which has 160 classes distributed among 130608 objects. This dataset is really huge and it is called ”LOGO-net 160”. For cropping the image into regions and search regions of interest(RoI) they used selective search algorithm, that efficient for this type of tasks. After features extraction with CNN, fully connected two layers divided into softmax predictor and bounding box regressor. This mathematical operations provided us classification of logo and it’s position on image.

Guys from this [1604.06083] paper demonstrate method for recognition, which based on Region-based Convolutional Networks.A distinctive feature of this approach to solving the problem is the recognition of multiple objects in the image.

CHAPTER 2

PROBLEM STATEMENT AND THESIS ORGANIZATION

2.1. Introduction

In this chapter, we will explain the main problems that researchers faced when they recognize and detect logos. And also, briefly explain how you can solve difficulties of this complex process. We will also present you the content of the thesis, which will briefly clarify what will be shown in the following chapters.

2.2. Statement of the Problem

Logos are 2-dimensional shapes of varying complexity,with interior and exterior contours that are not necessarily connected. Therefore the recognition process seems to be difficult because of its complexity.For this reason, the logo recognition process is a difficult task.In this problem, you can also highlight the moments when the method works very well with perfectly made images, but when using some images that may be deformed, inverted and blurred, then at such moments the method was simply useless.After all, such methods are usually trained on a perfect images dataset. So, in this case, the model can work with real images.[1-s2.0-S0031320302001280-main]

Because of any transformations such as rotation, shift and scaling, as well as the position in which the logo is placed, makes the task of recognition a special case, because the slightest shift can significantly affect the result of recognition, since the classical methods of computer vision are very sensitive to the slightest changes in images.Even lighting and illumination can greatly affect the result, as they strongly affect the inversion of any pixel.Most of the methods cant cope with the recognition of logos because they are very limited in terms of the application and the structure of the algorithm.The variety of logos and their size's requirements makes it very difficult to create a fixed model that will be adapted to this variety.Optimization methods in most models are not very suitable for the case of logo recognition.Complex geometric shapes of logos and the lack of information about the cascade of the logo on the image during the training of the model lead to the fact that the model is underfitting or overfitting.[1-s2.0-S092523121631387X-main]

Another problem in logo recognition is the limited number of datasets, and collecting your own dataset is very costly and hard work.With a rapid jump in the creation of multimedia technology, the number of logos is growing up very quickly, which makes it a difficult process in the protection of intellectual property, as

well as a very interesting and challenging task.[1612.08796]

Also, the problem is complicated by the fact that most of the available and targeted images for experiments are very limited, with a small number of classes and the same type.[1803.11417]

Despite the results of logo recognition models of the most ideal and convenient for the algorithm cases, the recognition of logos from real images is a particularly difficult problem, which can handle not every algorithm. One problem is when the logo is very small and in a distorted state. Also, present the problem of logos that are on the clothes and they just become vague. Still, pose a problem for those cases when the logos of any single company may be very different at different angles.[1511.02462]

After reviewing a decent number of methods, I decided to repeat the work of [1701.02620], and divide this problem into 2 main parts. The first is responsible for the detection of logos, or rather approximately different parts of the image. Second, make recognition using deep learning algorithms. In creating sample areas where you can find the logo, I want to use the selective search method, which will be able to divide the images into regions. And then these regions will go through CNN, reaching for softmax, which will classify and give us an answer, what kind of logo it is. But unlike other researchers, I want to try with different types of CNN architectures and try to change the SS, or try other methods in the floodplain distinguishing features of the image. Also for improvement of a result at training, the new class of the background is added. SS is a method that searches for regions of interest in an image that is somehow different from the rest of the image. But in turn, this algorithm finds the false parts of the image that are not the logo. This problem is solved by CNN, which will in most cases refer this part to a class where there is no logo.

2.3. Thesis Organization

The structure of the thesis is organized as follows:

- In Chapter 3, a general review of main methods of image segmentation and deep learning algorithms
- In Chapter 4, the proposed algorithm is presented. A review of the Selective Search algorithm and Convolutional Neural Network model, with softmax and prediction frameworks.
- In Chapter 5, an experimental study is provided in order to compare the performance of the proposed methods for logo recognition and logo detection.
- In Chapter 6, conclusions and a discussion on possibilities for future work are provided.

CHAPTER 3

REVIEW OF DEEP LEARNING AND PATTERN RECOGNITION ALGORITHMS

3.1. Introduction

This chapter provides a review of the well-known algorithms of image segmentation, pattern recognition, exhaustive search and deep learning methods.We will also explain how the main image segmentation methods work and how they developed in computer vision. Also will demonstrated methods forward and backpropagation. Between this two process, you can see the optimization process, which try to minimize function of error.

3.2. Computer Vision and Pattern Recognition

For a person, the perception of the outside world with your own eyes is a very simple task, be looking at any 2- or 3-dimensional object, you can safely tell about its shape and external structure.Looking at the crowd of people, the human brain can easily calculate the number of objects, can tell about their shape and condition. But what about the computer? Will the computer be able to handle the processing of objects that people see? Will the computer be able to find the difference between very similar objects? In this matter will help discipline called computer vision.This area is very closely related to signal processing, image processing, and video recording. As well as it includes machine learning with pattern recognition.Along with other Sciences like text processing and audio processing, science tries to create the ideal artificial intelligence that can think and act like a human.Image processing not only includes the transformation of images into a more comfortable and desired type but also this area along with computer vision will be able to show what is inside the image. Image processing not only includes the transformation of images into a more comfortable and desired look but also this area along with computer vision will be able to show what is inside the image. Also, this area helps in capturing movements inside the picture.[CVPR] Understanding what exactly is happening on images and perception of this process is an important process in AI. Draw conclusions depending on what you see is a fairly simple process for a person, but not for the computer. Since a computer without any reason cannot understand the essence of the process. The problem in object recognition is the appearance of these objects in new forms or compositions because the pre-built model cannot cope with it, because it has not seen the object in this format. These new formats can be represented

as an object in the expanded state, or it can be simply in motion. A huge number of new forms and aspects makes the object recognition a practically impossible task.[introTOCV]

3.3. Image Segmentation Methods

In practice, the importance and value is not always fully the image itself, namely what are the specific parts of the image, and sometimes just the number of channels of the image.The first and one of the most important technologies for understanding what is happening inside this image is segmentation.Since only a segmentation can be divided into important and different parts.After all, it helps to understand the image inside the image, as well as to extract useful information for us. These aspects are extremely important for programs where image recognition is paramount.For all these reasons, it can be understood that segmentation is a very important discipline within computer vision, and in turn, segmentation has a huge number of difficulties in implementing many methods.In short, segmentation is important for recognition, because it can pull out those areas that are very important for humans. And are the basis for all methods of recognition of contours and objects.There are many types of segmentation and a huge number of places where you can use them.One of the most common methods is threshold segmentation.The basis of this algorithm creates a segmentation of the image by its regions.This method searches for a threshold by a specific criterion to create a grayscale that will be distributed from other colors. This method sets a specific threshold for pixels and depending on the condition they change from 0 to 255 in grayscale.You can also mark a method called edge segmentation. This method is particularly the fact that he refers to the saturation of gray on the borders of any object.In the discipline of computer vision and related industries, there is no single segmentation method that can work in all cases. To use the segmentation method correctly, you need to consider the advantages and disadvantages. After all, each method will lead in different ways depending on the situation and the state in the image. And it is also very important to apply the correct parameters of segmentation methods. Since the parameters play a significant role in the algorithm.[1707.02051]

Segmentation, by itself, is splitting the image into several areas, depending on their structure, size, and saturation of any particular colors. These areas can include grouped pixels, which represent the object itself, and can represent a variety of shapes, such as an arc, circle, or just a line. Developed regions can be simple lines or full-fledged objects that can have boundaries separating them from other content. Since the area of interest may not cover the entire image, we are interested in using segmentation in such cases. Segmentation has two main goals that it pursues. The first is to expand the image to the desired regions. The second task is to change the representation.Considering the simplest cases, when the interesting part of the image is very different from the rest, the segmentation will not be a problem. After all, the area of our interest, especially its color and saturation help to clearly separate it from the rest of the image.

After all, the rest of the area does not have similar components as in the desired image area.But there are also severe cases where the boundaries are strongly distorted and erased as the color saturation is very similar, and the components do not differ from each other.Considering the second objective pursued by the segmentation can be sure it will ultimately give us a richer and more precise representation of the object within the image. Here our task is to gather pixels into one whole, into a more integral area, which is much useful and important for future research, because we create a clearer outline of the object. The perspective of an image can serve both as a useful tool and a very strong drawback since the borders can be clearly highlighted or even erased in the image. **Here will be one two images, seg1.png and seg2.png**Typically, classic segmentation techniques may not work well for images where the boundaries between the desired features are blurred and blurred, making this work practically impossible because the pixels are too similar and the features cannot be separated or isolated.To divide the image into several parts according to the regions, have to be extremely homogeneous as the level of gray. After all, black-and-white images are easier to work with due to algorithms. As well as the color and texture of the image are also important when dividing by regions. Neighboring areas of the desired object should have very different characteristics and features because the uniformity prevents the algorithm. Also, borders of the object should be evenly distributed, and also they should not be torn or distorted. Achieving all the above characteristics gives a certain amount of difficulty, after all, how would the objects did not have their uniform or completely, they still have dire and slits, which interfere with segmentation algorithms, making a homogeneous region in a heterogeneous region. Also, our eye can also be mistaken in terms of the homogeneity of the object inside the image, because sometimes there may be holes or cuts that are not subject to our eye, so the number of pixels that we can not see, can interfere with the segmentation.[ch10]

3.3.1. Thresholding

In the methods of segmentation are the segmentation types of image in parallel. The most common and easiest method is to segment an image using a threshold. This method is based on the use of gray color. After all, we know that translating the image into a black and white contour with it is more convenient to work with than 3 color channels. This method segments the image based on image separation by saturation and grayscale. It is able to divide image according to its local threshold, which is automatic, depending on the distribution of the white and black color. And also, you can split the image using a global defect which can be defined as static and automatic and manually. It can also be noted that the threshold can be dynamic because it changes from area to area by an image.Global threshold divides the image into the desired area and its background, which he considered not similar to the area of interest to us. The local one does this by going through the image, and depending on the situation and position, select a threshold

to be divided into the main part and the background part.THE most common and convenient method of threshold segmentation is - Otsu method. This method uses the interclass variance to separate areas of an image. The method is special and distinctive in that it selects only the global threshold. And the threshold is chosen by the maximum dispersion between all classes inside the image. The segmentation method has found extensive application due to the fact that it is very simple to calculate and does not require costly calculations and calculation when the algorithm itself. Also, due to a simple calculation and increases the speed of the algorithm. This algorithm can work very well when the boundaries between the object and the background are separated by an accurate and bold contrast line. In such cases, you can obtain accurate segmentation results for the image. But in the opposite case, when the boundaries are erased, this method is not able to cope, because it will not know exactly what is the object and its background. Noise can easily interfere with this method. Because the noise erases the boundaries. And uniformity also can ruin the quality of the method is the segmentation threshold.This method is effective in combined use with other methods.[1707.02051]

3.3.2. Clustering Methods

Clustering is a very powerful and unpredictable method in machine learning and computer vision. In computer vision, or rather in segmentation, it works by splitting image vectors into groups called clusters. Since clustering methods are not the same type, we can consider several types of clustering methods, but the essence of its work is based on similar points, which are very similar, and then they are grouped into separate clusters.The main problem of clustering is the correct splitting of the image into the correct sets of vectors. In this case, these vectors must be collected to have a similar value in the numbers, which means their structure must be similar. In these vectors consists mainly of the pixels of the image. Also, components can be indicators of the intensity in a given area, and 3 channel parameters that are related to each other. Texture, namely their calculated values can also be components. To associate pixels in groups, you can use any component or parameter that combines these pixels into a single value. Due to this, it is possible to find the associated objects and re-create the segmentation for the pixel count.[ch10]The *least squares error* is one of the most common measures to compare clusters that use the traditional method to break into groups. Clustering involves the process that determines the number of clusters κ. the same is created the number of groups from C_1 to C_κ. Each such cluster has its own personal measure of average m_κ. The formula of the error, said earlier, is calculated as follows:

$$D = \sum_{k=1}^{\kappa} \sum_{x_i \in C_\kappa}^{1} \|x_i - m_k\|^2 \tag{3.1}$$

This formula shows how close this object's data is to a particular cluster. This procedure will help you to see all the possible options for partitioning into K-th number of clusters. As a result, it will find the best option to minimize our error function D. the Disadvantage of this method is that it is impossible to calculate everything. For this reason, they find the closest number in value and divide the rest of the objects into clusters. It is also very difficult to find a global and optimal variant of the error function iteratively, for this reason, they usually resort to the random selection of clusters and selection of their mean values for further calculations. This method is called k-means clustering. There is also a method that is different from it. It is called isodata clustering. It uses a similar method of splitting and merging.This method is based on creating groups from their distance from the center of a particular cluster. Clusters are initialized randomly and iteratively go through the positions to find the most optimal point at which the error function will be reaching.[ch10]

This method is the simplest and fastest, and most efficient for large datasets. Because it's easy to scale, it's very responsive for large datasets. The iterative nature of this method makes the optimization process easier and more convenient for calculations. But it also has a number of drawbacks, such as the number of clusters and the parameters by which these clusters need to be calculated. The iterative method is bad because every step goes through the whole sample, which is very expensive and time-consuming to calculate. There is also a problem with non-convex clusters, as they are difficult and impossible to calculate.[1707.02051]

3.3.3. Edge detection

The edge segmentation method has its own feature to work faster due to the lack of a large amount of information that needs to be processed somehow because this method uses only the detected edges, which store all the necessary information for this method.Because of the very easy and understandable implementation, it is included along with other segmentation algorithms for recognition of an object. It has good ability to work in big data processing.The main advantage of this method we can call the extraction of accurate edge lines, which in the future will help to create their own borders and segment the image. Also when removing it takes out edge with the desired orientation. But we can not say for sure that the performance and reliability of this method are good or bad, because of the formulation of problems and the benefits of the method, each researcher must judge himself, based on the results that shows this method.This working method transforms the original image into divided regions which are clearly separated by edges in which the grey tone prevails. The edges that are at work of the algorithm become the boundaries between the object and its background, but again only the researcher can judge what is the background and what is the desired object. The benefit and contribution to the computer vision of this method are simply colossal. Also, a feature of this method in the creation of localization in the image of dependence on the level of gray color

in the certain region.What are these edges? This is basically a change from pixel to pixel intensity of any color depending on a certain direction. From the same edges are extracted the main objects that will be used in the future. Because of the importance of grey, exploring the gaps between them is very important. Breaks can represent points, lines, and edges as well. Since breaks are useful, and they have based the basic methods of segmentation at the edges. It is possible to allocate methods such as Roberts, Sobel and Kirsh edge detection methods.[1211csit20]

For the definition process, we can know for sure that not all points in the gradient have nonzero values. But at this point, we can say that not all points are important to us.During the improvement process, special pixels are highlighted. Under the peculiarity, we can say about the gradient change and about the intensity in this region. Also, to restore the image and get rid of unnecessary noise, we can recreate the filtering process.For example, we can talk about its two most common methods like Sobel operator and Laplacian of Gaussian Operator.**Sobel Operator:** It goes through all gradients and highlights the rich and intense frequencies that pass through the object boundaries in the image.

Table 3.1. **Sobel Operator**

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Laplacian of Gaussian Operator: For use this method, you need to calculate the derivative of the second order and create a mask on the Laplacian distribution, which is listed below:

Table 3.2. **Laplacian of Gaussian Operator**

0	-1	0
-1	4	-1
0	-1	0

The edge segmentation method has other methods, but all methods mainly work with masks that go through the entire image and converts it to the desired shape.[1210ijcsit14]

3.3.4. Histogram-based methods

Since the other segmentation methods require a full pass through the image, there is a method that works with histograms, and its speed is much faster than the rest. It involves the least time pass on the pixels of the

entire image.This process involves grouping the space of an entire dimension, in which all similar objects behave the same way and are grouped into sets, in other words building their histogram. Segmentation in this process is carried out by comparing all the existing groups in the set in its area inside the image, in which all the related parts will look like a group. The intervals between histograms represent a change as hills, where colors shimmer from each other. The peculiarity here is the gray color that will be distributed between the multi-modal histograms, and the regions will be defined by peaks. Then again you need to define a threshold that will determine what region we want. This motivates the need threshold value methods, focused on understanding and creating conclusions, in which the selected thresholds are closely related to the histogram and the quality of the region or area.[ch10]

3.3.5. Compression-based methods

The compression process is a more advanced segmentation method, so it includes 2 main processes, it is segmenting the image, and further compressing it. A result is a huge number of regions that do not intersect each other, and their combination can lead to a return to the original position, which in some cases is convenient. After the segmentation has occurred, the compression process continues, which compresses all created segments to a specific state or size. Segmentation on the output gives different objects, such as smoothed parts of the image, text, individual images, graphics, and areas that overlap the rest. Smooth areas use a compression method that uses an arithmetic encoder that operates on the basis of contrast and color palettes. The text uses a different method that uses text encoding. Classic JPEG-based compression methods are used for the image.[compressionBased]

The segmentation algorithm works together with intensity values. In intensity, it is important to keep in mind the similarity between pixels. Due to the fact that the object is taken as a whole, its edges can be saved intact, to continue to use. The image quality is also maintained. At the stage of compression, for each segment, its type is determined, and then the method by which it will be compressed is determined. For efficiency, segments are encoded into a specific format. Effective and reliable (adaptive) methods of image compression using remote sensing become more and more necessary both in quantity and in size images for archiving and transferring the purpose of the network is constantly growing. In addition, all remote sensing images have a huge amount frequency component. Therefore, it is important to reach a maximum the degree of compression while maintaining a reasonable the computational complexity of implementation and high the visual quality of the restored image and, possibly, only Segmentation based on image compression.Amount of segmentation compression methods of general and remote control are offered picture.An algorithm for lossless image compression is proposed, using the Segmentation of the size of the variable block. Duplication in the representation of a digital image can be divided into two categories: local and

global. In this work, a lossless image compression scheme is used, using redundancy at the local and global levels ensures maximum compression efficiency. This algorithm segments the image into blocks of variable size and encodes them depending on the characteristics presented pixels inside the block. The execution of this algorithm is superior to other lossless compression schemes, such as Huffman, arithmetic, Lempel-Ziv, and JPEG. But the characteristics of estimating the distribution of the image and the resulting efficiency of compression are a very difficult task because of the huge amount of computation.[fulltext24632013]

3.3.6. Dual clustering method

This method represents a combination of other methods we are familiar with. It uses the three most important features of image segmentation. The image is taken, segmented using a histogram-based method, then their density and completeness are checked by clustering, and last but not least the boundaries of these segments are checked by the integrity of their gradients. The method is carried out by creating two spaces we need. The first is responsible for the intensity and brightness of the image. Next, in another space is the original image itself, which has a dual feature. As mentioned earlier, the first space checks the intensity and brightness distribution. Everything happens in the clustering process.[Vadim V. Maximov, Alex Pashintsev Gestalt and Image Understanding. GESTALT THEORY 2012, Vol. 34, No.2, 143-166.]

3.3.7. Region-growing methods

The last in our list and also one of the most common segmentation methods is the method of regional growth. This method is based on searching for similar and consistent areas within the image. Also, during the search process, similar pixels are merged into a region, which includes similar areas of the image. The method is simple in that it simply merges all similar pixels together to form an area as a whole region.Due to the fact that this method shares a well-connected region, a region well is a clear boundary that provides an excellent segmentation. When the segmentation process is started, there is a search for and an increase in the criterion, which depends on several factors. Segmentation is due to separation and the creation of clear boundaries. Again, the disadvantage is too expensive a number of calculations during the process. Noise and unevenness can also affect the result of this method.[1707.02051]

Methods based on the region rely on the assumption that all neighboring pixels in the same region have a similar value or a certain range. This leads to a class of algorithms known as an area whose growth the technique of ”splitting and merging”, perhaps the most famous. The general procedure is to compare specific one-pixel function for its neighbors. If the homogeneity criterion is satisfied, then the pixel assigned to the same class as one or more of its neighbors. Choice of the criterion of homogeneity critical for even moderate success, and in all cases the results are disappointed by the noise. The fourth type hybrid

methods, which combine boundary and regional criteria. This class includes morphological Segmentation of the catchment area and a landing surface of variable order. The catchment method is usually applied to gradient image. This gradient image can be viewed as a topography with boundaries between regions ridges of hills. Segmentation is equivalent to filling the topography of the starting points of the boundaries of the region so that they are open to keeping water from different points of sowing from the meeting. Like the whole perverted method, the technique encounters difficulties with images in which the areas are both noisy, and blurred or fuzzy volume. In addition, this method is also very expensive to calculate. In this article, A segmentation method based on the method of growing the sown area is proposed, which is less sensitive to noisy image and quantitatively and avoid an explosion, leakage, segmentation, and segmentation problems.[1412.3958]

3.4. Supervised Learning

The essence of supervised learning is to find algorithms that build hypotheses based on previously solved problems and you already know the range of conclusions that will be on the output. This method helps to solve the problem for the future and make a forecast for future data. In other words, the goal is to create a concise model that will be trained on these features and their labels. However, the model will be used to predict the data, where there are only signs, but there are no markings. But we know the exact range of these labels.Algorithms for supervised learning can be as classifiers, as regressors.The essence of supervised learning is to find algorithms that build hypotheses based on previously solved problems and you already know the range of conclusions that will be on the output. This method helps to solve the problem for the future and make a forecast for future data. In other words, the goal is to create a concise model that will be trained on these features and their labels. However, the model will be used to predict the data, where there are only signs, but there are no markings. But we know the exact range of these labels.Algorithms for supervised learning can be as classifiers, as regressors.Since the training is inductive, because this algorithm is trained from a certain set of rules, which are usually presented as datasets, or certain parameters are important in the conclusion of the forecast. In short, the task of this tutorial is to recreate an algorithm that is ready for new data instances and works correctly.An important factor in creating an entire training processor supervised learning, data selection and feature selection are paramount, as data can play a cruel joke with you. The process of working with data we can imagine as its pre-processing and transformation. Since only digits that do not contain noise, unnecessary data or just emptiness are needed to work with these algorithms. Removing this kind of data is an important aspect of training.The data can be generated using different machine learning methods. It can also be noted that in most cases a well-assembled dataset is more important than the algorithm used. It can also be too a huge number of signs, as it makes the

learning process more difficult and time-consuming. For this reason, dimensionality reduction methods with minimal data loss in the dataset are also an important part of supervised learning.Typically, the feature selection process can be represented as the removal of unnecessary and redundant information, since the datasets are usually sparse. After all, with the usual data collecting data is collected regardless of their measure of need. Also, when we find dependencies, we can remove not strongly correlated features, or we can generate them on the basis of dependent features. This reduces redundancy and also contributes to good algorithm performance.The choice of the algorithm should be approached very carefully and subtly because fully convinced of the data, we can rely only on the algorithm itself.The simplest method of checking algorithms is its accuracy. But we also must not forget about other parameters like recall and precision. In the evaluation of the algorithm, we should consider all the indicators and settle on its previously set task.The simplest method of checking algorithms is its accuracy. But we also must not forget about other parameters like recall and precision. In the evaluation of the algorithm, we should consider all the indicators and settle on its previously set task. Since classification occupies an important part in the work of intelligent systems, its development and speed is an important prerogative. More advanced algorithms supervised learning are neural networks, Bayesian networks, etc. **Here will be one image about all parts of supervised learning** [sup3]

Passable or hidden layers that do not include inputs and outputs are very effective in training, as they add a more complex shape, and the training of convex and non-convex shapes is carried out more easily and possible. Nonlinearities, usually not solvable, can be differentiated and brought into a more convenient form. As well as the relationships between all layers are a good work out this algorithm. Due to the rapid development of neural networks, complex and nonlinear problems in the problems began to be solved with the help of deep learning algorithms. The methods of neural networks are unique in that their error on any instance returns to the very beginning of the perceptron diffusing and changing the weight in the direction we need to improve the result.[sup2]

3.5. Optimization

In computing the function, and composes the important role played by its minimization or else calling optimization. This process is accounted for by minimization functions that reduce the error value in the main target function. It's just a mathematical function that depends on the internal parameters of the model used. Typically, you can retrieve these parameters when you create a dependency on the $Y(X)$ target value calculation from a set of parameters X that will be used inside the prediction method. In a neural network, there are such concepts of weight $W(X)$ and their bias (B), the role of which is to do calculations inside the neural network with input data, converting, minimizing and pulling them on the output layers. Since

the neural network problem to learn to predict better, each iteration calculates what proportion of the error of the estimated answers from the present. And the task at this stage is to learn how to minimize this error. This process of minimization plays an extremely important role in the training of neural networks.Passable or hidden layers that do not include inputs and outputs are very effective in training, as they add a more complex shape, and the training of convex and non-convex shapes is carried out more easily and possible. Nonlinearities, usually not solvable, can be differentiated and brought into a more convenient form. As well as the relationships between all layers are a good work out this algorithm. Due to the rapid development of neural networks, complex and nonlinear problems in the problems began to be solved with the help of deep learning algorithms. The methods of neural networks are unique in that their error on any instance returns to the very beginning of the perceptron diffusing and changing the weight in the direction we need to improve the result.The preparation of the model should be approached very carefully, and take into account the parameters of the model, which effectively affects its construction as a regressor or classifier. The quality of the result and its forecast directly depend on the parameters and their value. To select the same parameters, you need to resort to a variety of versatile methods that will update and make the calculation of the corresponding optimization parameters. Model output is indirectly and directly related to the operation of selecting and searching the most optimal model parameters. Optimization methods can be divided into two main categories, according to which the same optimization algorithms operate. To the first category of optimization methods, we deduce *first-order optimization algorithms*. These are optimization algorithms that use the first-order differential to minimize the loss function $E(x)$. When this loss function is optimized, a gradient differential is used. The first-order derivative, which is calculated from the calculations of the main function, shows how much the function has changed at a certain point. At the same time, the function can decrease and increase at this point. When visualizing this function, we may notice a tangent to the point where the underlying surface loss is calculated.Here emerges the question, what still is this gradient value. If we look at a gradient in a simpler sense, it's just a vector. A vector that includes all possible variations of the generalization of the derivative (dy/dx). In short, the gradient is how much the specific value of $Y(x)$ from the x parameter changes.the Gradient value is calculated by a purely hourly derivative. Also by calculating the gradient that its output function creates a vector field.The calculated gradient value is represented by a Jacobin matrix that shows all partial derivatives.The difference between the second category of optimization methods, which simply uses a second-order derivative. Also a commonly used name for this species Hessian. This category is very similar, it simply creates a matrix as a vector space that is filled with second-order partial derivatives. The derived function shows not the change of the function, but the change previously calculated by the derivative of the first order. In this case, it is calculated how

much the function has warped. And also when visualizing, we can see that it is not a line, more parabola, quadratic surface that touches the line of our loss function [Anish Singh Walia, Types of Optim methods.] One of the most common and widely used optimization methods for neural networks is the gradient descent method. But on the other hand, the implementation of this method in libraries for in-depth training is different in some versions and functions, which causes some oddity among researchers.As mentioned earlier, gradient descent is used to minimize our E(X) function, which will be parameterized by a model that uses parameters from X in the form of O. Minimization is based on the aspiration of the same parameters to the anti-gradient.Antigradient is the opposite direction from the main gradient. Speed or as stated differently, the steps are determined by the rate of learning, which shows how to move our value in the direction of anti-gradient. We move along the slope until we reach the local minimum, or sometimes this local minimum is a global minimum.We can separate the gradient descent method by the amount of data that is used to execute the target function. Since the quality and speed of gradient descent change with different amounts of data, it was advisable to divide it into 3 types.[1609.04747][1801.06159][1801.03137][1801.09136][1803.02922]

Batch gradient descent. Batch gradient descent works with the entire dataset, and it is calculated during each iteration and with each sample data. It is the most classical method, but it is very expensive in terms of time. The parameters of the model θ change every era with this formula:

$$\theta = \theta - \eta \nabla_\theta E(\theta) \tag{3.2}$$

The difficulty in this process presents us with a dataset that may not fit in our RAM, and the problem in calculating the gradient will not be solved. Also the problem is that models can't learn on the fly with new data. After all, the model works so that going through the entire dataset, the process begins back, and cost more in time and capacity. The process includes a priority calculation errors on all of us selected the sample, then computed the vector turns into a matrix with proizvodnymi of the first order. Next, the calculated matrix uses a Loya update of the main parameters of the neural network or model to approach the minimum point of our target function. The advantage of this method is to calculate the global minimum and the local minimum completely and accurately. The first is computed for convex surfaces, and the second for non-convex surfaces, which is sometimes valuable for researchers.[1609.04747][1801.03137][1801.09136][1803.02922]

Stochastic gradient descent. This method, unlike the first one, calculates the gradient and updates the parameters for each package on which the model is trained. Our calculation is made between x_i to each of

its target $y(x_i)$:

$$\theta = \theta - \eta \nabla_\theta E(\theta; x_i; y(x_i)) \tag{3.3}$$

Because of the batch gradient descent work, which leads to redundancy, the time it takes is expensive, but SGD solves this problem because its gradient computation converges to only one computation in one era. And the instance that is selected for calculation is taken randomly from the dataset. And every time this process repeats itself. Due to the random selection of instances for training, the loss function is very strong and often fluctuates during training, which leads simultaneously to the choice of time for training and to the choice of initial parameters. The difference also lies in the fact that due to a random transition, emissions between local minima are very frequent. There is also a very important role is played by the learning rate, which allows you to navigate between the local minima. After all, when choosing a small learning rate, the transition can be very slow and on the contrary, will worsen the algorithm. And if you select too large, you can jump from the lows and again worsen the algorithm.[1609.04747][1801.03137][1801.09136][1803.02922]

Mini-batch gradient descent. This method is special in that it performs a gradient descent among a randomly selected sample, but also selects a certain range in the same sample. This improves the algorithm performance in terms of speed and performance, making it clear that the method works really well.[1801.03137][1801.09136][1803.02922] The formula is written as follows:

$$\theta = \theta - \eta \nabla_\theta E(\theta; x^{i:i+n}; y^{x_i:x_n}) \tag{3.4}$$

Momentum. Despite its performance, the storage system has a number of disadvantages, which also strongly affect the result and lead to slow growth, which further leaves the trained model in local lows. These transitions between lows and highs can be improved by adding a new value that will affect the gradient transition speed. A value, momentum, was taken to smooth the oscillations and control the speed. The effect of momentum we can notice below:

$$\begin{aligned} \upsilon_t &= \gamma\upsilon_{t-1} + \eta\nabla_\theta E(\theta) \\ \theta &= \theta - \upsilon_t \end{aligned} \tag{3.5}$$

The momentum assigned like γ, and value mostly is 0.9.Speaking in other words, our momentum is a pusher gradient descent.[1609.04747]

Nesterov Accelerated Gradient. This method is explained by the fact that it also allows us to increase the intensity of the change in our gradient descent rate. This is the calculation of the approximate next position of the parameter, which is of course approximated, but without completely updating the parameter, significantly increasing the speed of the algorithm. Since the function changes in the inclined, and allows you to increase the speed, as well as updating each individual parameter of the model[1609.04747].The formula is shown here:

$$\begin{aligned} \upsilon_t &= \gamma\upsilon_{t-1} + \eta\nabla_\theta E(\theta - \gamma\upsilon_{t-1}) \\ \theta &= \theta - \upsilon_t \end{aligned} \tag{3.6}$$

Adagrad. To improve methods that are based on gradient descent, we can adapt them to certain conditions. The first representative of such methods is Adagrad, which adapts its training by performing certain operations in the process. Under these operations, we mean the moments when the surround or a minor update of the gradient corresponding to its significance. Since not all data is perfect, it is usually very sparse and scattered. It has significantly increased the performance of stochastic gradient descent. While using the Adagrad method, we will not update all the parameters of Our o parameter matrix, and for each of them, we will find a separate learning rate, which will be changed at a certain time step t. First of all, we have to calculate the gradient of the target function in a certain period of time:

$$g_{t,i} = \nabla_{\theta_t} E(\theta_{t,i}) \tag{3.7}$$

SGD process update every parameter in our training parameters θ_i at each time step t, then our parameters updated wholly:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \times g_{t,i} \tag{3.8}$$

But Adagrad do something other. This process modify main learning rate η for each step at t time, here all parameters inside θ changed, based on the last calculated parameters:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \times g_{t,i} \tag{3.9}$$

So a visible advantage Adagrad's is automatically updated and customizable learning rate.But despite all the above advantages, there is a very big drawback of this method, these are positive gradient numbers, which in turn cause the main amount to grow, which causes a slowdown in the speed of learning and obtaining new data, which leads to poor learning with a large number of training.[1609.04747][1801.03137][1801.09136][1803.0292

The perfect form of Adagrad is Adadelta, which, instead of accumulating a huge number of gradients, cuts them off and gives a fixed number of exactly the same gradients. This size W is fixed and controlled.Instead of storing all the data that came out of the gradients, it is more efficient to use its mean $E[g^2]_t$ for all decreasing gradient squares, which gives only the mean and the gradient that was calculated at that point:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2 \tag{3.10}$$

Next, using momentum, we update the parameters using the update vector $\Delta\Theta_t$, which will be calculated by the gradient $g_{t,i}$ and learning rate η:

$$\begin{aligned} \Delta\theta_t &= -\eta g_{t,i} \\ \theta_{t+1} &= \theta_t + \Delta\theta_t \end{aligned} \tag{3.11}$$

Next one is updating parameters of Adagrad:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t \tag{3.12}$$

So, for improving results, we replace the gradient matrix with our mean matrix. And we can that our formula converts to other RMS function:

$$\begin{aligned} \Delta\theta_t &= -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \odot g_t \\ \Delta\theta_t &= -\frac{\eta}{RMS[g]_t} g_t \end{aligned} \tag{3.13}$$

Last thing about Adadelta, is that you don't need initial and static learning rate, because this algorithm do it for its own.[1609.04747][1801.03137][1801.09136][1803.02922]

3.6. Backpropagation

A multilayer neural network for training using the method of inverse spread, which in training practicing on perceptron neural network. The classical neural network is a compositional one, of which there are 3 layers. The input layer, hidden layer, and output layer are these 3 layers. In layers are neurons that build the Foundation for the work of the neural network. The importance, connection, and involvement of each neuron represent it, which in the training will change in a different range. At the very first iteration, weights are determined by random weights. After the values from the input neurons reach the most recent output layer, a loss function is calculated that shows how close or far you are away from the real value. After that, the loss function returns to the previously hidden layer, which updates its weight, after all, the process repeats many times to reach the minimum value of the loss function, as well as to improve the result of

prediction and guessing.To paint the back-propagation algorithm can be like this: First of all, you need to understand that the reverse propagation algorithm works in training and test mode. Since the algorithm is a representative of supervised learning algorithms, we already know labels during the learning process. Since the last layer allows you to calculate the loss functions, this data is returned back to the input layer, and then change the weight to one or the other side to achieve a certain result.First of all, you need to understand that the algorithm of reverse distribution works in training and test mode. Since the algorithm is a representative of supervised learning algorithms, we already know labels during the learning process. Since the last layer allows you to calculate the loss functions, this data is returned back to the input layer, and then change the weight to one or the other side to achieve a certain result. This is followed by the choice of the style in which the reverse distribution method will work. Before you start working with this method, you should choose a style of work, which is divided into an approach in which after each return value from the output layer immediately follows the update weights, as well as an approach in which under a certain condition and sequence to update all weights. The second approach wins in time because it does not require mathematical operations to update all weights.Also, in the reverse distribution process, you should understand that all hidden layers are input layers for the following layers, which allows them to be updated in their scales. The last layer, after it has received values last layers immediately calculates the value of the activation function, which is determined in advance.[back2]

This algorithm can stop only if the researcher considered that the loss function is small enough and the accuracy reached the desired measure. First of all, the data should go through all the layers, calculating the loss function at the end. Next, after that, you need to perform the reverse distribution operation for the output layer and for the hidden layer, then the output data updates the weights of all previous layers.[NeuralNetworks]

3.7. Neural Networks

One of the most powerful tools of today's intelligent machines is the neural network, which in the last few years has only gained a more powerful turnover in development. Since the penetration of neural networks is high, the area of their use is wide-range, as it can be used for regression, classification, and clustering. Initially, the neural network was invented to try to mathematically Paradize the brain and perform its most simple and primitive functions. But with the development of science, of the simple functions, it could perform even the most difficult. The initial task was not only to imitate the brain but also the ability to learn from examples. Under the best conditions with the data where they will snatch all cases, the neural network is capable of showing new trends and extensions in the data, which is very useful for intelligent machines.The simplest example of a neural network is a perceptron, which has 3 main layers that we passed

in advance. The main working and important component are the weights between the layers, which are influenced by the training of the neural network. Neural networks by type of training can be divided into two parts, this is when we know the answer interval, and when we do not know what will be the output. In short, regression/classification and clustering. Important components in the training of the neural network are the activation function, momentum, and pruning.Main object in neural network is neuron.**The Neuron** is a link that performs all the basic functions in the training of the neural network, using all the nodes in which there are weights for these neurons themselves. Here we can explain so that the stronger and stronger the relationship between neurons, the faster and more efficiently transmit data. This process is very similar to the work of the human brain.The architecture of a simple neural network can be explained very simply. It can be divided into three main layers. This is the input layer, followed by hidden layers, which can be a large number. The most recent layer, the output layer, is used to compute the activation function overall data that has passed the entire neural network. When learning back propagation, this loss function is used to update all weights.You can control the neural network by updating and installing new weights that show the importance of each neuron. Because of the unknown during training, weights at the beginning are always put out of random numbers. The learning process of neural network consists of straight and bra spread. First and foremost are calculated amounts were neurons, as well goes along the bias. The Apostle goes on with the present result. After that, the error is calculated, the error goes back, then the update was, and the process repeats depending on the need and the number of an era. Next, VA performs the function activation role, which is performed by the who so. The peculiarity of this process is that it can solve problems. That and makes NN very Mo weapons. To activate the function, we can use linear, threshold, polynomial and sigmoid functions. Also, the speed and power of the neural network can increase from the removal of unnecessary neurons and connections between them. This is called pruning.[Neural Networks]

3.7.1. Convolutional Neural Networks

Convolutional neural networks are a very powerful and productive tool for solving problems in deep and machine learning. Their mathematical content is a very important and practical is justified.Since convolution neural networks are primarily algorithms of supervised learning, you need to write a basic function that will build a hypothesis. To construct a function of the conjecture that will reduce the error function, we will consider in one regression classification formula. This we can do by designating our label one-dimensional size.We know that the prediction algorithm is based on a simple formula $Y = f(X)$, in which the function f will always be different depending on the approach and the type of solution. Our task is always to minimize the loss function $E(X)$, which will be the loss L itself, showing how far the predicted value has gone from the real value.Our problem is how to minimize the error that pulls out our function f consisting of a group

of all possible conjectures F:

$$\hat{f} = arg\min_{f \in F} E[L(Y, f(X))] \tag{3.14}$$

A function to represent all parameters in the model and data in cases that with the classifier that with the practical regression are the same. This representation of the parameters is carried out by changing and transforming all the parameters from the sample, and then this transformation occurs in the prediction function itself.[1605.09081] For the classifier and for the regressors, we can present this:

$$\begin{aligned} f(X) &= \langle \Phi(X), w \rangle - regression \\ f(X) &= sign(\langle \Phi(X), w \rangle) - classification \end{aligned} \tag{3.15}$$

The most powerful and accurate tool for recognition and to determine what is and what is happening in the image are actively used neural networks. They are specifically configured and made to work with images.For this reason, CNN forms its layer of three important values: width, height, and depth. The latter is used to work with color images.**Convolutional Layer**: These layers are used to address specific issues that MLPs often encounter or fully-connected NN. This solution combats redundancy and noise that classical computer vision techniques could not solve. The difference from other layers is that each neuron in a new layer, it's put together subsets of the previous layer. In short, we compress and weld the squares from the previous neuron, which solves the problem with a huge amount of data, and at the same time does it without losing all the data. The square is sprayed with height and width. Can be called differently as a receptive field that represents a subset of the pixels of the previous layer. The depth is also determined at the very beginning, as we decide to work only with black and white or RGB image. There is also a hyperparameter that corresponds to the number of convolutions we get at the output, it is called - stride. He controls the expense of gap between squares subsets. Mathematical it is clear that there will be fewer neurons at the output than at the input to regulate this process uses a method in which 0 is added to the boundaries. It's called padding. The number of neurons in the output we can calculate by taking away the number of neurons in the input number of receptive field, then adding padding to it, then divide it into stride, then adding to it only the number 1. **Pooling layer:**the layer which serves to unite as convolutional layers

only has the distinctive feature which helps to solve the main problems of NN. The process is similar to convolution, but the difference is that this layer is not parameterized. In this layer there is also no weights or rejection; instead, it simply combines all the neurons in a specific area. The most used method is to take the maximum number of the selected area, as well as methods in which to take the average number. Process on the idea of similar. [1803.02129]

3.7.2. Recurrent Neural Networks

AN that have a link to a recurrent function are called recurrent neural networks that are able to mathematically work with data that have a particular link and have a specific sequence. The essence of this neural network is that hidden layers save information from the past iteration, and work as memory inside the neural network. This structure allows RNNs to store, remember and process complex signals over long periods of time. RNNs can map the input sequence to the output sequence in the current time step and predict the sequence in the next time step.RNN, this as mentioned earlier, is a class of algorithms superior learning, in which a neural loop is connected, which repeats a certain order of time, and which has memory to work with a sequence. RNN is trained with a dataset where there is the concept of ”input - purpose”. Optimization comes at the expense of minimizing the difference between the real value and the forecast, but the work is in pairs, as well as these values are stored, and the size of the RNN can be significantly small from CNN. The neural network has 3 layers: the input layer, which contains all the original neurons. Next are the recurrent hidden layers, which store the past weight and changes. After all, this is the output layer. The entrance is n number of units. These data units look like a sequence of vectors at a certain time t as $..., x_{t-1}, x_t, x_{t+1}$, where each x_t is from t, $x_t = (x_1, x_2, ..., x_N)$ a group of all X units. Next, fully connected input signals are on hidden layers, where their relationship is defined by a matrix of weights W_{IH}. Next, the hidden layers create their M hidden units $h_t = (h_1, h_2, ..., h_M)$, which are all of them connected recursively. Hidden layers we can define as true neural network memory.Memory h_t in hidden layers is determined by the function's deduction formula from o_t, which in turn represents the linear sum of all layers:[1801.01078]

$$h_t = f_H(o_t) \tag{3.16}$$

$$o_t = W_{IH}x_t + W_{HH}h_{t-1} + b_h \tag{3.17}$$

Further hidden layers create a connection from the output layer through the weights in the W_{HO}. And the output layer has its own P number of units $y_t = (y_1, y_2, ..., y_P)$that are calculated by this formula:

$$y_t = f_O(W_{HO}h_t + b_o) \tag{3.18}$$

Where f_O is the activation function for solving nonlinearity and control numbers in RNN. RNN, in its simplicity, consists of nonlinear equations that simply repeat in a certain order of time. Then, in each interval, the loss function is predicted and deducted. Hidden layers perform a memory function that retains the weights and the rest of the desired and best parameters that were calculated in time series. The peculiarity here is that you can determine the future behavior of the network, and significantly improve the result. The simplicity of RN is also caused by the use of simple nonlinear functions, which create dynamics and good training on data in a certain interval.[1801.01078][1710.03414]

3.7.3. Capsules Neural Networks

Capsule neural networks have been a new approach to neural networks since October 17. Capsules in NN represent a group of neurons, where their vector, which is responsible for the activity, as well as this vector itself, represents the parameters of each instance of a particular type of entity, or in other words the pattern. This approach uses the length of the same activity vector to calculate and represent the probability that an instance pattern exists and to represent its parameters. Capsules that are active, calculate specific forecast parameters of an implementation of the capsules, whose levels are more higher. When this prediction occurs, and most of these predictions converge to the same result, the high-level capsule moves to the active capsule level. Usually, after convolutional networks used activation function, which outputs the data to enter in the capsule. The very first capsules serve by simple logic, as multidimensional transformable objects, which are very similar to reverse rendering. This process has not been applied before, which makes it very special.[1710.09829]

3.8. Summary

In this Chapter, we fully explained all the existing algorithms so briefly and clearly, but at the same time openly explained their algorithms and processes with maximum clarity. In the next Chapter, some methods which will be included in the basic work of the algorithm will be revealed. The following will show what kinds of algorithms we use to demonstrate the performance of SS and CNN.

CHAPTER 4

PROPOSED METHODS AND ALGORITHMS

4.1. Introduction

As mentioned earlier, methods for identifying and recognizing logos on the image is quite problematic. In this Chapter, we will tell you how we came to the solution of this problem. As the basis of all the work, I decided to repeat the steps and methods of work with researchers from the DISCo laboratory[1701.02620]. Just changing the segmentation search for image segmentation, I added some layers for CNN, taking as a basis one of the most popular CNN - AlexNet, but to solve the problem and train more different networks, we will use the same CNN based on VGG16 and will try to implement ResNet. This chapter will explain in detail what frameworks and pipeline we have created to solve this problem. Also will tell about the technologies and the main dataset, which will train our CNN.

4.2. Review of Pipeline

The problem with recognition is that we cannot know exactly where our logo will be located, which we want to recognize and find where it is. Before creating the algorithm itself, our task is to create a working dataset that will store the images and their target.The working pipeline, the first step is to get the input of the full image itself, then to carry out segmentation, filter segmented images, then CNN will hold the image through itself, in the end, there is a prediction that will give us a possible class of this logo on the image.To have the performance as high as possible in this pipeline, we use an object clause that is very review-oriented. For this reason, the CNN classifier must be created and trained to take into account that logo offered regions can contain many false positives or only part of the real logos. To solve these problems, we propose here a training structure and investigate the impact on the final efficiency of the recognition of different implementation options. **HERE WILL BE IMAGE OF MAIN PIPELINE**

4.2.1. Object Segmentation Method

Exhaustive search helps to find parts of the image where you want to consider the potential parts of the desired object.Although this model works well with specially selected objects, it has a number of drawbacks that significantly affect the detection of logos.After all, the search for every possible object has the ability to

be impossible. To solve this, we can use selective search.To improve the whole process and the data set for testing, we can use a combined method where we will use both methods described above. Since the number of possible objects will be more and less real and possible.Diversity in this task plays an important role, as we can cover more and more possible variants of this logo in the image.Since selective search is more useful to us, it will be helpful to familiarize yourself with its dependencies.The first and most important factor is to cover as many scales as possible because the logo can be small or large. We may not warn that. After all, the situation can be quite different. Also can make problems of objects which have no clear borders, for this reason, it is necessary to look through all options of the sizes of an object.Also, it should be noted that there is no exact and general solution of searches of any objects. It is impossible to make such a general detection system. Well, at the moment of course. For this reason, you should also look at the variety of objects and their contours, which can be very important in training. Speed is an important factor when searching for possible objects in an image. After all, such systems are built to determine the objects on the camera in a short period of time.This method is exclusive in that it is possible to configure this so that it worked by concentrating on the object and not on its borders.[ssForSegmentation]

4.2.2. Network architecture and training parameters

To create a convenient and correct working data loader, you need to collect all the available data, using their ready-made annotation you need to crop each photo with the object, or rather with the logo. After cropping logos, you need to prepare a pair of ”logo-class”. Since not all logos are perfect squares, many images will have a background that will not be relevant to the logo itself. In most cases, the logos are more perpendicular.This process achieved with ground-truth of logos in images.Methods for solving these problems will be proposed to improve efficiency and achieve a good result.The first step is to try to increase subsidies by creating and generating new data that will simply be created from already having data. Generation is carried out in such a way that when creating a logo object to look from different angles and perspectives, and increasing the size of the data.To bring all images to a more comfortable and uniform color scheme, in which there will be no frequent deviations or excessive noise, you need to try to normalize the contrast and the object itself.Also, a very important idea is to create a class that would mean no logo and would mean the background itself. Due to this, CNN will know about the background and learn how to more accurately and efficiently determine the logos themselves.In the process of testing CNN, a simple image comes to the entrance, where presumably there is a logo of a company. Then the image should be segmented to possibly find any existing objects. The further task is to drive all available segments on CNN. After that, the neural network will give us an answer from its classifier.

Table 4.1. CNN Architecture

N	Layers
1	Conv 64 filters of 11×11
2	ReLU activation function
3	MaxPooling with stride = 2
4	Conv 192 filters of 5×5
5	ReLU activation function
6	MaxPooling with stride = 2
7	Conv 384 filters of 3×3
8	ReLU activation function
9	Conv 256 filter of 3×3
10	ReLU activation function
11	Conv 256 filter of 3×3
12	ReLU activation function
13	MaxPooling with stride 2
14	Dropout
15	Fully connected,size 4096
16	ReLU activation function
17	Dropout
18	Fully connected,size 4096
19	ReLU activation function
20	Fully connected,size number of classes
21	SoftMax function - classification

4.3. Review of Logos Dataset

Flickr Logo s - 32 dataset is one of the largest and most extensive datasets, which is still publicly available in turn. Classes in it enough for research in this area, their 32 class, also there are samples, where logo there and is used in as a simple background. The dataset is just as unique as it is very convenient to study recognition and definition with real images where logos can appear in different positions and orientations. This dataset is divided into training, validation, and test set.The resulting images have been manually checked to make sure that the particular logo is really shown. The entire dataset is divided into three disjoint subsets P_1, P_2, and P_3, each containing images of all 32 classes. The first section P_1-the training set consists of 10 images that have been selected in such a way that they consistently show a single logo under various views with a minimum amount of background clutter. The other two sections P_2 (test case) and P_3 (test case = query case) contain 30 images per class. Unlike P_1, these images contain at least one instance of the logo, but in several cases, multiple instances.

To facilitate the development of high-precision classifiers is important to assess their sensitivity to images without a logo. Therefore, both sections P_2 and P_3 include an additional 3,000 images downloaded from Flickr with ”building”, ”nature”, ”people” and ” friends” requests. These images are negative images and complement our dataset. A brief summary of the data is given in the table below.[1801.11417]

Table 4.2. Dataset partitions/Subsets

Partition	Description	Images	Numbers of Images
P_1 (training set)	Hand-picked images	10 per classes	320 images
P_2 (validation set)	At least a single logo in image	30 per classes	3960 images
	Non logo images	3000	
P_3 (test set)	At least a single logo in image	30 per classes	3960 images
	Non logo images	3000	
P_1, P_2 **and** P_3	Three of them disjoint	All classes	**8240 images**

4.4. Concept of Technologies and Frameworks

PyTorch is a python package that provides two high-level features:**Tensor computation (like numpy) with strong GPU acceleration** and **Deep Neural Networks built on a tape-based autodiff system**.You can reuse your favorite python packages such as numpy, scipy and Cython to extend PyTorch when needed.At a granular level, PyTorch is a library that consists of the following components:

Usually one uses PyTorch either as:

Table 4.3. PyTorch components

Component	Definition
torch	a Tensor library like NumPy, with strong GPU support
torch.autograd	a tape based automatic differentiation library that supports all differentiable Tensor operations in torch
torch.nn	a neural networks library deeply integrated with autograd designed for maximum flexibility
torch.optim	an optimization package to be used with torch.nn with standard optimization methods such as SGD, RMSProp, LBFGS, Adam etc.
torch.multiprocessing	python multiprocessing, but with magical memory sharing of torch Tensors across processes. Useful for data loading and hogwild training.
torch.utils	DataLoader, Trainer and other utility functions for convenience
torch.legacy	legacy code that has been ported over from torch for backward compatibility reasons

- A replacement for numpy to use the power of GPUs.
- A deep learning research platform that provides maximum flexibility and speed

PyTorch provides Tensors that can live either on the CPU or the GPU, and accelerate compute by a huge amount.**Dynamic Neural Networks: Tape based Autograd**. PyTorch has a unique way of building neural networks: using and replaying a tape recorder.Most frameworks such as TensorFlow, Theano, Caffe and CNTK have a static view of the world. One has to build a neural network, and reuse the same structure again and again. Changing the way the network behaves means that one has to start from scratch.With PyTorch, we use a technique called Reverse-mode auto-differentiation, which allows you to change the way your network behaves arbitrarily with zero lag or overhead. Our inspiration comes from several research papers on this topic, as well as current and past work such as autograd, autograd, Chainer, etc.While this technique is not unique to PyTorch, its one of the fastest implementations of it to date. You get the best of speed and flexibility for your crazy research.PyTorch is not a Python binding into a monolothic C++ framework. It is built to be deeply integrated into Python. You can use it naturally like you would use numpy / scipy / scikit-learn etc. You can write your new neural network layers in Python itself, using your favorite libraries and use packages such as Cython and Numba. Our goal is to not reinvent the wheel where appropriate.PyTorch is designed to be intuitive, linear in thought and easy to use. When you execute a line of code, it gets executed. There isnt an asynchronous view of the world. When you drop into a debugger,

or receive error messages and stack traces, understanding them is straight-forward. The stack-trace points to exactly where your code was defined. We hope you never spend hours debugging your code because of bad stack traces or asynchronous and opaque execution engines.PyTorch has minimal framework overhead. We integrate acceleration libraries such as Intel MKL and NVIDIA (CuDNN, NCCL) to maximize speed. At the core, its CPU and GPU Tensor and Neural Network backends (TH, THC, THNN, THCUNN) are written as independent libraries with a C99 API. They are mature and have been tested for years. Hence, PyTorch is quite fast whether you run small or large neural networks. The memory usage in PyTorch is extremely efficient compared to Torch or some of the alternatives. Weve written custom memory allocators for the GPU to make sure that your deep learning models are maximally memory efficient. This enables you to train bigger deep learning models than before.

4.5. Summary

In this Chapter, I fully told about the approach to solving this problem. The method that will be used for segmentation is explained, as well as further work with segmented parts of the image. It was told about the architecture of CNN, also described the technology to be used. In the next Chapter, you will see the results of all the work described in this Chapter.

CHAPTER 5

SIMULATION RESULTS

5.1. Introduction

In this Chapter, we will fully show the results and experiments of our work. Let's show how the dataset will be divided and used for training CNN, as well as show what factors affect the algorithm. Comparative results with other works related to the identification and recognition of logos will be shown. Also, we thoroughly compare with the article, which was taken as the basis of the work.

5.2. Logos Dataset Preparing

Before you start training and segmenting images, you need to prepare the dataset, which will be convenient to work with, as well as to lack options where the logos are not correctly or not fully distributed between eras and packs.Also very important will be the process to remove duplicates, so when training was not overfitting. It was also important to distribute the datasets into several bundles. Another important point is that when dividing the dataset into training, validation and test sets. It is also necessary to distribute all these data correctly so that in one particular iteration of the kidneys and eras there is no oversaturation of the same class. Also, it is important to consider when training for precision and recall. After all, accuracy in some cases can play with us in a very bad joke. Class balancing can sometimes solve this problem, but counting those metrics is also very important.

Table 5.1. FlickrLogos-32 content

Definition	Images
Total images	8240
Images containing logo instances	2240
Train+Validation annotations	1803
Average annotations for class (Train + Validation)	40
Total annotation	3405

5.3. Checking Image Segmentation and Object Proposal

In this section, we'll talk about the SS method, from the [selectivesearch] article, and how it worked for our data. Since the color gamut of all logos is very different, we tested this method on different color spaces, segment similarity measures, as well as the value that give a threshold for all images to be segmented in the future. Since it is very difficult to find the optimal size for the segmentation method, we have taken a static value that will fit the total size, in which the camera will shoot the area where the logo is supposed to be. This process is more efficient and convenient in terms of developing future applications that will use this method.

5.4. Experiments on Training CNN and Evaluating results

CNN training was conducted on all parameters and various variants of the training, which were described above. This is a balancing of classes via eras and iterations on the batches, then the normalization of all batches and adding a new class, in images which do not meet the logos. We explained in detail the benefits of each of these parameters, now show the results in all parameters. Also, these parameters increase the probability of finding the logo when tested on segmented images.Since each parameter, which affects the result in the measure in relation to each other is very different, for this reason, we have introduced a single value over time to fully assess the final result of the trained CNN. Data from the "FlickrLogos-32" dataset were used for full training, validation, and testing. To make it convenient to view all the results of the neural network training, the data obtained will be written in Table-5.2 . There will be shown our CNN with all its versatile parameters that have been described above. All indicators will also be shown to evaluate the algorithm performance. Since we know that accuracy does not always show a useful and effective result for us. In order to solve such a problem, we will show the loss functions, precision top 1 and top 5. Just beyond this the show accuracy. You will also see the results of other works to see the results relative to others.

Table 5.2. Results of final classification

Config.	BG class	WB	CB	BN	Prec. TOP1	Prec. TOP5	Best Loss	Acc.
I	No	No	No	No	0.320	0.350	4.525	0.083
II	Yes	No	No	No	0.632	0.685	2.522	0.657
III	Yes	Yes	No	No	0.845	0.886	0.881	0.896
IV	Yes	Yes	Yes	No	0.921	0.947	0.035	0.905
V	Yes	Yes	Yes	Yes	0.935	0.975	0.009	0.968

From the results shown in Table 5.2, we can clearly see that each parameter adds more efficiency in object

Table 5.3. Legend to Table 5.2

Name of config.	Description
BG	Background class
WB	With background logo
CB	Class balancing in epoch
BN	Batch normalization

recognition and neural network training. The table can also consider that even with good accuracy other results may not be very satisfactory. The results from Table 5.2 can be interpreted as follows :

- In configuration I We can see that CNN can not properly and effectively recognize the logo we need because its precision is below the middle. The loss function is also high and inefficient for our work.
- When you add a class where logos are not present(configuration II), it is possible to significantly consider what precision is much increase in its value by a third. The loss function also changed and decreased. But even so, this variant of neural network training is not effective for us.
- Looking through the rest of the configuration, we can consider a good result for all indicators, but the latest configuration of V showed the best and the desired result for us. This result shows that with proper class balancing, normalizing batches, adding logo backgrounds, and adding backgrounds as classes helped to achieve a better result.

Table 5.4. Comparison of the best trained our CNN with other methods in logo recognition

Methods	Train Data	Precision	Accuracy
BoW SIFT	FL32	**0.991**	0.941
Romberg	FL32	0.981	0.752
Bianco	FL32	0.909	0.876
DeepLogo	FL32	N/A	0.896
Deep Learning	FL32	0.968	0.917
Ours(config. V)	FL32	0.975	**0.968**

5.5. Application Creating

In order to visualize the results and demonstrate how the algorithm works, we will create an application that will take the image to the input, then segment it, and then it will hold on CNN and make the classification of

the image. There are options to create a bot on Telegram or raise a full web application that will implement Pipeline, which was described above.

5.6. Summary

After conducting the results on ”FlickrLogos-32” with different configurations and testing on segmented images, we clearly saw the performance of CNN on object recognition. This NN was able to solve the localization problem because in most cases we saw that it would be able to find the logo wherever it was.The main thing is that it is properly segmented by the SS algorithm.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1. Conclusion

As said in Introduction, logo recognition is a very important and widely applicable tool in artificial intelligence applications and business projects. After all, with this tool you can solve very big problems in the analysis of the popularity of brands, the popularity of advertising, as well as it will be possible to preserve their intellectual property, avoiding plagiarism. But the problem still remains, because we just can not know where the logo may appear, in what format or position it will be. This problem remains unresolved. Classical and traditional methods in most cases cannot cope with the problems of pattern and object recognition. But CNN in most cases solves many problems of localization and feature extraction. In this thesis we have improved the existing method, using different parameters for segmentation, as well as using a different architecture of the neural network. Also, CNN learning time is also an important part of the study, because it is difficult to know when the loss function will reach the point of the global minimum. Our method showed good results in training on marked data. Also, the results of the study were shown in testing on logos, which were segmented.

REFERENCES

[1] Deniz, P. S. R., Adaptive Filtering Algorithms and Practical Implementation, 2008, Third Ed., LLC, NY, Springer.

[2] Haykin, S., Adaptive Filter Theory, 2002, Prentice Hall, Upper Saddle River, NJ.

[3] Hayes, M. H., Statistical Digital Signal Processing and Modeling, 1996, John Wiley & Sons. Inc., New York.

[4] Sondhi, M. M., The History of Echo Cancellation, IEEE Signal Processing Magazine, 2006, 23, 95-102.

[5] Gay, S. L., An Efficient Fast Converging Adaptive Filter for Network Echo Cancellation, Presented at the Thirty-Second Asilomar Conference on Signal, System and Amplifier, California, USA, November 1998, 394-398.

[6] Gilloire, A., Experiments with Sub-Band Acoustic Echo Cancellers for Teleconferencing, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP1987), April 1987, 2141-2144.

[7] Duttweiler, D. L., Proportionate Normalised Least Mean Square Adaptation in Echo Cancelers, IEEE Transactions on Speech and Audio Processing,2000, 5, 508-518.

[8] Romesburg, E. D., Echo Canceller for Non-Linear Circuits, U.S. Patent, 5, August 1998, 796-819.

[9] Benesty, J., Gansler, T., Morgan, D. R., Sondhi, M. M., Gay, S. L., Advances in Network and Acoustic Echo Cancellation. Berlin, Germany: Springer-Verlag, 2001. DOI: 10.1007/978-3-662-04437-7.

[10] Naylor, P. A., Cui, J., Brookes, M., Adaptive Algorithms for Sparse Echo Cancellation. Signal Processing, June 2006, 6, 1182-1192.

[11] Salman, M. S., Kukrer, O., Hocanin, A., Adaptive Filtering Fundamentals and Applications, 2011, LAP LMBERT, U.S.A.

[12] Sayed, A. H., Adaptive Filters, 2008, John Wiley, Hoboken, New Jersey, 163-167.

[13] Bellanger, M. G., Adaptive Digital Filters, 2001, Second Ed., Marcel Dekker, New York.

[14] Mathews, V. J., Zhenhua X., Stochastic Gradient Adaptive Filters with Gradient Adaptive Step-Sizes, International Conference on Acoustics, Speech, and Signal Processing (ICASSP1990), April 1990, 1385-1388.

[15] Widrow, B., Glover, J. R., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., Zeidler, J. R., Eugene Dong, Jr., Goodlin, R. C., Adaptive Noise Cancelling: Principles and Applications, Proceedings of the IEEE, December 1975, 1692-1716.

[16] Li, Y., Gu, Y., Tang, K., Parallel NLMS Filters with Stochastic Active Taps and Step-Sizes for Sparse System Identication, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2006), May 2006, 3.

[17] Pelekanakis, K., Chitre, M., Comparison of Sparse Adaptive Filters for Underwater Acoustic Channel Equalization/Estimation, IEEE International Conference on Communication Systems (ICCS2010), November 2010, 17, 395-399.

[18] Gui, G., Peng, W., Adachi, F., Improved Adaptive Sparse Channel Estimation Based on the Least Mean Square Algorithm, IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, April 2013, 3105-3109.

[19] Gay, S. L., Douglas, S. C., Normalized Natural Gradient Adaptive Filtering for Sparse and Non-Sparse Systems, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2002), Orlando, Florida, March 2002, 2, 1405-1408.

[20] Chen, Y., Gu, Y., Hero, A. O., Sparse LMS for System Identification, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2009), Taipei, Taiwan, April 2009, 3125-3128.

[21] Jin, J., Qing, Q., Yuantao, G., Robust Zero-Point Attraction LMS Algorithm on Near Sparse System Identification, IET Signal Processing, 2013, 3, 210-218.

[22] Gu, Y., Jin, J., Mei, S., l_0-Norm Constraint LMS for Sparse System Identification, IEEE Signal Processing Letters, 1985, 9, 774-777.

[23] Christina, B., Control of a Hands-Free Telephone Set, Signal Processing, ScienceDirect, 1997, 61, 131-143.

[24] Elko, G. W., Diethorn, E., Gansler, T., Room Impulse Response Variation Due to Thermal Fluctuation and its Impact on Acoustic Echo Cancellation, International Workshop on Acoustic Echo Noise Control (IWAENC2003), Kyoto, Japan, September 2003, 67-70.

[25] Peterson, P. M., Simulating the Response of Multiple Microphones to a Single Acoustic Source in a Reverberant Room, Journal Of the Acoustical Society of America, Nov. 1986, 5, 1527-1529.

[26] Donoho, D. L., Compressed Sensing, IEEE Transactions on Speech and Audio Processing, 2006, 4, 1289-1306.

[27] Widrow, B., Stearn, S. D., Adaptive Signal Processing, 1985, Printice Hall, New Jersey.

[28] Mandic, D. P., A Generalized Normalized Gradient Descent Algorithm, IEEE Signal Process Letters, February 2004, 11, 115-118.

[29] Cheng Y. F., Etter, D. M., Analysis of an Adaptive Technique for Modeling Sparse Systems, IEEE Transaction on Acoustics, Speech, and Signal Processing, February 1989, 2, 254-264.

[30] Candes, E. J., Wakin, M., An Introduction To Compressive Sampling, IEEE Signal Processing Magazine, March 2008, 2, 21-30.

[31] Salman, M. S., Sparse Leaky-LMS Algorithm for System Identification and its Convergence Analysis, International Journal of Adaptive Control and Signal Processing, 2008, DOI:10.1002/acs. 2428.

[32] Rey, V. L., Rey, H., Benesty, J., Tressens, S., A Family of Robust Algorithms Exploiting Sparsity in Adaptive Filters, IEEE Transactions on Audio, Speech, and Language Processing, May 2009, 4, 572-581.

[33] Etter, D. M., Identification of Sparse Impulse Response System Using an Adaptive Delay Filter, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP1985), April 1985, 10, 1169-1172.

[34] Khong, A. W. H., Xiang, L., Doroslovacki, M., Naylor, P. A., Frequency Domain Selective Tap Adaptive Algorithm for Sparse System Identification, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2008), Las Vegas, Nevada, March 2008, 229-232.

[35] Kawamuri, S., Hatori, M., A Tap Selection Algorithm for Adaptive Filters, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP1986), April 1986, 11, 2979-2982.

[36] Su, G., Jin, J., Gu, Y., Wang, J., Performance Analysis of l_0-Norm Constraint Least Mean Square Algorithm, IEEE Transactions on Signal Processing, 2011, 6, 2223-2235.

[37] Gu, Y., Jin, J., Mei, S., 2010. A Stochastic Gradient Approach on Compressive Sensing Signal Reconstruction Based on Adaptive Filtering Framework, IEEE Journal of Selected Topics in Signal Processing, 2011, 2, 409-420.

[38] Shi, K., Shi, P., Convergence Analysis of Sparse LMS Algorithms with l_1-norm Penalty Based on White Input Signal, Signal Image and Video Processing, Springer, May 2010, 12, 3289-3293.

[39] Slavakis, K., Kopsinis, Y., Theodoridis, S., Adaptive Algorithm for Sparse System Identification Using Projections onto Weighted l_1-Balls, IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP2012), March 2010, 3742-3745.

[40] Wu, F. Y., Tong, F., Non-Uniform Norm Constraint LMS Algorithm for Sparse System Identification, IEEE Communications Letters 2013, 2, 385-388.

[41] Martin, R. K., Sethares, W. A., Williamson, R. C., Johnson, C. R., Exploiting Sparsity in Adaptive Filters, IEEE Transactions on Signal Processing, August 2002, 8, 1883-1894.

[42] Harris, R. W., Chabries D. M., Bishop, F. A., A Variable Step (VS) Adaptive Filter Algorithm, IEEE Transactions on Acoustics, Speech and Signal Processing 1986, 2, 309-316.

[43] Kwong, R. H., Johnson, E. W., A Variable Step-Size LMS Algorithm, IEEE Transactions on Signal Processing,1992, 7, 1633-1642.

[44] Cui, J., Naylor, P. A. Brown, D. T., An Improved PNLMS Algorithm for Echo Cancellation in Packet-Switched Networks, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2004), Toulouse, France, 3, May 2004, 141-144.

[45] Deng, H., Doroslovacki, M., Improving Convergence of the PNLMS Algorithm for Sparse Impulse Response Identication, IEEE Signal Processing Letters,2005, 3, 181-184.

[46] Benesty, J., Morgan, D. R., Sondhi, M. M., A Better Understanding and an Improved Solution to the Specic Problems of Stereophonic Acoustic Echo Cancellation, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP1998), 2, 156-165.

[47] Salman, M. S., Jahromi, N., Hocanin, A., Kukrer, O., A Zero-Attracting Variable Step-Size LMS Algorithm for Sparse System Identification, IX International Symposium on Telecommunications (BIHTEL2012), Sarajevo, Bosnia and Herzegovina, October 2012, 1-4.

[48] Chartrand, R., Exact Reconstruction of Sparse Signals Via Non-Convex Minimization, IEEE Signal Processing Letters, 2007, 10, 707-710.

[49] Rao, B. D., Delgado, K. K., An Affine Scaling Methodology for Best Basis Selection, IEEE Transaction on Signal Processing, January 1999, 1, 187-200.

[50] Rao, B. D., Bongyong S., Adaptive Filtering Algorithms for Promoting Sparsity, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2003), June 2003, 6, 361-364.

[51] Aliyu, M. L., Alkassim, M. A., Salman, M.S., A p-Norm Variable Step-Size LMS Algorithm for Sparse System Identification, Signal Image and Video Processing, Springer, DOI: 10.1007/s11760-013-0610-7.

[52] Gwadabe T. R., Aliyu M. L., Alkassim, M.A., Salman M. S., Haddad H., A New Sparse Leaky LMS Type Algorithm. IEEE 22nd Signal Processing and Communications Applications Conference (SIU 2014), Trabzon, Turkey, April 2014.

[53] Loganathan P., Sparseness-Controlled Adaptive Algorithms for Supervised and Unsupervised System Identification, Ph.D. Thesis, 2011, Imperial College, London.

[54] Gui, G., Adachi, F., Improved Adaptive Sparse Channel Estimation Using Least Mean Square Algorithm, EURASIP Journal on Wireless Communications and Networking, March 2013, 1, 1-18.

[55] Kenney, J.F., Keeping, E.S., In Mathematics of Statistics, 1962, Third Ed. Van Nostrand, Princeton.

[56] Evans, J. B., Xue p., Liu, B., Analysis and Implementation of Variable Step-Size Adaptive Algorithms, IEEE Transaction on Signal Processing, 8, 2517-2535.

[57] Mader, A., Puder, H., Schmidt, G. U., Step-Size Control for Acoustic Echo Cancellation Filters-An Overview. Signal Processing, September 2000, 9, 1697-1719.

[58] Reddy, V. U., Shan, T. J., Kailath, T., Application of Modified Least-Squares Algorithms to Adaptive Echo Cancellation, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP1983), April 1983, 8, 53-56.

[59] Mayyas, K., Aboulnasr, T., Leaky LMS Algorithm: MSE Analysis for Gaussian Data, IEEE Transactions on Signal Processing, April 1997, 4, 927-934.

[60] Candes, E. J., Wakin, M., Boyd. S., Enhancing Sparsity By Reweighted l_1-Minimization, Journal of Fourier Analysis and Applications, October 2008, 14, 877-905.

[61] Claasen, T., Mecklenbrauker, W., Comparison of the Convergence of Two Algorithms for Adaptive FIR Digital Filters, IEEE Transactions on Acoustics, Speech and Signal Processing, 3, June 1981, 670-678.

[62] Hill, S.I., Williamson, R.C., Convergence of Exponentiated Gradient Algorithms, IEEE Transactions Signal Processing, Jun 2001, 6, 1208-1215.