ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

MONEY CLASSIFICATION FOR VISUALLY IMPAIRED PEOPLE

Graduation Project

Ira Shyti
150110903

Program: Computer Engineering
Department: Computer Sciences

Advisor: Prof. Dr. Gozde UNAL

June 2017

ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

MONEY CLASSIFICATION FOR VISUALLY IMPAIRED PEOPLE

Graduation Project

Ira Shyti
150110903

Program: Computer Engineering
Department: Computer Sciences

Advisor: Prof. Dr. Gozde UNAL

June 2017

Declaration of Authenticity

I declare that all material presented to Istanbul Technical University is my own work, or fully and specifically acknowledged wherever adapted from other sources. I understand that if at any time it is shown that I have significantly misrepresented material presented to Istanbul Technical University, any degree or credits awarded to me on the basis of that material may be revoked.

İstanbul, 07.06.2017

Ira Shyti

MONEY CLASSIFICATION FOR VISUALLY IMPAIRED PEOPLE

(SUMMARY)

Nowadays technology is being used to make the life of people easier and everyday new devices and applications are being developed to help us in our daily life. New researches are done every day in all different areas. A group of people that needs to use technology to make their everyday life activities easier to perform is the people with imparities.

Lots of applications exist and more are being developed to help this group of people. Our project is related to the people that have visual imparity. This project will help them with the identification of the money.

Since mobile phones and their cameras have become part of our lives, more applications are done that deal with image processing. This project is also related to this field, more precisely it deals with image classification. Given some sets of images, in our case those sets being types of Turkish Liras banknotes, the application will classify the label of the given image.

In this project, we will develop the backend of the image classification part of an application. In the future, both iOS and Android applications can be developed and make use from this project.

Firstly, we have developed the project using Scale Invarient Feature Transform and Bag of Words algorithm. After that we utilized another method in order to get a higher accuracy. We used Convolutional Neural Network (CNN). CNNs generally show significant improvement for image classification problems.In the project TensorFlow library is used to implement our CNN model. It is a high level library for deep neural networks.

We created two datasets. The first consists of about 2000 banknote images. The second dataset, which is the extended version of the first one, has more than 3000 images. The accuracy rate when used the first dataset is higher, but that is not realistic since the images used for classification in real life are more complex than the ones included in this dataset.

The banknotes classification part of the project is separated into two major subtasks being image training and image testing.

The accuracy results obtained using dataset one and dataset two with SIFT and BOW methods are almost 56% and 30% respectively. We optained a accuracy rate of 32% by using the CNN method on second daaset.

GÖRME ENGELLİ İNSANLAR İÇİN PARA SINIFLANDIRMA

(ÖZET)

Bugünlerde teknoloji insanların hayatını kolaylaştırıyor ve her gün günlük yaşamımıza yardımcı olacak yeni cihazlar ve uygulamalar geliştiriliyor. Her gün değişik alanlarda araştırmalar yapılıyor. Görme engelli insanlar da hayatlarını kolaylaştırabilecek böyle teknolojik cihazlar ve uygulamalara ihtiyaç duyuyorlar.

Bu insanlara yardım etmek için bir çok uygulama geliştirildi ve geliştirilmeye devam ediyor. Bu proje de görme engelli insanların hayatlarını kolaylaştırmayı amaç ediniyor. Bu uygulama görme engelli insanların para sınıflandırmalarında yardımcı olmayı hedefliyor.

Hayatımıza kamera ve mobil teknoloji girdiğinden beri görüntü işleme üzerine olan çalışmalar da geliştirildi. Bu uygulama da aynı alanda ve daha çok görüntü sınıflandırma olarak kategorilendirilebilir. Bu uygulama daha önceden verilen Türk Lirası test görsellerini baz alarak gösterilen banknotun para birimini algılıyor. Algıladığı banknotu daha önceden verilen test girdileriyle değerlendirip sınıflandırmayı yapıyor.

Bu projede uygulamanın görsel sınıflandırma yapan bölümününü geliştireceğiz. Bu geliştirme bittikten sonra herhangi bir mobil cihaza yazılan uygulama ile bulunduğu sunucudan kullanılabilecektir.

Görsel sınıflandırma yapmak için konvolüsyonel sinir ağları kullanılmıştır. Bu projede modeli oluşturmak için TensorFlow kütüphanesi kullanılmıştır. Bu, derin sinir ağları için yüksek seviye bir kütüphanedir. Proje aynı zamanda SIFT ve Bag of Words algoritmaları kullanarak da gerçekleştirildi. İkisinin de sonuçları raporda bulunmaktadır.

İki örnek veri seti tanımladım. İlki 2000'e yakın banknot görseli içeriyor. İkinci sette ise ilk setin genişletilmiş versiyonudur ve yaklaşık 3000 örnek görsel bulunmaktadır. İlk veri setinin tutarlılığı daha fazla çünkü gerçek hayattaki sınıflandırma görselleri veri setindeki görsellerden daha karmaşıktır.

Projenin banknot sınıflandırma kısmı görsel alıştırma ve görsel test olmak üzere iki ana alt parçaya ayrılmıştır. Görsel alıştırma, sınıflandırma için kullanılacak test verilerinin alınması için kullanılmıştır.

SIFT ve BOW metodlarını kullanarak iki veri setinde alınan tutarlılık yüzdeleri sırasıyla

%56 ve 30% dır. CNN metodu kullanarak ikinci veri setinde yakaladığımız tutarlılık oranı ise %32 dir.

Projenin amacı görme engelli insanların hem pratik hem de günlük hayatlarını kolaylaştırmak için parayı algılamalarını kolaylaştırabilecek bir sonuç üretmektir.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Introduction to the Subject

In this project, it is aimed to develop a software for classification of money for visually impaired people. As mentioned in the “Turkiye’deki engellilere ilişkin en detay bilgileri”, in 2013 there were approximately 213 thousand visually impaired people living in Turkey and they face diverse kinds of life challenges, one of which is related to money and finance [1]. They have difficulties in classifying the money so this project is expected to make this activity easier for them. They usually classify the money by their physical characteristics, using a method of folding the money that they put in their wallets and identify them by their shape, but the real struggle is identifying the money that other people give them. The application will be helpful on classifying Turkish lira banknotes, being 5 TL, 10 TL, 20 TL, 50 TL, 100 TL, 200 TL.

Nowadays, mobile phones can be easily used from blind people thanks to the new user interfaces, which have improved user-friendliness of the touch-screen devices for the blind users. These new devices can be controlled by touch gestures or voice. Android-based mobiles are equipped with TalkBack accessibility service and the phones that have an IOS operation system have Voice Over implemented on them. All of these devices have a digital camera and are more popular among users. The camera of the device will be used to take the image of the note that will be identified.
In order for the project to be useful for users of all kinds of mobile phones, it is decided that our project will run on a server and the mobile applications will work by connecting to this server.

It is expected that the application that will run on Android OS will be used more since Android device usage is much more than iOS device usage. When creating the applications that will perform the image classification, we should consider also the minimum system requirements in order to make this application useful for the most number of people.

The system of the application will be based on the server. The entire database and the code needed to make the application work properly will be stored on the server. It is decided to be this way to make it easy to develop Android and iOS applications that will use the same code. Another reason why we decided to place the project on server is because in order to run image classification projects high performance computers are needed. This way the classification computation time is decreased too. To make the applications work, the users will need Internet connection, but since nowadays Internet is accessible from anywhere it is not seen as an obstacle.

The project will perform real time processing for each captured frame of the bill since it is difficult for people with vision disabilities to capture high-resolution pictures. These people who will use it will just capture a picture of the banknote that they want to recognize.

As mentioned by Xu Liu in [2], some of these challenges are related to the limited processing power of the device and the complex background resulting in false outcome and the quality of the captured frame of the banknote because sometimes it may result to be blurry.

Since visually impaired people are not able to take pictures that will only show the banknotes, the images that they will capture and try to make classification will have other elements included. In order to solve this issue, there are images with different background and different amount of light in the dataset. In this way we make the training accurate for any case that will appear during the image classification.

In conclusion, the project aims to be functional and to help visually impaired people in Turkey with the identification of the banknotes. In the following sections of the report you will find more information about money classification, the algorithms used for making this project, the way that the software works and the results obtained from the tests that have been done.

1.2 Related Works

Similar projects have has been developed before in other countries. One example is implemented by Xu Liu, “Mobile Currency Reader for People with Visual Impairments”. This application aims to classify American dollar bills. It was presented to Bureau of Engraving and Printing as a solution for the classification of the money problem for blind people. In this example Viola and Jones` algorithm and Ada-boost machine learning techniques are used [2].

Another example of similar applications is the “LookTel Money Reader” which is an IOS application. It was launched in March 2011. This application does the classification of the money of different countries. Some of the banknotes it recognizes are US Dollar, Euro, British Pound, Canadian Dollar, and Australian Dollar. The application supports different languages also [3].

A mobile application for money recognition is also developed for Jordanian currency. In this application both paper notes and coins can be classified. For this project the developers have used SIFT algorithm to make the money classification [4].

To our knowledge, an equivalent application does not exist for Turkish banknotes. This is the goal of the current project. A second goal is to create an image database of Turkish bills of 5TL, 10 TL, 20 TL, 50 TL, 100 TL and 200 TL.

1.3. Report Summary

1. Definition Of The Project And Its Plan: A short definition of the project is written and the project plan is also shown. The project risks are presented in this part of the report also.

2. Theoretical Information: Theoretical information about image classification and the algorithms used for the project are explained. There is given theoretical information for each of the main algorithms that take part in the project.

3. Analysis And Modeling: Project requirements, criteria to mark the project successful, criteria that can make the project fail. There is shown the workflow of the application also with a chart.

4. Design, Verification and Test: The development of the banknote classification is explained in details. In this section you can find also the tests done on the application and their results.

5. Experimental Results: In experimental result section we evaluate the performance of our proposed method on the test data.

6. Results And Suggestion: Results of the project and what could have been done to improve is presented.

7. Resources: The sources used for this project.

2 DEFINITION OF THE PROJECT AND ITS PLAN

2.1 Project Scope

The project is related to image processing, more exactly with image classification. Given an image, the program has to decide the class it corresponds to. In our case, the classes are the types of Turkish banknotes, being 5, 10, 20, 50, 100 and 200 Lira.

The project is developed by using two different techniques of image classification.

The first one is by using Scale Invariant Feature Extraction commonly known as SIFT [16] and Bag of Words commonly known as BoW algorithm. Project is separated into two major parts, which are training and testing.

The second type of classification is done using convolutional neural networks with Tensorflow Library. For this project AlexNet architecture is used [10].After the classification is done, type of the banknote corresponding to the image will be given as an output by using trained CNN model.

The project is designed to work server based and to store the required database on the server. The Project is designed to recognize the banknotes independent of the amount of light on the environment, the scale of the banknote on the image and its orientation.

2.2 Project Plan

The tasks that are completed while doing the project are as listed below:

- Research on image processing
- Research on image classification
- Research on using OpenCV
- Analysis and comparisons of algorithms and methods
- Research on SIFT and BOW
- Development of project done with SIFT and BOW
- Testing for SIFT-BoW

- Research on deep learning and convolutional neural networks
- Research on VGG16 and AlexNet
- Development using AlexNet architecture
- Training
- Testing for CNN
- Documentation of work done

The activities listed are done on 4 terms. The first 3 terms are dedicated to researching and developing the project by using BoW and the last term's work is dedicated to CNN.

In the last 4 months we decided to develop the project using Deep Learning, believing that we will get better results for the image classification.

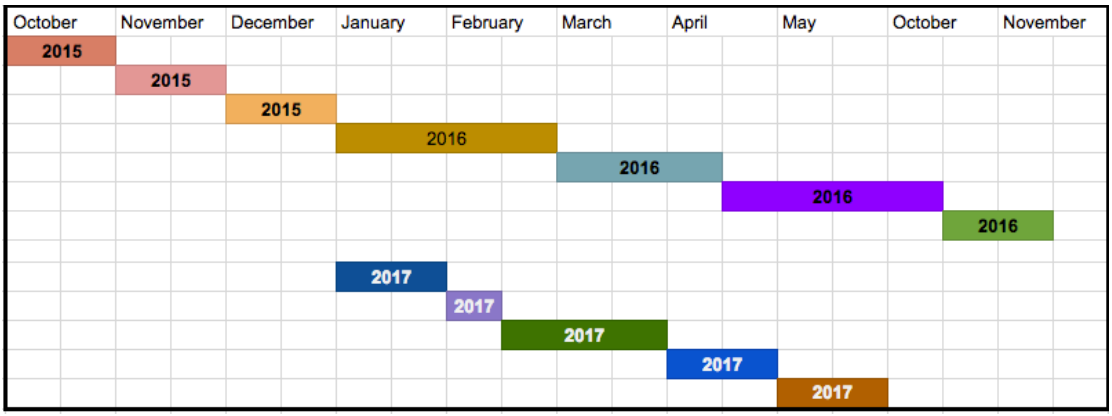

Figure 2.1: Gantt Chart of the project

The project is divided into 4 main parts, which are:

1. Understanding the topic and doing the research
2. Developing the project
3. Training &Testing
4. Writing down the documentation

2.3 Scheduling

More details for the duration of each task are given below:

Task	Start Date	End Date	Time Spent

Research on image processing	1 October 2015	1 November 2015	30 days
Research on image classification	1November 2015	1December 2015	30 days
Research on using OpenCV	1December 2015	1January 2016	30 days
Analysis and comparisons of algorithms and method	1February 2016	1April 2016	60 days
Research on SIFT and BOW	1April 2016	15May 2016	45 days
Development of project done with SIFT and BOW	15 May 2016	15 October 2016	60 days
Testing for SIFT-BoW	15 October 2016	15 November 2016	30 days
Research on deep learning and convolutional neural networks	1 January 2017	1 February 2017	30 days
Research on VGG16 and AlexNet	1 February 2017	15 February 2017	15 days
Development using AlexNet	15 February 2017	1 April 2017	45 days
Training	1 April 2017	1 May 2017	30 days
Testing	1 May 2017	1 June2017	30 days

Table 2.1: The scheduling of the tasks in the project

3 THEORETICAL INFORMATION

3.1 Image Classification

Image classification is one of the topics related with image processing. It deals with extraction of information from data sets. In the classification process, the image pixels are considered to be units, which are composed of the values from multiple bands. Image classification is the process that categorizes all the pixels of a digital image in a class. So, we can describe it basically as the tasks that takes an image as an input and the class that this image corresponds to or the probability of classes that best describe the image are given as an output. This project is based on money classification, which is one of the challenging computer vision problems.

The application has to identify corresponding class label from given input image. The project had to be divided into two major parts, which are model training and image classification. For deciding which algorithms were going to be used for the project, we have to consider lots of details. We had to pick the model that would give us the best accuracy in classification. The project should achieve successful results with the images that have different amount of light, different background and scale. Also our algorithm is expected to be position invariant. It also should work in the best way using the dataset that is created for it.

Firstly, we decided to collect money dataset since there is no known public dataset, which contains Turkish banknotes. The dataset had to be prepared. It contains approximately 3000 images from all the 6 types of banknotes of Turkish Lira. The dataset is needed for both of the training and testing parts of the application. The images are classified on 6 folders. We also use data augmentation in order to make the set of training data even larger. Data augmentation is the process that changes the array representation of the image while keeping its label same [5].

We have done the image classification using two different techniques for this project. In the first one, Bag of Words algorithm is used to make the classification and in the second one, the classification is done by making use of a relatively new topic, which is convolutional neural network.

While using convolutional neural network, firstly we tried to use the VGG16 architecture to make the image classification. We trained the model from scratch. Because of limitations in computer lab it was not possible to run the project using VGG16.

After that we decided to continue with the project by using AlexNet architecture. It is a less complex architecture than the VGG16 but is suitable to use for our project also. In the beginning we trained the model by initializing randomly and after that we used fine-tuning in order to compensate insufficient dataset. The architecture of VGG16 model consists of 13 convolutional layers and 3 fully connected layers. Five pooling layers are used in between the convolutional layers and after the last fully connected layer softmax is applied. We tried to apply this model using 5*5 filters and the images were resized to 224*224*3.

3.2 Convolutional Neural Network

Convolutional Neural Network, or referred to as CNN, has been one of the most important innovations in the field of computer vision. CNNs do take a biological inspiration from the visual cortex. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field [5]. We want the computer to be able to differentiate between all the images and figures out the unique features that make the given input part of a class. CNN is able perform image classification by looking for low level features such as edges and curves, and then building up to more abstract concepts through a series of convolutional layers. This is a general overview of what a CNN does. Explaining it in more details, the convolutional neural network takes an image as an input, passes it through different kinds of layers such as convolutional, pooling, nonlinear, fully connected, and gives as an output the class that the image belongs to. A great benefit of using CNN is that they have fewer parameters compared to fully connected networks and this makes it easier to train [7].

The first layer of a Convolutional Neural Network is usually a convolutional layer. The input of this layer is a pixel array, which dimensions are height and width of the image and the third dimension is the number of channels that the image has.

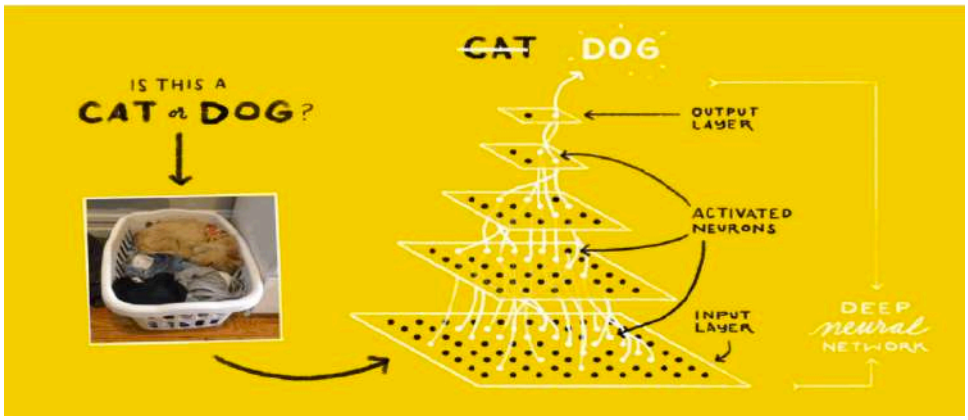

Fig 3.2: Image classification using CNN [8]

The CONV layer's parameters consist with a set of learnable filters. The size of the filters is small but they extend to the full depth of the volume of input. While sliding, or else known as convolving, this filter over the input image an activation map is created. The activation map is a two dimensional array that keeps the information of the filter at that position. The size of the output depends of three parameters, which are the depth, the stride and the padding. Depth corresponds to the number of filters we want to use. Stride is the number of pixels that we slide the filter through the image and lastly the zero-padding allows us to control the size of the output by adding zeros around the image. The correct formula for calculating the number of neurons is given below where N is the size of the input, F is the field size of neurons, P is the zero-padding and S is the stride applied. The result of it should be an integer.

$$N= (W – F + 2P)/S+1 \quad (1)$$

Pooling layer is used between convolutional layers and its main role is providing to translation invariance. Also it can be used to reduce parameter size, which is called sub-sampling. The size of filter in max pooling is usually 2*2 and stride equal to 2. Most often the MAX function is used in pooling layers, which mean it takes the highest parameter of the 4 parameters that are located in one filter. This layer gets an input of size W1 * H1 * D1 and the formula to calculate the dimensions of its output is shown below:

$$W2 = (W1 – F)/S + 1$$
$$H2 = (J1 – F)/S + 1$$
$$D2 = D1 \quad (2)$$

Another type of layer used in CNN is also activation layer. An activation function performs certain mathematical operations on a single number. The most common one to use is Rectified Linear Unit (ReLU). It takes the maximum between 0 and the input. ReLU accelerates the convergence of stochastic gradient descent.

$$F (x) = \max (0, x) \quad (3)$$

The last kind of layer used is the fully connected layer. The neurons in this layer are fully connected to all the neurons to the previous layer.

3.2.1 TensorFlow

TensorFlow is an open source software library released in 2015 by Google to make it easier for developers to design, build, and train deep learning models. It uses deep learning, which is a powerful part of artificial intelligence. It can run either in CPU or GPU. The underlying software of TensorFlow was build using C++ language but TensorFlow is implemented as a library in Python, which is one of the most common programming languages among the deep learning researchers [9]. For the project I will use this one.

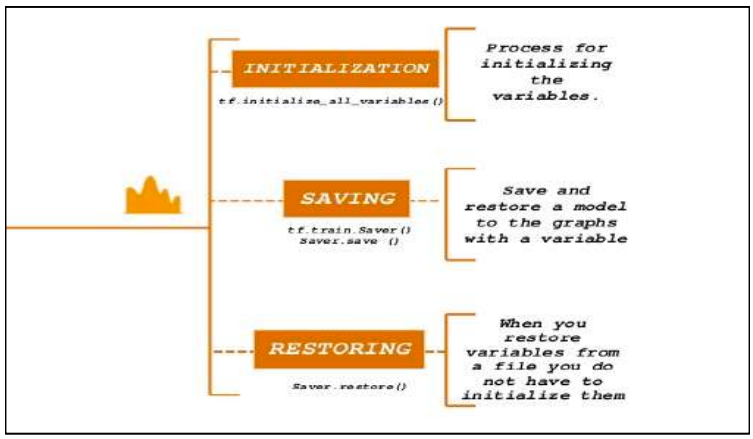

Fig 3.3: Main architecture of TensorFlow

The base unit of data in TensorFlow is a Tensor, which is an array of any dimensions that contains primitive values. The computations are represented as graphs that contain nodes. The nodes in the graph are called ops. It takes zero or more Tensors, makes computations and returns them. Graphs are launched in Sessions. TensorFlow sorts through these nodes in order to learn what is the image that it is given as an input. The first layer looks for basic elements of the image, determining general shape in picture. Then the system flows to the next data set looking for more specific elements in the photo.

3.2.2 AlexNet

Alex et al. create AlexNet network in 2012. It achieved a test error rate of 15.6 %, on ImageNet dataset, which was the best at that time [10]. The network was designed to make the classification among 1000 classes. Compared to modern architectures, the layout of this network is considered relatively simple.

The network contains eight learned layers from which five of them are convolutional layers and three of them are fully connected layers. The output of last fully connected layer is directed to a 1000-way softmax since the network was build to make classification among 1000 different classes.

The size of filters and the strides used in this network differ through the layers. There are three different layers used in total and their sizes are 11x11, 5x5 and 3x3. The strides used are respectively given as 4, 1 and 1. Response-normalization layers are used after the first and second convolutional layers. Max-pooling layers follow both response-normalization layers as well as the fifth convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer. The first convolutional layer takes as the input the images placed in the batch. The second convolutional layer takes the output of the first layer as an input after applying ReLU and maxpooling to it and uses 256 filters of size 5x5.The third, fourth, and fifth convolutional layers are connected to one another. There is no ReLU or maxpooling used between those layers. The third convolutional layer has 384 kernels of size 3×3×256 connected to the outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size 3×3×192, and the fifth convolutional layer has 256 kernels of size 3×3×192. The fully connected layers have 4096 neurons each.

The creators have trained their models using stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005. As the result, they claim that the small amount of weight decay was important for the model to learn [10].

The weights in each layer are initialized with standard deviation 0.01. The neuron biases in the second, fourth, and fifth convolutional layers, as well as in the fully-connected hidden layers are initialized with the constant 1. In the remaining layers the neuron biases are initialized with the constant 0. The learning rate is presented to be equal to 0.01.

Since deep convolutional neural networks with ReLU train much faster than the ones that lack it, it is considered as one of the most important feature of these networks. Learning in the neuron will happen if at least some training examples produce a positive input to a ReLU [10].

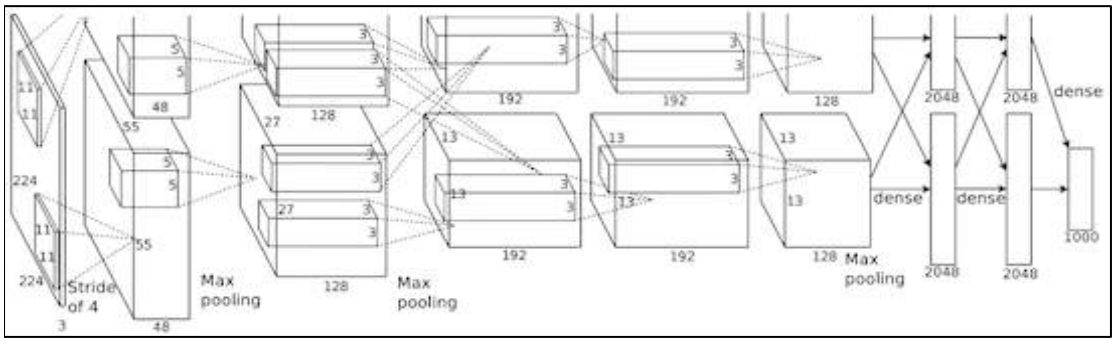

Fig 3.4: AlexNet Architecture [11]

An important layer in this architecture is also Pooling Layer. In AlexNet they use overlapping pooling, which means that the size of the filter used in pooling is greater than the stride, more precisely, the filter is of size 3x3 and the stride is equal to 2. In the paper they claim that this method reduces the error rate by 0.4 % on ImageNet dataset.

The dropout technique is used in the fully connected layers. This technique consists of setting to zero the output of each hidden neuron with probability 0.5. The neurons that are eliminated by the dropout in this way do not contribute to the forward pass and do not participate in back-propagation. Dropout is used in the first two fully connected layers and its aim is preventing the overfitting.

3.3 Money Classification using SIFT and BOW

3.3.1 OpenCV Library

Open Source Computer Vision Library or commonly known as OpenCV is an open source software library used for computer vision and machine learning. It was built for making computer vision applications easier to develop. There are approximately 2500 algorithm in the OpenCV library. Most of them are specialized for identifying objects, detecting and recognizing faces, tracking moving objects, finding similar images from an image database, etc. In this project we are going to work on the last mentioned type of algorithms. You can use it while developing with C++, C, JavaScript, Matlab and Python and it is compatible with Linux, Microsoft, Android and Mac OS operating systems [12].

3.3.2 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform is commonly known as SIFT. It was developed by David Lowe in 1999 for view-based object recognition of images [16]. It is known for successful application for image matching under real world conditions. What makes it a very useful algorithm is the fact that it is invariant to the illumination of the image and to the rotation and scale of the image. SIFT is designed for grayscale images mainly but it can be used for BGR images since colors contain significant information. Keypoints are interest points that correspond to some elements in the image that are detectable from different views. SIFT makes use of differences-of-Gaussians to obtain those interest points. A keypoint has four parameters whose being the x and y coordinates of its center, the orientation and its scale. Taylor series expansion of scale space is used to get more accurate location of extrema. The keypoint is rejected if the intensity is less than a given threshold.

An image descriptor is used for each keypoint in the image. A descriptor is a vector that describes the surroundings of a feature point. It can also be seen as a histogram for the gradient directions of location surrounding the interest point. Each image descriptor has 128 directions [13].

SIFT functionalities are available in OpenCV. We can use the function for finding keypoints and drawing them, for calculating descriptors and for matching keypoint in images.

3.3.3 Bag of visual Words (BOW)

Bag of visual words is one of the image classification method used in computer vision applications. It treats the image as a document and its features as words contained by this document. When we want to use BoW for images, they are treated as documents also. A histogram of image keypoint occurrence is calculated to compose feature vector. BoW model is largely unaffected by the orientation and the position of the object on image. Feature detection and representation can be done with it.

BoW usually is generated into two steps. Firstly we have to get the set of bag of features and the second step has to do with clustering those features into the created bags and calculating the histograms. Those histograms are the ones used for making the image classification [14]. One of the disadvantages of Bow is that they have a poor performance for localizing the objects within an image.

3.3.4 K Means Clustering

This is an iterative algorithm. It aims to partition the observations into a number of clusters. The input of the algorithm is a vector and the aim of the algorithm is to cluster each point of this vector to K sets. It is done to minimize the sum of distance of each point to the center of the cluster for each cluster.

$$\arg\min \sum_{i=1}^{k} \sum_{x \in S} \|x - \mu_i\|^2 \quad (4)$$

K-mean clustering aims to partition a given N set of observations where each of them is a d-dimensional real vector, into K classes so it minimizes the sum of squares within a cluster as shown in formula 4.

An example of clustering is shown in image 3.5.

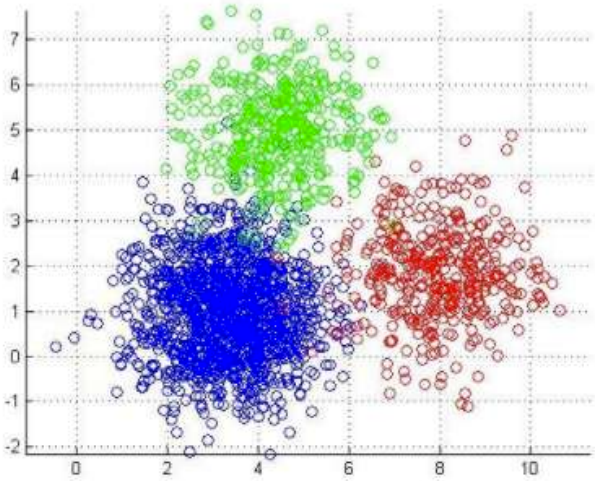

Fig 3.5: K-Mean Clustering example [15]

4 ANALYSIS AND MODELLING

Before starting to develop the project it is important to analyze how it works and what is needed to be done making it easy to use for the targeted audience, which in our case are the visually impaired people. While developing the backend, we should keep in mind that in the real world conditions, the images have different background scenes and quality.

The requirements of the project, the criteria that make the project successful and reasons why the application may fail to operate successfully are listed in the following section below.

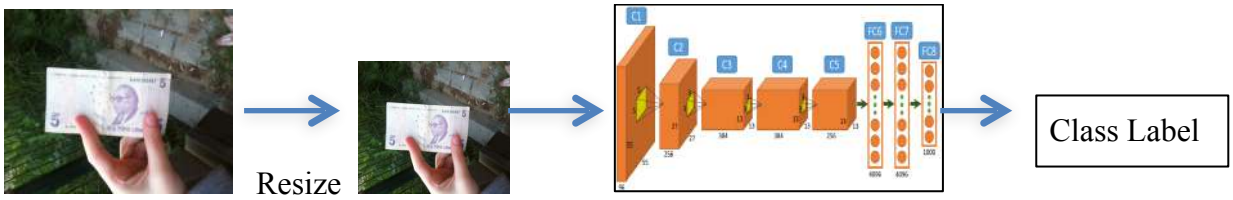

Fig 4.1: Image classification with AlexNet

4.1 Dataset

Collecting the dataset is a very important part of the development. There was no ready dataset related to the Turkish Lira banknotes so the dataset used for this project is created by me. The dataset of this project contains pictures of the Turkish banknotes which are 5 TL, 10 TL, 20 TL, 50 TL, 100 TL an 200 TL. It contains between 500 and 550 pictures from each of those banknotes. Dataset is not large enough to train a model that performs high accuracies.

In the beginning, a smaller dataset was used. The first dataset contained approximately 2000 pictures. This dataset included simple scenarios with homogeneous backgrounds and less occlusion. For instance, images in the first and second row of Fig. 4.2 show easy classify banknote pictures. The second dataset is creating by extending the first one with images in more difficult conditions such as complex backgrounds, occlusions and severe geometric transformations. For instance see images in rows 3, 4 and 5 in Fig. 4.2.

While taking the pictures of the dataset there were some details that should be considered.

The pictures are taken with a camera, which does not have a very good quality.
Because it is supposed that it should work with pictures of any quality and if we trained and tested the project with high quality pictures we might not have good results when using low quality images.

Another issue is that the banknotes should be photographed from different positions. Firstly they should be photographed back and front with different angles of view. In the dataset there are also pictures of folded banknotes since this is a possible way of showing the money to the camera.

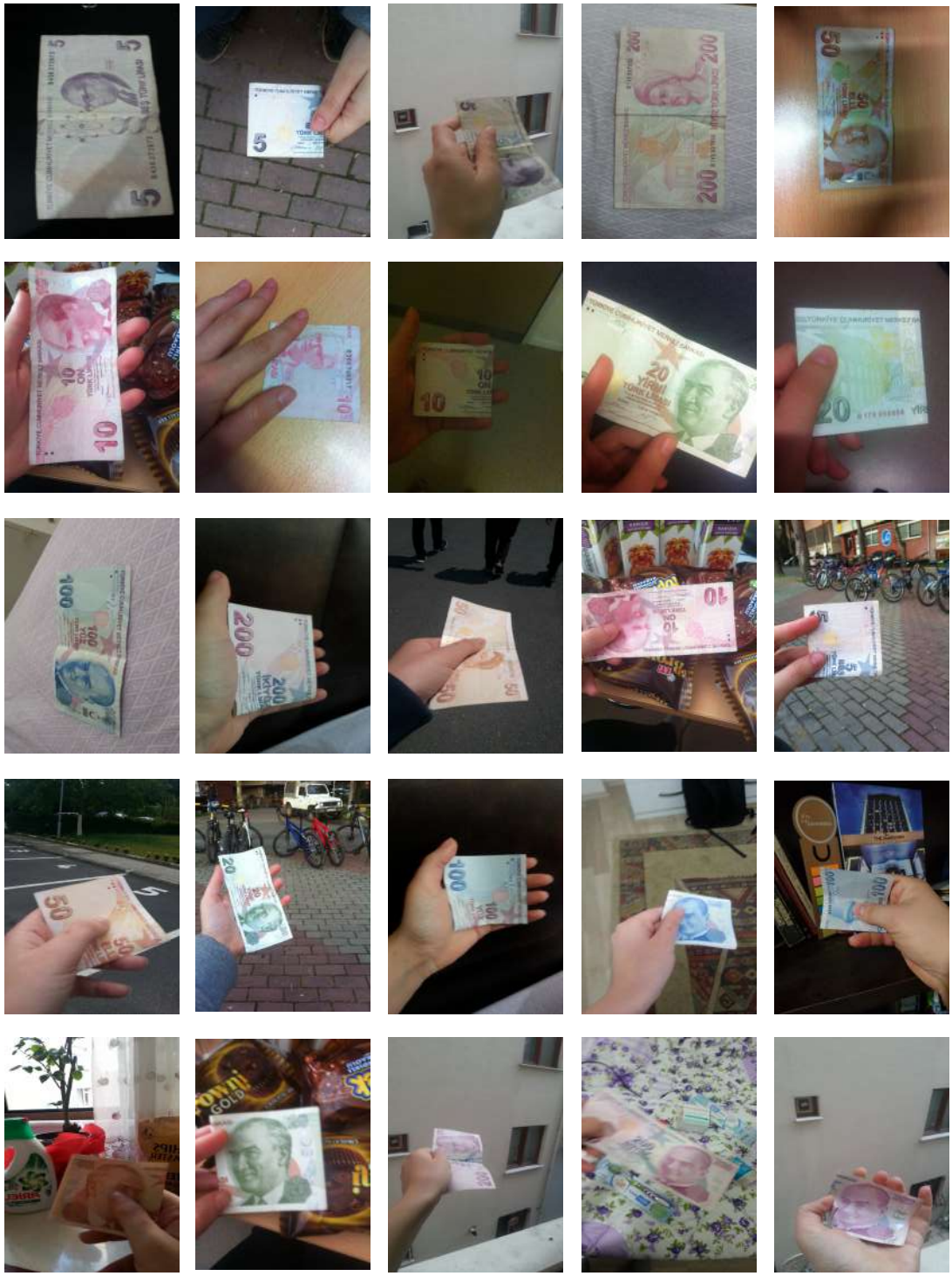

Fig 4.2: Examples of dataset images

Pictures of the dataset are also taken with different background based on the scenarios that may occur during the usage of the application. The scenarios are described in the paragraph below. Lighting condition is another substantial subject to be considered. It is a very important factor on the picture. Since visually impaired people cannot check the amount of light in the environment, the pictures are taken on places with various amount of light. In the dataset there are also some samples of data that show very clean images of banknotes are placed on a table and clean images are taken. A very few amount of pictures is taken showing blurry images of the banknotes. Images are blurry enough to classify them between each other. Figure 4.2shows some pictures from each class of image in the dataset.

When creating the dataset some cases had to be thought. The most common situation that the visually impaired people will use this application is while they are in a market, so while taking the pictures I created a background that looks like a market place. The pictures with this background were taken with different amount of light in the background. During this scenario, there are also taken pictures that are a bit blurry since the target audience for this project cannot check the quality of the image. Another scenario is that the users would try to classify the money while they are in an outdoor environment, having a walk or sitting in a park. To imitate that clean pictures are captured by putting the money on a desk and having no objects on the surrounding.

2. Criteria That Makes the Project Successful

1. The application must recognize Turkish Lira banknotes

2. The banknote recognition should be done within five seconds.

3. Criteria That Makes the Project Fail

1.The classification of banknote fails

1. Blurry image.
2. Illumination variation.
3. Partial occlusion.
4. Limited training data.

5 DESIGN, IMPLEMENTATION AND TESTING

5.1 Image Classification using SIFT and BOW

The image classification is divided into two sub-parts, which are respectively the image training and the image testing. OpenCV is used for developing the backend part of the project. Following sections are going to describe these parts in detail.

5.1.1 Training of Bag of Visual Words

This is the first part of the image classification. After the dataset is created the code development process started.

First of all we extract the keypoints from all the images in the dataset. Keypoints, or as they are commonly called the features, are some distinctive point in the image that correspond to a part of the image which can be located in different views. The maximum number of features that can be found in an image is decided to be 300. This number should be big enough so it covers most important parts of the image. There are different algorithms that can be used for detecting the features and one of them is SIFT algorithm. We use the SIFT algorithm that is implemented as a library on OpenCV. SIFT feature can be extracted from RGB images or grayscale images. Since in every banknote there are a lot of similarities, considering the part of the banknote that the picture of Mustafa Kemal Ataturk is shown, and also considering the fact that during the image testing only this part might be shown in camera, the colors of the image contain valuable information about the banknote and can help making the classification easier and more accurate. Based on those reasons feature detection is done on RGB colored images. Figure 5.2 show the sample outputs obtained from the SIFT algorithm. After the features are extracted, the descriptors of those keypoints are computed. As mentioned before, a descriptor is a vector that contains information about the surroundings of the detected feature. In total 916367 features were extracted from the dataset. Feature extraction and descriptor computation took nearly 703 seconds. The code used to compute those parts is shown in figure 5.1.

```
cv::SiftFeatureDetector detector( MAXIMAL_KEYPOINTS );
cv::SiftDescriptorExtractor extract;
detector.detect(image, keypoints);
extract.compute(image, keypoints,descriptor);
Mat output;
drawKeypoints(img, keypoints, output);
imwrite("sift_result.jpg", output);
```

Fig 5.1: SIFT algorithm for keypoints detection and descriptor computation

After this part the vector quantization of the features has to be done. For performing this task the K-Mean clustering algorithm is used. The right size of vocabulary had to be selected so it has a good accuracy and a fast performance time. In this case the vocabulary size is selected as twenty thousand. The function returns the cluster centers. This is the function that takes the most time to be completed. For all the images that are included in the dataset, when the code is computed in a computer with characteristics processor 2.5 GHz and Ram 16 GB it took about 18 hours to run this function.

Fig 5.2: SIFT results. The first and second examples show that SIFT has not detected features from our area of interest. The third and fourth images show more successful results of feature extraction.

In the next step the histograms are computed using the Bag of Words algorithm. Histograms are graphical representation of containing information about the contrast of image. It plots the number of pixels for each different tonal value.

5.1.2 Testing

For performing the image testing on the project, we obtain a histogram of visual words for the test image. Normalized scalar product between the query vector and the weighted histograms created in image training is used to perform the image retrieval. Ten images with the highest rank are chosen. We check whether the keypoints in the test image are in spatial consistency with the retrieved images. The type of the banknote, which has the most number of occurrences in the top ten selected images, is selected as the class where the test note corresponds.

5.1.3 Performance test

Pictures used for testing the success rate of the project are selected to be with different backgrounds, different amount of light in the image and different parts of the banknote are shown on them.

By selecting the vocabulary size of the clustering as 20000 we got the best accuracy without increasing the classification time too much. I tested the project with the value of vocabulary equal to 5000 and 10000 also.

Another parameter that affected the success rate was the number of image selected as best match images. I tested the project by setting this parameter equal to 20, 10 and 5. There was no accuracy difference between selecting this parameter as 20 and 10 but there was a slight difference in computational time. By selecting the best match size as 5 the success rate decreases by almost 20%. So based on those results the number of best matched images was selected as 10.

The average time to make the classification of an image is 0.4 seconds when the project is run on a computer with CPU equal to 2.5 GHz and ram equal to 16 GB.

5.2 Image Classification using Convolutional Neural Network

5.2.1 Inputs and Outputs

Images that are used as an input of network are of the JPG, jpg and jpeg format. It is important that the size of all images must be the same. For making use of AlexNet all the images are resized to 227 x 227. Since our dataset is small and in order to avoid over fitting data augmentation is used. In data augmentation, random flipping of the image, affecting brightness and contrast is done.

After running the project many times in order to get the best results out of it by changing some hyper parameters, the batch size in which the project gave better results was sixteen. So for this project the batch size is decided to be equal to sixteen.

For the output, the network calculates accuracy of predictions with based on the largest softmax value. It also shows to which class the image corresponds.

5.2.2 Training

We used the AlexNet network on this project. There are eight layers in the network, five of them are convolutional layers and the last three layers are fully-connected. The non-linearity ReLU is applied to the output of every layer.

The filter size of the first layer is 11x11 and the stride is 4. It uses 96 kernels. After this layer, non-linearity local response normalization and maxpool layers are used. For all normalization layers, the parameters used are the same and they are as: depth radius equal to 2, alpha equal to 2e-05, beta as 0.75 and bias as 1.0. All pooling layers

use also the same filter size and amount of stride. The filter size for this layer is 3x3 and the stride is given as 1.

```
def conv(x, size, n_filter, stride=(1, 1), padding='SAME',
         name="conv",
         bias=True,
         initializer_w=tf.random_normal_initializer(0., 0.01),
         initializer_b=tf.constant_initializer(0.)):
    with tf.variable_scope(name):
        w = tf.get_variable('w',
                            [size[0], size[1], x.get_shape().as_list()[-1], n_filter],
                            initializer=initializer_w, dtype=tf.float32)
        res = tf.nn.conv2d(x, w, [1, stride[0], stride[1], 1], padding=padding)
        if bias:
            b = tf.get_variable('b', [n_filter], initializer=initializer_b, dtype=tf.float32)
            res = tf.nn.bias_add(res, b)

        return res
```

Fig 5.3: Convolutional layer implementation in Tensorflow

The second convolutional layer uses a filter of size 5x5 with stride amount equal to 1. It uses 384 kernels. The last three convolutional layers use the same filter type that has size of 3x3 and the stride is also 1 for them. They only differ from the number of kernels they use. Layer 3 and 4 use 384 kernels and convolutional layer 5 uses 256 kernels.

Again, while making the test for getting out the best accuracy of the project, we noticed that by changing the number of nodes to 2048 the results were better. Since the project has to make classification between 6 different classes, the parameter of the last fully connected layer is set to 6. Also I changed the random normal initializer to 0.005 for the first and third layer and constant initializer to 0.01 for all the fully connected layers.

The accuracy depends also on the learning rate so we tested the project using different training rates and the best results were seen when it was equal to 0.0001.

Based on the researches done, using transfer learning helps the network to make a better training. Training from scratch does not show very good results when we do not have sufficient labeled data for the task to train a reliable model. Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For this project we imported the model of another trained dataset.

5.2.3 Testing

For making the testing of the accuracy of our network we used the data that were on the test folder. Those images were captured similar to the images used for training with different backgrounds, light and different position of banknote. The results we

got were not very satisfactory and we came to the conclusion that this was a result of the few images that were in the dataset and the complexity of those images.

Main reason of the poor performance is that the number of the images in the dataset is not enough to train such a complex and highly varying classification model.

6 EXPERIMENTAL RESULTS

In the beginning, the image classification method is implemented by using SIFT and Bag of Words algorithm. Firstly we did the training of images with a smaller dataset that included very clear images with simple backgrounds. After using the new created dataset, the accuracy results decreased. This means that the results we first got were not realistic. There are a lot of cases where the banknote part of the images was not selected as an important part of the images by SIFT and very few features were taken from this part. These cases decrease the accuracy rate of the project. The success rate that is achieved by using SIFT and BoW in our project is calculated as 30%.

We were expecting to get better results by developing the project using convolutional neural networks. Unfortunately, even if the results that we got were slightly better comparing to Bag of Words, we could not obtain the significant improvement. The accuracy result from using AlexNet was 32%.

You can see the results obtained by using the mentioned methods in Table 2.

Method	Accuracy
AlexNet (from scratch)	25%
AlexNet (pre trained model)	**32%**
Bag of Visual Words	30%
Bag of Words using dataset one	**56%**

Table 2: Performance comparison of different methods

7 CONCLUSION AND DISCUSSION

The best result that we obtained were 55% accuracy rate for dataset one and 30% for dataset to when using SIFT and BOW method. The best accuracy result obtained by using CNN on dataset two is 32%. The decrease in performance for the Dataset 2 is due to the wide variation and difficulty of images in the second dataset. The Dataset 2 is created keeping in mind real life scenarios, where visually impaired people will take pictures in uncontrolled environments.

In order to increase this rate and make the project more successful we are planning to increase the number of images in the dataset. From the results we have got, we came to the conclusion that the number of images in the dataset and their complexity affects the success rate of the image classification. We are expecting to get better results by enlarging the dataset and training over a larger number of images in different environments.

We hope that this project will be base for mobile applications so that the visually impaired people in Turkey can make use of it to make the everyday task of banknote recognition easier for them.

The first Turkish money dataset is introduced according to the best of our knowledge. We are planning to expand the dataset and make it public.

Accuracy of BoW - SIFT is dropped significantly when the new data is added to the dataset. This happened because the images added to the dataset have more complex backgrounds. If we had just presented the results taken by using the first dataset, the computed accuracy would be higher, however, the system would fail in a real life application.

8 REFERENCES

[1] “Turkiye’deki engellilere iliskin en detay bilgiler”, SGK Rehberi, October 2013

[2] X.Liu, “Mobile Currency Reader for People with Visual Impairments”.Retrieved from: http://src.acm.org/liu/liu.html

[3] “LookTel Money Reader”. Retrieved from: http://www.looktel.com/moneyreader

[4] I. Abu Doush, S. Al-Btoush, “Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms”, July 2016

[5]A.Deshpande, “A Beginner's Guide To Understanding Convolutional Neural Networks “, June 2016

[6] D.Frossard, “VGG16 in Tensorflow”, June 2016

[7]“Convolutional Neural Networks”. Retrieved from: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork

[8] J.Dean, “Jeff Dean On Large-Scale Deep Learning At Google”, March 2016

[9] C.Metz, “Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine”, Septemder 2015

[10] A. Krizhevsy, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, 2012

[11]Retrieved from:https://www.researchgate.net/figure/305378666_fig1_Figure-1-AlexNet-architecture-17

[12] “About OpenCV”. Retrieved from: http://opencv.org/about.html

[13] T. Lindeberg, “Scale Invariant Feature Transform”, 2012

[14] R. Bandara, “Bag-of-Features Descriptor on SIFT Features with OpenCV”, April 2014

[15] M. Chen, “Kmeans Clustering”, March 2017. Retrieved from:https://www.mathworks.com/matlabcentral/fileexchange/24616-kmeans-clustering?requestedDomain=www.mathworks.com

[16] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features”, 1999